Universidade Estadual de Londrina

EDUARDO ALVES MORAES

APLICAÇÃO DE APRENDIZADO DE MÁQUINA SUPERVISIONADO E TÉCNICAS DE CORRELAÇÃO NA ANÁLISE DE ALERTAS DE INTRUSÃO

Londrina
2018

EDUARDO ALVES MORAES

APLICAÇÃO DE APRENDIZADO DE MÁQUINA SUPERVISIONADO E TÉCNICAS DE CORRELAÇÃO NA ANÁLISE DE ALERTAS DE INTRUSÃO

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Bruno Bogaz Zarpelão

Londrina
2018

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

Moraes, Eduardo Alves.
Aplicação de Aprendizado de Máquina Supervisionado e Técnicas de Correlação na Análise de Alertas de Intrusão / Eduardo Alves Moraes. - Londrina, 2018.
62 f. : il.

Orientador: Bruno Bogaz Zarpelão.
Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2018.
Inclui bibliografia.

1. Sistema de detecção de intrusão - Tese. 2. Aprendizado de máquina supervisionado - Tese. 3. Redução de alertas falsos positivos - Tese. 4. Correlação de alertas de intrusão e clusterização - Tese. I. Zarpelão, Bruno Bogaz. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. III. Título.

EDUARDO ALVES MORAES

APLICAÇÃO DE APRENDIZADO DE MÁQUINA SUPERVISIONADO E TÉCNICAS DE CORRELAÇÃO NA ANÁLISE DE ALERTAS DE INTRUSÃO

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

_____

Orientador: Prof. Dr. Bruno Bogaz Zarpelão
Universidade Estadual de Londrina – UEL

_____

Prof. Dr. Rodolfo Miranda de Barros
Universidade Estadual de Londrina – UEL

_____

Prof. Dr. Alexandre de Aguiar Amaral
Instituto Federal Catarinense, Campus Camboriú – IFSC

Londrina, 11 de setembro de 2018.

Este trabalho é dedicado a aqueles que buscam superação, mesmo achando que não são capazes. O nosso coração é o lugar da morada de Deus e somente Ele nos capacita.

AGRADECIMENTOS

Agradeço a Deus por tudo, por me sustentar até este momento e pelas graças que ainda alcançarei, em nome de seu filho amado Jesus Cristo.

Agradeço minha amada esposa Adriana pela motivação e por estar sempre ao meu
lado.

Agradeço ao meu orientador pela dedicação, paciência e, acima de tudo, por acre-
ditar em mim.

Agradeço a todos os professores da Universidade Estadual de Londrina por com- partilhar os conhecimentos necessários para enriquecer ainda mais este trabalho.

Agradeço ao meu grande amigo Prof. Carlos Tojeiro da Faculdade de Tecnologia de Ourinhos que me ajudou na realização deste trabalho.

Agradeço também aos Mestres Cláudio Toshio Kawakani e Sean Carlisto de Alva- renga pela imensa ajuda e inspiração para realização deste trabalho.

Agradeço ao meu grande amigo Gabriel Vasquez por dividir esta jornada comigo.

"Mas, buscai primeiro o reino de Deus, e a sua justiça, e todas estas coisas vos serão acrescentadas."
(Bíblia Sagrada, Mateus 6:33)

MORAES, E. A. **Aplicação de aprendizado de máquina supervisionado e técnicas de correlação na análise de alertas de intrusão**. 2018. 62 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2018.

RESUMO

As tecnologias de invasão a redes de computadores vêm se sofisticando continuamente. Por este motivo, as organizações estão buscando cada vez mais o uso de ferramentas de segurança da informação contra ataques, visando a proteção de seus bens digitais. Para o combate a ações maliciosas nas redes de computadores, pode-se usar um Sistema de Detecção de Intrusão (IDS - *Intrusion Detection System*). Os IDS detectam vários tipos de comportamentos maliciosos em sistemas computacionais, que podem comprometer sua segurança e confiabilidade. Eles geram alertas no formato de *logs*, resultantes das análises efetuadas por meio do monitoramento dos pacotes que transitam pela rede de computadores, visando a detecção de atividades maliciosas. Com a informação obtida dos *logs*, é possível que administradores de rede tenham conhecimento do estado atual de seus ativos de redes, auxiliando-os no combate de possíveis invasões. Embora os IDS auxiliem na proteção dos sistemas, existe um problema: a geração de um grande volume de alertas, que sobrecarregam os administradores de rede. Além disso, alguns desses alertas podem estar reportando situações que, na verdade, não são ataques. Este trabalho apresenta uma proposta de correlação *off-line* de alertas de intrusão que tem duas características principais: (i) a redução do volume de alertas, utilizando uma filtragem por prioridades e aprendizado de máquina supervisionado para eliminação de alertas falsos positivos; (ii) identificação de relacionamentos entre os alertas de forma a evidenciar as estratégias de ataque utilizadas contra a rede em análise.

Palavras-chave: IDS. Alertas de intrusão. Correlação. Aprendizado de Máquina.

MORAES, E. A. **Applying supervised machine learning and correlation techniques to analyze intrusion alerts**. 2018. 62 p. Thesis (Master in Computer Science) – Universidade Estadual de Londrina, Londrina, 2018.

ABSTRACT

Technologies for computer network intrusion have become increasingly sophisticated. For this reason, organizations seeking to use information security tools against attacks to protect their digital assets. To tackle malicious actions on computer networks, an Intrusion Detection System (IDS) can be used. IDS detect various types of malicious behavior in computer systems, which can compromise their security and reliability. They generate alerts in the format of logs, resulting from the analysis carried out by monitoring the packets passing through the computer network, in order to detect malicious activities. With the information obtained from logs, it is possible for network administrators to be aware of the current state of their network assets, assisting them in combating potential intrusions. Although IDS help to protect systems, there is a problem: the generation of a large volume of alerts, which overwhelm network administrators. In addition, some of these alerts may be reporting situations that are not really attacks. This work presents a proposal for off-line correlation of intrusion alerts that has two main characteristics: (i) reduction of the volume of alerts, using priority based filtering and supervised machine learning to eliminate false positive alerts; (ii) identification of relationships between alerts in order to show the attack strategies used against the network under analysis.

Keywords: IDS. Intrusion Alerts. Correlation. Machine Learning.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AD	Árvore de Decisão
AM	Aprendizado de Máquina
API	Application Programming Interface
BPM	Business Process Management
CERT.br	Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil
CGI	Common Gateway Interface
DDoS	Distributed Denial of Service
DNS	Domain Name Server
HIDS	Host-based IDS
IDS	Intrusion Detection System
IP	Internet Protocol
IPSec	Intenet Protocol Security
ISP	Internet Service Provider
kNN	k Nearest Neighbour
NIDS	Network-based IDS
NTPX	Network Time Protocol eXploit
PHP	Personal Home Page
RF	Random Forest
SQL	Structured Query Language
SSDP	Simple Service Discovery Protocol
SSH	Secure Shell
SSL	Secure Socket Layer
SVM	Support Vector Machine
TCP	Transmission Control Protocol
TI	Tecnologia da Informação
UDP	User Datagram Protocol
UPnP	Universal Plug and Play

SUMÁRIO

1 INTRODUÇÃO

Desde os primeiros computadores desenvolvidos, a tecnologia vem evoluindo constantemente e tais evoluções trouxeram grandes ferramentas de comunicação como a Internet, entre outras. Como o aumento global da rede de computadores é constante, consequentemente, o número de usuários conectados vem aumentando paralelamente. Nas décadas de 80 e 90, o computador pessoal se popularizou, assumindo um papel fundamental na vida de indivíduos e organizações. Atualmente, as redes de computadores são os principais meios que as empresas, organizações e governos utilizam na disponibilização de diversos serviços de rede como *Web Services*, Bancos de Dados, Serviços em Nuvem, dentre outros. Enquanto surgem novas ferramentas com o intuito de melhoria em determinados segmentos, também são desenvolvidas ferramentas para o uso indevido das redes de computadores como: quebra de senhas de segurança, exploração de vulnerabilidades de sistemas, entre outras [3].

O maior desafio não é apenas manter a disponibilidade de serviços destas redes, mas também detectar a presença de indivíduos conectados, que por razões diversas, tentam impedir o fornecimento dos serviços, ou interceptar as informações trocadas entre os dispositivos. Grande parte dos administradores de sistemas computacionais, em um dado momento, irão se deparar com um evento de intrusão de segurança durante suas carreiras. Ter um plano de detecção de intrusão resultará em uma notificação mais ágil, minimizando as consequências e permitindo uma rápida recuperação, em caso de algum sinistro [4, 5].

Para a elaboração de um plano que garanta os aspectos básicos da informação (confidencialidade, integridade e disponibilidade), que garanta a continuidade dos serviços de redes de computadores, que gerencie os incidentes de segurança e que seja parte integral da governança corporativa, é necessária a elaboração de um conjunto estruturado de competências e habilidades estratégicas na área de tecnologia da informação, com foco no planejamento, implantação, controle, monitoramento de sistemas e gestão de projetos. A governança de TI (Tecnologia da Informação) é o conjunto de práticas, padrões e relacionamentos estruturados, assumidos não apenas pelos responsáveis da área de TI, mas também por executivos, gestores, técnicos e usuários de uma organização, com a finalidade de garantir controles efetivos, minimizar os riscos, ampliar o desempenho, otimizar a aplicação de recursos de TI a custos otimizados, prestar suporte à tomada de decisões estratégicas e consequentemente alinhar TI aos objetivos negociais, gerenciando os processos ligados à segurança da informação [6].

A informação é um ativo importante para as organizações. Devido a tal importância, é necessário que haja um plano de segurança eficaz, garantindo que *crakers* ou demais grupos de invasores não obtenham tais informações [7, 8]. Uma falha de segurança em

uma empresa pode acarretar grandes prejuízos, os quais podem ser refletidos em seu fluxo econômico como a perda da credibilidade da empresa perante a concorrência, podendo ocasionar um afastamento de investidores e clientes devido à falta de segurança, entre outras perdas incalculáveis [9].

Os incidentes de segurança reportados no Brasil têm mostrado uma tendência de crescimento nos últimos anos, segundo as estatísticas da CERT.br [10]. Dos 647.112 incidentes reportados, 59,33% são ataques do tipo *scan*, 17,87% são ataques do tipo fraude ou páginas falsas, 8,57% são ataques que comprometem servidores *Web*, 9,34% são ataques de negação de serviços distribuídos, 4,37% são ataques do tipo *worm*, 0,26% são ataques do tipo invasão à sistemas e os 2,27% restantes são as demais categorias de ataques menos expressivas. Tais informações reforçam a demanda por ferramentas dedicadas à defesa das redes de computadores.

Uma das principais ferramentas de defesa é o Sistema de Detecção de Intrusão (IDS – *Intrusion Detection System*). O IDS analisa os *hosts* e tráfego, gerando *logs*[1] e alertas ou até mesmo tomando ações pré-determinadas pelo administrador da rede, caso sejam detectadas atividades maliciosas como ataques a servidores, alterações de permissões de acessos, entre outras, fornecendo uma camada extra de segurança. Em suma, os IDS englobam o processo de monitoramento de redes e dos *hosts*, notificando a ocorrência de atividades não autorizadas [11]. Intrusão é toda atividade que pode resultar em alteração de privilégios de usuários e permissões em arquivos, instalação de *malware*, acesso não autorizado a arquivos e sistemas, ataques de negação de serviços, presença de *worms* e vírus, estouros de *buffers*, entre outros [12]. Com a análise dos *logs* gerados pelos IDS, os administradores conseguem efetivar um plano de detecção e resposta às intrusões.

No entanto, existe um problema relacionado à efetividade dos IDS: a geração de um grande volume de alertas, dos quais vários são considerados falsos, dificultando a análise feita pelos administradores de sistemas e analistas de segurança. Técnicas de correlação de alertas combinadas com aprendizado de máquina supervisionado podem facilitar a análise destes grandes volumes de informação e seus comportamentos [11, 13, 14, 15].

Neste trabalho, é proposto um processo de correlação de alertas de intrusão, iniciando com a etapa de filtragem de alertas de baixa prioridade. Em seguida, é realizada a remoção dos alertas falsos positivos. Para identificar os alertas falsos, um algoritmo de aprendizado de máquina supervisionado é utilizado para analisar informações como origem e destino do ataque, assinatura reportada no alerta, função que os *hosts* envolvidos desempenham na rede, e horário do evento. Para comparação e avaliação de desempenho, foram utilizados dois algoritmos supervisionados: o kNN (*k Nearest Neighbour*) e o *Random Forest*. O processo é concluído aplicando a técnica de correlação proposta no trabalho de Kawakani *et al.* [1], que é capaz de identificar os padrões de estratégia de ataque que

[1] Log: Processo de descrição e registro de eventos relacionados ao sistema computacional.

são normalmente utilizados numa determinada rede de computadores por meio de um histórico de alertas de intrusão.

As contribuições deste trabalho são as seguintes: (i) um método de remoção de falsos positivos baseado em aprendizado de máquina supervisionado que utiliza informações do próprio alerta e informações adicionais referentes à localização e ao papel dos nós de rede envolvidos no evento reportado; (ii) processo resultante da combinação do método de remoção de falsos positivos com técnicas de identificação de estratégias de ataque propostas por Kawakani *et al.* [1].

O restante deste trabalho é organizado da seguinte forma:

- No Capítulo 2 serão descritos os principais elementos deste trabalho como Sistemas IDS, geração de alertas, alertas falsos positivos, correlação de alertas e trabalhos relacionados.
- O Capítulo 3 descreve a proposta de correlação de alertas apresentada neste trabalho.
- O Capítulo 4 apresenta os resultados obtidos nos experimentos acompanhados de discussão dos mesmos.
- O Capítulo 5 apresenta a conclusão. São feitas as considerações finais e direção dos futuros trabalhos.

2 REFERENCIAL TEÓRICO

Neste capítulo, será apresentada a fundamentação dos assuntos envolvidos neste trabalho. Na Seção 2.1, é apresentada uma breve introdução sobre os IDS. Na Seção 2.2, são descritas as categorias de IDS. Na Seção 2.3, são descritas as metodologias utilizadas pelos IDS quanto a verificação dos dados. Na Seção 2.4, são discutidos os desafios no uso dos IDS quanto a sua eficiência. Na Seção 2.5, são discutidas técnicas de correlação de alertas de intrusão e na Seção 2.6, são destacados os trabalhos de outros autores na área de detecção de intrusão e correlação de alertas.

2.1 Funcionalidade de um IDS

Os primeiros IDS tiveram início na década de 80 com o trabalho de Anderson [16]. O autor observou que relatórios de auditoria em sistemas computacionais podiam conter informações que auxiliariam no rastreamento de usos impróprios dos sistemas e acessos indevidos, proporcionando um entendimento do comportamento de usuários mal-intencionados. Este trabalho forneceu o fundamento necessário para os estudos e desenvolvimento de IDS. Em 1985, Denning e Neumann [17] observaram que os invasores tinham hábitos diferentes de usuários comuns do sistema, e que estas diferenças poderiam ser notadas, tornando-se possível o desenvolvimento de uma ferramenta de detecção automatizada.

Existem ferramentas de proteção a redes de computadores contra ameaças externas e invasores como *firewalls*, uso de criptografia, sistemas de filtro de conteúdo, controle de acesso, dentre outras. Porém estes sistemas de proteção normalmente não são capazes de monitorar as atividades dos invasores que obtiveram acesso autenticado dentro de uma rede de computadores através de meios suspeitos. Devido a este fato, torna-se crucial o monitoramento das informações contidas nos pacotes de dados que atravessam estas barreiras de proteção, acrescentando assim uma camada de segurança dentro de um segmento de rede e nos *hosts*.

O IDS é um elemento importante dentro de um plano de segurança da informação [3], pois tem como principal objetivo analisar o tráfego de rede e os dispositivos que a compõem a fim de detectar atividades maliciosas, suspeitas, impróprias e incorretas dentro de um segmento de rede e nos *hosts*. É uma ferramenta especializada em ler e interpretar conteúdos de pacotes. Ao identificar o pacote como uma possível ameaça, o IDS executa suas atividades de acordo com a abordagem escolhida pelo administrador do sistema, podendo ser um registro de *log* ou alguma ação proativa [18].

A utilização de IDS não envolve apenas a geração de alertas para o administra-

dor do sistema, mas também a análise dos eventos de segurança, exames preventivos e obstruções de conexões suspeitas. O processo de detecção de intrusão é o de identificar atividades suspeitas que possam interferir nos princípios da integridade, confidencialidade e disponibilidade. Além disso, os IDS são capazes de distinguir de onde se originaram os ataques, de dentro ou fora da rede em questão, analisando arquivos locais em busca de rastros ou tentativas malsucedidas de conexão aos computadores de uma rede.

Uma das vantagens da utilização do IDS é o registro dos alertas em seus *logs*, que possuem informações úteis para o administrador de sistemas, tais como tentativas de ataques sofridas por dia, tipo de tentativa de ataque que foi utilizado, origem e destino do ataque [19]. Por meio dos *logs* do IDS, o administrador poderá ter o conhecimento de boa parte das tentativas de ataque que estão sendo realizadas em seu sistema, podendo se proteger melhor contra as futuras tentativas de invasões.

