UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Controlos de Cibersegurança em Ambientes MS Windows de Grandes Empresas: Integração Efetiva de Eventos Relevantes de Segurança no SIEM AlienVault USM

Rodrigo Serrano dos Santos

Mestrado em Segurança Informática

Trabalho de Projeto orientado por:
Prof. Doutora Maria Dulce Pedroso Domingos
Eng. José António dos Santos Alegria

2018

Agradecimentos

Agradeço à Superintendência das Tecnologias de Informação da Marinha e à Direção de Tecnologias da Informação e Comunicações que, respetivamente na figura do Sr. Contra-Almirante Superintendente e dos Srs. Diretor e Subdiretor, concederam apoio e disponibilidade que em muito contribuíram para a realização deste trabalho.

Ao Capitão-Tenente Paulo Jorge Baptista das Neves, que durante o período do mestrado, com o seu saber, disponibilidade, apoio e orientação contribuiu para alcançar os objetivos propostos.

Agradeço à Professora Doutora Dulce Domingos por orientar este projeto, pelos conselhos e observações relevantes que foram dados, e por se mostrar sempre disponível para ajudar e contribuir para o seu sucesso. Ao Engenheiro José Alegria, pela coorientação na Altice Portugal, agradeço a disponibilidade e confiança prestadas de forma incondicional, que contribuíram de forma decisiva para o sucesso deste projeto.

Agradeço a toda a equipa do SOC da Altice Portugal, em particular ao Engenheiro Jorge Silva, por ter estado sempre presente em todos os momentos de dúvida, desde o planeamento até à concretização deste trabalho, pela sua colaboração e disponibilidade.

Agradeço à minha família, e em especial à minha noiva Catarina, que formam um forte alicerce para alcançar os meus objetivos, através do amor, força, paciência, apoio, determinação e incentivo.

A todos aqueles que de forma direta ou indireta
contribuíram para a realização deste projeto,
o meu muito obrigado.

Resumo

Atualmente, vivemos numa era cada vez mais digital, onde o cibercrime tem vindo rapidamente a ganhar terreno através de redes organizadas mais sofisticadas, e com acesso a mais e melhores recursos. Neste contexto, os CSOC (*Cyber Security Operation Center*) são indispensáveis para detetar, analisar e combater esse avanço, e garantir a cibersegurança das organizações, sobretudo aquelas em que as suas redes assumem uma dimensão elevada, ou que esteja envolvida nos seus processos informação sensível.

Por forma a ser humanamente possível tratar a informação recebida pelo CSOC, esta deverá passar por um processo onde são coletados eventos das fontes de informação (dispositivos de rede, plataformas de segurança e sensores), sendo estes filtrados, normalizados, classificados e correlacionados num SIEM (*Security Information and Event Management*), dando origem à geração de alertas em caso de incidentes de segurança.

Este projeto visa adicionar fontes de informação que atualmente a Altice Portugal não dispõe no SIEM *Alienvault USM* (*Unified Security Management*), relativas aos eventos de segurança das inúmeras estações de trabalho e servidores *MS (Microsoft) Windows* a operar dentro da rede corporativa e infraestruturas de datacenter (perto de 25 mil). Esta informação irá permitir uma melhor cobertura da sua infraestrutura, e deteção de anomalias de segurança em tempo real. Estas capacidades são fundamentais, dado que a maioria dos ataques com recurso a *malware* são direcionados aos ecossistemas *MS Windows*.

Torna-se assim necessário, desenvolver um processo técnico robusto e eficiente de fazer chegar, de forma adequadamente filtrada ao SIEM, os eventos relevantes de segurança dos dispositivos *MS Windows* (tipicamente associados a *Indicators of Compromise* (IoC) conhecidos). De igual forma, é necessário desenvolver regras personalizadas no SIEM, que permitam endereçar os alertas relevantes ao CSOC, através da correlação e deteção de padrões anómalos dos eventos recebidos.

A ferramenta proposta neste projeto aumenta o nível de segurança da infraestrutura de rede da empresa, oferecendo uma melhor visibilidade dos sistemas MS Windows, de forma não intrusiva. Esta solução apresenta uma elevada capacidade de escalabilidade a nível da recolha e centralização de eventos, assim como na deteção de comportamentos anómalos, através da criação de diretivas personalizadas no SIEM.

Palavras-chave: Eventos de Segurança, *MS Windows*, SIEM, Deteção de Intrusões, Cibersegurança.

Abstract

Nowadays, we live in an increasingly digital age, where cybercrime has growing very quickly, through increasingly sophisticated organized networks and the ease of access to more and better resources. In this context, Cyber Security Operation Centers (CSOC) are indispensable to detect, analyze and face this advance, and guarantee the cyber security of organizations, especially those in which their networks are large, or sensitive information is involved in their processes.

In order to be humanly possible to handle the information received by the CSOC, it must pass through a process where network and platform events are collected by sensors, which are filtered, normalized, classified and correlated in a SIEM (Security Information and Event Management), to be able of rise alarms in case of security incidents.

This project aims to add sources of information that currently Altice Portugal does not have in the Alienvault USM (Unified Security Management) SIEM, related to the security events of MS (Microsoft) Windows workstations and servers operating within the corporate network and datacenter infrastructures (about 25 thousand). This will enable a better coverage of their infrastructure, and detection of security anomalies in real time. These are key capabilities, as most malware attacks target MS Windows ecosystems.

It is therefore necessary to develop a robust and efficient technical process to ensure that relevant security events of the MS Windows devices (typically associated with known Indicators of Compromise (IoC)) arrive in an adequately filtered way to the SIEM. Likewise, it is necessary to develop customized rules in the SIEM, which allow addressing relevant alerts to the CSOC team, through the correlation and detection of anomalous patterns of the events received.

The tool proposed in this project increases the level of security of the company's network infrastructure, offering better visibility of MS Windows systems, in a non-intrusive way. This solution reveals a high scalability in the event collection and centralization environment, as well in the detection of abnormal behaviors, through the creation of customized directives in SIEM Alienvault USM.

Keywords: Security Events, MS Windows, SIEM, Intrusion Detection, Cybersecurity.

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos e Siglas

AD Active Directory

AD CS Active Directory Certificate Services

AD DS Active Directory Domain Services

B2B Business-to-business

CEIP Customer Experience Improvement Program

CIDF Common Intrusion Detection Framework

CPU Central Processing Unit

CSE Cyber Security Engineering

CSIRT Computer Security Incident Response Team

CSOC Cyber Security Operations Center

Data Center Centro de Processamento de Dados

DBO Direção de Operações B2B

DHCP Dynamic Host Configuration Protocol

DIT Direção de Informação e Tecnologia

DNS Domain Name System

DoS Denial-of-service

FQDN Fully Qualified Domain Name

GPO Group Policy

HIDS Host-based IDS

ICT Information and Communications Technologies

IDS Intrusion Detection System

IIS Internet Information Services

IoC Indicators of Compromise

LDAP Lightweight Directory Access Protocol

MDM Mobile device management

MS Microsoft

MSRT Malicious Software Removal Tool

NIDS Network-based IDS

OMS Operations Management Suite

OU Organizational Unit

RAC Reliability Analysis Component

RAM Random-access Memory

SCOM System Center Operations Manager

SIEM Security Information and Event Management

SOC Security Operation Center

SSL Secure Socket Layer

USM Unified Security Management

vCPU CPU Virtual

WEC Windows Event Collector

WEF Windows Event Forwarding

Capítulo 1

Introdução

"More and more, modern warfare will be about people sitting in bunkers in front of computer screens, whether remotely piloted aircraft or cyber weapons."

(Philip Hammond)

1.1 Motivação

O contexto atual da cibersegurança caracteriza-se por uma crescente complexidade e volumetria de ataques, e pela existência de redes de cibercrime cada vez mais sofisticadas, com mais e melhores recursos. Estas redes são eventualmente suportadas por Estados, cuja economia pode-se basear em parte nesta atividade, dado o retorno financeiro que pode gerar, ou apenas por motivações políticas [1].

Este contexto leva a um aumento do risco para o funcionamento das organizações, com ciberataques progressivamente mais destrutivos e com capacidade efetiva de interrupção de setores de negócio vitais para o funcionamento da economia. Esta tendência tem-se verificado nos ataques mais recentes, como o *WannaCry* (maio de 2017) e o *NotPetya* (junho de 2017), que causaram danos a nível mundial e de forma transversal, afetando diversos setores, tais como a banca, a saúde, os operadores de serviços de comunicações, entre outros [1].

De forma a mitigar esse risco, os registos de eventos produzidos pelos sistemas operativos e pelas aplicações são de vital importância, uma vez que contribuem com informação para análise e investigação dos incidentes de cibersegurança [2]. A correlação de eventos e a sua análise é fundamental para identificar incidentes de segurança, violações de políticas de domínio e atividades fraudulentas [3]. Neste

contexto é essencial a existência de mecanismos que permitam efetuar a recolha, filtragem, normalização, análise e correlação desses eventos [2].

Muitas organizações, efetuam uma recolha centralizada de eventos apenas dos seus sistemas críticos de forma contínua. Este tipo de abordagem pode ser motivado por diversas razões, como por exemplo: a maioria das estações de trabalho executar o sistema operativo *MS (Microsoft) Windows*, que não permite uma integração direta dos eventos com a maioria dos SIEM (*Security Information and Event Management*) existentes; ou a enorme quantidade de dados que serão gerados na rede, dado o número de máquinas existentes. Estas dificuldades acabam, muitas vezes, por levar as organizações a negligenciar a recolha de eventos de segurança das estações de trabalho de forma centralizada, criando um vazio de informação relevante para permitir obter um panorama mais abrangente da cibersegurança na organização [4].

1.2 Enquadramento Institucional

A Altice Portugal, como líder na área de ICT (*Information and Communications Technologies*), possui um CSOC (*Cyber Security Operation Center*) e CSIRT (*Computer Security Incident Response Team*) de referência nesta área para clientes e para uso interno no Grupo Altice.

Desta forma, a organização tem como responsabilidade lidar com todas as ameaças à confidencialidade, integridade, autenticidade e disponibilidade da informação e sistemas que a suportam.

Para cumprir esse objetivo, o CSOC dispõe de sistemas SIEM para detetar, com base em eventos capturados em tempo real no máximo número possível de fontes de informação, padrões que indiciem anomalias de segurança ou mesmo ataques em preparação ou já em curso.

Este projeto está inserido na melhoria deste último ponto, na medida em que visa adicionar fontes de informação que atualmente a Altice Portugal não dispõe no SIEM, relativas aos eventos de segurança das estações de trabalho e servidores *MS Windows*.

1.3 Objetivos

Este projeto tem como objetivo adicionar ao SIEM *Alienvault Unified Security Management (*USM*)* da Altice Portugal, os eventos de segurança relevantes das estações de trabalho e servidores *MS Windows*, a operar dentro da rede corporativa e infraestruturas de datacenter, garantindo uma melhor cobertura da sua infraestrutura e deteção de anomalias de segurança em tempo real. Estas capacidades são fundamentais, dado que a maioria dos ataques com recurso a *malware* têm início, e propagam-se, através dos ecossistemas *MS Windows*, que são utilizados na maioria das grandes empresas, e dos quais estas se encontram dependentes.

Torna-se necessário, desenvolver um processo técnico robusto e eficiente de fazer chegar, de forma adequadamente filtrada ao SIEM, todos os eventos relevantes de segurança dos dispositivos *MS Windows* (tipicamente associados a *Indicators of Compromise* (IoC) conhecidos), de forma a permitir que este proceda à correlação e deteção de padrões anómalos, através da criação de regras personalizadas, que permitam endereçar os alertas relevantes ao CSOC.

Desta forma, é necessário definir quais os eventos de segurança considerados relevantes para uma correta monitorização dos sistemas, bem como as plataformas a serem utilizadas para a sua recolha, filtragem, tratamento e gestão centralizada.

Adicionalmente, por forma a ser possível efetuar a correlação e deteção automática de padrões anómalos, para detetar ataques e possibilitar uma rápida resposta aos incidentes, torna-se necessário desenvolver regras de correlação que permitam ao SIEM detetar, de forma efetiva, anomalias relevantes a serem endereçadas ao CSOC.

1.4 Contribuições

A contribuição deste projeto está diretamente relacionada com o enriquecimento da informação disponibilizada ao CSOC da Altice Portugal para permitir detetar os incidentes que estejam a ocorrer na rede. Assim, é possível identificar as seguintes contribuições:

- Assente na política de implementação de segurança em profundidade, este projeto veio adicionar uma nova camada de segurança na infraestrutura de

rede da Altice Portugal, permitindo uma maior visibilidade e deteção de ataques direcionados ao ecossistema *MS Windows*.

- Criação de um processo robusto, eficiente, e escalável, que permita efetuar a recolha, filtragem, tratamento e gestão centralizada dos eventos de segurança dos dispositivos *MS Windows*, e a sua integração com o SIEM da Altice Portugal.
- Deteção de ataques e geração de alertas de interesse, com vista a possibilitar uma rápida resposta aos incidentes, através da extração, interpretação e correlação da informação relevante dos eventos de segurança dos sistemas *MS Windows* recebidos pelo SIEM.

Em suma, este projeto visa tornar cada sistema *MS Windows* num sensor que irá contribuir com informação valiosa para melhorar o panorama global de monitorização de segurança.

1.5 Organização do documento

O documento está dividido em oito capítulos, da seguinte forma:

Capítulo 1: É o presente capítulo, e contém uma introdução onde são apresentados em traços gerais a motivação, o enquadramento institucional, os objetivos que são propostos alcançar, e as contribuições do projeto.

Capítulo 2: Descreve do ponto de vista teórico, o processo de deteção de intrusões e a sua contribuição para um SIEM. Neste capítulo é ainda apresentado o SIEM *Alienvault USM*.

Capítulo 3: Aborda, de forma genérica, algumas das soluções disponíveis para implementar um processo consolidado de recolha, filtragem, tratamento e gestão centralizada de eventos dos sistemas *MS Windows*, e a sua integração num SIEM.

Capítulo 4: Descreve e analisa a solução proposta pela *Logbinder* para efetuar a gestão e monitorização centralizada do ambiente *Windows Event Collection* (WEC).

Capítulo 5: Aborda a análise do problema que levou à elaboração deste projeto, bem como os requisitos que devem ser satisfeitos e a arquitetura de alto nível para a implementação da solução.

Capítulo 6: Descreve o processo de implementação e avaliação através dos diferentes casos de estudo analisados em ambiente de laboratório, bem como a arquitetura e recursos necessários para a concretização da implementação da solução em ambiente de produção.

Capítulo 7: Este capítulo efetua a avaliação do trabalho desenvolvido, com base no cumprimento dos objetivos e requisitos propostos, e nos resultados de desempenho obtidos em laboratório pela solução implementada.

Capítulo 8: Neste capítulo final são apresentadas as conclusões do projeto, bem como as opções tomadas com vista a atingir os objetivos propostos. Contempla ainda o trabalho futuro que poderá ser desenvolvido com vista a melhorar os processos implementados e amplificar a capacidade de análise, correlação e deteção automática de padrões anómalos, de forma a antecipar mais tipos de ataques, possibilitando uma rápida resposta aos incidentes.

Capítulo 2

Deteção de Intrusões

A deteção de intrusões é um importante componente de um sistema de segurança, uma vez que fornece informação relevante para permitir a deteção de ataques [5].

Este capítulo descreve os fundamentos teóricos obtidos através da consulta e análise bibliográfica relevante, que permitam compreender o processo de deteção de intrusões e a sua contribuição para um *Security Information and Event Management* (SIEM).

Apresenta ainda a plataforma *Alienvault Unified Security Management* (USM), uma plataforma de monitorização de segurança desenvolvida com a capacidade de operar em redes tradicionais, ambientes de *cloud* e redes híbridas. Esta plataforma é um dos SIEM atualmente utilizado pela Altice Portugal, e o selecionado para a implementação deste projeto. Este SIEM inclui capacidades essenciais de segurança, que permitem a deteção de ameaças e resposta a incidentes de forma integrada e efetiva [6].

2.1 Sistema de Deteção de Intrusões (IDS)

Um sistema de deteção de intrusões (IDS) é um importante componente de segurança, utilizado para identificar e isolar intrusões em sistemas de computadores, sendo que a informação fornecida por estes sistemas poderá ainda ser útil para auxiliar na análise forense de um ataque [5].

Apesar de existirem vários tipos de IDS, tais como sistemas baseados na análise de rede (*Network-based IDS*), e sistemas baseados na análise do equipamento local onde se encontra instalado (*Host-based IDS*), é possível utilizar o modelo *Common Intrusion*

Detection Framework (CIDF) para definir um conjunto de componentes que permitem representar a constituição de um IDS genérico [5].

O modelo CIDF, conforme Figura 2.1, é composto por quatro componentes [5]:

- ***Event (E) Box*:** Fornece informação dos eventos ao resto do sistema.
- ***Analysis (A) Box*:** Analisa os eventos produzidos pelas fontes de eventos. Esta análise pode ser baseada em diferentes técnicas, tais como:
    - Baseada em Comportamento (deteção de anomalias): Analisa o comportamento padrão do sistema através de técnicas de inteligência artificial, e alerta a ocorrência de comportamentos anómalos.
    - Baseada em Conhecimento (deteção através de assinatura): Dispõe de uma base de dados com padrões de ataques conhecidos, e utiliza algoritmos estatísticos para os comparar com os comportamentos que estão a ocorrer.
- ***Storage (D) Box*:** Armazena e mantém disponível a informação dos eventos.
- ***Countermeasure (C) Box*:** Executa ações para parar ataques em curso e/ou evitar futuros ataques.

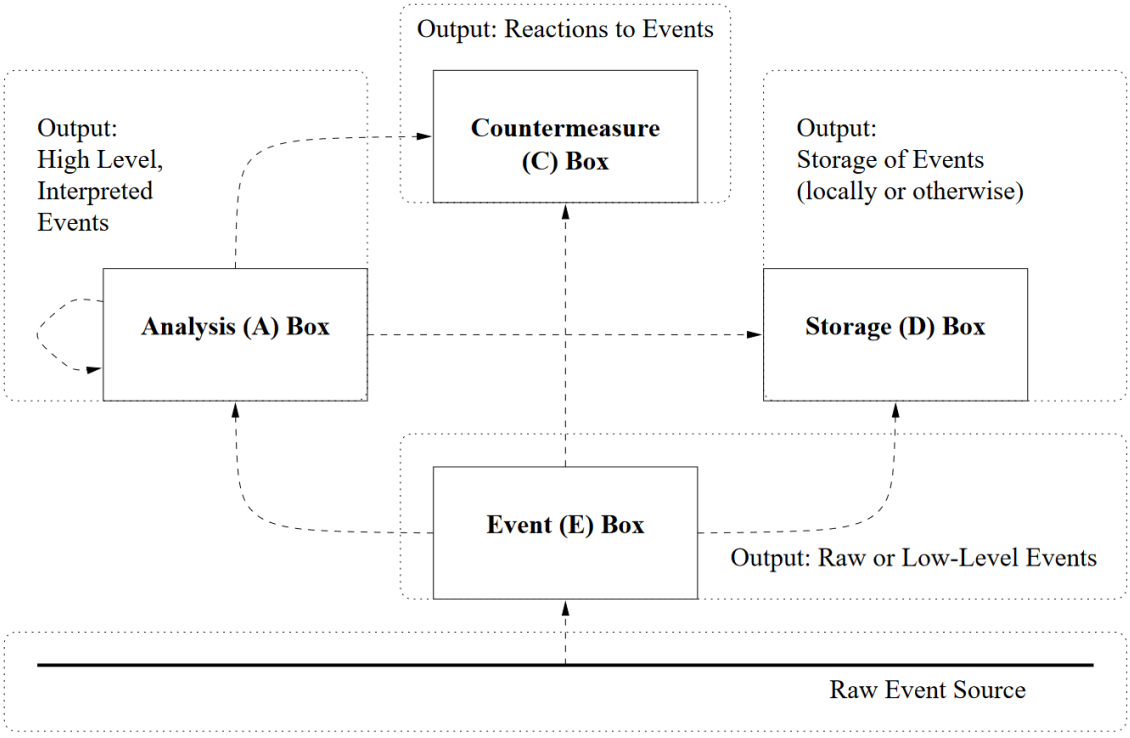


Figura 2.1 – Componentes do Modelo CIDF (extraído de [5])

2.2 *Security Information and Event Management (*SIEM*)*

Um *Security Information and Event Management* (SIEM) é a combinação de dois sistemas, um *Security Information Management* (SIM) e um *Security Event Management* (SEM). Os sistemas SIM coletam e armazenam a longo prazo os eventos para análise de tendências e criação de relatórios, enquanto os sistemas SEM interpretam e correlacionam os eventos para análise de segurança em tempo real, proporcionam uma melhor visualização gráfica do ambiente, e produzem notificações ou alertas, o que permite agilizar a realização de ações defensivas. Ao combinar essas duas tecnologias num SIEM, obtemos uma melhor identificação, análise e recuperação de uma eventual violação de segurança [7].

A utilização de um SIEM (ver Figura 2.2) proporciona uma melhor visibilidade da infraestrutura, através da análise de atividade e de comportamento. Ao aliarmos a capacidade de correlação das diferentes fontes de informação à visibilidade fornecida por um SIEM, permite que este nos impulsione para uma melhor identificação e resposta dos incidentes [8].

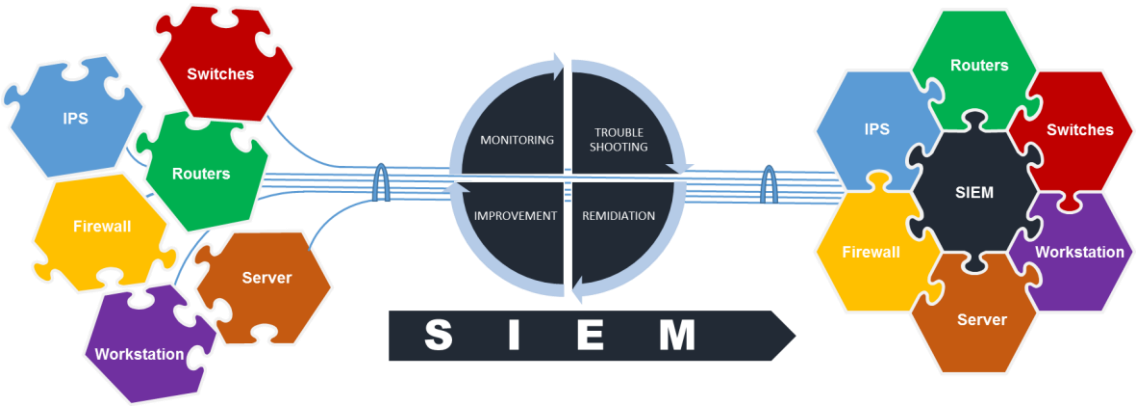


Figura 2.2 – Security Information and Event Management (SIEM) (extraído de[9])

2.2.1 Normalização de Eventos (*Parsing*)

Os eventos que chegam ao SIEM são na sua grande maioria no formato *Raw* e chegam a partir de diferentes sistemas, tais como computadores, servidores, *firewalls*, *routers*, *switches*, *proxy*, sistemas de deteção e/ou prevenção de intrusões, entre outros. Embora alguns destes sistemas compartilhem o mesmo formato de envio de eventos, existe uma grande quantidade de formatos e conteúdos de eventos diferentes [10].

A instalação de um SIEM requer a configuração das fontes de eventos, bem como da forma como os eventos recebidos de cada fonte específica deverão ser interpretados e transformados num formato padrão do próprio SIEM, de maneira a que possam ser armazenados na sua base de dados de eventos e utilizados para posterior correlação.

2.2.2 **Correlação de Eventos**

Numa organização é gerada uma grande quantidade de eventos de segurança a partir de diferentes sistemas, tais como computadores, servidores, *firewalls*, *routers*, *switches*, *proxy*, sistemas de deteção e/ou prevenção de intrusões, entre outros. Grande parte desses eventos não são gerados de forma isolada, ou seja, existe uma relação entre os eventos gerados pelos diferentes sistemas. Desta forma, a capacidade de correlacionar os eventos dos diversos sistemas, bem como os diversos eventos do mesmo sistema, tem um papel fundamental para garantir uma melhor e mais rápida identificação dos eventuais incidentes que estejam a ocorrer na rede.

É nesta fase que os eventos são relacionados entre si e agrupados, de forma a dar uma visualização mais alargada de um determinado caso de interesse. Em casos de incidentes específicos permitem ainda despoletar alertas. Por forma a despoletar alertas, grande parte dos SIEM disponibilizam um conjunto de regras de correlação que permitem detetar incidentes que estejam a ocorrer, e permitem também a criação customizada de novos tipos de correlação mais específicas ao ambiente de rede de cada organização [11].

2.2.3 **Alertas**

Os alertas têm como função notificar o mais próximo de tempo real possível sobre atividades ou comportamentos de interesse que estão a ocorrer na rede e nos sistemas, de forma a permitir tomar ações de mitigação automáticas, ou manuais. Estes podem ser gerados a partir de um único evento, ou a partir da correlação de diversos eventos que estão a ocorrer, de acordo com as regras definidas [12].

A geração de alertas automáticos é de vital importância para uma rápida identificação dos eventuais incidentes que estejam a ocorrer, uma vez que sem sistemas de alerta automatizados seria praticamente impossível identificar esses incidentes em tempo quase real, o que impossibilitaria a implementação de contramedidas de proteção

em tempo útil, reduzindo drasticamente a capacidade para evitar o comprometimento dos sistemas [12].

Para deteção destas atividades, deverá ser adotada uma estratégia que permita minimizar a ocorrência de falsos positivos e cumulativamente, de falsos negativos. Um número excessivo de falsos positivos, isto é, alertas a relatar anomalias não relevantes, geram "ruído" e consomem recursos especializados para serem analisados. Por outro lado, a existência de falsos positivos, isto é, a não deteção de uma atividade mal-intencionada, pode ter consequências nefastas para a organização.

2.3 SIEM na Altice Portugal

Com o aumento do número de dispositivos conectados, e a preocupação crescente com a sua segurança, em 2004, a Altice Portugal (à data Portugal Telecom) começou a fazer as primeiras auditorias de segurança às plataformas e serviços que estas disponibilizam.

Já no decorrer do ano de 2007, foi instalado o primeiro SIEM, tendo a escolha recaído pelo *ArcSight* da *Hewlett Packard*. Ao princípio, este começou por fazer a monitorização de perímetro, através dos eventos gerados pelas *Firewall*. Ao longo do tempo, foram sendo adicionadas fontes de eventos de outros sensores, de forma a permitir uma cobertura mais abrangente da rede.

Com o progressivo aumento da dimensão da infraestrutura, e com o objetivo de permitir efetuar uma separação entre a rede da Altice Portugal, e as redes dos clientes, que são geridas por esta, foi efetuado um estudo para a instalação de um segundo SIEM. O SIEM eleito para efetuar a monitorização da rede da Altice Portugal foi o *Alienvault USM*.

No princípio do ano de 2017 foi efetuada a implementação do SIEM *Alienvault USM*, permanecendo o SIEM *ArcSight* para a monitorização das redes dos clientes. Este SIEM, veio adicionar as funcionalidades de *Network-based IDS* (NIDS), e *Open Threat Exchange* (OTX), que não se encontravam presentes no *ArcSight*, permitindo aumentar a capacidade de deteção de intrusões.

Desde a sua implementação, até à presente data, a Altice Portugal tem efetuado esforços para adicionar o máximo possível de fontes de informação ao SIEM *Alienvault USM*, como é o caso deste projeto, que visa adicionar os eventos de segurança das

plataformas *MS Windows*, com o objetivo de enriquecer a informação que o SIEM disponibiliza ao CSOC.

2.4 *Alienvault USM*

O *Alienvault USM Appliance* é uma plataforma unificada que integra diversas tecnologias de segurança. Esta pode assumir uma configuração com apenas um servidor, ou uma configuração distribuída, com múltiplos servidores, de modo a garantir a escalabilidade e disponibilidade pretendida [13].

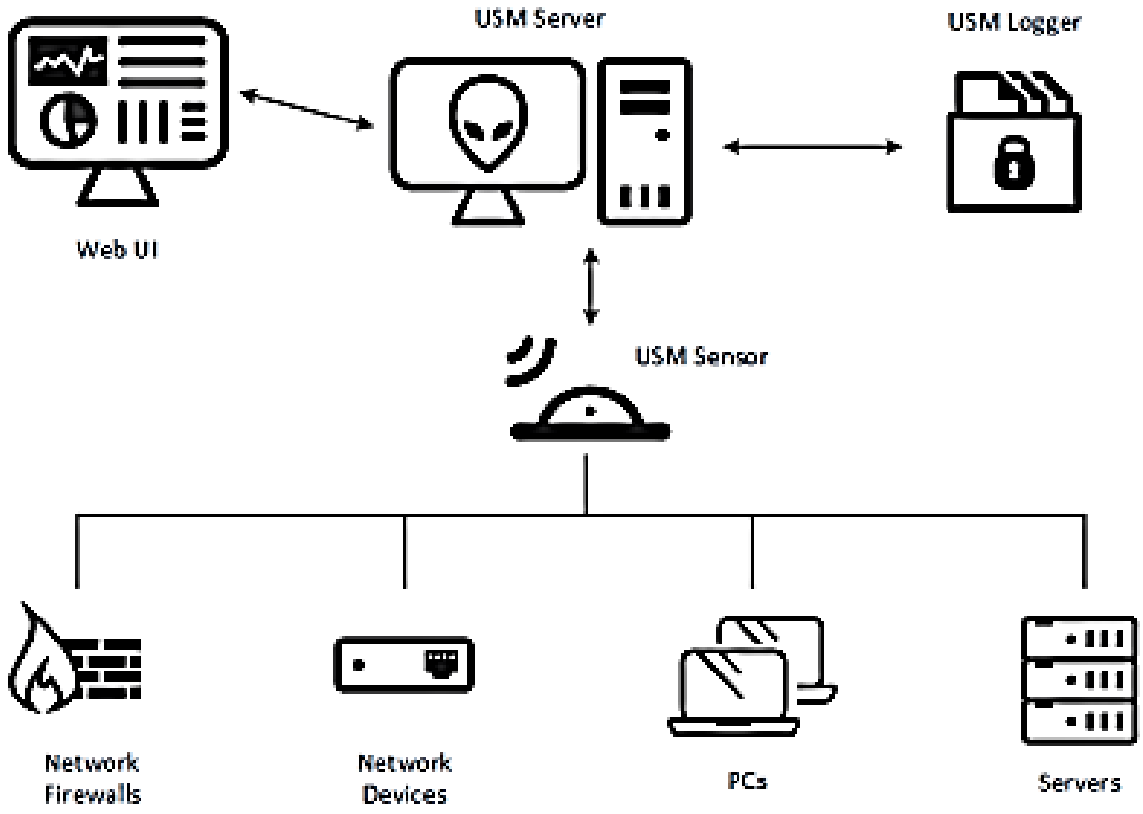


Figura 2.3 – Arquitetura de alto nível do AlienVault USM Appliance (extraído de *[13]*)

Os três componentes principais da arquitetura do *Alienvault USM Appliance* (ver Figura 2.3), que permitem a monitorização da rede são [13]:

- ***USM Sensor*:** Podem ser implementados em diversos pontos da infraestrutura de rede, permitindo coletar e normalizar informações de qualquer equipamento, reencaminhando-a de seguida para o *USM Server*.

- ***USM Server*:** Agrega e correlaciona a informação que os *USM Sensors* coletam. Fornece a capacidade de gestão, reporte e administração através de uma interface web, dando desta forma a capacidade SIEM ao *Alienvault*.
- ***USM Logger*:** Arquiva os eventos recebidos de forma segura para permitir análise forense e garantir conformidade.

O *Alienvault USM Appliance* tem a capacidade de coletar eventos de várias fontes, tais como equipamentos de rede, servidores ou aplicações. Esta plataforma disponibiliza diversos *plugins* para permitir a integração com os protocolos utilizados pelas fontes de eventos, tais como, *syslog*, *Windows Management Instrumentation* (WMI), *Security Device Event Exchange* (SDEE), entre outros. Os eventos recebidos são normalizados para extrair e armazenar a informação numa base de dados comum, permitindo melhorar a análise, pesquisa e correlação dos dados, para que estes contribuam de forma efetiva para a mitigação do risco, identificação de vulnerabilidades, deteção de ameaças, e priorização da resposta aos incidentes, através da análise de comportamentos [13].

2.4.1 Interpretação e Normalização de Eventos (*Parsing*)

A recolha de eventos é necessária para a operação do *Alienvault*, sendo que estes podem ser coletados de diversas fontes, tais como *firewalls*, *routers*, servidores, estações de trabalho, aplicações, entre outros. Todos os dados coletados das fontes de eventos são interpretados e normalizados para extrair e armazenar as informações em campos de dados comuns (Ver ANEXO A) que definem um evento, tais como o endereço IP, o nome de utilizador, entre outros. Após o processo de normalização, a informação dos eventos coletados das diversas fontes pode ser analisada em conjunto no *Alienvault*, permitindo efetuar pesquisas, correlações, deteção de ameaças e vulnerabilidades, e criação de alertas [14].

A interpretação dos eventos é efetuada através de *Plugins* que definem como transformar os eventos *Raw* específicos de cada sistema e normalizar essa informação em campos de dados de eventos comuns. O *Alienvault* dispõe de diversos *Plugins* para as fontes de eventos mais comuns, no entanto poderão ser criados *Plugins* customizados que se adaptem às necessidades de cada situação [14]. A normalização dos eventos *Raw* (ver Figura 2.4) em campos de dados de eventos comuns permite que o *Alienvault* exiba

a informação de forma uniforme e correlacione os eventos dos vários sistemas individuais para gerar alertas [15].

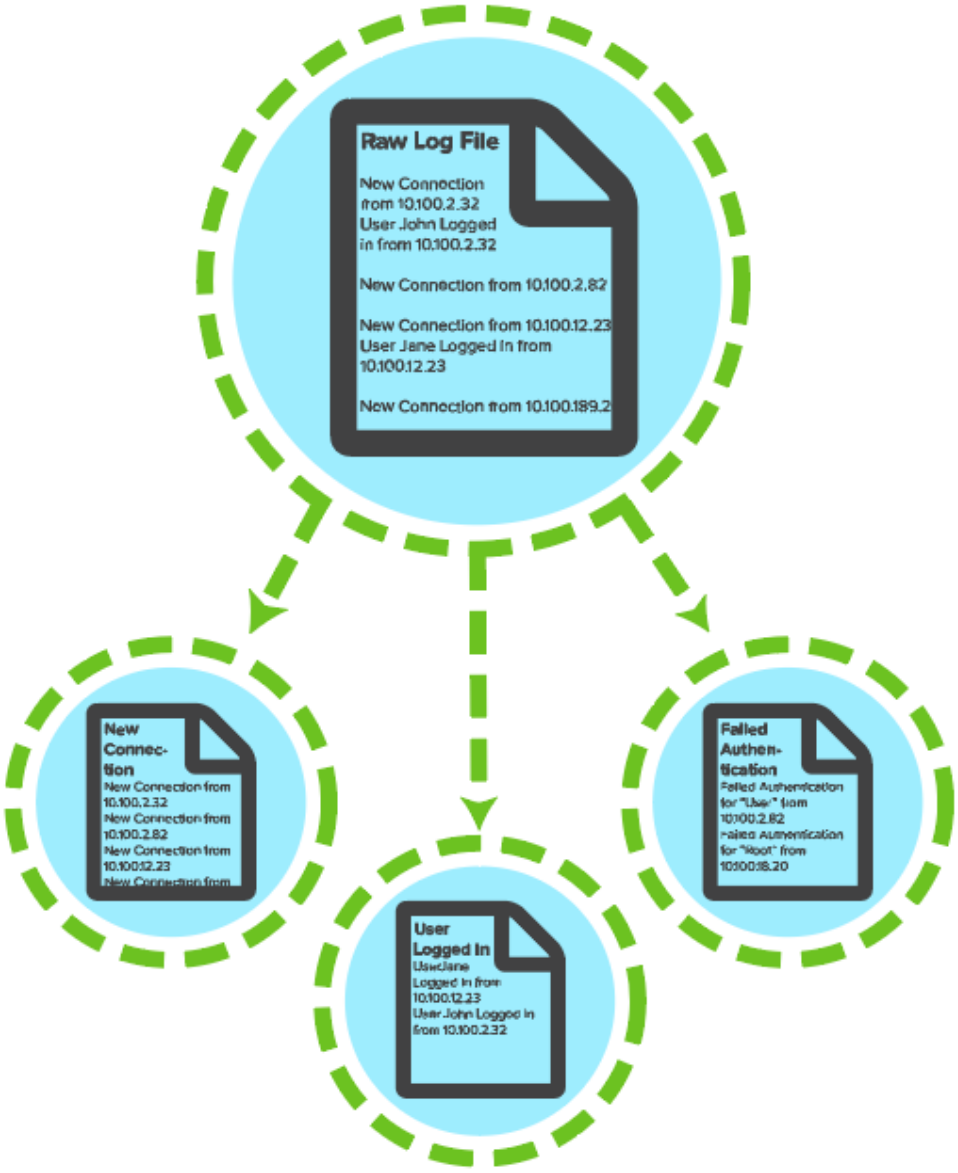


Figura 2.4 – Processo de Normalização de Eventos (extraído de *[15]*)

Por forma a abranger uma maior diversidade de sistemas, o Alienvault suporta dois tipos de *Plugins* [15]:

- ***Detector Plugins:*** Os sistemas a monitorizar escrevem os eventos num ficheiro de *log* localizado no *Alienvault*, de onde estes são extraídos por um *Plugin*.
- ***Monitor Plugins:*** Executam comandos, tais como *nmap* ou *tcptrack*, para recolher informação dos sistemas que se encontram a monitorizar.

2.4.2 Correlação de Eventos e Alertas

Após a normalização, processamento, análise e filtragem iniciais que o *Alienvault* realiza aos eventos, é efetuada a verificação dos seus campos, para que, caso estes correspondam aos critérios das regras de correlação, o evento seja encaminhado para o componente de correlação do *Alienvault*. Através da correlação é possível determinar padrões e sequências de eventos provenientes de diferentes sistemas. Os eventos recebidos podem ser processados múltiplas vezes pelo componente de correlação, uma vez que o mesmo evento pode ser processado por diferentes regras de correlação [14].

À medida que os eventos continuam a surgir no componente de correlação, o *Alienvault* gera alertas com base nas condições de eventos especificadas nas diretivas (regras de correlação). O *Alienvault* pode ainda ser configurado para receber informação de Indicadores de Compromisso (IOCs) através do programa *Open Threat Exchange* (OTX), que permite aumentar a capacidade de deteção de intrusões, através da correlação dos eventos de segurança com a informação de reputação dos endereços IP que é recebida através do OTX [14].

2.5 Resumo

Um SIEM permite coletar, normalizar, armazenar e correlacionar os eventos gerados pelas plataformas de segurança, possibilitando dessa forma a identificação e resposta a incidentes.

Este capítulo descreveu o processo de deteção de intrusões e a sua contribuição para um SIEM, assim como apresentou a plataforma *Alienvault Unified Security Management* (USM), que será o SIEM utilizado para a implementação deste projeto.

Capítulo 3

Eventos de Segurança do *Windows*

Os sistemas *MS Windows* disponibilizam uma grande quantidade de eventos, que de entre outros, são armazenados localmente em quatro *logs* principais: Aplicação, Segurança, Configuração e Sistema. Destes *logs*, o mais importante no âmbito desta tese é o *log* de Segurança, uma vez que regista as ações críticas do utilizador e/ou do sistema, conforme definido através das políticas de auditoria, tais como *logons* e *logoffs*, gestão de contas, acesso a objetos, entre outros. São uma das principais ferramentas utilizadas para detetar e investigar atividades não autorizadas, potenciais falhas de segurança, e servem como evidência no caso de violações de segurança. A *Microsoft* descreve-os como "*Your Best and Last Defense*" [16].

3.1 Formato dos Eventos do *Windows*

O *Windows Event Logging Framework* da *Microsoft* utiliza a estrutura de dados EVTX (oficialmente conhecido como *Windows Event Log*) a partir do sistema operativo *MS Windows Vista* e *Server 2008*, que sucedeu à estrutura EVT (conhecida como *Event Logging*), por esta não suportar o nível de requisitos de conformidade exigidos. A estrutura de dados EVTX inclui uma grande quantidade de novos recursos e aprimoramentos relativamente à estrutura de dados EVT, que inclui novas propriedades de eventos, o uso de canais para publicar eventos, a utilização do formato XML (*Extensible Markup Language*) para armazenamento dos eventos, e um novo Visualizador de Eventos [17].

