

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Guilherme Barbosa Junqueira Soares

**Sistema para identificar e bloquear endereços
de email descartáveis**

Uberlândia, Brasil

2019

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Guilherme Barbosa Junqueira Soares

**Sistema para identificar e bloquear endereços de email
descartáveis**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau
de Bacharel em Sistemas de Informação.

Orientador: Rodrigo Sanches Miani

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2019

Guilherme Barbosa Junqueira Soares

Sistema para identificar e bloquear endereços de email descartáveis

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Trabalho aprovado. Uberlândia, Brasil, 15 de julho de 2019:

Rodrigo Sanches Miani
Orientador

Professor
Pedro Henrique Aparecido Damaso de Melo

Professor
Rafael Pasquini

Uberlândia, Brasil
2019

Resumo

Com o surgimento de serviços de emails descartáveis surgiu a necessidade de alguns serviços online se protegerem contra cadastros falsos criados através de emails temporários.

Este trabalho possui como objetivo a criação de uma metodologia para a classificação de tais endereços de email e a aplicação de tal metodologia em um serviço web a ser consumido.

Dezenas de milhares de endereços de email foram submetidos à análise e classificação obtendo assim uma lista de domínios que foram devidamente classificados como legítimos ou descartáveis. Tal classificação agora pode ser obtida através de uma API que pode ser facilmente incorporada e utilizada livremente por qualquer serviço online.

Com a classificação de endereços de email pôde-se detectar uma taxa de aproximadamente 3,52% de endereços ilegítimos, o que eleva consideravelmente a qualidade das listas de endereços de email submetidas ao serviço objeto deste trabalho.

Palavras-chave: Email, spam, endereços de email descartáveis, classificação, API.

Lista de ilustrações

Figura 1 – Imagem ilustrando o real funcionamento do email. Adaptado de Oasis Open (2017).	13
Figura 2 – Imagem ilustrando o padrão MVT.	18
Figura 3 – Fluxograma ilustrando a classificação manual de domínio	25
Figura 4 – Fluxograma ilustrando o funcionamento básico da API	27
Figura 5 – Diagrama de entidades relacionais (ERD) do sistema	28
Figura 6 – Imagem do <i>hotsite</i>	28
Figura 7 – Imagem da página <i>Dashboard</i>	29
Figura 8 – Imagem da página <i>Requests</i>	30
Figura 9 – Imagem da página <i>Django administration</i>	31

Lista de tabelas

Tabela 1 – Exemplo de tabela DNS	14
Tabela 2 – Exemplo de classificação de domínios	23
Tabela 3 – Número de domínios classificados	25
Tabela 4 – Serviços de classificação de email	32

Lista de abreviaturas e siglas

A Record	<i>Address Record</i>
API	<i>Application Programming Interface</i>
BDEA	<i>Block Disposable Email Address</i>
BTE	<i>Block Temporary Email</i>
CNAME Record	<i>Canonical Name Record</i>
DEA	<i>Disposable Email Address</i>
DKIM	<i>DomainKeys Identified Mail</i>
DNS	<i>Domain Name System</i>
Email	<i>Eletronic Mail</i>
ICANN	<i>Internet Corporation for Assigned Names and Numbers</i>
IP	<i>Internet Protocol</i>
MDA	<i>Mail Delivery Agent</i>
MUA	<i>Mail User Agent</i>
MTA	<i>Mail Transfer Agent</i>
MX Record	<i>Mail Exchanger Record</i>
SaaS	<i>Software as a Service</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SPAM	Envio de email não solicitados
TLD	<i>Top Level Domain</i>
TXT Record	<i>Text Record</i>

Sumário

1	INTRODUÇÃO	9
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	Email	12
2.2	DNS	14
2.2.1	Tabela DNS	14
2.3	Spam	15
2.4	<i>Disposable Email Address</i>	16
2.5	<i>Software as a Service</i>	16
2.6	Tecnologias	17
2.6.1	Web Framework	17
2.6.2	Serverless	19
2.6.3	Application Programming Interface (API)	19
3	TRABALHOS CORRELATOS	21
3.1	Trabalhos acadêmicos	21
3.2	Serviços	21
3.2.1	Block Disposable Email Address	22
3.2.2	Kickbox	22
4	DESENVOLVIMENTO	23
4.1	Metodologia de classificação	23
4.1.1	Identificação de provedores de emails descartáveis	23
4.1.1.1	Lista inicial de domínios	24
4.1.1.2	Classificação	24
4.1.2	Identificação entradas DNS válidas	26
4.2	Sistema	26
4.2.1	Modelagem	26
4.2.2	Interface	27
4.2.3	API	28
4.2.4	Administração	30
4.3	Análise	31
4.3.1	Amostra	31
4.3.2	Resultados	32
5	CONCLUSÃO	33

