Mestrado em Informática e Sistemas

DNS como mecanismo de proteção e fonte de informação em segurança cibernética

Relatório de Estágio apresentado para a obtenção do grau de Mestre em Informática e Sistemas
Especialização em Tecnologias de Informação e do Conhecimento

Autor
Sérgio Filipe Lourenço Ribeiro

Orientador
Mestre Luís Eduardo Faria dos Santos
Doutora Teresa Raquel Corga Teixeira Rocha
Professores do Departamento de Engenharia Informática e Sistemas
Instituto Superior de Engenharia de Coimbra

Supervisor
Mestre André Manuel Pereira Pinheiro
Dognaedis

Coimbra, junho, 2019

Agradecimentos

À minha família, em especial ao meu pai José, à minha mãe Elvira e à minha irmã Andreia por todo apoio e compreensão durante este ciclo.

À minha namorada Ana Patrícia por toda ajuda e paciência.

À família Dognaedis que mais uma vez esteve presente, agora na conclusão deste desafio pessoal.

Resumo

Atualmente, a cibersegurança é cada vez mais uma preocupação para as organizações. Todos os dias surgem novas ameaças e, para ser possível dar-lhes resposta, é necessário ter pessoas, processos e tecnologias prontamente alinhados e preparados. Como forma de detetar ciber ameaças, a Dognaedis desenvolveu o Portolan - uma plataforma de ciber inteligência. Através da mesma é possível analisar informação relevante em tempo real, bem como otimizar e apoiar as tomadas de decisão no âmbito dos serviços prestados aos clientes, nomeadamente *Managed Security Services*, *Cyber Intelligence* ou Consultoria.

O *Domain Squatting* consiste no registo de um domínio DNS que verbal e/ou visualmente apela a marcas populares e confiáveis. Também conhecida por Cybersquatting, esta prática começou por ser usada como forma de obter mais valias na venda de novos domínios. Porém, cedo, atores maliciosos começaram a explorá-la para recolher credenciais e disseminar malware. O projeto desenvolvido pretendeu dotar o Portolan de autonomia na deteção de tais práticas ílegítimas.

O fenómeno do *Domain Squatting* foi estudado nas suas diversas vertentes. Com base no conhecimento adquirido foi desenvolvida uma estratégia de reconhecimento autónomo desta atividade com um grau de precisão de 81.45%. A técnica foi integrada com sucesso no Portolan. Desenvolveu-se ainda uma Firewall DNS que, retirando partido da nova funcionalidade do Portolan, consegue fornecer proteção em tempo real face à ameaça em estudo. O seu desempenho varia entre os 341 e 2331 domínios analisados por segundo com uma latência média por domínio de 2 a 200 milisegundos, quando animada por uma máquina virtual com 1 vCPU, 1 GB de RAM e 20 GB de disco SSD a correr Ubuntu 18.04 LTS.

Palavras-chave: DNS, *Domain Squatting*, Portolan, DNS Firewall

Abstract

Cibersecurity is increasingly a concern and requirement for organizations. New threats arise every day and therefore people, processes and technologies should be prepared to respond to them. In order to detect cyber threats, Dognaedis developed Portolan - a cyber intelligence platform. Through it, it is possible to analyze relevant information in real time, as well as to optimize and support the decision making in the scope of the services provided to clients, namely Managed Security Services, Cyber Intelligence and Consultancy.

Domain Squatting consists in registering a DNS domain that verbally and/or visually appeals to popular and trusted brands. Also known as Cybersquatting, this practice began by being used as a way of increasing profits while selling new domains. But soon, malicious actors began exploiting it to gather credentials and spread malware. The project was intended to give to Portolan the autonomy in detecting such illegal activity.

The phenomenon of Domain Squatting has been studied in its various aspects. Based on the knowledge acquired, was developed a strategy of autonomous recognition of this activity with a precision of 81.45 %. The technique was successfully integrated into Portolan. In addition a DNS Firewall was developed that, taking advantage of Portolan's new functionality, was able to provide real-time protection against the threat under study. Its performance varies between 341 and 2331 domains analyzed per second with an average latency per domain of 2 to 200 milliseconds, in a virtual machine with 1 vCPU, 1 GB of RAM and 20 GB of SSD disk running Ubuntu 18.04 LTS .

Keywords: DNS, *Domain Squatting*, Portolan, DNS Firewall

Conteúdo

Lista de Figuras

Lista de Tabelas

Glossário

Blacklist Mecanismo de controlo que permite o acesso a outros elementos (emails, Urls, IPs, domínios), excepto os explicitamente mencionados. XI, 48, 49

Ciber Inteligência É a informação referente a ciber ameaças, depois de recolhida e avaliada no contexto onde esta insere. 4

C&C Servidor responsável por dar instruções de controlo a agentes maliciosos, instalados ou executados em máquinas comprometidas, com o objetivo de realizar as ações pretendidas por um ator malicioso. XI, XIII

Framework É uma camada de abstração que une vários pontos comuns de diversos softwares. Permite a reusabilidade de recursos e elementos. XI, 30

Landing Pages Uma *landing page*, no contexto de segurança da informação, é uma página criada com o objetivo de causar danos ao utilizador final (Exemplo: Phishing). É a página para onde o utilizador é direcionado (*land*), após aceder inadevertidamente a um link suspeito. IX, XI, 27

Malware Termo genérico para qualquer tipo de software informático com intenção maliciosa. 4, 49

Data Mining *Data Mining* é o processo de descoberta de padrões em conjuntos de dados, envolvendo métodos de *machine learning*. 40

Phishing O phishing consiste em utilizar métodos tecnológicos que levem o utilizador a revelar dados pessoais e/ou confidenciais. XI, 3, 4, 40

SandBox *SandBox* é uma abordagem de segurança de isolamento de um sistema, com o objetivo de mitigar possíveis falhas que possam por em causa a Segurança de um Sistema ou Organização. XI, 44

Squatters São atores maliciosos que registam domínios similares ao de uma terceira entidade/-nome, para fins ilícitos. XI, 18, 19

Whitelist Mecanismo de controlo que permite o acesso aos elementos (emails, Urls, IPs, domínios) explicitamente mencionados. XI, 48

Acrónimos

SGBD Sistema de Gestão de Base de Dados. XIII, 31, 33

API Application Programming Interface. 43, 45, 46
AS Autonomous system. 20

C&C Servidor de Comando e Controlo. XI, 15
CA Autoridades Certificadoras. 37

DDos *Distributed Denial of Service*. 14
DNS Domain Name System. 1, 2, 7, 14, 16, 18–21, 56

FN False Negative. 43
FP False Positive. 43

GPL *GNU General Public License*. 42

HSTS HTTP Strict Transport Security. 37

ICANN Internet Corporation for Assigned Names and Numbers. 2
IDNA Internationalized Domain Names in Applications. 18, 38
IPS Intrusion Prevention System. 20

MITM *Man-in-the-Middle*. XIII, 15, 37

RR *Resource Record*. 11, 12

SPF Sender Policy Framework. 9, 51

TLD Top Level Domain. 19–21, 36–38
TP True Positive. 43
TTL *Time-to-Live*. 12

UI *User Interface*. V, 29–32, 35, 45, 53, 55

1

Introdução

Este documento descreve o trabalho realizado na Dognaedis no âmbito do estágio do Mestrado de Informática e Sistemas - Tecnologias de Informação. O objetivo central foi dotar o Portolan - uma ferramenta de ciber inteligência da Dognaedis - de capacidade de detetar e reagir a ameaças que exploram o Domain Name System (DNS) como meio de proliferação.

1.1 A Dognaedis

A DOGNAEDIS é uma empresa focada em cibersegurança e segurança da informação, criada em 2010 por uma equipa de investigadores do CERT-IPN e da Universidade de Coimbra. A equipa fundadora da Dognaedis esteve anteriormente na origem de um CSIRT (Computer Security Incident Response Team), o CERT-IPN, localizado no Instituto Pedro Nunes.

Desde março de 2016 que integra o grupo Prosegur, um dos líderes mundiais no setor de segurança, com presença nos cinco continentes e com mais de 150 000 colaboradores. Para além do SOC (*Security Operations Center*) da Dognaedis a operar em Portugal, a Prosegur tem outros 4 SOC espalhados pelo mundo.

A Dogneadis possui as certificações ISO 27001 - Gestão de Segurança de Informação e ISO 9002 - Gestão da Qualidade.

Apresenta-se como prestadora de serviços especializados e focados na eficiência, mas também

desenvolve um papel fundamental na produção de tecnologias e soluções inovadoras na área de segurança de informação e áreas subjacentes. Maioritariamente uma prestadora de serviços, a Dognaedis oferece serviços em 5 principais áreas: *Cyber Intelligence*, *Managed Security Services*, Consultoria, Auditoria e *Testing* e Tecnologias de Segurança.

1.2 O Portolan

O Portolan é uma plataforma de ciber inteligência, desenvolvida pela Dognaedis, e atualmente utilizada na ajuda ao desempenho das mais diversas atividades praticadas pela empresa. Tem por objetivo ser um agregador de informação de cibersegurança em tempo real de apoio à tomada de decisões relacionadas com a segurança da Dognaedis e dos seus clientes. É uma plataforma modular e adaptável, com capacidade para dar resposta a necessidades emergentes de recolha de informação de múltiplas fontes de informação, como por exemplo: IRC, Pastebin, blacklists ou IDS. Esta informação, depois de enriquecida, é canalizada para os sistemas utilizados na operação diária da empresa.

Sendo o Portolan uma plataforma modular e orientada para a tomada de decisão, este estágio enquadrou-se no desenvolvimento de um módulo que adicione à mesma a capacidade automatizada de deteção e proteção de ameaças que utilizem o DNS como meio de proliferação.

1.3 Apresentação do Problema

Vivemos numa Era em que a tecnologia assume cada vez mais um papel preponderante na vida das pessoas e empresas, estando presente sob as mais variadas formas, desde o telemóvel que anda sempre connosco, ao frigorífico que se encontra ligado à *cloud* e cuja temperatura pode ser controlada remotamente por esse mesmo *smartphone*. Apesar dos utilizadores se basearem em símbolos e palavras (daqui em diante referenciados como domínios - ex: *www.google.com*), para identificar os dispositivos com que interagem, estes encontram-se identificados por endereços numéricos de 32/128 bits (denominados endereços IPv4/v6) de comprimento e formato preconvencionado, explorados pelos protocolos IPv4/v6 (ex: 216.58.201.132). A representação numérica, utilizada no estabelecimento da comunicação entre os diversos dispositivos, não se torna prática para os internautas. Como resposta a este problema, foi desenvolvido em 1985 um sistema (DNS) que possibilitasse identificar os dispositivos comunicantes através de nomes, e por conseguinte, simplificasse a sua utilização por parte dos utilizadores finais da tecnologia TCP/IP.

Dados de 2016 do *Internet Corporation for Assigned Names and Numbers* (ICANN) [13] indi-

cam que durante esse mesmo ano foram registados, no total, mais de 252 milhões de domínios. Este número pode ser justificado pela facilidade no registo e divulgação de domínios e com a «explosão» da Internet.

Contudo, estas facilidades no registo foram aproveitadas por atores maliciosos que viram neste protocolo um recurso valioso para a disseminação das suas campanhas. Uma das técnicas exploradas é o registo de domínios visual e/ou foneticamente semelhantes a domínios já registados ou a entidades existentes (Ex: facebook.com vs facevook.com), com o objetivo de criar uma falsa crença ao utilizador final de estar a aceder ao serviço pretendido. Múltiplas motivações podem existir para este tipo de atividade maliciosa, tais como:

- «Sequestro» de Domínios - Registo de um domínio com o intuito de obter um proveito financeiro junto do proprietário da marca registada.
- *Domain Parking* - Redirecionamento para *sites* com anúncios publicitários, com o fim de gerar tráfego e tirar proveito financeiro.
- *Scam* - Esquema fraudulento usado essecialmente em engenharia social, em que os alvos são direcionados a uma página que acreditam ser confiável.
- *Hit Stealling* - Angariar clientes de concorrentes ou causar danos na imagem dos mesmos.

Este tipo de atividade tem impacto financeiro e reputacional nas entidades visadas. Apesar de existirem instituições que estão sensibilizadas para esta problemática ajudando na resolução de disputas (Ex: WIPO - *World Intellectual Property Organization*), a demora para a resolução das mesmas continua a ser uma razão de preocupação. Se a este facto for acrescentado um tempo de deteção tardio, estão criadas condições para danos avultados. Imagine-se o cenário do banco *bancoexemplo.com*, que foi alvo de uma campanha de *Phishing*, com a utilização da sua imagem, tendo sido registado o domínio *bancoxemplo.com* (vs *bancoexemplo.com* - domínio legítimo) e alojada uma *Landing Page*, cujo objetivo era obter de forma ilícita as credenciais de acesso dos seus clientes. Num cenário normal, o *banco exemplo* tomaria as medidas adequadas apenas quando já existissem clientes afetados. Porém, se tivesse existido uma monitorização ativa do domínio quando este foi registado, a deteção desta ameaça seria antecipada e os danos contidos.

Com a expansão da internet e com a limitação do espaço público de endereçamento IPv4, múltiplas técnicas têm vindo a ser utilizadas para melhor aproveitar o mesmo. Técnicas como NAT ou a utilização de *Reverse Proxies* são alguns exemplos, fazendo com que muitas das vezes não seja exequível a aplicação de medidas de contenção (Ex: criação de uma regra de Firewall) ao nível IP. Devido a esta limitação, uma forma extremamente eficaz e que é utilizada, por exemplo, em bastantes ISPs para controlar o acesso a conteúdos protegidos por direitos de autor, é o bloqueio ao nível do protocolo DNS. Esta técnica poderá também ser utilizada para prevenir a comunicação com domínios cuja reputação indicie um risco para a segurança da infraestrutura

ou dos utilizadores. No exemplo do parágrafo anterior, esta técnica poderia ser utilizada para bloquear o acesso ao *site* malicioso bancoxemplo.com assim que a ameaça fosse inicialmente detetada.

1.4 Objetivos

No âmbito do problema apresentado, o objetivo do presente trabalho é a identificação de um conjunto de técnicas e tácticas maliciosas comumente utilizadas para realizar ataques, tirando partido do protocolo DNS. Finda esta fase, será detalhado um conjunto de soluções, a serem implementadas de forma complementar ao Portolan, com o intuito de detetar e bloquear os vetores de ataque identificados.

A primeira fase deste trabalho, na ótica de uma abordagem preditiva, centra-se no estudo do desenvolvimento de uma tecnologia capaz de prever e detetar, através de técnicas de *Domain Squatting* (secção 2.1.3), ataques baseados em DNS (disseminação de *Phishing*, *landing pages*, fraude, etc) pela análise de domínios recém-criados ou recentemente utilizados na geração de certificados. Desta fase, espera-se que resulte inteligência capaz de alimentar a fase seguinte.

A segunda fase, já numa abordagem reativa, consiste na implementação de um *Recursive* DNS *Server*, ou DNS Firewall (secção 2.1.3), capaz de proteger os utilizadores de ameaças detetadas através da tecnologia desenvolvida na primeira fase, bem como outras ameaças, como por exemplo, comunicação com servidores responsáveis por alojar *Malware*. Esta componente do projeto deverá ainda ser capaz de produzir e recolher *Ciber Inteligência* relevante, como por exemplo a informação de ativos infetados, com base nas tentativas de resolução previamente realizadas pelos diversos ativos.

1.5 Planeamento e Metodologia

O planeamento inicial foi estimado de acordo com a percepção inicial da complexidade do projeto, bem como as estimativas iniciais de alocação de esforço ao mesmo, referidas na tabela 1.1.

Periodo	1º Semestre	2º Semestre
Total Horas	384	736
Distribuição do esforço	34%	66%

Tabela 1.1: Distribuição de esforço por semestre

O projeto, cuja proposta se encontra anexa a este relatório (B), previa uma dimensão de 1120 horas de trabalho, dividas por sete meses de trabalho, tal como ilustra a Figura 1.1. Importa no entanto referir que, em consonância com a entidade de acolhimento deste estágio, o mesmo foi extendido durante um período de doze meses. Esta alteração ao planeamento inicial é justificada pela necessidade de alocação permanente noutros projetos desenvolvidos, no âmbito da atividade profissional do autor na Dognaedis, e que impediram o cumprimento da estimativas iniciais de desenvolvimento deste projeto. Anexo a este relatório (A) encontra-se o gráfico de Gantt, representativo do total de tarefas realizadas e da duração do projeto.

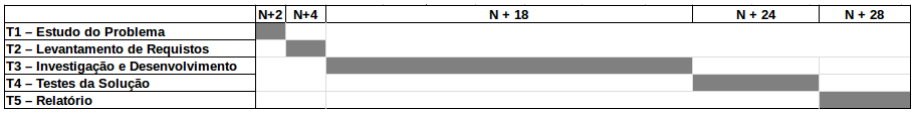


Figura 1.1: Gantt - Estimativa inicial das diversas fases do projeto

Durante a realização deste trabalho foi adotada uma metodologia, baseada em Scrum, utilizando também caraterísticas de outras metodologias como as visualizações Kanban. O Scrum é uma *framework* iterativa e incremental que permite a entrega de produtos de forma faseada, através de ciclos de desenvolvimento mais curtos. Assim, e de uma forma mais fácil, é possível adaptar o produto à visão final do cliente.

A periodicidade definida para os *Sprints* foi mensal, sendo que era habitualmente realizada uma reunião no início de cada nova iteração, bem como uma reunião intermédia para avaliação do progresso. Nessa reunião, o trabalho realizado era revisto bem como novas tarefas a realizar adicionadas à nova iteração.

1.6 Estrutura do Documento

Este relatório encontra-se dividido em 8 capítulos conforme se detalha em seguida:

1. **Introdução** - Este primeiro capítulo serve de enquadramento ao trabalho que foi desenvolvido ao longo do estágio. É apresentada uma descrição do problema e das motivações que levaram à concretização deste mesmo projeto. Ainda aqui são expostos os objetivos definidos assim como o planeamento e a metodologia levados a cabo.

2. **Estado da Arte** - Do segundo capítulo faz parte uma análise detalhada ao trabalho que já foi desenvolvido no âmbito do *Domain Squatting* e tecnologias de proteção associadas.

3. **Análise de Requisitos** - O terceiro capítulo identifica os requisitos funcionais e atributos de qualidade acordados com a entidade de acolhimento deste estágio.

4. **Enquadramento** - No quarto capítulo é dada ao leitor uma contextualização da arquitetura do Portolan e tecnologias de suporte da solução - informação essencial para compreender os capítulos seguintes.

5. **Solução Proposta** - O quinto capítulo apresenta o resultado do estudo realizado e a sua implementação. São apresentados os módulos de criação de inteligência, apresentação de resultados e resposta às ameaças identificadas.

6. **Validação da Solução** - No sexto capítulo são apresentados os testes realizados à solução, bem como os resultados obtidos.

7. **Conclusões** - Por fim, no sétimo e último capítulo, são apresentadas as considerações finais do resultado deste estágio, bem como sugestões de trabalho futuro que poderão complementar a solução.

2

Estado da Arte

Este capítulo apresenta o estudo comparativo de soluções existentes e estudos previamente realizados que se enquadram no âmbito deste estágio. Na secção 2.1 é analisado em detalhe o DNS, os problemas de segurança e respetivas evoluções que este sofreu ao longo do tempo, bem como as preocupações abordadas neste projeto, descritas mais em detalhe nas secções subsequentes.

2.1 Domain Name System (DNS)

O DNS foi criado com o objetivo de providenciar um mecanismo que permitisse a utilização de nomes como referência para dispositivos e serviços em rede. Desta forma é eliminada a necessidade do utilizador memorizar o endereço lógico (IP) associado a um determinado dispositivo, com o qual pretenda estabelecer uma comunicação. Por isso, grande parte das comunicações IP estabelecidas são precedidas por uma consulta (*query*) DNS. Por exemplo, um utilizador que pretenda aceder diretamente à página web do ISEC deverá introduzir no seu navegador (*browser*) a sequência *www.isec.pt* e não o endereço IP (193.137.78.36), cuja composição é mais difícil de memorizar e mais propensa a erros. Ainda que o utilizador tenha a comodidade de fornecer a sequência *www.isec.pt*, o navegador precisa de pedir a sua tradução para o endereço IP que lhe está associado antes de estabelecer a sessão de rede desejada. Para o efeito, consulta um serviço do seu sistema operativo denominado *resolver*.

O DNS foi sofrendo múltiplas evoluções, podendo considerar-se a sua origem no Standford Research Institute, entidade que nos primórdios da Internet era responsável por manter e atualizar o mapeamento entre nomes e endereços numéricos num único ficheiro, o HOSTS.txt. Com o

aumento do número de registos, esta solução deixou de ser viável e escalável. Como alternativa surgiram em 1983 as primeiras especificações deste sistema (RFC 882 e 883), mais tarde atualizadas pelos RFCs 1034 e 1035.