O IDS possui dois modelos de utilização: modo passivo e modo reativo [20]. No modo passivo, o IDS não realiza nenhuma ação proativa em relação ao ataque, por exemplo, o descarte do pacote. Neste cenário, o IDS gera um *log* e envia para o administrador da rede para visualização. No modo reativo, o IDS não só realiza o envio de *logs* para o administrador como também executa ações pré-definidas para cada tipo de ataque detectado.

2.2 Posicionamento dos IDS

Um IDS pode ser classificado quanto ao seu posicionamento: sistemas de detecção baseados em *host* (HIDS - *Host-based IDS*), sistemas de detecção baseados em redes (NIDS - *Network-based IDS*) e sistemas distribuídos ou híbridos, conforme explicado adiante [21].

2.2.1 Sistemas baseados em *host* (HIDS)

A detecção baseada em *host* tem como ênfase a instalação do sistema de detecção em máquinas específicas. Normalmente o HIDS é instalado em servidores ou máquinas com funções críticas, que necessitam de um monitoramento e segurança mais específicos, de acordo com a necessidade da rede [22, 23]. O HIDS possui mecanismos que possibilitam monitorar diversas atividades da máquina hospedeira, como: análise do uso de *hardware*, sistema operacional, alterações em pastas, alterações nos privilégios dos usuários, entre outras atividades disponíveis. Um *trojan*, por exemplo, programado para alterar permissões de acessos do sistema pode ser instalado em um servidor, sendo programado para que toda vez que o servidor for reiniciado, o *trojan* execute atividades como alteração de permissão de usuários, dando possíveis permissões indevidas para que *black hats* invadam o servidor. Esta ação maliciosa provavelmente passaria despercebida pelo administrador da rede caso não houvesse nenhum tipo de monitoramento. Porém com a utilização de

HIDS, tal alteração de arquivos seria alertada. Após a detecção de alterações nos arquivos, o HIDS gera *logs* registrando todas as alterações ocorridas.

Uma das vantagens da utilização do HIDS é que, como as análises são realizadas em determinados ativos, ele permite que as regras de detecção de intrusão possam ser elaboradas de acordo com a necessidade de cada *host* [18]. A desvantagem é que o HIDS limita-se apenas no *host* que está sendo monitorado.

2.2.2 Sistemas baseados em redes (NIDS)

O NIDS realiza o monitoramento dos pacotes que transitam dentro de um segmento de rede. O processo de detecção é realizado com a captura dos pacotes, análise dos cabeçalhos e seus conteúdos [23]. O processo de verificação de pacotes acontece com a integração de dois elementos do NIDS: os sensores e o *console*. Os sensores são responsáveis pela captura dos pacotes, pela formatação e pela análise do tráfego da rede. Normalmente são instalados em diversos pontos dentro de um segmento de rede. Já o *console* realiza o gerenciamento integrado dos sensores, funcionando como uma interface para o usuário.

O NIDS conta com dois padrões de instalação: modo *inline* e modo passivo. No modo *inline*, o NIDS é instalado em pontos críticos de conexões entre redes, capturando e analisando o tráfego em tempo real [19]. Um ponto positivo deste padrão é a possibilidade da detecção e ação proativa em tempo real. Em contrapartida deve-se lembrar que, em determinadas organizações, o fluxo de dados dentro de um segmento de rede é alto e que a presença de um IDS no modo *inline* pode causar queda de desempenho.

No modo passivo, também conhecido como promíscuo, o NIDS atua na análise de uma cópia do tráfego que é enviada para ele. Um ponto negativo é o tempo de resposta a invasões, ou seja, um ataque possivelmente chegará ao alvo enquanto o NIDS realiza sua análise.

Há diferentes NIDS sendo utilizados no mercado atualmente como Suricata, Bro IDS e *Security Onion*. Para este trabalho, o IDS utilizado será o SNORT[1], que é um NIDS e, portanto, utiliza métodos de detecção previstos para esse tipo de IDS. O SNORT foi escolhido pois, além de ser *open source*, é o IDS mais utilizado em sua categoria, realiza a análise de tráfego em tempo real e faz os registros dos eventos de segurança, utilizando recursos mínimos de processamento. O SNORT verifica os eventos e combina as inspeções em protocolos e assinaturas, gerando os alertas de intrusão.

2.2.3 Sistemas distribuídos ou híbridos

Em paralelo aos pontos fortes e fracos dos HIDS e NIDS, existem também *softwares* que combinam as vantagens oferecidas pelos dois tipos de sistemas, que são os IDS

[1] SNORT: https://www.snort.org/

distribuídos ou híbridos. Sua definição ainda é complexa e indistinta, pois varia muito de acordo com a sua funcionalidade, sua aplicação no ambiente, topologia e diferencia-se também de um fabricante para outro [18]. Uma característica importante que define um IDS distribuído ou híbrido refere-se à utilização de vários IDS funcionando como sensores ou agentes, que manipulam informações dos *hosts* e da rede, reportando os eventos encontrados para um IDS único que funciona como servidor, para concluir um gerenciamento centralizado.

A vantagem dos IDS distribuídos ou híbridos é a utilização das capacidades técnicas tanto de um HIDS quanto de um NIDS. Sendo assim, ataques que não são detectados por um NIDS, podem ser suportados por um HIDS [3]. O IDS distribuído ou híbrido capacita a aplicação com as diversas vantagens de duas categorias de IDS, os baseados em *host* e os baseados em rede, oferecendo uma visão holística da rede e dos sistemas pontuais monitorados.

2.3 Metodologias de detecção de intrusão

Para que a análise dos dados seja realizada, o IDS precisa seguir uma metodologia de verificação [24]. A detecção é baseada na premissa de que as atividades intrusivas são diferentes das ações normais e, portanto, são possíveis de serem detectadas [25]. Um IDS pode ser classificado em três categorias no que diz respeito a sua forma de detecção de intrusão: baseado em assinaturas, baseado em anomalias ou baseado em especificações.

2.3.1 Detecção baseada em assinatura

As assinaturas são regras que definem características de invasões ou ataques já conhecidos, e são previamente configuradas nos sensores do IDS [21]. Este processo permite que o IDS realize uma comparação entre todos os dados coletados com as assinaturas, verificando a ocorrência de ataques que já são conhecidos em seu banco de dados. Um exemplo de IDS baseado em assinaturas é o SNORT, utilizado neste trabalho.

Como novas formas de ataques são criadas constantemente, as regras de detecção precisam ser atualizadas com frequência [23], evitando que ataques sejam bem-sucedidos por serem desconhecidos. O método por assinatura é eficaz na detecção de intrusos com ataques já conhecidos, porém quando o sistema se depara com ataques desconhecidos de sua base de assinaturas, a chance do pacote indevido ser detectado pelo IDS é nula.

Neste ponto, pode-se evidenciar uma desvantagem: o método de detecção baseado em assinaturas, mesmo sendo o método mais utilizado, contém algumas limitações e desvantagens, já que para um IDS identificar um tipo de ataque com precisão, ele compara uma informação obtida do ataque com a assinatura registrada em um banco de dados. Caso esta informação sobre o ataque não esteja registrada e o ataque seja ainda desco-

nhecido, o IDS fica impossibilitado de identificar o ataque e, consequentemente, de gerar alertas.

2.3.2 Detecção baseada em anomalia

Os IDS baseados em anomalias criam dinamicamente uma espécie de perfil de comportamento, registrando todas as atividades rotineiras durante um período determinado. Quando o *host* ou a rede tem uma alteração no comportamento previsto no perfil diagnosticado pelo IDS, ele interpreta como uma anomalia, caracterizando uma tentativa ou ocorrência de um ataque [18].

Uma das vantagens do método de detecção por anomalia é a capacidade de detectar ataques ainda desconhecidos pelo IDS. Mesmo que o ataque seja desconhecido, ele tende a ter um comportamento fora do perfil anteriormente diagnosticado, sendo detectado pelo IDS. Outra vantagem é produzir informações que podem ser usadas na definição de novas assinaturas para o IDS baseado em assinaturas.

Uma desvantagem reside na fase de criação do perfil. Se o *host* ou a rede sofrerem algum tipo de ataque que não seja identificado a tempo, o IDS registra no perfil como uma ação normal, permitindo que futuros ataques do mesmo gênero passem despercebidos pelo IDS [18, 24]. Outra desvantagem é a geração de um volume grande de alertas falsos devido ao comportamento imprevisível de usuários e sistemas. Muitas sessões podem ser necessárias para coleta de amostras de dados do sistema na etapa de criação de perfil, de modo a caracterizar padrões de comportamento normais.

2.3.3 Detecção baseada em especificação

Também chamado de método híbrido, na detecção baseada em especificação, o IDS combina as estratégias de detecção por anomalias e por assinaturas por meio de técnicas de expressões regulares onde são criadas previamente as especificações de funcionamento da rede e dos sistemas [26]. Estas especificações são compostas por comportamentos considerados normais e situações consideradas como tentativas de invasão [27]. Uma vantagem é o baixo volume de alertas falsos positivos, caso as especificações sejam configuradas corretamente por um especialista com conhecimento sobre o sistema monitorado. A desvantagem é que o volume das especificações tende a aumentar conforme a necessidade de segurança, deixando o IDS lento.

2.4 Desafios no uso de IDS

Com relação à eficácia na detecção de intrusão [21], o primeiro desafio enfrentado pelo IDS são as interpretações equivocadas dos dados analisados, que são divididas em dois grupos: falsos positivos e falsos negativos.

- **Falsos Positivos (FP):** ocorrem quando um evento, após ser analisado pelo IDS, é caracterizado como uma tentativa de ataque, quando na verdade se trata de um evento legítimo.
- **Falsos Negativos (FN):** ocorrem quando eventos maliciosos passam despercebidos pelo IDS. Neste caso, o IDS analisa o evento e define-o como legítimo, quando na verdade ele faz parte de um ataque.

Outros desafios dos IDS seriam os avanços tecnológicos constantes [21, 28]. Áreas como infraestrutura e protocolos vêm se tornando cada vez mais complexas. Um exemplo dessa situação é a utilização de protocolos de criptografia como SSL (*Secure Socket Layer*), IPSec (*Intenet Protocol Security*), entre outros protocolos.

Algumas tecnologias de criptografia codificam o campo de dados dos pacotes, outras o pacote inteiro [29]. Neste cenário, um IDS pode ser ineficaz já que os dados de um ataque estão encobertos pela criptografia aplicada por algum tipo de protocolo, impedindo que o IDS analise os pacotes.

2.5 Correlação de alertas de intrusão

Correlação de alertas de intrusão é a identificação da relação entre eventos de segurança, levando em consideração suas similaridades. Podemos considerar um exemplo, em que tem-se uma lista de alertas de intrusão relacionados à alteração de um arquivo PHP [2] de um servidor *Web* que foi identificada pelo IDS. Desta lista, extrai-se um atributo em particular: endereço IP do *host* de origem. Uma forma simples de correlação implicaria na identificação de outros eventos de segurança (tentativa de acesso com privilégios administrativos ao servidor *Web*, por exemplo) que contém o mesmo endereço IP do *host* de origem [30]. A Figura 1 mostra um exemplo de correlação.

```
05/20/2016-16:53:12.899517 [**] [1:130:4] WEB-MISC adminlogin access
[**] [Classification: Attempted Recon] [Priority: 2] {TCP} [redacted].35:
62061 -> [redacted].12:80

05/20/2016-16:53:23.296857 [**] [1:26:4] WEB-PHP admin.php access [**]
[Classification: Web Application Activity] [Priority: 2] {TCP}
[redacted].35: 62081 -> [redacted].12:80
```

Figura 1 – Exemplo de correlação de alertas.

No exemplo da Figura 1, observa-se dois alertas em sequência referentes ao mesmo endereço IP de origem do *host*. Além de possuírem o mesmo endereço IP, os alertas

[2] PHP: acrônimo recursivo para PHP: *Hypertext Preprocessor*, mas a sigla também é conhecida originalmente por *Personal Home Page*.

ocorreram com pouca diferença de tempo, aproximadamente 11 segundos de diferença entre um alerta e outro. Com base nas assinaturas apresentadas nos alertas, é possível observar uma tentativa de acesso com privilégio de administrador em um servidor *Web*.

A correlação de alertas de IDS pode ser classificada em três métodos: método baseado em regras, método baseado em causa e consequência e método baseado em similaridades [31].

- **Método baseado em regras:** os alertas são correlacionados de acordo com uma base de conhecimentos de cenários de ataques. Quando uma tentativa de invasão ocorre e os alertas são gerados, o método faz a correlação, procurando na base de conhecimento os cenários com sintomas similares ao problema informado [32, 33, 34]. Algumas desvantagens: geração de grande volume de dados, alto consumo de poder computacional, método baseado apenas em vulnerabilidades conhecidas, sem garantias que os alertas seguirão o mesmo cenário armazenado na base de conhecimento.
- **Método baseado em causa e consequência:** a correlação ocorre com a identificação de um conjunto de alertas preparatórios, que remetem a eventos que denotam que um ataque está sendo preparado, que está relacionado com outro conjunto de alertas resultantes, que remetem a eventos que são consequência de uma condição prévia. Para a correlação ser válida, um alerta preparatório precisa ter em suas consequências ao menos um atributo que se repita (ou se relacione) nas causas do alerta resultante (Exemplo: o *timestamp* do alerta preparatório precisa ser anterior ao do alerta resultante) [34, 35].
- **Método baseado em similaridade:** realiza o agrupamento dos alertas com base nos atributos semelhantes, como número de porta de origem ou destino, endereço IP de origem ou destino, protocolo, descrição do alerta, *timestamp*. A correlação é feita por meio de análise estatística, conjuntos de regras, pontuações ponderadas, algoritmos de clusterização, distância Euclidiana, distância de Jaccard e outras métricas relacionadas à similaridade [35]. Como exemplo, pode-se comparar os dois alertas da Figura 1 e dizer que estão correlacionados se apresentarem o mesmo endereço IP como origem [36, 37].

2.6 Trabalhos relacionados

Os trabalhos que propõem soluções para lidar com grandes volumes de alertas gerados pelos IDS podem ser organizados em dois grupos.

Os trabalhos do primeiro grupo propõem a correlação ou priorização de alertas como processos principais na redução do grande volume de dados. Esses trabalhos até

podem incluir métodos de redução de falsos positivos, mas não têm isso como objetivo final ou como elemento principal da proposta.

Uma das primeiras abordagens nessa linha foi a de Valdes e Skinner [38], que tem foco único e exclusivo na correlação de alertas. A correlação de alertas é alcançada por meio da clusterização daqueles que possuem atributos similares. Para cada atributo do alerta, é definida uma função específica envolvendo cálculos probabilísticos para determinar o grau de similaridade entre os alertas.

No trabalho de Lee *et al.* [39], também foi proposto um método que correlaciona os alertas por meio de similaridade entre seus atributos. Os autores desenvolveram um sistema que filtra alertas redundantes e os agrega em *clusters* de hiper-alertas com base em atributos específicos (endereço IP de origem e tipo de ataque). Logo após, é calculada a similaridade entre os hiper-alertas levando em conta os demais atributos como o endereço IP, as portas de origem e destino, tipo de ataque e *timestamp* do alerta.

Treinen e Thurimella [40] propuseram uma abordagem com base em mineração de dados e regras de associação entre os alertas para encontrar relacionamentos e dependências causais. Nesta abordagem, os alertas são modelados em um grafo direcionado, em que os vértices representam os endereços IP de origem ou destino do alerta e as arestas conectam os vértices da origem para o destino. Técnica semelhante é utilizada também por Kawakani *et al.* [1]. Cada modelagem criada auxilia o administrador da rede na identificação de tentativas de ataques.

Zurutuza *et al.* [41] também propuseram uma técnica de correlação de alertas com base em mineração de dados e regras de associação. Bem semelhante aos trabalhos de Treinen e Thurimella [40] e de Kawakani *et al.* [1], os autores desenvolveram uma abordagem dividida em quatro etapas. A primeira etapa é o pré-processamento de alertas onde alguns atributos são selecionados. Na segunda etapa, os alertas são agrupados com base nas similaridades dos atributos selecionados na primeira etapa por meio de um algoritmo de clusterização denominado *Expectation Maximization*. Na terceira etapa, para cada *cluster* de alertas formado, regras de associação são derivadas utilizando o algoritmo denominado *Apriori*. Na quarta etapa é atribuído um rótulo para cada regra: tráfego normal, tráfego suspeito ou ataque.

Soleimani e Ghorbani [42] propuseram um método de redução de alertas por meio de filtragem, separando os alertas considerados mais críticos dos que possuíam baixa prioridade. Para identificar os alertas críticos, os autores separaram um grupo e analisaram os atributos e a sequência temporal. Deste ponto eram identificados dois grupos de ataques: ataques críticos simples e ataques críticos do tipo multiestágio. No passo seguinte, os autores utilizaram o algoritmo de árvore de decisão para filtrar os eventos de segurança, identificando os alertas de baixa prioridade e os de alta prioridade.