Conclusão	33
REFERÊNCIAS	34
APÊNDICES	36
APÊNDICE A – DOCUMENTAÇÃO COMPLETA <i>BLOCK TEMPORARY EMAIL</i>	37
A.1 Getting started	37
A.2 What is Block Temporary Email	37
A.3 Do's and Don'ts	37
A.4 Endpoints	38
A.5 Requests and Resposes	38
A.6 All API's actions	38
A.6.1 Check	38
A.7 Rate limiting	39
A.8 How to get Username and API Key	40
A.9 Examples	40
A.9.1 Python	40
APÊNDICE B – LISTA DE DOMÍNIOS IDENTIFICADOS COMO PROVEDORES DE EMAIL DESCARTÁVEIS	41

1 Introdução

O email (eletronic mail ou e-mail) é uma das primeiras aplicações efetivas para comunicação pela Internet, surgindo na década de 60. O protocolo utilizado atualmente para troca de emails, SMTP (*Simple Mail Transfer Protocol*), foi introduzido em 1982 ([Internet Engineering Task Force, 1982a](#)) e tendo sua última atualização em 2015 ([Klensin J, 2015](#)). Tal protocolo especifica quais mensagens podem ser trocadas por dispositivos através da referência de um endereço, conhecido como endereço de email (*email address*). Tais endereços são criados e mantidos por servidores específicos (*SMTP Server*) de maneira descentralizada que tem o papel de armazenar os emails recebidos e enviar novos emails.

Com a modernização da Internet, surgiram os primeiros serviços públicos de email (Hotmail, Gmail, Yahoo, entre outros), que facilitaram o acesso aos servidores de email, proporcionando a criação de endereços eletrônicos de forma barata ou até mesmo gratuita a qualquer usuário. Rapidamente o endereço de email consolidou-se como identificação universal de um usuário na Internet, sendo na grande maioria dos casos o principal requisito para um cadastro em qualquer que seja o serviço online. Criando assim um vínculo entre sua identidade física e seu endereço eletrônico.

Pela grande facilidade de acesso a este serviço, agentes maliciosos observaram a oportunidade de atingir milhões de pessoas a um custo mínimo e esta prática ficou conhecida como spam ([Internet Engineering Task Force, 1982b](#)), que consiste no envio de mensagens não solicitadas em massa. O spam possui o objetivo principal de difundir publicidade ou programas maliciosos. Houve então a necessidade de criar barreiras ao recebimento de tais mensagens e assim, diversos métodos foram criados para identificar, filtrar e bloquear os mesmos. Dentre eles, a análise do cabeçalho e autenticação DKIM (*DomainKeys Identified Mail*) ([ALLMAN et al., 2007](#)), *blacklists* de domínios ([COOK et al., 2006](#)) e até mesmo inteligência artificial para reconhecimento de padrões de email indesejados ([ANDROUTSOPOULOS et al., 2000](#)).

Um outro método para barrar o recebimento de spam foi a criação de serviços de emails temporários ou descartáveis, também conhecido como *Disposable Email Address* (DEA), que tem como objetivo fornecer um endereço de email temporário que se auto destrói após o uso. Ele consiste em criar um email a ser usado em um cadastro em um serviço online qualquer, substituindo assim a necessidade do uso do seu endereço de email pessoal e verdadeiro. Este método também possui o benefício do usuário manter sua privacidade, escondendo seu endereço de email real e causando uma disruptura de sua identidade física/virtual ([SEIGNEUR; JENSEN, 2003](#)).

Alguns sites que fornecem tais serviços gratuitamente:

- Temp-mail (<https://temp-mail.org/pt/>);
- Mohmail (<https://www.mohmal.com/pt/>);
- Gerrilla Mail (<https://www.guerrillamail.com/pt/>).

Com a facilidade de obtenção de endereços de email gratuitos como citado anteriormente e a criação de emails descartáveis (DEA) há também a facilidade do uso de múltiplos endereços criados de forma arbitrária e muitas vezes automatizada para cadastro em serviços online sem que haja a contrapartida requerida pelo serviço de possuir um usuário real e único. Tais serviços fornecem benefícios aos usuários na maioria das vezes gratuitos porém com limitações e fornecem a opção de *upgrade* para planos pagos e mais completos, sendo o cadastro gratuito uma forma de teste ao usuário e até mesmo como publicidade do serviço.

Podemos citar como exemplo o serviço de cursos a distância [Solyd Treinamentos \(2019\)](#), que oferece treinamentos *online* em tecnologia da informação, alguns deles gratuitos. Tais treinamentos gratuitos tem como objetivo a captura de um endereço de email real para que treinamentos mais completos e pagos possam ser oferecidos posteriormente ao aluno, porém caso o aluno crie uma conta utilizando um endereço de email descartável o site em questão não poderá contatar o aluno posteriormente e efetuar uma possível venda.

A automação e criação de múltiplas contas tem como objetivo obter vantagens sobre esses sistemas. Como esperado, cada usuário indesejado representa um custo maior às empresas, surgindo então a necessidade de uma melhor filtragem dos usuários cadastrados.

A filtragem poderia constituir-se de um método com os seguintes passos: bloquear usuários de serviços de emails gratuitos, restringindo o uso do serviço somente a usuários de domínios corporativos; restringir usuários de serviços de emails descartáveis ou temporários; restringir emails conhecidos como robôs ou spam encontrados em *blacklists*.