O formato de dados XML (ver Figura 3.1), foi criado para permitir a partilha de dados estruturados num formato em que seja possível definir os seus elementos. Este formato aumenta consideravelmente a granularidade que pode ser aplicada na filtragem

e visualização dos eventos, permitindo efetuar uma filtragem por qualquer um dos campos que definem o evento [17].

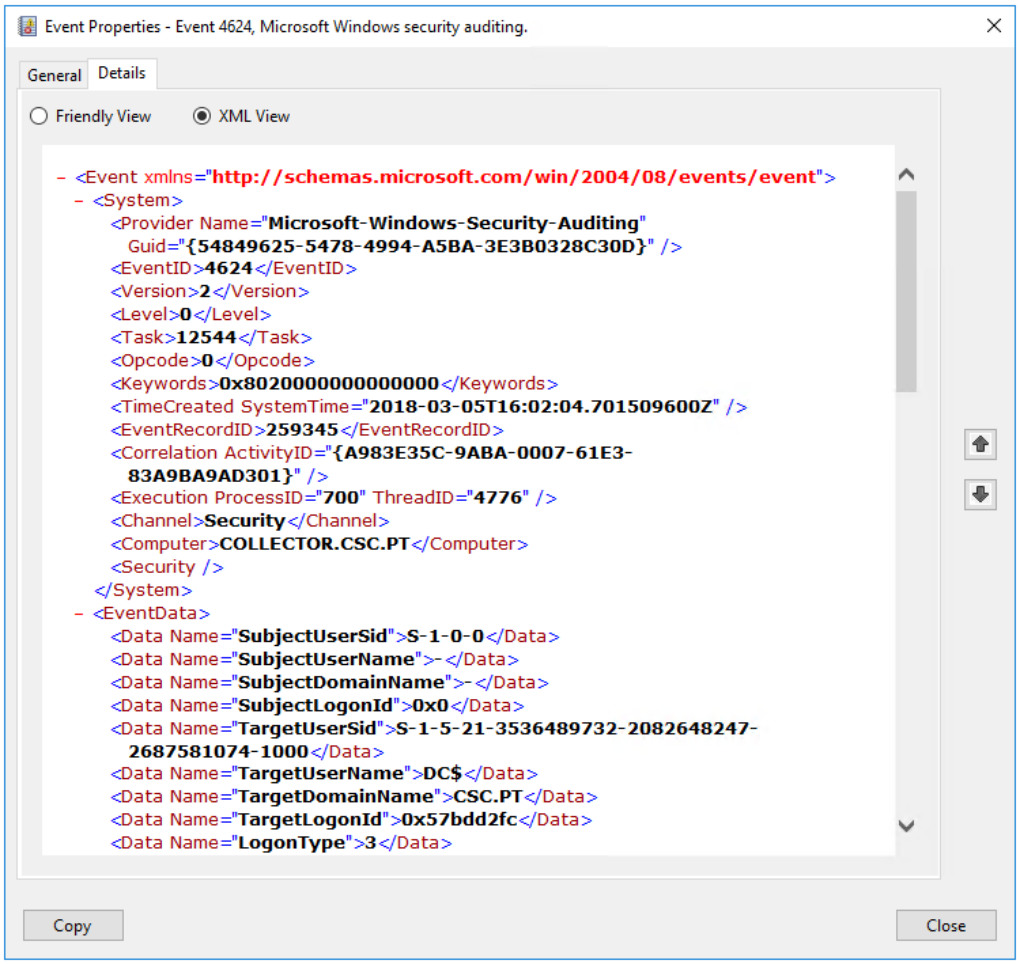


Figura 3.1 – Exemplo de Vista XML de um Evento do Windows

3.2 Auditoria de Eventos do *Windows*

A auditoria de eventos permite detetar ataques que já ocorreram, ou que estão a decorrer. No entanto, a determinação dos objetos a ser auditados deve ser seletiva, tendo por base as atividades de interesse a monitorizar (Subsecção 5.2.2), uma vez que uma auditoria muito abrangente cria sobrecarga no sistema, e a quantidade excessiva de dados gerados faz com que o *log* de segurança se torne demasiado grande e difícil de gerir [18].

Em ambientes *MS Windows*, para que seja efetuado o *log* dos eventos é necessário configurar e habilitar as políticas de auditoria, sendo que algumas já se encontram pré-

habilitadas no momento da instalação do sistema operativo. Estas políticas servem apenas para definir os tipos de eventos que serão gerados e armazenados nos *logs* [18].

As políticas de auditoria dividem-se em grupos, cada um composto por diversas subcategorias, que uma vez habilitadas são responsáveis por auditar um conjunto de eventos. Os principais grupos de políticas de auditoria, conforme se pode observar na Figura 3.2, são: Início de Sessão de Conta; Gestão de Contas; Controlo Detalhado; Acesso DS; Início de Sessão/Fim de Sessão; Acesso a Objetos; Alteração de Política; Utilização de Privilégios; Sistema; Auditoria de Acesso a Objetos Globais.

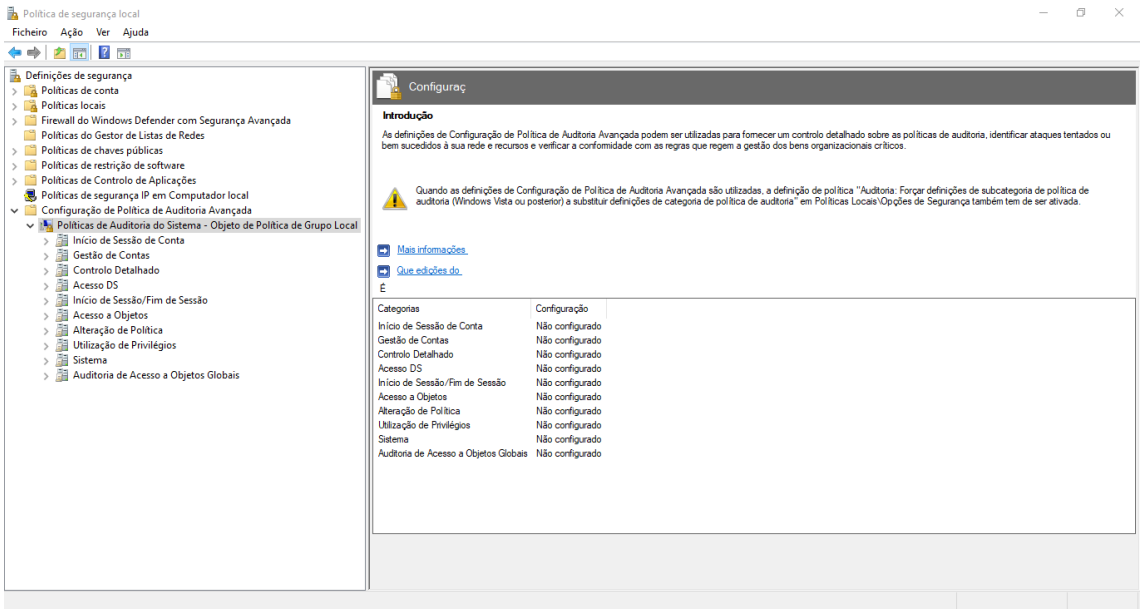


Figura 3.2 – Políticas de Auditoria do Sistema

Uma das maiores preocupações dos administradores é garantir que em ambientes com um elevado número de máquinas *MS Windows*, estas políticas são implementadas de forma homogénea. Para tal, caso estejamos num ambiente *MS Windows* com *Active Directory*, através da criação de *Group Policy* (GPO), é possível configurar as políticas de auditoria de forma centralizada, e aplicar a grupos, ou a todas as máquinas da organização de forma simples, garantindo uma configuração coerente e homogénea [19].

3.3 Processo de Recolha e Centralização de Eventos

A recolha e centralização de eventos é um processo que engloba a recolha de eventos de uma, ou mais, fontes de eventos e o seu armazenamento numa estrutura de

dados centralizada. Neste processo, um ou mais coletores de eventos deverão ser utilizados como repositório centralizado de todos os eventos gerados pelos computadores (fontes de eventos) a monitorizar.

Esta recolha pode ser feita através da instalação de agentes que correm nas máquinas fonte de eventos, permitindo o seu envio para os coletores, ou de forma menos intrusiva, sem a necessidade de instalação desses agentes. Sendo que o tipo de processo de recolha deverá ser escolhido com base na arquitetura da rede e capacidades específicas das ferramentas a utilizar em cada um destes processos.

3.3.1 Recolha de Eventos através da Instalação de Agentes

A abordagem tradicional para a recolha de eventos envolve a instalação de agentes em todos os sistemas onde é necessário recolher dados. Esta etapa de instalação pode ser executada manualmente para cada sistema, ou automatizada por meio de um servidor de instalação centralizado. Em qualquer dos casos, o custo de instalação, manutenção e atualização é tipicamente proporcional ao número de sistemas que requerem a instalação do agente [20].

No entanto, este tipo de abordagem pode fornecer benefícios, uma vez que o agente poderá ser configurado para monitorizar parâmetros que normalmente não aparecem no *log* de eventos, tais como o processamento ou a utilização de espaço em disco. Outro dos principais motivos para a utilização de um agente é quando existe a necessidade de tradução do formato nativo do *log* do sistema antes do seu envio para o sistema de destino [21].

A principal desvantagem deste tipo de instalação é a utilização de recursos do sistema onde o agente se encontra instalado [21].

3.3.2 Recolha de Eventos sem a Instalação de Agente (*Agentless*)

A abordagem *Agentless*, visa a recolha dos eventos sem a instalação de *software* adicional nos sistemas onde é necessário recolher dados, utilizando apenas os recursos já disponíveis nesse sistema [20].

O principal benefício desta abordagem é não ser necessário instalar, atualizar e manter *software* adicional em cada um dos sistemas onde é necessário efetuar a recolha de dados. No entanto, a capacidade de monitorização de parâmetros específicos que não

estejam contemplados nos *logs*, bem como a tradução do formato nativo do *log* poderá ser limitada, ou mesmo impossível de efetuar [21].

No caso dos sistemas operativos *MS Windows*, a Microsoft disponibiliza o *WinRm* em todas as versões a partir do sistema operativo *Windows Vista*, que é um serviço que visa permitir a gestão remota dos sistemas. Este serviço permite executar algumas das operações que normalmente apenas seriam passíveis de executar por um agente, tais como o envio de eventos iniciado pelo cliente, mas tem a grande vantagem de não ser necessária a instalação de *software* adicional.

3.4 Ferramentas da *Microsoft* para Recolha de Eventos

De forma a possibilitar uma gestão centralizada dos eventos das plataformas *MS Windows*, a *Microsoft* disponibiliza soluções baseadas em agentes, soluções que dispensam a necessidade de instalação de agentes, e soluções baseadas em *cloud*. Serão descritas nas próximas secções algumas das soluções disponibilizadas pela *Microsoft*, para permitir analisar aquela que mais de adapta a cada tipo de ambiente.

3.4.1 Windows Event Collection

O *Windows Event Collection* é uma solução própria da *Microsoft*, e encontra-se integrado no protocolo de gestão remota do *Windows* (*WS-Management protocol*). É compatível com os sistemas operativos a partir do *MS Windows XP SP2+* e *Windows Server 2003 SP1+*, sendo que a partir do *MS Windows Vista* e *MS Windows Server 2008*, os serviços necessários para a sua utilização encontram-se instalados por omissão em todos os sistemas operativos, não sendo necessário recorrer à instalação de agentes.

A utilização da recolha de eventos requer a configuração das máquinas que irão reencaminhar os eventos, bem como da máquina que irá receber os eventos (coletor). Esta funcionalidade está dependente do serviço de Gestão Remota do *Windows* (WinRM) e do serviço Recolector de Eventos do *Windows* (Wecsvc).

Como tal, os sistemas operativos da *Microsoft* permitem através da criação de subscrições (ver Figura 3.3), configurar o reencaminhamento de uma cópia dos seus eventos para um repositório centralizado (*Windows Event Collector*), onde estes podem ser visualizados, tratados e analisados em conjunto.

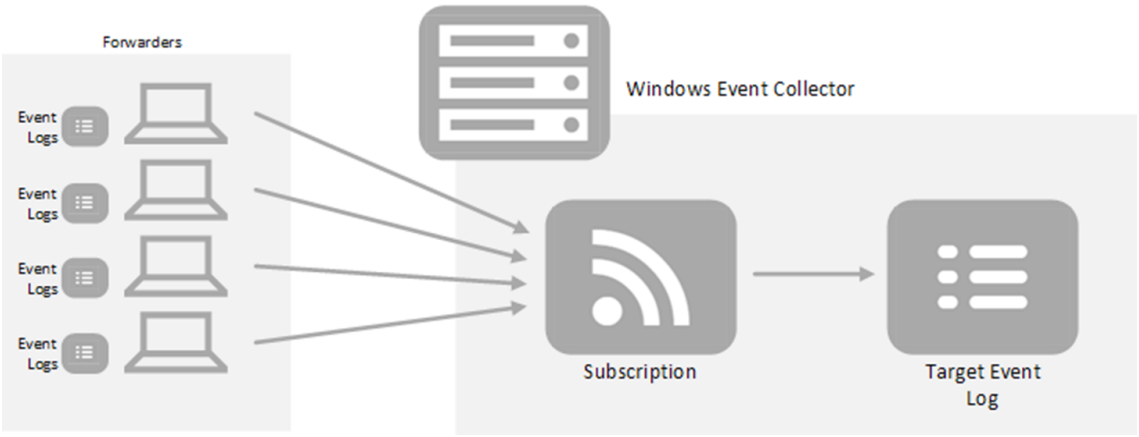


Figura 3.3 – Arquitetura do Ambiente "Windows Event Collection" (adaptado de [22])

As subscrições são criadas no coletor, e permitem especificar exatamente as máquinas (fontes de eventos) e os eventos que serão recolhidos, e em que registo serão armazenados no coletor. Assim que uma subscrição está ativa e os eventos estão a ser recolhidos, estes podem ser visualizados e tratados no coletor, tal como qualquer outro evento local. A *Microsoft* permite que sejam efetuados dois tipos de subscrições no WEC, sendo que estes não poderão ser utilizados em conjunto [23]:

- **Subscrições iniciadas pelo coletor:** As fontes de eventos têm de ser especificadas no coletor no instante da criação da subscrição, para que este as possa contactar para iniciar o envio de eventos. Este tipo de subscrição deverá ser utilizado quando se sabe quais as máquinas que irão reencaminhar eventos, o número de máquinas é pequeno, ou de forma a limitar as máquinas que têm permissão de escrita no coletor [24].
- **Subscrições iniciadas pela fonte:** Permite a criação de uma subscrição no coletor de eventos, sem a necessidade de especificar as fontes que irão reencaminhar os eventos. As fontes que irão reencaminhar os eventos poderão ser posteriormente especificadas de forma local, ou de forma centralizada, através de políticas de grupo (GPO). Este tipo de subscrição é útil quando não se quer especificar à partida todas as fontes de eventos no *Windows Event Collector*, ou quando as fontes de eventos estão inseridas num domínio em que existe uma grande rotatividade de máquinas a ser adicionadas e/ou retiradas [24].

As Subscrições podem ser aplicadas a máquinas específicas, mas também a grupos da *Active Directory*, como é possível ver na Figura 3.4. Por forma a reduzir o tráfego de rede gerado pelo reencaminhamento de eventos, deverão ser definidos filtros de eventos através do interface gráfico, ou através de *XML query* no caso de se pretender criar filtros mais avançados [22].

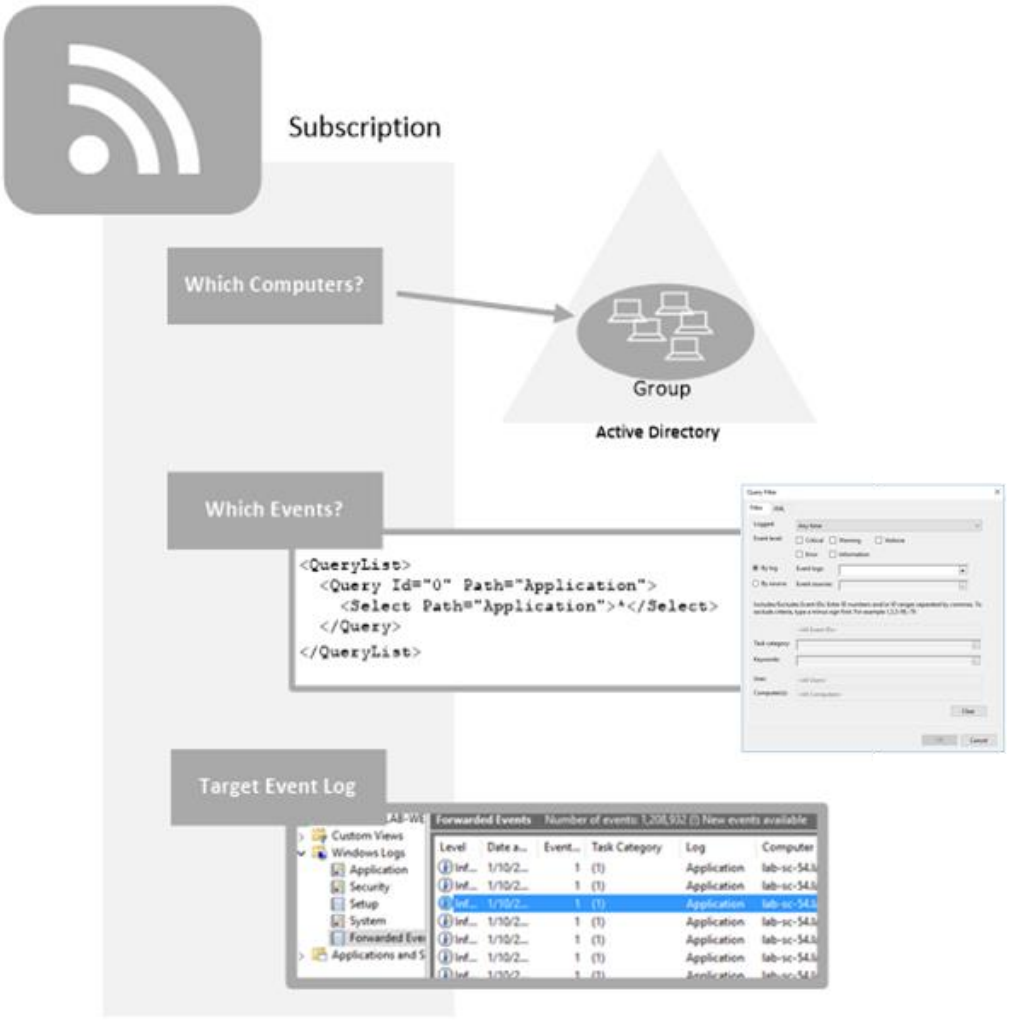


Figura 3.4 – Arquitetura Lógica de uma Subscrição (adaptado de [22])

No entanto, por um computador estar no grupo em que a Subscrição é aplicável não significa que comece de imediato a enviar eventos. Isso porque, adicionalmente é preciso configurar a conexão com o seu respetivo Coletor de Eventos. Para tal, é necessário ter privilégios de administração nas máquinas que serão configuradas como fontes de eventos, por forma a configurar e ativar a chave de registo "Computer Configuration \ Policies \ Administrative Templates \ Windows Components \ Event Forwarding \ Configure target Subscription Manager", o que poderá ser efetuado através de uma Política de Grupo (ver Figura 3.5) [22].

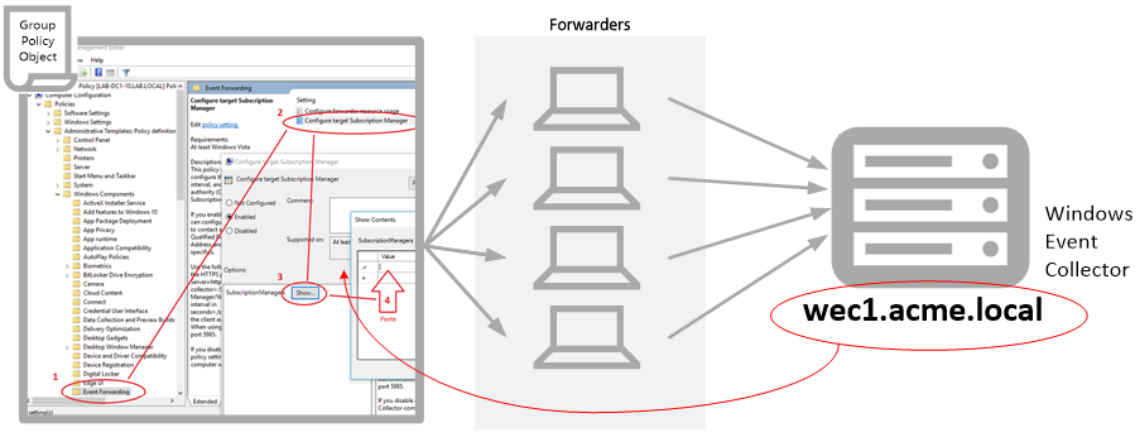


Figura 3.5 – Configuração do "Target Subscription Manager" (extraído de [22])

3.4.2 Microsoft System Center Operations Manager

O *Microsoft System Center Operations Manager* (SCOM) permite efetuar a monitorização de serviços, dispositivos e operações de forma centralizada, através da análise de eventos, tais como os eventos de segurança, de modo a identificar rapidamente potenciais problemas. A sua constituição base compreende um grupo de gestão que é composto por três componentes principais, o *management server*, a *operational database*, e a *reporting data warehouse database*, podendo estes coexistir num único servidor, ou serem distribuídos por vários servidores [25].

- ***Management Server:*** É o ponto central, e tem como função administrar a configuração do grupo de gestão, administrar e comunicar com os agentes para recolha dos dados e comunicar com as bases de dados.
- ***Operational Database:*** É uma base de dados *SQL Server* que contém os dados de configuração, e armazena todos os dados de monitorização recolhidos e processados. A *operational database* retém dados a curto prazo, por definição durante 7 dias.
- ***Reporting Data Warehouse Database:*** É uma base de dados *SQL Server*, onde os dados escritos na *operational database* são replicados, com o objetivo de serem armazenados a longo prazo, para efeitos de histórico e elaboração de relatórios.

Por forma a efetuar a monitorização, o Microsoft *System Center Operations Manager* requer a instalação de um agente nas fontes de eventos, para permitir efetuar a

recolha dos dados, comparação de amostras de dados com os valores predefinidos, criação de alertas e execução de respostas, de acordo com as configurações enviadas pelo *management server* [25].

O agente pode ser configurado para atuar como um agente de *proxy*, permitindo desta forma reencaminhar dados para o *management server* em nome de outro computador ou dispositivo, permitindo a monitorização de computadores e dispositivos nos quais não é possível instalar um agente [25].

Uma das principais vantagens do Microsoft *System Center Operations Manager* é a utilização de uma base de dados SQL Server, onde os eventos recebidos, após normalização, são armazenados de forma centralizada, permitindo desta forma efetuar consultas de forma rápida e simples.

3.4.3 **Serviço de Telemetria do *Windows***

O serviço de telemetria do *Windows* tem vindo a sofrer alterações ao longo dos últimos anos, sendo que o maior salto foi dado com o lançamento do sistema operativo *Windows 10* e *Windows Server 2016*. Nas versões anteriores do *Windows* e do *Windows Server*, a Microsoft usava a telemetria apenas para verificar se as assinaturas do *Windows Defender* se encontravam atualizadas, se as instalações do *Windows Update* tinham sido bem-sucedidas, e coletar informações de confiabilidade através do *Reliability Analysis Component* (RAC) e do *Customer Experience Improvement Program* (CEIP)[26].

No *Windows 10* e no *Windows Server 2016*, foi introduzida uma nova funcionalidade que permite controlar os dados de telemetria que se pretende enviar para a *cloud* da *Microsoft*, através de uma ligação *Secure Socket Layer* (SSL), podendo estes ser definidos através das Configurações da Privacidade, *Group Policy* (GPO), ou através do *Mobile Device Management* (MDM). A *Microsoft* tem vindo gradualmente a introduzir esta funcionalidade nas versões mais antigas do *Windows*, a partir do *Windows 7*, através de pacotes de atualizações. Desta forma, torna possível a recolha de dados técnicos vitais, sobre o desempenho dos dispositivos e *softwares* que estão a ser executados, de forma a manter os sistemas atualizados, seguros e com bom desempenho [26].

O serviço de telemetria é categorizado em quatro níveis cumulativos:

- ***Security:*** Informações necessárias para manter o *Windows* seguro, incluindo dados do *Connected User Experience and Telemetry*, *Malicious Software Removal Tool* (MSRT) e *Windows Defender*.
- ***Basic:*** Informações básicas do dispositivo, tal como os dados de qualidade, compatibilidade e utilização das aplicações. Inclui ainda todos os dados do nível *Security*.
- ***Enhanced:*** Informações adicionais sobre como o sistema operativo e as aplicações são usadas, dados de desempenho e dados de confiabilidade avançados. Inclui ainda os dados dos níveis *Basic* e *Security*.
- ***Full:*** Todos os dados necessários para identificar e ajudar a corrigir problemas, além de dados dos níveis *Enhanced*, *Basic* e *Security*.

Estes níveis de telemetria estão nativamente disponíveis em todas as edições de desktop e móvel do *Windows 10*, exceto o nível *Security*, que está limitado apenas ao *Windows 10 Enterprise*, *Windows 10 Education*, *Windows 10 Mobile Enterprise*, *Windows 10 IoT Core* (*IoT Core*) e *Windows Server 2016* [26].

Apesar de não estar publicamente disponível em que formato, nem de que forma é possível obter os dados, a *Microsoft* permite às organizações, através da configuração de um ID comercial único em todas as máquinas da organização, ter acesso aos eventos de telemetria recolhidos. A utilização deste serviço torna-se vantajosa, na medida em que a *Microsoft*, através do *Operations Management Suite* (OMS), disponibiliza ainda diversas soluções de produtividade baseada em *cloud* que permitem às organizações tirar partido desses dados, tal como o *Windows Analytics*, e contribuir para melhorar a eficiência operacional dos dispositivos *Windows* [27].

3.4.4 **Resumo**

As três soluções apresentadas na secção 3.4 são proprietárias da *Microsoft*, garantem compatibilidade com as plataformas a monitorizar, e permitem alcançar os objetivos propostos para o projeto. Para efetuar a implementação na infraestrutura da Altice Portugal, a solução escolhida é o *Windows Event Collection* (ver secção 3.4.1) com subscrição iniciada pela fonte, uma vez que dispensa a instalação de agentes, é

escalável, permite uma simples instalação e configuração de forma centralizada, e garante que os eventos recolhidos são tratados dentro da própria infraestrutura da organização.

3.5 Ferramenta de Gestão de Eventos *NXLog*

O *NXLog* é um *software* de gestão de eventos disponível para diversas plataformas, tais como *MS Windows* e GNU/Linux. Este *software* encontra-se disponível em duas versões: o *NXLog Community Edition*; e o *NXLog Enterprise Edition* [28].

O *NXLog Community Edition* é um *software open source* de gestão de eventos, de elevado desempenho, que permite a recolha de eventos reencaminhados em diversos formatos, incluindo suporte para protocolos específicos de diversas plataformas, tais como, *Windows Eventlog* (a partir do *MS Windows XP*), *Linux kernel logs*, *Android device logs* e *syslog* [28].

O *NXLog Enterprise Edition*, além das características presentes na versão *Community Edition*, contém, entre outras, melhorias ao nível do suporte de fontes de *logs* adicionais, como eventos *CheckPoint* LEA e SNMP, recursos de gestão e monitorização de agentes, recolha remota dos eventos do *Windows* e módulos de entrada e saída ODBC para leitura e/ou gravação de informação em bases de dados [28].

A *NXLog* disponibiliza ainda a aplicação *NXLog Manager*, que pode ser utilizada juntamente com o *NXLog Enterprise Edition*, permitindo efetuar a gestão e monitorização de forma centralizada das diversas instâncias *NXLog Enterprise Edition* através de uma consola de gestão web [28].

O conceito chave do *NXLog* é o de lidar com os eventos, preservando a sua estrutura, não sendo necessário converter todos os eventos para um protocolo comum, no entanto, caso se pretenda, também permite fazer a sua conversão para um outro protocolo. A arquitetura leve, modular e *multithread*, possibilita a esta ferramenta uma grande escalabilidade, e uma capacidade de processamento, filtragem e conversão de *logs* na ordem das centenas de milhar de eventos por segundo [28].

Este *software* possibilita a recolha e centralização dos eventos das plataformas *MS Windows*, mas obriga que seja instalada e configurada uma instância do *NXLog* em cada fonte de eventos. No entanto, uma vez que é possível configurar o envio dos eventos no

formato *syslog*, permite que estes sejam encaminhados para o SIEM *Alienvault USM* a partir do coletor de eventos, ou mesmo enviados diretamente das fontes de eventos, suprimindo desta forma a necessidade de utilização de um coletor de eventos.

3.6 Resumo

Efetuar a recolha e monitorização em tempo quase real dos eventos de segurança dos sistemas *MS Windows* é um recurso extremamente valioso como parte de um processo de monitorização de segurança mais amplo. As estações de trabalho e os utilizadores que as utilizam tornaram-se o alvo da atual guerra cibernética, pelo que torna-se essencial recolher e analisar os eventos produzidos por esses sistemas [29].

Este capítulo descreveu de forma genérica, com base nos fundamentos teóricos adquiridos, algumas das soluções disponíveis para implementar um processo consolidado de recolha, filtragem, tratamento e gestão centralizada de eventos dos sistemas *MS Windows*, e a sua integração num SIEM para uma melhor e mais rápida identificação dos eventuais incidentes que estejam a ocorrer na rede, tornando cada sistema *MS Windows* num sensor que irá contribuir com informação valiosa para melhorar o panorama global de monitorização de segurança.

Capítulo 4

Gestão e Monitorização do Ambiente WEC

Conforme apresentado no Capítulo 3, o *Windows Event Collection* é uma tecnologia de base dos sistemas operativos *MS Windows*, desenvolvida pela Microsoft, e constitui um núcleo flexível e eficiente de encaminhamento de eventos. No entanto, apresenta algumas lacunas ao nível da capacidade de gestão corporativa, monitorização de estado, relatórios de funcionamento, ou recursos que permitam uma simples escalabilidade do sistema, tal como o balanceamento de carga automático.

Neste capítulo é estudada a solução proposta pela *Logbinder* para efetuar a gestão e monitorização centralizada do processo de recolha, filtragem e tratamento dos eventos das plataformas com o sistema operativo *MS Windows*.

4.1 *Logbinder – Supercharger*

O *Supercharger* é uma plataforma de monitorização e gestão do ambiente *Windows Event Collection* desenvolvida pela *Logbinder*. Embora não seja requisito para a implementação de um ambiente WEC, o *Supercharger* veio colmatar as lacunas existentes, permitindo uma gestão simples e visibilidade instantânea do ambiente WEC [30].

4.1.1 **Arquitetura**

A arquitetura do *Supercharger* pressupõe um servidor designado de *Supercharger Manager* e um agente designado de *Supercharger Agent,* ou *Supercharger Controller*, que é instalado em cada um dos Coletores de Eventos do *Windows* (WEC) [31].

O *Supercharger Manager* suporta Coletores de vários domínios, não sendo necessário existir uma relação de confiança entre esses domínios. Em ambientes

menores, é possível executar o *Supercharger Manager* diretamente num WEC. No caso de apenas existir um Coletor de Eventos do *Windows* é possível instalar apenas o *Supercharger Manager* diretamente nesse Coletor, uma vez que este já inclui uma instância do serviço necessário para gerir as Subscrições locais [31].

A Figura 4.1 apresenta uma arquitetura que o *Supercharger* permite implementar, composta por três domínios diferentes. Um dos domínios é composto por dois Coletores, enquanto os outros domínios possuem apenas um Coletor cada. O *Supercharger Controller* (ou *Supercharger Agent*) encontra-se instalado nos quatro Coletores de Eventos do *Windows*, e um quinto computador tem configurado o *Supercharger Manager* para efetuar a gestão e monitorização dos WEC [31].

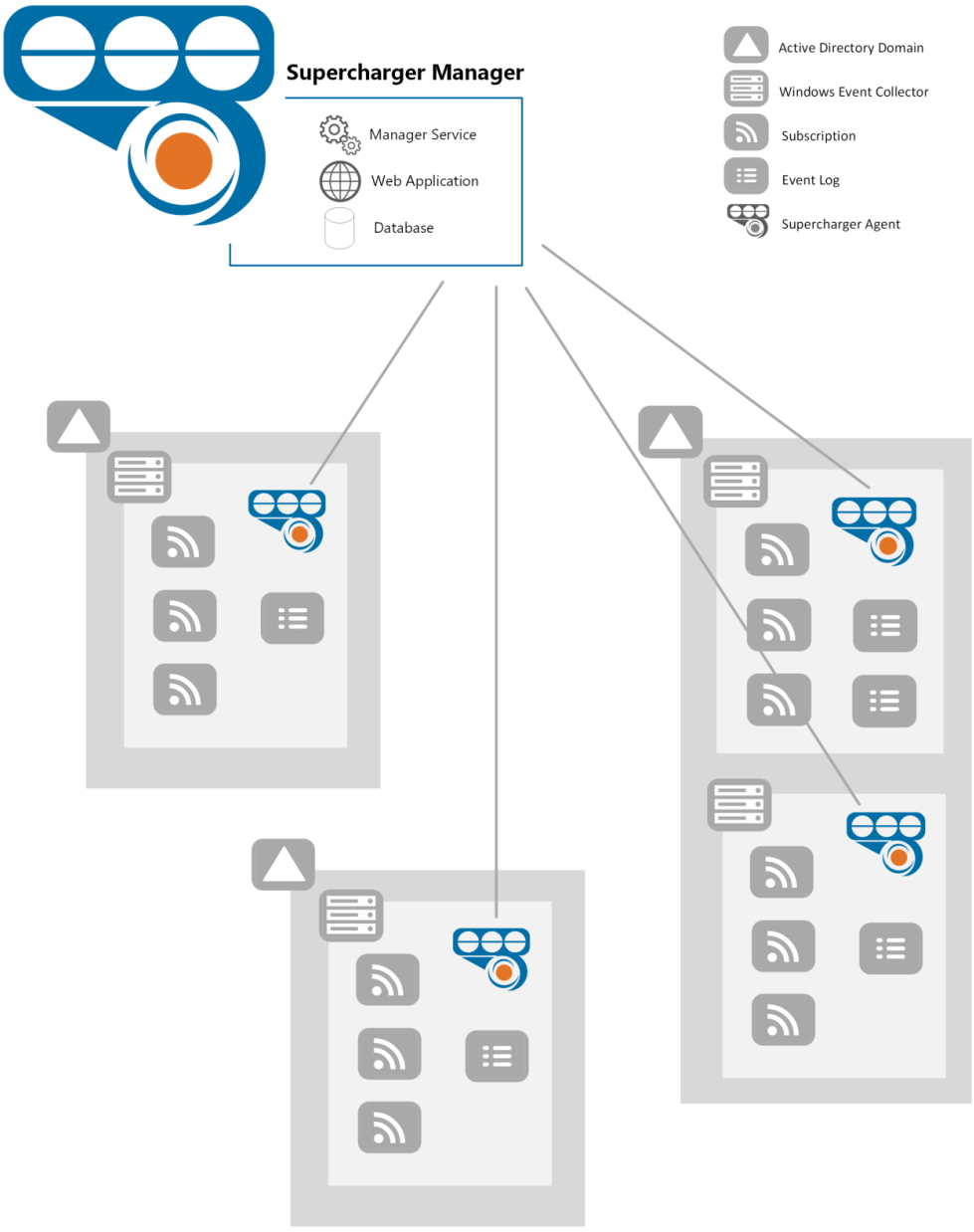


Figura 4.1 – Arquitetura Física do Supercharger (extraído de [31])

A arquitetura lógica do *Supercharger* (ver Figura 4.2) baseia-se nas Subscrições do WEC e nos servidores *Windows* que as hospedam. Os objetos que a constituem incluem as Políticas do Coletor, as Políticas de Subscrição e os Filtros. Uma das suas grandes vantagens é permitir distribuir a carga pelos diversos coletores de eventos pertencentes a uma determinada subscrição de forma automática, através de uma funcionalidade que permite a criação de Subscrições Distribuídas [32].

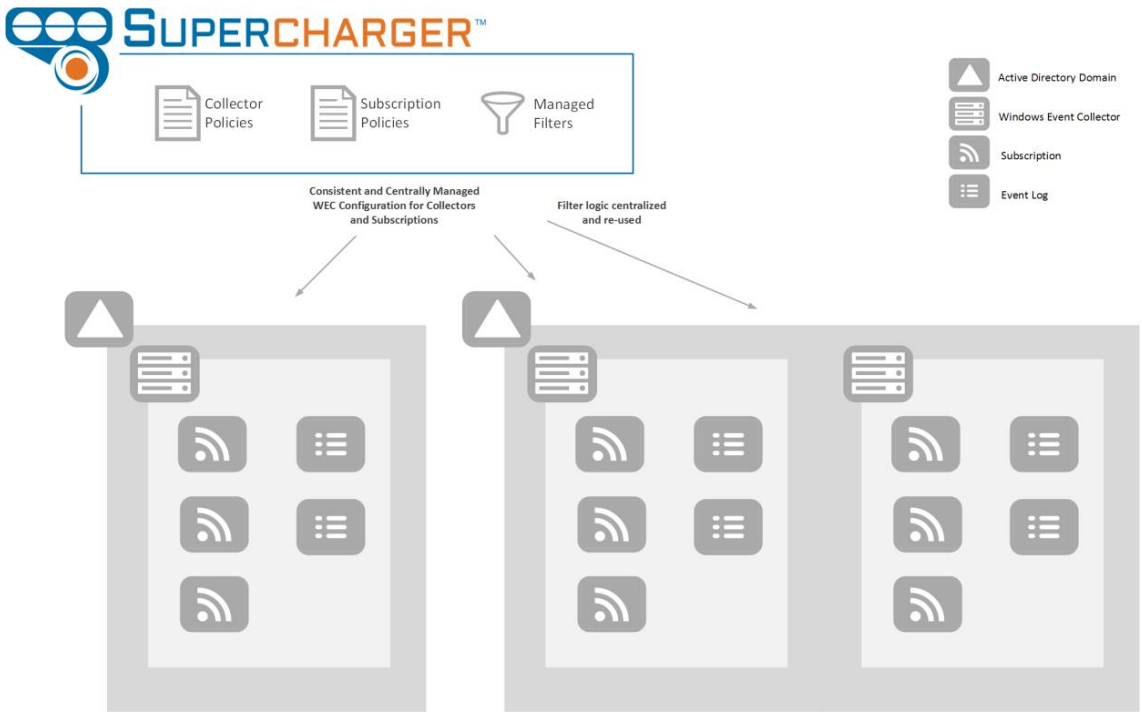


Figura 4.2 – Arquitetura Lógica do Supercharger (extraído de [32])

- **Políticas do Coletor:** Definem as configurações dos WEC. Permite atribuir a mesma Política a vários Coletores, de forma a garantir que estão todos configurados de forma coerente. O *Supercharger* disponibiliza uma Política de Coletor padrão, sendo possível definir Políticas personalizadas adicionais.
- **Políticas de Subscrição:** São semelhantes às Políticas de Coletor, mas são aplicadas ao nível da subscrição. O *Supercharger* e o WEC têm várias opções de configuração que determinam a forma como as subscrições funcionam no ambiente WEC, e a forma como podem ser controladas pelo *Supercharger*. Ao criar uma Política de Subscrição no *Supercharger*, esta poderá ser aplicada a várias subscrições, de forma a garantir que são todas configuradas de forma coerente.

- **Filtros (*Managed Filters*):** Uma das características do WEC é a sua capacidade para criar filtros avançados que definem exatamente quais os eventos que deverão ser reencaminhados pelas fontes de eventos, permitindo reduzir o tráfego de dados na rede. Depois de criado um filtro no *Supercharger Manager*, este pode ser atribuído a várias subscrições.

