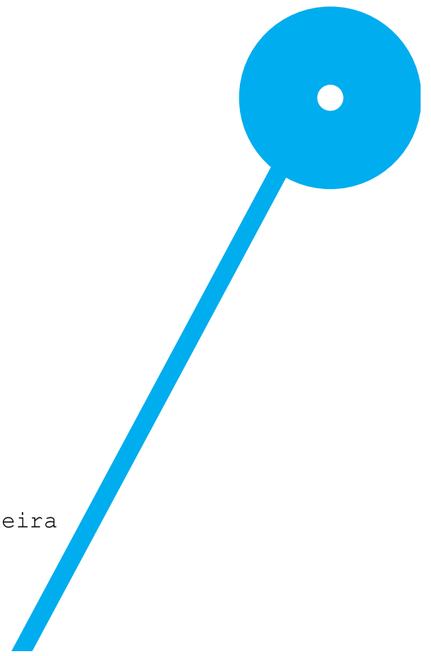
ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO
POLITÉCNICO
DO PORTO

P.PORTO

M

MESTRADO
ENGENHARIA INFORMÁTICA



Evolutive Software Asset Management

Paulo Jorge da Silva Teixeira

10/2020

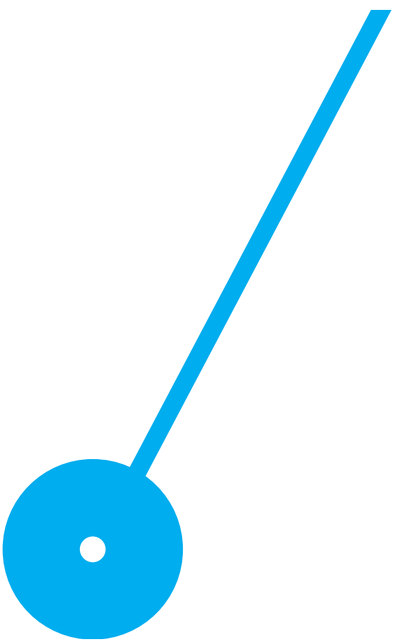
Paulo Jorge da Silva Teixeira. Evolutive Software Asset Management.

P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO
POLITÉCNICO
DO PORTO

M

MESTRADO
ENGENHARIA INFORMÁTICA



Evolutive Software Asset Management

Paulo Jorge da Silva Teixeira

10/2020

P.PORTO

escola
superior
de tecnologia
e gestão
POLITÉCNICO
DO PORTO

P.PORTO

M

MESTRADO
ENGENHARIA INFORMÁTICA

Evolutive Software Asset Management

Paulo Jorge da Silva Teixeira

João Paulo Magalhães

Dedicatória

Dedico este trabalho aos meus pais, que me educaram a nunca desistir.

"O seu trabalho irá preencher uma grande parte da sua vida, e a única maneira de ficar realmente satisfeito é fazer o que acredita ser um ótimo trabalho. E a única maneira de fazer um ótimo trabalho é amar o que faz. Se ainda não o encontrou, continue a procurar. Não se acomode. Como acontece com todos os assuntos do coração, saberá quando o achou."

Steve Jobs

Agradecimentos

Um Obrigado.

Um Obrigado a todos os que me ajudaram nesta última etapa da faculdade.

Um Obrigado ao meu Orientador João Paulo Magalhães pela sua disponibilidade, sugestões e críticas na realização deste projeto.

Um Obrigado à Professora Doutora Dorabela Gamboa pela disponibilidade com que sempre brindou os mestrandos em Engenharia Informática.

Um Obrigado à entidade de acolhimento AET Europe por receber o meu Projeto, pela disponibilização de todos os meios concedidos para que me fosse possível realizá-lo.

Um Obrigado à minha mãe por acreditar em mim.

Um Obrigado à minha noiva por me lembrar sempre para nunca desistir dos nossos sonhos.

Um Obrigado aos meus amigos da faculdade por me recordarem que não estou sozinho nesta etapa.

E, por fim, não menos importante, um obrigado a todos quanto comigo se cruzaram neste curto, mas tão importante percurso da minha vida e partilharam as suas experiências de vida. Por serem muitos não os menciono individualmente.

Obrigado!

Resumo

Uma boa gestão de ativos de software é crucial para implementar práticas eficazes de segurança. Esta gestão é essencial para ajudar a combater ataques que podem traduzir-se em custos elevados para as empresas, tanto em termos financeiros como na sua reputação. Ao realizar a gestão de ativos estamos a reduzir os riscos legais e de segurança associados a estes ativos, bem como aumentar o desempenho operacional através da redução da taxa de falhas e aumento de disponibilidade. Em suma, a gestão de ativos permite a gestão da informação de forma simples e unificada o que se traduz numa otimização de processos e custos inerentes ao *software* e seu licenciamento e ainda numa resposta mais eficaz a potenciais problemas de segurança.

Nesta dissertação foi desenvolvida uma solução, denominada E-Sam, de gestão de ativos mais concretamente de software e licenças associadas com o intuito de ajudar as organizações a mitigar potenciais problemas de segurança. O modelo de suporte à solução é constituído por uma aplicação central que efetua a recolha dos dados, alimentada por um conjunto de agentes configuráveis e extensíveis que são executados em múltiplos dispositivos. Os dados relativos aos ativos são armazenados e é feita a sua análise, nomeadamente a pesquisa de vulnerabilidades que possam existir e afetem esses ativos. Esta solução é importante para as organizações na medida em que automatiza o processo de recolha de informação sobre ativos, análise a existência de vulnerabilidades conhecidas e despoleta alarmes que podem ajudar a sua rápida mitigação. Considerando a dimensão e complexidade dos parques informáticos, a nível de hardware e software, dispor de soluções automatizadas deste tipo é um aspeto crucial para a manutenção e bom funcionamento dos sistemas e consequentemente das organizações suportadas por estes.

Palavras-chave: Gestão de ativos, Gestão de licenças de software, Segurança, Vulnerabilidades, Agentes remotos

Abstract

Good software asset management is crucial to implementing effective security practices. This management is essential to help fight attacks that can translate into high costs for companies, both in financial terms and in their reputation. By performing asset management, we are reducing the legal and security risks associated with these assets, as well as increasing operational performance by reducing the failure rate and increasing availability. In short, asset management allows the management of information in a simple and unified manner, which translates into an optimization of processes and costs inherent to software and its licensing, as well as a more effective response to potential security problems.

In this dissertation, an asset management solution was developed, named E-Sam, more specifically software and associated licenses in order to help organizations mitigate potential security problems. The solution support model consists of a central application that collects data, powered by a set of configurable and extensible agents that run on multiple devices. The data related to the assets are stored and analyzed, namely the search for vulnerabilities that may exist and affect these assets. This solution is important for organizations as it automates the process of gathering information about assets, analyzes the existence of known vulnerabilities and triggers alarms that can help its rapid mitigation. Considering the size and complexity of computer parks, in terms of hardware and software, having automated solutions of this type is a crucial aspect for the maintenance and smooth functioning of the systems and, consequently, of the organizations supported by them.

Keywords: Asset management, Software license management, Security, Vulnerabilities, Remote agents

Índice

DEDICATÓRIA ..... **II**

AGRADECIMENTOS ..... **III**

RESUMO ..... **IV**

ABSTRACT ..... **V**

CAPÍTULO 1 - INTRODUÇÃO **10**

1.1 **ESTRUTURA DA DISSERTAÇÃO** 12

CAPÍTULO 2 - ESTADO DE ARTE **13**

2.1 **GESTÃO DE ATIVOS DE TECNOLOGIA DA INFORMAÇÃO** 13

2.2 **BENEFÍCIOS DA GESTÃO DE ATIVOS** 17

2.3 **PRINCÍPIOS DA GESTÃO DE ATIVOS** 18

2.4 **CICLO DE VIDA DOS ATIVOS** 20

2.5 **AGENTES REMOTOS** 21

2.6 **GESTÃO DE VULNERABILIDADES** 22

2.7 **SOLUÇÕES DE MERCADO EXISTENTES** 24

2.7.1 *Manage Engine Asset Explorer* *25*

2.7.2 *InvGate Assets* *27*

2.7.3 *Asset Tiger* *29*

2.7.4 *Snipe-IT* *30*

2.7.5 *Comparação das Soluções Tecnológicas* *31*

CAPÍTULO 3 – METODOLOGIA DE INVESTIGAÇÃO **34**

3.1 **CARACTERIZAÇÃO DA ENTIDADE DE ACOLHIMENTO** 34

3.2 **FERRAMENTAS UTILIZADAS** 34

3.3 CENÁRIO DE IMPLEMENTAÇÃO 39

3.4 DESENVOLVIMENTO E CONFIGURAÇÃO DA SOLUÇÃO 42

3.4.1 Funcionalidades *42*

3.4.2 Métodos desenvolvidos *42*

3.4.3 Configuração da aplicação central *49*

3.5 RESUMO DO CAPÍTULO 55

CAPÍTULO 4 – RESULTADOS OBTIDOS EM PROVA DE CONCEITO 56

4.1 SUMÁRIO DO CAPÍTULO 62

CAPÍTULO 5 - CONSIDERAÇÕES FINAIS 64

BIBLIOGRAFIA 66

ANEXOS 70

ANEXO 1 – *RESPOSTA DA CHAMADA A UMA CHAMADA À API VULDB* 70

APÊNDICES 73

APÊNDICE 1 – EXEMPLO APLICAÇÃO MOBILE POWER BI 73

APÊNDICE 2 – CÓDIGO PARA RECOLHA DE LICENÇAS DE SOFTWARE DO WINDOWS *REGISTRY* ATRAVÉS DE POWERSHELL 73

APÊNDICE 3 – TAREFA CRIADA *WINDOWS SCHEDULER* 74

APÊNDICE 4 – PÁGINA WEB REPOSITÓRIO DOS AGENTES 74

APÊNDICE 5 – *SCHEMA* BASE DE DADOS 75

APÊNDICE 6 – *REGRAS DEFINIDAS NO POSTGRES PARA CHAMADAS REST (POSTREST)* 78

Lista de Quadros

Tabela 1- Exemplos de vulnerabilidades [35] 23

Tabela 2 - Tabela de comparação de Características das Soluções Alternativas..... 31

Tabela 3 - Tabela de comparação entre soluções alternativas e o E-Sam 57

Lista de Figuras

Figura 1 - Ativos e o seu contexto organizacional [6] 14

Figura 2 - Fase do ciclo de ITSM [15] 16

Figura 3 - Princípios da gestão de ativos (Baseada em [6]) 19

Figura 4 - Ciclo de vida dos ativos segundo a ISO 27001:2013 A.8 (Baseada em [6]) 20

Figura 5 - As cinco fases em foco no AssetExplorer [39] 26

Figura 6 - Asset Explorer Demo Main Menu 27

Figura 7 - Menu Solução InvGate 28

Figura 8 - Adição de um novo ativo, AssetTiger 30

Figura 9 - Soluções tecnológicas AET Europe 34

Figura 10 - Magic Quadrant for BI & Analytics Platforms 2015 (Gartner, 2017) 37

Figura 11 - Magic Quadrant for BI & Analytics Platforms 2017 (Gartner, 2017) 38

Figura 12 - Arquitetura proposta para gestão de ativos 39

Figura 13 - Modelo relacional E-Sam 40

Figura 14 - Exemplo NVD API 41

Figura 15 - Código executável a ser distribuído pelos funcionários da empresa 43

Figura 16 - Script para obter licenças Linux 44

Figura 17 - Método find_programs 45

Figura 18 - Método get_system_info 46

Figura 19 - Método get_location 47

Figura 20 - Parte do Método para procurar vulnerabilidades e inserir na base de dados 48

Figura 21 - Método para agendar uma nova procura em ambientes Windows 49

Figura 22 - PostgREST API [61] 53

Figura 23 – Criação e assinatura de um token de autenticação 54

Figura 24 - Criação de relações entre tabelas Power BI 55

Figura 25 - Exemplo de relatório vulnerabilidades por Nome, Risco e Editor 58

Figura 26 - Exemplo relatório ao selecionar um tipo de vulnerabilidades (high) 59

Figura 27 - Vulnerabilidades por dia e por Internet Protocol (IP) 60

Figura 28 - Relatório vulnerabilidades High e localização 60

Figura 29 - Licenças encontradas Windows Linux 61

Figura 30 - Alertas criados no Power BI 62

Figura 31 - Email de alerta recebido pelo Power BI 62

Figura 32 - Exemplo dos relatórios do Power BI no Smartphone 73

Figura 33 - Abordagem inicial em PowerShell script 73

Figura 34 - Windows Scheduler tarefa SAM 74

Figura 35 - Certificada autenticação cliente 74

Figura 36 - Página de download dos agentes e certificado servidor Apache 75

Lista de Abreviaturas

AC - Autoridade Certificadora

API - Application Programming Interface

ASN - Autonomous System Number

COM - Component Object Model

CPE - Common Platform Enumeration

CVE - Common Vulnerabilities and Exposures

E-Sam - Evolutive Software Asset Management

Https - Hypertext Transfer Protocol Secure

IDE - Integrated Development Environment

IP - Internet Protocol

ITAM - Gestão de ativos de TI (IT Asset Management)

ITIL - Information Technology Infrastructure Library

JSON - JavaScript Object Notation

NVD - National Vulnerability Database

OSVDB - Open Source Vulnerability Database

RGPD - Regulamento Geral sobre a Proteção de Dados

REST - Representational State Transfer

SAM- Software Asset Management

SCAP - Security Content Automation Protocol

SO - Sistema Operativo

SQL - Structured Query Language

SSL - Secure Sockets Layer

TI - Tecnologia da Informação

Capítulo 1 - Introdução

O número de equipamentos interligados em rede continua a aumentar. Se por um lado temos cada vez mais funcionalidades disponíveis através da digitalização dos processos, por outro as pessoas, máquinas e informação que se interliga no ciberespaço estão constantemente expostas a perigos diversos. Os ataques informáticos estão a ficar mais complexos em resultado dos próprios atores maliciosos que estão cada vez mais evoluídos em termos de técnicas, ferramentas e procedimentos que adotam para levar a cabo os seus ataques. O número e proximidade das ciberameaças tem feito com a consciência de que as empresas, independentemente do seu tamanho ou sector de atividade, precisam de estar preparadas para lidar com os ciberataques [1].

Hoje em dia, de forma a implementar práticas eficazes de segurança informática nas organizações, torna-se crucial estas apresentarem uma boa gestão de seus ativos. A gestão de ativos é um conjunto de boas práticas que permite às empresas controlar e realizar valor com os seus ativos desde a sua aquisição até ao seu abate. Como ativos temos por exemplo: equipamentos informáticos; contratos; equipamentos utilizados no processo produtivo; marcas; ferramentas e materiais; know-how. A gestão dos ativos é essencial no âmbito da cibersegurança, visto permitir mitigar ataques informáticos internos e externos às organizações que possam interferir com os mesmos causando impacto nas organizações. De acordo com um relatório divulgado pelo Eurostat em [2], uma em cada oito empresas europeias reportaram problemas na área das tecnologias de informação e comunicação. Segundo o mesmo estudo, os problemas comuns incluem falhas de software ou hardware, ataques distribuídos de negação de serviço (DDoS, na sigla inglesa), em que o propósito é sobrecarregar uma rede de computadores com muita informação de forma a paralisá-la, e ataques de *ransomware*, isto é, bloqueio de um sistema informático, com o criminoso a exigir à vítima o pagamento de um resgate para que a ligação/libertação de recursos seja executada. Cerca de 45% das empresas da União Europeia mantêm registos de incidentes de segurança para análise mesmo ao implementar e utilizar medidas de proteção para prevenir ataques informáticos. Em Portugal, as empresas apresentam uma média de 58% destes registos. De acordo com os estudos efetuados pela Accenture em [3], o número de ataques de *malware* atingiu um valor recorde no ano 2018. Segundo a empresa de segurança SonicWall [3], foram registados mais de 10 mil milhões de ataques de *malware* nesse mesmo ano. É o terceiro ano consecutivo de crescimento deste tipo de ataques, a nível global. A empresa norte-americana ressalva que os ataques estão a crescer não só em número como em variedade. Em 2018, foram identificadas mais de 391 mil novas variedades de ataques.

Os ciberataques representam custos avultados para as empresas. Em 2018, segundo [3], as empresas gastaram em média 2,3 milhões de euros, valor gasto para resolver problemas ligados a *malware*. Trata-se de um aumento de 12% relativamente a 2017. Nestes custos milionários incluem-se, por exemplo, o tempo gasto para recuperação dos sistemas, contratação de empresas externas e outros valores associados. Pelas contas da empresa de segurança Kaspersky, detalhadas no relatório *IT Security: "cost-center or strategic investment?"* [1] um ataque informático representa uma despesa

média de 87 mil dólares, o equivalente a 74 mil euros, às pequenas e médias empresas. Além disso, o relatório da Kaspersky descobriu outro elemento relevante: o custo médio dos ataques é significativamente superior quando o alvo dos mesmos não são diretamente às empresas, mas sim parceiros externos como fornecedores de serviços de Infraestrutura. O diretor de negócios corporativos da Kaspersky Lab, Alessio Aceti, afirmou em [1] que "enquanto incidentes de cibersegurança que envolvem terceiros são prejudiciais para empresas de qualquer tamanho, o seu impacto financeiro tem a capacidade de criar o dobro do prejuízo. Isto deve-se a uma escala maior e mais global do desafio em que as ameaças se transformam rapidamente, mas as empresas e a legislação mudam lentamente. Quando regulamentos como o Regulamento Geral sobre a Proteção de Dados (RGPD) forem aplicados e se detetarem empresas que ainda não adotem o regulamento, as multas de incumprimento só vão aumentar a conta final". A Aon Portugal na sua publicação [4], estima que até 2021 as perdas decorrentes de ataques cibernéticos deverão atingir os seis biliões de dólares a nível global. Preveem que as empresas irão perder cada vez mais dinheiro na sequência de falhas de segurança, tais como, os custos processuais após um ataque, as multas regulatórias - que aumentam após a implementação do Regulamento Geral de Proteção de Dados - e as quebras nas receitas devido a interrupções na atividade são as principais consequências financeiras.

Além disso, os ataques informáticos também afetam a reputação das organizações por falta de controlo e/ou gestão. Anabela Araújo, Chief Broking Officer e diretora de Sinistros da Aon Portugal, numa entrevista publicada em [4], afirmou que "a crise de reputação resultante de um cyber ataque pode comprometer o valor de mercado de uma empresa, destruir a lealdade à marca, limitar os esforços de transformação digital e até levar a uma diminuição do *rating* de crédito. Uma estratégia eficaz de resiliência cibernética pode ajudar a mitigar perdas financeiras imediatas e de longo prazo."

Os ciberataques, podem criar o pior pesadelo no mundo dos negócios: o seu encerramento. Em [5], a Malwarebytes apresenta um estudo que envolve 1 054 empresas na América do Norte, França, Reino Unido, Alemanha, Austrália e Singapura, que mostra que os mais prejudicados são as pequenas e médias empresas sendo que 22% delas, quando atingidas, têm de encerrar a atividade imediatamente.

A gestão de ativos não é por si só a solução para as ciberameças e ciberataques. Porém, a gestão de ativos permite conhecer e controlar os bens ao dispor do negócio, permitindo atuar proactivamente na sua proteção ou então mais rapidamente perante cenários de ataque mitigando os impactos. De forma a empreender uma ótima gestão de ativos, em primeiro lugar, as empresas precisam de saber os ativos que possuem, que ativos são de facto utilizados pelos colaboradores, o seu estado de atualização e integridade, o seu licenciamento e custos associados. O conhecimento dessas informações permite às empresas agir rapidamente e dar resposta mais eficaz a potenciais problemas de segurança, tais como, aumento do nível do risco e exposição, ataques informáticos, conformidade e regulamentação como o RGPD.

Devido à evolução e importância da gestão de ativos e cibersegurança, este projeto tem como principal motivação a criação de um serviço de pesquisa e gestão de ativos, mais concretamente sobre

licenças e programas de software, que permite efetuar eficazmente a sua manutenção e a pesquisa proactiva sobre vulnerabilidades conhecidas que envolvam esses ativos. O serviço será composto por uma aplicação central alimentada por múltiplos agentes, que tem a função de identificar e recolher ativos em múltiplos dispositivos. A aplicação recebe a informação enviada pelos agentes de forma segura, efetuando a gestão dos ativos e operações associadas, como a notificação de expiração, necessidade de novas licenças, verificação de atualizações, verificação de vulnerabilidades, entre outros. A aplicação deverá ser leve, configurável e capaz de ser replicada para responder a aumentos de carga. Os agentes arquitetados de forma modular e evolutiva, para que se possam estender futuramente a outros tipos de ativos. Ou seja, os agentes irão possuir algumas características, tais como, uma identificação única e podem ser configurados ou atualizáveis por ação da aplicação central. Também irão ter a capacidade de adaptação a ambientes distintos, tais como, Windows, Linux e diferentes estruturas de ficheiros.

O serviço no seu todo deve assegurar a alta disponibilidade e adaptação da aplicação/agentes a novas necessidades da gestão de ativos. No que diz respeito à primeira, o cenário contempla um sistema de alta disponibilidade, mais propriamente a utilização de dois servidores em que o segundo é um espelho do primeiro. Relativamente à segunda característica, a aplicação e os agentes devem ser arquitetados de forma a que possam ser evolutivos, isto é, possam ser adicionadas novas funcionalidades. Para que se possam detetar ativos com vulnerabilidades de forma proactiva, está contemplada a integração com Application Programming Interface (API's) e com uma base de dados com entradas de erros/bugs/possíveis vulnerabilidades, para correlacionar estes dados com a aplicação central.

1.1 Estrutura da dissertação

Esta dissertação está estruturada em cinco capítulos. O capítulo dois refere-se ao Estado de Arte. O capítulo três aborda a metodologia de investigação utilizada e onde é efetuada uma contextualização da empresa onde se realizou o projeto, o respetivo enquadramento metodológico da investigação e os programas e ferramentas utilizados. No quarto capítulo são apresentados os resultados obtidos, os quais são discutidos face ao Estado de Arte. Por último, no quinto capítulo, intitulado de considerações finais são apresentadas as conclusões do projeto bem como os contributos do mesmo, as limitações existentes na realização do projeto e deixadas sugestões de trabalho futuro.

Capítulo 2 - Estado de Arte

Este capítulo divide-se em sete seções. Ao longo das seções contextualizamos o tema gestão de ativos ao abordar as suas definições, os seus benefícios, os seus princípios e o seu ciclo de vida. Além disso, definimos o conceito de agentes remotos ao mostrar a sua utilidade e importância para o desenvolvimento deste trabalho. Por fim, referimos as várias soluções de gestão de ativos existentes no mercado.

2.1 Gestão de Ativos de Tecnologia da Informação

Segundo a Norma ISO 27001 A.8 [6], um ativo é definido como "Qualquer item de valor económico de propriedade de um indivíduo ou empresa". No que diz respeito à Norma ISO/IEC 55000 [7], "ativo é um item, algo ou entidade que tem valor real ou potencial, para uma organização". O seu valor irá variar entre diferentes organizações e partes interessadas. Por outro lado, segundo a ISO/IEC 19770-1:2017 [8], documento que especifica processos, requisitos, implementação e manutenção de um sistema de gestão de ativos de TI (ITAM), um ativo de TI é um "item, coisa ou entidade que pode ser usada para adquirir, processar, armazenar e distribuir informações digitais e tem valor potencial ou real para uma organização". Neste projeto adotamos esta última definição.