Tais restrições podem ser implementadas internamente, seguindo as necessidades de cada empresa, porém tal objetivo poderia se tornar economicamente inviável, pois precisaria de uma grande alocação de recursos para o desenvolvimento e manutenção de um serviço de filtragem próprio realmente efetivo.

Uma melhor abordagem seria a contratação de um serviço especializado para a identificação e filtragem de endereços de email não desejados.

Considerando os aspectos abordados, espera-se que este trabalho obtenha sucesso na criação de um serviço que será capaz de reduzir o número de cadastros não desejados

(DEA) em serviços na Internet, fornecendo um meio de filtragem e catalogação de reputação em tempo de cadastro através de um serviço sob demanda em nuvem, conhecido como *Software as a Service* (SaaS).

Este trabalho tem como objetivo solucionar a necessidade de criação de um método para a validação de endereços de email confiáveis, diminuindo o número de cadastros inválidos ou falsos em sites de serviços *online*.

Tal método será baseado em classificação via *blacklist* e *whitelist* de domínios, essa classificação será feita a partir de análise manual em conjunto com análises em múltiplos outros serviços de classificação de domínios.

O sistema então ficará disponível publicamente através de uma API (*Application Programming Interface*) *web*.

Este trabalho está dividido em capítulos: na fundamentação teórica serão apresentados os principais conceitos e tecnologias por trás da criação do mesmo, após isso serão apresentados trabalhos acadêmicos e serviços correlatos a este, no desenvolvimento serão descritas as metodologias e processos utilizados para a construção deste trabalho e avaliação dos resultados decorrentes do mesmo, e ao final será apresentada a conclusão com considerações finais e possíveis trabalhos a serem propostos no futuro.

2 Fundamentação Teórica

Neste capítulo serão descritos e explicados os principais conceitos e tecnologias que fundamentam a elaboração deste trabalho.

2.1 Email

O email é um meio de comunicação largamente utilizado pelos usuários da rede mundial de computadores, tem como objetivo a troca de documentos estáticos em texto de maneira distribuída. O protocolo que comanda a distribuição e recepção de email é o SMTP ([Internet Engineering Task Force, 1982a](#); [Klensin J, 2015](#)).

Destinatário, remetente, assunto, corpo e anexos são os componentes principais de uma mensagem de email e todas essas informações são passadas a um servidor SMTP para que a mensagem seja enviada. Um servidor SMTP pode estar presente em um serviço local, em uma máquina dedicada dentro de uma infraestrutura corporativa ou até mesmo em um outro país, não havendo qualquer tipo de controle ou restrição ou regulação a este tipo de serviço, bastando a aplicação/servidor respeitar os princípios descritos no protocolo. As trocas de mensagens são feitas entre endereços de email, que convencionalmente possuem um nome de usuário seguido de um nome de domínio ou IP separados por um símbolo de arroba (@), exemplo: *bob@example.com*, em que "bob" é o usuário e identificador único dentro do domínio definido como "example.com". Com o uso no dia a dia o envio e recebimento de emails parece uma tarefa trivial, feita ponta a ponta, porém envolve passos bem mais complexos como podemos ver na figura 1. O envio e o recebimento de email pode ser detalhado em três fases bem definidas:

1. Escrita do email: o remetente precisa de um software denominado MUA (Mail User Agent) para redigir o email e enviá-lo, tal software na prática pode ser conhecido como clientes locais de email como Microsoft Outlook ou serviços de *webmail* como Gmail, Hotmail ou Yahoo. No MUA é redigido o corpo do email e são definidas algumas informações que vão no cabeçalho como destinatário, assunto e informações ocultas definidas pelo MUA como endereços de IP de origem e informações de autenticação, entraremos em detalhes mais a frente. O remetente também pode ser definido livremente pelo MUA, embora na prática isso seja restrito pela maioria dos softwares e serviços disponíveis, o protocolo dá a liberdade para que o usuário possa definir livremente o remetente, acarretando alguns problemas de segurança a serem discutidos posteriormente neste trabalho.
2. Roteamento do email: o MUA então passa o email ao MTA/MDA (*Mail Transfer*

Agent/Mail Delivery Agent), funções geralmente exercidas pelo mesmo software, entre eles o Postfix e Sendmail. Estes tem a função de identificar o destinatário e rotear o email. Em caso de email local: entregar o email ao usuário local; em caso de email externo: identificar o servidor responsável pelo recebimento de emails, isso é feito através de uma consulta do MX Record na tabela DNS do domínio do destinatário (falaremos mais sobre consultas DNS posteriormente), tal consulta retorna um endereço de IP, no qual, deverá haver um serviço MTA equivalente para o recebimento e manejo das mensagens de email.