Taha *et al.* [43] propuseram um modelo de redução de alertas baseado em componentes distintos de correlação denominados de agentes. Cada agente utiliza um critério diferente para realizar o agrupamento de alertas baseado num conhecimento determinado previamente. O modelo remove alertas duplicados e, em seguida, identifica a probabilidade de um ataque ter sucesso e possíveis cenários de ataques.

Yang *et al.* [44] propuseram um sistema de análise e priorização de alertas, bem como visualização de ataques multiestágio. Os autores definiram quatro etapas: processamento de alertas do IDS por meio de modelo de comparação de similaridade e clusterização, geração de regras de associação de alertas, elaboração de perfil e comportamento do atacante e, finalmente, predição de visualização de ataques do tipo multiestágios. É na etapa de comparação de similaridade, denominada pelos autores de modelo *flocking*, que os alertas classificados erroneamente são distinguidos dos alertas verdadeiros.

Liu *et al.* [45] também apresentam uma abordagem para correlação e visualização. Em sua proposta, os autores definiram um modelo dividido em quatro etapas. A primeira etapa trata da formatação das informações dos alertas no mesmo padrão. A segunda etapa trata da filtragem dos alertas na identificação de falsos positivos. A terceira etapa trata da correlação e agrupamento de alertas, onde são analisados os cenários de ataques críticos, a identificação de perfil do atacante e a identificação dos alvos. A quarta etapa trata da visualização de cenários de ataques por meio de grafos. Os autores descrevem com detalhes cada etapa do modelo, porém a segunda etapa não apresenta informações mais precisas sobre o processo de filtragem de falsos positivos.

Saad e Traore [46] propõem um modelo de correlação de alertas com base nas similaridades dos atributos dos alertas. O modelo é baseado na taxonomia das classes de ataques para medir o grau de similaridade entre alertas. Os autores realizaram um levantamento teórico para a elaboração da taxonomia das classes e, em seguida, aplicaram o modelo em um grupo de eventos de segurança, reduzindo o grande volume de alertas e formando grupos representados como hiper-alertas.

Shittu *et al.* [47] propuseram uma abordagem de visualização de alertas de intrusão, utilizando um agente de correlação de alertas que melhora a compreensão de cenários de ataques. Dado um determinado conjunto de eventos gerados por um IDS, a abordagem dos autores seleciona o método mais eficaz para a correlação de eventos de segurança.

Fayyad e Meinel [48] propuseram um modelo de predição de estratégias de ataque em tempo real, utilizando grafos de ataque atualizados. O modelo é comporto por três etapas. A primeira etapa contempla a filtragem de alertas falsos positivos e de alertas duplicados. É nesta etapa em que os alertas são agrupados de acordo com sua similaridade. A segunda etapa identifica as sequências de alertas, as quais são mapeadas no grafo de ataque. Esta é a etapa na qual acontece a atualização do modelo de predição. Na última etapa é feita a predição de próximas estratégias de ataque em tempo real.

Elshoush [14] também propôs um processo que visa diminuir o volume de dados a ser analisado durante a correlação, aumentando a sua eficiência. Primeiramente, o autor removeu uma serie de alertas de baixo risco. Depois, os alertas restantes foram agrupados de acordo com os seus *hosts* de origem e destino. Os alertas que não foram agrupados nesta fase foram classificados como falsos. Os autores consideraram que alertas falsos têm um comportamento aleatório e por isso acabam não sendo agrupados de acordo com os critérios utilizados no trabalho. Em nosso trabalho, observou-se que alertas falsos não estão ligados obrigatoriamente a situações incomuns ou a comportamentos aleatórios. Pelo contrário, alertas falsos têm algumas características em comum, que costumam diferenciá-los de alertas verdadeiros.

Hachmi *et al.* [49] propuseram um processo de redução de volumes de alertas, dividido em duas etapas. Na primeira etapa os autores utilizaram algoritmo de clusterização *K-Means* e geraram um agrupamento de meta-alertas. Na segunda etapa utilizaram técnica de aprendizado de máquina para classificar e identificar alertas verdadeiros e falsos. Para a etapa de classificação e identificação de falsos positivos, compararam dois algoritmos: SVM *(Support Vector Machine)* e Árvore de Decisão. Neste trabalho os autores filtraram os alertas identificados como falsos positivos, reduzindo o volume de alertas a serem analisados.

Ghasemigol e Ghaemi-Bafghi [50] propuseram uma abordagem de correlação dos alertas de intrusão utilizando o conceito de entropia parcial de alerta para agrupá-los em *clusters* menores. Os autores afirmam que pode-se representar grandes volumes de alertas em um grupo menor de hiper-alertas por meio de função de entropia. O número de hiper-alertas pode ser limitado por meio de duas funções: princípio de entropia máxima ou conceito de entropia parcial de alertas.

Kawakani *et al.* [1] propuseram uma abordagem baseada no agrupamento de alertas. Duas etapas se destacam: a primeira etapa é a correlação dos alertas e a observação do histórico destes alertas para a identificação das estratégias de ataque utilizadas. A segunda etapa é a associação dos alertas em tempo real às estratégias de ataque descobertas na primeira etapa. Nesta proposta, de maneira similar ao trabalho de Elshoush [14], alertas que não atenderam aos critérios de agrupamento foram removidos do processo de correlação. No entanto, não houve uma análise mais criteriosa para certificar se eles eram mesmo falsos ou não.

No trabalho de Chakir *et al.* [51] foi proposto um modelo de priorização de alertas com base num processo de avaliação de risco dos ativos de rede que sofreram tentativas de ataques. O modelo de Chakir *et al.* utiliza atributos como prioridade, confiabilidade e valor do ativo como fatores de criticidade no processo de classificação de risco que um alerta pode gerar neste ativo. O modelo identifica quais alertas possuem criticidade alta com base nos seus níveis de risco e filtra os alertas de baixo risco, classificando-os como

falsos.

No segundo grupo de trabalhos, os pesquisadores priorizam a identificação de falsos positivos como o objetivo final do processo. Julisch e Dacier [5] propuseram uma avaliação de duas técnicas de análise de alertas de intrusão. Uma técnica é a mineração de regras de ataques para priorização de alertas, visualização de estratégias de ataques e identificação de ferramentas usadas nos ataques, pois tais ferramentas tendem a gerar alertas com a mesma sequência de assinaturas para diferentes alvos. A outra técnica é a clusterização, onde os alertas que apresentam atributos similares são agrupados para reduzir a quantidade de dados. Os autores notaram que, com a primeira técnica, alguns alertas apresentavam um certo grau de homogeneidade e repetitividade, caracterizando um provável cenário de ataque. Alertas que não apresentavam tais característica, os autores consideravam falsos positivos.

Novamente, Shittu *et al.* [52], num trabalho mais recente, propõem uma métrica para priorização de alertas de intrusão usando a correlação dos alertas. Primeiramente, os alertas são correlacionados em grafos. Então, os alertas pertencentes a grafos que correspondem aos comportamentos menos frequentes recebem prioridades mais altas. Os alertas com prioridades mais baixas são identificados como alertas falsos, sendo filtrados.

Vidal *et al*. [53] propuseram a utilização de algoritmo de clusterização *K-Means* na priorização de alertas. No trabalho, primeiramente, há uma fase de treinamento na qual o IDS analisa qual é o comportamento normal da rede. Logo após esta fase, o IDS gera alertas para pacotes que se distanciam deste comportamento normal. De acordo com a proposta, para cada novo alerta gerado, é calculada a distância entre o pacote que causou a geração do alerta e os pacotes normais do treinamento. Quanto maior a distância entre o pacote responsável pelo alerta e os pacotes não maliciosos do treinamento, menor será a prioridade do alerta. Alertas com prioridade baixa são considerados falsos.

Hachmi *et al.* [54] propuseram em um trabalho mais recente a detecção de alertas falsos e *outliers* por meio de correlação de alertas dividida em três etapas. Na primeira etapa os autores aplicaram a correlação baseada em similaridade numa base de alertas de intrusão para criação de dois conjuntos de meta-alertas: alertas falsos e alertas verdadeiros, utilizando o algoritmo *K-Means*. Na segunda etapa aplicaram um processo de comparação e cardinalidade entre os eventos sobre os meta-alertas gerados na etapa anterior utilizando um algoritmo denominado *Binary Otimization Problem* (BOP) para identificação e eliminação de *outliers*. A terceira etapa consiste na detecção de falsos-positivos por meio do algoritmo *Binary Classification Algorithm* (BCA) que se baseia nos atributos dos alertas. Os autores utilizaram a base DARPA 99 para a criação dos meta-alertas e se limitaram apenas nos atributos que constavam nos alertas, não se baseando em informações externas relacionadas a eles.

Neste trabalho, propõe-se uma abordagem de correlação de alertas de intrusão

onde analisa-se, primeiramente, o comportamento dos alertas para identificação e remoção de falsos positivos, e na sequência, aplica-se técnicas de correlação *off-line* propostas por Kawakani *et al.* [1]. A contribuição deste trabalho está centrada na combinação da filtragem de alertas por prioridade com um processo de identificação de falsos positivos que utiliza aprendizado de máquina supervisionado e informações externas que não estão presentes nos alertas. A união destas duas funcionalidades com técnicas de correlação *off-line* propostas por Kawakani *et al.* [1] resultará na elaboração de grafos de estratégia de ataque que tornarão mais evidente a real situação do ambiente computacional estudado.

3 PROCESSO PARA REDUÇÃO DE ALERTAS FALSOS POSITIVOS E CORRELAÇÃO DE ALERTAS

Este capítulo apresenta a proposta de um processo para correlação de alertas, cuja função é reduzir a quantidade de alertas que não são relevantes para análise do administrador, filtrando aqueles que possuem prioridade baixa e utilizando algoritmos de aprendizado de máquina para identificar os falsos positivos. Logo em seguida é feito o processo de correlação para elaboração de grafos que representam as estratégias de ataques usadas contra a rede monitorada. Para visualizar o processo proposto, foi elaborado um modelo BPM *(Business Process Management)*, uma ferramenta de gerenciamento que otimiza resultados de desempenho de processos com agilidade operacional [55]. No caso do processo de redução de alertas falsos positivos e correlação de alertas proposto neste trabalho, o BPM fornece uma visão específica do funcionamento de cada etapa, proporcionando um melhor entendimento. O modelo BPM é exibido na Figura 2.

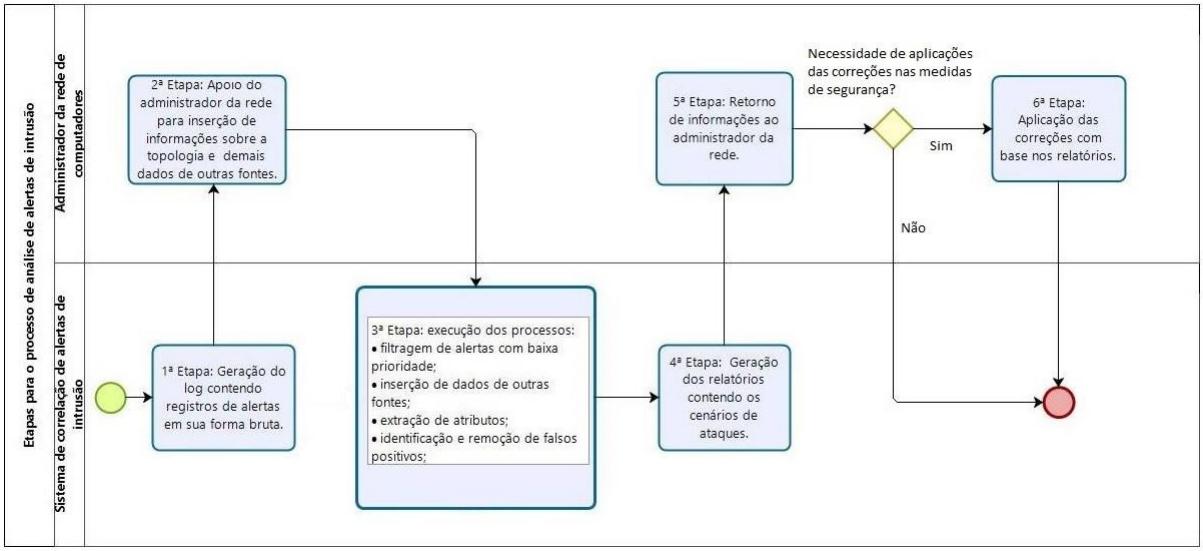

Figura 2 – Modelo BPM das etapas para o processo de análise de alertas de intrusão.

Conforme ilustrado na Figura 2, a primeira etapa é a geração dos alertas pelo IDS no formato de *logs*. A segunda etapa é realizada pelo próprio administrador da rede que deve alimentar o modelo com informações adicionais. Na terceira etapa são realizados os processos de filtragem de alertas de baixa prioridade e remoção de falsos positivos, com base nos *logs* de alertas gerados pelo IDS e nas informações providas pelo administrador sobre a topologia. Na quarta etapa são gerados os relatórios com os cenários de ataques. Na quinta etapa, as estratégias de ataque são geradas a partir dos cenários de ataque e são reportadas ao administrador. O administrador da rede fará, então, a aplicação de medidas de segurança e as correções necessárias na sexta etapa. Nas seções que seguem,

será descrito o processo proposto em detalhes.

3.1 Coleta de alertas gerados pelo IDS e filtragem por prioridade

O primeiro passo do modelo proposto é a coleta dos alertas gerados pelo IDS. Logo após coletá-los, os alertas serão filtrados de acordo com a prioridade de cada um. Em geral, nos IDS, há uma classificação dos alertas que permite aos administradores de sistemas um entendimento rápido e uma priorização mais efetiva de cada evento. No IDS SNORT, por exemplo, cada alerta tem uma prioridade determinada por um valor: nível 1 para prioridade alta, nível 2 para prioridade média, e nível 3 para prioridade baixa.

Na prioridade 1, os alertas representam uma ameaça muito grande ao sistema, precisando de uma análise imediata. A prioridade 2 é considerada importante, podendo se tornar uma ameaça caso não seja analisada e tratada. Na maioria dos casos, pode-se desconsiderar os alertas de prioridade 3 em uma primeira análise, pois são aqueles que não representam uma ameaça, mas sim uma característica mais informativa sobre o estado do sistema.

A Figura 3 mostra um exemplo de alertas com prioridade baixa, que reportam uma tentativa de comunicação com a porta 80 do servidor de autenticação de um *wireless access point*, gerando alertas com assinatura *(portscan) TCP Portsweep* com prioridade igual a 3. Estes alertas foram gerados para a rede utilizada no estudo de caso apresentado na Seção 4. Este tipo de evento é muito típico e normalmente está relacionado à solicitação de um *host* para que suas credenciais sejam verificadas durante a conexão com o *access point*. Mesmo que haja uma intenção maliciosa, esta tentativa de conexão tem um baixo grau de risco.

```
04/17/2016-16:52:23.705315 [**] [122:3:0] (portscan) TCP Portsweep [**]
[Classification: Attempted Information Leak] [Priority: 3] {TCP}
10.0.4.172:3556 -> 10.0.0.1:80

04/17/2016-16:54:23.913522 [**] [122:3:0] (portscan) TCP Portsweep [**]
[Classification: Attempted Information Leak] [Priority: 3] {TCP}
10.0.5.100:3447 -> 10.0.0.1:80
```

Figura 3 – Exemplos de alertas de prioridade baixa.

Na filtragem por prioridade, o modelo proposto prevê que todos os alertas de uma determinada prioridade sejam separados do conjunto de alertas que será submetido ao restante do processo. Após a filtragem de acordo com a prioridade definida pelo IDS, se inicia o processo de identificação de falsos positivos utilizando aprendizado de máquina supervisionado com a inserção de novas informações de outras fontes à base de alertas que serão analisados.

3.2 Inserção de informações de outras fontes

Para demonstrar a importância de inserir informações de outras fontes à base de alertas, toma-se como exemplo um alerta gerado pelo IDS utilizado no estudo de caso apresentado na Seção 4. O alerta é apresentado na Figura 4.

```
04/17/2016-13:52:23.053985 [**] [1:2001219:20] ET SCAN Potential SSH
Scan [**] [Classification: Attempted Information Leak] [Priority: 2]
{TCP} [redacted].146:38454 -> [redacted]:22
```

Figura 4 – Exemplo de alerta gerado por um IDS.

A Tabela 1 detalha quais são as informações providas no alerta. São informações básicas, que indicam algumas características do evento, mas não conseguem mostrar sozinhas, por exemplo, qual é a localização geográfica dos *hosts* envolvidos no ataque ou qual é a função desses *hosts* nas suas respectivas redes.

Tabela 1 – Descrição do alerta.

