

Universidade Federal do Rio de Janeiro

Núcleo de Computação Eletrônica

Máximo Eduardo Egger

**CORRELAÇÃO DE EVENTOS PARA DETECÇÃO DE
INTRUSÃO:**

**Um experimento com Open Source Security Information
Management (OSSIM)**

Rio de Janeiro

2009

Máximo Eduardo Egger

CORRELAÇÃO DE EVENTOS PARA DETECÇÃO DE INTRUSÃO: Um experimento com Open Source Security Information Management (OSSIM)

Monografia apresentada para obtenção de título de Especialista em Gerencia de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Rede de Computadores e Tecnologia Internet do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

Orientadora:

Prof^a. Mônica Ferreira da Silva, D. Sc., UFRJ, Brasil

Rio de Janeiro


2009

Máximo Eduardo Egger

**CORRELAÇÃO DE EVENTOS PARA DETECÇÃO DE INTRUSÃO: Um
experimento com Open Source Security Information Management (OSSIM)**

Monografia apresentada para obtenção de título de Especialista em Gerencia de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Rede de Computadores e Tecnologia Internet do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

Aprovado em junho de 2009



Prof^a Mônica Ferreira da Silva, D. Sc., UFRJ

RESUMO

EGGER, Máximo Eduardo. **CORRELAÇÃO DE EVENTOS PARA DETECÇÃO DE INTRUSÃO: Um experimento com Open Source Security Information Management (OSSIM)**. Monografia (Especialização em Gerência de Redes de Computadores e Tecnologia Internet) - Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2009.

Trabalhos recentes têm mostrado que é possível identificar anormalidades em redes de computadores automaticamente usando técnicas de aprendizagem ou de mineração de dados. Para verificar como o uso destas técnicas ajuda na identificação de anormalidades foi escolhida a ferramenta *Open Source Security Information Management* (OSSIM) para análise. Para esta análise foi montado um experimento a fim de permitir a verificação da instalação e configuração da ferramenta e o uso das técnicas de mineração para a correlação dos eventos de uma rede de computadores. Os resultados obtidos mostram a viabilidade da utilização dessas técnicas para a correlação e análise de eventos de modo automático em uma rede de computadores. Também foi possível identificar as dificuldades para instalação e configuração da ferramenta e do impacto para a correlação de eventos quando a quantidade de eventos foi elevada.

Palavras-chave: correlação de eventos computacionais, detecção de intrusão em redes de computador, gerência de segurança da informação e OSSIM.

ABSTRACT

EGGER, Máximo Eduardo. **CORRELAÇÃO DE EVENTOS PARA DETECÇÃO DE INTRUSÃO: Um experimento com Open Source Security Information Management (OSSIM)**. Monografia (Especialização em Gerência de Redes de Computadores e Tecnologia Internet). Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2009.

Recent studies have shown that it is possible to identify abnormalities in networks of computers automatically using techniques of learning and data mining. To see how the use of these techniques helps in the identification of abnormalities was chosen Open Source Security Information Management (OSSIM) tool for analysis. For this analysis, an experiment was set up to allow verification of the installation and configuration of the tool and the techniques of data mining for correlation of events from a computer network. The results show the feasibility of using these techniques for the correlation and analysis of events in automatic mode on a computer network. It was also difficult to identify the tool's installation and configuration and the impact on the events correlation when the number of events was high.

Key-words: log correlation, intrusion detection in computer network, security information management and OSSIM.

LISTA DE FIGURAS

	Página
Figura 2.1 – Classificação de IDS segundo Campello	14
Figura 3.1 – Componentes básicos do OSSIM	26
Figura 3.2 – Componentes básicos do OSSIM	27
Figura 3.3 – Tela Principal	28
Figura 3.4 – Exemplo de métricas	31
Figura 3.5 – Exemplo de relatórios de segurança	32
Figura 3.6 – Exemplo de relatórios de vulnerabilidades	33
Figura 3.7 – Exemplo de relatórios de disponibilidade	33
Figura 3.8 – Exemplo de relatórios de rede	34
Figura 3.9 – Exemplo de Analisador Forense	34
Figura 3.10 – Exemplo de relatório executivo	35
Figura 3.11 – Métricas	35
Figura 3.12 – Respostas automáticas	36
Figura 3.13 – Organização hierárquica de servidores e sensores	38
Figura 3.14 – Gerenciamento de inventário	39
Figura 3.15 – Medidas de Conformidade	40
Figura 3.16 – Exemplos de objetos métricos	40
Figura 3.17 – Exemplos de objetos métricos	41
Figura 3.18 – Exemplos de métrica tabular	41
Figura 5.1 – Arquitetura das máquinas OSSIM	43
Figura 5.2 – Detecção de novo sensor de agente	47
Figura 5.3 – Gráfico de volume de eventos e alarmes	50
Figura 5.4 – Regras do snort – visualização de regra p2p	51
Figura 5.5 – Exemplos de correlação com elevação do risco	51
Figura 5.6 – Termômetro de serviço	52
Figura A.1 – Arquivo de configuração do OSSIM	57
Figura D.1 – Visualização das Diretivas de correlação para o vírus conficker	65

LISTA DE TABELAS

Tabela 3.1 – Efeitos

Página
30

LISTA DE QUADROS

	Página
Quadro 3.1 – Funcionalidades do OSSIM	27
Quadro 5.1 – Seção plugins do arquivo /etc/ossim/agent/config.cfg	45
Quadro 5.2 – Arquivo lea.conf	45
Quadro 5.3 – Arquivo fw1-loggrabber.conf	46
Quadro 5.4 – Seção output-server do arquivo /etc/ossim/agent/config.cfg	46
Quadro 5.5 – Reiniciando o aplicativo ossim-agent	46
Quadro 5.6 – Verificando o conteúdo do arquivo fw-1.log	47
Quadro 5.7 – Verificando o conteúdo do arquivo agent-ossim.log	47
Quadro A.1 – Arquivo de configuração do OSSIM	57
Quadro B.1 – Arquivo de configuração fw1ngr60.cfg	59
Quadro C.1 – Comando para instalação do agente	61
Quadro C.2 – Arquivo de configuração do agente	62
Quadro D.1 – Arquivo directives.xml	66
Quadro D.2 – Arquivo virus.xml	69

LISTA DE ABREVIATURAS E SIGLAS

HIDS	Sistema de Detecção de Intruso baseado em Máquina em inglês <i>Host-based Intrusion Detection System</i>
HIMS	Sistema de Monitoramento de Integridade de Máquina em inglês <i>Host Integrity Monitoring System</i>
HIPAA	Health Insurance Portability and Accountability Act
IDS	Sistema de Detecção de Intruso em inglês <i>Intrusion Detection System</i>
IPS	Sistema de Prevenção de Intruso em inglês <i>Intrusion Prevention System</i>
ISO	International Organization for Standardization
NIDS	Sistema de Detecção de Intruso baseado em Rede em inglês <i>Network-based Intrusion Detection System</i>
KDD	Descoberta de Conhecimento em Banco de Dados em inglês Knowledge Discovery in Databases
PCI	Payment Card Industry
OSSIM	Open Source Security Information Management
RSS	Really Simple Syndication
SOX	Sarbanes-Oxley
ToS	Type-of-Service

SUMÁRIO

	Página
1 INTRODUÇÃO	11
1.1 MOTIVAÇÃO	11
1.2 OBJETIVO	12
2 REFERENCIAL TEÓRICO	13
2.1 SISTEMAS DE DETECÇÃO DE INTRUSO (IDS)	13
2.2 PROBLEMAS COM IDS	15
2.3 CORRELACIONAMENTO DE LOGS	16
2.4 DATA MINING	17
2.4.1 Algoritmos clássicos, técnicas de mineração de dados e visualização	18
2.4.2 Análise de eventos por Mineração de Dados	22
2.4.3 Ferramentas para Visualização	23
2.4.4 Detecção de Intrusão por Mineração de Dados	23
3 OSSIM	24
3.1 CARACTERÍSTICAS DO OSSIM	24
3.2 DETECÇÃO E CORRELAÇÃO	29
3.3 RELATÓRIOS	31
3.4 GERENCIAMENTO	36
3.4.1 Avaliação de Risco e Tomada de Decisão	36
3.4.2 Respostas Automáticas	36
3.4.3 Gerenciamento de Incidente	37
3.4.4 Central de Gerenciamento de Política	37
3.4.5 Gerenciamento do inventário	38
3.5 CONFORMIDADE	39
4 METODOLOGIA DE PESQUISA	42
4.1 TIPO DA PESQUISA	42
4.2 OBJETIVO DA PESQUISA	42
4.3 DESENVOLVIMENTO DO EXPERIMENTO	42
5 EXPERIMENTO	43
5.1 APRESENTAÇÃO DO AMBIENTE	43
5.2 CONFIGURAÇÃO DO AMBIENTE	44
5.3 AVALIAÇÃO DA INSTALAÇÃO E DA CONFIGURAÇÃO	48
5.4 DADOS COLETADOS E CORRELACIONADOS	49
6 CONCLUSÕES	53
6.1 TRABALHOS FUTUROS	54
REFERÊNCIAS	55
ANEXO A – Arquivos de configuração do OSSIM	57
ANEXO B – Configuração plugin para o FireWall1 NG R65	59
ANEXO C – Arquivos de configuração do agente OSSIM	61
ANEXO D – Arquivo de diretiva de correlação virus.xml	65

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Com o uso cada vez maior da rede mundial de computadores pela sociedade, pelos governos e pelas organizações comerciais, as redes de computadores passaram a ser de grande vulnerabilidade. Seu comprometimento, principalmente por atividades maliciosas, resulta em enormes prejuízos. (USA, 2003)

A informação e a proteção dos recursos são dois dos mais importantes interesses na maior parte das organizações. Para serem capazes de garantir a integridade de seus dados, bem como sua segurança, as organizações necessitam de sistemas de detecção de intrusão com o objetivo de se defenderem e se recuperarem de ataques (ABAD, 2003).

Com a contínua expansão da internet e a evolução tecnológica que tem possibilitado o aumento da velocidade das redes, a detecção de intrusão vem sendo forçada a buscar novos paradigmas. A análise do tráfego, com o aumento da taxa de transmissão de dados, tem se tornado uma tarefa árdua em função da necessidade cada vez maior de capacidade de processamento para se conseguir analisar todo o tráfego gerado em uma rede de dados (CARVALHO FILHO, 2000; CORDEIRO, 2006; VAZ, 2004).

Para essa análise os sistemas de detecção de intrusão (IDS) tentam detectar ataques pelo monitoramento dos eventos dos sistemas ou pelo comportamento da rede. Enquanto muitos IDSs existentes necessitam de definições manuais de comportamento normal e anormal, trabalhos recentes têm mostrado que é possível identificar anormalidades automaticamente usando técnicas de aprendizagem ou de mineração de dados (*data mining*).

1.2 OBJETIVO

Desta forma, pretende-se verificar **as facilidades e as dificuldades de se utilizar uma ferramenta *open source* para correlação de eventos¹ de várias fontes que faça uso de mineração de dados para a detecção de intruso em uma rede de computadores.**

A proposta deste trabalho é o estudo da ferramenta Open Source Security Information Management – OSSIM². Para isso, no Capítulo 2 será visto o referencial teórico das áreas envolvidas para que esta ferramenta seja compreendida não somente pela sua praticidade, mas também pela parte conceitual. No Capítulo 3 serão apresentadas algumas informações sobre o OSSIM, no que ele poderá ser usado, por quem, requisitos e instalação. No Capítulo 4 será apresentada a metodologia de pesquisa utilizada. Já no Capítulo 5 será mostrado como configurar e utilizar o OSSIM para um ambiente básico de teste, como o usado neste trabalho. No Capítulo 6, por fim serão apresentadas as conclusões do trabalho e a perspectiva de trabalhos futuros.

¹ Registros gerados pelos programas e comumente chamados de LOG.

² Disponível em <http://www.ossim.net>.

2 REFERENCIAL TEÓRICO

2.1 SISTEMAS DE DETECÇÃO DE INTRUSO (IDS)

A detecção de intrusão é o processo de monitoramento e análise de eventos ocorridos em sistemas computacionais ou em redes de dados, com o objetivo permitir uma medida emergencial que interrompa a atividade hostil, além de informações que possibilitem determinar ocorrência do ataque, localização do ataque – o que ocorreu e onde ocorreu e a identificação do atacante para as medidas cabíveis. (CARVALHO FILHO, 2000)

Para realizar a detecção de intrusos, os IDS realizam a coleta e a análise do tráfego da rede, procurando detectar anomalias que representem alguma atividade ilícita.

Os sistemas de detecção de intrusão normalmente são classificados segundo o método de detecção e segundo a arquitetura.

Quanto ao método de detecção pode ser: **baseado em comportamento**, quando o sistema de detecção de intrusão procura identificar desvios em relação às características de tráfego consideradas normais, ou, **baseado em assinaturas**, quando o sistema procura identificar atividades que correspondam a padrões já conhecidos.

Quanto à arquitetura, pode ser dividida segundo o alvo e segundo a localização. A arquitetura segundo o alvo pode ser: **baseado em host**, quando os dados utilizados como entrada para o IDS são provenientes de eventos e aplicações de detecção de intrusão executadas em uma estação; **baseado em rede**, quando os dados sobre o tráfego da rede são usados para detectar intrusões; e **híbrido**, quando as arquiteturas anteriores coexistem;

A arquitetura baseada na localização pode ser: **centralizada**, quando as funções de coleta, análise e gerência estão localizados em um mesmo ponto; **hierárquica**, quando as funções de coleta, análise e gerência estão distribuídas mas possuem relações de hierarquia; e **distribuída**, quando as funções de coleta, análise e gerência estão livremente distribuídas.

Mas ainda, segundo Campello (2001), o sistema de detecção pode ser classificado pelo comportamento pós-deteção e pela frequência de uso (FIG. 2.1).

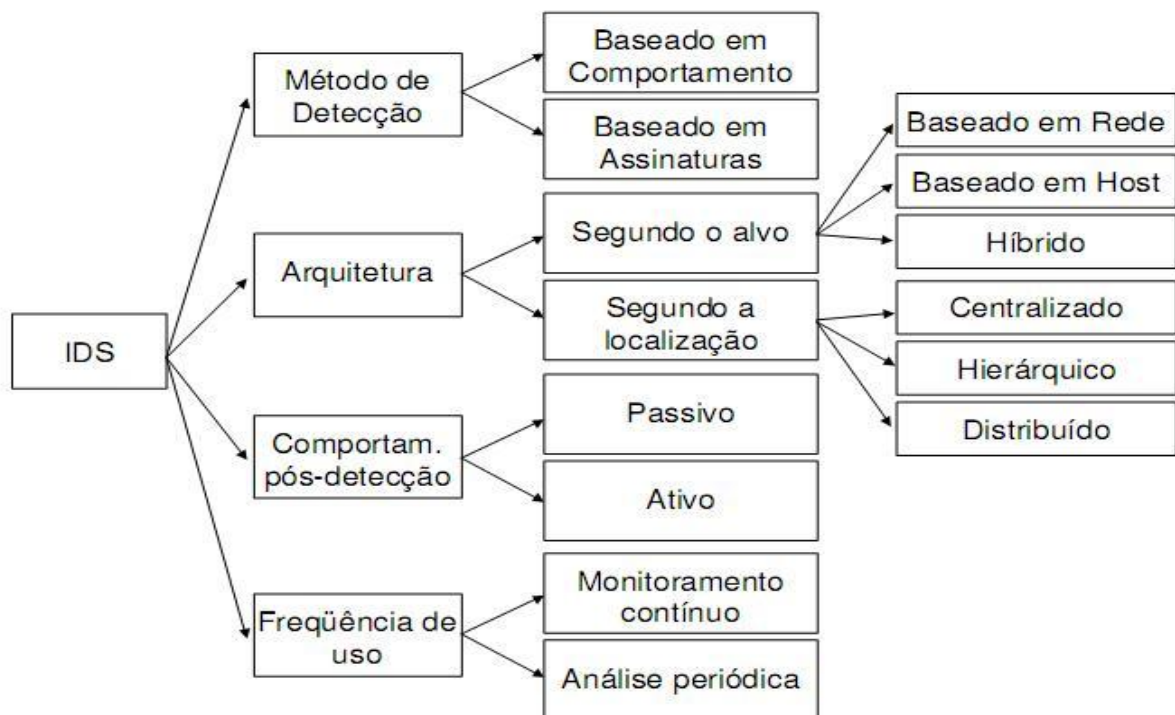


FIGURA 2.1 Classificação de IDS segundo Campello (2001)

Quanto ao comportamento pós-deteção, poderia ser **passivo** ou **ativo**, depois de realizada a detecção positiva de intrusão. Por exemplo, durante a detecção de uma varredura de portas, poder-se-ia modificar dinamicamente as regras de um *firewall*, no sentido de bloquear o acesso da máquina atacante. Esta seria uma atitude ativa de um IDS.