Assim, a solução que foi posta em prática é composta por uma base de dados distribuída e um protocolo de comunicação. Esta infraestrutura, denominada de *Domain Name Space*, organiza-se numa estrutura em árvore invertida, onde cada «folha» contém uma *label* (i.e nome) e a informação associada à mesma (*Resource Records*). Cada folha descende de nós superiores, sendo estes, domínios superiores do respectivo subdomínio associado à folha.

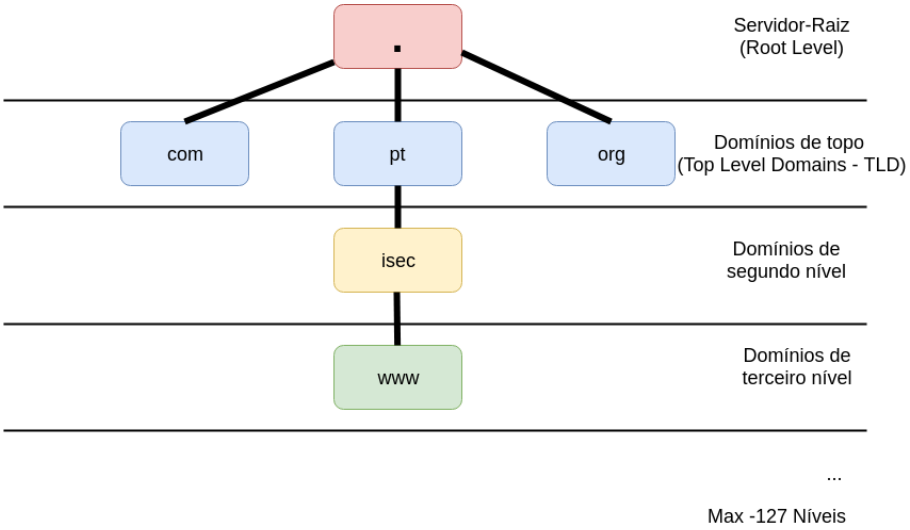


Figura 2.1: *Domain Name Space*

Um domínio é então composto por essa *label* concatenada com as *label*s dos nós superiores, separados pelo caractere (ponto). Por exemplo a *label www*, perfaz o subdomínio *www.isec.pt*, do domínio *isec.pt*, sendo este por sua vez subdomínio do domínio pt, como se pode verificar pela figura 2.1.

De forma a garantir um padrão e uma harmonização do sistema, foram colocadas algumas restrições na criação de domínios. São exemplo: o número máximo de níveis (127); cada *label* conter no máximo 63 caracteres; e um domínio ser composto no máximo por 253 caracteres limitados a [A-Z, a-z, 0-9 e - (hífen)], sendo que o hífen não pode estar presente no início ou fim da *label*.

Como referido anteriormente, a cada nó, para além da *label*, podem estar também associadas diversas fichas de recursos (*Resource Records* - RR), que consistem na informação de registos DNS associados a esse domínio. Existem vários tipos de registos DNS, sendo que os mais utilizados e necessários à compreensão do trabalho realizado, encontram-se presentes na tabela 2.1.

Tipo	Descrição	Exemplo de Utilização
A	Endereço de uma máquina (IPv4)	Exemplo: 8.8.8.8
AAAA	Endereço de uma máquina (IPv6)	Exemplo: 2a00:1450:4003:804::2003
NS	*Authoritative Name Server* - Servidor original responsável por resolver esta zona e por manter os registos originais da mesma.	Exemplo: ns1.google.com
CNAME	«Alcunha» do domínio.	Exemplo: *google.com* -> *www.google.com*
SOA	Indica o *Start of Authority* do domíno. Incluí o DNS server primário (NS), e-mail da entidade/pessoa responsável, número de série utilizado na validação de alterações da zona, bem como alguns tempos utilizados para atualização da mesma.	
PTR	Utilizado em resoluções recursivas (a que domínio está associado este IP).	Exemplo: 8.8.8.8.in-addr.arpa -> *google-public-dns-a.google.com*.
MX	Devolve o(s) servidor(es) de envio e recepção de e-mail*Message Transfer Agent(s) (MTA)* associados a esse domínio.	MX *aspmx.l.google.com*.
TXT	Utilizado para registo de informação diversa associada ao domínio.	Exemplo:*Sender Policy Framework* (SPF).

Tabela 2.1: DNS - *Resource Records*

Para mais fácil compreensão, é apresentada uma simples configuração (2.1) de uma zona DNS no servidor BIND, onde se encontram enumerados alguns dos tipos de registos mais utilizados.

```
$TTL    3h
@       IN      SOA    ns1.exemplo.com. admin.exemplo.com. (
 1         ; Serial
 3h        ; Refresh em 3 horas
 1h        ; Tempo de retry - 1 hora
 1w        ; Expira em 1 semana
 1h )      ; TTL
;
@       IN      NS      ns1.exemplo.com.

exemplo.com.    IN    MX      10      mail.exemplo.com.
exemplo.com.    IN    A       192.168.0.10
ns1                   IN      A       192.168.0.10
www                   IN      CNAME   exemplo.com.
mail                  IN      A       192.168.0.10
```

Listagem 2.1: Exemplo de uma zona DNS Bind

2.1.1 Modo de Operação

Como referido anteriormente, sempre que é solicitado o endereço associado a um domínio, uma sequência de operações é realizada. Esta sequência é ilustrada na Figura 2.2.

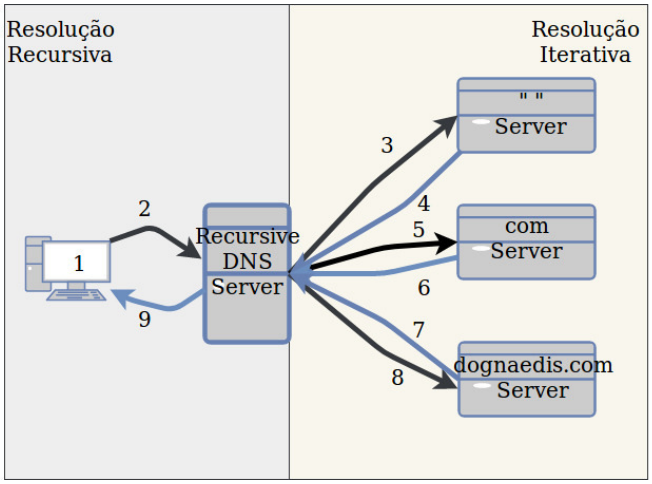


Figura 2.2: Resolução DNS

Legenda do processo de resolução:

1. O serviço ou aplicação, ao tentar aceder a um endereço, questiona numa primeira fase o *resolver* - serviço local de resolução de nomes do sistema operativo local. Este verifica primeiro na sua *cache* se tem a resposta à pergunta solicitada.

2. Caso a questão anterior obtenha uma resposta negativa envia um pedido ao servidor local de DNS configurado no sistema operativo, perguntando qual o endereço IPv4 (A *Record*),

associado ao domínio *dognaedis.com*. Importa referir que o servidor de DNS configurado pode ser um servidor de resolução público (ex: 8.8.8.8 - Google) ou um local. Algumas aplicações como, por exemplo, os *browsers* implementam a sua própria *cache*, evitando operações de resolução ao serviço local.

3. O servidor recursivo de DNS verifica na sua *cache* as zonas pelas quais é responsável por resolver no domínio *dognaedis.com*. Caso não consiga responder ao pedido, o servidor dá início a um processo de resolução recursiva. Na primeira fase submete o nome para resolução a um servidor de raiz (root server). Existem 13 conjuntos de servidores [12] desta natureza em toda a Internet operados por entidades distintas. Os servidores de raiz guardam os endereços do servidores DNS responsáveis por gerir os domínios de topo (.com, .org, ...).

4. Ao receber o pedido (*dognaedis.com*), o servidor de raiz constata que desconhece o nome mas retorna o endereço de um ou vários servidores DNS capazes de resolver nomes registados no domínio .com.

5. O servidor recursivo, de seguida, envia ao servidor autoritativo do domínio .com um pedido solicitando o endereço de *dognaedis.com*.

6. O servidor autoritativo do domínio .com não sabendo qual o endereço IP (A *Resource Record* (RR)) associado ao domínio *dognaedis.com*, responde com o endereço do servidor autoritativo (NS RR) do seu domínio.

7. O servidor recursivo de DNS, por fim, solicita ao servidor autoritativo do domínio *dognaedis.com*, um pedido solicitando o endereço de IPv4 de *dognaedis.com*.

8. O servidor autoritativo do domínio *dognaedis.com* verifica a sua zona, encontra o domínio solicitado e responde com o endereço IP 206.125.169.26.

9. O servidor recursivo de DNS devolve ao cliente que solicitou o pedido o endereço 206.125.169.26 e, por sua vez, o serviço local de resolução de nomes desse cliente transmite-o à aplicação que o solicitou.

DNS *cache*

Como explicado anteriormente, o DNS permite a utilização de nomes para iniciar o estabelecimento de uma ligação, sem a necessidade do conhecimento prévio do endereço IP associado ao mesmo, ficando o sistema operativo do utilizador responsável por obter esse mesmo endereço. Contudo, esta operação tem impacto no desempenho de uma ligação e, de forma a evitar que a cada pedido tenha de ser realizada uma resolução recursiva, os diversos serviços de resolução de nomes podem guardar numa base de dados temporária (*cache*) as resoluções previamente realizadas. De forma a aumentar o desempenho deste protocolo, todos os níveis de resolução (*DNS Resolvers*) pelo qual o pedido passa, armazenam uma *cache*, que é verificada antes de poderem redirecionar o pedido para outro *DNS Resolver*. Esta configuração, sendo opcional, é

definida pelo *Time-to-Live* (TTL) e indica durante quanto tempo a resolução realizada é válida e pode ser armazenada em *cache*. Esta é definida no servidor responsável pela zona, podendo ser adaptada individualmente para cada RR. A cada resolução o *DNS Resolver*, anexa à resposta o respectivo TTL atualizado. Um TTL negativo indica durante quanto tempo a resolução é válida no caso de um RR não existir ou ter sido obtida uma resposta inválida a uma resolução previamente realizada. Um TTL 0 significa que não deve ser armazenada *cache*, enquanto que um TTL positivo indica que deve ser armazenada durante esse número de segundos no caso de uma resposta positiva.

2.1.2 Protocolo de Comunicação

Para que que todas as operações descritas na secção 2.1.1 e restantes operações fossem possíveis, foi definido um protocolo de comunicação que permitisse a transferência de informação. Este protocolo encontra-se implementado sobre UDP e TCP, atendendo o serviço por omissão no porto 53. Para resoluções de DNS a comunicação cliente-servidor é feita através de UDP. As ligações TCP tipicamente são apenas utilizadas quando o tamanho da resposta excede os 512 Bytes ou para outras operações como transferência de zonas entre servidores DNS. As mensagens transmitidas neste protocolo contemplam o seguinte formato:

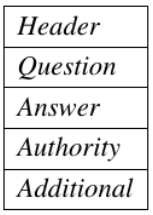


Figura 2.3: DNS - Estrutura de uma mensagem

A secção *Header* encontra-se sempre presente e inclui os campos que especificam quais da secções restantes estão presentes, bem como se a mensagem se trata de uma resposta ou de um pedido. A secção *Question* é apenas usada para descrever a informação associada ao pedido de resolução. Contempla os seguintes campos:

- **QNAME** - O domínio a resolver, representado por uma sequência de *labels*.
- **QTYPE** - Tipo de *Resource Record* pretendido. Exemplo: A - Endereço IPv4 da máquina.
- **QCLASS** - Especificação de qual a classe do pedido. Exemplo: IN - Internet.

Já as secções *Answer*, *Authority* e *Additional* contemplam todas o mesmo formato (Figura 2.4), com um número variável de *Resource Records* declarado na secção *header*.

Name
Type
Class
TTL
RDLENGTH
RDDATA

Figura 2.4: DNS - Estrutura de um *Resource Record*

O campo *Name*, especifica o domínio a que o registo está associado. Por sua vez, o campo *Type* refere-se ao tipo de registo associado à resposta. O campo *Class* especifica a classe da informação contida no campo RDATA. Este último contém a resposta ao pedido realizado que, no caso de um A *record*, contém a informação IPv4. Já o tamanho do *RDATA* é especificado no campo RLENGTH.

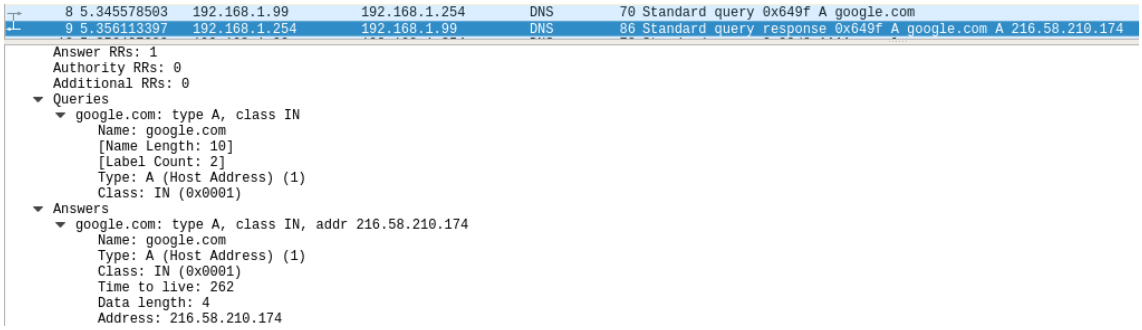


Figura 2.5: Resolução DNS - Captura de tráfego

A figura 2.5 ilustra uma resolução DNS, exemplificando o formato de transmissão de mensagens referido no seu protocolo. A primeira secção (*Query* i.e *Question*) contempla os campos:

- *Name* (i.e. *QName*) - google.com
- *Type* (i.e. *QType*) - A
- *Class* (i.e. *QClass*) - IN

Por sua vez, a secção *Answers* descreve o resultado do pedido realizado:

- *Name* - google.com
- *Type* - A
- *Class* - IN
- *Time to live (TTL)* - 262
- *Data Length* (i.e. RDLENGTH) - 4
- *Address* (i.e. RDDATA) - 216.58.210.174

2.1.3 Preocupações de Segurança no DNS

A relevância do protocolo DNS e o seu modo de operação fazem com seja um alvo de atores maliciosos e uma ferramenta utilizada em atividades que atentam à segurança da informação. De seguida, são descritas algumas das técnicas mais utilizadas pelos atacantes. É também apresentada uma tabela que expõe os possíveis metódos de mitigação e o enquadramento destas técnicas no projeto desenvolvido.

DNS *cache* Poisoning

DNS cache Poisoning é um tipo de ataque que explora vulnerabilidades no protocolo DNS, relacionadas com a impossiblidade de validação da integridade e autenticidade de uma resposta. A sua exploração tem por objetivo desviar utilizadores para servidores ilegítimos. Como referido na secção 2.1.1, para aumentar o desempenho do protocolo, foi implementada uma solução de *cache* que previne que a cada resolução, seja necessário fazer pedidos iterativos desde os *root servers*. Contudo, se esta *cache* for adulterada e uma vez que não existe forma de validar se a resposta obtida é legítima, o servidor de DNS irá fornecer informação incorreta aos seus clientes, podendo por exemplo direcioná-los para servidores maliciosos. Esta falha de segurança pode tomar contornos mais preocupantes se tivermos em conta que a informação adulterada pode propagar-se por vários servidores e não estar apenas contida num ambiente restrito (por exemplo: LAN).

Ataques de amplificação e *Distributed Denial of Service* (DDoS)

Como referido anteriormente, o DNS utiliza principalmente o protocolo UDP como protocolo de transporte. Dado que o UDP é um protocolo *stateless* e não oferece autenticidade quanto à sua origem, esta limitação pode também ser utilizada em ataques de amplificação [33], cujo objetivo passa por causar uma disrupção de serviço num alvo definido previamente. Essencialmente, o ataque consiste em múltiplos pedidos de resolução DNS, cujo o ip de origem é adulterado (*Spoofing*) para o IP da vítima, fazendo com que servidor de DNS Recursivo envie as suas respostas para o mesmo. Tipicamente os atores maliciosos utilizam pedidos de DNS ANY para obterem toda a informação associada a um domínio, fazendo com que o tamanho da resposta seja largamente superior ao pedido inicialmente realizado (amplificação).

DNS *Tunneling*

Os túneis DNS consistem na transmissão de dados associados a outros protocolos ou aplicações camuflados como pedidos e respostas do protocolo de comunicação usado pelo DNS. Tipicamente estão associados a uma fase avançada de um ataque, após o comprometimento de um ou mais sistemas, sendo utilizados na exfiltração de informação ou para receber instruções de

um C&C. Para este tipo de ataque ser possível, um atacante tem de controlar o servidor DNS autoritativo do domínio em causa para que este possa receber e responder ao pedido solicitado.

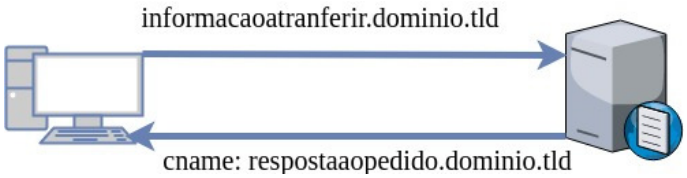


Figura 2.6: Resolução DNS

Suporte para múltiplos *charsets* - Punycode

A representação de domínios encontra-se limitada a um subconjunto do código ASCII. Contudo, esta representação cria limitações na sua utilização por parte de países que usem outras representações nas suas linguagens que não estejam contempladas neste subconjunto (Ex: Mandarim, Cirílico). Foi então criada uma solução, denominada IDN - Internacionalized Domain Names - que consiste num algoritmo de tradução de domínios em Unicode para ASCII. No entanto, esta solução apresenta um risco de segurança para utilizadores menos atentos, dada a similaridade visual de alguns caracteres de outros alfabetos com a representação ASCII, podendo leva-los a aceder a páginas maliciosas sem que estes se apercebam. Por exemplo, pode ser criado um domínio visualmente semelhante ao domínio *mybank.com* (Figura 2.7) com o objetivo de simular a sua aplicação bancária e obter as credenciais dos seus utilizadores.

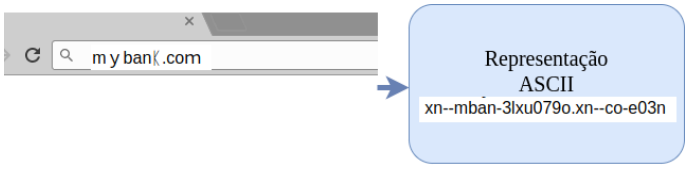


Figura 2.7: Punycode - Exemplo

Questões de Privacidade e *EavesDropping*

Como referido anteriormente, grande parte das comunicações IP são precedidas por uma resolução DNS. Dependendo da forma como estes registos são guardados (pelos diversos servidores de DNS), estes podem ser equiparados a um extrato bancário, registando os acessos de um utilizador ou organização. Apesar destes registos causarem alguma discussão relativamente a questões de segurança, têm um valor inegável em diversas áreas como, por exemplo, o Marketing. Adicionalmente, o protocolo de DNS não oferece qualquer confidencialidade durante a transmissão dos pedidos e respostas, sendo possível através de um ataque *Man-in-the-Middle*(*Man-in-the-Middle* (MITM)) observar, sem qualquer restrição, os pedidos de DNS realizados e, consequentemente, monitorizar a atividade de uma empresa/organização.

Domain Squatting

O *Domain Squatting*, também conhecido como *CyberSquatting*, consiste no registo de um domínio associado a uma marca registada de uma outra entidade, com uma intenção maliciosa. Múltiplas técnicas podem ser utilizadas, desde a simples variação do TLD (Ex: .com, .pt, .org, ...), à troca de apenas um caractere aproveitando um erro de escrita - *typo* (Ex: isec.pt - iseec.pt), ou até mesmo o uso de palavras foneticamente semelhantes (Ex: *for* vs *four*).

Zone Walking

As *DNS Securtiy Extensions* [9] foram introduzidas com o objetivo de mitigar algumas fragilidades no seu protocolo, nomeadamente no que respeita à integridade e autenticidade da informação transmitida. Estas extensões com elas trouxeram novos registos como RSSIG, DNSKEY, DS e NSEC. A fragilidade encontra-se no registo NSEC, que identifica o próximo registo de DNS válido para essa zona, funcionando como uma lista ligada. Desta forma é possível enumerar por completo a lista de subdomínios associada a uma zona de DNS, mesmo que alguns destes domínios não sejam públicos.

DNS AXFR

Para replicação das bases de dados de DNS (zonas), é executado um pedido de DNS Zone Transfer (AXFR) por parte do servidor secundário ao servidor principal. Caso o servidor principal não tenha configurado correctamente quais os endereços dos servidores secundários ou quais os servidores que podem replicar a sua zona, qualquer *resolver* pode obter todos os registos de DNS associados à mesma. Esta má configuração é muitas vezes utilizada por atacantes numa fase inicial de um ataque para realizar a enumeração da infraestrutura alvo, à semelhança da fragilidade descrita anteriormente 2.1.3.