Atributo	**Valor**
Mês / Dia / Ano	04/17/2016
Hora / Minuto / Segundo	13:52:23.053985
Código da Assinatura	[1:2001219:20]
Assinatura	*ET SCAN Potential SSH Scan*
Classificação	*[Classification: Attempted Information Leak]*
Prioridade	*[Priority: 2]*
Protocolo	{TCP}
Endereço de IP de Origem	xxx.xxx.xxx.146
Porta de Origem	38454
Endereço de IP de Destino	(Endereço IP do Servidor *Web* da faculdade)
Porta de Destino	22

Buscando as informações em outras fontes e inserindo-as junto aos dados contidos nos alertas, é possível descobrir mais alguns detalhes sobre o evento que podem ajudar a identificar se o alerta é falso ou não. Esta busca é feita por uma API (*Application Programming Interface*) que possibilita a inserção automáticas das informações adicionais nos alertas.

Utilizando um recurso de geolocalização de endereços IP como o GeoIP2 [1], pode-se extrair informações mais detalhadas sobre o endereço IP do atacante, como cidade, país, latitude, longitude, raio de ação, provedor de Internet e organização.

Como estas informações estão relacionadas ao endereço IP, é importante considerar que existe a possibilidade do endereço IP ser falsificado pelo atacante. A Tabela

[1] GeoIP2: https://www.maxmind.com/pt/geoip-demo

2 apresenta estas informações para o endereço IP xxx.xxx.xxx.146. Este endereço IP é apontado como endereço do atacante no alerta da Figura 4.

Tabela 2 – Informações de geolocalização do endereço IP.

Atributo	**Valor**
Código do País	CN
Localização	Changzhou, Jiangsu, China, Ásia
Coordenadas aproximadas (Latitude, Longitude)	31.7833, 119.9667
Raio de ação (em Km)	50
Internet Service Provider (ISP)	China Telecom
Organização	Liyang Hongkou Primary School

Fonte: https://www.maxmind.com/pt/geoip-demo

Os dados destacados na Tabela 2 podem melhorar a qualidade das informações contidas no alerta. Neste caso, o endereço IP xxx.xxx.xxx.146 está localizado na China, província de Jiangsu. A princípio, muitos analistas de segurança pensariam que, pelo endereço IP, o ataque teria sido realizado por algum invasor tentando conexão SSH na porta 22, utilizando um serviço de *proxy* externo ou a Rede Tor [2], dificultando sua localização exata. Porém, o provedor do serviço de Internet é a China Telecom, e a organização é uma escola primária denominada Liyang Hongkou. De acordo com consulta no Google Maps [3], a escola realmente existe e as coordenadas do endereço IP coincidem com a localização física. Assim, levanta-se também a possibilidade de que o invasor tenha comprometido um sistema desta escola, passando a utilizar este endereço IP. De qualquer forma, todas as características ajudam a mostrar que o alerta é verdadeiro, já que a rede alvo pertence a uma faculdade brasileira, cujos serviços não são normalmente acessados por *hosts* na China, de acordo com seus administradores.

Na proposta deste trabalho, busca-se, além de informações inseridas automaticamente sobre a localização geográfica dos endereços IP, informações sobre a topologia da rede fornecidas pelo administrador, determinando quais dos *hosts* envolvidos fazem parte da rede alvo, e caso façam parte, qual seriam as funções destes *hosts* (Ex.: função de servidor, *host* interno, etc.). A seguir, é apresentada uma lista de informações consideradas relevantes na identificação de alertas falsos:

- *Timestamp*: informações de data e horário. Momento que ocorreu o alerta.
- Assinatura: descrição única de uma tentativa de invasão. Somente o código da assinatura foi utilizado.
- Prioridade: indica o grau de severidade de uma tentativa de invasão.

[2] Rede Tor: *software* que proporciona o navegação anônima na Internet, escondendo informações que possam identificar um usuário e as suas atividades.

[3] Google Maps: https://www.google.com.br/maps

- Porta de origem e destino: o número da porta de origem ou destino que indica a aplicação à qual se destinam os dados (Ex.: alertas que possuem a porta 80 como destino, normalmente possuem assinaturas relacionadas a vulnerabilidades de servidores *Web*).
- Endereço IP: identificação do *host* de origem e de destino.
- Localização: indica se o endereço IP de origem ou destino pertence ou não ao mesmo país onde localiza-se a rede de computadores monitorada pelo IDS.
- Função do *host*: no caso do *host* pertencente a rede monitorada, o administrador indica se esse *host* é um servidor ou não.

Na próxima seção, quando forem apresentados os atributos extraídos a partir dessas informações, será explicado como cada atributo, e consequentemente cada informação, contribui para o processo de identificação de falsos positivos.

Outras informações também foram destacadas, porém, não foram inseridas por não serem consideradas relevantes para a próxima etapa de extração de atributos:

- Coordenadas aproximadas: latitude e longitude aproximadas relacionada ao endereço IP. Estes dados não são considerados como informações importantes para este trabalho, pois já temos os endereços IP de origem e destino e a localização física dos mesmos.
- Raio de precisão: fornece, em quilômetros, o raio de ação segundo a latitude e longitude do endereço IP, segundo GeoIP2.
- ISP (*Internet Service Provider*): nome do provedor de serviço de Internet, da organização provedora, ou nome do sistema autônomo associado ao endereço IP. Não é considerada uma informação importante para este trabalho, pois na maior parte das buscas, este campo aparece em branco. Segundo o GeoIP2, o nome do ISP está disponível para cerca de 40% das redes empresariais, governamentais e de ensino.
- Organização: organização de registro relacionado ao endereço IP.
- Domínio: domínio de segundo nível relacionado ao endereço IP como:"exemplo.com"ou "exemplo.com.br", mas não o domínio completo como "algo.exemplo.com".
- Código metropolitano: está disponível somente para endereços IP dos Estados Unidos.

3.3 Extração de atributos

As informações adicionais inseridas de outras fontes juntamente com as informações contidas no próprio alerta são usadas para extrair atributos que serão a entrada para o algoritmo de aprendizado de máquina supervisionado usado na identificação dos alertas falsos. Do conjunto de informações inicialmente levantado para os alertas, são extraídos os seguintes atributos para o processo de identificação de alertas falsos:

- **Atributos referentes ao período**: contribuem na identificação de alertas falsos conforme o horário da ocorrência do alerta. Baseado em observações feitas na rede do estudo de caso, normalmente os alertas falsos aparecem com mais frequência no horário com mais atividade de usuários legítimos na rede. É atribuído um valor lógico para estes atributos, o valor "0"significa que é falso e o valor "1"significa que é verdadeiro:
  - *manhã*: indica se o alerta foi gerado entre 6h00 e 11h59;
  - *tarde*: indica se o alerta foi gerado entre 12h00 e 17h59;
  - *noite*: indica se o alerta foi gerado entre 18h00 e 23h59;
  - *madrugada*: indica se o alerta foi gerado entre 0h00 e 5h59;
- **Atributos referentes ao tipo de ataque**: auxiliam no entendimento da natureza do ataque. Determinadas assinaturas e prioridades podem apresentar uma possibilidade maior de ter alertas falsos:
  - *assinatura*: código da assinatura presente no alerta;
  - *prioridade*: prioridade atribuída pelo IDS ao alerta;
- **Atributos referentes à localização física do *host* e aos serviços atacados**: a ideia é identificar *hosts* e serviços de rede que têm maior propensão a gerar alertas falsos e usar esta informação para classificar os alertas (foi utilizado a versão IPv4):
  - *porta_origem*: porta de origem do ataque presente no alerta;
  - *porta_destino*: porta de destino do ataque presente no alerta;
  - *parte1_end_origem*: 1º octeto do endereço IP de origem;
  - *parte2_end_origem*: 2º octeto do endereço IP de origem;
  - *parte3_end_origem*: 3º octeto do endereço IP de origem (o quarto octeto que indica o endereço do *host* não foi necessário, pois na maioria dos casos, os três primeiros já permitem a identificação da sub-rede);
  - *parte1_end_dst*: 1º octeto do endereço IP de destino;
  - *parte2_end_dst*: 2º octeto do endereço IP de destino;

  - *parte3_end_dst*: 3º octeto do endereço IP de destino (o quarto octeto não foi necessário, pois na maioria dos casos os três primeiros já permitem a identificação da sub-rede);
  - *país_origem*: indica se o endereço IP de origem está localizado no mesmo país da rede monitorada pelo IDS;
  - *país_dst*: indica se o endereço IP de destino está localizado no mesmo país da rede monitorada pelo IDS;

- **Atributos referentes à função do *host*** auxilia na identificação de alertas falsos com base na funcionalidade do *host* de origem ou destino. Por meio da análise do endereço IP, identifica se o *host* de origem ou destino pertence à rede monitorada. Em caso positivo, é possível verificar se ele é um servidor da rede interna ou não. O campo foi preenchido a partir da análise dos endereços IP e do conhecimento sobre a alocação destes endereços na rede:
  - *origem_servidor* : a partir da análise do endereço IP de origem, indica se a origem é um servidor da rede monitorada;
  - *dst_servidor*: a partir da análise do endereço IP de destino, indica se o destino é um servidor da rede interna;
  - *origem_host_externo*: a partir da análise do endereço IP de origem, indica se a origem é um equipamento externo à rede monitorada;
  - *dst_host_externo*: a partir da análise do endereço IP de destino, indica se o destino é um equipamento externo à rede monitorada.

3.4 Treinamento com dados rotulados e classificação dos alertas

O treinamento é executado com o objetivo de encontrar um padrão que melhor descreva os alertas e assim gerar um modelo capaz de prever se o alerta é falso ou verdadeiro. Na etapa de treinamento dos dados, os algoritmos supervisionados construirão um modelo de classificação com base nos alertas rotulados pelo administrador de rede como falsos ou verdadeiros. Na fase de classificação, o modelo gerado pelos algoritmos supervisionados na fase de treinamento é usado para analisar os alertas e atribuir um rótulo, classificando-os em verdadeiros ou falsos. Nos experimentos apresentados na Seção 4, os algoritmos *kNN* e *Random Forest* foram utilizados para que os seus desempenhos fossem comparados.

O *kNN* (*k Nearest Neighbour*) é um algoritmo de classificação supervisionada que determina o rótulo de classificação de um alerta baseado em alertas vizinhos provenientes de um conjunto de treinamento. A variável *k* representa a quantidade de vizinhos mais próximos que serão utilizados para averiguar a qual classe o novo alerta pertence e pode

ser ajustada para se obter uma melhor classificação. O valor de *k* varia de acordo com a base de alertas, mas é recomendável que seja um número ímpar ou primo [56].

O algoritmo *Random Forest* (RF) utiliza o método *ensemble learning* que, por meio de um subconjunto de atributos selecionados aleatoriamente para formar os dados de treinamento a partir do conjunto original, gera vários classificadores do tipo AD (Árvore de Decisão) e combina seus resultados por meio de um mecanismo de votação [57]. Ao final do processo de combinação dos resultados, é atribuída uma classificação considerando a classe que recebeu o maior número de votos entre todas as ADs. [58] [59].

3.5 Identificação de padrões de estratégia de ataque

Após remover os falsos positivos, se inicia no processo proposto a fase de análise das relações entre os alertas para descobrir as estratégias de ataque. Para implementar essa fase no processo proposto, escolhemos adotar técnicas propostas por Kawakani *et al.* [1].

O trabalho de Kawakani *et al.* [1] tinha como objetivo propor um método que reduzisse a grande quantidade de alertas gerados pelo IDS e que fornecesse informações mais precisas para o administrador da rede tomar as decisões relacionadas às medidas de segurança. O método de Kawakani *et al.* gera hiper-alertas com base em estratégias de ataque identificadas num histórico de alertas e é formado por dois correlacionadores: um correlacionador *off-line* e um correlacionador *online*. A Figura 5 exibe o correlacionador *off-line* de Kawakani *et al.*

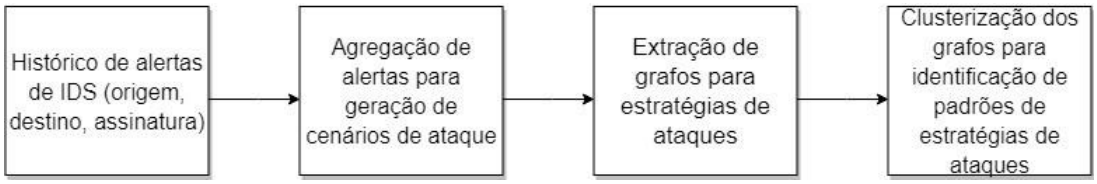

Figura 5 – Modelo de correlação *off-line* de alertas de Kawakani *et al.* [1].

Para este trabalho, adotamos técnicas presentes no correlacionador *off-line* de Kawakani *et al.*, que recebe como entrada um histórico de alertas e é responsável por processá-los a fim de encontrar os padrões de estratégia de ataque utilizados pelos atacantes. Conforme ilustrado na Figura 5, o correlacionador *off-line* de Kawakani *et al.* é composto por quatro fases:

- Fase 1: com base no histórico de alertas, o correlacionador inicia um processo de análise e organização com o objetivo de identificação de cenários de ataques.

- Fase 2: o correlacionador inicia o processo de agregação, agrupando alertas em cenários de ataques, com base nos endereços IP e proximidade no tempo entre os alertas.
- Fase 3: inicia-se a etapa de extração de grafos de estratégia de ataque, com base em cada cenário identificado na fase de agregação.
- Fase 4: todas as estratégias de ataques similares são agrupadas em *clusters*.

No trabalho de Kawakani *et al.* [1], os cenários de ataque formados na fase 2 são representados como componentes conexos. Para construir os componentes conexos, é necessário analisar os endereços IP de origem e destino dos alertas e o intervalo de tempo corrido entre eles. Para exemplificar a construção de um componente conexo, podemos utilizar os alertas representados na Tabela 3.

Tabela 3 – Exemplo de alertas de IDS.

Timestamp	**Origem**	**Destino**	**Assinatura**
05/15/2016 22:45:01	192.168.0.1	192.168.0.254	A
05/15/2016 22:45:02	192.168.0.1	192.168.0.254	B
05/15/2016 22:45:10	192.168.0.2	192.168.0.254	C
05/15/2016 22:45:15	192.168.0.2	192.168.0.254	D
05/15/2016 22:45:22	192.168.0.2	192.168.0.254	D
05/16/2016 12:24:34	192.168.0.3	192.168.0.254	E
05/16/2016 12:24:45	192.168.0.3	192.168.0.254	F
05/16/2016 12:24:57	192.168.0.3	192.168.0.254	F

De acordo com a Tabela 3, os *hosts* 192.168.0.1, 192.168.0.2 e 192.168.0.3 estão atacando o *host* 192.168.0.254 com seis assinaturas diferentes. O componente conexo é definido como um grafo direcionado e representa um cenário de ataque por meio do conjunto de vértices e arestas [1]. Os vértices representam os endereços IP e as arestas representam a direção do ataque entre os endereços IP de um determinado cenário de ataque.

Ainda de acordo com a Tabela 3, nota-se a existência de dois grupos de alertas: um grupo é formado pelas assinaturas "A", "B", "C" e "D" e outro formado pelas assinaturas "E" e "F", separados por uma diferença de tempo de aproximadamente 14 horas entre os grupos. No trabalho de Kawakani *et al.* [1], há um limiar de tempo n usado para dividir os cenários de ataque. Se dois alertas apresentam uma diferença de tempo entre eles maior que o limiar n, então esses alertas são colocados em cenários de ataque diferentes. Levando-se em consideração um limiar de tempo n igual a 1 hora entre um grupo e outro, é possível extrair dois cenários distintos de ataques (componentes conexos) em nosso exemplo. A Figura 6 ilustra a representação visual dos dois cenários de ataques por meio de componentes conexos.

Figura 6 – Exemplo de componentes conexos, com base na Tabela 3.

Com os componentes conexos devidamente formados, é possível extrair os grafos de estratégia de ataque que apresentam a sequência de assinaturas que ocorreu em cada cenário de ataque. Esta sequência de assinaturas também é definida por meio de um grafo direcionado, onde o conjunto de vértices representa as assinaturas dos alertas e o conjunto de arestas representa a relação de sequência entre as assinaturas dos alertas [1]. A Figura 7 ilustra grafos de estratégia de ataque com base na Tabela 3.

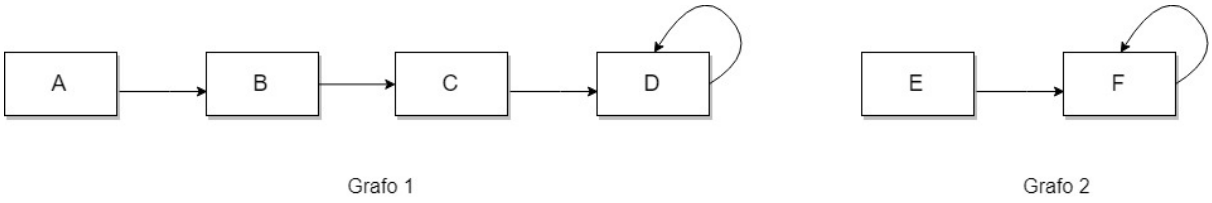

Figura 7 – Grafos de estratégia de ataque.