Em relação à classificação quanto à frequência de uso, basicamente classifica-se o sistema como atuando na monitoração de forma contínua ou em processamentos periódicos. Um sistema que atua de forma contínua possui a capacidade de atuar de forma ativa, mas exige grande capacidade de processamento de alguns componentes do sistema. Um sistema que faz o processamento periodicamente, não será capaz de atuar de forma ativa.

2.2 PROBLEMAS COM IDS

Em ambientes que fazem uso de IDS existem duas situações indesejáveis que ocorrem com frequência: **falsos positivos** ou falsos alertas e **falsos negativos**, quando os ataques reais não são detectados (VAZ, 2004).

Na situação de falsos positivos são gerados alertas que podem confundir o administrador de rede na sua análise, enquanto na situação de falso negativo implica em um comprometimento da rede caso o ataque seja bem sucedido, tendo em vista que o administrador não sabe que um ataque está ocorrendo e uma ação não será executada.

2.2.1 Redes de alta velocidade e a detecção de intrusão

Outra situação ocorre com o aumento da velocidade das redes de dados, onde a quantidade de informação que deve ser processada acaba sendo um problema para a grande maioria dos IDS, principalmente quando estamos em uma rede Gigabit Ethernet.

Para lidar com o grande volume de tráfego em redes de alta velocidade sem perda de pacotes durante a análise foi desenvolvido o conceito de fluxo: uma seqüência unidirecional de pacotes entre um dado par origem-destino. A identificação de cada fluxo é obtida pela combinação dos seguintes campos:

endereço IP de origem e destino, número da porta de origem e destino, tipo de protocolo, ToS Byte e interface lógica de entrada (CARVALHO FILHO, 2000).

O Netflow desenvolvido pela CISCO NETWORKS. (CISCO, 2004) é um exemplo de ferramenta que utiliza este conceito.

Com ferramentas como o NetFlow um grande volume de informações são registradas, embora sua análise por processo manual de identificação represente uma tarefa muito difícil.

2.3 CORRELACIONAMENTO DE EVENTOS (LOGS)

Entende-se por evento um registro de transação que consiste de um ou mais arquivos do tipo texto ou em formatos específicos gerados por certas aplicações, que permite a um analista visualizar as atividades que ocorrem em seus sistemas ou redes. Os eventos são gerados por dispositivos computacionais relacionados aos componentes de um sistema, tais como: aplicativos, sistema operacional ou dispositivos de *hardware*, bem como a interação entre sistemas formando redes de computadores. Eventos podem representar avisos de atividades normais, alertas ou erros em algum dos componentes do sistema. (GRÉGIO, 2007)

Os tipos de eventos variam conforme suas fontes geradoras, que vão desde o sistema operacional instalado, até a finalidade do sistema em si (servidor, computador de usuário, roteador), não sendo limitados a um sistema específico. Vários dispositivos ou classes de sistemas são capazes de gerar logs, entre os quais são citados por Schmidt (2006): Sistema operacional; Aplicações; Firewall; Sistema de Detecção de Intrusão; Anti-vírus; e Dispositivos de rede.

Cada um destes tipos de logs tem o seu formato específico, definido pelo fabricante ou mantenedor do software/dispositivo gerador do log.

Segundo Grégio (2007), com esta variedade de tipos de eventos surgem os primeiros problemas: dúvidas sobre quais eventos deve-se priorizar, quais as formas para sumariá-los de modo a possibilitar a análise e falta de experiência para discernir quais eventos são úteis e quais são descartáveis.

Entretanto, o principal problema relacionado à atividade de registro de eventos é a enorme massa de dados que tem de ser vasculhada por um analista em busca de situações anormais na rede ou nos sistemas sob sua responsabilidade. Isto faz com que esta tarefa não possa ser completada em um intervalo de tempo razoável e, por consequência, torna a implementação de ações ineficiente no que diz respeito à minimização ou mitigação de incidentes de segurança em redes de computadores e sistemas de informação.

Assim, para que o correlacionamento de uma grande quantidade de eventos possa gerar informação útil para a identificação de tentativas de ataque, ou de eventos maliciosos, no tempo adequado, é necessário que sejam utilizados processos automáticos de mineração de dados.

2.4 MINERAÇÃO DE DADOS

A mineração de Dados ou *Data Mining* é uma das etapas em processos de descoberta de conhecimentos úteis, previamente desconhecidos, em bancos de dados (Knowledge Discovery in Databases ou KDD). O processo de descoberta de conhecimento é composto de várias etapas independentes. No caso particular do sub-processo de mineração de dados, sua composição pode ser pelo uso e análise de resultados de várias técnicas estatísticas, de inteligência artificial e reconhecimento de padrões, de modelagem de bancos de dados, heurísticas, computação gráfica e visualização, computação de alto desempenho, etc. Outro

sub-processo é o de visualização, que pode tanto permitir a inferência sobre uma estrutura ou organização nos dados a ser analisados quanto apresentar resultados de análise de forma clara (FAYYAD et al., 1996 *apud* GRÉGIO, 2007).

Ainda segundo Grégio (2007), o processo de descoberta de conhecimento envolve o levantamento de requisitos e necessidades, e estudos iniciais sobre o problema a ser atacado, sendo composto por algumas fases:

1. Compreender o domínio da aplicação e entender as expectativas do usuário final do processo.
2. Criar/selecionar uma coleção de dados para aplicação.
3. Pré-processar e limpar os dados (eliminar ruídos e dados irrelevantes).
4. Transformar (reduzir e reprojeter) os dados (encontrar atributos úteis e interessantes).
5. Escolher as tarefas, métodos, modelos, parâmetros etc. do processo de mineração de dados e executar este processo.

Grégio (2007) ressaltou ainda o aspecto exploratório de mineração de dados, pois se faz necessário o conhecimento de um especialista nos dados que estão sendo analisados e algum conhecimento das técnicas de mineração de dados para sugerir a melhor abordagem para a análise dos dados. E mesmo assim pode ser que a “melhor abordagem” só seja descoberta depois de muitas tentativas com os diversos algoritmos e parâmetros existentes.

2.4.1 Algoritmos clássicos, técnicas de mineração de dados e visualização

Nesta seção serão apresentados alguns algoritmos clássicos, técnicas de mineração de dados e visualização segundo Grégio (2007).

O processo de classificação dos dados envolve basicamente a criação de uma função que seja capaz de, usando dados de entrada e um algoritmo, prever a classe ou categoria discreta correspondente a estes dados. Como entrada para algoritmos deste tipo, temos uma quantidade razoável de dados para os quais as classes são conhecidas. Com isso cria-se um classificador, o que consiste da fase de treinamento.

Em seguida, utilizam-se dados para os quais as classes não são conhecidas para indicar classes para os mesmos. Assume-se que dados desconhecidos “próximos” de dados conhecidos terão a mesma classe dos dados conhecidos.

Algumas das técnicas de classificação são:

- a) Sistemas especialistas, onde um especialista define regras que serão aplicadas aos dados dos logs para decidir a que categoria eles devem pertencer;
- b) Árvores de decisão, onde um algoritmo cria uma estrutura parecida com um sistema especialista mas usando análise dos dados para definir regras que classificam os dados. A estrutura criada pode ser facilmente interpretada por humanos, e recriada automaticamente no evento de novos dados ou fenômenos;
- c) Algoritmos estatísticos, que podem criar automaticamente (através da análise do espaço de atributos dos dados) funções de discriminação por regiões ou hiperplanos. Diferentes algoritmos podem usar diferentes medidas estatísticas para calcular as funções que definem as regiões de classificação e rejeição; e

- d) Algoritmos baseados em redes neurais, que são usados frequentemente como algoritmos estatísticos, podem criar funções de separação mais precisas, mas possui uma maior complexidade.

A técnica de regressão de dados é similar à classificação, exceto que ao invés de obter uma classe discreta para os dados, tem-se um valor numérico real (valor objetivo) e como resultado do treinamento do algoritmo obtém-se uma função numérica para o cálculo deste valor.

Como entrada para algoritmos de regressão deve-se ter uma quantidade razoável de dados para os quais os valores da função são conhecidos, e a função será então estimada usando um de vários modelos existentes. Os valores da função podem então ser calculados para as entradas.

Agrupamento ou clusterização (clustering) é o nome dado a técnicas que identificam grupos semelhantes entre si no espaço amostral, o que provavelmente corresponde a eventos semelhantes em um log. Estas técnicas também assumem que dados em um grupo devem ser diferentes de dados em outro grupo.

Nesta técnica os dados de entrada não precisam ter identificação de classes, já que os grupos serão identificados sem esta informação. Os algoritmos identificam determinado número de grupos de dados e calculam a associação dos dados de entrada aos grupos de saída. Adicionalmente estatísticas e outras informações sobre os grupos podem ser criadas.

A qualidade do agrupamento pode ser medida com algumas métricas, mas comparações entre algoritmos são complicadas. É necessário uma fase de pós-processamento para fazer uma correspondência entre os grupos e classes de eventos. Existem duas técnicas principais de agrupamento: a hierárquica que começa juntando dados e formando grupos pequenos até que todos os dados

pertençam a um único grupo, e que permite a exploração de diversas opções de agrupamento; e a particional, que agrupa dados iterativamente até que o número de grupos solicitado esteja estável.

Algoritmos de agrupamento são então quase sempre iterativos, necessitando a comparação de dados entre si várias vezes até que seja possível chegar a grupos estáveis, o que pode causar custos de processamento e tornar a aplicação inviável para grandes massas de dados.

Técnicas de descoberta de associação trabalham de forma diferente das apresentadas até agora. Basicamente estas técnicas tentam descobrir elementos que ocorrem (ou não) em comum em coleções de dados. Os dados de entrada são estruturas com associações (por exemplo, lista de artigos comprados, pequenas séries temporais multivariadas, existência ou não de um determinado fato em uma entrada de log, etc.) e o algoritmo identifica a existência de elementos em comum e suporte para esta existência.

O algoritmo tenta descobrir regras do tipo “Se X ocorre na base de dados, então Y também ocorre (com alguma relação X)” ou de co-ocorrência, como “se X, Y e Z ocorrem na base de dados então A também ocorre (com alguma relação X, Y e Z)”. Consideramos X, Y e Z como os antecedentes da associação; A é o conseqüente.

Implementações de algoritmos de associação podem usar várias métricas de qualidade que indicam qual é a credibilidade da regra de associação descoberta. Uma destas métricas é suporte em uma associação, que indica o quão rara a associação é em bases de dados. Outra métrica é a confiança em uma associação: o antecedente pode ocorrer várias vezes na base de dados mas nem sempre com o mesmo conseqüente associado.

2.4.2 Análise de eventos por Mineração de Dados

Uma vez que as técnicas tradicionais, apresentadas na seção 2.4.1 para análise de eventos são computacionalmente custosas, dado que envolvem configurações por vezes complicadas de ferramentas, não reduzem de maneira inteligente o volume de dados, despendem uma quantidade de tempo absurda e são inteiramente dependentes do conhecimento do analista, novas abordagens devem ser estudadas para fornecer soluções melhores para estes problemas.

Várias pesquisas na aplicação de técnicas de mineração de dados em problemas envolvendo segurança da informação tem sido realizadas (DOKAS et al., 2002; MENA, 2003; LEE; STOLFO, 1998 apud GRÉGIO, 2007). Alguns trabalhos e técnicas associadas a aplicações de mineração de dados que podem ser empregadas à análise de logs foram citadas por Grégio (2007) e estão enumerados a seguir:

- a) Modelos estatísticos (VALDES; SKINNER, 2001 apud GRÉGIO, 2007);
- b) Vizinhos mais próximos (STEARLEY, 2006 apud GRÉGIO, 2007);
- c) Algoritmos genéticos (MÉ, 1998 apud GRÉGIO, 2007);
- d) Redes neurais artificiais (CANSIAN, 1997);
- e) Mapas auto-organizáveis de Kohonen (RAMADAS et al., 2003; RHODES et al., 2000 apud GRÉGIO, 2007).
- f) Máquinas de vetores de suporte (MUKKAMALA et al., 2002 apud GRÉGIO, 2007);

2.4.3 Ferramentas para Visualização

Segundo Grégio (2007), a utilização de técnicas de visualização em eventos pode representar uma grande ajuda na interpretação dos acontecimentos na grande massa de logs coletada. Existem diversas ferramentas que implementam formas diferentes de visualização, algumas delas específicas para eventos ou tráfego de rede.

2.4.4 Detecção de Intrusão por Mineração de Dados

Em face às limitações dos sistemas de detecção de intrusão tradicionais, que usam técnicas baseadas em abuso ou anomalia, vários trabalhos vêm tentando superar os problemas conhecidos e desenvolver sistemas de detecção de intrusão com técnicas de mineração de dados. Espera-se com isso aumentar a taxa de identificação de comportamento intrusivo e reduzir a taxa de falsos positivos.

Como apresentado na seção 2.4.2, as técnicas de mineração de dados são muitas e variadas, aspectos como a capacidade explicativa, complexidade, custo computacional e capacidade de aprendizagem incremental dos modelos são determinantes na hora de selecionar qual o melhor a ser utilizado em cada situação (ZURUTUZA; URIBEETXEBERRIA, 2005 apud GRÉGIO, 2007).

3 OSSIM – Open Source Security Information Management

3.1 CARACTERÍSTICAS DO OSSIM

O OSSIM visa unificar a rede de monitoramento, segurança, correlação e qualificação, em um único sistema permitindo uma ampla compilação. Para isso o sistema integra e gerencia várias ferramentas de código aberto como (ALIEN, 2008):

- *Arpwatch* – utilizada para a detecção de alterações no endereço MAC.
- *P0f* – utilizada para a detecção passiva de Sistema Operacional.
- *Pads* e *Nmap* – utilizadas para detecção de alterações nos serviços da rede.
- *Nessus* – utilizada para a análise de vulnerabilidade e correlação cruzada (IDS vs Security Scanner).
- *Snort* – IDS, também utilizado para a correlação cruzada com o *nessus*.
- *Spade* – utilizada para obter conhecimentos sobre os ataques sem assinatura. A ferramenta detecta conexões anômalas analisando as portas utilizadas e o destino.
- *Tcptrack* – utilizada para obter informações sobre as conexões TCP de uma interface específica para a correlação de ataque.
- *Ntop* – utilizada para construir uma base de dados com informações da rede e obter detecção de anomalia por comportamento anormal.
- *Nagios* – utilizada para monitorar os hosts e produzir informações sobre a disponibilidade de serviço.
- *Osiris* – Host Integrity Monitoring System (HIMS) utilizada para monitorar máquinas Windows e coletar em tempo real dados sobre: modificações em

arquivos utilizando checksums, modificações em portas, mudanças de usuários e de kernel.

- *Snare* – utilizada para monitorar máquinas Windows e coletar em tempo real dados sobre: criação, modificação e acesso de arquivos, uso de USB, Login e Logoff e instalação e execução de programas.
- OCS-NG – solução de inventário Cross-Platform.
- OSSEC – detecção de integridade, rootkit e registro.
- OSSIM C/A – monitor que requisita informações de comprometimento ou de ataque aos agentes OSSIM

Além de utilizar as ferramentas descritas anteriormente, o OSSIM ainda possui os seguintes módulos:

- Sistema de coleta;
- Correlação lógica, cruzada e de inventário;
- Gerenciador de inventário (hosts, redes, grupos e serviços);
- Gerenciador de políticas de coleta e de configuração;
- Gerenciador de resposta e incidente; e
- Conformidade de segurança e Risk Metrics Scoredash

Os componentes de uma instalação básica são (Fig. 3.1):

- Uma base de dados.
- Um servidor onde está instalada a ferramenta de correlação, qualificação e avaliação de risco.
- Vários agentes que fazem coleta informações a partir de uma série de dispositivos. (Fig. 3.2)

- Um *daemon*³ de controle que faz trabalhos de manutenção. É chamado *framework*.
- O *frontend* é baseado na Web, unificando todas as informações recolhidas e proporcionando a capacidade de controlar cada um dos componentes (Fig. 3.3). Para facilitar o entendimento das funcionalidades existentes em cada página do *frontend* existe um *link* apontando para uma página na internet mantida pelo desenvolvedor.

