UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

ROBOPHISH: UM SISTEMA DE ROBÔS DE *SOFTWARE* (RPA) PARA AGILIZAR E AUTOMATIZAR O TRATAMENTO DO *PHISHING*

Inês Alexandra Silva da Fonseca Rodrigues

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

Trabalho de projeto orientado por:
Prof.ª Doutora Ana Luísa do Carmo Correia Respício

2020

Agradecimentos

Em primeiro lugar gostaria de agradecer Eng.º José Alegria que acompanhou de perto este projeto e demonstrou sempre imensa disponibilidade para ajudar a guiar este percurso. Um grande agradecimento também à professora Ana Luísa Respício, minha orientadora, por toda a dedicação e disponibilidade que mostrou para orientar este projeto. A todos os colegas da DCY que me acolheram, apoiaram e ajudaram um sincero obrigada. Saliento em particular o Eng.º Carlos Cabral que trabalhou de perto comigo para que este projeto fosse concluído com sucesso.

Agradeço a todos os meus colegas de trabalho na Altice Portugal, Sara Nascimento, Cátia Rodrigues, José Águas, João Miranda e Beatriz Sécio, com vocês esta jornada foi mais fácil. Obrigada pelos sorrisos, pela motivação e pelos bons momentos passados. Em particular gostaria de agradecer à Sara e à Cátia que me acompanharam mais de perto, motivando e ajudando-me sempre. Obrigada por me aturarem, as vossas palavras, boa disposição e apoio constante foram fundamentais para o desenvolvimento deste projeto.

Quero agradecer em especial aos meus pais, Helena e Paulo Rodrigues, e ao meu irmão Rodrigo Rodrigues. Sem vocês nada disto seria possível, o vosso apoio incondicional permitiu que chegasse até aqui. Vocês são sempre a minha maior motivação, tudo o que consegui na minha vida foi graças a vós. Obrigada por tudo o que fizeram e fazem por mim, por escolherem sempre ver o melhor em mim, pela paciência, pelo carinho e pela dedicação.

Agradeço também ao resto da minha família, às minhas avós Helena e Graciete, tio Henrique, tia Cristina e prima Ana. Obrigada por me acompanharem desde sempre, por todo o apoio que me deram até hoje, por me ajudarem a crescer e por acreditarem em mim.

Por fim agradeço a todos os meus amigos que me acompanharam nesta jornada, desde que iniciei a minha vida académica até hoje. Todos vocês fazem parte da minha vida e também graças ao vosso apoio cheguei até aqui. Os momentos passados convosco foram distrações necessárias e bem-vindas que tornaram este projeto mais fácil de concluir. Joca Barbosa, Catarina Fitas, Gabriela Gonçalves, Álvaro Teles, Aurora Zanga e Catarina Faria um beijinho especial para vocês que me acompanham há muitos anos e que nunca deixaram de fazer parte da minha vida. Obrigada pela paciência infinita, pelo carinho e amizade, por serem sempre um ombro amigo nos momentos difíceis e um sorriso rasgado nos mais felizes.

Aos meus pais, ao meu irmão e às minhas avós.

Resumo

Atualmente o *phishing* apresenta-se como uma ameaça relevante e pertinente, dado o aumento significativo da quantidade e complexidade deste ataque. É recorrente os utilizadores receberem grandes quantidades de emails de *phishing*, quer no contexto empresarial como no pessoal, sendo possível que estes representem um impacto muito significativo nas vítimas.

A sofisticação deste tipo de ataques é cada vez maior, verificando-se um aumento de ataques de *spear phishing*, altamente direcionados, bem redigidos e eficazes. Estes ataques servem frequentemente como mecanismo para levar à infeção com *malware* mais sofisticado que, por sua vez, pode incluir novos tipos de *ransomware* ou *wipers*. Concorrentemente os processos atuais de resposta a esta crise ainda são pouco integrados, pouco eficientes e muito *ad-hoc*. Estes fatores levam a um dificuldade acrescida na prevenção e mitigação do *phishing*, fazendo deste um problema atual e complexo.

Neste projeto são analisados os atuais processos e iniciativas da Altice Portugal e é proposta uma solução inovadora mais automatizada e integrada, tirando partido de tecnologias emergentes: Robotic Process Automation (RPA) e *Crowdsourcing*.

O sistema *RoboPHISH* surge então com o objetivo de colmatar este problema, tendo como objetivo principal o bloquear os emails maliciosos enquanto são dadas respostas de forma eficaz e rápida aos colaboradores. O *RoboPHISH* é uma ferramenta capaz de tirar partido do RPA para extrair e analisar emails que os colaboradores consideraram suspeitos e atribuir uma classificação aos mesmos. O sistema está divido em três componentes principais, a primeira corresponde à denúncia dos emails, a segunda à análise e atribuição de categoria dos mesmos e por último a atuação junto das plataformas para mitigar o problema.

O sistema foi desenvolvido, testado e encontra-se neste momento em produção na Direção de Cibersegurança (DCY), mostrando ser uma ferramenta útil e eficaz no combate ao *phishing* e apresentando resultados promissores.

Palavras-chave: *Phishing*, Automação, RPA, Cibersegurança, *Crowdsourcing*

Abstract

Nowadays phishing attacks are a relevant and pertinent threat given the substantial increase in their quality and quantity. Mailboxes filled with phishing emails are common occurrences, whether in a personal of professional context, possibly impacting significant their victims.

The sophistication in these attacks is increasing, with a greater number of spear phishing attacks, highly targeted, well written and effective. These attacks frequently serve as a vehicle to install sophisticated malware that may contain new types of ransomware or wipers. The current processes to respond to this crisis are not fully capable of dealing with it, being poorly integrated, inefficient and ad-hoc. These problems lead to a difficulty in the prevention and mitigation of phishing attacks, making this a current and complex problem.

In this project the current processes and initiatives of the company (Altice Portugal) are analyzed and a more automated and integrated innovative solution is proposed. This solution takes advantage of emerging technologies: Robotic Process Automation (RPA) and Crowdsourcing.

The RoboPHISH system emerges with the aim of addressing this problem, with the main goal of blocking malicious emails while giving efficient and quick responses to the employees. RoboPHISH is a tool capable of taking advantage of RPA to extract and analyze emails that employees consider suspicious and assign a rating to them. The system is divided into three main components, the first being the email report, the second the analyses and categorization of the email and the third the mitigation of the attack in the correct platforms.

The system was developed, tested and is currently in production at Direção de Cibersegurança (DCY), showing to be a useful and effective tool in combating phishing and presenting promising results.

Keywords: Phishing, Automation, RPA, Cibersecurity, Crowdsourcing

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Motivação

Nos dias de hoje uma das formas mais frequentes de comunicação no mundo empresarial é o correio eletrónico (email). Marcar reuniões, distribuir tarefas, trocar informação entre colaboradores, são tarefas tipicamente executadas através deste meio de comunicação. Algumas empresas já tratam de certos assuntos, mais informais, utilizando outras tecnologias, tais como mensagens instantâneas ou chamadas, mas para as comunicações oficiais o mais comum ainda é o email.

A troca de emails veio facilitar muito a maneira como comunicamos, esta passou a ser uma necessidade fundamental tanto na vida das pessoas no geral como em particular no trabalho dos colaboradores das empresas. Neste momento o correio electrónico é indispensável, ao ponto das interrupções neste serviço poderem causar danos significativos. Um estudo da *Spiceworks* apurou que desde Agosto de 2017 a Agosto de 2018, das 762 empresas de IT questionadas, 30% afirmou ter perdido receita devido a falhas de serviço ou negócio no geral [16].

Dentro de uma grande empresa, com milhares de funcionários, o número de emails em circulação é bastante elevado. Isto faz com que os serviços de email de uma empresa sejam um alvo perfeito para os agentes maliciosos. Os emails de *phishing*[1] são uma maneira acessível de conseguir comprometer uma máquina ou obter informação. O tipo de *phishing* pode variar conforme a maneira como é feito o ataque ou com o objetivo principal de um agente malicioso. Um dos objetivos mais comuns é a obtenção de credenciais de um dado utilizador, para mais tarde utilizar noutras máquinas e sistemas ou até mesmo vendê-las na *Internet*. Outro objetivo pode ser infetar as máquinas da empresa com vírus, *e.g. malware* (*software* desenvolvido para danificar ou obter acesso não autorizado a um computador) ou *ransomware* (*software* desenvolvido para bloquear o acesso a um computador até que uma quantia monetária seja paga).

Visto que dentro de uma empresa há colaboradores que têm acesso a informação muito sensível, desde dados pessoais de clientes a acessos a máquinas críticas da empresa, é habitual que as organizações sejam constantemente inundadas com emails de *phishing*.

Conforme as tecnologias de proteção e mitigação de ataques de *phishing* evoluem, é natural que os agentes maliciosos modifiquem as suas técnicas também. Embora já exista *software* capaz

[1] Prática fraudulenta de enviar emails que supostamente pertencem a empresas respeitáveis, a fim de induzir indivíduos a revelar informações pessoais.

de reconhecer e bloquear imediatamente as tentativas de *phishing* menos avançadas, cada vez vamos tendo emails maliciosos mais sofisticados e mais direcionados a utilizadores específicos.

No caso particular da Altice Portugal a empresa sofre frequentemente ataques de *phishing*, embora muitos sejam bloqueados pelas ferramentas *anti-spam/ phishing* da empresa, e nunca cheguem aos utilizadores. Caso algum colaborador receba um email não filtrado, e carregue num *link* malicioso, existem ferramentas que servem como barreira entre a *Internet* e os utilizadores, bloqueando IPs maliciosos e monitorizando o tráfego. De forma a aumentar os cuidados de cibersegurança dos colaboradores existem formações, *pop-ups* de consciencialização e testes regulares de *phishing*.

Os emails maliciosos que não são bloqueados pelas ferramentas da empresa, devem ser identificados e reportados pelos colaboradores que os recebem. O processo de denúncia destes emails ainda não é feito de forma estandardizada, o que significa que chegam à caixa de correio dos analistas de cibersegurança emails reportados como *phishing* de maneiras bastante diferentes. Para além disto, a análise destes emails reportados tende a consumir muito tempo aos analistas, o que muitas vezes pode levar a falhas neste processo.

1.2 Objetivos

Este projeto pretende atuar no caso particular da análise dos emails denunciados como *phishing* e SPAM pelos colaboradores. O objetivo principal deste projeto é a automatização e inovação dos processos atuais de resposta a incidentes de *phishing* bem como a padronização das ações efetuadas sempre que há um email reportado pelo colaborador. O projeto tem por base uma tecnologia atual e emergente, RPA, e algoritmos inovadores de *crowdsourcing* para tomar decisões sobre os emails.

A uniformização do modo como é reportado *phishing* na Altice Portugal é extremamente importante para manter coerência e facilitar a análise dos emails comprometidos. De forma a conseguir esta uniformização foi feita a adição de um botão para reportar *phishing* no Outlook. Este botão, que mais tarde será alargado para todos os colaboradores, facilitará a denúncia de emails potencialmente maliciosos.

1.3 Contribuições

A grande contribuição deste projeto foi o desenho e concretização do sistema *RoboPHISH* que se apresenta como uma ferramenta capaz de fazer frente ao problema do *phishing*. O sistema está divido em três componentes principais: denúncia de emails, análise e atribuição de categorias dos mesmos e atuação junto das plataformas para mitigar o problema. Neste projeto foi fundamental conseguir a automatização de todo o processo de análise dos emails denunciados pelos colaboradores. Este robô foi desenvolvido considerando a informação recolhida no levantamento de processos, efetuando as alterações necessárias para incrementar a eficácia do processo, em comparação com o processo manual atual.

Para além das contribuições supramencionados, este projeto tem uma grande componente de

inovação, visto que para além do processo de denúncia de *phishing* foi necessário desenvolver e implementar um algoritmo de *crowdsourcing*. O algoritmo tem como função principal atribuir um valor de *accuracy* aos colaboradores que denunciam emails suspeitos, podendo desta forma tomar decisões sobre o estado do email (SPAM, *phishing* ou *clean*). O algoritmo desenvolvido deverá tomar decisões adequadas sobre os emails reportados como SPAM ou *phishing*, sendo este o ponto fulcral e revolucionário do projeto. Desenvolver este algoritmo inovador, funcional e com resultados adequados é um dos objectivos mais importantes do sistema *RoboPHISH*.

1.4 Estrutura do documento

O documento está estruturado da seguinte forma:

- Capítulo 2 - **Contexto** - Descrição detalhada do problema geral de *phishing*, incluindo o tipo de ataques, como são feitos e quais os alvos principais. Exposição das medidas mais comuns e eficazes na mitigação de ataques de *phishing*. Uma avaliação sobre o *phishing* na Altice Portugal, tanto do tipo e frequência de ataques sofridos, como do cenário atual de mitigação e prevenção dos mesmos, será detalhada neste capítulo.

- Capítulo 3 - **Ferramentas** - Descrição das ferramentas utilizadas no âmbito do projeto, focando em particular os instrumentos mais fundamentais à analise, deteção e bloqueio de ataques de *phishing*. As ferramentas mencionadas são: Blue Prism (BP), Anubis, HPSM, Elastic Search e VirusTotal.

- Capítulo 4 - **Arquitetura do RoboPHISH** - Proposta de uma solução automática de análise de emails reportados como *phishing* - sistema *RoboPHISH*. Apresentação das arquiteturas e requisitos necessários.

- Capítulo 5 - **Implementação do RoboPHISH** - Apresentação do sistema *RoboPHISH* implementado. Demonstração do processos implementados em BP bem como do código em Visual Basic (VB).

- Capítulo 6 - **Resultados** - Apresentação e discussão dos resultados do sistema *RoboPHISH*. Apresentação de métricas e gráficos para acompanhar ao longo do tempo.

- Capítulo 7 - **Conclusão** - Discussão final sobre os resultados do sistema no geral, vantagens, desvantagens. Para além disso são mencionadas as dificuldades sentidas e trabalho futuro.

Capítulo 2

Contexto

2.1 *Phishing*

O *phishing* é um dos maiores veículos para cometer crimes informáticos apesar de não ser um método recente, continua a causar inúmeros transtornos tanto à vida pessoal como profissional de muitas pessoas. O Google bloqueia, aproximadamente, mais de 100 milhões de ataques de *phishing* todos os dias, com operações que podem durar desde 7 minutos até 13 horas [25].

O *phishing* é um tipo de ataque caracterizado pelo uso de engenharia social, como forma de cometer cibercrimes. O propósito destes ataques é conseguir manipular as vítimas, levando-as a fornecer dados pessoais, acessos privilegiados ou informação confidencial aos *hackers*.

Embora já existam ataques que utilizam SMS (*smishing*) e chamadas de voz (*vishing*), o email continua a ser o serviço de eleição dos ataques de *phishing*. A rápida distribuição de um elevado número de mensagens faz com que o correio eletrónico seja um veículo frequentemente utilizado. Tanto a comunicação de empresas entre si como a comunicação entre empresas e clientes é habitualmente feita através de email, fazendo deste o meio ideal para conseguir obter informação classificada.

Um ataque de *phishing* bem sucedido a um conjunto de colaboradores de uma empresa, processa-se tipicamente da seguinte forma:

- O agente malicioso consegue uma lista de emails de uma empresa através de informação obtida *online*, utilizando redes sociais, informação disponível na *Internet* ou informação obtida ilegalmente na *dark web*;
- Os emails são construídos com o intuito de aliciar a vítima a carregar no URL malicioso, que normalmente está dissimulado;
- O email malicioso não é bloqueado pelas *firewalls* e pelos sistemas que filtram o email da empresa, chegando à caixa de correio do colaborador;
- O colaborador sem suspeitar carrega no URL malicioso, que pode redirecioná-lo para uma página *web* que rouba as suas credenciais ou instala *malware* na sua máquina;
- O agente malicioso foi bem sucedido, acedendo assim a informação possivelmente crítica e sensível.

Os ataques mais sofisticados têm, normalmente, um propósito muito específico, mas podem ser definidos alguns objetivos gerais. O acesso a credenciais do *user*, quer seja para vender na *dark web*[1] ou para utilizar noutras máquinas, tipicamente com mais acessos, é um dos objetivos mais frequentes. A obtenção direta dos dados pessoais e informações confidenciais do *user* também podem ser o alvo do *hacker*. A utilização do *phishing* como meio para a instalação de *malware* (*ransomware*, *spyware*, *wipers*) na máquina da vítima é também cada vez mais comum [8]. Neste tipo de ataque o objetivo é aceder às máquinas das vítimas para as inviabilizarem ou para apagarem ou acederem a dados ou informação confidencial.

2.1.1 *Untargeted Bulk Phishing*

Apesar de existir uma quantidade exacerbante de ataques de *phishing*, temos uma grande disparidade entre os mesmos, variando de ataques relativamente inexperientes, com pouca sofisticação até ataques extremamente complexos e direcionados. Temos um elevado número de ataques que se inserem na categoria de ataques inexperientes (pouco profissionais), tornando-os mais fáceis de identificar. Erros de gramática, palavras mal escritas e um tom muito informal fazem com que estas tentativas de *phishing* sejam pouco credíveis. A falta de credibilidade destes ataques não impede que ainda assim um grande grupo de utilizadores, menos experientes, não possa estar susceptível aos mesmos.

Os ataques de *Untargeted Bulk Phishing* são caracterizados pelo envio de emails iguais para um elevado número de destinatários, sem grande conhecimento sobre os mesmos. Um email onde é pedido que o utilizador clique num *link* de forma a ganhar uma raspadinha, poderá ser um exemplo deste tipo de ataque (Figura 2.1). Estes tipos de *phishing*, não direcionados a um grupo específico de utilizadores, acabam por ser mais fáceis de identificar devido ao elevado número de mensagens enviadas, aos padrões de discurso nos emails, ao facto de muitas vezes retratarem um cenário pouco realista e de estarem escritos num português incorreto.

Figura 2.1: Exemplo de Email de *Bulk Phishing*

2.1.2 *Spear Phishing*

Para além dos tipos de ataques supramencionados, de identificação mais fácil, temos tipos mais complexos, cada vez mais difíceis de detetar. Os ataques direcionados têm uma maior proba-

[1] Conteúdo online que não é indexado nos motores de pesquisa comuns

bilidade de serem bem sucedidos, visto que focam um grupo de pessoas específico. Neste contexto direcionar um ataque significa construir um ataque com um indivíduo ou conjunto de indivíduos específicos em mente, baseando-se na informação recolhida sobre os mesmos.

Os ataques *targeted* de *Spear Phishing* pressupõem um conhecimento significativo sobre as vítimas, fazendo destes uma ameaça muito maior. O objetivo é conseguir informação sobre a futura vítima, analisando a sua presença *online*, de forma a aumentar a probabilidade desta ser vítima de *phishing*. Um número mais pequeno e direcionado de ataques pode ser muito mais eficaz, visto que as vítimas sentir-se-ão mais confiantes da legitimidade do email. Se os *hackers* descobrirem que o seu alvo, por exemplo, tem uma conta num banco específico, poderão tentar imitar a correspondência oficial do banco, aumentando assim a probabilidade da vítima pensar que se trata de um email fidedigno [8].

O sucesso destes ataques está altamente dependente da capacidade do criminoso recolher informação relevante sobre a vítima. Obter esta informação é cada vez mais acessível, visto que a presença *online* da população tem vindo a aumentar. Os dados partilhados nas redes sociais, muitas vezes voluntariamente, são uma excelente forma de obter informação sobre gostos, interesses e dados pessoais. O Linkedin[2] é uma rede com particular interesse, visto que pode facilitar muita informação sobre o local e colegas de trabalho dos seus subscritores [8].

Olhando para o mundo empresarial também é cada vez mais fácil obter informações sobre a estrutura e organização de uma empresa. Nos dias de hoje, a grande parte das empresas tem uma forte presença *online*, onde fornecem dados relevantes. Outra possibilidade é a análise das ofertas de emprego da empresa, que muitas vezes detalham o tipo de tecnologias presentes. Para além disto, há sempre a possibilidade de obter informação ilegalmente através da *dark web*, onde muitas vezes é possível encontrar informação de emails e *passwords*.

2.1.3 *Whaling*

Cada vez mais preocupante no mundo do cibercrime são os ataques direcionados de *whaling*. Este tipo de ataques, também denominados de CEO *fraud*, são altamente direcionados, precisos e requerem um grande nível de conhecimento sobre a empresa alvo. Os cibercriminosos personificam um cargo *senior* ou importante de uma empresa, com o intuito de enganar os colaboradores que podem sentir-se reticentes em contrariar uma ordem de um superior, mesmo que esta possa parecer pouco comum. Os invasores estudam também a maneira como os cargos superiores comunicam com os seus colaboradores, imitando a sua forma de comunicação [12]. Empresas que utilizam o email como meio primário de comunicação e em que os pedidos de transferências bancárias por email são prática comum, estão mais vulneráveis a este tipo de ataques.

Com dados obtidos ilegalmente de contas e passwords, é possível enviar mensagens através de um email empresarial comprometido, com o intuito de levar os colaboradores a transferir grandes quantias de dinheiro para a conta bancária do criminoso [8]. Quando os criminosos não conseguem obter as contas reais, utilizam endereços de email muito semelhantes, com letras de diferença, para tentar enganar a vítima. (Figura 2.2)

[2]https://www.linkedin.com/

Esta nova forma de fazer ataques de *phishing* é extremamente perigosa e difícil de detetar, visto que os emails para além de estarem bem escritos e serem muito idênticos aos emails reais, podem até ser enviados de endereços reais comprometidos. Empresas como o Snapchat e a Mattel já foram vítimas deste tipo de ataques, sendo que a segunda perdeu perto de $3M com um *whaling* que personificava o CEO e pedia uma transferência ao departamento financeiro [12].

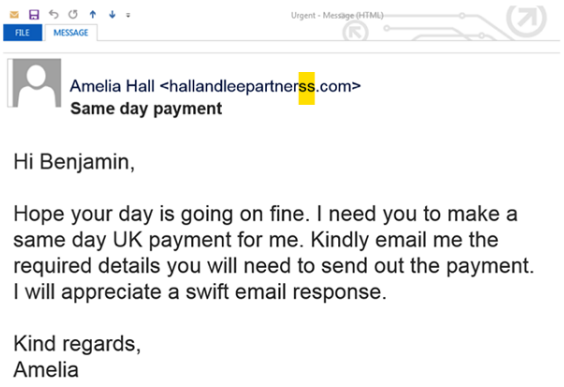


Figura 2.2: Exemplo de um email de *Whaling*

2.1.4 *Phishing* no contexto da Altice Portugal

Apesar da Altice ser uma multinacional, neste projeto o foco será o caso específico da Altice Portugal, que é uma empresa inserida no contexto nacional. Portugal não é um país que se apresente como um grande *target* a nível mundial, os ataques de maiores dimensões são tipicamente direcionados a outros países. Como consequência deste facto, muitas das tentativas de *phishing* acabam por ser bastante amadoras, embora frequentes e em grande quantidade. Um número elevado de emails maliciosos é apanhado pela ferramenta *anti-spam/ phishing* da Altice, Anubis (Secção 3.2). Podemos observar nas estatísticas da plataforma, que desde 31/10/2019 a 30/11/2019 (30 dias), as percentagens de *good vs bad mail* são quase 50/50, com a percentagem de *bad mail* ligeiramente superior a 50% (Figura 3.4).

Em relação aos emails que não são filtrados pelo Anubis, de momento não são mantidas estatísticas sobre os mesmos. Formalmente não está definido um processo uniforme e calendarizado, embora possam ser recolhidas esporadicamente. Manter informações regulares sobre este tipo de ataque poderá ser relevante, não só para estatísticas como para conseguir perceber se os esforços feitos nesta área estão a ser eficazes na diminuição destes ataques. Mesmo sem estatísticas formais é reportada, em particular pelos quadros superiores da empresa, uma grande quantidade de ataques de *phishing*.

2.2 Como prevenir e mitigar ataques de *phishing*

Para combater os ataques de *phishing* são necessárias várias vertentes, visto que estes vão evoluindo de maneiras muitas vezes imprevisíveis. É possível surgir um ataque nunca antes visto, criando uma necessidade de atenção e cuidado extra por parte das empresas. Numa vertente tec-

nológica têm de existir ferramentas e protocolos extremamente seguros implementados, de modo a conseguir filtrar emails maliciosos e proteger assim as máquinas da empresa. Os processos implementados devem permitir a proteção, *report* e mitigação dos ataques de *phishing*.

2.2.1 *Sender Policy Framework*

O *Sender Policy Framework* (SPF) é um protocolo de autenticação de email, que permite que o dono de um domínio indique quais os servidores de email que podem enviar mensagens daquele domínio. O registo de SPF é um registo de entrada do Domain Name System (DNS)[3] que apresenta a lista de endereços de IP autorizados a enviar mensagens. Sempre que um email é enviado para o domínio definido, o SPF é verificado, isto é, os provedores de email verificam o registo de SPF, procurando o nome do domínio listado no DNS. Caso o endereço de IP não esteja listado naquele registo SPF, a mensagem falha a autenticação [20].

Embora este seja um protocolo importante que ajuda no combate ao *phishing* tem as suas limitações. Os registos SPF têm de estar sempre atualizados, ou seja, sempre que há alguma alteração ou adição os mesmos têm de ser modificados, o que pode ser pouco prático, caso hajam alterações frequentes. Apesar desta ser uma verificação feita pelos provedores de email, nem sempre a mensagem é bloqueada pelos mesmos. Talvez o maior problema do SPF, e a razão pela qual este protocolo sozinho não oferece proteção suficiente, seja o facto do SPF não proteger contra os casos em que os agentes maliciosos falsificam o nome do remetente. Para além destas limitações o SPF não funciona com emails redirecionados [20].

2.2.2 *DomainKeys Identified Mail*

O *DomainKeys Identified Mail* (DKIM) é um protocolo que permite a transmissão de mensagens, de forma a que possam ser verificadas pelos provedores das caixas de email. O DKIM verifica que o conteúdo das mensagens é fidedigno, isto é, que o conteúdo não foi alterado desde o remetente até à fonte. Esta verificação é possível através da autenticação criptográfica.

A configuração do DKIM pode variar, de forma a incluir mais ou menos campos na sua assinatura. O remetente decide que elementos do email quer assinar e estes têm de se manter inalterados durante a transição, de outra forma a autenticação falha. Um *hash* [4] é criado automaticamente a partir dos campos selecionados, após a sua criação o *hash* é cifrado utilizando uma chave privada. Só o remetente tem acesso à chave privada. Do lado do remetente o processo está completo, do lado do destinatário é necessário validar a mensagem [19].

Para validar a assinatura, o provedor da caixa de correio deverá executar uma *query* DNS para encontrar a chave pública para esta mensagem. Esta chave pública é a única que corresponde à chave privada que assinou o email. O provedor de email seleciona os elementos do email assinados pelo DKIM e gera o seu próprio *hash* desses elementos. Os dois *hashes* são comparados e se forem iguais sabemos que a mensagem não foi alterada [19].

[3] Sistema para converter nomes em endereços IP numéricos na *Internet*.

[4] Função que converte um conjunto de caracteres noutro conjunto de caracteres único.

2.2.3 Formação

Formação ou treino sobre ataques de *phishing* é um meio muito explorado de os tentar mitigar e prevenir. Este método baseia-se no facto do ataque de *phishing* só ser bem sucedido se a vítima acreditar no mesmo e por isso carregar nos *links* ou ficheiros maliciosos. Os efeitos desta formação não são consensuais, com estudos que afirmam que esta é ineficaz e outros que notam que esta apresenta melhorias significativas, em comparação com indivíduos não formados.

Segundo alguns peritos a educação para a segurança é colocar o peso dos ataques nas vítimas, o que não é eficaz [22]. Considerando que a motivação principal de um utilizador nunca será a segurança, mas sim cumprir o seu trabalho ou objetivo [9], esperar que os mesmos estejam sempre atentos e façam da segurança uma prioridade poderá não ser realista. Para além disso é impossível ensinar o desconhecido, isto é, os novos tipos de ataque serão sempre uma surpresa, não sendo possível abordá-los nas formações.

Por outro lado, existem estudos que revelam que após a formação os utilizadores conseguiram identificar melhor os casos de *phishing*, embora os resultados possam ser temporários. Os melhores resultados são os de treino contínuo, resultando num aumento da confiança em detetar um email malicioso por parte dos colaboradores.

Confiar unicamente nas formações de segurança pode ser perigoso visto que a empresa pode conseguir reduzir o número de vítimas de *phishing*, mas basta um colaborador carregar num URL malicioso para o ataque ser bem sucedido. No entanto, a formação aliada a outros meios tecnológicos para mitigar os ataques de *phishing* deverá ser uma opção vantajosa.

2.2.4 *Security Keys*

Desde 2017 que a Google tem estado a testar uma nova forma de manter a segurança das contas dos seus 8500 colaboradores. A empresa revelou que a medida tem sido altamente bem sucedida, visto que nenhum dos seus colaboradores foi vítima de *phishing*, especificamente de *account takeover*, desde que esta foi adotada. A medida aplicada é a utilização de *Security Keys*, estas são chaves físicas, na forma de uma pen USB que guarda informação das credenciais de cada *user*, autenticando-o. Para isto acontecer é apenas necessário ter uma *Security Key* e um *website* que suporte a utilização da mesma [24]. Claro que isto pressupõe que tenha de haver excelente segurança física desta chave USB, mas segundo a empresa os resultados são promissores.

2.2.5 Prevenção e mitigação de ataques de *phishing* na Altice

A Direção de Cibersegurança (DCY) está dividida em diversas áreas, com responsabilidades diferentes mas interligadas. O combate ao *phishing* exige diversos tipos de processos, muito distintos, para conseguir não só proteger mas também educar os colaboradores e mitigar estes ataques.

Para abordar as várias vertentes do combate ao *phishing* na Altice Portugal serão feitas três divisões principais: tecnologia, pessoas e processos. Utilizar uma abordagem por camadas significa ter vários controlos de segurança para proteger pontos de entrada vulneráveis diferentes. Para a cibersegurança ser eficaz é necessário não só ter boas ferramentas para proteção mas também sa-

ber gerir as pessoas e os processos. Estas três áreas são bastante complexas e muitas vezes difíceis de relacionar. Alinhar estas áreas, ajuda a unificar as diferentes vertentes da cibersegurança, prevenindo um potencial ataque.

Tecnologia

Dentro da Altice Portugal a prevenção de ataques de *phishing* é feita, na vertente tecnológica, através das ferramentas Anubis e Zscaler. O Anubis (Secção 3.2) é uma ferramenta *anti-spam/ phishing* que bloqueia emails potencialmente maliciosos. Os emails maliciosos são colocados inicialmente em quarentena, passados 40 dias são removidos permanentemente. Caso seja encontrado algum falso positivo é possível retirar de quarentena e devolver à caixa de correio do colaborador. De momento a versão 6.0.10 está operacional, mas em breve a Altice Portugal vai atualizar para a versão 7.0.1. As principais diferenças na nova versão são:

- Implementação do Mail Protection Service (MPS), uma solução de proteção de email para empresas oferecida tanto na nuvem como localmente, com atualização dos seus componentes;
- Nova implementação do filtro *blacklist*, com ferramentas de simulação para testar e modificar filtros.

Com a necessidade empresarial de mover aplicações para a nuvem, é útil ter uma ferramenta que ofereça segurança nesta transição. A ferramenta Zscaler permite uma experiência de navegação na *Internet* segura, servindo como primeira barreira entre os utilizadores e a mesma. A ferramenta liga os seus utilizadores às aplicações, independentemente do aparelho ou localização, de forma segura. Quando um colaborador entra no motor de pesquisa tem de fornecer o seu *login* para poder conseguir aceder a uma página *web*. De forma a manter um histórico, para análise e estatística os *logs* do Zscaler são guardados numa base de dados.

Pessoas

A vertente tecnológica deve proteger ao máximo o serviço de email da empresa, mas infelizmente esta nem sempre é suficiente. Os colaboradores são a última barreira entre o *malware* e as máquinas da empresa, por este motivo é necessário investir na formação dos mesmos. Num cenário ideal ninguém se deixa enganar pelo email malicioso, contudo isto não é o que se verifica na realidade.

De forma a educar os colaboradores, com o intuito de minimizar o número de vítimas de *phishing*, a Altice Portugal tem formação obrigatória nesta área. Esta formação está englobada na área de *Awareness* da DCY. Esta área engloba todas as ações de sensibilização e formação da empresa. *Awareness* é um termo Inglês que significa um conhecimento ou perceção de uma situação ou facto. Tal como o nome indica o objetivo principal é conseguir aumentar a perceção dos colaboradores para as boas práticas de segurança. As atividades promovidas nesta área são variadas, contendo formações, *pop-ups* de alerta e testes controlados de *phishing*.

A destacar dentro desta área é a iniciativa de ataques simulados de *phishing*. Esta iniciativa da DCY consiste da realização periódica de ataques simulados fornecidos pela empresa CybeReady. Após a realização dos ataques a empresa compila estatísticas sobre o número de colaboradores que foram vítimas dos ataques falsos. Para além das estatísticas é fornecida uma pequena formação imediata. Sempre que um colaborador carrega numa hiperligação deste email é indicado que este era um ataque simulado e que o colaborador não deverá interagir com este tipo de email. Através destas simulações periódicas e dos seus resultados é possível avaliar se os colaboradores vão aprendendo e sendo cada vez menos vítimas destes ataques. Os resultados têm sido positivos visto que a percentagem de colaboradores que seguiram os URLs tem vindo a diminuir desde que a campanha foi iniciada.

Processos

A maneira como os processos são executados na Altice Portugal é fundamental para não permitir situações em que a segurança da empresa seja comprometida e para dar resposta quando ocorrem incidentes de segurança. Os processos devem ser sempre cumpridos, garantindo assim que tudo o que seja fora do esperado possa ser considerado suspeito. Se as comunicações entre a chefia e os colaboradores estiver padronizada, bem como o tipo de tarefas que devem ser realizadas por email, sempre que houver um pedido anormal, os colaboradores terão mais confiança que se trata de um pedido ilegítimo.