Os ativos são classificados por tangíveis e intangíveis (Norma ISO/IEC 19770-1:2017). [7] Ativos tangíveis geralmente referem-se a ativos físicos como equipamentos, inventário e propriedade pertencente a organização. Ativos intangíveis são ativos não físicos. Segundo o SNC (Sistema de Normalização Contabilística) [9], um ativo intangível "é um ativo não monetário que não possui substância física". Alguns exemplos de ativos intangíveis são: marcas, direitos de uso, licenças e direitos de propriedade intelectual. Além disso, os colaboradores de uma organização, bem como a reputação da organização, também são ativos importantes que não devem ser esquecidos. Em termos de classificação, ativos intangíveis podem ser divididos de várias formas. Por exemplo, *o Finacial Accounting Standard Board* (FASB) 141, emitido em Junho de 2001, é a primeira norma contabilística, em termos mundiais, a enumerar exemplos de ativos intangíveis que satisfazem os critérios para o reconhecimento separadamente do mais intangível dos intangíveis, o *goodwill*. De acordo com a norma americana são cinco as principais categorias de ativos intangíveis: os relacionados com marketing, os relacionados com clientes, os relacionados com as artes, os baseados em contratos e os baseados em tecnologia. [10]

Os ativos intangíveis baseados em tecnologia são referidos por [11] como relacionados com inovações ou avanços tecnológicos. Normalmente, os benefícios económicos futuros desses ativos são protegidos por direitos legais ou contratuais. Bruce Mackenzie exemplifica este tipo de intangível como:

- Softwares;
- Tecnologia patenteada;
- Bases de dados;

- Tecnologia não patenteada;
- Segredos de Comercialização.

Por outro lado, a ISO 27001, classifica os tipos de ativos, e a compreensão vital do seu contexto de negócios, conforme a Figura 1.

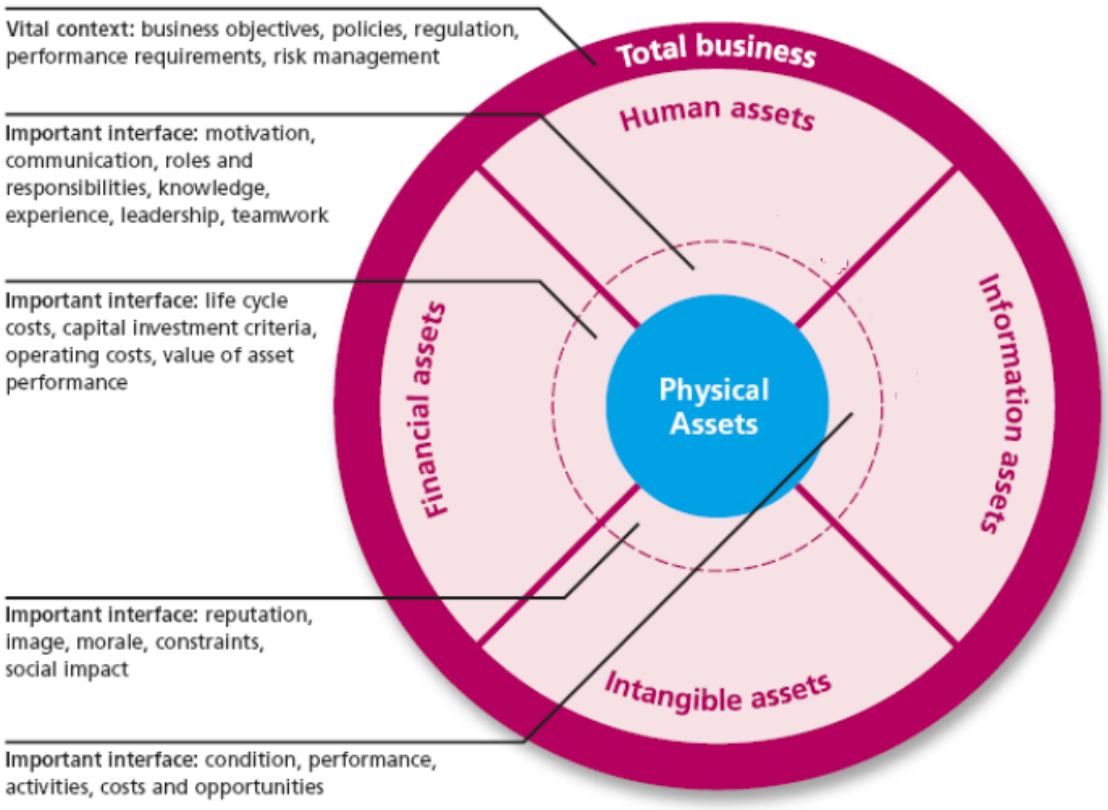


Figura 1 - Ativos e o seu contexto organizacional [6]

A maioria das empresas foca-se na gestão de ativos tangíveis e esquecem-se da importância da gestão dos ativos intangíveis. Verifica-se pela Figura 1, que os ativos tangíveis e intangíveis estão interligados. Logo, a falta de gestão de algum ativo irá afetar outros. Pretesh Biswas em [6], afirma que as organizações que dependem fortemente de ativos físicos devem reconhecer que deficiências na gestão de outros tipos de ativos podem ter um impacto profundo no desempenho geral ou de longo prazo de seus ativos físicos e, portanto, no seu desempenho organizacional. Todas as empresas devem reconhecer que todos os ativos precisam de ser geridos da mesma maneira. De acordo com o autor, para além dos ativos tangíveis, os ativos a ter em conta são:

- Ativos humanos: os comportamentos, conhecimento e competência da força de trabalho têm uma influência fundamental no desempenho dos ativos físicos.
- Ativos financeiros: recursos financeiros são necessários para investimentos em infraestrutura, operação, manutenção e materiais;

- Ativos de informação: dados e informações de boa qualidade são essenciais para desenvolver, otimizar e implementar o plano de gestão de ativos.
- Ativos intangíveis: a reputação e a imagem da organização podem ter um impacto significativo no investimento em infraestrutura, estratégias operacionais e custos associados.

Neste projeto consideramos os ativos de software e de informação como ativos intangíveis. Para uma organização ser eficiente, torna-se necessário gerir os seus ativos. Um dos ativos menos geridos pelas empresas são os ativos intangíveis, mais concretamente as licenças de software. Do ponto de vista de Jadir Breda em [12], cerca de 80% da implementação da gestão desses ativos nas empresas é devido a uma auditoria interna/externa e/ou pressões pela falta de gestão. Ou seja, só são iniciadas após uma causa negativa. Apenas 20% das empresas procedem a esta implementação por conhecerem os seus benefícios.

Para tratar destas particularidades, as normas ISO/IEC 19770 definem requisitos de gestão de ativos de TI. A gestão de ativos de TI é, de acordo com [13], um conjunto de práticas de negócios que unem funções financeiras, contratuais e de inventário para dar apoio à gestão do ciclo de vida de uma infraestrutura de TI. A gestão de ativos de TI fornece os meios para obter visibilidade completa do inventário da infraestrutura de TI e assim obter uma compreensão aprofundada de:

- Quais os sistemas e equipamentos existentes;
- Onde os componentes residem;
- Como são usados;
- Se estão licenciados;
- Se têm suporte;
- Quem os desenvolveu;
- Se têm um prazo de validade.

Uma *framework* muito importante para a gestão de ativos é a *Information Technology Infrastructure Library* (ITIL) [14]. Trata-se de um conjunto de publicações e boas práticas para a gestão dos serviços de TI. A série de livros do ITIL cobrem tópicos de gestão de TI que procuram alinhar os serviços de TI com as necessidades do negócio. Fornece às organizações orientação e descrição detalhada de várias práticas importantes com procedimentos e tarefas que podem ser personalizados para qualquer organização. O ITIL é a abordagem mais usada para a gestão de serviços de TI no mundo.

A ITIL v4 começa com o livro ITIL Foundation, que foi lançado em 18 de fevereiro de 2019. O ITIL v3 consiste numa série de 5 volumes. Cada volume cobre uma fase do ciclo de ITSM diferente. A Figura 2 identifica cada uma destas fases.

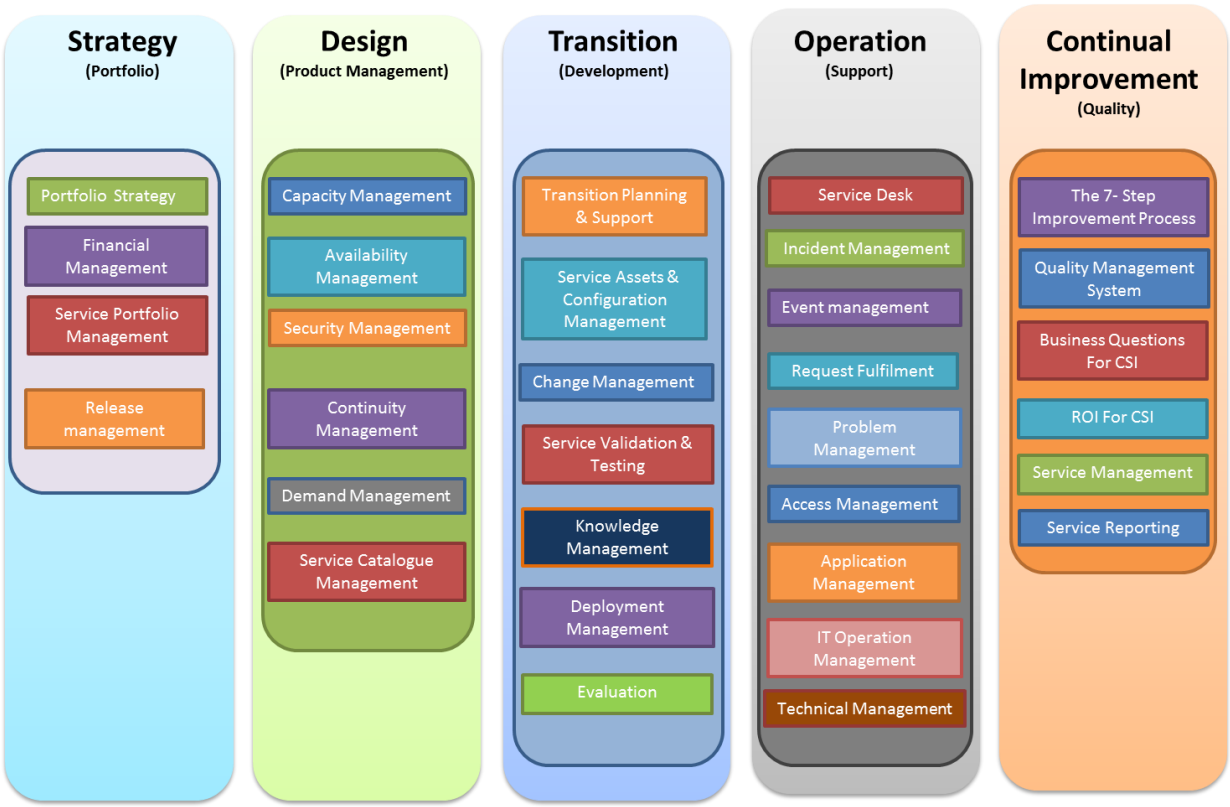


Figura 2 - Fase do ciclo de ITSM [15]

Cada uma destas fases, de acordo com [16], define um conjunto de orientações que procuram completar o ciclo referente aos diversos serviços, nomeadamente:

- Estratégia de Serviço: fase onde se define a direção estratégica dos serviços de TI, quem são os seus clientes e quais os serviços que serão disponibilizados.
- Desenho de Serviço: fase que inclui a avaliação dos processos de gestão do negócio (nível serviço, disponibilidade, capacidade, etc.) para desenhar e desenvolver novos serviços ou melhorar serviços já oferecidos.
- Transição de Serviço: fase que cobre a transição do desenvolvimento para as operações, incluindo testes e controlo de qualidade.
- Operação de Serviço: fase onde são coordenadas e executadas as atividades e processos necessários para entregar os serviços aos clientes e utilizadores, gerindo os níveis de serviços acordados.
- Melhoria Contínua de serviço: fase que procura manter os níveis de qualidade, sendo o seu propósito alinhar e realinhar continuamente os serviços de TI de acordo com as necessidades do cliente, identificando e implementando melhorias aos serviços de TI que suportam os processos de negócio.

Neste projeto focamos a atividade *Software Asset Management* (SAM). SAM é definido pela ITIL como toda a infraestrutura e processos necessários para a gestão, controlo e proteção efetiva dos ativos de software de uma organização, em todos os períodos do seu ciclo de vida. Fundamentalmente

destinado a fazer parte da estratégia de negócios de TI de uma organização, os objetivos do SAM são reduzir custos e limitar negócios e riscos legais relacionados à propriedade e utilização de software. SAM é particularmente importante para grandes organizações com relação à redistribuição de licenças e gestão de riscos legais associados à propriedade e expiração de software. As tecnologias SAM rastreiam a expiração da licença, permitindo que a empresa funcione de forma ética e dentro dos regulamentos de conformidade de software [14].

Segundo [17], a gestão de ativos de software é uma especialização da gestão de ativos de TI focada especificamente em ativos de software e que visa limitar riscos comerciais, legais e de segurança relacionados com a utilização de software. Atualmente, as empresas de forma a conseguirem ser competitivas necessitam de software e de tecnologia da informação (TI) eficientes para suportar os seus negócios. Uma das preocupações que as empresas têm é apresentar licenças legais devido à sua importância e por isso torna-se vital ter uma forma eficaz de as gerir. Periodicamente, os fornecedores de software auditam os seus clientes para garantir a conformidade com as normas aplicáveis, no entanto, tal não chega para a correta gestão das mesmas [18].

2.2 Benefícios da Gestão de Ativos

Na secção anterior verificamos que a gestão de ativos se baseia na importância de identificar, localizar, classificar e atribuir propriedade para os ativos mais importantes numa organização para garantir que sejam protegidos adequadamente. De modo a proteger efetivamente os dados, as organizações devem estar cientes dos riscos dos ciberataques e serem capazes de, no contexto de serviços de negócios, avaliar as vulnerabilidades que afetam a capacidade das empresas em realizar operações diárias. Da mesma forma, a concretização de ameaças à cibersegurança podem, sem saber, expor a propriedade intelectual de uma organização, além de colocar em risco os dados de privacidade de clientes, parceiros e funcionários. Assim, uma gestão de ativos de software eficaz deve permitir que as organizações abordem questões de cibersegurança, tais como:

- Gerir com segurança ativos de software através da elaboração de inventários de acordo com a norma ISO 27001:2013 A.8 Asset Management [6];
- Fornecer visibilidade total dos ativos para garantir uma infraestrutura segura e assim poder fornecer uma defesa mais eficaz contra-ataques, isto é, permitir às empresas ter consciência de todos os ativos que possuem de forma a saber como protegerem-se;
- Proteger a organização contra perda de dados tais como dados pessoais de clientes, fornecedores e até divulgações de produtos e processos internos [19].

De acordo com [20], existem vários benefícios na gestão de ativos de software, entre eles:

- Gerir de forma eficiente o consumo de licenças de software e o gasto com as mesmas;
- Reduzir o risco e custo por não conformidade contratual de licenças de software;

- Obter controlo sobre o estado do software com processos automatizados de SAM, através de pesquisas periódicas do software existente nos vários sistemas da organização;
- (Re)Negociar contratos a partir do uso de software na empresa e respetivas necessidades, incluindo o que é importante e como é utilizado.

Uma das grandes vantagens da utilização de um software de Gestão de Ativos é reduzir os custos empresariais bem como dar um melhor controlo financeiro. Tipicamente, segundo o apresentado em [21], um bom software de Gestão de Ativos fornece as seguintes funcionalidades:

- Prevenção da compra ou atualização desnecessária de licenças;
- Oferecer a capacidade de planeamento do futuro e criação de orçamento que reflita com maior precisão os custos sobre os seus ativos e processos;
- Tendo em conta que a organização detém o controlo sobre os seus ativos e processos, pode aproveitar e efetuar compras em grande escala sobre os itens corretos e assim usufruir dos descontos sobre estes;
- Controlo do ciclo de vida do ativo de forma constante, o que proporciona uma prevenção constante de situações desnecessárias e dispendiosas.

Um bom exemplo da importância de gerir os ativos é a análise feita pela empresa Gartner e publicada em [19]. Após iniciar a gestão de ativos, foi feita uma análise e verificaram que uma empresa pode reduzir entre 5% a 35% dos gastos relacionados com TI. Um outro exemplo é a pesquisa efetuada pela consultora Forrester Research. O estudo de 2008 mostra que a adoção de um plano de gestão de ativos consegue reduzir até 10% nos custos de suporte de tecnologia e alcançar uma economia média de 50 dólares por computador e 300 dólares por servidor.

Não gerir licenças de software pode também causar muitos prejuízos financeiros. Um bom exemplo apresentado em [22] é o caso do exército dos EUA que pagou cerca de 40 milhões de euros em multas após instalar aplicações não licenciadas no ano de 2013.

Em suma, segundo a empresa KPMG no relatório [23], a gestão de ativos de software é um conceito de negócio desenvolvido para reduzir custos de TI, diminuir riscos inerentes à propriedade e utilização de software bem como aumentar a eficiência das TI. Por exemplo, a existência de um inventário atualizado que permita avaliar adequadamente a localização, o estado e a utilização dos ativos existentes numa organização é, de acordo com [24], um aperfeiçoamento eficaz na gestão da cibersegurança.

2.3 Princípios da gestão de ativos

Para além dos benefícios da gestão de ativos, torna-se vital ter noção dos seus princípios da gestão de ativos. A gestão de ativos é uma visão holística que pode unir diferentes partes de uma organização em busca de objetivos estratégicos partilhados [6]. Os princípios e atributos-chave de uma gestão de ativos de sucesso são ilustrados na Figura 3.

Figura 3 - Princípios da gestão de ativos (Baseada em [6])

Em primeiro lugar, a gestão de ativos deve ser efetuada de forma holística. Deve-se olhar para o quadro completo, ou seja, as implicações combinadas de gestão de todos os aspetos. Isso inclui a combinação de diferentes tipos de ativos, as interdependências funcionais, as contribuições de ativos dentro de sistemas de ativos, as diferentes fases do ciclo de vida de ativos bem como as atividades correspondentes. Por outro lado, a gestão de ativos deve ser sistemática, ou seja, uma abordagem metódica, promovendo decisões e ações consistentes, repetíveis e auditáveis. Deve igualmente considerar os ativos no seu contexto de sistema de ativos e otimizar o valor dos sistemas de ativos, incluindo desempenho sustentável, custo e riscos, em vez de otimizar ativos individuais de forma isolada. A gestão também deve ser feita com base no risco, sendo que as empresas devem concentrar recursos e despesas e definir prioridades adequadas aos riscos identificados e aos custos versos benefícios associados. Outro princípio na gestão de ativos é ser ideal, ou seja, estabelecer o melhor compromisso de valor entre os fatores concorrentes, como desempenho, custo e risco, associados aos ativos ao longo dos seus ciclos de vida. A gestão deve considerar as consequências de longo prazo e atividades de curto prazo para garantir que seja feita a provisão adequada dos requisitos e obrigações futuras, como sustentabilidade económica ou ambiental, desempenho do sistema, responsabilidade social e outros objetivos de longo prazo. Por fim, a gestão deve reconhecer que as interdependências e os efeitos combinados são vitais para o sucesso. Isso requer uma combinação dos atributos acima, coordenados para fornecer uma abordagem conjunta e potenciadora de valor [6].

Uma organização, de modo a aplicar uma boa gestão de ativos, deve integrar os princípios descritos anteriormente.

2.4 Ciclo de vida dos ativos

Uma organização, de modo a apresentar uma boa Gestão de Ativos, necessita de ter conhecimento dos seus benefícios e princípios, mas também do ciclo de vida que os diversos ativos apresentam. Compreender que os ativos têm um ciclo de vida é um conceito-chave na Gestão de Ativos e, portanto, digno de análise. Existem dezenas de maneiras diferentes de representar o ciclo de vida, mas a Figura 4 faz uma representação simples segundo ISO 27001:2013 A. 8 Asset Management [6].

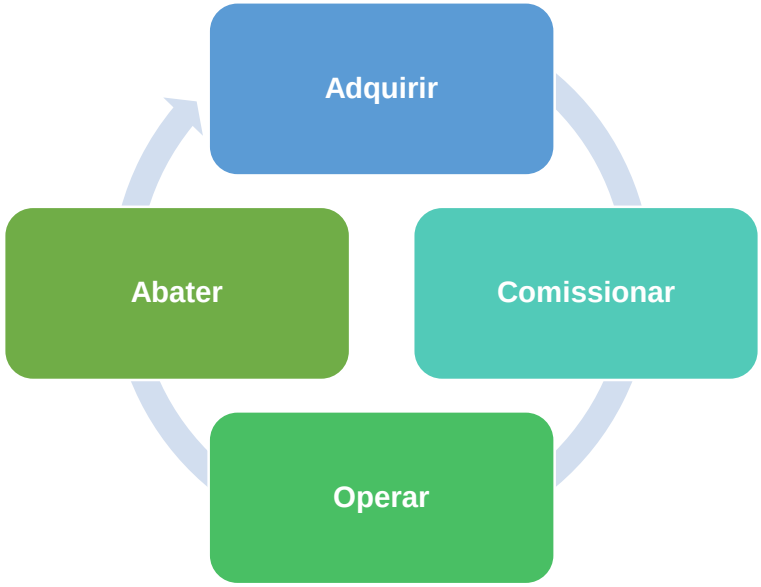


Figura 4 - Ciclo de vida dos ativos segundo a ISO 27001:2013 A.8 (Baseada em [6])

Pela Figura 4 verifica-se que o ciclo de vida tem quatro fases: adquirir, comissionar, operar e descartar.

A fase de adquirir abrange tudo o que é necessário para planear, projetar e comprar um ativo. Alguns diagramas de ciclo de vida capturam o planeamento como uma função separada. A aplicação adequada dessas atividades garante que o ativo é o adequado para a finalidade.

A fase comissionar abrange as atividades de instalação/criação ou construção do ativo e garantia de que está totalmente funcional. É facto reconhecido que existe uma maior incidência de falhas após a primeira instalação/construção de um ativo.

A fase de operar, normalmente, é a maior parte do ciclo de vida de um ativo durante o qual ele fornece a função para a qual foi projetado. Durante este período, o ativo deve estar sujeito a monitorização, manutenção, renovação e atualização adequada para atender a mudanças nas condições ou requisitos operacionais.

A fase de abate é geralmente a fase mais negligenciada. Os ativos podem durar séculos o que dificulta considerar o seu abate. Este período inclui as seguintes etapas:

- Remoção efetiva do ativo da operação;
- A eliminação ou reciclagem do mesmo ou dos seus componentes;
- A alimentação no planeamento para o ativo de substituição (se uma substituição for necessária) para determinar os requisitos operacionais com base na eficácia da operação e os modos de falha encontrados [6].

Relativamente a este projeto, o mesmo incide nas fases de operar e abater. Na fase de operar focamo-nos na monitorização dos ativos e na criação de alertas para a sua manutenção, renovação e atualização. No que diz respeito à fase de abate, esta é aplicada também através dos alertas devolvidos pela aplicação para que depois o administrador de sistemas possa remover os ativos.

2.5 Agentes Remotos

De modo a efetuar a gestão de ativos é necessário recorrer a uma solução que permita identificar e recolher ativos em múltiplos dispositivos. Uma solução possível passa pela utilização de agentes remotos, ou seja, uma componente de software que corre no sistema destino, recolhe a informação necessária e envia a mesma para um sistema remoto.