Supercharger Manager

O *Supercharger Manager* é composto por três componentes:

- ***Web Application:*** Disponibiliza uma interface web através da qual é possível interagir com o *Supercharger* para gerir o ambiente WEC [31].
- ***Manager Service:*** Permite efetuar uma gestão centralizada da instalação do *Supercharger*, bem como a gestão de qualquer subscrição local, no caso da máquina onde se encontra instalado o *Supercharger Manager* também ser um Coletor (executa a função do *Supercharger Agent*) [31].
- **Base de dados SQL Express:** Instalada e gerida pelo *Supercharger* para armazenamento de configurações e dados aplicacionais [31].

Estes permitem gerir os coletores de forma centralizada, e monitorizar o estado de cada objeto, fornecendo visibilidade instantânea do estado do ambiente WEC. No caso do *Supercharger Manager* detetar que o *Supercharger Controller* de algum WEC que se encontra a ser gerido não está a reportar informação, ele irá alertar essa alteração do estado, que se refletirá na cor do respetivo objeto que é apresentado no interface web. Ao abrir a respetiva caixa de diálogo do visualizador é possível saber a razão pela qual o objeto se encontra nesse estado [33]. As suas principais funcionalidades são [34]:

- Visualização centralizada do estado de todos os Coletores e Fontes de Eventos.
- Gestão centralizada de todos os Coletores de Eventos.
- Alerta quando o estado de qualquer Subscrição não cumpre com os requisitos da sua Política de Subscrição. Este alerta pode ser visualizado através da alteração da cor do estado do objeto no interface web, através da criação de um evento representativo dessa alteração, ou através do envio de um e-mail.

- Equilibrar a carga de um número elevado de fontes de eventos por vários coletores, através de Subscrições Distribuídas.
- Correlação das fontes de eventos presentes no WEC com a informação dos computadores e grupos da *Active Directory* [35]:
    - Determinista: Enumera cada grupo da AD associado à assinatura, e compara com o número de fontes de eventos que estão a reportar atualmente ao WEC - Tem em consideração o estado do computador na AD.
    - Empírico: Compara a quantidade de fontes de eventos ativas, com a quantidade de fontes existentes no passado.
    - Arbitrário: Compara o número de fontes de eventos especificada na subscrição, com o número de fontes de eventos que se encontram atualmente a reportar.
- Exclui fontes de eventos (WEF) desatualizadas.
- Construção de filtros (*Managed Filters*) que limitam o ruído na fonte.
- Aplicar políticas de configuração WEC coerentes em coletores e subscrições.
- Análise de desempenho dos coletores, incluindo a carga do CPU e o número de eventos recebidos por segundo, para planeamento da escalabilidade do sistema.

Supercharger Controller (Supercharger Agent)

O *Supercharger Controller* (*aka agent*) é executado como um serviço no WEC, e tem como função fornecer os dados de estado deste ao *Supercharger Manager* e processar os comandos enviados automaticamente pelo *Supercharger Manager*, ou enviados através da interface web. Algumas das principais funcionalidades do *Supercharger Controller* são [33]:

- Criação, eliminação e modificação de Subscrições, conforme instrução do *Supercharger Manager*.
- Análise periódica das Subscrições locais do WEC.
- Análise de desempenho, incluindo a carga do CPU e o número de eventos recebidos por segundo.

- Monitorização e administração do estado dos serviços críticos, incluindo o WinRM e o *Windows Event Collection*.
- Garante a aplicação da configuração das Políticas de Subscrição nas Subscrições atribuídas.
- Exclui fontes de eventos (WEF) desatualizadas.
- Executa consultas à *Active Directory* em nome do *Supercharger Manager*.

O *Supercharger Controller* comunica com o *Supercharger Manager* estritamente via SQL (TCP 1433), utilizando a funcionalidade *SQL Service Broker*. A única exceção é quando o *Supercharger Controller* descobre que o *Supercharger Manager* foi atualizado. Nesta situação, este comunica via HTTP para descarregar a versão mais recente do instalador do controlador, de forma a efetuar a atualização automática. Se a atualização automática falhar por algum motivo, é possível proceder-se à atualização manual do controlador, seguindo o mesmo procedimento de uma nova instalação.

Além das instruções enviadas automaticamente pelo *Supercharger Manager* para o *Supercharger Controller*, também é possível enviar instruções através do interface web para serem executados comandos a pedido no WEC, tais como [33]:

- Análise do estado das fontes de eventos (WEF)
- Reiniciar o serviço do *Supercharger Controller*
- Reiniciar o sistema operativo do Coletor
- Limpeza do registo de fontes de eventos (WEF) antigas no WEC

4.1.2 **Funcionalidades**

O *Supercharger* apresenta diversas funcionalidades que visam melhorar a capacidade, e aumentar a simplicidade de gestão e configuração do ambiente WEC, tais como:

Criação de Subscrições

Através do Visualizador de Subscrições do *Supercharger* é possível configurar os atributos da subscrição no WEC, bem como os atributos adicionais que permitirão ao *Supercharger* fornecer a informação de alarmística pretendida. Desta forma, dá-nos um controlo centralizado dos coletores de eventos (WEC) a partir de qualquer ponto,

através da interface web, deixando de ser necessário aceder aos coletores para configurar as respetivas subscrições [36].

Com o objetivo de permitir uma instalação graciosa em ambientes onde já existem coletores de eventos configurados, e a adição, modificação, ou exclusão manual de subscrições nos coletores de eventos que se encontram a ser geridos pelo *Supercharger*, o *Supercharger Controller* procura por quaisquer subscrições existentes no coletor de eventos (em intervalos de cerca de cinco minutos) e cria os seus registos associados no *Supercharger Manager*. Esta análise pode ainda ser feita a pedido, através do comando *CollectorAnalysisCommand* [36].

Quando uma subscrição é alterada no *Supercharger Manager*, é enviada a respetiva informação ao *Supercharger Controller* que se encontra instalado nesse coletor, que será responsável pela aplicação das novas configurações no WEC de forma imediata [36].

Subscrições Distribuídas / Loadbalancer

Em redes com milhares de computadores, e/ou centenas de servidores, apenas um WEC poderá não ser suficiente para suportar o elevado número de eventos gerados pelas fontes, desta forma, o *Supercharger* na versão *Enterprise* permite distribuir e equilibrar a carga automaticamente entre vários coletores de eventos com o recurso às Subscrições Distribuídas [37].

Uma subscrição distribuída atua como uma subscrição normal ao nível do WEC, mas no *Supercharger* esta é criada ao nível do domínio. Após estar criada, terão de ser atribuídos 2 ou mais coletores desse domínio à Subscrição Distribuída. Em seguida, são especificadas as fontes de eventos que serão distribuídas entre esses coletores. Isso é chamado de *Forwarder Superset* [37].

O *Forwarder Superset* pode ser especificado selecionando um grupo da *Active Directory* que possui todas as fontes, ou em vez disso, especificar um filtro LDAP (*Lightweight Directory Access Protocol*) personalizado. Terá também de ser definida uma Política de Subscrição e um *"Managed Filter"* para a Subscrição Distribuída, de forma a permitir ao *Supercharger* criar uma subscrição WEC em cada um dos Coletores atribuídos à Subscrição Distribuída [37].

Para permitir que o *Supercharger* distribua as fontes de eventos pelos coletores é necessário criar uma Unidade Organizacional (OU) na *Active Directory* (AD) e delegar

no *Supercharger* autoridade para criar e gerir grupos nessa OU. Uma vez que o *Supercharger* necessita de criar um grupo para cada Coletor atribuído à Subscrição Distribuída (ver Figura 4.3) para repartir os WEF existentes no *Forwarder Superset* entre esses grupos [37].

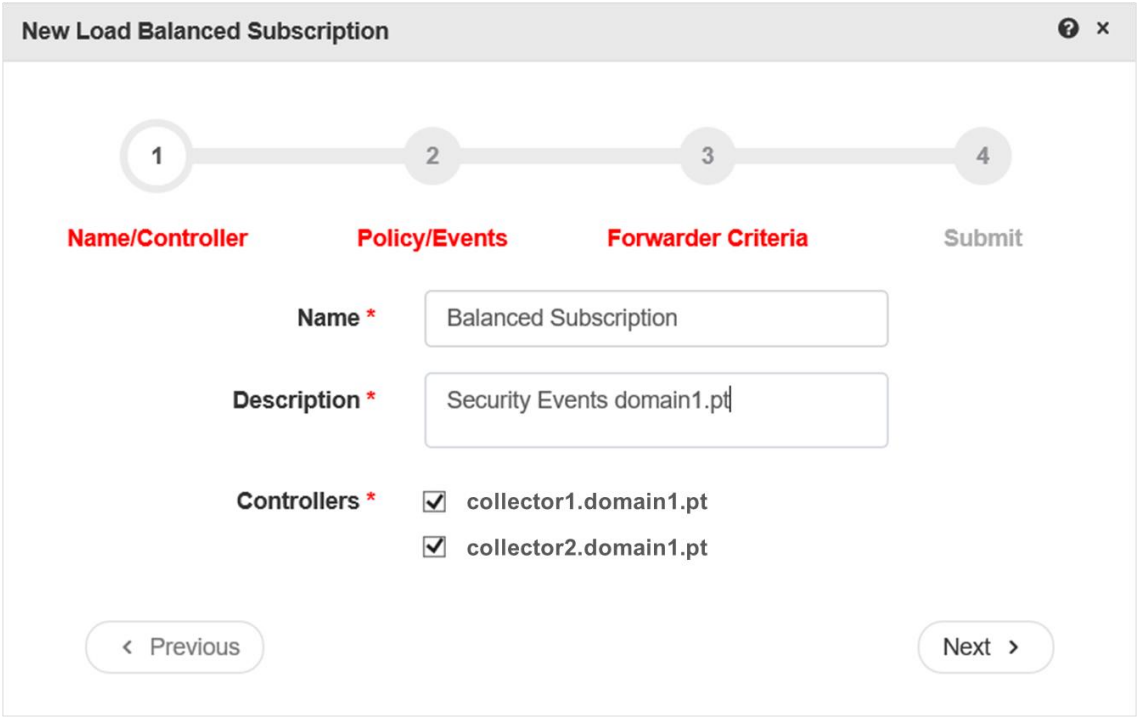


Figura 4.3 – Criação de uma Subscrição Distribuída

Políticas de Subscrição

As Políticas de Subscrição permitem assegurar uma configuração coerente das configurações do WEC entre várias Subscrições (mesmo em coletores e domínios diferentes) semelhante à forma como as Políticas de Grupo funcionam em ambientes *Windows* [36].

A maioria das configurações das subscrições do WEC não são diretamente acessíveis quando se cria ou edita uma subscrição, uma vez que são geridas pela Política de Subscrição atribuída a essa subscrição. Sendo que as Políticas de Subscrição existentes podem ser editadas, ou criadas novas Políticas, de forma a satisfazer as necessidades de cada ambiente WEC. Por omissão, o *Supercharger* tem disponível duas Políticas de Subscrição [36]:

- ***Discovered Subscriptions Policy:*** Política apenas de leitura, atribuída automaticamente sempre que o *Supercharger* descobre uma subscrição que não foi criada por este. Nesta política a opção *"Enforce WEC settings"* estará sempre desmarcada, o que significa que o *Supercharger* não fará qualquer modificação baseada em Políticas de Subscrição. Outras configurações, como *"Pruning Old WEC Sources"* também estão desabilitadas, e o *"Health Assessment Basis"* utilizado é o determinístico. Basicamente, as subscrições descobertas são tratadas como *hands-off* pelo *Supercharger* até ser atribuída manualmente uma nova Política de Subscrição pelo administrador [36].
- ***Default Subscription Policy:*** Esta é a Política de Subscrição padrão do sistema. A criação de novas políticas customizadas começa com as configurações desta Política. Por norma, as configurações padrão são as mostradas na Figura 4.4 [36]:

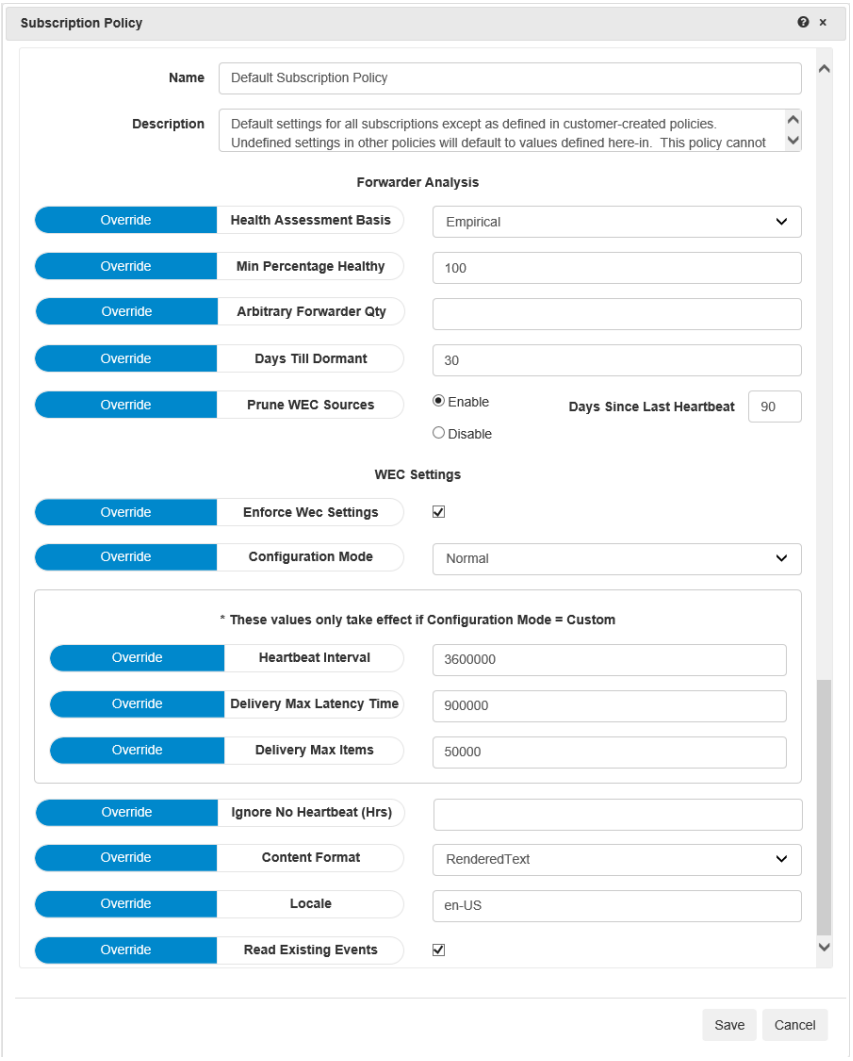


Figura 4.4 – Política de Subscrição Padrão

Criação de Filtros (Managed Filters)

Através da criação de filtros personalizados, e da utilização de alguns filtros pré-existentes no *Supercharger*, é possível de forma simples reduzir o ruído de eventos reencaminhados pelas fontes. Estes filtros vão ser aplicados nas fontes que geram os eventos, de forma a selecionar quais os registos de eventos, e quais os eventos desses *logs* que devem ser encaminhados para o coletor, permitindo maximizar a eficiência da relação sinal-ruído [36].

Os filtros personalizados podem ser criados através de *XML XPath query* (ver Figura 4.5), ou através do *Managed Filter*.

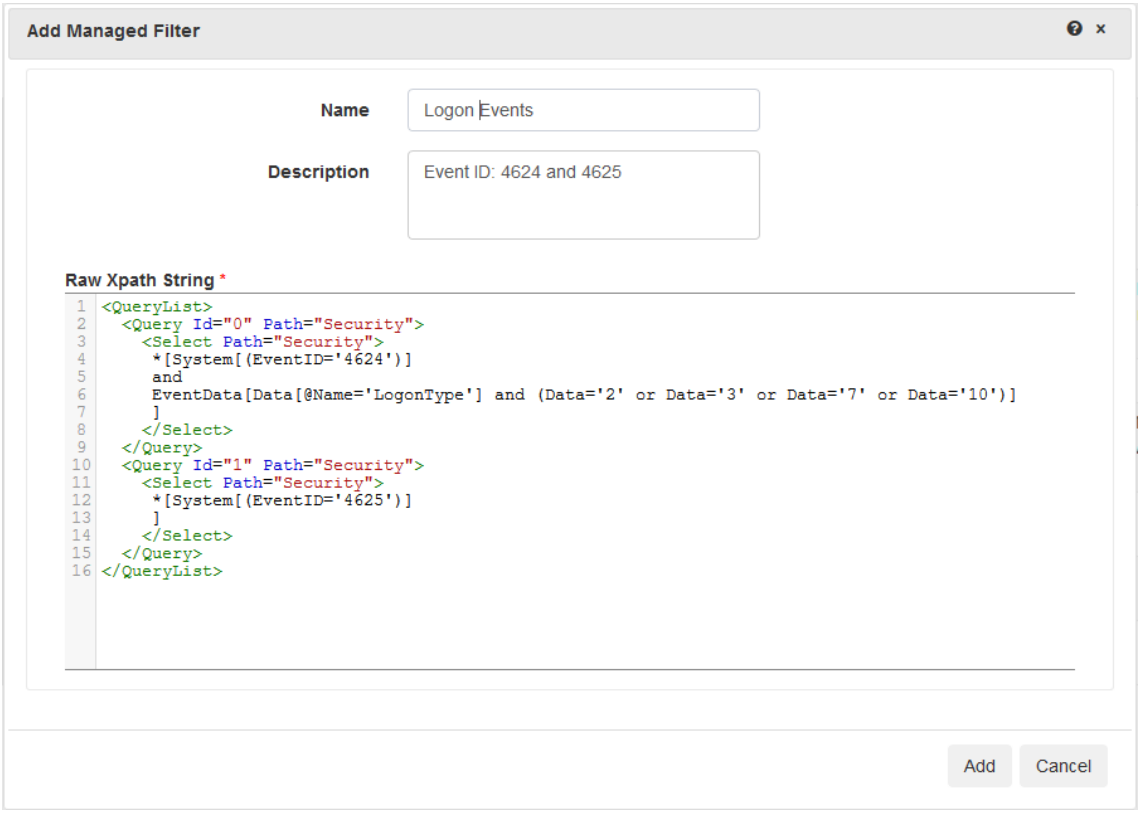


Figura 4.5 – Editor de Filtros Através de "XML XPath Query"

De forma a não exigir o conhecimento da sintaxe *XML query*, bem como da estrutura dos eventos do *Windows* para a criação de filtros através de *XML XPath query*, o *Supercharger* disponibiliza o interface gráfico do *Managed Filter* (ver Figura 4.6) que facilita a criação de filtros relativos ao *Log de Segurança do Windows*.

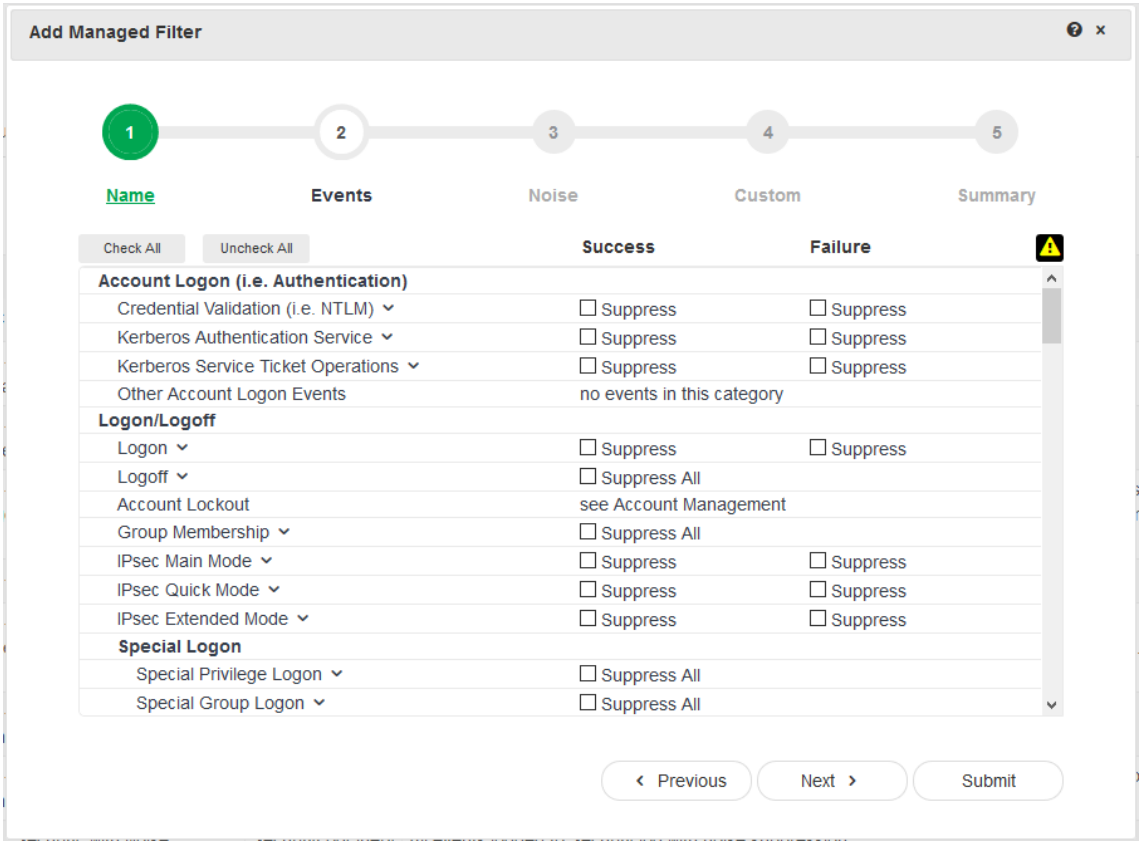


Figura 4.6 – Editor de Filtros "Managed Filter"

Depois de criado um filtro no *Supercharger*, este pode ser atribuído a várias Subscrições. As subscrições existentes nos WEC, que sejam descobertas pelo *Supercharger* (que não tenham sido criadas pelo *Supercharger*) exibem o filtro através do *XML XPath query* previamente definido na subscrição [36].

Limpeza do registo de WEF's antigos no WEC

O *Windows Event Collector* (WEC) cria uma chave de registo e várias subchaves para cada computador de origem que envia eventos para uma determinada subscrição. No entanto, o WEC nunca apaga esses objetos de registo, mesmo depois das fontes de eventos deixarem de ser válidas, podendo originar dois problemas [36]:

- Nos ambientes em que haja uma grande renovação de fontes de eventos (WEF), iremos observar um aumento excessivo do número de registos de fontes de eventos antigas, o que degrada o desempenho do WEC e torna o *Event Viewer* irresponsivo [36].

- O *Supercharger* não determina com precisão o estado das subscrições quando é selecionado o modo Empírico. O modo de avaliação Empírico, utiliza o número de fontes de eventos existentes no registo para avaliar o estado atual da subscrição, pelo que se existir no registo um elevado número de fontes de eventos desatualizadas, o *Supercharger* irá apresentar a subscrição com um estado inferior ao que realmente se encontra [36].

Para resolver estes problemas, o *Supercharger* dispõe de uma funcionalidade opcional, que pode ser ativada ou desativada através dos *Subscription Policy objects*, e permite eliminar de forma automática as fontes de eventos antigas através da exclusão da sua chave de registo existente no WEC. Se essa fonte de eventos voltar a ser ativada no futuro, o WEC irá automaticamente recriar a chave [36].

Análise de Estado e Alarmística

O *Supercharger* determina o estado de todos os objetos existentes no seu ambiente WEC, permitindo visualizar não apenas um nível macro, mas também consegue alcançar o pormenor de visualização do estado de cada uma das fontes de eventos. A visualização macro do ambiente WEC é obtida através da cor dos ícones existentes no *dashboard*, sendo que estes podem assumir quatro cores (Cinzento – Sem informação disponível; Vermelho – Erro; Amarelo – Alerta; Verde – Sem problemas) [36].

A cada Subscrição do Coletor de eventos é necessário associar uma Política de Subscrição que servirá como base de avaliação do estado desse ambiente WEC, e define os critérios de avaliação dessa Subscrição. O principal objetivo deverá ser manter o ambiente WEC com todos os objetos a aparecer com a cor verde. Para tal, o *Supercharger* permite criar Políticas de Subscrição personalizadas, para que a alarmística seja a mais adequada para cada tipo de ambiente WEC a monitorizar [36].

A alarmística do *Supercharger* funciona aproximadamente em tempo real para mudanças de estado de qualquer objeto, desde o nível das Subscrições até ao nível dos domínios. No entanto, não é necessário aceder ao *dashboard* para obter a informação do estado do ambiente WEC, esta informação poderá ser obtida através dos eventos gerados pelo *Supercharger Manager* (ver Tabela 4.1), que verifica as alterações de estado em todos os coletores. No caso da cor do estado de um objeto alterar, o *Supercharger* cria um alerta no registo de Aplicações local [36].

Tabela 4.1 – ID's dos Eventos de Alarmística Gerados Pelo Supercharger (adaptado de [33])

	Cor do Novo Estado e Severidade do Evento			
	Informação		Alerta	Erro
	Cinzento	Verde	Amarelo	Vermelho
Domínio	103	102	101	100
Coletor	203	202	201	200
Subscrição	303	302	301	300
Subscrição Distribuída	403	402	401	400

Outra forma de receber a informação da alarmística do *Supercharger* é através de um Relatório enviado por SMTP (por padrão, diariamente, às 7h) que lista cada domínio, subscrição distribuída, coletor e subscrição, indicando o seu estado, cor, e as razões porque se encontra nesse estado [36].

Monitorização de Desempenho dos Coletores

O *Supercharger* monitoriza a utilização do processador (CPU) de cada coletor de eventos em tempo real e o número de eventos recebidos por segundo em cada log. Disponibiliza ainda a média e o pico de eventos dos últimos 7 dias para serem exibidos junto ao respetivo objeto (ver Figura 4.7) [36].

Figura 4.7 – Monitorização de Desempenho dos Coletores (extraído de [36])

4.2 Resumo

Este capítulo descreveu a arquitetura, funcionalidades e capacidades da plataforma *Supercharger* da *Logbinder*, que permite efetuar a gestão e monitorização centralizada do ambiente *Windows Event Collection*.

Capítulo 5

Análise de Requisitos e Arquitetura

"No problem can be solved until it is reduced to some simple form. The changing of a vague difficulty into a specific, concrete form is a very essential element in thinking."

(John Pierpont Morgan)

Neste capítulo são especificados os requisitos e a arquitetura necessários para a implementação de um sistema que permita colmatar a atual lacuna da falta de visibilidade e alarmística adequada ao nível da deteção de incidentes de segurança relacionados com os sistemas *MS Windows* da Altice Portugal.

5.1 Definição do Problema

O problema exposto pela Altice Portugal para a realização deste projeto tem por base a atual lacuna ao nível da capacidade de análise e deteção de forma centralizada e automatizada dos eventos de segurança relativos às estações de trabalho *MS Windows*. Por forma a dotar o seu CSOC (*Cyber Security Operations Center*) desta capacidade foram apresentando os seguintes objetivos:

- Desenvolver um processo técnico robusto e eficiente, que faça chegar, de forma adequadamente filtrada ao SIEM, dados de todos os eventos relevantes de segurança, da totalidade dos sistemas *MS Windows* a operar dentro de uma rede corporativa e infraestruturas de *data center*.
- Desenvolver regras de correlação que permitam ao SIEM detetar, de forma efetiva, anomalias relevantes a serem endereçadas ao CSOC.

5.2 Análise de Requisitos

A disponibilização de eventos por parte dos sistemas *MS Windows*, e a sua utilização ativa para monitorização das atividades maliciosas em tempo real são dois conceitos totalmente distintos. Os eventos de segurança são uma fonte essencial de informação durante uma investigação forense após uma violação de segurança, no entanto, por si só não permitem a deteção ativa de atividades maliciosas. De forma a poderem ser úteis para a defesa ativa da rede, através da deteção de ataques e implementação das contramedidas adequadas em tempo útil, estes deverão ser monitorizados e analisados o mais próximo de tempo real possível, sendo que isto torna-se cada vez mais importante, quanto cada vez mais os ataques se tornam mais sofisticados [12].

Para proceder à definição dos requisitos para implementação do processo de recolha de eventos do *Windows*, é necessário ter em conta uma correta estratégia de monitorização dos eventos do *Windows*, bem como definir corretamente as atividades de interesse a monitorizar.

5.2.1 Estratégia de Monitorização dos Eventos *Windows*

Uma vez que não existe uma estratégia de monitorização de *logs* única que se adeque a todas as organizações, é necessário perceber como deverá ser implementado o processo de monitorização, e qual a infraestrutura de suporte necessária, de forma a garantir o seu sucesso. Para tal, existem algumas perguntas que deverão ser feitas na fase de desenho da estratégia, por forma a determinar os requisitos e garantir uma correta implementação [12]:

- O que é necessário monitorizar (a um nível macro)?
- Quais os sistemas e os componentes que deverão ser monitorizados?
- Que informações os sistemas devem registar nos *logs* de segurança?
- Qual será o processo de captura e análise dos *logs* de segurança?
- Com que frequência os dados dos *logs* de segurança devem ser analisados?
- Por quanto tempo é necessário manter os *logs* de segurança?

De forma a responder a essas questões, deverão ser consideradas em primeiro lugar as leis e regulamentos aplicáveis, e de seguida as políticas e o apetite ao risco da

organização, a fim de garantir que a estratégia de monitorização e gestão atenda a todas as necessidades da organização. Além disso, é necessário compreender as capacidades técnicas dos sistemas a monitorizar, e das tecnologias de suporte disponíveis que permitam implementar o processo de captura e análise [12].

5.2.2 Definição a Alto Nível das Atividades de Interesse a Monitorizar

O ponto de partida para qualquer organização que procure estabelecer ou melhorar os seus processos de monitorização de eventos é definir a alto nível os tipos de atividades que pretende monitorizar como possíveis indicadores de comportamento potencialmente malicioso ou anómalo [12].

Uma vez que não existe uma linguagem universal para a representação de eventos, é necessário na fase de planeamento definir os requisitos das atividades de interesse a alto nível, de forma a permitir que exista flexibilidade suficiente para que no momento da implementação seja feita a seleção de um conjunto de eventos (mensagens ou alertas) específicos dos sistemas, que serão associados a uma atividade de alto nível [12].

Exemplos de atividades de interesse de alto nível:

- Deteção de Ataques de *Bruteforce*
- Deteção de Uso Indevido de Credenciais
- Deteção de Escalação de Privilégios
- Deteção de Ataques *Denial-of-service* (DoS) a Contas de Utilizador

A definição a alto nível das atividades de interesse a monitorizar deverá garantir o cumprimento das políticas de segurança e o apetite ao risco da organização.

5.3 Requisitos Não-Funcionais

Com o objetivo de garantir os requisitos de integração dos eventos de segurança dos sistemas *MS Windows* no SIEM *Alienvault USM*, uma eficiente monitorização e gestão desse processo, e a capacidade de retenção de dados por um período considerável para eventual análise forense, é necessário implementar uma arquitetura robusta, redundante, e capaz de suportar o normal funcionamento da infraestrutura existente,

mas que sobretudo tenha capacidade sobrante suficiente para suportar períodos críticos em que seja gerada uma elevada quantidade de eventos por segundo. Como tal, são definidos os seguintes requisitos não-funcionais para o sistema:

- **Compatibilidade:** Deverá ser compatível com a infraestrutura existente, e ser o menos intrusivo possível para as fontes de eventos, devendo privilegiar a não instalação de agentes nas máquinas.
- **Capacidade:** Suportar o normal funcionamento da infraestrutura existente, e garantir capacidade sobrante suficiente para suportar períodos críticos em que seja gerada uma elevada quantidade de eventos por segundo.
- **Escalabilidade:** O sistema a implementar deverá permitir uma fácil escalabilidade horizontal, tendo em conta a atual dimensão e distribuição geográfica da organização.
- **Segurança:** Sendo uma solução que visa contribuir para a monitorização do estado da segurança dos sistemas existentes na rede, deve ser implementada tendo por base as melhores práticas ao nível de segurança, permitindo garantir a disponibilidade, confidencialidade e integridade da informação tratada.
- **Desempenho:** Os eventos devem chegar ao SIEM em tempo útil para análise e deteção de anomalias.
- **Robustez:** O sistema deverá ser desenhado de forma a resistir à falha de alguns dos seus componentes.
- **Disponibilidade:** O sistema deverá manter-se operacional, tanto quanto necessário, para a deteção efetiva de eventos relevantes.

5.4 Requisitos Funcionais

Os requisitos funcionais descrevem as condições necessárias para atingir os objetivos propostos, permitindo alcançar o sucesso na realização do projeto. Assim, foram definidos os seguintes requisitos funcionais:

5.4.1 Espectro de Máquinas

Este projeto destina-se a ser aplicado à totalidade dos sistemas *MS Windows* a operar dentro da rede corporativa e infraestruturas de *data center* da Altice Portugal. Atualmente, esta dispõe de mais de quinze mil estações de trabalho com sistema operativo *MS Windows* geograficamente distribuídas, e de mais de oito mil servidores com sistema operativo *MS Windows* que se encontram distribuídos em dois grandes polos, um localizado em Lisboa, e outro na Covilhã.

Por motivos de priorização e complexidade de instalação, foi definido pela empresa que o projeto será implementado numa primeira fase apenas na infraestrutura de servidores e estações de trabalho geridos pela Direção de Informação e Tecnologia (DIT), que contempla a totalidade das estações de trabalho existentes na Altice Portugal e cerca de seis mil servidores, e será estendido aos restantes cerca de dois mil servidores que se encontram na infraestrutura gerida pela Direção de Operações B2B (DBO) apenas no decorrer do ano de 2019.

5.4.2 Condições para Geração de Alertas no SIEM

Para permitir a geração de alarmes relevantes a serem endereçados ao CSOC, os eventos de segurança gerados pelos sistemas *MS Windows*, terão de ser encaminhados a um SIEM para análise, correlação e geração de alertas em tempo real.

Este deve ser configurado de forma a detetar situações anómalas a nível de segurança das plataformas *MS Windows*, tendo como objetivo alertar todos os incidentes críticos que estejam a ocorrer em tempo quase real. As atividades de interesse a monitorizar no âmbito deste projeto são:

- Deteção de Ataques de *Bruteforce*
- Deteção de Uso Indevido de Credenciais (através de movimentos laterais)
- Deteção de Ataques *Denial-of-service* (DoS) a Contas de Utilizador

Para deteção destas atividades, deverá ser adotada uma estratégia que permita minimizar a ocorrência de falsos positivos, uma vez que um número excessivo de alertas a relatar anomalias não relevantes e a existência de falsos positivos geram "ruído", o que torna difícil (ou mesmo impossível) determinar quais os alertas verdadeiramente maliciosos, fazendo com que o sistema de alertas se torne inútil.

Para reduzir o número de falsos positivos produzidos pelo SIEM, sem que isso resulte na existência de falsos negativos, deverá ser analisada a atividade normal da rede, por forma a ser possível definir criteriosamente os limites para as regras de correlação de eventos. Devendo ser gerados alertas sempre que esses limites sejam ultrapassados.

Para efetuar o estudo dos casos de interesse para a geração de alertas, pode-se utilizar como base a informação dos casos apresentados através dos *webinars* disponibilizados pela *Windows Ultimate Security*[1].

5.5 Arquitetura

Para garantir os requisitos de integração dos eventos de segurança dos sistemas *MS Windows* no SIEM *Alienvault* definidos nas secções anteriores, uma eficiente monitorização e gestão de todo o processo, e a capacidade de retenção de dados por um período considerável para eventual análise forense, foi desenhada uma arquitetura, conforme mostra a Figura 5.1.

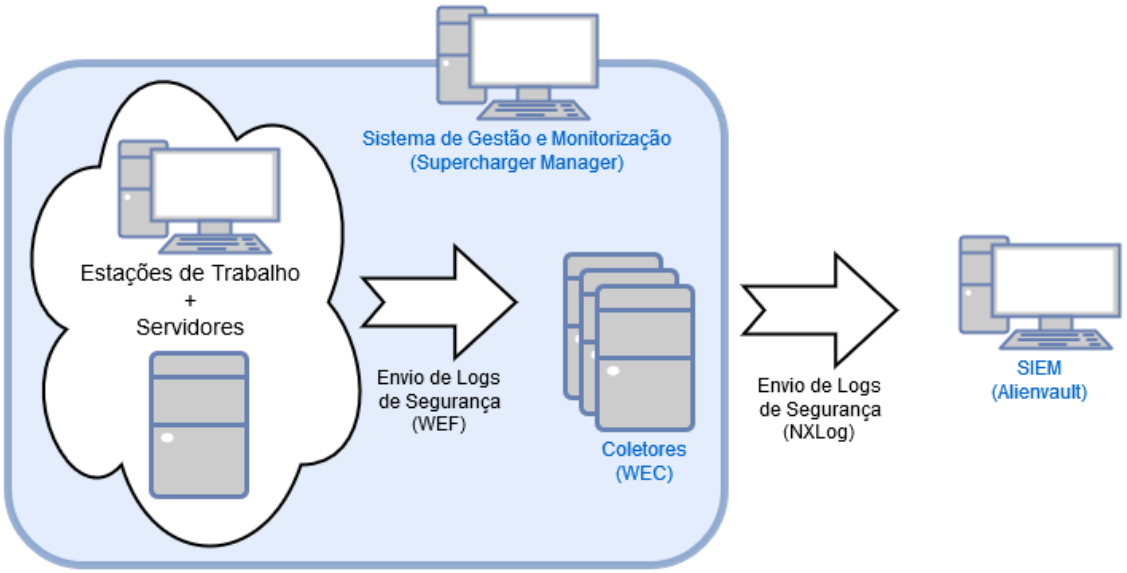


Figura 5.1 – Arquitetura

Através do *Windows Event Collection*, os eventos de segurança de interesse dos sistemas *MS Windows*, que serão as fontes de eventos (WEF), irão ser encaminhados

[1] https://www.ultimatewindowssecurity.com/webinars/

para um ou mais Coletores de Eventos (WEC), onde serão armazenados. De seguida, os eventos serão filtrados pela ferramenta NXLog, e encaminhados para o SIEM *Alienvault* apenas aqueles que serão utilizados por este para correlação e geração de alertas.

A plataforma *Supercharger*, embora não seja necessária para a implementação do ambiente *Windows Event Collection*, será utilizada neste projeto para alcançar os requisitos de gestão e monitorização definidos.

5.5.1 Eventos do *Windows* de Interesse

O objetivo deste projeto foca-se na recolha de eventos de segurança dos sistemas *MS Windows*, no entanto, é desejável, se possível, que a solução proposta seja escalável de forma a ter a capacidade de processar todos os tipos de eventos considerados relevantes que sejam gerados por estes sistemas.

Uma vez que os sistemas *MS Windows* podem chegar a ter mais de quatrocentos eventos de segurança distintos, dependendo da versão do sistema operativo, é necessário definir quais os eventos de interesse para serem armazenados num repositório centralizado para eventual análise forense, e quais os relevantes para serem endereçados ao SIEM para a monitorização em tempo real e geração de alertas. Desta forma é necessário definir a alto nível, quais os tipos de ações que deverão ser monitorizadas, para que seja efetuada a filtragem correta dos eventos de interesse para cada versão do sistema operativo.

Tendo por base a atual dimensão e distribuição geográfica da organização, é expectável que a nível global seja gerado uma elevada quantidade de eventos de segurança por segundo. Como tal, mais uma vez, torna-se imprescindível garantir uma correta seleção e filtragem na fonte, por forma a minimizar o impacto que a transmissão dessa elevada quantidade de eventos possa vir a ter na rede.

Para permitir uma correta seleção dos eventos, o estudo dos eventos de interesse é baseado na informação disponibilizada na página web da *Windows Ultimate Security*[1], que é uma das divisões pertencentes ao *Monterey Technology Group*, que por sua vez também é responsável pela divisão da *Logbinder*, onde é desenvolvido o *Supercharger*.

[1] https://www.ultimatewindowssecurity.com/

5.5.2 Recolha e Tratamento dos Dados

O processo de encaminhamento e recolha de dados deverá garantir uma eficiente transmissão dos eventos entre as fontes de eventos, o sistema de armazenamento centralizado, e o SIEM, para que este último possa receber a informação normalizada com o menor atraso possível, permitindo assim detetar eventuais anomalias em tempo quase real. É importante utilizar protocolos de comunicação seguros para transmissão dos dados, prevenindo desta forma modificações não autorizadas.