Em termos de suspeitas de *phishing* o procedimento geral para o colaboradores é o seguinte: caso o colaborador receba algum email suspeito de *phishing*, deverá fazer *foward* do mesmo para o endereço específico de *phishing* da Altice. De momento a equipa que trata destes emails tem a tarefa de os analisar posteriormente. Muitas vezes os colaboradores não se restringem ao *phishing* recebido no email profissional e enviam *phishing* do pessoal ou até mesmo de familiares, dificultando a tarefa da equipa que analisa estes *reports*.

Os processos de resposta a incidentes também devem estar muito bem definidos. Infelizmente, por incumprimento dos processos definidos, falta de atenção ou erro, acabam por haver situações de infeção, isto é, situações em que o ataque é bem sucedido e consegue recolher dados do colaborador ou infetar uma máquina. Para resolver estas situações é necessário que estejam montados processos eficazes na equipa Computer Security Incident Response Team (CSIRT). Na Altice esta é a equipa de resposta a incidentes de segurança da informação, que está integrada no Security Operations Center (SOC). O CSIRT trata e coordena os incidentes de segurança informática de acordo com a classificação definida na Política de Classificação de Incidentes, disponível na Rede Nacional de CSIRT, rede em que a equipa da Altice está enquadrada. Para além disto a disseminação de alertas e recomendações de segurança, através da informação recebida dos sistemas de monitorização de segurança da Altice, é responsabilidade desta equipa.

2.3 Robotic Process Automation (RPA)

Os processos de automatização são cada vez mais frequentes dentro do mundo empresarial, a ideia de que é possível, com relativa facilidade, automatizar um conjunto de tarefas rotineiras é bastante apelativa. Todos os dias os colaboradores de uma empresa têm de tratar de inúmeros processos considerados repetitivos e pouco interessantes. Estes aumentam a fadiga e cansaço mental do colaborador, o que acaba por levar a situações de erro ou de procrastinação das tarefas em questão.

Apesar da automatização estar presente no mercado de trabalho há bastante tempo, a aplicação de RPA nas empresas é relativamente recente. É importante definir e compreender bem este conceito de forma a explorar as suas capacidades ao máximo e ter noção das suas limitações.

Pelo IEEE temos a seguinte definição de RPA *"preconfigured software instance that uses business rules and predefined activity choreography to complete the autonomous execution of a combination of processes, activities, transactions, and tasks in one or more unrelated software systems to deliver a result or service with human exception management."*

Por Tornbohm, 2017 temos *"RPA tools perform [if, then, else] statements on structured data, typically using a combination of user interface interactions, or by connecting to APIs to drive client servers, mainframes or HTML code. An RPA tool operates by mapping a process in the RPA tool language for the software robot to follow, with runtime allocated to execute the script by a control dashboard."* (Tornbohm, 2017).

Simplificando estas definições podemos dizer que as ferramentas de RPA utilizam regras, definidas pelo programador, de forma a executar processos que interagem com vários sistemas, como um humano faria. Desta forma os processos repetitivos baseados em regras são os mais adequados para automatizar. Considerando por exemplo um relatório que tem de ser escrito todos os dias, com informação de uma tabela que se encontra numa página *web*, este processo seria um excelente candidato a ser automatizado.

Visto que esta área está a ganhar muito interesse rapidamente, é normal que em muitos casos as empresas não compreendam completamente que tipo de processos podem ser automatizados. Isto pode fazer com que hajam expectativas pouco realistas em relação àquilo que o RPA pode fazer. RPA deve ser utilizado para tratar dos casos mais frequentes e repetitivos, isto corresponde a 20% dos casos, os restantes são demasiados complexos ou infrequentes para serem bons candidatos a automatização. Alguns fornecedores de RPA, como o Blue Prism (BP) ajudam a perceber qual o grau de automatização de um processo, isto é, quão bom candidato é o processo em questão para automatizar [4].

Por outro lado, o investimento na automatização de processos, feita da maneira correta, apresenta bastantes vantagens. A facilidade de uso de RPA é uma das suas características mais apelativas, visto que não é necessário aprender uma linguagem de programação, embora seja vantajoso ter algum *background* nesta área. A maior parte das ferramentas de RPA apresentam uma interface simples e acessível, de forma a facilitar o uso a diferentes tipos de utilizadores da ferramenta. As ferramentas não necessitam de ser integradas, isto é, funcionam sobre os sistemas existentes, atuando sobre os mesmos como um ser humano. Este é um ponto a favor destes sistemas, visto que há

outras formas de automação que requerem que o sistema seja todo modificado para as acomodar.

Temos então o RPA como uma tecnologia emergente e atual, que embora apresente alguns desafios, oferece também inúmeras vantagens sobre outros processos de automação. A utilização deste tipo de tecnologia pode aumentar bastante a produtividade de uma empresa, em particular em processos rotineiros de gestão.

2.4 *Crowdsourcing*

O *Crowdsourcing* é um modelo que tira partido das opiniões de grupo (*crowd*) para tomar decisões. Pode ser descrito como um *outsourcing* [5] para multidões. O *Crowdsourcing* pode ser aplicado a vários setores, desde ciências, à educação, passando pelas tecnologias e ciências sociais. Em qualquer setor podemos ter vantagem na consulta da opinião de um grupo de pessoas, visto que podemos democratizar a solução de problemas e acelerar a inovação das soluções aos mesmos. Este modelo pretende conseguir uma inovação da maneira de pensar dentro de um projeto, onde antes pequenas amostras da população eram consultadas ou apenas os responsáveis dos projetos sugeriam novas ideias, agora temos todo um novo conjunto de pessoas e ideias a explorar. É um desafio aos modelos de negócio e organizações tradicionais, oferecendo grandes capacidades de reinvenção de processos convencionais. Um dos exemplos mais populares de *crowdsourcing* é a Wikipedia[6], o seu conceito serve para exemplificar como podemos recorrer a este método para conseguir obter resultados. Neste caso temos uma plataforma de informação em que vários utilizadores podem acrescentar ou modificar conteúdo. Embora um utilizador normal possa editar bastantes artigos, a partir de um determinado número de dias e edições os privilégios do utilizador são aumentados automaticamente. Desta forma é possível aceder à criação de novas páginas, edição de mais artigos e carregamento de ficheiros [33].

[5] Ato de obter (bens ou serviços) de um fornecedor externo, em vez de uma fonte interna

[6] https://www.wikipedia.org/

Capítulo 3

Ferramentas

Este capítulo apresenta as ferramentas fundamentais utilizadas no âmbito do sistema *RoboPHISH*. Embora a Altice Portugal disponha de inúmeras ferramentas, neste capítulo serão apenas focadas as que têm interesse no contexto do *RoboPHISH*. As ferramentas utilizadas apenas marginalmente ou para extrair alguma informação mas que não correspondam a interações do sistema não serão apresentadas no capítulo.

3.1 Blue Prism

O Blue Prism (BP) é uma ferramenta de RPA confiável e segura com possibilidade de implementação local ou na nuvem. O BP não necessita de integração, visto que manipula outro *software* através da interface já existente.

A ideia do BP é criar uma força de trabalho virtual, constituída por robôs de *software*. Estes robôs permitem a automatização de processos administrativos de *backend*, baseados em regras, que tendem a ser repetitivos e demorados. Com esta força de trabalho é possível reduzir o custo destes processos, tanto monetário como temporal.

Para utilizar o BP não é necessário ter conhecimentos em nenhuma linguagem de programação em particular. Os colaboradores, dado algum tempo e treino, conseguem compreender facilmente a lógica dos conceitos da ferramenta e dentro de uma curva de aprendizagem relativamente curta, realizar processos.

3.1.1 Componentes

Em termos da interface gráfica do BP temos seis separadores principais: *Home*, *Studio*, *Control*, *Analytics*, *Releases* e *System*.

No separador *Home* (Figura 3.1) temos um conjunto de estatísticas sobre a disponibilidade, número total de automatizações e maiores tabelas (em termos de espaço ocupado) na base de dados. Dentro do *Studio* temos uma listagem de todos os Processos e Objetos, que normalmente estão divididos por pastas, de acordo com a sua função. Neste separador podemos criar, editar e apagar processos e objetos. No separador *Control* temos uma visão geral dos processos já publicados, as máquinas (recursos) onde os podemos executar e as filas de trabalho, que contêm os dados para os mesmos. É neste separador que são executados os processos, através da atribuição

de um processo a uma determinada máquina. Dentro do *Analytics* temos o mesmo conjunto de estatísticas que aparecem no separador inicial (*Home*), aqui podemos criar e editar as mesmas. O separador *Releases* funciona como um sistema de controlo de versões. Uma *release* no fundo é um *save* do projeto, no estado em que está naquele momento, sendo possível guardar objetos, variáveis de sistema, processos, tudo o que for necessário para o projeto ser executado. Por fim temos o separador *System*, onde são alteradas as definições do sistema, é aqui que podemos criar, editar e apagar filas de trabalho, *users* e variáveis de sistema.

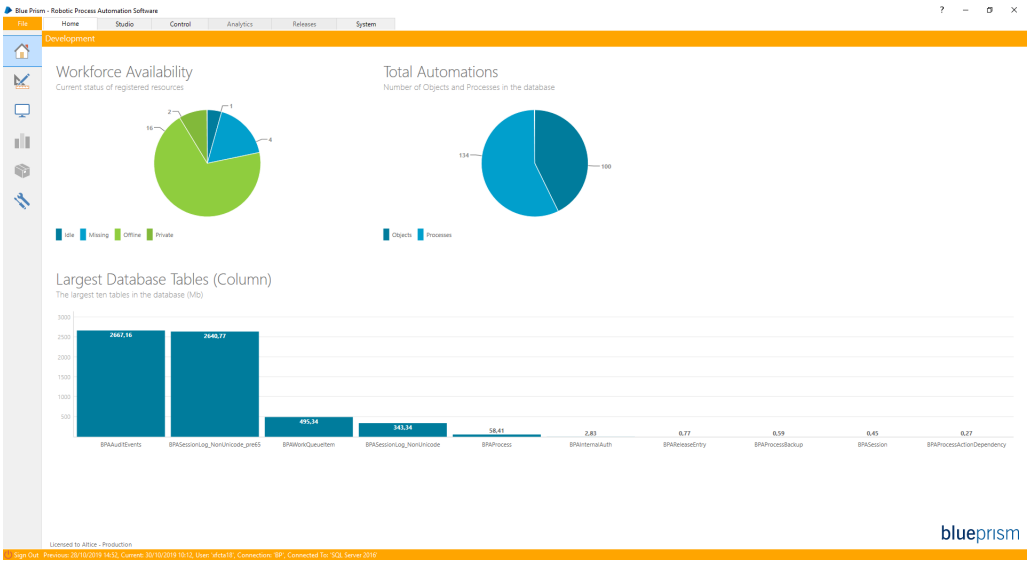


Figura 3.1: Home page do BP

3.1.2 Processos

Os processos no BP são criados no *Process Studio*, a partir do separador *Studio*. Ao criarmos um processo temos uma página quadriculada onde podemos arrastar e largar diferentes *tools*, que correspondem a pedaços de código, de forma a construir um fluxograma que representa uma parte de um caso de uso. A maneira como modelamos os processos é semelhante a um diagrama de fluxo, temos formas geométricas que simbolizam ações (objetos), variáveis, ciclos e condições, ligadas entre si (Figura 3.2).

Existe uma *tool* para fazer a ligação a outras *pages* do mesmo processo. Dividindo os processos em diferentes *pages* temos código que é mais fácil de manter e compreender. Cada *page* deve corresponder a uma divisão coerente do processo.

É boa prática construir o diagrama *top down*, isto é, o processo deve seguir uma ordem lógica de passos de cima para baixo, assim qualquer pessoa consegue entender o que está a ser feito no processo. As exceções (ações que interrompem o programa e lançam um erro) devem ser colocadas à direita do diagrama. Também há um conjunto de cores diferentes para as variáveis globais, locais ou de sistema. Para além do mencionado anteriormente, as boas práticas comuns da programação também devem ser cumpridas, tais como as que se referem à modularização, nomes significativos de variáveis e documentação do código.

3.1.3 *Queues*

As *queues* são filas de trabalho do BP que servem para guardar os *items* a processar. Os *items* são guardados nas filas de trabalho, podendo ser utilizados por diferentes processos. O objetivo das *queues* é não só guardar o trabalho até um determinado ponto, no caso de ocorrer algum erro, mas também dividir o código por diferentes processos que vão trabalhar sobre as mesmas *queues*. A ideia é termos um *item* da *queue* que vai passando por vários processos e atualizando o seu estado conforme estas atualizações forem completas com sucesso. Todas as *queues* de trabalho têm os seguintes campos:

- *Item Key*: Corresponde ao identificador único da queue em questão;
- *Priority*: Representa a prioridade com que deve ser tratado cada *item*, caso não esteja preenchido têm todos a mesma prioridade;
- *Status*: Campo onde é guardado o estado do *item* que vai sendo atualizado ao longo dos processos;
- *Tags*: Campo onde podemos guardar uma ou mais *keywords* ou pequenas informações sobre o *item*;
- *Resource*: Representa o recurso (máquina) que tratou do *item* em questão;
- *Attempt*: Número de tentativas necessárias para completar com sucesso o *item*;
- *Created*: Corresponde à data de adição do *item* à *queue*;
- *Last Updated*: Corresponde à data mais recente em que o *item* foi atualizado;
- *Completed*: Corresponde à data em que *item* foi dado como completo;
- *Total Work Time*: Representa o tempo total que o *item* demorou a ser processado;
- *Exception*: Campo que representa se um *item* foi mal processado, isto é, caso tenha ocorrido algum erro de processamento;
- *Exception Reason*: Campo que contém o erro em particular que ocorreu no *item*. Só é preenchido caso o *Exception* seja preenchido.

Na Figura 3.3 temos o exemplo dos campos supramencionados de uma das *queues* do BP. Para além destes campos é possível definir campos específicos, no processo, para os *items* de uma determinada *queue*. Estes campos novos devem conter a informação específica de cada item, funcionando mais ou menos como uma tabela.

Existem quatro categorias fundamentais no que toca ao processamento dos *items* da *queue*: *pending*, *locked*, *completed* e *exception*. Quando um *item* entra pela primeira vez na *queue*, a sua categoria é *pending*. Conforme o *item* específico é requisitado por um processo este fica *locked* até ser dada ordem de o libertar. O *item* pode ser libertado e retornado ao estado de *pending*, pode ser dado como *completed* (completo) ou pode ser marcado como *exception* (exceção). Um *item* só

pode estar numa categoria, isto é, se foi *completed* quer dizer que foi executado com sucesso, se foi *exception* ocorreu algum erro.

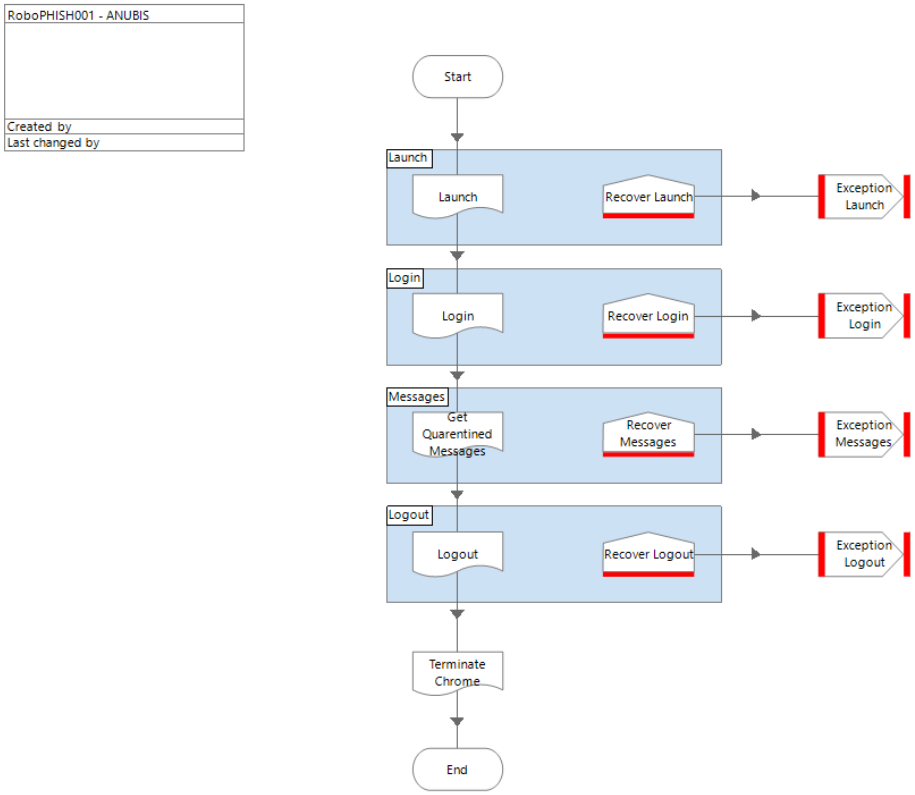


Figura 3.2: Exemplo de uma *main page* de um processo

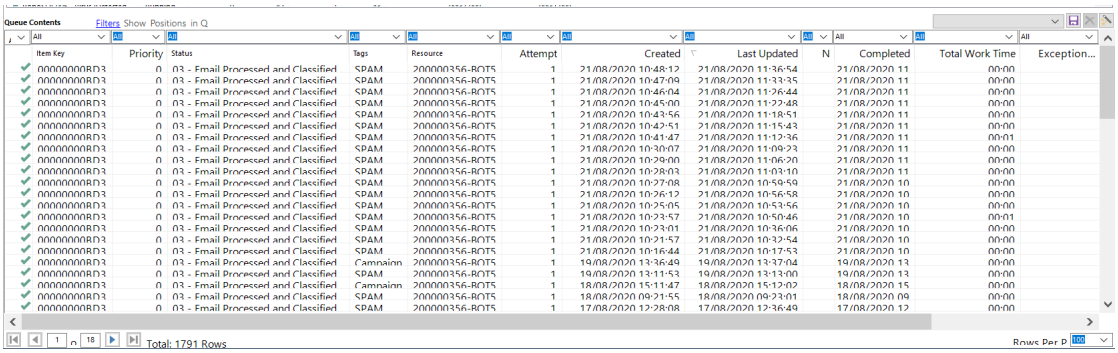


Figura 3.3: Exemplo de uma *queue* do BP

3.1.4 Objetos

Tal como nos processos, os objetos também são diagramas de fluxo, onde temos um conjunto de ferramentas representadas por formas geométricas. Todas as coisas que podemos fazer num processo podemos fazer num objeto mas os objetos têm *tools* que não existem nos processos. Os objetos são feitos especificamente para interação com aplicações, isto é, têm ferramentas especiais para conseguir identificar elementos nas diferentes aplicações.

Cada objeto tem de ser associado a uma aplicação específica, que tem de ser definida na criação do mesmo, podendo ser alterada mais tarde. Podemos também passar logo certos parâmetros a determinadas aplicações, por exemplo, uma página *web* ao Google Chrome ou ao Internet Explorer. Sempre que a aplicação é lançada é aberta na página referente ao URL definido.

Após a definição da aplicação vamos definindo os objetos com os quais queremos interagir na mesma. Por exemplo, uma pesquisa simples num *browser* requer identificar a *search bar* e o botão de pesquisa. Quando já temos estes elementos podemos desenvolver a lógica do objeto à volta dos mesmos, com as ferramentas disponíveis.

Quando temos os objetos criados, podemos publicá-los e usá-los em qualquer processo. A ideia de separar os objetos dos processos também facilita a reutilização dos mesmos. Podemos ter um processo que trabalha com o Notepad, que é utilizado por vários processos. Para além dos objetos criados pelos *developers* também temos alguns que já existem no BP, para interagir com o Excel, Outlook e File System por exemplo.

3.2 AnubisNetworks

A AnubisNetworks [3] fornece soluções para combater ameaças cibernéticas em tempo real. O serviço de proteção de email foi projetado com o objetivo de detetar fraude, *malware* e SPAM, com grande capacidade de controlo de entrega e das funcionalidades de *routing*. Estas plataformas de segurança podem ser utilizadas na nuvem ou em máquinas locais, permitindo às empresas escolher qual a opção mais adequada para si.

Na interface *web* da ferramenta Anubis existem seis separadores principais *Dashboard*, *Messages*, *Mail Filtering*, *Mail Control*, *Domain Settings* e *Hierarchy*. No *Dashboard* temos as estatísticas, representadas em diferentes gráficos, dos emails recebidos, bloqueados e em quarentena, é possível filtrar por período de tempo (Figura 3.4). O separador mais utilizado no contexto deste projeto é o *Messages*, onde podemos encontrar listas com as várias mensagens que são descartadas ou postas em quarentena. Podemos aplicar diferentes filtros para selecionar as mensagens que queremos visualizar. A lista obtida contém a data da mensagem, remetente, destinatário, assunto, *report*, ação e informação sobre a que hierarquia pertence a mensagem. O corpo da mensagem em si não está presente nesta lista. Temos ainda uma lista de cores que nos permite ver em que categoria estão estas mensagens, que podem ser: *Clean*, *Spam*, *HighSpam*, *Infected* e *Policy Blocked*.

3.2.1 Mailspike

A Mailspike, empresa-mãe da AunubisNewtorks, fornece uma lista gratuita de bloqueios de IP. Através do endereço *web http://mailspike.org/iplookup.html* podemos verificar a reputação de um determinado IP e podemos também solicitar a exclusão de IPs.

Os dados de reputação dos IPs têm um *score* associado, que é calculado através das características e comportamento dos endereços de IP que enviam emails de entrega direta (*direct-to-MX email*[1]). Com base no *score*, existem 11 categorias de reputação de IPs, desde *Worst possible re-*

[1] *Direct-to-MX* ignora o servidor SMTP do remetente e entrega o email diretamente ao servidor de mensagens do destinatário (MX) [6].

putation até *Excellent Reputation*. Também há dados imediatos, que não dependem da reputação anterior de um IP, estes são obtidos através da deteção de comportamentos virais de um ou mais IPs [2].

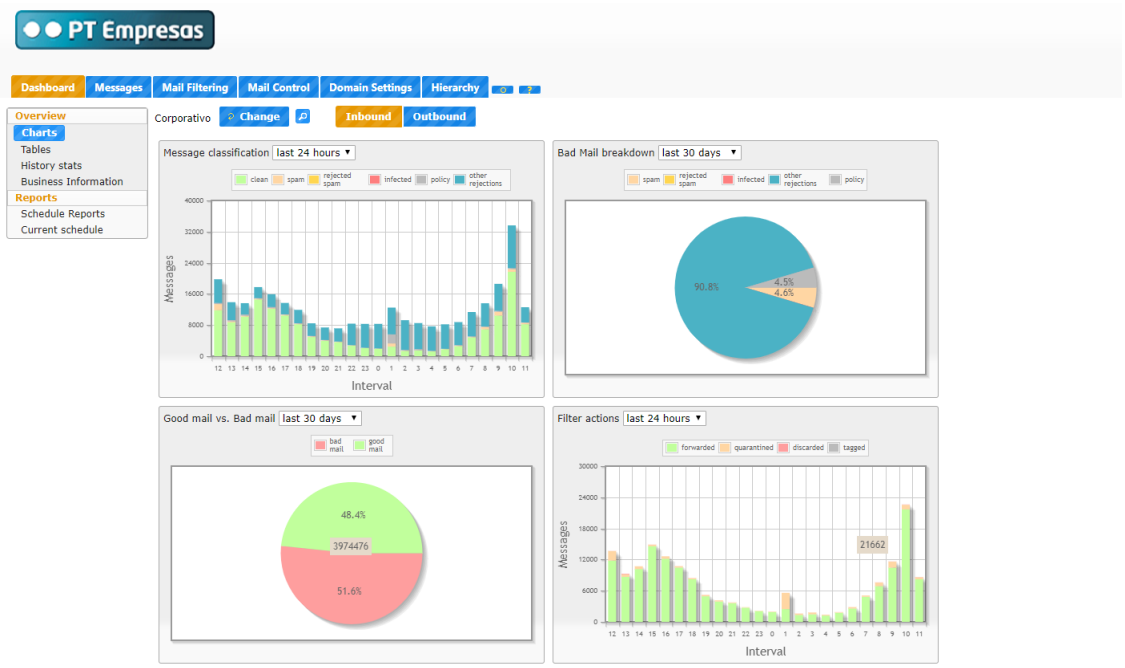


Figura 3.4: Home page do Anubis

3.3 *HP Service Manager* (HPSM)

O *HP Service Manager* (HPSM) é uma aplicação da HP[2] que serve para criar, apagar e gerir *tickets* que correspondem a denúncias de incidentes. Através desta aplicação é possível que os membros das equipas de resolução de incidentes possam tomar medidas diretas sobre *tickets* abertos na aplicação.

É possível aceder ao sistema de gestão de *tickets* através da interface *web* do HPSM. Na Altice Portugal só algumas contas têm acesso a esta interface, que corresponde à parte de gestão dos *tickets* e onde estão disponíveis todas as funcionalidades. Caso um colaborador queira abrir um *ticket* terá de o fazer através de uma plataforma interna da Altice Portugal chamada ONEDESK. Esta plataforma permite a abertura (indiretamente) dos *tickets* no HPSM.

Para abrir um *ticket* é possível selecionar um *template* existente adequado ao incidente em questão. Quando um *ticket* é aberto no HPSM este é atribuído a um elemento da equipa SOC para resolução do mesmo. A partir do momento em que o SOC resolve o incidente é então enviado um email para o colaborador com o intuito de avaliar a sua satisfação com a resolução do problema.

[2]https://www8.hp.com/pt/pt/home.html

3.4 Elastic Search

O Elastic Search é um motor de análise e pesquisa *RESTful* [3] distribuído, altamente escalável. Atualmente é um dos motores mais populares, devido à sua capacidade de lidar com um número crescente de casos de uso, guardando e analisando grandes volumes de dados rapidamente e em tempo real.

O Elastic Search é rápido, escalável, relevante e resiliente. É uma boa opção para empresas que necessitam de fazer análise, em tempo real, de um grande volume de dados. A sua flexibilidade na maneira como permite a inserção de dados é também um ponto a favor. O ambiente distribuído em que a ferramenta opera permite o processamento de dados em paralelo, diminuindo assim o tempo de pesquisa. É utilizado frequentemente para análise de *logs*, visto que é possível fazer uma pesquisa rápida e centralizada dos mesmos.

A interface do Elastic é simples e relativamente fácil de navegar, facilitando assim o uso inicial da ferramenta. É possível fazer análises quase em tempo real, é oferecida integração com várias plataformas (Kibana, Beats, Logstash) e suporte para várias linguagens (Java, Python, Node.js).

O Elastic Search vem integrado com o Kibana, uma ferramenta de visualização e *report*. O Kibana permite a visualização dos dados do Elastic Search, fazendo queries sobre os mesmos e selecionando a forma mais adequada de os visualizar (gráficos, tabelas). Com o Kibana é possível pesquisar entre vários documentos, com a sua visualização interativa é possível ir selecionando diferentes elementos e chegar a diferentes *views*. A possibilidade de explorar um módulo de *machine learning* também está presente na ferramenta.

3.5 VirusTotal

O VirusTotal é um serviço *web*, atualmente pertencente à Google, que analisa hiperligações e ficheiros com o objetivo de detetar presença de *malware* nos mesmos. O VirusTotal funciona agregando os resultados de diferentes antivírus, *websites*, ferramentas de análise de domínios e contribuições de utilizadores para devolver um parecer sobre o nível de segurança de um domínio ou ficheiro. Os serviços do VirusTotal são gratuitos e beneficiam tanto os utilizadores, que podem navegar de forma mais segura, como os antivírus que têm assim uma forma de fazer publicidade ao seu produto. O VirusTotal conta com mais de 70 antivírus e *blacklists* de domínios, que servem para analisar os ficheiros/ hiperligações inseridos pelos utilizadores. Para além da utilização da interface *web* é possível submeter ficheiros através de uma *desktop app*, extensões do navegador e *Application Programming Interface* (API)[4] [31].

Quando um utilizador do VirusTotal submete uma hiperligação ou ficheiro os resultados são enviados tanto para o próprio como para os antivírus parceiros do VirusTotal, desta forma os próprios utilizadores ajudam a melhorar o sistema.

[3] REST significa *Representational State Transfer* e representa um princípio de arquitetura para serviços *web*

[4] Conjunto de funções e procedimentos que permitem a criação de aplicações que acedem aos recursos ou dados de um sistema operacional, aplicação ou outro serviço

3.6 Microsoft Exchange Email (Outlook)

O Microsoft Outlook é uma aplicação da Microsoft que serve como cliente de email, calendário, gestor de contactos e outras funções de gestão de informação pessoal. Esta aplicação está integrada no conjunto de aplicações do Microsoft Office. Num contexto empresarial é utilizado como um *software* através do Microsoft Exchange Server.

O Microsoft Exchange Server é o servidor de email da Microsoft, isto é, é o servidor que gere e administra os emails e todas as outras funcionalidades do Outlook. Os emails que são recebidos, enviados, calendários, tarefas e notas são todos geridos e guardados no servidor da Microsoft. Utilizar esta solução de email apresenta vantagens para as empresas, visto que os utilizadores têm os seus emails guardados e centralizados num servidor, em vez de localmente. A partilha de calendários entre os colaboradores também pode ser bastante útil para gerir e marcar reuniões.

Uma outra vantagem do Exchange Server, particularmente pertinente a este projeto, é a capacidade de filtragem de emails do servidor. O Exchange Server protege ativamente as comunicações dos seus utilizadores, com vários filtros *anti-spam* e *anti-phishing*, possuindo também vários motores de deteção de *malware*, garantindo assim alguma proteção dos emails dos utilizadores.

Dentro da Altice Portugal o serviço de email utilizado é o da Microsoft - Microsoft Exchange Email. Os computadores dos colaboradores da empresa têm o Outlook instalado (Oulook para MAC, Outlook 2009, Outlook 2013 ou Outlook 2016) e quem quiser também pode utilizar a interface *web* (Office 365).

A proteção conferida pelo servidor da Microsoft, embora útil, muitas vezes é insuficiente para combater as ameaças aos emails dos colaboradores da Altice Portugal. Assim, é necessário associar esta proteção a outras plataformas e aplicações, para conseguir combater os ataques mais persistentes.

Capítulo 4

Descrição do Sistema *RoboPHISH*

Nos capítulos anteriores são abordados os processos mais eficazes de combate ao *phishing* mundialmente e os processos atualmente implementados na Altice Portugal. Através desta informação é possível propor uma solução para o problema do *phishing* nesta empresa. A solução proposta não pretende melhorar todos os processos existentes mas sim automatizar um processo em particular, o de análise e de *report* de *phishing*, de forma a mitigar parcialmente este problema.

O sistema *RoboPHISH* está inserido na nova iniciativa da Altice Portugal, *PhishFighting*, uma iniciativa criada especialmente para ajudar a combater e mitigar os ataques de *phishing* na empresa.

Neste capítulo será descrita a arquitetura do BP dentro da DCY, bem como a arquitetura e detalhe dos requisitos do sistema desenvolvido, *RoboPHISH*. O capítulo inclui também a esquematização dos processos, referindo a finalidade e os pontos fundamentais dos mesmos.

4.1 Arquitetura BP DCY

O sistema *RoboPHISH* está inserido num sistema maior de robôs de *software* BP (*bots*) já existente no início do trabalho (Figura 4.1). Os *bots* estão todos localizados somente num único servidor, na Covilhã, sendo que cada um deles é uma virtualização (VMWare [1]). Cada *bot* é acedido por *Remote Desktop Connection*[2], que executa diferentes processos em BP.

Na Altice existem sete *bots* numerados do zero ao cinco, embora apenas cinco executem processos (Tabela 4.1). O *bot* 0 serve de *Control Room*, isto é, coordena todos os outros. Sempre que é necessário iniciar ou parar a execução de algum processo é possível entrar no *bot* 0 e através da aplicação BP iniciar ou parar a execução de um dos outros *bots*. O sétimo *bot* corresponde à base de dados onde é guardada toda a informação relacionada com o BP, desde tabelas pré-existentes, à informação da instalação da ferramenta até aos *logs*(registos) dos processos.

Cada *bot* pode estar responsável por vários processos porém só pode ser executado um de cada vez, ou seja, todos os processos devem estar agendados de modo a que os seus horários não se sobreponham. Desta forma os *bots* que têm mais que um processo ou vários sub-processos (Tabela 4.1) executam-nos em horários diferentes. O agendamento dos processos pode ser feito por dia,

[1] Empresa que fornece *software* e serviços de virtualização e computação na nuvem.

[2] Tecnologia da Microsoft que permite que um computador local se conecte e controle um PC remoto numa rede ou na *Internet*.

hora, minuto, semana, mês ou ano. Embora o agendamento possa dar a entender que alguns *bots* estão mais sobrecarregados que outros (*i.e. bot* 3) isto nem sempre é verdade, visto que foram definidas algumas políticas de organização dos processos por *bot* na empresa. Por exemplo, o *bot* 3 só executa processos que correm uma vez por dia, todos os outros têm agendamentos de horas e minutos.

A nomenclatura dos processos da Altice Portugal segue sempre um conjunto bem definido de regras visto que facilita a leitura, organização e compreensão dos nomes dos processos. O processo deve sempre começar por RoboX, sendo que o X é o nome da categoria a que pertence. Neste momento existem seis categorias (*SOC*, *IAM*, *CSVMS*, *PHISH*, *CISO* e *ALL*) relacionadas com a função de cada processo.

Na nomenclatura dos processos, após RoboX, temos três números que correspondem ao número do processo em questão. A numeração está por ordem cronológica de quando foram desenvolvidos os processos. Pode acontecer que ao longo do tempo tenham sido apagados processos e desta forma a numeração poderá saltar alguns valores.