"O termo agente não tem significado único. Porém, pode ser compreendido como uma entidade que trabalha autónoma e continuamente. Não existe uma abordagem padrão para uma tecnologia baseada em agentes, nem mesmo uma definição única do que seja um agente. O conceito de agentes envolve um variado número de propostas relativas a teorias e arquitetura de agentes." [25]. Segundo Neide dos Santos, um agente é considerado um assistente de tarefas, isto é, uma entidade de software que emprega técnicas de Inteligência Artificial com o objetivo de assistir o utilizador na realização de determinada tarefa, agindo de forma autónoma. Este autor considera também um agente como um programa independente que pode realizar uma ou mais tarefas em prol do utilizador ou máquina. Os utilizadores precisam comunicar ao agente 'o que' e 'quando' fazer alguma coisa, mas não "como" fazer alguma coisa [26]. Do ponto de vista de Wooldridge, M. J. & Jennings, N. R. um agente pode ser visto como um sistema computacional localizado num ambiente e ser capaz de realizar ações autónomas nesse ambiente para corresponder aos seus objetivos [27]. O ambiente referido anteriormente é apresentado de forma genérica, podendo ser entendido como qualquer meio físico ou lógico, composto por aspetos heterogêneos ou não. As ações autónomas que um agente pode desempenhar num ambiente, podem ser quaisquer ações que não precisem de intervenção humana para ser realizadas. Contudo, essa definição compreende apenas os agentes reativos. Por outro lado, Björn Hermans em [28] refere que um agente é normalmente um programa de software que apoia um utilizador na realização de alguma tarefa ou atividade. Os autores Vavasori, F. B. & Gauthier, F. A. em [29] referem que um agente é uma entidade de software que realiza um conjunto de operações em benefício do utilizador ou de outro programa, utilizando certo grau de independência ou autonomia. Ao fazê-lo, emprega algum conhecimento ou representação dos objetivos ou preferências do utilizador. Segundo a definição de Jacques, A. P. & Oliveira, M. F., em [30] agente é uma entidade virtual ou real, que está apta para perceber e representar parcialmente o seu ambiente. Um agente também possui a

capacidade de comunicar com outros agentes e pode assumir um comportamento autónomo que é uma consequência de suas observações, de seu conhecimento e de suas interações com outros agentes. Ferreira, F. L. & Bercht, M., em [31] consideram um agente como uma "entidade real ou virtual, imersa num ambiente no qual é capaz de agir, dispondo de capacidade de perceção e de representação parcial deste ambiente, podendo comunicar com outros agentes e possuir um comportamento autónomo, consequência das suas observações, do seu conhecimento e das suas intenções com outros agentes." Espera-se que os agentes assumam um papel importante no mundo da TI para resolver os problemas complexos do mundo real. Estes comportam-se como humanos de maneira inteligente, autônoma, cooperativa e social para resolver problemas ou dar suporte a utilizadores [32]. Os agentes podem ser caracterizados pelas suas funcionalidades e devem ser descritos pelas características que podem ou não possuir. As características mais comuns são autonomia, orientação a objetivos, colaboração, flexibilidade, comunicação, adaptabilidade e mobilidade. Os agentes têm a capacidade de iniciativa e controlar as suas próprias ações, além de funcionarem sem a intervenção direta de um humano. São capazes de realizar tarefas complexas e tomar a decisão de como estas tarefas serão divididas e qual a ordem de execução. Além disso, tem a habilidade para alterar pedidos, para aperfeiçoar ou rejeitar os pedidos já satisfeitos e comunicar os resultados à aplicação central. Escolhem dinamicamente que ações evocar, em resposta ao estado do seu ambiente externo. Também comunicam com outros agentes para obter informações. Além disso, automatizam preferências dos seus utilizadores e adaptam-se às mudanças no seu ambiente. Podem ser transportados de uma máquina para outra e atravessar diferentes arquiteturas de sistemas e plataformas [33].

Neste projeto seguimos a definição de agentes remotos definido por Vavasori, F. B. & Gauthier, F. A., apresentada em [29], ou seja, a designação de agente no âmbito do projeto deve ser vista como um componente de software que irá executar de forma independente e automática tarefas no lugar de um humano. Nesta fase não está contemplada a criação de agentes inteligentes, mas sim, um agente capaz de receber instruções sobre a recolha de dados de ativos e executar as mesmas nos sistemas onde a recolha deve ser feita. No entanto, idealmente os agentes seriam capazes de evoluir e absorver conhecimento autonomamente como por exemplo, com a partilha de conhecimento entre agentes.

2.6 Gestão de Vulnerabilidades

A Gestão de Vulnerabilidades tem como objetivo antecipar a gestão de segurança de redes ao mitigar os riscos de falhas e *exploits* em códigos ou arquiteturas que possam comprometer *endpoints* ou ativos de rede. A Gestão de Vulnerabilidades baseia-se em práticas e processos de rotina. Esta gestão tem como objetivo diminuir as falhas. De forma a efetuar esta gestão deve-se seguir os seguintes passos. Em primeiro lugar, as empresas devem procurar vulnerabilidades. Esse processo deve incluir scans periódicos de rede, logs de firewall, *pentests* ou uso de ferramentas automatizadas como um scan de vulnerabilidades. Depois devem identificar as vulnerabilidades. Esse processo envolve a análise dos scans de rede e *pentests*. Essas análises podem encontrar anomalias que

sugerem ataques de *malware* ou outras atividades maliciosas que tenham tomado vantagem sob uma vulnerabilidade [34].

De seguida, encontra-se a Tabela 1, onde são referidos alguns exemplos de vulnerabilidades.

Tabela 1- Exemplos de vulnerabilidades [35]

Ativo	Ameaça	Vulnerabilidades	Risco
Documento em papel	Fogo	Documento não é armazenado em armário à prova de fogo	Risco relacionado a perda de disponibilidade da informação
Documento em papel	Fogo	Não existe cópia de segurança do documento	Potencial perda de disponibilidade
Documento em papel	Acesso não autorizado	Documento não está trancado em um armário	Potencial perda de confidencialidade
Documento digital	Falha de disco	Não existe cópia de segurança do documento	Potencial perda de disponibilidade
Documento digital	Vírus	Programa antivírus não está adequadamente atualizado	Potencial perda de confidencialidade, integridade e disponibilidade
Documento digital	Acesso não autorizado	Esquema de controlo de acesso não é definido apropriadamente	Potencial perda de confidencialidade, integridade e disponibilidade
Documento digital	Acesso não autorizado	O acesso foi atribuído de forma indevida	Potencial perda de confidencialidade, integridade e disponibilidade
Administrador do sistema	Indisponibilidade desta pessoa	Não há substituto para esta posição	Potencial perda de disponibilidade

Administrador do sistema	Erros frequentes	Falta de formação	Potencial perda de informação e disponibilidade

O processo seguinte é verificar as vulnerabilidades. Esse processo inclui averiguar se as vulnerabilidades identificadas podem de fato ser exploradas em servidores, aplicações, redes e outros sistemas. Isso também inclui a classificação de severidade de uma vulnerabilidade e o nível de risco que ela apresenta à empresa. De seguida deve-se mitigar as vulnerabilidades. Esse processo consiste em descobrir, de acordo com os recursos e limitações da empresa, como prevenir essas vulnerabilidades de serem exploradas antes que um *patch* de correção esteja disponível, ou como aplicar o *patch* da forma mais rápida possível.

Por fim, aplicar *patches*. Esse é o processo de colocar os *patches* disponibilizados pelos fabricantes e aplicá-los em todos os sistemas presentes no ambiente em tempo útil. Muitas vezes esse processo pode ser automatizado [34].

No entanto, de forma a poder aplicar a gestão de vulnerabilidades efetiva é necessário que uma empresa tenha os ativos identificados e catalogados. Ou seja, antes de tudo, é preciso mapear quais ativos são cruciais para o dia a dia operacional e estratégico do negócio. Quais destes ativos de informação, se danificados, podem parar a operação? Este mapeamento é deveras importante, precisamente para garantir que os controlos são aplicados em todos os recursos relevantes do negócio. Depois da identificação de todos os ativos é de suma importância saber as consequências perante a perda, dano ou roubo de ativos. Qual é o tipo de impacto que acontece caso cada ativo seja comprometido? Se a informação estiver indisponível por um período de tempo, qual é o impacto e o seu prejuízo? Com estas questões, a empresa irá entender o grau de risco em caso de indisponibilidade, perdas, roubos ou alterações das informações. Só depois podemos aplicar a gestão de vulnerabilidades e as várias etapas descritas anteriormente.

Por fim, deve-se testar as reais vulnerabilidades das informações. Os testes vão demonstrar se existe risco real das informações serem perdidas ou não. A partir disso, retire quais são as opções para proteger as informações importantes para a empresa. O processo de gestão de vulnerabilidades está intrinsecamente interligado com o processo de riscos de segurança da informação. A gestão de riscos de segurança da informação é cíclica, deve ser revista, e atualizada dentro de períodos regularmente definidos [36].

2.7 Soluções de mercado existentes

No mercado atual existem soluções tecnológicas para efetuar uma gestão de ativos das organizações. No entanto, tal como é apresentado de seguida, a maioria foca-se mais em dispositivos ou em software aplicacional e são normalmente dispendiosas, o que torna impraticável a sua aquisição

para a maioria das organizações. Existem também algumas soluções que são muito acessíveis. Contudo, carecem de robustez e de capacidade de funcionamento, nomeadamente de recolha de informação a um nível esperado por uma organização empresarial. Das soluções analisadas, nenhuma efetua a interligação com serviços de vulnerabilidades, algo que neste trabalho é considerado como fundamental para aumentar a segurança relacionada com a posse de ativos de software ou de informação.

Das várias soluções, achamos essencial referir as seguintes: Manage Engine Asset Explorer, InvGate Assets, Asset Tiger e Snipe-IT.

2.7.1 Manage Engine Asset Explorer

O AssetExplorer é uma ferramenta de gestão baseada em Web IT Asset Managament (ITAM), que ajuda a monitorizar e gerir os ativos de uma empresa, focando todo o ciclo de vida dos ativos. A ferramenta dispõe de várias formas para reconhecer a situação e localização dos ativos numa determinada rede. Podem-se gerir os ativos de software e hardware, assegurar o cumprimento de licenças de software e seguir as ordens de compra e contratos. É de fácil instalação e é simples de manusear. O AssetExplorer ajuda a entender a conformidade das licenças de software e o uso de software não autorizado na organização e auxilia a agir de forma proactiva para conter o uso ilegal e os problemas decorrentes do mesmo. Esta ferramenta gere todos os tipos de licenças. Os tipos de licença para Microsoft, Adobe e Symantec veem configurados por padrão [37].

Esta solução é caracterizada pelas seguintes funcionalidades [38]:

- Descoberta de ativos na rede;
- Gere e monitoriza ativos de software e hardware;
- Gere o ciclo de vida completo dos ativos de TI;
- Garante a conformidade das licenças de software;
- Rastreia pedidos e contratos de compra.

Por outro lado, apresenta pontos fracos, tais como:

- Só está disponível para o sistema operativo Windows;
- A versão profissional pode ser inacessível para muitas pequenas empresas, pois é demasiado elevado para a quantidade dos ativos que possui;
- A interface Web e algumas funções poderiam ser simplificadas.

O AssetExplorer é comercializado focando-se em cinco fases da Gestão de Ativos, que estão ilustrados na Figura 5.

Figura 5 - As cinco fases em foco no AssetExplorer [39]

As cinco fases ilustradas na Figura 5 constituem o ciclo de vida do ativo no contexto do AssetExplorer. Este ciclo engloba:

1. A aquisição do ativo;
2. A sua implementação no negócio e sua descoberta;
3. A manutenção desse ativo;
4. O suporte que pode ser fornecido a este;
5. A substituição do respetivo ativo ou a sua desativação.

Para visualização da solução AssetExplorer basta aceder ao link http://demo.assetexplorer.com/. Através deste endereço é apresentada a janela de autenticação da aplicação.

Ao carregar no respetivo Login da aplicação é apresentado o aspeto global da solução AssetExplorer. Tal como se pode ver pela Figura 6, a disposição da aplicação é simples e amigável para os utilizadores, sendo composta por Menus numa frame do lado esquerdo da janela e um frame principal centralizada, onde é exposta toda a informação da interação com a aplicação. A informação pode ser visualizada em formato gráfico ou em forma de formulários.

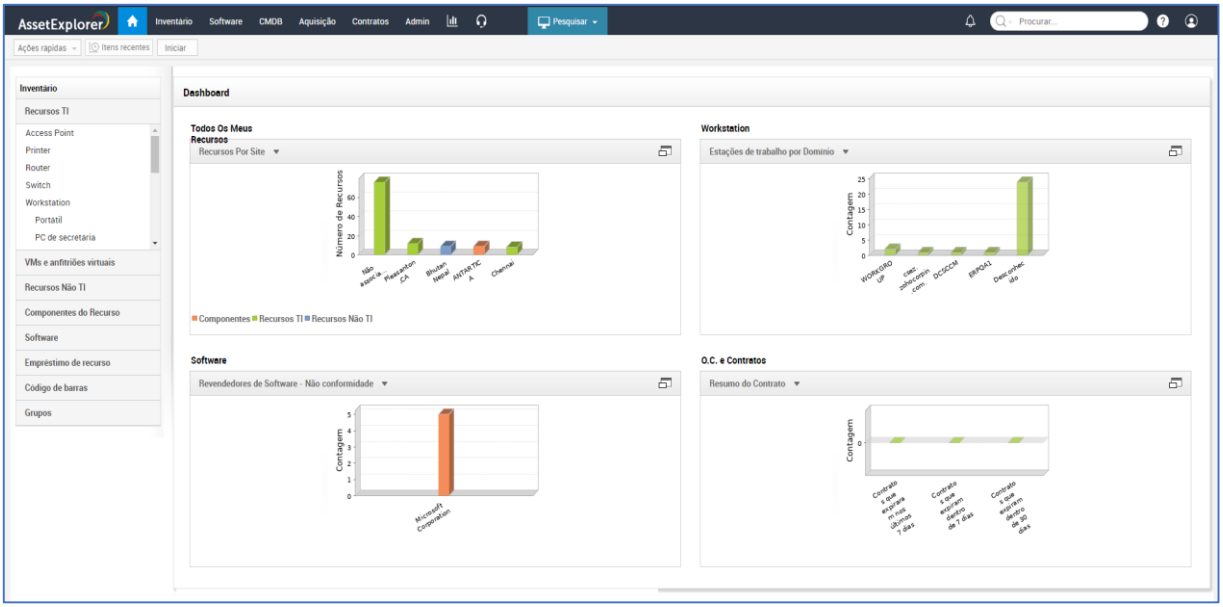


Figura 6 - Asset Explorer Demo Main Menu

2.7.2 InvGate Assets

A InvGate Assets é uma ferramenta ITAM e SAM (Gestão de Ativos de Software) orientada ao cumprimento de objetivos.

A gestão de Ativos de TI é geralmente uma funcionalidade corporativa subvalorizada, já que insiste mais nas atividades de gestão de serviços (ITSM), na necessidade de preservar o negócio e em assegurar que se esteja a utilizar o nível correto de ativos TI. Consequentemente, as empresas podem estar a perder uma quantidade significativa dos seus orçamentos de TI, dado que por norma não têm uma visão global dos gastos integrada com a respetiva utilização dos ativos. Exemplos de desperdícios incluem, novas despesas com aquisição de software e hardware quando estes poderiam ser reutilizados ou adaptados a uma nova utilização, gasto com suporte prestado por terceiros ou ainda gastos em manutenção em ativos de TI que não são utilizados. Porém, as organizações mais conscientes percebem como utilizar dados, técnicas e ferramentas de gestão de ativos de TI para maximizar o *Return Of Investment* (ROI) em ativos de TI, diminuir *o Total Cost of Ownership* (TCO) e reduzir os riscos por exemplo de incumprimento legal devido à falta de licenciamento de software.

O InvGate é caracterizado pelas seguintes funcionalidades [40]:

- Inventário de ativos;
- Monitorizar ativos;
- Administração de Licenças de Software;
- Gerir o ciclo de vida dos ativos;
- Network Discovery.

Por outro lado, apresenta pontos fracos, tais como:

- É uma aplicação que tem de ser instalada em todos os postos da empresa bem como o motor de base de dados. Esta ocupação de recursos para certas máquinas pode ser negativa;
- Só está disponível para o sistema operativo Windows;
- O preço pode ser inacessível para muitas pequenas empresas, pois é demasiado elevado para a quantidade dos ativos que possui.

O demo da aplicação pode-se obter a partir do *link* https://www.invgate.com/pt/free-trial/. No entanto, é necessário o preenchimento de um formulário para poder obter o demo. Após o correto preenchimento do formulário, basta aceder ao email e fazer o respetivo *download*. Depois da aplicação instalada e executada irá ser apresentada a janela principal do InvGate Assets, tal como ilustrado na Figura 7.

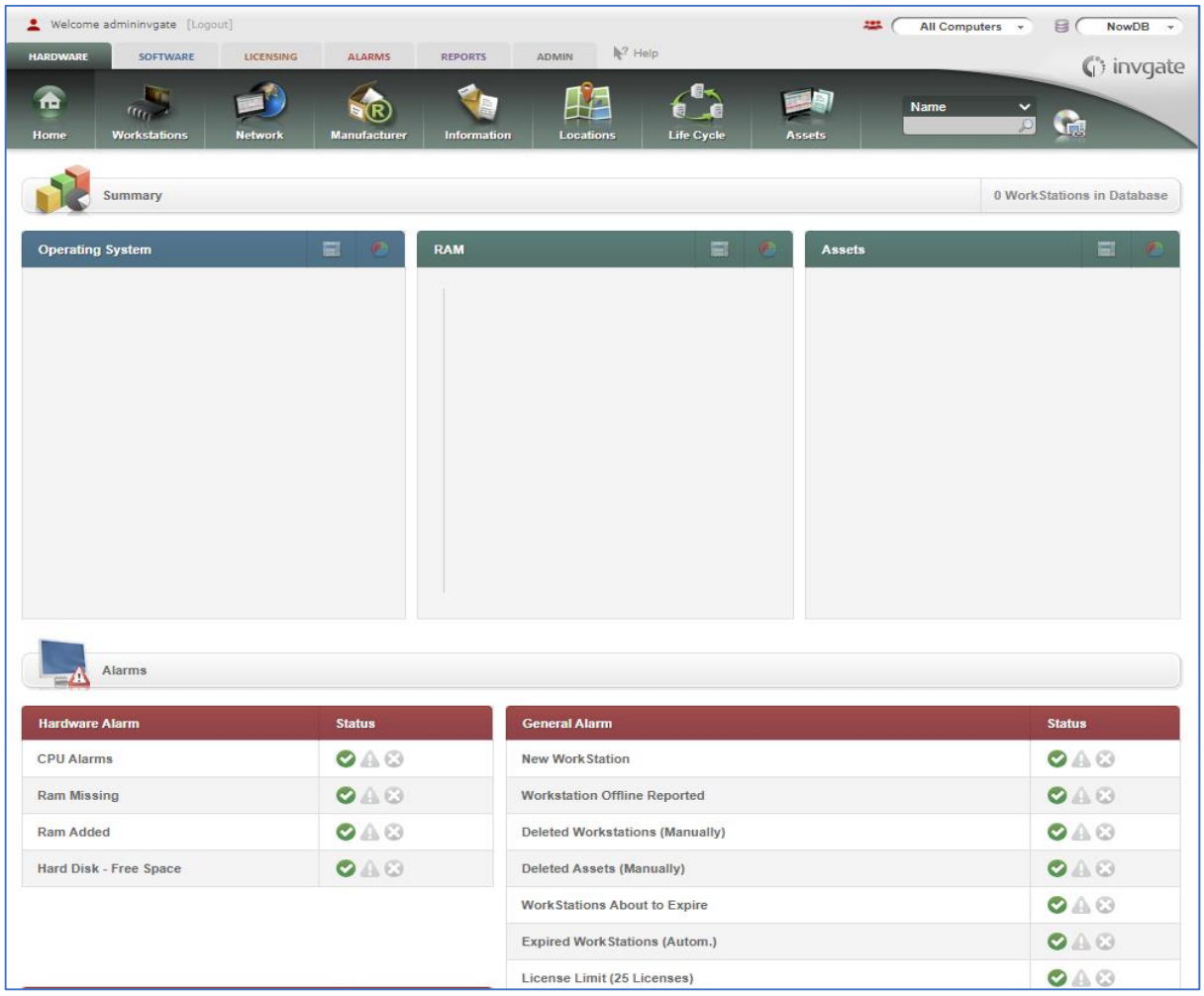


Figura 7 - Menu Solução InvGate

2.7.3 Asset Tiger

AssetTiger é uma ferramenta de software *cloud-based*. A mesma permite que os utilizadores mantenham o controlo de *check-ins* / *check-outs* de ativos bem como a monitorização e o registo de todas as interações dos ativos. Com esta ferramenta também podemos realizar auditorias internas e definir alarmes para evitar a falha de manutenção de ativos. Os utilizadores podem gerir licenças de ativos e definir alertas de email para renovações.

AssetTiger é o primeiro programa de gestão de ativos totalmente funcional e totalmente gratuita. A versão gratuita suporta até 250 ativos.

O AssetTiger é caracterizado pelas seguintes funcionalidades [41]:

- Inventário de ativos;
- Monitorizar ativos;
- Capacidade de produção de relatórios;
- Gerir o ciclo de vida dos ativos.

Por outro lado, apresenta pontos fracos, tais como:

- Dispõe de pouco suporte tanto a nível de documentos como canais de ajuda a novos clientes;
- Não possui pesquisa de ativos na rede;
- O plano gratuito é limitado a 250 ativos.

A demo da aplicação pode ser feita através de um registo a partir do *link* https://www.myassettag.com/assettiger. Após o registo, basta aceder ao email para ser feita a respetiva validação. Após conclusão destes passos poderemos então testar a aplicação e adicionar ativos bem como criar alertas como ilustrado na Figura 8.

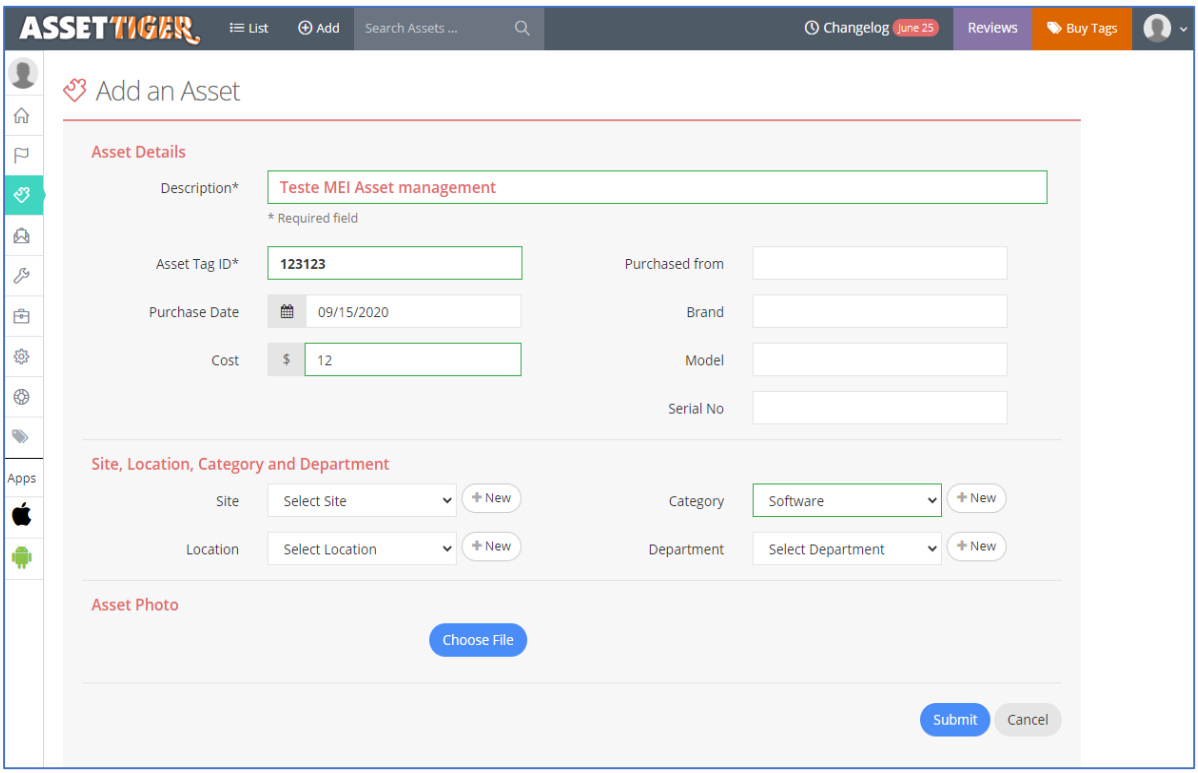


Figura 8 - Adição de um novo ativo, AssetTiger

2.7.4 Snipe-IT

Snipe-IT é uma ferramenta *open source*. Permite o rastreamento ponta a ponta de ativos, incluindo o seu histórico (*check-ins, check-outs* e manutenção), status atual (instalado, pendente, arquivado, roubado, etc.) e localização.

