POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Relatório Prático

Mestrado em Cibersegurança e Informática Forense

Gestão e Análise de Relatórios de Segurança

Sistemas SIEM em Contexto Organizacional

Filipe Henriques, 2180066

Jéssica Pedrosa, 2180067

Patrícia Silva, 2180068

Tiago Martins, 2182716

Leiria, maio de 2019

Resumo

Dado o aumento da ocorrência de ataques, é de facto relevante para as empresas, não só apresentarem respostas eficazes aos eventos adversos que os seus ativos sofrem, mas também possuir alguma solução que permita uma deteção rápida desses mesmos eventos indesejados. Deste modo, a existência e utilização de sistemas de SIEM (*Security Information and Event Management*) torna-se um elemento-chave para as empresas, tornando-se importante nos dias de hoje perceber o funcionamento destes num contexto organizacional. Assim, para a realização deste trabalho, foi criado um cenário personalizado no qual foram realizados ataques num período temporal previamente definido para, posteriormente, se observar a deteção dos mesmos pelo sistema de SIEM selecionado para efeitos do presente trabalho. Deste modo, com a realização de alguns ataques espera-se perceber alguns aspetos, como por exemplo, de que modo estes sistemas, através dos dados que recolhem, identificam a existência de um incidente de segurança.

Palavras-chave: ataque, evento, SIEM, *Wazuh*, alerta

Esta página foi intencionalmente deixada em branco

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

API - *Application programming interface*

CPU - *Central Process Unit*

GUI – *Graphic User Interface*

IDS - Sistema de Deteção de Intrusões

IP – *Internet Protocol*

IT - *Information Technology*

PHP - *Hypertext Preprocessor*

SIEM - *Security Information and Event Management*

SO – Sistema Operativo

SSH - *Secure Shell*

Índice

1. Introdução

No âmbito da unidade curricular de Gestão e Análise de Relatórios de Segurança, do Mestrado Cibersegurança e Informática Forense da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, foi elaborado o presente relatório de modo a elaborar uma análise da segurança de sistemas em contexto organizacional com o recurso a sistemas de SIEM.

Com o aumento da realização de ataques contra sistemas informáticos de organizações e empresas, estas sentem a necessidade de realizar uma gestão objetiva dos eventos ocorridos de modo a simplificar a observação e análise de eventuais ataques e, assim, eventualmente, auxiliar a resolução de possíveis problemas provocados pelos ataques executados. Deste modo, para colmatar essa referida necessidade, as organizações, nos dias de hoje, recorrem a sistemas de SIEM que recolhem dados, podem efetuar o processamento dos dados recolhidos, realizando o armazenamento destes e correlacionando a informação permitindo identificar incidentes de segurança e, eventualmente, dar uma resposta aos mesmos. Alguns dos sistemas de SIEM existentes no mercado permitem também realizar a análise dos dados recolhidos sendo que, alguns destes, permitem ainda que as organizações criem regras personalizadas de modo a despoletar alertas de acordo com as diferentes situações.

No contexto deste trabalho, de modo a se entender melhor o funcionamento de um sistema de SIEM, foi elaborado um cenário em que se utiliza um sistema de SIEM, máquinas que são monitorizadas e atacadas, sendo que estes mesmos ataques ocorrem num determinado período de tempo definido. Posteriormente, para permitir um melhor entendimento dos eventos ocorridos no cenário criado, bem como da interpretação destes efetuada pelo sistema de SIEM, será realizada uma análise/observação dos resultados obtidos. O SIEM em si foi disponibilizado na entrega deste trabalho, na forma de máquinas virtuais em *pens* para efeitos de avalização.

O presente relatório está dividido em 7 capítulos, sendo que o capítulo 2 apresenta o conceito de sistema de SIEM e, por sua vez, a arquitetura de um sistema de SIEM e os objetivos, vantagens e desvantagens da utilização de um sistema de SIEM; é ainda apresentado tanto o estudo comparativo realizado entre sistemas de SIEM bem como o sistema de SIEM escolhido e a razão dessa mesma escolha. Já o capítulo 3 expõe a

infraestrutura tecnológica do sistema de SIEM selecionado para o presente trabalho bem como outros recursos que possam ser necessários para a instalação. O capítulo 4 expõe o processo de instalação do sistema de SIEM selecionado para o trabalho. Relativamente ao cenário de testes, este encontra-se descrito no capítulo 5. No capítulo 6 são apresentados os ataques que foram realizados, a monitorização e análise dos eventos ocorridos no *plugin* para o *Kibana* e, por fim, a definição e configuração de alertas no sistema de SIEM. Já no capítulo 7 é abordado o processo de criação do *data set* e ainda a análise realizada ao mesmo. Por fim, no capítulo 8 encontram-se expostas as conclusões obtidas com a realização deste trabalho.

2. Estado da Arte

Uma vez que a temática do presente trabalho se prende com o estudo de sistemas de SIEM, neste capítulo pareceu relevante abordar o conceito de sistema de SIEM, bem como os seus objetivos, principais características, vantagens e desvantagens da utilização do mesmo. Também será apresentada tanto a arquitetura, como os principais componentes de um sistema de SIEM. Para além disto, será ainda exposto o estudo comparativo de sistemas de SIEM que foi realizado e, por conseguinte, que sistema de SIEM foi selecionado para este trabalho e a razão que levou a essa mesma escolha.

2.1 Sistema de SIEM

Um sistema de SIEM, é um sistema que oferece vários serviços, nomeadamente, gestão de *logs*, conformidade regulatória com sistemas de IT (*Information Technology*), correlação de eventos, respostas ativas e segurança nos *endpoints*. [1]

A gestão de *logs* num sistema de SIEM envolve a configuração de diferentes *nodes*, *nodes* estes que podem ser tanto *hardware* (como um *router*) ou *software* (como um sistema ou uma aplicação) no sistema de IT. Estes mesmos *nodes* enviam eventos relevantes ocorridos nestes para uma base de dados centralizada. A gestão desta base de dados é realizada através de uma aplicação de SIEM que processa e normaliza os dados dos eventos recebidos dos *nodes*. Este serviço permite ao sistema de SIEM efetuar análises em tempo real e de *data mining* sobre o desempenho do comportamento e segurança de todos os sistemas de IT a enviarem dados para o sistema de SIEM. [1]

No serviço de conformidade regulatória com o IT são executadas as auditorias sobre os ficheiros *log* recebidos dos sistemas geridos no serviço anterior, através da aplicação de filtros/regras e de temporizadores, no qual se verificam registos como a frequência das alterações de *password*, de *updates* nos antivírus e *antispywares*, de forma a regular a conformidade com as regras da organização. É também realizada a identificação de violações na conformidade sobre os requisitos impostos pela organização, através dos filtros ou regras criadas para validar essas conformidades. [1]

Com o serviço de correlação de eventos é aumentado o nível de inteligência adquirido pela organização. O que a correlação de eventos permite fazer é ensinar o sistema de

SIEM a considerar outras condições antes de este decidir ativar um alerta. Por exemplo, quando o desempenho do CPU (*Central Process Unit*) de um servidor alcança uma alta percentagem, este aumento pode ser causado por diferentes fatores. Assim, com a correlação de eventos é possível ao sistema de SIEM efetuar a investigação e considerar outros eventos que, apesar de não estarem diretamente relacionados com o problema do CPU, irão ajudar a desenhar uma melhor imagem sobre o estado do servidor através da remoção de possíveis causas do problema. [1]

O serviço de resposta ativa é aquele que tira partido de todos os outros serviços anteriores. Com toda a informação recebida e as regras estabelecidas e correlacionadas, um utilizador neste serviço tem a opção de responder diretamente ao evento ou de deixar que o sistema de SIEM automaticamente tome uma ação corretiva perante o evento – seja este uma ameaça ou uma má configuração. Esta última opção deve ser apenas utilizada caso o sistema de SIEM instalado esteja devidamente implementado para tomar tais medidas corretivas. [1]

Por fim, o serviço de segurança nos *endpoints* permite validar o estado da segurança nos vários sistemas da organização. Isto envolve monitorizar, por exemplo, se a *firewall* num computador está ativa, se um servidor está ligado, verificar as últimas alterações nos antivírus ou identificar quando um *node* fique infetado com um *spyware*. [1]

Do ponto de vista empresarial, os sistemas de SIEM são grandes ferramentas para ajudar no aumento da segurança e auxiliar na administração dos sistemas tecnológicos da organização. Estes também ajudam as organizações a cumprirem certas legislações governamentais e comerciais, evitando assim coimas que poderão prejudicar o negócio da organização. [1]

2.1.1 Arquitetura de um Sistema de SIEM

Como se pode visualizar na Figura 1, um sistema de SIEM é uma ferramenta que recolhe e correlaciona eventos que ocorrem numa rede. Permite ainda a criação de regras e alertas para possibilitar a identificação de eventos fora do normal, sendo tudo isto sempre de forma centralizada.

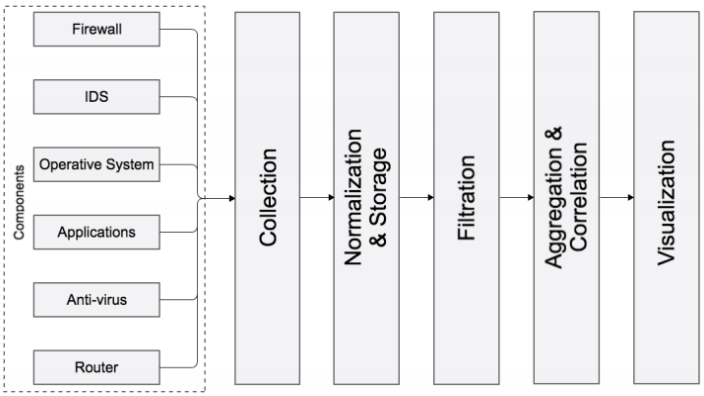


Figura 1 - Arquitetura de um sistema SIEM [2]

A arquitetura do sistema de SIEM tem como fonte de recolha de dados os vários componentes na rede, tais como *firewalls*, antivírus, *routers*, entre outros. Estes componentes, como se pode ver na Figura 1, são tanto *hardware* como *software*, que interagem com o sistema de SIEM, o que faz com que os dados recolhidos venham em vários formatos. É na fase de normalização e armazenamento que estes dados são, portanto, normalizados e guardados numa base de dados. Só depois disto é que são aplicados os filtros para remover dados que não sejam relevantes para a organização. Após estas fases, o sistema de SIEM está pronto para efetuar a agregação dos dados e aplicar a correlação de eventos, sendo os resultados visualizados num *dashboard* sob a forma de alertas, gráficos ou relatórios. [2]

2.1.2 Objetivos, vantagens e desvantagens da utilização de sistemas de SIEM

Os objetivos principais na utilização de um sistema de SIEM são aumentar a segurança da rede e dos seus *endpoints*, melhorar o funcionamento da rede e simplificar o trabalho de administração na organização. [3]

Uma das grandes vantagens da utilização de um sistema de SIEM está no uso de *dashboards* ou interfaces gráficas que são disponibilizadas por uma grande parte dos sistemas de SIEM existentes. Estes oferecem uma melhor visualização do estado da rede na qual o sistema de SIEM está a monitorizar. [4] Outra vantagem dos sistemas de SIEM é o facto de estes organizarem e centralizarem uma grande quantidade de eventos, o que possibilita responder de forma rápida às ameaças encontradas. Deste modo, é possível

avaliar de forma eficiente o estado de segurança da infraestrutura informática da organização. [5]

Por outro lado, a utilização de um sistema de SIEM tem as suas desvantagens, nomeadamente, o facto de estes serem bastantes caros e difíceis de administrar. [5] Isto deve-se ao facto dos sistemas de SIEM, normalmente, lidarem com uma grande quantidade de dados o que leva ao aumento da quantidade de falsos positivos influenciando, assim, o desempenho da resposta eficaz contra situações de ameaça. [6]

Um outro problema é a indução de uma falsa sensação de conforto causada aos analistas do sistema de SIEM. Tal faz com que estes deixem de utilizar outros métodos diferentes para a auditoria da infraestrutura informática da organização, deixando assim uma vulnerabilidade na infraestrutura. [7]

É claro que a complexa implementação e gestão de um sistema de SIEM justifique a sua utilização ocorrer maioritariamente em organizações de grande e média dimensão, sendo que a manutenção e os custos elevados inerentes ao funcionamento deste sistema afaste as organizações mais pequenas e com menos recursos para implementarem uma solução SIEM. Para além disso, a instalação deste tipo de ferramentas exige uma análise profunda da rede informática, requer *hardware* para suportar a grande quantidade de processamento e armazenamento de dados e ainda equipas de técnicos especializados apenas para monitorizar e melhorar este sistema de SIEM. [3]

2.2 Estudo Comparativo

Ainda no contexto deste trabalho, foi efetuado um estudo comparativo de um conjunto de plataformas selecionadas, nomeadamente, o *Wazuh*, o *SIEMonster* e ainda o *Graylog*. Este estudo comparativo consiste em efetuar o levantamento das funcionalidades e características apresentadas por estes e assim perceber as vantagens e desvantagens dos mesmos. A Tabela 1 apresenta as funcionalidades gerais presentes ou não em cada uma das três plataformas anteriormente referidas.