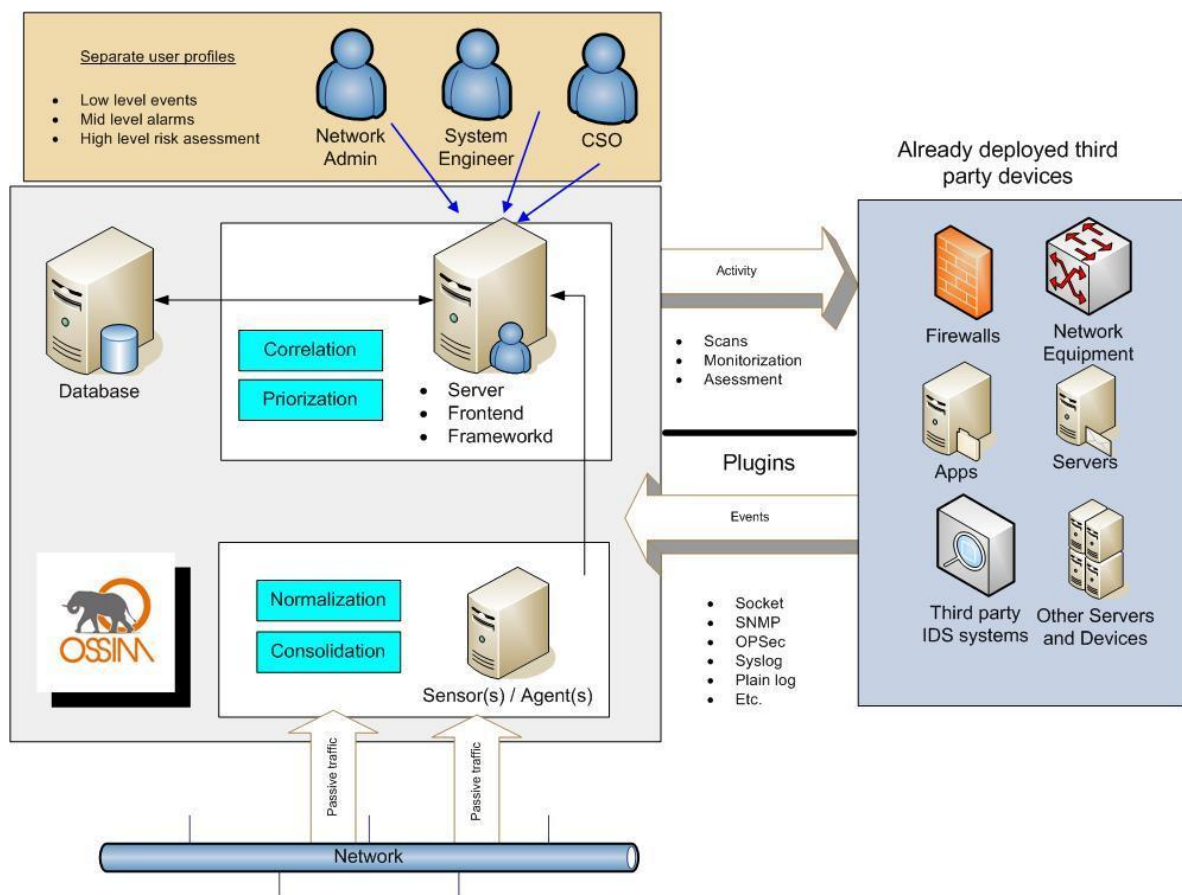


Figura 3.1 Componentes básicos do OSSIM – Fonte OSSIM (2008)

³ É um programa que roda em segundo plano (*background*), que realiza uma função ou tarefa de sistema específica. O termo *Daemon* (demônio) vem da mitologia grega, onde os *Daemons* são espíritos guardiões que influenciam o caráter ou a personalidade de uma pessoa. (NEMETH, 2007, p. 608)

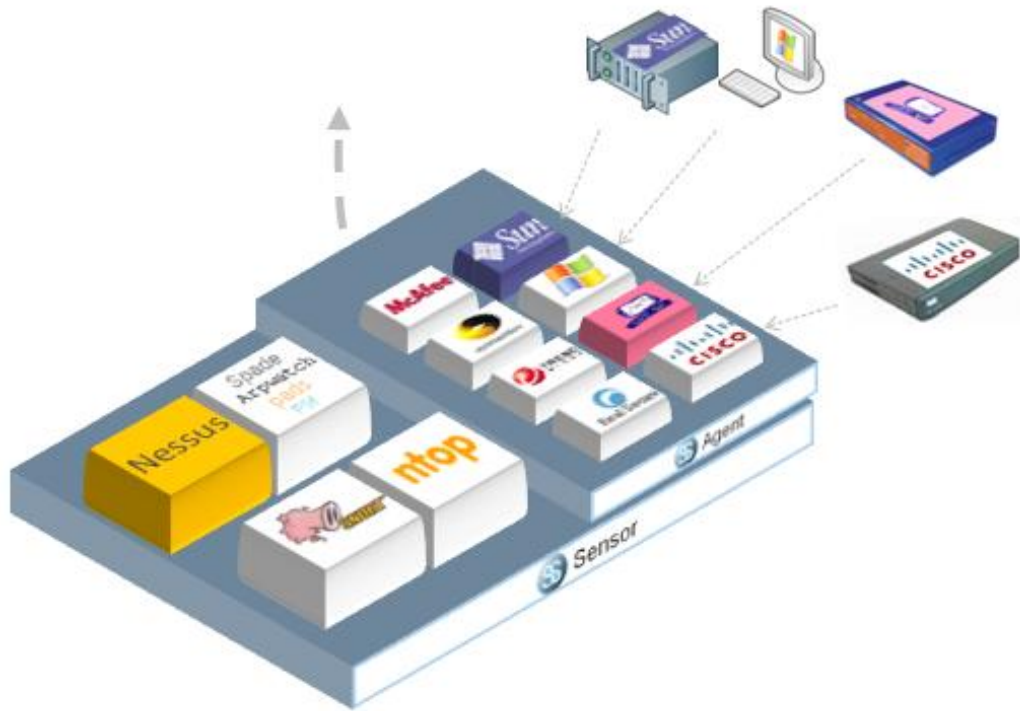


Figura 3.2 Componentes básicos do OSSIM – Fonte OSSIM (2008)

O quadro 3.1 apresenta as funcionalidades do OSSIM do nível mais baixo até o mais alto:

Quadro 3.1 – Funcionalidades do OSSIM. Fonte (OSSIM,2008)

Detecção				Resposta	
Monitoração >	Coleta >	Correlação >	Avaliação de Risco >	Gerenciamento de Incidentes >	Medidas de
Escanear				Relatório e Análise	Segurança
Inventário					

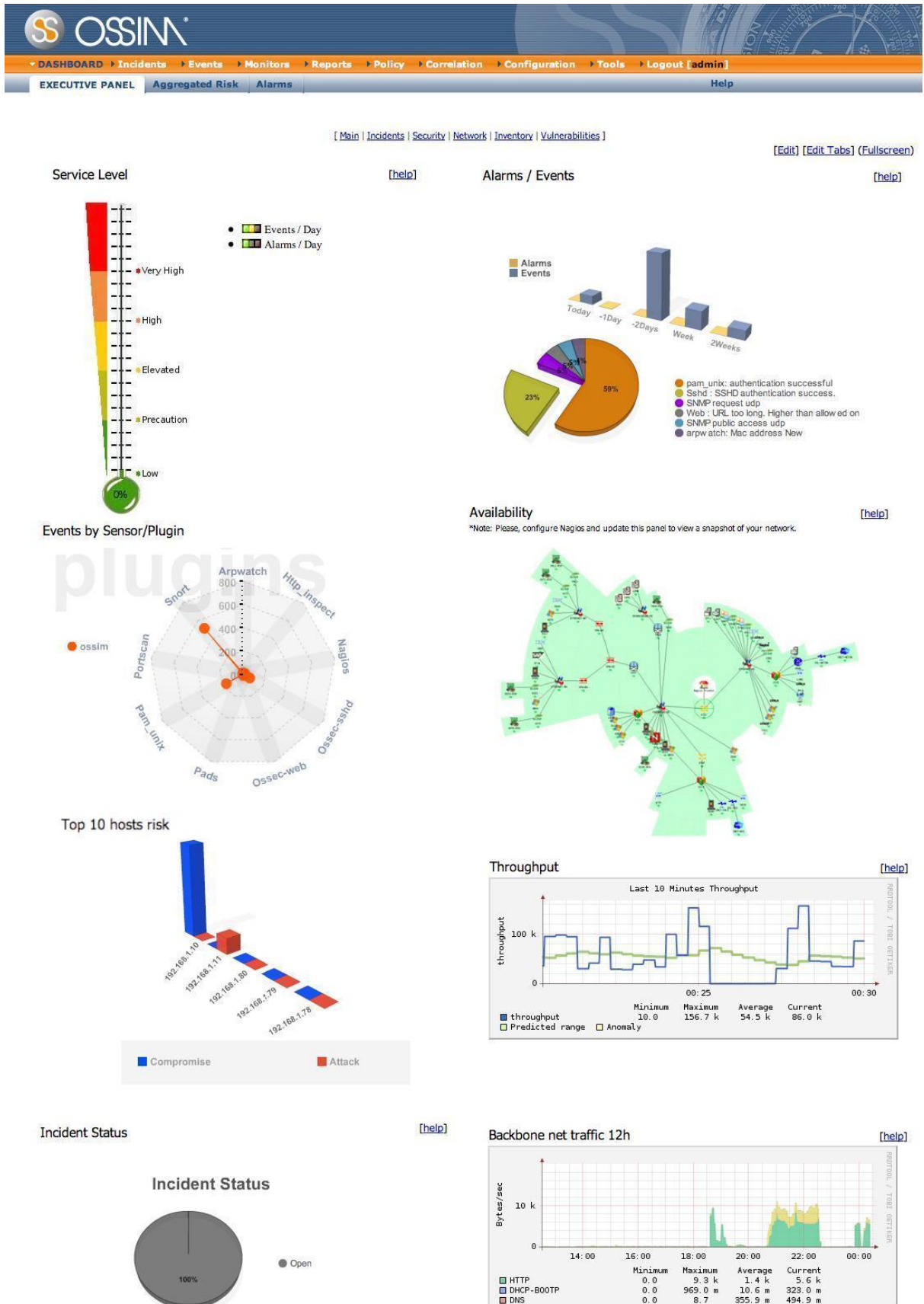


Figura 3.3 – Tela Principal – Fonte OSSIM (2008)

3.2 DETECÇÃO E CORRELAÇÃO

O sistema de coleta de eventos concentra os mesmos por meio de agentes coletores que utilizam detectores e monitores. A diferença entre eles é que as informações produzidas pelos detectores são continuamente enviados pelos agentes ao servidor e no caso dos monitores, o envio das informações ocorre como resposta a uma requisição do servidor ao agente que utiliza o monitor para verificar se a condição requisitada foi encontrada ou não.

Os principais detectores utilizados pelo OSSIM são o snort (NIDS), snare, osiris (HIMS) e spade. E os monitores são nmap, ntop e nessus. Existe também a possibilidade de utilizar detectores e monitores externos, tais como firewall, ids, roteadores, vpn e servidores. Isto é possível pelo uso de *plugins* nos agentes.

Após concentração dos alertas gerados pelos monitores e das informações geradas pelos detectores, o OSSIM inicia o seu trabalho incluindo três métodos para descartar os falsos positivos, e priorizar ou descobrir padrões mais complexos que tenham sido ignorados pelos detectores. São eles:

- **Priorização** – os alertas são recebidos através de um processo de contextualização desenvolvido pela definição de uma política de segurança topológicas em combinação com o inventário dos sistemas.
- **Avaliação de risco** – Cada caso é avaliado em relação aos seus riscos associados, em outras palavras, em proporção com os ativos de risco, da ameaça representada pelo evento, e pela probabilidade real.
- **Correlação** – onde é feita uma análise de uma coleção de eventos para obter informações de maior valor. O OSSIM realiza três tipos de correlação: Lógica – que permite o administrador criar diretivas de correlação ou regras lógicas para reunir pequenos eventos diferentes de diferentes fontes em um novo

padrão. Cruzada – que cruza informações dos IDS (ex.: snort) e dos scanners de vulnerabilidades (ex.: nessus), aumentando ou diminuindo a prioridade do evento no caso de estarmos vulnerável ou não ao ataque detectado. Inventário – verifica o serviço e o tipo/versão de sistema operacional que o ataque pode afetar e depois verifica se o host atacado possui o Sistema Operacional/Serviço ativo, descartando o evento se não for verdadeiro.

Um exemplo típico de correlação poderia ser uma “detecção de *worm*”⁴ depois de localizar um número de conexões anormais. Também poderiam ser criados diferentes tipos de níveis de correlação, e correlacionado algum destes alarmes “detecção de *worm*” poderíamos produzir um alarme mais abstrato do tipo “Alarme de Praga”.

A tabela 3.1 mostra como as propriedades mencionadas acima são afetadas por estes processos:

Tabela 3.1 – Efeitos. Fonte OSSIM (2008)

	Processamento	Efeito
Priorização	Avaliar uma ameaça pela contextualização de um evento	Melhora da confiabilidade
Avaliação de risco	Avaliar o risco em relação ao valor dos ativos	Melhora da confiabilidade
Correlação	Comparar vários eventos para obter informações de maior valor	Melhora da confiabilidade, da sensibilidade e de abstração

3.3 RELATÓRIOS

São muitos os relatórios que podem ser feitos pelos diferentes utilizários do OSSIM. Os relatórios variam de relatórios de baixo nível de detalhamento até relatórios de alto nível com métricas personalizadas.

O sistema permite ainda definir métricas personalizadas em painéis específicos para mostrar situações de segurança e da rede (Fig. 3.4).