5.5.3 Armazenamento dos Dados

Uma vez definido o espectro de máquinas e o método de recolha e tratamento dos eventos de segurança, deve ser determinado qual o método indicado para o seu armazenamento. No caso deste projeto, foi definido pela empresa, que por motivos operacionais, os dados dos eventos recolhidos serão armazenados de forma centralizada e normalizada durante um período mínimo de um mês, de forma a permitir contribuir para uma eventual análise forense.

O sistema de armazenamento de dados deve ainda assegurar a segurança dos dados armazenados, bem como a sua redundância em caso de falha do sistema.

5.5.4 Gestão e Monitorização do Ambiente WEC

O sistema de gestão e monitorização deve permitir efetuar a gestão centralizada e uma eficiente monitorização do ambiente *Windows Event Collection*. Este deve ainda, garantir a uniformização das configurações aplicadas aos coletores de eventos e respetivas fontes de eventos, bem como permitir detetar atempadamente eventuais falhas no reencaminhamento dos eventos.

5.6 Resumo

Este capítulo descreveu a problemática que levou à realização deste projeto, e definiu os requisitos e a arquitetura de alto nível para a implementação da solução.

Capítulo 6

Implementação da Solução

Após uma cuidada análise dos requisitos, torna-se possível iniciar a implementação da solução de forma a cumprir com os objetivos propostos para este projeto.

A implementação será efetuada em duas fases, sendo que a primeira fase será implementada em ambiente de laboratório, que tenta reproduzir de forma mais fiel possível o ambiente de rede onde a solução vai ser implementada, e posteriormente, numa segunda fase, esta será implementada em ambiente de produção, integrada com o SIEM *Alienvault USM* da Altice Portugal.

Neste capítulo será descrito o processo de desenvolvimento da solução, bem como os recursos necessários para efetuar a recolha, filtragem, tratamento e gestão centralizada de eventos das máquinas que executam o sistema operativo *MS Windows*, e a sua integração com o *Alienvault USM* da Altice Portugal.

6.1 FASE 1 – Ambiente de Laboratório

Este capítulo descreve, o ambiente de laboratório disponibilizado e as diferentes arquiteturas implementadas nos casos de estudo, tendo em vista a escolha da arquitetura que melhor se adapta à implementação do projeto em ambiente de produção.

Descreve ainda as configurações e regras de correlação necessárias que permitem ao SIEM *Alienvault OSSIM*, com base nos eventos de segurança recebidos dos dispositivos *MS Windows*, detetar padrões que indiciem anomalias de segurança ou mesmo ataques em preparação ou já em curso.

6.1.1 **Concretização do Ambiente de Laboratório**

O ambiente de laboratório disponibilizado pela organização consiste numa estação virtualizada com o sistema operativo *CentOS Linux 7 (Core)*, com oito processadores virtuais (vCPU), quarenta gigabytes de RAM, e trezentos gigabytes de espaço em disco. Sobre esta plataforma foi instalado o sistema de virtualização *Oracle VM VirtualBox*, onde foram montadas as máquinas e o ambiente de rede necessário para simular os Casos de Estudo apresentados na Secção 6.1.2 , e implementar a arquitetura final do ambiente de laboratório.

Para implementação do ambiente de laboratório foram criados três domínios, onde o *dominio1.pt* é o domínio principal. O domínio *dominio2.dominio1.pt* é um domínio filho do domínio *dominio1.pt*, e dispõe de relação de confiança com este. E o domínio *dominio3.pt*, que se encontra em outra árvore e não tem qualquer relação de confiança com os outros dois domínios.

Cada um desses domínios é composto por um Controlador de Domínio, com o sistema operativo *Windows Server 2016 x64*, (com as funcionalidades de *Active Directory Domain Services* – AD DS, e as funcionalidades de *Domain Name System – DNS*), e um Coletor de Eventos com o sistema operativo *Windows Server 2016 x64*. Foi ainda instalado nos vários domínios um conjunto de máquinas para simularem os clientes WEF, garantindo uma correta amostragem dos diversos sistemas operativos existentes, entre as versões *Windows XP SP3/Windows Server 2003* e *Windows 10/Windows Server 2016*.

Adicionalmente a esta configuração base, no domínio *dominio1.pt* foram adicionadas as funcionalidades de *Dynamic Host Configuration Protocol* (DHCP), *Active Directory Certificate Services* (AD CS) e *Internet Information Services* (IIS) ao Controlador de Domínio. Neste domínio de topo, à imagem de como será implementado em ambiente de produção, foi instalado o SIEM *Alienvault OSSIM*, bem como o sistema *Supercharger Manager* (de acordo com o ANEXO B) numa máquina com o sistema operativo *Windows Server 2016 x64*, para permitir efetuar a gestão e monitorização de todo o ambiente WEC.

Após a configuração do ambiente de rede necessário, os Coletores de Eventos do Windows foram configurados (de acordo com o ANEXO C) para receber eventos através do protocolo HTTPS e permitir que o *Supercharger Manager* possa efetuar a sua monitorização e gestão remota. De seguida, foi instalado e configurado o *NXLog*

Community Edition (de acordo com o ANEXO D), por forma a normalizar e reencaminhar os eventos recebidos nos WEC, através de *syslog*, para o SIEM *Alienvault OSSIM*.

Para que as fontes de eventos (WEF) encaminhem os eventos relevantes para o seu respetivo Coletor de Eventos, e uma vez que se pretende efetuar as configurações de forma o mais centralizada possível, foi necessário criar políticas de grupo ao nível do Controlador de Domínio para configurar o serviço de reencaminhamento de eventos (conforme ANEXO E), políticas de grupo para definir o Coletor de Eventos para onde o WEF deverá reencaminhar os eventos (conforme ANEXO E), assim como o nível de auditoria das máquinas (conforme ANEXO E).

Uma vez que existem determinadas especificidades na configuração do ambiente WEC, bem como na comunicação dos agentes do *Supercharger* com o *Supercharger Manager* entre domínios, foram efetuados três casos de estudo, para permitir determinar a melhor localização lógica dos coletores de eventos do *Windows*.

As diferentes arquiteturas presentes nos casos de estudo definidos na secção 6.1.2 foram configuradas através das políticas de grupo que definem o Coletor de Eventos para onde o WEF deverá reencaminhar os eventos, devendo estas ser configuradas a partir do Controlador de Domínio.

6.1.2 **Variantes da Arquitetura – Casos de Estudo**

Foram implementadas em ambiente de laboratório, três variantes da arquitetura definida na secção 5.5 , que diferiram entre si ao nível de localização lógica dos coletores de eventos. A criação destas variantes tem como objetivo apurar a capacidade dos coletores de eventos para coletar eventos entre domínios com diferentes graus de confiança entre si, e encaminhá-los para o SIEM *Alienvault OSSIM*.

O estudo das duas primeiras variantes da arquitetura serve de base para definir a arquitetura do caso de estudo final, a ser implementado em ambiente de laboratório, e posteriormente em ambiente de produção, bem como apurar a capacidade de gestão e monitorização, entre diferentes domínios, do sistema *Supercharger*.

Caso de Estudo 1 – Cada domínio dispõe do seu servidor WEC

Neste Caso de Estudo, cada domínio/subdomínio dispõe do seu próprio coletor de eventos, conforme mostra a Figura 6.1.

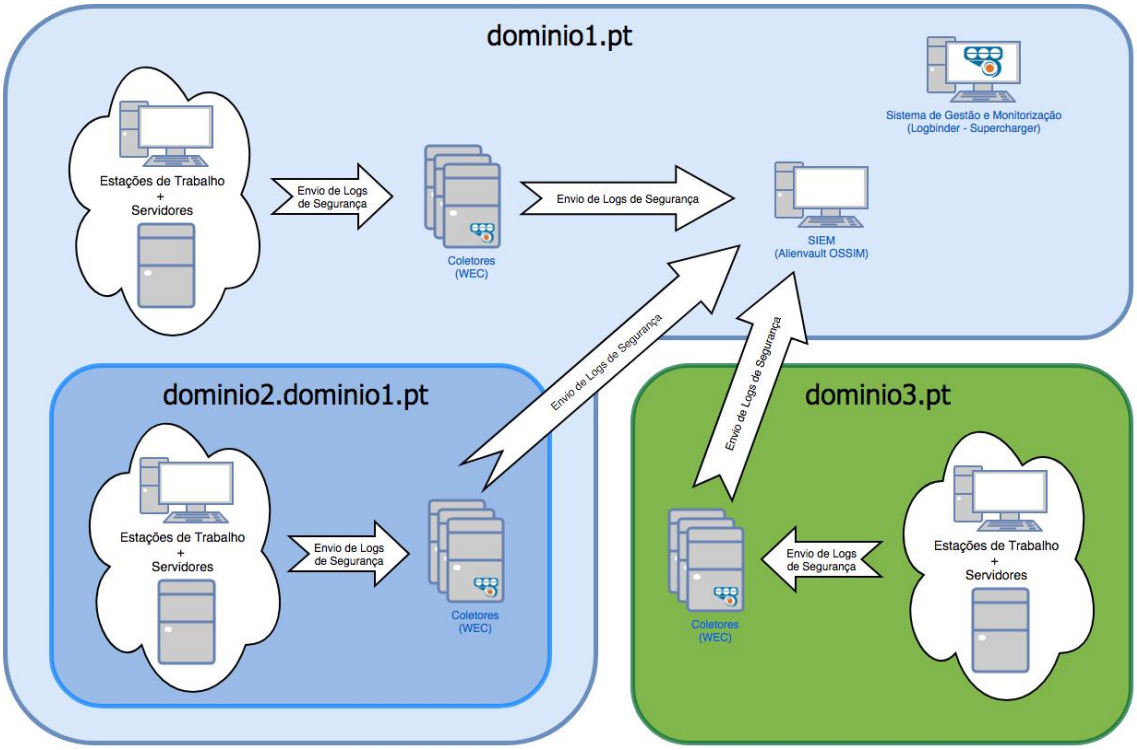


Figura 6.1 – Arquitetura de Rede do Ambiente de Laboratório (Caso Estudo 1)

As máquinas de cada domínio/subdomínio, que serão as fontes de eventos, foram configuradas para reencaminhar os eventos de segurança relevantes para o coletor de eventos existente no seu domínio, de modo a permitir que este os armazene de forma centralizada, e reencaminhe aqueles que são suscetíveis de gerar alarmes diretamente para o SIEM *Alienvault OSSIM*.

Este caso de estudo permite analisar a capacidade do *software NXLog* para reencaminhar os eventos entre diferentes domínios, com ou sem relação de confiança, bem como a capacidade do *Supercharger Manager*, para monitorizar e gerir os ambientes WEC desses domínios.

Através da análise deste caso de estudo, verificou-se que o *Supercharger* não apresenta qualquer limitação quando utilizado para gestão e monitorização de servidores WEC que se encontrem num domínio diferente do domínio em que o *Supercharger Manager* se encontra instalado, tenham estes, ou não, relação de confiança com o domínio onde se encontra o *Supercharger Manager*.

O *software* de gestão de eventos, *NXLog*, utilizado para normalizar e encaminhar os eventos do coletor para o SIEM, também revelou ser agnóstico em relação ao domínio, não apresentando limitações na utilização entre domínios, independentemente da sua relação de confiança.

Uma vez que neste caso de estudo as fontes de eventos (WEF) encontram-se no mesmo domínio dos coletores de eventos (WEC), não foi verificado qualquer tipo de limitação, tanto a nível de configuração, como a nível de recolha de eventos. Neste caso, é possível configurar de igual forma subscrições iniciadas pela fonte, e/ou subscrições iniciadas pelo coletor (ver secção 3.4.1).

Caso de Estudo 2 – Todos os servidores WEC no mesmo domínio

Neste Caso de Estudo, foram instalados coletores de eventos apenas no domínio *dominio1.pt*, de forma a permitir reduzir o número de coletores de eventos necessários e centralizar num único ponto a recolha de eventos dos diversos domínios, conforme mostra a Figura 6.2.

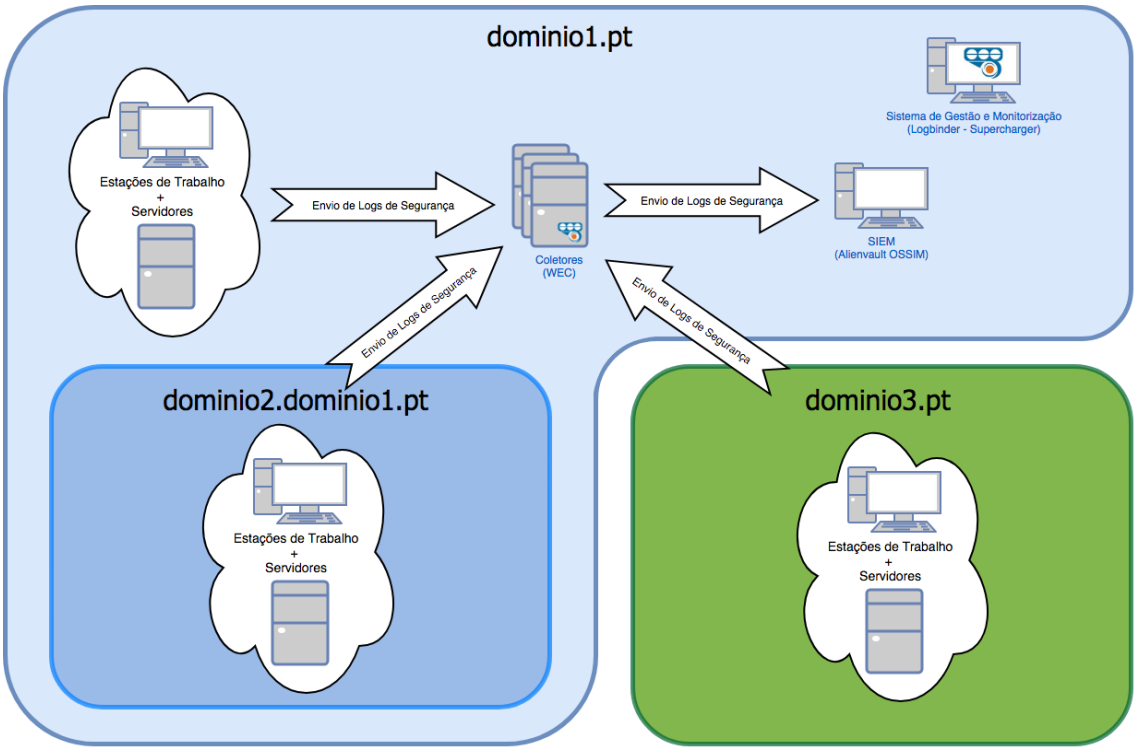


Figura 6.2 – Arquitetura de Rede do Ambiente de Laboratório (Caso Estudo 2)

As máquinas de cada domínio/subdomínio, que serão as fontes de eventos, foram configuradas para reencaminhar os eventos de segurança relevantes para os coletores de

eventos existentes no domínio *dominio1.pt*, de modo a permitir que este os armazene de forma centralizada, e reencaminhe aqueles que são suscetíveis de gerar alarmes diretamente para o SIEM *Alienvault OSSIM*.

Este caso de estudo permite analisar a capacidade do ambiente *Windows Event Collection* para reencaminhar os eventos entre diferentes domínios, que disponham, ou não, de relação de confiança entre si.

Contrariamente ao verificado com o *Supercharger* e com o *NXLog*, no caso de estudo 1, que não apresentam limitações na utilização entre domínios com diferentes relações de confiança, o ambiente *Windows Event Collection* apresenta algumas particularidades na relação entre domínios, que terão de ser tidas em conta para escolher a melhor arquitetura a implementar para cada ambiente em particular.

Neste caso, foram verificadas algumas limitações de configuração do ambiente *Windows Event Collection*, uma vez que não é possível configurar subscrições iniciadas pelo coletor para serem aplicadas nos WEF que se encontram em domínios que não têm relação de confiança com o domínio onde está instalado o WEC. Assim, apenas é possível configurar subscrições iniciadas pela fonte, e tem se ser gerado e instalado um certificado de autenticação em cada WEF, para que este se possa autenticar perante o WEC. No caso de os domínios terem relação de confiança, não existe qualquer limitação na implementação do ambiente *Windows Event Collection*.

Caso de Estudo 3 – Arquitetura Escolhida

Depois de analisados os casos de estudo 1 e 2, a solução escolhida que melhor se adapta para a implementação do projeto, é uma arquitetura mista, conforme mostra a Figura 6.3, onde os coletores de eventos são instalados apenas nos domínios que não dispõem de relação de confiança entre si.

Os domínios com relação de confiança encaminham os eventos para o coletor existente no seu domínio de topo, de modo a permitir que este os armazene de forma centralizada, e reencaminhe apenas aqueles que são suscetíveis de gerar alarmes para o SIEM *Alienvault*.

Esta solução torna possível configurar subscrições iniciadas pelo coletor em todos os domínios, uma vez que todos os WEF terão relação de confiança com o seu respetivo WEC. Adicionalmente permite reduzir a necessidade de utilização e configuração de certificados de autenticação, que apenas são necessários nos casos em que o WEF e o

WEC se encontrem em domínios sem relação de confiança, sendo que nesse caso existe ainda a limitação de apenas ser possível criar subscrições iniciadas pela fonte.

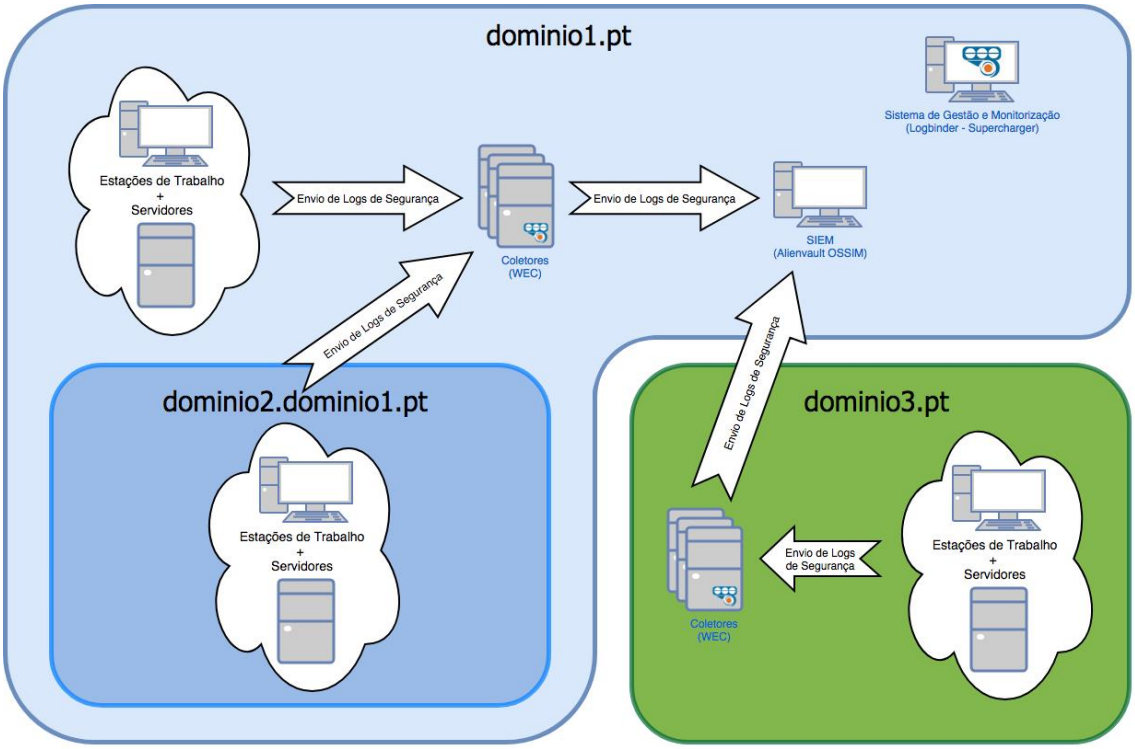


Figura 6.3 – Arquitetura de Rede Final do Ambiente de Laboratório

6.1.3 Configuração do *Alienvault*

Os eventos de segurança das plataformas *MS Windows* que são recebidos pelo WEC, são posteriormente normalizados e encaminhados pelo *software NXLog*, através do protocolo *syslog*, para o *Alienvault OSSIM*.

Após receção desses eventos no formato *Raw*, o SIEM *Alienvault OSSIM* dispõe de *plugins* que permitem interpretar e normalizar os dados recebidos, e proceder ao seu armazenamento em campos de dados comuns (Ver ANEXO A) para posterior correlação e geração de alertas que podem ser configurados através de diretivas.

Dessa forma, é necessário personalizar o *plugin* que interpreta os eventos de segurança do *Windows* recebidos através de *syslog*, para que este possa extrair a informação relevante dos eventos *Raw*, bem como criar Diretivas para correlacionar a informação extraída dos eventos recebidos e gerar alertas de interesse.

Plugin NXLog

Os eventos *Raw* recebidos via *NXLog* pelo *Alienvault* são armazenados num ficheiro de log localizado em */var/log/nxlog.log*. O *plugin* responsável por extrair e interpretar esses eventos é configurado em */etc/ossim/agent/plugins/nxlog.cfg*.

Para implementação deste projeto, foi personalizado no *plugin* responsável por interpretar os eventos de segurança do Windows recebidos através do *NXLog*, os seguintes ID's de eventos:

- 4624 – O *logon* de uma conta foi efetuado com êxito.
- 4625 – Falha no *logon* de uma conta.
- 4740 – Uma conta de utilizador foi bloqueada.
- 4768 – Foi solicitado o *ticket* (TGT) de autenticação *Kerberos*.
- 4771 – Falha na pré-autenticação *Kerberos*.
- 4778 – Uma sessão foi reconectada a uma estação *Windows*.
- 4801 – A estação de trabalho foi desbloqueada.

Embora este *plugin* permita interpretar outros eventos de segurança, apenas foram personalizados os ID's dos eventos descritos anteriormente, de acordo com o ANEXO F, uma vez que são estes os eventos de segurança considerados relevantes para a criação das Diretivas que permitem monitorizar as atividades de interesse definidas na secção 5.4.2 .

Por forma a possibilitar a extração, interpretação, normalização e correlação de alguma informação de interesse dos eventos *Raw* para ser armazenada em campos de dados, foi necessário desenvolver as funções personalizadas constantes no ANEXO G. Estas funções deverão ser adicionadas ao ficheiro de configuração localizado em */etc/ossim/agent/plugins/custom_functions/winnxlog_functions.cfg*.

Adicionalmente, para ser possível dotar o sistema com a funcionalidade de verificação do nível de privilégios da conta de utilizador responsável pera geração de um determinado evento de segurança, foi desenvolvido um *script* em *Python* (ver ANEXO H), que deverá ser executado periodicamente, e que permite identificar as contas de domínio com privilégios de administração, e armazenar essa informação num ficheiro de texto que será utilizado como fonte de informação para algumas das funções personalizadas utilizadas pelo *plugin*.

Diretivas (Regras de Correlação)

Para permitir a geração de alertas, de acordo com as atividades de interesse definidas na secção 5.4.2 , foram desenvolvidas diretivas, conforme ANEXO I, que permitem correlacionar os eventos recebidos das plataformas *MS Windows*.

Tendo por base alguns dos casos apresentados nos *webinars* da *Windows Ultimate Security*, foram definidas as seguintes atividades de interesse de alto nível:

Atividade de Interesse 1 – *Deteção de Ataques de Bruteforce*

Para esta atividade de interesse foram efetuadas quatro diretivas, que permitem identificar ataques de *bruteforce*, através dos eventos de falha de autenticação (ID 4625):

- Falhas de Autenticação em Contas de Domínio com Privilégios;
- Falhas de Autenticação em Contas de Domínio Sem Privilégios;
- Falhas de Autenticação em Contas Locais;
- Falhas de Autenticação a Partir da Mesma Origem.

Estas diretivas permitem identificar a existência de múltiplas falhas de autenticação num curto período de tempo, bem como perceber se o ataque está a ser dirigido a contas locais, ou a contas de domínio com, ou sem, privilégios.

Para esta atividade de interesse, não serão utilizados os eventos *Kerberos* que são gerados no controlador de domínio (ID 4768 e ID 4771), uma vez que não é possível correlacionar esses eventos de forma determinística, com os eventos gerados nas plataformas onde é efetuada a autenticação (ID 4625). Assim, para que não sejam gerados dois alarmes para o mesmo incidente, optou-se por correlacionar apenas os eventos de falha de autenticação (ID 4625), uma vez que disponibilizam informação mais relevante que os eventos *Kerberos*.

Atividade de Interesse 2 – *Deteção de Uso Indevido de Credenciais*

Esta atividade de interesse visa identificar o uso indevido de credenciais, através da presença de movimentos laterais. Para tal, foi efetuada a seguinte diretiva, que visa detetar este tipo de comportamento anómalo:

- Utilizador Autenticado em Várias Estações.

Para determinar a presença de um utilizador que se autentique em múltiplas estações num curto espaço de tempo, foram utilizados os eventos de *logon* com êxito (ID 4624), restabelecimento de sessão (ID 4778) e desbloqueio da estação de trabalho (ID 4801).

Neste caso, não serão utilizados os eventos de *logoff* (ID 4634 e ID 4647), ou sessão desconectada (ID 4779), uma vez que no caso de a sessão não ser terminada corretamente, estes eventos não são gerados. Verificou-se também, que estes eventos apenas são reencaminhados para o coletor de eventos, e posteriormente para o *Alienvault*, quando a estação é novamente iniciada, podendo originar falsos positivos.

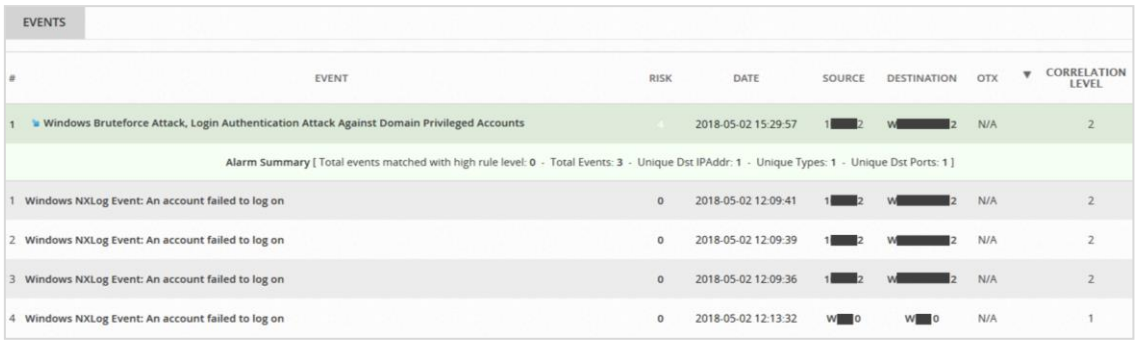
Atividade de Interesse 3 – *Deteção de Ataques DoS a Contas de Utilizador*

Esta atividade de interesse permite identificar ataques de *Denial-of-service* a contas de utilizador, através da deteção de múltiplos eventos de bloqueios de conta (ID 4740), num período de tempo. Para esse efeito foram criadas duas diretivas:

- Múltiplos Bloqueios de Conta;
- Múltiplos Bloqueios de Conta a Partir da Mesma Origem.

Embora ambas as diretivas utilizem os mesmos eventos para correlação, a primeira é mais abrangente e permite detetar ataques distribuídos, enquanto a segunda é mais específica, permitindo gerar alertas a partir de um menor número de eventos, e identificar o endereço a partir do qual o ataque está a ser efetuado.

As diretivas apresentadas nestas três atividade de interesse de alto nível, e presentes no ANEXO I, foram avaliadas em ambiente de laboratório, através da geração de eventos nos WEF, com intervalos de tempo e número de ocorrências adaptados para o ambiente existente, como por exemplo, a diretiva para deteção de falhas de autenticação em contas de domínio com privilégios (ver alerta gerado na Figura 6.4).



EVENTS

#	EVENT	RISK	DATE	SOURCE	DESTINATION	OTX	CORRELATION LEVEL
1	Windows Bruteforce Attack, Login Authentication Attack Against Domain Privileged Accounts		2018-05-02 15:29:57	1[redacted]2	W[redacted]2	N/A	2
	Alarm Summary [Total events matched with high rule level: 0 - Total Events: 3 - Unique Dst IPAddr: 1 - Unique Types: 1 - Unique Dst Ports: 1]						
1	Windows NXLog Event: An account failed to log on	0	2018-05-02 12:09:41	1[redacted]2	W[redacted]2	N/A	2
2	Windows NXLog Event: An account failed to log on	0	2018-05-02 12:09:39	1[redacted]2	W[redacted]2	N/A	2
3	Windows NXLog Event: An account failed to log on	0	2018-05-02 12:09:36	1[redacted]2	W[redacted]2	N/A	2
4	Windows NXLog Event: An account failed to log on	0	2018-05-02 12:13:32	W[redacted]0	W[redacted]0	N/A	1

Figura 6.4 – Exemplo de Alerta Gerado pelo Alienvault

Esta diretiva foi testada em laboratório para 1 ocorrência no nível de correlação um, mais 3 ocorrências no nível de correlação dois num espaço temporal de 180 segundos, conforme Figura 6.5, enquanto no caso de estudo foi definido um intervalo de tempo de 120 segundos para 1 ocorrência no nível de correlação um, mais 10 ocorrências no nível de correlação dois.

```
<directive id="500009" name="Windows Bruteforce Attack, Login
Authentication Attack Against Domain Privileged Accounts"
priority="5">
   <rule type="detector" name="Domain Priviliged Account
    authentication failure" from="ANY" to="ANY" port_from="ANY"
    port_to="ANY" reliability="+0" occurrence="1" plugin_id="1817"
    plugin_sid="4625" userdata6="Yes">
      <rules>
         <rule type="detector" name="Domain Priviliged Accounts -
          Multiple authentication failures" from="ANY" to="ANY"
          port_from="ANY" port_to="ANY" reliability="10"
          occurrence="3" time_out="180" plugin_id="1817"
          plugin_sid="4625" userdata6="Yes"/>
      </rules>
   </rule>
</directive>
```

Figura 6.5 – Exemplo de Diretiva para Deteção de Falhas de Autenticação

6.1.4 Conclusão

Os diversos casos de estudo analisados nesta secção permitiram avaliar que a arquitetura que mais se adequa à implementação do projeto em ambiente de produção é uma arquitetura mista, conforme Figura 6.3, bem como definir regras de correlação de eventos que permitam ao SIEM detetar padrões que indiciem anomalias de segurança relevantes, que possam ser implementadas em ambiente de produção.

Em ambiente de laboratório foi possível determinar que as regras de correlação funcionam de acordo com o espectável, no entanto não foi possível aferir com precisão a existência de falsos positivos e de falsos negativos. Esta limitação deveu-se ao facto de em ambiente de laboratório não dispor de outra ferramenta que pudesse utilizar para comparar os resultados obtidos, de forma a verificar a existência de falsos negativos. Por outro lado, a ausência de eventos gerados pelo normal funcionamento da infraestrutura de uma organização, uma vez que os eventos foram gerados em ambiente controlado, tornou complexo aferir com precisão a existência de falsos positivos.

6.2 FASE 2 – Ambiente de Produção

Conforme referido na secção 5.4.1 , foi definido pela empresa que este projeto será implementado apenas na infraestrutura de servidores e estações de trabalho geridos pela Direção de Informação e Tecnologia (DIT), e será estendido posteriormente à infraestrutura gerida pela Direção de Operações B2B (DBO), já fora do âmbito deste projeto.

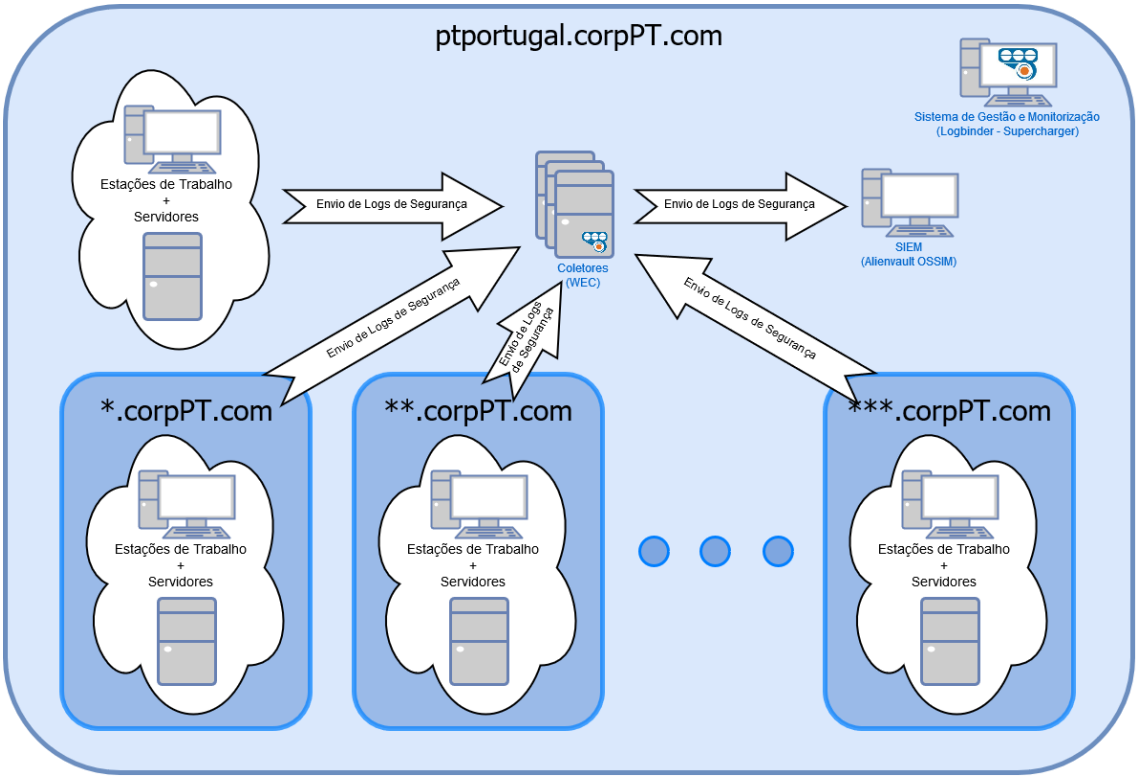


Figura 6.6 – Arquitetura de Rede do Ambiente de Produção

Após implementação e avaliação dos diferentes Casos de Estudo em ambiente de laboratório, conforme Capítulo 6.1 , verificou-se que a arquitetura mais adequada para implementação em ambiente de produção na infraestrutura de servidores e estações de trabalho geridos pela Direção de Informação e Tecnologia (DIT), consiste na arquitetura em que todos os WEC e o *Supercharger Manager* se encontram instalados no domínio principal (acordo Figura 6.6), uma vez que os outros domínios têm relação de confiança com este.

Com base na arquitetura definida, efetuou-se o estudo para definir os recursos e requisitos necessários para a implementação do projeto.

6.2.1 Recursos de *Hardware*

Para implementação deste projeto em ambiente de Produção, a empresa decidiu que por motivos operacionais, a implementação dos servidores WEC e *Supercharger Manager* serão efetuadas em máquinas virtualizadas. Estas máquinas terão instalado de base o sistema operativo *Windows Server 2016* de 64 bits.

O dimensionamento para implementação da solução na infraestrutura de servidores e estações de trabalho geridos pela DIT (com cerca de vinte e dois mil sistemas *MS Windows*), foi efetuado tendo por base os dois polos existentes (Picoas e Covilhã) com uma distribuição de carga não equitativa, que permita suportar simultaneamente a falha de um servidor WEC em cada polo, ou mesmo a falha de um dos polos. Sendo que no caso da falha de um dos polos, considera-se que todos os servidores WEC do polo operacional se encontram a funcionar.

Uma vez que a *Microsoft* não disponibiliza métricas concretas para o cálculo do dimensionamento do ambiente WEC, foram tidos em conta alguns dos fatores que esta aponta como limitativos para a escalabilidade do sistema:

- Máximo de dez mil eventos por segundo e dez mil WEF, por cada WEC[1];
- O tamanho médio que um evento ocupa no *log* são 500 bytes[2];
- O tamanho máximo de um *log* de eventos é 2 terabytes[4];

[1] https://docs.microsoft.com/en-us/windows/security/threat-protection/use-windows-event-forwarding-to-assist-in-intrusion-detection

[2] https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/dd349798(v=ws.10)

- Largura de banda da rede e número de portas TCP abertas disponíveis[3];
- Velocidade de escrita em disco[3].

Da observação do ambiente *MS Windows* da Altice Portugal, determinou-se que em média cada WEF gera cerca de dez mil eventos num período de vinte e quatro horas, o que permitiu dimensionar o espaço em disco necessário, conforme Tabela 6.1.

Tabela 6.1 – Dimensionamento do Espaço Necessário em Disco

	PERÍODO DE 24H	**PERÍODO DE 30 DIAS**
Nº MÉDIO DE EVENTOS GERADOS POR WEF	10.000	300.000
ESPAÇO OCUPADO NO LOG POR WEF	5 MB	150 MB
ESPAÇO OCUPADO NO LOG PELOS WEF DA DIT	0,11 TB	3,30 TB

Tendo por base os fatores limitativos apontados pela *Microsoft*, para permitir suportar o número de WEF existentes e garantir uma capacidade sobrante em momentos em que exista um grande pico de eventos na rede, são necessários três servidores WEC. De forma a cumprir com os requisitos de dimensionamento referidos anteriormente, é necessário um total de seis servidores WEC (três em cada polo), e um servidor *Supercharger Manager*, com as características definidas na Tabela 6.2.

Tabela 6.2 – Características dos Servidores Virtuais

	SERVIDOR WEC	**SERVIDOR SUPERCHARGER**
CPU (Nº *CORES*)	8 vCPU	8 vCPU
RAM	16 GB	16 GB
DISCO	3 TB	1 TB

6.2.2 Requisitos de *Software*

Para implementação deste projeto é necessária a instalação e/ou configuração de algum *software* específico, de acordo com a Tabela 6.3.

Tabela 6.3 – Requisitos de Software

	REQUISITOS DE *SOFTWARE*
FONTES DE EVENTOS (WEF)	Windows Remote Management
COLETOR DE EVENTOS DO WINDOWS (WEC)	Windows Remote Management NXLog Community Edition Supercharger aka Agent
SUPERCHARGER MANAGER	Supercharger Manager

Nas fontes de eventos (WEF), que correspondem aos dispositivos *MS Windows* a operar dentro da rede corporativa e infraestruturas de *data center* da Altice Portugal, tem de ser configurado o serviço *Windows Remote Management* de forma a reencaminhar os eventos de segurança relevantes para o WEC respetivo. Esta configuração pode ser efetuada localmente, ou através de políticas de grupo.

Os servidores WEC têm como base o sistema operativo *Windows Server 2016 x64*, sendo necessário configurar o serviço *Windows Remote Management* de forma a permitir receber eventos de dispositivos remotos através de HTTPS. Estes servidores terão ainda instalado o *Supercharger Controller*, de forma a permitir a gestão e monitorização remota pelo *Supercharger Manager*. Por fim, é necessário instalar o *software NXLog Community Edition* que terá de ser configurado para encaminhar os eventos recebidos para o SIEM *Alienvault USM*.

O servidor *Supercharger Manager*, tem como base o sistema operativo *Windows Server 2016 x64*, sendo necessário instalar e configurar a aplicação *Supercharger Manager*, que irá permitir a gestão e monitorização centralizada de todos os WEC.

Uma vez que o *Windows Remote Management* é um serviço do Windows, além do licenciamento da versão do sistema operativo, não requer nenhum tipo de licenciamento específico para a sua utilização, assim como o *NXLog Community Edition*, que é distribuído sobre a forma de *software* de fonte aberta (*open source*). Já a aplicação *Supercharger Manager*, requer a aquisição de licenciamento, de acordo com o número e tipo de estações WEF que se pretenda monitorizar[1].

[1] https://www.logbinder.com/Products/Supercharger/Pricing

6.2.3 **Implementação**

De acordo com o dimensionamento e requisitos apresentados nas secções 6.2.1 , e 6.2.2 , deverão ser instalados e configurados (com base no ANEXO B e ANEXO C) no domínio *ptportugal.corppt.com* seis servidores WEC, e um servidor *Supercharger Manager*, todos com o com o sistema operativo *Windows Server 2016 x64*. Estes servidores irão estar distribuídos por dois polos (Picoas e Covilhã), de acordo com a Figura 6.7, sendo que serão instalados três servidores WEC em cada um dos polos, e o servidor *Supercharger Manager* no polo da Covilhã.