Evolução da segurança no DNS

O DNS foi sofrendo múltiplas evoluções ao longo do tempo e muitas destas tiveram como origem questões de segurança que foram sendo levantadas. Como já foi evidenciado, o desempenho associado ao protocolo foi desde sempre o requisito não funcional obrigatório no DNS, exemplo dessa preocupação é a *cache* no DNS. No seguimento dessa preocupação (desempenho) e para reduzir o impacto nas transferências e atualizações de zonas de DNS, em 1995 surgiu a opção IXFR, que permite que apenas sejam transferidos os registos de DNS atualizados associados. Em 1997 surgiu o RFC2535 que viria introduzir o DNSSEC e que tem como objetivo principal garantir a autenticidade e integridade das respostas obtidas a um pedido DNS, adicionando uma assinatura digital às mesmas. Contudo, o DNSSEC na sua versão original veio introduzir uma vulnerabilidade que permitia enumerar todos os domínios associados a uma determinada zona através do registo NSEC. Em 2008, o RFC 5155 veio introduzir o NSEC3 e assim mitigar esta vulnerabilidade.

Contudo, têm ainda vindo a surgir diversas alterações e propostas de melhoria neste protocolo estando, no entanto, ainda em fase de aceitação ou desenvolvimento. Por exemplo, o RFC 7816[5], para reduzir a informação divulgada durante uma resolução DNS ao longo dos diversos *DNS Servers*, propõe uma minimização dos domínios durante a realização de um pedido de resolução. Por exemplo, se pretender aceder ao endereço de *www.isec.pt*, em vez de enviar *www.isec.pt* no pedido inicial, primeiro ao *root server* é enviada a query *.pt*, de seguida ao servidor responsável da zona *.pt* é enviado o pedido de *isec.pt* e só por fim é enviado ao servidor responsável pela zona isec.pt o domínio completo. Adicionalmente, para garantir a confidencialidade da informação transmitida existem duas propostas *Domain over TLS* RFC 7858 [35], que propõe o uso do protocolo TCP (porta 853) para transmissão da informação cifrada e o RFC 8094 *DNS over Datagram Tranport Layer Security*, que tenta aplicar o conceito anterior mas através do protocolo UDP.

A tabela 2.2 vem enumerar as preocupações de segurança no DNS descritas anteriormente, as soluções existentes e quais dos problemas enumerados são mitigados com o resultado deste estágio.

Vulnerabilidade	**Mitigação**
DNS *cache* Poisoning	DNSSEC
Distributed Denial of Service	Proibição de tráfego outbound *spoofed*. ACLs nos servidores de DNS Recursivos.
DNS *Tunneling*	Monitirização do tráfego DNS, através network IDS/IPS.
Punycode	Apesar de algumas aplicações (ex: Browsers) permitirem que os domínios internacionais apareçam no formato punycode, esta solução não é ainda um standard. Este projeto vai permitir detetar e bloquear comunicações a domínios suspeitos punycode.
Questões de Privacidade e *Eaves-Dropping*	QNAME e Domain over TLS/DNS over DTLS.
Domain Squatting	O estágio aqui reportado pretende estudar esta vulnerabilidade, comparando soluções e propondo estratégias complementares que melhorem a mitigação da mesma.
Zone Walking	NSEC3
AXFR	Configuração dos IPs autorizados a realizar a transferência de zona.

Tabela 2.2: Preocupações de segurança no DNS e medidas de mitigação.

2.2 O Domain Squatting e DNS Firewall

Durante o estudo realizado sobre o *Domain Squatting* foram identificadas as técnicas mais comuns e utilizadas por *Squatters*. Dessa análise foram identificadas:

- *Typo Squatting* - técnica descrita por Banerjee, Rahman e Faloutsos [4]. Nesta técnica o ator malicioso tipicamente explora erros na escrita do domínio (typo) por parte do utilizador. Os erros mais comuns e que, por sua vez, são mais explorados são:
  - Duplicação de caractere. Exemplo: *isec.pt* vs *iisec.pt*.
  - Substituição de caractere por outro. Exemplo: *isec.pt* vs *1sec.pt*.
  - Omissão de caractere. Exemplo: *isec.pt* vs *isc.pt*.
  - Adição de um caractere. Exemplo: *isec.pt* vs *isec1.pt*.
  - Troca de um caractere por um caractere adjacente do teclado. Exemplo: *isec.pt* vs *usec.pt*.
- *Sound Squatting* - técnica descrita por Nikiforakis et al. [21]. Esta técnica visa explorar erros linguísticos através da substituição de sílabas ou palavras por outras foneticamente semelhantes.
- *Bit Squatting* - técnica descrita por Spaulding, Upadhyaya e Mohaisen [28]. No *Bit Squatting* o atacante tenta explorar falhas de *hardware*, nomeadamente em memória. Por exemplo, se um pente defeituoso de memória alterar um bit durante a cópia de um domínio antes da sua resolução, irá alterar por completo o domínio e por sua vez o endereço a contactar. A taxa de sucesso deste tipo da técnica é mais baixa que as restantes, contudo se pensarmos na possibilidade desta falha de hardware estar presente por exemplo num *DNS Server* ou Proxy, esta taxa poderá aumentar.
- *Internationalized Domain Names in Applications* (IDNA) - técnica descrita por Baasanjav [3]. O DNS suporta apenas um conjunto de caracteres ASCII. Assim, outras línguas que utilizam caracteres não contidos nesse subconjunto (Ex: Chinês, Cirílico) não são nativamente suportados. De forma a dar suporte aos mesmos, foi criado um algoritmo que permite converter do conjunto Unicode para o conjunto ASCII suportado pelo DNS. A exploração desta técnica consiste na utilização de caracteres visualmente semelhantes (i.e. homoglífos) e que quando renderizados por um cliente (Ex: *Browser*) aparentam ser outros caractares.

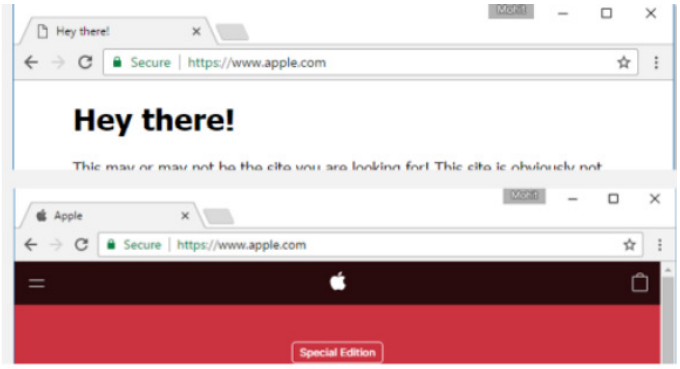


Figura 2.8: IDNA - apple.com

Na Figura 2.8, é demonstrada a prova de conceito desta técnica. Enquanto que, em ambas as janelas do *Browser*, visualmente os domínios aparentam ser idênticos (*apple.com*), na prática o primeiro domínio corresponde a *xn–80ak6aa92e.com*. Esta ilusão é possível pela utilização de caracteres não permitidos no DNS, o que torna necessária a utilização deste algoritmo de codificação. O resultado são dois domínios visualmente idênticos mas que, sob um registo de DNS, representam dois recursos diferentes.

- Troca de *Top Level Domain* (TLD) - Consiste na troca do TLD por outro. Exemplo: *isec.pt* vs *isec.com*.

De estudos realizados por diversos autores e que serviram de base para a execução deste trabalho, resultaram conclusões bastante similares. A maioria dos *Squatters* têm preferência em domínios com o TLD «.com» como alvos, domínios de curta dimensão, habitualmente 10 caracteres e pertencentes ao Top da Alexa [1]. Este *ranking* resulta do tratamento da informação recolhida por este serviço da Amazon, proveniente dos seus utilizadores e é definido através da combinação de visitantes únicos/páginas vistas por website. Num estudo de Khan et al. [16] realizado em 2015, foram recolhidos os logs provenientes de resoluções DNS, bem como de *proxies* HTTP de um Campus Universitário e foi criada uma metodologia que permitiu identificar potenciais domínios de *Typo Squatted*, através da comparação do acesso URLs/Domínios semelhantes num curto espaço de tempo. Desta análise foi possível confirmar que numa grande percentagem dos casos identificados, o utilizador é redirecionado para o domínio legítimo após o acesso inicial ao domínio suspeito. Foi assim possível identificar quais as técnicas mais utilizadas pelos atores maliciosos.

Numa outra abordagem por Banerjee, Rahman e Faloutsos [4], foi extraído o top 900 do Alexa, criando de seguida mais de 3 milhões de variações desses mesmos domínios utilizando técnicas simples como adição ou remoção de um, dois e três caracteres, bem como permutações e substituições de caracteres. Após a criação destes domínios, os respetivos URLs foram acedidos. Dos resultados obtidos, surgiram as seguintes condições para classificação de domínios *squatted*:

- Site Suspeito com tamanho <= 10 diferindo para o domínio original apenas num caractere;
- Número de HTTP *Redirects* durante o acesso ao site suspeito >= 6;

- Durante o acesso ao site suspeito, existe um redirecionamento final para um TLD biz, net, org;
- *Autonomous system* (AS) entre 2600-2800 ou 5100;
- Tamanho total da página suspeita <=31 KB;
- Durante o acesso ao site suspeito um *popup* surgia com o dominio casalemedia/sedoparking.com;
- Presença de palavras suspeitas no conteúdo da página.

Já por Szurdi et al. [30] foi adotada uma abordagem idêntica. Contudo cingiu-se ao TLD .com para seleção dos domínios originais. Posteriormente, foram aplicadas variações como remoção do caractere (ponto) utilizando prefixos comuns (exemplo: www, ftp), omissão de caracteres, permutações, substituições, duplicações de caracteres, adição e remoção de um caracter. Mais tarde, um conjunto de domínio maliciosos «.com» foi retirado de diversas blacklists. Ambos os *datasets* foram enriquecidos com atributos associados às resoluções de DNS (Geo localização, AS, etc) e informações possíveis de obter através do protocolo WHOIS (Registrar, Registrant, etc). De seguida, foi desenvolvido um algoritmo de classificação baseado nos resultados obtidos.

Outras abordagens foram estudadas, quer de classificação de domínios maliciosos, quer de classificações de páginas *Phishing* - técnica esta bastante utilizada em casos de *Squatting*. Num estudo realizado por Munro [20] a classificação centrou-se numa análise baseada em atributos gerados a partir do próprio domínio, como o número total de dígitos no domínio, TLD, número de vogais ou número de consoantes.

Já numa perspetiva reativa de contenção de comunicações de DNS suspeitas, surgiu em 2010 o conceito de Response Policy Zones (RPZ), também conhecido como DNS Firewall. Este conceito originário do servidor DNS Bind, e que vem desde então sendo adaptado e aplicado noutros produtos e soluções, permite que um servidor de DNS altere as suas respostas de acordo com o pedido realizado, prevenindo assim que comunicações maliciosas seja estabelecidas. Soluções como *Intrusion Prevention System* (IPS), como por exemplo o Snort, também têm a capacidade de bloquear tráfego DNS considerado suspeito, contudo dada a necessidade de inspeção completa do tráfego TCP/UDP, faz com que habitualmente esta não seja primeira opção no que ao DNS diz respeito.

2.3 Produtos Semelhantes

Durante a análise de mercado realizada não foram identificadas soluções que agregassem a capacidade de identificação e análise de domínios associados a *Domain Squatting* e a capacidade de prevenção. Então após o estudo anterior 2.2, foram examinadas diversas soluções também nestas duas perspectivas: a identificação de domínios suspeitos e o bloqueio de comunicações

DNS maliciosas. A tabela 2.3 apresenta o resultado relativo à primeira perspetiva referida, no que respeita às técnicas suportadas:

Transformação	**Typo Finder[11]**	**DnsTwist [31]**	**UrlCrazy [32]**
Adição de um caractere	✓	✓	✓
Adição de um subdomínio	✓	✓	✓
Duplicação de caractere	✓	✓	✓
Supressão de caractere	✓	✓	✓
Troca de caractere	✓	✓	✓
Adição de um ponto extra	✓		
Remoção de um ponto			✓
Homoglifo	✓	✓	✓
Pluralização de palavras	✓	✓	✓
Permutação TLD	✓	✓	✓
BitSquatting	✓	✓	✓
Adição de Hífens	✓	✓	
Erros ortográficos comuns			✓
Homofonos			✓
Punycode			✓
Análise genérica por palavras chave			
Keyboard Typos	✓	✓	

Tabela 2.3: Análise ferramentas de *Squatting*

Já relativamente à segunda perspetiva sobre as soluções ativas e prevenção de comunicações com domínios maliciosos foram identificadas as seguintes soluções:

- **Cisco Umbrella**
- **BlueCat DNS**
- **Efficient IP**
- **InfoBlox**

As soluções identificadas apresentam-se todas como capazes de mitigar ameaças baseadas em DNS, contudo destas destaca-se a solução Cisco Umbrella, pela possibilidade de gestão cloud e possibilidade de aplicar os mesmos controlos de segurança fora do perímetro de controlo, através de agentes instalados nos diversos dispositivos terminais.

3

Análise de Requisitos

Este capítulo enumera e descreve os requisitos definidos pela Dognaedis enquanto entidade acolhedora deste projeto de estágio. Os requisitos foram recolhidos após diversas reuniões com o *Product Owner* da Dognaedis. São, então, descritos primeiramente os requisitos funcionais e, de seguida, os atributos de qualidade que deveriam estar presentes no produto final.

3.1 Requisitos Funcionais

As tabelas abaixo descrevem os diversos requisitos funcionais que deveriam ser cumpridos para que o resultado obtido na nova extensão do Portolan fosse considerado positvo.

ID	FR01
Titulo	Coletor de domínios recém-criados.
Descrição	O sistema deve ser capaz de receber listas de domínios recém-criados e adaptar os resultados obtidos à taxonomia existente no ambiente onde será inserido.
Dependências	

Tabela 3.1: FR01 - Colector de domínios recém-criados.

ID	FR02
Titulo	Identificação de domínios considerados suspeitos.
Descrição	O sistema implementado deve ser capaz de poder identificar o registo de domínios que possam ser utilizados para usurpação de uma entidade, através de técnicas de *Domain Squatting*.
Dependências	FR01

Tabela 3.2: FR02 - Identificação de domínios considerados suspeitos.

ID	FR03
Titulo	Implementação de um sistema classificador de domínios.
Descrição	O sistema implementado deve ser capaz de poder classificar os domínios previamente identificados (malicioso/não malicioso) de acordo com indicadores de suspeição.
Dependências	FR02

Tabela 3.3: FR03 - Implementação de um sistema classificador de domínios.

ID	FR04
Titulo	Análise automatizada de *Landing Pages*
Descrição	O sistema deve ser capaz de poder analisar o conteúdo HTTP associado ao domínio suspeito de forma a confirmar a maliciosidade do domínio.
Dependências	FR02

Tabela 3.4: FR04 - Análise automatizada de *Landing Pages*

ID	FR05
Titulo	Consulta dos resultados da Análise
Descrição	Deve ser dada a possibilidade ao utilizador do Portolan de consultar os resultados obtidos, como por exemplo nome do domínio, data do registo, entidade a que pode estar associado, classificação e captura de ecrã da página (se existir).
Dependências	FR03 e F04

Tabela 3.5: FR05 - Consulta dos resultados da Análise

ID	FR06
Titulo	Possibilidade de marcar um resultado como Falso Positivo.
Descrição	Deve ser dada a possibilidade ao utilizador do Portolan de consultar os resultados obtidos e marcar um resultado como falso-positivo.
Dependências	FR05

Tabela 3.6: FR06 - Possibilidade de marcar um resultado como Falso Positivo.

ID	FR07
Titulo	Adicionar/Editar informação de Entidades a Monitorizar.
Descrição	O administrador deverá conseguir adicionar novas entidades a monitorizar nos registos de DNS, bem como editar a informação já existente.
Dependências	

Tabela 3.7: FR07 - Adicionar/Editar informação de Entidades a Monitorizar.

ID	FR08
Titulo	Remover Entidades a Monitorizar.
Descrição	O administrador deverá conseguir remover entidades a monitorizar nos registos de DNS.
Dependências	

Tabela 3.8: FR08 - Remover Entidades a Monitorizar.

ID	FR09
Titulo	Implementação do servidor de DNS recursivo
Descrição	Deverá ser implementado um servidor de DNS recursivo onde serão feitas as resoluções de DNS.
Dependências	

Tabela 3.9: FR09 - Implementação de DNS recursivo

ID	FR10
Titulo	Bloqueio do acesso a domínios maliciosos.
Descrição	Quando solicitado o acesso a um domínio malicioso, este deverá ser negado através do protocolo DNS, de forma a mitigar riscos para a segurança.
Dependências	FR09

Tabela 3.10: FR10 - Bloqueio do acesso a domínios maliciosos.

ID	FR11
Titulo	Alimentação automáticas das *Blacklists* no Servidor de DNS.
Descrição	Deverá ser possível, de forma periódica, atualizar as listas de Domínios Maliciosos que o servidor de DNS não deverá resolver.
Dependências	FR09

Tabela 3.11: FR11 - Alimentação automáticas das *Blacklists* no Servidor de DNS.

3.2 Requisitos Não Funcionais/Atributos de Qualidade

Nesta secção são descritos os diversos atributos de qualidade, definidos pela entidade acolhedora, que deveriam estar presentes no projeto divididos em três categorias: Performance, Segurança e Escalabilidade.

3.2.1 Performance

ID	Q01
Titulo	Análise rápida aos novos domínios.
Descrição	De forma a que detecção e possível reacção a potenciais ameaças seja célere, o sistema deverá ser capaz de analisar os domínios recém-criados de forma, num curto período.

Tabela 3.12: Q01 - Análise rápida aos novos domínios.

ID	Q02
Titulo	Baixa latência nas resoluções DNS.
Descrição	Sempre que for feita uma resolução de DNS através do DNS Resolver, irá ser feita uma análise às diversas blacklists existentes para validar se o domínio solicitado é ou não malicioso. Esta operação deverá ser breve e optimizada.

Tabela 3.13: Q02 -Baixa latência nas resoluções DNS.

3.2.2 Segurança

ID	Q03
Titulo	Validação de campos de entrada na UI.
Descrição	Injecção de código é uma vulnerabilidade de software e dada a natureza do mesmo, deverá ser evitada e mitigada.

Tabela 3.14: Q03 - Validação de campos de entrada na UI.

ID	Q04
Titulo	Cuidados na análise do HTML de páginas suspeitas.
Descrição	Uma técnica utilizada por atores maliciosos é a utilização de código malicioso para explorar vulnerabilidades em Browsers de clientes, bem como nos dispositivos de rede a que estes estão ligados. Deverão existir cuidados para evitar que esse risco seja explorado.

Tabela 3.15: Q04 - Cuidados na análise do HTML de páginas suspeitas.

ID	Q05
Titulo	Anonimização no acesso a *Landing Pages*
Descrição	De forma a evitar que o potencial ator malicioso consiga identificar a origem do pedido, o acesso deverá ser feito através de mecanismos de anomização, como redes de proxies ou VPN.

Tabela 3.16: Q05 - Anonimização no acesso a *Landing Pages*

3.2.3 Escalabilidade

ID	Q06
Titulo	O DNS Resolver deverá poder escalar horizontalmente.
Descrição	No caso de grande afluência de pedidos de DNS deverá ser possível instalar mais instâncias sincronizando a configuração entre as mesmas.

Tabela 3.17: Q06 - O DNS Resolver deverá poder escalar horizontalmente.

4

Enquadramento

Este capítulo serve de contexto à solução desenvolvida no âmbito do estágio realizado, estando divido em duas secções. A primeira secção descreve a arquitectura do Portolan, enquanto que a segunda descreve as principais tecnologias utilizadas durante a implementação do produto resultante deste estágio.

4.1 O Portolan

Como referido na secção 1.2, o módulo de *Domain Squatting* irá ser inserida no Portolan, um produto da Dognaedis e, como tal, deverá obedecer à sua arquitetura. O Portolan é desenvolvido em Python e dividido em duas áreas base, o Core e a UI, conforme referido na Figura 4.1. O Core é responsável por todo o processamento de informação, enriquecimento e armazenamento, enquanto que a UI fornece a capacidade de orquestração e operacionalização do Core, consulta da informação e sua disseminação.

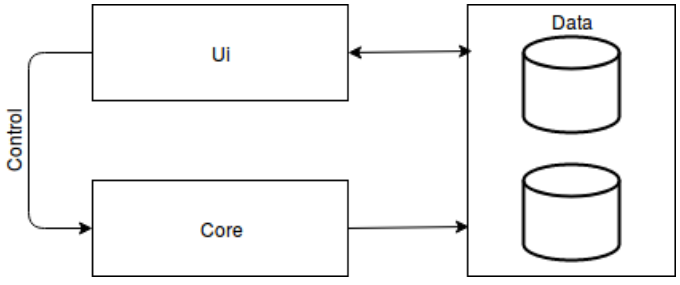


Figura 4.1: Arquitectura Portolan

4.1.1 Core

O Core é responsável por obter a *Security Intelligence* do Portolan. Dada a diversidade de informação e fontes, este foi desenhado para ser modular e extensível. Na figura 4.2 é apresentado o seu diagrama de comunicação. Os conceitos necessários para a sua compreensão são os seguintes:

- **Bot** - O Bot é uma unidade atómica responsável por realizar uma tarefa. Exemplo: Recolher domínios maliciosos da lista malwaredomainlist.com.
- **Pipeline** - Conjunto de bots. Este conjunto é essencial no enriquecimento da informação recolhida pelo primeiro Bot da sequência.
- **Sink** - Adicionado ao final da Pipeline e consiste no destino final da informação. (Ex: Servidor Syslog, Base de Dados.)