3. Entrega: o MTA do remetente então repassa o email ao MTA do destinatário que repassa ao MDA que deixará o email disponível em sua respectiva caixa de emails. A leitura do mesmo será efetuada por clientes de email, na prática utilizando serviços de *webmail* ou *softwares* que implementam protocolos de leitura de email como POP3 (MYERS; ROSE, 1996) e IMAP (CRISPIN, 2003). Muitas vezes o MTA do remetente passa o email por alguns filtros anti-spam e antivírus antes de efetuar a entrega de fato, porém essa etapa não é obrigatória e nem sempre está presente.

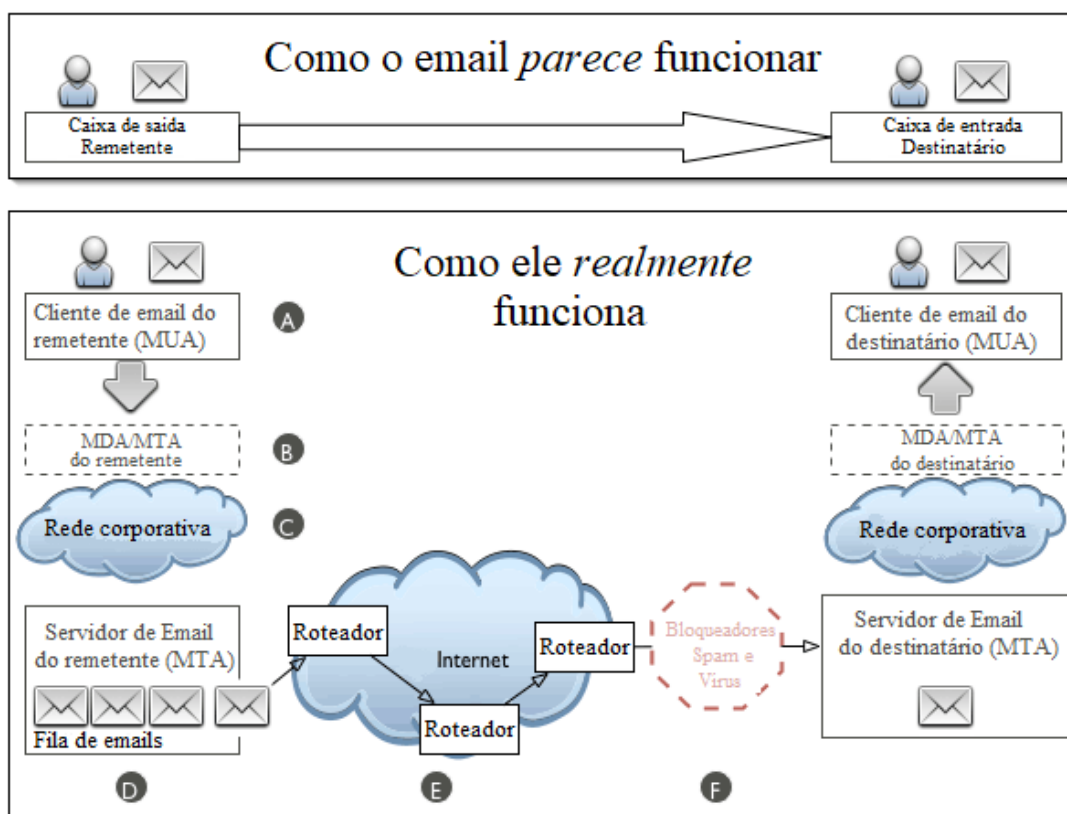


Figura 1 – Imagem ilustrando o real funcionamento do email. Adaptado de [Oasis Open \(2017\)](#).

2.2 DNS

Domain Name System (DNS) é um protocolo de gerenciamento de nomes de domínios, funciona de maneira hierárquica e descentralizada ([MOCKAPETRIS, 1987](#)). Possui como função traduzir nomes, chamados de domínios em endereços de IP com o objetivo de facilitar ao acesso global de servidores pela *World Wide Web*. Devido à implementação de DNS somos capazes de decorar e acessar rapidamente centenas de *sites* sem a necessidade de sabermos seus respectivos endereços de IP. Exemplo: o domínio *example.com* redireciona para o IP *93.184.216.34*.

O DNS pode ser gerenciado de maneira local através de servidores DNS ou acessado de maneira global através de servidores raiz espalhados pelo globo. Assim podemos criar um sistema de nome de domínios interno a uma organização em que os nomes traduzidos serão redirecionados em IPs privados ou podemos acessar um sistema global e distribuído que possui 13 servidores raiz (*root servers*), nenhum desses servidores armazenam todos os domínios registrados no mundo, porém quando acessados redirecionam a chamada para o respectivo servidor TLD (*Top Level Domain Server*) contendo o domínio requisitado. O gerenciamento de registro de domínios é regulado pela ICANN (*Internet Corporation for Assigned Names and Numbers*) ([ICANN, 2012](#)).

2.2.1 Tabela DNS

A tabela DNS ou *DNS records* é uma tabela que possui uma lista de IPs e seus respectivos alias para um determinado domínio e seus subdomínios. Na Tabela 1 temos um exemplo de um *DNS record* para o domínio *example.com*.