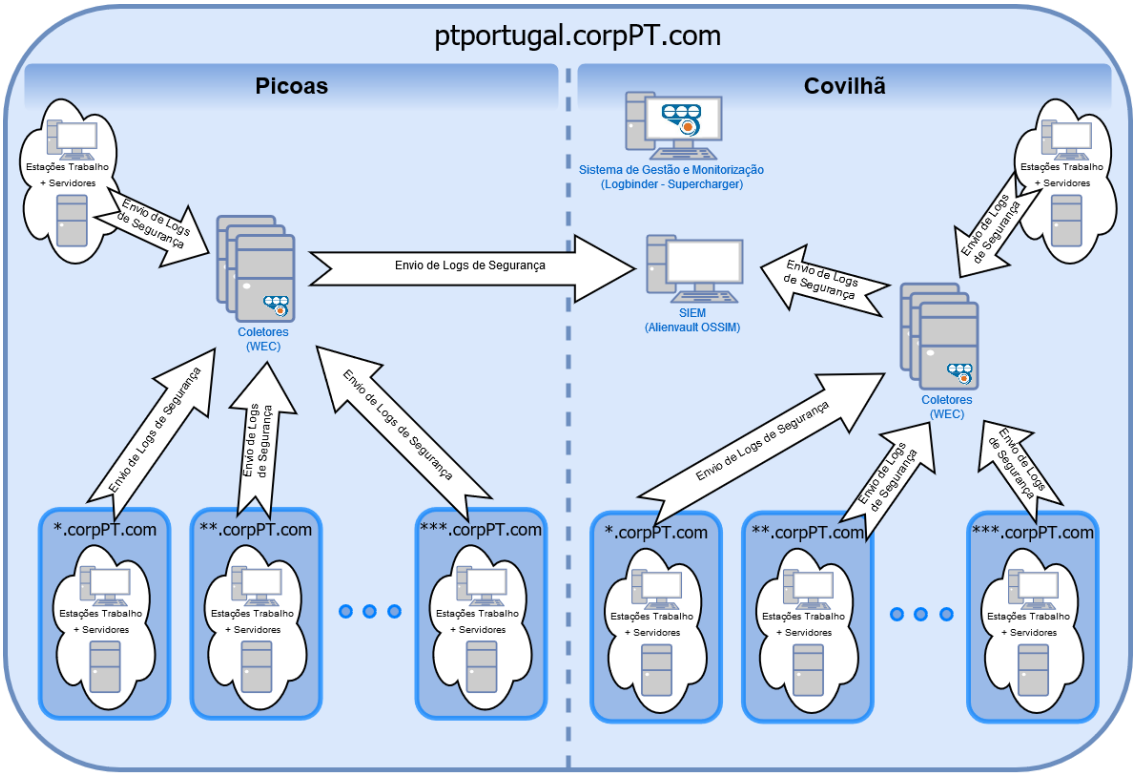


Figura 6.7 – Localização Física

Após instalação e configuração os servidores WEC, e do *Supercharger Manager*, deverão ser criadas as subscrições, através da página web do *Supercharger Manager*, de acordo com o ANEXO J, para que os WEF comecem a reencaminhar os eventos selecionados para o respetivo servidor WEC.

De seguida, deverá proceder-se à configuração do *NXLog Community Edition* (de acordo com o ANEXO D) em todos os WEC, por forma a permitir que os eventos recolhidos por estes sejam enviados por *syslog*, através da rede de gestão, para o SIEM *Alienvault USM* da Altice Portugal existente no domínio *ptportugal.corpPT.com*, e que já se encontra atualmente a receber informação de diversas fontes de eventos.

Após a configuração de todo o ambiente WEC, torna-se necessário configurar o *Alienvault USM* para interpretar e correlacionar os eventos que lhe são enviados pelos servidores WEC através do *NXLog*. Desta forma, para implementação deste projeto em ambiente de produção, deverão ser configurados os *plugins* e diretivas do Alienvault, de acordo com as configurações aplicadas em ambiente de laboratório, descritas na secção 6.1.3 .

6.2.4 **Conclusão**

Nesta secção foram descritos os requisitos de *software*, recursos de *hardware* e procedimentos para a implementação do projeto em ambiente de produção. Devido à organização não ter disponibilizado os recursos necessários, em tempo útil, não foi possível efetivar a instalação do projeto em ambiente de produção.

A solução proposta para implementação em ambiente de produção pretende garantir o cumprimento dos requisitos definidos no Capítulo 5. Assim sendo, os requisitos de compatibilidade e escalabilidade são alcançados com escolha do *Windows Event Collection* para a recolha e centralização dos eventos. Com a análise do dimensionamento e distribuição dos servidores WEC apresentada na secção 6.2.1 , garante-se o cumprimento dos requisitos de capacidade, robustez e disponibilidade. Por último, pretende-se assegurar o requisito de segurança através da utilização de protocolos de comunicação seguros, assim como a utilização da rede de gestão para efetuar a transmissão de dados através de *syslog* entre os WEC e o *Alienvault USM*.

Ficando a faltar neste ponto garantir os requisitos de desempenho da solução, a sua análise em ambiente de laboratório será apresentada no Capítulo 7.

6.3 Resumo

Este capítulo descreveu o processo de implementação e avaliação dos diferentes casos de estudo analisados em ambiente de laboratório, bem como a arquitetura e recursos necessários para a concretização da implementação da solução em ambiente de produção, de acordo com os requisitos apresentados.

Capítulo 7

Avaliação da Solução

Após apresentada a arquitetura final no capítulo anterior, neste capítulo será efetuada a avaliação da solução, com base no cumprimento dos objetivos propostos e nos resultados obtidos.

Devido a não ter sido possível a organização disponibilizar, em tempo útil, os recursos necessários para efetuar a instalação da solução proposta em ambiente de produção, a avaliação restringiu-se à análise funcional, conforme apresentado na secção 6.1 , e análise de desempenho em ambiente de laboratório, que será apresentada neste capítulo, não tendo sido recolhidos dados em ambiente de produção.

7.1 Configurações

A análise da solução final foi efetuada no ambiente de laboratório disponibilizado pela organização, que conforme descrito na secção 6.1.1 , consiste numa estação virtualizada com o sistema operativo *CentOS Linux 7 (Core)*, com oito processadores virtuais (vCPU) com velocidade de processamento de 2,60GHz, quarenta gigabytes de RAM, e trezentos gigabytes de espaço em disco.

Sobre esta plataforma foi instalado o sistema de virtualização *Oracle VM VirtualBox*, onde foram montadas as máquinas e o ambiente de rede necessário para implementar a arquitetura escolhida, de acordo com a Figura 6.3, com o objetivo de se proceder à análise funcional e de desempenho desta solução.

7.2 Cenários

Foram criados três cenários, para permitir determinar o desempenho e capacidade de processamento de eventos do ambiente *Windows Event Collection*, assim como do *software* de gestão de eventos *NXLog Community Edition*, e do SIEM *Alienvault OSSIM*.

O primeiro cenário foi criado com o objetivo de avaliar o desempenho do *NXLog Community Edition*, tendo sido utilizado um servidor WEC, com 4 processadores virtuais e 4 gigabytes de memória RAM, a partir do qual foram gerados cerca de quatrocentos mil eventos, com o recurso a um script. Esses eventos foram armazenados no log *ForwardedEvents*, para posteriormente serem utilizados nos testes de desempenho do *NXLog*, conforme descrito na secção 7.3.1 .

O segundo cenário foi desenvolvido para avaliação de desempenho do SIEM *Alienvault OSSIM*, conforme descrito na secção 7.3.2 , tendo-se utilizado para tal o servidor WEC e os eventos armazenados no log *ForwardedEvents* do primeiro cenário, e um SIEM *Alienvault OSSIM* com 6 processadores virtuais e 16 gigabytes de memória RAM.

Por fim, o terceiro cenário foi desenvolvido para avaliação de desempenho do ambiente *Windows Event Collection*. Para este cenário foram utilizadas as mesmas máquinas do cenário anterior, e mais duas máquinas WEF para gerar eventos, conforme descrito na secção 7.3.3 .

7.3 Apresentação de Resultados e Análise

Nesta secção serão descritos os testes de desempenho realizados em cada um dos cenários enumerados na secção 7.2 , e efetuada a apresentação e análise dos resultados obtidos.

7.3.1 Primeiro Cenário – Avaliação do *NXLog Community Edition*

Para proceder à avaliação de desempenho do *NXLog Community Edition*, utilizou-se um conjunto de quatrocentos mil eventos armazenados no log *ForwardedEvents* do servidor WEC, para serem interpretados e encaminhados pelo *NXLog*. Os eventos foram encaminhados para um ficheiro de texto localizado no próprio WEC, de forma a

verificar a capacidade de processamento do *NXLog* sem as eventuais limitações que o SIEM possa ter na receção e processamento dos eventos.

Foram efetuados vários testes de modo a analisar o seu desempenho, consoante o número de caminhos em paralelo (*Routes*) que o *NXLog* possa executar no processo de receção, interpretação e encaminhamento dos eventos. Os resultados desses testes são apresentados na Figura 7.1.

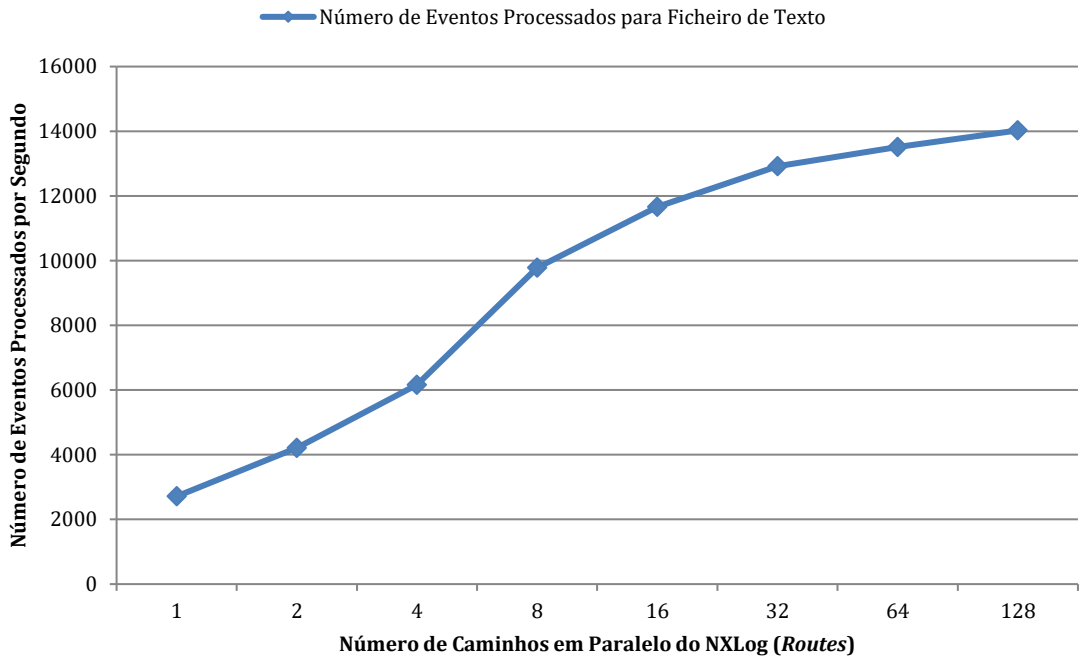


Figura 7.1 – Desempenho do NXLog

Conforme apresentado na Figura 7.1, através do número de eventos processados para ficheiro de texto, é possível verificar que a capacidade de processamento do *NXLog* aumenta de forma logarítmica, consoante o aumento do número de caminhos em paralelo. No caso da máquina onde foi efetuada a análise de desempenho do *NXLog*, que dispunha de 4 processadores virtualizados e 4GB de memória RAM, este aumento verificou-se até cerca de catorze mil eventos por segundo.

Desta forma, conclui-se que o *NXLog* é um *software* de gestão de eventos de elevado desempenho, e permite efetuar uma escalabilidade vertical através da adição de recursos de *hardware* que permitam aumentar o número de caminhos em paralelo.

7.3.2 Segundo Cenário – Avaliação do SIEM *Alienvault OSSIM*

Após apresentar os testes de desempenho do *NXLog* na secção anterior, para este cenário foram efetuados os mesmos testes, mas com os eventos a serem encaminhados para o SIEM *Alienvault OSSIM*, de forma a permitir determinar a capacidade de processamento de eventos do SIEM. Neste caso, foi ainda possível determinar o tempo que o *Alienvault OSSIM* demora a processar o evento, através da adição de um *timestamp* no *Raw* do evento, com o instante em que este foi enviado pelo *NXLog* para o *Alienvault OSSIM*.

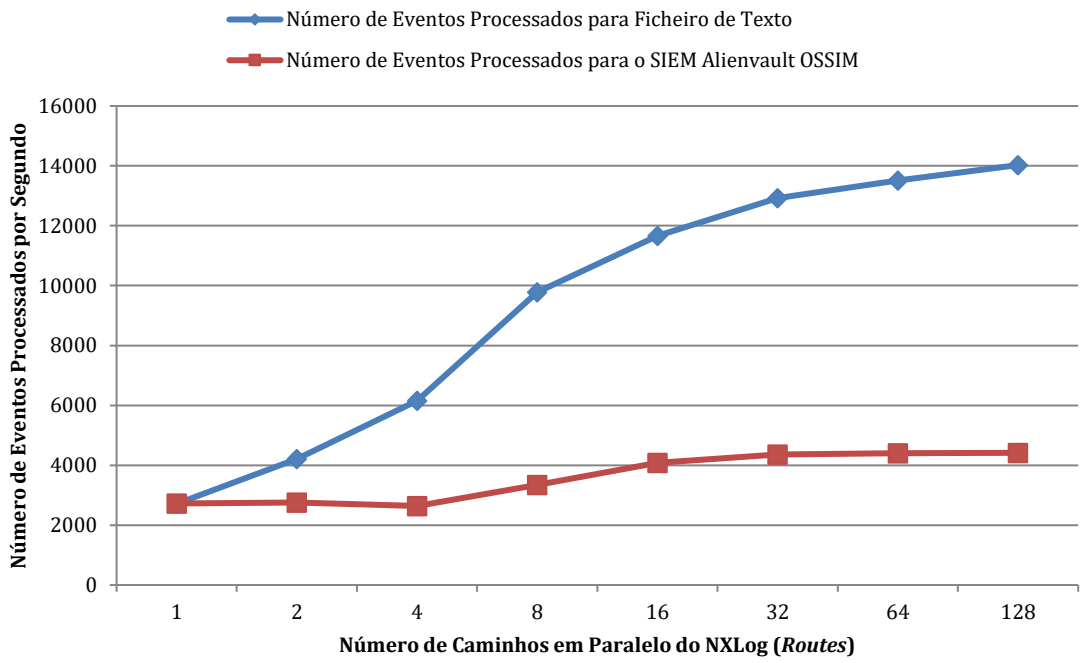


Figura 7.2 – Desempenho do Alienvault OSSIM Comparativamente ao NXLog

Ao encaminhar os eventos para o *Alienvault OSSIM*, de acordo com os resultados apresentados na Figura 7.2, não foi possível ultrapassar uma média de 5000 eventos processados por segundo, no entanto verificou-se um tempo de processamento médio de eventos bastante baixo (cerca de 100 milissegundos). Verificou-se ainda que o *Alienvault OSSIM* apresenta um limite de 1024 eventos por segundo após algum tempo de estar a processar uma taxa de eventos superior a este valor, de acordo com o apresentado na Figura 7.3, que é o mesmo limite apresentado para a versão base de subscrição do *Alienvault USM Appliance All-in-One*.

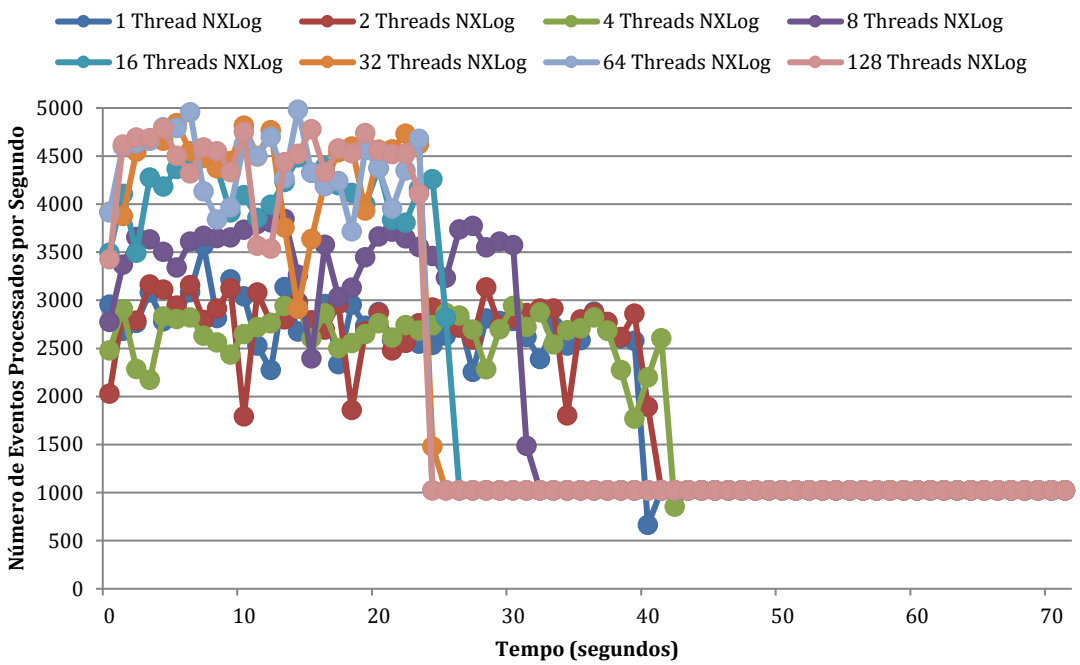


Figura 7.3 – Desempenho do Alienvault OSSIM

Desta forma, os dados apresentados na Figura 7.2 correspondem à média do número de eventos processados por segundo, e à média de tempo de processamento dos dados recolhidos no período que antecedeu a limitação dos 1024 eventos por segundo. Após o instante que *Alienvault OSSIM* passa a processar o máximo de 1024 eventos por segundo, caso o número de eventos recebidos se mantenha constante, o tempo de processamento dos eventos começará a aumentar de forma linear.

7.3.3 Terceiro Cenário – Avaliação do Ambiente WEC

A avaliação do ambiente *Windows Event Collection* dividiu-se em duas fases.

A primeira fase permitiu determinar a capacidade de reencaminhamento de eventos do ambiente WEC, através do cálculo do número máximo de eventos por segundo que chegam ao SIEM *Alienvault OSSIM*.

A segunda fase teve como objetivo determinar o atraso médio de encaminhamento de um evento desde a máquina onde é gerado, até chegar no SIEM *Alienvault OSSIM*.

Para determinar o número máximo de eventos por segundo que o servidor WEC consegue processar, foi configurada uma subscrição para o WEC receber os eventos que serão gerados nas máquinas WEF. Foi configurado de igual forma o *NXLog* para

reencaminhar os eventos recebidos para um ficheiro de texto com o *timestamp* do instante em que foi processado pelo *NXLog*.

Após a configuração da subscrição, e todos os WEF estarem efetivamente a encaminhar eventos para o WEC, foi desligada a conectividade de rede entre os WEF e o WEC para se proceder à geração de eventos. Neste ponto, foram gerados cerca de duzentos mil eventos em cada WEF.

Para monitorizar o ritmo máximo de envio de eventos, a ligação de rede dos WEF com o WEC foi reestabelecida no mesmo instante, e foi monitorizado o número de eventos que se encontravam a ser escritos no ficheiro por segundo, de acordo com a Figura 7.4.

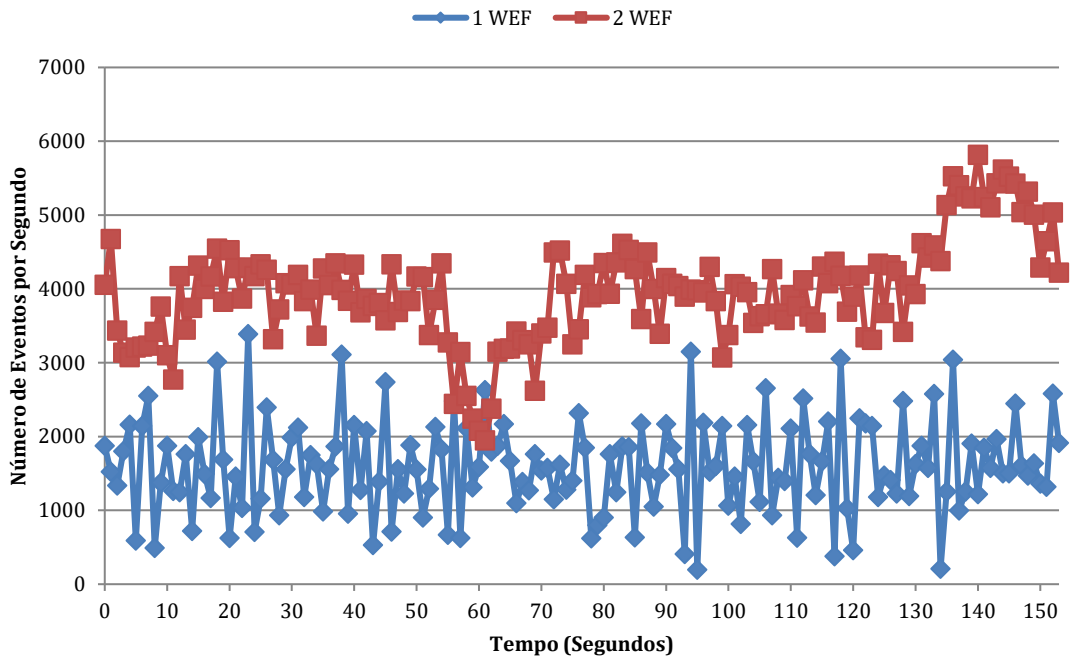


Figura 7.4 – Número Médio de Eventos Por Segundo Suportado Pelo WEC

Conforme apresentado na Figura 7.4, verifica-se que cada WEF envia em média cerca de 1600 eventos por segundo. Com o objetivo de determinar o atraso médio de encaminhamento dos eventos desde a máquina onde são gerados, até chegar ao SIEM *Alienvault OSSIM*, foi utilizada a mesma subscrição que permitiu determinar o número máximo de eventos por segundo, no entanto, em vez de se gerar os eventos com as máquinas WEF desconectadas do servidor WEC, foi executado o *script* de geração de eventos nas máquinas WEF com a ligação de rede estabelecida, para que os eventos fossem encaminhados em tempo real.

A Figura 7.5 apresenta o número total de eventos gerados por segundo pelas máquinas, assim como a tendência do tempo médio de atraso no encaminhamento de um evento, de forma a perceber quais os limites de processamento e de atraso existentes no ambiente WEC.

Como limitação do ambiente de laboratório onde foram efetuados os testes, constatou-se não ser possível gerar em tempo real mais de 1000 eventos por segundo, uma vez que a execução em paralelo de várias instâncias do script de geração de eventos causava a exaustão dos processadores da máquina sobre a qual está montado o ambiente de virtualização do laboratório. Desta forma, não foi possível determinar qual o comportamento do tempo de processamento de eventos para ritmos de eventos superiores a 1000 eventos por segundo.

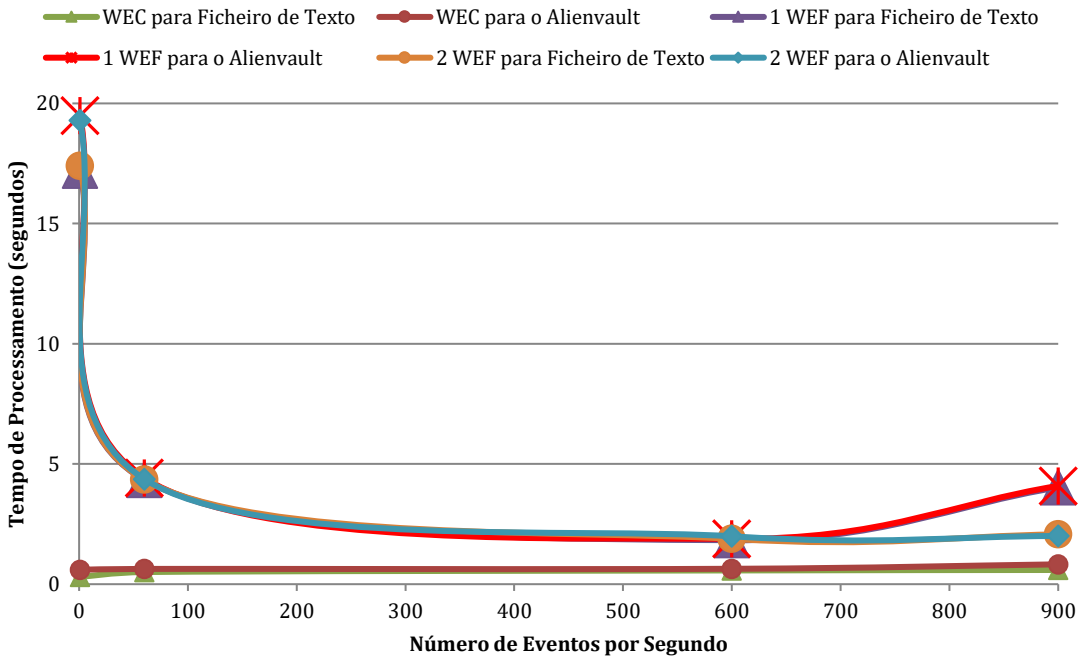


Figura 7.5 – Desempenho do Windows Event Collector

Conforme apresentado na Figura 7.5, verifica-se que em média os eventos gerados no WEC e reencaminhados pelo *NXLog* apresentam um tempo de processamento inferior a um segundo. Pelo que se pode concluir que o *NXLog* introduz um atraso pouco significativo no processamento de eventos, independentemente da taxa de eventos por segundo.

Para avaliar a capacidade de processamento do ambiente WEC, foram efetuados testes numa primeira fase com apenas um WEF a encaminhar eventos para o WEC, e numa segunda fase com dois WEF a encaminhar eventos para o WEC.

De acordo com a Figura 7.5, verificou-se que para taxas de eventos por segundo mais baixas, o ambiente *Windows Event Collection* apresenta tempos de processamento médio na ordem dos vinte segundos, e diminui gradualmente esse valor para taxas de eventos por segundo mais elevadas, até apresentar um atraso médio de cerca de três segundos. Este comportamento é explicável através das características de encaminhamento de eventos do ambiente WEC, que retém os eventos até que exista um conjunto de eventos para encaminhar de forma agrupada, minimizando assim a utilização de largura de banda, e pelas configurações da subscrição escolhida (*Minimum Latency*), que tem configurada uma latência máxima de trinta segundos.

Com os resultados apresentados na Figura 7.5, é possível ainda constatar que as estações WEF começam a apresentar um incremento de latência no encaminhamento dos eventos para o WEC, para taxas superiores a seiscentos eventos por segundo.

7.4 Conclusão

Após analisados os resultados, verifica-se que a solução apresentada permite atingir os objetivos propostos, tanto a nível funcional, como a nível de requisitos de desempenho em ambiente de laboratório.

Devido a limitações ao nível do CPU máximo disponibilizado para execução do ambiente de laboratório, e aos elevados recursos de CPU que os scripts necessitam, apenas foi possível gerar em tempo real 1000 eventos por segundo antes da estação *CentOS*, que serve de base ao sistema de virtualização, chegar ao ponto de exaustão dos processadores, o que não foi suficiente para testar os limites do servidor WEC, que segundo a recomendação da Microsoft, é de dez mil eventos por segundo.

A nível de desempenho do servidor WEC, e do *software* de gestão de eventos *NXLog*, verificou-se que mesmo com máquinas com características bastante inferiores às que foram propostas para implementação em ambiente de produção (secção 6.2.1), é possível atingir uma taxa de processamento de cerca de 4000 eventos por segundo, com um tempo médio de atraso bastante reduzido. A maior limitação é causada pelo SIEM implementado em ambiente de laboratório, que apenas permite processar uma taxa de

cerca de 1000 eventos por segundo. No entanto, o SIEM utilizado em laboratório é uma versão *open source* do SIEM *Alienvault USM* da Altice Portugal, que atualmente dispõe de um sensor com licença para processar 5000 eventos por segundo.

Capítulo 8

Conclusões e Trabalho Futuro

A integração de eventos de segurança das plataformas *MS Windows* num SIEM é um valioso recurso, que permite aumentar a capacidade de ciber inteligência, tornando a deteção de intrusões mais eficiente. A monitorização e correlação de eventos de segurança das plataformas *MS Windows* proporcionam à organização uma visão abrangente, em tempo real, do nível de segurança da sua infraestrutura, melhora o tempo de resposta a incidentes, assim como auxilia na sua mitigação.

Este projeto focou-se no processo de recolha, filtragem, tratamento e gestão centralizada de eventos das plataformas *MS Windows*, de forma a serem endereçados ao SIEM *Alienvault*, e na criação de um conjunto de regras de correlação que permitam ao SIEM gerar alertas de segurança relevantes.

Após a análise das diversas soluções para recolha de eventos de segurança das plataformas *MS Windows* apresentadas, a escolha recaiu no *Windows Event Collection*, que mostrou ser a solução menos intrusiva, uma vez que dispensa a instalação de agentes, é escalável, proprietária da *Microsoft*, e não necessita de licenciamento.

Adicionalmente, com o objetivo de proporcionar uma melhor monitorização e gestão do ambiente *Windows Event Collection* foi utilizado o *Supercharger* da *Logbinder*, que embora não seja requisito para a implementação de um ambiente WEC, permite melhorar a capacidade de gestão corporativa, melhorar a monitorização de estado, visualizar relatórios de funcionamento, e efetuar o balanceamento de carga automático.

De forma a permitir o envio dos eventos de segurança para o SIEM *Alienvault*, foi escolhido o *software* de gestão de eventos *NXLog Community Edition*, devido a ser um *software open source* de elevado desempenho, incluir suporte para recolha de eventos através do protocolo *Windows Eventlog*, e permitir configurar o seu envio no formato *syslog*, que é um dos protocolos suportados pelo SIEM *Alienvault*.

Após selecionadas as ferramentas e plataformas a utilizar para fazer chegar os eventos de segurança das plataformas *MS Windows* ao SIEM *Alienvault*, foi definida a melhor arquitetura a implementar em ambiente de produção, e desenvolvidos *plugins* e diretivas personalizadas no SIEM *Alienvault*, de forma a interpretar e correlacionar os eventos de segurança.

De seguida, foi efetuada a análise da solução proposta em ambiente de laboratório, uma vez que não foi possível a Altice Portugal disponibilizar, em tempo útil, os recursos necessários para efetuar a instalação da solução proposta em ambiente de produção. Desta análise, concluiu-se que a solução apresentada permite atingir os objetivos propostos, tanto a nível funcional, como a nível de desempenho, e apresenta uma elevada capacidade de escalabilidade.

Por último, para deteção de comportamentos anómalos através dos eventos de segurança das plataformas *MS Windows*, e criação de alarmística adequada, foram desenvolvidas regras de correlação no SIEM para deteção de três tipos de comportamentos: ataques de *bruteforce*; uso indevido de credenciais; e ataques de DoS a contas de utilizador.

Em suma, este projeto evidenciou ser ambicioso, complexo, abrangente e um processo de desenvolvimento contínuo, através da criação de regras de correlação no SIEM *Alienvault*, que permitam acrescentar novas funcionalidades de deteção de comportamentos anómalos.

8.1 Trabalho Futuro

No decorrer deste projeto foram identificadas algumas oportunidades de melhoria, que visam aprimorar a qualidade do produto final.

A primeira melhoria deve-se com o facto de a organização não ter conseguido disponibilizar em tempo útil os recursos necessários à implementação do projeto em ambiente de produção, aconselhando-se desta forma, a implementação do projeto em ambiente de produção assim que for viável.

Por forma a minimizar o número de falsos positivo e de falsos negativos gerados pelas regras de correlação desenvolvidas neste projeto, é aconselhável que após a sua implementação em ambiente de produção, se proceda a um período de experimentação

que permita afinar as regras de correlação de acordo com o ambiente presente na infraestrutura de rede da Altice Portugal.

Para melhorar a capacidade de deteção de comportamentos anómalos relativos a autenticações de domínio, e uma vez que os eventos deste tipo de autenticação são gerados na plataforma onde é feita a autenticação e no controlador de domínio, é vantajoso conseguir correlacionar os eventos de ambas as fontes, sem criar uma duplicação de alarmes. A correlação destes eventos permite identificar comportamentos anómalos dentro da própria rede, que possam indiciar um problema de falha no processo de encaminhamento de eventos, ou mesmo a existência de um ataque mais elaborado, que impeça as máquinas afetadas de encaminhar os eventos de segurança, como forma de ocultar a sua presença.

Por fim, este projeto não deve ser encarado como finito, mas como um processo contínuo de constante aperfeiçoamento, sendo recomendável o desenvolvimento regras de correlação que permitam identificar novas categorias de comportamentos suspeitos. Uma vez que o SIEM da Altice Portugal recebe eventos e *flows* de outras plataformas de segurança, aconselha-se a criação de regras que permitam correlacionar esses eventos com os eventos das plataformas *MS Windows*, com o objetivo de tornar a capacidade de deteção de intrusões mais efetiva.

Bibliografia

[1] Banco de Portugal, "Cibersegurança no sistema financeiro: riscos, cooperação e governação," 30 junho 2017. [Online]. Available: https://www.bportugal.pt/sites/default/files/anexos/documentos-relacionados/intervpub20170630.pdf. [Acedido em 29 setembro 2017].

[2] Centro Nacional de Cibersegurança (CNCS), Reação a Incidentes - Roadmap para a criação de capacidades mínimas, DN-PL – V.1.0, Lisboa.

[3] A. Madani, S. Rezayi e H. Gharaee, "Log management comprehensive architecture in Security Operation Center (SOC)," em *International Conference On Computational Aspects of Social Networks (CASoN)*, 2011.

[4] M. Rothman, "Continuous Security Monitoring: The Challenge of Full Visibility," 30 janeiro 2014. [Online]. Available: https://www.tripwire.com/state-of-security/security-data-protection/defining-continuous-security-monitoring-3/. [Acedido em 24 5 2018].

[5] T. H. Ptacek e T. N. Newsham, Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection, Secure Networks, Inc., 1998.

[6] Alienvault, "AlienVault Unified Security Management (USM) Platform," [Online]. Available: https://www.alienvault.com/products. [Acedido em 26 outubro 2017].

[7] J. Anjorin, "Is a SIEM Right For You?," 8 agosto 2017. [Online]. Available: https://www.sedarasecurity.com/is-a-siem-right-for-you/. [Acedido em 19 abril 2018].

[8] Gartner, "Security Information and Event Management (SIEM)," [Online].

Available: https://www.gartner.com/it-glossary/security-information-and-event-management-siem/. [Acedido em 19 abril 2018].

[9] KREEYAA, "Cyber Security," [Online]. Available: http://www.kreeyaa.com/cyber-security/. [Acedido em 24 maio 2018].

[10] TATA Cyber Security Community, "Journey of Security Incident & Event Management (SIEM)," TATA Consultancy Services, 27 outubro 2016. [Online]. Available: https://securitycommunity.tcs.com/infosecsoapbox/articles/2016/10/26/journey-security-incident-event-management-siem. [Acedido em 19 abril 2018].

[11] University of Houston-Clear Lake, "Security Information and Event Monitoring (SIEM)," [Online]. Available: https://www.uhcl.edu/computing/information-security/tips-best-practices/siem. [Acedido em 19 abril 2018].

[12] PCI Security Standards Council, *Information Supplement: Effective Daily Log Monitoring*, 2016.

[13] AlienVault, "USM Appliance System Overview," [Online]. Available: https://www.alienvault.com/documentation/usm-appliance/system-overview/system-overview.htm. [Acedido em 27 outubro 2017].

[14] Alienvault, "Event Collection, Processing, and Correlation Workflow," [Online]. Available: https://www.alienvault.com/documentation/usm-appliance/system-overview/about-usm-event-collection-processing-correlation.htm. [Acedido em 19 abril 2018].

[15] Alienvault, "About Plugins," [Online]. Available: https://www.alienvault.com/documentation/usm-appliance/plugin-management/about-plugins.htm. [Acedido em 19 abril 2018].

[16] ManageEngine, "The 8 most critical Windows security event IDs," [Online]. Available: https://download.manageengine.com/products/active-directory-audit/kb/the-eight-most-critical-windows-event-ids.pdf. [Acedido em 05 março

2018].

[17] B. Charter, “EVTX and Windows Event Logging,” SANS Institute, 2008.

[18] Microsoft, “Configuring Audit Policies,” [Online]. Available: https://technet.microsoft.com/en-us/library/dd277403.aspx. [Acedido em 15 abril 2018].

[19] Microsoft, “Introdução a Group Policy (GPO),” [Online]. Available: https://technet.microsoft.com/pt-br/library/cc668545.aspx. [Acedido em 15 abril 2018].

[20] “Agentless data collection,” [Online]. Available: https://en.wikipedia.org/wiki/Agentless_data_collection. [Acedido em 20 abril 2018].

[21] I. Koecher, “Agent vs Agentless: Why you should monitor (event) logs with an agent-based log monitoring solution,” 14 março 2017. [Online]. Available: https://www.eventsentry.com/blog/2017/03/agent-vs-agentless-why-you-should-monitor-event-logs-with-an-agent-based-log-monitoring-solution.html. [Acedido em 20 abril 2018].

[22] Logbinder, “Windows Event Collection,” [Online]. Available: https://support.logbinder.com/SuperchargerKB/50097/1-Windows-Event-Collection. [Acedido em 1 março 2018].

[23] Microsoft, “Subscrições de Eventos,” [Online]. Available: https://technet.microsoft.com/pt-PT/library/4aa6403f-d4b8-43a4-a70d-ceb7f88c524e. [Acedido em 16 outubro 2017].

[24] Microsoft, “Windows Event Collector (Windows),” [Online]. Available: https://msdn.microsoft.com/en-us/library/windows/desktop/bb427443(v=vs.85).aspx. [Acedido em 16 outubro 2017].

[25] Microsoft, “Operations Manager Documentation,” [Online]. Available: https://docs.microsoft.com/en-us/system-center/scom/?view=sc-om-1711. [Acedido em 17 novembro 2017].

[26] Microsoft, “Configure Windows telemetry in your organization (Windows 10),” [Online]. Available: https://docs.microsoft.com/en-us/windows/configuration/configure-windows-telemetry-in-your-organization. [Acedido em 17 novembro 2017].

[27] Microsoft, “Windows Analytics | Data-Driven insights that reduce the cost of deploying, servicing, and supporting Windows 10,” [Online]. Available: https://www.microsoft.com/en-au/windowsforbusiness/windows-analytics. [Acedido em 22 novembro 2017].

[28] NXLog, “NXLog Community Edition,” [Online]. Available: https://nxlog.co/products/nxlog-community-edition. [Acedido em 26 outubro 2017].

[29] R. Anthony, “Detecting Security Incidents Using Windows Workstation Event Logs,” SANS Institute, 2013.

[30] Logbinder, “Supercharger for Windows Event Collection - Supercharger Free,” [Online]. Available: https://www.logbinder.com/Products/Supercharger/. [Acedido em 1 março 2018].

[31] Logbinder, “Supercharger Architecture - Physical,” [Online]. Available: https://support.logbinder.com/SuperchargerKB/50098/2-Supercharger-Architecture-Physical. [Acedido em 1 março 2018].

[32] Logbinder, “Supercharger Architecture - Logical,” [Online]. Available: https://support.logbinder.com/SuperchargerKB/50100/4-Supercharger-Architecture-Logical. [Acedido em 1 março 2018].

[33] Logbinder, “Collectors,” [Online]. Available: https://support.logbinder.com/SuperchargerKB/50106/4-Collectors. [Acedido em 1

março 2018].

[34] Logbinder, “Supercharger for Windows Event Collection,” [Online]. Available: https://www.logbinder.com/blog?p=443d8a52-0f44-4fe0-ae66-f9382a1cf4fc. [Acedido em 1 março 2018].

[35] Logbinder, “Forwarder Analysis,” [Online]. Available: https://support.logbinder.com/SuperchargerKB/50105/3-Forwarder-Analysis. [Acedido em 1 março 2018].