Figura 3.4 – Exemplos de métricas - Fonte OSSIM (2008)

Os Relatórios de Segurança apresentam uma visão geral da situação de segurança, agregando dados a partir de diferentes pontos de vista e exibindo-os em formatos gráficos tais como: Eventos de segurança, Incidentes, Vulnerabilidades,

⁴ **Worm** (verme, em português), em computação, é um programa auto-replicante, semelhante a um vírus. Entretanto um vírus infecta um programa e necessita deste programa hospedeiro para se

Anomalias, Tendência da Rede, Perfis de computadores, Recursos de computadores monitorados, etc. (Fig. 3.5)

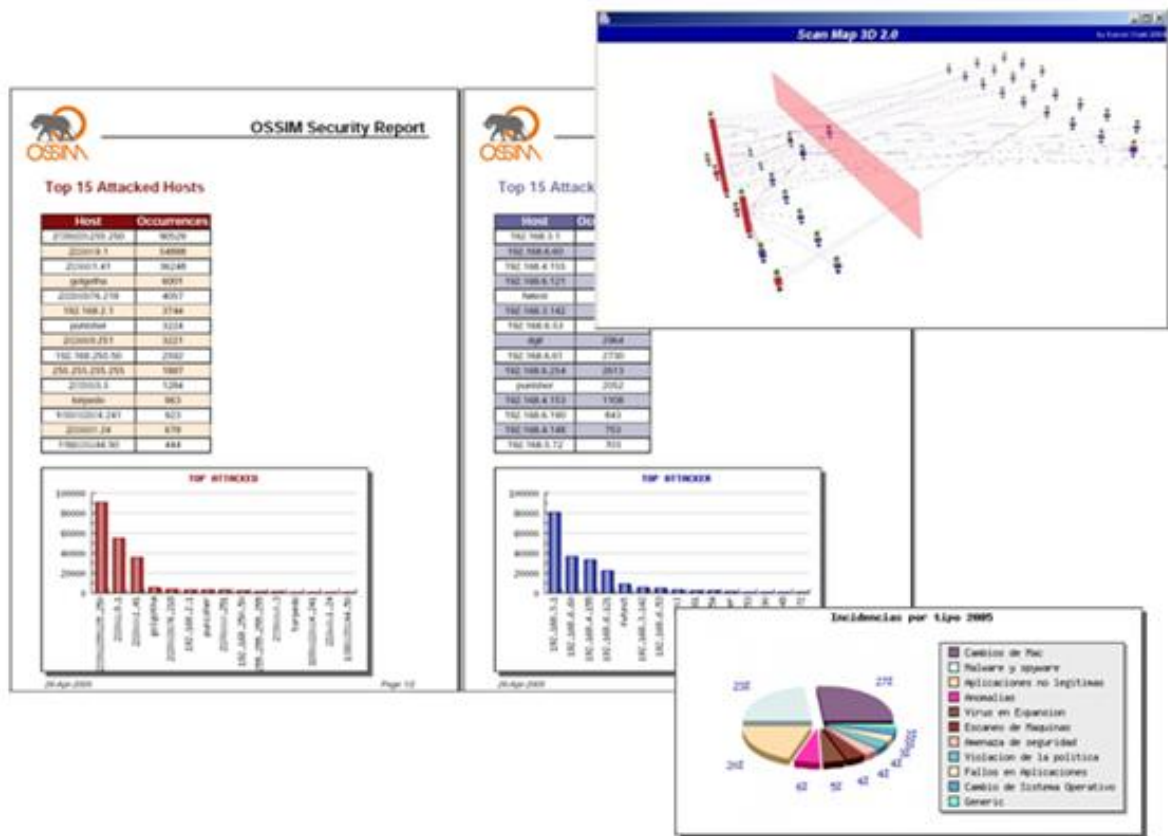


Figura 3.5 – Exemplos de relatórios de segurança - Fonte OSSIM (2008)

Os Relatórios de Vulnerabilidade mostram relatórios agregados e por vulnerabilidade de computadores. (Fig. 3.6)

Os Relatórios de Disponibilidade oferecem informações detalhadas sobre computadores e níveis de serviço. (Fig. 3.7)

Os Relatórios de Rede oferecem visão em tempo real do tráfego de rede, bem como estatísticas e perfis detalhados para cada máquina na rede (Fig. 3.8)

O Analisador Forense permite a pesquisa através de milhões de eventos armazenados para obter detalhes do que aconteceu em um determinado momento e a criação de visualizações personalizadas para a revisão periódica. (Fig. 3.9)

propagar, já o Worm é um programa completo e não precisa de outro programa para se propagar.

Ainda podem ser criados os Relatórios Executivos personalizados utilizando-se de vários tipos de gráficos, gráficos de tendências, de pizzas, mapas, fontes RSS⁵, nuvens, termômetros, etc. (Fig. 3.10)

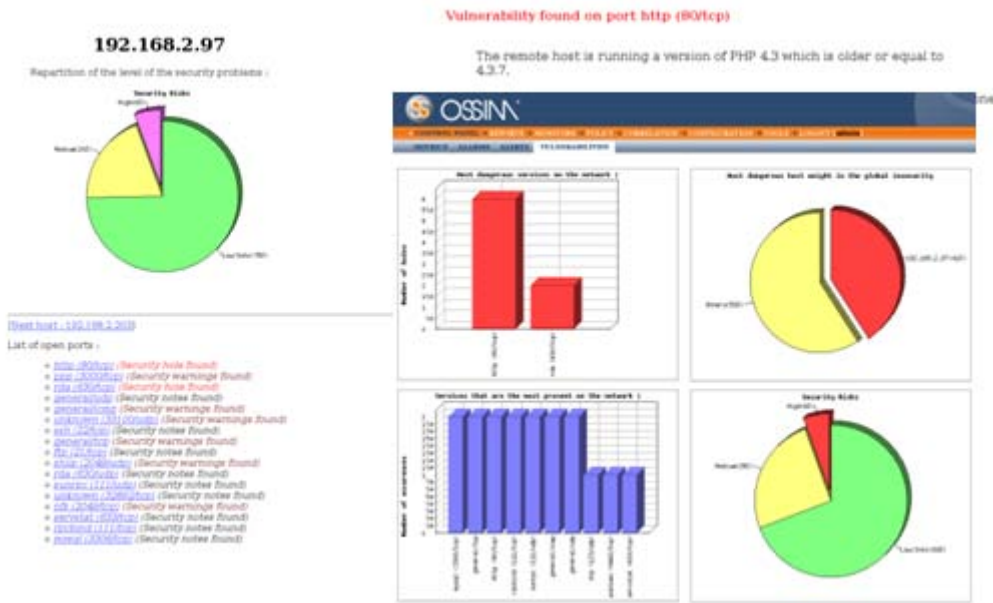


Figura 3.6 – Exemplos de relatórios de vulnerabilidades - Fonte OSSIM (2008)

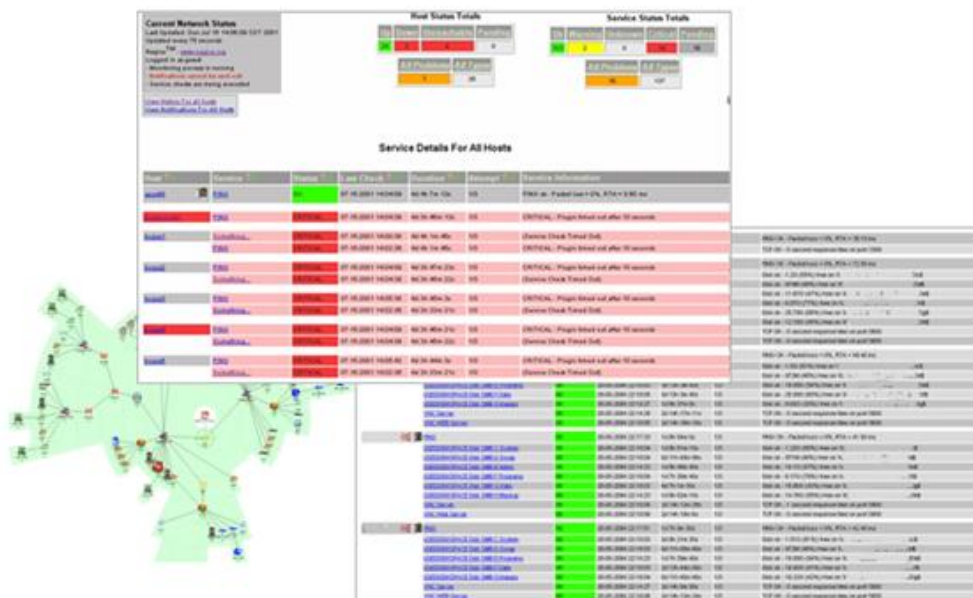


Figura 3.7 - Exemplo de relatórios de disponibilidade - Fonte OSSIM (2008)

⁵ RSS é um subconjunto de "dialetos" XML que servem para agregar conteúdo. É usado principalmente em sitios de notícias e blogs(contração do termo "Web log"). A abreviatura do RSS é



Figura 3.8 – Exemplos de relatórios de rede - Fonte OSSIM (2008)

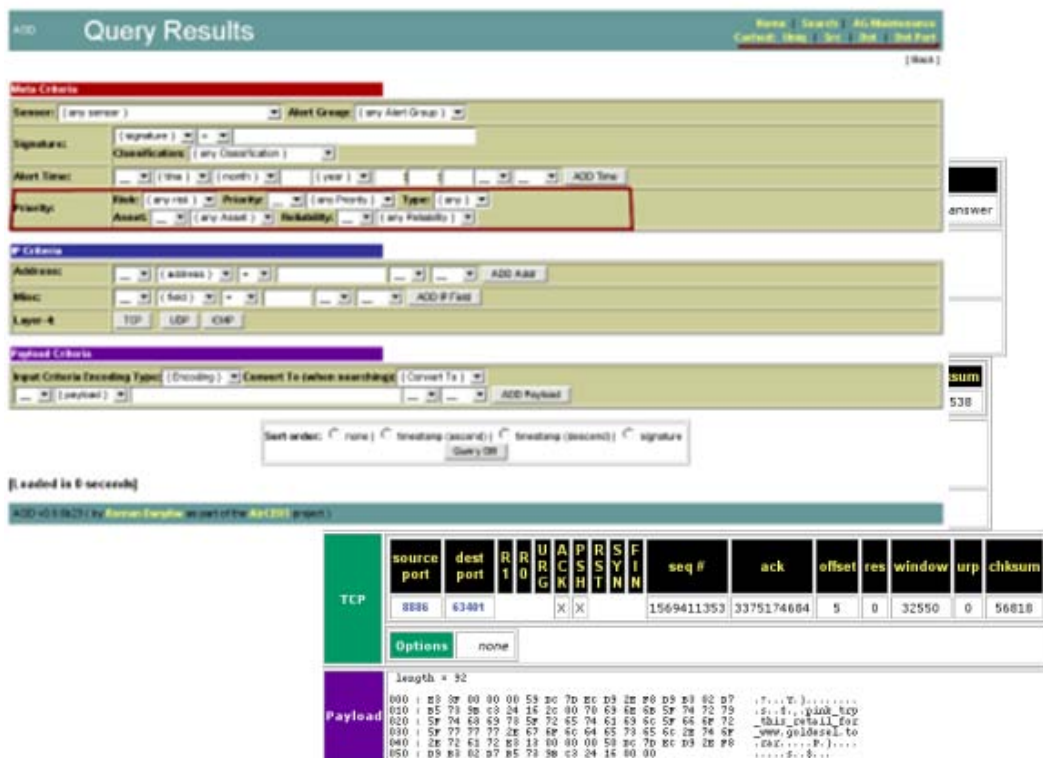


Figura 3.9 – Exemplos de Analisador Forense - Fonte OSSIM (2008)

usada para se referir aos seguintes padrões: Rich Site Summary (RSS 0.91), RDF Site Summary (RSS 0.9 e 1.0) e Really Simple Syndication (RSS 2.0).

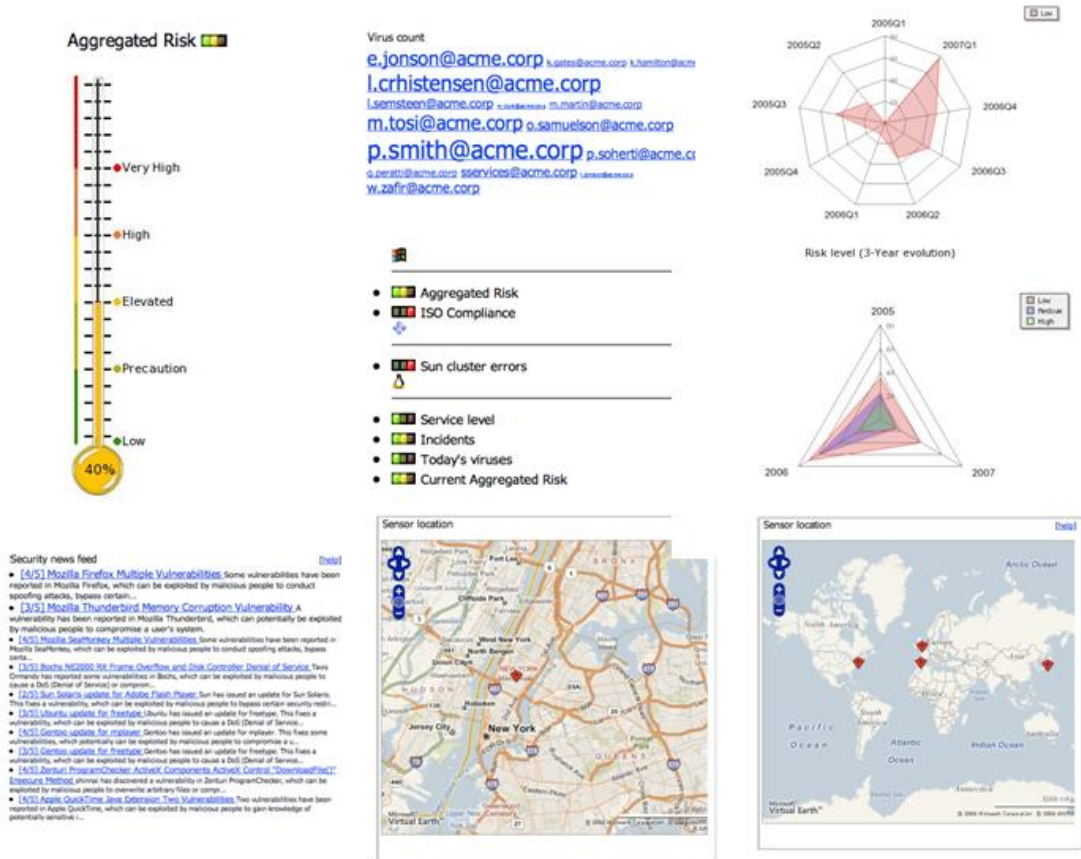


Figura 3.10 – Exemplos de relatório executivo - Fonte OSSIM (2008)



Figura 3.11 – Métricas - Fonte OSSIM (2008)

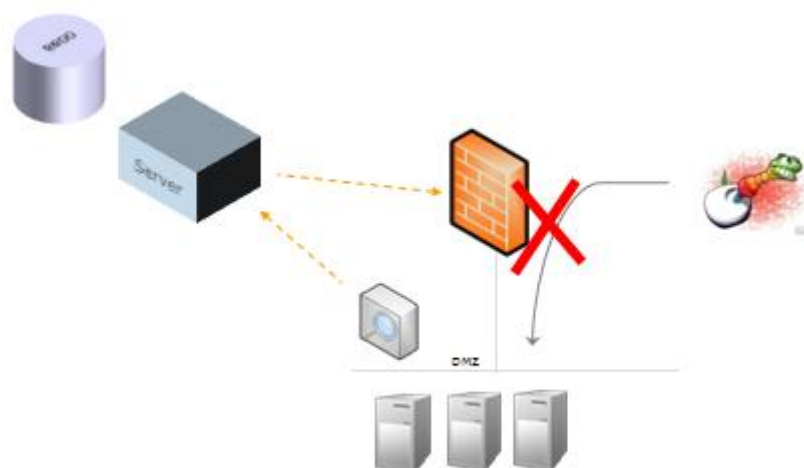


Figura 3.12 – Respostas automáticas - Fonte OSSIM (2008)

3.4 GERENCIAMENTO

3.4.1 Avaliação de Risco e Tomada de Decisão

Todas as decisões no OSSIM são motivadas por parâmetros de risco, neste caso o risco é calculado para cada evento recebido pelo OSSIM utilizando os ativos, as ameaças e os parâmetros confiabilidade.

Existe também uma pontuação da métrica de riscos no Painel de instrumentos (RiskMetrics Score Dashboard) que consolida o parâmetro de risco para cada objeto da rede produzindo uma visualização total da situação de risco de cada host e da rede.

Cada vez que uma situação de risco é produzida respostas automáticas ou manuais são geradas para contê-la.

3.4.2 Respostas Automáticas

Uma vez que é recebido um alarme confirmado de um ataque bem sucedido, nós podemos acionar respostas automáticas, para atenuar este ataque.

Estas respostas podem ser ações predefinidas como enviar um e-mail, bloquear uma conexão no nível de firewall ou desativar uma porta de um switch.

Depois de definir um conjunto de ações genéricas é estabelecida uma política de ações de disparo, com um conjunto de variáveis pré-definidas, tais como origem do IP, data, *plugin* ou qualquer outro campo do registro sendo substituído em tempo real por valores atuais.

3.4.3 Gerenciamento de Incidente

O OSSIM inclui um Gerenciador de Incidente que controla a atribuição de tarefas a serem executadas como ações resultantes de eventos de segurança.

O Gerenciador de Incidente permite-nos atribuir registros de observação para a maioria das ferramentas de relatórios do OSSIM, tais como o Painel de alarme, o console forense ou o Painel de instrumentos de pontuação da métrica de riscos (RiskMetrics Score Dashboard). Cada registro de observação mostra o responsável, o status, as ações tomadas e o monitoramento do fluxo de trabalho a partir da criação do registro para cada situação real. Todos os registros de observação são armazenadas no banco de dados e uma ferramenta de pesquisa permite filtrá-las. Um relatório de Exploração do processamento destes dados pode ser impresso automaticamente. Também é possível extrair tendências de incidente e implementar métrica para medir a situação e monitorar sua evolução ao longo do tempo.