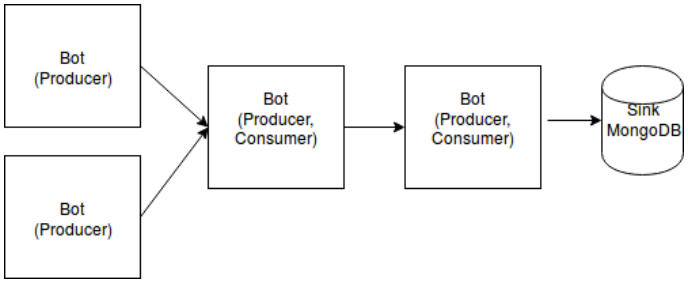


Figura 4.2: Portolan - Exemplo Pipeline

A modularidade do Portolan permite que sejam adicionadas novas fontes de dados, com um baixo tempo de desenvolvimento. Esta característica é possível uma vez que cada Bot descende de um Bot abstracto (StreamBot) responsável pela implementação de todas funcionalidades comuns, garantindo também assim a conformidade com os padrões de software da solução.
Existem dois tipos de Bots, *Producer* e *Consumer*, *Consumer*. Os *Producers* são o primeiro elemento de uma Pipeline, sendo apenas responsáveis por recolher a informação associada à sua origem, enquanto que os *Producer* e *Consumer* recebem a informação de outro Bot e fazem o seu enriquecimento (ex: realizar o mapeamento de IP, Domínio) e enviam o resultado para o elemento seguinte (Bot ou Sink). A comunicação entre Bots é realizada através de Redis.

4.1.2 User Interface (UI)

A UI (cujo funcionamento lógico pode ser visto na figura 4.3) é responsável pela configuração das diversas Pipelines e instruções aos diversos Bots (Start, Stop), bem como a consulta da informação e operacionalização dos diversos módulos do Portolan. É uma interface Web, desenvolvida também em Python através da *Framework* Django e à semelhança do Core é também ela modular, sendo composta por diversos módulos denominados de Aplicações (Apps).

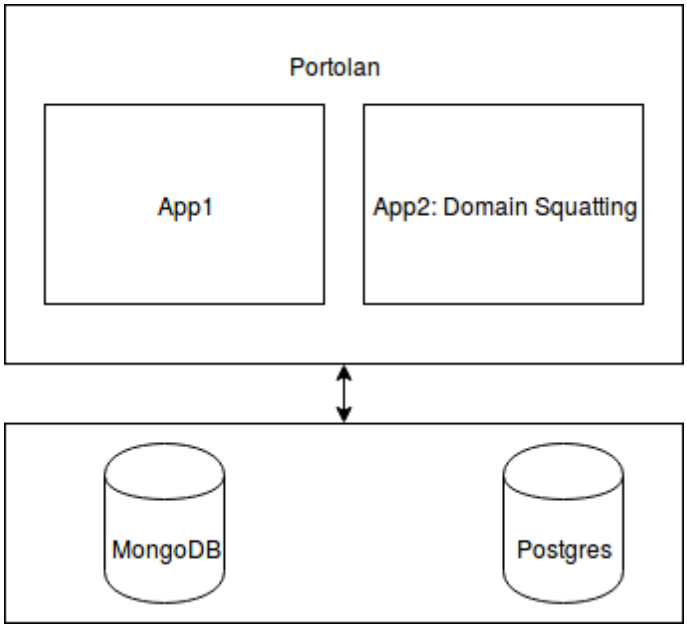


Figura 4.3: Portolan - UI

A UI utiliza dois SGBD, um relacional (PostgreSQL) e outro não relacional (MongoDB). A relacional está integrada directamente com o Django e é responsável por armazenar todas as configurações, enquanto que a base de dados MongoDB armazena todos os eventos obtidos pelo Core.

4.2 Tecnologias de Suporte

Esta secção apresenta e descreve as principais tecnlogias (*frameworks*, bibliotecas, software) utilizadas durante da implementação do produto final deste estágio. É apresentada uma breve descrição, bem como um enquadramento da mesmas no contexto do trabalho desenvolvido.

4.2.1 Python

O Python [26] é uma linguagem de programação *Open Source* de alto nível, de *script* e orientada a objetos. Foi criada por Guido van Rossum em 1991 e gerida pela organização Python Software Foundation. No contexto do projeto desenvolvido foi utilizado no Portolan Core, bem como nos demais scripts de suporte criados para integração com a DNS Firewall.

4.2.2 Django

Django[7] é uma *framework* desenvolvida em Python e lançada em 2005, que utiliza o padrão Model-Template-View no desenvolvimento de aplicações web. Os modelos (*Model*) permitem

um mapeamento de objetos com a camada de dados (ORM). As vistas (*views*) representam os dados que são apresentados ao utilizador, mais precisamente que informação é apresentada. Já os templates são a camada de apresentação dos dados que são mostrados ao utilizador. Os templates apresentam a informação que é retornada pelas respetivas vistas. No contexto deste estágio o Django é a *framework* utilizada na UI no Portolan.

4.2.3 Flask

O Flask[10] é uma *framework* web Python, basead nas bibliotecas WSGI e JINJA2. Dada a sua flexibilidade e simplicidade é aconselhada para o desenvolvimento de pequenas aplicações web ou webservices. Esta *framework* foi utilizada para o desenvolvimento de um webservice, solicitado durante a análise HTTP, que é responsável por fazer uma análise ao Website suspeito num ambiente isolado (sandbox).

4.2.4 Selenium

Selenium[27] é uma biblioteca multi-plataforma frequentemente utilizada em testes automatizados de software de plataformas Web. Esta é composta por um *driver*, que interage com um *Browser* por forma a executar as acções solicitadas (ex: clicar num link ou submissão automatizada de um formulário). Foi utilizada na análise realizada aos Websites suspeitos, sendo usada essencialmente a funcionalidade *save_screenshot*.

4.2.5 OpenCV

A OpenCV[15] é biblioteca criada pela Intel em 2000, utilizada no processamento de Imagens, Estruturas de Dados ou Álgebra Linear. Foi usada durante o processo de análise de imagens do Bot HTTP Classifier.

4.2.6 Redis

O Redis[25] é uma estrutura de dados em memória, utilizada como Base de Dados, Cache ou *Message Broker*. É utilizada em dois contextos diferentes: como *Message Broker* durante a comunicação entre Bots do Portolan e como Base de Dados, para armazenamento dos domínios em *Blacklist/Whitelist* na DNS Firewall.

4.2.7 PostgreSQL

O PostgreSQL[24] é um SGBD *open source*. É utilizado no Portolan UI, para armazenamento da informação geral da aplicação, como configurações, ou resultados identificados no âmbito deste projeto.

4.2.8 Unbound

O Unbound DNS[22] é um servidor de DNS recursivo, criado pela empresa NLnet Labs. Devido ao seu desempenho e capacidade de extensão, através de módulos de Python, foi utilizado para implementar a DNS Firewall.

4.2.9 Weka

O Weka[34] é um software desenvolvido na Universidade de Waikato na Nova Zelândia e é composto por uma coleção de algoritmos de *machine learning*, bem como ferramentas de tratamento de dados, classificação, regressão, *clustering* e visualização. Durante este projeto foi utilizado no processo de *Data Mining* para desenvolvimento do algoritmo de classificação de domínios maliciosos.

5

Solução Proposta

O módulo de *Domain Squatting* foi criado com o objetivo de identificar domínios que possam consistir numa ameaça para as organizações. Inserindo-se diretamente no Portolan, no seu Core estão implementados 5 Bots responsáveis pela identificação de domínios potencialmente suspeitos, a sua análise e classificação. A UI permite a configuração deste módulo e análise de resultados. Uma visão global da arquitetura deste módulo é referida na Figura 5.1.

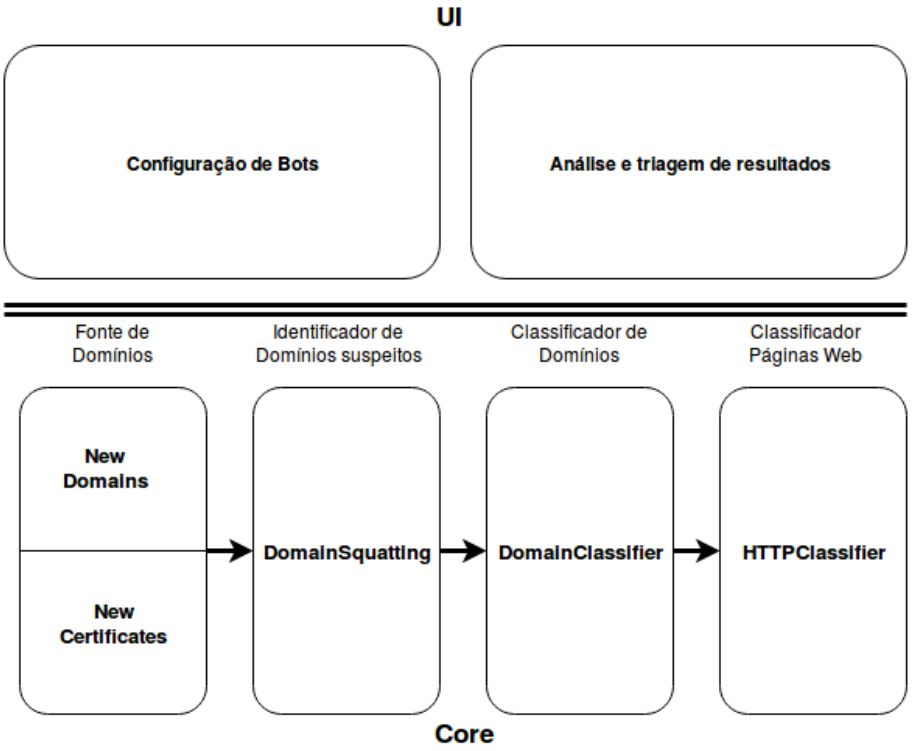


Figura 5.1: Arquitetura Domain Squatting

5.1 Domain Squatting - Core

Nesta subsecção são apresentados os diversos Bots implementados, resultantes do estudo realizado no âmbito do *Domain squatting*, análise e classificação de domínios.

5.1.1 Feed Domínios - NewDomains

O primeiro elemento da Pipeline de *Domain Squatting* é o bot *NewDomains*. Este é responsável por recolher periodicamente uma lista de domínios recém-registados para posteriormente serem analisados. A maior dificuldade na implementação deste Bot foi identificar uma fonte confiável e completa. Desde 2012, com o início do programa de novos TLDs do ICANN [14], que novos domínios têm sido delegados a *Registrars*, sendo que atualmente já existem mais de 1500 disponíveis. Apesar de uma parte destes novos TLDs ser privada, estando associado a marcas ou grupos e desta forma dever em teoria estar sujeito a um escrutínio aquando do registo, a lista disponível é extensa, e para uma monitorização eficaz foi necessário encontrar uma fonte o mais completa possível. Desta forma fez-se uma comparação de algumas fontes disponíveis, cujo requisito obrigatório era a sua extração via HTTP/HTTPS.

Fonte	Total de TLDs Anunciados
Whoxy	2026
Whoisds	550
whoisxmlap	1250

Tabela 5.1: Comparação fontes de novos Domínios Registados

Apesar de, aparentemente, a fonte mais completa dos candidatos avaliados ser a Whoxy, após uma análise mais detalhada, verificou-se que dos 2026 TLDs anunciados apenas 1437 continham registos, o que vai ao encontro da lista total de domínios já delegados pelo ICANN. Ainda assim, este candidato mostrou-se ser o mais completo e, por isso, acabou por ser o escolhido como fonte para a implementação.

Uma vez que a lista de domínios é atualizada diariamente, o bot foi implementado de forma a ser executado a cada 24 horas. Durante a sua execução, este realiza um pedido HTTP extraindo o ficheiro diário de atualizações de novos registos de domínios. Após esta operação este ficheiro é tratado enviando os seus resultados para o nó seguinte da pipeline (DomainSquatting - 5.1.3).

5.1.2 Feed Domínios - NewCertificates

Durante a investigação realizada foi possível perceber que, em muitos casos, os atores maliciosos apenas iniciam a campanha alguns meses depois do registo do domínio. Outro indicador que foi possível perceber em fontes externas de *Phishing* foi que, em alguns casos os atores maliciosos reutilizavam um domínio genérico já existente numa nova campanha, criando apenas um novo subdomínio, direcionado a esta nova campanha. Tendo em conta estes indicadores, identificou-se uma limitação na solução de monitorizar apenas os domínios recém-criados e, por isso, foi criado um Bot capaz de recolher informação associada à criação/renovação de certificados SSL/TLS, permitindo assim complementar em parte a abordagem inicial.

A implementação deste Bot encontra-se assente no protocolo *Certificate Transparency*[17], RFC 6962. Este protocolo surgiu devido à necessidade de colmatar algumas lacunas na emissão de certificados, nomeadamente a possibilidade da emissão e atribuição de certificados relativos a domínios não pertencentes a quem os solicitou, por *Autoridades Certificadoras* (CA). Por exemplo, em Agosto 2011[18] a CA TURKTRUST inadvertidamente emitiu dois certificados *.google.com, tendo este erro apenas sido detectado em Dezembro de 2012. Na sua essência, a implementação deste protocolo consiste no anúncio da emissão de um determinado certificado pelas CA. Estes logs são agregados de forma pública podendo posteriormente ser usados tanto por diversas aplicações, como pelos proprietários dos domínios. Este método de funcionamento não vai inviabilizar erros como o de 2011, mas vai permitir uma redução do tempo de resposta aos mesmos. Atualmente, diversos *Browsers* já utilizam estes logs para validação dos

certificados através de uma técnica chamada *Certificate Pinning*, que essencialmente verifica a correspondência do certificado enviado pelo servidor num pedido HTTPS com a assinatura do mesmo contido nos logs de *Certificate Transparency*, sendo retornado um erro de *HTTP Strict Transport Security* (HSTS) caso não exista correspondência, evitando assim ataques MITM.

Assim, o Bot *NewCertificates* liga-se com uma periodicidade de uma hora aos repositórios públicos de logs, extraindo a informação relevante dos novos certificados emitidos ou renovados, nomeadamente:

- CN - Domínio(s) para o(s) qual/quais este certificado é válido;
- *Not Before* - Data de início de validade do certificado;
- *Not After* - Data limite de validade do certificado;
- CA - Entidade Certificadora.

5.1.3 Identificador de Domínios Suspeitos - DomainSquatting

A função principal deste Bot é identificar domínios suspeitos (semelhantes ou possivelmente associados a uma entidade) e descartar os restantes. Esta identificação é realizada com o recurso a 3 atributos: domínio, nome da entidade e nome de produtos/marcas. A utilização destes mesmos atributos permitiu adicionar a capacidade de deteção de outros domínios suspeitos, mesmo quando estes diferem bastante do domínio original, estando contudo relacionados à mesma entidade. Por exemplo, a Dognaedis é titular do domínio dognaedis.com, caso a análise fosse apenas realizada recorrendo ao domínio, apenas poderiam ser detetadas permutações do mesmo. Contudo, caso ocorresse registo de um domínio contendo o nome Portolan, não seria possível detetar este evento.

Para desenvolvimento deste Bot foram estudadas as técnicas mais utilizadas por atores maliciosos sendo então possível, através da aplicação das mesmas neste Bot, gerar domínios que pudessem apresentar risco para as entidades visadas. O maior desafio na identificação dos domínios foi o número elevado de combinações que seriam necessárias gerar caso fossem aplicadas múltiplas técnicas simultaneamente, o que levaria à criação de um *dataset* de grande dimensão para os domínios a monitorizar. Por exemplo, para o domínio dognaedis.com o algoritmo criado para a geração de *typos* retorna 448 resultados. Se a estes fosse aplicada também a técnica de troca de TLD, teríamos 448*1500 combinações e assim sucessivamente por cada técnica aplicada. Inicialmente esta abordagem foi testada mas em termos computacionais era bastante exigente (não cumprindo o requisito Q01). Para além da limitação anterior, esta abordagem também não era totalmente eficaz pois continuavam a existir técnicas que poderiam não ser detetadas, tais como a combinação de IDNA com permutação de TLD. Por estes motivos optou-se por adotar uma abordagem alternativa, ilustrada no diagrama de fluxo da figura 5.2.

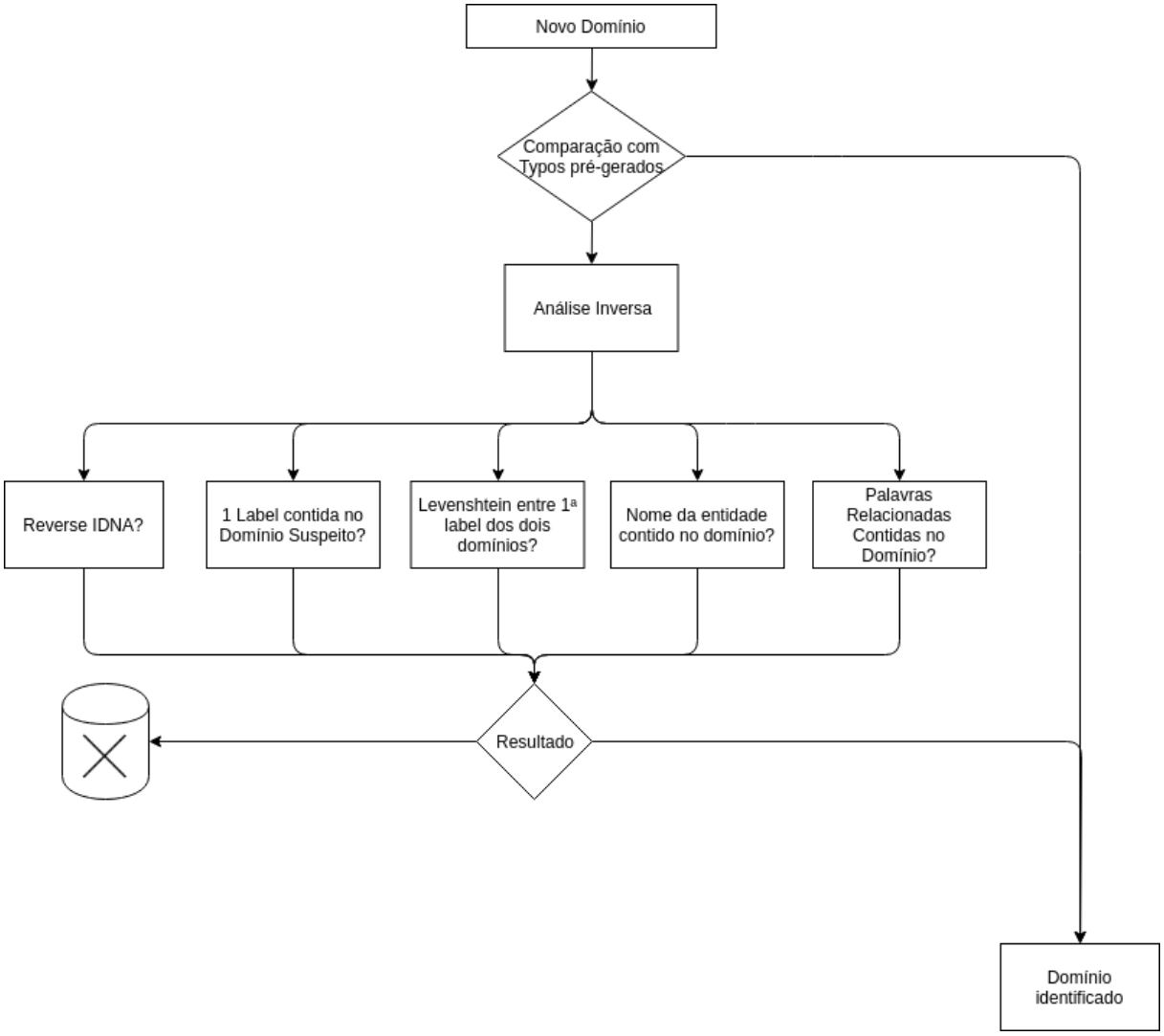


Figura 5.2: Análise do Bot *Domain Squatting*

Um domínio recém-registado começa por ser comparado com um conjunto pré-gerado de *Typos* mais comuns. Caso haja uma correspondência, o domínio é imediatamente enviado para o Bot seguinte para ser analisado. No entanto, não sendo possível confirmar na primeira validação se o domínio é suspeito, este não é imediatamente descartado, passando por um conjunto de funções que visam encontrar um índice de similaridade entre o domínio novo e o domínio a comparar. Caso o resultado da análise seja positivo (i.e. domínio suspeito) é enviado para análise para o Bot seguinte. Caso contrário as suspeitas sobre o domínio são desconsideradas. As diversas funções de comparação são explicadas de seguida:

- **Reverse IDNA** - O caractere cirílico «ĭ», pode ser considerado um homoglifo de «i» por ser visualmente semelhante ao mesmo. Da mesma forma que os caracteres «ë» e «e» são semelhantes. Esta comparação visual estende-se a centenas de caracteres suportados no *charset* Unicode. Sendo caracteres diferentes durante a análise realizada o resultado iria ser negativo, mesmo tendo em conta o seu aspeto visual. Tomando o exemplo dos dois caracteres cirílicos o exemplo seguinte irá ser utilizado para demonstração da função. **IDNA**:*xn–isc-kma.pt* - **Unicode**: *ĭsëc.pt*.