Após gerar os grafos de estratégia de ataque, o último passo é identificar os padrões de estratégia de ataque. O correlacionador *off-line* realiza o agrupamento dos grafos de estratégia de ataque similares e identifica um padrão, pois a mesma estratégia pode ocorrer diversas vezes na base de dados de alertas. Para medir o grau de coincidência entre os grafos de estratégia de ataque, o correlacionador utiliza uma cálculo estatístico que compara a similaridade e diversidade de conjuntos de amostras de alertas denominado índice de Jaccard [60], gerando uma matriz de similaridade entre os grafos de estratégia de ataque.

Para comparar dois grafos de estratégia de ataque, o método de Kawakani *et al*. [1] propõe medir a similaridade entre os conjuntos de assinaturas presentes neles. É considerada a relação existente entre o número de assinaturas em comum e o número total de assinaturas encontradas quando se comparam os dois conjuntos de assinaturas (A e B). Basicamente, o índice de similaridade reflete o tamanho da interseção dividido pelo tamanho da união dos conjuntos de assinaturas:

$$\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (3.1)$$

Onde:

- $|A \cap B|$: número de assinaturas em comum entre A e B;
- $|A \cup B|$: número total de assinaturas.

Se:

- Jaccard(A, B) = 1: todas as assinaturas são comuns entre A e B;
- Jaccard(A, B) = 0: não existe assinaturas em comum entre A e B;

A matriz de similaridade resultante, contendo o índice de Jaccard para todos os pares de grafos de estratégia de ataque, é utilizada para agrupar os grafos em uma estrutura hierárquica de acordo com as suas similaridades. Na Figura 8 é mostrado um exemplo de uma matriz de similaridade.

	Grupo A	Grupo B	Grupo C	Grupo D	Grupo E
Grupo A	1	0,8	0,4	0,15	0
Grupo B	0,8	1	0,5	0,2	0
Grupo C	0,4	0,5	1	0,7	0,6
Grupo D	0,15	0,2	0,7	1	0,85
Grupo E	0	0	0,6	0,85	1

Similaridade(Grupo A, Grupo B) = 0,8

Figura 8 – Exemplo de uma matriz de similaridade.

Esta estrutura hierárquica normalmente apresentada no formato de árvore com suas ramificações é denominada dendrograma. No dendrograma, os nós pais agrupam os exemplos representados pelos nós filhos, formando a estrutura hierárquica de agrupamento [61]. Na Figura 9 é mostrado um exemplo de um dendrograma.

Essa técnica permite analisar os *clusters* em diferentes níveis de granularidade, pois cada nível do dendrograma descreve um conjunto diferente de agrupamentos, formando os padrões de estratégia de ataque. Este processo é denominado de clusterização hierárquica aglomerativa, onde os grafos de estratégia de ataque são inicialmente distribuídos de modo

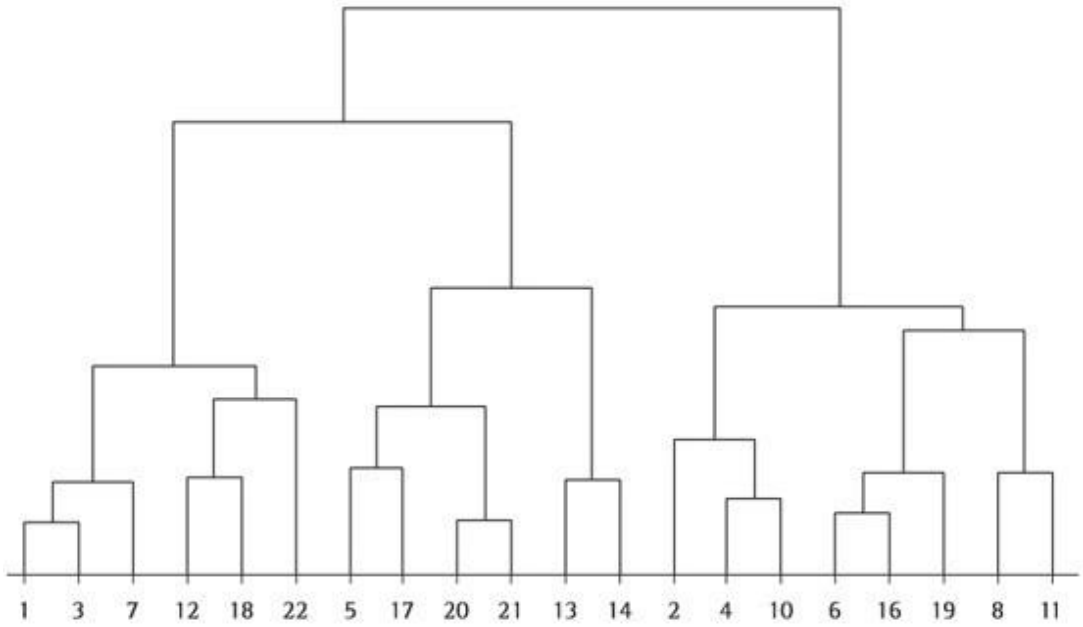

Figura 9 – Exemplo de um dendrograma [2].

que cada *cluster* tenha apenas um grafo. Estes *clusters* são agrupados de forma recursiva considerando a medida de similaridade (índice de Jaccard), até que todas as amostras pertençam a apenas um *cluster*.

Em suma, a proposta deste trabalho é aplicar as três últimas fases do correlacionador de Kawakani *et al.* [1]: (i) a agregação dos alertas em componentes conexos, (ii) a extração dos grafos de estratégia de ataque e (iii) a clusterização para identificação dos padrões de estratégia de ataque.

4 RESULTADOS E DISCUSSÃO

Este capítulo apresenta o estudo de caso realizado em uma rede corporativa real para a avaliação do modelo proposto neste trabalho. Os testes foram divididos em duas partes. Primeiramente foram utilizados os alertas referentes ao mês de março de 2016 para testar o desempenho do processo de filtragem de alertas por prioridade e redução de falsos positivos por aprendizado de máquina supervisionado.

Logo após, realizou-se de avaliação do processo de correlação de alertas proposto na Seção 3.5 deste trabalho. Para isto foram utilizados os alertas de três datas escolhidas aleatoriamente: 15 de março de 2016, 28 de abril de 2016 e 7 de maio de 2016. A topologia da rede de computadores onde foi aplicado este estudo é representada na Figura 10.

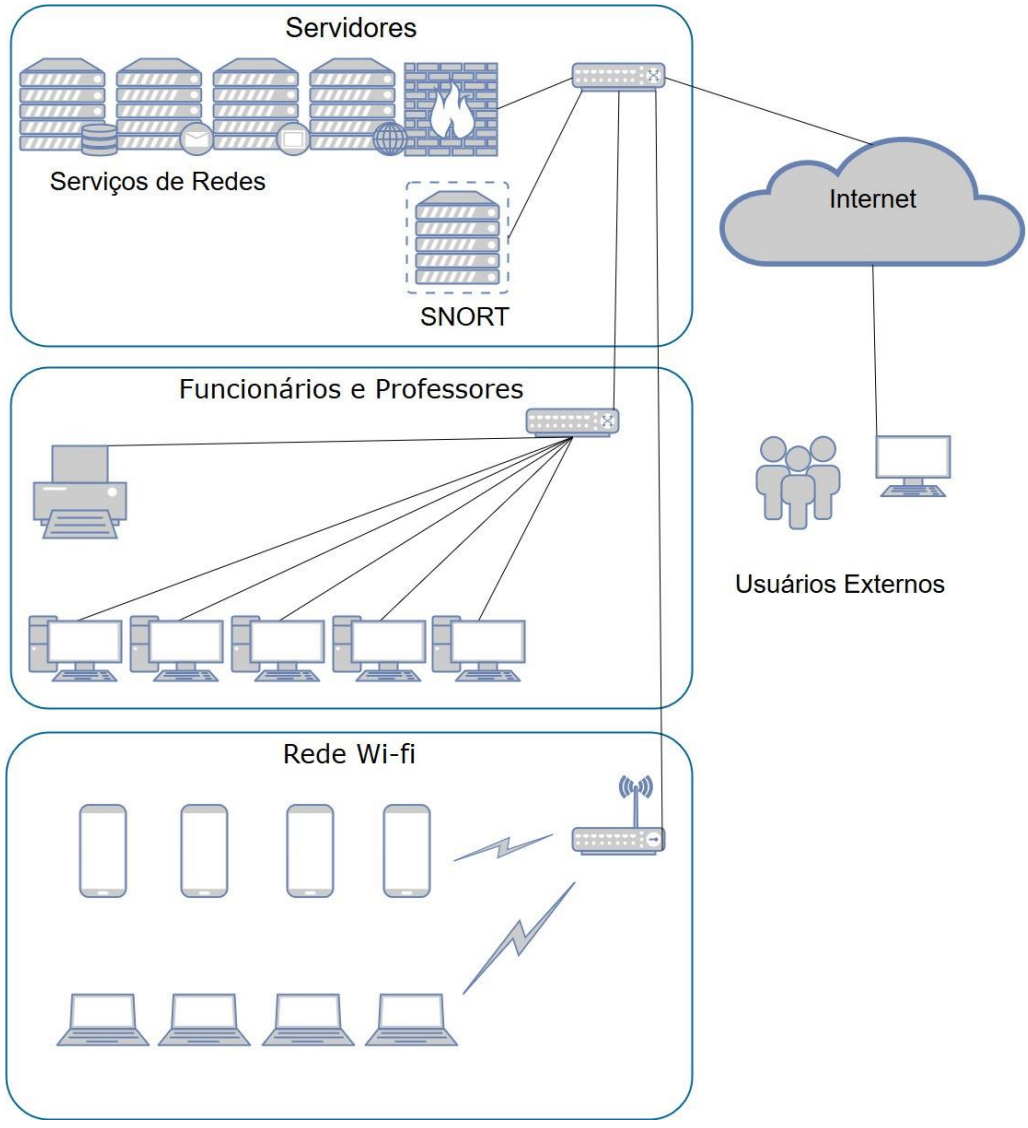

Figura 10 – Topologia da rede de computadores utilizada nos testes.

A rede de computadores visualizada na Figura 10 pertence a Faculdade de Tecnologia de Ourinhos (FATEC-OU). Ela é acessada por cerca de 1.750 usuários, entre alunos, professores e funcionários, tanto internamente por uma rede sem fios e pela rede cabeada dos departamentos, como externamente. Os servidores disponibilizam principalmente serviços Web, como a página principal da faculdade, o *blog*, serviços acadêmicos, materiais de ensino e bancos de dados acadêmicos.

Com base em estatísticas mensais coletadas pela administração da rede, nestes servidores são estabelecidas cerca de 1.000 conexões diárias. Na rede monitorada, há um IDS SNORT instalado, cujas assinaturas são atualizadas pelo próprio sistema, que gera em média 230.000 alertas diários, ou seja, mais de 2 alertas por segundo.

4.1 Redução de alertas falsos positivos e avaliação de desempenho dos algoritmos supervisionados

Para a avaliação de desempenho da filtragem de alertas por prioridade e da remoção de falsos positivos, foram utilizados os alertas referentes ao mês de março de 2016 gerados pelo IDS SNORT, pois neste mês ocorreram dois fatos importantes que geraram certas mudanças no comportamento dos alertas. O primeiro fato foi o início das atividades práticas nos laboratórios de informática da faculdade para o primeiro semestre de 2016, ocorrido no início da terceira semana deste mês, o que costuma propiciar ações atípicas na rede e um grande aumento do número de alertas. O segundo fato foi a ocorrência de um feriado prolongado na quarta semana, causando a diminuição das atividades da rede e alterando o volume e as características dos alertas. A divisão por semanas ocorreu para se obter resultados mais precisos, pois se a análise fosse feita no mês como um todo, não se conseguiria observar se fatos como início das aulas práticas e feriados teriam algum impacto nos resultados. A Tabela 4 mostra a quantidade de alertas por prioridade gerados neste mês.

Tabela 4 – Total de alertas por prioridade referente ao mês de março de 2016.

	Prioridade 1	**Prioridade 2**	**Prioridade 3**	**Total geral**
Semana 1	239	930	26.704	27.873
Semana 2	601	4.124	275.168	279.893
Semana 3	9.878	2.466	1.302.703	1.315.047
Semana 4	445	4.524	284.177	289.146
Semana 5	479	3.095	296.459	300.033
TOTAL	**11.642**	**15.139**	**2.185.211**	**2.211.992**

O primeiro passo do processo de redução de alertas falsos positivos descrito no Capítulo 3 é a filtragem de todos os alertas com uma determinada prioridade. No estudo de caso, filtrou-se todos os alertas com prioridade 3, pois são aqueles que representam normalmente uma característica informativa sobre o estado do sistema. Cerca de 91%

destes alertas são do tipo *Portsweep* ou *Open Port*, ou seja, tratam de uma varredura por portas abertas na rede. Normalmente, na rede monitorada, este alerta é gerado quando um dispositivo móvel de um aluno, funcionário ou professor solicita uma autenticação ao *access point* para poder se conectar à Internet. Os outros 9% de alertas com esta prioridade são majoritariamente dos tipos:

- *ICMP test detected*: evento de segurança gerado quando um teste é realizado para verificar se o servidor está respondendo;
- *Protocol mismatch*: evento de segurança gerado quando um teste é realizado para verificar se um serviço está respondendo em uma determinada porta;

Seguindo o modelo proposto, foram adicionadas aos alertas informações de outras fontes de dados sobre os eventos de segurança como a localização geográfica do endereço IP de origem e de destino, e a função do endereço IP na rede analisada. Então, foi realizada a extração dos atributos elencados na Seção 3.3, formando o conjunto de dados que será analisado pelos algoritmos de classificação supervisionada. Um algoritmo supervisionado trabalha em duas etapas. Primeiro, há um treinamento no qual o algoritmo recebe como entrada um conjunto de dados rotulados. Neste treinamento, o algoritmo constrói um modelo de classificação com base nos dados rotulados. No estudo de caso, os alertas foram rotulados como verdadeiros ou falsos pelos profissionais da administração de redes da faculdade. Na segunda etapa, o classificador usa o modelo para classificar novas instâncias de dados que são apresentadas a ele. Utilizou-se dois algoritmos de classificação supervisionada descritos na Seção 3.4: *k Nearest Neighbour* (*kNN*) e *Random Forest*.

Para medir o desempenho do classificador, foram utilizadas as seguintes métricas:

- Precisão: no conjunto de alertas classificados como verdadeiros, esta métrica mostra quantos realmente eram verdadeiros;

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{4.1}$$

- Sensibilidade: considerando o conjunto de alertas que realmente eram verdadeiros, esta métrica mostra quantos destes alertas foram classificados corretamente;

$$\text{Sensibilidade} = \frac{VP}{VP + FN} \tag{4.2}$$

- Especificidade: considerando o conjunto de alertas que realmente eram falsos, esta métrica mostra quantos destes alertas foram classificados corretamente;

$$\text{Especificidade} = \frac{VN}{VN + FP} \tag{4.3}$$

- F-score: avalia a cobertura e a precisão da classificação, considerando a média ponderada da precisão e da sensibilidade. Quanto mais próxima de 1, melhores são a cobertura e a precisão da classificação;

$$F\text{-}score = \frac{2 * (Precisão * Sensibilidade)}{(Precisão + Sensibilidade)} \quad (4.4)$$

TP, FP, TN, e FN correspondem respectivamente a *true positive* (positivo verdadeiro), *false positive* (falso positivo), *true negative* (negativo verdadeiro), e *false negative* (falso negativo).

Para avaliar os dois classificadores escolhidos, utilizou-se o método denominado validação cruzada, cuja proposta é particionar o conjunto de dados em k subconjuntos de mesmo tamanho. A partir deste ponto, um dos subconjuntos é utilizado para teste e o restante para treinamento. As rodadas de avaliação são repetidas k vezes e em cada uma destas rodadas um subconjunto diferente é usado para teste e os outros são usados para treinamento. Foi aplicado o método de validação cruzada em cada uma das semanas do mês de março de 2016. Em cada semana, dividiu-se os alertas em 7 subconjuntos, isto é, $k = 7$. A Figura 11 mostra os resultados obtidos com a aplicação do algoritmo *kNN*.