[36] Logbinder, “Functionality,” [Online]. Available: https://support.logbinder.com/SuperchargerKB/37. [Acedido em 1 março 2018].

[37] Logbinder, “Load Balancing Many Forwarders Across Multiple Collectors,” [Online]. Available: https://support.logbinder.com/SuperchargerKB/50107/5-Load-Balancing-Many-Forwarders-Across-Multiple-Collectors. [Acedido em 1 março 2018].

[38] Alienvault, “Review Event Details,” [Online]. Available: https://www.alienvault.com/documentation/usm-appliance/events/event-details-fields.htm. [Acedido em 12 abril 2018].

[39] Portugal Telecom - CSIRT, 25 fevereiro 2015. [Online]. Available: https://conteudos.telecom.pt/Documents/EN/pt/security/portugal-telecom-csirt.pdf. [Acedido em 28 setembro 2017].

[40] J. Shenk, “SANS Seventh Annual Log Management Survey Report,” A SANS Whitepaper – April 2011.

[41] Microsoft, “Windows Events,” 2017. [Online]. Available: https://msdn.microsoft.com/en-us/library/windows/desktop/aa964766(v=vs.85).aspx. [Acedido em 16 outubro 2017].

Anexos

ANEXO A **– Campos dos Eventos do Alienvault**

Tabela A.1 – Campos de Eventos do Alienvault (extraído de [38])

Campos de Dados	Descrição
Date	Data e hora do evento.
AlienVault Sensor	Sensor que processou o evento
Device IP	Endereço IP do Sensor *USM Appliance* que processou o evento.
Event Type ID	ID atribuído pelo *USM Appliance* para identificar o tipo de evento.
Unique Event ID#	Número de identificação único atribuído ao evento pelo *USM Appliance*.
Protocol	Protocolo utilizado pelo ativo de origem/destino do evento. Por exemplo, TCP IP.
Category	Taxonomia do evento, por exemplo, *Authentication* ou *Exploit*.
Sub-Category	Subcategoria da taxonomia do evento. Por exemplo, *Denial of Service*, dentro da categoria *Exploit*.
Data Source Name	Nome da aplicação externa que produziu/enviou o evento para o *USM Appliance*.
Data Source ID	ID associado à aplicação externa que produziu o evento.
Product Type	Tipo de produto que gerou a taxonomia do evento. Por exemplo, *Operating System* ou *Server*. **Nota:** Eventos relacionados com reputação de IP têm *Product Types*, enquanto eventos do *OTX pulse* não têm.
Additional Info	Se o evento for gerado por um URL suspeito, por exemplo, este campo indica o URL. Quando presente, esse URL fornece informação adicional e referências sobre os componentes associados ao evento.
Priority	Nível de prioridade, baseado no valor do tipo de evento. Cada tipo de evento tem um valor de *Priority*, que é utilizado para calculo do risco.
Reliability	Nível de confiabilidade, baseado no valor da confiabilidade do *Event Type*. Cada tipo de evento tem um valor de *Reliability*, que é utilizado para calculo do risco.

Risk	Nível de Risco do event0: Baixo = 0, Médio = 1, Alto > 1 **Nota:** O cálculo do Risco é baseado na seguinte fórmula: Valor do ativo * *Reliability* do evento * *Priority* do evento / 25 = *Risk*
OTX Indicators	Número de indicadores associados com reputação de IP, ou eventos *OTX pulse*.
Source / Destination	Endereço IP e nome da origem e destino, respetivamente, do evento. Se a origem, ou o destino for um ativo, é possível clicar com o botão direito sobre ele e aceder a informação detalhada sobre esse ativo.
Hostname	Nome do ativo de origem/destino do evento. Se o nome do ativo estiver registado na lista de *Assets*, ao clicar no ativo é possível aceder à sua informação detalhada.
MAC Address	*Media Access Control* (MAC) do ativo, se disponível.
Port	Porta externa ou interna de Origem/destino do evento.
Latest Update	Última vez que foi atualizada as propriedades do ativo pelo *USM Appliance*.
Username & Domain	Nome de Utilizador e Domínio associados ao ativo, que gerou o evento.
Asset Value	Valor do ativo, se estiver definido na lista de ativos.
Location	Se o país do ativo de origem/destino for conhecido, mostra a bandeira nacional respetiva.
Context	Se o ativo pertencer a um grupo de entidades definido pelo utilizador, o *USM Appliance* exibe o contexto.
Asset Groups	Quando o ativo de origem/destino do evento pertence a uma, ou mais, grupos de ativos que se encontrem configurados, este campo mostra a lista de grupos a que o ativo pertence.
Networks	Quando o ativo de origem/destino do evento pertence a uma, ou mais, redes que se encontrem configuradas, este campo mostra a lista de redes a que o ativo pertence.
Logged Users	Lista de utilizadores que estão/estiveram conectados no ativo. Pode ser detectado através de um *scan* ao ativo.
OTX IP Reputation	(*Yes*/*No*) Quando exista ou não reputação de IP que identifique o IP como suspeito.
Service	Lista de serviços ou aplicações detetadas na porta de origem/destino.
Port	Porta usada pelo serviço ou aplicação.
Protocol	Protocolo usado pelo serviço ou aplicação.
Raw Log	Informação do evento no formato *Raw*.
Filename	Nome do ficheiro associado ao evento.
Username	Nome de utilizador associado ao evento.
Password	Password associada ao evento.
Userdata 1-9	Campos de dados customizados pelo utilizador.
Payload	Conteúdo do evento.
Rule Detection	Regra do *AlienVault NIDS* que detetou o evento.

ANEXO B – **Instalação do *Supercharger Manager***

Para proceder à instalação do *Supercharger Manager*, deve-se descarregar o instalador a partir da página web da *Logbinder*[1]. É necessário ter em atenção que a máquina onde será instalado o *Supercharger Manager* tem que pertencer a um domínio, e a instalação deverá ser feita com uma conta com privilégios de administração, uma vez que essa conta será adicionada automaticamente ao grupo de Administradores do *Supercharger*. No caso de esta ter, ou ter tido instalado o *SQL Server 2016*, deverá ser contactado o suporte[2] da *Logbinder* antes de proceder à instalação.

Ao executar o instalador, irá aparecer uma janela, de acordo com a Figura B.1.

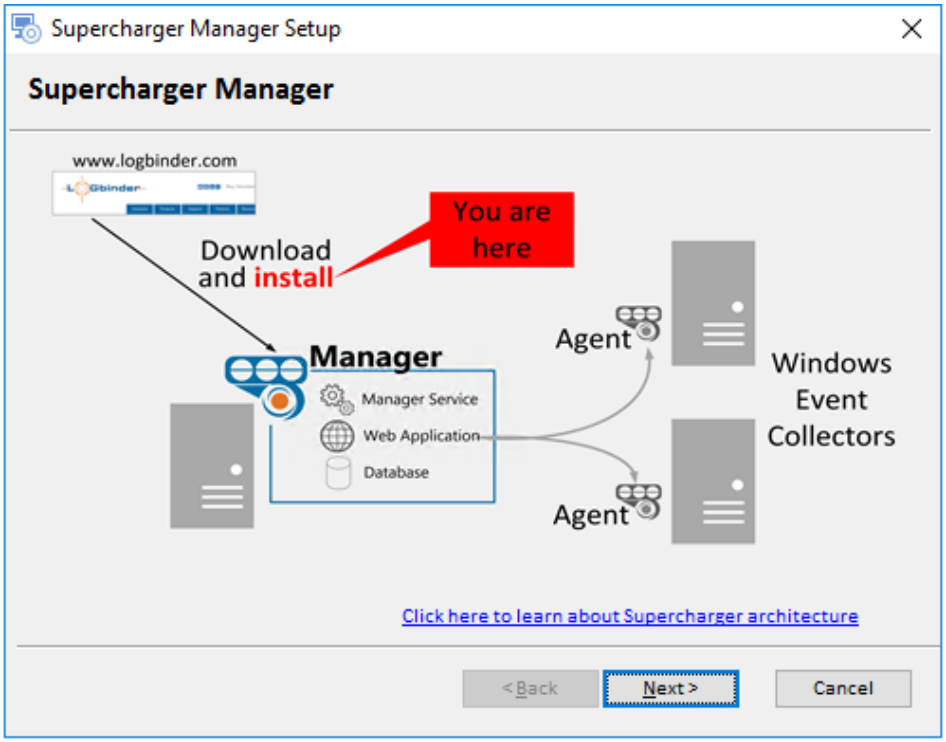


Figura B.1 – Instalação do Supercharger Manager (Passo 1 de 5)

[1] https://www.logbinder.com/form/scdownload (Versão Gratuita)
[2] https://forum.logbinder.com/

Ao pressionar "*Next*", irá aparecer uma janela, de acordo com a Figura B.2, onde são selecionados os pré-requisitos a instalar.

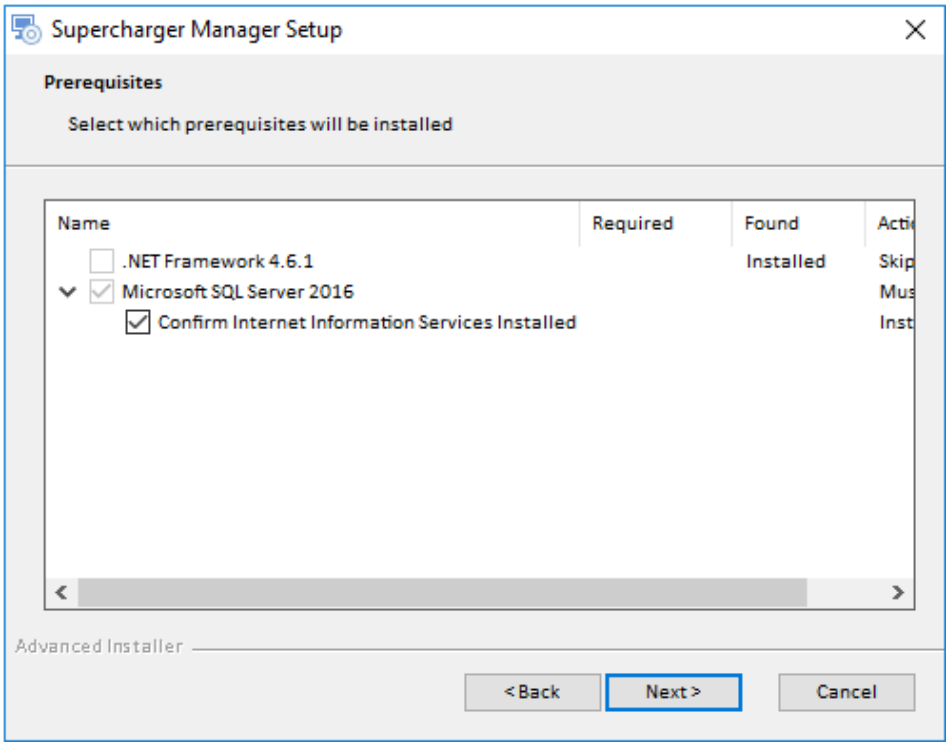


Figura B.2 – Instalação do Supercharger Manager (Passo 2 de 5)

De seguida, é necessário aceitar os termos da licença, conforme Figura B.3.

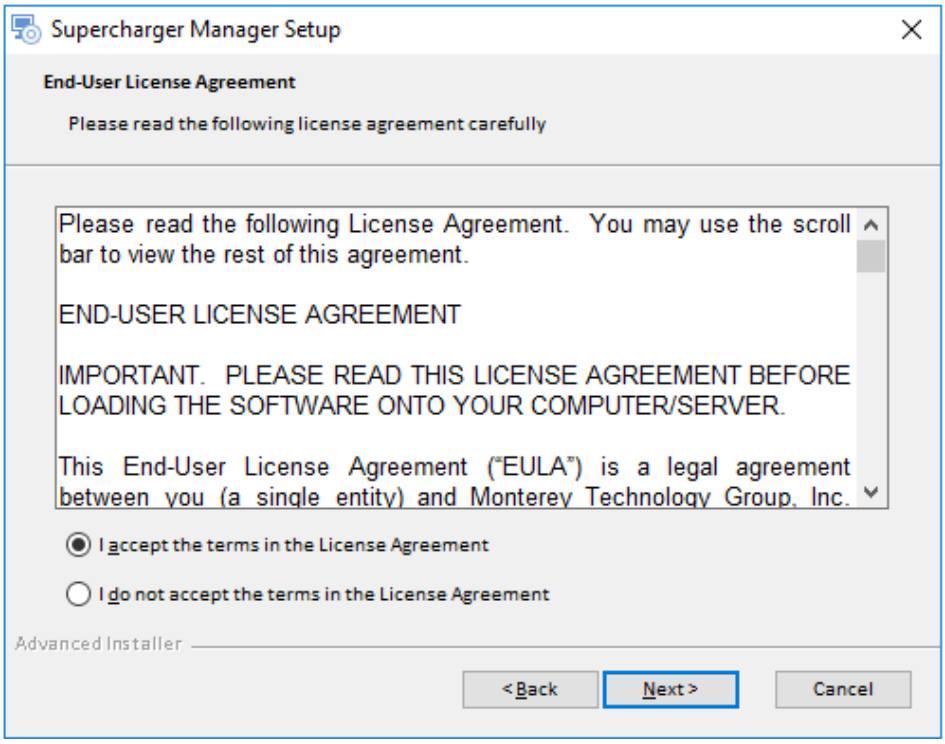


Figura B.3 – Instalação do Supercharger Manager (Passo 3 de 5)

Após a instalação dos pré-requisitos e configuração da instalação do *Supercharger* Manager, inicia-se a instalação do *Supercharger* ao pressionar "*Install*" (Figura B.4).

Figura B.4 – Instalação do Supercharger Manager (Passo 4 de 5)

Concluída a instalação com sucesso, é apresentada a janela conforme Figura B.5.

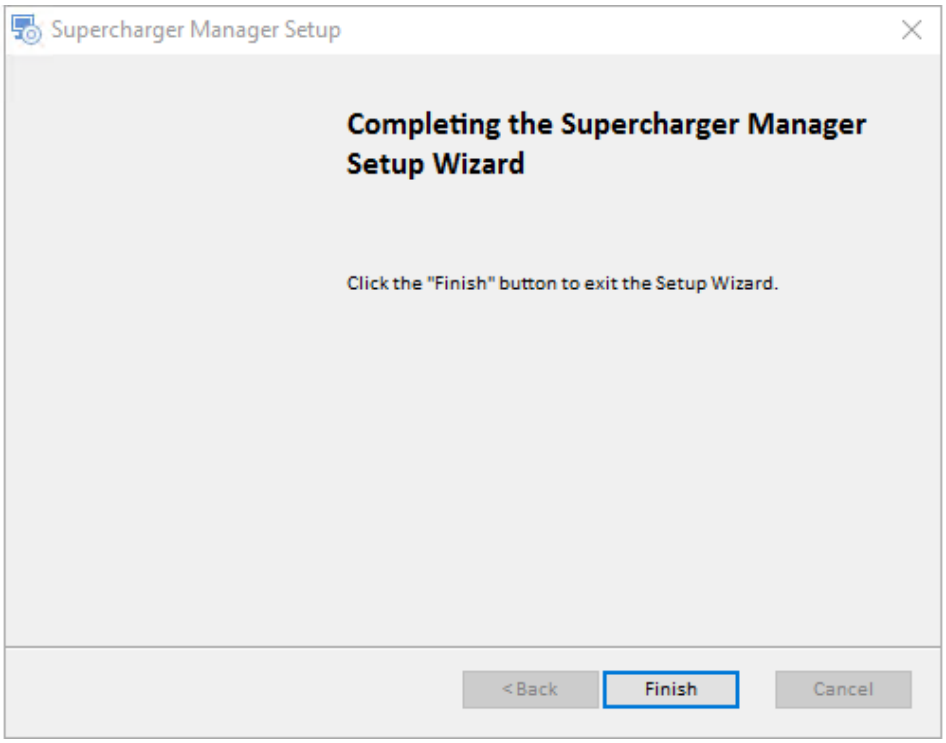


Figura B.5 – Instalação do Supercharger Manager (Passo 5 de 5)

A partir deste ponto as configurações são efetuadas a no interface web (ver Figura B.6), no endereço http://localhost/Supercharger/, que deverá ser aberto automaticamente pelo programa de instalação.

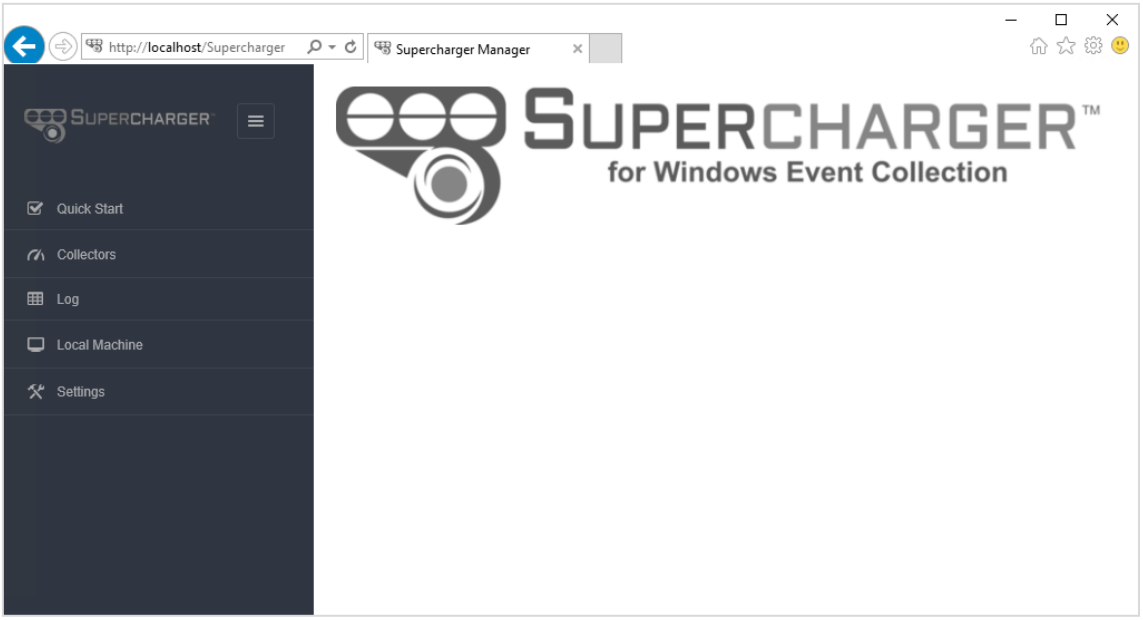


Figura B.6 – Interface Web do Supercharger Manager

ANEXO C – Configuração dos Coletores de Eventos do Windows

Para configuração dos Coletores de Eventos do Windows (WEC), devem ser seguidos os seguintes passos:

1. *Instalação do Supercharger Controller*

No caso de existir uma instância do *Supercharger Manager* ativa, e se pretenda que o WEC seja gerido por este, deve-se (a partir do WEC) aceder ao URL http://<Supercharger_FQDN>/Supercharger/LocalMachine (ver Figura C.1) e executar os seguintes procedimentos:

1.1. Descarregar e executar o ficheiro de instalação do *Supercharger Controller*.

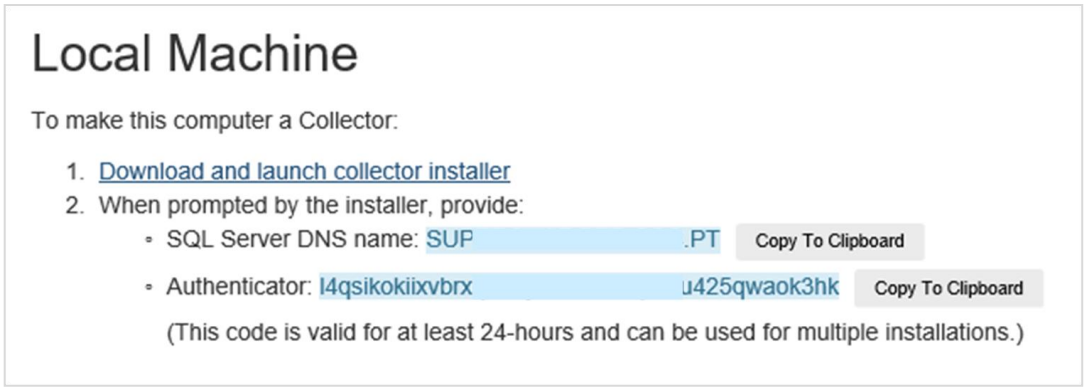


Figura C.1 – Página Web do Supercharger

1.2. Inserir o nome do servidor SQL, e o código de autenticação fornecidos, quando for solicitado pelo instalador (conforme Figura C.2). O código de autenticação disponibilizado é válido por um período de 24 horas.

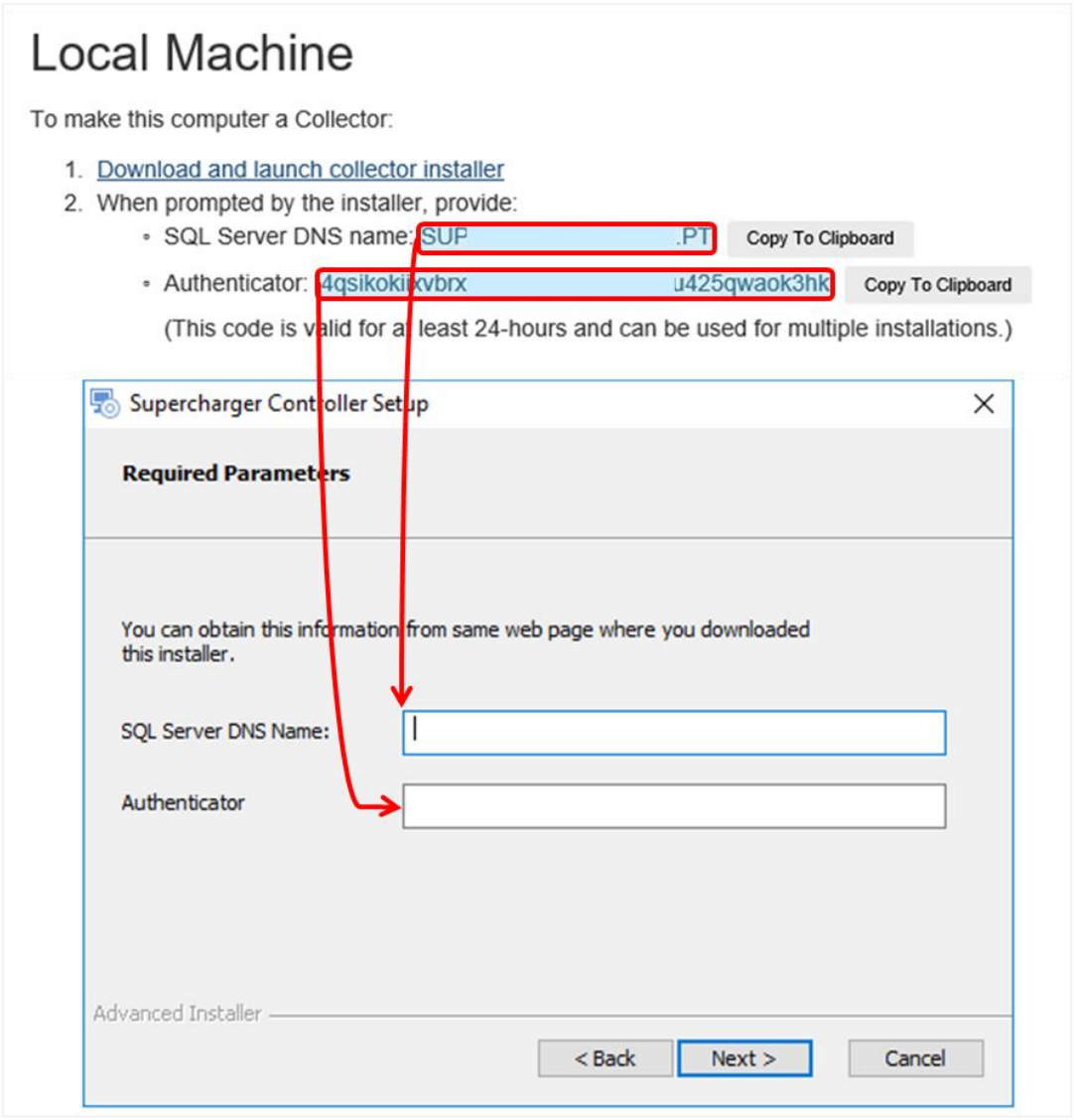


Figura C.2 – Instalação do Supercharger Controller

1.3. Aceder ao URL http://<Supercharger_FQDN>/Supercharger/Collectors, e selecionar *"Approve"* (conforme Figura C.3), para autorizar o *Supercharger Manager* a gerir o WEC.

Figura C.3 – Aprovação de um Coletor do Windows

NOTA: Após efetuar este passo não é necessário efetuar os passos 2 e 3, no entanto podem ser efetuados para confirmar se está tudo corretamente configurado.

2. *Configurar o Serviço de Gestão Remota do Windows*

Para configurar o serviço de Gestão Remota do *Windows* (*WinRM*) é necessário abrir uma janela de comando com privilégios de Administrador (*RunAs Administrator*), e executar o comando "*winrm qc"*. De seguida, caso o serviço não esteja configurado, deverão surgir duas questões, que deverão ser respondidas de forma afirmativa, conforme Figura C.4.

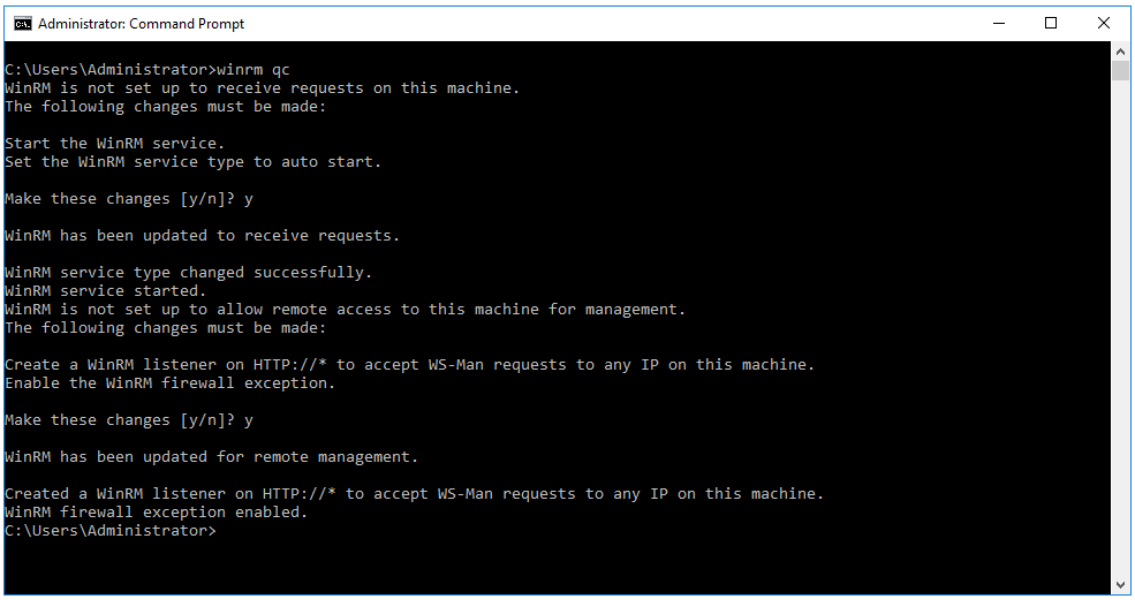


Figura C.4 – Configuração do Serviço de Gestão Remota do Windows

3. *Configurar o Serviço do Coletor de Eventos do Windows*

Para configurar o serviço *Coletor de Eventos do Windows* e garantir que é possível criar subscrições, e manter as configurações depois de reiniciar o sistema operativo, é necessário abrir uma janela de comando com privilégios de Administrador (*RunAs Administrator*), e executar o comando "*wecutil qc*". Caso o serviço não esteja configurado, deverá surgir uma questão, que deverá ser respondida de forma afirmativa, conforme Figura C.5.

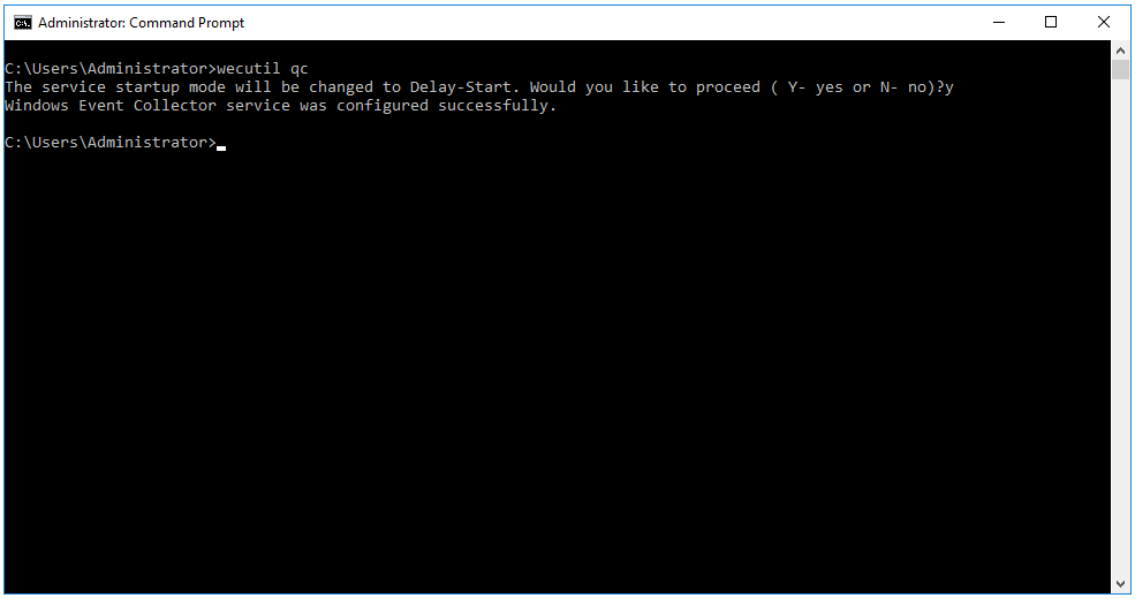


Figura C.5 – Configuração do serviço Coletor de Eventos do Windows

4. *Configuração do WEC para utilização do Protocolo HTTPS (Opcional)*

Para configuração do protocolo de comunicação seguro, através de HTTPS, é necessário o servidor WEC ter instalado um certificado de autenticação de servidor (*Server Authentication*), e efetuar os seguintes procedimentos:

4.1. Abrir uma janela de comando com privilégios de Administrador (*RunAs Administrator*), e executar o comando "*winrm create winrm/config/listener?Address=*+Transport=HTTPS @{Hostname="<WEC_FQDN_hostname>";CertificateThumbprint="<WEC_Server_Authentication_Certificate_Thumbprint>";Port="5986"}*".

O comando "*winrm enumerate winrm/config/listener*", conforme Figura C.6, permite confirmar se o WEC está corretamente configurado para receber os eventos do *Windows* através de HTTPS.

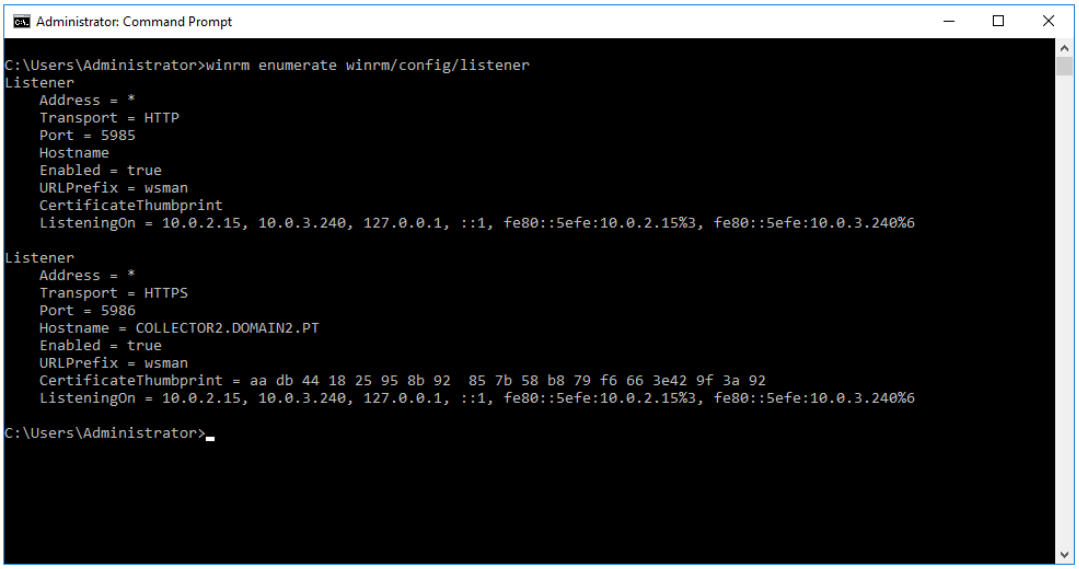


Figura C.6 – Enumerar as Configurações dos Canais em Escuta pelo WEC

4.2. Configurar as subscrições para utilização do Protocolo HTTPS. Para tal, é necessário abrir o visualizador de eventos do *Windows*, aceder às propriedades da subscrição desejada, clicar no botão "*Advanced…*" e selecionar o protocolo HTTPS, conforme Figura C.7.

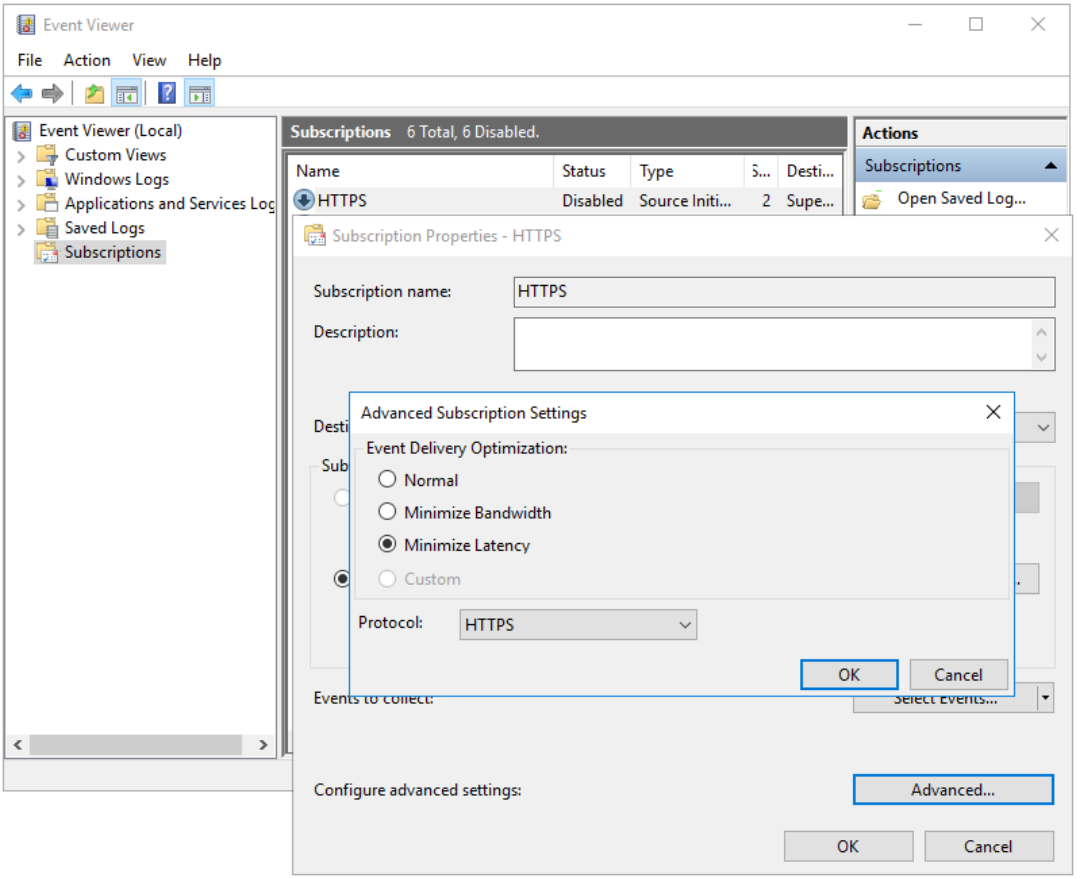


Figura C.7 – Configurar uma Subscrição para Utilizar o Protocolo HTTPS

ANEXO D – **Instalação e Configuração do *NXLog Community Edition***

Para proceder à instalação e configuração do *NXLog Community Edition*, para reencaminhar os eventos do log de eventos reencaminhados para o *Alienvault*, devem ser seguidos os seguintes passos:

1.4. Descarregar e instalar a versão mais atualizada do *NXLog Community Edition*[1] disponibilizada na página web da *NXLog*.

1.5. Descarregar o ficheiro *patterndb.xml*[2] disponibilizado na página web da *Alienvault*, e colocar em "*C:\Program Files (x86)\nxlog\conf\patterndb.xml*".

1.6. Criar o ficheiro *nxlog.conf*, com o conteúdo descrito a baixo, em "*C:\Program Files (x86)\nxlog\conf\nxlog.conf*".

1.7. Alterar no ficheiro *nxlog.conf* o *OUTPUT_DESTINATION_ADDRESS* para corresponder ao endereço IP do *Alienvault*, e o *OUTPUT_DESTINATION_PORT* para corresponder à porta pela qual vai ser efetuada a comunicação (por omissão é a 514).

1.8. Reiniciar o serviço *NXLog*.

Conteúdo do ficheiro nxlog.conf

```
define ROOT C:\Program Files (x86)\nxlog
define LOGFILE %ROOT%\data\nxlog.log

define OUTPUT_DESTINATION_ADDRESS XXX.XXX.XXX.XXX
define OUTPUT_DESTINATION_PORT XXX

Moduledir %ROOT%\modules
CacheDir %ROOT%\data
Pidfile %ROOT%\data\nxlog.pid
SpoolDir %ROOT%\data
```

[1] https://nxlog.co/products/nxlog-community-edition/download
[2] https://www.alienvault.com/documentation/resources/downloads/usm-appliance/patterndb.xml

```
LogFile %ROOT%\data\nxlog.log


##
## Extensions:
##

## Support character conversions:
<Extension charconv>
    Module          xm_charconv
</Extension>

##
## Inputs:
##

## This nxlog servers heartbeat:
## MarkInterval defines the interval in minutes of the heartbeat-
messages.
## Mark defines the text which is sent.
<Input in_nxlog_heartbeat>
    Module          im_mark
    MarkInterval    10
    Mark            The nxlog service is alive.
    Exec            $EventType = 'Application'; $Channel = 'nxlog-ce';
$EventID = 8347;
</Input>

## Windows event log:
<Input in_windows_events>
    Module          im_msvistalog
    SavePos         FALSE
    ReadFromLast    TRUE

    # Limit the log forwarding to collected events:
    Query          <QueryList>                                          \
                       <Query Id='0' Path='ForwardedEvents'>            \
                           <Select Path='ForwardedEvents'>*</Select> \
                       </Query>                                         \
                   </QueryList>
</Input>

##
## Transformation:
##

## Custom CSV format for nxlog and sysmon-nxlog plugin.
<Extension transform_alienvault_csv>
    Module          xm_csv
    Fields          $EventTime, $EventType, $Severity, $Channel,
$Hostname, $EventID, $SourceName, $AccountName, $AccountType, $Domain,
$Message
    FieldTypes      string, string, string, string, string, string,
string, string, string, string, string
    Delimiter       ;
</Extension>

##
## Filters:
##
```

```
## Match events by Windows event ID.
## This sets $PatternID in case it matches.
<Processor match_events>
    Module          pm_pattern
    PatternFile     %ROOT%\conf\patterndb.xml
</Processor>

##
## Outputs:
##

## Process and forward Windows logs:
<Output out_alienvault_csv>
    Module          om_udp
    Host            %OUTPUT_DESTINATION_ADDRESS%
    Port            %OUTPUT_DESTINATION_PORT%

    # If the EventID doesn't exist in 'patterndb.xml' it gets dropped:
    Exec            if not defined $PatternID or not defined $Message
{ drop(); }

    # Replace newlines, tabs and carriage returns with blanks:
    Exec            $Message = replace($Message, "\t", " "); $Message
= replace($Message, "\n", " "); $Message = replace($Message, "\r", "
");

    # Ensure that commonly undefined values are set:
    Exec           if not defined $AccountName { $AccountName = "-"; }
    Exec           if not defined $AccountType { $AccountType = "-"; }
    Exec           if not defined $Domain { $Domain = "-"; }

    # Ensure we send in the proper format:
    Exec            transform_alienvault_csv->to_csv(); $raw_event =
$Hostname + ' WIN-NXLOG ' + $raw_event;
</Output>

## Output internal nxlog messages:
<Output out_alienvault_nxlog_csv>
    Module          om_udp
    Host            %OUTPUT_DESTINATION_ADDRESS%
    Port            %OUTPUT_DESTINATION_PORT%

    Exec            if not defined $Message { drop(); }

    # Replace newlines, tabs and carriage returns with blanks:
    Exec            $Message = replace($Message, "\t", " "); $Message
= replace($Message, "\n", " "); $Message = replace($Message, "\r", "
");

    # Ensure that commonly undefined values are set:
    Exec           if not defined $AccountName { $AccountName = "-"; }
    Exec           if not defined $AccountType { $AccountType = "-"; }
    Exec           if not defined $Domain { $Domain = "-"; }

    # Ensure we send in the proper format:
    Exec            transform_alienvault_csv->to_csv(); $raw_event =
$Hostname + ' WIN-NXLOG ' + $raw_event;
</Output>

##
## Routes:
```

```
##

## Route for Windows logs:
<Route route_windows_logs>
    Path            in_windows_events => match_events =>
out_alienvault_csv
</Route>

## Route for internal nxlog heartbeat messages:
<Route route_nxlog_messages>
    Path            in_nxlog_internal, in_nxlog_heartbeat =>
out_alienvault_nxlog_csv
</Route>
```

ANEXO E – **Políticas de Grupo**

Para concretização do Ambiente de Laboratório, foram criadas as seguintes Políticas de Grupo (GPO), que permitam uma gestão centralizada das fontes de eventos do Windows (WEF).