Além destas funcionalidades, o software oferece recursos de suporte a auditorias de ativos, alertas de vencimento de garantias e licenças, assinaturas digitais, gestão de licenças, relatórios, integração com leitores de código de barras e leitores de QR code.

Snipe-IT oferece um plano que é gratuito, acomoda utilizadores e ativos ilimitados e oferece suporte à comunidade GitHub.

O Snipe-IT é caracterizada pelas seguintes funcionalidades [42]:

- Inventário de ativos;
- Monitorizar ativos;
- Capacidade de produção de relatórios;
- Gerir o ciclo de vida dos ativos;
- Integração de notificações com a aplicação Slack;
- Auditoria de ativos rápida e fácil;

- Aplicação multiplataforma.

Por outro lado, apresenta pontos fracos, tais como:

- A versão gratuita não possui suporte;
- Não possui pesquisa de ativos na rede;
- Aplicação difícil de instalar.

A demo da aplicação pode ser feita através de um registo a partir do *link* https://www.myassettag.com/assettiger. Após o registo, basta aceder ao email para ser feita a respetiva validação. Após conclusão destes passos, poderemos então testar a aplicação e adicionar ativos bem como criar alertas como ilustrado na Figura 8.

2.7.5 Comparação das Soluções Tecnológicas

A Tabela 2 apresenta o sumário das características das quatro soluções referidas anteriormente, bem como a comparação entre elas.

Tabela 2 - Tabela de comparação de Características das Soluções Alternativas

Solução / Características	Asset Explorer	InvGate Assets	Asset Tiger	Snipe-IT
Instalação Centralizada	Sim	Não	Não	Sim
Preço licenças **+++ Muito Alto** **++ Alto** **+ Razoável** **- Baixo**	++	+++	++	+
Flexibilidade	Não	Não	Não	Não
User-Friendly	Sim	Não	Sim	Sim
Multiplataforma	Não	Não	Sim	Sim
Relatórios customizáveis	Sim	Sim	Sim	Sim

Disponibilização de Indicadores gráficos	Sim	Sim	Sim	Sim
Baseado em tecnologia *Open Source*	Sim	Não	Não	Sim
Web Based	Sim	Sim	Sim	Sim
Verificação de Vulnerabilidades	Não	Não	Não	Não
Pesquisa automática de Ativos	Sim	Não	Não	Não

Tendo como base a tabela anterior, pode-se efetuar a análise critica das vantagens e desvantagens das soluções, tendo em conta que as soluções que apresentam as caraterísticas referidas apresentam vantagem, o fato de não as terem é considerado desvantagem:

- Instalação centralizada – É uma caraterística que é imprescindível na atualidade, pois possibilita que as soluções sejam instaladas apenas uma vez e num local. Neste caso apenas as soluções “InvGate Assets” e “Asset Tiger” não apresentam uma instalação centralizada, o que limita a portabilidade dos sistemas, que por sua vez é uma desvantagem.
- Preços das licenças – O preço da solução é muito importante para o cliente, tendo em conta se os seus orçamentos são suficientemente reduzidos. Neste sentido, pode-se verificar na tabela acima que em termos de qualidade versus preço a solução “Snipe-IT” são as que mais se destacam.
- Flexibilidade – Atualmente a flexibilidade do software é muito importante porque o negócio das empresas está em constante mutação e as soluções de software devem acompanhar essas alterações. Relativamente às soluções analisadas verificou-se que são todas indesejavelmente rígidas relativamente a desenvolvimentos à medida, portanto pouco flexíveis.
- *User-Friendly* - Um fator importante na adaptação das soluções aos seus utilizadores é a facilidade que estes têm em navegar pela solução e para executar as ações que permitam resolver os seus problemas, em termos de software. Neste aspeto apenas se destaca negativamente a solução “InvGate Assets”.
- Permite criação de relatórios customizáveis – A extração de relatórios é muito importante para as empresas, pois estes poderão ter informação importante para efetuar decisões de negócios que permitam evitar gastos desnecessários. Se esta funcionalidade for editável torna-se mais flexível, possibilitando que os utilizadores escolham que tipos de informação são extraídos do software. Todas as soluções têm esta funcionalidade, o que é considerado uma vantagem.

- Indicadores Gráficos – Esta caraterística é um complemento à referida anteriormente e permite visualizar informação através de diferentes tipos de gráficos (gráficos de linhas, barras, pie, etc.). Todas as soluções têm esta funcionalidade.
- Baseado em tecnologia Open Source – Esta caraterística pode ser considerada uma vantagem do ponto de vista tecnológico e de preço do produto final, tendo em conta que a tecnologia Open Source possibilita desenvolvimentos mais flexíveis e livres de licenças, o que baixa o preço do produto final. Neste aspeto destacam-se negativamente as soluções “InvGate Assets” e “Asset Tiger”.
- Web Based – É uma das soluções atuais do mercado, que possibilita aos utilizadores acederem à solução através da internet ou intranet e de um browser. Todas as soluções têm a vantagem de dispor desta funcionalidade.
- Verificação de Vulnerabilidades – É um dos elementos chave para que as empresas estejam atualizadas sobre o tipo de vulnerabilidades que podem ter de enfrentar e assim conseguirem reagir antecipadamente. Nenhuma das ferramentas analisadas neste projeto apresenta esta funcionalidade ou indicam planos para a sua inclusão.
- Pesquisa automática de Ativos – A par da anterior, esta é uma característica fundamental visto que leva a um levantamento exaustivo e mais correto dos ativos existentes das empresas. Esta funcionalidade apenas pode ser encontrada na solução “Asset Explorer”.

Capítulo 3 – Metodologia de investigação

Tendo como base a revisão da literatura apresentada no capítulo dois, aqui descreve-se a metodologia de investigação utilizada neste projeto, de forma a alcançar os objetivos propostos e referidos na introdução.

Este capítulo divide-se em quatro seções. A primeira seção refere-se à Caracterização da Entidade de Acolhimento. A segunda secção denominada por Ferramentas utilizadas, dá ênfase às ferramentas escolhidas para a elaboração do projeto. Na terceira secção, Cenário de implementação, é feita uma descrição da arquitetura da ferramenta E-Sam (Evolutive Asset Management). Na quarta e última seção, cujo título é Desenvolvimento e configuração da solução, são indicados os procedimentos executados durante o desenvolvimento deste projeto.

3.1 Caracterização da entidade de acolhimento

Este projeto foi desenvolvido na empresa AET Europe. Fundada em 1998 é considerada líder global na área de soluções de segurança digital. Esta organização é especializada na criação de soluções seguras de identificação, autenticação, assinatura digital, consentimento e gestão de credenciais.

Em termos de missão, a AET Europe compromete-se a “Criar uma solução tecnológica perfeita na identificação do utilizador autenticado e autorizado”. Para além disso, tem como visão “Todos poderem beneficiar da tecnologia que oferecem” [43].

Na Figura 9 podemos ver as três soluções tecnológicas que a AET Europe oferece:

Figura 9 - Soluções tecnológicas AET Europe

Devido ao aumento de pedidos de funcionalidades de gestão de ativos por parte de empresas cliente, houve a necessidade de criar uma ferramenta que permita fetuar a gestão das licenças de software complementando outros serviços já disponibilizados noutros produtos da empresa

3.2 Ferramentas utilizadas

Para o desenvolvimento da ferramenta E-Sam (Evolutive Asset Management) optou-se pela utilização da linguagem Python. Esta linguagem de programação criada por Guido van Rossum é independente de plataforma, pronta para executar qualquer tipo de programa. É uma linguagem interpretada, o que significa que não é preciso compilar o código-fonte para executá-lo, o que oferece

vantagens como velocidade de desenvolvimento e também desvantagens, como menor velocidade de execução.

O Python tornou-se muito popular devido a algumas das suas características:

- É gratuito, mesmo para fins comerciais;
- O número de bibliotecas que dispõe, tipos de dados e funções incorporadas que ajudam a executar muitas tarefas comuns sem precisar programar do zero;
- A simplicidade e rapidez com que os programas são criados;
- Um programa Python pode ter 3 a 5 menos linhas de código do que seu equivalente em Java ou C;
- O número de plataformas em que podemos desenvolver, como UNIX, Windows, OS/2, Mac e outros [44].

O Python apresenta várias outras características: utilização genérica, multiplataforma, interpretada, interativa, orientada a objetos, funções e bibliotecas e sintaxe clara. Relativamente à primeira característica referida, a linguagem de programação Python permite criar muitos tipos de programas, tais como:

- Aplicações de negócios para capturar, analisar e processar dados;
- Aplicações Web dinâmicos;
- Jogos em 2D e 3D;
- Aplicações financeiros e científicas;
- Aplicações móveis [45].

A característica multiplataforma deve-se ao facto das versões do Python estarem disponíveis em muitos sistemas de computador diferentes. Ser interpretado significa que o código não é compilado para um determinado ambiente computacional, sendo capaz de executar em qualquer ambiente usando para tal um interpretador. Na realidade, é aplicada uma compilação, mas é realizada de forma transparente para o programador. Em determinados casos, quando se executa o código pela primeira vez, o bytecode[1] é guardado no sistema e servem para adequar o programa ao ambiente tornando a execução do código mais eficiente. Por outro lado, o Python é interativo por ter um interpretador de linha de comandos no qual podemos inserir instruções. Cada instrução executa e produz um resultado visível, que pode ajudar a entender melhor a linguagem e testar os resultados da execução de partes de código rapidamente. A programação orientada a objetos é suportada em Python e, em muitos casos, oferece uma maneira fácil de criar programas com componentes reutilizáveis. O Python tem muitas funções incluídas na própria linguagem, para o tratamento de números, ficheiros, entre outros. Além

[1] é um idioma intermediário mais abstrato entre linguagem de máquina e linguagem de programação, usado para descrever as operações que compõem um programa

disso, existem muitas bibliotecas que podem ser importadas no código para conseguir interagir com outros componentes específicos. Por fim, pode-se dizer que o Python tem uma sintaxe muito visual, graças a uma notação indentada (com margens) de conformidade obrigatória [44].

Resumidamente, no que se refere a linguagens de programação, o Python é relativamente simples e eficiente. Essa combinação contribuiu para a sua imensa popularidade. Além da linguagem de programação em si, o Python apresenta um elevado número de bibliotecas que podem reduzir tarefas complexas a apenas algumas linhas de código [45].

Para facilitar a implementação utilizou-se o *integrated development environment* (IDE) Pycharm. Este fornece análise de código e um depurador gráfico. É multiplataforma, com versões para Windows, macOS e Linux. A versão instalada, foi o PyCharm Community Edition 2020, que é gratuito e Open Source [46].

Para disponibilização de uma *dashboard* para o utilizador final utilizou-se o Microsoft Power BI que é uma ferramenta de *Business Intelligence* da Microsoft. O Power BI é um pacote de ferramentas de análise de negócios que oferece *insight*[2] às empresas. Tem como função a ligação a várias de fontes de dados e assim simplifica a recolha e preparação dos dados. Permite elaborar relatórios que podem ser acedidos via Web browser e através dos dispositivos móveis [47]. Com esta ferramenta é possível consolidar, tornar coerentes e representar de forma visual as informações que se encontram em diversas fontes. O Power BI pode por exemplo ser utilizado de forma simples para pequenos projetos no Excel e base de dados local, ou até mesmo para grandes e robustos projetos a nível empresarial. Com esta ferramenta é possível criar uma visão 360 graus sobre a informação relativa a uma dada organização atualizada em tempo real [48]. Pode-se ainda entender o Power BI como um serviço online através do qual se pode criar *dashboards*, partilhar relatórios e incorporar todos os dados que são importantes para as organizações. No servidor central da ferramenta E- Sam foi instalado o Power BI Desktop, uma ferramenta dedicada de criação de relatórios que permitiu transformar dados, criar relatórios e visualizações avançadas e publicar as mesmas facilmente através do serviço Power BI [47].

Marcelo de Tarcio em [49] afirmou haver 28 motivos para se usar o Power BI e entre os mais relevantes apontou:

- Power BI interliga-se com praticamente todas as fontes de dados existentes;
- Power BI tem alta performance e alta taxa de compressão;
- O sistema de visualização do Power BI é novo e *user friendly*;
- Há sempre novas visualizações a ser criadas;
- Power BI está na nuvem;

[2] *Insight* é a compreensão de uma causa e efeito específicos dentro de um determinado contexto.

- Power BI tem uma aplicação para Smartphone e Tablets como ilustrado no apêndice 1;
- Podemos receber relatórios via email automaticamente;
- Power BI é muito barato e possui também uma versão gratuita;
- Tem um controlo de acesso e segurança robusto;
- O Power BI tem um ciclo de atualização contínuo. Atualmente há uma periodicidade mensal de atualização. As melhorias são feitas tanto no Power BI Services, na aplicação mobile e também no Power BI Desktop.
- Power BI tem um sistema de pesquisa em linguagem natural;
- Grandes empresas já aderiram ao Power BI e será a ferramenta de análise mais utilizada em breve como podemos observar nas figuras seguintes [49].

Nos gráficos seguintes apresentamos um estudo da Gartner sobre a evolução do Power BI desde o ano de 2015. Este estudo mostra o quanto, segundo a Gartner, o Power BI tem evoluído em termos de posicionamento no mercado, sendo hoje em dia uma solução líder.

Figura 10 - Magic Quadrant for BI & Analytics Platforms 2015 (Gartner, 2017)

Figura 11 - Magic Quadrant for BI & Analytics Platforms 2017 (Gartner, 2017)

Para que seja possível armazenar os dados recolhidos pela ferramenta E-Sam utilizou-se o sistema de gestão de base de dados PostgreSQL. O PostgreSQL é muito poderoso, *Open Source* e estende a linguagem *Structured Query Language* (SQL) combinada com muitos recursos que armazenam e escalam com segurança as cargas de trabalho. As origens do PostgreSQL remontam a 1986 como parte do projeto POSTGRES na Universidade da Califórnia em Berkeley e conta já com mais de 30 anos de desenvolvimento ativo na plataforma central. PostgreSQL ganhou uma forte reputação devido à sua arquitetura comprovada, confiabilidade, integridade de dados, conjunto de recursos robustos, multiplataforma, extensibilidade e dedicação da comunidade *Open Source* por detrás do software para fornecer soluções inovadoras e de alto desempenho de maneira consistente. O PostgreSQL tornou-se uma base de dados relacional *Open Source* preferida de por muitas organizações.

O PostgreSQL é altamente extensível. Por exemplo, pode-se definir os próprios tipos de dados e construir funções personalizadas. Muitos dos recursos exigidos pelo padrão SQL são suportados, embora às vezes com sintaxe ou função ligeiramente diferentes. A partir da versão 12 lançada em outubro de 2019, o PostgreSQL está em conformidade com pelo menos 160 dos 179 recursos obrigatórios para conformidade com o SQL 2016 Core [50].

3.3 Cenário de implementação

O cenário proposto para a realização deste trabalho é ilustrado na Figura 12. O mesmo é constituído por uma aplicação central que comunica com vários agentes de forma segura. A aplicação central é composta por uma base de dados Postgres que guarda as informações obtidas pelos agentes, nomeadamente sobre ativos de licenças de software e conta com o Power BI para disponibilização de relatórios. Os agentes de software foram desenvolvidos para ser distribuídos pelos vários dispositivos de uma dada empresa de forma a que se possa recolher o máximo de informação possível. Os agentes foram desenvolvidos de forma a permitir a sua evolução, isto é, que possam obter instruções da aplicação central para adotar novas/mais funções, como por exemplo, pesquisas por outro tipo de ativos como certificados digitais, código/scripts, chaves criptográficas, ficheiros entre outros.

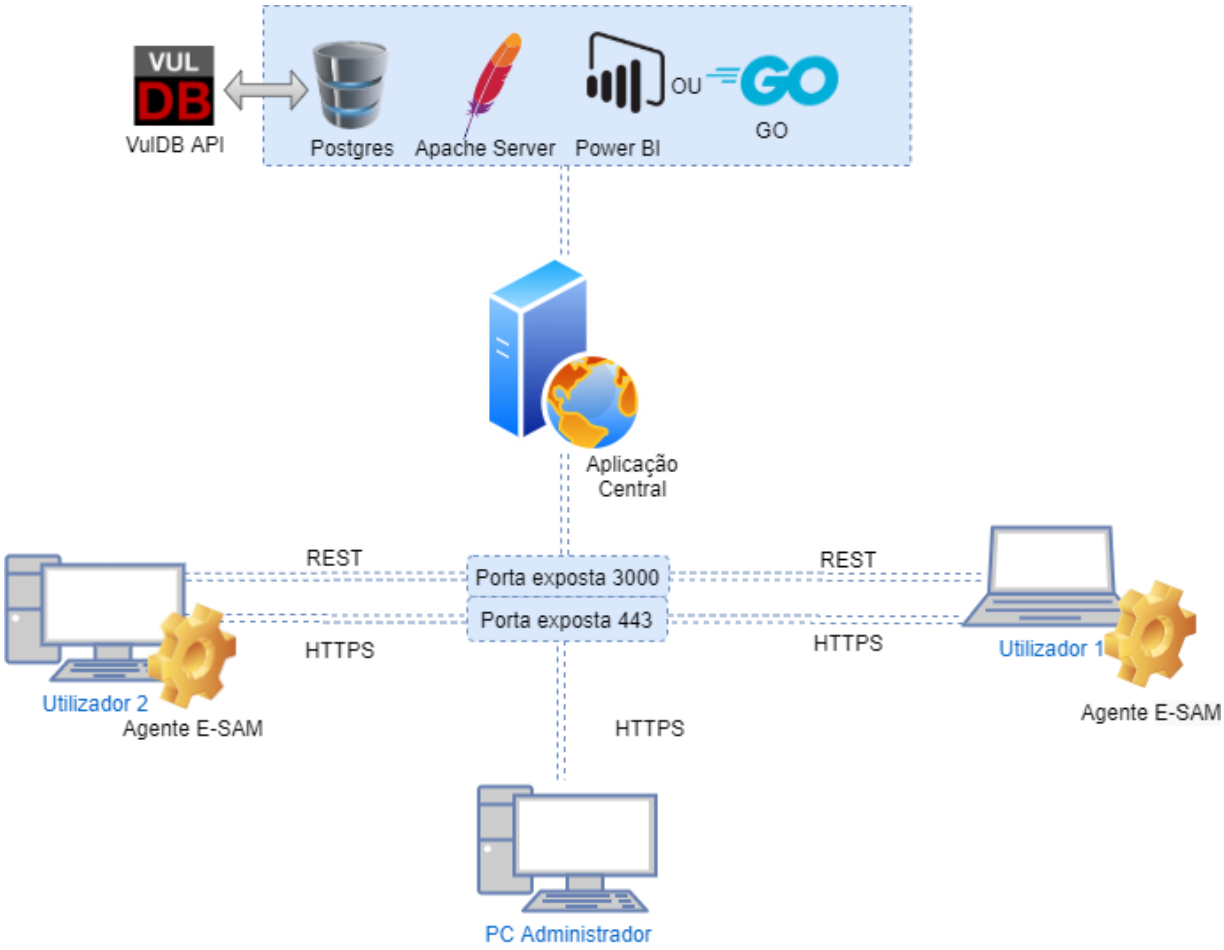


Figura 12 - Arquitetura proposta para gestão de ativos

Na versão aqui apresentada a informação recolhida pelos agentes é relativa às licenças de software associada às várias aplicações instaladas nos ativos físicos de uma organização. Estes ativos físicos são computadores que podem executar diversos sistemas operativos. A informação recolhida pelos agentes é enviada para a aplicação central e são registadas na base de dados elementos como

data/hora em que foi feito o envio, versão instalada e a própria licença cujo registo servirá também de mecanismo de backup dessa mesma licença.

Com a informação recolhida é possível desencadear várias notificações sobre licenças expiradas ou desnecessárias.

Para obter um armazenamento consistente e um acesso aos dados eficiente em bases de dados relacionais, é necessário saber organizá-los. A técnica utilizada para alcançar este fim é designada por normalização. A normalização é um processo que permite examinar os atributos de uma entidade de forma a prevenir eventuais anomalias que possam surgir, principalmente nas operações CRUD (*Create, Read, Update, Delete*) e tem como principais objetivos, estruturar relacionalmente uma base de dados de forma a suportar dados de um determinado universo, reduzir a redundância dos dados, garantir a consistência dos dados, reduzir o espaço de armazenamento e proporcionar facilidade de manipulação e manutenção dos dados [51] [52]. Quando se aplicam as regras de normalização de dados, por vezes certos atributos dão origem a novas tabelas, o que acaba por gerar no final um número maior de tabelas do que as que existiam originalmente. Mas, por outro lado, esse aumento do número de tabelas pode ser compensado pelos benefícios que advém da normalização, como por exemplo, maior rapidez no acesso a dados ou a redução da quantidade de dados armazenados. Na figura seguinte apresentamos o modelo relacional da ferramenta E-Sam desenvolvida neste projeto.

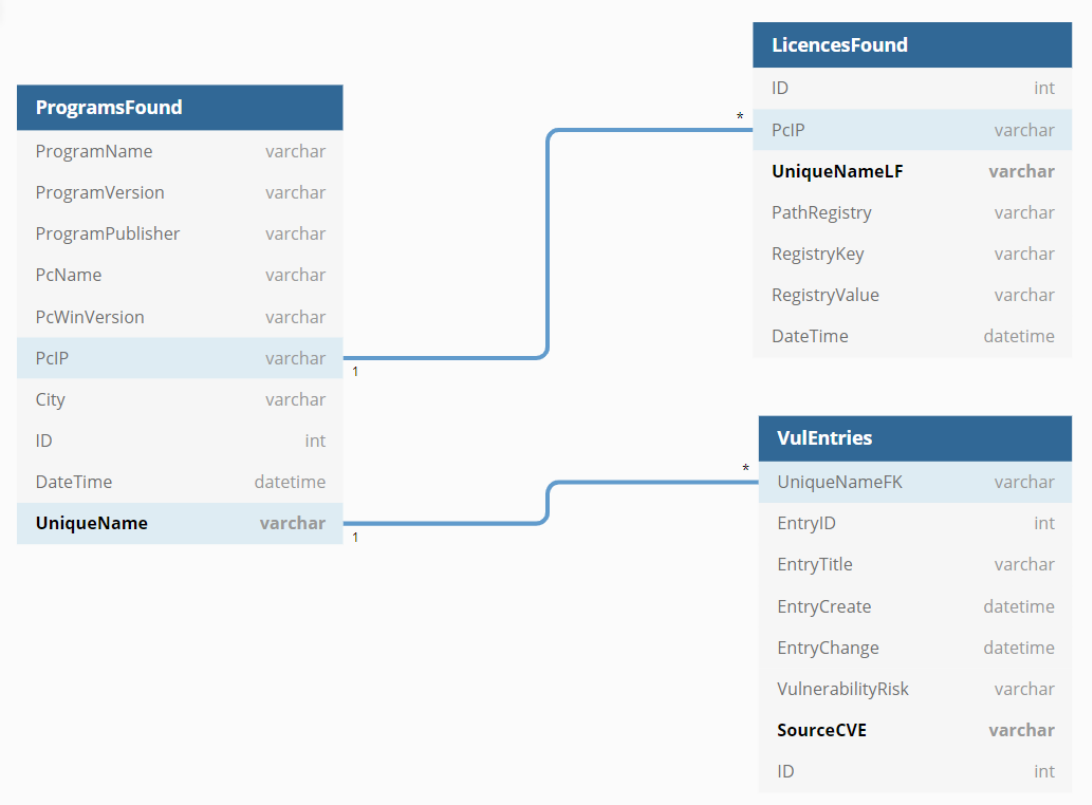


Figura 13 - Modelo relacional E-Sam

Para que seja possível obter informações sobre software que necessita de updates ou é malicioso e criar um sistema de alertas para vulnerabilidades, foi necessário correlacionar os dados recolhidos com duas bases de dados *Open Source Vulnerability Database* (OSVDB). Estas base de dados são independentes e *Open Source,* criadas pela e para a comunidade. O objetivo destes projetos é fornecer informações técnicas precisas, detalhadas, atuais e imparciais sobre uma série de vulnerabilidades e problemas associados a servidores, aplicações, entre outros [53].