Em cada processo após a primeira palavra, que indica a categoria e número do processo, temos uma breve descrição do processo em questão. Dentro de cada um destes processos podemos ter vários sub-processos. A nomenclatura para estes sub-processos consiste do primeiro nome igual ao processo-pai, seguido do número de sub-processo e finalizando com uma breve descrição daquela parte concreta do processo (exemplo: *RoboPHISH001 - 01 - Save Emails to Folder and Queue*).

Os nomes de processo apresentados na Tabela 4.1 tratam-se dos nomes dos agendamentos dos processos, que podem conter vários processos. Por exemplo o agendamento *RoboPHISH001 - Get and Classify Emails* contém os processos *RoboPHISH001 - 01 - Save Emails to Folder and Queue* e *RoboPHISH001 - 02 - Analyse Emails from Queue*.

Processo	BOT 1	BOT 2	BOT 3	BOT 4	BOT 5
RoboSOC000 - Feed All Queues		X			
RoboSOC001 - Daily Reports from Arcsight			X		
RoboSOC002 - Notify Customers of DDoS ...		X			
RoboSOC003 - Zscaler Threat Reports	X				
RoboSOC004 - Anubis Outbound SPAM			X		
RoboSOC009 - ALLOT DDoS Alerts			X		
RoboSOC010 - DigiCert Domain Validation			X		
RoboIAM002 - Weekly Reports from SAPRH		X			
RoboIAM007 - ELEARNING Reports			X		
RoboCSVMS - Cycognito			X		
RoboPHISH001 - Get and Classify Emails					X
RoboPHISH001 - Get Unclassified Emails					X
RoboCISO002 - Daily Alerts			X		
RoboCISO002 - Hourly Alerts				X	
RoboCISO001 - Critical Patching			X		
RoboALL000 - Log Queues into KIBANA			X		

Tabela 4.1: Distribuição de processos por *bot*

Os cinco *bots* existentes na empresa comunicam com o servidor aplicacional e com as diferentes aplicações, externas e internas. No caso particular do *RoboPHISH* as aplicações principais são:

- Base de dados SQL, ligada à ferramenta Tableau[3] que permite trabalhar sobre os dados em questão, extraindo estatísticas relevantes sobre o processo;
- HPSM, ferramenta de *tickets* para comunicação com SOC;
- Anubis (Secção 3.2) e Zscaler (Secção 2.2.5) para bloquear os endereços de email e URLs maliciosos;
- Google Chrome, em particular VirusTotal (Secção 3.5), para ajudar a classificar os emails;
- Outlook (Secção 3.6), aplicação de email da Altice Portugal, utilizada para a comunicação entre robô e colaborador.

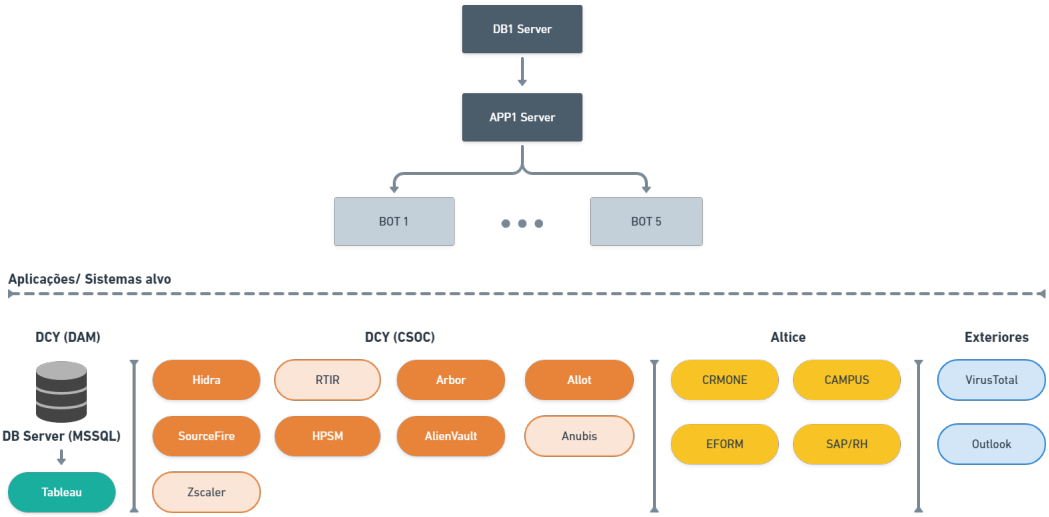


Figura 4.1: Arquitetura BP DCY

4.2 Arquitetura *RoboPHISH*

O sistema *RoboPHISH* está dividido em vários processos que tratam da interação com diversas ferramentas (Figura 4.2). Vários processos podem ser executados no mesmo *bot*, no caso particular do *RoboPHISH* são três processos a serem executados no *bot* cinco.

Quando um email é enviado para um endereço interno da Altice Portugal este passa primeiro pela plataforma anti-spam Anubis (Secção 3.2), seguindo para o Exchange (Outlook) e finalmente chegando ao colaborador. Caso chegue em email malicioso ao colaborador, isto significa que o email passou por estas três plataformas e não foi identificado como *phishing* ou SPAM por nenhuma. Quando o colaborador interage com as hiperligações do email, assumindo que está na rede interna da Altice Portugal, tem de passar pelo Zscaler (Secção 2.2.5) para aceder à *Internet*.

[3]https://www.tableau.com/

Temos assim definidos os pontos de passagem do email malicioso até chegar ao colaborador, bem como os pontos de passagem das interações do colaborador com o email (Figura 4.2).

O sistema pretende atuar sobre estes pontos de passagem, impedindo desta forma o ataque em questão e prevenindo futuros ataques. O botão de denúncia de SPAM/ *phishing* é colocado no Outlook do colaborador, fazendo assim com que o email passe por mais um ponto: o sistema *RoboPHISH*. Após passar pelo sistema, com um resultado de *phishing*, é necessário bloquear este domínio nas plataformas a montante e bloquear as suas hiperligações nas plataformas a jusante. Foi selecionada a plataforma Anubis para bloquear o email, visto que é a primeira plataforma que o email encontra e é a mais fácil de personalizar. Em relação ao bloqueio das hiperligações será feito pelo Zscaler visto que é a plataforma responsável por esta ação. De qualquer das formas existirá sempre um risco associado, visto que os colaboradores podem aceder às hiperligações fora da rede interna da Altice, mas estes casos estão fora do alcance do sistema. Caso o email seja de SPAM é apenas bloqueado no Anubis. Para além dos bloqueios nas plataformas possíveis e necessárias, o colaborador também será informado, por email, da classificação da sua denúncia. Caso se trate de um email de *phishing* o colaborador deverá ser informado que não poderá interagir com o conteúdo do email.

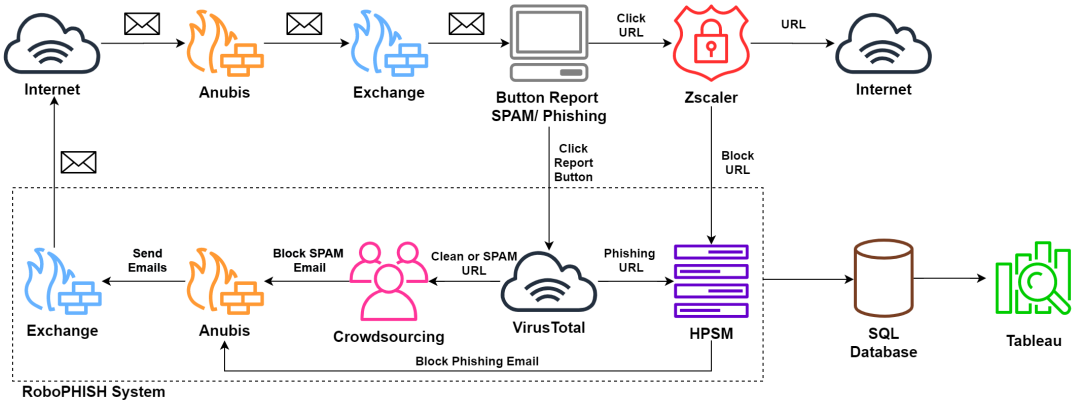


Figura 4.2: Arquitetura *RoboPHISH*

4.3 Esquematização dos processos *RoboPHISH*

Neste sub-capítulo, são apresentados os pontos principais do processo, que será subdividido em vários sub-processos. Todos os diagramas dos processos são apresentados e explicados neste sub-capítulo.

4.3.1 Botão Outlook

Para facilitar a denúncia de um email suspeito (de SPAM ou *phishing*) por parte de um colaborador, de forma padronizada, foi necessário concretizar um botão no Outlook. Antes da implementação do *RoboPHISH*, quando um colaborador encontrava um email suspeito envia-o para o SOC a expor a situação ou abria um *ticket* para reportar o que pensava ser um email comprometido. Com a adição deste botão no Outlook, o utilizador só necessita de selecionar o email em questão e este é enviado em anexo para a caixa de correio *RoboPHISH*.

A solução do botão *SPAM/ Phishing* teve como requisitos as seguintes funcionalidades:

- Possibilidade de seleção de um ou mais emails para denúncia;
- Opção de escolha do tipo de denúncia, SPAM ou *Phishing*, por parte do colaborador;
- Enviar o(s) email(s) potencialmente malicioso(s) em anexo para a caixa de correio do *RoboPHISH*;
- Mover o(s) email(s) potencialmente malicioso(s) para a pasta *Junk* e adicionar um prefixo ao *subject* para informar o colaborador que o *report* já foi feito;
- Informar o colaborador que a denúncia foi bem sucedida.

4.3.2 Processos RPA

Para além da implementação de um botão de denúncia de emails suspeitos, é necessário programar um sistema (conjunto de processos RPA) para analisar estas denúncias. Toda a análise é feita recorrendo a RPA, nomeadamente BP, em conjunto com algumas operações feitas através do SQL *Server*. De seguida é apresentada uma representação de alto nível do processo geral de análise:

- Aceder à caixa de correio criada para reportar incidentes de *phishing*, e extrair o(s) email(s) potencialmente maliciosos (anexados aos emails de denúncia);
- Analisar os *links* do(s) email(s) reportado(s) pelos colaboradores, consultando os mesmos no VirusTotal (Secção 3.5) para chegar a uma decisão relativamente à classificação "SPAM", "*Phishing*"ou "*Clean*";
- Caso não haja informação no VirusTotal é feita uma avaliação através de um algoritmo de *crowdsourcing* para chegar a uma decisão relativamente à classificação "SPAM", "*Phishing*" ou "*Clean*";
- Caso não seja possível de todo chegar a uma conclusão, enviar o email para os analistas SOC;
- Quando houver confirmação da classificação do email (pelo SOC ou pelo robô) enviar um email ao colaborador com a resposta adequada;
- No caso do email ser considerado *phishing* ou SPAM, aceder à interface *web* de *report* do Anubis (Capítulo 3.2) para bloquear futura correspondência proveniente daquele endereço de email;
- No caso do email ser considerado *Phishing*, aceder à interface *web* de tickets do HPSM para bloquear no Zscaler os *links* provenientes daquele email.

Para demonstrar as interações dos robôs com as aplicações é útil recorrer a algum tipo de diagrama. Este diagrama representa o todo ou parte do processo e ajuda a representar os passos efetuados pelo robô. Para o sistema *RoboPHISH* o tipo de diagrama escolhido foi System Sequence Diagram (SSD).

Um SSD é um subtipo de diagramas de sequência, que faz parte de uma linguagem denominada de United Modeling Language (UML). Esta linguagem serve especificamente para a construção, leitura e interpretação de diagramas. Em particular os diagramas SSD são ferramentas úteis para detalhar como as operações são executadas, demonstrando como são realizadas as interações entre objetos. Nestes diagramas temos também a representação da ordem cronológica das interações, de cima para baixo (mais antigo para mais recente). Utilizando este tipo de diagramas é possível representar um caso de uso, capturando a interação de alto nível entre o utilizador e o sistema (ou entre vários sistemas) [15] [23].

Foram escolhidos os SSDs para representar os casos de uso do *RoboPHISH* pois estes são adequados ao tipo de interações feitas pelo robô. Os robôs interagem com as aplicações quase como um utilizador interage com um sistema, utilizando muitas vezes a interface web dos sistemas necessários. Para além disso podemos representar cronologicamente as ações do robô, ajudando assim a informar a ordem com que as ações se processam no BP.

Em seguida são apresentados vários diagramas que ilustram a execução de cada um dos processos. Estes diagramas podem representar o caso de uso inteiro ou apenas partes do mesmo.

RoboPHISH001 - 01 - *Save Emails to Folder and Queue*

Neste caso de uso o robô interage com o Outlook, de forma a extrair a informação necessária dos emails. O robô tem uma *Inbox* dedicada a este projeto em particular e é nesta que se encontram os emails denunciados pelos colaboradores. Cada email enviado por um colaborador pode conter em anexo um ou mais emails potencialmente maliciosos. A Figura 4.3 demonstra as interações do robô com o Outlook e com o Command Prompt (CMD).

- O robô acede à *Inbox* designada para o *RoboPHISH* e extrai os emails que tenham no seu *subject* a palavra "Análise" para uma *Collection*. Esta parte do processo é demonstrada na função *GetReceivedEmailsFromInbox()*;
- O robô acede a cada um dos elementos da *Received Items Collection*, devolvida por *GetReceivedEmailsFromInbox()*, e extrai para uma pasta de rede todos os anexos de cada um dos emails;
- Na função *OpenCMD()* o robô abre o terminal e envia um comando para abrir um determinado email (extraído no passo anterior) através do Outlook em *batch*;
- Já com a janela do email aberta, o robô retira a informação do *Subject* do email (*GetSubject()*), do *Header* (*GetHeader()*) e dos *links* (*GetLinks()*);
- No final o robô fecha o Outlook (*TerminateOutlook()*) e o terminal (*CloseCMD()*), deixando tudo no seu estado original.

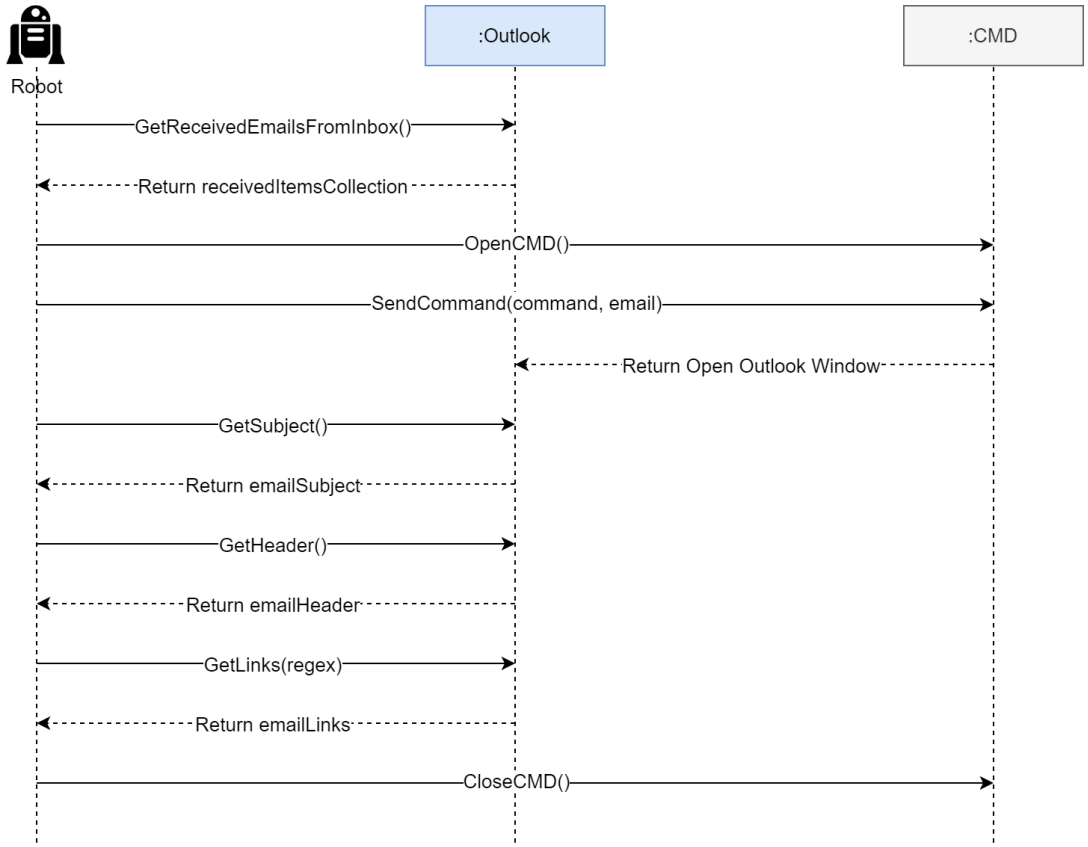


Figura 4.3: SSD que representa o caso de uso *Save Emails to Folder and Queue*

RoboPHISH001 - 02 - *Analyse Emails from Folder*

Neste caso de uso o robô analisa os campos recolhidos do email, com o objetivo de atribuir uma classificação inicial ao mesmo. Para conseguir fazer esta análise, o robô recorre às informações do email recolhidas no processo anterior: *RoboPHISH001 - 01 - Save Emails to Folder and Queue*. Para além de realizar uma análise inicial, o robô atua imediatamente sobre os emails classificados como SPAM ou *phishing*, isto é, envia email de resposta aos colaboradores, bloqueia o *sender* no Anubis e bloqueia os *links* maliciosos no Zscaler. A análise requer interações com o *Chrome* (*VirusTotal*, interface web do HPSM e interface *web* do *Abuse Report Form*), com o SQL *Server* e com o Outlook.

As Figuras 4.4 e 4.5 representam a estrutura geral do caso de uso, na forma de SSDs. Por motivos de legibilidade o diagrama está dividido em duas figuras. Para além da divisão foram necessários alguns diagramas complementares, que fazem parte do caso de uso. Estes diagramas também estão à parte do diagrama geral, devido à grande extensão do mesmo. O diagrama da Figura 4.4 representa a sequência de passos necessária para classificar, inicialmente, um email. Caso cada uma das condições seja cumprida, o colaborador é imediatamente informado, Figura 4.5, e os bloqueios necessários são efetuados (a Figura 4.6 corresponde ao bloqueio no Anubis e a Figura 4.7 ao bloqueio no Zscaler).

Descrição detalhada do SSD representado na Figura 4.4:

- No primeiro passo o robô acede aos *items* com classificação pendente, obtidos no módulo

01, estes correspondem aos emails recolhidos que ainda não tiveram um processamento inicial. Este passo está descrito na função *GetPendingEmails()*;

- A primeira verificação feita trata-se de *Campaign*, isto é, verificação de ataque simulado de *phishing*. Esta verificação é feita recorrendo a uma tabela que contém os domínios da campanha, no SQL. A função *GetCampaignList()* representa o pedido SQL da tabela, sendo que o resultado é devolvido para uma coleção, *listOfCampaigns*;

- Na função *GetAllClassifiedEmails()* está representada a verificação da existência de emails iguais já classificados, guardados no BP;

- A função *GetURLblacklist()* representa a interação com o SQL *Server*, em que são devolvidos os conteúdos da tabela em questão para uma *Collection* com o nome *listOfURLs*. Esta tabela contém os URLs que já foram previamente classificados como *malicious* ou *phishing* pelo VirusTotal;

- No passo seguinte o robô acede ao *Chrome*, na página do VirusTotal. A função *OpenBrowserInVirusTotal()* representa esta interação;

- Na função seguinte, *SendURLtoVirusTotal(url)* está representada a pesquisa de um determinado URL, presente no email suspeito, no VirusTotal. São devolvidos vários campos para uma coleção BP, entre eles temos o campo de *phihsing* pesquisa e data da última atualização. Estes campos estão representados na variável *listOfClassifications*.

- Após ter sido pesquisado o URL em questão, o robô coloca outra vez a página inicial do VirusTotal onde pode ser feita uma nova pesquisa. A função *GoToHomepage()* representa esta ação;

- Visto que para cada email podemos ter vários URLs, os últimos dois pontos são feitos num *loop*. Cada iteração do *loop* representa um URL a pesquisar no VirusTotal;

- Na função *CheckURLicu()* verificamos se algum dos domínios têm a extensão .icu, visto que por observação foi determinado que estes são tipicamente domínios de SPAM;

- Na função *CheckCrowdsourcing()* é aplicado o algoritmo de *crowdsourcing* ao email, este algoritmo será descrito detalhadamente na Subseção 4.3.3.

Descrição detalhada do SSD representado na figura Figura 4.5:

- Caso qualquer um dos pontos anteriores seja verdade, o robô deverá enviar um email ao colaborador que reportou o email de SPAM ou *phishing*. Isto está exemplificado numa sequência de *flags*, cada uma para um dos pontos em questão;

- Para além de enviar email ao colaborador, deverá ser atualizada a informação no próprio robô, ajudando assim a construir uma fonte de emails já classificados.

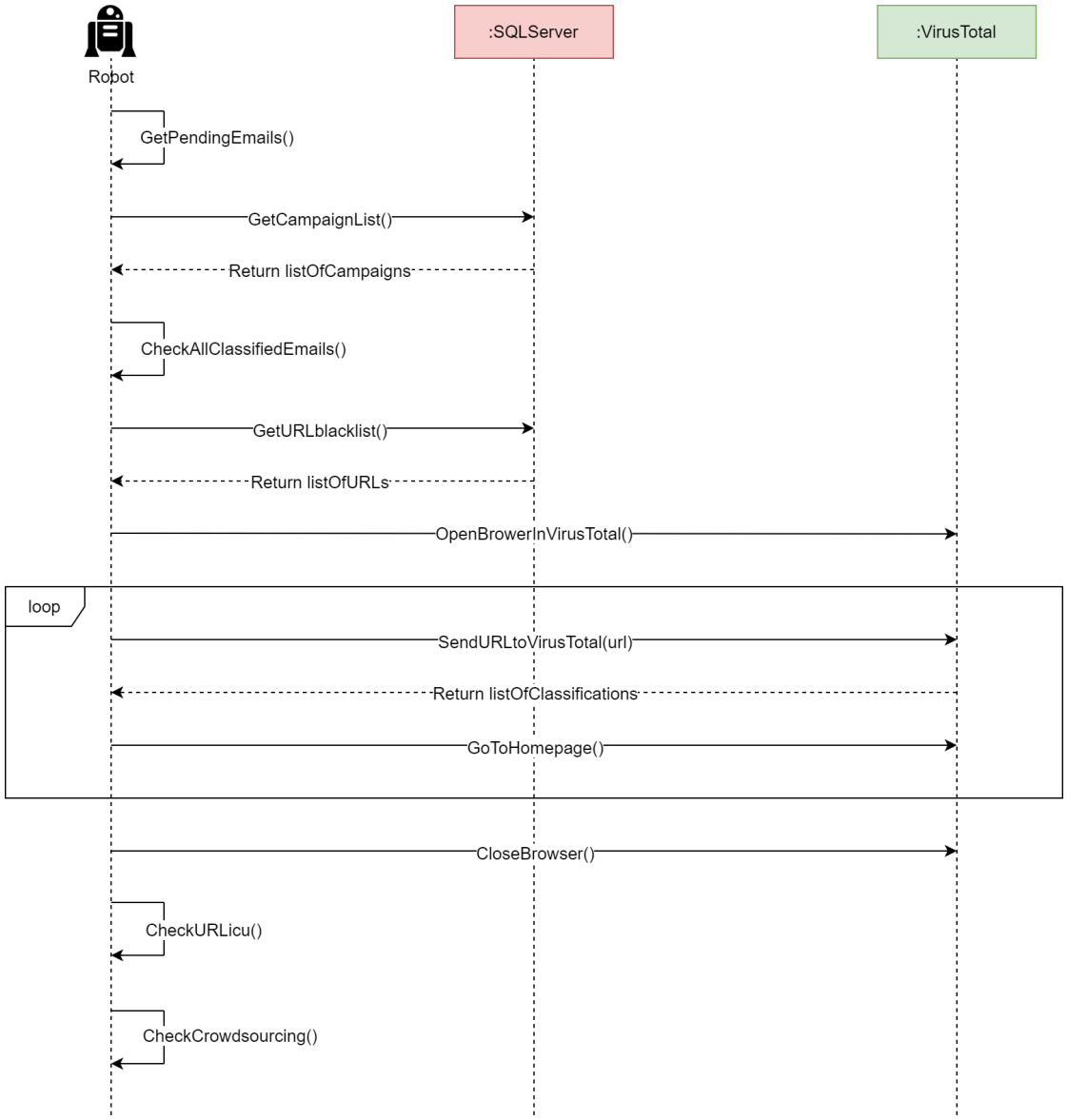


Figura 4.4: SSD que representa a primeira parte do caso de uso *Analyse Emails from Folder*

Na Figura 4.6 temos o diagrama de interação do robô com o *Chrome*, em particular no Abuse Report Form (ARF). Este SSD serve para demonstrar a forma a reportar um endereço a bloquear no Anubis. Este diagrama representa uma parte do caso de uso *RoboPHISH001 - 02 - Analyse Emails from Folder*, mas devido à extensão do caso e à independência do bloco do ARF em particular, foi colocado à parte. Esta interação é feita diretamente com o *Chrome*, sendo que após a submissão é o *Chrome* que envia os emails necessários, através do Outlook.

Descrição detalhada do SSD representado na figura Figura 4.6:

- Na função *OpenBrowserInARF()* o robô lança o *browser Google Chrome* na página do ARF;
- De seguida é anexado o email malicioso a bloquear no Anubis, sendo necessário fornecer o caminho completo do email. Este passo é demonstrado na função *AttachEmailToARF-form(email)*, sendo que a variável email representa o caminho completo do mesmo;
- Na função *FillForm(emailReceiver, bcc)* o robô preenche o resto dos campos do formulário,

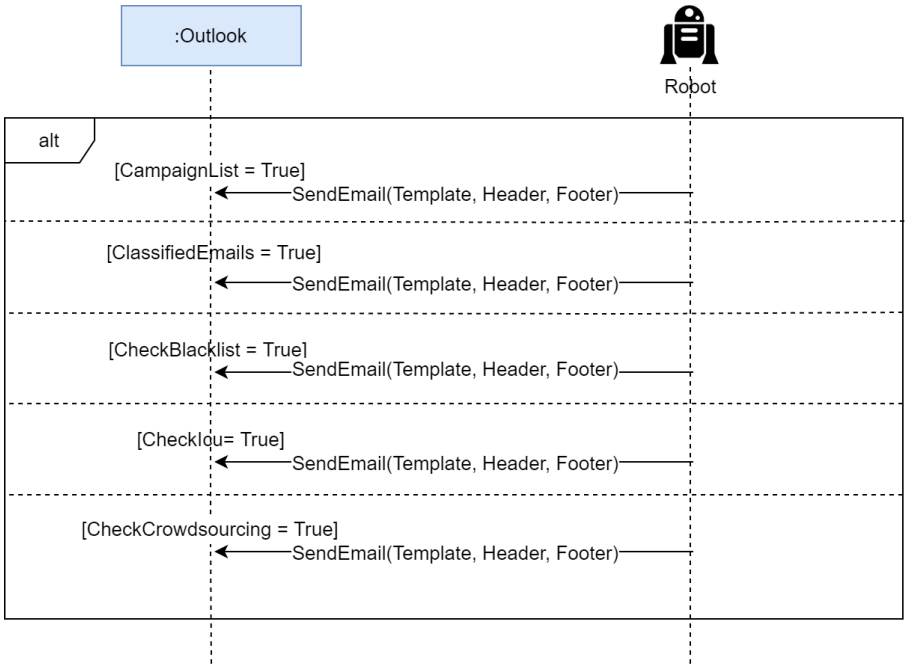


Figura 4.5: SSD que representa a segunda parte do caso de uso *Analyse Emails from Folder*

adicionando o destinatário do *report* (Anubis) e colocando em Bcc o endereço do CSIRT e do *RoboPHISH*;

- Após o preenchimento de todo o formulário este é submetido, sendo assim enviado um email padrão para as entidades submetidas;
- A função final corresponde à ordem dada pelo robô para fechar o *browser*, *CloseBrowser()*.

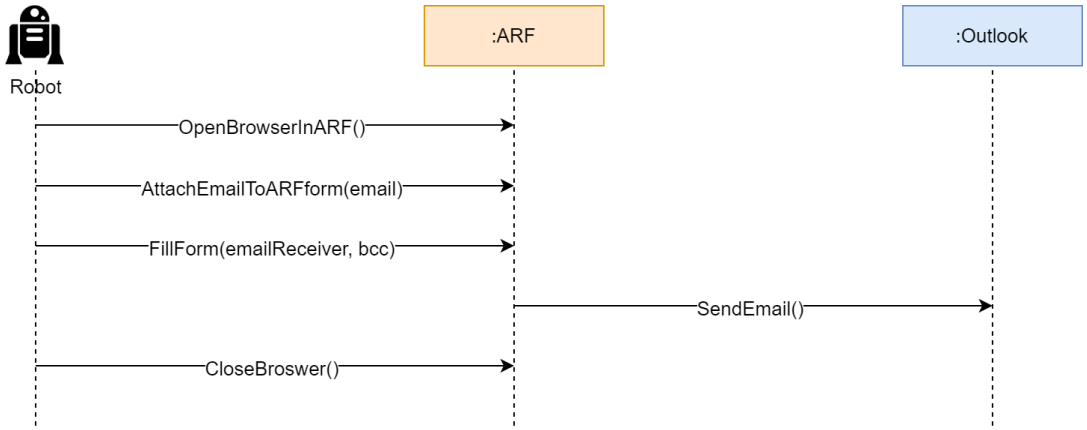


Figura 4.6: SSD que representa o bloco ARF do caso de uso *Analyse Emails from Folder*

Na figura Figura 4.7 temos o SSD da interação do robô com o *Chrome*, em particular no HPSM. O HPSM é uma das plataformas de *tickets* na Altice Portugal, onde através desta é possível abrir um *ticket* de natureza variada, a ser resolvido pela entidade adequada. No caso deste *ticket* em particular, é utilizado um *template* para reportar um *url* malicioso. Este diagrama representa uma parte do caso de uso *RoboPHISH001 - 02 - Analyse Emails from Folder*, mas devido à extensão do caso e à independência do bloco do HPSM em particular, foi colocado à parte. Esta interação

é feita diretamente com o *Chrome*, sendo que após a submissão é o *Chrome* que envia os emails necessários, através do Outlook.

Descrição detalhada do SSD representado na figura Figura 4.7:

- Na primeira função, *OpenBrowserInHPSM()*, o robô abre o *Chrome* na página *web* do HPSM;
- No passo seguinte o robô acede à página de *Incident Management*, nesta página temos os *templates* todos que dizem respeito à gestão de incidentes. Isto está representado na função *GoToIncidentManagementPage()*;
- Já dentro da página de gestão de incidentes o robô seleciona o *template* para registar um pedido de bloqueio de URLs, *InSecTemplate*. Este passo está representado na função *ApplyTemplate(InSecTemplate)*;
- Na função *SendEmail()* temos a representação do envio de email que é feito automaticamente através da aplicação Outlook;
- Por fim o robô fecha o *browser*, representado pela função *CloseBrowser()*.

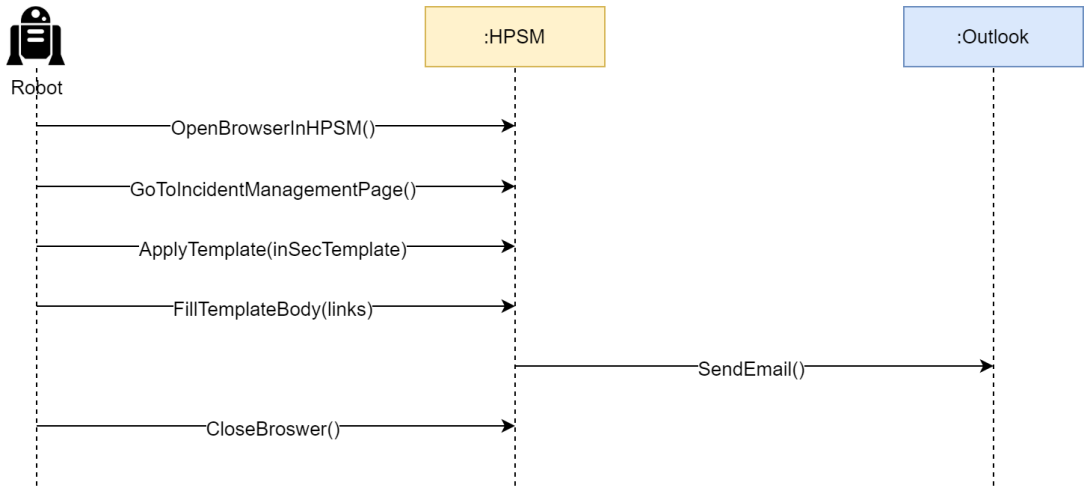


Figura 4.7: SSD que representa o bloco HPSM do caso de uso *Analyse Emails from Folder*

RoboPHISH001 - 03 - Scan Not Classified

No caso de uso correspondente ao processo *RoboPHISH001 - 03 - Scan Not Classified*, o objetivo principal é tratar dos emails que não conseguiram uma classificação inicial. Após a extração da informação do email no primeiro caso de uso e a análise inicial dos mesmos no segundo, é necessário tentar classificar novamente os emails que ficaram por classificar. O robô tem a informação necessária no próprio BP, sendo que este caso tem maioritariamente operações internas, apenas comunicando com a aplicação do Outlook. A Figura 4.8 demonstra as interações dentro do robô bem como as interações com o Outlook.