A solução para este problema foi agrupar um conjunto de homoglifos para cada caractere suportado no DNS. Assim que o domínio passa por esta função, caso o domínio seja um IDNA *xn–isc-kma.pt*, é convertido para Unicode *ĭsëc.pt*. De seguida, cada caractere do domínio em Unicode é processado individualmente e, caso corresponda a um homoglifo de um caractere do conjunto suportado, é feita uma permutação pelo mesmo (exemplo: ĭ -> i). No final é obtido um domínio (isec.pt) que irá ser comparado então com o legítimo.

- **1ª Label Contida no Domínio Suspeito** - Esta função extrai a primeira *label* do domínio (exemplo: isec) e verifica se esta está contida no domínio a comparar.

- **Levenshtein entre a 1ª label dos Domínios** - A Distância de Levenshtein[19] é uma métrica utilizada para avaliar a quantidade de operações necessárias (adicionar, trocar, remover um caractere) para transformar uma sequência de caracteres noutra. Nesta função é utilizado o rácio de Levenshtein que retorna um índice de similaridade entre as duas sequências, onde 1 significa que os objectos comparados são idênticos. Esta função vem então aumenta a capacidade de deteção de potenciais *typos* que não tenham sido previamente identificados na fase inicial deste Bot.

  Nota: Foi equacionado durante o desenvolvimento o algoritmo de Hamming. No entanto, este apenas suporta a comparação de conjuntos da mesma dimensão, tendo por tal motivo a escolha recaído em Levenshtein.

- **Nome da Entidade contido no Domínio** - Por vezes o domínio de uma entidade não contem o nome da mesma. Para suportar esta possibilidade, nesta função, é verificado se o mesmo se encontra presente no domínio a comparar. Por exemplo, o domínio *cgd.pt* é pertencente ao banco *Caixa Geral de Depósitos*. No entanto, o seu domínio é apenas um acrónimo e uma análise exclusiva ao mesmo não iria identificar variações criadas com base no seu nome.

- **Palavras Relacionadas contidas no Domínio** - Verifica se o domínio contém palavras relacionadas com a empresa como por exemplo produtos ou marcas.

  A tabela 5.2, apresenta a análise comparativa do Bot criado, com as soluções analisadas no estado da arte 2.3.

Transformação	Typo Finder[11]	DnsTwist [31]	UrlCrazy [32]	*Proposta Dognaedis*
Adição de um caráter	✓	✓	✓	✓
Adição de um subdomínio	✓	✓	✓	✓
Duplicação de caráteres	✓	✓	✓	✓
Supressão de caráteres	✓	✓	✓	✓
Troca de Carateres	✓	✓	✓	✓
Adição de um ponto extra	✓			✓
Remoção de um ponto			✓	✓
Homoglifo	✓	✓	✓	✓
Pluralização de palavras	✓	✓	✓	
Permutação TLD	✓	✓	✓	✓
BitSquatting	✓	✓	✓	✓
Adição de Hífens	✓	✓		✓
Erros ortográficos comuns			✓	✓
Homofonos			✓	✓
Punycode			✓	✓
Análise genérica por palavras chave				✓
Keyboard Typos	✓	✓		✓

Tabela 5.2: Tabela Comparativa 2 - Ferramentas de Squatting

5.1.4 Classificador de Domínios - DomainClassifier

Este Bot tem por objetivo classificar os domínios identificados pelo Bot anterior, através de técnicas de *Data Mining*. Para o efeito foi criado um *Dataset* de treino composto por duas classes: domínios Maliciosos e Não Maliciosos. Para compor a classe de domínios não maliciosos extraíram-se os 20.000 domínios/websites mais visitados do ranking Alexa [1]. Para os domínios maliciosos extraíram-se 30.000 registos do Portolan e que anteriormente já tinham sido reportados em atividades maliciosas. Ao conjunto de domínios mencionado adicionaram-se então atributos, como por exemplo o TLD, *Registrar* ou Geo-Localização do A Record. A lista final de atributos utilizados na construção deste conjunto foi a seguinte:

- tld - *Top Level Domain*;
- domain_protected - O registo do domínio está anonimizado? (Verdadeiro/Falso);
- mx_record - Tem servidor de e-mail associado? (Verdadeiro/Falso);
- max_consecutive_digits - Número máximo de dígitos consecutivos no domínio;
- domain_length - Tamanho do domínio;

- n_letters - Total de letras no domínio;
- entropy - Entropia de Shannon. Medida utilizada para medir a «aleatoriedade» do domínio. Este atributo é bastante relevante para deteção de domínios associados a malware. [29];
- max_consecutive_vowels - Número máximo de vogais consecutivas no domínio;
- n_vowels - Total de vogais;
- n_numbers - Total de dígitos;
- max_consecutive_nonvowels - Número máximo de consoantes consecutivas no domínio;
- n_labels - Número de *labels* no domínio;
- has_hifen - O domínio tem hífen? (Verdadeiro/Falso);
- ip_geo_cc - Geo Localização do A Record (caso exista);
- asn - *AS Number* associado ao A Record (caso exista);
- registrant - Pessoa ou Organização que fez o registo;
- registrant_email_domain - Domínio do email da Pessoa ou Organização que fez o registo;
- registry_duration - Duração do Registo (em dias);
- registrant_country - País da Pessoa ou Organização que fez o registo;
- registrar - Entidade utilizada para o registo do domínio;
- malicious - Classificação do Domínio. Malicioso? (Verdadeiro/Falso).

Após o tratamento de dados (remoção de entradas nulas, correlações entre dois atributos, etc.) o *Dataset* final ficou composto por 35819 domínios, cuja dispersão é mostrada através da tabela 5.3.

Maliciosos	**Não Maliciosos**
20346	15473

Tabela 5.3: Dispersão do *Dataset* de treino por classe.

Para a análise de dados foi utilizado o Weka [34, p. 1], um projeto *open source* sob licença *GNU General Public License* (GPLv3) da Universidade de Waikato na Nova Zelândia e que reúne uma coleção de algoritmos de *Data Mining*. Neste *software* foram testados os algoritmos de classificação JRIP (Classificador por regras), RandomForest e NaiveBayes. Os testes basearam-se na execução desses mesmos algoritmos, sobre o *Dataset* criado. Os resultados dos testes realizados podem ser verificados na tabela 5.4.

Medida	JRIP	NaiveBayes	RandomForest
Precisão	87.90%	86.57%	89.69%
Especificidade	90.02%	74.3%	89.76%

Tabela 5.4: Resultado da avaliação dos algortimos de classificação - Precisão e Especifidade

Após a análise de resultados o algoritmo que demonstrou melhores resultados com os dados de treino foi o JRIP. A figura 5.3 e a tabela 5.5 permitem ajudar a perceber melhor o motivo desta decisão.

TP	Elemento dos itens relevantes, corretamente classificado como relevante.
FP	Elemento dos itens irrelevantes, incorretamente classificado como relevante.
FN	Elemento dos itens relevantes, incorretamente classificado como irrelevante.
TN	Elemento dos itens irrelevantes, corretamente classificado como irrelevante.

Tabela 5.5: Possíveis resultados - Classificação Binária

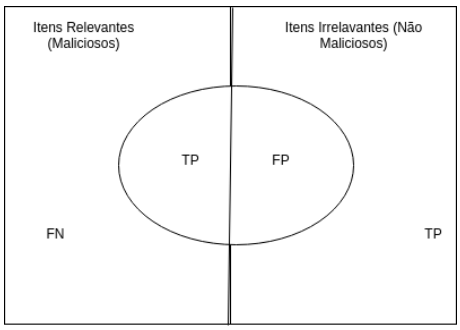


Figura 5.3: *Precisão e Especifidade*

Tendo em conta os elementos anteriores, importa então considerar duas medidas:

- Precisão é uma medida que permite avaliar a correta classificação dos elementos analisados. (*True Positive* (TP)/TP+*False Positive* (FP))
- Especificidade é a medida que avalia o número total de elementos relevantes (domínios maliciosos) que são identificados do total da sua amostra. (TP/TP+*False Negative* (FN))

Como o *dataset* não era balanceado (distribuição uniforme de classes no *Dataset*), o algoritmo JRIP na eventualidade de nenhuma regra ter tido correspondência durante a análise e por razões estatísticas, a regra final irá indicar que o domínio é malicioso. Como neste contexto é preferível ter mais Falsos Positivos e menos Falsos Negativos, ou seja uma especificidade mais próxima de 1, este método de funcionamento satisfaz esse requisito. O motivo pelo qual é preferível ter

menos Falsos Negativos, é o potencial impacto que uma incorreta classificação de um domínio como benígno poderá ter.

Contudo, mesmo que determinado domínio não seja considerado malicioso durante esta primeira análise, é ainda feita uma análise posterior pelo Bot seguinte, uma vez que mais indicadores poderão confirmar a suspeição sobre o mesmo. A Figura 5.4 descreve o método de funcionamento deste algoritmo.

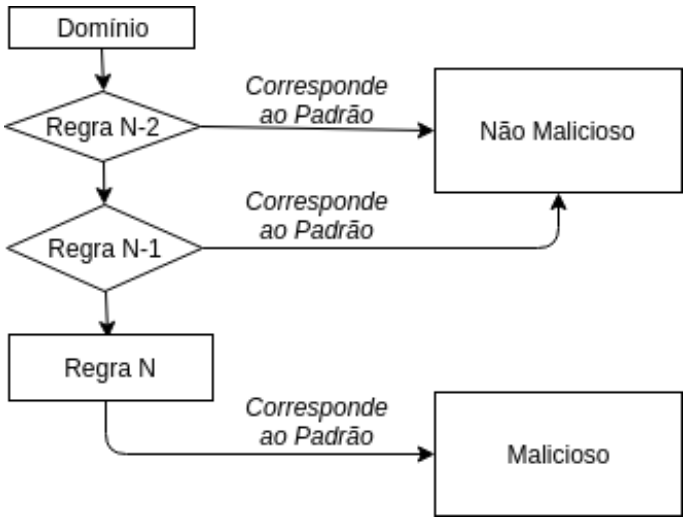


Figura 5.4: Análise do Bot DomainClassifier

5.1.5 Classificador HTTP - HTTPClassifier

O último elemento da Pipeline é o Bot HTTPClassifier. Este é responsável numa primeira instância por analisar o conteúdo HTTP, caso exista, associado ao A record do domínio. Por fim, este guarda o resultado da análise na base de dados relacional, através de uma *Application Programming Interface* (API) criada para o efeito. A arquitetura deste Bot está ilustrada na figura 5.5:

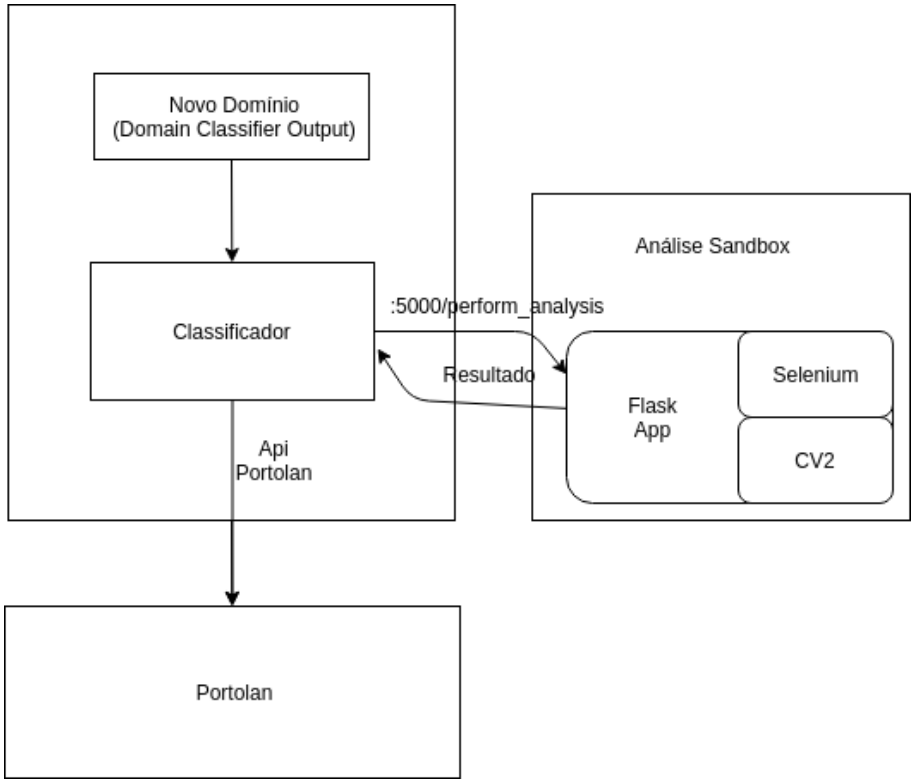


Figura 5.5: Bot HTTPClassifier

Quando chegado a esta fase, o domínio já passou por um conjunto de validações realizadas pelos Bots anteriores. Este Bot tem por objectivo confirmar, no caso de existência de conteúdo HTTP, se o domínio suspeito já contém alguma *Landing Page* que possa causar algum tipo de confusão a potenciais utilizadores, através de usurpação de entidade. Para tal é feita uma análise desse conteúdo. Tendo em conta que o conteúdo da página poderá ser malicioso, de forma a prevenir potenciais infeções do sistema e ambiente envolvente, a referida análise é realizada num ambiente isolado, aqui denominado de *SandBox*.

A *Sanbox* contém um servidor HTTP que serve uma aplicação desenvolvida em Flask (uma *framework* em Python). Esta, após receber o domínio suspeito, realiza a sua análise em diversas fases. Numa primeira fase é realizada uma comparação de imagens do domínio original, com o domínio suspeito. Para tal é usada a biblioteca Selenium, que permite emular a página através de um *Browser* e obter uma captura de ecrã da página já renderizada pelo mesmo. O objetivo desta comparação é tentar identificar semelhanças entre o site original e o domínio que foi considerado semelhante e potencialmente malicioso. Para a comparação de imagens foi utilizada a biblioteca opencv-python, sendo necessário numa primeira fase realizar um tratamento inicial, nomeadamente a conversão das imagens para escala de cinza e o redimensionamento das imagens em uso. A primeira transformação é realizada nativamente pelo algoritmo utilizado e é motivada pela redução da dimensão e dos dos modelos de comparação (em vez de RGB, as imagens escala de cinza contêm apenas uma). Já a segunda transformação é realizada de forma a que a imagem do domínio original tenha uma resolução <= à do domínio suspeito.

Caso a validação anterior não confirme a semelhança no conteúdo da página, é analisado o seu HTML, realizando uma pesquisa por palavras-chave como, por exemplo, o nome da entidade e/ou produtos que possam estar associados ao mesmo. Foi testada a abordagem de analisar a similaridade do HTML mas verificou-se que os resultados eram pouco fiáveis e precisos. Verificou-se, por exemplo, que páginas visualmente idênticas podiam ter um HTML substancialmente diferente, utilizando elementos diferentes (div, table, etc) o que levaria a que os índices de similidaridade apurados fossem baixos.

Após a análise concluída o resultado é retornado ao Bot HTTPClassifier, sendo a classificação do domínio atualizada e inserida na UI via API Rest.

5.2 Domain Squatting - Interface de Utilizador (UI)

A *User Interface* (UI) permite realizar uma análise aos resultados obtidos, bem como alterar as configurações que serão utilizadas pelos diversos bots durante as fases de deteção e classificação dos domínios.

5.2.1 Visualização e Análise de Resultados

Após um domínio ser analisado este é guardado diretamente na base de dados relacional do Portolan via API. Posteriormente, é possível a um analista de cibersegurança consultar o seu resultado e os detalhes do mesmo. A vista principal está dividida em duas áreas. Na primeira delas é apresentada uma tabela com a listagem de todos os domínios classificados. Na segunda área anunciam-se os domínios analisados e cuja análise do Bot HTTPClassifier foi positiva (i.e. identificada uma similaridade). A Figura 5.7 ilustra a referida visualização.

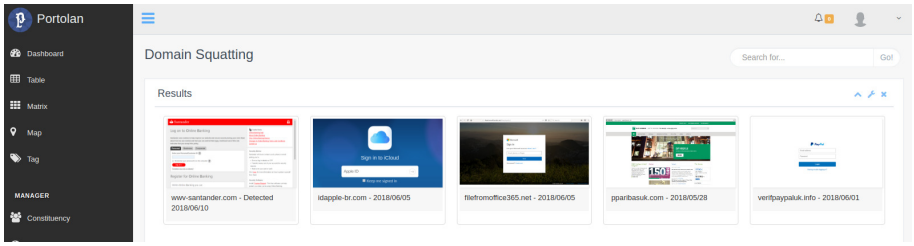


Figura 5.6: User-Interface - Dashboard de visualização de resultados

Acedendo ao resultado da análise é possível confirmar os detalhes do domínio como por exemplo o *Registrar*, nome do domínio ou data do registo. O analista pode ainda verificar os resultados da sua análise, bem como marcar o resultado como Falso Positivo, caso esta classificação tenha resultado de uma classificação incorreta. Esta funcionalidade satisfaz o requisito FR06 e irá remover o domínio da lista de domínios suspeitos.

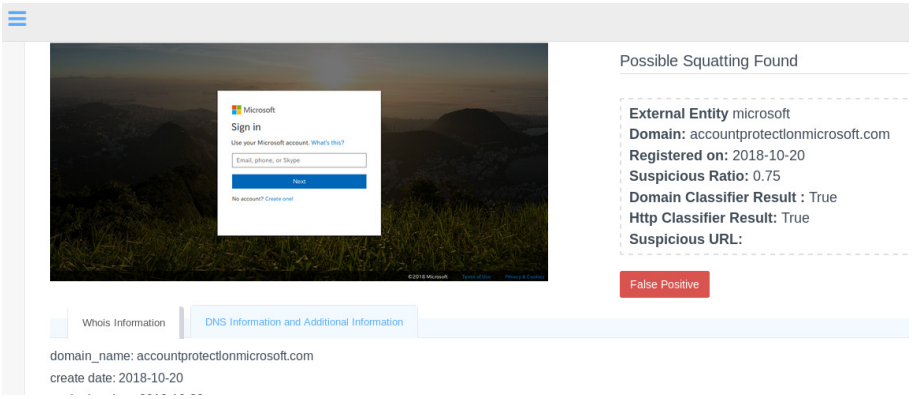


Figura 5.7: User-Interface - Detalhe do Resultado

5.2.2 Administração

Como requisito funcional, foi solicitada a possibilidade de poder fazer a edição das entidades cujos registos de DNS são monitorizados (requisitos FR07 e FR08). Esta funcionalidade, ilustrada na figura 5.8, é permitida através da área de gestão criada na aplicação de *Domain Squatting*. Para que fosse possível à Dognaedis não fornecer apenas este serviços aos seus clientes mas também ser capaz de detetar campanhas mais genéricas que habitualmente utilizam entidades de grande dimensão, a lista de entidades a monitorizar é dividida em dois grupos: Entidades Internas e Externas. Em ambos a configuração é idêntica sendo possível a adição/remoção de novos domínios, entidades, palavras-chave e/ou imagens a ter em conta durante a análise HTTP.

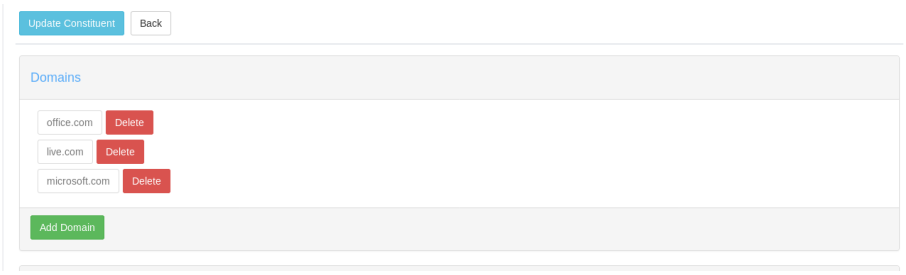


Figura 5.8: User-Interface - Dashboard de Administração | Detalhe da entidade

5.2.3 API de Comunicação com DNS Firewall

Através do requisito (FR11), verificou-se a necessidade de implementação de uma solução que permitisse a partilha dos resultados obtidos. A resposta mais coerente, de acordo com a arquitetura do Portolan, passou pela implementação de uma API que exportasse os resultados obtidos. De forma a ser flexível, esta permite que sejam definidos os critérios de extracção. São eles:

- *Hours* - Resultados Identificados desde as últimas N horas. (0 < Hours <= 24).
- *Constituent* - Resultados apenas desta entidade. Depende dos privilégios da API_KEY.