As duas bases de dados de vulnerabilidades são a VulDB e a *National Vulnerability Database* (NVD) *Common Vulnerabilities and Exposures* (CVE)/*Common Platform Enumeration* (CPE) API. A primeira é a base de dados de vulnerabilidade número um em todo o mundo contando com mais de 159 mil entradas disponíveis. É uma comunidade que documenta as vulnerabilidades mais recentes desde 1970. Além dos detalhes técnicos, são fornecidas informações adicionais de inteligência de ameaças, como níveis de risco atuais e previsões de preços de exploração das vulnerabilidades [54]. No que diz respeito à base de dados NVD, esta é o repositório governamental dos EUA de dados de gestão de vulnerabilidades. É baseada em padrões representados com recurso ao protocolo *Security Content Automation Protocol* (SCAP). Os dados permitem a automação da gestão de vulnerabilidade, medição de segurança e conformidade. O NVD inclui base de dados de falhas de software relacionadas à segurança, configurações incorretas, nomes de produtos e métricas de impacto [55].

A integração entre o E-Sam e estas base de dados foi feita através das API's disponibilizadas nos Web sites oficiais. É muito importante dispor da API, pois trata-se de bases de dados que estão em constante atualização e assim é possível às organizações manterem-se sempre atualizadas sobre riscos inerentes à utilização de determinada solução de software. Na Figura 14 é ilustrado um exemplo de um retorno da API via site oficial.

Vuln ID	Summary	CVSS Severity
CVE-2020-6576	Use after free in offscreen canvas in Google Chrome prior to 85.0.4183.102 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. **Published:** September 21, 2020; 4:15:15 PM -0400	*V3.x:*(not available) *V2.0:*(not available)
CVE-2020-6575	Race in Mojo in Google Chrome prior to 85.0.4183.102 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. **Published:** September 21, 2020; 4:15:15 PM -0400	*V3.x:*(not available) *V2.0:*(not available)
CVE-2020-6574	Insufficient policy enforcement in installer in Google Chrome on OS X prior to 85.0.4183.102 allowed a local attacker to potentially achieve privilege escalation via a crafted binary. **Published:** September 21, 2020; 4:15:15 PM -0400	*V3.x:*(not available) *V2.0:*(not available)
CVE-2020-6573	Use after free in video in Google Chrome on Android prior to 85.0.4183.102 allowed a remote attacker who had compromised the renderer process to potentially perform a sandbox escape via a crafted HTML page. **Published:** September 21, 2020; 4:15:15 PM -0400	*V3.x:*(not available) *V2.0:*(not available)
CVE-2020-6571	Insufficient data validation in Omnibox in Google Chrome prior to 85.0.4183.83 allowed a remote attacker to perform domain spoofing via IDN homographs via a crafted domain name. **Published:** September 21, 2020; 4:15:15 PM -0400	*V3.x:*(not available) *V2.0:*(not available)

Figura 14 - Exemplo NVD API

Em suma, para recolher os ativos de uma empresa será disponibilizado um ficheiro executável com um certificado para autenticação. Após o utilizador executar o ficheiro disponibilizado, irá ser feito o *download* do agente via *Hypertext Transfer Protocol Secure* (https) e vai ser agendada uma nova procura através do Windows *Task Scheduler* ou com recurso ao Crontab em Debian. De seguida, o agente é executado e recolhe as informações sobre os programas e licenças de software existentes no computador. Após finalizada a procura a informação é enviada para o servidor central com *Certificate-Based Authentication*, que é a utilização de um certificado digital para identificar um utilizador, máquina ou dispositivo antes de conceder acesso a um recurso, rede, aplicação. O envio da informação para a aplicação central de forma segura foi ainda complementado com a utilização da ferramenta PostgREST [56]. Para que a ferramenta E-Sam consiga comunicar com o servidor foi gerado um certificado digital. À entidade que utiliza este certificado digital foi dada permissão de acesso para a aplicação externa. No final do envio, todos os ficheiros gerados localmente são apagados. Por fim, e de forma quase imediata, os relatórios criados no Power BI são atualizados com os novos dados recolhidos.

3.4 Desenvolvimento e configuração da solução

Neste subcapítulo apresenta-se o desenvolvimento e configuração da ferramenta E-Sam, mais propriamente, detalham-se as suas funcionalidades e os métodos desenvolvidos.

3.4.1 Funcionalidades

As funcionalidades implementadas são as seguintes:

- Agentes que identificam e recolhem ativos em múltiplos dispositivos;
- Servidor central para receber as informações recolhidas pelos agentes;
- Agendamento de novas execuções dos agentes nos sistemas operativos;
- Sistema de analise de vulnerabilidades com recurso à integração da aplicação central com base de dados de vulnerabilidades.

3.4.2 Métodos desenvolvidos

Considerando que alguns dos métodos criados são semelhantes, e com o objetivo de não tornar o relatório repetitivo, são apresentados apenas alguns dos métodos da ferramenta E-Sam.

Como referido anteriormente para facilitar a instalação dos agentes foi criado um executável Python. Este executável foi desenvolvido com instruções específicas para que obtenha a última versão dos agentes da aplicação central bem como para os executar localmente em cada máquina da organização. Faz também com que seja eliminada toda a informação gerada pelos agentes e é também calendarizado no gestor de tarefas do Windows para que execute com uma determinada periocidade.

Para que seja possível a sua execução é necessário incluir os certificados de cliente gerados, tal como apresentado no capítulo anterior.

Na Figura 15 apresenta-se um excerto de código correspondente ao executável a ser distribuído pelos utilizadores. Este código contempla uma ligação ao servidor Https com recurso a um certificado digital para autenticação, permitindo dessa forma o *download* dos agentes.

```
if oper_system == 'Windows':
    # Load certificates
    context = ssl.SSLContext(ssl.PROTOCOL_SSLv23)
    context.load_verify_locations("./SAM/Auth/ca.crt")
    context.load_cert_chain("./SAM/Auth/client.pem", keyfile="./SAM/Auth/client_key.key")
    remote_agent_zip = "./SAM/RemoteAgent.zip"
    # Connect to the server
    try:
        conn = http.client.HTTPSConnection("e-sam.com", context=context)
        conn.set_debuglevel(3)
        conn.request("GET", "/RemoteAgent/RemoteAgent.zip")
        r2 = conn.getresponse()
    except conn as error:
        logger.error(error)

    # Transfer remote agent from server
    try:
        with open('C:/SAM/RemoteAgent.zip', 'wb') as f:
            f.write(r2.read())
            # Close connection
            conn.close()
    except f as error:
        logger.error(error)

    # Unzip file to local computer
    try:
        if os.path.exists(remote_agent_zip):
            with zipfile.ZipFile(remote_agent_zip, 'r') as zip_ref:
                zip_ref.extractall('./SAM/RemoteAgentTemp')
    except os as error:
        logger.error(error)
    try:
        # Delete file .zip transfered from webserver
        if os.path.exists(remote_agent_zip):
            os.remove(remote_agent_zip)
    except os as error:
        logger.error(error)

    try:
        # Make connection with DB
        connection = psycopg2.connect(user="postgres", password="password", host="e-sam.com", port="5432",
                                      database="e-sam", sslmode='require', sslrootcert='./Auth/client.crt')
```

Figura 15 - Código executável a ser distribuído pelos funcionários da empresa

De seguida são apresentados alguns dos métodos implementados.

Método find_licences_linux e find_licences_win

Para pesquisa de licenças de software instaladas em dispositivos que executem um sistema operativo baseado em Debian foi reutilizado o projeto do GitHub dpkg-licenses. Este projeto disponibiliza uma ferramenta desenvolvida em *shell script* e através da mesma consegue-se obter a versão, descrição e licença associada aos pacotes de software instalados no sistema.

Na Figura 16 apresenta-se um excerto do código que pesquisa por estas licenças em ambientes unix. O método `find_licences_linux` chama a ferramenta dpkg-licenses que por sua vez irá guardar os resultados num ficheiro. Após terminar a ferramenta E-Sam percorre o ficheiro gerado pela ferramenta dpkg.

```
#!/bin/bash

set -e

CSV=0

case "$1" in
  --help|-h)
    cat >&2 <<.e
Lists all installed packages (dpkg -l similar format) and prints their licenses

Usage: $0
.e
    exit 1
    ;;
  -c)
    CSV=1
    ;;
esac

SCRIPTLIB=$(dirname $(readlink -f "$0"))/lib/
test -d "$SCRIPTLIB"


COLUMNS=2000 dpkg -l | grep '^.[iufhwt]' | while read pState package pVer pArch pDesc; do
  license=
  for method in "$SCRIPTLIB"/reader*; do
    [ -f "$method" ] || continue
    license=$("$method" "$package")
    [ $? -eq 0 ] || exit 1
    [ -n "$license" ] || continue
    # remove line breaks and spaces
    license=$(echo "$license" | tr '\n' ' ' | sed -r -e 's/ +/ /g' -e 's/^ +//' -e 's/ +$//')
    [ -z "$license" ] || break
  done
  [ -n "$license" ] || license='unknown'

  if [[ $CSV -eq 1 ]]
  then
    echo -n "\"$package\" <>\"$pVer\" <>\"$pArch\" <>\"$pDesc\" <>\"$license\"<!>"
  else
    echo -n "${package:0:30} <>" "${pVer:0:30} <>" "${pArch:0:6} <>" "${pDesc:0:60} <>" "$license<!>"
  fi
done
```

Figura 16 - Script para obter licenças Linux

Para obtenção destas licenças em ambientes Windows existe a necessidade de percorrer o Windows *Registry*. Desta forma, numa fase inicial deste projeto adotou-se uma abordagem diferente e que tirava partido de uma script PowerShell. O script procurava por entradas com valores definidos como por exemplo “ProducID” ou “Serial” e caminhos específicos definidos no Windows *Registry*. Podemos ver algum do código desenvolvido em PowerShell no Apêndice 2. Considerando que a abordagem era limitativa porque nem sempre os ativos usam as mesmas palavras chave no Windows *Registry*, optou-se por implementar um programa em Python que tira partido do módulo winreg. Este módulo expõe uma interface de muito baixo nível para o Windows *Registry* e espera-se que no futuro um novo módulo winreg seja criado oferecendo uma interface de nível superior para a API [57].

Método find_programs

O método `find_programs` é responsável pela procura de todos os programas existentes no computador local. Na Figura 17 é apresentado o módulo executado em ambientes Windows. O método `find_programs` procura no Windows *Registry* todos os programas instalados. Dessa forma obtém-se informações sobre os programas encontrados tais como o nome, versão e quem disponibiliza o software. De forma a ser possível encontrar todos os programas instalados em ambientes Windows foi utilizado o módulo python winreg. Nos ambientes Unix foi criado um script a partir do projeto GitHub dpkg-licenses.

```
161  def find_programs(hive, flag):
162      a_reg = winreg.ConnectRegistry(None, hive)
163      a_key = winreg.OpenKey(a_reg, r"SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall",
164                             0, winreg.KEY_READ | flag)
165
166      count_sub_key = winreg.QueryInfoKey(a_key)[0]
167
168      software_list = []
169
170      for i in range(count_sub_key):
171          software = {}
172          try:
173              asubkey_name = winreg.EnumKey(a_key, i)
174              asubkey = winreg.OpenKey(a_key, asubkey_name)
175              software['name'] = winreg.QueryValueEx(asubkey, "DisplayName")[0]
176
177              try:
178                  software['version'] = winreg.QueryValueEx(asubkey, "DisplayVersion")[0]
179              except EnvironmentError:
180                  software['version'] = 'undefined'
181              try:
182                  software['publisher'] = winreg.QueryValueEx(asubkey, "Publisher")[0]
183              except EnvironmentError:
184                  software['publisher'] = 'undefined'
185              software_list.append(software)
186          except EnvironmentError:
187              continue
188
189      return software_list
```

Figura 17 - Método find_programs

Método get_system_info

O método `get_system_info` (Figura 18), foi desenvolvido com o intuito de poder guardar informações sobre a máquina local associadas às licenças e programas encontrados. A partir deste método pode-se guardar informações na base de dados como nome do computador, IP, processador, arquitetura, entre outros. Estas informações permitem criar relatórios mais detalhados e, se necessário, identificar univocamente máquinas com vulnerabilidades.

Este módulo tira partido de um outro denominado por platform. Como o título indica, o módulo platform padrão fornece informações ao nível da plataforma, dados de software e de hardware, como por exemplo a arquitetura, o processador, dados do sistema operativo, entre outros mais específicos. A maioria de suas informações são multiplataforma. No entanto, também disponibiliza informações específicas de uma determinada plataforma, como Windows, Mac OS X, Unix e Java [58].

```
96  # Method to get system info
97  def get_system_info(logger):
98      info_list = []
99      info = {}
100     try:
101         info['platform'] = platform.system()
102         info['platform-release'] = platform.release()
103         info['platform-version'] = platform.version()
104         info['architecture'] = platform.machine()
105         info['hostname'] = socket.gethostname()
106         info['ip-address'] = socket.gethostbyname(info['hostname'])
107         info['mac-address'] = ':'.join(re.findall('..', '%012x' % uuid.getnode()))
108         info['processor'] = platform.processor()
109         info_list.append(info)
110         return info_list
111     except Exception as error:
112         message = ("Error getting system info", error)
113         logger.error(message)
```

Figura 18 - Método get_system_info

Método get_location

O método `get_location`, foi desenvolvido com o intuito de guardar informações sobre a localização da máquina local associada às licenças e programas encontrados. O código do método é apresentado na Figura 19 e a informação que disponibiliza relativa à geolocalização das máquinas permite que estes possam ser posteriormente usados para análises. Paralelamente permite também que passe a ser possível a efetuar a gestão dos ativos tendo em consideração a sua localização geográfica.

Para a implementação deste método tirou-se partido do módulo python urllib. O módulo urllib permite aceder a web sites, fazer download de dados, analisar dados e fazer qualquer pedido GET e POST. Este módulo juntamente com a API IPinfo.io, permite procurar por um IP e obter informação como: dados de geolocalização (cidade, região, país, código postal, latitude e longitude), detalhes do autonomous system number (ASN) (operadora de rede, nome de domínio associado e tipo, como negócio, hospedagem ou empresa) e dados da empresa (nome e domínio da empresa que usa o endereço IP). O código do método `get_location` é apresentado na Figura 19.

```
73  def get_location():
74      with urllib.request.urlopen('http://ipinfo.io/json') as url:
75          data = json.load(url)
76          IP = data['ip']
77          org = data['org']
78          city = data['city']
79          country = data['country']
80          region = data['region']
81          return city
82
```

Figura 19 - Método get_location

Método insert_db_vul_vuldb

O método `insert_db_vul_vuldb` é um dos pilares da ferramenta E-Sam. Este método, como se pode observar na Figura 20, comunica com a API da base de dados de vulnerabilidades VulDB, sendo passado como argumento o nome do programa e versão. Depois de fazer um pedido a esta base de dados é retornada uma resposta em *JavaScript Object Notation* (JSON) na qual, em caso de existirem vulnerabilidades, se podem retirar informações como por exemplo o nível de risco, o CVE ou a descrição de vulnerabilidade. Um exemplo da resposta enviada pela API VulDB é apresentado no Anexo 1.

```python
def insert_db_vul_vuldb(program_name, version, unique_name, logger):
    try:
        # Make connection with DB
        connection = psycopg2.connect(user="postgres", password="password", host="e-sam.com", port="5432",
                                      database="e-sam", sslmode='require', sslrootcert='./client.crt')
        cursor = connection.cursor()
    except (Exception, psycopg2.Error) as error:
        message = ("Failed to connect the DB", error)
        logger.error(message)
        sys.exit()

    # url endpoint
    url = 'https://vuldb.com/?api'

    # request
    post_fields = {'apikey': 'bb1547f9969f854cae72c260d019534d', 'search': program_name + version}

    request = Request(url, urlencode(post_fields).encode())
    json_str = urlopen(request).read().decode()

    output = json.loads(json_str)
    print(output)

    try:
        for i in output['result']:
            create_date = int(i["entry"]["timestamp"]["create"])
            change_date = int(i["entry"]["timestamp"]["change"])
            entry_create = time.strftime("%Y-%m-%dT%H:%M:%SZ", time.localtime(int(create_date)))
            entry_change = time.strftime("%Y-%m-%dT%H:%M:%SZ", time.localtime(int(change_date)))
            entry_id = (i["entry"]["id"])
            entry_title = (i["entry"]["title"])
            vulnerability_risk = (i["vulnerability"]["risk"]["name"])
            source_cve = (i["source"]["cve"]["id"])
            postgres_insert_query = """
            INSERT INTO public."vulEntries"("entryId", "entryTitle",
            "entryCreate", "entryChange", "vulnerabilityRisk", "sourceCVE", "uniqueNamePFound")
            VALUES (%s,%s,%s,%s,%s,%s,%s)"""
            record_to_insert = (entry_id, entry_title, entry_create, entry_change,
                                vulnerability_risk, source_cve, unique_name)
```

Figura 20 - Parte do Método para procurar vulnerabilidades e inserir na base de dados

Métodos schedule_unix e schedule_win

De modo a que só seja necessária a intervenção do utilizador uma única vez durante a execução da ferramenta E-Sam foram desenvolvidos dois métodos para que a pesquisa fosse agendada e executada automaticamente numa data/hora especifica. Para efeitos de testes foi definido que a ferramenta E-Sam fosse executada todos os dias.

Para ser possível agendar tarefas em ambientes Unix foi utilizado o módulo em Python chamado Crontab. O nome é derivado da palavra grega "Chronos", que significa "tempo". As funções disponíveis permitem-nos aceder ao Cron, programar tarefas, definir restrições, entre outros [59].

Por outro lado, para que fosse possível agendar tarefas em ambientes Windows foi utilizado o módulo win32com que basicamente permite interagir com objetos Component Object Model (COM) e automatizar aplicações do Windows com Python. Os objetos COM permitem controlar as aplicações do Windows a partir de outro programa. Os objetos COM são definidos no Windows *Registry* [60]. Um

excerto do método `schedule_win` utilizado para agendar a execução da ferramenta E-Sam, neste exemplo todos os dias, pode ser visualizado na Figura 21. Após a execução deste método é agendada no Windows *Scheduler* esta tarefa, como se pode observar no Apêndice 3.

```python
import datetime
import win32com.client

start_time = datetime.datetime.now() + datetime.timedelta(minutes=1)

computer_name = ""  # leave all blank for current computer, current user
computer_username = ""
computer_userdomain = ""
computer_password = ""
action_id = "SAM"  # arbitrary action ID
action_path = r"C:\SAM\Python37\python.exe"  # executable path (could be python.exe)
action_arguments = r"C:\SAM\SoftwareAssetManagement.py" # arguments (could be something.py)
action_workdir = r"C:\SAM"  # working directory for action executable
author = "Paulo Teixeira"  # so that end users know who you are
description = "SAM v1.5"  # so that end users can identify the task
task_id = "SAM"
task_hidden = True  # set this to True to hide the task in the interface
username = ""
password = ""
run_flags = "TASK_RUN_NO_FLAGS"  # see dict below, use in combo with username/password
# define constants
TASK_TRIGGER_DAILY = 2
TASK_CREATE = 2
TASK_CREATE_OR_UPDATE = 6
TASK_ACTION_EXEC = 0
IID_ITask = "{148BD524-A2AB-11CE-B11F-00AA00530503}"
RUNFLAGSENUM = {
    "TASK_RUN_NO_FLAGS": 0,
    "TASK_RUN_AS_SELF": 1,
    "TASK_RUN_IGNORE_CONSTRAINTS": 2,
    "TASK_RUN_USE_SESSION_ID": 4,
    "TASK_RUN_USER_SID": 8
}
# connect to the scheduler (Vista/Server 2008 and above only)
scheduler = win32com.client.Dispatch("Schedule.Service")
scheduler.Connect(computer_name or None, computer_username or None, computer_userdomain or None,
                  computer_password or None)
rootFolder = scheduler.GetFolder("\\")

#(re)define the task
taskDef = scheduler.NewTask(0)
```

Figura 21 - Método para agendar uma nova procura em ambientes Windows

3.4.3 Configuração da aplicação central

Nesta subsecção é detalhada a configuração da aplicação central da ferramenta E-Sam. Para realizarmos esta configuração foi instalada e configurada uma máquina virtual Windows server 2019.

Nesta máquina configuramos um servidor Apache para que seja possível tornar acessível o *download* externo dos agentes bem como criar um repositório central onde está sempre disponível a sua versão mais recente. Para isso foi criada uma página Web a partir da qual se pode efetuar o seu *download* para que posteriormente sejam instalados nos computadores clientes. O servidor Apache foi

configurado com autenticação *Transport Layer Security* (TLS), para isso foi necessário criar dois certificados digitais, um para configuração da Apache e outro para autenticação do utilizador com recurso ao OpenSSL.

Um certificado digital é uma credencial digital que fornece informações sobre a identidade de uma entidade, ou seja, é uma forma digital de identificação bem como repositório de algumas informações adicionais. Um certificado digital é emitido por uma autoridade, conhecida como Autoridade Certificadora (AC). Por ser emitido por uma autoridade certificadora, é da responsabilidade dessa autoridade garantir a validade das informações presentes no certificado digital. Essa verificação pode ser realizada pela própria AC ou por uma Autoridade de Registo (AR) [61].

Este documento eletrónico (certificado digital) prova a titularidade de uma chave pública pela entidade associada. Além disso, um certificado digital é válido apenas por um determinado período de tempo definido nas políticas do certificado e que normalmente está relacionado com o tamanho da chave pública. Por exemplo, a entidade de certificação do estado Português tem um certificado digital com validade de 10 anos (com uma chave de 4096 bits), enquanto os cidadãos têm certificados digitais com validade de 5 anos (com chaves de 1024/2048 bits) [61].

Para a criação de certificados digitais é necessária a execução de vários comandos. De seguida, são apresentados e explicados alguns desses comandos que, devem ser utilizados para que seja possível obter um certificado digital. Em primeiro lugar é necessária a criação de uma AC que permita simular um certificado digital de uma cadeia hierárquica de pelo menos dois níveis. Para fins de teste, esta AC substitui uma AC reconhecida na Internet (como por exemplo a VeriSign). Todos os certificados digitais irão ser assinados digitalmente por esta AC de raiz.

De seguida descrevem-se os passos necessários para a criação de uma AC.

1. Primeiro, procede-se à criação de um ficheiro de solicitação de certificado em inglês *Certificate Signing Request* (CSR), isto é, um bloco de texto codificado que é posteriormente enviado para uma autoridade de certificação como forma de solicitar um certificado. O "assunto" (-subj) descreve o assunto/titular do certificado. Para isso, é necessário executar o comando: "`openssl req -passout pass:projeto -subj "/C=PT/ST=Porto/L=Felgueiras/O=ESTG/OU=ESTG Students/CN=ServerProjeto /emailAddress=810000@estg.ipp.pt" -new > testecsr.csr`".
2. De seguida, é criado um ficheiro para armazenar a chave privada. Assim, não é necessário colocar *password* sempre que é preciso assinar um certificado. Para tal, executa-se o comando: "`openssl rsa -passin pass:projeto -in privkey.pem -out testekey.key`".
3. Nesta etapa é criado um certificado digital X.509 a partir do CSR. O comando que se segue cria um certificado assinado com a chave privada da AC com validade de 365 dias: "`openssl x509 -in testecsr.csr -out testecert.cert -req -signkey testekey.key -days 365`".
4. Para concluir, é necessário criar um ficheiro codificado em PKCS#12 que contenha o certificado e a chave privada. O comando que se segue define a password no ficheiro com extensão `p12`

como `default`: "`openssl pkcs12 -passout pass:default -export -nokeys -cacerts -in testecert.cert -out testep12.p12 -inkey testekey.key`"

Finalizados estes passos obtém-se uma AC de certificação (`testecert.cert`), que é instalada no servidor Web e um ficheiro de chave privada (`testekey.key`) que poderá ser utilizado para assinar certificados de utilizador.