Tabela 1 - Estudo comparativo entre as plataformas *Wazuh*, *SIEMonster* e *Graylog*

Funcionalidades	**Comparação entre *Wazuh*, *SIEMonster* e *Graylog***		
	Wazuh	***SIEMonster***	***Graylog***
Licença	*Open Source*	*Community Edition*	*Open Source*
Análise de *logs*	Sim	Sim	Sim
Deteção de intrusões/anomalias	Sim	Sim	Não
Identificação de vulnerabilidades	Sim	Sim	Não
Despoleta alertas	Sim	Sim	Sim
Resposta a Incidentes	Sim	Sim	Não
Relatórios	Sim	Não	Não

É de ter em consideração que a recolha de informação para este estudo foi efetuada nos sítios oficiais das plataformas, sendo que, grande parte da informação apresentada correspondia a informação comercial.

Através da informação disponibilizada conseguiu-se perceber que muitas funcionalidades que o *Wazuh* apresenta são gratuitas e estas estão presentes tanto na versão gratuita, como nas versões pagas. Contudo, no caso do *SIEMonster* e do *Graylog*, as funcionalidades que estes apresentam variam da versão gratuita para a paga. O *Wazuh*, o *Graylog* e o *SIEMonster* são *open source* e possuem a versão gratuita. É de mencionar que para a recolha de informação para a Tabela 1 foi considerada unicamente a versão gratuita destas três plataformas.

Relativamente ao *Graylog* e à funcionalidade de deteção de intrusões/anomalias, aparentemente o *Graylog*, por *default*, não se comporta como um IDS (Sistema de Deteção de Intrusões), no entanto, é possível que este se comporte também como um IDS através de complementos como o *Snort*. [8] O *SIEMonster* aparentemente consegue detetar intrusões/anomalias através da funcionalidade *Threat Intelligence* que permite, por sua vez, parar ataques em tempo real.

É de referir que estas três soluções apresentam outras opções escaláveis ao negócio sendo que, em alguns casos, como o da plataforma *Wazuh*, é necessário contactar a empresa proprietária da plataforma para se saber mais detalhes.

Observando os dados recolhidos sobre as três plataformas verifica-se que a solução gratuita mais completa é o *Wazuh* uma vez que este disponibiliza todas as funcionalidades levadas em consideração neste estudo comparativo. Por outro lado, a solução mais incompleta acaba mesmo por ser o *Graylog*, sendo que este não apresentou a maioria das funcionalidades da tabela. Deste modo, dado os resultados obtidos com o estudo comparativo efetuado, o sistema SIEM selecionado para o projeto é o *Wazuh*.

3. Infraestrutura tecnológica e recursos necessários para a instalação do SIEM

A arquitetura do *Wazuh* baseia-se no envio de registos *log* de uma dada máquina para um servidor, de modo a, posteriormente, esses dados serem visualizados e analisados. Todavia, há que ter em conta que, existem alguns cenários possíveis para a arquitetura do *Wazuh*, sendo de seguida apresentados dois deles. É de referir que os componentes existentes em ambos os cenários apresentam um dado conjunto de elementos como se verá posteriormente. [9]

Um dos cenários de arquitetura apresenta três componentes - nomeadamente o *Wazuh server*, o *Elastic Stack* e o *Monitored server*. Neste cenário, o *Wazuh server* é composto pelo *Wazuh manager*, *Filebeat* e o *Wazuh API*; o *Elastic Stack* é constituído pelos elementos *Logstash*, *Elasticsearch*, *Kibana* e ainda a *Wazuh App*; o componente *Monitored server* tem como elemento o *Wazuh agent*, podendo ser visto, de uma forma simplista, como a máquina responsável pelo envio de registos *log* para o sistema de SIEM, ou seja, a máquina que está a ser monitorizada pelo sistema de SIEM. Neste cenário, o *Monitored server* envia os dados para o servidor, servidor este que posteriormente analisa e envia os resultados obtidos para o *Elasticsearch*, que se encontra incluído no componente *Elastic Stack*. Relativamente ao elemento *Filebeat* presente no componente *Wazuh server*, este será usado para o envio seguro de alertas e/ou eventos para o *Elastic Stack*. É de referir que este elemento é utilizado dado o *Elastic Stack* e o *Wazuh Server* estarem em máquinas diferentes. Este cenário apresentado é um cenário simples, podendo existir, caso se pretenda, mais *Monitored servers*. Em alguns casos, por exemplo, se o volume de dados for elevado, poderá ser usada mais do que uma máquina com o *Elastic Stack*. [9]

Um cenário ainda menos complexo que o anteriormente apresentado encontra-se na Figura 2. O cenário que se encontra na referida figura corresponde ao cenário que será adotado para este projeto. Este cenário consiste em o *Wazuh server*, para além de apresentar os seus elementos, apresenta também os elementos do *Elastic Stack*. Neste caso, o elemento *Filebeat* já não é utilizado pois os elementos do *Elastic Stack* encontram-se no *Wazuh Server*. É de referir que nas secções seguintes deste relatório, o componente *Monitored server*, será referido como *Wazuh agent*.

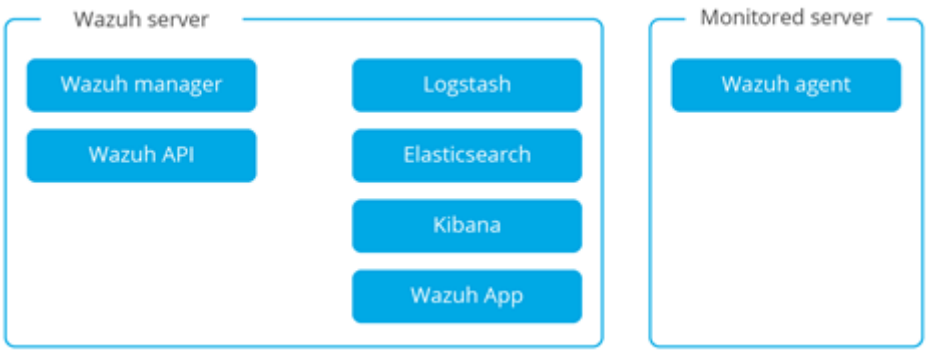


Figura 2 - Um dos cenários de arquitetura do *Wazuh* [9]

De modo a se compreender de uma forma mais clara os processos executados pelos três componentes bem como o fluxo dos dados entre os mesmos, pode-se consultar a Figura 3. Nesta figura, é possível visualizar tanto os processos executados, como também os *sockets* que são utilizados para algumas comunicações. De uma forma geral, pode-se observar que os processos do *agent* enviam informação a um dos processos do *manager*, sendo o *manager* também o responsável pelo envio de alertas e eventos. A informação acaba depois por ser enviada para um dos processos do *Elastic*, mais especificamente, o *Logstach* que efetua o mapeamento dos dados para o *Elasticsearch* que, por sua vez, remete os dados para o *Kibana* que utiliza o *Wazuh App* para realizar *queries* à *RESTful API*. [9]

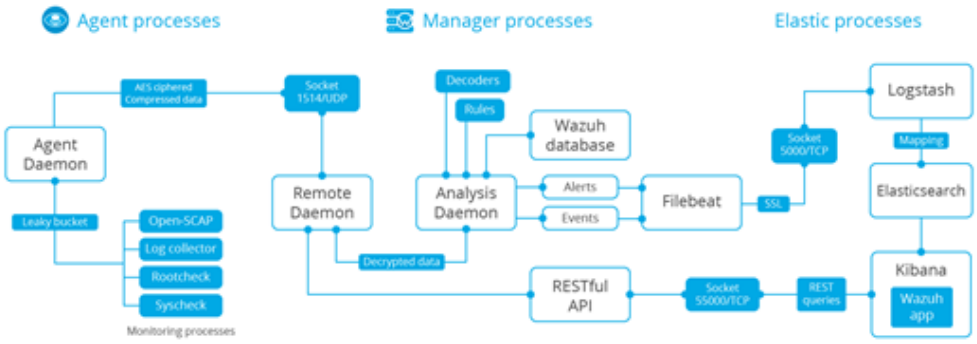


Figura 3 - Fluxo dos dados entre os componentes da arquitetura do *Wazuh* [9]

4. Instalação, Configuração do SIEM e Respetiva Ligação aos *Agents*

Para a instalação do *Wazuh*, optou-se por uma arquitetura *single-host*, onde uma máquina que desempenha o papel de servidor *Wazuh* contém o *Wazuh Manager*, o *Wazuh API* e o *Elastic Stack* na mesma, como foi representado na Figura 2.

Na instalação do servidor *Wazuh*, foi utilizada uma máquina virtual com o sistema operativo *CentOS 7*. Nesta máquina foram instalados o *Wazuh Manager*, *Wazuh API* e o *Elastic Stack* que consiste, por exemplo, no *Elasticsearch* e no *Kibana* com o *Wazuh App* incorporado neste. Depois, para a instalação do *software* de monitorização das máquinas monitorizadas, foi instalado o *Wazuh Agent* em duas máquinas virtuais - uma máquina com o sistema operativo *Windows 7* e outra com o *Ubuntu 19.04*. Este processo de instalação baseou-se na documentação do *Wazuh 3.8* focada para a distribuição *Linux CentOS*, e no caso do *Wazuh Agent*, focada nos seus respetivos sistemas operativos.

O *Wazuh* disponibiliza documentação detalhada relativa ao processo de instalação para diversos sistemas operativos. Deste modo, para a criação do cenário de testes optou-se por utilizar um sistema operativo presente na documentação, nomeadamente o *CentOS 7*, de modo a facilitar o processo de instalação do sistema de SIEM. A instalação do *Wazuh* está dividida em três passos que correspondem à instalação do *Wazuh Manager*, à integração do *Elastic Stack* com o *Wazuh* e à instalação do *Wazuh Agent*.

4.1. Wazuh Manager

A instalação do *Wazuh Manager* consiste inicialmente em adicionar o repositório do *Wazuh* à lista de repositórios do *CentOS*, algo que deve ser realizado visto que muitos dos repositórios necessários para instalar os pacotes pedidos na documentação do *Wazuh* não constarem na lista de repositórios pré-configurada no SO (Sistema Operativo). De seguida, instalou-se o pacote "*wazuh-manager*". Esta instalação de pacotes é efetuada através do gestor de pacotes "*yum*". Após a instalação é ativado o serviço referente ao *Wazuh Manager*. De modo a que os *logs* que o *Wazuh Manager* analisa possam ser acedidos, foi necessário configurar o ficheiro *"/var/ossec/etc/ossec.conf"*, alterando o <*logall*> para *yes*, como se pode visualizar na Figura 4.

```
<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>yes</logall>
    <logall_json>yes</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>ossecm@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <queue_size>131072</queue_size>
  </global>
```

Figura 4 - Ficheiro de configurações *ossec.conf*

Depois do *Wazuh Manager*, é necessário instalar o *Wazuh API*. Este tem um pré-requisito, que é a instalação do *NodeJS*. O *NodeJS* é instalado através da instalação do seu pacote com o gestor "*yum*". Com isto, é realizada a instalação do *Wazuh API* também através do seu pacote a partir do "*yum*", e, de seguida, ativado o serviço do *Wazuh API* no SO. Segundo a documentação do *Wazuh*, é neste ponto que é instalado o *Filebeat*, mas como o cenário de teste para o presente trabalho trata-se de uma arquitetura de *single-host*, não é necessário realizar a instalação deste, por razões já apontadas anteriormente.

4.2. Elastic Stack

O próximo passo no processo de instalação do sistema de SIEM, consiste na instalação do *Elasticsearch*, *Logstash* e *Kibana* que pertencem ao *Elastic Stack*, sendo que, para tal, é necessário como pré-requisito a utilização do *Java*, mais concretamente *Java 8 OpenJDK*. Deste modo, foi instalado também através do gestor "*yum*" o *Java 8 OpenJDK*.

Após isto, é procedida à instalação do *Elastic Stack*, neste caso, o *Elasticsearch* em primeiro lugar. Aqui é necessário a adição do repositório *Elastic* - para que seja possivel aceder ao pacote do *Elasticsearch*, e, posteriormente, do *Logstash* e do *Kibana* - e utilizar o "*yum*" para instalar o *Elasticsearch*. Com o *Elasticsearch* instalado, é ativado o seu serviço no sistema.