- *malicious_ratio* - Retornar apenas os resultados com rácio superior ou igual ao definido. Este rácio é atualizado pelos diversos Bots durante do processo de classificação do domínio. (0 <= ratio <= 1).

```
POST /domain_squatting/export_results HTTP/1.1
{
  "API_KEY": <char_str>,
  "constituent": <char_str>,
  "malicious_ratio": <float>,
  "hours": <int>,
}
```

Listagem 5.1: Estrutura de um pedido de exportação de resultados

A resposta ao pedido, em caso de sucesso, retorna um *Status Code* 200, com os resultados presentes num array em formato JSON no corpo da resposta. Exemplo da resposta:

```
[
{
  "domain_info": {
    "domain":<str>,
    "registrar":<str>,
    "create_date":<YYYY-MM-DD>,
    "expiration_date":<YYYY-MM-DD>,
    "registrar_company": <str>,
    "registrant_email": <str>,
    "registrant_country": <str>,
    "administrative_email": <str>,
    "administrative_country": <str>,
    }
  "date_of_identification": <YYYY-MM-DD>
  "malicious_ratio": <float>
}
]
```

Listagem 5.2: Estrutura de uma resposta de exportação de resultados

Os seguintes *HTTP Status Code* podem ser retornados ao comunicar com a API:

- **200** - Pedido realizado com sucesso.
- **400** - Pedido incorrecto.
- **401** - API_KEY inválida ou sem permissão para aceder ao recurso solicitado.

5.3 Solução Proposta - DNS Firewall

Conforme referido na secção 2.1.1, quando uma aplicação inicia uma comunicação IP com base no nome de um domínio, a resolução do mesmo pode ou não ser localmente conhecida. Caso esse dispositivo ainda não conheça o IP a que se deve ligar, ou seja o resultado da resolução não esteja contido na sua cache local, é realizado um pedido a um servidor responsável por lhe devolver o resultado. É neste ponto que entra a segunda parte deste projeto: a proteção. As resoluções de DNS passam a ser realizadas por um servidor específico, aqui denominado de DNS Firewall. Caso o domínio solicitado seja considerado malicioso, será devolvida uma resposta NXDOMAIN (*Non-Existent Domain*), evitando assim que seja estabelecida uma ligação ao mesmo, tal como ilustrado pela Figura 5.9.

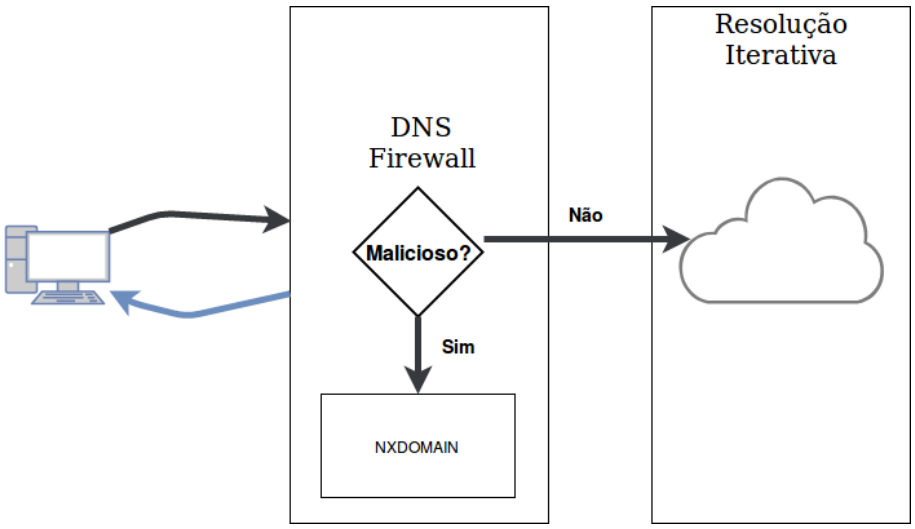


Figura 5.9: Servidor DNS Firewall - Modelo genérico de funcionamento

Para a implementação da DNS Firewall foi utilizado o servidor de DNS Unbound[22]. Este servidor de DNS é extensível e modular, suportando protocolos como *Domain over TLS* RFC 7858 [35], DNSSec e Query Name Minimization [5]. Para além do Unbound foi também considerado o servidor BIND, para a implementação da DNS Firewall. No entanto, devido à necessidade de *restarts* periódicos para a atualização das variadas listas e a complexidade acrescida na configuração da mesmas, fizeram com que a escolha tenha recaído no Unbound. O módulo de firewall criado assenta numa configuração de um script externo responsável pela resolução. Este, ao receber a query DNS, verifica se se trata de um domínio não malicioso (i.e., presente na *Whitelist* mantida pelo módulo de DNS *squatting* do Portolan). A primeira razão para a implementação desta funcionalidade foi prevenir o impacto de possíveis falsos positivos, causados por uma incorreta classificação de domínios. Estes falsos positivos que, adicionados à *Blacklist*, iriam impossibilitar o acesso aos recursos pretendidos. A outra razão foi a otimização dos tempos de resposta de domínios considerados confiáveis, uma vez que esta validação ocorre numa primeira fase, evitando assim percorrer as diversas *Blacklist*s. Após a consulta da *Whitelist*, são consultadas as diversas *Blacklists*. Estas foram divididas em diversas categorias (taxonomias), de forma a

poderem ser atríbuidas ações distintas, bem como o registo da categoria de atividade maliciosa detetada. Caso o pedido realizado seja permitido (i.e., não estar contido em *blacklists*), a DNS Firewall inicia então o processo de resolução iterativa.

As listas anteriormente são atualizadas periodicamente através de um acesso à API (5.2.3), estando assentes numa base de dados partilhada Redis[25] e contêm a seguinte estrutura:

- DB 1 - *Whitelist*
  - **KEY 1** - dominiowhitelisted1.com
  - **KEY N** - ...
- DB 2 - Blacklist - *Malware*
  - **KEY 1** - dominiomalware1.com
  - **KEY N** - ...
- DB 3 - Blacklist - *Squatting*
  - **KEY 1** - dominiosquatting1.com
  - **KEY N** - ...
- DB 4 - *Content Filtering*
  - **KEY 1** - dominio1.com
  - **KEY N** - ...

O motivo da utilização do Redis está relacionado com o requisito Q06, que comprendia a necessidade de poder escalar horizontalmente a solução, isto é, aumentar o número de DNS Servers disponíveis. Outra vantagem desta abordagem é a não necessidade de *restart* do servidor durante updates de listas, uma vez que estas estão sempre disponíveis nesta Base de Dados. Esta base de dados consegue suportar até 250 milhões de *Keys*, número considerado suficiente para o fim proposto. O motivo pelo qual se optou por este esquema de base de dados, foi a necessidade da redução da latência, induzida pelas constantes iterações feitas fora do motor da base de dados.
Outro fator considerado relevante para a decisão, foi a possibilidade do Redis suportar nativamente pesquisas com máscaras nas chaves (*Keys*). Isto vem permitir o uso por exemplo de *wildcards*, que permitem bloquear toda a àrvore de um determinado domínio mesmo que na *Blacklist* esteja contido apenas um registo A. Imagine-se por exemplo o domínio *www.dominiomalicioso.com* adicionado à *Blacklist* de *Malware*. Mais tarde o mesmo ator malicioso criou uma nova campanha mas com o domínio *www2.dominiomalicioso.com*. Caso ocorresse uma resolução à DNS Firewall deste último domínio, esta não iria ser bloqueada. Com o uso de *wildcards* é possível fazer uma pesquisa na base de dados por **.dominiomalicioso.com*, permitindo conter quaisquer campanhas que possam surgir relacionadas com o primeiro indicador.
Apesar da Base de Dados escolhida garantir um bom desempenho, *queries* constantes à mesma

a cada resolução de DNS iriam adicionar alguma latência aos pedidos. Assim foi proposta uma solução complementar que vem minimizar este impacto. Cada servidor de DNS tem configurada uma cache em memória de resoluções realizadas recentemente referentes às listas anteriores. A Figura 5.10 ilustra as validações realizadas em cada pedido de DNS e a sua sequência.

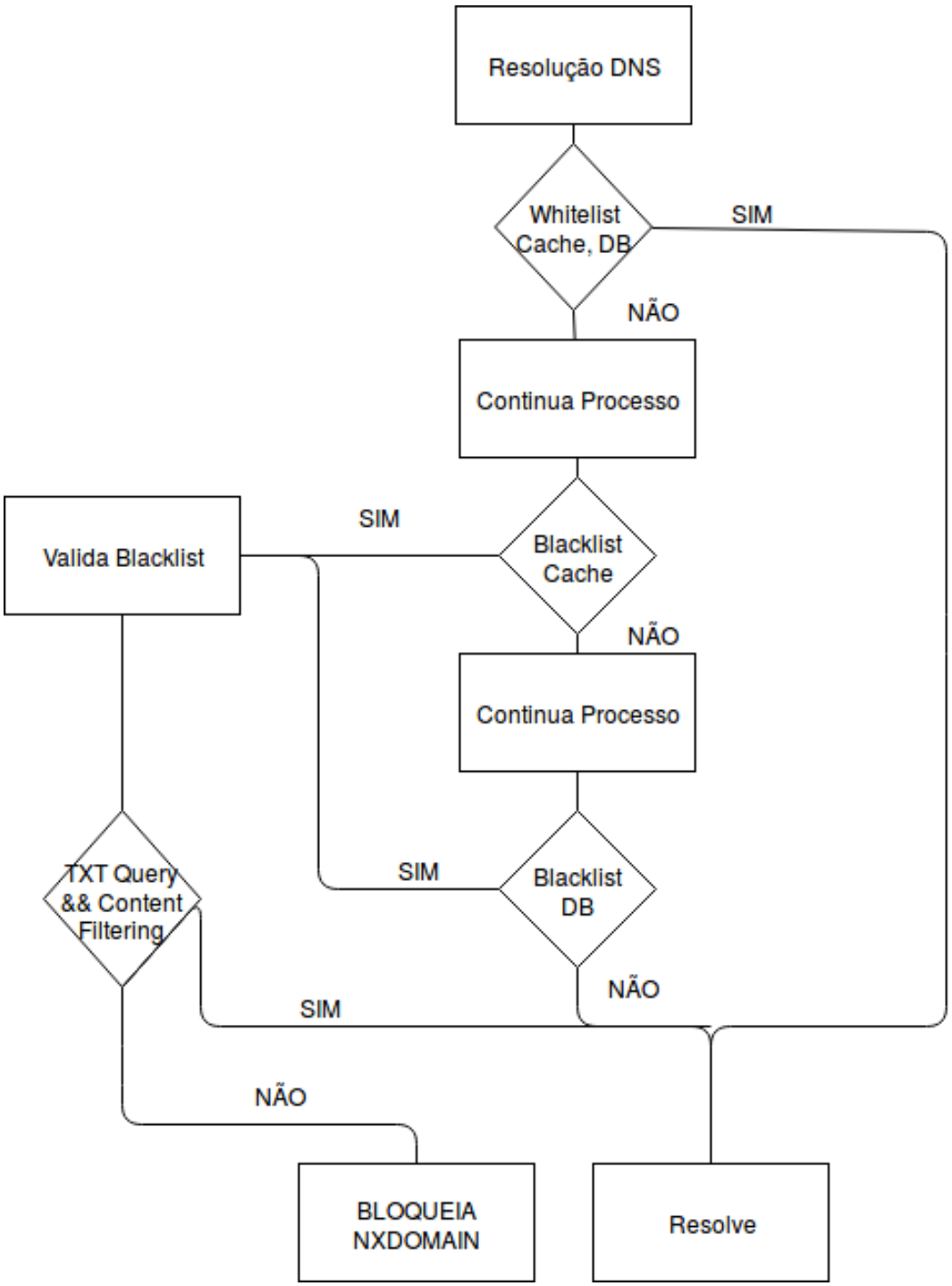


Figura 5.10: Servidor DNS Firewall - Fluxo de resolução DNS

Através da figura anterior 5.10 é possível verificar que existe uma exceção nas *blacklists* para o bloqueio das resoluções DNS. Esta ocorre quando é feito um pedido TXT de um domínio apenas bloqueado por política/*Content Filtering*. A justificação para esta decisão é a relevância

da informação contida nos registos TXT como o SPF, informação essa utilizada para validar a origem de um e-mail. Imaginemos o cenário, onde num ambiente corporativo, não é autorizado o acesso ao e-mail pessoal (Exemplo: hotmail.com). Com este método de funcionamento é possível bloquear o acesso a *www.hotmail.com*, mas nesse mesmo ambiente continuar a poder receber e-mails com essa origem. Tal a acontece pela validação realizada pelo servidor e-mail, durante o processo de receção de novas messagens. Este ao receber um e-mail, realiza uma validação quanto à origem do mesmo. A informação a utilizar durante este processo está contida no *SPF* do próprio domínio. Caso todos os registos do domínio *hotmail.com* estivessem bloqueados, o servidor seria incapaz de validar o e-mail recebido o que, dependendo da política *anti-spam* aplicada, poderia impedir a receção de mensagens.

A implementação deste componente do projeto ficou assente num servidor virtualizado Ubuntu 18.04 LTS, com 1 Virtual CPU, 1GB Ram de 20GB Disco SSD.

6

Validação da Solução

Este capítulo tem por objetivo detalhar os testes realizados sobre o trabalho desenvolvido, de forma a garantir que cumpre o requisitos definidos.

6.1 Testes de Software

Durante todo o desenvolvimento foram realizadas validações manuais sobre os resultados que vinham a ser obtidos. Estas validações permitiam delinear as estratégias seguintes do desenvolvimento. Contudo, foram utilizadas outras metodologias que permitem garantir a integridade e a conformidade do produto desenvolvido, sendo estas baseadas em testes unitários de software e testes de segurança.

No core do Portolan, todos os elementos criados no decorrer deste projeto foram alvo de testes unitários de software. Estes foram concebidos para validar fluxos dos diversos bots, campos de entrada, resultados de operações lógicas e aritméticas e implementados através da biblioteca de python unittest. Assim, sempre que uma nova funcionalidade ou alteração no código seja implementada, os bots têm de passar nos testes já pré-concebidos. Caso falhem significa que o resultado que está a ser obtido não é o esperado e dessa forma a *build* falha, não estando esta versão pronta para passar à próxima fase do ciclo de desenvolvimento de software.

De forma a cumprir os requisitos de segurança, na UI o código desenvolvido foi alvo de testes de segurança. Inicialmente esses testes foram feitos de forma manual para garantir a cobertura de vulnerabilidades lógicas e de subversão de fluxos. Numa fase seguinte foram realizados

testes com recurso a ferramentas automatizadas por forma a garantir uma maior cobertura de testes e eficiência. Para o efeito foram usadas várias ferramentas como Burp, SqlMap, Owasp Zap e Nikto, não tendo sido identificadas vulnerabilidades que fossem contra o compromisso de segurança da aplicação.

6.2 Domain Squatting - Resultados

Durante o desenvolvimento desta solução foram continuamente realizados testes, não só para avaliar a eficiência dos algoritmos realizados, mas também para produzir alguma inteligência que pudesse ser utilizada na operação da empresa. Existiram bastantes falsos positivos no início, nomeadamente na classificação HTTP e na identificação dos domínios *squatting*. Após a reformulação dos algoritmos, os resultados passaram a ser bastante mais confiáveis. A Figura 6.1 representa a distribuição de domínios corretamente identificados, isto é que apresentam uma semelhança considerável ou que aparentam deliberadamente fazer-se passar por determinada entidade. Para este estudo foram identificados e analisados 1143 domínios pela plataforma criada.

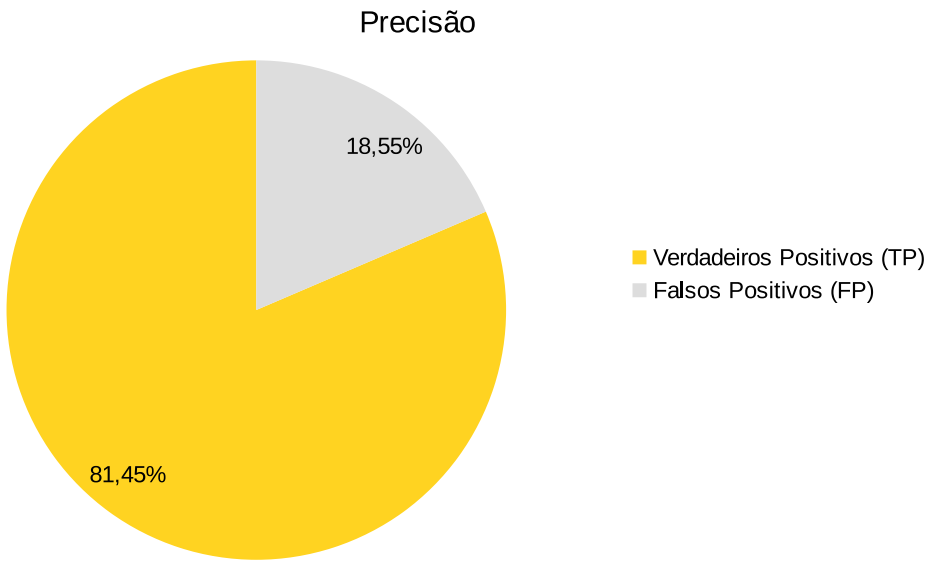


Figura 6.1: Precisão da identificação de domínios de Squatting

Como verificado através da figura anterior, a precisão foi de 81.45%, o que pode ser considerado um bom resultado. O valor de Falsos Positivos na identificação pode ser justificado com a utilização de domínios com 1ª *label* de curta dimensão (<5 carateres) e entidades/palavras-chave com nomes genéricos e que facilmente podem ter outro significado noutro contexto (ex: Apple, Windows, Office). Vejamos o exemplo anterior «office», apesar de poder retornar falsos positivos associados, por exemplo, ao ramo imobiliário mas que no entanto permite identificar ameaças reais tais como:

- *edp-office365[.]com*

- *mssecurityoffice365[.]com*
- *ww-woffice[.]com*

Todos os resultados anteriores, identificados pela plataforma, não iriam ser identificados se apenas fosse utilizada a entidade microsoft ou domínio microsoft[.]com para monitorização. Apesar deste número de falsos positivos, é possível reduzir a utilização dos mesmos ao definir o critério de confiança (malicious_ratio) a utilizar na exportação dos resultados obtidos. A Figura 6.2, demonstra que cerca de 79,1% dos domínios considerados suspeitos têm um índice >=0,65, enquanto que apenas 69,7% do total de falsos positivos encontra-se presente neste intervalo. Contudo, ao analizar-se o intervalo >= 0.6 verifica-se que praticamente todos os domínios identificados (94,7%) são selecionados (+15.6% em relação ao anterior). Em contrapartida, também aumenta bastante o número de falsos positivos neste intervalo, tendo um crescimento ainda superior, 96% (+26,3%).

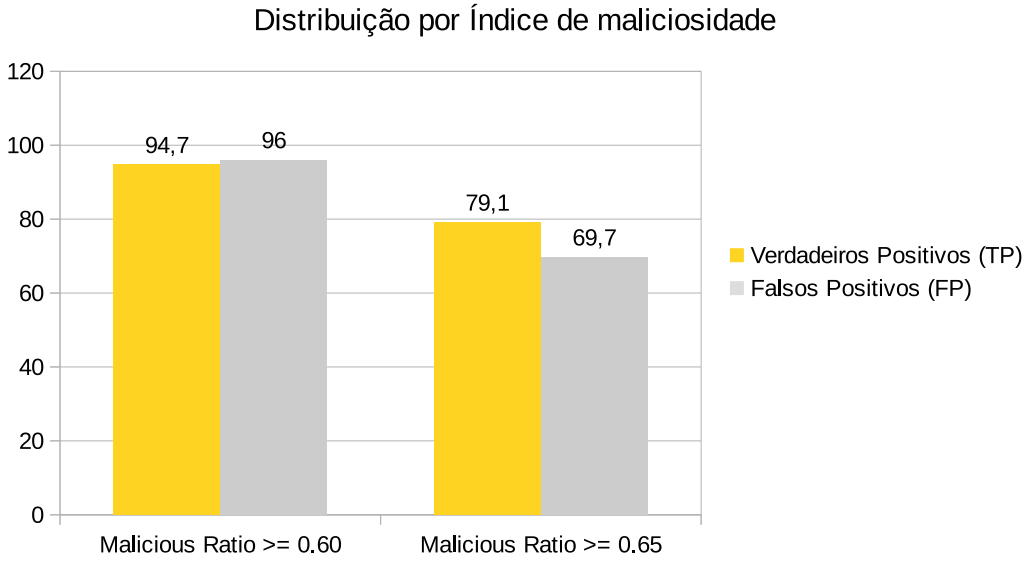


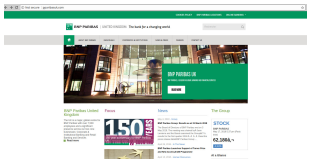
Figura 6.2: Distribuição dos resultados

Os resultados anteriores demonstram que existiram sempre falsos positivos e que tráfego não malicioso poderá ser bloqueado indevidamente. Contudo, se se abordar uma política de confiança zero [2] numa organização, esta métrica é de relevância menor. Importa voltar a referir que esta plataforma tem a possibilidade de retificar os domínios incorretamente classificados pela plataforma em duas fases distantas. A primeira via UI marcando o resultado como Falso Positivo e a segunda através da *whitelist* na Firewall DNS, que tem prevalência sobre as *blacklists*.