3.4.4 Central de Gerenciamento de Política

A Central de gerenciamento de Política define como cada evento individual em cada elemento de uma grande rede deve se comportar relacionado-os com as seguintes ações:

- Correlacionamento
- Transmissão
- Priorização
- Armazenamento
- Consolidação

Com isto é possível a criação de uma organização hierárquica de servidores e sensores, e de diferentes níveis de correlação e de pontos de armazenamento. Todos as configurações são ajustadas em modo gráfico. (Fig. 3.13)

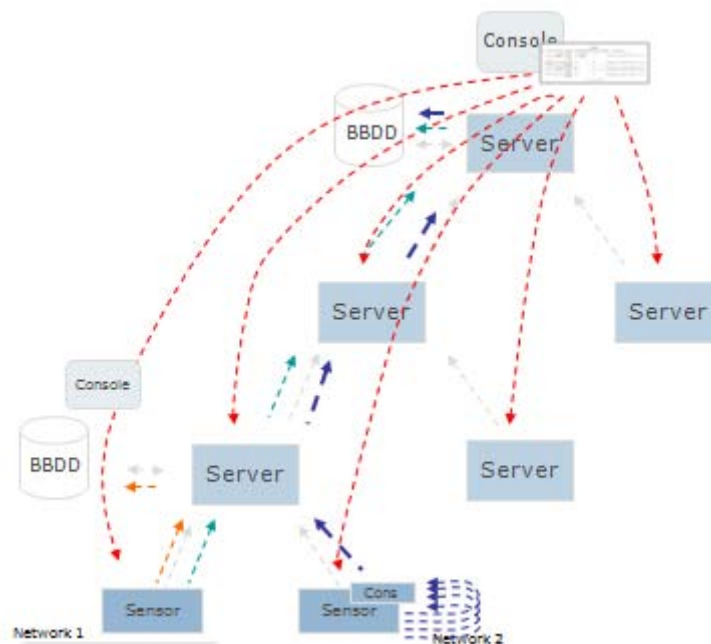


Figura 3.13 – Organização hierárquica de servidores e sensores.
Fonte OSSIM (2008)

3.4.5 Gerenciamento do inventário

O gerenciamento do inventário dos equipamentos conectados as redes podem ser realizado de quatro formas diferentes no OSSIM (Fig. 3.14):

- Passiva (automático e sem agentes) , usando as ferramentas *p0f*, *pads* e *arpwatch*;
- Ativa (automático e sem agentes), usando a ferramenta *nmap*;

- Agente (automático e com agentes), usando a ferramenta OCS; e
- Manual.

Os dados inventariados podem ser usados para correlacionar eventos, descartando ou priorizando-os. Isto é feito verificando se o sistema operacional e os serviços da máquina alvo são compatíveis aos do ataque.

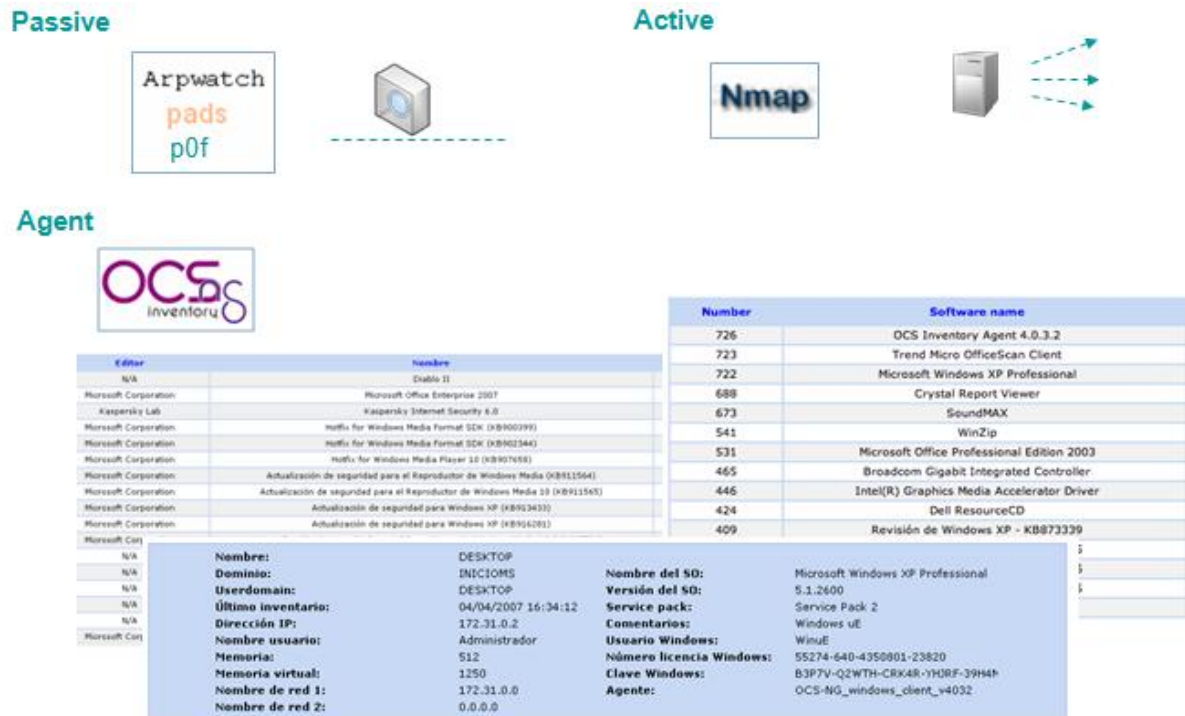


Figura 3.14 – Gerenciamento de inventário - Fonte OSSIM (2008)

3.5 CONFORMIDADE

Painéis executivos são personalizados para enviar e informar o cumprimento de regulamentos específicos como: ISO2700X, SOX, HIPAA, ISO-17799, PCI e outras.

O OSSIM permite a medição, mostrando as tendências e criando métricas para o cumprimento das normas e dos regulamentos de segurança, fornecendo um quadro completo para a conformidade normativa. (Fig. 3.15)

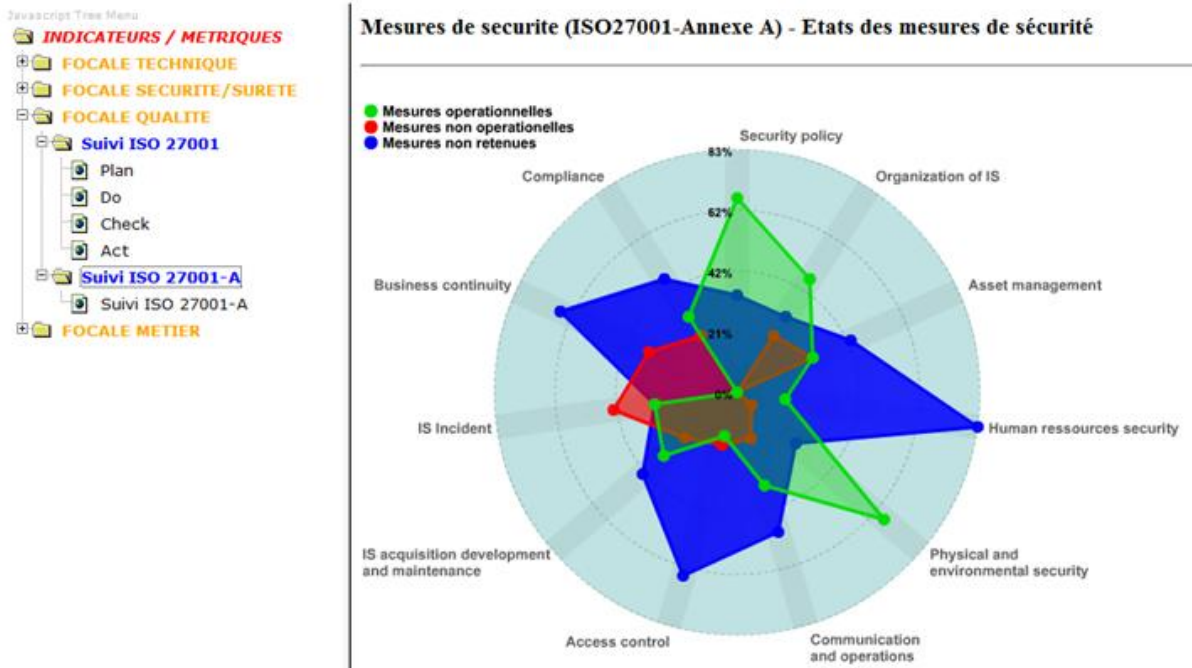


Figura 3.15 – Medidas de Conformidade - Fonte OSSIM (2008)

Esses objetos métricos são apresentados como gráficos dos parâmetros monitorados, dando uma visualização rápida do resumo mensurável da situação de segurança. (Fig. 3.16)

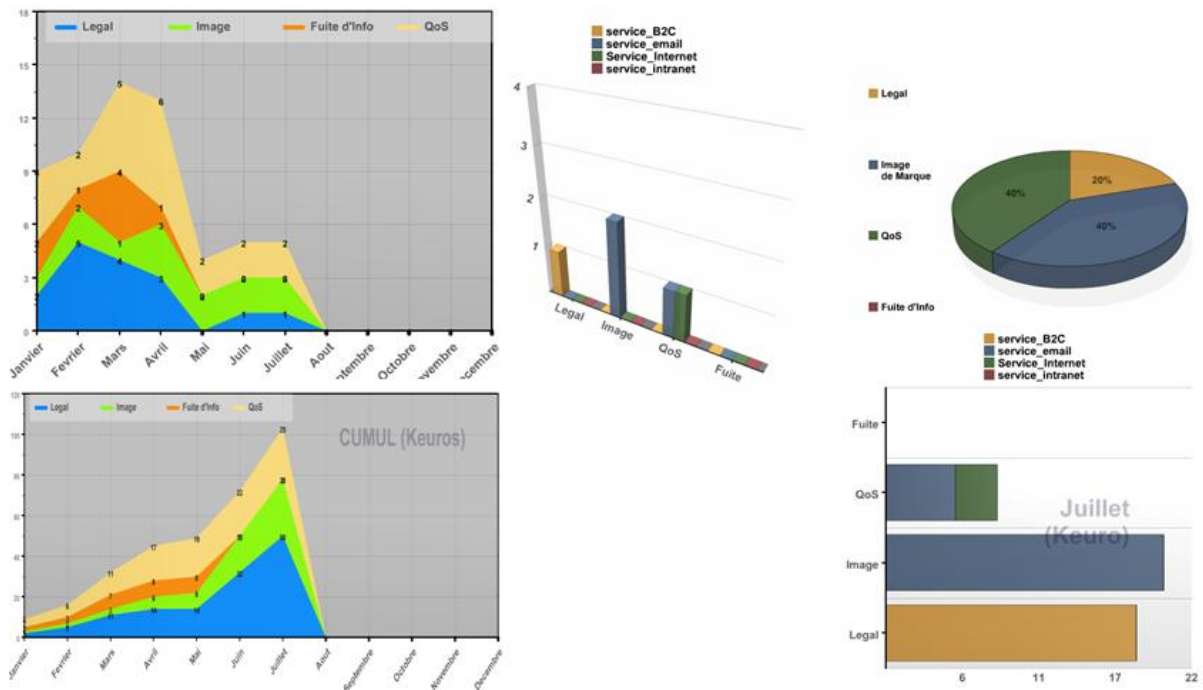


Figura 3.16 – Exemplos de objetos métricos - Fonte OSSIM (2008)

Ambos os controles, técnico e não-técnico, podem ser monitorados obtendo uma visão completa da regulamentação normativa. (Fig. 3.17 e Fig. 3.18)

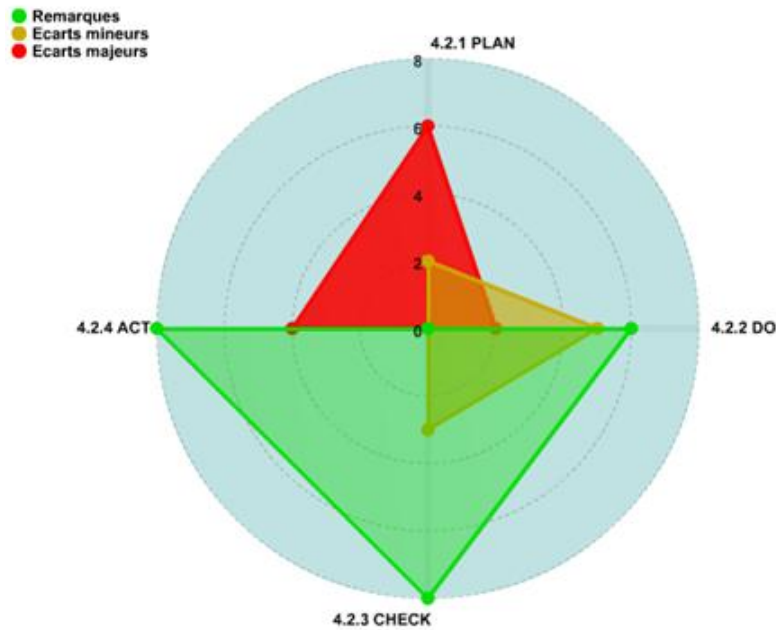


Figura 3.17 – Exemplos de objetos métricos - Fonte OSSIM (2008)

Access	A	B	C	D	E	F	G
1	A.5.1	0	Information security policy	none		1	
2	A.5.1	1	Information security policy document	none		1	
3	A.5.1	2	Review of the information security policy	none		0	
4	A.6.1	0	Internal organization	none		1	
5	A.6.1	1	Management commitment to information security	none		1	
6	A.6.1	2	Information security coordination	none		1	
7	A.6.1	3	Allocation of information security responsibilities	none		2	
8	A.6.1	4	Authorization process for information processing facilities	none		0	
9	A.6.1	5	Confidentiality agreements	none		2	
10	A.6.1	6	Contact with authorities	none		1	
11	A.6.1	7	Contact with special interest groups	none		1	
12	A.6.1	8	Independent review of information security	none		0	
13	A.6.2	0	External parties	none		2	
14	A.6.2	1	Identification of risks related to external parties	none		0	
15	A.6.2	2	Addressing security when dealing with customers	none		1	
16	A.6.2	3	Addressing security in third party agreements	none		0	
17	A.7.1	0	Responsibility for assets	none		2	
18	A.7.1	1	Inventory of assets	none		0	
19	A.7.1	2	Ownership of assets	none		1	
20	A.7.1	3	Acceptable use of assets	none		0	
21	A.7.2	0	Information classification	none		1	
22	A.7.2	1	Classification guidelines	none		2	
23	A.7.2	1	Information labelling and handling	none		0	
24	A.8.1	0	Prior to employment	none		1	
25	A.8.1	1	Roles and responsibilities	none		0	
26	A.8.1	2	Screening	none		1	
27	A.8.1	3	Terms and conditions of employment	none		0	
28	A.8.2	0	During employment	none		0	
29	A.8.2	1	Management responsibilities	none		0	
30	A.8.2	2	Information security awareness, education and training	none		0	
31	A.8.2	3	Disciplinary process	none		0	

Figura 3.18 – Exemplos de métrico tabular - Fonte OSSIM (2008)

4 METODOLOGIA DE PESQUISA

4.1 TIPO DA PESQUISA

O tipo da pesquisa foi bibliográfica e experimental e quanto aos objetivos a serem atingidos, descritiva, pois descreveu as características, as configurações da ferramenta utilizada e as facilidades e dificuldades encontradas.

4.2 OBJETIVO DA PESQUISA

Tendo em vista que atualmente a correlação de eventos e a detecção de intruso têm se mostrado tarefas impossíveis de serem completadas plenamente por um analista humano sem lançar mão de diversas ferramentas e recursos, este trabalho pretende apresentar um experimento com a ferramenta Open Source Security Information Management – OSSIM, que faz uso de técnicas de mineração de dados para realizar a tarefa de correlacionamento de eventos, a fim de melhorar a detecção de intruso.

4.3 DESENVOLVIMENTO DO EXPERIMENTO

O experimento foi será realizado utilizando duas redes ethernet, uma administrativa para as estações de trabalho e a outra para o gerenciamento da ferramenta. Estas redes foram montadas com *switches* gerenciáveis. Para a proteção da rede administrativa foi utilizado um *firewall*. E para coletar os eventos do *firewall* e do tráfego de entrada e saída da rede administrativa foram utilizados dois computadores configurados com a ferramenta OSSIM. A coleta e análise dos dados foram obtidas a partir do *frontend* web da ferramenta.