Descrição detalhada do SSD representado na Figura 4.8:

- O robô inicialmente pega em todos os emails que já foram analisados uma vez pelo módulo 02 mas não obtiveram classificação. Este passo está representado na função *GetPending-NotClassifiedEmails()*;

- De seguida o robô vai analisar estes emails sem classificação inicial. A primeira verificação consiste da função *CheckSentToSOC()*, nesta o robô verifica se aquele email está à espera de resposta há mais de 4 horas e se o colaborador que reportou o email tem boa reputação. Quando ambas as condições se verificam será enviado um email a pedir análise manual ao SOC;

- A segunda verificação feita pelo robô serve para atribuir o estado *Clean* a uma mensagem que esteja por classificar há mais de 24 horas. A ideia é informar o colaborador que o *RoboPHISH* não encontrou problemas evidentes e classificar a mensagem como *Clean* para a base de dados futura. Este passo está representado na função *CheckClean()*;

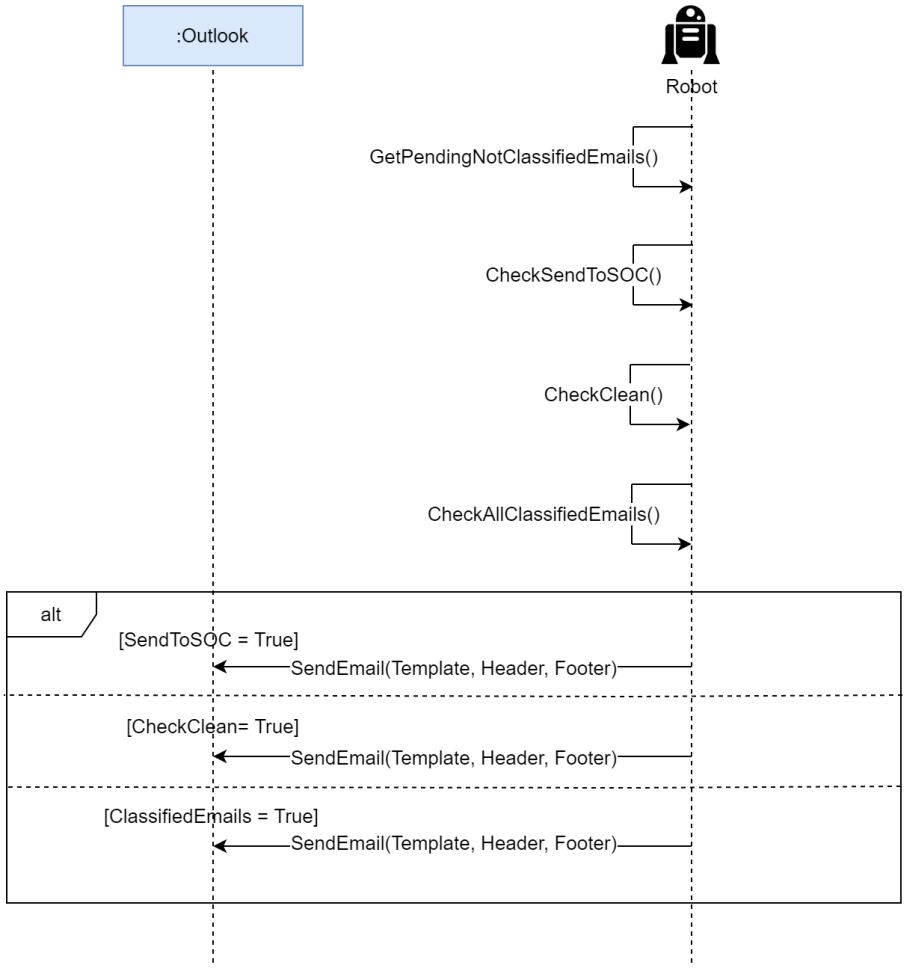


Figura 4.8: SSD que representa o caso de uso *Scan Not Classified*

- O último caso corresponde a uma nova análise dos emails já classificados, presentes no BP. Caso haja algum igual que foi classificado entretanto, a mesma classificação deverá ser atribuída. A função *CheckAllClassifiedEmails()* representa este passo;

- Caso qualquer uma das condições anteriores se verifique o robô deve informar adequadamente o colaborador que denunciou o email. Esta interação com o Outlook está descrita dentro do bloco alt (*alternative*).

4.3.3 Algoritmo de *Crowdsourcing*

O algoritmo de *crowdsourcing* tem uma função essencial: serve para ajudar a classificar os emails que não conseguem ser classificados de outra forma. Para além desta função, que é a principal, podemos também premiar os utilizadores com melhor participação.

O algoritmo consiste da atribuição de um valor por cada resposta correta, conseguindo desta forma encontrar um valor de exatidão (*accuracy*) para cada utilizador. Depois são definidos *thresholds* para este valor de exatidão, podendo assim um utilizador com um valor maior que X (*threshold* definido) ajudar a classificar um email.

Atribuição de pontos

A atribuição dos pontos é efetuada cada vez que um colaborador faz uma denúncia de um email. Os pontos são distribuídos da seguinte forma:

- Caso o colaborador classifique um email como SPAM ou *phishing* e o *RoboPHISH* confirme um dos dois, são então atribuídos 10 pontos a este utilizador.

- Caso o colaborador classifique um email como SPAM ou *phishing* e o *RoboPHISH* não o consiga confirmar ao fim de 48 horas, então este colaborador não recebe pontos (0 pontos).

Inicialmente a definição do algoritmo valorizava quem acertava completamente, isto é, apenas dava a pontuação completa (10 pontos) às denúncias exatamente corretas. As denúncias parcialmente corretas recebiam apenas meia pontuação (5 pontos). Exemplificando:

- Caso o colaborador classifique um email como SPAM e o *RoboPHISH* atribua a classificação de *phishing* ou o colaborador classifique como *phishing* e o *RoboPHISH* atribua SPAM, então o colaborador recebe apenas 5 pontos.

- Caso o colaborador classifique um email como SPAM e o *RoboPHISH* confirme que é SPAM ou o colaborador classifique como *phishing* e o *RoboPHISH* confirme o *phishing*, são então atribuídos 10 pontos a este utilizador.

- Caso o colaborador classifique um email como SPAM ou *phishing* e o *RoboPHISH* não consiga determinar ao fim de 24 horas, então este colaborador não recebe pontos (0 pontos).

Esta definição foi abandonada verificando-se, após testar, que desfavorecia muito os colaboradores. A ideia é que estes utilizadores ajudem a classificar os emails, que se apercebam que há qualquer coisa de errado com o email recebido. Por vezes podem não ter a classificação exatamente correta mas conseguiram distinguir que o email era indesejado, que é o que se pretende.

Cálculo da exatidão

A exatidão está definida como o grau de aproximação entre o valor de uma medição e o valor verdadeiro da grandeza medida. Neste caso, a exatidão do utilizador diz respeito a quão perto as denúncias do mesmo estão dos resultados esperados.

Após a definição adequada das pontuações é possível calcular a exatidão (*accuracy*). Este valor é calculado a cada email denunciado pelo colaborador, tal como a atribuição de pontos. Cada vez que um colaborador denuncia um email, este é analisado e após a decisão do robô são atribuídos pontos e calculada novamente a *accuracy*.

A fórmula definida para a *accuracy* de cada utilizador é a seguinte:

$$Accuracy = NumberOfCorrectReports/TotalNumberOfReports$$

O valor do *NumberOfCorrectReports* corresponde então ao número de emails bem classificados pelo colaborador. O valor do *TotalNumberOfReports* corresponde ao número total de denúncias feitas pelo colaborador (incorretas e corretas).

Calculando a exatidão desta forma os resultados poderiam ser pouco indicativos nas primeiras execuções. Caso o colaborador, por exemplo, acertasse a primeira denúncia ficaria logo com *accuracy* de 100%. Para evitar este cenário, a *accuracy* só é utilizada para fazer decisões a partir da 10ª denúncia, embora seja calculada desde a primeira denúncia.

Classificação dos colaboradores

A ideia da classificação dos colaboradores é colocar cada utilizador do sistema numa categoria, para desta forma podermos selecionar só os que estão numa determinada categoria para classificar um email. Para chegar aos valores definidos para a classificação de um colaborador foi necessário um período de testes. Apenas testando a adesão dos colaboradores foi possível perceber quais os valores de *accuracy* adequados para cada categoria.

As categorias definidas foram as seguintes:

- ***Gurus***: Inicialmente trata-se de um conjunto de colaboradores estritamente internos e explicitamente nomeados pelo Comité *PHISHFighting* da DCY, por serem considerados como confiáveis na classificação de emails como SPAM ou *phishing*. Na fase de testes na DCY, temos um conjunto de pelo menos cinco elementos da DCY, com a possibilidade de promover mais elementos através de um valor de *accuracy* superior a 75%. Em fase de *full production* o número de Gurus deverá estabilizar nos melhores 50 elementos (em atividade e exatidão) da Altice Portugal do universo Office365.

- ***Experts***: Inicialmente trata-se de um conjunto vazio. Depois serão propostos novos elementos, automaticamente e incrementalmente, pelo algoritmo de *crowdsourcing*. Um colaborador é automaticamente promovido a *Expert* se tiver contribuído com pelo menos 10 classificações e tiver pelo menos 50% de *accuracy*. Quando esse valor de *accuracy* baixar, deixa de ser categorizado como *Expert* e, assim, deixa de contar para o processo de voto. Alguns destes *Experts* poderão ser depois promovidos a *Gurus* pelo Comité *PHISHFighting* da DCY, no processo de garantir os TOP 50 no âmbito geral da Altice Portugal.

- ***Informed***: Este conjunto de utilizadores é exclusivamente proposto, automaticamente e incrementalmente, pelo algoritmo de *crowdsourcing*. Considera-se que um utilizador é *Informed* se este tiver contribuído com pelo menos dez denúncias e tiver menos 50% de *accuracy*. Estes utilizadores poderão evoluir a *Experts*, caso o seu valor de *accuracy* aumente para mais de 50%.

- ***No Classification Yet***: Quando um utilizador não chega a ter dez denúncias (condição base para classificação), é mantido sem classificação. A sua *accuracy* é calculada e guardada mas o colaborador não pertence a nenhuma categoria ainda.

Decisões do algoritmo

A classificação dos utilizadores serve para auxiliar a tomada de decisões sobre os emails que o *RoboPHISH* foi incapaz de classificar através das plataformas utilizadas nos passos antecedentes. A categoria mais importante e com mais poder decisivo é a do *Guru*, seguindo-se o *Expert* e por fim *Informed*.

O último bloco de decisão do robô (*Analyse Emails from Folder*) é o algoritmo de *crowdsourcing*. Dentro deste bloco o robô verifica se o email suspeito foi enviado por um *Guru*. A regra para classificação por *Guru* é então a seguinte: caso não haja mais forma nenhuma de classificar o email, o robô atribui a classificação dada pelo *Guru*. Para além dos *Gurus* temos mais um elemento que pode ajudar a classificar um email, os *Experts*.

A opinião dos *Experts* é utilizada no módulo 03, *Scan Not Classified*. Neste módulo, o robô pretende enviar para o SOC os emails de pessoas com classificação de *Expert*. A regra para classificação por *Expert* é então a seguinte: caso o email esteja por classificar há mais de 4 horas e o emissor seja um *Expert*, o robô deverá então enviar o email ao SOC para classificar manualmente. Um colaborador do *soc* deverá então abrir o email, fazer a verificação, e classificar o email. Esta regra foi colocada para não inundar o SOC com emails por classificar, assim só os emails das pessoas com o segundo melhor grau de reputação são avançados para o SOC.

Caso o email não atinja estes requisitos, deverá continuar a tentar classificar o email, no máximo durante 24 horas. Após este período de tempo é considerado que não foi possível classificar o email e o robô assume o email como *Clean*. Para o utilizador é enviado um email com o *template Seems Clean*, neste email é informado que o robô não encontrou ameaças no email mas se o colaborador não desejou este email, não deverá interagir com o mesmo de qualquer das formas. A Figura 4.9 representa as decisões tomadas pelo robô, no que diz respeito ao algoritmo de decisão, através de uma árvore.

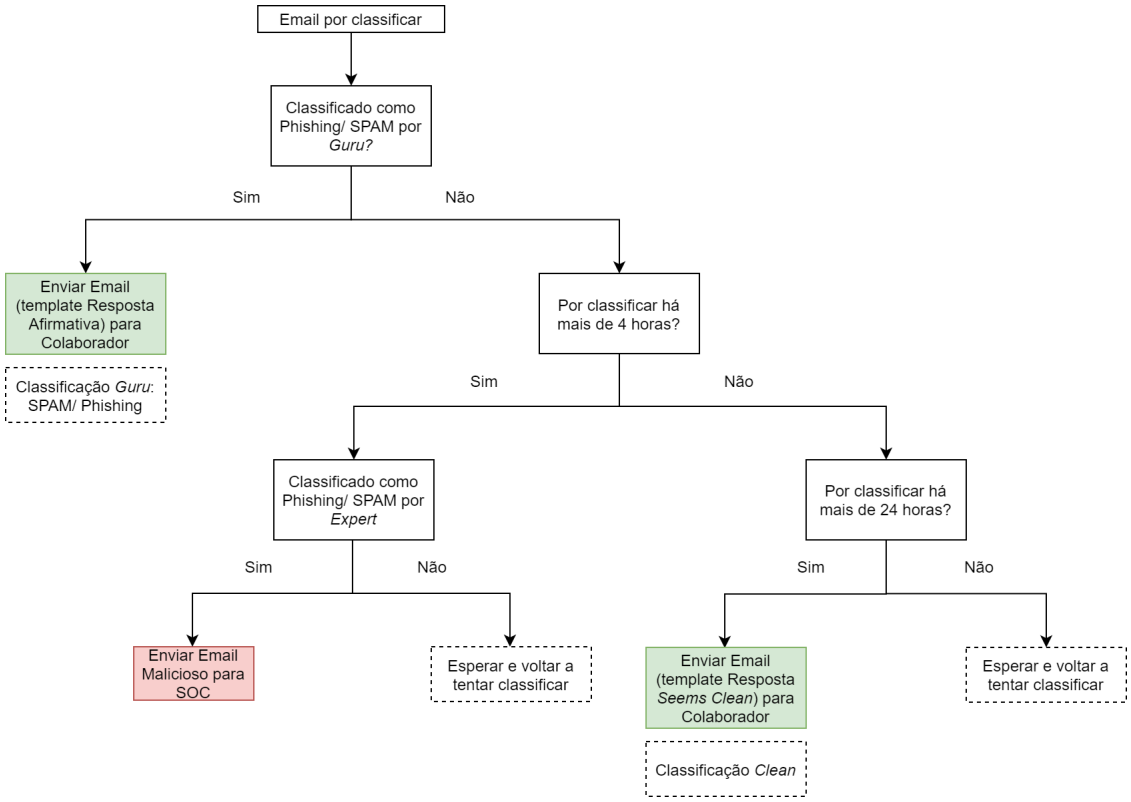


Figura 4.9: Árvore para classificação de um email (*crowdsourcing*) do *RoboPHISH*

Capítulo 5

Implementação do *RoboPHISH*

Antes de iniciar a etapa de implementação do processo de denúncia aos ataques de *phishing*, foi realizada uma reunião com os analistas SOC onde foi feito o levantamento do processo *report phishing* executado pelos mesmos. Após o levantamento, foi possível concluir quais as etapas que são possíveis automatizar.

Para além de automatizar o atual processo de *report* do *phishing*, o sistema *RoboPHISH* pretende inovar e trazer uma alternativa mais eficaz de analisar o *phishing*. O novo processo de denúncia e análise implementado é muito distinto do processo anterior, onde os emails eram analisados manualmente, um a um, pela equipa SOC.

Nas secções seguintes são descritos os processos implementados, referindo as ferramentas utilizadas, reforçando os objetivos do processo, explicitando detalhadamente a implementação do mesmo e referindo as dificuldades na elaboração.

Os processos foram implementados recorrendo às macros do Outlook (VB) e ao BP e foram executados, numa fase de testes, em máquinas locais e numa fase de produção em máquinas remotas (situadas na Covilhã - Portugal).

5.1 *Outlook Button to Report Phishing*

A elaboração da primeira parte do projeto corresponde à implementação de um botão de denúncia de *phishing*, desenvolvido recorrendo a macros do Outlook 2013. Uma macro é uma sub-rotina num módulo de código, que serve para atingir um objetivo mais rapidamente, poupando tempo nas tarefas repetitivas. As macros são escritas em VB, uma linguagem de programação da Microsoft orientada a eventos [32]. A Figura 5.1 representa a função *PrepareEmail* do botão *SPAM/ Phishing*, desenvolvido em VB.

A macro desenvolvida permite a denúncia de um ou mais emails suspeitos por parte dos colaboradores. O botão *SPAM/ Phishing*, que é adicionado à *Ribbon*[1], deverá executar a macro. O botão é de utilização simples e intuitiva, lançando mensagens de ajuda sempre que o utilizador não o utiliza corretamente.

[1]Menu do Outlook que contem múltiplas *tabs*, cada uma com vários grupos de comandos. Estas *tabs* servem para executar as tarefas mais comuns no Outlook [5].

Através do botão *SPAM/ Phishing* o email suspeito é enviado em anexo e o assunto da mensagem é definido como ”Phishing - Análise” ou ”SPAM - Análise” (definido conforme a escolha do utilizador).

```
'corre APENAS 1x, em PRIMEIRO lugar
Function PrepareEmail(emailType As String)

    Dim phishingaddress As String
    Dim subject         As String
    Dim body            As String
    Dim junkFolder      As Outlook.Folder
    Dim MyDate
    Dim MyTime
    Dim Category        As Categories
    Dim cat             As Category


    'Enviar para:
    'eh para robophish@telecom.pt
    phishingaddress = "ines-a-rodrigues@telecom.pt"

    'Get date
    MyDate = Date

    'Get time
    MyTime = Time

    body = "Suspeita de email(s) de " & emailType
    & vbNewLine & MyDate & " " & MyTime

    subject = emailType & " - Análise"

    'email novo, em branco
    Set objMail = Application.CreateItem(olMailItem)

    objMail.To = phishingaddress
    objMail.subject = subject
    objMail.body = body

End Function
```

Figura 5.1: Exemplo de uma das funções em VB

A execução do botão *SPAM/ Phishing* é feita através de múltiplas funções, *ReportSPAMPhish*, *getEmailType*, *PrepareEmail*, *AddAttachments*, *AddToJunk* e *FinalSendEmail*, descritas em alto nível nos pontos seguintes:

1. Quando o colaborador seleciona um ou mais emails e carrega no botão *SPAM/ Phishing*:

   (a) Cada email é processado individualmente (iterar sobre o conjunto de emails enviados pelo colaborador);

   (b) É feita a verificação se o email é interno, ou seja, se pertence ao domínio @*telecom.pt*;

   (c) Em caso afirmativo (email interno) é lançado um *pop-up* com um aviso a alertar qual(ais) o(s) email(s) interno(s) selecionado(s) (Ilustrado na Figura 5.2). Dentro do *pop-up* são apresentadas duas opções de resposta ao colaborador: **OK** ou **Cancel**. Esta decisão representa a continuação do processo de *report* (OK) ou término imediato da macro (Cancel);

   (d) Caso o colaborador tenha selecionado a opção **OK**, representada na Figura 5.2, é lançado um formulário onde o colaborador seleciona qual o tipo de email malicioso

(SPAM ou *phishing*) que este quer enviar para análise. O valor escolhido pelo colaborador é guardado numa variável;

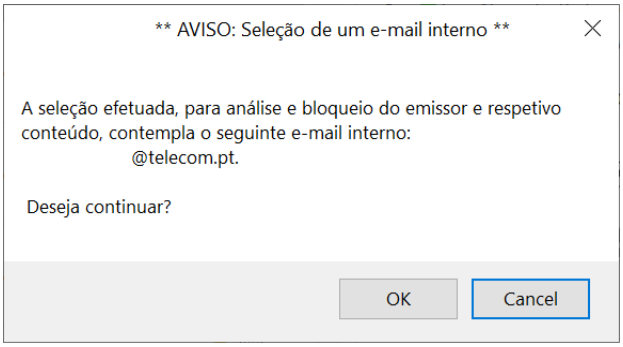


Figura 5.2: Exemplo do *pop up* de um email interno

(e) De seguida é criado um email novo, *Report Email*, com o *subject*: ”Phishing - Análise” ou ”SPAM - Análise”, dependendo da opção selecionada pelo colaborador;

(f) O *Report Email* terá o *body*: ”Suspeita de email(s) de SPAM dd/MM/AAAA HH:mm:ss”ou ”Suspeita de email(s) de Phishing dd/MM/AAAA HH:mm:ss”, dependendo da opção selecionada pelo colaborador;

(g) O(s) email(s) selecionado(s), é/são colocado(s) em anexo no *Report Email*;

(h) O(s) email(s) selecionado(s) é/são movido(s) para a pasta *Junk* (email de lixo) e o *subject* de cada um leva o prefixo: ”[moved To Junk by RoboPHISH]”;

(i) O *Report Email* criado é enviado para a caixa de correio do *RoboPHISH*;

(j) Paralelamente é criada uma *Rule*[2], denominada *Block Senders*, para bloquear os emails dos remetentes selecionados. Apenas são adicionados os remetentes que não façam parte de nenhum domínio interno. Esta regra move automaticamente para a pasta *Junk* todos os futuros emails deste remetente. No caso de já ter sido criada a regra anteriormente (noutras denúncias) esta é reutilizada, ou seja, somente são adicionados os remetentes à regra existente;

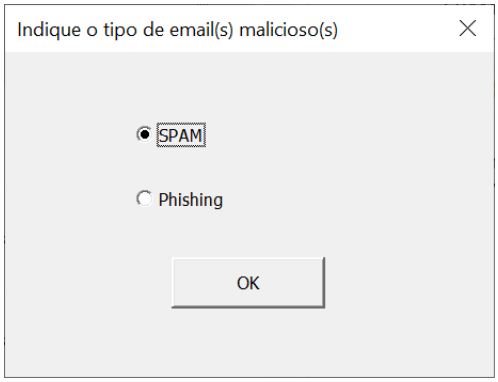


Figura 5.3: Exemplo do *pop-up* de email interno

[2]Uma *rule* do Outlook é uma ação executada automaticamente quando as condições indicadas são satisfeitas [17].

(k) Após o envio do *Report Email*, devidamente preenchido, é lançado um *pop-up* com uma mensagem de sucesso, notificando o colaborador que a sua denúncia está concluída e que receberá *feedback* brevemente (Figura 5.4).

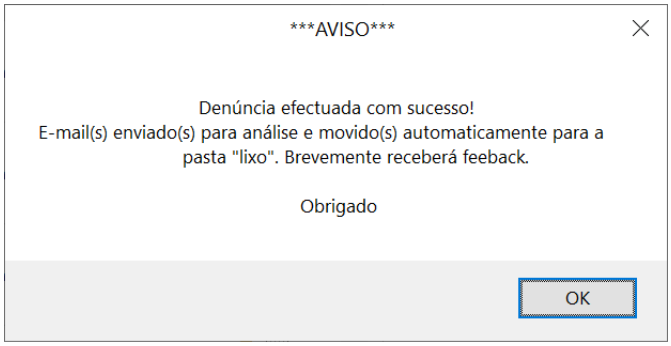


Figura 5.4: Exemplo do *pop up* de sucesso

2. Quando o colaborador tenta denunciar o email malicioso, através do botão *SPAM/ Phishing*, mas não seleciona nenhum email da sua caixa de correio:

   (a) É lançado um *pop-up* com uma mensagem de erro, indicando que o colaborador deverá selecionar pelo menos um email (ilustrado na Figura 5.5).

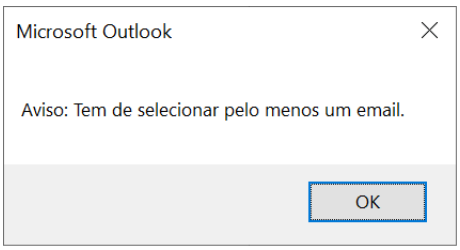


Figura 5.5: Exemplo do *pop up* de um email interno

Após a criação da macro foi necessário adicionar um novo menu e o botão *SPAM/ Phishing* ao menu principal do Outlook (*Ribbon*). O Outlook disponibiliza vários ícones e dentro destes foi selecionado o ícone do escudo vermelho por ser chamativo e aludir ao tema da proteção e segurança. O botão está associado à macro criada, sendo que sempre que um utilizador carrega no botão, o código é executado. A Figura 5.6 representa o aspeto e posicionamento do botão na *Ribbon* do Outlook.

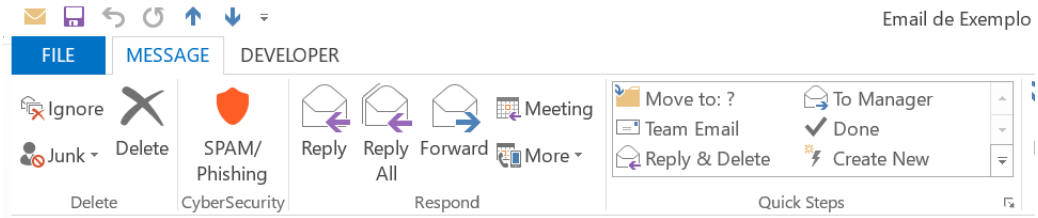


Figura 5.6: Apresentação do botão *SPAM/ Phishing* na *Ribbon* do Outlook

Para além do desenvolvimento da macro foi também desenvolvido um manual para instalação da mesma (Apêndice A). Este manual serviu não só para a DCY conseguir testar o botão, mas

também para permitir ao *RoboPHISH* ter acesso a uma amostra de emails para analisar. O manual tem instruções claras e sucintas, com capturas de ecrã, para ajudar a compreender todos os passos necessários para a instalação.

5.2 *RoboPHISH001 - 01 - Save Emails to Folder and Queue*

A primeira parte da automação do processo de análise de SPAM/ *phishing*, consiste na leitura, extração e armazenamento da informação dos emails denunciados pelos colaboradores. A informação é extraída tanto do email que contém o email suspeito em anexo (***Report Email***), como do próprio email suspeito (***Suspicious Email***). O processo construído interage com as seguintes aplicações: Outlook, CMD e Notepad.

Para cada ***Report Email*** são extraídos os seguintes campos (com exemplos para clarificação):

- *Entry ID*: 00000000BD34E1B8FEF6DF4DA227D36E61C7BE180700D33A7712480D8145B52E8F009A7D7F8100000000010C0000D33A7712480D8145B52E8F009A7D7F8100003564A9E90000
- *Sender Email Address*: nome-do-colaborador@telecom.pt
- *Sender Name*: Nome Colaborador
- *Email Subject*: SPAM - Análise
- *Sent On*: 23/05/2020 15:07
- *Received On*: 23/05/2020 15:07

Para cada ***Suspicious Email*** são extraídos os seguintes campos (com exemplos para clarificação):

- *Email Subject*: Peça o seu empréstimo sem sair de casa e receba o dinheiro na conta em 48h!
- *Email Body*: Um crédito mais barato 100% on-line. Simula o teu projecto. Assina online (...)
- *Malicious Email Sender*: newsletter@emailmalicioso.com
- *Malicious Email Receiver*: nome-do-colaborador@telecom.pt
- *Header Text*: Received: from localhost (localhost [000.0.0.0]) by PCINMAILTE (Postfix) with ESMTP id X for nome-do-colaborador.pt; Thu, 13 Aug 2020 12:50:41 +0100 (BST)
- *Links*: http://linkmalicioso.icu/

A extração dos campos do *Report Email* é feita através de um objeto existente no BP, que retira para uma coleção a informação dos emails presentes na Inbox - *Get Received Items (Basic)*. Através deste objeto conseguimos obter todos os campos do *Report Email* contudo, não é possível utilizar o mesmo objeto para obter os campos do *Suspicious Email*. Para combater esta dificuldade,

os emails suspeitos anexados ao email de denúncia são transferidos para serem posteriormente analisados, através de um objeto criado para este efeito - *Save Attachments (.msg)*. O objeto *Save Attachments (.msg)* consiste de um bloco de código em VB que extrai os *.msg* anexados de um email. Desta forma é possível ter todos os emails suspeitos transferidos para a pasta selecionada (pasta de rede do *RoboPHISH*).

Após a extração de cada um dos *Suspicious Emails*, anexados ao email de denúncia, é necessário analisar cada um deles individualmente. Isto é feito recorrendo a um ciclo, que seleciona cada um dos emails guardados na pasta definida, abre o email no Outlook (através do CMD) e retira a informação de cada um. Para conseguir fazer isto foi utilizado um objeto do BP que lança o CMD com os parâmetros definidos (/k "[Outlook Path]"/f "[Complete File Path]"). Desta forma é aberto o *Suspicious Emails* no Outlook, sendo assim possível identificar os elementos necessários através de um novo objeto criado para este efeito - *RoboPHISH001 - Outlook*.

O objeto *RoboPHISH001 - Outlook* tem então como objetivo identificar a janela do Outlook aberta, e identificar os elementos necessários através do *Application Modeller*. Neste caso são apenas necessários o *Body* e o *Subject*. Primeiro é necessário criar uma ação para identificar a janela aberta do Outlook (*Attach Outlook*), para que a janela fique associada à ação possibilitando assim interagir com a mesma. De seguida foi criada a ação *Get Subject* para extrair o assunto e corpo do email. Para além desta ação é necessário também obter o *Header* e os *Links*. Para isso foram criadas mais duas ações para este objeto. Na ação *Get Header* o robô acede a *File*, seguido de *Properties* e retira e guarda o campo *Header*. Na ação *Click View Source* o robô carrega no botão *Actions*, selecionando depois *View Source*, abrindo assim uma janela do Notepad com o *Source Code*[3] do email (Figura 5.7). Dentro do objeto *RoboPHISH001 - Outlook* falta somente a ação que fecha a aplicação - *Terminate Outlook*.

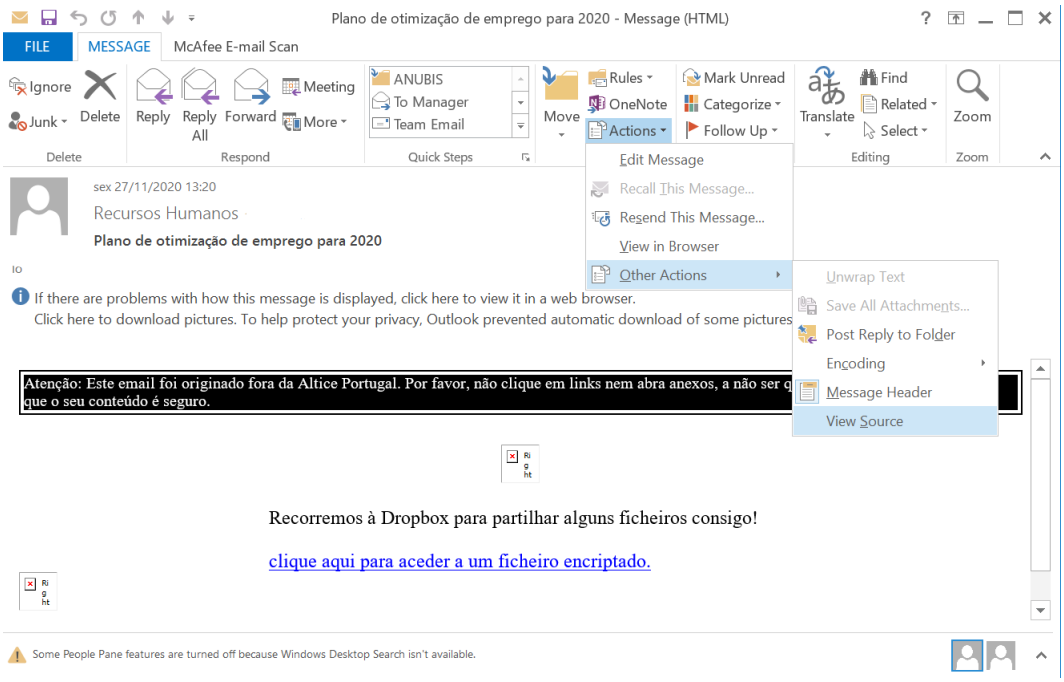


Figura 5.7: Exemplo de uma janela de email do Outlook

[3]Código HTML do email

Dentro do processo são chamadas as novas ações do objeto pela seguinte ordem: *Attach Outlook*, *Get Subject*, *Get Header*, *Click View Source* e *Terminate Outlook*.

Após a execução das ações do objeto *RoboPHISH001 - Outlook*, o robô terá o Outlook fechado e uma janela do Notepad aberta contendo o *Source Code*, em HTML, do email. Para ser possível a interação com este conteúdo foi criado um objeto novo - *RoboPHISH001 - Notepad*.

Tal como no objeto criado para as interações com o Outlook, as interações com o Notepad necessitam de um *Attach*, para ligar a janela aberta ao objeto, e de um *Terminate* para fechar a janela no fim. O primeiro elemento a ser identificado em qualquer interação com aplicações é a janela da aplicação em si, deste modo é possível executar as ações *Activate and Maximize* de forma a tornar a janela visível e maximizá-la. Para obter a informação do Notepad foi necessário ainda de criar o objeto *Read Info Notepad*, que seleciona o texto inteiro (CRTL + A) e copia (CRTL + C e ação BP *Get Clipboard*) o mesmo, guardando-o na variável Texto.

A variável Texto que contem o texto da *Source Code* do *Suspicious Email* necessita de ser trabalhada para ser possível retirar as hiperligações. Neste caso foi utilizada uma ação do BP, *Regex V2*, que aplica expressões regulares a uma variável do tipo *text* e devolve os resultados para uma *Collection*. Para este efeito foi criada uma expressão regular que identifica quais os "hrefs", que contêm a *substring* "http". Com base nas características específicas dos emails em HTML é possível utilizar a seguinte expressão regular:

$$< a\backslash s + (? : [\ \hat{}\ >]*?\backslash s+)?href = ([\backslash ?"'])http(.*?)[\backslash ?\backslash 040"']$$

O campo *Header Text*, recolhido através do objeto *RoboPHISH001 - Outlook*, também necessita de ser trabalhado, de modo a ser retirada alguma informação, nomeadamente *Malicious Email Sender* e *Malicious Email Receiver*. Para obter a informação necessária foram utilizadas, novamente, expressões regulares (que correspondem ao *Sender* e *Receiver* respetivamente):

$$From : .*?@.*? >$$
$$To : .*?@.*? >$$

À medida que os valores são retirados das aplicações Outlook e Notepad, são colocados nos campos de uma *Collection* (com o nome *Updated Collection*) para mais tarde poderem ser adicionados às filas de trabalho (*Queues*). Neste conjunto de processos são necessárias duas filas de trabalho, uma para os emails (RoboPHISH - EMAILS) e outra para os colaboradores/ utilizadores (RoboPHISH - USERS) (Figura 5.8).