Tabela 1 – Exemplo de tabela DNS

Tipo	Nome	Valor
A	example.com	93.184.216.34
A	www.example.com	93.184.216.34
MX	example.com	93.184.216.34
CNAME	ftp.example.com	example.com

Na Tabela 1 podemos notar diferentes tipos de entradas, como: A, MX e CNAME. Esses são os tipos de entradas mais comuns. Cada entrada possui uma função distinta de tradução de nomes:

- A: possui a função de alias para um domínio, quando se acessa determinado endereço gravado em "Nome" o mesmo será traduzido para o IP indicado no "Valor".
- CNAME: possui a função parecida com A, porém pode traduzir e redirecionar nomes de domínio para outros nomes de domínio. No exemplo acima temos que

ftp.example.com será redirecionado para *example.com* e consequentemente para o IP *93.184.216.34*.

- MX: a entrada MX (Mail Exchange) indica em qual IP está localizado o MTA do domínio, portando quando um email é enviado a *bob@example.com*, o MTA do remetente sabe que precisa enviar o email para o IP *93.184.216.34*, onde há um servidor SMTP esperando por emails a serem entregues ao destinatário.

2.3 Spam

Spam é o envio de emails não desejados em massa. Possui o objetivo de difundir publicidade ou programas maliciosos (*malware*). O termo originou-se de uma esquete do grupo humorístico britânico Monty Python se referindo à carne suína enlatada da marca Spam produzida pela Hormel Foods Corporation ([Internet Engineering Task Force, 1982b](#)).

A prática se popularizou devido à facilidade de coleta de endereços e email e envio em massa a milhares ou até mesmo milhões de emails a um baixíssimo custo, bastando o acesso a um servidor SMTP, que pode ser criado e hospedado localmente e uma lista de emails.

Com o crescimento da prática de envios de spam houve a necessidade de se adotar contramedidas a fim de sanitizar as caixas de entrada e separar emails legítimos do lixo eletrônico. Serão listadas algumas técnicas adotadas por servidores de emails para barrar tais mensagens indesejadas:

- *Blacklists*: como o nome sugere, são listas negras contendo mensagens, endereços de email, domínios e IPs conhecidos pelas práticas de spam, trata-se de uma técnica primitiva, porém eficaz, que depende da ação de usuários e especialistas na detecção e denúncia de prováveis emails indesejados ([COOK et al., 2006](#)).
- Análise de cabeçalhos e DKIM: a implementação do SMTP deixa a especificação do cabeçalho de email livre para ser manipulado, podendo assim um remetente falsificar sua identificação, provendo um endereço de email ou domínio de origem falsos, tal técnica é conhecida como *email spoofing*. Para sanar tal problema foi proposto por [Allman et al. \(2007\)](#) o uso de chaves DKIM (*DomainKeys Identified Mail*) com o objetivo de garantir a autenticidade e integridade do email. A autenticação é feita através de um par de chaves pública/privada geradas pelo dono do domínio. A chave pública é publicada na tabela DNS através de uma entrada do tipo TXT e a chave privada fica disponível no servidor SMTP de saída, quando o email é enviado o servidor então assina a mensagem usando a chave privada e inserindo tal assinatura no cabeçalho do email, ao receber o email o destinatário então confirma

tal assinatura através de uma consulta à chave pública na tabela DNS do domínio, garantindo assim que a mensagem foi enviada e autorizada pelo remetente e que o cabeçalho e conteúdo da mesma não foi modificado no caminho.

- Inteligência artificial: como mostrado por [Androutsopoulos et al. \(2000\)](#), são empregadas técnicas da análise de conteúdo e cabeçalho de email, usando técnicas de *machine learning* para identificar padrões em mensagens indesejadas e fazer a catalogação em conjunto com outras análises qualitativas citadas acima. Assim torna-se possível filtrar spam antes mesmo que ele seja posto em uma *blacklist* e obedeça a padrões de autenticação modernos.

2.4 Disposable Email Address

Assim como as técnicas discutidas acima, o uso de *Disposable Email Address* (DEA) ou endereços de email descartáveis também se tornou uma arma contra o recebimento de spam. Porém diferente das abordagens já citadas essa é uma medida adotada por usuários e não por provedores de email. Trata-se na criação de endereços de emails temporários e descartáveis que geralmente serão utilizados apenas uma única vez. Tal uso é feito para o cadastro em qualquer tipo de serviço online que requisite um endereço de email confirmado para a ativação do cadastro, protegendo assim seu verdadeiro endereço de email contra qualquer tipo de vazamento, distribuição ou comercialização que tal provedor de serviço possa efetuar com a posse de seu email, além de oferecer uma proteção à privacidade do usuário como já abordado por [Seigneur e Jensen \(2003\)](#).

Um exemplo de um serviço online que oferece a criação de emails descartáveis gratuitamente é 10 Minute Mail (<https://10minutemail.com>). Ao acessar o site, automaticamente já é criado um endereço único capaz de receber emails e mostrá-los para leitura. Tal endereço será automaticamente destruído dentro de dez minutos. Na página sobre o serviço podemos notar a motivação citada pelos autores:

"Por que você usaria isto? Talvez você queira registrar-se em um site que solicite que você forneça um endereço de email no qual receberá um email de validação. E talvez você não queira fornecer seu endereço de email verdadeiro e acabar em uma série de listas de spam. Isto é bom e descartável." ([10 Minute Mail, 2017](#)).