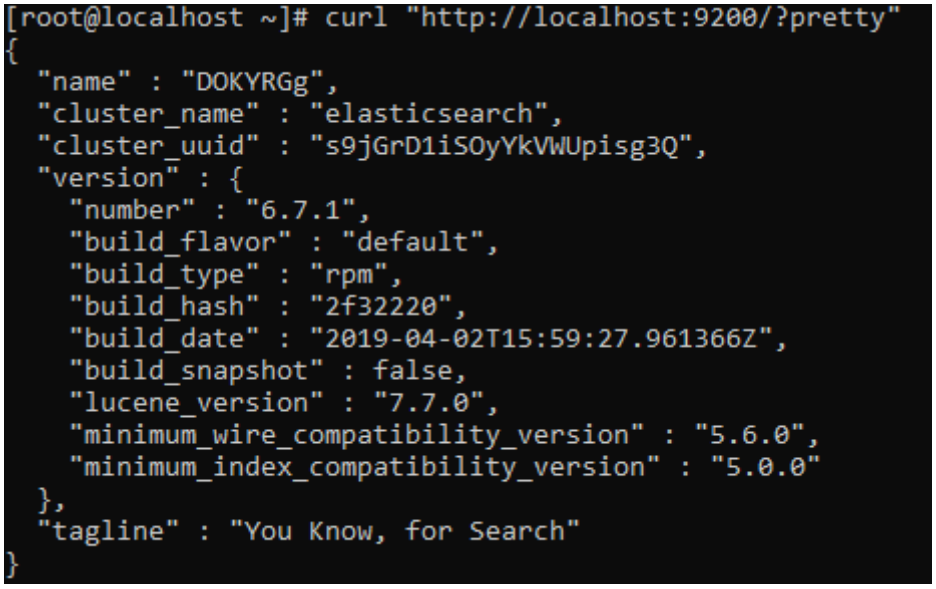


Figura 5 - Teste sobre o estado do *Elasticsearch*

Após a realização de um teste para verificar o estado do servidor *Elasticsearch* instalado (ver Figura 5), é carregado para o servidor um *template* disponibilizado pelo *Wazuh* para o *Elasticsearch*. Este *template* é necessário para o bom funcionamento do *Wazuh App* com o *Kibana*.

Com isto procede-se à instalação do *Logstash*, sendo este instalado através do pacote referente ao *Logstash* com o gestor "*yum*". Sendo o funcionamento do *Logstash* dependente do tipo de arquitetura utilizada para o cenário, é necessário transferir o ficheiro de configurações disponibilizado pelo *Wazuh.* O *download* é efetuado através da ferramenta de linha de comandos *curl.* Também para o serviço do *Logstash* conseguir aceder e ler o ficheiro de alertas "*alerts.json*", é necessário adicionar este serviço ao grupo *OSSEC*. Por fim, é ativado o serviço do *Logstash* no SO.

Como último passo na instalação do *Elastic Stack* no servidor é instalado o *Kibana* (a interface *web* do *Elastic Stack)* e o *Wazuh App*. A instalação do *Kibana* é realizada da mesma forma que o *Elasticsearch* ou o *Logstash*, através do gestor "*yum*". Já o *Wazuh App* é depois instalado através do comando "*sudo -u kibana NODE_OPTIONS="--max-old-space-size=3072" /usr/share/kibana/bin/kibana-plugin install https://packages.wazuh.com/wazuhapp/wazuhapp-3.9.0_6.7.2.zip"*.

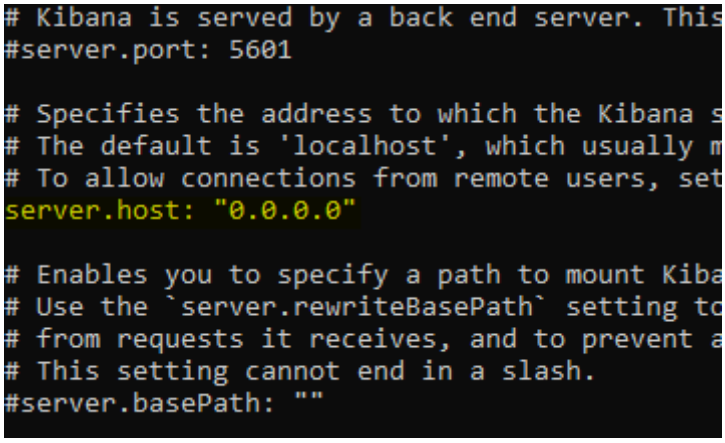


Figura 6 - Ficheiro de configurações *kibana.yml*

Como se pode visualizar na Figura 6, foi necessário realizar a alteração do ficheiro "*/etc/kibana/kibana.yml"* para configurar o valor do *server.host* para "0.0.0.0" de modo a que o *Kibana* fosse possivel ser acedido externamente, e não apenas de forma interna no *localhost*. Com isto terminado, é ativo o serviço do *Kibana* no sistema. A partir deste momento será possivel aceder à interface *web* do *Kibana*, configurar as opções do *Wazuh App* para registar o *RESTful API* do nosso servidor *Wazuh* (Figura 7) e ainda reiniciar o serviço do *Wazuh API*. Assim, será possivel efetuar a análise dos eventos dos *logs* recebidos no servidor *Wazuh* registado no *Wazuh App* no *Kibana*.

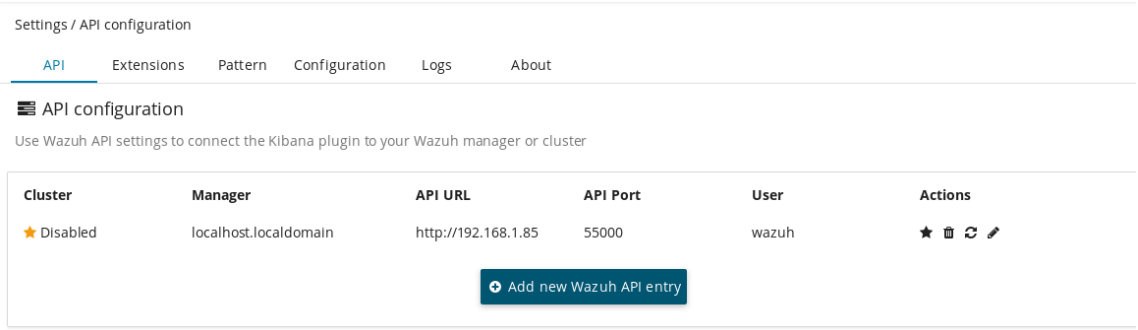


Figura 7 - Configuração de API do *Wazuh Manager*

4.3. Wazuh Agent

O modo de instalação do *Wazuh Agent* depende do SO do dispositivo ou da máquina no qual irá ser instalado. Como no cenário de testes do presente trabalho optou-se por máquinas monitorizadas com sistemas operativos diferentes, a instalação do *Wazuh Agent* diferenciou-se entre elas. Os dois SO utilizados foram o *Windows 7* e o *Ubuntu*.

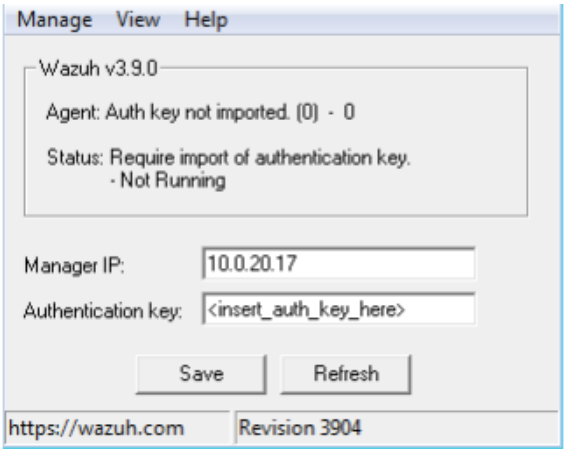


Figura 8 - Wazuh Agent Manager

A instalação do *Wazuh Agent* no *Windows* é efetuada através do ficheiro executável que é disponibilizado na página *web* do *Wazuh*, sendo que depois é procedida à instalação via GUI (*Graphic User Interface*) ou via linha de comandos. Neste cenário foi optado pelo GUI, que consiste na execução do ficheiro executável transferido para a máquina *Windows*. Como se pode ver na Figura 8Figura 1**Error! Reference source not found.**, o GUI possui dois campos, nomeadamente, o IP da máquina onde o *Wazuh Manager* está inserido e a chave de autenticação. Aqui é necessário preencher o campo do IP primeiro e selecionar a opção *Refresh*, depois para obter a chave de autenticação é preciso realizar o registo automático deste *Agent*, Para isto, foi utilizada a linha de comandos no *Windows* para executar o programa *agent-auth.exe*, configurar o ficheiro "*C:\Program Files (x86)\ossec-agent\ossec.conf*" e alterar o endereço na secção <*client*><*server*> para o do *Wazuh Manager*. Com isto, é ativado o *Agent* e a chave de autenticação é atribuída automaticamente.

Na instalação do *Wazuh Agent* no *Ubuntu*, o processo inicia-se por adicionar o repositório do *Wazuh* no *Ubuntu*, sendo depois executada a instalação do respetivo pacote através do gestor de pacotes "*yum*". Agora que o *Agent* se encontra instalado, é necessário registar e configurar o *Agent* para este poder comunicar com o *manager*. Este passo é bastante parecido com o registo do *Agent* no *Windows*, sendo que aqui é executado também o programa *agent-auth*, configurado o ficheiro "/var/ossec/etc/ossec.conf" e alterado o endereço na secção <*client*><*server*> para o endereço IP do *Wazuh Manager*. Depois basta iniciar o serviço do *Wazuh Agent*.

Após a instalação do *Wazuh Agent* no *Ubuntu*, foi necessário efetuar configurações adicionais. O *Wazuh Agent* vem inicialmente com configurações padrão para a recolha de alguma informação dos *syslogs* presentes nas seguintes localizações:

- /var/ossec/logs/active-responses.log
- /var/log/auth.log
- /var/log/syslog
- /var/log/dpkg.log
- /var/log/kern.log

No cenário de testes pensou-se também em ter a máquina virtual *Ubuntu* a funcionar como um *webserver*. Para isto, instalou-se o *Apache*, o PHP (*Hypertext Preprocessor*) e o MariaDB. No entanto, o *Wazuh* não recolhia automaticamente os *logs* destes serviços, pelo que se tiveram de adicionar os seguintes ficheiros/localizações:

- /var/log/apache2/*.log
- /var/log/mysql/*.log

5. Cenário de Testes

Antes de mais foi necessário definir o cenário de testes no qual a segurança dos sistemas irá ser testada. Deste modo, o cenário elaborado para este trabalho é composto por 5 máquinas virtuais em 3 máquinas físicas, que, por sua vez, desempenharão papéis distintos. Deste modo, uma destas máquinas físicas - um computador *desktop* com o sistema operativo *Arch Linux* - irá suportar 3 das máquinas virtuais sendo uma delas o servidor *Wazuh* e as restantes os *Wazuh agentes*. As restantes máquinas virtuais encontram-se em cada uma das duas máquinas físicas (Figura 9).

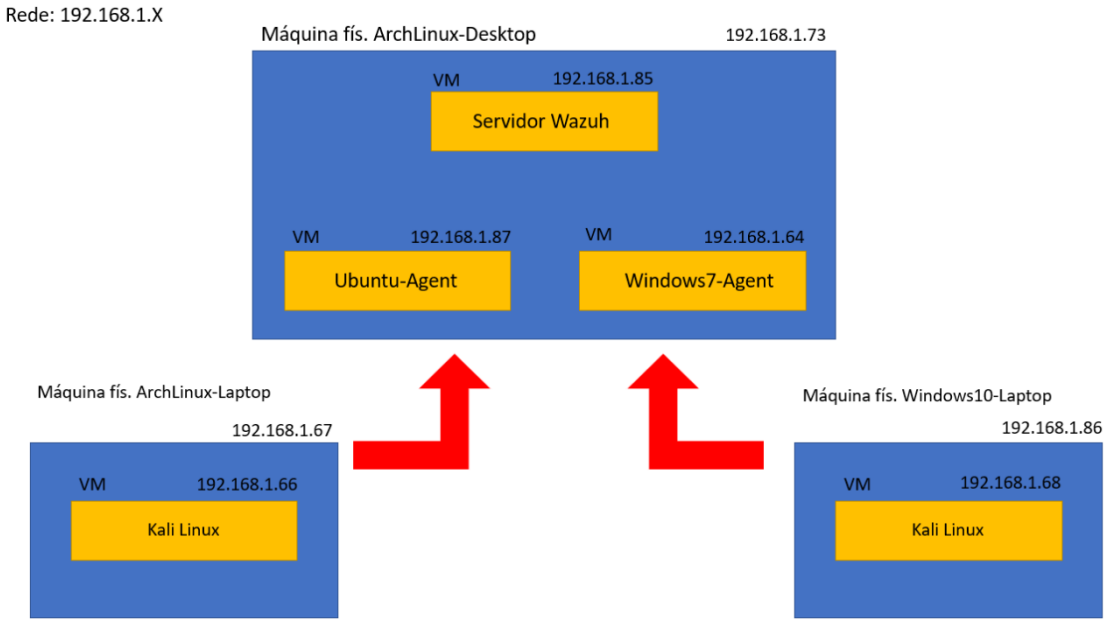


Figura 9 - Cenário de testes

As duas máquinas físicas portáteis do cenário, a correr, cada uma, uma máquina virtual *Kali Linux*, serão as máquinas virtuais responsáveis por realizar os ataques às outras duas máquinas virtuais, sendo estas, por sua vez, os *Wazuh agents* do cenário. No capítulo 3, já foi referida a arquitetura utilizada, tendo já sido mencionado os elementos de cada componente deste cenário. No entanto, para recapitular de uma forma breve, o *Wazuh server*, possuí alguns elementos como o *Wazuh Manager*, *Wazuh API* e o componente *Elastic Stack*; portanto, esta é a máquina virtual que receberá os registos *log* das máquinas que estão a ser monitorizadas e ainda a máquina na qual se fará a análise/observação dos dados recolhidos, através do *Kibana*, com a aplicação das regras definidas. Por sua vez,

as duas máquinas monitorizadas - uma com o sistema operativo *Windows 7* e outra com o *Ubuntu 19.04* - contêm o elemento *Wazuh agent*; deste modo, estas são aquelas que enviam os registos *log* para o *Wazuh Server*.