As imagens seguintes são exemplificativas de alguns dos resultados identificados e que no momento da sua análise já continham *landing pages* que tentavam simular as páginas reais.

(a) Domínio - *cd-gmail[.]com*

(b) Domínio - *appleid-i[.]cloud*



(c) Domínio - *pparibasuk[.]com*

(d) Domínio - *jpmorogan[.]com*

Figura 6.3: Exemplos Identificados

Durante o desenvolvimento deste projeto foram ainda identificados domínios considerados suspeitos, relacionados com clientes da Dognaedis. Tais domínios foram reportados às entidades visadas, para que pudessem ser aplicadas as medidas de mitigação necessárias.

6.3 DNS Firewall - Testes de Performance

Durante a implementação da DNS firewall, foi sempre uma prioridade dar resposta ao requisito de qualidade Q02 - Baixa latência nas resoluções DNS. Para tanto foi utilizada a ferramenta DNSPerf[23] em vários cenários. Os cenários testados englobaram o uso de *blacklists* com e sem a resolução em cache, para um total de 500 domínios considerados maliciosos, bem como a resolução iterativa de 3000 domínios presentes no top 10000 do Alexa. A tabela 6.1 apresenta os resultados obtidos de resoluções por segundo e latência nos cenários testados.

Testes Realizados	**Resoluções por segundo**	**Latência** em milisegundos (média)
Blacklist sem cache	341	50
Blacklist em cache	1455	2
Resolução iterativa sem cache, sem *blacklist*	407	200
Resolução iterativa em cache, sem *blacklist*	2331	30

Tabela 6.1: *DNS Perf* - Testes de Performance

Quanto ao número de resoluções por segundo, como seria de esperar, os resultados melhoraram consideravelmente (4-5 vezes) após o domínio estar em cache. Apesar desta melhoria, os primeiros resultados já eram bastante satisfatórios dado que no primeiro cenário (sem cache), caso cada domínio fosse resolvido uma única vez, seria possível responder a 1227600 resoluções por

hora. Já relativamente ao segundo indicador medido, a latência, como seria de esperar os resultados de blacklists apresentam um valor mais baixo, devido à não necessidade da realização da resolução iterativa. Importa ainda referir que os valores apresentados são apenas a média, e que incluem respostas como *SERVFAIL* ou *Refused*, que por sua vez aumentam os tempos médios. Os valores mínimos para cada um dos cenários acima referidos, foram respetivamente 1.9, 1.6, 4 e 3 ms.

6.4 Validação de Requisitos

Para o projeto poder ser considerado satisfatório os requisitos acordados e definidos no capítulo 3 deveriam ser cumpridos. A tabela 6.2 vem constatar os resultados obtidos.

FR01 - Colector de domínios recém-criados	✓
FR02 - Identificação de domínios considerados suspeitos	✓
FR03 - Implementação de um sistema classificador de domínios	✓
FR04 - Análise automatizada de Landing Pages	✓
FR05 - Consulta dos resultados da Análise	✓
FR06 - Possibilidade de marcar um resultado como Falso Positivo	✓
FR07 - Adicionar/Editar informação de Entidades a Monitorizar	✓
FR08 - Remover Entidades a Monitorizar	✓
FR09 - Implementação do servidor de DNS recursivo	✓
FR10 - Bloqueio do acesso a domínios maliciosos	✓
FR11 - Alimentação automáticas das Blacklists no Servidor de DNS	✓
Q01 - Análise rápida aos novos domínios	✓
Q02 - Baixa latência nas resoluções DNS	✓
Q03 - Validação de campos de entrada na UI	✓
Q04 - Cuidados na análise do HTML de páginas suspeitas	✓
Q05 - Anonimização no acesso a Landing Pages	✓
Q06 - O DNS Resolver deverá poder escalar horizontalmente	✓

Tabela 6.2: Validação de Requisitos satisfeitos.

7

Conclusões

O *Domain Squatting* é uma técnica cada vez mais utilizada, através da qual atores maliciosos registam domínios de grafia ou fonética semelhante a domínios legítimos, para iludir internautas que, de antemão, se encontram predispostos a depositar confiança em nomes, marcas e produtos conhecidos. Do presente estágio resultaram duas ferramentas desenvolvidas pelo autor: a) Um novo módulo para a ferramenta de ciber-inteligência Portolan da Dognaedis, que lhe conferiu a funcionalidade de identificar ameaças de *Domain Squatting*; b) um servidor local DNS, denominado aqui de DNS Firewall, capaz de interagir com o Portolan e vigiar, em tempo real, acessos externos, vedando o acesso a domínios potencialmente maliciosos.

Concluído o projeto de estágio, pode-se afirmar que o balanço do trabalho realizado é bastante positivo, tendo sido possível cumprir todos os requisitos especificados. O trabalho desenvolvido foi desafiante, tanto devido à diversidade de tecnologias utilizadas, bem como deviao ao tema do mesmo. Por exemplo, a recolha da informação associada ao registo de domínios, habitualmente realizada através do protocolo Whois, foi alterada após as alterações impostas pelo novo Regulamento Geral de Proteção de Dados. Essas alterações vieram adicionar confidencialidade do *Registrant*, informação relevante que era utilizada no modelo inicial de classificação de domínios. Outro desafio foi a dimensão de possíveis permutações que um domínio poderia ter para ser considerado suspeito. Este factor, quando conjugado com o requisito da velocidade de análise deste módulo, fez com que esta identificação passasse de um modelo exclusivamente com um *dataset* pré-gerado e estático, para um modelo híbrido, com uma análise dinâmica através de algoritmos que permitem identificar a similaridade de palavras e um *dataset* de menor dimensão apenas com os *typos* mais comuns.

Para além do conhecimento adquirido, os resultados obtidos são também bastantes satisfatórios tanto na componente de identificação e classificação de domínios *squatting*, bem como nos testes de desempenho obtidos nas resoluções feitas pela DNS Firewall desenvolvida. Relativamente à primeira parte, de uma amostra retirada dos resultados obtidos, a precisão foi de 81,45%, valor considerado positivo. Já sobre os testes de desempenho realizados o resultado mais baixo, permite resolver «apenas» 1.2 milhões de domínios por hora, valor largamente superior à média de resoluções na entidade acolhedora durante as horas de tráfego mais congestionadas. A latência foi igualmente satisfatória, tendo sido possível obter valores bastante mais baixos que os valores obtidos em diversos servidores de DNS recursivos públicos e que podem também utilizar mecanismos de firewall.

Durante o desenvolvimento deste projeto, resultaram diversos novos Indicadores (domínios maliciosos), associados a marcas ou entidades conhecidas. Tais resultados permitiram validar a solução e divulgá-la numa conferência de de cibersegurança, C-DAYS 2018, organizada pelo Centro Nacional de Cibersegurança. A apresentação pode ser consultada no Youtube [6]. Contudo, não foram apenas identificados domínios associados a entidades externas. A plataforma desenvolvida classificou também resultados associados a clientes da Dognaedis, o que demonstra a valia desta no contexto da atividade da empresa. Apesar dos domínios identificados não conterem ainda *landing pages*, foram reportados e desta forma, permitiu que fosse aplicado o processo resposta a incidentes.

Como trabalho futuro, este projeto poderá ser extendido a outras fontes de informação que não apenas domínios recém-criados, tais como fontes de informação passivas de atividade na rede, permitindo assim criar inteligência (Indicadores de Compromisso), numa perspectiva mais forense e reativa. Foi verificado durante a investigação que muitas vezes os domínios criados não são utilizados imediatamente após o seu registro, havendo o risco do domínio ser removido das *blacklists* do Portolan. Desta forma, e caso seja possível integrar logs de outras fontes como *Proxies* HTTP ou *Network IDS*, caso exista um acesso a um domínio ainda não identificado, será possível realizar a mesma análise e aplicar o mesmo método de classificação. A sugestão anterior, é também extensível aos logs de DNS gerados pela DNS Firewall de resoluções feitas a domínios desconhecidos, permitindo assim uma abordagem circular em que Portolan alimenta a DNS Firewall mas, a DNS Firewall pode também fornecer inteligência ao Portolan para posterior análise.

Ainda como possível tarefa futura identifica-se a integração da DNS Firewall com Honeypots, alterando a resposta devolvida de NXDOMAIN para o enderço IP da Honeypot. Esta técnica viria a permitir poder extrair informação, capaz de produzir inteligência, que alimente outros sistemas de proteção na rede, como *Intrusion Prevention Systems* ou *Proxies*. Aproveitando a referência a *Proxies* HTTP, estes poderão também fazer parte de uma solução futura que utilize o Portolan como fonte de dados. A solução atual previne o acesso a domínios considerados suspeitos mas não permite, por exemplo, a aceitação do risco por parte do utilizador

e, consequentemente, o acesso a um domínio que possa ter sido incorretamente classificado. A implementação deste tipo de funcionalidade requer que seja possível guardar estado, algo que não é possível garantir exclusivamente através de um servidor de DNS. Por isso, o uso de proxies com recurso a identificadores de Sessão (*Cookies*), podem ser uma abordagem futura a considerar.

8

Bibliografia

[1] Alexa. *Alexa - Alexa Internet - About Us*. 2018. URL: https://www.alexa.com/about.

[2] Palo Alto. *What is a zero trus architecture*. 2018. URL: https://www.paloaltonetworks.com/cyberpedia/what-is-a-zero-trust-architecture.

[3] Undrah B Baasanjav. «Linguistic diversity on the internet: Arabic, Chinese and Cyrillic script top-level domain names». Em: *Telecommunications Policy* 38.11 (2014), pp. 961–969.

[4] Anirban Banerjee, Md Sazzadur Rahman e Michalis Faloutsos. «SUT: Quantifying and mitigating url typosquatting». Em: *Computer Networks* 55.13 (2011), pp. 3001–3014.

[5] Stephane Bortzmeyer. *DNS query name minimisation to improve privacy*. Rel. téc. 2016.

[6] Centro Nacional de Cibersegurança. *C-days Apresentação (Video)*. 2018. URL: https://www.youtube.com/watch?v=5qGX_b1fRGc.

[7] Django. *The Web Framework for perfectionists with deadlines*. 2005. URL: https://www.djangoproject.com/.

[8] DNSPerf. *DNSPerformance - Compare the speed and uptime of enterprise and commercial DNS services*. 2019. URL: https://www.dnsperf.com/ (acedido em 01/05/2019).

[9] DNS.PT. *DNSSEC*. URL: https://www.dns.pt/pt/dnssec/ambito/.

[10] Flask. *Flask (A Python MicroFramework)*. 2010. URL: http://flask.pocoo.org/.

[11] NCC Group. *TypoFinder*. 2016. URL: https://github.com/nccgroup/typofinder (acedido em 01/03/2018).

[12] IANA. *IANA - Root Servers*. URL: https://www.iana.org/domains/root/servers.

[13] ICANN. *Domain Name Registrations - ICANN*. 2016. URL: https://www.icann.org/cct-metrics-domain-name-registration-2016-06-27-en.

[14] ICANN. *New gTLD Program*. 2018. URL: https://icannwiki.org/New_gTLD_Program.

[15] Intel. *OpenCV*. 2000. URL: https://opencv.org/.

[16] Mohammad Taha Khan et al. «Every second counts: Quantifying the negative externalities of cybercrime via typosquatting». Em: (2015), pp. 135–150.

[17] Adam Langley, Emilia Kasper e Ben Laurie. *Certificate Transparency*. 2013. URL: https://tools.ietf.org/html/rfc6962.

[18] Adam Langley, Emilia Kasper e Ben Laurie. *Certificate Transparency - Presentation*. 2013. URL: https://csrc.nist.gov/csrc/media/events/workshop-on-improving-trust-in-the-online-marketpl/documents/presentations/kasper_ca-workshop2013.pdf.

[19] Levenshtein. *Distância de Levenshtein*. URL: https://en.wikipedia.org/wiki/Levenshtein_distance.

[20] John Munro. *Clairvoyant Squirrel: Large Scale Malicious Domain Classification*. 2013. URL: https://www.slideshare.net/jasontrost/flo-con-clairvoyant-squirrel-final.

[21] Nick Nikiforakis et al. «Soundsquatting: Uncovering the use of homophones in domain squatting». Em: (2014), pp. 291–308.

[22] Nlnetlabs. *NLnet Labs - Unbound*. 2006. URL: https://nlnetlabs.nl/projects/unbound/about/.

[23] Inc. Nominum. *DNSPerf - test performance of DNS Server*. 2019. URL: https://linux.die.net/man/1/dnsperf.

[24] PostgreSQL. *PostgreSQL: The world's most advanced open source database*. 1989. URL: https://www.postgres.org/.

[25] Redis. *Redis*. 2009. URL: https://redis.io/.

[26] Guido van Rossum. *Welcome to Python.org*. 1991. URL: https://www.python.org/.

[27] Selenium. *Selenium - Web Browser Automation*. URL: https://www.seleniumhq.org/.

[28] Jeffrey Spaulding, Shambhu Upadhyaya e Aziz Mohaisen. «The landscape of domain name typosquatting: Techniques and countermeasures». Em: (2016), pp. 284–289.

[29] Splunk. *Random Words on Entropy and DNS*. 2015. URL: https://www.splunk.com/blog/2015/10/01/random-words-on-entropy-and-dns.html.

[30] Janos Szurdi et al. «The long "taile" of typosquatting domain names». Em: *23rd {USENIX} Security Symposium ({USENIX} Security 14)*. 2014, pp. 191–206.

[31] Marcin Ulikowski. *DnsTwist*. 2015. URL: https://github.com/elceef/dnstwist (acedido em 02/03/2018).

[32] *UrlCrazY*. 2013. URL: https://github.com/hardwaterhacker/URLCrazy (acedido em 04/03/2018).

[33] US-CERT. *DNS Amplification Attacks*. 2013. URL: https://www.us-cert.gov/ncas/alerts/TA13-088A.

[34] Weka. *Weka*. 2018. URL: https://www.cs.waikato.ac.nz/ml/weka/.

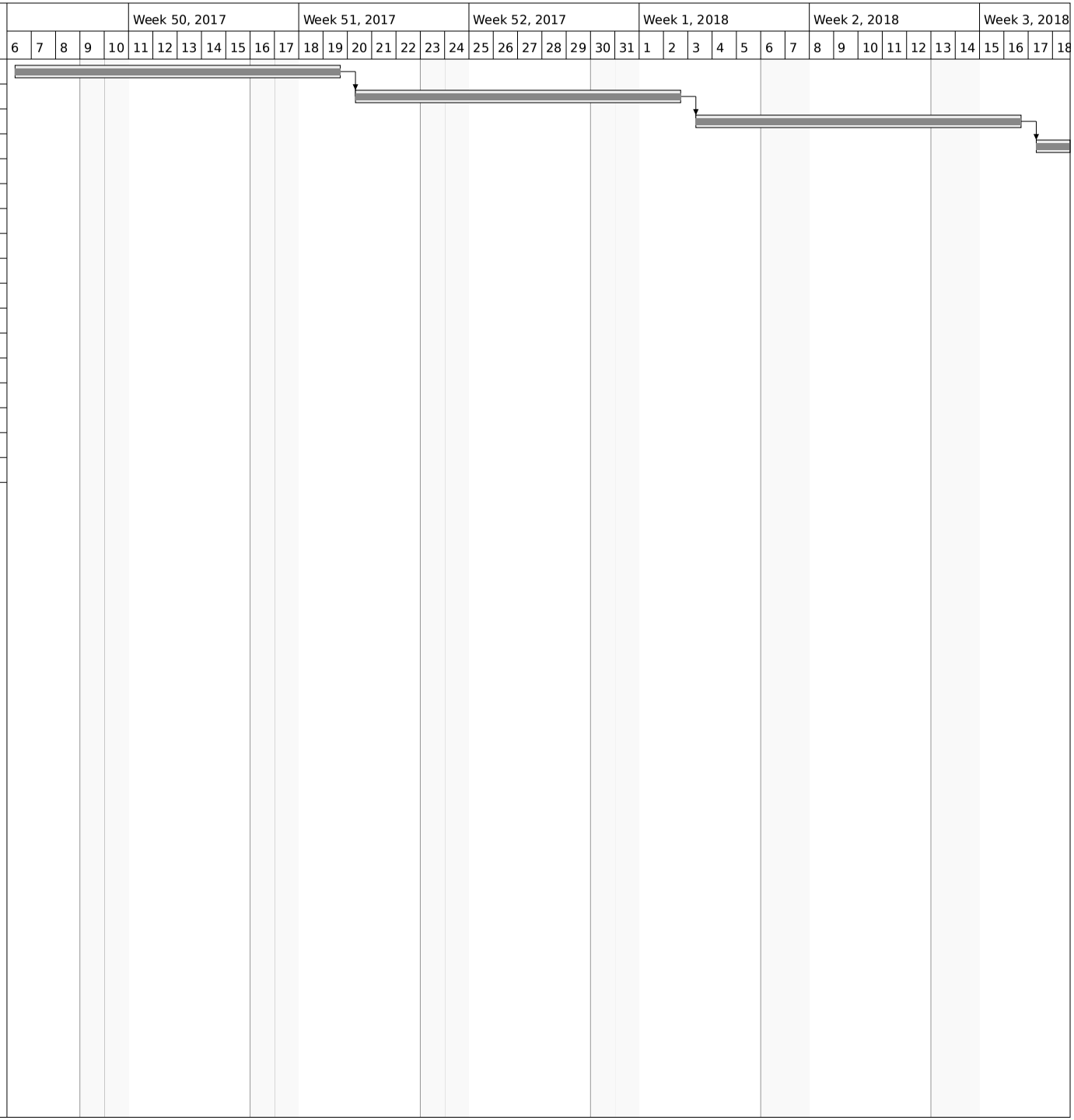
[35] Duane Wessels et al. «Specification for DNS over Transport Layer Security (TLS)». Em: (2016).