2.5 Software as a Service

Software as a Service (SaaS) é um termo adotado para se referir a um modelo de negócios e licenciamento de softwares sob demanda, hospedados de maneira centralizada que geralmente possuem taxas de assinatura recorrentes ou proporcionais à demanda necessária. SaaS foi projetado para separar a posse e propriedade do software com

seu uso, como relatado por [Turner, Budgen e Brereton \(2003\)](#). Neste modelo o software e seu código ficam protegidos em servidores centralizados controlados pelo provedor de tal serviço oferecendo uma camada de uso ao usuário final.

Essa nomenclatura faz parte de produtos oferecidos "*as-a-Service*" com a revolução em *cloud computing* como discutido por [Cheang \(2014\)](#).

O objetivo final deste trabalho é a elaboração de um SaaS que ofereça o benefício ao usuário de poder catalogar e filtrar endereços de email.

2.6 Tecnologias

Nesta seção serão discutidas as principais tecnologias utilizadas para o desenvolvimento da ferramenta de classificação de endereços de email.

Será desenvolvido um serviço *web* que permitirá a consulta e classificação de endereços de email via API, este mesmo serviço permitirá a visualização de dados e estatísticas e a completa administração da plataforma.

2.6.1 Web Framework

Um Web Framework é um conjunto de ferramentas e códigos reusáveis e modulares que permitem o desenvolvimento de aplicações *web* dinâmicas de maneira mais rápida, organizada, segura e escalável. Tais frameworks expõem bibliotecas e APIs para o desenvolvimento ágil de aplicações e permite que o usuário se preocupe apenas com a lógica de negócio que deseja fornecer ao seu cliente.

No desenvolvimento da plataforma será utilizado um *Web Framework* baseado em Python: *Django Web Framework*.

O Django é um *framework* de alto nível que permite o desenvolvimento de aplicações *web* complexas e modulares a partir de um núcleo cheio de recursos que fornecem a base para a construção de qualquer aplicação.

Este *framework* utiliza um padrão de projeto chamado MVT (*Model - View - Template*), tal padrão divide o projeto em:

- *Model*: declara, relaciona e controla as principais entidades presentes na aplicação, fornecendo assim uma API para interação com um banco de dados relacional ao mesmo tempo que permite a adoção de lógicas de negócio específicas a cada entidade, permitindo assim um código reusável e de fácil manutenção;
- *View*: recebe e processa as entradas de usuário e é responsável pela resposta da requisição, correlacionando assim os dados, via *models*, e a forma de exibição, via

templates ou requisições HTTP puras;

- *Template*: permite que lógicas de exibição sejam utilizadas para renderizar páginas HTML ao usuário;

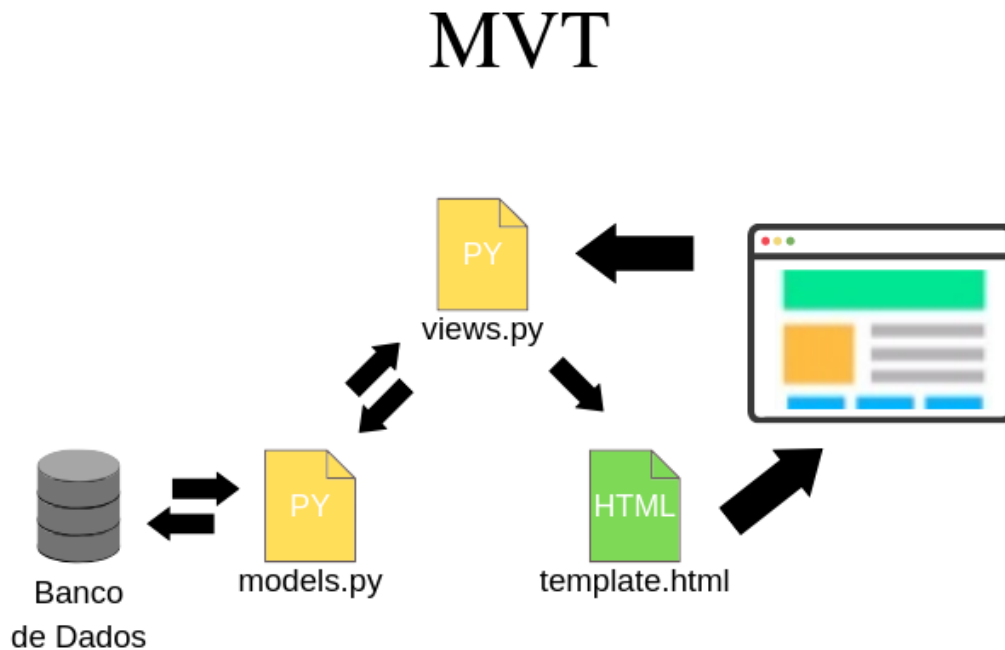


Figura 2 – Imagem ilustrando o padrão MVT.