Após o robô retirar todas as informações do email e de ter preenchido todos os campos necessários da *queue*, o estado do *item* é definido com o valor *01 - Email Saved in Folder*. Isto significa que o *Suspicious Email* foi guardado para análise e estão reunidas as informações necessárias para este ser analisado.

A Figura 5.8 corresponde a um pequeno diagrama desenvolvido para demonstrar a interação dos processos *RoboPHISH* com as *queues* correspondentes. No primeiro processo são adicionadas todas as informações necessárias para o processo de análise às *queues*. Os restantes processos consomem esta informação e, após análise, atualizam os campos necessários.

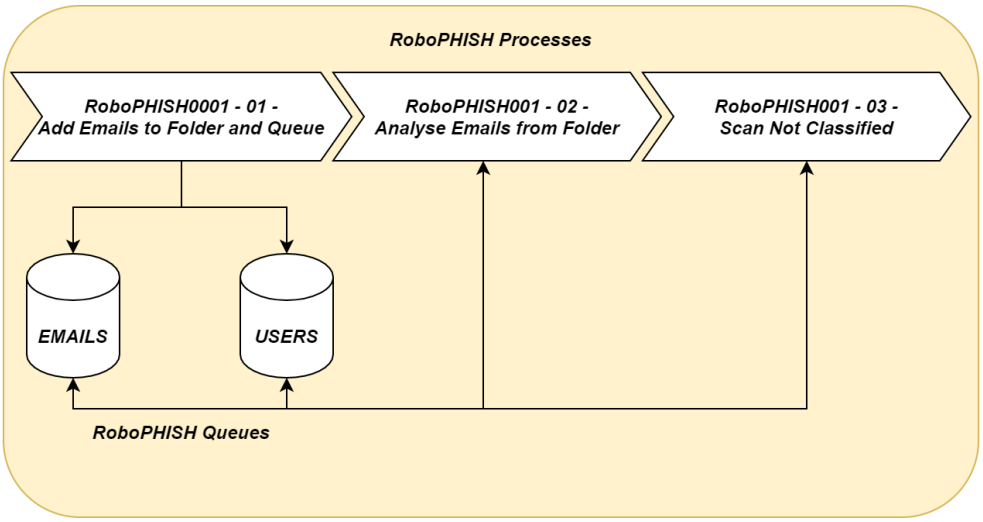


Figura 5.8: Diagrama de interação dos processos *RoboPHISH* com as suas *queues*

5.2.1 *Queue RoboPHISH - EMAILS*

Na fila de trabalho *RoboPHISH - EMAILS* é guardada toda a informação relacionada com os emails, portanto a informação mencionada na Secção 5.2. Os campos desta *Queue* são os seguintes: *AttachmentID*, *EntryID*, *ReceivedOn*, *SentOn*, *SenderName*, *SenderEmailAddress*, *EmailBody*, *EmailSubject*, *HeaderText*, *MaliciousEmailSender*, *MaliciousEmailReceiver*, *VirusTotalLinks*, *UserSuspected*, *ScanCybeReady*, *ScanQueue*, *ScanVT*, *ScanCrowdSourcing* e *ScanSOC*.

Alguns dos campos presentes foram mencionados anteriormente, bem como a forma de os obter. Dentro dos restantes campos temos o *AttachmentID* que serve como identificador de cada email suspeito, portanto tem de ser sempre único. Este campo corresponde ao *EntryID* (identificador único de cada email do Outlook) do *Report Email* juntamente com o sufixo ATT e o número do anexo.

$$AttachmentID = EntryID(Report\ Email) + ATTX$$

Onde X varia de 1 a N, sendo que N é o número de anexos (*Suspicious Emails*) presentes no email de denúncia.

O campo *Email Subject* do *Report Email* sofre uma alteração, a palavra Análise é retirada, ficando só ”PHISHING” ou ”SPAM”. Este valor é colocado na *Queue* no campo *UserSuspected*, um campo dedicado ao valor que o utilizador selecionou na denúncia do email.

O campo *Links* da *Collection* do *Suspicious Email* passa para o campo da *Queue* com o nome *VirusTotalLinks*. Este campo em particular trata-se de uma *Collection* visto que podemos ter várias hiperligações dentro de um email. Esta *Collection* será alterada no módulo 02 (*Analyse Emails from Folder*) após análise dos *links* (Secção 5.3).

Os restantes campos serão explicados, analisados e preenchidos nos processos *RoboPHISH001 - 02 - Analyse Emails from Folder* e *RoboPHISH001 - 03 - Scan Not Classified*.

5.2.2 *Queue RoboPHISH - USERS*

A fila de trabalho, *Queue RoboPHISH - USERS*, contém toda a informação relacionada de cada colaborador especificamente. O propósito da criação desta *queue* resulta do algoritmo de *crowdsourcing* e das estatísticas sobre o comportamento de cada colaborador. Estas informações serão fundamentais para calcular a *accuracy* para cada utilizador, podendo desta forma atribuir pesos diferentes às opiniões dos mesmos (Secção 5.5). Os campos desta *Queue* são os seguintes: *UserEmail*, *UserName*, *CountReports*, *CountReportsSPAM*, *CountReportsPhishing*, *CountClean*, *CountPhishing*, *CountSPAM* e *Rating*, descritos da seguinte forma:

- *UserEmail*: campo que corresponde ao email do utilizador tal como o campo *SenderEmailAddress* da *queue* de emails, sendo que ambos os campos são obtidos da mesma forma. Este campo serve de identificador único da *queue* visto que não existem emails iguais;

- *UserName*: campo correspondente ao nome do utilizador tal como o campo *SenderName* na *queue* de emails. Os valores deste campo são extraídos da mesma forma para as duas *queues*. Esta informação é guardada para facilitar o acesso ao nome do colaborador, caso seja necessário uma resposta mais personalizada por email;

- *CountReports*: campo (do tipo *Number*) onde é guardado o número de denúncias de SPAM/ *phishing* feitas por um determinado utilizador. Este valor é incrementando sempre que é feita uma denúncia por um colaborador;

- *CountReportsSPAM*: campo (do tipo *Number*) semelhante ao *CountReports* mas correspondente apenas às denúncias de SPAM. Sempre que um colaborador denuncia SPAM o valor é incrementado;

- *CountReportsPhishing*: campo (do tipo *Number*) semelhante ao *CountReportsSPAM* mas correspondente apenas às denúncias de *phishing*. Sempre que um colaborador denuncia *phishing* o valor é incrementado;

- *CountClean*: no campo *CountClean* é guardado o número de denúncias classificadas como *Clean*, isto é, denúncias que não foram confirmadas nem pelo *RoboPHISH*, nem pela equipa SOC. O campo é do tipo *Number* e só é preenchido após 24 horas sem classificação do email;

- *CountPhishing*: no campo *CountPhishing* (do tipo *Number*) é guardado o número de denúncias classificadas como *phishing*, isto é, denúncias que foram confirmadas pelo *RoboPHISH*;

- *CountSPAM*: no campo *CountSPAM* (do tipo *Number*) é guardado o número de denúncias classificadas como SPAM, isto é, denúncias que foram confirmadas pelo *RoboPHISH*;

- *Rating*: campo (do tipo *Number*) que contém o valor do *rating* de cada colaborador. Este campo corresponde ao valor da *accuracy* calculada a cada denúncia, através do algoritmo de *crowdsourcing*.

5.3 *RoboPHISH001 - 02 - Analyse Emails from Folder*

A implementação do segundo processo foi feita através de BP, recorrendo tanto a objetos já existentes como a objetos criados propositadamente para este processo. Neste processo o robô trabalha maioritariamente sobre as suas duas filas de trabalho, *emails* e *users*. Para além das interações com os elementos das filas de trabalho, o robô interage com o Outlook e o *Chrome* (*VirusTotal*, ARF e HPSM).

Em primeiro lugar, o robô acede à *queue* de emails, *RoboPHISH - EMAILS*, e extrai para a *Collection Pending Items* o identificador BP de todos os emails no estado pendente na *queue*. Este passo é feito através da ação *Get Pending Items* do BP. Depois de ter esta informação de todos os emails pendentes numa *Collection*, o robô vai iterar sobre cada um dos *items* da mesma. Primeiro é necessário extrair a informação de todos os campos da *queue* daquele item em particular. Isto é feito através da ação *Get Item Data*, que devolve para uma *Collection*, *Data from Queue*, os vários campos daquele item.

A primeira verificação feita corresponde ao estado do email, visto que este processo corresponde ao processo de análise inicial e queremos apenas os emails que foram extraídos no processo 01 (Secção 5.2) e que nunca foram analisados. Após a seleção dos emails, no estado 01 - *Email Saved in Folder*, o robô está então preparado para começar a analisar cada um deles.

Na *main page* do processo temos a estrutura das decisões, organizadas em diferentes *pages* (lógica semelhante à de funções). Cada uma das diferentes decisões de classificação fica numa página à parte, sendo que cada página é chamada pela *main page*.

5.3.1 *Verify Campaign*

A primeira *page* corresponde à verificação de email de campanha, *Verify Campaign*. Na Altice Portugal são feitos ataques simulados de *phishing*, fornecidos pela empresa CybeReady, que para além de analisar a quantidade de colaboradores que não reconheceram o ataque simulado, também proporciona formação aos mesmos. Estes ataques simulados servem como medida para avaliar se as formações e políticas *anti-phishing* estão a ser bem sucedidas, educando também os colaboradores que possam não estar tão sensibilizados (Secção 2.2.5). O processo tem acesso à lista de domínios da CybeReady, colocada em SQL, podendo assim excluir os emails que vêm dos mesmos. O robô transfere então a lista de domínios do SQL para uma *Collection* chamada *Whitelist*. É feito um *loop* sobre esta *Collection*, verificando em cada iteração se o domínio presente na *whitelist* é igual ao domínio do *sender* do email potencialmente malicioso. Assim que é encontrado um destes domínios na *Whitelist* o robô sai do ciclo e da *page*, adicionando ao campo *ScanCybeReady* da *Collection Data from Queue* a palavra *Campaign* e colocando uma *flag*, *CybeReady Flag*, a *True*.

5.3.2 *Verify Queue Classification*

A segunda *page*, *Verify Queue Classification*, corresponde à verificação de emails semelhantes já classificados na *queue*. A *queue* de emails vai mantendo informação sobre a classificação (*Phishing*, SPAM ou *Clean*) dos mesmos, sendo possível consultar estes itens posteriormente. A

ideia desta etapa é classificar mais rapidamente um email igual ou muito semelhante a um email já classificado anteriormente. Para isso o robô acede a todos os itens da *queue* dados como *Completed* através da ação *Get Completed Items*. Nesta ação podemos especificar vários filtros, entre eles o *Tag Filter*, desta forma podemos especificar os elementos que têm a *tag* de *Phishing* ou SPAM. Só é possível selecionar uma *tag* de cada vez, por isso temos de ter uma ação separada para o *Phishing* e outra para SPAM, *Get Completed Items Phish* (ilustrado na Figura 5.9) e *Get Completed Items SPAM*. Tudo o que é feito para os itens classificados como *phishing* é feito para os de SPAM, desta forma será demonstrado apenas o caso geral.

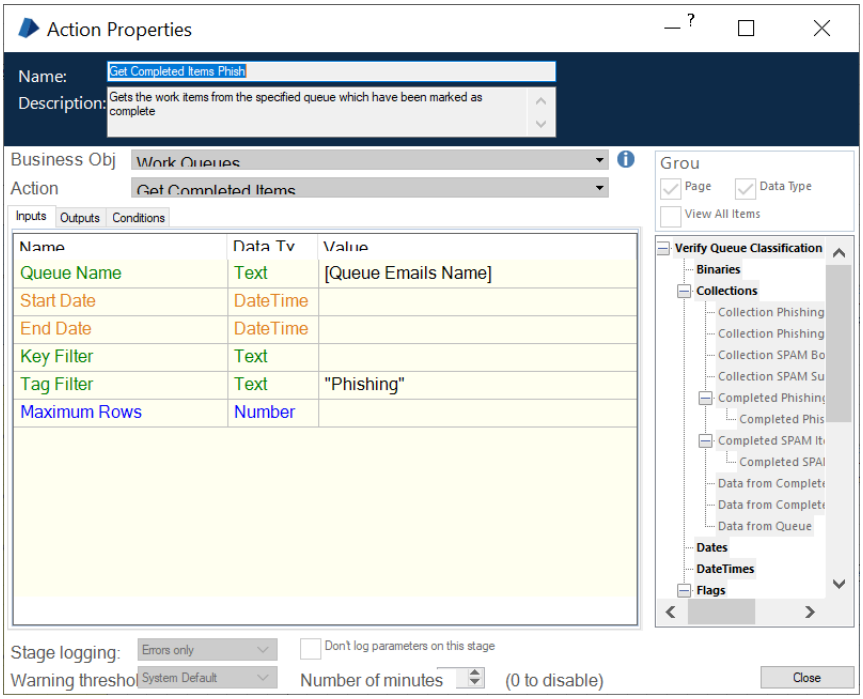


Figura 5.9: Ação *Get Completed Items* com a opção de *Phishing* selecionada

Tal como no caso dos *Pending Items* é devolvido um identificador BP daquele item na *queue*. Depois de obter a lista de identificadores, é possível iterar sobre a mesma e obter o conteúdo de cada item para a *Collection Data from Completed Queue*. Dentro de cada item é então feita a verificação de semelhança ao item atual, isto é, verificar se o *EmailSubject* do *Suspicious Email* é igual ao *EmailSubject* do email classificado anteriormente. Caso seja, é colocado o campo *ScanQueue* do email suspeito como *Phishing* ou SPAM, é colocada a *QueuePhishing Flag* ou a *Queue SPAM Flag* a *True*, o robô sai do ciclo e da página em questão. Caso contrário, o ciclo é percorrido até ao fim e o robô sai da *page* com ambas as *flags* a *False*.

5.3.3 *Verify Links VirusTotal*

A *page Verify Links VirusTotal* corresponde ao bloco mais complexo para o robô, isto é, o bloco que consome mais tempo, recursos e com mais probabilidade de falhar (levantamento exceção) Para a interação com o VirusTotal não existem objetos específicos do BP, portanto têm de ser criados objetos novos que interajam com esta página *web*. Foi então criado o objeto *RoboPHISH001 - VirusTotal* para fazer esta interação. Cada *page* dentro deste novo objeto corresponde a uma ação

que pode ser chamada no processo principal. Dentro deste objeto existem cinco ações principais: *Launch*, *Attach*, Pesquisa por URL, Voltar à *Homepage* e *Terminate*.

Antes de qualquer ação é necessário definir a aplicação com a qual este objeto vai interagir. Estes parâmetros são definidos no *Application Modeller* (apresentado na Figura 5.10). Neste caso temos de fornecer o caminho das pastas onde se encontra a aplicação do *Chrome*, `C:\Program Files (x86)\Google\Chrome\Application\chrome.exe`, e a página *web* que deverá ser aberta, `http s://www.virustotal.com/gui/home/url`.

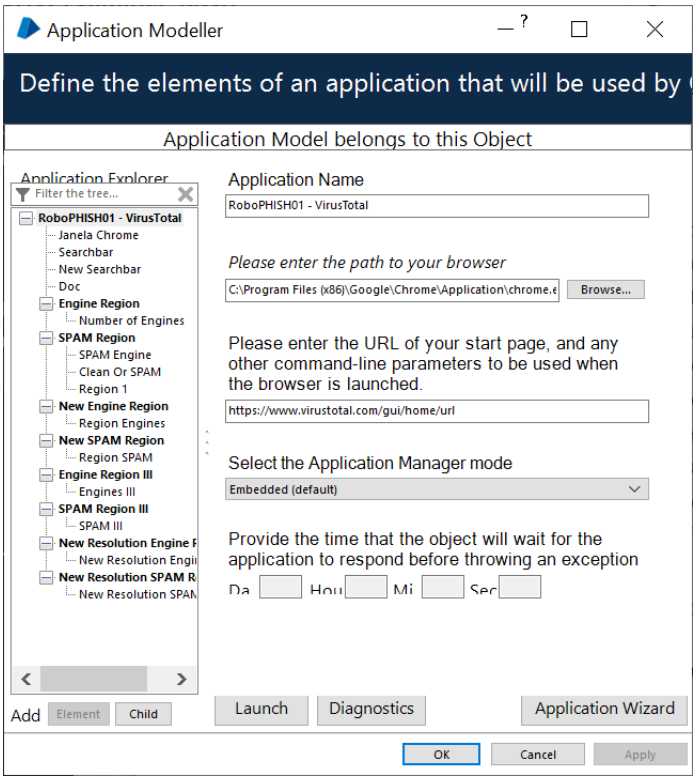


Figura 5.10: Parâmetros do *Application Modeller* do objeto que interage com o VirusTotal

Na primeira ação, *Launch*, o robô abre a aplicação *Chrome* na página do VirusTotal. Esta ação é a mais simples de todas, sendo que o robô já tem os parâmetros definidos no *Application Modeller*. É apenas necessário identificar na página do VirusTotal um elemento qualquer, para confirmar que a página foi aberta corretamente. As identificações dos elementos são todas feitas através do *Application Modeller* também, sendo que depois estes elementos são chamados na página do objeto. Neste caso foram escolhidos dois elementos, a janela que foi aberta e a *searchbar* do *Chrome*. Caso o robô não encontre pelo menos um destes elementos é lançada uma *Exception* (erro, levantamento de exceção).

A segunda ação trata-se do *Attach* e serve para ligar a página aberta ao robô evitando problemas de conexão ou de outras páginas abertas. Só é possível fazer *attach* do robô se houver apenas uma única página *web* aberta. Caso haja mais do que uma página *web* aberta, é lançada uma exceção e o programa é impedido de continuar.

A pesquisa por URL é a ação principal deste objeto e trata-se da pesquisa de um URL no VirusTotal. O desenvolvimento deste objeto teve vários desafios relacionados com esta pesquisa,

visto que certas funcionalidades do BP não foram possíveis de utilizar. O BP tem vários modos de identificar os diferentes elementos de uma página *web*. No caso da página *web* do VirusTotal o BP não consegue identificar os elementos HTML, sendo assim necessário recorrer ao método de identificação mais falível, *Region Mode*. Neste método o robô não identifica nenhum elemento da página, é apenas indicada uma região do ecrã onde deverá aparecer a informação que o robô deve recolher. Este método é sensível tanto a qualquer mudança de localização da informação na página, bem como à resolução do ecrã em questão.

Visto que não foi possível identificar a barra de pesquisa do VirusTotal, o robô tem de clicar na zona da página onde se encontra a mesma (para a selecionar) e enviar os caracteres do URL depois de clicar. As coordenadas (X,Y) indicadas para clicar na barra da página inicial são: (550,550). Depois de serem enviados os caracteres para a barra de pesquisa do VirusTotal, o robô deverá enviar um *Enter* para submeter o URL em questão. Visto que não é possível identificar os elementos da página de resultados, é necessário colocar um *Wait* (espera) de 15 segundos. Com esta espera é possível dar tempo suficiente à página de resultados para carregar todos os seus elementos. Ao fim dos 15 segundos o robô vai então retirar duas regiões necessárias, *Number of Engines* e *SPAM Field*, utilizando o *Region Mode* (Figura 5.11). A região *Number of Engines* corresponde à frase: *X engines detected this URL*, representando que X *engines* do VirusTotal consideraram aquele link *malicious* ou *phishing*. A região *SPAM Field* corresponde ao valor do primeiro *engine*, no caso da Figura 5.11 seria o valor do *engine* Certego - Malicious. Este valor é retirado porque caso o link seja apenas SPAM ou *suspicious* não é contabilizado no *Number of Engines*, aparece apenas na descrição dos *engines*. Por fim é retirado um terceiro campo, *URL from Searchbar*, que corresponde ao link de pesquisa no VirusTotal presente na barra de pesquisa do *Chrome*. Esta hiperligação para além de conter o *hash* (SHA-256) do link analisado, pode também ser utilizada para confirmação manual da classificação do robô.

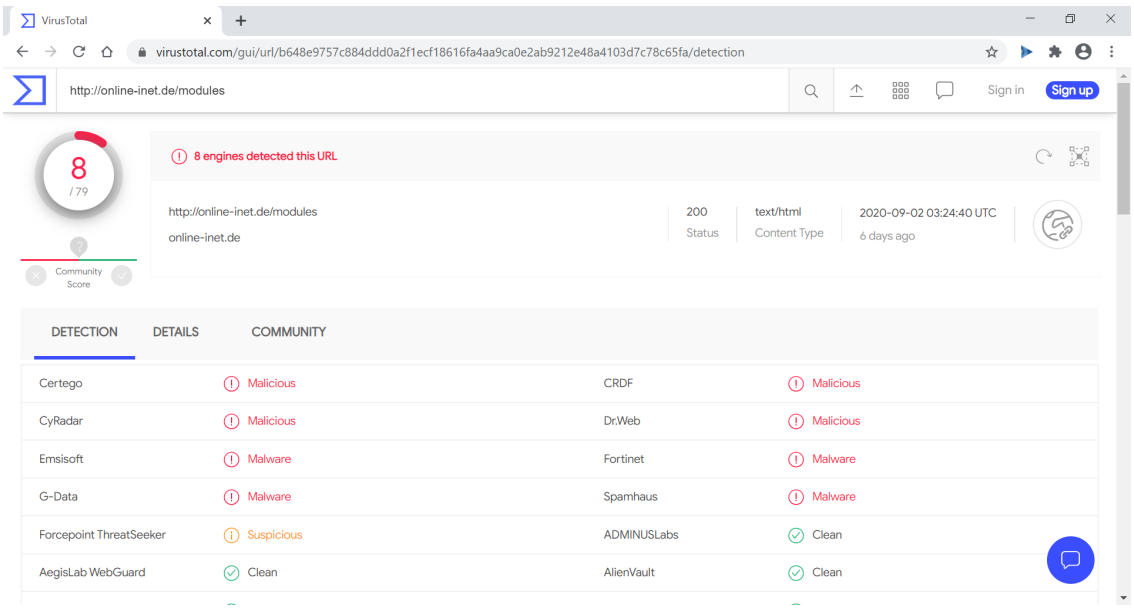


Figura 5.11: Exemplo de um ecrã do VirusTotal que mostra um URL classificado como *phishing*

A ação Voltar à *Homepage* foi criada para o robô voltar a aceder à página de pesquisa com o

objetivo de pesquisar novos links. Esta ação é bastante simples, consistindo apenas de escrever na barra de pesquisa do *Chrome* o endereço inicial: `https://www.virustotal.com/gui/home/url`. Após a escrita é enviado um *Enter* para submeter o link.

A ação final, *Terminate*, consiste do término da aplicação *Chrome*. O robô executa uma ação que termina o objeto aberto naquele *Application Modeller*.

Estas novas ações criadas no objeto *RoboPHISH001 - VirusTotal* são utilizadas no processo como ações normais do BP. Na página *Verify Link VirusTotal* o robô acede à *Collection* que contém os links, vinda do campo *VirusTotalLinks*, e adiciona três novos campos à mesma (*Number of Engines*, *SPAM Field*, *VirusTotal URL*). De seguida, o robô vai procurar se o link já se encontra na *blacklist* guardada em SQL. Esta *blacklist* contem links maliciosos, identificados previamente pelo robô. O robô transfere então para uma *Collection*, com o nome *Blacklist*, os links previamente classificados. Se alguma das hiperligações do *Suspicious Email* a ser analisado estiver nesta coleção, o robô sai do processo colocando a *flag VirusTotal Phishing* a *True* e colocando na *Collection VirusTotalLinks* o valor "Phishing".

Caso o robô não encontre nenhum dos links na *Blacklist* é necessário ir ao VirusTotal tentar classificar os mesmos. Em primeiro lugar o robô executa a ação *Launch* do objeto *RoboPHISH001 - VirusTotal*, deixando assim a janela do *Chrome* aberta na página de pesquisa por URL e fazendo logo de seguida o *Attach*. A partir daqui o robô itera sobre a coleção *VirusTotalLinks* e executa a ação Pesquisa por URL no VirusTotal para cada um destes links. Esta ação retorna três variáveis que, dependendo do seu valor, resultam numa determinada classificação (representada na Tabela 5.1).

URL Classification	*Number of Engines*	*SPAM Field*
CLEAN	*No engines detected this URL*	*Clean*
SPAM	*No engines detected this URL*	*SPAM/ Suspicious*
PHISHING	*(1 or more) engines detected this URL*	*Phishing/ Malicious*

Tabela 5.1: Classificação de links com base nas variáveis retornadas pelo VirusTotal

Consoante a classificação destes links, *Clean*, SPAM ou *Phishing*, poderá ser possível atribuir uma classificação ao email. No caso de haver pelo menos um link classificado como SPAM ou *Phishing* é colocada a *True* a *flag VirusTotal SPAM* ou *VirusTotal Phishing*, respetivamente. Para além de ser devolvida esta variável é também colocado o campo *VirusTotalLinks* com o valor de "Phishing"ou "SPAM". No caso da classificação de todos os links ser *Clean*, o processo corre até ao fim retornando ambas as *flags* a *False*.

5.3.4 *Verify SPAM icu*

A *page* seguinte, *Verify SPAM icu*, representa uma verificação do *top-level domain*[4] do link. Através da observação de vários emails de SPAM foi possível perceber que uma grande quantidade de emails de SPAM continha links com o *top-level domain* `.icu`. Após pesquisa interna, foi

[4] *Top-level domain* refere-se à parte mais à direita dos domínios, após o ponto (ex .com). É um dos domínios de nível mais alto no DNS [21].

possível chegar à conclusão que não haviam emails classificados como *Clean* que apresentassem este domínio. O objetivo desta página é verificar se algum dos domínios de nível mais alto dos links terminam em `.icu`, de modo a classificar o email como SPAM. As hiperligações estão a ser guardadas até ao primeiro subdomínio (de forma a não mandar bloquear domínios inteiros) logo é necessário trabalhar o link para obter o *top-level domain*. Para além desta modificação da *string* do link, o robô apenas tem de confirmar se o link modificado termina em `.icu`. Em caso afirmativo, robô deverá colocar a *True* a *Queue SPAM flag*, bem como colocar "SPAM"no campo *ScanQueue*. Caso contrário, a *page* executa até ao fim e a *flag* fica com o valor *False*.

5.3.5 *Verify Crowdsourcing*

Na *page Verify Crowdsourcing* todas as decisões são feitas através do algoritmo de *Crowdsourcing* (Subseção 4.3.3). Existem outras páginas que usam os resultados deste algoritmo, que estão separadas para facilitar a lógica do processo e porque foram desenvolvidas posteriormente. A ideia deste método de classificação é classificar como "pelo menos SPAM", os emails que foram enviados por mais de N pessoas e num curto espaço de tempo para o *RoboPHISH*. Dentro desta *page*, o robô vai trabalhar sobre o *Suspicious Email* que está a tentar classificar, comparando-o com os que tem por classificar neste momento. Por exemplo, quando é chamada esta *page* é passado o email que está a ser analisado neste momento (*Suspcicious Email*) e assim que o robô entra na mesma executa a ação *Get pending items* para ir buscar todos os itens que também estão por classificar (estando ou não no estado 01). Após ter estes itens o robô vai comparar o email atual (*Suspcicious Email*), a classificar, com estes emails pendentes. A comparação será feita com o corpo e assunto dos emails mas não se trata se uma comparação simples de igualdade. Para tentar colmatar os casos em que os emails são muito semelhantes, apenas com algumas variações nos nomes ou na escrita de algumas palavras, é utilizado um algoritmo que devolve a distância entre duas linhas de texto. O algoritmo escolhido, já existente numa ação BP, foi a distância de Jaro-Wrinkler[5]. Através desta ação é devolvido um valor (percentagem) de proximidade ou igualdade entre dois textos, sendo que zero são totalmente distintos e 100 são exatamente iguais. O valor desta semelhança é calculado entre o texto do email em classificação e o texto de cada um dos emails pendentes (*loop pending emails*). Se em algum dos casos esta semelhança for superior a 80% é considerado que estes emails são suficientemente semelhantes para serem iguais. Cada email "igual" encontrado é adicionado a uma nova *Collection*, *Equal Coll*, que ficará com todos os emails considerados iguais. Se esta coleção tiver mais de N emails (definido como dois para testes dentro da DCY) então são considerados "SPAM". De seguida é feita a verificação para saber se estes emails serão elevados a *phishing* ou não. Para este efeito o robô itera sobre a coleção *Equal Coll*, que contém apenas os emails iguais, e verifica se um ou mais foram denunciados por um *Expert* ou *Guru*. No caso, afirmativo são retirados os valores de *UserSuspected* (suspeita do utilizador) e contabilizados. Se dentro deste conjunto houverem mais *Gurus* e *Experts* que tenham denunciado este email como *phishing* então o email é classificado como "Phishing". Caso seja maior ou igual número de suspeitas de SPAM, então é classificado como "SPAM". No final são devolvidas as *flags CrowdSourcing SPAM Flag* e *CrowdSourcing Phishing Flag* a *True*, consoante

[5]Métrica de textos que mede a semelhança entre dois textos ou palavras em percentagem. [1].

a classificação do email.

5.3.6 *Verify Guru Vote*

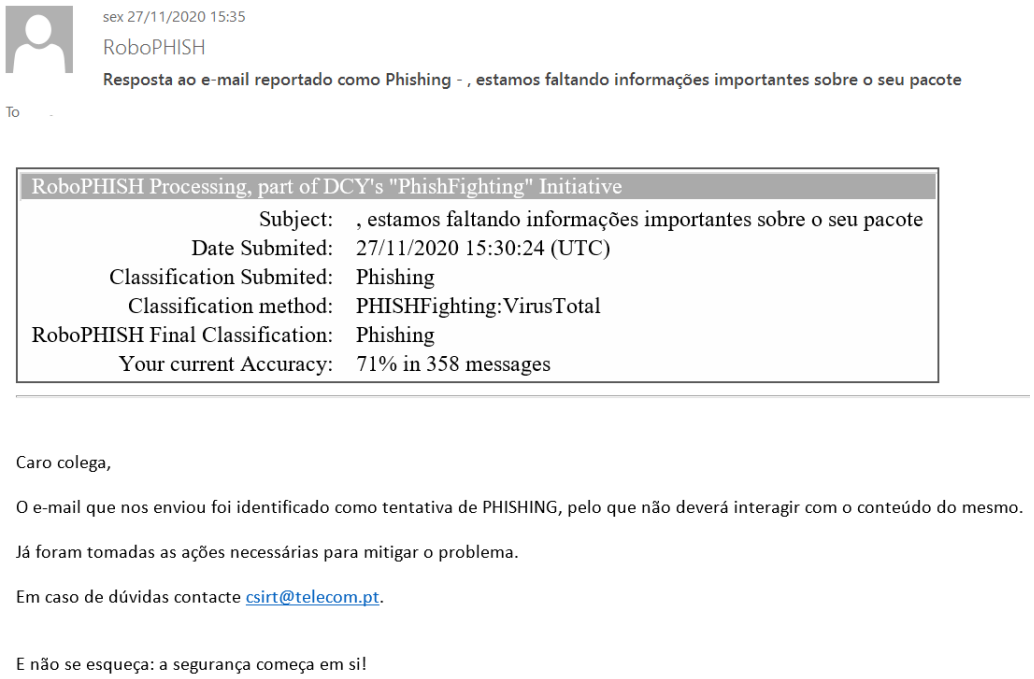
Por fim o robô tem a *page Verify Guru Vote* que mostra a decisão final, caso o emissor do *Report Email* seja *Guru*. Esta é a última etapa de classificação visto que o *Guru* é o decisor final, ou seja, apenas quando o robô não conseguiu classificar por nenhum dos outros métodos. No processo é verificado se o *SenderEmailAdress* corresponde a um utilizador da categoria *Guru*. Caso seja, é atribuída a classificação do campo *UserSuspected* (SPAM ou *Phishing*) ao *Suspicious Email*. Para atribuir esta classificação é colocada a *flag CrowdSourcing Phishing Flag* ou *CrowdSourcing SPAM Flag* a *True* e para além disso é preenchido o campo *ScanCrowdsourcing* da coleção com o valor adequado (”SPAM” ou ”Phishing”).

5.3.7 *Update Queue*

As *flags* devolvidas de cada *page* para a *main page* servem para atualizar as informações caso estas tenham o valor *True*. Em cada uma das decisões, sempre que pelo menos uma das *flags* estiver a *True*, são atualizadas as informações nas *queues* e é enviado um email ao colaborador. Deste modo, foram desenvolvidas mais duas *pages*: *Update Queue* e *Send Emails* que têm como objetivo guardar na *queue RoboPHISH - EMAILS* todas as atualizações feitas nas páginas anteriores e acabar de preencher os campos necessários. Sempre que um email é classificado, o campo que representa o modo como foi classificado é preenchido na respetiva *page* de classificação (por exemplo: campo *ScanQueue*). No entanto esta alteração é feita na coleção local e para esta ser submetida na *queue* é necessário executar, nesta *page*, um *Set Data*. Nesta *page* é também necessário preencher a *Tag* do item da *queue*, consoante a informação vinda da *flag* (”SPAM” ou ”Phishing”). Para além da *Tag*, é feita atualização do estado do item que passa do estado ”01 - *Email Saved in Folder*” para estar no estado ”03 - *Email Processed and Classified*”.