Para criar um certificado digital para um determinado utilizador é necessário realizar os seguintes passos:

1. Criar um ficheiro CSR para o utilizador. Se for de interesse, pode-se fornecer um assunto apropriado: "`openssl req -passout pass:testeuser -subj "/C=PT/ST=Porto/L=Felgueiras/O=ESTG/OU=Student Software Group/CN=Paulo Teixeira/emailAddress=8150524@estg.ipp.pt " -new > paulocsr.csr`";
2. Criar um ficheiro de chave privada sem password: "`openssl rsa -passin pass:testeuser -in privkey.pem -out paulokey.key`";
3. Criar um certificado X.509 para o novo utilizador. Assina-se digitalmente usando a chave privada do utilizador e certifica-se o mesmo usando a chave privada da AC com validade de 365 dias: "`openssl x509 -req -in paulocsr.csr -out paulocert.cert -signkey paulokey.key -CA testecert.cert -CAkey testekey.key -CAcreateserial -days 365`";
4. Criar um ficheiro codificado em PKCS#12: "`openssl pkcs12 -passout pass:default -export -in paulocert.cert -out paulorp12.p12 -inkey paulokey.key`".

Para criar mais certificados, basta executar os passos acima indicados. É preciso manter os ficheiros de chaves seguros e eliminá-los quando estes deixarem de ser necessários. O ficheiro de chave privada da AC não pode ser eliminado, pois este é necessário para assinar os certificados digitais finais.

Após emitir os certificados para o servidor Web foi feita a instalação e configuração de um Apache server com recurso à ferramenta XAMPP. Esta é uma distribuição do Apache fácil de instalar que contém MariaDB, PHP e Perl. Para que o servidor Apache esteja disponível para ligações seguras (Https) e autenticação com certificado digital, como se pode constatar no apêndice 4, configurou-se o ficheiro "httpd.conf" com um entrada VirtualHost como a que se segue:

```
<VirtualHost _default_:443>
        DocumentRoot "C:/Apache/htdocs"
        ServerName www.e-sam.com
        ServerAdmin 8150524@estg.ipp.pt
        ErrorLog "C:/Apache/apache/logs/error.log"
        TransferLog "C:/Apache/apache/logs/access.log"
        SSLCertificateFile "conf/ssl.crt/server.crt"
```

```
        SSLCertificateKeyFile "conf/ssl.key/server.key"
        SSLCertificateChainFile "conf/ca/ca.crt"
        SSLCACertificatePath "conf/ca"
        SSLCACertificateFile "conf/ca/ca.crt"
        SSLVerifyClient require
</VirtualHost>
```

De seguida, e para armazenar os dados recolhidos pelos agentes na aplicação central foi instalado o SGBD Postgres. A script que contém o modelo de dados é apresentada no apêndice 5. O Postgres foi instalado e configurado de modo a usar um certificado digital para autenticação e foi também definida a gama de IPs para os quais aceita ligações. Estas configurações foram feitas de modo a garantir a segurança do envio dos dados bem como o acesso à base de dados, pois possui dados sensíveis dos utilizadores. As configurações da gama de IPs foram feitas no ficheiro "`pg_hba.conf`", tendo-se adicionado a seguinte configuração:

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# IPv4 local connections:
host    all             all             192.168.1.86/32         md5
```

Para configurar o uso de um certificado digital para autenticação foram realizados os passos acima descritos para criação de uma nova AC bem como a criação do certificado de autenticação. O ficheiro "`postgressql.conf`" foi alterado de forma a conter os caminhos para o certificado de AC, utilizador e a respetiva chave. Esta configuração é consiste no ajuste dos seguintes parâmetros:

```
# SSL

ssl = on
ssl_ca_file = 'ca.crt'
ssl_cert_file = 'ssl.crt'
ssl_key_file = 'ssl.key'
```

Para que seja inserida a informação recolhida pelos agentes na base de dados de forma segura, recorreu-se à utilização da ferramenta PostgREST. O PostgREST é um servidor web autónomo que transforma uma base de dados PostgreSQL numa API RESTful. É distribuído como um único binário, com versões compiladas para as principais distribuições de Linux / BSD / Windows [62].

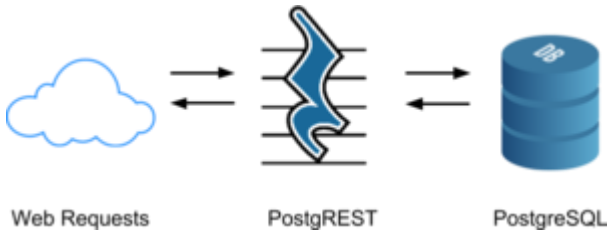


Figura 22 - PostgREST API [62]

Após realizado o download da versão mais recente do PostgREST foi necessário criar um ficheiro de configuração `settings.conf`. O PostgREST utiliza este ficheiro para saber como se conecta à base de dados. Um exemplo de configuração pode ser consultado de seguida:

```
db-uri = "postgres://user:pass@localhost:5432/e-sam"
db-schema = "api"
db-anon-role = "unauthorized"
jwt-secret = "<password>"
```

Relativamente a esta configuração podemos dizer que o parâmetro `db-uri`, trata-se do formato de conexão padrão. No que diz respeito ao parâmetro `db-schema`, aqui colocamos o nome do *schema* de base de dados que iremos expor aos clientes *Representational State Transfer* (REST). Em relação ao parâmetro `db-anon-role`, este é usado para definirmos as regras configuradas na base de dados a serem utilizadas quando nenhuma autenticação de cliente é fornecida. Por último, definimos o parâmetro `jwt-secret` para que os clientes se autentiquem com a API utilizem JSON *Web Tokens*. Estes são objetos JSON assinados criptograficamente e que usam uma password conhecida apenas pelo servidor. Como os clientes não sabem a password, eles não podem violar o conteúdo dos seus *tokens*. Esta senha deve ser constituída pelo menos por 32 caracteres. Para concluirmos esta conexão (agentes – API), necessitamos ainda de criar e assinar um token de autenticação. A criação destes tokens de autenticação pode ser feita acedendo o web site jwt.io como podemos ver na Figura 23.

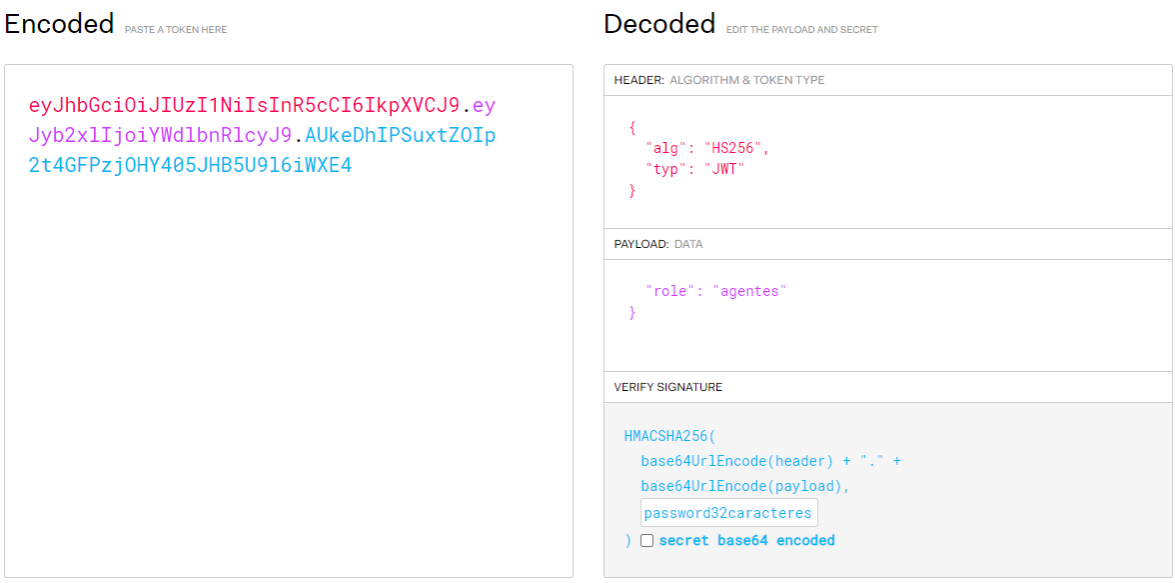


Figura 23 – Criação e assinatura de um token de autenticação

Em suma, como anteriormente referiu-se, o token de autenticação criado servirá para autenticação dos agentes com a API através de chamadas POST feitas pelos agentes. Podemos ver um exemplo desta chamada desenvolvida em python de seguida:

```
import requests
TOKEN=
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyb2xlIjoiYWdlbnRlcyJ9.lTNWw9jGp1jfetIpH4qB-T-fov58F0IJPlzwW7PBViU"

headers = {'Authorization': 'Bearer ' + TOKEN}
payload = {"Column1": "Data1", "Column2": "Data1", "Column3": "Data1"}