Entre os principais recursos do Django estão:

- Usuários e papéis: permite que usuários sejam criados com diferentes tipos e níveis de permissão e controle.
- Autenticação e controle de sessão: permite o cadastro e o *login* seguro de usuários;
- ORM (*Object Relational Mapper*): fornece uma API para a interação com bancos de dados relacionais, transformando as entidades do banco em classes (*models*) e objetos, tal API permite que o banco de dados seja manipulado de maneira segura e força a utilização de um padrão de projeto coerente ao Django;
- Django Admin: fornece uma interface de administração para fácil controle da aplicação *web*.

2.6.2 *Serverless*

A arquitetura *Serverless*, uma das vertentes do *Cloud-computing*, permite que aplicações sejam postas em produção sem a necessidade de manter uma infraestrutura de servidores própria. Ao invés disso são utilizados serviços que provêm contêineres efêmeros que executam a aplicação pelo período necessário e se desligam. Tais serviços são provisionados e executados através de virtualização em uma estrutura redundante de *clusters* de servidores físicos.

Tal arquitetura apresenta algumas vantagens a serem exploradas por este trabalho, entre elas:

- Nenhuma manutenção com servidores e sistemas operacionais;
- Baixo custo;
- Alta escalabilidade;
- Alta disponibilidade;
- Segurança;
- Alto desempenho;

Será utilizado o serviço *serverless* AWS Lambda. Que permite um fácil *deploy* da aplicação *web*, possuindo uma disponibilidade próxima de 100% a um baixo custo. Este serviço é completamente gerenciado pela AWS e não é necessária a instalação e manutenção de servidores Linux, tão pouco há necessidade em se preocupar com a segurança de tais elementos.

O modelo de cobrança é por tempo de uso de CPU em uma escala de 100 ms. O preço para a utilização da CPU por 100 ms é de US\$ 0,000001667, utilizando 1024 MB disponíveis para memória. Ou seja, em uma requisição em que a CPU foi ocupada por 540 ms a cobrança será arredondada para 600 ms e a conta será de US\$ 0,000010002. Para 1 milhão de requisições temos a quantia aproximada de US\$ 10,00.

Tal serviço pode ser facilmente escalado e consegue lidar com milhões de requisições simultâneas sem que haja degradação da velocidade ou disponibilidade.

2.6.3 *Application Programming Interface (API)*

Application Programming Interface ou API como a tradução literal aponta trata-se de uma interface de programação de aplicações. Tem como objetivo permitir a interação entre diferentes programas utilizando padrões para a comunicação e controle.

Este trabalho expõe publicamente uma API do tipo REST (*Representational State Transfer*), sendo este um padrão muito comum em aplicações *web*.

A API Rest utiliza-se de requisições via protocolo HTTP feitas a uma URI, tais requisições enviam e recebem cargas de dados que podem ser em vários formatos como: XML, HTML, JSON, entre outros.

Este trabalho adota o padrão JSON (*JavaScript Object Notation*) para as transações na API.

Exemplo de um JSON que descreve alunos e seus respectivos nome e número de matrícula:

```
{"Alunos": [
  { "nome": "João Silva", "matricula": 1 },
  { "nome": "Edson Arantes do Nascimento", "matricula": 2 },
  { "nome": "Renata Reis", "matricula": 3 }
]}
```

3 Trabalhos Correlatos

Nesta seção alguns trabalhos já realizados que se relacionam com o tema deste serão abordados. Será discutida a importância para a retomada da privacidade utilizando emails descartáveis segundo [Seigneur e Jensen \(2003\)](#) e a implementação de serviços análogos ao proposto por este trabalho feitos por [BDEA \(2017\)](#) e [KickBox \(2019\)](#).

3.1 Trabalhos acadêmicos

[Seigneur e Jensen \(2003\)](#) propõe o uso de DEA como ferramentas para o aumento da privacidade e proteção anti-spam. É proposta e implementada uma ferramenta e um protocolo, REAP (*Rolling Email Address Protocols*), para automação da criação e manutenção de endereços de emails descartáveis que podem ser destruídos assim que haja o comprometimento do mesmo por spam. O trabalho alerta para a perda de privacidade ocasionada pela assinatura de serviços utilizando sempre o mesmo endereço de email que muitas vezes pode ser comercializado ou distribuído a outros serviços sem a devida autorização. O uso de um endereço de email real e único em diversos serviços também pode ser usado para o cruzamento de informações e identificação precisa de um usuário, portanto o uso de um DEA proporciona a disruptura de uma identidade física/virtual impossibilitando o cruzamento de informações e identificação de usuários via endereços de email compartilhados entre serviços.