5.3.8 *Send Emails*

A *page Send Emails* serve para enviar os emails de resposta adequados aos colaboradores. Cada tipo de resposta requer um *template* de email definido para este efeito. A ideia é sempre avisar quais os cuidados a ter com emails maliciosos, agradecer a participação e informar o colaborador da sua *accuracy*. Conforme a classificação do *Suspicious Email*, é então enviado um email de resposta ao colaborador. Sempre que um colaborador denuncia, correta ou incorretamente, um email deverá receber um email de resposta. Todos os *templates* vêm com um cabeçalho inicial com algumas informações sobre o método, classificação do email e *accuracy* do colaborador (representado na Figura 5.12). O objetivo destes emails é passar alguma informação para incentivar mais denúncias por parte dos colaboradores. O texto dos *templates* do email foi escrito em HTML, e está guardado numa tabela SQL. O robô simplesmente analisa a classificação do *Suspicious Email* e seleciona o *template* adequado. Posteriormente verifica a forma como a classificação foi obtida, as datas de envio do *Suspicious Email* e vai buscar a *accuracy* do utilizador à *queue RoboPHISH - USERS*, substituindo depois estes campos no *template*.



sex 27/11/2020 15:35

RoboPHISH

Resposta ao e-mail reportado como Phishing - , estamos faltando informações importantes sobre o seu pacote

To

RoboPHISH Processing, part of DCY's "PhishFighting" Initiative	
Subject:	, estamos faltando informações importantes sobre o seu pacote
Date Submited:	27/11/2020 15:30:24 (UTC)
Classification Submited:	Phishing
Classification method:	PHISHFighting:VirusTotal
RoboPHISH Final Classification:	Phishing
Your current Accuracy:	71% in 358 messages

Caro colega,

O e-mail que nos enviou foi identificado como tentativa de PHISHING, pelo que não deverá interagir com o conteúdo do mesmo.

Já foram tomadas as ações necessárias para mitigar o problema.

Em caso de dúvidas contacte csirt@telecom.pt.

E não se esqueça: a segurança começa em si!

Figura 5.12: Exemplo de um *template* de resposta de email classificado como *phishing*

5.3.9 *Send to ARF Block Anubis*

Para além das duas *pages* supramencionadas, há alguns métodos de classificação que requerem bloqueios nas plataformas a montante e a jusante. As *pages Send to ARF Block Anubis* e *Send URL Block to HPSM* são chamadas após determinados métodos de classificação. Para o robô poder interagir com as duas plataformas de bloqueio foi necessária a criação de um objeto, *RoboPHISH001 - ARF Chrome*, e adição de novas ações a um objeto já existente, *RoboSOC - HPSM*.

No caso da *page Send to ARF Block Anubis*, o robô vai preencher um formulário para reportar um determinado email à Anubis. A Anubis quando receber esta indicação deverá tomar medidas para impedir a passagem de emails iguais. O formulário necessário encontra-se num endereço *web* da Altice e é acedido através do *Chrome*. Por conseguinte é necessário um objeto novo para tratar desta interação. Neste novo objeto, *RoboPHISH001 - ARF Chrome*, existem quatro ações: *Launch*, *Attach*, *Send Abuse Report to Anubis* e *Terminate*. As ações *Launch*, *Attach* e *Terminate*, tal como nos restantes objetos criados servem para lançar o *Chrome* na página necessária, fazer a ligação da página ao robô e terminar a janela do *Chrome* aberta. A ação *Send Abuse Report to Anubis* tem como objetivo o preenchimento do formulário por parte do robô. Primeiramente o robô vai à pasta (passada como variável) e anexa ao formulário o ficheiro a reportar, através do nome do ficheiro existente na *queue* (campo *AttachmentID*). Posteriormente preenche os campos do formulário: *Submit Report To*, *to*, *cc*[6] e *bcc*[7]. O campo *Message Body* já vem parcialmente preenchido e quando é feita a submissão do formulário, são substituídos alguns valores automaticamente (Figura 5.13). Por fim, o robô carrega no botão *Submit* de forma a enviar o formulário

[6] *Carbon Copy*: Enviar para lista de destinatários visível

[7] *Blind Carbon Copy*: Enviar para lista de destinatários oculta

para a Anubis, enviando também uma cópia para o email do SOC e do *RoboPHISH*.

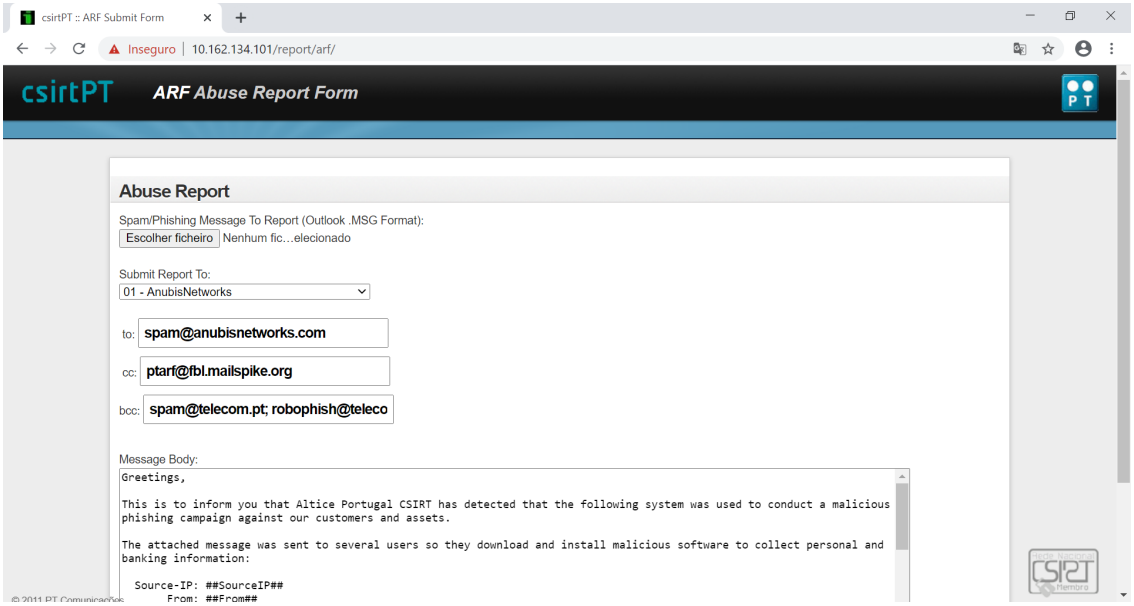


Figura 5.13: Exemplo de um formulário ARF de denúncia de email

Em primeiro lugar, na *page* do processo, é necessário verificar se o *Suspicious Email* em questão está no formato *.msg* ou *.eml*, visto que o ARF só aceita ficheiros com a extensão *.msg*. Caso o ficheiro tenha a extensão *.eml*, é necessário converter para *.msg*. Para este efeito foi criada uma ação, *Convert eml to msg* no objeto *RoboPHISH001 - Outlook*. Esta ação abre o ficheiro *.eml* e guarda-o como *.msg* numa pasta dedicada a este efeito. Após ter o ficheiro no formato correto (*.msg*), o robô acede à página de submissão de denúncia de emails para a Anubis, através da ação *Launch* do objeto criado. Após lançar a página é feito o *Attach* da mesma. De seguida, é invocada a ação principal, *Send Abuse Report to Anubis*, onde é feito o preenchimento e submissão do formulário.

Por fim é necessário guardar a informação do bloqueio que foi efetuado. Para este efeito, foi criada uma tabela SQL *RoboPHISH_ARF_tickets* onde estão os dados recolhidas durante a execução do processo. Nesta tabela são guardadas as seguintes informações: *ID* da mensagem (*MSG_KeyValue*) e data de criação do formulário ARF (*ANUBIS_CreateDT*).

5.3.10 *Send URL Block to HPSM*

Para facilitar o bloqueio de URLs no Zscaler, é necessário recorrer ao sistema de *tickets* da Altice, HPSM. Este processo ocorre na *page*, *Send URL Block to HPSM*, que chama as ações do objeto *RoboSOC - HPSM*. As ações *Launch and Login*, *Attach*, *Logout* e *Close* já existiam para outros *tickets* de outros processos. Foi apenas necessário criar as ações *Click Incident Management*, *Apply Incident Template*, *Fill Ticket Description* e *Submit Ticket*. A primeira nova *page* criada, *Click Incident Management*, serve apenas para carregar no menu *Incident Management* e de seguida selecionar *Open New Incident* dentro do mesmo. A *page Apply Incident Template* recebe uma variável de texto com o nome do *template* a selecionar. De seguida, o robô carrega no botão *Apply Template* e escolhe de uma lista de *templates* o que foi passado na variável. Por

fim a última página criada, *Submit Ticket*, serve apenas para carregar *Enter* de forma a submeter o formulário.

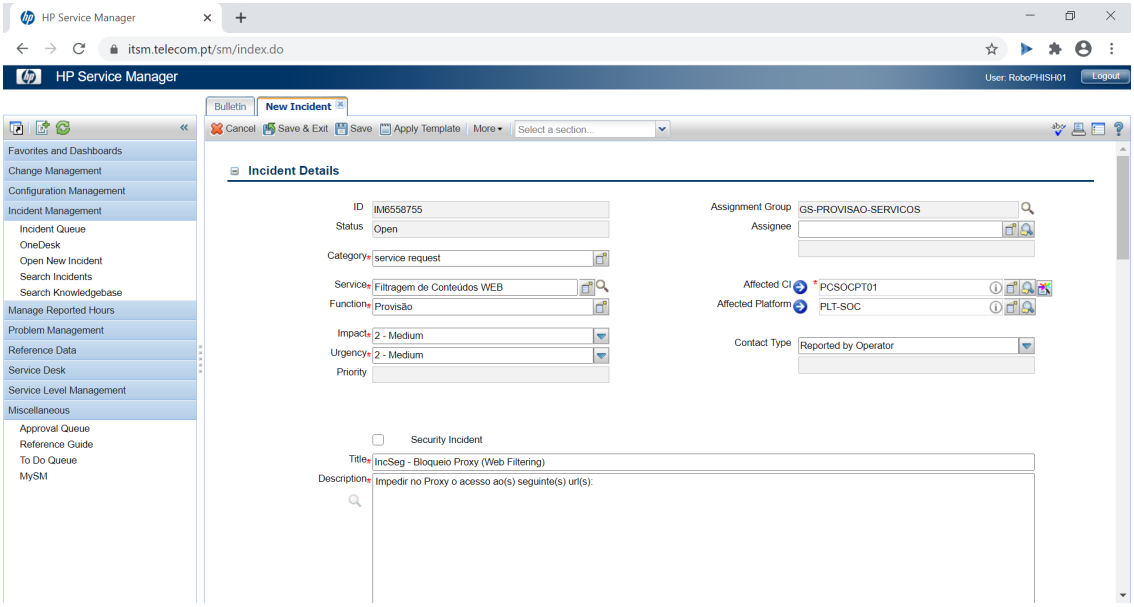


Figura 5.14: Exemplo de *ticket* HPSM de bloqueio de URL

Antes de serem chamadas as novas ações, é nesta página que é adicionado o URL à *blacklist* SQL. O robô acede então à tabela (representada na Figura 5.15) e executa a seguinte *query* para devolver os URLs iguais:

*select * from TESTE.dbo.RoboPHISH_URL_blacklist where URL = '%URL%';*

Se a *query* retornar um valor diferente de NULL, isto significa que este URL já existe na *blacklist* e por esse motivo, não é necessário adicionar novamente. Embora este URL já esteja na *blacklist*, existe a possibilidade de não ter sido aberto um *ticket* no HPSM. Deste modo é então verificado se o campo *HPSM_ID* da *blacklist* está a NULL. Caso esteja este URL terá de ser enviado para bloqueio e é então adicionado à *Collection Links to Send*. Caso a *query* inicial tenha retornado o valor NULL são adicionados os campos *URL*, *Engines*, *VTClassification*, *ScanDt*, *#msg* e *URL_HASH* à tabela através da seguinte *query*:

insert into TESTE.dbo.RoboPHISH_URL_blacklist (URL, Engines, VTClassification, ScanDt, #msg, HPSM_ID, HPSM_CreateDT, HPSM_CloseDT, ARF_ID, ANUBIS_CreateDT, ANUBIS_CloseDT, URL_HASH) values ('%URL%', '%ENGINES%', '%VT%', '%DATE%', '%MSG%', NULL, NULL, NULL, NULL, NULL, NULL, '%hash%');

Os valores são substituídos pelas variáveis correspondentes que estão na *Collection Data from Queue*. Para além de executar a *query* o robô deverá guardar cada um destes URLs na *Collection Links to Send*. De seguida, o robô abre um *ticket* HPSM para cada um dos URLs presentes na *Collection Links to Send*. Isto é feito através das ações criadas no objeto descrito anteriormente, *RoboSOC - HPSM*. A primeira ação é a *Launch and Login* no HPSM, onde são passados os valores *userame* e *password*. De seguida é feito o *Attach* do robô à janela HPSM, aberta pela ação anterior.

De seguida é chamada a ação *Apply Incident Template*, passando a variável *IncSeg - Bloqueio Proxy (Web Filtering)*, que corresponde ao nome do *template* a selecionar. Após esta ação são chamadas as ações *Fill Ticket Description*, *Submit Ticket*, *Logout* e *Close*, respetivamente. Depois da abertura do *ticket*, é necessário atualizar a informação na URL *blacklist*. Uma nova *query* é feita, desta vez para ir buscar o URL, em particular o que tem o campo *HPSM_ID* a NULL. De seguida é necessário atualizar este valor e a data de criação, *HPMD_CreateDT*.

Results | Messages

	URL	Engines	VTclassification	ScanDt	#msg	HPSM_ID	HPSM_CreateDT	HPSM_CloseDT
1	http://www.tobulb.buzz/kxgqn	2 engines detected this URL	Phishing	2020-10-09 16:33:48.000	00000000...	IM6583480	2020-10-09 16:36:04.000	2020-10-09 18:02:31.000
2	https://safeairyes.com/trk	1 engines detected this URL	Phishing	2020-10-06 12:33:09.000	00000000...	IM6573098	2020-10-06 12:35:11.000	NULL
3	http://www.pumpitem.buzz/krxmalggcs	1 engines detected this URL	Phishing	2020-10-06 12:28:49.000	00000000...	IM6573091	2020-10-06 12:30:53.000	NULL
4	http://www.slicesweep.cyou/rkmajo	1 engines detected this URL	Phishing	2020-10-06 12:22:08.000	00000000...	NULL	NULL	NULL
5	http://www.Marsspot.cyou/kjlfobk	1 engines detected this URL	Phishing	2020-10-06 11:58:51.000	00000000...	NULL	NULL	NULL
6	http://www.sweepmajor.buzz/Lldude	1 engines detected this URL	Phishing	2020-10-03 14:55:41.000	00000000...	NULL	NULL	NULL
7	https://kinomeja.com/trk	1 engines detected this URL	Phishing	2020-10-02 10:43:24.000	00000000...	IM6566186	2020-10-02 10:45:27.000	2020-10-02 12:09:46.000
8	http://www.danceessay.cyou/pxehwtpnw	1 engines detected this URL	Phishing	2020-10-01 17:16:53.000	00000000...	NULL	NULL	NULL
9	http://sgtr.eomail6.com/ls	1 engines detected this URL	Phishing	2020-10-01 16:56:41.000	00000000...	IM6564444	2020-10-01 16:58:48.000	2020-10-02 09:39:40.000
10	http://portal.docstoreinternal.net/?	3 engines detected this URL	Phishing	2020-09-29 12:32:44.000	00000000...	IM6558141	2020-09-29 12:34:51.000	2020-09-09 14:52:50.000

Figura 5.15: Tabela SQL *blacklist* com os campos do HPSM

5.3.11 *No Classification Found Yet*

Falta apenas mencionar a condição final, isto é, na eventualidade do robô não ter devolvido a *True* nenhuma das *flags* das decisões de classificação do email. Neste cenário é enviado um email inicial ao colaborador e são atualizados valores das *queue RoboPHISH - EMAILS* com a informação recolhida. O *template* do email para o utilizador agradece a denúncia e refere que será dado *feedback* assim que possível. As atualizações dos valores da *queue* servem maioritariamente para o campo *VirusTotalLinks* que terá os seus campos com o valor *Clean* e para atualizar o estado que passa a "02 - *Email Processed and Not Classified*".

5.4 *RoboPHISH001 - 03 - Scan Not Classified*

Este processo, *Scan Not Classified*, serve para tentar classificar os emails que não conseguiram classificação inicial. Em primeiro lugar, o robô obtém os *items* pendentes da *queue RoboPHISH - EMAILS*, através da ação *Get Pending Items*. De seguida é necessário ver qual destes se encontram no estado "02 - *Email Processed and Not Classified*", visto que só estes não foram classificados no processo anterior (Secção 5.3). Após a recolha dos itens pendentes, no estado pretendido, para uma *Collection* o robô vai iterar sobre a mesma para analisar os emails novamente. As duas primeiras verificações consistem nas condições de paragem do ciclo, que são:

- Primeira condição: email enviado por *Expert* e por classificar há mais de quatro horas;
- Segunda condição: email por classificar há mais de 24 horas.

Para verificar a primeira condição é necessário aceder à *queue RoboPHISH - USERS* e ver se o emissor do email está nesta *queue* com a categoria *Expert*. No caso afirmativo, é necessário recolher a data e hora atual, através de uma ação do BP para este efeito. De seguida as datas (do email e atual) são comparadas através da seguinte expressão:

```
[ReceivedOn DateTime] <= [Current DateTime] AND
[Current DateTime] <= [ReceivedOn + SOCTimeSpan]
```

A variável *ReceivedOn DateTime* diz respeito ao valor retirado da *queue RoboPHISH -EMAILS* e corresponde à data em que a denúncia de *Suspicious Email* chegou à caixa de correio do *RoboPHISH*. A *Current DateTime* foi a data atual recolhida no processo e a *ReceivedOn + SOCTimeSpan* (data recepção + quatro horas) corresponde ao número de horas necessárias para enviar o email ao SOC. Se o resultado da expressão for *True*, isso significa que é necessário enviar o email para o SOC analisar manualmente e classificar. O email segue então em anexo para o endereço de email do SOC com a informação que o robô, através dos métodos de classificação desenvolvidos, não foi capaz de categorizar o email e com o pedido de análise manual.

A segunda condição de paragem verifica se já passaram 24 horas de modo a classificar este email como *Clean* e dar uma resposta ao utilizador. Para este efeito é necessário verificar a seguinte expressão:

```
[ReceivedOn DateTime] <= [Current DateTime] AND
[Current DateTime] <= [ReceivedOn + CleanTimeSpan]
```

As variáveis são as mesmas usadas na expressão anterior excepto a *ReceivedOn + CleanTimeSpan* (data recepção + 24 horas) que corresponde ao número de horas necessárias para classificar um email como *Clean*. Caso esta expressão devolva *True*, é fundamental classificar o email como *Clean* na *queue* e enviar uma resposta adequada ao colaborador. Na resposta enviada, o *template* não menciona especificamente que o email não apresenta qualquer ameaça, dizendo apenas que não foi possível classificar o mesmo mas se o email não foi solicitado o colaborador não deverá interagir com o mesmo.

Caso o robô não pare em nenhuma das condições anteriores (ambas a *False*) então é necessário fazer a verificação final. Esta consiste em saber se, entretanto, houve alguma atualização em relação à classificação de algum email igual ou semelhante na *queue RoboPHISH - EMAILS*. A nova análise consiste em ir verificar à *queue* se há emails iguais já classificados. A *page* criada para este efeito, *Verify Queue Classification*, é igual à *page* com o mesmo no nome do processo da Subseção 5.3.2. No caso afirmativo, são tomadas as mesmas ações que na página da Subseção 5.3.2, ou seja, é enviado um email de resposta ao colaborador (*Send Emails*) e a *queue RoboPHISH - EMAILS* é atualizada (*Update Queue*) .

5.5 Algoritmo de *Crowdsourcing*

A implementação do algoritmo foi feita através de uma combinação de SQL com RPA. A pontuação de cada utilizador, *Accuracy*, é calculada através de SQL mas as decisões do algoritmo em si são feitas no BP.

Em SQL são mantidas as tabelas com toda a informação necessária ao algoritmo. A primeira tabela relevante corresponde à pontuação atribuída a cada colaborador quando este erra ou acerta uma denúncia (Figura 5.16).

Results | Messages

	TAG	UserSuspected	Points
1	CAMPAIGN	Phishing	10
2	CAMPAIGN	SPAM	10
3	Phishing	Phishing	10
4	Phishing	SPAM	10
5	SPAM	Phishing	10
6	SPAM	SPAM	10
7	Clean	Phishing	0
8	Clean	SPAM	0

Figura 5.16: Tabela SQL de pontos atribuídos aos utilizadores do sistema

Para além da tabela de pontuação é mantida uma tabela de classificações atribuídas a cada colaborador, com base no intervalo definido de *accuracy* (Figura 5.17).

Results | Messages

	id	tag	point_mim	point_max
1	81	Guru	75	100
2	82	Expert	50	75
3	83	Informed	1	50
4	84	No Tag Yet	0	0

Figura 5.17: Tabela SQL de categorias atribuídas aos utilizadores do sistema

Para atualizar a pontuação e categoria de cada utilizador é corrido um *script* SQL de hora a hora. Este *script* vai a todos os elementos da *Users Queue*, passada para SQL, e atualiza o valor da *accuracy* de cada um deles. Este valor é atualizado com base nos valores guardados nos campos *CountReports* (*Users Queue*) e no campo *tag* (*Emails Queue*) e guardado no campo *Rating* (*Users Queue*). Para além deste valor é também atualizada a categoria dos utilizadores, que pode ter sido alterada devido ao novo valor de *accuracy*. A atualização da categoria é guardada no campo *tag* (Users Queue).

Quando um colaborador faz uma denúncia, correta ou incorreta, a sua *accuracy* e categoria podem ser atualizadas com um atrasado de uma hora no máximo. Mantendo sempre esta tabela atualizada é possível então tomar as decisões necessárias com base nestes valores. Toda a parte das decisões é feita no BP, nos processos detalhados anteriormente. A classificação dos utilizadores ajuda o robô a tomar decisões em três circunstâncias, sendo duas delas no segundo processo (*Analyse Emails from Folder*) e uma delas no terceiro (*Scan Not Classified*).

5.5.1 Decisão *Crowdsourcing*

Dentro do segundo processo a Subseção 5.3.5, *Verify Crowdsourcing*, corresponde à análise de vários emails semelhantes, enviados num espaço de tempo relativamente curto para classificar como, no mínimo, "SPAM". Inicialmente não é necessário ter em conta qualquer categoria dos colaboradores, contudo adotando a abordagem *crowdsourcing* é considerada a opinião dos mesmos para que seja tomada uma decisão. Na segunda componente da decisão (elevar ou não a *phishing*) as classificações dos colaboradores são consideradas. Para avaliar se o email deverá ser considerado "*Phishing*", são retiradas as classificações apenas dos colaboradores com categoria *Guru* ou *Expert*. Caso o número *phishings* seja superior que o de SPAMs, neste subconjunto, então o email é classificado como *phishing* (Figura 5.18).

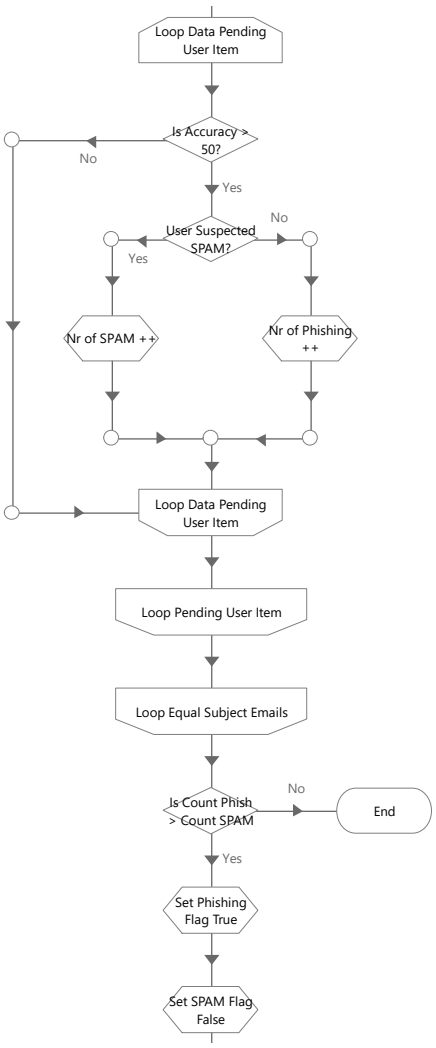


Figura 5.18: Bloco de código BP que classifica o email através do *crowdsourcing*

5.5.2 Decisão *Guru Vote*

O segundo processo, *RoboPHISH001 - 02 - Analyse Emails from Folder* (Secção 5.3), contém mais uma decisão resultante do algoritmo de *crowdsourcing*, Subseção 5.3.5. A *accuracy* do colaborador leva à atribuição de determinadas categorias, dentro destas o *Guru* que é o decisor final. Caso não tenha sido possível classificar o email, a decisão final consiste em verificar se a denúncia foi feita por um *Guru*. Em caso afirmativo, é colocado o *UserSuspected* (”SPAM” ou ”*Phishing*”) como classificação final do email (Figura 5.19).

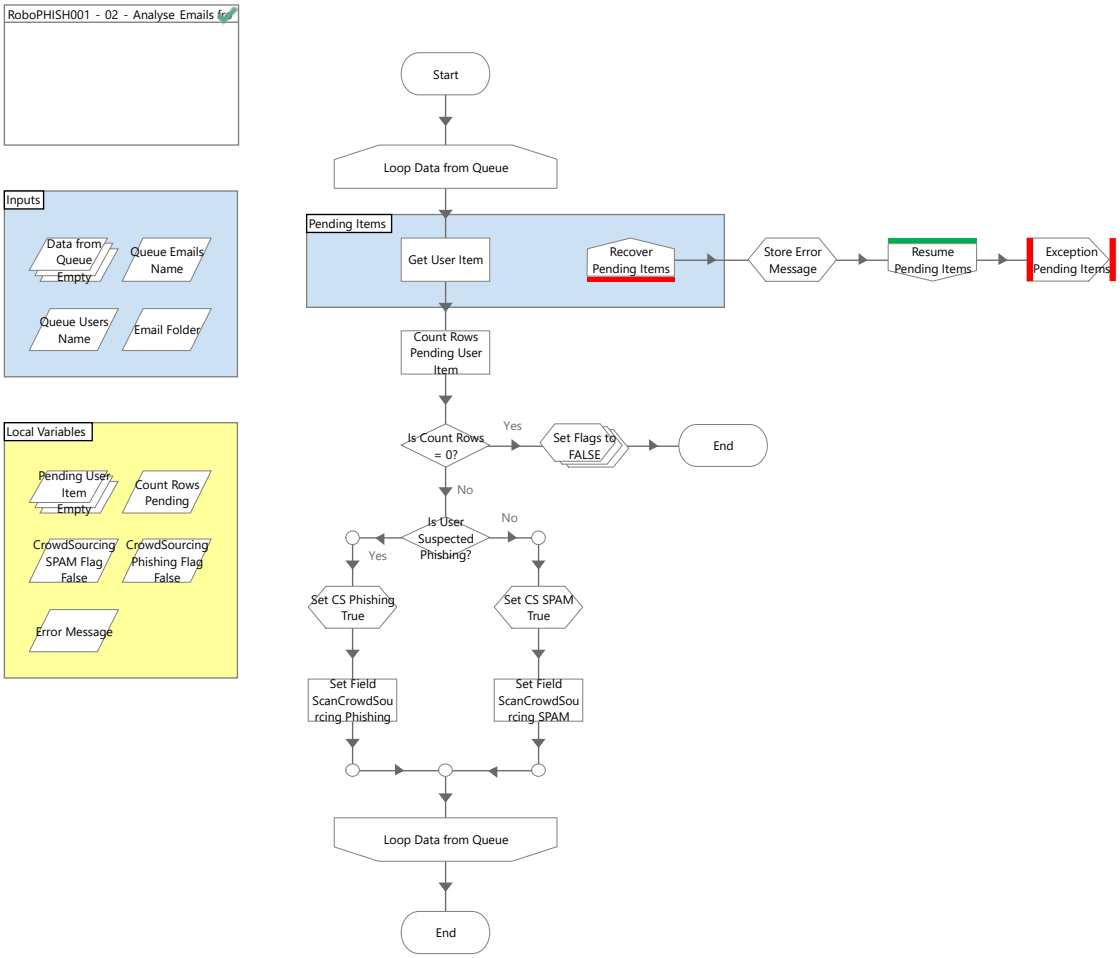


Figura 5.19: Bloco de código BP que classifica o email através do *GuruVote*

5.5.3 Decisão *Expert*

Na Secção 5.4, onde está presente o processo que tenta classificar os emails que não foram classificados anteriormente, a condição relativa ao algoritmo diz respeito apenas aos emails não classificados há mais de quatro horas. O robô verifica se os emails não classificados há quatro ou mais horas, foram enviados por um colaborador com categoria *Expert*. Em caso afirmativo, os emails são enviados para o SOC analisar (Figura 5.20).

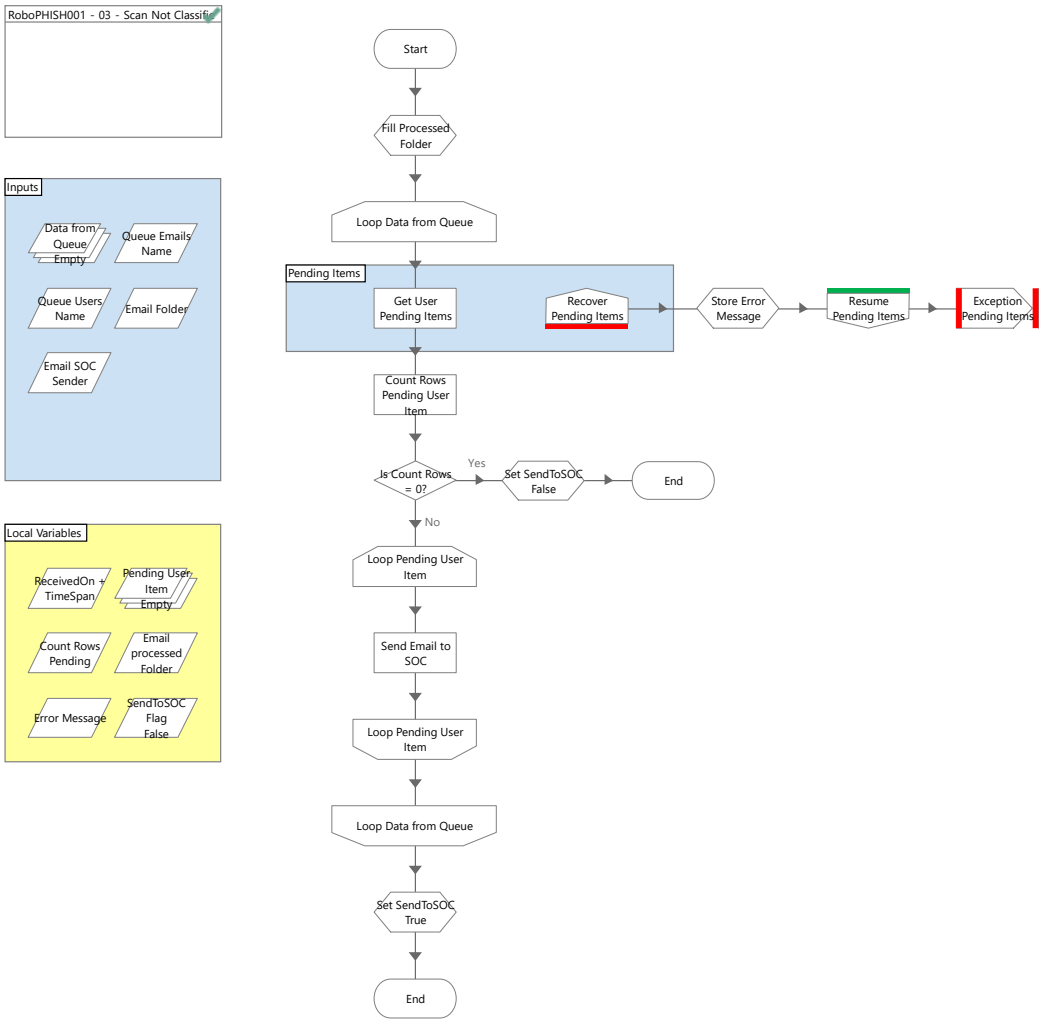


Figura 5.20: Bloco de código BP que email para o SOC através da denúncia do *Expert*

Capítulo 6

Resultados

Neste capítulo é apresentada toda a extração, investigação e análise de resultados do sistema *RoboPHISH*. Para tal são extraídas as métricas possíveis para avaliar este projeto. Visto que existem várias componentes do processo é possível recolher métricas individuais ou avaliar o projeto como um todo. As métricas mais gerais dizem respeito ao tempo médio de execução de cada processo e percentagem de erro. As métricas mais específicas dos casos de uso deste sistema dizem respeito ao número de denúncias por semana, número de utilizadores ativos, classificação de emails e classificação de utilizadores.

6.1 Ambiente de produção

Os botões de denúncia foram instalados em todos os colaboradores DCY, desta forma todos os testes foram realizados dentro desta população. O desenvolvimento dos processos foi feito em parte numa máquina local da Altice Portugal e em parte num servidor único na Covilhã. Quando os processos *RoboPHISH* foram colocados em produção, foram executados apenas no servidor da Covilhã. Os processos correm com *schedules* (agendamentos) definidos da seguinte forma: *RoboPHISH001 - 01 - Save Emails to Folder and Queue* e *RoboPHISH001 - 02 - Analyse Emails from Folder* correm de cinco em cinco minutos enquanto que o *RoboPHISH001 - 03 - Scan Not Classified* corre de uma em uma hora. Toda a informação das *queues* do BP é passada para tabelas SQL, através do processo BP *RoboSOC000 - Feed All Queues*, sendo assim possível trabalhar sobre as mesmas para retirar as métricas adequadas.