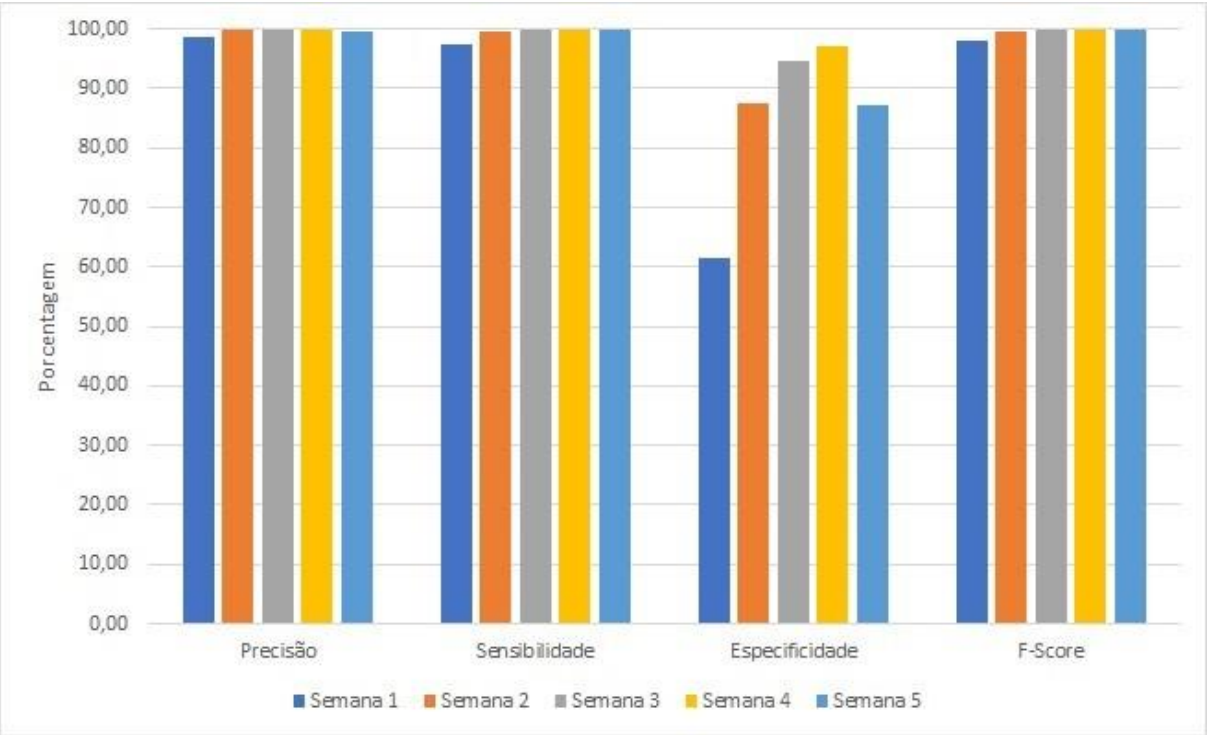

Figura 11 – Resultados para o algoritmo *kNN*.

Na Figura 11, pode-se observar que os resultados obtidos com o algoritmo de aprendizado de máquina *kNN* foram próximos para todas as semanas, com exceção da semana 1. Nesta semana, a métrica Especificidade (equação 4.3) teve o resultado de

61,54%, diferente das outras semanas que variaram entre 87,34% e 97,19%. Este resultado indica que na semana 1, o *kNN* encontrou dificuldades em classificar corretamente os alertas falsos, apontando erroneamente que alguns seriam verdadeiros. Este fato pode ter ocorrido em decorrência da quantidade de alertas falsos positivos ser muito pequena (53 alertas falsos positivos) quando comparada com a quantidade de alertas verdadeiros (1.116 alertas verdadeiros). O desbalanceamento entre as classes pode afetar o desempenho do *kNN*. Apesar deste resultado ruim para a semana 1, nota-se que os resultados para todas as métricas nas outras semanas foram bons, ficando sempre próximos a 100%. A Figura 12 mostra os resultados obtidos com a aplicação do algoritmo *Random Forest*.

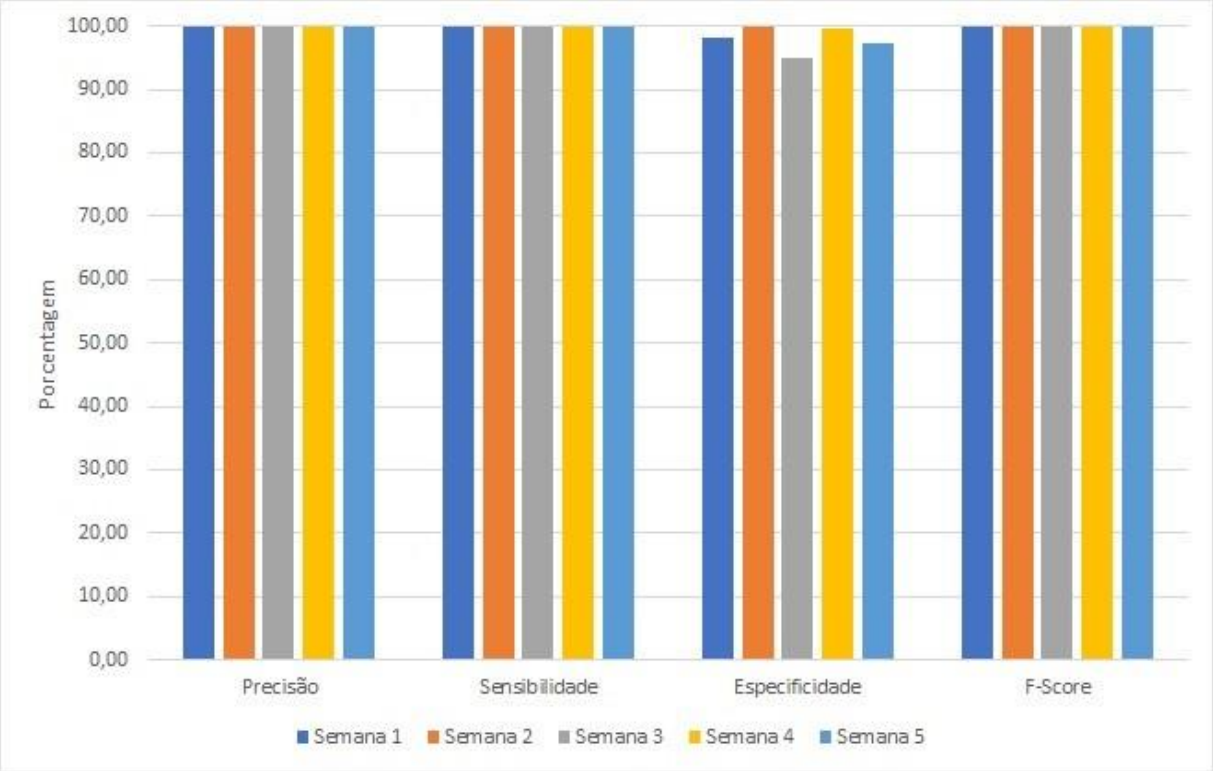

Figura 12 – Resultados para o algoritmo *Random Forest*.

Na Figura 12, os resultados obtidos com o *Random Forest* foram praticamente iguais para todas as semanas, sem exceção. Todas as métricas, para todas as semanas se situam entre 95,08% e 100%. O *Random Forest* conseguiu superar o *kNN* na semana 1, quando este último teve um resultado ruim para a Especificidade. No caso do *Random Forest*, mesmo com o desbalanceamento das classes para a semana 1, a Especificidade alcançada foi bastante alta. Isto ocorre porque o *Random Forest*, ao ser um algoritmo de *ensemble learning*, é mais robusto para lidar com o desbalanceamento de classes. Em geral, os resultados do *Random Forest* foram levemente melhores que os obtidos com o *kNN*.

A seguir, serão apresentados um caso de alerta classificado acertadamente como alerta falso e um caso de alerta classificado corretamente como alerta verdadeiro. O obje-

tivo é mostrar dois casos típicos que ilustram o funcionamento da proposta. A Figura 13 exibe um alerta classificado corretamente como falso positivo.

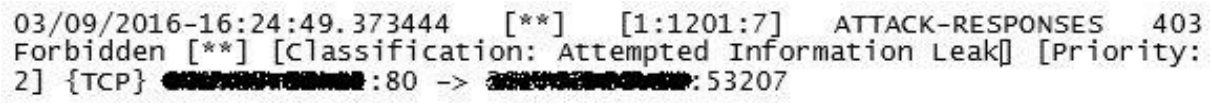

Figura 13 – Exemplo de alerta falso positivo.

O alerta exibido na Figura 13 se refere a uma resposta de um servidor *Web* interno da faculdade para outro servidor interno, responsável por hospedar o *blog* da faculdade. Esta resposta indica que uma página está proibida de ser acessada, por estar passando por alguma atualização ou inserindo um novo aviso interno para os alunos ou professores. Apesar de ser apenas uma resposta informativa, ela foi classificada pelo IDS como ataque. É importante observar que os atributos que extraímos para cada alerta ajudam a mostrar que o alerta é falso. Como exemplo, pode-se destacar os atributos *origem_servidor* e *dst_servidor*, que neste caso mostram que o alerta retrata um suposto ataque entre dois servidores da rede da faculdade.

A Figura 14 exibe um alerta classificado corretamente como verdadeiro.

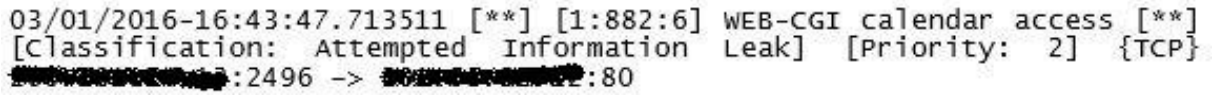

Figura 14 – Exemplo de alerta verdadeiro.

O alerta exibido na Figura 14 é referente a um ataque realizado por um *host* externo, localizado na cidade de Jacarezinho, no estado do Paraná. O atacante tenta executar um *script* de manipulação de calendário escrito em linguagem *Perl*, que permite a execução de comandos sem verificação de entrada de dados. O modelo proposto neste trabalho classificou esta ocorrência como ataque ao analisar características como o *host* de origem, sua localização física e a assinatura do alerta.

O modelo proposto para redução de falsos positivos teve um ótimo desempenho no mês avaliado. A aplicação do algoritmo *Random Forest* teve um desempenho melhor pois lida melhor com classes desbalanceadas e cria múltiplas ADs de forma aleatória para gerar um modelo de classificação.

4.2 Processo de correlação de alertas

Para avaliar o desempenho do processo de correlação de alertas proposto como um todo, foram realizados três testes comparativos. Em cada um destes testes, primeiramente,

foi testado o modelo de correlação de alertas de Kawakani *et al.* [1] sobre a base de alertas do estudo de caso, onde seguiu-se exatamente a implementação do trabalho deles. Na sequência, foi testado o processo de correlação proposto neste trabalho, com a mesma base de alertas do estudo de caso. Utilizou-se os alertas referentes à seguintes datas: 15 de março de 2016, 28 de abril de 2016 e 7 de maio de 2016, todas escolhidas aleatoriamente. Na Tabela 5, é exibido um resumo dos alertas gerados nessas datas contabilizados por prioridade.

Tabela 5 – Total de alertas por prioridade nos dias escolhidos para testes.

DATA	**15 / 03 / 2016**	**28 / 04 / 2016**	**07 / 05 / 2016**
Prioridade 1	8.528	112	102
Prioridade 2	537	4.402	416
Prioridade 3	381.118	348.226	205.674
TOTAL GERAL	**390.183**	**352.740**	**206.192**

O correlacionador proposto neste trabalho foi implementado utilizando a linguagem de programação Java[1], versão 1.8.0_152. O sistema operacional utilizado foi o *Windows* 10 *Pro* com 16 GB de memória. O sistema de gerenciamento de banco de dados utilizado para armazenar os alertas foi o MariaDB [2] na versão 10.1.25. A linguagem de programação R [3], versão 3.4.3, foi utilizada para realizar a clusterização hierárquica aglomerativa com a função *hclust* do pacote *stats* na mesma versão da linguagem de programação R.

Na clusterização hierárquica aglomerativa foi aplicado o método *Ward*, um procedimento de agrupamento calculado com a soma de quadrados entre os dois *clusters* feita sobre todas as variáveis. Este método foi adotado método pois resulta em agrupamentos aproximadamente de tamanhos semelhantes devido a sua minimização de variação interna [62].

A primeira data testada foi 15 de março de 2016. Primeiramente foi testado o modelo de correlação de Kawakani *et al.* [1], utilizando os 390.183 alertas. O modelo de Kawakani *et al.*, diferentemente do nosso trabalho, filtra estratégias de ataques que contenham apenas uma assinatura, como solução para reduzir o volume de alertas. Dessa forma, foram descartados 533 alertas. Foram gerados 3 *clusters*, indicando 3 padrões de estratégia de ataque neste dia. O *cluster* 1 é composto por 30 alertas, o *cluster* 2 é composto por 389.594 alertas e o *clusters* 3 é composto por 26 alertas. O *cluster* 2 é o que possui o maior número de alertas e representa 99,98% dos alertas inseridos no modelo. Dos 389.594 alertas pertencentes a este *cluster*, 366.628 são alertas do tipo *Portsweep* ou *Open*

[1] https://www.oracle.com/br/java/index.html
[2] https://mariadb.org/
[3] https://www.r-project.org/

Port, ou seja, relativos majoritariamente a dispositivos que tentam se autenticar no *access-point*. É importante salientar que, se fosse utilizado o método de correlação proposto neste trabalho, estes 366.628 alertas seriam filtrados e, consequentemente, seriam removidos.

O *cluster* 2 teve 38 assinaturas distintas. Foram detectados também 987 endereços IP de origens distintos e 9.395 endereços IP de destino distintos. Nota-se que o modelo de Kawakani *et al.* [1] priorizou a correlação, não distinguindo se o alerta inserido era de baixa ou alta prioridade, se era um alerta verdadeiro ou falso positivo. Um exemplo de tipo de alerta de baixa prioridade que o modelo de Kawakani *et al.* [1] não filtrou é o *"ICMP Destination Unreachable Communication with Destination Host is Administratively Prohibited"*. Este tipo de alerta é gerado quando um tipo específico de pacote ICMP (RFC 2726 [63]) é identificado na rede. Esse pacote ICMP é gerado quando um datagrama não consegue alcançar o seu destino. Esse evento pode até indicar algum problema com o roteamento, mas normalmente não demanda qualquer ação corretiva imediata do administrador de rede e, por isso, é considerado de baixa prioridade [64].

Como exemplo de alertas falsos positivos que o modelo de Kawakani *et al.* [1] não identificou, temos aqueles relacionados à assinatura *"ATTACK-RESPONSES 403 Forbidden"*. Na Seção 4.1 deste trabalho, essa assinatura foi apresentada. O alerta apenas se refere a uma resposta relacionada a uma atualização de um servidor interno da faculdade para outro servidor interno, considerado erroneamente pelo modelo de Kawakani *et al.* como um alerta verdadeiro. Houve 70 ocorrências desta assinatura.

Dando sequência, o método de correlação proposto neste trabalho foi aplicado ao mesmo conjunto de alertas. Os 390.183 alertas passaram pelo processo de filtragem por prioridade e redução de falsos positivos. Restaram apenas 8.980 alertas e estes geraram 4 *clusters*, ou seja, 4 padrões de estratégia de ataque. Percebe-se um número bem reduzido de alertas, otimizando o processo de análise e tomada de decisão nas medidas de segurança. O *cluster* 1 é composto por 8 alertas, o *cluster* 2 é composto por 8.930 alertas, o *cluster* 3 é composto por 36 alertas e o *cluster* 4 é composto por 8 alertas. O *cluster* 3 pode ser visualizado por meio do grafo apresentado na Figura 15.

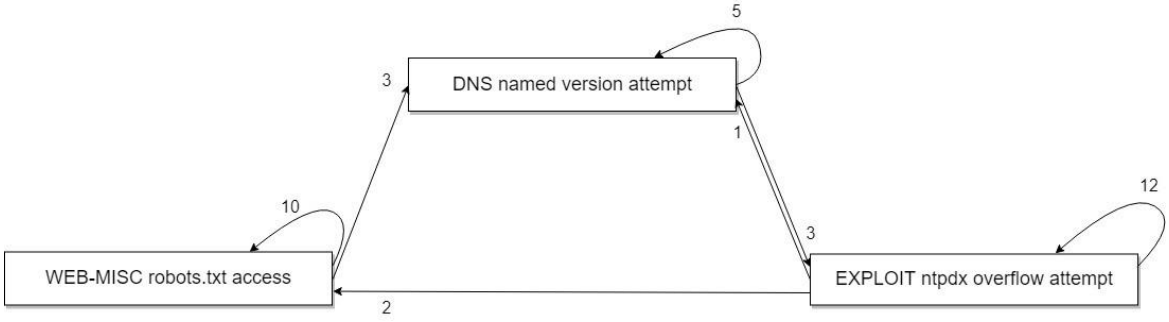

Figura 15 – Grafo de estratégia de ataque do *cluster* 3 de 15 de março de 2016.

Com base no grafo de estratégia de ataque representado na Figura 15, é possível observar que o atacante teve, como objetivo final, um ataque do tipo *overflow* (estouro) no serviço de tempo que é executado em segundo plano, conhecido como NTPX *(Network Time Protocol eXploit)*. Para isto, o mesmo atacante analisou qual parte do servidor *Web* pode ser acessada (*robots.txt*) e qual a versão do servidor DNS (*Domain Name Server*).