5 EXPERIMENTO

5.1 APRESENTAÇÃO DO AMBIENTE

A topologia de rede utilizada para análise da ferramenta foi tipo estrela monitorada com um *server host* (SH) e um *agent host* (AH) OSSIM. As duas máquinas OSSIM foram configuradas com a imagem disponível no site do desenvolvedor (OSSIM, 2008). A diferença entre as configurações do SH e do AH são os aplicativos em execução e os detectores e monitores habilitados. O servidor executa os aplicativos *ossim-server*, *ossim-framework*, *ossim-agent* e o agente só o *ossim-agent*.

Quanto aos detectores e monitores, o SH foi configurado para armazenar e correlacionar os dados coletados pelos detectores: *p0f*, *osiris*, *arpwatch*, *snort*, *pads*, *pam_unix*, *rrd*, *sudo*, *nagios* e pelos monitores: *nmap*, *ntop*, *ossim-ca*. E o AH foi configurado para coletar os eventos de um detetor externo por meio de um *plugin* e o envio dos eventos coletados para o SH.

Para permitir a coleta do tráfego de dados de entrada e saída desta rede, uma porta do *switch* CISCO foi configurada como porta *span* da porta utilizada para a conexão externa. O tráfego da rede foi analisado pelo SH (Fig. 5.1).

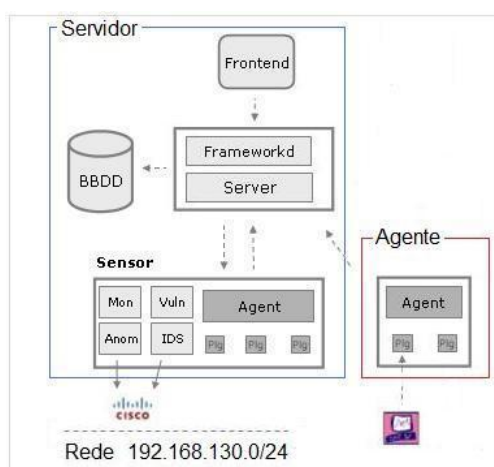


Figura 5.1 – Arquitetura das máquinas OSSIM

Onde temos:

- Rede administrativa – 192.168.130.0/24 – formada por 31 computadores, um switch Cisco de 48 portas, um firewall CheckPoint FW1, servidor de arquivo e servidor WEB;
- Rede de gerência – 192.168.0.0/24 – utilizada para o gerenciamento dos SH e AH e dos ativos de rede (switches e firewall); e
- Uma máquina de ataque.

5.2 CONFIGURAÇÃO DO AMBIENTE

Inicialmente foi realizada a instalação e a configuração do SH virtualizado em um servidor VMware ESXi 3 (VMWARE, 2008) usando a imagem do instalador (OSSIM, 2008):

- <http://downloads.alienvault.com/ossim-installer-1.0.6.iso>
- Versão: 1.0.6 Liberada:04/09/2008
- MD5: a8fa7ad0a97347dc08f3e9ef7ba75b54

Após a instalação foi necessário configurar a segunda placa de rede (eth1) alterando o arquivo */etc/network/interfaces*. Isto foi necessário para monitorar a rede por meio de outra placa de rede sem IP e deixando a primeira *interface* (eth0) somente para gerenciamento do servidor e para a visualização do *frontend* (Fig. 4.3).

Para definir a interface eth1 como sendo de monitoramento do SH, foi alterado o arquivo de configuração */etc/ossim/ossim_setup.conf*. Esta interface foi

configurada sem IP. Para tornar a esta configuração ativa foi necessário executar o script `/home/ossim/dist/reconfig.pl` (ANEXO A).

Na máquina AH foi ativado o plugin `fw1ngr60` para a coleta dos dados do firewall FW-1. Esta ativação foi realizada adicionando a linha do quadro 5.1 na seção `plugins` no arquivo `/etc/ossim/agent/config.cfg` (ANEXO B).

Quadro 5.1 – Seção `plugins` do arquivo `/etc/ossim/agent/config.cfg`

```
[plugins]
fw1ngr60=/etc/ossim/agent/plugins/fw1ngr60.cfg
```

Também foi necessário instalar o pacote `fw1-loggrabber-1.1.1-linux.tar.gz`, um programa de linha de comando para coletar os eventos do firewall FW-1, encontrado diretório: `/usr/share/ossim/www/downloads/` ou na página do fabricante do *firewall*. Para não precisar alterar o arquivo de configuração `fw1ngr60.cfg`, descompactamos o pacote no diretório: `/opt/` e alteramos os arquivos exemplos do pacote como apresentados nos quadros 5.2 e 5.3:

- `lea.conf-unix-sample` gravado como `lea.conf` – ip e porta do firewall FW1

Quadro 5.2 – Arquivo `lea.conf`

```
lea_server ip X.X.X.X # onde X.X.X.X é o IP do servidor Firewall FW-1
lea_server port 50003
```

- `fw1-loggrabber.conf-unix-sample` gravado como `fw1-loggrabber.conf`. Caso seja observado erros durante a execução do programa pode-se aumentar o valor do parâmetro `DEBUG_LEVEL` para 3 e obter mais informações de sua execução.

Quadro 5.3 – Arquivo fw1-loggrabber.conf

```
DEBUG_LEVEL="0"
FW1_OUTPUT="logs"
FW1_TYPE="ng"
FW1_MODE="normal"
ONLINE_MODE="yes"
RESOLVE_MODE="no"
SHOW_FIELDNAMES="yes"
RECORD_SEPARATOR="|"
DATEFORMAT="std"
LOGGING_CONFIGURATION="file"
OUTPUT_FILE_PREFIX="/var/log/ossim/fw-1"
OUTPUT_FILE_ROTATESIZE=1048576000
```

O firewall FW1 também foi configurado para permitir o acesso do AH sem criptografia.

Para enviar os eventos coletados pelo AH para o SH foi alterado a seção do arquivo `/etc/ossim/agent/config.cfg` no AH:

Quadro 5.4 – Seção output-server do arquivo `/etc/ossim/agent/config.cfg`

```
[output-server]
enable=True
ip=x.x.x.x #onde X.X.X.X é o ip do SH
port=40001
```

Após as alterações foi reconfigurado o pacote e reiniciado o aplicativo `ossim-agent` com o comando do quadro 5.5.

Quadro 5.5 – Reiniciando o aplicativo `ossim-agent`

```
# dpkg-reconfigure ossim-agent
# /etc/init.d/ossim-agent restart
```

A comunicação entre o AH e o SH foi verificada por meio da página WEB na guia Policy > Sensors. A figura 5.2 mostra o aviso de um novo sensor. Para permitir que o SH receba os dados é necessário adicionar o novo agente/sensor.



Figura 5.2 – Detecção de novo agente como sensor.

Para verificar o recebimento dos eventos do FW-1 obtidos com o programa fw1-loggrabber foi verificado o conteúdo do arquivo /var/log/ossim/fw-1.log no AH, com o comando apresentado no quadro 5.6.

Quadro 5.6 – Verificando o conteúdo do arquivo fw-1.log

```
# tail -f /var/log/ossim/fw-1.log
```

O próximo passo foi verificar o envio de informações do AH para o SH por meio do arquivo /var/log/ossim/agent-ossim.log (Quadro 5.7).

Quadro 5.7 – Verificando o conteúdo do arquivo agent-ossim.log

```
# tail -f /var/log/ossim/ossim-agent.log
```

Para verificar a possibilidade de criar novas diretivas de correlacionamento foi adicionado no SH uma para o worm conficker que estava disponível na página do

desenvolvedor (OSSIM, 2008). Para isto foram alterados os arquivos do diretório `/etc/ossim/server/` (ANEXO D).

Para atualizar para a última versão dos pacotes do SH OSSIM foi executado o script `/home/ossim/dist/ossim-update.pl`.

O sincronismo de tempo das máquinas deve ser implementado. O comando `ntpdate <ip do servidor de tempo>` adicionado ao `/etc/crontab` para ser executado a cada hora foi suficiente.

Para provocar eventos e verificar a capacidade de correlação da ferramenta foi configurada uma máquina com a distribuição linux BackTrack3 da Remote-Exploit.org (BACKTRACK,2009).

5.3 AVALIAÇÃO DA INSTALAÇÃO E DA CONFIGURAÇÃO

Com relação a avaliação da instalação e da configuração foi obtido o seguinte:

- a instalação realizada com o instalador é fácil e rápida para se criar uma máquina SH. Mas, para a máquina AH o instalador não ajudou, pois os pacotes não necessários na criação da máquina AH foram instalados, dificultando a configuração do agente. Um resultado melhor foi obtido instalando o pacote `ossim-agent` em uma máquina com a distribuição Debian recomendada. Também foi realizado a mesma instalação do `agent` em uma máquina com a distribuição linux Ubuntu, que funcionou corretamente. O agente instalado no Ubuntu também funcionou virtualizado.
- o uso de plugins para obter dados de outros dispositivos necessitou de um melhor entendimento do seu funcionamento e da forma com é feito a

“normalização” dos dados de cada dispositivo devido a diferenças nos dados informados em cada equipamento.

- a conexão entre o agente e o servidor funcionou normalmente quando se estava utilizando os plugins já disponíveis e configurados na instalação. Após ter sido habilitado o plugin `fw1ngr60.cfg`, o envio de dados não ocorreu, embora o AH estivesse coletando os dados do firewall. Para resolver o problema foi necessário habilitar a execução dos programas `ossim-server` do SH e o `ossim-agent` no AH em modo `DEBUG`, a fim de obter mais detalhes sobre a falha na comunicação. A solução foi executar o programa `fw1-loggrabber` fora do controle do `ossim-agent`.
- para realizar as configurações disponíveis somente no console é necessário um conhecimento intermediário sobre a distribuição linux `debian` que é o sistema operacional do instalador `OSSIM`.

Com relação a atualização dos programas instalados nas máquinas, o programa `/home/ossim/dist/ossim-update.pl` funcionou bem, mas foi observado perda da conexão entre o agente e o servidor quando a versão dos pacote `ossim-agent` instalado no AH foi atualizada que apresentou uma incompatibilidade no formato dos dados transmitidos pelo agente ao servidor. Foi feita a reinstalação da versão `1:0.9.9-4`.

5.4 DADOS COLETADOS E CORRELACIONADOS

Com as máquinas conectadas e funcionando foi iniciada a observação dos dados coletados, da correlação e dos alarmes.

Primeiro ponto observado foi a quantidade de informação disponível em um mesmo lugar. Com o inventário das máquinas realizado no modo automático, passamos a verificar as vulnerabilidades encontradas com o *nessus* e posteriormente anomalias devido a troca de endereço IP para um endereço MAC conhecido (98% destas informações estavam relacionadas a máquinas em reparo e 2% a alterações não autorizadas). Também foram obtidos: gráficos de intensidade de tráfego feitos com os dados coletados pelos sensores; estatísticas obtidas pelo *ntop*; mapas de disponibilidades das máquinas cadastradas com o *nagios*.

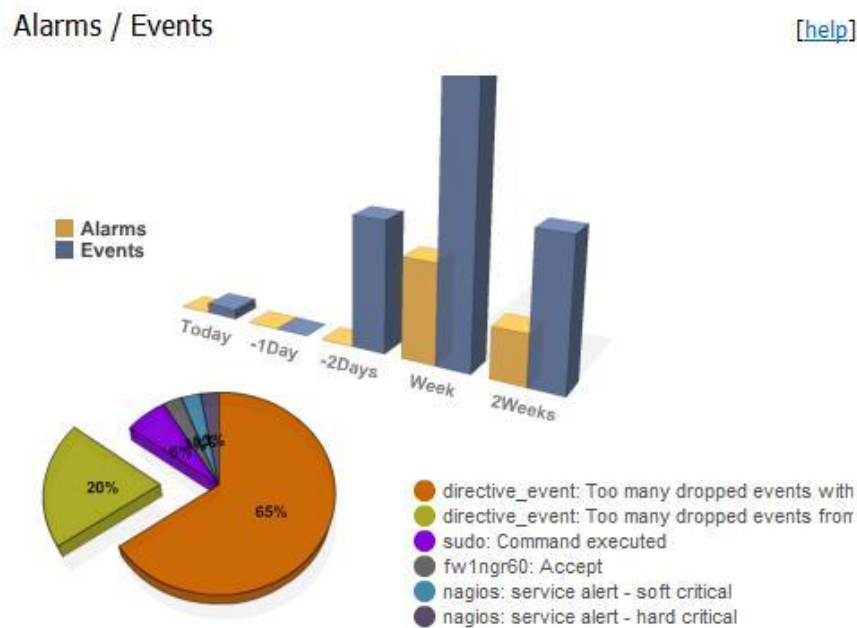


Figura 5.3 – Gráfico de volume de eventos e alarmes.

Outra facilidade observada foi a simplicidade para realizar varredura em máquinas com o *nmap* e o *nessus*.

Outro item observado foi a existência de um visualizador das regras do snort instaladas que permite ao administrador identificar quais as regras utilizadas e bem como a necessidade de sua atualização (Fig. 5.4).

p2p.rules									
Name	Action	Protocol	SRC IP	SRC Ports	Dir	DEST IP	DEST Ports	Content	Options
"P2P napster login"	alert	tcp	\$HOME_NET	any	->	\$EXTERNAL_NET	8888	"[00 02 00]"	offset: 1 depth: 3 rev: 8 classtype: policy-violation flow: to_server,established
"P2P napster new user login"	alert	tcp	\$HOME_NET	any	->	\$EXTERNAL_NET	8888	"[00 06 00]"	offset: 1 depth: 3 rev: 8 classtype: policy-violation flow: to_server,established
"P2P napster download attempt"	alert	tcp	\$EXTERNAL_NET	any	->	\$HOME_NET	8888	"[00 CB 00]"	offset: 1 depth: 3 rev: 7 classtype: policy-violation flow: to_server,established

Figura 5.4 – Regras do snort – visualização de regra p2p

Durante a avaliação foi observado que a ferramenta coleta e analisa todos os eventos de conexão entre o AH e SH e eventos internos (pam_unix e ssh) gerando grande volume de informações. Para um ambiente de produção seria recomendável não coletar estes eventos.

Com relação a avaliação da correlação dos dados foi observado que ela ocorre agrupando vários alarmes a cada nível de correlação elevando o valor do risco. Isto facilitou a análise de vários eventos relacionados destacados pela elevação do risco (Fig. 5.5).

#	Id	Alarm	Risk	Date	Source	Destination	Correlation Level
1	402	TCP Portscan against ossim	5	2009-04-04 10:38:26	192.168.0.101:ANY	ossim:ANY	2
Alarm Summary [Total Events: 2 - Unique Dst IPAddr: 1 - Unique Types: 2 - Unique Dst Ports: 1]							
1	401	portscan: Open Port	0	2009-04-04 10:38:14	192.168.0.101:ANY	ossim:ANY	2
2	400	portscan: TCP Portscan	0	2009-04-04 10:38:14	192.168.0.101:ANY	ossim:ANY	1
Total events matched after highest rule level, before timeout: 1							
Back to main							
#	Id	Alarm	Risk	Date	Source	Destination	Correlation Level
1	7396	Too many dropped events with destination 10.5.143.255	4	2009-05-01 18:13:04	10.5.131.89:netbios-ns	10.5.143.255:netbios-ns	4
Alarm Summary [Total Events: 10 - Unique Dst IPAddr: 1 - Unique Types: 1 - Unique Dst Ports: 2]							
2	7375	Too many dropped events with destination 10.5.143.255	3	2009-05-01 18:13:03	10.5.131.89:netbios-ns	10.5.143.255:netbios-ns	3
Alarm Summary [Total Events: 5 - Unique Dst IPAddr: 1 - Unique Types: 1 - Unique Dst Ports: 2]							
3	7367	Too many dropped events with destination 10.5.143.255	2	2009-05-01 18:13:02	10.5.131.89:netbios-ns	10.5.143.255:netbios-ns	2
Alarm Summary [Total Events: 4 - Unique Dst IPAddr: 1 - Unique Types: 1 - Unique Dst Ports: 2]							
Total events matched after highest rule level, before timeout: 3							

Figura 5.5 – Exemplos de correlação com elevação do risco

Outro gráfico interessante é o que indica o nível de alarmes não processados (Fig. 5.6). Quando foi habilitado o recebimento e o envio dos eventos do *firewall* simulando uma rede de 300 máquinas o valor de 100 foi atingido no período de uma noite.