Tal como já foi mencionado anteriormente, no sistema *Ubuntu* foi também configurado um *webserver*, com o *Apache*, *PHP* e MariaDB, onde foi criada uma página simples em PHP vulnerável a ataques do tipo *SQL Injection*, de modo a expandir o tipo de ataques que podem ser feitos.

Os ataques que serão realizados às duas máquinas monitorizadas serão efetuados em períodos temporais definidos e variáveis consoante o ataque em causa. Assim, o período temporal de ocorrência de ataques durará 48 horas, sendo que estas horas correspondem aos dias 7 e 8 de maio de 2019.

6. Simulação do cenário de teste

Com a implementação do cenário de teste efetuada, foi dado início à fase de simulação do cenário definido. Como já foi referido anteriormente, a simulação do cenário irá apenas ocorrer durante os dias de 7 a 8 de maio. Deste modo, foram executados alguns ataques às duas máquinas a partir destas datas, nomeadamente, os ataques:

- **Ataque de força bruta no *login* por *SSH* (*Secure Shell*) tirando recurso da ferramenta *hydra***. Este ataque consistiu na utilização de duas listas de palavras, as quais iram ser utilizadas para efetuar as tentativas de *login* na máquina alvo do cenário (máquina *Ubuntu*). O ataque inicia-se por executar o comando da ferramenta *hydra*, "*hydra -L userlist.txt -P passwordlist.txt ssh://<ip da máquina alvo>*", na linha de comandos do *Kali*. Este comando requer a utilização de duas listas de palavras que foram criadas previamente, contendo nomes de utilizadores comuns numa lista e *passwords* mais comuns na outra lista. Este ataque em concreto, foi executado apenas uma vez pelas duas máquinas de ataque em alturas de tempo separadas, nomeadamente, dia 7 e 8 de maio, respetivamente. De notar, que também foram efetuadas tentativas de *login* manuais, de forma a simular a utilização normal do sistema.
- **Ataque de SQL *Injection* tirando recurso da ferramenta *SQLMap***. *SQL Injection* é um ataque que consiste na inserção de código *SQL* em *inputs*, de forma a manipular o resultado de um pedido à base de dados. Este ataque em concreto, consistiu na execução de um comando na ferramenta *SQLMap* na linha de comandos, "*sqlmap -u http://192.168.1.87/verify.php --data="username=admin" --level=1 --risk=1 --dump-all*". Este comando irá executar vários *scripts* de ataques de baixo nível e de impacto, na página do endereço IP alvo com apenas pedidos *POST*. Este ataque foi executado duas vezes no dia 7 de maio ao *webserver* presente na máquina *Ubuntu*.
- **Ataque de *port scanning* tirando recurso da ferramenta *Nmap***. O ataque de *port scanning* consiste na execução de pedidos a portos existente numa máquina alvo, de forma a encontrar portos abertos e os seus serviços. Este ataque foi realizado através da execução do comando "*nmap -sV 192.168.1.87*" da ferramenta *Nmap* na linha de comandos. O que este comando irá fazer é precisamente o que um ataque de *port scanning* faz, que é a procura de portos

abertos e a determinação de informação de serviços nestes. Este ataque foi executado uma vez no dia 8 de maio.

- **Ataque com a *framework Metasploit***. Utilizando a *framework Metasploit* presente no *Kali*, foi criado um executável malicioso que, por sua vez, foi disponibilizado num servidor *web* para que este pudesse ser transferido para a máquina alvo, através de um ataque *man-in-the-middle*, no qual se fosse pedido um executável do *Chrome*, o ficheiro malicioso seria transferido. Este ficheiro fornecia acesso a uma sessão na máquina alvo – máquina esta a *Windows 7* - através de uma *shell meterpreter*.

Com os ataques concluídos, a fase de monitorização e análise foi iniciada através da utilização do *Wazuh App* do SIEM *Wazuh* no *Kibana*.

6.1. Monitorização e análise de eventos no *Kibana*

Para aceder às funcionalidades de monitorização e análise que o *Wazuh* proporciona, acede-se ao acesso endereço local "*http://localhost:5601*" num *browser*, acabando-se por aceder ao *Kibana* onde se encontra o *Wazuh App*.

Como já foi referido anteriormente, o *Wazuh* proporciona uma interface gráfica onde poderão ser realizadas monitorizações e análises de eventos. Esta tira partido do *Kibana* com a integração do *plugin* do *Wazuh App*. O *Wazuh App* oferece uma forma rápida para visualizar *clusters* de informação, agentes e alertas; oferece uma interface de utilizador de fácil uso para interagir com a API e com o *Wazuh Manager* ao apresentar informação importante num formato conveniente.

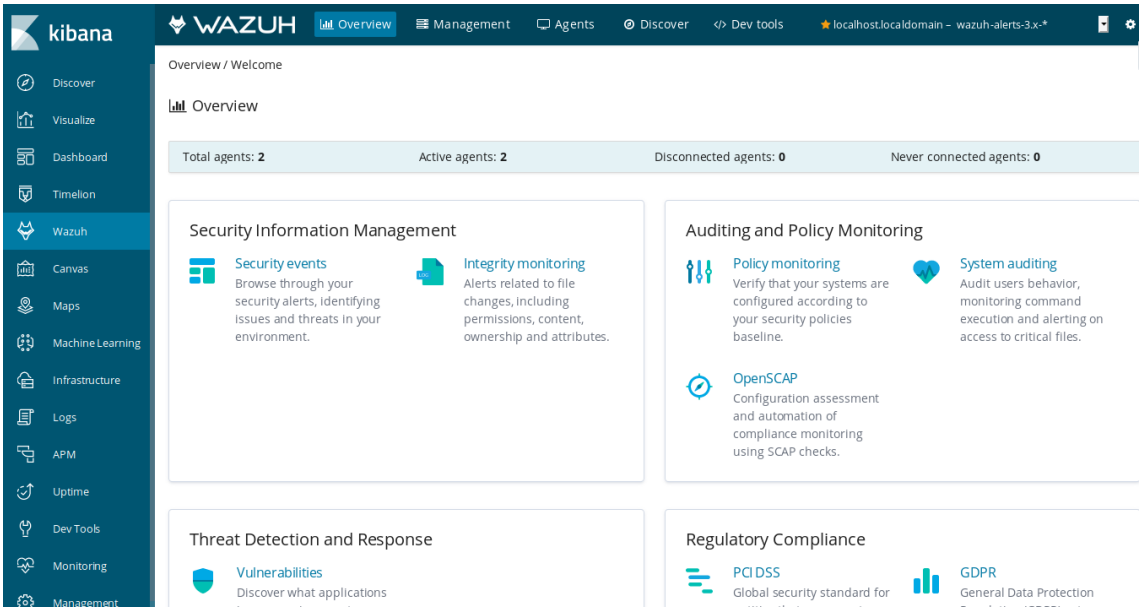


Figura 10 - Interface do *Wazuh App*

Como se pode visualizar na Figura 10, o *Wazuh App* é acessível no *Kibana* através da barra de navegação lateral, e este está organizado em 6 secções principais:

- ***Overview***: secção principal do *Wazuh App*; nela encontram-se várias opções e informações sobre os agentes, monotorização de eventos e do sistema interno dos agentes. Mas, são nomeadamente as opções de *Security Events* e de *Integrity Monitoring* que se destacam. É também nesta secção que é encontrada a subsecção relativa às vulnerabilidades encontradas nas aplicações dos agentes e a subsecção de completude regulatória com leis e regulações.
- ***Management***: secção que guarda informação sobre a administração, estados sobre os vários componentes do *Wazuh*. Nesta secção são essencialmente visualizadas as *rulesets* criadas, os grupos de *rulesets*, os *logs*, os relatórios gerados anteriormente (por exemplo de pesquisas), entre outros.
- ***Agents***: secção que lista todos os agentes e apresenta informações gerais para cada um. Também oferece uma barra de procura mais detalhada para a lista de agentes. Caso seja necessário encontrar mais informação sobre estes agentes, basta escolher um deles na lista e será apresentada, noutra página, a informação no mesmo formato que a secção do *Overview*.
- ***Discover***: secção que permite explorar de forma interativa todos os alertas despoletados pelo *Wazuh*. Aqui estão disponíveis todos os alertas a partir de

um filtro e escala temporal definidos, como, por exemplo, todos o alertas deste ano ou todos os alertas nos últimos 15 minutos.

- ***Dev tools***: secção que oferece ao utilizador uma interface para interagir com a API do *Wazuh*. Aqui podem ser realizados pedidos e receber as respetivas respostas.
- ***Settings***: secção que permite configurar e personalizar o próprio *Wazuh App*.

Neste cenário, a monitorização e análise foi efetuada através das funcionalidades de monitorização de eventos e de informação de sistema nas secções do *Overview* e do *Agent* (página *Overview* de cada agente).

Ao usar o *Security Events* no *Overview*, é possivel analisar todos os eventos. Aqui foi necessário selecionar o intervalo de tempo no qual se queria visualizar, que neste caso é o intervalo de tempo entre os dias 7 e 8 de maio. Como se pode ver na Figura 11, com o *Overview* de eventos de segurança foi possivel perceber o número total de alertas recebidos por todos os agentes, ver quantos alertas de nível elevado ocorreram e conhecer a quantidade de eventos de autenticação. Também com os dois gráficos de "*Alert level evolution"* e "*Alerts"*, foi possível verificar as alturas onde ocorreram a maior parte dos ataques, sendo essas as zonas onde se destacam uma maior quantidade de alertas acionados.

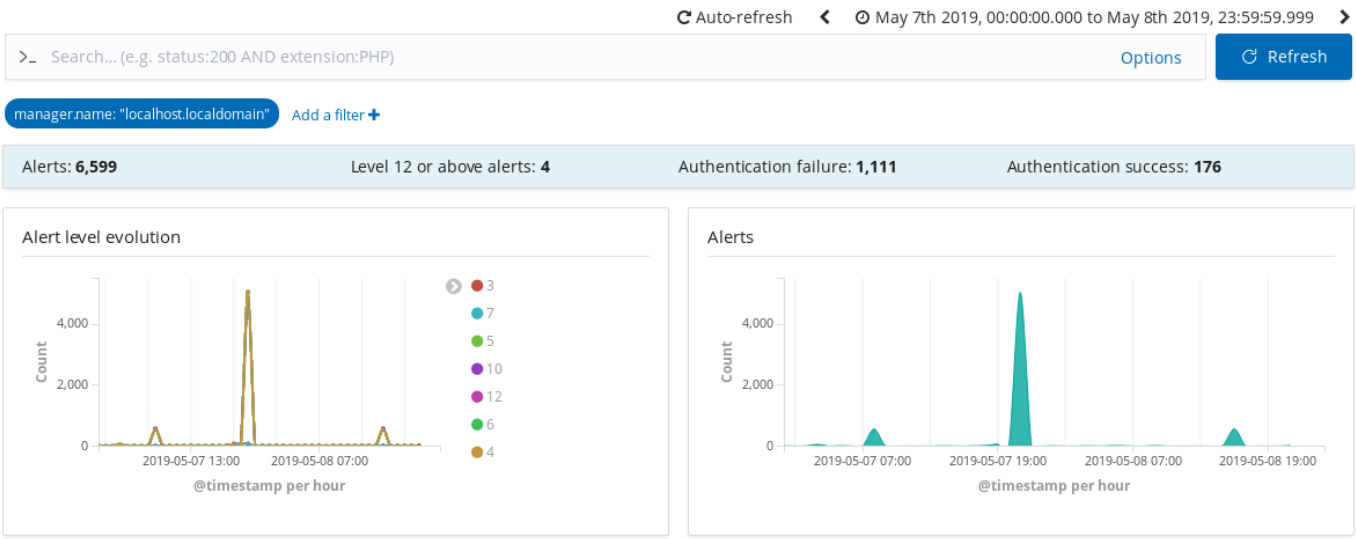


Figura 11 - Vista geral dos eventos de segurança no *Overview*

Depois, na Figura 12, continuando na mesma página do *Overview*, com os seguintes gráficos, foi possível analisar quais os agentes enviaram mais eventos, os grupos de regras

de alertas (ou tipo de ataques) mais despoletados e o estado dos agentes ao longo do período de tempo selecionado anteriormente.