requests.post("http://e-sam.com:3000/vulEntries", headers=headers, data=payload)
```

Como o intuito de facultar permissões apenas para ações necessárias na base de dados, configuramos uma regra nova chamada `agentes`. Esta configuração é necessária para que os agentes consigam enviar a informação recolhida para API e pode ser consultada na Apêndice 6.

Ainda relativamente à configuração da aplicação central, foi instalado o Power BI Desktop. A partir desta ferramenta como já mencionado é possível criar relatórios. O Power BI foi configurado de forma a obter os dados do Postgres. Tal é feito através do menu Página Inicial – Obter Dados – Base de Dados – Base de Dados PostgresSQL. Também se configurou no Power BI as relações entre tabelas, para tal, e como ilustrado na Figura 24, acedeu-se ao menu Modelação – Gestor de Relações e criar as relações.

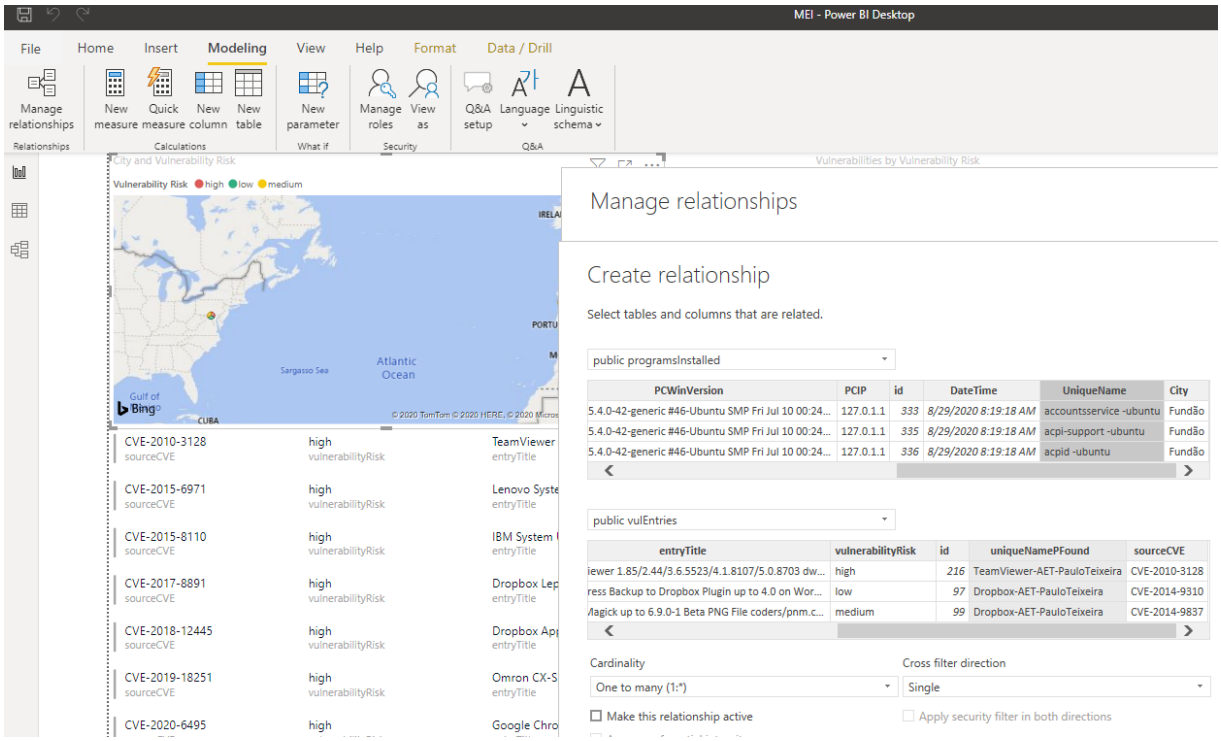


Figura 24 - Criação de relações entre tabelas Power BI

Por fim relativamente ao Power BI, criou-se três tipos de alertas com condições para que sejam enviados emails. Foi criado um alerta para cada tipo de vulnerabilidades (*High*, *Medium* e *Low*). No caso dos agentes reportarem uma vulnerabilidade do tipo *High* é envidado um email de hora em hora, sendo que se reportarem 25 vulnerabilidades do tipo *Medium* será enviado outro email também de hora em hora e por fim se forem reportados 50 do tipo *Low* será também enviado mas este só uma vez por dia. Estas regras podem facilmente ser ajustadas.

3.5 Resumo do capítulo

Neste capítulo foi apresentado a solução E-Sam. Esta solução é composta por agentes que executam localmente nos dispositivos de uma organização e têm a capacidade de recolher informação sobre ativos nesses mesmos dispositivos. Os dados recolhidos são enviados para um sistema central, a partir do qual se podem gerar alertar e criar visualizações sobre os dados usando uma ferramenta de Business Intelligence como facilitador do processo, toda a comunicação é assegurada por certificados digitais promovendo a segurança da informação que circula no canal de comunicação e o controlo de acessos.

Capítulo 4 – Resultados obtidos em prova de conceito

De forma a projetar e desenvolver a solução E-Sam, foram considerados aspetos relevantes descritos no estado de arte, mais concretamente na Tabela 2, que apresenta as principais vantagens das soluções existentes. Foram ainda tidos em consideração os seguintes requisitos para o E-Sam:

- Deverá ser possível fazer a gestão clara dos ativos das empresas, sendo o focus nesta fase no software e licenças associadas;
- Deverá ser uma solução flexível, para que possa ser adaptada a diferentes tipos de negócio e indústrias;
- Deverá ser uma aplicação Web-based, que será centralizada, ou seja, será instalada num servidor e poderá ser acessível através de qualquer computador que esteja ligado na rede;
- Deverá ser desenvolvido em tecnologias Open Source, para poder ser comercializado através de licenças de baixo custo;
- Deverá ser *user-friendly*, sem, no entanto, menosprezar o detalhe;
- A pesquisa dos Ativos deverá ser feita automaticamente e ao mesmo tempo ser capaz de fazer a verificação de vulnerabilidades existentes.

Na Tabela 3 apresentamos uma comparação entre as características do E-Sam com quatro soluções mencionadas no estado de arte. Na tabela podemos constatar que a solução desenvolvida não só corresponde com características semelhantes às existentes no mercado, mas também é inovadora no ponto de verificação de vulnerabilidades. Podemos também dizer que é uma das poucas que faz pesquisa automática na rede por ativos bem como a única a utilizar uma arquitetura de agentes para realizar esta tarefa. Por fim, podemos referir que a solução E-Sam por se encontrar num estádio inicial de desenvolvimento e maturidade, não suporta ainda a diversidade de ativos suportada pelas soluções de mercado analisadas. Como esta solução foi desenhada e desenvolvida de forma a poder ser facilmente expansível, essa desvantagem será rapidamente ultrapassada no trabalho futuro a realizar.

Tabela 3 - Tabela de comparação entre soluções alternativas e o E-Sam

Solução / Características	Asset Explorer	InvGate Assets	Asset Tiger	Snipe-IT	E-Sam
Instalação Centralizada	Sim	Não	Não	Sim	Sim
Preço licenças **+++ Muito Alto** **++ Alto** **+ Razoável** **- Baixo**	++	+++	++	+	-
Flexibilidade	Não	Não	Não	Não	Sim
User-Friendly	Sim	Não	Sim	Sim	Sim
Multiplataforma	Não	Não	Sim	Sim	Sim
Relatórios customizáveis	Sim	Sim	Sim	Sim	Sim
Disponibilização de Indicadores gráficos	Sim	Sim	Sim	Sim	Sim
Baseado em tecnologia *Open Source*	Sim	Não	Não	Sim	Sim
Web Based	Sim	Sim	Sim	Sim	Sim
Verificação de Vulnerabilidades	Não	Não	Não	Não	Sim

Pesquisa automática de Ativos	Sim	Não	Não	Não	Sim

Para testar a ferramenta E-Sam em diferentes sistemas operativos procedeu-se à sua execução num computador pessoal com o sistema operativo (SO) Windows 10 e criou-se uma máquina virtual com SO Ubuntu 20.04.1 LTS. Testou-se a recolha de ativos, em concreto o software e respetivas licenças de utilização e também se testou a pesquisa de vulnerabilidades para esses mesmos ativos.

De seguida, pode-se observar alguns exemplos de relatórios criados no Power BI (Figuras 25, 26, 27 e 268. Os relatórios foram elaborados para que se possa disponibilizar informação relativa aos ativos e vulnerabilidades existentes. Para a criação destes relatórios foram seguidas algumas questões chave, como por exemplo quantas máquinas estão infetadas por IP, riscos de vulnerabilidades por nome de programa, número de ativos recolhidos por dia, mapa geográfico a representar riscos de vulnerabilidades por localização das máquinas, entre outros.

Na Figura 25 podem-se observar exemplos de relatórios criados pela ferramenta, nomeadamente agrupando as vulnerabilidades por nome, risco e empresas detentoras das respetivas aplicações. Pelo gráfico de barras verifica-se que as aplicações com mais vulnerabilidades encontradas são aplicações da Google e da Microsoft. Ao complementar com o gráfico do tipo *pie chart*, percebe-se que esses programas são o Google Chorme e o Microsoft Edge.

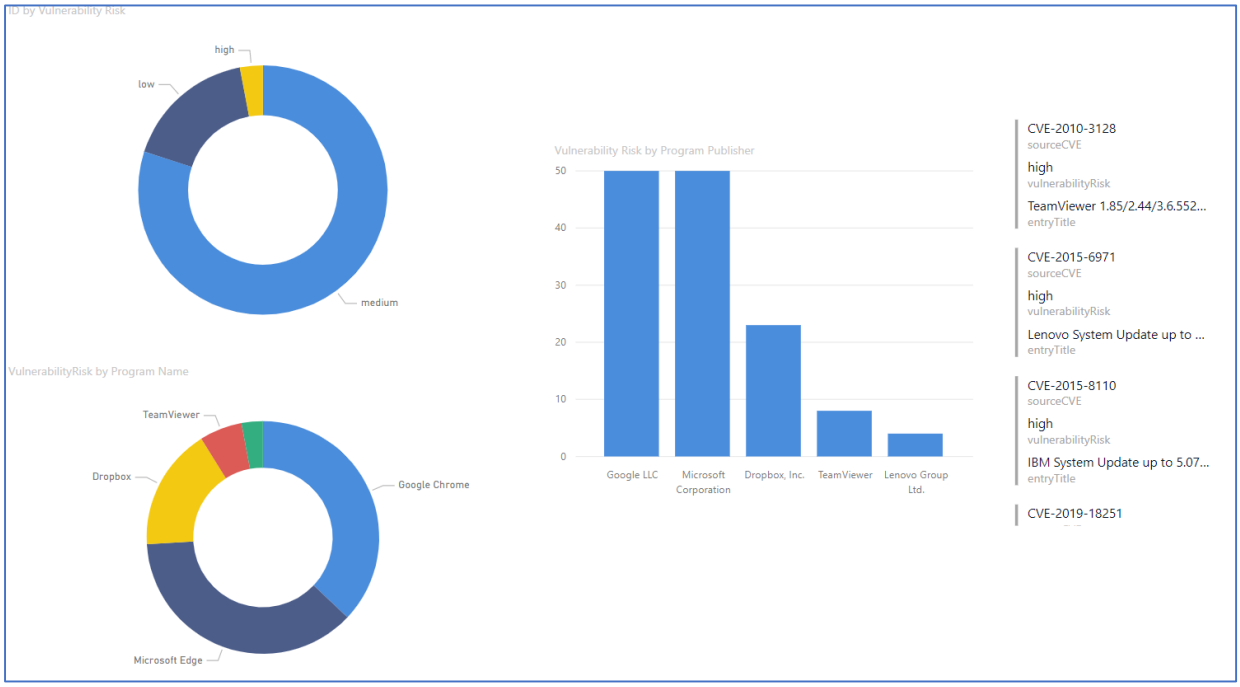


Figura 25 - Exemplo de relatório vulnerabilidades por Nome, Risco e Editor

Os relatórios são interativos, portanto podendo-se selecionar os diferentes dados disponíveis nos gráficos como se ilustra na Figura 26. Nesta figura, de modo a apresentar as vulnerabilidades referentes ao nível de severidade *high,* selecionou-se esse mesmo campo no gráfico "ID by Vulnerability Risk". Desta forma verifica-se que os programas identificados com risco "high" são o Lenovo Vantage Service e o Team Viewer. Nesta página pode-se ainda observar uma breve descrição da vulnerabilidade bem como o CVE associado, por exemplo o CVE-2010-3128. Este CVE diz que a vulnerabilidade do caminho de pesquisa não confiável no TeamViewer 5.0.8703 (e anteriores) permite que utilizadores locais, e possivelmente invasores remotos, executem código arbitrário e conduzam ataques de roubo de DLL por meio de um cavalo de tróia `dwmapi.dll` localizado na mesma pasta que o ficheiro de extensão `tvs` ou `tvc` [63].

Figura 26 - Exemplo relatório ao selecionar um tipo de vulnerabilidades (high)

Na Figura 27, estão representados os ativos de software, aplicações enviadas pelos agentes no dia 19 e 20. Ao interagir com o gráfico e pressionar o dia 19 podem-se observar que foram recolhidos cerca de 40 ativos de software neste dia. Estes ativos foram recolhidos pelo agente colocado no computador AET-PauloTeixeira e este agente recolheu informação de ativos de software como por exemplo o BlueX Browser, ConsentID, Dropbox, entre outros.

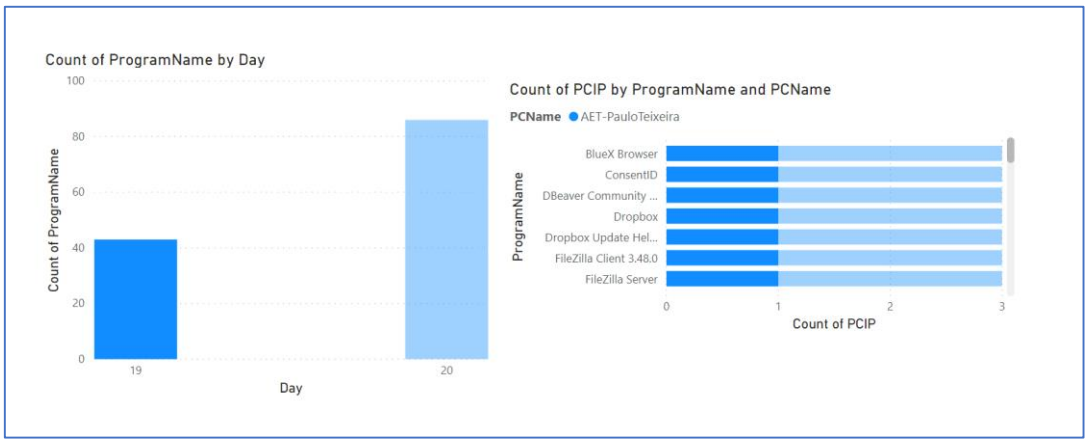


Figura 27 - Vulnerabilidades por dia e por Internet Protocol (IP)

A Figura 28 apresenta outro tipo de relatório onde é representada a localização geográfica dos ativos de software com vulnerabilidades de risco *high.* Através deste relatório pode-se visualizar quais as vulnerabilidades existentes por tipo e por local geográfico, bem como informações complementares como o IP e versão do sistema operativo do computador infetado. Neste caso, ao selecionar os dados relativos ao risco *high*, é possível observar que uma determinada empresa dispõe de equipamentos com vulnerabilidades localizadas em Portugal e Singapura. Através deste relatório é também possível obter informações sobre as máquinas e vulnerabilidades como por exemplo a máquina com o IP 192.168.154.1 e que possui o SO Windows 10 tem uma vulnerabilidade CVE-2015-6971. Este CVE informa que o programa Lenovo System Update (anteriormente ThinkVantage System Update) em versões anteriores à 5.07.0013 permite que os utilizadores locais enviem comandos para o serviço System Update (SUService.exe) e obtenham privilégios ao iniciar executáveis assinados da Lenovo [64].

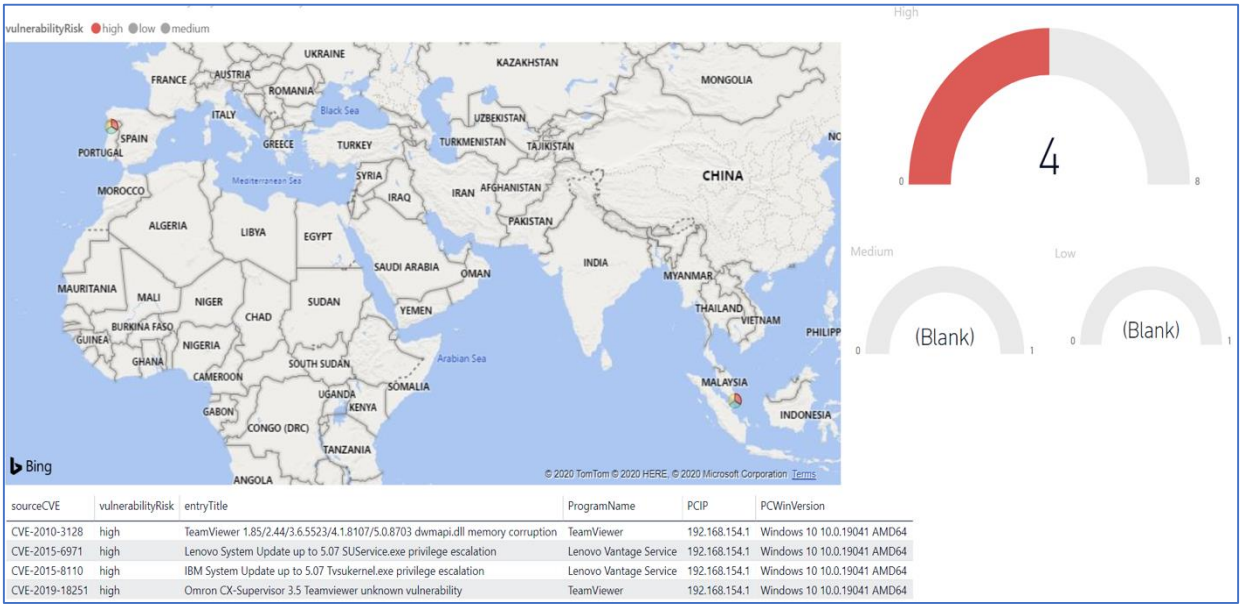


Figura 28 - Relatório vulnerabilidades High e localização

A Figura 29 apresenta algumas das licenças encontradas pelos agentes nos equipamentos com SO Windows e Linux utilizados nos testes. Devido ao equipamento Windows ser pessoal e conter licenças de programas pagos optou-se por ocultar essa informação. Ao observar a figura seguinte pode-se ver algumas das licenças que os agentes encontraram nos equipamentos com Windows 10, com o IP 192.168.154.1 e no equipamento com Linux. Algumas das licenças correspondem aos programas VMware e Microsoft Office do equipamento Windows. Relativamente à máquina Linux pode-se observar que algumas licenças correspondem à aplicação servidor Apache e algumas licenças *Berkeley Source Distribution* (BSD). As licenças BSD são um tipo de licença com baixas restrições para software *Open Source* que não impõe requisitos aquando da sua redistribuição. Por esse motivo, as licenças BSD são utilizadas para a distribuição de muitos softwares *freeware*, *shareware* e *Open Source*. A licença BSD Unix original foi escrita pela primeira vez em 1969 [65].

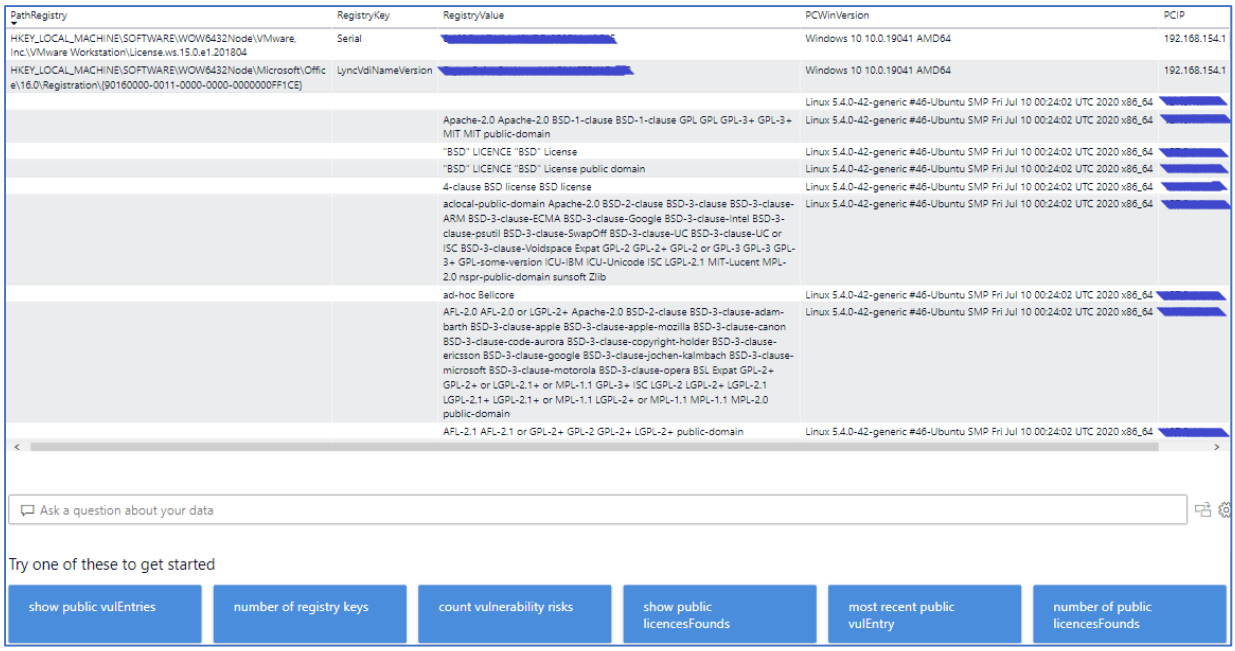


Figura 29 - Licenças encontradas Windows Linux

Na Figura 30, estão representados os alertas criados para que posteriormente os responsáveis pela gestão do serviço possam receber emails com informações de alerta. Pelos gráficos pode-se observar que uma determinada empresa possui equipamentos contendo quatro vulnerabilidades "*high*", 108 "*medium*" e 23 "*low*".

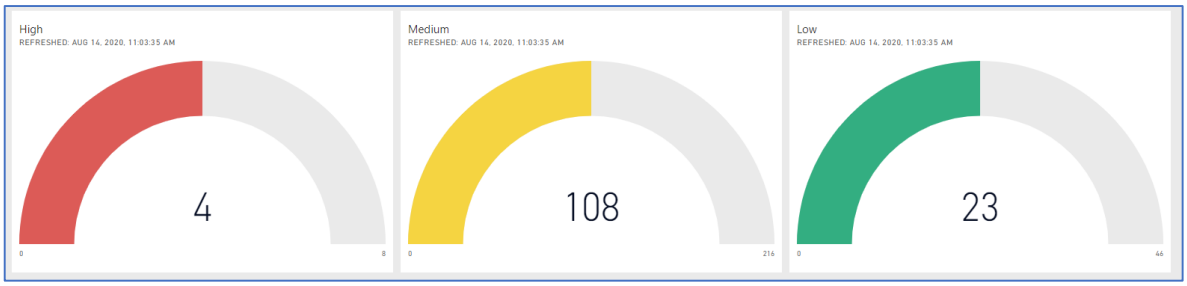


Figura 30 - Alertas criados no Power BI

Foram criadas regras para que o responsável pela gestão de ativos receba emails com informações relativas às vulnerabilidades. Estas métricas variam consoante a dimensão da empresa. No caso e para efeitos de teste, as métricas foram definidas de modo a enviar um email de hora em hora, caso as vulnerabilidades "*high*" atinjam o valor 1 e as vulnerabilidades "*medium*" atinjam o valor 25. No caso das vulnerabilidades "*low"*, a métrica definida foi enviar um email de 24 em 24 horas, sempre que o valor atingido seja igual ou superior a 50 vulnerabilidades. Neste caso, o gestor irá receber de hora em hora um email relativo às vulnerabilidades do tipo "*high*" e do tipo "*medium*". Na Figura 31 é apresentado um exemplo de um email recebido pelos gestores dos ativos.

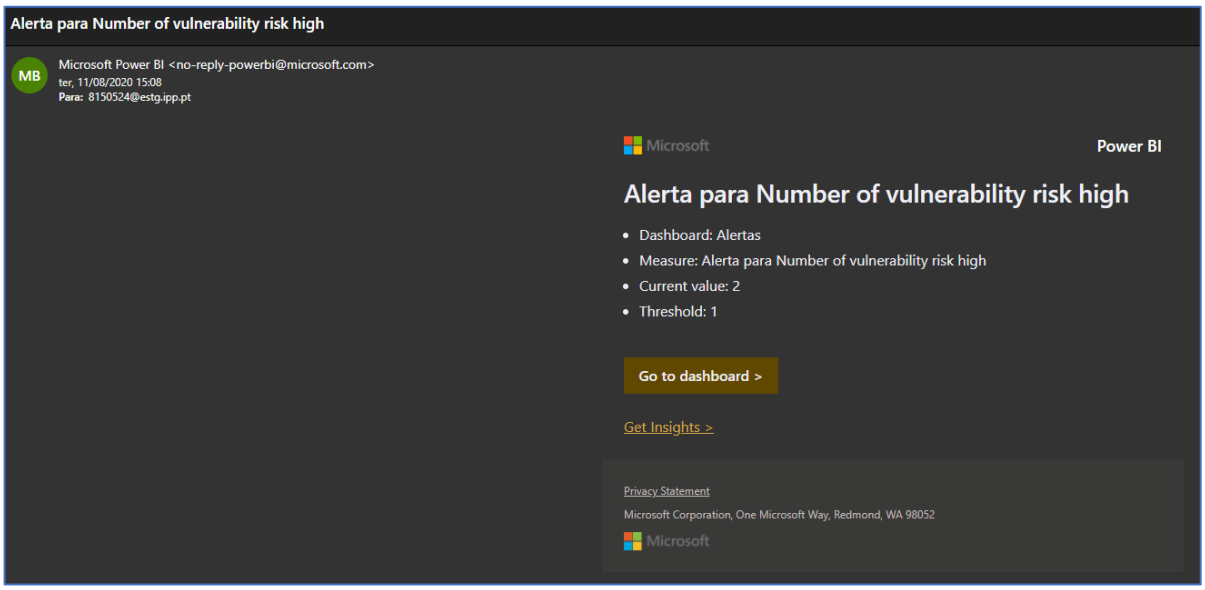


Figura 31 - Email de alerta recebido pelo Power BI

4.1 Sumário do capítulo

Neste capítulo foi apresentada uma prova de conceito do E-Sam. Foi criado um ambiente e os agentes foram postos em prática recolhendo informação dos ativos nos equipamentos que compõe o ambiente de testes. Os dados recolhidos foram enviados para a base de dados central, através de um

serviço Web exposto no porto 443. A partir da base de dados, o Power BI gerou um conjunto de dashboards permitindo observar os ativos e o seu estado relativamente ao licenciamento e existência de vulnerabilidades. A verificação de vulnerabilidades implicou a consulta a base de dados de vulnerabilidades tirando partido da API que estas disponibilizam. A prova de conceito mostra que a solução é funcional e que através da mesma a gestão de ativos fica mais simples e automatizada facilitando o trabalho dos gestores de ativos.

Capítulo 5 - Considerações Finais

Este último capítulo foca quatro aspetos. Em primeiro lugar apresenta as conclusões retiradas do projeto proposto no início deste trabalho. Posteriormente refere os contributos do trabalho, as limitações existentes e, por fim, as propostas para trabalho futuro.

O projeto apresentado tinha por objetivo a criação de um serviço de pesquisa e gestão de ativos de software, com indicação inicial específica em licenças e aplicações de software. O objetivo da solução é permitir efetuar a recolha e manutenção dessas licenças e aplicações ao mesmo tempo que se efetua a pesquisa proactiva sobre vulnerabilidades conhecidas nesses ativos.

Pelos resultados obtidos, pode-se afirmar que foi criada uma solução cujo objetivos propostos foram alcançados com sucesso. A criação da solução E-Sam, ferramenta de gestão de ativos de software e licenças, permite às empresas aplicar no seu ambiente de trabalho uma gestão eficaz de ativos bem como a identificação e potencial diminuição de vulnerabilidades relacionadas com os ativos.

Para a criação da solução foram utilizadas várias ferramentas. Para o desenvolvimento da ferramenta E-Sam utilizou-se a linguagem Python. Para disponibilização de uma *dashboard* para o utilizador final utilizou-se o Microsoft Power BI que é uma ferramenta de *Business Intelligence* da Microsoft. Para que fosse possível armazenar os dados recolhidos pela ferramenta E-Sam utilizou-se o sistema de base de dados PostgreSQL.

Comparativamente a quatro soluções mencionadas ao longo do projeto, verificou-se que a solução E-Sam apresenta, não só, características semelhantes, como também introduz melhorias que podem ser consideradas inovadoras e diferenciadoras. Por exemplo, relativamente à característica instalação centralizada, a solução apresenta essa funcionalidade tal como a solução Asset Explorer e InvGate Assets. Em termos monetários, foi possível criar uma solução de baixo custo comparativamente a outras soluções. No que diz respeito à característica flexibilidade, o E-Sam é a única que apresenta um elevado grau, dado permitir melhoria continua ao nível dos agentes, ou seja, permite que os agentes sejam estendidos com novas funcionalidades. As características *user-friendly*, multiplataforma, criação de relatórios customizáveis, disponibilização de indicadores gráficos e acessível via *Web*, tal como a maioria das soluções, também foram alcançadas. O E-Sam, tal como a Asset Explorer e Snipe-IT, é baseada em *Open Source.* O E-Sam possui uma característica única, sendo ela a verificação de vulnerabilidades. Hoje em dia, torna-se fulcral dispor dessa funcionalidade devido ao aumento de ataques informáticos e necessidade de adotar medidas preventivas. Considera-se esta funcionalidade muito importante, pois permite além da recolha dos ativos numa determinada organização identificar de forma automática se os ativos estão vulneráveis a algum tipo de ataque conhecidos. Desta forma é possível atuar proactivamente no sentido de mitigar o impacto de potenciais ciberataques. Por fim, a solução apresenta também a característica de pesquisa automática de ativos, característica esta que também se verifica na Asset Explorer (uma solução boa, mas mais dispendiosa).

Este projeto contribuiu para enriquecer a literatura sobre gestão de ativos e gestão de vulnerabilidades. Através deste projeto, diversas empresas podem integrar a ferramenta E-Sam de serviço de pesquisa e gestão de licenças de software de modo a gerir melhor os seus ativos. Pelo facto da ferramenta ser desenvolvida com base em pedidos de empresas clientes da AET Europe que operam nos setores da defesa e que pretendem aplicá-la no decorrer dos próximos anos, a solução alcançada demostra a sua viabilidade comercial. Os agentes têm identificação unívoca na comunicação entre aplicação-agente. É possível a reconfiguração e alteração de modo de operação dos agentes. Por isso, ter uma boa uma solução de SAM como a E-SAM não só ajuda apenas a gerir custos e mitigar riscos, mas também a manter flexibilidade e agilidade com estruturas de custos previsíveis.

O projeto não está isento de limitações. O acesso escasso de ativos para a simulação dificultou a obtenção de resultados reais e em escala. Nota-se que este projeto está limitado a uma única empresa e as conclusões propostas não podem ser generalizadas para todos os casos. Além disso, devido à atual pandemia do vírus COVID-19, não foi possível montar um cenário real nas máquinas dos colaboradores no escritório em Portugal da AET Europe. A realização desse cenário é fundamental para apresentar futuramente o projeto E-Sam aos possíveis clientes da entidade de acolhimento.

Relativamente a trabalho futuro, existe a necessidade de desenvolver mais projetos sobre gestão de ativos devido à sua inegável importância. De uma forma mais especifica, é vital novos estudos e projetos que integrem a ferramenta E-Sam, tais como, integrar a ferramenta *certificate auto discovery tool* (CADT) neste projeto, bem como a gestão de outro tipo de ativos físicos como *routers*, *switches*, entre outros. Por outro lado, futuras empresas poderão colocar em prática este projeto de gestão de ativos e comparar os resultados antes e após a implementação da gestão de ativos.

Bibliografia

[1] R. d. R. Ferreira, “Falhas de segurança custam em média 74 mil euros às PME,” Ntech.news, 2017. [Online]. Available: https://www.ntech.news/falhas-seguranca-custam-74-mil-euros-pme/. [Acedido em 17 Agosto 2020].

[2] K. Pequenino, “Empresas portuguesas entre as que menos sofreram ataques informáticos,” publico.pt, 2020. [Online]. Available: https://www.publico.pt/2020/01/14/tecnologia/noticia/empresas-portuguesas-menos-sofreram-ataques-informaticos-uniao-europeia-1900332. [Acedido em 17 Agosto 2020].

[3] C. Rocha, “Ataques informáticos atingem recorde em 2018, custando milhões às empresas,” insider.dn.pt, 2019. [Online]. Available: https://insider.dn.pt/featured/ataques-informaticos-recorde-2018/16405/. [Acedido em 15 Agosto 2020].

[4] F. Almeida, “Empresas vão perder 6 biliões com ciberataques,” executivedigest.sapo.pt, 2019. [Online]. Available: https://executivedigest.sapo.pt/empresas-vao-perder-6-milhoes-de-milhoes-com-ciberataques/. [Acedido em 15 Agosto 2020].

[5] L. Pereira, “22% das pequenas empresas atingidas por ransomware têm de fechar as portas,” olhardigital, 2017. [Online]. Available: https://olhardigital.com.br/fique_seguro/noticia/22-das-pequenas-empresas-atingidas-por-ransomware-tem-de-fechar-as-portas/70129. [Acedido em 12 Agosto 2020].

[6] P. Biswas, “ISO 27001:2013 A. 8 Asset management,” 2019. [Online]. Available: https://isoconsultantkuwait.com/2019/12/08/iso-270012013-a-8-asset-management/. [Acedido em 12 Julho 2020].

[7] ISO, “Asset management — Overview, principles and terminology,” 2014. [Online]. Available: https://www.iso.org/obp/ui#iso:std:iso:55000:ed-1:v2:en. [Acedido em 10 Março 2020].

[8] ISO, “IT asset management systems — Requirements,” 2017. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:19770:-1:ed-3:v1:en. [Acedido em 23 Março 2020].

[9] P. Editora, SNC - SISTEMA DE NORMALIZAÇÃO CONTABILÍSTICA, 2019.

[10] “Statement of Financial Standards no 141: Bussiness Combination,” Financial Accounting Standard Board, FASB (2001a), Junho 2001. [Online]. Available: http://www.fasb.org/cs/BlobServer?blobcol=urldata&blobtable=MungoBlobs&blobkey=id&blobwhere=1175820919868&blobheader=application%2Fpdf. [Acedido em 20 Agosto 2020].

[11] J. L. Santos, J. Gomes, L. Fernandes, P. Pinheiro e P. Schmidt, em *Ativos intangíveis: fonte de vantagem competitiva*, Porto Alegre, 2006.

[12] J. Breda, “SOFTWARE ASSET MANAGEMENT (SAM) – “O QUE OS OLHOS NÃO VEEM O BUDGET NÃO SENTE”,” 2019. [Online]. Available: https://blog-br.softwareone.com/software-asset-management. [Acedido em 12 Julho 2020].

[13] “IT asset management best practices,” [Online]. Available: https://www.manageengine.com/products/service-desk/it-asset-management/. [Acedido em 14 Junho 2020].

[14] R. Young e U. Arora, “The Business Value of Software Asset Management,” 2016.

[15] “COBIT vs ITIL vs TOGAF: Which Is Better For Cybersecurity?,” [Online]. Available: https://www.upguard.com/blog/cobit-vs-itil-vs-itsm-which-is-better-for-cybersecurity-and-digital-resilience. [Acedido em 22 Julho 2020].

[16] “CICLO DE VIDA ITIL,” [Online]. Available: http://www.itilportugal.pt/ciclo-vida-itil/. [Acedido em 24 Julho 2020].

[17] A. M. Q. Varela , M. P. Méxas e G. M. Drumond , “The scenario of software asset management (SAM) in large and midsize companies,” *Independent Journal of Management & Production,* pp. 301-320, 2018.

[18] “ISO/IEC 19770-1,” itamstandards, [Online]. Available: http://www.itamstandards.org/iso-iec-19770-1/. [Acedido em 12 Março 2020].

[19] P. Geelen e P. Almeida, “Software Asset Management - SAM,” Microsoft, 2017. [Online]. Available: https://social.technet.microsoft.com/wiki/pt-br/contents/articles/13564.software-asset-management-sam.aspx. [Acedido em 10 Dezembro 2019].

[20] “Gestão de Ativos de Software e Otimização de Licenciamento,” kpmg, 2016. [Online]. Available: https://assets.kpmg/content/dam/kpmg/br/pdf/2016/10/br-sam-kpmg-snow-2016.pdf. [Acedido em 12 Fevereiro 2020].

[21] R. Gaspar, “Open Asset Management,” 2014.

[22] “Top 12 Benefits of Software Asset Management,” 2017. [Online]. Available: https://ca.insight.com/en_CA/content-and-resources/2017/02062017-the-top-12-benefits-of-software-asset-management.html. [Acedido em 12 Janeiro 2020].

[23] “Gerenciamento de Ativos de Software,” 2007. [Online]. Available: https://www.kpmg.com.br/publicacoes/Gerenciamento_Ativos_30jun.pdf. [Acedido em 15 Janeiro 2020].

[24] M. Stone, C. Irrechukwu e L. Kauffman, “IT Asset Management,” 2018. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1800-5.pdf. [Acedido em 15 Janeiro 2020].

[25] T. H. Chaves e A. Castro Jr., “Elementos de um Ambiente Multiagente com Intermediação para Suporte à Aprendizagem,” 2020.

[26] N. d. Santos, “Agentes de Software em Ambientes Educacionais Mediados por Computador,” *Revista Brasileira de Informática na Educação – V.11 N. 1,* 2003.

[27] M. J. Wooldridge e N. R. Jennings, “Intelligent Agents: Theory and Practice,” 1995.

[28] B. Hermans, “Intelligent Software Agents on the Internet: An Inventory Offered Functionality of (near) Future Developments,” 1996. [Online]. Available: https://firstmonday.org/ojs/index.php/fm/article/download/514/435. [Acedido em 9 Janeiro 2020].

[29] F. B. Vavasori e F. A. Gauthier, “Proposta de Ferramenta e Agentes Inteligentes para um Ambiente de Ensino/Aprendizagem na Web,” *Anais do IX Simpósio Brasileiro de Informática na Educação,* 1998.

[30] A. P.-. Jacques e M. F. Oliveira, “Agentes de Software para Análise das Interações em um Ambiente de Ensino a Distância,” *Anais do IX Simpósio Brasileiro de Informática na Educação,* 1998.

[31] F. L. Ferreira e M. Bercht, “Agentes Pedagógicos como Apoio à Avaliação de Competência Técnica,” *Anais do XI Simpósio Brasileiro de Informática na Educação,* 2000.

[32] G. Sakarkar e N. M. Shelke, “A New Classification Scheme for Autonomous Software Agent,” 2009.

[33] G. Weiss, Multiagent Systems a Modern Approach to Distributed Artificial Intelligence, 209.

[34] R. “Gestão de Vulnerabilidades, o que é?,” [Online]. Available: https://realprotect.net/blog/gestao-de-vulnerabilidades-o-que-e/. [Acedido em 13 Setembro 2020].

[35] D. Kosutic, “Avaliação de riscos da ISO 27001: Como combinar ativos, ameaças e vulnerabilidades,” [Online]. Available: https://advisera.com/27001academy/pt-br/knowledgebase/avaliacao-de-riscos-da-iso-27001-como-combinar-ativos-ameacas-e-vulnerabilidades/. [Acedido em 12 Setembro 2020].

[36] “Gestão de Risco de Ativos de Segurança da Informação,” [Online]. Available: https://www.gat.digital/blog/risco-de-ativos-em-seguranca-da-informacao/. [Acedido em 12 Setembro 2020].

[37] “IT Asset Management Tool for your Business,” 2020. [Online]. Available: https://www.manageengine.com/products/asset-explorer/?utm_source=PCMag&utm_medium=PPC&utm_campaign=AEReview. [Acedido em 19 Agosto 2020].

[38] B. Turner, “Best software asset management tools of 2020: SAM software to reduce licensing costs,” 2020. [Online]. Available: https://www.techradar.com/best/best-software-asset-management-tools. [Acedido em 18 Agosto 2020].

[39] “IT Asset Management Tool for your Business,” [Online]. Available: https://www.manageengine.com/products/asset-

explorer/#:~:text=AssetExplorer%20makes%20it%20easier%20to,software%20inventory%20of%20these%20assets.. [Acedido em 15 Agosto 2020].

[40] “InvGate Assets,” [Online]. Available: https://www.invgate.com/assets/. [Acedido em 2020].

[41] “The free*, cloud-based asset management tool,” [Online]. Available: https://www.myassettag.com/assettiger. [Acedido em 21 Agosto 2020].

[42] “SNIPE-IT ASSET MANAGEMENT,” [Online]. Available: https://snipeitapp.com/product. [Acedido em 22 Agosto 2020].

[43] “AET Europe,” 2020. [Online]. Available: https://www.aeteurope.com/. [Acedido em 9 Janeiro 2020].

[44] “Python - O que é e para que serve,” [Online]. Available: https://www.lojadelayouts.com/blog/python/. [Acedido em 18 Setembro 2020].

[45] “O que é o Python? Introdução,” [Online]. Available: https://docs.microsoft.com/pt-br/learn/modules/python-introduction/1-introduction. [Acedido em 18 Setembro 2020].

[46] “Get Your Educational Tool,” [Online]. Available: https://www.jetbrains.com/pycharm-edu/download/#section=windows. [Acedido em 12 Janeiro 2020].

[47] “Integração Power BI com ERP PRIMAVERA,” [Online]. Available: https://www.inovflow.pt/site/power-bi/. [Acedido em 19 Setembro 2020].

[48] M. Pereira, “Power BI: O que é e para que serve,” [Online]. Available: https://www.voitto.com.br/blog/artigo/o-que-e-power-bi. [Acedido em 19 Setembro 2020].

[49] M. Tarcio, “28 MOTIVOS PARA VOCÊ COMEÇAR A USAR O POWER BI AGORA!,” [Online]. Available: https://uaismart.com/28-motivos-para-voce-comecar-a-usar-o-power-bi-agora/. [Acedido em 18 Setembro 2020].

[50] “What is PostgreSQL? Why use PostgreSQL?,” [Online]. Available: https://www.postgresql.org/about/. [Acedido em 20 Setembro 2020].

[51] G. Powell, Beginning Database Design, Wiley Publishing Inc, 2006.

[52] T. Teorey, S. Lightstone, T. Nadeau e H. Jagadish, Database Modeling and Design: Logical Design, Fourth Edition, Elsevier Inc, 2006.

[53] C. Wright, “The IT Regulatory and Standards Compliance Handbook,” [Online]. Available: https://www.sciencedirect.com/topics/computer-science/open-source-vulnerability-database. [Acedido em 21 Setembro 2020].

[54] “VulDB,” [Online]. Available: https://vuldb.com/. [Acedido em 2 Janeiro 2020].

[55] “NATIONAL VULNERABILITY DATABASE,” [Online]. Available: https://nvd.nist.gov/. [Acedido em 3 Janeiro 2020].

[56] “What Is Certificate-Based Authentication and Why Should I Use It?,” [Online]. Available: https://www.globalsign.com/en/blog/what-is-certificate-based-authentication. [Acedido em 10 Março 2020].

[57] “winreg – Windows registry access¶,” [Online]. Available: https://docs.python.org/3.0/library/winreg.html. [Acedido em 10 Janeiro 2020].

[58] “El módulo platform – Información del sistema y ordenador,” [Online]. Available: https://recursospython.com/guias-y-manuales/platform-sistema-y-ordenador/. [Acedido em 15 Fevereiro 2020].

[59] S. Robinson, “Scheduling Jobs with python-crontab,” [Online]. Available: https://stackabuse.com/scheduling-jobs-with-python-crontab/. [Acedido em 2020 Julho 10].

[60] A. Bile, “Programmatic way of Creating Task in Windows Task Scheduler,” [Online]. Available: https://medium.com/@ajay.bile007/programmatic-way-of-creating-task-in-windows-task-scheduler-8673c5d5b897. [Acedido em 20 Junho 2020].

[61] “Understanding Digital Certificates,” Microsoft, [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/tn-archive/bb123848(v=exchg.65)?redirectedfrom=MSDN. [Acedido em 10 Outubro 2020].

[62] J. Nelson, “Tutorial 0 - Get it Running,” [Online]. Available: http://postgrest.org/en/v7.0.0/tutorials/tut0.html. [Acedido em 02 Fevereiro 2020].

[63] “CVE-2010-3128 Detail,” [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2010-3128. [Acedido em 20 Agosto 2020].

[64] “CVE-2015-6971 Detail,” [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2015-6971. [Acedido em 20 Agosto 2020].

[65] M. Rouse, “BSD licenses,” [Online]. Available: https://whatis.techtarget.com/definition/BSD-licenses. [Acedido em 15 Setembro 2020].

[66] “What is ITIL Best Practice?,” Axelos, [Online]. Available: https://www.axelos.com/best-practice-solutions/itil/what-is-itil. [Acedido em 21 Janeiro 2020].

[67] B. Mackenzie e a. et, IFRS 2012 - Interpretação e aplicação, KPMG, 2012.

Anexos

Anexo 1 – *Resposta da chamada a uma chamada à API VulDB*