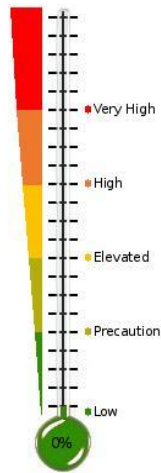


Figura 5.6 – Termômetro de serviço

6 CONCLUSÃO

Este trabalho apresentou a análise da ferramenta OSSIM para o correlacionamento de eventos fornecidos por vários programas, com enfoque nas facilidades e dificuldades de uso e na aperfeiçoamento da detecção de intruso.

Com relação às facilidades observadas pelo uso da ferramenta OSSIM temos a correlação de eventos gerados por várias ferramentas *open source* e o gerenciamento integrado destas mesmas ferramentas. Em redes de computadores que já fazem uso de alguma destas ferramentas, o uso do OSSIM trará um incremento das informações para o gerenciamento da rede por permitir correlacionamento de dados de outras ferramentas com as que já estão em uso. E em redes onde não se possui ferramentas instaladas para monitoramento, o OSSIM permitirá a criação de uma gerencia de uma forma prática e simples.

Com relação a capacidade de correlacionar eventos do SH foi observado que a arquitetura de teste empregada não obteve o desempenho esperado com taxas de envio de 6 eventos por segundo pelo agente. Para melhorar este desempenho seria necessário a criação de níveis intermediários de correlacionamento e separar o servidor de banco de dados a fim de diminuir a concentração de eventos em um mesmo servidor.

As dificuldades para o uso da ferramenta foram a configuração da arquitetura da rede de gerenciamento somente em modo texto do linux e o fato do instalador não permitir definição do tipo da instalação de acordo com o papel da máquina tais como: agente, servidor de correlacionamento, servidor intermediário, *frontend* e banco de dados.

Também foi observado a ausência de um editor de diretivas de correlação e de um editor de plugins que permitisse criar e alterar as “expressões regulares” utilizadas dentro do plugin para obter os dados de interesse de cada evento coletado por plugins. Para a edição de “expressões regulares” foi utilizada a ferramenta *kodos* disponível para os sistemas operacionais windows e linux. No linux utiliza-se o comando `#apt-get install kodos`.

Assim concluo que a ferramenta pode ser excelente para iniciar um gerenciamento de uma rede a fim de obter informações sobre o funcionamento e orientar as decisões do administrador. Caso seja do interesse do administrador da rede, o desenvolvedor do sistema comercializa *hardware* dedicado para o OSSIM, bem como treinamento, certificação e suporte.

6.1 TRABALHOS FUTUROS

Algumas propostas surgiram no decorrer deste projeto, as quais são agora apresentadas como possíveis caminhos de pesquisa:

- Avaliar as soluções comerciais para monitoramento de rede de computadores que realizam correlação de eventos com fluxos de 1Gbs.
- Avaliar um modelo que defina o melhor posicionamento de pontos de coleta de eventos e o tipo de eventos coletados em cada ponto, de forma a otimizar o correlacionamento e as informações para o gerenciamento de uma rede, bem como o número de níveis de correlacionamento.
- Avaliar quais as representações gráficas das ferramentas ajudam a diminuir o tempo de resposta a incidente em uma rede.

REFERÊNCIAS

- ABAD C.; TAYLOR J. et al. **Log Correlation for Intrusion Detection**: A Proof of Concept. Proc. of the 19th Annual Computer Security Applications Conf. ACSAC, 2003. Disponível em: <http://citeseer.ist.psu.edu/abad03log.html>. Acessado em 30 mai. 2008.
- ALLEN, W. **Mixing Wheat with the Chaff**: Creating Useful Test Data for IDS Evaluation. Security & Privacy, IEEE, 2007, v. 5, p. 65-67.
- BACKTRACK3. **Distribuição linux focada em teste de penetração**. Disponível em: <http://www.remote-exploit.org/backtrack.html>. Acessada em out de 2008.
- CAMPELLO, R.; WEBER, R. Minicurso: **Sistemas de Detecção de Intrusão**. XIX Simpósio Brasileiro de Redes de Computadores, SBRC, 2001, Florianópolis: UFSC, 2001.
- CARVALHO FILHO, B. P.; MONTES FILHO, A. **Detecção de Intrusão em Redes de alta Velocidade**. In: WORKSHOP DOS CURSOS DE COMPUTAÇÃO APLICADA DO INPE, 4. (WORCAP), 2004, São José dos Campos. **Anais...** São José dos Campos: INPE, 2004. CD-ROM, On-line. Disponível em: <http://urlib.net/lac.inpe.br/worcap/2004/10.05.19.05>. Acesso em: 15 jun. 2008.
- CASAL, J.; KARG, D. **Open Source Security Information Management (OSSIM)**. Disponível em <http://www.ossim.com/> Acesso em jan. de 2009.
- CISCO. **Cisco Information Technology at Work Case Study**: Cisco IOS NetFlow Technology, 2004. Disponível em: http://www.cisco.com/en/US/tech/tk812/tech_briefs_list.html. Acessado em 30 mai. 2008.
- CORDEIRO, W; VASCONCELLOS, V. et al. **Uma Proposta para a Identificação de Atividades Maliciosas em Arquivos de Log**. XI Workshop de Gerência e Operação de Redes e Serviços, 2006, Curitiba.
- FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P.; UTHURUSAMY, R.. **Advances in knowledge discovery and data mining**. Cambridge, MA, USA: MIT Press, 1996. ISBN 0262560976.
- GRÉGIO, A. R. A. **Aplicação de Técnicas de Data Mining para a Análise de Logs de Tráfego TCP/IP**. Dissertação de Mestrado em Computação Aplicada, orientada pelo Prof. Dr. Antonio Montes Filho e pelo Prof. Dr. Rafael Duarte Coelho dos Santos, apresentada em 27 de fevereiro de 2007.
- HAINES, J.; RYDER K. et al.. **Validation of sensor alert correlators**. Security & Privacy, IEEE, 2003, v. 1,p. 46-56
- MELLO, T. E. B; HEXSEL, R. A. **Correlacionamento Distribuído de Alertas em Sistemas de Detecção de Intrusão**. Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, 2005, Florianópolis, p. 1-14

MILITELLI, L. C. **Proposta de um agente de aplicação para detecção, prevenção e contenção de ataques em ambientes computacionais**. Orientador: João Antônio Zuffo. São Paulo, 2007. Dissertação. (Mestrado em Engenharia Elétrica) - Escola Politécnica, 2007

MULLINS, D. et al. **Network Intrusion Detection: Automated and Manual Methods Prone to Attack and Evasion**. Security & Privacy, IEEE, 2006, v. 4, p. 36-43.

NEMETH, Evi; HEIN, Trent R.; SNYDER, Garth. **Manual completo do Linux: guia do administrador**. 2. ed. São Paulo: Pearson Prentice Hall, 2007

OSSIM. **Open Source Security Information Management**. Disponível em: <<http://www.ossim.net/home.php>>. Acesso em 15 set. de 2008.

PIATETSKY-SHAPIRO, G. **From data mining to knowledge discovery: an introduction**. Oct 2003. Disponível em: <http://www.kdnuggets.com/dmcourse/other_lectures/data-mining-to-knowledge-discovery.ppt>. Acesso em jan. de 2007.

RAMOS, G. R. ; A. M. C. & T. A. S.. **Monitoramento e Defesa do Perímetro de Redes TCP/IP Utilizando o NetFlow**. VI SIGE - Simpósio de Guerra Eletrônica 2004. São José dos Campos, 29 de novembro a 01 de dezembro de 2004.

SCHMIDT, K. J. **Threat analysis using log data**. INSECURE Magazine, n. 5, p. 17–26, Jan 2006. Disponível em: <<http://www.insecuremag.com>>. Acessado em 14 jun. 2008.

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO. **Manual para Elaboração e Normalização de Dissertações e Teses**. 3.ed. Rio de Janeiro: Sistema de Bibliotecas e Informação – SiBI, 2008. Disponível em: <http://www.sibi.ufrj.br/manual_teses.pdf>. Acessado em 21 dez. 2008.

U.S.A. White House. **The National Strategy to Secure Cyberspace** [Estratégia Nacional para Segurança do Ciberespaço], U.S. Executive Office of the President, 2003, 60 p. Disponível em: <http://www.whitehouse.gov/pcipb/>. Acessado em 14 jun. 2008.

VAZ, T.; BORTOLETTO, T.; CAMÕES, G. A. **Sistemas de Detecção de Intrusão Livres: suas limitações e uma arquitetura proposta sobre concentração de mensagens e correlacionamento de eventos**. IV ERBASE - IV Escola Regional de Computação Bahia-Sergipe, 2004.

VMWARE. **Sistema Operacional Vmware ESXi 3: Virtualização de servidores**. Disponível em: <<http://www.vmware.com>>. Acessado em 14 jun. 2008.

ANEXO A - Arquivos de configuração do servidor OSSIM

Quadro A.1 – Arquivo de configuração do OSSIM

```
#
# /etc/ossim/ossim_setup.conf
#
interface=eth0
language=en
profile=all-in-one

[database]
acl_db=ossim_acl
db_ip=
db_port=3306
event_db=snort
ossim_db=ossim
osvdb_db=osvdb
pass=
type=mysql
user=root

[expert]
profile=server

[sensor]
detectors=snare, p0f, osiris, arwatch, snort, pads, ssh, pam_unix, rrd,
sudo, iptables
interfaces=eth1
ip=
monitors=nmap, ping, ntop, ossim-ca
name=ossim
priority=5

[server]
server_ip=
server_port=40001
server_plugins=osiris, pam_unix, ssh, snare, sudo
```

Descrição dos parâmetros:

- **interface:** A placa de rede principal (administração). O endereço IP desta placa de rede será utilizado para o acesso ao OSSIM.
- **language:** Define a linguagem do framework. Desabilitada na versão 1.0.
- **profile:** Sensor, server, etc..Desabilitada na versão 1.0.
- **Database:** Informações do banco de dados que serão utilizadas.
 - pass: Se em branco, uma senha randômica será gerada.

- `db_ip`: Se deixado em branco, o endereço IP da interface principal será usado.
- **Sensor**. Opções dos Sensores.
 - `detectors`: quais detectores serão habilitados.
 - `interfaces`: Mais do que uma placa de rede poderá ser especificada, devendo estarem separadas por vírgula. Todas as placas de rede devem estar ativas para o bom funcionamento.
 - `ip`: Se deixado em branco, o endereço IP da placa de rede principal será usado. Vários processos associados a esta placa de rede serão iniciados separadamente para cada sensor ativo: `arpwatch`, `pads`, `p0f`, `snort`.
 - `monitors`: quais monitores serão habilitados
 - `name`: nome do sensor. Inserido também no banco de dados.
 - `priority`: prioridade do sensor que será utilizado.

Após qualquer alteração neste arquivo é necessário executar o script:

```
/home/ossim/dist/reconfig.pl
```

ANEXO B - Configuração do plugin para o FireWall1 NG R65

Quadro B.1 – Arquivo de configuração fw1ngr60.cfg

```
;; Firewall-1 NG R60 (fw1ngr60)
;; plugin_id: 1504
;;
;; $Id $
;;

[DEFAULT]
plugin_id=1504

[config]
type=detector
enable=yes
source=log
location=/var/log/ossim/fw-1.log
# create log file if it does not exists,
# otherwise stop processing this plugin
create_file=false
#This process is started by fw1-loggrabber
process=fw1-loggrabber
start=yes
stop=yes
startup=/opt/fw1-loggrabber-1.11.1-linux/fw1-loggrabber -c /opt/fw1-
loggrabber-1.11.1-linux/fw1-loggrabber.conf -l /opt/fw1-loggrabber-1.11.1-
linux/lea.conf
shutdown=kill -9 `pidof fw1-loggrabber`
# use {translate($n)} for translations
[translation]
ACCEPT=1
REJECT=2
DROP=3
DENY=3
Inbound=4
Outbound=5
__DEFAULT__=99

[fw1ngr60-generic]
event_type=event
regexp=loc=\d+\\|time=(\d+\\-\d+\\-
\d+\\s+\d+:\d+:\d+)\\|action=(\w+)\\|orig=(\d+\\.\\d+\\.\\d+\\.\\d+)\\|i\\f_dir=(\w+
)\\|i\\f_name=(\\w\
-
)+\\|has_accounting=(\d+)\\|uid=<([\\S,]+)>\\|product=([a-zA-Z0-9\
-
&s]+)\\|__policy_id_tag=product=[a-zA-Z0-9\
-
&s]+\\[db_tag=\\{([\\S\
-
]+)\\};mgmt=(\w+);date=(\d+);policy_name=(\w+)\\|src=(\d+\\.\\d+\\.\\d+\\.\\d+)\\
|s_port=(\d+)\\|dst=(\d+\\.\\d+\\.\\d+\\.\\d+)\\|service=(\d+)\\|proto=(\w+)\\|rule=
(\d+)
date=${1}
plugin_sid={translate($2)}
src_ip=${13}
dst_ip=${15}
protocol=${17}
src_port=${14}
dst_port=${16}
userdata4=Action: {$2}, firewall: {$3}, policy_name: {$12}, rule: {$18},
if_name: {$5}, product: {$8}, uid: {$7}
```

Para receber os eventos do FireWall-1 NG R65 foi utilizado o arquivo `/etc/ossim/agent/plugins/fw1ngr60.cfg` que é apresentado no quadro a seguir

ANEXO C - Arquivos de configuração do agente OSSIM

Para criar uma máquina agente OSSIM pode-se instalar o pacote `ossim-agent` na distribuição linux Ubuntu

Quadro C.1 – Comando para instalação do agente

```
# apt-get install ossim-agent
```

ou podemos utilizar o instalador e desabilitar os outros serviços do OSSIM (`ossim-framework`, `ossim-server` e `ossim-mysql`)

Após a instalação altere o arquivo `/etc/ossim/agent/config.cfg`. Neste arquivo você pode:

- definir o modo de execução do agente
- definir os arquivos de logs
- alterar o endereço do servidor OSSIM
- ativar ou desativar *watchdog*
- ativar ou desativar *plugins*
- configurar *plugins*

O arquivo foi alterado da seguinte forma:

- o IP do sensor foi atribuído a interface `eth0`
- o IP da interface de gerencia foi atribuído a interface `eth1`
- para enviar os dados para o servidor OSSIM foi alterada a seção

```
[output-server]
enable = True
ip = 192.168.0.X
port = 40001
```

- para enviar os dados coletados para o banco de dados do servidor OSSIM deve ser alterada a seção

```
[output-db]
enable = True
type = mysql
host = <IP do servidor>
base = ossim_events
user = root
pass = suasenha
```

- para escolher os plugins que serão executados deve-se retirar o `#` do início da linha

```
[plugins]
arpwatch=/etc/ossim/agent/plugins/arpwatch.cfg
#cisco-ids=/etc/ossim/agent/plugins/cisco-ids.cfg
```

```
#cisco-pix=/etc/ossim/agent/plugins/cisco-pix.cfg
#cisco-router=/etc/ossim/agent/plugins/cisco-router.cfg
#cisco-vpn=/etc/ossim/agent/plugins/cisco-vpn.cfg
fw1ngr60=/etc/ossim/agent/plugins/fw1ngr60.cfg
ossim-agent=/etc/ossim/agent/plugins/ossim-agent.cfg
p0f=/etc/ossim/agent/plugins/p0f.cfg
pads=/etc/ossim/agent/plugins/pads.cfg
```

- pode-se escolher também monitores

```
[monitor]
nmap=/etc/ossim/agent/plugins/nmap-monitor.cfg
ntop=/etc/ossim/agent/plugins/ntop-monitor.cfg
ossim-ca=/etc/ossim/agent/plugins/ossim-monitor.cfg
ping=/etc/ossim/agent/plugins/ping-monitor.cfg
```

No quadro C.2, segue o arquivo de configuração original.