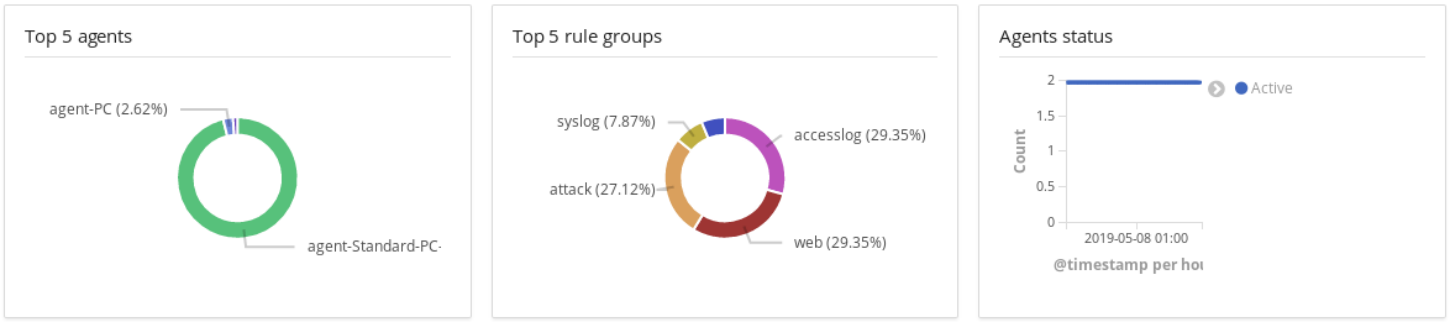


Figura 12 - Continuação do conteúdo na página referente à figura anterior

Na Figura 13, que continua a fazer parte da mesma página do *Overview*, é também possível perceber melhor, através da tabela de alertas, quais o alertas de maior nível que ocorreram, qual a regra pertencente a esse nível e o número de vezes que aconteceu o alerta.

Alerts summary

Rule ID	Description	Level	Count
5108	System running out of memory. Availability of the system is in risk.	12	2
40112	Multiple authentication failures followed by a success.	12	2
5712	sshd: brute force trying to get access to the system.	10	3
5720	sshd: Multiple authentication failures.	10	4
5551	PAM: Multiple failed logins in a small period of time.	10	36
31151	Multiple web server 400 error codes from same source ip.	10	352
2902	New dpkg (Debian Package) installed.	7	1
2904	Dpkg (Debian Package) half configured.	7	2
533	Listened ports status (netstat) changed (new port opened or closed).	7	7
550	Integrity checksum changed.	7	73

Figura 13 - Sumário de alertas do *Overview* de eventos de segurança

É também nesta tabela que é possivel detetar os ataques que se elaborou no cenário de testes; com a exceção do ataque do *port scanning* com o *Nmap* e o acesso ao sistema

através da *shell metepreter*. É possível detetar o ataque de força bruta nas regras 5712 e 5720, e o ataque de SQL *injection* na regra 31151.

Para efetuar uma análise mais especifica a um agente optou-se por escolher a secção de *Agents* e escolher o agente que mais ataques efetuamos, nomeadamente, o o agente *Ubuntu* (IP 192.168.1.87). Ao escolher a opção de *Security Events*, tal como no *Overview*, ir-se-á encontrar uma análise igual, mas mais orientada ao agente em questão. Como se pode confirmar na Figura 14, este agente recebeu claramente ataques de SQL *Injection* e ataques de força bruta, como se pode ver nos gráficos "*Top 5 alerts*" e "*Top 5 groups*". É também de realçar que o *Wazuh* permite conciliar leis e regulamentos com os alertas, o que se pode verificar com o gráfico "*Top 5 PCI DSS Requirements*" que mostra os requerimentos mais encontrados nos alertas.

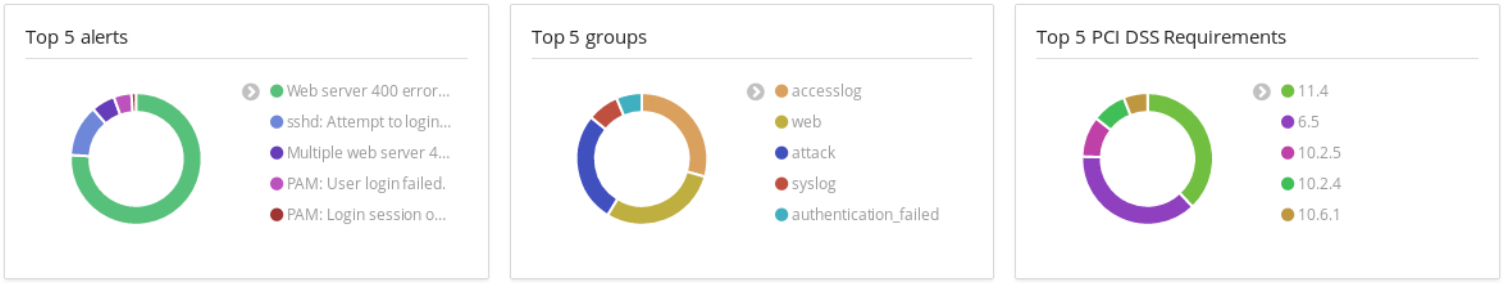


Figura 14 - 1ª parte do *Overview* do *agent-Ubuntu*

Tal como no *Overview* dos eventos de segurança, também é possível visualizar a evolução dos níveis de alertas e o progresso temporal dos alertas recebidos, mais especificamente para o agente *Ubuntu* (respetivamente Figura 15 e Figura 16)

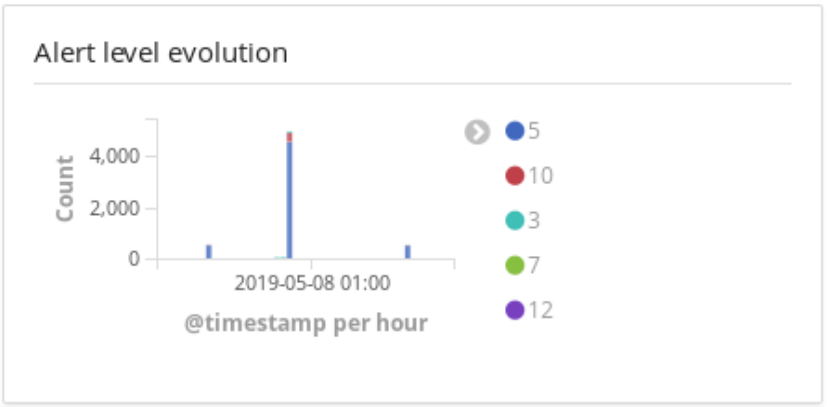


Figura 15 - 2ª parte do *Overview* do *agent-Ubuntu*

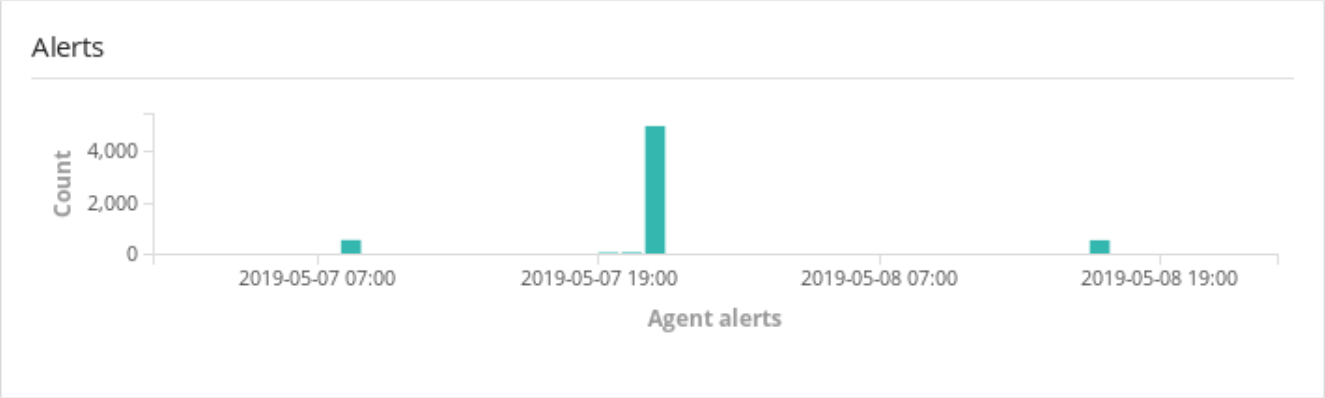


Figura 16 - 3ª parte do *Overview* do *agent-Ubuntu*

Por fim, na Figura 17 tem-se em maior detalhe informação sobre os alertas ocorridos neste agente. Comprova-se que o ataque de SQL *Injection* foi aquele que mais alertas acionou no *Wazuh*.

Alerts summary

Rule ID	Description	Level	Count
31101	Web server 400 error code.	5	4,589
5710	sshd: Attempt to login using a non-existent user	5	768
31151	Multiple web server 400 error codes from same source ip.	10	352
5503	PAM: User login failed.	5	264
5501	PAM: Login session opened.	3	70
5502	PAM: Login session closed.	3	66
5715	sshd: authentication success.	3	46
5551	PAM: Multiple failed logins in a small period of time.	10	36
5716	sshd: authentication failed.	5	36
516	System Audit event.	3	28

Figura 17 - 4ª parte do *Overview* do *agent-Ubuntu*

É de concluir que com esta análise foi possivel encontrar toda a informação necessária para perceber que ataques foram efetuados, as alturas em que ocorreram e a sua frequência de ocorrência. Mas, como o ataque de *port scanning* não foi capturado pelas regras existentes no *Wazuh*, sendo por isso necessário criar uma nova regra (ou alerta) para que este seja capturado pelo *Wazuh*. De notar que, o ataque que teve como alvo a máquina *Windows* não foi identificado pelo *Wazuh*, não tendo sido detetado qualquer registo nos registos de *logs* enviados para o servidor.

6.2. Definição e configuração de alertas de segurança

Para a criação e configuração de um novo alerta de segurança no *Wazuh*, é necessário o adicionamento de um novo *Ruleset* ao *Wazuh*. Este processo passo pela configuração de um novo *decoder* no ficheiro "*/var/ossec/etc/decoders/local_decoder.xml*" que vai ler e encontrar nos *logs* uma entrada especifica, e depois configurar uma nova entrada de código no ficheiro "*/var/ossec/etc/rules/local_rules.xml*" com a regra que vai ser acionada pelo *decoder* especificado anteriormente.

De acordo com a documentação foi criada um *decoder* e uma regra de modo a acionar um alerta mais específico para *SQL injection*. As linhas de log para as quais foi desenvolvido este código foi a seguinte:

[Fri May 10 16:00:54.740629 2019] [php7:notice] [pid 1968] [client 192.168.1.66:34684] PHP Notice: Undefined index: password in /var/www/html/verify.php on line 11

O código do *decoder* desenvolvido foi o seguinte:

```
<decoder name="php-notice">
    <parent>apache-errorlog</parent>
    <prematch offset="after_parent">^PHP Notice:  Undefined index: \w+ in /var/www/html/\w+</prematch>
    <regex offset="after_parent">^PHP Notice:  Undefined index: (\w+) in /var/www/html/(\w+)</regex>
    <order>field,page</order>
</decoder>
```

Enquanto que a regra criada foi:

```
<rule id="100010" level="5">
    <decoded_as>php-notice</decoded_as>
    <description>Possible SQL Injection Attempt</description>
</rule>
```

No entanto, apesar de ter sido criado de acordo com a documentação e testado com a ferramenta *ossec-logtest,* esta regra não era acionada, nem era mostrado nenhum ecrã de erro pelo que após várias tentativas de criar este alerta sem sucesso, decidiu-se parar.

7. Criação e Análise do *Data Set*

Neste trabalho, para além da análise dos dados recolhidos pelo *Wazuh* ser realizada com o *Kibana*, esta também foi concretizada de um outro modo, que consiste na análise de *data sets* criados a partir de ficheiros *log* que, por sua vez, contêm os eventos recolhidos. A ferramenta *Rapidminer* permite não só a criação do *data set*, mas também a sua análise; já outras ferramentas que permitem a sua análise são o *Microsoft Excel* e a ferramenta *Power BI*. É de referir que no presente trabalho, esta técnica de análise dos eventos ocorridos foi unicamente aplicada para se obter um *data set* relativo aos ataques *SSH* efetuados à máquina *Ubuntu 19.04*, devido ao formato de cada tipo de ataque efetuado ser diferente, como se irá perceber de seguida.

Assim, em primeiro lugar, para se criar o *data set* pretendido, foi realizada a extração do ficheiro *log* (formato *.txt*) do *Wazuh* na pasta */var/ossec/logs/archive/archives.log*. Após a sua extração, com uma rápida observação do conteúdo do ficheiro, verificou-se que os dados deste não apresentam uma formatação definida, apresentando vários eventos de várias fontes de dados. A Figura 18 é um exemplo de parte do conteúdo de um ficheiro *log*.