6.2 Percentagem de erros de execução

A primeira métrica selecionada para avaliar o sucesso do sistema foi a percentagem de erros de execução dos processos BP. Visto que toda a parte de análise, resposta e bloqueios do sistema é feita através do BP, é importante perceber se os processos do sistema apresentam um número baixo de erros. É de notar que estes erros são excepções lançadas pelo próprio BP (ações que interrompem o programa e lançam uma exceção). Estas exceções dizem respeito a falhas na implementação dos robôs em si, são situações de exceção não recuperáveis, que atingiram o número máximo de *retries*. Para calcular a taxa de erros a fórmula usada é a seguinte:

`Número de execuções com exceção/Número de total de execuções`

Pelo contrário temos a taxa de execuções bem sucedidas com a seguinte expressão:

`Número de execuções bem sucedidas/Número de total de execuções`

Em ambos os casos apenas é necessário multiplicar por 100 para obter as percentagens. Os valores necessários foram retirados das *queries* feitas sobre as filas de trabalho guardadas no SQL e dizem respeito ao período de tempo desde que o processo em questão foi colocado em produção até 26 de Novembro de 2020.

6.2.1 *RoboPHISH001 - 01 - Add Emails to Folder and Queue*

O primeiro processo foi colocado em produção no *bot* cinco a 4 de Abril de 2020. Para um total de 45211 execuções feitas no espaço de tempo indicado obtivemos 45036 que terminaram com sucesso e 175 que terminaram com exceção (erro). Através destes valores é possível calcular a taxa de erro, seguindo a fórmula descrita, chegando ao valor de 0.0039 (arredondado às quatro casas decimais). Convertendo para percentagem temos um valor de 0.39%. Pelo contrário temos uma percentagem de sucesso de 0.9961 (arredondado às quatro casas decimais), correspondendo a uma percentagem de 99,61%. Através destes valores é possível afirmar que a percentagem de erro foi bastante baixa, sendo este um parâmetro importante para avaliar a qualidade do sistema.

6.2.2 *RoboPHISH001 - 02 - Analyse Emails from Folder*

O segundo processo foi colocado em produção no *bot* cinco a 28 de Março de 2020. Esta data é inferior à do primeiro processo porque durante um período de tempo os processos foram trocados de *bots* devido a problemas no *bot* em si. Anteriormente o processo 01 estava a ser executado no *bot* quatro. Para simplificar a análise, visto que esta transição de *bots* não foi feita ao mesmo tempo, foram contadas as execuções feitas a partir do momento em que os processos foram colocados no *bot* cinco. Dentro das 47690 execuções do processo, 44434 foram bem sucedidas e 3256 resultaram em exceção (erro). A taxa de erro, calculada através destes valores, corresponde a 0.0682 (arredondado às quatro casas decimais), correspondendo a uma percentagem de 6,82%. Pelo contrário, a taxa de sucesso é 0.9317 (arredondado às quatro casas decimais), correspondendo a uma percentagem de 93,17%. Este processo apresenta a maior percentagem de erro visto que foi o que apresentou mais dificuldades, em particular na interação do robô com o VirusTotal (*website*). As dificuldades foram superadas, pelo que neste momento a percentagem de erro será muito menor.

6.2.3 *RoboPHISH001 - 03 - Analyse Emails from Folder*

O terceiro processo sendo o último a ser desenvolvido e colocado em produção, foi executado apenas no *bot* cinco. O processo foi executado 7025 vezes desde que foi colocado em produção a 21 de Maio de 2020. Dentro do total de execuções 7000 foram completas com sucesso e 25 com exceção (erro). Através destes valores é possível calcular a taxa de erro, obtendo um valor de 0.0036 (arredondado às quatro casas decimais). Convertendo para percentagem o valor obtido

é 0.36%, considera-se uma percentagem baixa. Em relação à taxa de sucesso o valor é 0.9964, convertendo para percentagem 99,64 %.

6.3 Tempo médio de execução do processo

Para cada processo é possível isolar o tempo médio de quando este é executado com e sem erro. Tipicamente as execuções com exceção são mais demoradas visto que muitas vezes há um N número de tentativas até a exceção ser lançada. Para o primeiro processo, *RoboPHISH001 - 01 - Add Emails to Folder and Queue*, o tempo médio de execução normal é 11 segundos sendo que quando existem exceções passa a 208 segundos. Para o segundo processo, *RoboPHISH001 - 02 - Analyse Emails from Folder*, o tempo médio de execução sem erros é 10 segundos, passando para 67 segundos na presença de exceções. Para o terceiro processo, *RoboPHISH001 - 03 - Analyse Emails from Folder*, o tempo médio de processamento varia entre 47 (sem erros) e 162 (com erros).

A análise destes valores permite concluir que os processos foram executados em tempo aceitável. É importante que a classificação seja feita o mais rapidamente possível de forma a permitir que o robô classifique o maior número de emails. É de notar que estes valores são para as denúncias de um grupo de testes restrito, os colaboradores DCY, e que ao fazer o *rollout* do sistema para uma população maior é esperado que os valores subam.

6.4 Número de denúncias por semana

O número de denúncias por semana é uma métrica relevante para ser acompanhada ao longo do tempo. Podem haver algumas razões para uma diminuição ou aumento deste valor. Algumas das causas possíveis para a diminuição de denúncias:

- O sistema *RoboPHISH* está a cumprir o seu objetivo, fazendo os bloqueios necessários a montante, e desta forma os emails maliciosos deixam de chegar aos colaboradores;
- A quantidade de SPAM/ *phishing* recebido pelos colaboradores diminui (logo estes têm menos emails para denunciar);
- Os utilizadores deixaram de usar o sistema porque acharam pouco eficiente, aplicável ou útil.

Em relação às possíveis causas para o aumento das denúncias do sistema, temos o contrário:

- O sistema *RoboPHISH* não está a cumprir o seu objetivo, desta forma continuam a chegar muitos emails maliciosos que não estão a ser bloqueados;
- O número de SPAM/ *phishing* recebido pelos colaboradores aumentou devido a um aumento da quantidade de emails enviados pelos agentes maliciosos;
- Os utilizadores estão a usar o sistema mais frequentemente porque o consideram eficiente, aplicável ou útil.

Para acompanhar esta evolução do número de denúncias foi desenvolvido um gráfico no *Tableau* [1] utilizando como fonte de dados a *queue RoboPHISH - EMAILS* retirada do SQL. O gráfico representado na Figura 6.1 mostra o número total de denúncias (*reports*) feitos por semana.

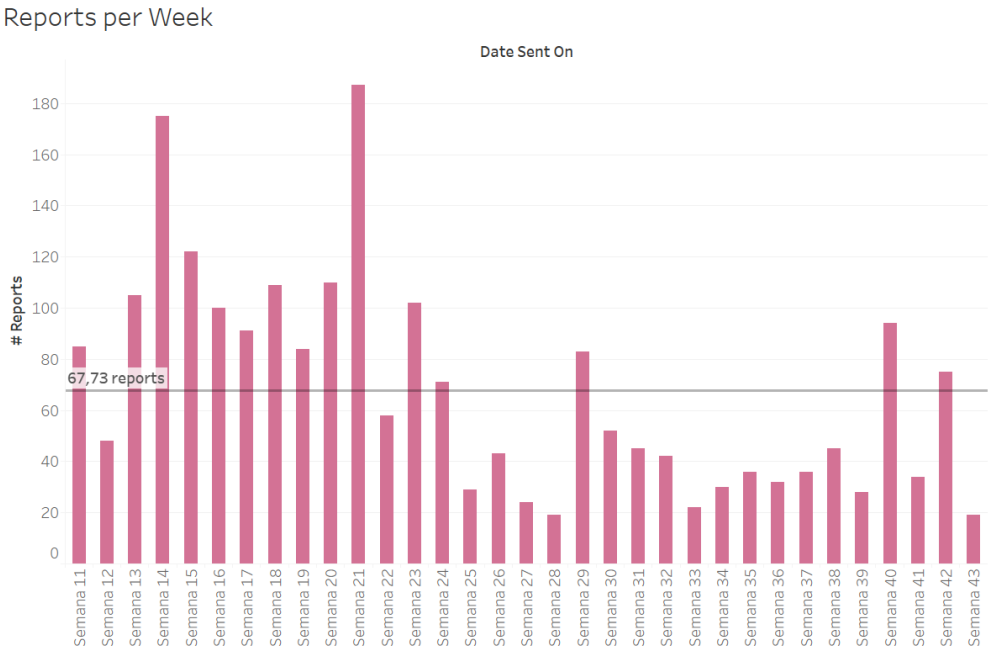


Figura 6.1: Número de denúncias (*reports*) por semana

Através da observação do gráfico é possível chegar ao valor médio de denúncias (*reports*) feitos por semana, aproximadamente 68 denúncias. É possível também concluir que nas primeiras semanas a utilização esteve maioritariamente acima desta média, ocorrendo grandes picos na semana 14 e 21. Ao longo das semanas a participação dos utilizadores foi diminuindo havendo poucas semanas em que o número de denúncias esteve acima da média. Analisando os emails recebidos inicialmente é possível perceber que vários utilizadores reportaram uma grande quantidade de emails das suas pastas de *junk*. Sendo que os emails já presentes (mais antigos) foram reportados inicialmente, nas seguintes utilizações os emails são reportados à medida que entram nas caixas de correio dos colaboradores, sendo esta uma possível causa para os valores das denúncias serem menores.

6.5 Número de utilizadores ativos

O número de utilizadores ativos é uma métrica relevante que foi calculada durante a duração do projeto. É importante calcular este valor para perceber se o sistema foi usado ativamente pela maior parte da população que teve acesso ao mesmo.

A amostra de utilizadores que fizeram parte dos testes ao sistema é toda a DCY, sendo que existem 49 pessoas (7 dos quais mestrandos) na direção. Foi possível verificar que 34 pessoas (aproximadamente 69%) carregaram no botão pelo menos uma vez. No entanto para tentar chegar

[1] Ferramenta de visualização de dados que transforma os dados brutos num formato de fácil compreensão [11].

a um valor de utilizadores ativos é necessário chegar à média de denúncias que cada utilizador fez ao longo das semanas.

Para obter o valor médio desejado, foi então necessário obter o número médio de denúncias feitas por utilizador. O espaço de tempo definido foi uma semana, portanto os valores retirados são: o número médio de denúncias feitas por utilizador numa semana. Através destes parâmetros é possível criar um gráfico de forma a comparar a média do número de denúncias que cada utilizador fez (Figura 6.2). O gráfico foi feito recorrendo ao *Tableau* utilizando como fonte de dados as *queues RoboPHISH - EMAILS* e *RoboPHISH - USERS* retiradas do SQL.

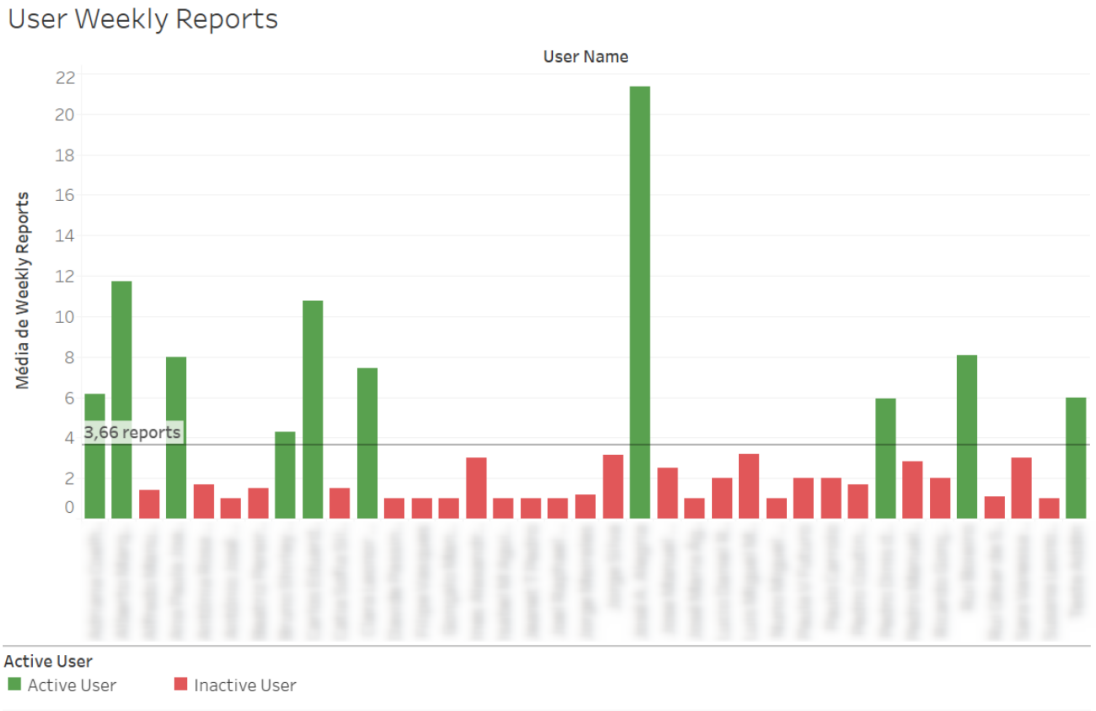


Figura 6.2: Classificação de emails (*tag*) por tempo

Após a criação do gráfico foi definido um *threshold*, um número mínimo para o utilizador ser considerado ativo. O valor foi escolhido através da observação do número médio de denúncias, por utilizador, por semana: `3,66` denúncias. É possível observar este valor no gráfico correspondente à Figura 6.2. Através desta visualização é também possível observar a vermelho o número de utilizadores inativos (abaixo da média), 27 utilizadores, e a verde os ativos (acima da média), 10 utilizadores. Analisando as denúncias é possível observar que há um número elevado de utilizadores inativos, desta forma foram consultados alguns destes colaboradores. Falando com os colaboradores em questão foi então possível concluir que há muitos destes que recebem pouco SPAM ou *phishing*. Especialmente as contas mais recentes e que ainda não estão em nenhuma lista *online* sofrem muito poucos ou nenhuns ataques de *phishing*, sendo que os seus *reports* são maioritariamente email de campanha.

6.6 Classificação de emails

Acompanhar a classificação dos emails pelo *RoboPHISH* ao longo do tempo é fundamental para perceber se o sistema está a conseguir classificar uma percentagem significativa de emails. Os emails podem ser classificados como *Clean*, *SPAM* e *Phishing*. No caso de ainda estarem a aguardar classificação a *tag* do email deverá estar a *null*, havendo por isso mais uma categoria onde poderão estar inseridos os emails. O sistema *RoboPHISH* deverá conseguir classificar o máximo de emails possível. Com este objetivo em mente é importante acompanhar a quantidade de emails classificados, tentando sempre que este número seja elevado, mantendo claro a precisão do sistema.

Com a finalidade de acompanhar o número de classificações (*SPAM*, *Phishing*, *Clean* e *Null*) feitas pelo robô foi desenvolvido um gráfico no *Tableau*, representado na Figura 6.3, que corresponde à classificação *RoboPHISH* de emails por semana. Como foi possível constatar no gráfico anterior (Figura 6.2) houve mais *reports* e portanto mais classificações feitas pelo robô nas primeiras semanas. Para além disso a ideia é analisar quais as classificações atribuídas pelo sistema. Inicialmente podemos observar uma percentagem de emails classificados como *Clean* bastante elevada (barras azuis). Ao longo das semanas esta percentagem foi diminuindo significativamente, este facto pode ser explicado por três fatores principais: alterações feitas ao sistema, diminuição da quantidade de denúncias e aumento da base de dados de emails classificados.

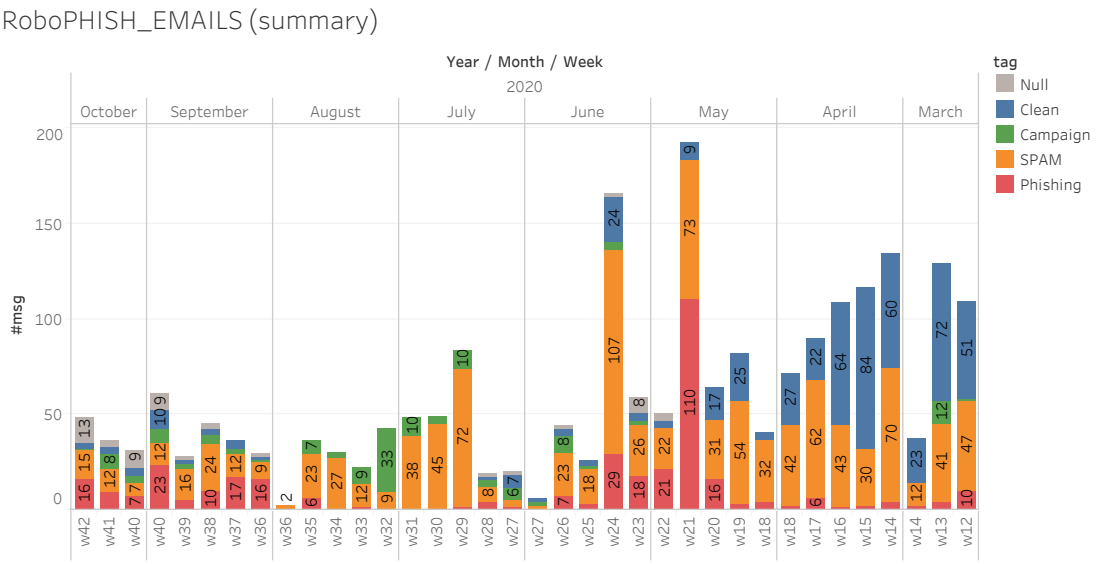


Figura 6.3: Classificação de emails (*tag*) por tempo

O acompanhamento deste valor ao longo do tempo permitiu perceber que o robô estava a deixar muitos emails por classificar, em algumas semanas mais de metade. Desta forma foi então possível fazer melhorias ao sistema, adicionando dois novos métodos de classificação: *Verify SPAM icu* (Subseção 5.3.4) e *Verify Guru Vote* (Figura 5.19). Para além disso foi adicionado o módulo *Send Expert Votes to SOC* que permitiu enviar uma parte dos emails por classificar ao SOC e foram feitos pequenos ajustes ao algoritmo de classificação dos utilizadores. Podemos confirmar no gráfico que estas novas adições ajudaram a diminuir muito o número de emails por

classificar, que inicialmente era muito elevado. Nas semanas de 12 a 19 temos números elevados de emails classificados como *Clean*. Nas semanas seguintes podemos constatar que este número é muito menor.

6.7 Classificação de colaboradores

Na secção anterior, Secção 6.6, é apresentada a distribuição temporal da classificação dos emails pelo *RoboPHISH*. Tal como no caso dos emails os colaboradores que utilizam o sistema também têm uma classificação associada. Esta classificação é guardada no campo *tag* da *queue* de emails (passada para SQL) e pode ter os seguintes valores: *No Tag Yet*, *Informed*, *Expert* ou *Guru* (Secção 4.3.3). A Figura 6.4 representa a tabela da categorização dos utilizadores do sistema *RoboPHISH* desenvolvida recorrendo ao *Tableau*.

RoboPHISH_USERS

tag	Rating	User_Email	#msg	#ReportSPAM	#ReportPhishing
Guru	100	[redacted]	11	3	8
		[redacted]	76	0	76
	86	[redacted]	830	370	460
	85	[redacted]	13	2	11
	76	[redacted]	110	65	45
	0	[redacted]	4	3	1
Expert	70	[redacted]	298	188	110
	65	[redacted]	284	256	28
	58	[redacted]	12	2	10
	54	[redacted]	63	9	54
	51	[redacted]	34	22	12
Informed	46	[redacted]	140	56	84
	45	[redacted]	133	81	52
	41	[redacted]	11	0	11
	40	[redacted]	45	33	12
	35	[redacted]	17	4	13
	27	[redacted]	40	32	8
No Tag Yet	0		1	0	1
		[redacted]	8	6	2
		[redacted]	10	5	5
		[redacted]	4	1	3
		[redacted]	3	1	2
		[redacted]	5	2	3
		[redacted]	1	0	1
		[redacted]	2	0	2
		[redacted]	4	0	4
		[redacted]	8	0	8
		[redacted]	1	0	1
		[redacted]	3	1	2
		[redacted]	4	1	3
		[redacted]	5	0	5
		[redacted]	3	1	2
		[redacted]	1	0	1
		[redacted]	2	1	1
		[redacted]	2	2	0
		[redacted]	2	0	2
		[redacted]	5	0	5
		[redacted]	4	1	3

tag
Guru
Expert
Gentile
No Tag Yet

Figura 6.4: Classificação de utilizadores (*tag*), número de denúncias e *rating* (atualmente *accuracy)*

A análise desta tabela permite ter uma ideia do panorama geral dos utilizadores do sistema.

Na categoria de *Guru* existem apenas cinco utilizadores, quatro deles promovidos tendo em conta a *accuracy* e um deles por nomeação direta (elemento da equipa SOC). Em relação às restantes categorias, todos os elementos estão inseridos nas mesmas apenas pela sua *accuracy* (processo automático). É possível observar que existem cinco elementos na categoria *Expert* e seis na categoria *Informed*. Todos os restantes utilizadores estão inseridos na categoria *No Tag Yet*, logo têm menos de 10 denúncias. Fazendo uma avaliação geral, estes resultados vão de encontro ao que foi referido no gráfico anterior representado na Figura 6.2. Há uma grande quantidade de utilizadores que não denuncia muitos emails, em princípio porque não recebe muito SPAM ou *phishing*.

Capítulo 7

Conclusão

O sistema *RoboPHISH* tinha como objetivo implementar com sucesso um sistema eficaz e autónomo de denúncia, análise e mitigação de *phishing*. Tendo em conta os objetivos propostos o projeto foi bem sucedido. A utilização do sistema *RoboPHISH* permitiu aos colaboradores um meio de denúncia rápido, simples e intuitivo. A análise dos emails potencialmente maliciosos, desenvolvida recorrendo ao RPA, cumpriu o objetivo de retirar pressão sobre a equipa SOC melhorando e inovando relativamente ao processo manual.

Através da avaliação dos resultados apresentados pelo sistema, é possível concluir que o sistema foi bem construído, apresentado taxas de erros de execução baixas. Como percentagem máxima de erro temos a do processo *RoboPHISH001 - 02 - Analyse Emails from Folder*, 6,82% por cento. Os outros dois processos rondam os 0,40% de percentagem de erros, valores bastante reduzidos e considerados bastante bons. A execução no BP da componente de análise e resposta do processo foi bastante rápida, sendo que este ponto também é importante para o sucesso do sistema. Somando os tempos de execução dos processo, em condições normais, temos 68 segundos (cerca de 1 minuto). Considerando o pior cenário, todos os processos falham, os processos demoram em média 437 segundos (cerca de 7 minutos), o que mesmo assim é um tempo bastante reduzido.

Acompanhando os gráficos desenvolvidos é possível perceber que o sistema foi usado, pelo menos uma vez, por uma grande percentagem da população que o testou e que houve uma redução no número de denúncias feitas pelos utilizadores ao longo do tempo. Esta redução pode ser explicadas por vários fatores, alguns colaboradores recebem poucos emails maliciosos e no início da utilização do sistema vários colaboradores reportaram os emails acumulados que tinham nas suas caixas de *junk*. Desta forma vemos um aumento acentuado do número de denúncias nas primeiras semanas de utilização do sistema, seguido de uma diminuição destes valores.

O sistema *RoboPHISH* está neste neste momento em produção na DCY e apresenta bons resultados e aceitação. O *RoboPHISH* apresenta várias vantagens face ao processo anterior, exclusivamente manual. A libertação dos recursos humanos da equipa SOC para tarefas mais pertinentes foi atingida visto que a quantidade de emails a analisar manualmente reduziu significativamente. Tanto o RPA como o algoritmo de *crowdsourcing* mostraram ser boas soluções para ajudar a combater o problema do *phishing* na empresa. A concretização de um algoritmo de *crowdsourcing* revelou ser uma solução inovadora e eficaz que permitiu a classificação de um maior número de mensagens. Através da utilização do sistema as respostas aos colaboradores foram asseguradas,

ponto fundamental quer para incentivar o colaborador a reportar novamente, quer para aumentar a *awareness* do mesmo para o problema do *phishing*. Os bloqueios nas plataformas corretas foram assegurados diminuindo o tempo de bloqueio dos emails e URLs, visto que estes são pedidos imediatamente após a classificação do email.

Atendendo ao tempo de desenvolvimento do projeto, foi possível desenvolver um sistema funcional e capaz de ser colocado em produção. No entanto há alterações e incrementos que podem ser feitos ao sistema de modo a aperfeiçoar o funcionamento do mesmo. De forma a reduzir o tempo de execução e aumentar a fiabilidade seria importante passar o módulo de classificação do VirusTotal para API. O VirusTotal dispõe de uma API, sendo possível integrar os pedidos à mesma no BP. O VirusTotal, *website* e API, permite a análise de ficheiros maliciosos. Esta *feature*, embora tenho sido explorada, acabou por não ser desenvolvida visto que não seria exequível em tempo útil. Numa iteração futura seria também uma mais valia, para a classificação de emails, acrescentar esta verificação dos ficheiros anexados aos emails potencialmente maliciosos. O sistema será estendido futuramente a todos os colaboradores da Altice Portugal e será continuado num próximo projeto.

Acrónimos

ARF Abuse Report Form.

BP Blue Prism.

CMD Command Prompt.

CSOC Security Operations Center.

DCY Direção de Cibersegurança.

RPA Robotic Process Automation.

SOC Security Operations Center.

SQL Structured Query Language.

SSD System Sequence Diagram.

UML United Modeling Language.

VB Visual Basic.

Glossário

Anubis Ferramenta *anti-spam/ phishing* que efetua proteção de email e filtragem de emails maliciosos através da deteção de *malware* e SPAM no email.

ARF Formulário *web* para pedir bloqueio de emails na ferramenta Anubis.

Blue Prism Ferramenta de RPA que permite a criação de uma força de trabalho virtual constituída por robôs de *software*.

Exchange Servidor de email da Microsoft através do qual é feita toda a gestão de emails da Microsoft [10].

HPSM Ferramenta na Altice Portugal que permite a criação e gestão de tickets que correspondem a incidentes.

Macro Uma macro é um conjunto de ações que podem ser executadas várias vezes para automatizar tarefas. A criação de macros é feita recorrendo à linguagem de programação Visual Basic [18].

Phishing Tipo de ataque informático caracterizado pelo uso de engenharia social como forma de manipular as suas vítimas levando-as a fornecer dados pessoais, acessos privilegiados ou informação confidencial.

RPA (Robotic Process Automation) *Software* que utiliza regras, definidas pelo programador, para conseguir a execução automática de tarefas, processos e transações que interagem com vários sistemas de *software*.

SPAM Emails não solicitados enviados indiscriminadamente a um número elevado de utilizadores.

SQL É uma linguagem de programação utilizada especificamente para gerir dados de uma base de dados relacional [29].

Visual Basic Visual Basic é uma linguagem de programação e desenvolvimento, orientada a eventos, criada pela Microsoft [30].

Zscaler Ferramenta que permite uma experiência de navegação na *Internet* segura, condicionando o acesso dos utilizadores à mesma.

Bibliografia

[1] Andrew1234. Jaro and jaro-winkler similarity. https://www.geeksforgeeks.org/jaro-and-jaro-winkler-similarity/, 2020. [Online; acedido a 13-10-2020].

[2] Anubisnetworks. About mailspike.org. http://mailspike.org/about.html, -. [Online; acedido a 20-11-2019].

[3] Anubisnetworks. Email security service. https://www.anubisnetworks.com/, -. [Online; acedido a 20-11-2019].

[4] Blueprism. Embracing RPA and choosing the right processes to automate - 7 steps to success (part 5). https://www.blueprism.com/resources/blog/7-steps-to-success-part-5-embracing-rpa-and-choosing-the-right-processes-to-automat/, -. [Online; acedido a 18-02-2019].

[5] CFCGlobal. Outlook 2010 - getting to know outlook 2010. https://edu.gcfglobal.org/en/outlook2010/getting-to-know-outlook-2010/1/, -. [Online; acedido a 15-09-2020].

[6] DirectToMX. Directtomx. http://www.directtomx.com/, -. [Online; acedido a 20-11-2019].

[7] Europol EC3. Internet organised crime threat assessment. *Europol Public Information*, 2019.

[8] Europol EC3. Spear phishing - a law enforcement and cross-industry perspective. *Europol Public Information*, 2019.

[9] Joris Evers. Security expert: User education is pointless. https://www.cnet.com/news/security-expert-user-education-is-pointless/, 2006. [Online; acedido a 28-11-2019].

[10] R & g technologies. Microsoft exchange. https://rgtechnologies.com.au/resources/microsoft-exchange/, -. [Online; acedido a 21-01-2021].

[11] GURU99. What is tableau? uses and applications. https://www.guru99.com/what-is-tableau.html, -. [Online; acedido a 16-09-2020].

[12] Kaspersky. What is a whaling attack? https://www.kaspersky.com/resource-center/definitions/what-is-a-whaling-attack. [Online; acedido a 15-11-2019].

[13] Lexico. Meaning of malware in English. https://www.lexico.com/definition/malware, -. [Online; acedido a 16-05-2019].

[14] Lexico. Meaning of ransomware in English. https://www.lexico.com/definition/ransomware, -. [Online; acedido a 16-05-2019].

[15] Lucidchart3. System sequence diagrams in uml. https://www.lucidchart.com/pages/uml-system-sequence-diagram, -. [Online; acedido a 12-05-2020].

[16] Craig McDonald. Is your business prepared to handle a sudden email outage? today? https://www.mailguard.com.au/blog/is-your-business-prepared-handle-sudden-email-outage-today, 2019. [Online; acedido a 18-11-2019].

[17] Microsoft. Create custom actions rules in outlook for windows. https://support.microsoft.com/en-us/office/create-custom-actions-rules-in-outlook-for-windows-c6a15a50-5b4f-43ea-9bcf-be8616db8a98#, -. [Online; acedido a 11-06-2020].

[18] Microsoft. Quick start: Create a macro. https://support.microsoft.com/en-us/office/quick-start-create-a-macro-741130ca-080d-49f5-9471-1e5fb3d581a8, -. [Online; acedido a 21-01-2021].

[19] Matt Moorehead. How to explain dkim in plain English. https://www.validity.com/how-to-explain-dkim-in-plain-english/, 2015. [Online; acedido a 04-11-2019].

[20] Matt Moorehead. How to explain spf in plain English. https://www.validity.com/how-to-explain-spf-in-plain-english/, 2015. [Online; acedido a 04-11-2019].

[21] namecheap. What is a tld? https://www.namecheap.com/domains/what-is-a-tld-definition/, -. [Online; acedido a 10-09-2020].

[22] J. Nielsen. User education is not the answer to security problems. http://www.useit.com/alertbox/20041025.html, 2004. [Online; acedido a 28-11-2019].

[23] Visual Paradigm. What is sequence diagram? https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/, -. [Online; acedido a 12-05-2020].

[24] Rob Pegoraro. Google made its employees impervious to phishing using usb security keys. https://www.fastcompany.com/90207052/google-made-its-employees-impervious-to-phishing-using-usb-security-keys, 2018. [Online; acedido a 20-11-2019].

[25] Rob Pegoraro. We keep falling for phishing emails, and google just revealed why. https://www.fastcompany.com/90387855/we-keep-falling-for-phishing-emails-and-google-just-revealed-why, 2019. [Online; acedido a 14-11-2019].

[26] Lexico powered by Oxford. Definition of outsource in English. https://www.lexico.com/en/definition/outsource, -. [Online; acedido a 14-04-2020].

[27] Blue Prism. Blue prism. https://www.blueprism.com/pt/, 2017. [Online; acedido a 30-10-2019].

[28] Techopedia. Microsoft outlook. https://www.techopedia.com/definition/355/microsoft-outlook, 2017. [Online; acedido a 09-06-2020].

[29] Techopedia. Structured query language (sql). https://www.techopedia.com/definition/1245/structured-query-language-sql, 2021. [Online; acedido a 21-01-2021].

[30] TechTerms. Visual basic. https://techterms.com/definition/visualbasic, 2007. [Online; acedido a 21-01-2021].

[31] VirusTotal. How it works. https://support.virustotal.com/hc/en-us/articles/115002126889-How-it-works. [Online; acedido a 18-03-2019].

[32] Vários. Writing an outlook macro. https://docs.microsoft.com/en-us/office/vba/outlook/concepts/getting-started/writing-an-outlook-macro, 2017. [Online; acedido a 10-11-2019].

[33] Wikipedia. Wikipedia:user access levels. https://en.wikipedia.org/wiki/Wikipedia:User_access_levels, -. [Online; acedido a 03-06-2020].