Configuração do Serviço WinRM

Para configuração do serviço *WinRM* nas máquinas com sistema operativo *Windows XP SP2+* e *Windows Server 2003 SP1+*, é necessário garantir que o pacote de atualização KB968930, que contém a versão 2.0 do *WinRM*, se encontra instalado. As máquinas com o sistema operativo superior ao *Windows Vista* e *Windows Server 2008* já têm a versão 2.0 do *WinRM* instalada por omissão. O arranque automático do serviço *WinRM* pode ser configurado através da GPO "*Computer Configuration\Preferences \Control Panel Settings\Services"*, onde é necessário criar um novo serviço e configurar conforme Figura E.1.

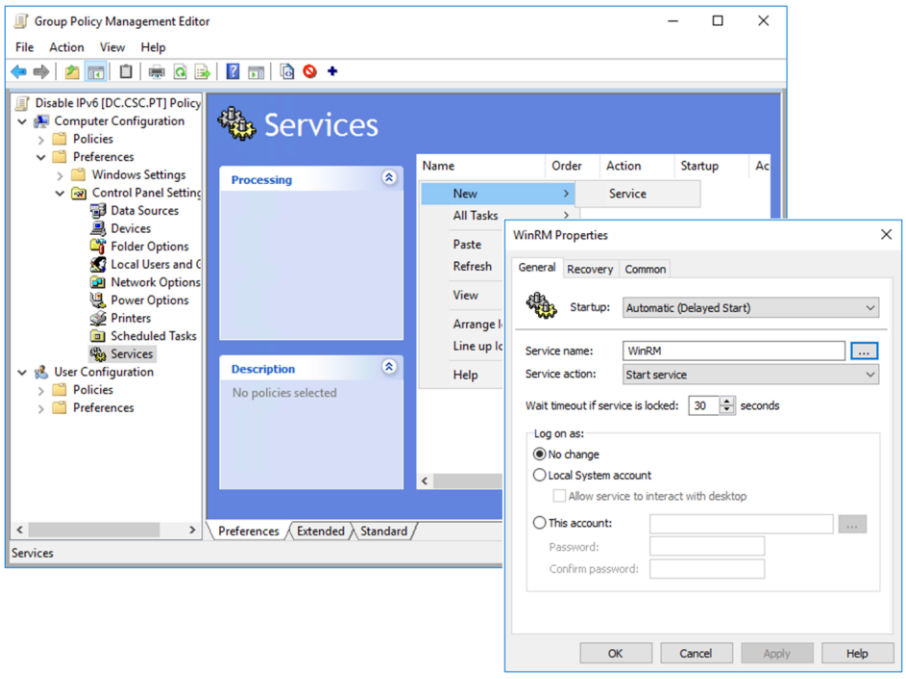


Figura E.1 – Configuração da GPO de Ativação Automática do WinRM

Permissões de Acesso ao Log de Eventos de Segurança do Windows

Para o serviço *WinRM* poder recolher eventos de segurança do Windows, é necessário dar permissões de acesso ao *log* de segurança de cada WEF. Através da criação da GPO "*Computer Configuration\Policies\Administrative Templates\Windows Components\Event Log Service\Security\Configure Log Access"* (conforme Figura E.2), é possível configurar esta permissão de forma centralizada.

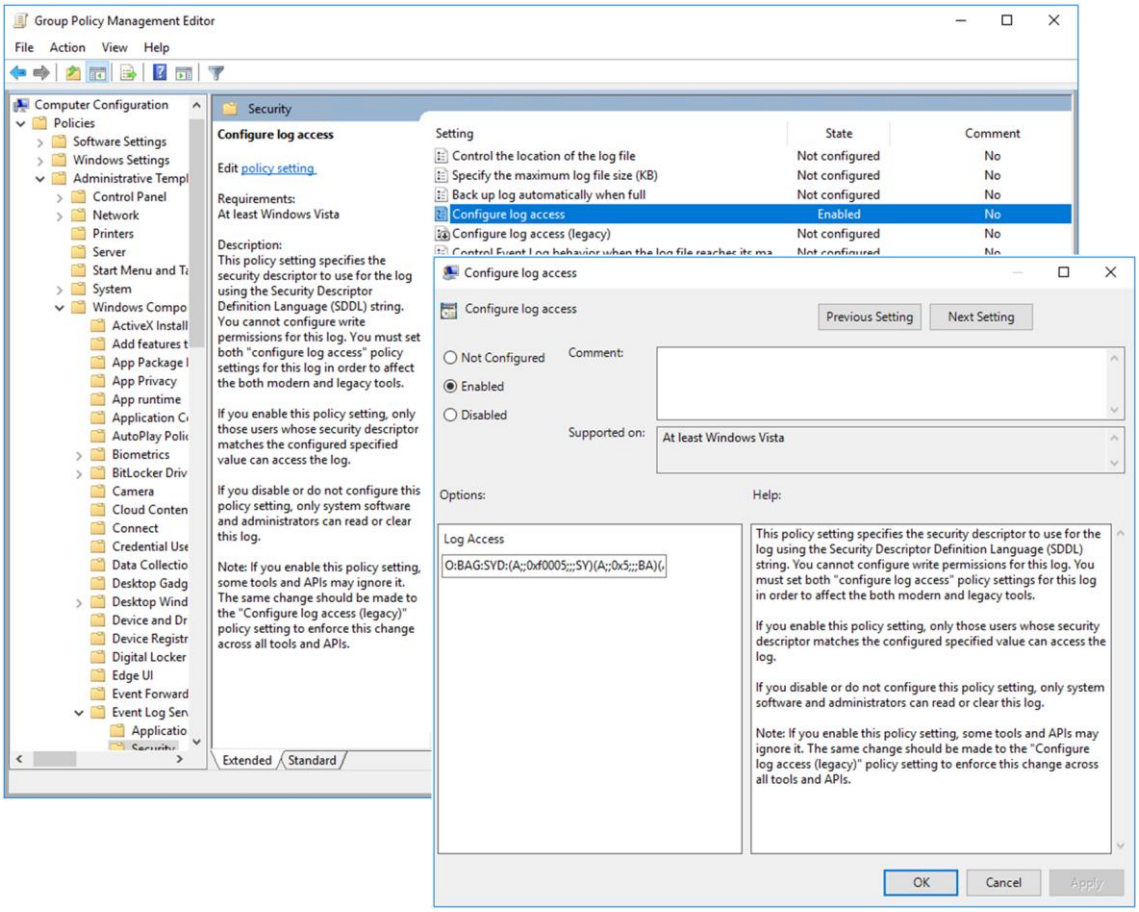


Figura E.2 – Configuração de Permissões de Acesso ao Log de Segurança

Devido às diferenças existentes entre os diversos sistemas operativos da Microsoft, é necessário criar uma GPO para as máquinas com sistema operativo *Windows XP SP2+* e *Windows Server 2003 SP1+*, com o valor do *Log Access* "*O:BAG:SYD:(A;;CC;;;NS)*", e outra GPO para as máquinas com o sistema operativo superior ao *Windows Vista* e *Windows Server 2008*, com o valor do *Log Access* "*O:BAG:SYD:(A;;0xf0005;;;SY)(A;;0x5;;;BA)(A;;0x1;;;S-1-5-32-573)(A;;0x1;;;NS)*".

Criação de Regras de Firewall

Para garantir o correto funcionamento do ambiente WEC é necessário garantir que a *firewall* dos WEF permite comunicação de entrada nas portas 5985 (*WinRM* HTTP), 5986 (*WinRM* HTTPS) e 80 (Modo de Compatibilidade do *WinRM*). Esta configuração pode ser aplicada através da GPO "*Computer Configuration\Policies\Windows Settings\Security Settings\ Windows Firewall with Advanced Security\Inbound Rules*", conforme Figura E.3.

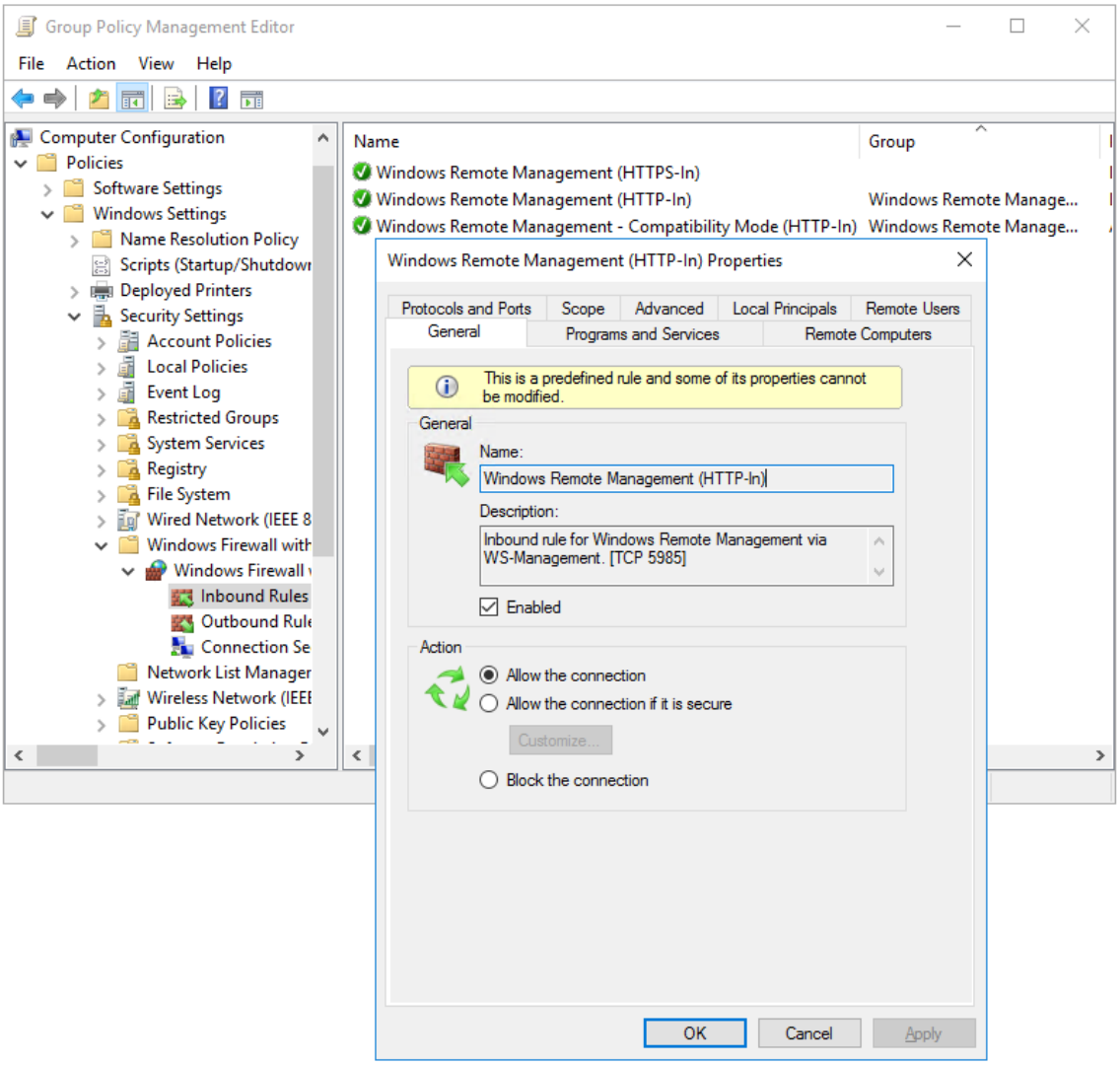


Figura E.3 – Configuração de Regras de Firewall

Configuração do WEC Para Onde o WEF Encaminha os Eventos

Para que as fontes de eventos recebam as subscrições e encaminhem os eventos para o seu respetivo Coletor de Eventos, é necessário configurar o Coletor de Eventos

de destino em cada WEF, sendo que esta configuração pode ser feita através da GPO "*Computer Configuration\Policies\Administrative Templates\Windows Components\Event Forwarding\Configure Target Subscription Manager*", no formato "*Server=http://<WEC_FQDN>:5985/wsman/SubscriptionManager/WEC*", conforme Figura E.4.

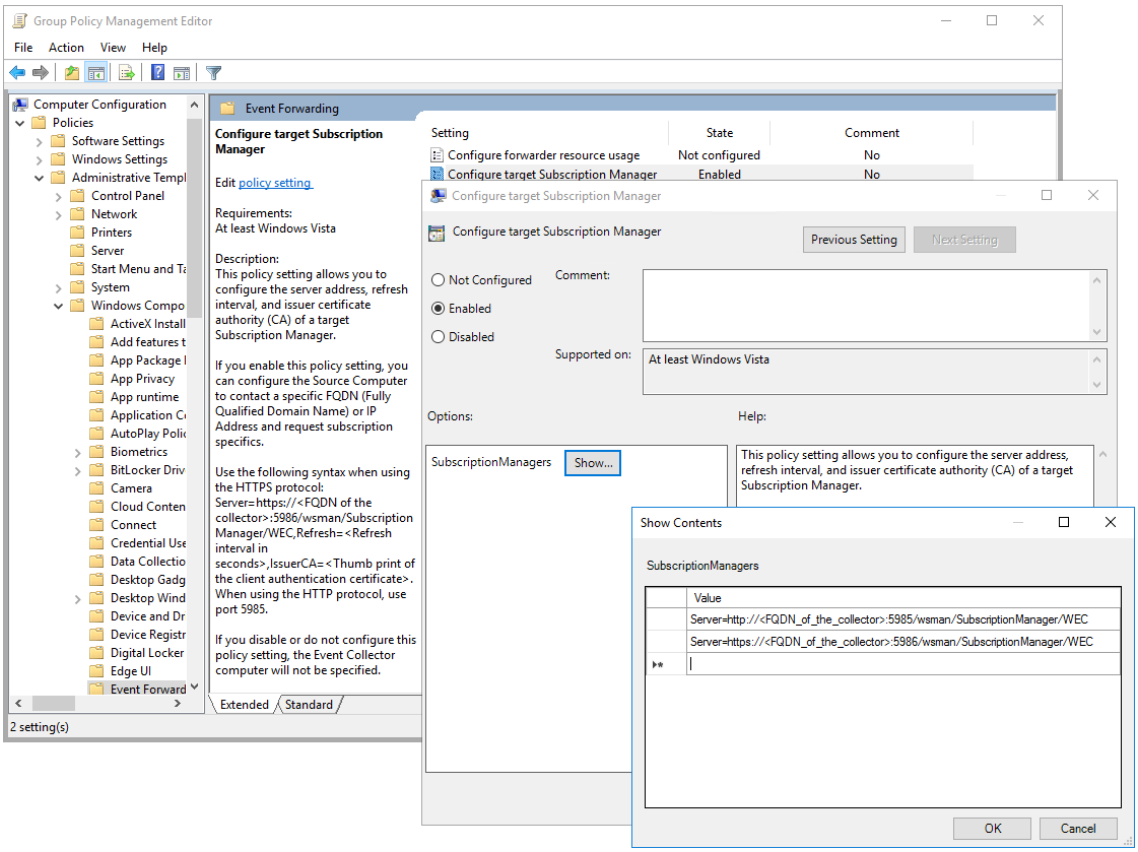


Figura E.4 – Configuração do WEC Gestor de Subscrição

Configuração do Nível de Auditoria do Windows

A configuração do nível de auditoria do *Windows* pode ser feita através da GPO "*Computer Configuration\Policies\Windows Settings\Security Settings\Advanced Audit Policy Configuration\Audit Policies*", conforme Figura E.5.

As políticas de auditoria dividem-se em grupos, cada um composto por diversas subcategorias, que uma vez habilitadas são responsáveis por auditar um conjunto de eventos. Estas políticas são utilizadas para definir os tipos de eventos que serão gerados e armazenados nos *logs*.

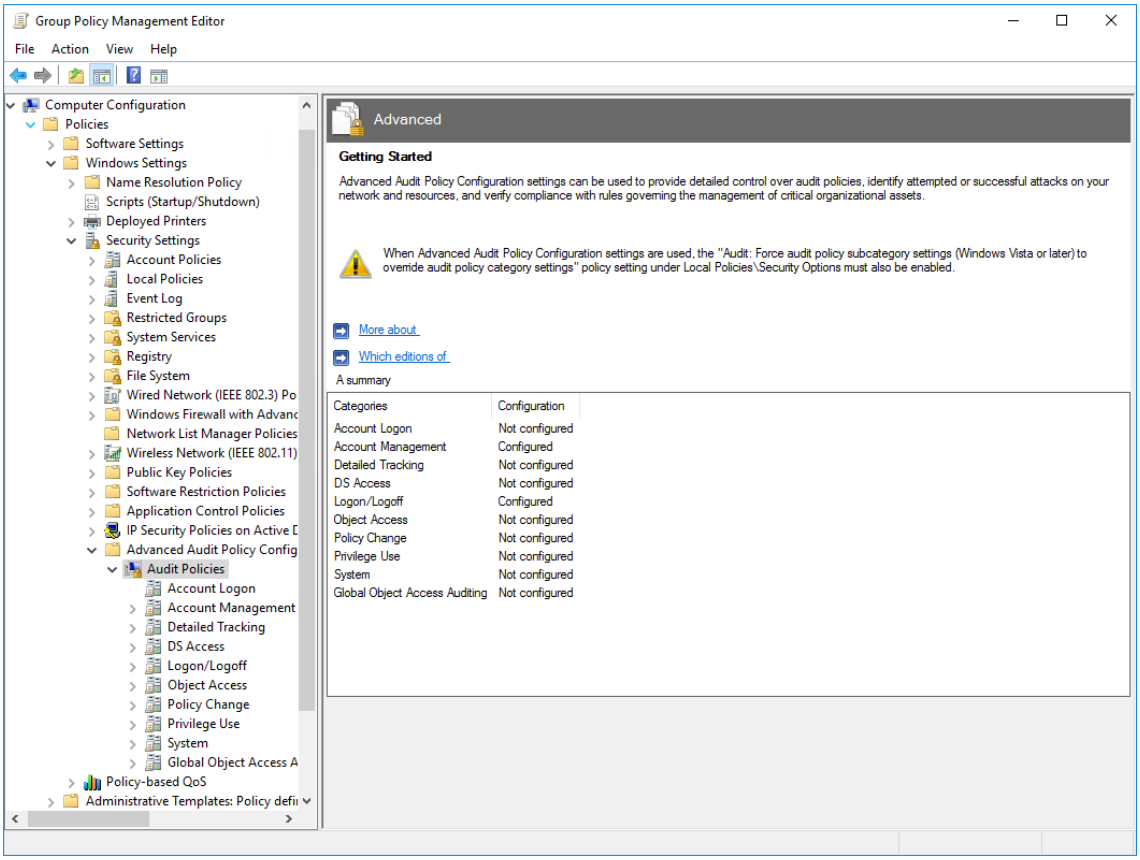


Figura E.5 – Configuração do Nível de Auditoria do Windows

Como é possível ver na Figura E.6, ao selecionar a auditoria dos eventos de sucesso e falha de *Logon*, esses eventos serão armazenados no log de segurança do *Windows*. Pelo contrário, se for definida uma política para não auditar determinados tipos de eventos, é possível efetuar uma filtragem dos eventos que são armazenados no *log* de segurança das plataformas *MS Windows*.

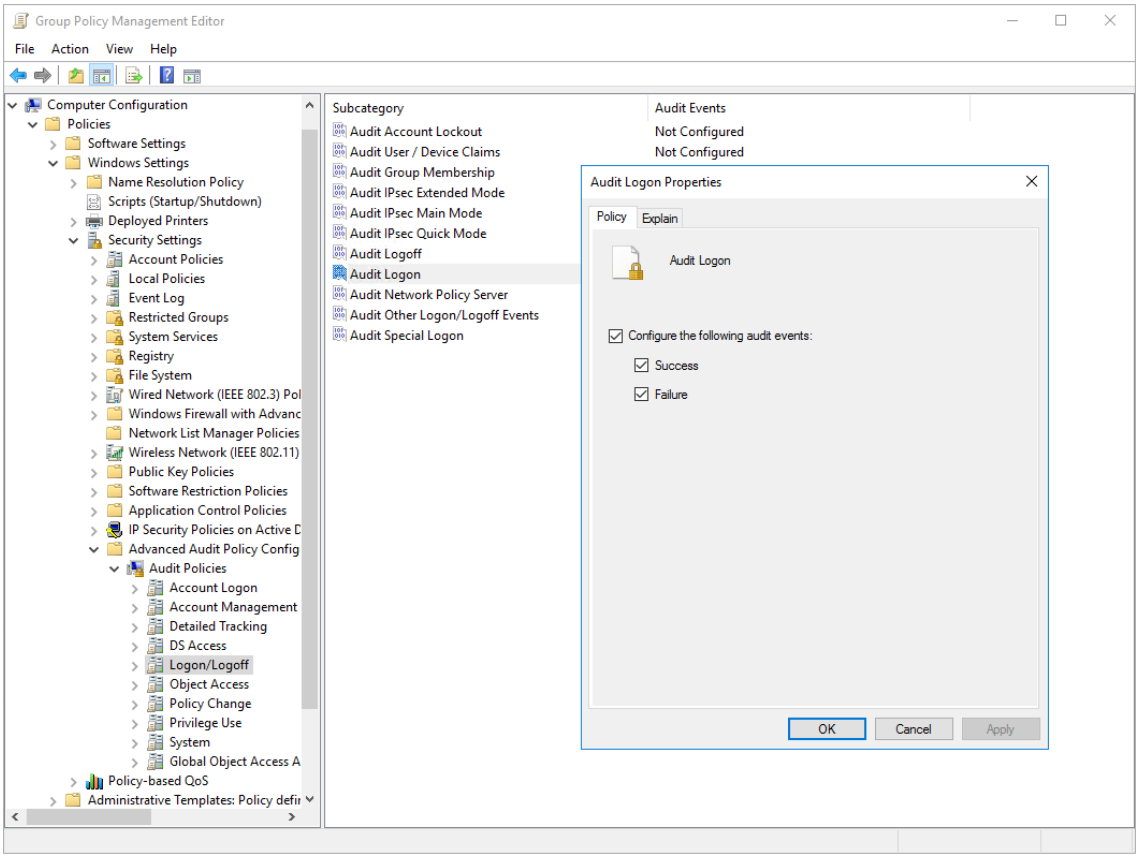


Figura E.6 – Configuração de Auditoria dos Eventos de "Logon"

ANEXO F – **Configurações do *Plugin NXLog* no *Alienvault***

As seguintes configurações deverão ser adicionadas, e/ou modificadas no ficheiro de configuração do *Plugin NXLog* do *Alienvault*, localizado em /etc/ossim/agent/plugins/nxlog.cfg.

Evento com o ID 4624 – O logon de uma conta foi efetuado com êxito.

```
[0041 - successful logon]
# win-id 4624
precheck=";4624;"
event_type=event
regexp='^(?P<incoming_date>\w{3}\s{1,2}\d{1,2}\s\d{2}:\d{2}:\d{2}) \S+
WIN-NXLOG (?P<date>\d{4}-\d{2}-\d{2}
\d{2}:\d{2}:\d{2});"(?P<event_type>AUDIT_SUCCESS)";"(?P<severity>INFO)
";"(?P<channel>Security)";"(?:-
|(?P<hostname>[^"]*))";(?P<event_id>4624);"(?P<source_name>[^"]+)";"(?
P<account_name>[^"]*)";"(?P<account_type>[^"]*)";"(?P<domain>[^"]*)";"
An account was successfully logged on\.\s+Subject:\s+Security
ID:\s+\S+\s+Account Name:\s+(?P<subject_account_name>.*?)\s+Account
Domain:\s+(?P<subject_account_domain>.*?)\s+Logon ID:\s+.*?\s+Logon
Type:\s+(?P<logon_type>.*?)\s+.*?(?:Elevated
Token:\s+(?P<elevated_token>.*?)\s+.*?)?\s+(?:Impersonation
Level:\s+.*?\s+)?New Logon:\s+Security ID:\s+\S+\s+Account
Name:\s+(?P<logon_account_name>.*?)\s+Account
Domain:\s*(?P<logon_account_domain>.*?)\s+Logon ID:\s+.*?\s+Logon
GUID:\s+\S+\s+Process Information:\s+Process ID:\s+\S+\s+Process
Name:\s+.*?\s+Network Information:\s+Workstation
Name:\s+(?P<workstation_name>.*?)\s*Source Network Address:\s+(?:-
|(?P<src_addr>\S+))\s+Source Port:\s+(?:-
|(?P<src_port>\S+))\s+Detailed Authentication Information:\s+Logon
Process:\s+\S+\s+Authentication
Package:\s+(?P<authentication_package>\S+)'
date={normalize_date($date)}
plugin_sid={:postfix_auth_event_sid($event_id, $logon_account_name,
$logon_account_domain, $hostname)}
src_ip={:select_srcip($src_addr, $hostname, $workstation_name)}
src_port={$src_port}
dst_ip={resolv($hostname)}
device={$hostname}
username={$logon_account_domain}\{$logon_account_name}
userdata1={$logon_account_domain}\{$logon_account_name}
userdata2={$workstation_name}
userdata3={translate2($logon_type,$4624)}
```

```
userdata4={$authentication_package}
userdata5={$source_name}
userdata6={:check_user_privileges($hostname, $logon_account_domain,
$logon_account_name)}
userdata7={$severity}
userdata8={$event_type}
userdata9={$hostname}
```

Evento com o ID 4625 – Falha no logon de uma conta.

```
[0042 - failed logon]
# win-id 4625
precheck=";4625;"
event_type=event
regexp='^(?P<incoming_date>\w{3}\s{1,2}\d{1,2}\s\d{2}:\d{2}:\d{2}) \S+
WIN-NXLOG (?P<date>\d{4}-\d{2}-\d{2}
\d{2}:\d{2}:\d{2});"(?P<event_type>AUDIT_FAILURE)";"(?P<severity>ERROR
)";"(?P<channel>Security)";"(?:-
|(?P<hostname>[^"]*))";(?P<event_id>4625);"(?P<source_name>[^"]+)";"(?
P<account_name>[^"]*)";"(?P<account_type>[^"]*)";"(?P<domain>[^"]*)";"
An account failed to log on.\s+Subject:\s+Security ID:\s+\S+\s+Account
Name:\s+(?P<subject_account_name>.*?)\s+Account
Domain:\s+(?P<subject_account_domain>.*?)\s+Logon ID:\s+\S+\s+Logon
Type:\s+(?P<logon_type>\S*)\s+Account For Which Logon
Failed:\s+Security ID:\s+\S+\s+Account
Name:\s+(?P<logon_account_name>.*?)\s+Account
Domain:\s+(?P<logon_account_domain>.*?)\s+Failure
Information:\s+Failure
Reason:\s+(?P<logon_failure_reason>[^\.]*)\.\s+Status:\s+(?P<status>\S
+)\s+Sub Status:\s+(?P<sub_status>\S+)\s+Process Information:\s+Caller
Process ID:\s+\S+\s+Caller Process Name:\s+\S+\s+Network
Information:\s+Workstation Name:\s+(?P<workstation_name>.*?)\s+Source
Network Address:\s+(?:-|(?P<src_addr>\S+))\s+Source Port:\s+(?:-
|(?P<src_port>\S+))\s+Detailed Authentication Information:\s+Logon
Process:\s+\S+\s+Authentication
Package:\s+(?P<authentication_package>\S+)'
date={normalize_date($date)}
plugin_sid={:postfix_auth_event_sid($event_id, $logon_account_name,
$logon_account_domain, $hostname)}
src_ip={:select_srcip($src_addr, $hostname, $workstation_name)}
src_port={$src_port}
dst_ip={resolv($hostname)}
device={$hostname}
username={$logon_account_domain}\{$logon_account_name}
userdata1={$logon_account_domain}\{$logon_account_name}
userdata2={$workstation_name}
userdata3={translate2($logon_type,$4624)}
userdata4={translate2($status,$4625)}
userdata5={translate2($sub_status,$4625)}
userdata6={:check_user_privileges($hostname, $logon_account_domain,
$logon_account_name)}
userdata7={$severity}
userdata8={$event_type}
userdata9={$hostname}
```

Evento com o ID 4740 – Uma conta de utilizador foi bloqueada.

```
[0074 - user account locked out]
# win-id 4740
```

```
precheck=";4740;"
event_type=event
regexp='^(?P<incoming_date>\w{3}\s{1,2}\d{1,2}\s\d{2}:\d{2}:\d{2}) \S+
WIN-NXLOG (?P<date>\d{4}-\d{2}-\d{2}
\d{2}:\d{2}:\d{2});"(?P<event_type>AUDIT_SUCCESS)";"(?P<severity>[^"]*
)";"(?P<channel>[^"]*)";"(?P<hostname>[^"]*)";(?P<event_id>4740);"(?P<
source_name>[^"]*)";"(?P<account_name>[^"]*)";"(?P<account_type>[^"]*)
";"(?P<domain>[^"]*)";"A user account was locked
out.\s+Subject:\s+Security ID:\s+[\w\d-]+\s+Account
Name:\s+(?P<subject_account_name>.*?)\s+Account
Domain:\s+(?P<subject_account_domain>.*?)\s+Logon ID:\s+\S+\s+Account
That Was Locked Out:\s+Security ID:\s+[\w\d-]+\s+Account
Name:\s+(?P<locked_account_name>.*?)\s+Additional
Information:\s+Caller Computer
Name:\s+(?P<caller_computer_name>.*)"(?:#\d{3})?$'
date={normalize_date($date)}
plugin_sid=4740
src_ip={resolv($caller_computer_name)}
dst_ip={resolv($caller_computer_name)}
device={$hostname}
username={$subject_account_domain}\{$locked_account_name}
userdata1={$locked_account_name}
userdata2={$caller_computer_name}
userdata3={$subject_account_domain}
userdata4={$channel}
userdata5={$source_name}
userdata6={$account_type}
userdata7={$severity}
userdata8={$event_type}
userdata9={$hostname}
```

Evento com o ID 4768 – A Kerberos authentication ticket (TGT) was requested.

```
[0115 - Kerberos TGT]
# win-id 4768
precheck=";4768;"
event_type=event
regexp='^(?P<incoming_date>\w{3}\s{1,2}\d{1,2}\s\d{2}:\d{2}:\d{2}) \S+
WIN-NXLOG (?P<date>\d{4}-\d{2}-\d{2}
\d{2}:\d{2}:\d{2});"(?P<event_type>AUDIT_(?:SUCCESS|FAILURE))";"(?P<se
verity>[^"]*)";"(?P<channel>[^"]*)";"(?P<hostname>[^"]*)";(?P<event_id
>4768);"(?P<source_name>[^"]*)";"(?P<account_name>[^"]*)";"(?P<account
_type>[^"]*)";"(?P<domain>[^"]*)";"A Kerberos authentication ticket
\(TGT\) was requested.\s.*Account
Name:\s+(?P<subject_account_name>.*?)\s+Supplied Realm
Name:\s+(?P<subject_account_domain>.*?)\s+\s.*?Client
Address:\s+(?P<source_workstation>.*?)\s+Client
Port:\s+(?P<source_workstation_port>.*?)\s+.*?Ticket
Options:\s+(?P<ticket_option>.*?)\s+Result
Code:\s+(?P<result_code>.*?)\s+Ticket Encryption
Type:\s+(?P<ticket_encryption_type>.*?)\s+'
date={normalize_date($date)}
plugin_sid=4768
src_ip={resolv($source_workstation)}
src_port={$source_workstation_port}
dst_ip={resolv($hostname)}
device={$hostname}
username={$subject_account_domain}\{$subject_account_name}
userdata1={translate2($ticket_encryption_type,$kerberos)}
userdata4={$channel}
```

```
userdata5={$source_name}
userdata6={:check_user_privileges2($event_id, $subject_account_domain,
$subject_account_name, $account_type)}
userdata7={$severity}
userdata8={$event_type}
userdata9={$hostname}
```

Evento com o ID 4771 – Falha na pré-autenticação Kerberos.

```
[0121 - Kerberos pre-authentication failed]
# win-id 4771
precheck=";4771;"
event_type=event
regexp='^(?P<incoming_date>\w{3}\s{1,2}\d{1,2}\s\d{2}:\d{2}:\d{2}) \S+
WIN-NXLOG (?P<date>\d{4}-\d{2}-\d{2}
\d{2}:\d{2}:\d{2});"(?P<event_type>AUDIT_FAILURE)";"(?P<severity>[^"]*
)";"(?P<channel>[^"]*)";"(?P<hostname>[^"]*)";(?P<event_id>4771);"(?P<
source_name>[^"]*)";"(?P<account_name>[^"]*)";"(?P<account_type>[^"]*)
";"(?P<domain>[^"]*)";"Kerberos pre-authentication
failed.\s.*?Security ID:\s.*Account
Name:\s+(?P<subject_account_name>.*?)\s+\s.*?Client
Address:\s+(?P<source_workstation>.*?)\s+Client
Port:\s+(?P<source_workstation_port>.*?)\s+.*?Ticket
Options:\s+(?P<ticket_option>.*?)\s+Failure
Code:\s+(?P<ticket_encryption_type>.*?)\s+'
date={normalize_date($date)}
plugin_sid=4771
src_ip={resolv($source_workstation)}
src_port={$source_workstation_port}
dst_ip={resolv($hostname)}
device={$hostname}
username={:resolv_username_4771($hostname, $subject_account_name)}
userdata1={translate2($ticket_encryption_type,$kerberos)}
userdata4={$channel}
userdata5={$source_name}
userdata6={:check_user_privileges2($event_id, $hostname,
$subject_account_name, $account_type)}
userdata7={$severity}
userdata8={$event_type}
userdata9={$hostname}
```

Evento com o ID 4778 – A session was reconnected to a Window Station.

```
[0101 - A session was re|dis connected from|to a Window Station]
# win-id 4778, 4779
precheck=";477"
event_type=event
regexp='^(?P<incoming_date>\w{3}\s{1,2}\d{1,2}\s\d{2}:\d{2}:\d{2}) \S+
WIN-NXLOG (?P<date>\d{4}-\d{2}-\d{2}
\d{2}:\d{2}:\d{2});"(?P<event_type>AUDIT_SUCCESS)";"(?P<severity>[^"]*
)";"(?P<channel>[^"]*)";"(?P<hostname>[^"]*)";(?P<event_id>477[89]);"(
?P<source_name>[^"]*)";"(?P<account_name>[^"]*)";"(?P<account_type>[^"
]*)";"(?P<domain>[^"]*)";"(?P<message>A session was
(?:re|dis)connected (?:to|from) a Window
Station.)\s+Subject:\s+Account
Name:\s+(?P<subject_account_name>.*?)\s+Account
Domain:\s+(?P<subject_account_domain>.*?)\s+.*?Client
Address:\s+(?P<client_address>.*?)\s+'
date={normalize_date($date)}
```

```
plugin_sid={$event_id}
src_ip={:resolv_srcip($client_address, $hostname)}
dst_ip={resolv($hostname)}
device={$hostname}
username={$subject_account_domain}\{$subject_account_name}
userdata4={$channel}
userdata5={$source_name}
userdata6={$account_type}
userdata7={$severity}
userdata8={$event_type}
userdata9={$hostname}
```

Evento com o ID 4801 – A estação de trabalho foi desbloqueada.

```
[0104 - The workstation was (un)locked]
# win-id 4800, 4801
precheck=";480"
event_type=event
regexp='^(?P<incoming_date>\w{3}\s{1,2}\d{1,2}\s\d{2}:\d{2}:\d{2}) \S+
WIN-NXLOG (?P<date>\d{4}-\d{2}-\d{2}
\d{2}:\d{2}:\d{2});"(?P<event_type>AUDIT_SUCCESS)";"(?P<severity>[^"]*
)";"(?P<channel>[^"]*)";"(?P<hostname>[^"]*)";(?P<event_id>480[01]);"(
?P<source_name>[^"]*)";"(?P<account_name>[^"]*)";"(?P<account_type>[^"
]*)";"(?P<domain>[^"]*)";"(?P<message>The workstation was
(?:un|)locked.)\s+Subject:\s+Security ID:\s+[\d\w-]+\s+Account
Name:\s+(?P<subject_account_name>.*?)\s+Account
Domain:\s+(?P<subject_account_domain>.*?)\s+'
date={normalize_date($date)}
plugin_sid={$event_id}
src_ip={resolv($hostname)}
dst_ip={resolv($hostname)}
device={$hostname}
username={$subject_account_domain}\{$subject_account_name}
userdata4={$channel}
userdata5={$source_name}
userdata6={$account_type}
userdata7={$severity}
userdata8={$event_type}
userdata9={$hostname}
```

ANEXO G – **Funções Personalizadas do *Plugin* *NXLog* do *Alienvault***

As seguintes funções deverão ser adicionadas, e/ou modificadas no ficheiro de configuração das funções personalizadas do *Pluguin NXLog* do *Alienvault*, localizado em /etc/ossim/agent/plugins/custom_functions/winnxlog_functions.cfg.