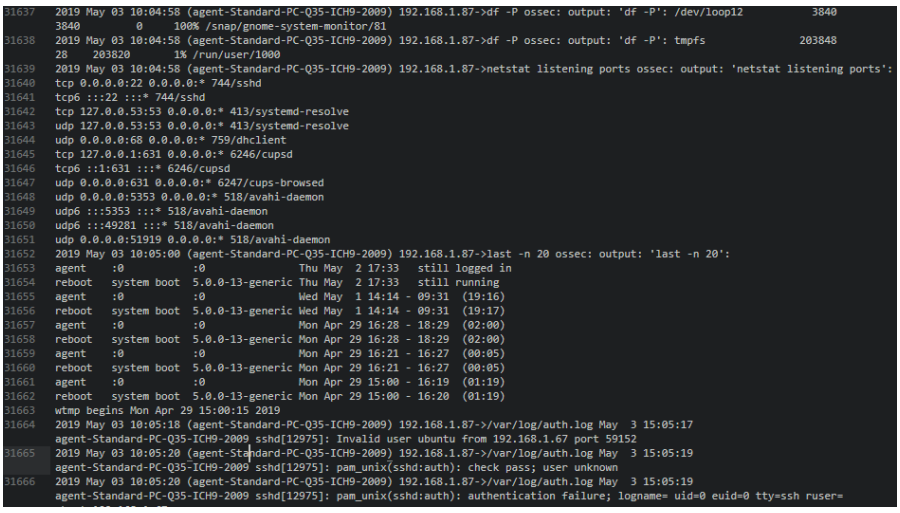


Figura 18 - Exemplo do conteúdo do ficheiro *log*

Para o caso do serviço *SSH*, reparou-se que existiam várias linhas referentes a eventos relacionados com este, no entanto, muitas destas linhas tinham uma estrutura divergente entre elas, dependendo da informação que apresentam. Assim, através de uma análise visual do ficheiro *log*, tentou-se perceber qual a estrutura das linhas referentes aos registos de eventos relativos a ataques ao *SSH* e que apresentassem informação mais pertinente

para o *data set*. Assim, decidiu-se retirar as linhas especificas a um ataque, por exemplo, a utilização de *username* ou *password* incorreta, bem como as linhas referentes a uma aceitação de uma *password* para um utilizador.

Deste modo, aquando a criação do *data set* no *Rapidminer*, foi necessário recorrer a um dado conjunto de operadores para se manipular os dados e filtrar a informação que se pretendia. Para começar, foi necessário juntar os ficheiros *log* referentes a vários dias e começar a filtrar consoante a informação que apresentavam no início. Por exemplo, as linhas pretendidas começavam sempre pelo ano, logo, foi filtrado todo o conjunto de dados de modo a se obter as linhas que iniciassem pelo ano. Após isto, foram utilizados vários operadores de forma a criar e filtrar atributos das linhas que se pretendiam, dependendo da formatação/estrutura da própria linha.

Por fim, foi obtido um *data set* com os seguintes atributos:

- DateAgent: data do registo no agente
- MonthAgent: mês da data do registo no agente
- DayAgent: dia da data do registo no agente
- TimeAgent: hora/minuto do registo no agente
- HourAgent: hora do registo no agente
- MinuteAgent: minuto do registo no agente
- Agent: nome do agente do *Wazuh*
- IdProcess: id do processo de autenticação
- Message: mensagem presente na linha
- User: *username* dado na tentativa de autenticação
- IpAlvo: endereço IP do agente do *Wazuh*
- IpAtaque: endereço IP de onde provêm a tentativa de *login*
- State: estado da linha
    - 0 – sucesso
    - 1 – *username* inválido
    - 2 – *password* errada
    - 3 – *password* errada e *username* inválido
    - -1 – falha na obtenção do estado

Com a análise do ficheiro *log* foi possível também perceber que existe uma situação em que existem dois registos de "erro" relativos ao mesmo processo de autenticação, ou seja, à mesma tentativa de autenticação. Esta situação deve-se ao facto de a tentativa de autenticação em causa ocorrer com a utilização de um *username* inválido e, possivelmente, por conseguinte, apresentar uma *password* errada. Assim, existe uma linha referente ao *username* ser inválido e outra referente ao facto de existir uma *password* errada para um *username* inválido. A Figura 19 permite exemplificar esta ocorrência (linha nº 31669 e a linha nº 31672) em que é possível chegar à conclusão que esta informação pertence ao mesmo pedido de autenticação, pois apresenta o mesmo id, nomeadamente, o id 12983. Já as outras duas linhas que estão na figura abaixo referem-se a linhas, que apesar de estarem relacionadas com o processo de autenticação SSH, foram desprezadas para a criação do *data set* visto não terem a mesma formatação/estrutura das linhas já mencionadas anteriormente e, para além disso, não apresentam mais informação relevante para o contexto.

```
31669   2019 May 03 10:05:32 (agent-Standard-PC-Q35-ICH9-2009) 192.168.1.87->/var/log/auth.log May  3 15:05:32
        agent-Standard-PC-Q35-ICH9-2009 sshd[12983]: Invalid user wazuh from 192.168.1.67 port 59154
31670   2019 May 03 10:05:34 (agent-Standard-PC-Q35-ICH9-2009) 192.168.1.87->/var/log/auth.log May  3 15:05:34
        agent-Standard-PC-Q35-ICH9-2009 sshd[12983]: pam_unix(sshd:auth): check pass; user unknown
31671   2019 May 03 10:05:34 (agent-Standard-PC-Q35-ICH9-2009) 192.168.1.87->/var/log/auth.log May  3 15:05:34
        agent-Standard-PC-Q35-ICH9-2009 sshd[12983]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
        rhost=192.168.1.67
31672   2019 May 03 10:05:36 (agent-Standard-PC-Q35-ICH9-2009) 192.168.1.87->/var/log/auth.log May  3 15:05:36
        agent-Standard-PC-Q35-ICH9-2009 sshd[12983]: Failed password for invalid user wazuh from 192.168.1.67 port 59154 ssh2
```

Figura 19 - Exemplo de registos associados ao processo de autenticação

A Figura 20 representa o processo de criação do *data set* no *Rapidminer* já a partir da informação em formato bruto dos ficheiros *log* utilizados.

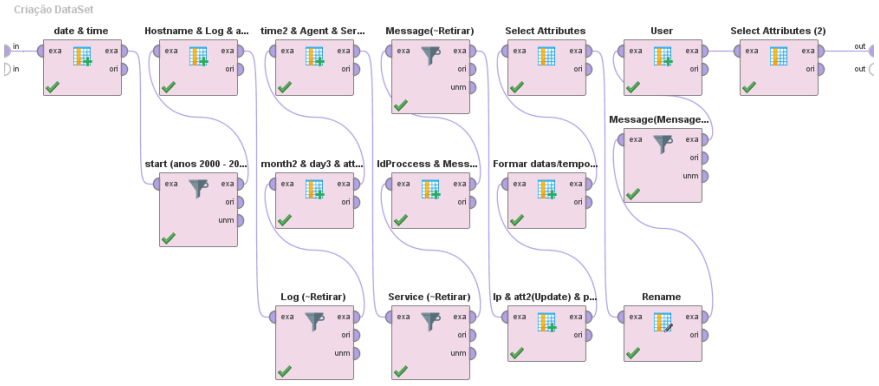


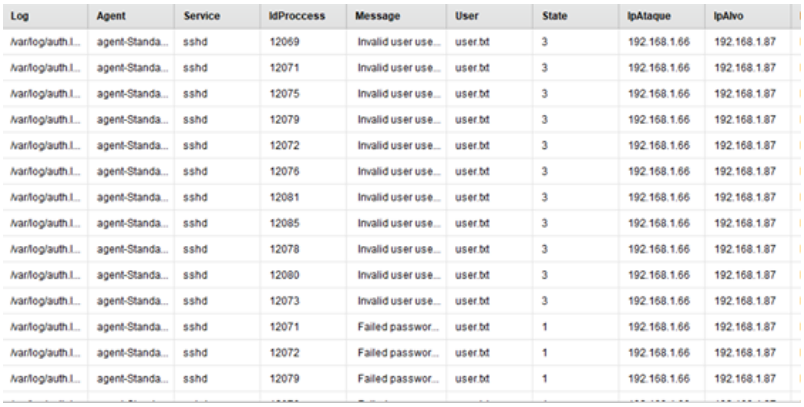
Figura 20 - Processo do *Rapidminer* para criação do *data set*

Como se pode constatar na figura cima, foi necessário utilizar-se vários operadores de forma a que, de um atributo que correspondia a toda a linha do ficheiro *log*, obter-se os 15 atributos já mencionados anteriormente através da filtragem/seleção das linhas

pretendidas e ainda da geração de atributos. A Figura 21 apresenta o conjunto de dados obtido com a junção dos ficheiros *log*, sem qualquer tipo de transformação. Já uma pequena parte do *data set* obtido encontra-se na Figura 22.

att1
2019 May 07 02:32:48 (agent-Standard-PC-Q35-ICH9-2009) 192.168.1.87->last -n 20 ossec: output: 'last -n 20':
agent :0 :0 Thu May 2 17:33 still logged in
reboot system boot 5.0.0-13-generic Thu May 2 17:33 still running
agent :0 :0 Wed May 1 14:14 - 09:31 (19:16)
reboot system boot 5.0.0-13-generic Wed May 1 14:14 - 09:31 (19:17)
agent :0 :0 Mon Apr 29 16:28 - 18:29 (02:00)
reboot system boot 5.0.0-13-generic Mon Apr 29 16:28 - 18:29 (02:00)
agent :0 :0 Mon Apr 29 16:21 - 16:27 (00:05)
reboot system boot 5.0.0-13-generic Mon Apr 29 16:21 - 16:27 (00:05)
agent :0 :0 Mon Apr 29 15:00 - 16:19 (01:19)
reboot system boot 5.0.0-13-generic Mon Apr 29 15:00 - 16:20 (01:19)
wtmp begins Mon Apr 29 15:00:15 2019
2019 May 07 02:35:36 (agent-Standard-PC-Q35-ICH9-2009) 192.168.1.87->/var/log/syslog May 7 07:35:35 agent-Standard-PC-Q35-I...

Figura 21 - Resultado da junção dos ficheiros *log*



Log	Agent	Service	IdProccess	Message	User	State	IpAtaque	IpAlvo
/var/log/auth.l...	agent-Standa...	sshd	12069	Invalid user use...	user.txt	3	192.168.1.66	192.168.1.87
/var/log/auth.l...	agent-Standa...	sshd	12071	Invalid user use...	user.txt	3	192.168.1.66	192.168.1.87
/var/log/auth.l...	agent-Standa...	sshd	12075	Invalid user use...	user.txt	3	192.168.1.66	192.168.1.87
/var/log/auth.l...	agent-Standa...	sshd	12079	Invalid user use...	user.txt	3	192.168.1.66	192.168.1.87
/var/log/auth.l...	agent-Standa...	sshd	12072	Invalid user use...	user.txt	3	192.168.1.66	192.168.1.87
/var/log/auth.l...	agent-Standa...	sshd	12076	Invalid user use...	user.txt	3	192.168.1.66	192.168.1.87
/var/log/auth.l...	agent-Standa...	sshd	12081	Invalid user use...	user.txt	3	192.168.1.66	192.168.1.87
/var/log/auth.l...	agent-Standa...	sshd	12085	Invalid user use...	user.txt	3	192.168.1.66	192.168.1.87
/var/log/auth.l...	agent-Standa...	sshd	12078	Invalid user use...	user.txt	3	192.168.1.66	192.168.1.87
/var/log/auth.l...	agent-Standa...	sshd	12080	Invalid user use...	user.txt	3	192.168.1.66	192.168.1.87
/var/log/auth.l...	agent-Standa...	sshd	12073	Invalid user use...	user.txt	3	192.168.1.66	192.168.1.87
/var/log/auth.l...	agent-Standa...	sshd	12071	Failed passwor...	user.txt	1	192.168.1.66	192.168.1.87
/var/log/auth.l...	agent-Standa...	sshd	12072	Failed passwor...	user.txt	1	192.168.1.66	192.168.1.87
/var/log/auth.l...	agent-Standa...	sshd	12079	Failed passwor...	user.txt	1	192.168.1.66	192.168.1.87

Figura 22 - Pequena parte do data set obtido

7.1. Análise do *Data Set*

Com o *data set* criado, foi iniciada a fase de análise do mesmo de modo a se retirar algumas conclusões da informação presente. Neste trabalho, a análise do *data set* criado foi efetuada recorrendo-se a três ferramentas distintas, mais precisamente, o *Rapidminer*, o *Microsoft Excel* e ainda a ferramenta *Power BI*.

Com o *Rapidminer* foi feita alguma análise, mais precisamente, algumas contagens gerais de possíveis eventos. Por exemplo, foi chegada à conclusão que houve ao todo 3 IPs que efetuaram tentativas de *login* falhadas, mais precisamente o IP 192.168.1.66 com 151 tentativas, o IP 192.168.1.67 com 47 tentativas e o IP 192.168.1.68 com 148 tentativas, perfazendo assim um total de 300 tentativas para o período de dois dias (Figura 23). Por si só, esta situação já é muito estranha e alarmante.