```
{
    "response": {
        "version": "3.19",
        "format": "json",
        "status": "200",
        "lang": "en",
        "items": 3,
        "consumption": 1,
        "remaining": 1997,
        "querylimit": 100,
        "querylimitmax": 500,
        "timestamp": "1569396519",
        "rtt": 0,
        "etag": "285302e7a19f108f-339212fb83e8faa8-dcca48101505dd86"
    },
    "request": {
        "timestamp": "1569396519",
        "apikey": "valid",
        "userid": "1",
        "details": 0,
        "sort": "entry_timestamp_create",
        "fields":
"vulnerability_cwe,vulnerability_cvss2_vuldb_basescore,vulnerability_cvss2_nvd_basescore",
        "type": "id",
        "value": "5,23,42"
    },
    "result": [
        {
            "entry": {
                "id": "42",
                "title": "Cisco Secure ACS up to 3.1.1 on Windows Admin memory corruption",
                "timestamp": {
                    "create": "1051056000",
                    "change": "1528838409"
                }
            },
            "vulnerability": {
                "risk": {
                    "value": "2",
                    "name": "medium"
                },
                "cwe": "CWE-119",
                "cvss2": {
                    "vuldb": {
                        "basescore": "6.8"
                    },
                    "nvd": {
                        "basescore": "7.5"
                    }
                }
            },
            "advisory": {
```

```
            "date": "1051056000"
         },
         "source": {
            "cve": {
               "id": "CVE-2003-0210"
            }
         }
      },
      {
         "entry": {
            "id": "23",
            "title": "Linux Kernel up to 2.2.23 proc\/pid\/mem mmap PROT_READ denial of
service",
            "timestamp": {
               "create": "1048809600",
               "change": "1516378012"
            }
         },
         "vulnerability": {
            "risk": {
               "value": "1",
               "name": "low"
            },
            "cwe": "CWE-404",
            "cvss2": {
               "vuldb": {
                  "basescore": "1.9"
               },
               "nvd": {
                  "basescore": "2.1"
               }
            }
         },
         "advisory": {
            "date": "1048809600"
         },
         "source": {
            "cve": {
               "id": "CVE-2002-1380"
            }
         }
      },
      {
         "entry": {
            "id": "5",
            "title": "Linux Kernel up to 2.4.19 privilege escalation",
            "timestamp": {
               "create": "1044316800",
               "change": "1496840061"
            }
         },
         "vulnerability": {
            "risk": {
               "value": "1",
               "name": "low"
            },
            "cwe": "CWE-269",
            "cvss2": {
               "vuldb": {
```

```
                "basescore": "4.1"
            },
            "nvd": {
                "basescore": "3.6"
            }
        }
    },
    "advisory": {
        "date": "1044316800"
    },
    "source": 10
    }
```

Apêndices

Apêndice 1 – Exemplo aplicação mobile Power BI

Na figura seguinte temos um exemplo de um relatório no Smartphone. Como foi referido anteriormente, o Power BI possui uma aplicação móvel. Esta funcionalidade facilita e muito a gestão das vulnerabilidades bem como estamos sempre atualizados sobre riscos inerentes.

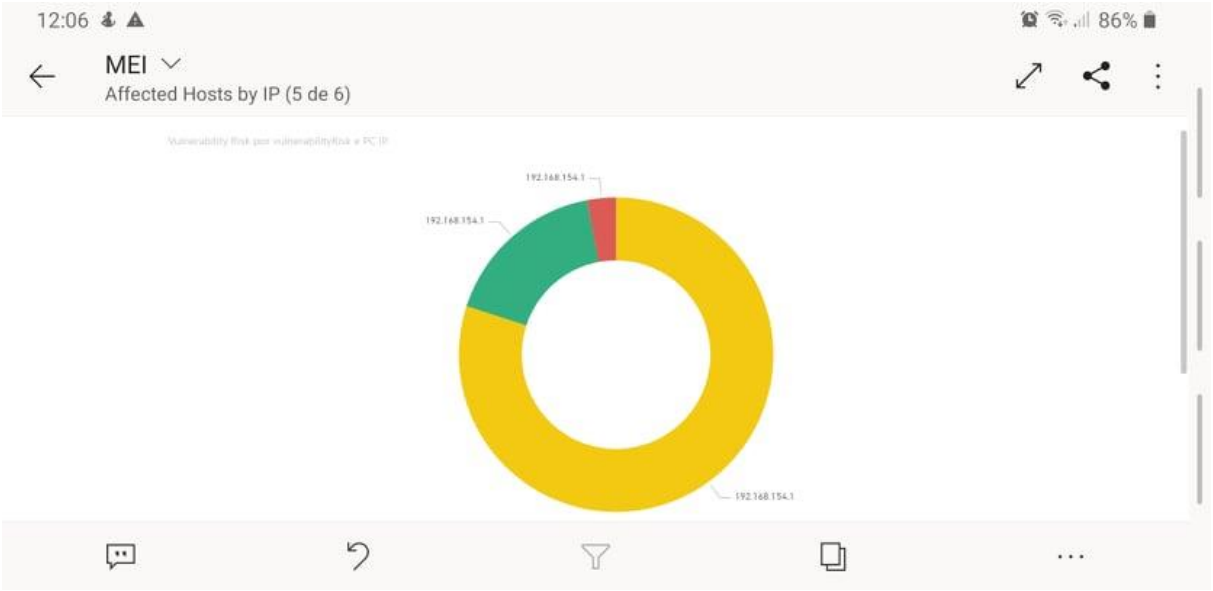


Figura 32 - Exemplo dos relatórios do Power BI no Smartphone

Apêndice 2 – Código para recolha de licenças de software do Windows *Registry* através de PowerShell

Neste apêndice é apresentado uma das primeiras abordagens para recolha de licenças de software com recurso ao Windows *Registry* e PowerShell.

```
$regex = 'ProductID'

Get-ChildItem HKLM:\SOFTWARE -ErrorAction SilentlyContinue -Recurse | ForEach-Object {
  foreach ($value in $_.GetValueNames() -match $regex) {
    if ($data = $_.GetValue($value)) {
      [PSCustomObject]@{
        Key = $_.Name
        Value = if ($value) { $value } else { '(default)' }
        Data = $data
      }
    }
  }
}
```

Figura 33 - Abordagem inicial em PowerShell script

Apêndice 3 – Tarefa criada *Windows Scheduler*

Relativamente à figura seguinte temos o resultado da execução do método de agendamento da ferramenta E-Sam.

Name	Status	Triggers	Next Run Time	Last Run Time
MicrosoftEd...	Ready	Multiple triggers defined	9/28/2020 8:53:11 AM	9/27/2020 5:59:05 PM
MicrosoftEd...	Ready	At 8:53 AM every day - After triggered, repeat every 1 hour for a duration of 1 day.	9/27/2020 7:53:11 PM	9/27/2020 6:53:12 PM
MonitorGro...	Ready	At 10:00 AM every day	9/28/2020 10:00:00 AM	9/27/2020 5:55:45 PM
npcapwatch...	Ready	At system startup		9/27/2020 5:59:03 PM
NvProfileUp...	Ready	At 12:25 PM every day	9/28/2020 12:25:54 PM	9/27/2020 4:11:53 PM
NvProfileUp...	Ready	At log on of any user		9/27/2020 6:01:05 PM
nWizard_{B2...	Ready	At log on of any user		9/27/2020 5:59:05 PM
OneDrive St...	Ready	At 4:00 AM on 5/1/1992 - After triggered, repeat every 1.00:00:00 indefinitely.	9/28/2020 1:04:36 PM	8/8/2018 12:50:50 PM
OneDrive St...	Ready	At 10:00 AM on 5/1/1992 - After triggered, repeat every 1.00:00:00 indefinitely.	9/28/2020 10:22:48 AM	9/27/2020 4:11:53 PM
OneDrive St...	Ready	At 6:00 PM on 5/1/1992 - After triggered, repeat every 1.00:00:00 indefinitely.	9/28/2020 6:38:54 PM	11/30/1999 12:00:00 AM
ProtonVPN ...	Ready	On event - Log: Application, Source: ProtonVPN, Event ID: 1		11/30/1999 12:00:00 AM
RtHDVBg_D...	Running	At log on of any user		9/27/2020 5:59:35 PM
RtHDVBg_P...	Running	At log on of any user		9/27/2020 5:59:35 PM
RTKCPL	Ready	At log on of any user		9/27/2020 5:59:35 PM
SAM	Ready	At 6:57 PM every day	9/27/2020 6:57:17 PM	9/27/2020 6:56:17 PM

Figura 34 - Windows Scheduler tarefa SAM

Apêndice 4 – Página Web repositório dos agentes

Na figura 35 podemos ver o certificado configurado para autenticação no servidor Web configurado.

Relativamente à figura 36, esta representada a página web criada configurada com certificado SSL com o objetivo de servir como repositório dos agentes. Através desta página é possível fazer download dos agentes manualmente.

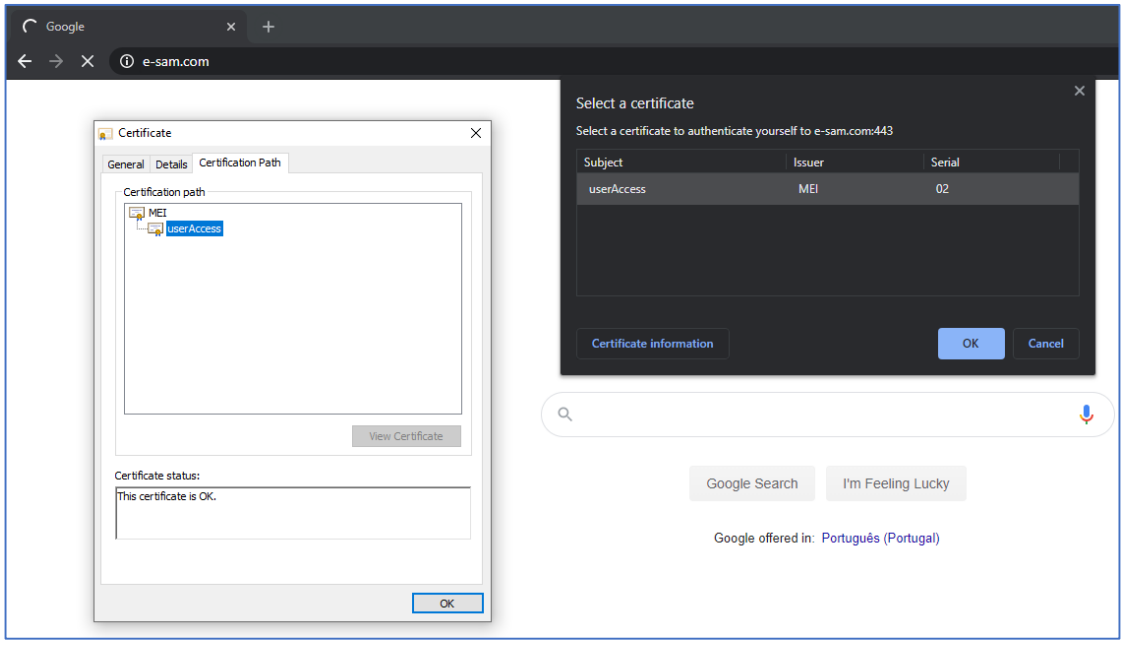


Figura 35 - Certificada autenticação cliente

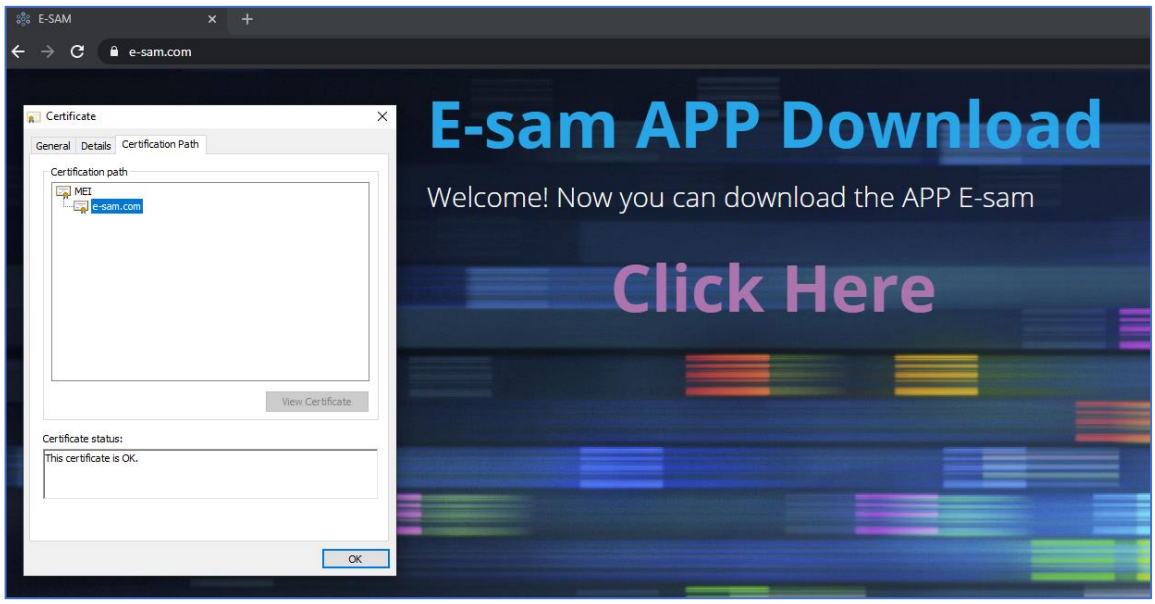


Figura 36 - Página de download dos agentes e certificado servidor Apache

Apêndice 5 – *Schema* base de dados

Neste apêndice é apresentado o *schema* da base de dados criada.

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 12.2
-- Dumped by pg_dump version 12.2

-- Started on 2020-09-26 11:58:07

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

--
-- TOC entry 2842 (class 1262 OID 16393)
-- Name: e-sam; Type: DATABASE; Schema: -; Owner: postgres
--

CREATE DATABASE "e-sam" WITH TEMPLATE = template0 ENCODING = 'UTF8'
LC_COLLATE = 'English_United States.1252' LC_CTYPE = 'English_United States.1252';
```

```
ALTER DATABASE "e-sam" OWNER TO postgres;

\connect -reuse-previous=on "dbname='e-sam'"

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

--
-- TOC entry 207 (class 1259 OID 41006)
-- Name: liceid_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public.liceid_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public.liceid_seq OWNER TO postgres;

SET default_tablespace = '';

SET default_table_access_method = heap;

--
-- TOC entry 206 (class 1259 OID 40998)
-- Name: licencesFound; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."licencesFound" (
    id bigint DEFAULT nextval('public.liceid_seq'::regclass) NOT NULL,
    "PCIP" text,
    "PathRegistry" text,
    "RegistryKey" text,
    "RegistryValue" text,
    "DateTime" time without time zone,
);


ALTER TABLE public."licencesFound" OWNER TO postgres;

--
-- TOC entry 203 (class 1259 OID 24620)
-- Name: pid_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public.pid_seq
    START WITH 1
```

```
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public.pid_seq OWNER TO postgres;

--
-- TOC entry 202 (class 1259 OID 16433)
-- Name: programsInstalled; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."programsInstalled" (
    "ProgramName" text,
    "ProgramVersion" text,
    "ProgramPublisher" text,
    "PCName" text,
    "PCWinVersion" text,
    "PCIP" text,
    id bigint DEFAULT nextval('public.pid_seq'::regclass),
    "DateTime" timestamp without time zone,
    "UniqueName" text NOT NULL,
    "City" text
);


ALTER TABLE public."programsInstalled" OWNER TO postgres;

--
-- TOC entry 205 (class 1259 OID 24631)
-- Name: vulid_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public.vulid_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public.vulid_seq OWNER TO postgres;

--
-- TOC entry 204 (class 1259 OID 24623)
-- Name: vulEntries; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."vulEntries" (
    "entryChange" date,
    "entryCreate" date,
    "entryId" bigint,
    "entryTitle" text,
    "vulnerabilityRisk" text,
    id bigint DEFAULT nextval('public.vulid_seq'::regclass) NOT NULL,
    "uniqueNamePFound" text,
    "sourceCVE" text NOT NULL
);
```

```
ALTER TABLE public."vulEntries" OWNER TO postgres;

--
-- TOC entry 2706 (class 2606 OID 16440)
-- Name: programsInstalled UniqueName; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public."programsInstalled"
    ADD CONSTRAINT "UniqueName" PRIMARY KEY ("UniqueName");


--
-- TOC entry 2710 (class 2606 OID 41005)
-- Name: licencesFound licencesFound_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public."licencesFound"
    ADD CONSTRAINT "licencesFound_pkey" PRIMARY KEY ("UniqueName");


--
-- TOC entry 2708 (class 2606 OID 32862)
-- Name: vulEntries vulEntries_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."vulEntries"
    ADD CONSTRAINT "vulEntries_pkey" PRIMARY KEY ("sourceCVE");


-- Completed on 2020-09-26 11:58:07

--
-- PostgreSQL database dump complete
--
```

Apêndice 6 – *Regras definidas no Postgres para chamadas REST (PostRest)*

Neste apêndice são apresentadas regras definidas na base de dados para que possam ser feitas chamadas REST pelos agentes.

```
CREATE ROLE unauthorized noinherit login password 'password';

SET LOCAL ROLE agentes;

CREATE ROLE agentes nologin;
GRANT agentes TO authenticator;
GRANT USAGE ON schema api TO agentes;
```

```
GRANT INSERT ON api."licencesFound" TO agentes;
GRANT USAGE, SELECT ON SEQUENCE liceid_seq TO agentes;

GRANT INSERT ON api."programsInstalled" TO agentes;
GRANT USAGE, SELECT ON SEQUENCE pid_seq TO agentes;

GRANT INSERT ON api."vulEntries" TO agentes;
GRANT USAGE, SELECT ON SEQUENCE vulid_seq TO agentes;
```