Outro ponto importante observado no processo de correlação de alertas proposto foi a elaboração das estratégias de ataque contendo a quantidade correta de alertas. Por exemplo, alertas com a assinatura *"WEB-MSC robots.txt access"* foram considerados corretamente um ataque pelos dois métodos. Porém, o modelo proposto considerou 102 ocorrências desta assinatura como sendo um ataque, enquanto o método de Kawakani *et al.* [1] considerou 98 ocorrências no mesmo dia. Isto deve-se ao fato de o método de Kawakani *et al.* excluir cenários de ataques que contém alertas com apenas uma assinatura, mesmo estes alertas sendo considerados verdadeiros.

A segunda data testada foi 28 de abril de 2016. Seguindo o mesmo procedimento adotado na data anterior, foi testado o modelo de correlação de Kawakani *et al.* [1], onde os 352.740 alertas foram inseridos no correlacionador *off-line*. O modelo considerou 349.063 alertas, descartando 3.677 alertas. Foram gerados 3 *clusters*, indicando 3 padrões de estratégia de ataque neste dia. O *cluster* 1 é composto por 173 alertas e o *cluster* 2 é composto por 13 alertas. Juntos, os dois *clusters* não chegam à 1% do total de alertas. O *cluster* que possui o maior número de alertas é o 3, com 348.875, cerca de 99,94%. Portanto, em um único *cluster*, existem 348.875 alertas, onde 323.350 são alertas do tipo *Portsweep* ou *Open Port*, ou seja, são alertas normalmente de baixa prioridade e que não demandam atenção imediata do administrador. Restariam 25.525 alertas a serem analisados.

Como já havia sido observado na análise do dia 15/03/2016, a falta de uma etapa de filtragem e detecção de falsos positivos prejudica os resultados do trabalho de Kawakani *et al.* Como exemplo de alertas de baixa prioridade, temos os alertas com a assinatura *"SCAN UPnP service discover attempt"*. Este alerta ocorre devido ao serviço SSDP (*Sim- ple Service Discovery Protocol*) do sistema operacional Microsoft Windows estar habilitado por padrão [65].

Segundo Mazerik [66], dispositivos *Universal Plug and Play* (UPnP) são muitos suscetíveis à ataques de negação de serviços distribuídos (DDoS). No artigo de Pincovscy [67], é analisada a relação do protocolo SSDP com ataques DDoS e o potencial deste protocolo neste tipo de ataque. Uma medida que colabora na mitigação de ataques DDoS é bloquear o tráfego que utiliza o serviço UPnP ou simplesmente desabilitar o serviço. Na rede de computadores do estudo de caso este procedimento foi adotado, porém a assinatura deste alerta de baixa prioridade ocorre devido à presença de uma quantidade significativa de *hosts* com o sistema operacional Microsoft Windows. Conforme o *cluster* 3,

houve 26 ocorrências desta assinatura, que foi inserida no padrão de estratégia de ataque, mas não deveria ter sido, pois nesse caso não representa uma informação importante para o analista.

Como outro exemplo de alerta falso positivo, temos novamente os alertas com a assinatura *"ATTACK RESPONSES 403 Forbidden"*. Conforme o *cluster* 3, no estudo realizado, houve 78 ocorrências desta assinatura. Basicamente ocorreu a mesma situação, como no exemplo anterior: esta assinatura está presente nos grafos de estratégia de ataque, porém não corresponde à real situação do ambiente monitorado pois se tratam de alertas falsos positivos.

Na sequência, um outro teste foi realizado utilizando o processo de correlação de alertas proposto neste trabalho. Dos 352.740 alertas referentes ao período em questão, apenas 4.395 alertas foram inseridos na fase de formação de componentes conexos. Os demais alertas foram considerados de baixa prioridade e alertas falsos positivos.

Foram gerados 6 *clusters*, o dobro se comparado com o teste feito com o modelo de correlação de Kawakani *et al.* [1] feito anteriormente, indicando 6 padrões de estratégia de ataque neste dia.

O processo de correlação proposto também gerou um volume menor de assinaturas por *cluster* comparado com o modelo de correlação de Kawakani: apenas 9 assinaturas. Ao analisar os padrões de estratégia de ataque pode-se verificar que o *cluster* 1 é composto por 117 alertas, o *cluster* 2 é composto por 120 alertas, o *cluster* 3 é composto por 12 alertas, o *cluster* 4 é composto por 16 alertas, o *cluster* 5 é o maior com 4.124 alertas e o *cluster* 6 é o menor com apenas 6 alertas. O *cluster* 5 pode ser visualizado por meio do grafo apresentado na Figura 16.

Com base na Figura 16, visualmente se torna simples a compreensão do cenário de ataque para um administrador de redes. O entendimento das estratégias de ataque também foi facilitado por elas conterem um menor número de assinaturas. Neste *cluster* existem 4.124 alertas que passaram pela redução de falsos positivos e chegaram ao final do processo de clusterização.

Analisando as assinaturas, é possível observar que os atacantes representados neste cenário realizaram um processo de varredura para verificar se os alvos estavam respondendo. Logo em seguida realizaram uma tentativa de acesso ao *script* de teste CGI (*Common Gateway Interface*) do servidor *Web*. A assinatura *"BAD-TRAFFIC same SRC/DST"* é referente à ocorrência de tráfego incomum envolvendo alguns endereços IP de origem ou destino, podendo comprometer a rede. Este tipo de assinatura é muito comum em redes com conexões complexas e com volume muito grande de *hosts*.

A terceira data testada foi 7 de maio de 2016, novamente aplicando inicialmente o modelo de correlação de Kawakani *et al.* [1] com a inserção de 206.192 alertas no

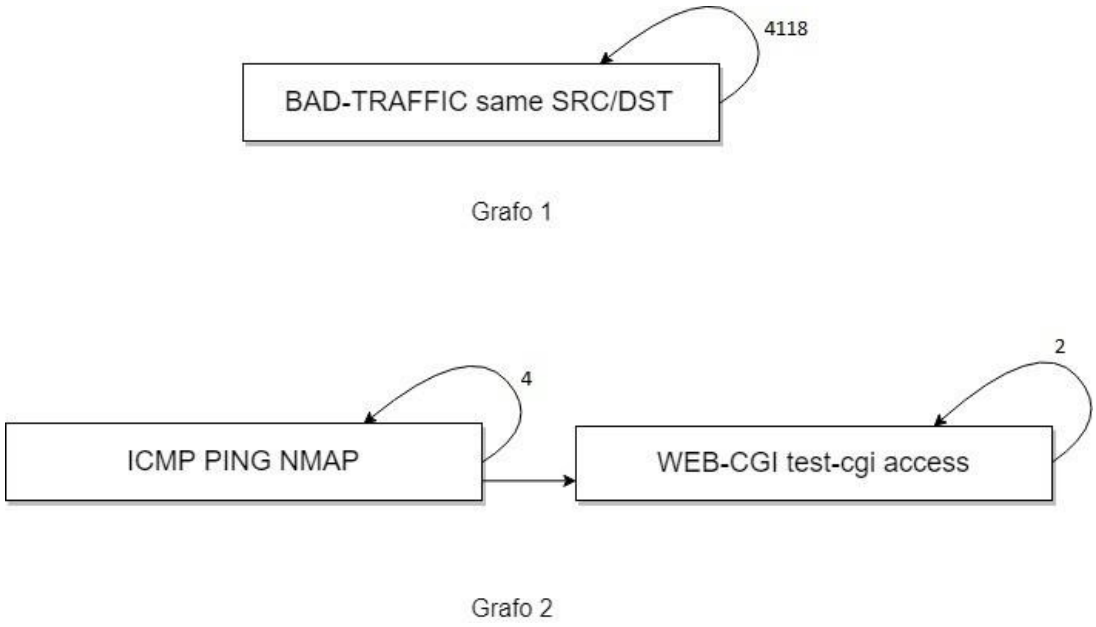

Figura 16 – Grafo de estratégia de ataque do *cluster* 5 de 28 de abril de 2016.

correlacionador *off-line*. O modelo considerou 204.016 alertas, descartando 2.176 alertas. Foram gerados 4 *clusters*, indicando 4 padrões de estratégia de ataque para este dia. O *cluster* 1 é composto por 180 alertas, o *cluster* 2 composto por 203.543 alertas, o *cluster* 3 composto por 279 e o *cluster* 4 composto por 14 alertas. O *cluster* 2 é o que possui o maior número de alertas, com 203.543, representando cerca de 99,77% de todos os alertas inseridos no modelo. Destes alertas, 194.652 são alertas do tipo *Portsweep* ou *Open Port*.

Como um exemplo de alerta de baixa prioridade não filtrado pelo modelo de Kawakani *et al.* [1] temos aqueles com a assinatura *"MS-SQL ping attempt"*, uma espécie de enumerador que procura especificamente por servidores SQL (*Structured Query Language*) da Microsoft, que não requerem autenticação [68]. Este tipo evento é considerado de pouca relevância porque, primeiro, ele simplesmente se trata de uma atividade comum de escaneamento. Apesar de permitir que o atacante obtenha informações importantes para o ataque, ela acaba necessitando de menos atenção que outros alertas mais críticos. Além disso, não há servidores desse tipo na rede.

Como um exemplo de alerta falso positivo não removido pelo modelo de Kawakani *et al.* [1] temos aqueles relacionados à assinatura *"WEB-IIS .htr access"* onde o atacante explora arquivos de fragmentação com extensão *.HTR* para ter acesso aos arquivos presente em servidores Microsoft IIS [69]. Este tipo de alerta foi gerado para a comunicação entre dois servidores internos (Servidor *Web* e o *Mikrotik*) e, um outro detalhe, os dois servidores possuem o sistema operacional Linux.

Na sequência, mais um teste foi realizado utilizando o processo de correlação de alertas proposto neste trabalho. Dos 206.192 alertas referentes ao período em questão,

apenas 424 alertas foram inseridos na fase de formação de componentes conexos, gerando 5 *clusters*. Os demais alertas foram considerados de baixa prioridade e alertas falsos positivos. O *cluster* 4 pode ser visualizado na Figura 17.

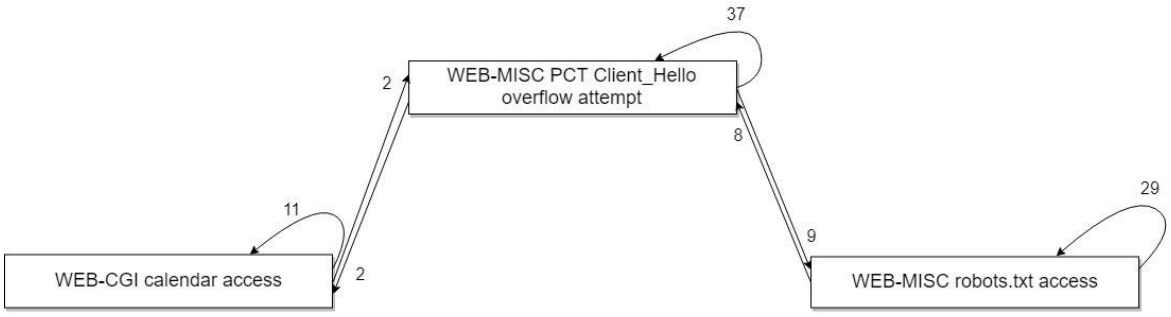

Figura 17 – Grafo de estratégia de ataque do *cluster* 4 de 7 de maio de 2016.

Com base no grafo de estratégia de ataque representado na Figura 17, o atacante realizou uma tentativa de acesso ao arquivo CGI responsável pela gerência do calendário das páginas localizadas no servidor *Web*. Este arquivo CGI contém o nome e a senha do administrador, necessários para manipular as configurações do calendário, segundo o documento CVE-2000-0432 [70]. O atacante também analisou qual ou quais diretórios do servidor *Web* poderiam ser acessados com a assinatura *"WEB-MISC robots.txt access"* e realizou uma tentativa de ataque do tipo *overflow* com a assinatura *"WEB-MISC PCT Client-Hello overflow attempt"*.

Fazendo uma análise comparativa no dia 07/05/2016, tanto o método de Kawakani *et al.* como o processo de correlação proposto geraram estratégias de ataque contendo alertas que continham a assinatura *"WEB-CGI calendar access"*. Porém, no processo proposto, foram computados 122 alertas com esta assinatura, identificados como alertas verdadeiros, enquanto que o método de Kawakani *et al.* computou 68 alertas com esta assinatura, descartando 54 alertas considerados verdadeiros.

A Tabela 6 mostra um resumo da comparação entre o processo de correlação de alertas proposto neste trabalho e o método de correlação de Kawakani *et al.*

Analisando os critérios comparativos apresentados, observa-se que o modelo proposto nesse trabalho tem as seguintes vantagens:

- Minimiza o problema do grande volume de alertas gerados pelo IDS, pois existem processos destinado à filtragem de alertas por prioridade e identificação de falsos positivos por meio de aprendizado de máquina;
- A filtragem de alertas por prioridade proporciona uma visão mais detalhada das estratégias de ataques, pois somente os alertas de caráter urgente ou importante são

Tabela 6 – Tabela comparativa entre os processos de correlação de alertas de intrusão.

Critério comparativo	**Processo de correlação de alertas proposto**	**Método de correlação de Kawakani *et al.***
· Filtragem de alertas de baixa prioridade antes do processo de correlação	· Sim	· Não
· Inserção de informações adicionais ao histórico de alertas	· Sim, funções dos *hosts*, informações sobre geolocalização inseridas por meio de fontes externas	· Inexistente
· Identificação de falsos positivos antes do processo de correlação	· Sim	· Não
· Facilidade na interpretação das estratégias de ataques	· Fácil entendimento	· Médio entendimento. Presença de muitos alertas no mesmo *cluster*

tratados pelo correlacionador na geração dos grafos com os cenários e estratégias de ataque;

- A identificação dos alertas falsos positivos por meio de aprendizado de máquina supervisionado antes do processo de correlação proporciona um histórico de alertas muito próximo da realidade do ambiente computacional estudado, favorecendo a geração de estratégias de ataque mais precisas.

5 CONCLUSÃO

Este trabalho apresentou uma nova abordagem de correlação *off-line* de alertas de intrusão por meio de um processo com 5 fases: redução de alertas de baixa prioridade, filtragem de falsos positivos, agregação de alertas em cenários de ataques, extração de grafos com a estratégia de ataque, unificação das estratégias de ataques em *clusters*. A abordagem obteve sucesso e alcançou os objetivos propostos.

Primeiramente, alertas menos significativos foram removidos da análise, contribuindo para a redução do volume de alertas. Para tanto, foi utilizada a atribuição de prioridades que o próprio IDS realiza. Os alertas classificados como sendo de baixa prioridade pelo IDS foram filtrados.

A etapa de redução de falsos positivos, cujo primeiro passo foi a inserção de dados de outras fontes, também contribuiu para a redução do volume de alertas. Com a inserção dos dados de outras fontes e o apoio do administrador da rede de computadores, foi possível identificar a localização física dos dispositivos e as funções dos *hosts* reportados nos alertas. Esta etapa contribuiu para extração de atributos que fossem realmente relevantes para classificar cada alerta como falso ou verdadeiro.

Em seguida, a identificação e redução de alertas falsos positivos foi alcançada por meio de técnicas de aprendizado de máquina supervisionado. Dois algoritmos foram utilizados e observou-se que ambos atingiram bons resultados: *kNN* e *Random Forest*. Dos algoritmos, o *Random Forest* foi o algoritmo que apresentou, de modo discreto, os melhores resultados quando comparado com o *kNN*, devido à sua capacidade de lidar melhor com classes desbalanceadas.

Com a aplicação do processo proposto, obteve-se estratégias de ataques intuitivas, onde foi possível evidenciar as seguintes contribuições:

- No momento da construção dos cenários com as estratégias de ataques não foram considerados os eventos que não demandavam uma ação corretiva imediata, pois no processo de correlação não havia alertas de baixa prioridade;
- O modelo proposto buscou garantir que os alertas inseridos no correlacionador fossem realmente verdadeiros, incluindo em seu escopo fases de filtragem e identificação de falsos positivos bem estruturados;
- Foi possível identificar eventos específicos como: atualização de informações de servidores, comunicação entre servidores internos, número de ocorrências exatas de determinadas assinaturas de ataques.

- Todos os cenários gerados pelo modelo proposto, mesmo os mais simples, foram capazes de fornecer informações para a extração de estratégias de ataques, pois eles eram formados por eventos de segurança classificados como verdadeiros e de alta prioridade.
- O método forneceu uma visão intuitiva das estratégias de ataques extraídas dos cenários ao administrador da rede, provendo suporte para tomada de decisões relacionadas à segurança da informação.

Como sugestão para trabalhos futuros, pretende-se utilizar, como alternativa, outros algoritmos de clusterização citados neste trabalho para uma análise comparativa: método por particionamento [49, 53, 54] ou baseados em densidade juntamente com função de entropia [50].