IpAtaque	count(IdPro...
192.168.1.66	151
192.168.1.67	1
192.168.1.68	148

Figura 23 - Tentativas de *login* falhadas por IP da máquina atacante

Assim, de forma a entender melhor o caso, foi efetuada uma pesquisa para se perceber se as tentativas de *login* falhadas eram feitas por utilizadores válidos ou não. Com o *Rapidminer*, chegou-se à conclusão de que foram feitas 260 tentativas de *login* que falharam, com utilizadores inválidos e 40 com utilizadores válidos (Figura 24). De notar, que todas estas tentativas foram feitas para o mesmo *agent*, sendo este o computador com o sistema operativo *Ubuntu*.

Agent	State	count(IdProc...
agent-Standa...	2	40
agent-Standa...	3	260

Figura 24 - Tentativas de *login* falhadas por utilizadores válidos e por utilizadores inválidos

Para verificar se ocorreu também alguma situação atípica nos *logins* válidos, foi feita uma pesquisa, sendo que o resultado foi o de 48 *logins* com sucesso feitos pelo utilizador "agent", sendo que a máquina de IP 192.168.1.67 fez 46 e os outros 2 IPs já referidos fizeram um *login* com sucesso cada um (Figura 25).

Agent	IpAtaque	User	count(IdPro...
agent-Standa...	192.168.1.66	agent	1
agent-Standa...	192.168.1.67	agent	46
agent-Standa...	192.168.1.68	agent	1

Figura 25 - Tentativas de *login* efetuadas com sucesso

Utilizando o *Power BI* foi feita uma pesquisa para observar a quantidade de *logins* falhados por nome de utilizador. O resultado é visível pela Figura 26, podendo observar-se que os valores das tentativas variam entre 23 e 10. Isto deixa um pouco em dúvida, se foi efetuado um ataque de força bruta automatizado ou não, ou seja, com o uso de uma ferramenta ou não. No entanto, o dicionário de palavras que terá sido utilizado para o

ataque terá sido sem dúvida, pequeno. Ficar-se-á com uma melhor perceção, se for feita uma análise com base no tempo. Mas antes, foi elaborada uma pesquisa para determinar os *top* 4 nomes de utilizador mais utilizados nas tentativas de *login* (Figura 27). Pode-se observar através dessa figura que os nomes mais utilizados foram "root", "username", "login" e "guest"

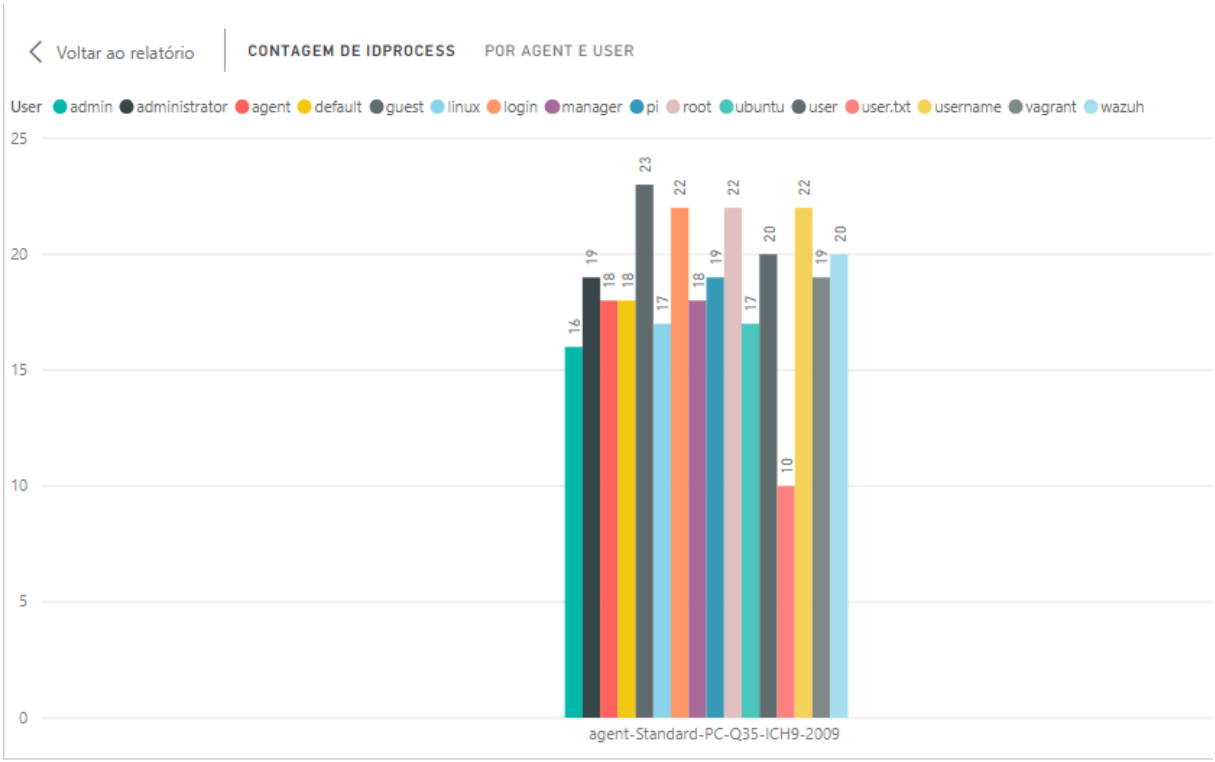


Figura 26 - Quantidade de tentativas de *login* falhados por nome de utilizado

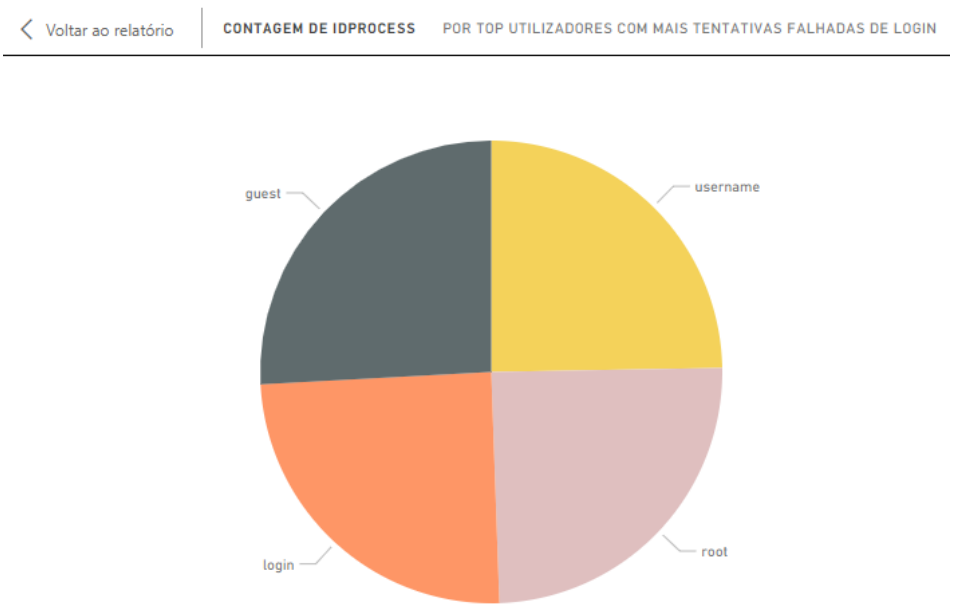


Figura 27 - *Top* 4 de nomes de utilizador mais utilizados nas tentativas de *login*

Relativamente à análise do *data set* através do *Microsoft Excel*, esta foi realizada através de tabelas dinâmicas, sendo que as análises realizadas neste tiveram em consideração o fator temporal. Primeiramente foi efetuada uma análise com o objetivo de se perceber a distribuição das tentativas de *login* ao longo do tempo (data e hora), obtendo-se o gráfico presente na Figura 28. Através da análise deste percebe-se que as tentativas de *login* ocorreram nos dias 7 e 8 de maio em horas distintas. É de referir que se verificou que o dia em que ocorreu mais tentativas foi o dia 8 de maio, pelas 16:16H. Relativamente ao dia 7 de maio os ataques ocorreram com pico de ocorrência pelas 08H.

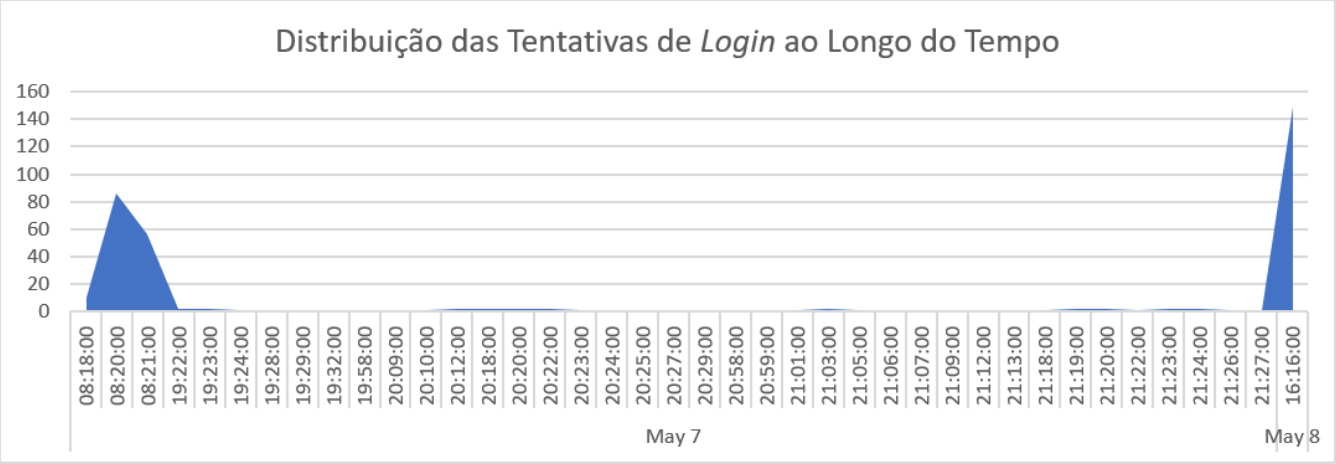


Figura 28 - Distribuição das tentativas de *login* ao longo do tempo

De seguida, de modo a se perceber que tentativas de *login* foram realizadas por utilizadores válidos ao longo do tempo, foi realizada mais uma análise cujo resultado pode ser observado na Figura 29. Este gráfico permite perceber que os utilizadores válidos – nomeadamente, o "agent" e o "root" – realizaram algumas tentativas de *login* falhadas tanto no dia 7 como no dia 8 de maio. O utilizador com mais tentativas foi o "agent" que, por sua vez, também corresponde ao único utilizador que fez *login* com sucesso neste intervalo de tempo.

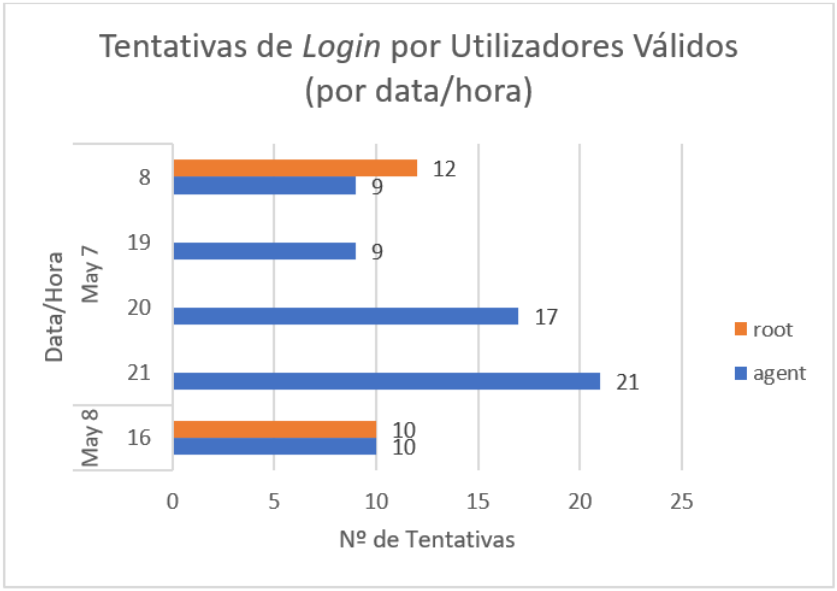


Figura 29 - Tentativas de *login* por utilizadores válidos (por data/hora)

Foi ainda realizada uma análise para se obter as tentativas de *login* falhadas pelos utilizadores. A Figura 30, que apresenta o gráfico obtido pela análise, permite verificar que as tentativas de *login* falhadas foram mais intensas no dia 8 de maio, sendo que as tentativas realizadas ocorreram todas no espaço de 1 minuto. Deste modo, pode-se desconfiar que estas tentativas de *login* foram realizadas com recurso a ferramentas de ataque *brute-force*, sendo, por isso, um ataque mais automatizado. Também se verifica pelo baixo número de utilizadores que surge no gráfico que o dicionário utilizado a nível de utilizadores e/ou *passwords* foi pequeno, como já se suspeitava anteriormente. É de referir que o utilizador com mais tentativas de *login* no dia 7 de maio foi o "root" – que é um utilizador válido – e no dia 8 de maio foi o "guest" – que é um utilizador inválido.