Função postfix_auth_event_sid:

```
Start Function postfix_auth_event_sid
    def postfix_auth_event_sid(self, sid, username, domain,
workstation_name='', logon_account_name='-', logon_account_domain='-
'):
        """
        Postfix a plugin_sid:
        domain != workstation_name and username == a user
           AD related event -> return sid
        domain != workstation_name and username == a service$
           AD related service event -> return sid + '1'
        domain == workstation_name and username == a user
           local user login forwarded to the AD -> return sid + '2'
        domain == workstation_name and username == a service$
           local service login forwarded to the AD -> return sid + '3'
        """
        if (domain == '-' and username == '-'):
            domain = logon_account_domain
            username = logon_account_name

        if domain == workstation_name.partition('.')[0]:
            if username.endswith('$'):
                return int(str(sid) + '3')
            return int(str(sid) + '2')
        elif username.endswith('$'):
            return int(str(sid) + '1')
        return int(sid)
End Function
```

Função select_srcip:

```
Start Function select_srcip
import socket
def select_srcip(self, src_addr, hostname, workstation_name):
```

```
    """
    Return a workstation name if IP is missing, or null.
    """
    if ((src_addr == '-' or src_addr == '') and (workstation_name ==
'-' or workstation_name == '')):
        try:
            return "%s" % socket.gethostbyname(hostname)
        except Exception, e:
            return "%s" % hostname
    elif (src_addr == '::1' or src_addr == '127.0.0.1'):
        try:
            return "%s" % socket.gethostbyname(hostname)
        except Exception, e:
            return "%s" % hostname
    elif ((src_addr == '-' or src_addr == '') and (workstation_name !=
'-' and workstation_name != '')):
        try:
            return "%s" % socket.gethostbyname(workstation_name)
        except Exception, e:
            return "%s" % workstation_name
    else:
        return "%s" % src_addr
End Function
```

Função check_user_privileges:

```
Start Function check_user_privileges
def     check_user_privileges(self,     hostname,     domain,     user,
user_privileges = '-'):
    """
    # CHECK USER PRIVILEGES
    """
    # Set PRIVILIGED_USERS file location
    privileged_users_file                                              =
'/etc/ossim/agent/plugins/custom_functions/PrivilegedAccounts'
    user=user.split("$", 1)[0]

    #Chech if it is a Domain User
    if(domain.lower().partition('.')[0]                                !=
hostname.lower().partition('.')[0]):
        #Chech User Privileges Level
        user_with_domain = (domain + "\\" + user).strip('\'').lower()
        # Set the USER_PRIVILEGES if the event don't have it set
        if  (user_with_domain  !=  ''  and  (user_privileges  ==  ''  or
user_privileges == '-' or user_privileges is None)):
            user_privileges = "No"
            with open(privileged_users_file, "r") as f:
                # Check if USER exists in the PRIVILEGED_USERS_FILE
                for line in f:
                    if(user_with_domain == line.strip().lower()):
                        user_privileges = "Yes"
                        break
                f.close()

    #Return Account Privilege Level
    return ("%s" % user_privileges)
End Function
```

Função check_user_privileges2:

```
Start Function check_user_privileges2
def     check_user_privileges2(self,     event_id,     domain,     user,
user_privileges = '-'):
    """
    # CHECK USER PRIVILEGES
    """
    # Set PRIVILIGED_USERS file location
    privileged_users_file                                                =
'/etc/ossim/agent/plugins/custom_functions/PrivilegedAccounts'

    user=user.split("$", 1)[0]
    if (event_id == '4771'):
        domain = domain.split(".", 1)[1]
    #Chech User Privileges Level
    user_with_domain = (domain + "\\" + user).strip('\'').lower()
    # Set the USER_PRIVILEGES if the event don't have it set
    if   (user_with_domain   !=   ''   and   (user_privileges   ==   ''   or
user_privileges == '-' or user_privileges is None)):
        user_privileges = "No"
        with open(privileged_users_file, "r") as f:
            # Check if USER exists in the PRIVILEGED_USERS_FILE
            for line in f:
                if (user_with_domain == line.strip().lower()):
                    user_privileges = "Yes"
                    break
            f.close()

    #Return Account Privilege Level
    return ("%s" % user_privileges)
End Function
```

Função resolv_username_4771:

```
Start Function resolv_username_4771
def resolv_username_4771(self, dc_name, user):
   """
   Return a domain\username.
   """
   domain = dc_name.split(".", 1)[1]
   return "%s\%s" % (domain, user)
End Function
```

ANEXO H – *Script* para Identificar Contas de Utilizador com Privilégios

```
#!/usr/bin/env python
from commands import getoutput
import ldap

########## COMANDOS PARA CORRER A QUERY RECURSIVA A TODO O DOMINIO
#############
##
## LOCALIZACAO: cd /etc/ossim/agent/plugins/custom_functions/
##
## COMANDO: python -c "execfile('Get_PrivilegedAccounts.py');
main(<Dominio de topo, no formato: 'DC=DOMINIO1,DC=PT'>, '<Dominio da
conta para efetuar login em todas as AD>', '<Conta para efetuar login
em todas as AD>', '<Password>', <Grupos de interesse, no formato:
['CN=Administrators,CN=Builtin,DC=DOMINIO1,DC=PT', 'CN=Domain
Admins,CN=Users,DC=DOMINIO1,DC=PT', ...]>, '<localizacao onde vai ser
escrito o ficheiro com a listagem de contas de utilizador>')"
##
##
#########################################################################
##########

#Retorna a listagem de todos os dominios existentes.
##O filtro da query é feito pelo grupo (CN=Users), uma vez que este
grupo existe em todos os dominios.
def get_domains_list(full_domain, login_account_domain,
login_account_name, password_login):
    domain_result_set = set([])
    result_data = ([])
    try:
        search_domain =
full_domain.lower().partition('dc=')[2].replace(',dc=', '.')
        result_data = filter(None, getoutput('ldapsearch -LLL -H
ldap://' + search_domain + ':3268 -b ' + full_domain.lower() + ' -D "'
+ login_account_domain + '\\' + login_account_name + '" -w "' +
password_login + '" "CN=Users" ""').splitlines())
        for j in range(len(result_data)):
            if(result_data[j].startswith("dn: ")):
                domain_result_set.add("DC=" +
result_data[j].partition('DC=')[2])
        return(list(domain_result_set))
    except:
        pass
```

```
#Retorna a lista de todas as contas de utilizador que pertencem aos
grupos especificados no "filterBy_distinguishedName".
def get_nested_user_accounts(full_domain, login_account_domain,
login_account_name, password_login, filterBy_distinguishedName):
    try:
        nested_user_result_set= set([])
        search_domain =
full_domain.lower().partition('dc=')[2].replace(',dc=', '.')
        for entry in filterBy_distinguishedName:
            try:
                nested_user_result_data = filter(None,
getoutput('ldapsearch -LLL -H ldap://' + search_domain + ':3268 -b ' +
full_domain.lower() + ' -D "' + login_account_domain + '\\' +
login_account_name + '" -w "' + password_login + '"
"(&(memberof:1.2.840.113556.1.4.1941:=' + entry +
')(objectClass=user))" ""').splitlines())
                for j in range(len(nested_user_result_data)):

nested_user_result_set.add(nested_user_result_data[j])
            except (ldap.LDAPError, e):
                print (e)
        return(list(nested_user_result_set))
    except:
        return("")



#Retorna a lista de todos os grupos de utilizador que pertencem aos
grupos especificados no "filterBy_distinguishedName".
def get_nested_child_group_accounts(full_domain, login_account_domain,
login_account_name, password_login, filterBy_distinguishedName):
    try:
        child_group_result_set= set([])
        search_domain =
full_domain.lower().partition('dc=')[2].replace(',dc=', '.')
        for entry in filterBy_distinguishedName:
            try:
                child_group_result_data = filter(None,
getoutput('ldapsearch -LLL -H ldap://' + search_domain + ':3268 -b ' +
full_domain.lower() + ' -D "' + login_account_domain + '\\' +
login_account_name + '" -w "' + password_login + '"
"(&(memberof:1.2.840.113556.1.4.1941:=' + entry +
')(objectClass=group))" ""').splitlines())
                for j in range(len(child_group_result_data)):
                    if ((full_domain.lower() != ("dc=" +
child_group_result_data[j].lower().partition('dc=')[2])) and
(len(child_group_result_data[j]) > 0)):

child_group_result_set.add(child_group_result_data[j])
            except (ldap.LDAPError, e):
                print (e)
        return(list(child_group_result_set))
    except:
        return("")



#Retorna a lista de todas as contas de utilizador que pertencem aos
grupos especificados no "filterBy_distinguishedName", de forma
recursiva.
```

```
def get_full_nested_user_accounts(full_domain, login_account_domain,
login_account_name, password_login, filterBy_distinguishedName):
    users_result_set= set([])
    groups_result_set= set([])
    checked_group = ([])

    users_result_data = get_nested_user_accounts(full_domain,
login_account_domain, login_account_name, password_login,
filterBy_distinguishedName)
    for k in range(len(users_result_data)):
        users_result_set.add(users_result_data[k])
    groups_result_data = get_nested_child_group_accounts(full_domain,
login_account_domain, login_account_name, password_login,
filterBy_distinguishedName)
    for l in range(len(groups_result_data)):
        groups_result_set.add(groups_result_data[l])
    while groups_result_set:
        try:
            check_group = groups_result_set.pop()
            if (check_group not in checked_group):
                checked_group.append(check_group)
                filterBy_child_distinguishedName = []

                full_child_domain = ("DC=" +
check_group.upper().partition('DC=')[2])

filterBy_child_distinguishedName.append(check_group.partition('dn:
')[2])
                users_result_data =
get_nested_user_accounts(full_child_domain, login_account_domain,
login_account_name, password_login, filterBy_child_distinguishedName)
                for k in range(len(users_result_data)):
                    users_result_set.add(users_result_data[k])

                groups_result_data =
get_nested_child_group_accounts(full_child_domain,
login_account_domain, login_account_name, password_login,
filterBy_child_distinguishedName)
                for l in range(len(groups_result_data)):
                    groups_result_set.add(groups_result_data[l])
        except:
            pass
    return(list(users_result_set))



#Retorna uma lista de contas de utilizador formatada para ser escrita
no ficheiro.
def parse_users_data(result_set):
    if len(result_set) == 0:
            print ("No Results.")
            pass
    output = set([])
    for i in range(len(result_set)):
        name =
result_set[i].lower().partition('cn=')[2].partition(',')[0]
        domain =
result_set[i].lower().partition('dc=')[2].replace(',dc=', '.')

        domain_list = domain.split('.')
        list_length = len(domain_list)
```

```
        if(list_length > 0):
            print_domain = domain_list[0]
            for i in range(1, (list_length + 1)):
                if (print_domain and name):
                    output.add(print_domain + "\\" + name)
                    if (i < list_length):
                        print_domain += '.' + domain_list[i]
    return(sorted(list(output)))


#Cria o ficheiro com a listagem de contas de utilizador.
def write_to_file_users(output_file_location, users):
    try:
        f= open(output_file_location,"w+")
        f.write("\n".join(users))
        f.close()
        print("File %s Successfully Writed." % output_file_location)
        return
    except:
        print("Error while writing in the file " +
output_file_location + ".")
        pass


#Retorna um ficheiro com a listagem de todos os utilizadores
pertencentes aos grupos especificados pelo
"filterBy_distinguishedName", ao fazer query a todos os Global Catalog
dos sub-dominios, de forma recursiva.
##Ao especificar no "filterBy_distinguishedName" os grupos com
privilegios de administracao e possivel obter uma listagem de todos os
utilizadores com privilegios existentes nos diversos dominios.
def main(full_domain, login_account_domain, login_account_name,
password_login, filterBy_distinguishedName, output_file_location):
    print("\nSTARTED...\n")
    user_accounts_set = set([])
    domain_list = get_domains_list(full_domain, login_account_domain,
login_account_name, password_login)
    domain_list.sort(key=len, reverse=False)
    for i in range(len(domain_list)):
        try:
            user_accounts_data =
get_full_nested_user_accounts(domain_list[i], login_account_domain,
login_account_name, password_login, filterBy_distinguishedName)
            for j in range(len(user_accounts_data)):
                user_accounts_set.add(user_accounts_data[j])
        except:
            pass
    users = parse_users_data(list(user_accounts_set))
    write_to_file_users(output_file_location, users)
    print("END")
```

ANEXO I – **Diretivas (Regras de Correlação)**

Falhas de Autenticação em Contas de Domínio Com Privilégios

```
<directive id="500009" name="Windows Bruteforce Attack, Login
Authentication Attack Against Domain Privileged Accounts"
priority="5">
   <rule type="detector" name="Domain Priviliged Account
    authentication failure" from="ANY" to="ANY" port_from="ANY"
    port_to="ANY" reliability="+0" occurrence="1" plugin_id="1817"
    plugin_sid="4625" userdata6="Yes">
      <rules>
         <rule type="detector" name="Domain Priviliged Accounts -
          Multiple authentication failures" from="ANY" to="ANY"
          port_from="ANY" port_to="ANY" reliability="10"
          occurrence="10" time_out="120" plugin_id="1817"
          plugin_sid="4625" userdata6="Yes"/>
      </rules>
   </rule>
</directive>
```

Falhas de Autenticação em Contas de Domínio Sem Privilégios

```
<directive id="500010" name="Windows Bruteforce Attack, Login
Authentication Attack Against Domain Non-Privileged Accounts"
priority="4">
   <rule type="detector" name="Domain Non-Priviliged Account
    authentication failure" from="ANY" to="ANY" port_from="ANY"
    port_to="ANY" reliability="+0" occurrence="1" plugin_id="1817"
    plugin_sid="4625" userdata6="No">
      <rules>
         <rule type="detector" name="Domain Non-Priviliged Accounts -
          Multiple authentication failures" from="ANY" to="ANY"
          port_from="ANY" port_to="ANY" reliability="8" occurrence="60"
          time_out="120" plugin_id="1817" plugin_sid="4625"
          userdata6="No"/>
      </rules>
   </rule>
</directive>
```

Falhas de Autenticação em Contas Locais

```
<directive id="500016" name="Windows Bruteforce Attack, Login
Authentication Attack Against Local Accounts" priority="4">
   <rule type="detector" name="Local Account authentication failure"
    from="ANY" to="ANY" port_from="ANY" port_to="ANY" reliability="+0"
    occurrence="1" plugin_id="1817" plugin_sid="46252">
```

```
      <rules>
         <rule type="detector" name="Local Accounts - Multiple
          authentication failures" from="ANY" to="ANY" port_from="ANY"
          port_to="ANY" reliability="8" occurrence="30" time_out="120"
          plugin_id="1817" plugin_sid="46252"/>
      </rules>
   </rule>
</directive>
```

Falhas de Autenticação a Partir do Mesma Origem

```
<directive id="500017" name="Windows Bruteforce Attack, Login
Authentication Attack From SRC_IP" priority="5">
   <rule type="detector" name="Account Authentication Failure"
    from="ANY" to="ANY" port_from="ANY" port_to="ANY" reliability="+0"
    occurrence="1" plugin_id="1817" plugin_sid="4625,46252">
      <rules>
         <rule type="detector" name="Multiple Authentication Failures
          From SRC_IP" from="1:SRC_IP" to="ANY" port_from="ANY"
          port_to="ANY" reliability="10" occurrence="20" time_out="120"
          plugin_id="1817" plugin_sid="4625,46252"/>
      </rules>
   </rule>
</directive>
```

Utilizador Autenticado em Várias Estações

```
<directive id="500008" name="Windows Attack, User Account (USERNAME)
Authenticated In multiple Workstations" priority="4">
   <rule type="detector" name="An account was successfully logged on"
    from="ANY" to="ANY" port_from="ANY" port_to="ANY" reliability="+0"
    occurrence="1" plugin_id="1817" plugin_sid="4624,4778,4801">
      <rules>
         <rule type="detector" name="An account was successfully
          logged on in multiple Workstations (More than 10)" from="ANY"
          to="ANY" port_from="ANY" port_to="ANY" reliability="8"
          occurrence="10" time_out="180" plugin_id="1817"
          plugin_sid="4624,4778,4801" sticky_different="DST_IP"
          username="1:USERNAME"/>
      </rules>
   </rule>
</directive>
```

Múltiplos Bloqueios de Conta

```
<directive id="500013" name="Windows Bruteforce Attack, Excessive
Account Lockouts" priority="5">
   <rule type="detector" name="Windows Account Lockout detected"
    from="ANY" to="ANY" port_from="ANY" port_to="ANY" reliability="+0"
    occurrence="1" plugin_id="1817" plugin_sid="4740">
      <rules>
         <rule type="detector" name="Windows Excessive Account
          Lockouts detected" from="ANY" to="ANY" port_from="ANY"
          port_to="ANY" reliability="10" occurrence="20" time_out="120"
          plugin_id="1817" plugin_sid="4740"/>
      </rules>
   </rule>
```

```
</directive>
```

Múltiplos Bloqueios de Conta a Partir da Mesma Origem

```
<directive id="500014" name="Windows Bruteforce Attack, Excessive
Account Lockouts From Workstation SRC_IP" priority="5">
   <rule type="detector" name="Windows Account Lockout detected"
    from="ANY" to="ANY" port_from="ANY" port_to="ANY" reliability="+0"
    occurrence="1" plugin_id="1817" plugin_sid="4740">
      <rules>
         <rule type="detector" name="Windows Excessive Account
          Lockouts detected" from="1:SRC_IP" to="ANY" port_from="ANY"
          port_to="ANY" reliability="10" occurrence="5" time_out="120"
          plugin_id="1817" plugin_sid="4740"/>
      </rules>
   </rule>
</directive>
```

ANEXO J – **Criação de uma Subscrição**

Para criação de uma subscrição é necessário confirmar que o servidor WEC se encontra configurado de acordo com o ANEXO C, e as políticas de grupo para auditoria e encaminhamento de eventos se encontram configuradas de acordo com o ANEXO E.

De seguida, as configurações são efetuadas no interface web do *Supercharger Manager*, através do URL http://<Supercharger_FQDN>/Supercharger/Collectors (conforme Figura J.1).

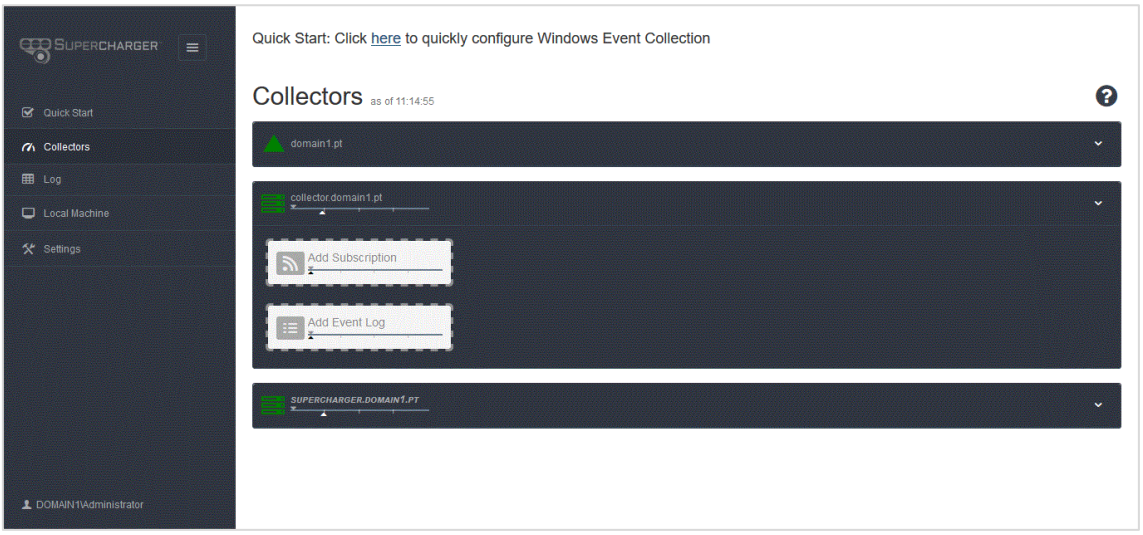


Figura J.1 – Interface Web do Supercharger para Configuração dos WEC

O primeiro passo é criar um log de eventos customizado, que será utilizado para armazenamento dos eventos recebidos dos WEF. Para tal, deve-se selecionar o WEC respetivo e clicar no ícone "*Add Event Log*". Será aberta uma nova janela (conforme Figura J.2) onde serão introduzidas as configurações do log de eventos que se pretende criar. Caso se pretenda utilizar um log de eventos do sistema, ou um customizado já existente, não é necessário executar este passo.

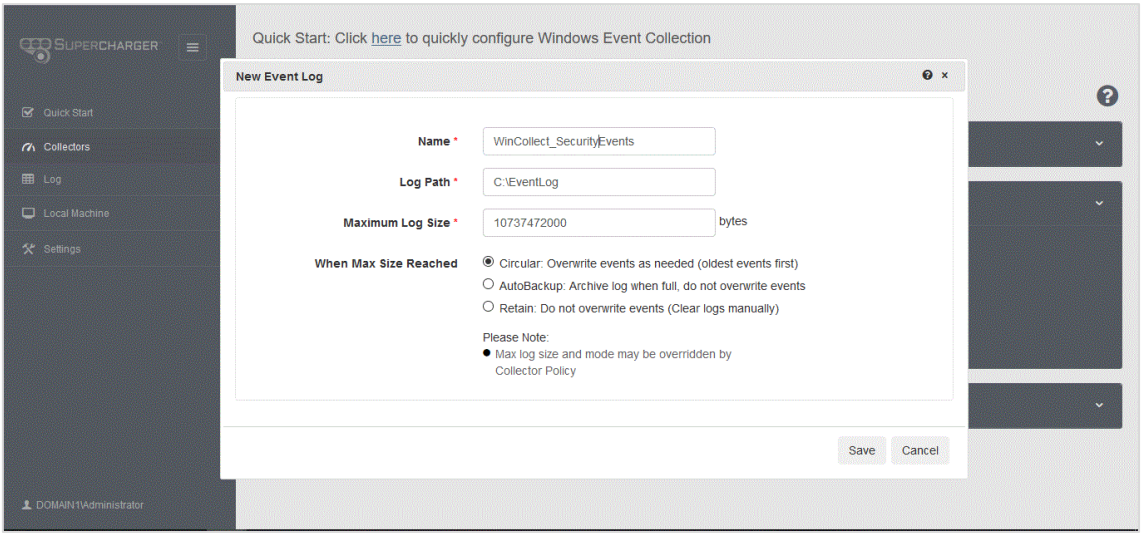


Figura J.2 – Interface Web para Criação de um Log de Eventos

Ao clicar no botão "*Save*", as configurações do novo log de eventos serão salvas, e dentro de alguns instantes o ficheiro de log é criado no WEC, e aparecerá no interface web do *Supercharger*, de acordo com a Figura J.3.

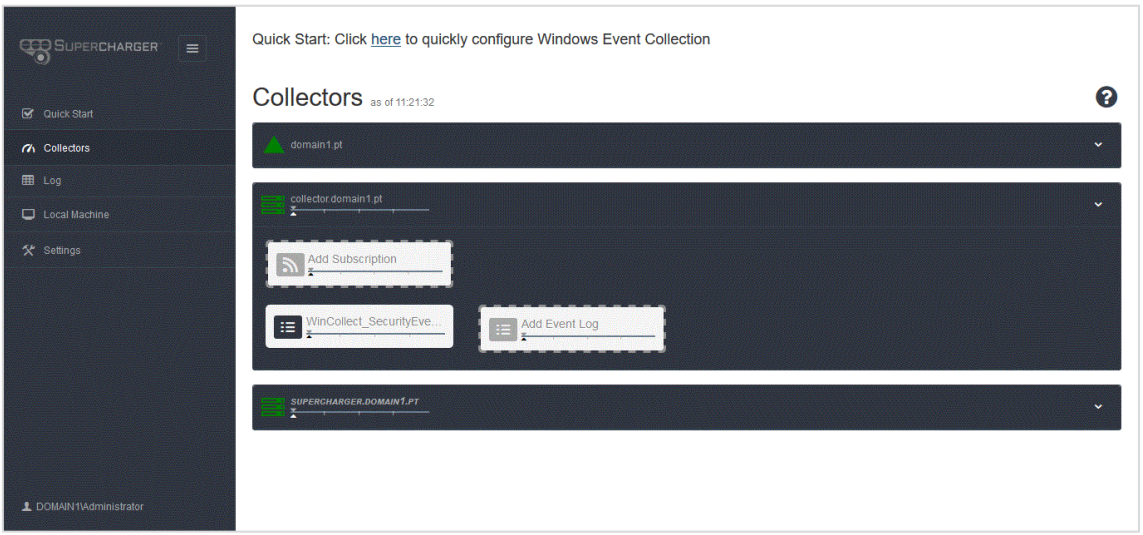


Figura J.3 – Interface Web para Visualização do Estado dos WEC

Após a configuração do log de eventos estar completa, poderá dar-se início à configuração da subscrição, ao clicar no ícone "*Add Subscription*". Será aberta uma nova janela (conforme Figura J.4) onde serão introduzidas as configurações da nova subscrição. A configuração da subscrição divide-se em cinco passos: Descrição e Seleção do Log de Destino; Políticas de Subscrição; Seleção dos WEF; Filtros de Eventos; Submissão da Subscrição.

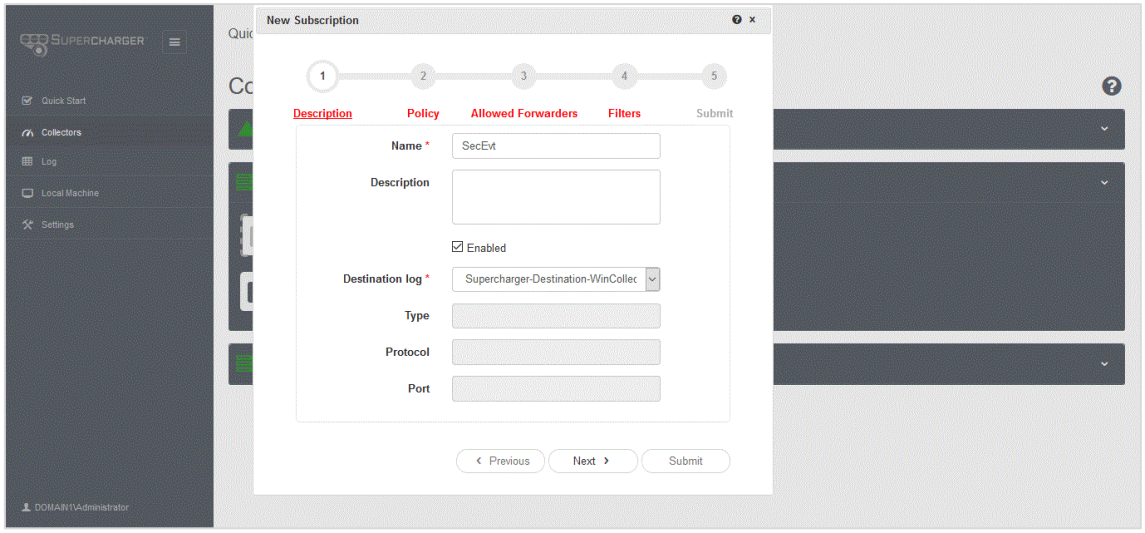


Figura J.4 – Configuração de uma Subscrição (Passo 1 de 5)

Na Descrição e Seleção do Log de Destino, conforme Figura J.4, é configurado o nome da subscrição, e o log, onde os eventos recebidos serão armazenados. Ao clicar no botão "*Next*" iremos para a página de configuração das Politicas de Subscrição, conforme Figura J.5.

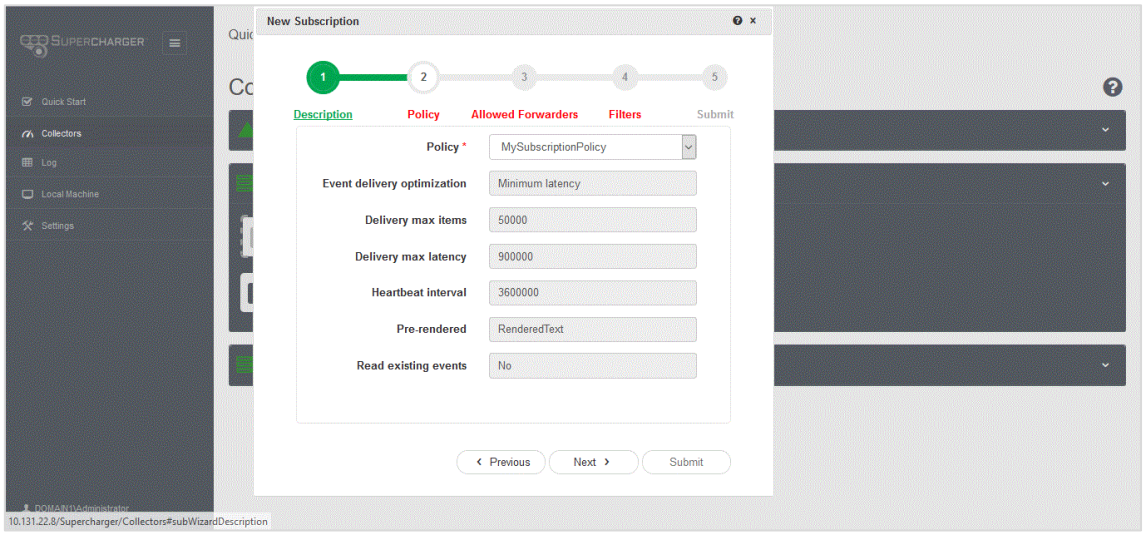


Figura J.5 – Configuração de uma Subscrição (Passo 2 de 5)

Nesta página serão selecionadas as configurações da subscrição, de acordo com a lista de políticas de subscrição que se encontram definidas no *Supercharger*. Esta lista contém as políticas que vêm integradas na instalação do *Supercharger*, sendo possível adicionar políticas customizadas. Após a configuração das políticas de subscrição, ao clicar no botão "*Next*" iremos para a página de configuração dos WEF, conforme Figura J.6.

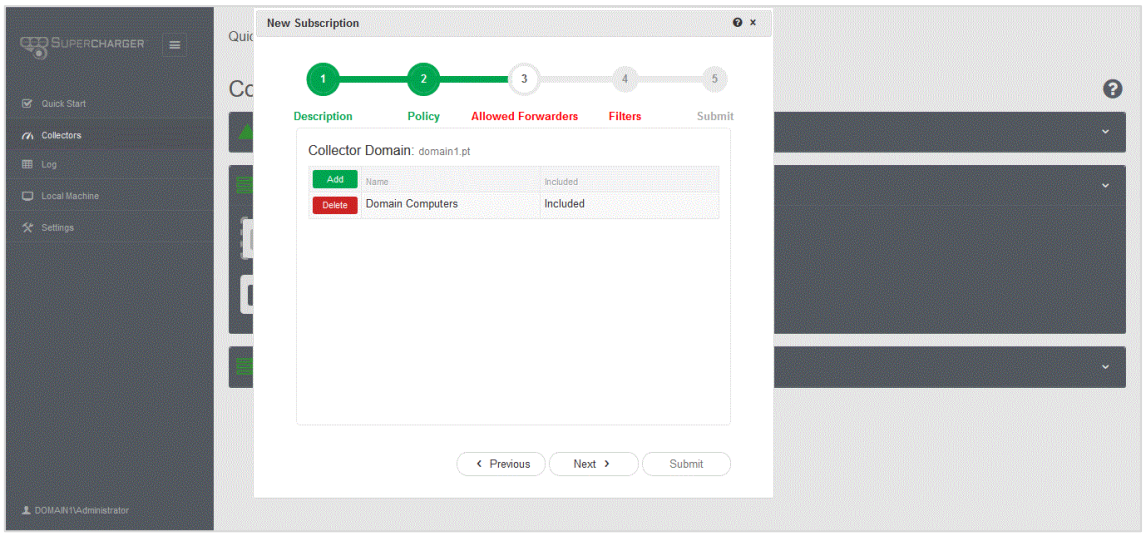


Figura J.6 – Configuração de uma Subscrição (Passo 3 de 5)

Ao clicar no botão "*Add*" é possível adicionar os computadores *MS Windows* que se pretende que reencaminhem eventos para o WEC de forma individual, ou através da seleção de um grupo de máquinas, conforme mostra na Figura J.6.

Após a seleção dos WEF, ao clicar no botão "*Next*" iremos para a página de configuração dos filtros de eventos, conforme Figura J.7.

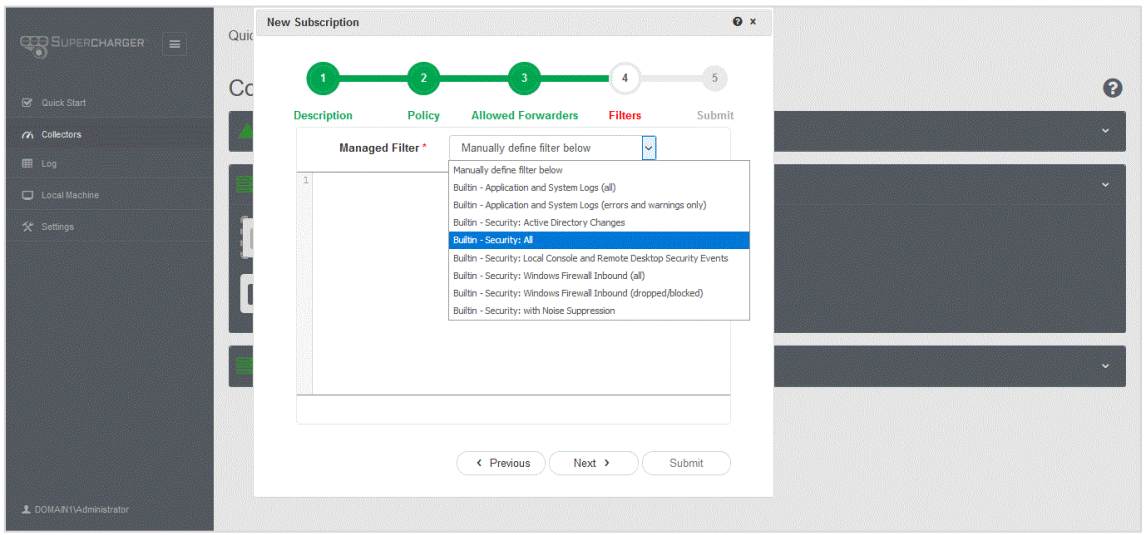


Figura J.7 – Configuração de uma Subscrição (Passo 4 de 5)

Nesta página é possível selecionar um dos filtros de eventos que vêm integrados na instalação do *Supercharger*, ou um filtro que tenha sido customizado e se encontre guardado na lista de *Managed Filters* do *Supercharger*. Existe também a possibilidade de definir um novo filtro customizado de forma manual, através da opção "*Manually*

define filter below". Após configurado o filtro de eventos, ao clicar no botão "*Next*" iremos para a página de submissão da subscrição, conforme Figura J.8.

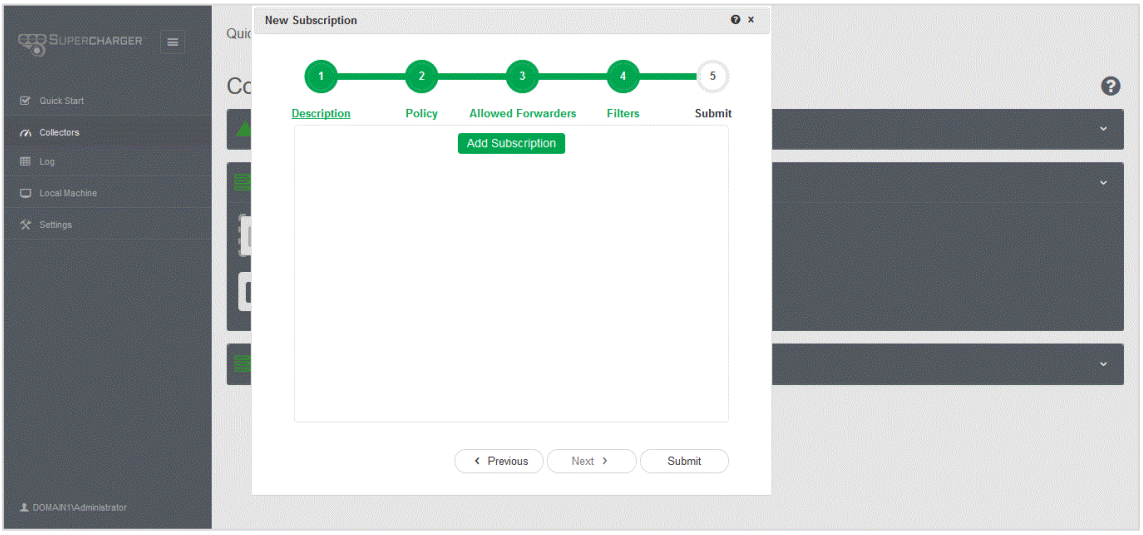


Figura J.8 – Configuração de uma Subscrição (Passo 5 de 5)

Nesta página, ao selecionar "*Add Subscription*", ou "*Submit*", a subscrição configurada será enviada para o WEC respetivo, onde será criada através do *Supercharger Agent*. Passados alguns instantes deverá ser possível visualizar a subscrição, bem como o seu estado, na interface web do *Supercharger*, conforme Figura J.9.

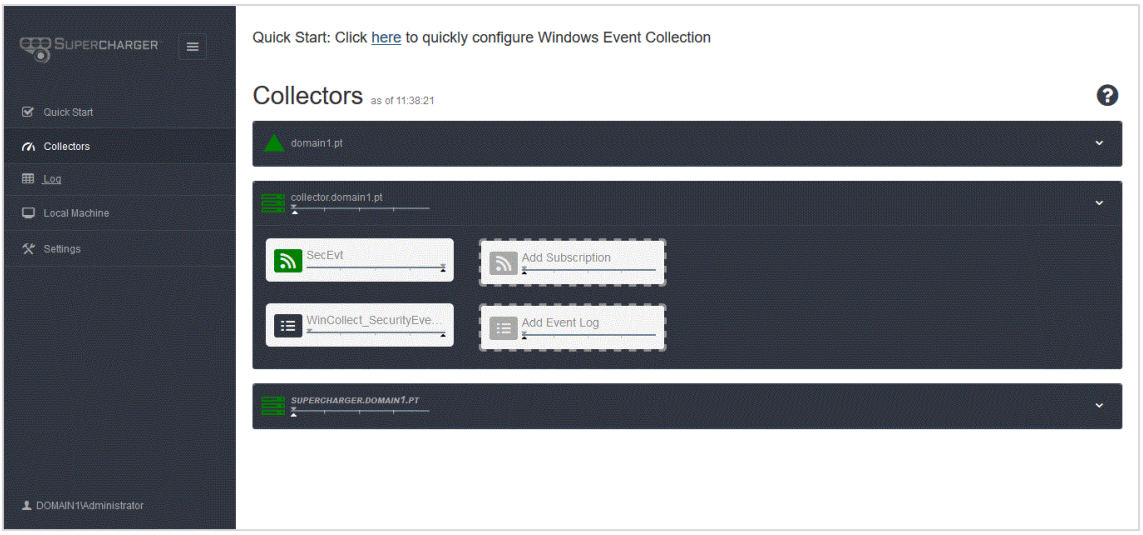


Figura J.9 – Visualização do Estado dos WEC e das Subscrições

Uma vez que o *Supercharger* apenas permite configurar subscrições HTTP, caso seja necessário que os WEF enviem os eventos para o WEC através de um protocolo de

comunicação seguro (HTTPS), é necessário efetuar os passos constantes no ponto 4.2 do ANEXO C, no coletor de eventos onde foi criada a subscrição.

Após efetuar o procedimento anterior, é possível confirmar na interface web do *Supercharger* que as configurações se encontram aplicadas, conforme Figura J.10.

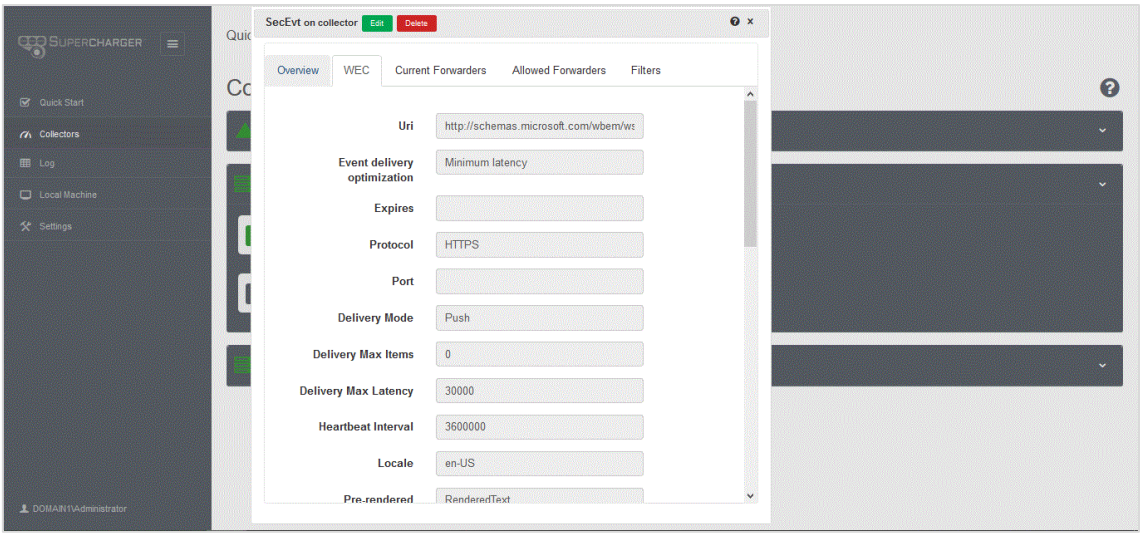


Figura J.10 – Visualização das Configurações da Subscrição