Anexos

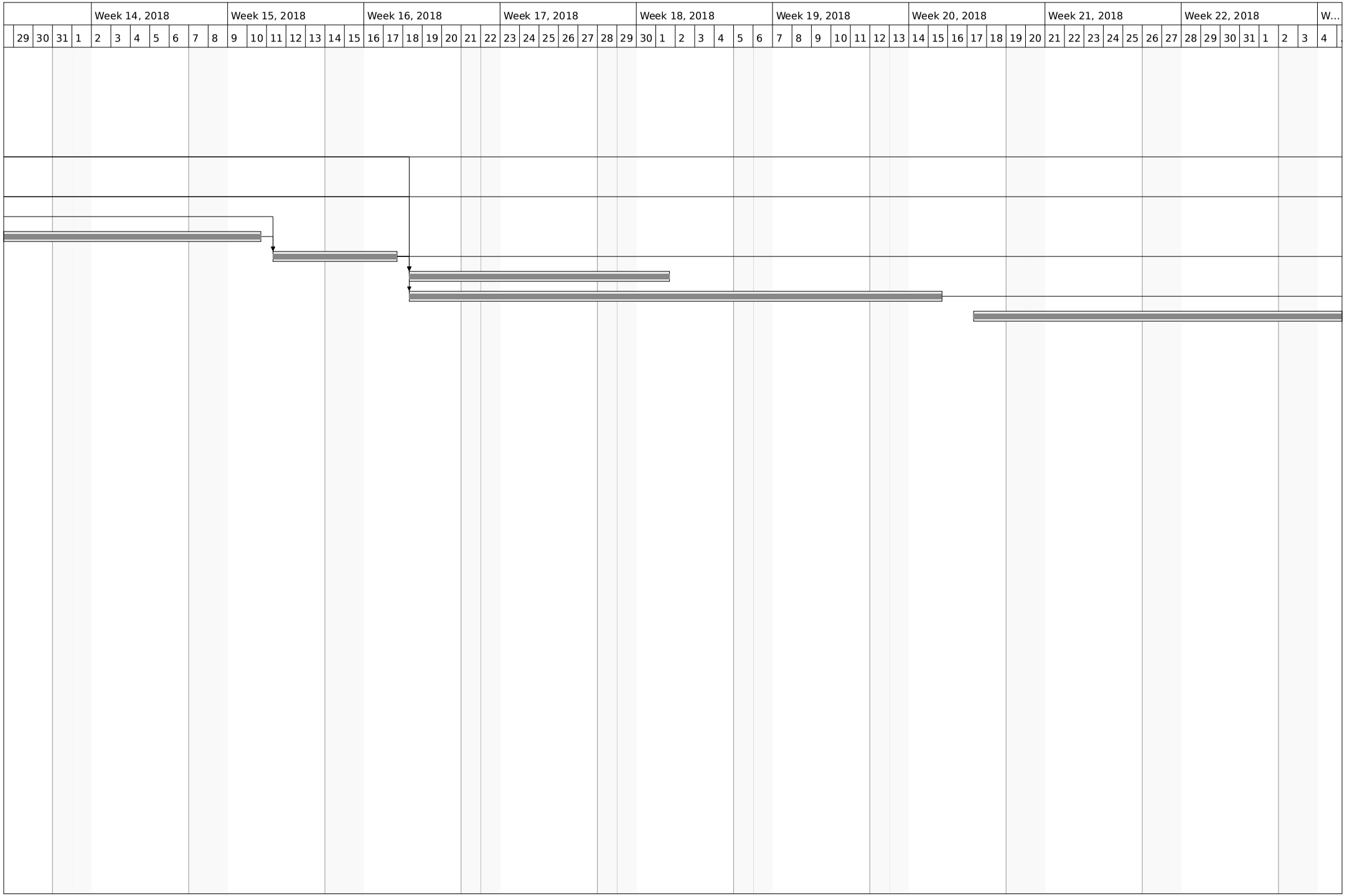
A

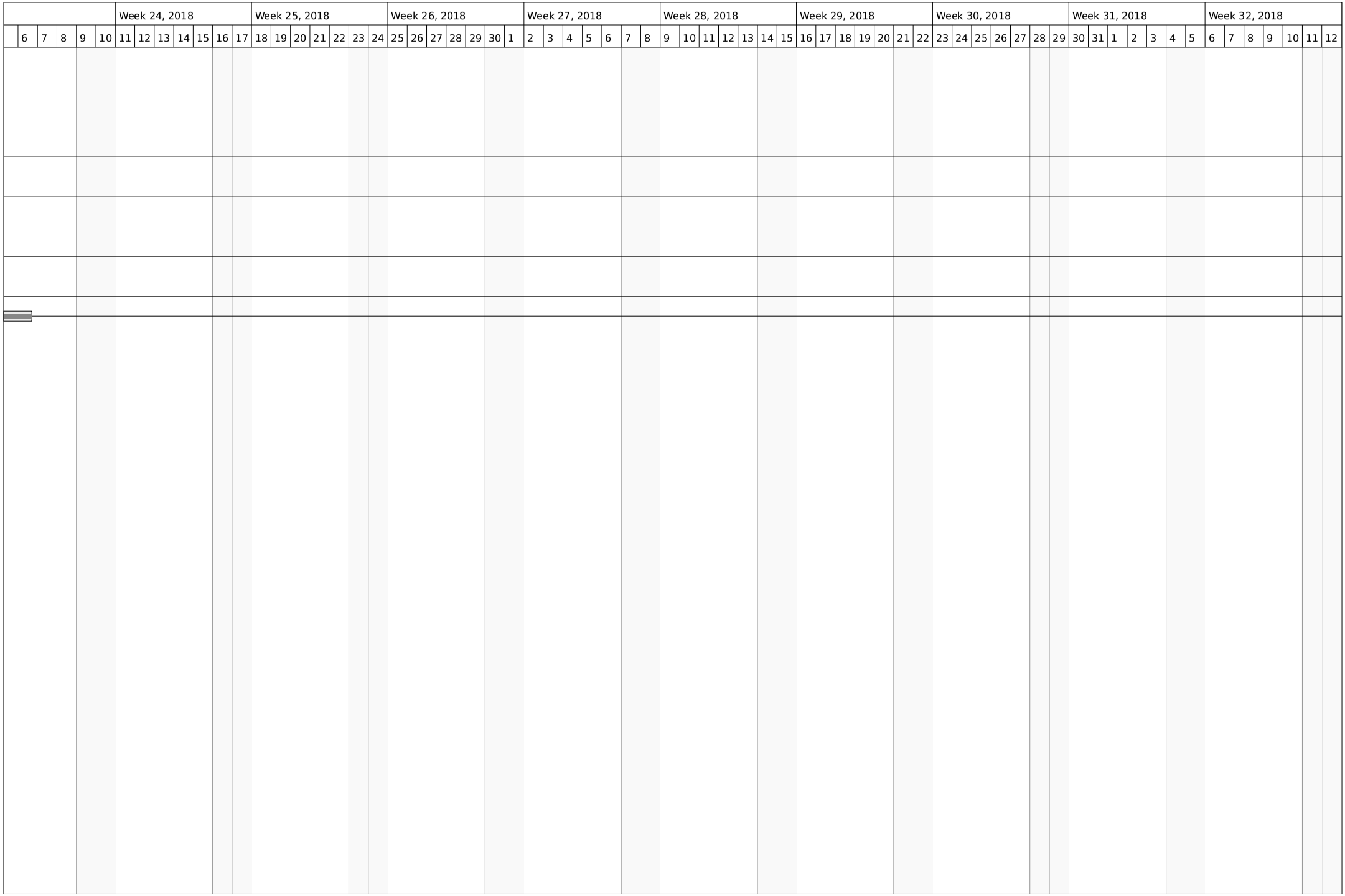
Gantt

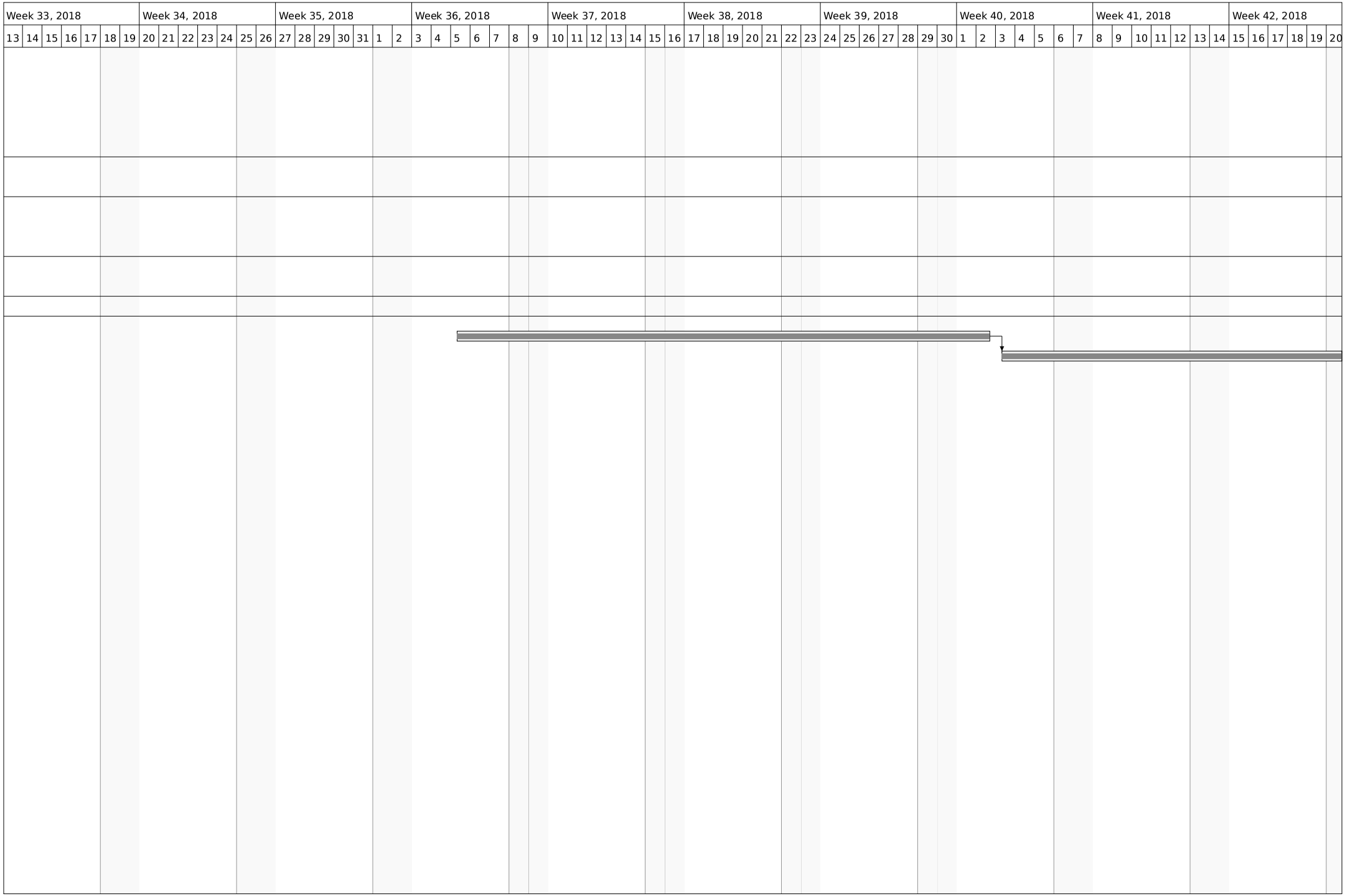
Name	Work
Análise do Problema	10d
Estado da Arte - Leitura de Papers e RFCs	10d
Análise comparativa Ferramentas Domain Squatting	10d
Desenvolvimento Algoritmo Geração de Typos	10d
Análise de fontes de domínios recém-criados e implementação de respetivo Bot	6d
Implementação do Bot de Domain Squatting	4d
Análise de dados e e desenvolvimento do algortimo de Classificação de Domínios	10d
Implementação do Bot Domain_Classifier	4d
Análise métodos comparativos de Websites	10d
Implementação analisador de aplicações web	15d
Implementação do Bot HTTP_Classifier	5d
Triagem de Falsos Positivos e Optimização dos diversos algoritmos	10d
Criação de Interface de Análise de Resultados	20d
Criação de Interface de Gestão de Configurações	15d
Análise de abordagens para a DNS Firewall	20d
Implementação da DNS Firewall	20d
Testes Finais da Solução	15d

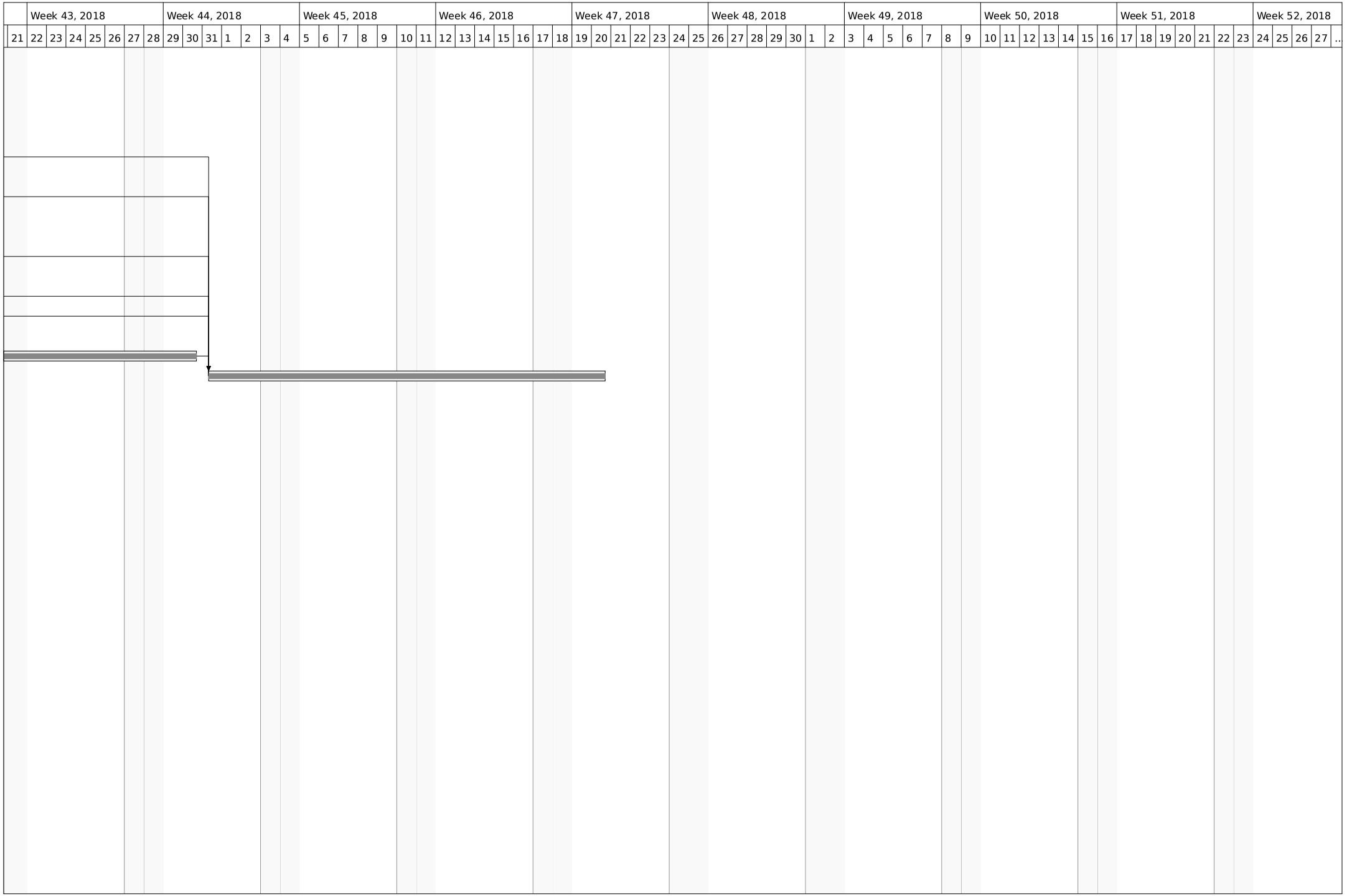



Week 4, 2018
Week 5, 2018
Week 6, 2018
Week 7, 2018
Week 8, 2018
Week 9, 2018
Week 10, 2018
Week 11, 2018
Week 12, 2018
Week 1...
19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 ..


Week 14, 2018
Week 15, 2018
Week 16, 2018
Week 17, 2018
Week 18, 2018
Week 19, 2018
Week 20, 2018
Week 21, 2018
Week 22, 2018
W...
29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4


Week 24, 2018
Week 25, 2018
Week 26, 2018
Week 27, 2018
Week 28, 2018
Week 29, 2018
Week 30, 2018
Week 31, 2018
Week 32, 2018
6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12


Week 33, 2018
Week 34, 2018
Week 35, 2018
Week 36, 2018
Week 37, 2018
Week 38, 2018
Week 39, 2018
Week 40, 2018
Week 41, 2018
Week 42, 2018
13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20


Week 43, 2018
Week 44, 2018
Week 45, 2018
Week 46, 2018
Week 47, 2018
Week 48, 2018
Week 49, 2018
Week 50, 2018
Week 51, 2018
Week 52, 2018
21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 ..

B

Proposta de Estágio

DEPARTAMENTO DE ENGENHARIA
INFORMÁTICA E DE SISTEMAS

PROPOSTA DE PROJECTO

Ano Lectivo de 2017/2018

em Mestrado em Informática e Sistemas - Tecnologias de Informação e do Conhecimento

TEMA

DNS as Security Shield and Security Intelligence Feed

SUMÁRIO

Cada vez mais os ataques à segurança da informação têm como alvo as pessoas, pois estas sendo mais suscetíveis a erros, são muitas vezes o meio mais fácil de comprometer uma infraestrutura. Muitos estudos recentes, apontam por exemplo o *Phishing* como principal meio de proliferação de *malware*, sendo uma das formas mais simples e eficazes para a sua distribuição, a usurpação de identidades associadas a entidades confiáveis e credíveis. Uma das técnicas utilizadas, consiste no registo de domínios semelhantes e que apelem ao erro do utilizador final.

Este estágio propõe o desenvolvimento de uma solução, que ofereça uma barreira adicional de segurança numa infraestrutura, tendo como foco o protocolo DNS. Esta solução deverá passar pela implementação de um *SinkHole* DNS, alimentado por diversas fontes de inteligência e ser capaz de produzir também Indicadores de Compromisso dos seus diversos utilizadores finais.

A solução contemplará ainda uma segunda abordagem, a monitorização contínua dos registos de novos endereços DNS, linguisticamente e visualmente semelhantes (ex:*punycode*) a instituições de interesse e/ou clientes, que analise o seu estado, com o objetivo de antecipar potenciais ameaças relacionadas com essas mesmas entidades (Ex: SpearPhishing).

1. ÂMBITO

A Dognaedis é uma empresa com foco em questões de segurança de informação, fornecendo aos seus clientes, serviços e soluções de segurança como Resposta a Incidentes e *Digital Security*, áreas que irão ser complementadas com esta solução.

Actualmente a Dognaedis dispõe de uma solução de Security Intelligence alimentada por fontes externas e internas, como por exemplo *honeypots*, que serve de suporte a serviços prestados internamente através da correlação da informação produzida com por exemplo Network IDS/IPS.

Este projeto irá enquadrar-se como mais uma fonte para essa plataforma, enquanto que deverá ao mesmo tempo prevenir a comunicação com ameaças externas através de DNS.

2. OBJECTIVOS

O estágio proposto, tendo uma forte componente de investigação, pretende ser uma solução de relevo na deteção antecipada de ameaças que utilizam o protocolo DNS como meio de disseminação. Dada a facilidade no registo de domínios, são cada vez mais frequentes campanhas direcionadas ou associadas a entidades de relevo, cujo o objetivo passa por criar uma falsa sensação de segurança a potenciais alvos e assim atingir o seu objetivo final. Desta forma pretende-se ajudar a resolver este problema através de duas abordagens.

A primeira, numa abordagem preditiva, centra-se no desenvolvimento de uma tecnologia capaz de prever e detetar, ataques baseados em DNS (disseminação de phishing, *landing pages*, etc) através da análise de domínios recém-criados. Desta fase, espera-se que resulte inteligência capaz de alimentar a fase seguinte.

A segunda, já numa abordagem reativa, consiste na implementação de um DNS Resolver, capaz de proteger os utilizadores de ameaças detetadas através da tecnologia desenvolvida na primeira fase, bem como outras ameaças, alimentado por outros Indicadores de Compromisso, cuja reputação indicie um risco para a segurança das pessoas e entidades. Esta componente do projeto deverá ainda e ser capaz de produzir e recolher ciber-inteligência relevante, como por exemplo a informação de ativos infetados, com base nas tentativas de resolução previamente realizadas.

3. PROGRAMA DE TRABALHOS

O estágio consistirá nas seguintes atividades e respetivas tarefas:

- ***T1*** – *Estudo preliminar do problema* – Nesta fase, o aluno pretende ter uma visão global do problema.
- ***T2*** – *Levantamento de Requisitos* – Tendo em conta o meio onde será inserida solução, o aluno deverá perceber quais os requisitos funcionais e não funcionais para as diversas funcionalidades que este projeto compreende.
- ***T3*** – Desenvolvimento – Investigação e desenvolvimento de uma solução que preencha os requisitos identificados anteriormente.
- ***T4*** – Testes – Testes funcionais e avaliação dos atributos de qualidade do trabalho desenvolvido anteriormente.
- ***T5*** – Relatório – Relatório descritivo do trabalho desenvolvido.

4. CALENDARIZAÇÃO DAS TAREFAS

As Tarefas acima descritas, incluindo os testes de validação de cada módulo, serão executadas de acordo com a seguinte calendarização:

O plano de escalonamento dos trabalhos é apresentado em seguida:

INI		Início dos trabalhos
M1	(INI + 2 Semanas)	Tarefa T1 terminada
M1	(INI + 4 Semanas)	Tarefa T2 terminada
M2	(INI + 18 Semanas)	Tarefa T3 terminada
M5	(INI + 24 Semanas)	Tarefa T4 terminada
M6	(INI + 28 Semanas)	Tarefa T5 terminada

5. RESULTADOS

Os resultados a apresentar serão apresentados ao longo das metas anteriormente definidas, sendo, no entanto, expectáveis duas fases de avaliação:

- Dezembro de 2017 – Avaliação preliminar da arquitetura e soluções propostas, sendo para tal avaliado o resultado das tarefas T1 e T2, validando a exequibilidade das soluções propostas.
- Julho de 2018 - Avaliação da solução e dos resultados obtidos pela mesma. Será também a avaliada a qualidade da documentação fornecida pelo aluno.

6. LOCAL DE TRABALHO

O trabalho irá ser desenvolvido no escritório da Dognaedis, em Coimbra.

7. METODOLOGIA

Embora sejam salvaguardadas 4 semanas no final para a elaboração da documentação descritiva e de suporte ao trabalho realizado, esta deverá contudo ser escrita ao longo da execução projeto.

É previsto que a elaboração deste projeto, seja assente numa metodologia ágil, com reuniões periódicas semanais que irão acompanhar o progresso do mesmo.

8. ORIENTAÇÃO

ISEC:

Luis Santos – (lsantos@isec.pt)
Teresa Rocha – (teresa@isec.pt)

Entidade de Acolhimento:

André Pinheiro (ampp@dognaedis.com)
Digital Security Manager, Mestre em Engenharia Informática