Apêndice A

Tutorial de Instalação *RoboPHISH*

1º) Guardar ficheiro que contém Macro

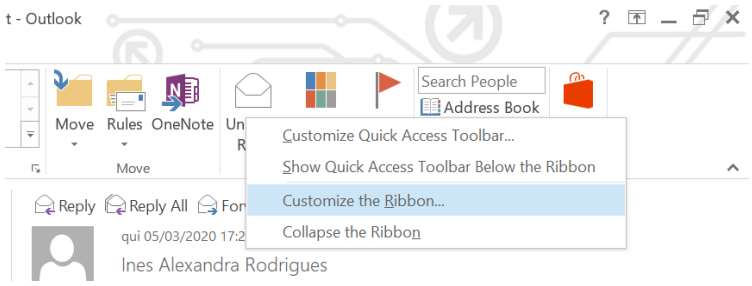
- Transferir o anexo do *email* para uma diretoria à escolha

- Extrair o ficheiro (.ZIP) previamente transferido

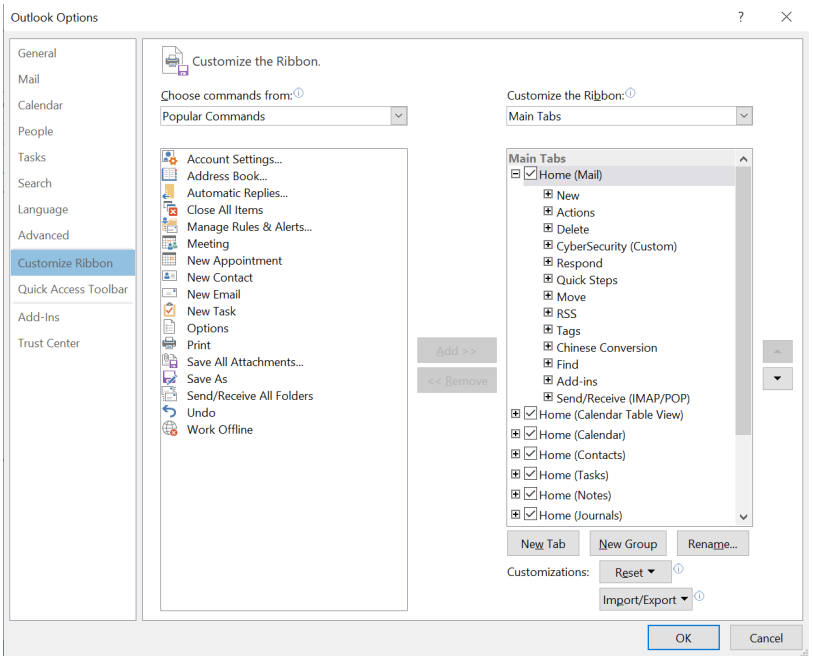
2º) Adicionar o separador *DEVELOPER* ao Outlook

- Carregar no botão direito do rato sobre a barra de funcionalidades do Outlook (*Ribbon*)

- Escolher a opção: "*Customize the Ribbon...*"



Deverá aparecer uma janela como a seguinte

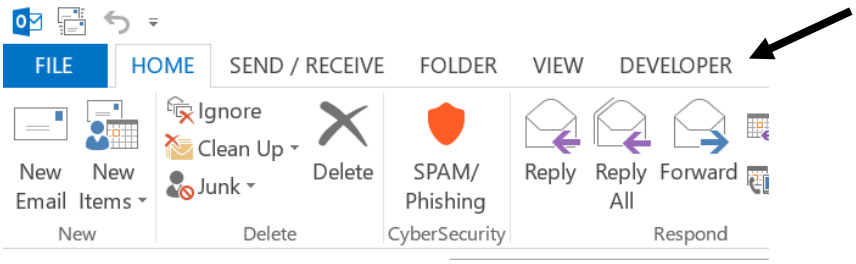


-Fazer *scroll* na janela da direita até encontrar: ☐ *Developer*

- Marcar o pisco no ☐ *Developer*

- Carregar OK

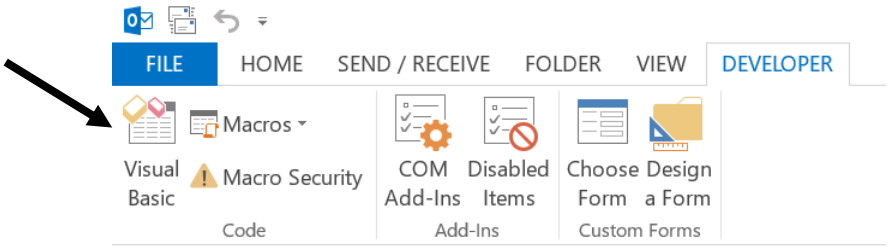
- Após aplicadas as alterações, deverá aparecer um novo separador na barra superior, como demonstrado na seguinte imagem:



3º) Importar Macros no Visual Basic editor

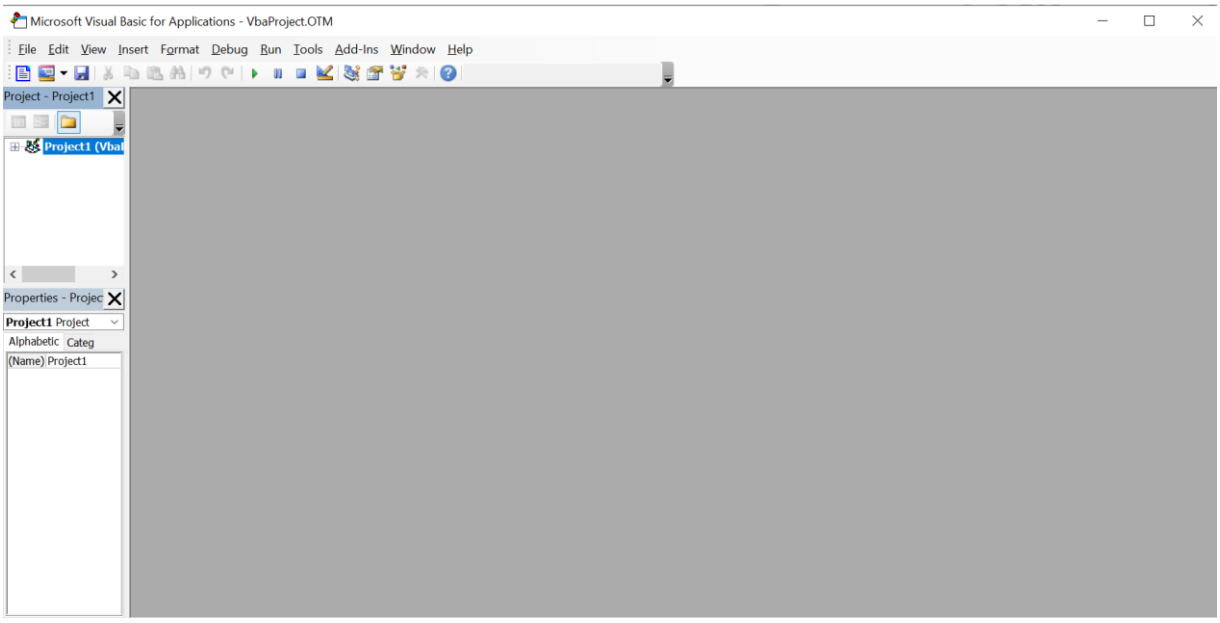
- Selecionar o separador *Developer*

- Deverão aparecer as seguintes opções:



- Selecionar a opção Visual Basic

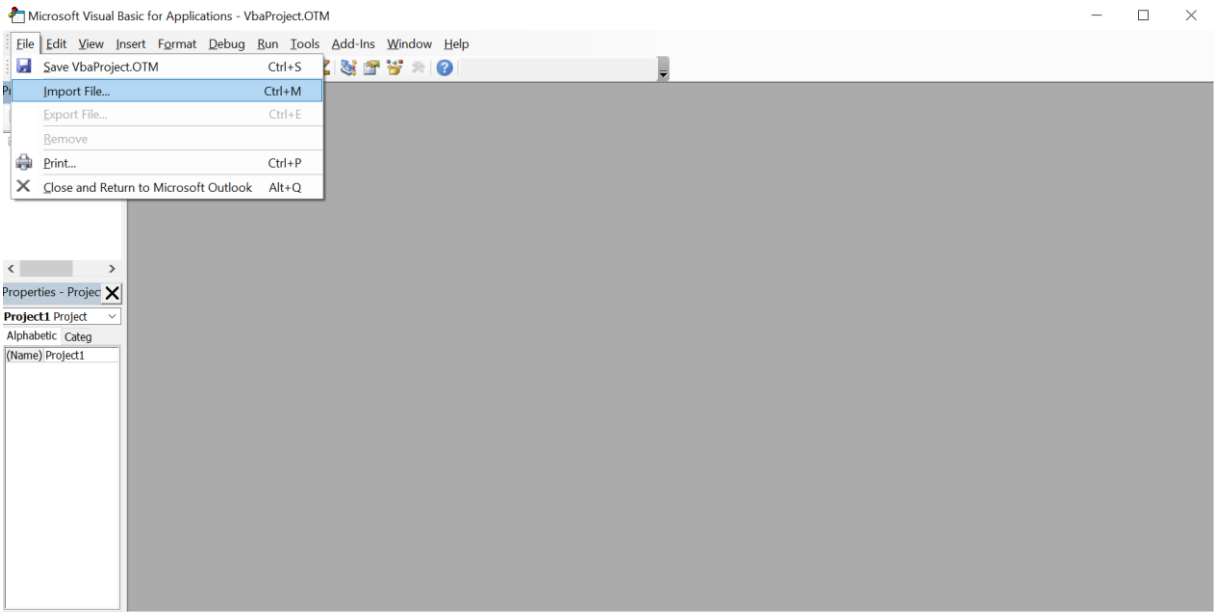
Deverá aparecer uma janela como a seguinte



- Selecionar a opção: *File*

- No *dropdown* selecionar a opção "*Import File…*"

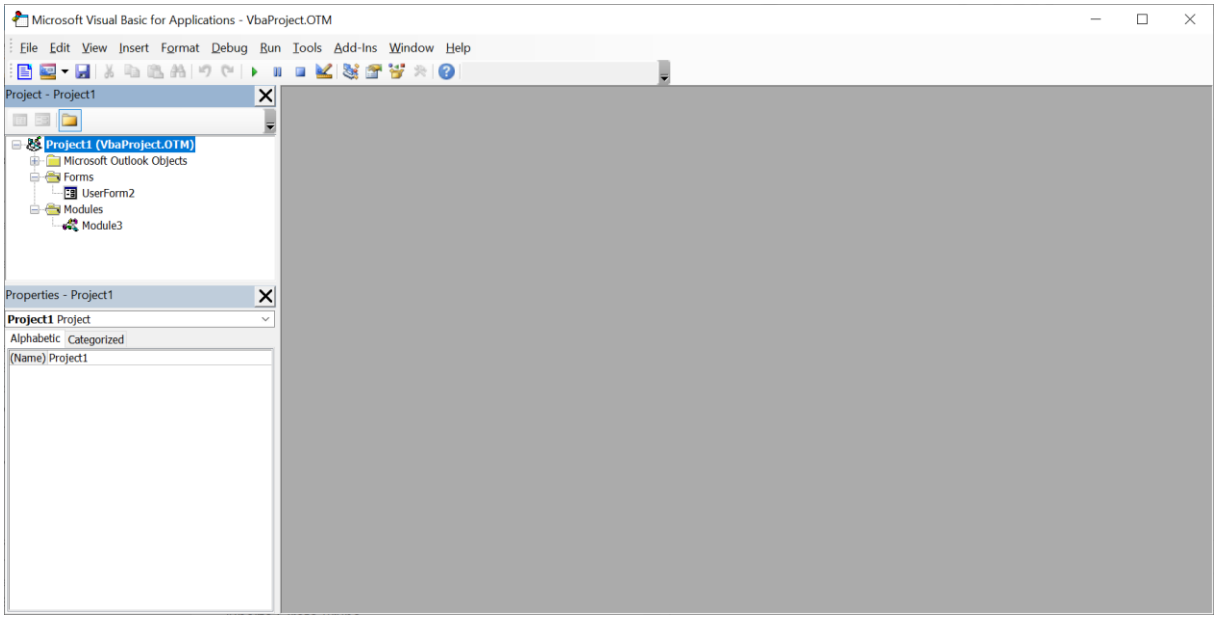
- Ir para a diretoria onde guardou os ficheiros do botão



- Selecionar o formulário: "UserForm2.frm"

- Repetir o processo, importando agora o módulo: "Module3.bas"

Deverão aparecer os dois módulos na janela do lado esquerdo



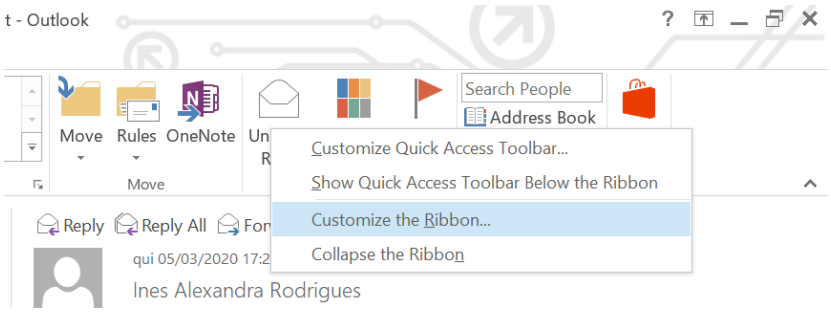
- Guardar as alterações (carregando no símbolo 🖫 ou File -> Save)

- Fechar a janela

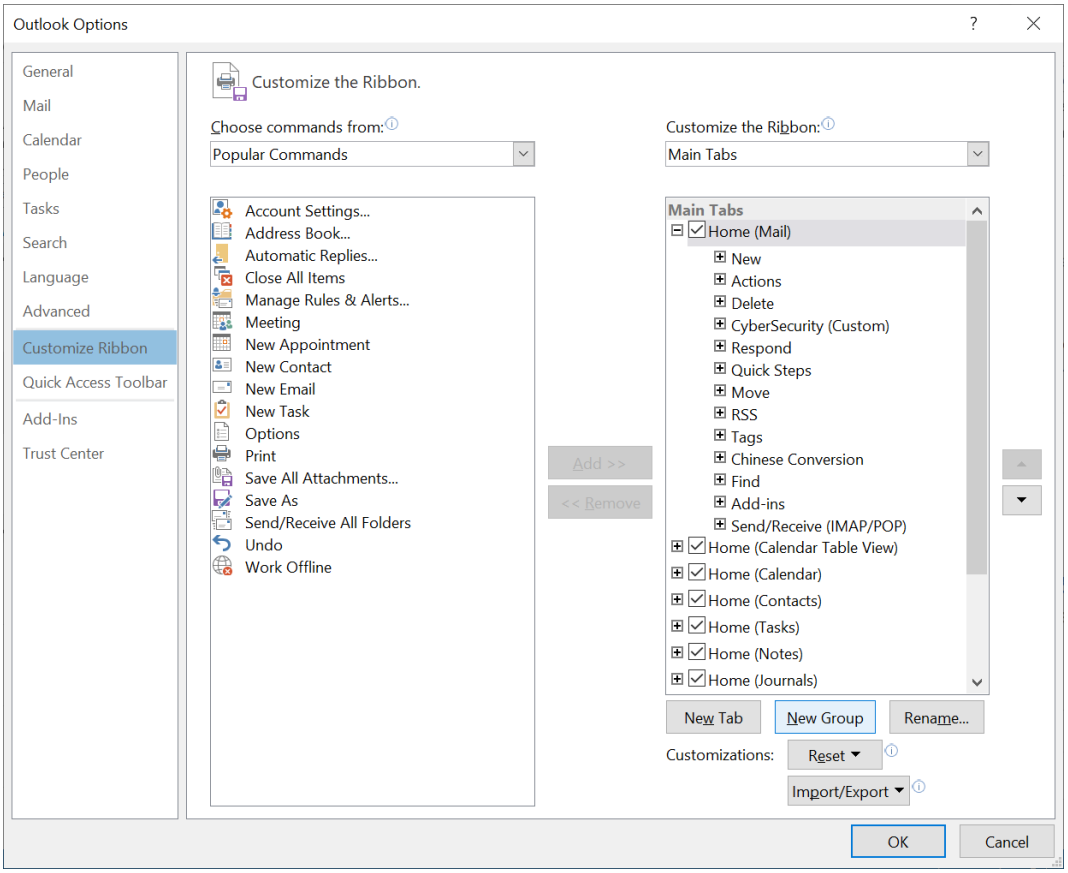
4º) Criar botão na *Home* e associar à Macro

- Voltando ao separador *Home*, voltar a alterar a barra de funcionalidades do Outlook (*Ribbon*)

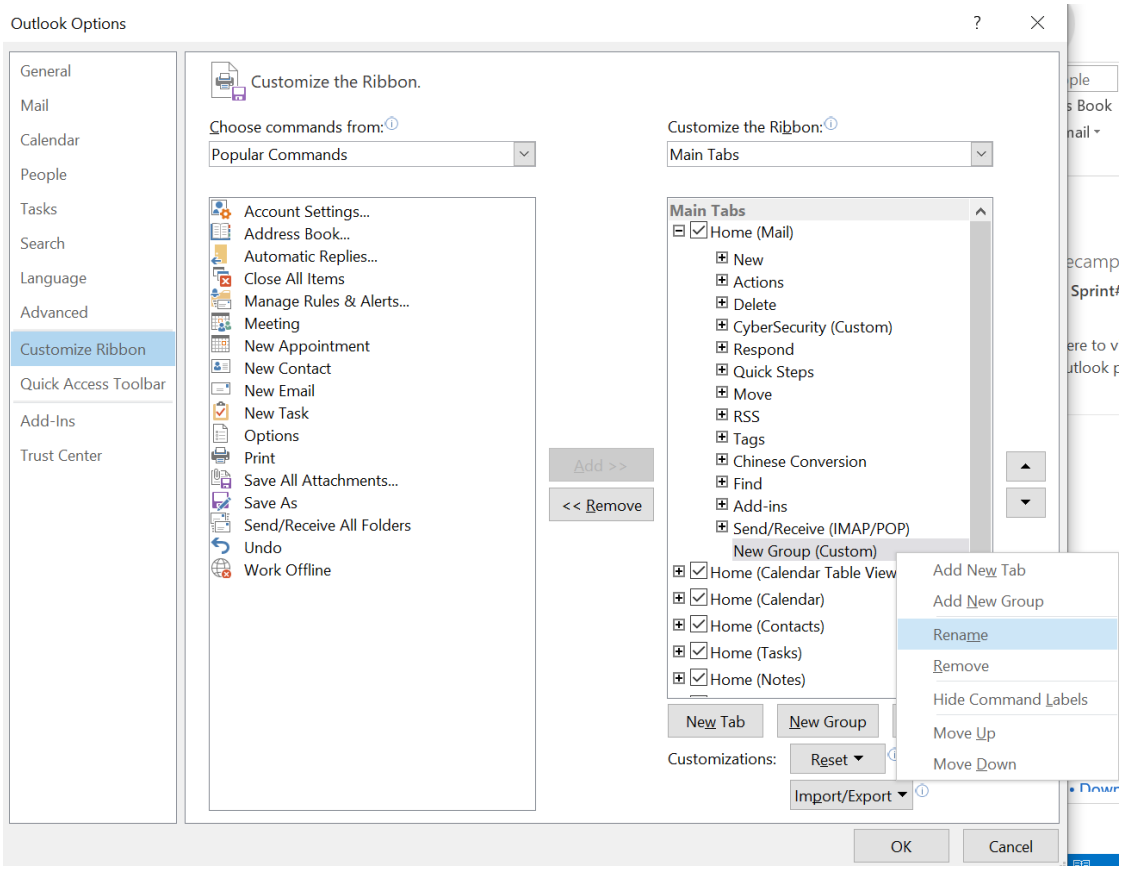
- Botão direito do rato sobre a Ribbon e selecionar a opção: "*Customize the Ribbon…*"



- Dentro do separador *Home (Mail)* criar um novo grupo utilizando o botão New Group



- Carregar com o botão direito do rato sobre o novo grupo e escolher a opção *Rename*



- Alterar o nome do grupo para **CyberSecurity** (não é necessário selecionar nenhum símbolo neste passo)

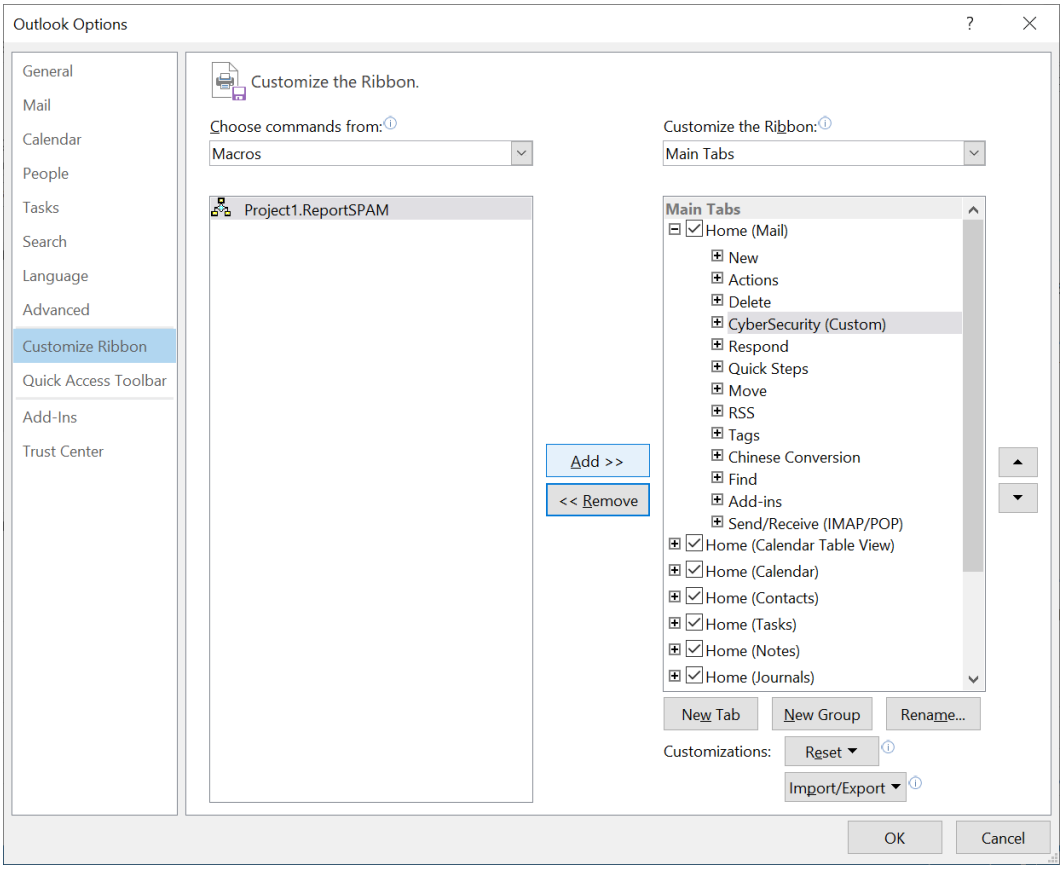
- Puxar o grupo CyberSecurity de forma a este ficar entre *Delete* e *Respond*

- Selecionar o novo grupo (CyberSecurity)

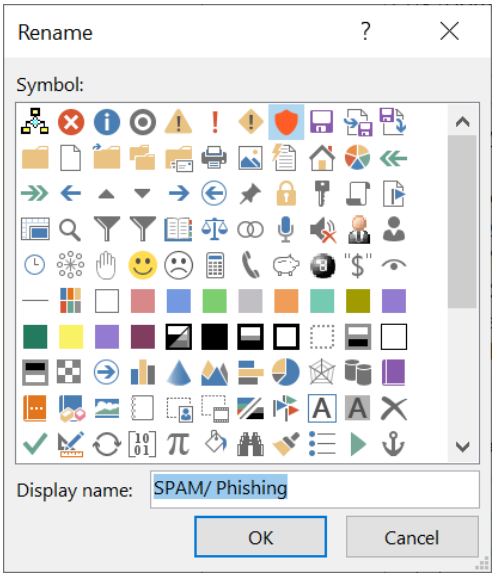
- No *dropdown* da esquerda, "*Choose commands from*", selecionar a opção Macros

- Selecionar Project1.ReportSPAM

- Carregar na opção Add >>

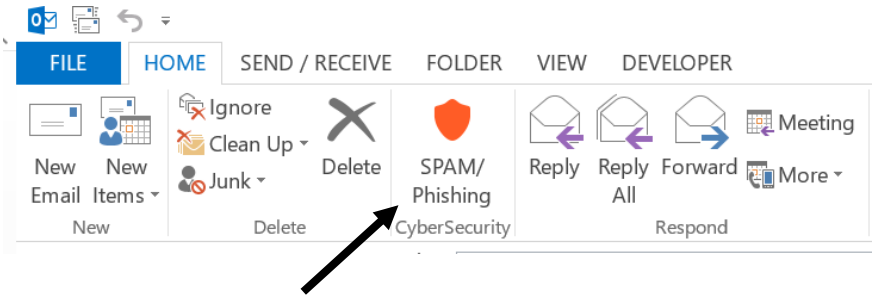


- Já dentro do grupo deverá aparecer a macro selecionada (Project1.ReportSPAM)
- Carregar com o botão do lado direito do rato sobre a mesma e selecionar *Rename*
- Alterar o nome para: **SPAM/ Phishing** e selecionar o símbolo do escudo vermelho



- Carregar OK e fechar a janela *Outlook Options*

Deverá aparecer o seguinte botão dentro do separador *Home*

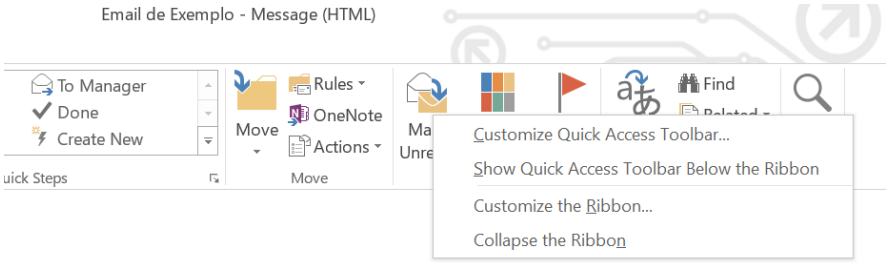


5º) Criar botão dentro do *email* e associar à Macro

- Selecionar e abrir um email qualquer

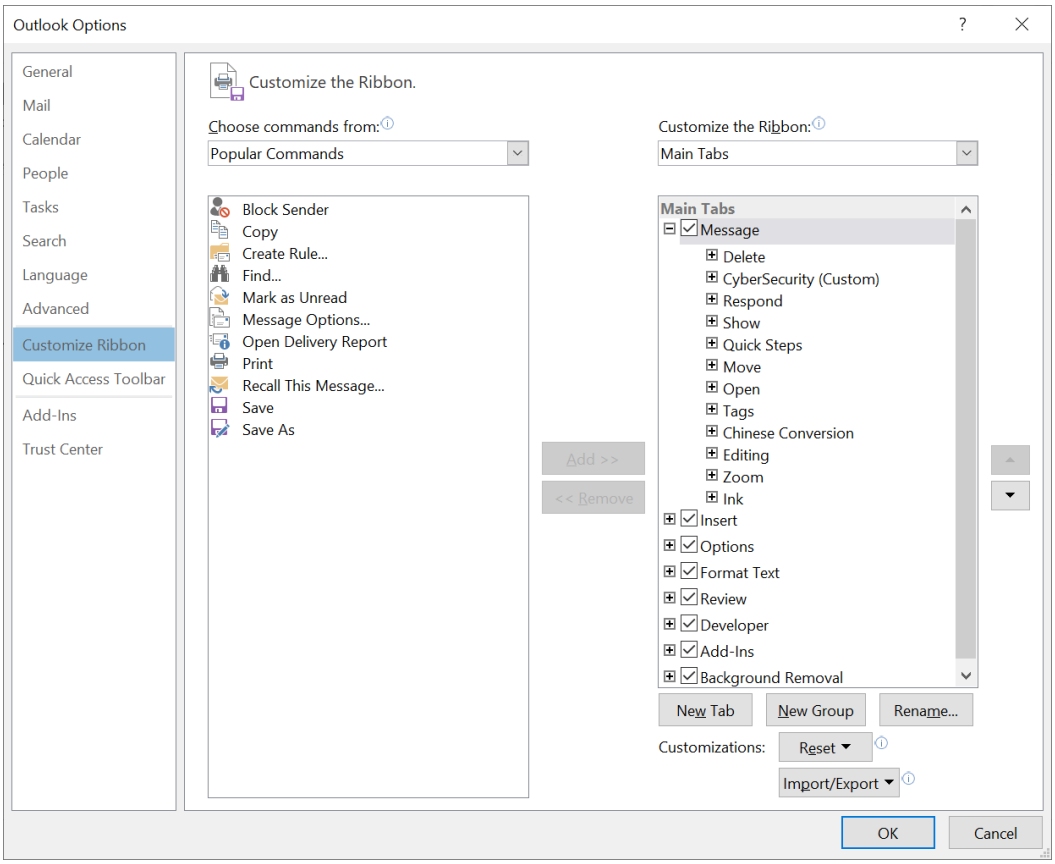
- Dentro do email aberto repetir o ponto 4

- Botão direito do rato sobre a Ribbon (desta vez a barra de funcionalidades dentro do email) e selecionar a opção: "*Customize the Ribbon...*"



- Dentro do separador *Message* criar um novo grupo utilizando o botão New Group

- Carregar com o botão direito do rato sobre o novo grupo e escolher a opção *Rename*



- Alterar o nome do grupo para **CyberSecurity** (não é necessário selecionar nenhum símbolo neste passo)

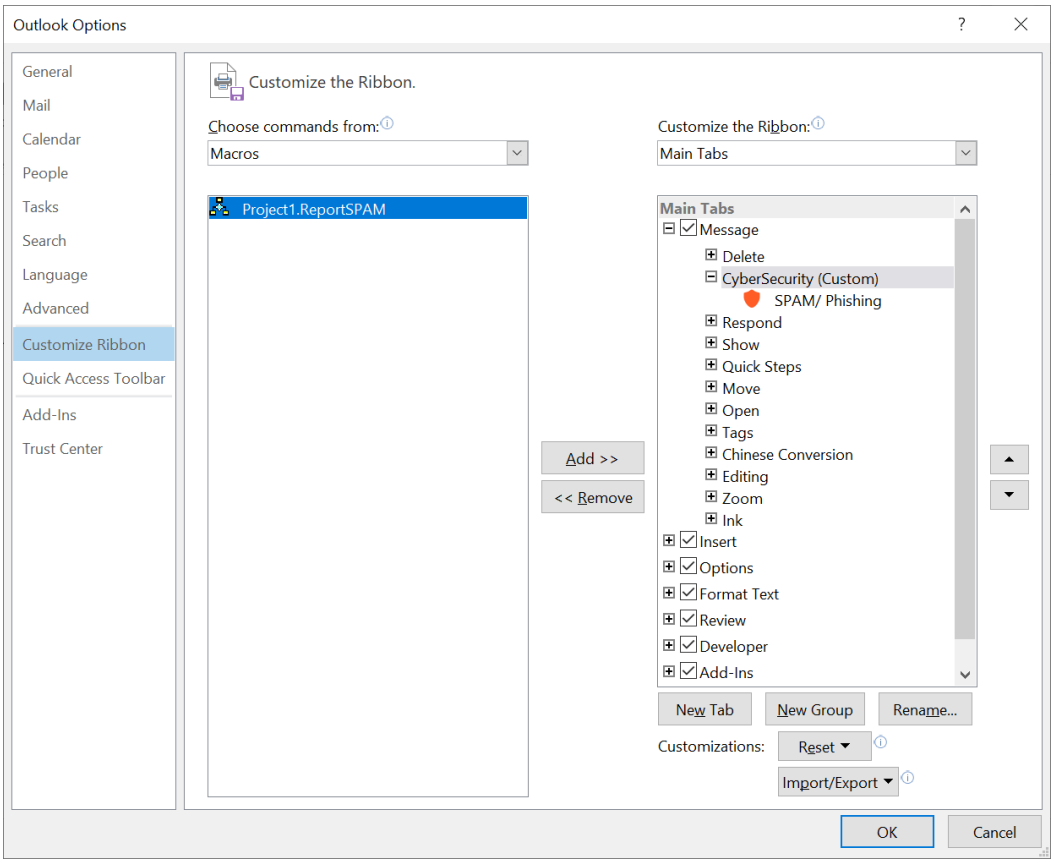
- Puxar o grupo CyberSecurity de forma a este ficar entre *Delete* e *Respond*

- Selecionar o novo grupo (CyberSecurity)

- No *dropdown* da esquerda, "*Choose commands from*", selecionar a opção Macros

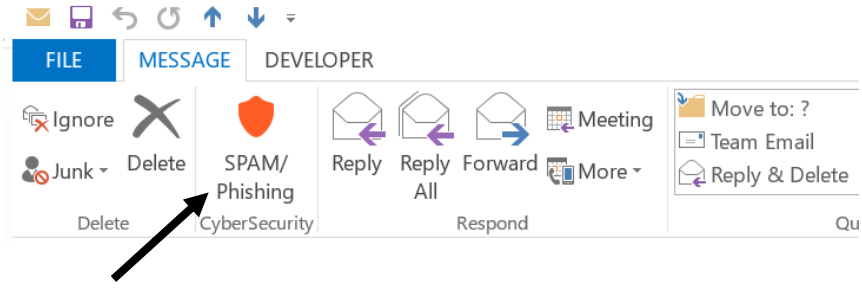
- Selecionar Project1.ReportSPAM

- Carregar na opção Add >>



- Já dentro do grupo deverá aparecer a macro selecionada (Project1.ReportSPAM)
- Carregar com o botão do lado direito do rato sobre a mesma e selecionar *Rename*
- Alterar o nome para: **SPAM/ Phishing** e selecionar o símbolo do escudo vermelho
- Carregar OK e fechar a janela *Outlook Options*

Deverá aparecer o seguinte botão dentro do separador *Message*



5º) Permitir que o Outlook corra macros (com notificação)

- No canto superior esquerdo selecionar: File
- No menu da esquerda (azul) selecionar *Options*
- Selecionar a última opção: *Trust Center*
- Dentro de *Trust Center* carregar no botão Trust Center Settings
- Selecionar *Macro Settings*
- Selecionar a opção: "*Notifications for all macros*"
- Carregar OK
- Reiniciar o Outlook

Nota: Após reiniciar o Outlook botão já estará operacional. Será necessário dar permissão à macro para correr, após carregar no botão.