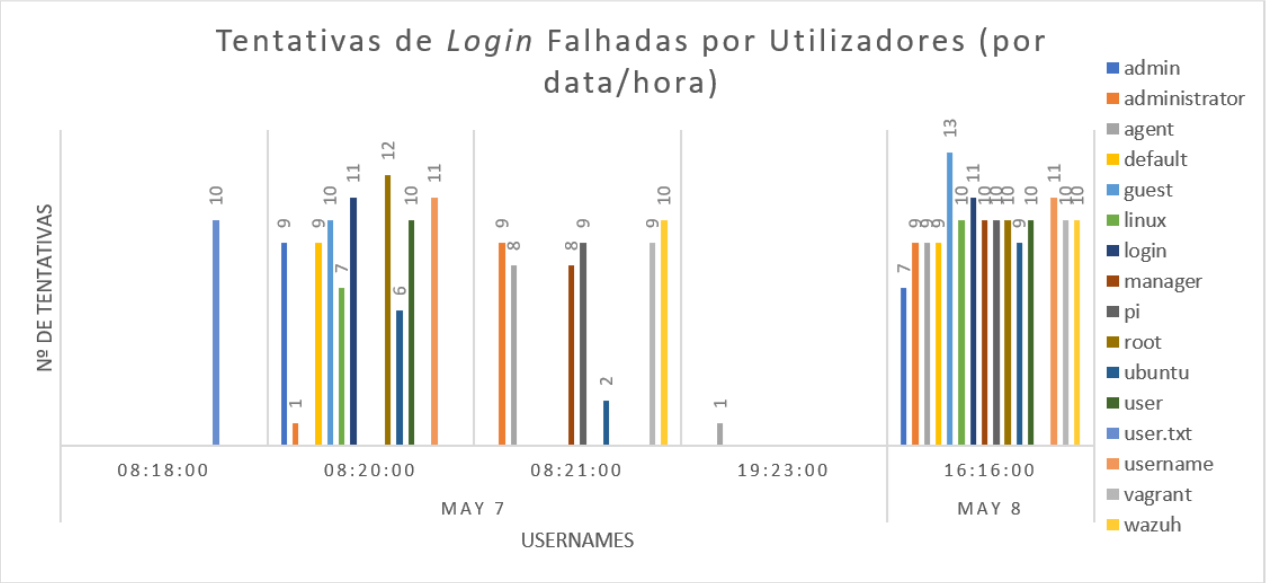


Figura 30 - Tentativas de *login* falhadas por utilizadores (por data/hora)

Por fim, através desta ferramenta, foi realizada uma última análise aos dados de modo a se entender se foram realizados *logins* com sucesso após os picos de tentativas de *login* identificados nas análises anteriores. Para isso concretizou-se uma análise que teve como resultado o gráfico na Figura 31. Com a observação da mesma percebe-se que a primeira tentativa de *login* com sucesso ocorreu pelas 08:21H no dia 7 de maio - que se insere no tempo correspondente ao pico de tentativas de *login*. Já no dia 8 de maio, também no tempo correspondente ao pico das mesmas, verifica-se a existência de uma tentativa de *login* com sucesso (*state* = 0). É de referir que durante o período entre os picos foram efetuadas maioritariamente *logins* com sucesso, sendo que o seu número não é alarmante.

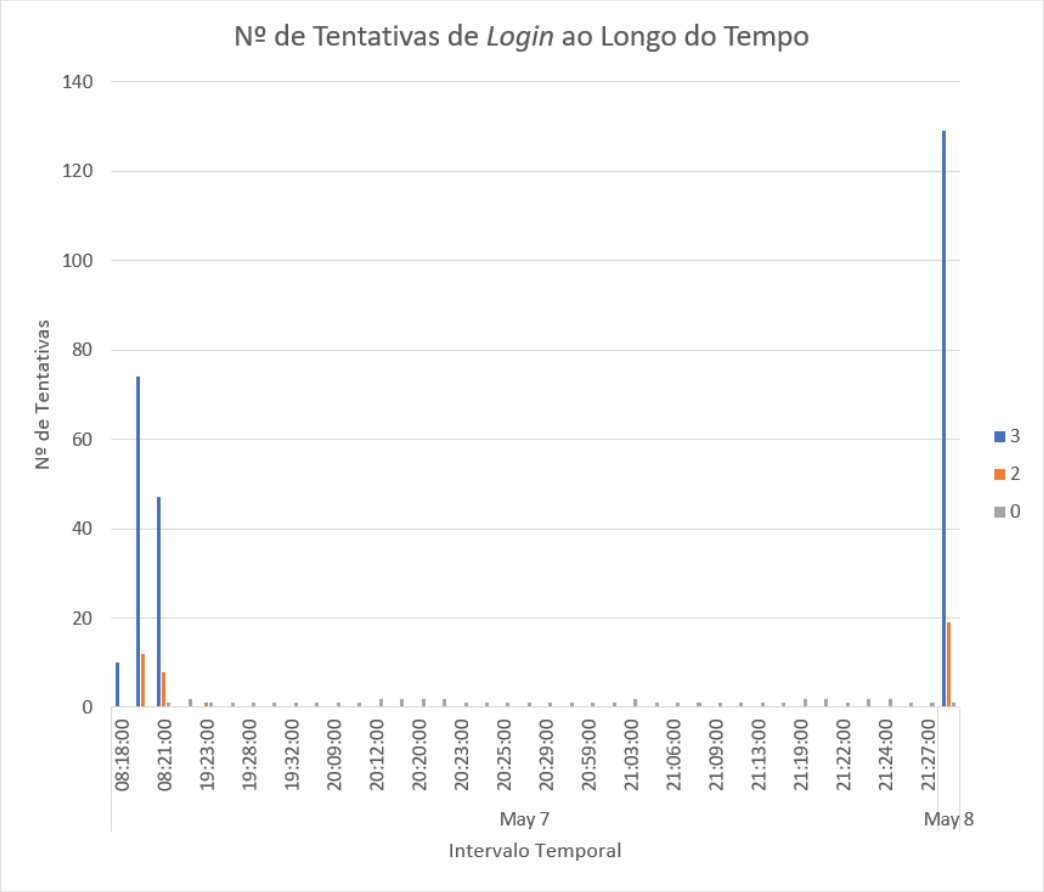


Figura 31 - Nº de tentativas de *login* ao longo do tempo

Conclui-se assim que os ataques, tanto do dia 7 como do dia 8, tiveram sucesso devido à tentativa de *login* com sucesso (*state* = 0) que existe em cada um desses períodos de picos. Cruzando com a informação já obtida através das outras pesquisas, é possível determinar que o utilizador "agent" foi aquele que ficou comprometido na sequência dos ataques.

8. Conclusões

Para este trabalho foi concebido um cenário que permitisse a observação do funcionamento de um sistema de SIEM que monitorizasse um dado conjunto de máquinas. Foi utilizado o sistema de SIEM *Wazuh*, tendo esta escolha por base o facto de esta plataforma ser a solução gratuita mais completa.

Relativamente à instalação, esta foi um processo simples dado que a documentação do *Wazuh* fornece todo o suporte e informação necessária para a realização desta fase. Contudo, é de notar que a instalação e configurações utilizadas foram unicamente em ambiente de linha de comandos.

Para o cenário em causa os recursos utilizados foram suficientes sendo que o sistema apresentou um desempenho eficiente. No entanto, num cenário de maiores dimensões é necessário ter uma maior atenção aos recursos da máquina correspondente ao *Wazuh server*.

Relativamente aos ataques realizados, todos estes foram bem-sucedidos, no entanto dois deles não foram identificados pelo *Wazuh* nomeadamente o ataque ao *Windows* que deu acesso a uma *shell* e o *port-scanning* com o *Nmap*. Contudo, provavelmente, com algumas configurações e regras especificas fosse possível detetá-los com o *Wazuh*.

Os dados recolhidos foram analisados de duas formas distintas - através da análise pelo *plugin* do *Wazuh* para o *Kibana* e através da análise do *data set* criado referente aos ataques SSH efetuados à máquina *Ubuntu* – sendo que ambas as análises permitiram obter algumas conclusões.

Com a primeira forma de análise mencionada, verificou-se que a interface é simples e intuitiva, sendo a sua utilização fácil. Apresenta várias funcionalidades interessantes como a análise em tempo real e os vários filtros que se podem aplicar, podendo assim se obter resultados mais direcionados para o objetivo do utilizador. É possível também realizar-se facilmente pesquisas e obter os resultados em vários tipos de gráficos, podendo-se, posteriormente, caso se pretenda, obter relatórios das pesquisas efetuadas. É de referir que é ainda possível interagir com a API e assim, por exemplo, executar *scripts* que permitem realizar pesquisas automatizadas. Também é possível visualizar-se, de forma simples, que alertas estão a ser despoletados e até mesmo visualizar-se os alertas de acordo com o seu nível de severidade. Verificou-se a existência de outras

funcionalidades que não foram exploradas para este trabalho como, por exemplo, o *scanning* de vulnerabilidades nos *agents* e o *scanning* de ficheiros maliciosos com o *VirusTotal*. Ao longo da utilização do *plugin* do *Wazuh* para o *Kibana* verificou-se que através da interface da sua interface não é possível, por exemplo, realizar configurações, criar alertas ou editar, sendo necessário efetuar estas tarefas através da linha de comandos. Apesar de ser possível extrair os alertas, constatou-se que não é possível extrair os ficheiros *log* que despoletam esses alertas.

No que diz respeito ao sistema de SIEM *Wazuh*, uma das vantagens é este permitir a implementação de diferentes tipos de arquitetura pudendo ser possível distribuir a "carga" pelos recursos computacionais disponíveis. Outro aspeto positivo é este já ter algumas configurações e alertas implementados, não sendo necessário configurar o sistema a partir do zero. O facto de este sistema de SIEM estar disponível para várias plataformas é um outro aspeto a ter em consideração. Já um aspeto negativo é o facto de o *Wazuh* não permitir obter os vários ficheiros *log* que os *agents* criam, sendo só possível obter um ficheiro *log* com todos os registos que os *agents* enviam. Para concluir, dada a quantidade de funcionalidades que o *Wazuh* apresenta, à sua boa documentação e ainda o facto de este disponibilizar tudo isto de forma gratuita dependendo do cenário em causa, tornam esta solução bastante apelativa.

Devido aos formatos dos ficheiros *log* recolhidos, a criação do *data set* não é um processo trivial e, por isso, foi optado fazer-se unicamente para o serviço SSH. A análise foi efetuada com o *Rapidminer*, o *Power BI* e o *Microsoft Excel*, notando-se que a análise com o *Rapidminer* é muito mais limitada, mas com o *Power BI* e com o *Microsoft Excel* é mais alargada, simples e intuitiva. Com a análise deste *data set* foi possível detetar o que era já detetado utilizando-se o SIEM, no entanto, uma análise utilizando o SIEM fornece muito mais informação do que uma análise feita com o *data set* criado.

Em suma, com este trabalho foi possível compreender o funcionamento de um sistema de SIEM e a sua importância nos dias de hoje, bem como a sua inserção em contexto empresarial para a deteção de incidentes de segurança e a sua posterior resposta. Assim, verifica-se que os objetivos delineados para este trabalho foram cumpridos.

Referências

[1] D. R. Miller, S. Harris, A. A. Harper, S. VanDyke e C. Blask, Security Information and Event Management (SIEM) Implementation, Network Pro Library, 2011.

[2] L. M. d. S. V. Ferreira, "A multi-level model for risk assessment in SIEM," Universidade de Lisboa - Faculdade de Ciências (FC), 2017.

[3] L. Sousa, "Sonorização de eventos gerados por um SIEM," Politécnico do Porto, 2016.

[4] P. Marques, "Assessment on the effectiveness of design diversity for network security and monitoring," Universidade de Lisboa, 2018.

[5] K.-O. Detken, D. Scheuermann e B. Hellmann, "Using extensible metadata definitions to create a vendor-independent SIEM system," International Conference in Swarm Intelligence, 2015.

[6] S. Zeinali, "Analysis of security information and event management (siem) evasion and detection methods," Tallinn University of Technology, 2016.

[7] P. Rodrigues, "Resilient event collection in SIEM systems," Universidade de Lisboa, 2013.

[8] Graylog, "VISUALIZE AND CORRELATE IDS ALERTS WITH OPEN SOURCE TOOLS," [Online]. Available: https://www.graylog.org/post/visualize-and-correlate-ids-alerts-with-open-source-tools. [Acedido em abril 2019].

[9] Wazuh, "Documentation Wazuh - Architecture," [Online]. Available: https://documentation.wazuh.com/current/getting-started/architecture.html. [Acedido em abril 2019].