Quadro C.2 – Arquivo de configuração do agente

```
# /etc/ossim/agent/config.cfg
# ossim agent configuration file
# general
#
[daemon]
daemon = True
pid = /var/run/ossim-agent.pid

[log]
file = /var/log/ossim/agent.log
error = /var/log/ossim/agent_error.log
stats = /var/log/ossim/agent_stats.log

# uncomment this line if you want to log to a remote syslog
# remember you need to startup syslog with the -r option to enable
# the facility to receive messages from the network
#syslog = localhost

# verbose level
# these are the levels allowed by agent:
# debug, info, warning, error and critical
# you can use the -v and -vv command line
# arguments to increase this verbose level
verbose = info

# default values, can be overridden in each plugin rule
[plugin-defaults]
sensor = 127.0.0.1
interface = any
date_format = %Y-%m-%d %H:%M:%S ; format, not date itself
ossim_dsn=mysql:localhost:ossim:root:yoursecretpassword
tzone=0
```

Continuação do Quadro C.2 – Arquivo de configuração do agente

```
[watchdog]
enable = True
interval = 30          ; seconds between checks
restart_interval=3600 ; seconds between plugin process restart

# Consolidate events captured within "time" seconds into a single one
[event-consolidation]
enable = False

# time for the event consolidation checks
# Be careful increasing this time interval as the events are stored in
memory
# In addition, notice that many correlation rules need a short time
between
# event matching so server correlation will not work as expected with a
# high "time" value
time = 10

# consolidation filters
# If there are more than one filter, they are managed as OR rules
by_plugin = 1001-1150,1501-1550,4001-4010
#by_src_ip = 127.0.0.1
#by_dst_ip = 127.0.0.1
#by_sensor = 127.0.0.1

# output
#
# write events into a plain file (server injectable)
[output-plain]
enable = False
file = /var/log/ossim/agent-plain.log

# send events and receive/send control messages from/to server
[output-server]
enable = True
ip = 127.0.0.1
port = 40001
send_events = True

[output-server-pro]
enable = False
ip = 127.0.0.1
port = 40001

# plugins
# (list of plugins to be loaded)
```

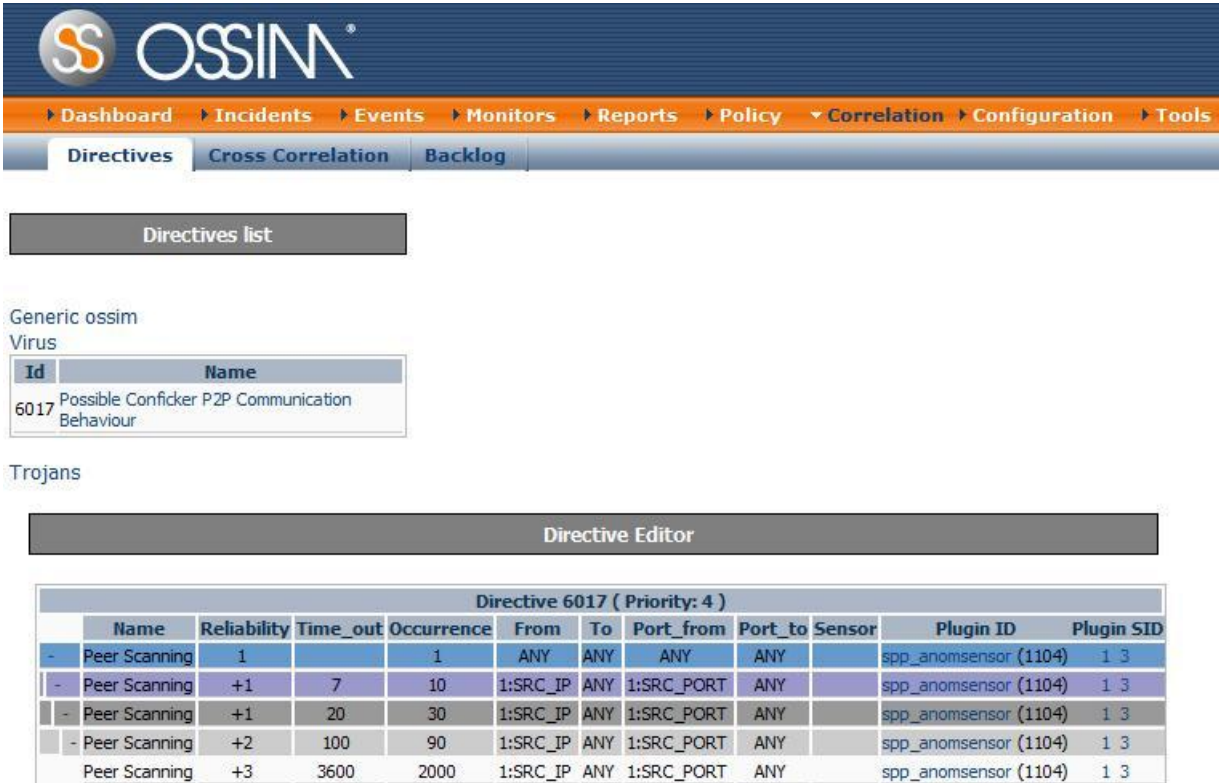
Continuação do Quadro C.2 – Arquivo de configuração do agente

```
[plugins]
# detectors
#
#apache=/etc/ossim/agent/plugins/apache.cfg
arpwatch=/etc/ossim/agent/plugins/arpwatch.cfg
#cisco-ids=/etc/ossim/agent/plugins/cisco-ids.cfg
#cisco-pix=/etc/ossim/agent/plugins/cisco-pix.cfg
#cisco-router=/etc/ossim/agent/plugins/cisco-router.cfg
#cisco-vpn=/etc/ossim/agent/plugins/cisco-vpn.cfg
#clamav=/etc/ossim/agent/plugins/clamav.cfg
#clurgmgr=/etc/ossim/agent/plugins/clurgmgr.cfg
#fwlngr60=/etc/ossim/agent/plugins/fwlngr60.cfg
#gfi=/etc/ossim/agent/plugins/gfi.cfg
#heartbeat=/etc/ossim/agent/plugins/heartbeat.cfg
#iis=/etc/ossim/agent/plugins/iis.cfg
#intrushield=/etc/ossim/agent/plugins/intrushield.cfg
#iphone=/etc/ossim/agent/plugins/iphone.cfg
#iptables=/etc/ossim/agent/plugins/iptables.cfg
#mwcollect=/etc/ossim/agent/plugins/mwcollect.cfg
#nagios=/etc/ossim/agent/plugins/nagios.cfg
#netgear=/etc/ossim/agent/plugins/netgear.cfg
#netscreen-manager=/etc/ossim/agent/plugins/netscreen-manager.cfg
#netscreen-firewall=/etc/ossim/agent/plugins/netscreen-firewall.cfg
#ntsyslog=/etc/ossim/agent/plugins/ntsyslog.cfg
#osiris=/etc/ossim/agent/plugins/osiris.cfg
ossec=/etc/ossim/agent/plugins/ossec.cfg
ossim-agent=/etc/ossim/agent/plugins/ossim-agent.cfg
p0f=/etc/ossim/agent/plugins/p0f.cfg
pads=/etc/ossim/agent/plugins/pads.cfg
pam_unix=/etc/ossim/agent/plugins/pam_unix.cfg
#postfix=/etc/ossim/agent/plugins/postfix.cfg
#realsecure=/etc/ossim/agent/plugins/realsecure.cfg
rrd=/etc/ossim/agent/plugins/rrd.cfg
snort=/etc/ossim/agent/plugins/snort.cfg
snortunified=/etc/ossim/agent/plugins/snortunified.cfg
#spamassassin=/etc/ossim/agent/plugins/spamassassin.cfg
#squid=/etc/ossim/agent/plugins/squid.cfg
#symantec-ams=/etc/ossim/agent/plugins/symantec-ams.cfg
ssh=/etc/ossim/agent/plugins/ssh.cfg
#stonegate=/etc/ossim/agent/plugins/stonegate.cfg
sudo=/etc/ossim/agent/plugins/sudo.cfg
#syslog=/etc/ossim/agent/plugins/syslog.cfg
#snare=/etc/ossim/agent/plugins/snare.cfg
#tarantella=/etc/ossim/agent/plugins/tarantella.cfg

# monitors
#
nmap=/etc/ossim/agent/plugins/nmap-monitor.cfg
ntop=/etc/ossim/agent/plugins/ntop-monitor.cfg
#opennms=/etc/ossim/agent/plugins/opennms-monitor.cfg
ossim-ca=/etc/ossim/agent/plugins/ossim-monitor.cfg
ping=/etc/ossim/agent/plugins/ping-monitor.cfg
#tcptrack=/etc/ossim/agent/plugins/tcptrack-monitor.cfg
#nessus=/etc/ossim/agent/plugins/nessus-monitor.cfg
```


ANEXO D – Arquivo de diretiva de correlação virus.xml

Pelo *frontend* do SH OSSIM foi acessado a página *Correlation -> Directives* e verificado que só existiam dois grupos na lista de diretivas de correlação: Generic ossim e Trojans. Assim um novo grupo “Virus” de diretivas de correlação foi criado seguindo as instruções disponíveis na página do *frontend* e foi associada a esse novo grupo, a diretiva para o worm conficker. (Fig.D.1)



The screenshot shows the OSSIM web interface. At the top, there is a navigation bar with the following items: Dashboard, Incidents, Events, Monitors, Reports, Policy, Correlation, Configuration, and Tools. Below this, there are tabs for Directives, Cross Correlation, and Backlog. The main content area is titled "Directives list" and shows two groups: "Generic ossim" and "Virus". Under the "Virus" group, there is a table with the following data:

Id	Name
6017	Possible Conficker P2P Communication Behaviour

Below the "Virus" group, there is a section for "Trojans" and a "Directive Editor" for directive 6017 (Priority: 4). The editor shows a table with the following data:

Directive 6017 (Priority: 4)											
	Name	Reliability	Time_out	Occurrence	From	To	Port_from	Port_to	Sensor	Plugin ID	Plugin SID
-	Peer Scanning	1		1	ANY	ANY	ANY	ANY		spp_anomsensor (1104)	1 3
-	Peer Scanning	+1	7	10	1:SRC_IP	ANY	1:SRC_PORT	ANY		spp_anomsensor (1104)	1 3
-	Peer Scanning	+1	20	30	1:SRC_IP	ANY	1:SRC_PORT	ANY		spp_anomsensor (1104)	1 3
-	Peer Scanning	+2	100	90	1:SRC_IP	ANY	1:SRC_PORT	ANY		spp_anomsensor (1104)	1 3
	Peer Scanning	+3	3600	2000	1:SRC_IP	ANY	1:SRC_PORT	ANY		spp_anomsensor (1104)	1 3

Figura D.1– Visualização das Diretivas de correlação para o vírus conficker

Para isso foi alterado o arquivo `/etc/ossim/server/directives.xml` adicionando as linhas em negrito do quadro D.1.

Quadro D.1– Arquivo directives.xml

```

<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE directives
  SYSTEM '/etc/ossim/server/directives.dtd'
  [
    <!ENTITY generic SYSTEM '/etc/ossim/server/generic.xml'>
    <!ENTITY trojans SYSTEM '/etc/ossim/server/trojans.xml'>
    <!ENTITY virus SYSTEM '/etc/ossim/server/virus.xml'>
  ]>

<directives>
  &generic;
  &trojans;
  &virus;

  <groups>
    <group name="GroupTest1">
      <append-directive directive_id="1"/>
    </group>
  </groups>

</directives>

```

Também foi adicionado o arquivo virus.xml com a nova diretiva para o worm conficker no diretório /etc/ossim/server/, onde as tags XML tem o seguinte significado (Quadro D.2):

- para a diretiva temos as seguintes tags
 - **id** – para identificar a diretiva, possuindo valor numérico e único.
 - **name** – descrição da diretiva que será mostrado nos relatórios de alertas.
 - **priority** – importância da diretiva.
- para as regras
 - **type** – pode ser “detector” ou “monitor”. As regras com tipo Detector são recebidas automaticamente do agente onde elas estão gravadas (snort, spade, apache, etc.). As regras tipo monitor devem ser solicitadas pelo servidor.
 - **name** – nome dado a regra ex.: "Peer Scanning".

- **priority** – prioridade usada dentro da diretiva, só usada na primeira regra que combinada com os dados dos hosts e da política qualificará o ataque. O valor 0 indica que pode ser ignorado e quanto maior o valor maior a importância.
- **Reliability** – cada regra tem sua própria e é importante para a correlação. A medida que o processo de correlação é realizado este valor é somado aos de cada regra verificada. Sua ideia é ser uma medida de probabilidade em tempo real, onde o seu valor pode ser de 0 até 10 e representaria a percentagem de não ser um falso positivo.
- **time_out** – tempo em segundos para a expiração da regra.
- **occurrence** – quantas vezes a regra deve ocorrer antes de avançar para a próxima regra. Combinada com um `time_out` grande pode ser utilizado para detecção de varreduras lentas usando o detector de anomalia *spade*.
- **from** – ip de origem. Valores possíveis: ANY, X.X.X.X(IPV4) e referenciada (1:SRC_IP – onde 1 é a posição da regra e SRC_IP ou DST_IP são ip de origem e de destino da regra referenciada).
- **to** – ip de destino. Valores possíveis: ANY, X.X.X.X(IPV4) e referenciada (1:SRC_IP – onde 1 é a posição da regra e SRC_IP ou DST_IP são ip de origem e de destino da regra referenciada).
- **port_from** – porta de origem ou seqüência de portas separadas por vírgula. Também pode-se usar ANY e referenciada (1:SRC_IP – onde 1 é a posição da regra e SRC_IP ou DST_IP são ip de origem e de destino da regra referenciada).

- **port_to** – porta de destino ou seqüência de portas separadas por vírgula. Também pode-se usar ANY e referenciada (1:SRC_IP – onde 1 é a posição da regra e SRC_IP ou DST_IP são ip de origem e de destino da regra referenciada).
- **plugin_id** – identifica o plugin. Ex.:"1104" é o *spade*
- **plugin_sid** – são valores associados aos eventos gerados pelos plugins. Ex.: 1 representa e o 3 representa. Os valores já cadastrados podem ser consultados e alterados na página configuration>plugins.
- **condition** – este parâmetro e os três seguintes são válidos para type=monitor e alguns detectores. Valores possíveis: eq – igual, ne – diferente, lt - menor, gt - maior, le - menor ou igual e ge – maior ou igual.
- **value** – o valor que deve ser comparado logicamente
- **interval** – similar ao time_out usado em regras type=monitor
- **absolute** – valor *true* ou *false*. Se for verdadeiro a condição do valor deve ocorrer dentro do intervalo. Se for falso, o valor dentro do tempo deve ocorrer de acordo com a condição
- **sticky**="true"
- **sticky_different**="DST_IP"
- **protocol** – TCP,UDP, ICMP, ARP

Quadro D.2– Arquivo virus.xml

```

<directive id="6017" name="Possible Conficker P2P Communication Behaviour"
priority="4">

  <rule type="detector" name="Peer Scanning" reliability="1" occurrence="1"
    from="ANY" to="ANY" port_from="ANY" port_to="ANY" plugin_id="1104"
    plugin_sid="1,3" protocol="UDP">

<rules>
  <rule type="detector" name="Peer Scanning" reliability="+1" time_out="7"
    occurrence="10" from="1:SRC_IP" to="ANY" port_from="1:SRC_PORT"
    port_to="ANY" plugin_id="1104" plugin_sid="1,3" sticky="true"
    sticky_different="DST_IP" protocol="UDP">

<rules>
  <rule type="detector" name="Peer Scanning" reliability="+1" time_out="20"
    occurrence="30" from="1:SRC_IP" to="ANY" port_from="1:SRC_PORT"
    port_to="ANY" plugin_id="1104" plugin_sid="1,3" sticky="true"
    sticky_different="DST_IP" protocol="UDP">

<rules>
  <rule type="detector" name="Peer Scanning" reliability="+2" time_out="100"
    occurrence="90" from="1:SRC_IP" to="ANY" port_from="1:SRC_PORT"
    port_to="ANY" plugin_id="1104" plugin_sid="1,3" sticky="true"
    sticky_different="DST_IP" protocol="UDP">

<rules>
  <rule type="detector" name="Peer Scanning" reliability="+3" time_out="3600"
    occurrence="2000" from="1:SRC_IP" to="ANY" port_from="1:SRC_PORT"
    port_to="ANY" plugin_id="1104" plugin_sid="1,3" sticky="true"
    sticky_different="DST_IP" protocol="UDP"/>
</rules>
  </rule>
</rules>
  </rule>
</rules>
  </rule>
</rules>
  </rule>
</directive>

```