Ainda como proposta para trabalhos futuros, pode-se melhorar a etapa de filtragem por prioridade utilizando outros IDS que incluam diferentes níveis de prioridade. Um exemplo é o IDS da empresa Symantec [71], que proporciona sete níveis de prioridade de alertas: informativo, aviso de perigo, evento secundário, evento primário, crítico, fatal e erro. O IDS SNORT possui 3 níveis de prioridade, propondo uma visão mais simples da gravidade de uma tentativa de ataque (alta, média ou baixa). Outra melhoria relacionada à filtragem de prioridade seria a implementação de um procedimento de filtragem que não dependa apenas da atribuição de prioridade determinada pelo IDS, mas também do relacionamento dos eventos de segurança com outros eventos relacionados aos *hosts* da rede monitorada, utilizando, por exemplo, a concatenação dos *logs* do sistema operacional com os alertas de intrusão [72].

REFERÊNCIAS

[1] KAWAKANI, C. T. et al. Intrusion alert correlation to support security management. In: BRAZILIAN COMPUTER SOCIETY. *Proceedings of the XII Brazilian Symposium on Information Systems on Brazilian Symposium on Information Systems: Information Systems in the Cloud Computing Era-Volume 1*. [S.l.], 2016. p. 42.

[2] SILVA, J. M. da et al. Impacto das funções desempenhadas pelos gerentes nos resultados da incubadora: Survey realizada na Rede Mineira de Inovação. *Production*, SciELO Brasil, v. 22, n. 4, p. 718–733, 2012.

[3] NAKAMURA, E. T.; GEUS, P. L. de. *Segurança de Redes em Ambientes Cooperativos*. [S.l.]: Novatec Editora, 2007.

[4] KUROSE, J. F. et al. *Redes de Computadores e a Internet: uma abordagem top-down*. [S.l.]: Pearson, 2010.

[5] JULISCH, K.; DACIER, M. Mining intrusion detection alarms for actionable knowledge. In: ACM. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2002. p. 366–375.

[6] FREITAS, M. P. C. d. Governança de TI. *Gerência de Projetos de Tecnologia da Informação-Unisul Virtual*, 2010.

[7] LYRA, M. R. Segurança e Auditoria em Sistemas de Informação. *Rio de Janeiro: Ciência Moderna*, 2008.

[8] CAMPBELL, C.; CRISTIANINI, N. Simple learning algorithms for training support vector machines. *University of Bristol: Bristol, UK*, 1998.

[9] SILVA, A. E. N. d. *Segurança da Informação: Vazamento de informações*. [S.l.]: Ciência Moderna, 2012.

[10] CERT.BR - Centro de Estudos Resposta e Tratamento de Incidentes de Segurança no Brasil. *Estatísticas dos Incidentes Reportados ao CERT.br*. 2017. Disponível em: <https://www.cert.br/stats/incidentes/>. Acesso em: 20 de novembro de 2017.

[11] GRANADILLO, G. G.; EL-BARBORI, M.; DEBAR, H. New types of alert correlation for security information and event management systems. In: IEEE. *New Technologies, Mobility and Security (NTMS), 2016 8th IFIP International Conference on*. [S.l.], 2016. p. 1–7.

[12] EBRAHIMI, A. et al. Automatic attack scenario discovering based on a new alert correlation method. In: IEEE. *Systems Conference (SysCon), 2011 IEEE International*. [S.l.], 2011. p. 52–58.

[13] ALVARENGA, S. C. d.; ZARPELÃO, B. B.; MIANI, R. S. Discovering attack strategies using process mining. In: *The Eleventh Advanced International Conference on Telecommunications*. [S.l.: s.n.], 2015. p. 119–125.

[14] ELSHOUSH, H. T. I. An innovative framework for collaborative intrusion alert correlation. In: IEEE. *Science and Information Conference (SAI), 2014*. [S.l.], 2014. p. 607–614.

[15] VERMA, R. et al. Security analytics: essential data analytics knowledge for cybersecurity professionals and students. *IEEE Security & Privacy*, IEEE, v. 13, n. 6, p. 60–65, 2015.

[16] ANDERSON, J. P. Computer security threat monitoring and surveillance. *Technical Report, James P. Anderson Company*, 1980.

[17] DENNING, D.; NEUMANN, P. G. *Requirements and model for IDES - A Real-time Intrusion Detection Expert System*. [S.l.]: SRI International, 1985.

[18] CASWELL, B.; BEALE, J. *Snort 2.1 intrusion detection*. [S.l.]: Syngress, 2004.

[19] LHOTSKY, B. *Instant OSSEC host-based intrusion detection system*. [S.l.]: Packt Publishing Ltd, 2013.

[20] POTLURI, S.; DIEDRICH, C. High performance intrusion detection and prevention systems: A survey. In: ACADEMIC CONFERENCES AND PUBLISHING LIMITED. *ECCWS2016-Proceeding for the 15th European Conference on Cyber Warfare and Security*. [S.l.], 2016. p. 260.

[21] PIETRO, R. D.; MANCINI, L. V. *Intrusion detection systems*. [S.l.]: Springer Science & Business Media, 2008. v. 38.

[22] BACE, R. G. *Intrusion detection*. [S.l.]: Sams Publishing, 2000.

[23] BURTON, J. D. *Cisco security professional's guide to secure intrusion detection systems*. [S.l.]: Syngress Publ., 2003.

[24] PAQUET, C. *Implementing Cisco IOS network security (IINS):(CCNA security exam 640-553)(authorized self-study guide)*. [S.l.]: Pearson Education India, 2009.

[25] GHORBANI, A. A.; LU, W.; TAVALLAEE, M. *Network intrusion detection and prevention: concepts and techniques*. [S.l.]: Springer Science & Business Media, 2009. v. 47.

[26] UPPULURI, P.; SEKAR, R. Experiences with specification-based intrusion detection. In: SPRINGER. *Recent Advances in Intrusion Detection*. [S.l.], 2001. p. 172–189.

[27] LYDON, A. *Compilation for Intrusion Detection Systems*. Tese (Doutorado) — Ohio University, 2004.

[28] SILVEIRA, K. *Os desafios para os sistemas de detecção de intrusos (IDS)*. Boletim bimestral sobre tecnologia de redes produzido e publicado pela RNP – Rede Nacional de Ensino e Pesquisa, 2000. Disponível em: <https://memoria.rnp.br/newsgen/0011/ids.html>. Acesso em: 20 de novembro de 2017.

[29] ZOCHIO, M. *Introdução à Criptografia: Uma abordagem prática*. NOVATEC, 2016. ISBN 9788575225158. Disponível em: <https://books.google.com.br/books?id=kOnZDAAAQBAJ>. Acesso em: 20 de novembro de 2017.

[30] BROWN, L.; STALLINGS, W. *Segurança de Computadores: Princípios e Práticas*. Elsevier Brasil, 2017. ISBN 9788535264500. Disponível em: <https://books.google.com.br/books?id=y2DcAwAAQBAJ>. Acesso em: 20 de novembro de 2017.

[31] ZHAI, Y.; NING, P.; XU, J. Integrating ids alert correlation and os-level dependency tracking. In: SPRINGER. *ISI*. [S.l.], 2006. p. 272–284.

[32] MIZOGUCHI, F. Anomaly detection using visualization and machine learning. In: IEEE. *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2000.(WET ICE 2000). Proeedings. IEEE 9th International Workshops on*. [S.l.], 2000. p. 165–170.

[33] ABAD, C. et al. Log correlation for intrusion detection: A proof of concept. In: IEEE. *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*. [S.l.], 2003. p. 255–264.

[34] SILVA, A.; GUELFI, A. Sistema para identificação de alertas falso positivos por meio de análise de correlacionamentos e alertas isolados. *the 9th IEEE I2TS*, 2010.

[35] SHITTU, R. O. *Mining intrusion detection alert logs to minimise false positives & gain attack insight*. Tese (Doutorado) — City University London, 2016.

[36] VALEUR, F. et al. Comprehensive approach to intrusion detection alert correlation. *IEEE Transactions on dependable and secure computing*, IEEE, v. 1, n. 3, p. 146–169, 2004.

[37] DEBAR, H.; WESPI, A. Aggregation and correlation of intrusion-detection alerts. In: SPRINGER. *Recent Advances in Intrusion Detection*. [S.l.], 2001. p. 85–103.

[38] VALDES, A.; SKINNER, K. Probabilistic alert correlation. In: SPRINGER. *International Workshop on Recent Advances in Intrusion Detection*. [S.l.], 2001. p. 54–68.

[39] LEE, S. et al. Real-time analysis of intrusion detection alerts via correlation. *Computers & Security*, Elsevier, v. 25, n. 3, p. 169–183, 2006.

[40] TREINEN, J. J.; THURIMELLA, R. A framework for the application of association rule mining in large intrusion detection infrastructures. In: SPRINGER. *International Workshop on Recent Advances in Intrusion Detection*. [S.l.], 2006. p. 1–18.

[41] ZURUTUZA, U. et al. Combined data mining approach for intrusion detection. In: *SECRYPT*. [S.l.: s.n.], 2007. p. 67–73.

[42] SOLEIMANI, M.; GHORBANI, A. A. Critical episode mining in intrusion detection alerts. In: IEEE. *Communication Networks and Services Research Conference, 2008. CNSR 2008. 6th Annual*. [S.l.], 2008. p. 157–164.

[43] TAHA, A. E. et al. Agent based correlation model for intrusion detection alerts. In: IEEE. *Intelligence and Security Informatics (ISI), 2010 IEEE International Conference on*. [S.l.], 2010. p. 89–94.

[44] YANG, L. et al. Alerts analysis and visualization in network-based intrusion detection systems. In: IEEE. *Social Computing (SocialCom), 2010 IEEE Second International Conference on*. [S.l.], 2010. p. 785–790.

[45] LIU, L.; ZHENG, K. F.; YANG, Y. X. An intrusion alert correlation approach based on finite automata. In: IEEE. *Communications and Intelligence Information Security (ICCIIS), 2010 International Conference on*. [S.l.], 2010. p. 80–83.

[46] SAAD, S.; TRAORE, I. A semantic analysis approach to manage ids alerts flooding. In: IEEE. *Information Assurance and Security (IAS), 2011 7th International Conference on*. [S.l.], 2011. p. 156–161.

[47] SHITTU, R. et al. Visual analytic agent-based framework for intrusion alert analysis. In: IEEE. *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2012 International Conference on*. [S.l.], 2012. p. 201–207.

[48] FAYYAD, S.; MEINEL, C. Attack scenario prediction methodology. In: IEEE. *Information Technology: New Generations (ITNG), 2013 Tenth International Conference on*. [S.l.], 2013. p. 53–59.

[49] HACHMI, F.; LIMAM, M. A two-stage process based on data mining and optimization to identify false positives and false negatives generated by intrusion detection systems. In: IEEE. *Computational Intelligence and Security (CIS), 2015 11th International Conference on*. [S.l.], 2015. p. 308–311.

[50] GHASEMIGOL, M.; GHAEMI-BAFGHI, A. E-correlator: an entropy-based alert correlation system. *Security and Communication Networks*, Wiley Online Library, v. 8, n. 5, p. 822–836, 2015.

[51] CHAKIR, E. M.; MOUGHIT, M.; KHAMLICHI, Y. I. An efficient method for evaluating alerts of intrusion detection systems. In: IEEE. *Wireless Technologies, Embedded and Intelligent Systems (WITS), 2017 International Conference on*. [S.l.], 2017. p. 1–6.

[52] SHITTU, R. et al. Outmet: A new metric for prioritising intrusion alerts using correlation and outlier analysis. In: IEEE. *Local Computer Networks (LCN), 2014 IEEE 39th Conference on*. [S.l.], 2014. p. 322–330.

[53] VIDAL, J. M.; OROZCO, A. L. S.; VILLALBA, L. J. G. Quantitative criteria for alert correlation of anomalies-based NIDS. *IEEE Latin America Transactions*, IEEE, v. 13, n. 10, p. 3461–3466, 2015.

[54] HACHMI, F.; BOUJENFA, K.; LIMAM, M. A three-stage process to detect outliers and false positives generated by intrusion detection systems. In: IEEE. *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*. [S.l.], 2015. p. 1749–1755.

[55] KO, R. K.; LEE, S. S.; LEE, E. W. Business Process Management (BPM) Standards: A Survey. *Business Process Management Journal*, Emerald Group Publishing Limited, v. 15, n. 5, p. 744–791, 2009.

[56] DINIZ, F. A.; SILVA, T. R. da; ALENCAR, F. E. S. Um estudo empírico de um sistema de reconhecimento facial utilizando o classificador KNN. *Revista Brasileira de Computação Aplicada*, v. 8, n. 1, p. 50–63, 2016.

[57] DIETTERICH, T. G. Ensemble methods in machine learning. In: SPRINGER. *International workshop on multiple classifier systems*. [S.l.], 2000. p. 1–15.

[58] GUEDES, A. R. M.; GUIMARÃES, V. L. *Sistema de Reconhecimento Baseado em Random Forest para Caracteres de Captchas*. Universidade Federal de Ouro Preto - Departamento de Computação, 2014. Disponível em: <http://www.decom.ufop.br/menotti/rp142/trab/trab1-dp2-artigo.pdf>. Acesso em: 20 de novembro de 2017.

[59] LOPES, T. D. et al. Aplicacão do algoritmo Random Forest como classificador de padrões de falhas em rolamentos de motores de indução. In: *Simpósio Brasileiro de Automação Inteligente*. [S.l.: s.n.], 2017.

[60] SILVESTRE, R. *Comparação da Florística, Estrutura e Padrão Espacial em três Fragmentos de Floresta Ombrófila Mista no Estado do Paraná*. Dissertação (Mestrado) — Universidade Federal do Paraná, 2013.

[61] METZ, J.; MONARD, M. C. Clustering Hierárquico: Uma metodologia para auxiliar na interpretação dos clusters. *In: XXV Congresso da Sociedade Brasileira de Computação, Anais do Encontro Nacional de Inteligência Artificial, 2005 jul 22-29; São Leopoldo, BR*, 2005. Disponível em: <http: //bibliotecadigital.sbc.org.br/bibliotecadigital>. Acesso em: 10 de março de 2018.

[62] HAIR, J. *Análise Multivariada de Dados*. [S.l.]: Porto Alegre: Bookman, 2005.

[63] NORDMARK, E. Stateless IP/ICMP Translation Algorithm (SIIT). RFC Editor, n. 2765, p. 1–26, February 2000. ISSN 2070-1721. Disponível em: <http://www.rfc-editor.org/rfc/pdfrfc/rfc2765.txt.pdf>. Acesso em: 10 de março 2018.

[64] ALDEID.COM. *Snort-alerts/ICMP-Destination-Unreachable-Communication-with-Destination-Host-is-Administratively-Prohibited*. 2010. Disponível em: <https://www.aldeid.com/wiki/Snort-alerts/ ICMP-Destination-Unreachable-Communication-with-Destination-Host-is-\ Administratively-Prohibited#Identification>. Acesso em: 10 de março de 2018.

[65] LEE, C.; HELAL, S. Protocols for service discovery in dynamic and mobile networks. *International Journal of Computer Research*, v. 11, n. 1, p. 1–12, 2002.

[66] MAZERIK, R. *DDoS on UPnP Devices*. 2015. Disponível em: <https: //resources.infosecinstitute.com/ddos-upnp-devices/#gref>. Acesso em: 10 de março de 2018.

[67] PINCOVSCY, J. A. Ataques de negação de serviço por reflexão amplificada utilizando o protocolo SSDP. 2016. CENTRO DE INSTRUÇÃO DE GUERRA ELETRÔNICA. Disponível em: <http://bdex.eb.mil.br/jspui/bitstream/1/1012/1/ Pincovscy_TCC.pdf>. Acesso em: 10 de março de 2018.

[68] STIAWAN, D. et al. Penetration testing and mitigation of vulnerabilities Windows Server. *IJ Network Security*, v. 18, n. 3, p. 501–513, 2016.

[69] RIECK, K.; LASKOV, P. Detecting unknown network attacks using language models. In: SPRINGER. *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. [S.l.], 2006. p. 74–90.

[70] MASSICOTTE, F.; LABICHE, Y. An analysis of signature overlaps in intrusion detection systems. In: IEEE. *Dependable Systems & Networks (DSN), 2011 IEEE/IFIP 41st International Conference on*. [S.l.], 2011. p. 109–120.

[71] SYMANTEC ENDPOINT PROTECTION. *Níveis de gravidade de alertas e eventos*. 2018. Symantec. Disponível em: <https://support.symantec.com/pt_BR/article.HOWTO124337.html#v112562035>. Acesso em: 20 de julho de 2018.

[72] PRELUDE IDS. *Prelude Compatibility - Manual User*. [S.l.], 2018. Communication System. Disponível em: <https://www.prelude-siem.org/projects/prelude/wiki/PreludeCompatibility>. Acesso em: 20 de julho de 2018.

TRABALHOS PUBLICADOS PELO AUTOR

Trabalhos publicados pelo autor durante o programa.

1. Moraes, E. A., Tojeiro, C. A. C., Miani, R. S., Zarpelão, B. B. **Análise de Alertas de Sistemas de Detecção de Intrusão: Uso de Aprendizado Supervisionado na Redução de Alertas Falsos Positivos**, XVII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais: SBSEG 2017, SBC - Sociedade Brasileira de Computação, p. 182-195, Brasília - DF, 2017, (Qualis CC 2017, B3).