3.2 Serviços

Paradoxalmente o mesmo DEA utilizado para combater spam tornou-se uma poderosa ferramenta para a prática de spam. Usando essa ideia, programas podem gerar automaticamente milhares de diferentes emails e automatizar processos de cadastros, comentários, *reviews*, *ratings* e inúmeras outras ações em serviços online que requerem um endereço de email para autenticação. [BDEA \(2017\)](#) e [KickBox \(2019\)](#), assim como este trabalho, propõem uma solução para a prática de spam e cadastro anônimo gerado por serviços DEA. O trabalho prevê três alternativas para um serviço online que sofra com cadastro indesejado de usuários:

- continuar recebendo cadastros indiscriminadamente, sejam eles de emails legítimos ou temporários;
- manter uma *blacklist* local, com domínios e endereços DEA;

- usar *blacklists* públicas que podem ocasionar em falsos positivos e também falsos negativos.

Entretanto este trabalho propõe uma nova alternativa: o uso de um serviço online sob demanda e centralizado, com uma equipe dedicada a manter uma *blacklist* sempre atualizada e funcional, minimizando praticamente a zero o número de falsos positivos e negativos.

3.2.1 Block Disposable Email Address

Este serviço pode ser acessado em <https://www.block-disposable-email.com/cms/> e conta com uma API para a classificação de emails descartáveis porém possui um *dashboard* defasado o que dificulta o acompanhamento das chamadas de API, possui um alto tempo de resposta e um banco de dados desatualizado com muitos falsos negativos.

3.2.2 Kickbox

Kickbox é uma empresa que oferece serviços de classificação de emails para publicidade e desenvolvedores, eles oferecem uma API aberta e gratuita para a checagem de endereços de emails descartáveis. Este serviço pode ser acessado em <https://open.kickbox.io>, dentre as vantagens possui uma API de fácil integração, não necessita de chave e possui um baixíssimo tempo de resposta, como desvantagem apresenta alguns falsos negativos.

4 Desenvolvimento

Este capítulo mostra o desenvolvimento e execução deste trabalho, mostrando metodologias, resultados e análises do serviço implementado.

4.1 Metodologia de classificação

O grande desafio desde trabalho foi desenvolver uma metodologia eficiente para classificação e identificação de emails descartáveis.

A abordagem escolhida foi a utilização de *blacklists* e *whitelists*, em um detalhado banco de dados de domínios.

Os seguintes itens foram checados em cada domínio:

- Se trata-se de um conhecido provedor de emails descartáveis;
- Se possui entradas MX em sua tabela DNS.

Cada um desses itens possuem respostas booleanas, a serem correlacionados em uma tabela:

Tabela 2 – Exemplo de classificação de domínios

Domínio	Descartável	DNS Válido
gmail.com	Falso	Verdadeiro
example.com	Falso	Falso
email-lab.com	Verdadeiro	Verdadeiro

Na tabela acima podemos notar a assinatura de um domínio de emails descartáveis: *email-lab.com*, possui tanto entradas DNS válidas como é um conhecido provedor de emails descartáveis.

4.1.1 Identificação de provedores de emails descartáveis

A identificação de domínios utilizados em endereços de email descartáveis não é uma tarefa trivial e a abordagem mais efetiva foi a utilização de checagem manual, através de coleta de informações usando DNS, checagem de sites ligados ao domínio e à utilização de motores de busca para encontrar referências a determinado domínio, checagem de listas públicas e checagem em serviços de classificação de domínios.

4.1.1.1 Lista inicial de domínios

Para a constituição do primeiro banco de dados de domínios a abordagem inicial foi encontrar listas de domínios de possíveis provedores de email descartáveis. Foram encontradas algumas listas, porém sem confirmação que tais domínios eram usados por endereços de emails temporários, a checagem final será feita posteriormente.

Outro meio de buscas de informações foi a utilização do Google para encontrar sites que provém endereços de emails temporários e assim adicionar tais endereços manualmente à nossa lista inicial de domínios.

Também foram adicionados à primeira lista domínios publicamente conhecidos como provedores de emails gratuitos como: *gmail.com*, *textithotmail.com* entre outros.

Uma lista de emails de usuários, com aproximadamente 160 mil cadastros, foi cedida pelo Fórum CaveiraTech (JUNQUEIRA, 2019). Todos os domínios utilizados para o cadastro foram extraídos e utilizados para a constituição da lista inicial.

Um total de 3144 domínios foram adicionados à lista listagem inicial de domínios, porém ainda sem classificação.

4.1.1.2 Classificação

Foram classificados como legítimos todos os domínios de emails utilizados por provedores gratuitos publicamente conhecidos como: *gmail.com*, *hotmail.com* entre outros.

Alguns TLDs de uso controlado foram admitidos como seguros, entre eles TLDs de uso governamental, militar educacional como exemplo: *.gov*, *.gov.br*, *.mil*, *.br*. Tais domínios pertencentes a estes TLDs foram classificados como seguros.

Os domínios sem classificação então foram submetidos à uma checagem DNS, que será discutida posteriormente nesse trabalho. Os que foram desaprovados na checagem então naturalmente já não podem ser provedores de email e foram devidamente classificados como DNS falho, porém não descartáveis.

Feito isso, todos os domínios restantes foram submetidos à checagem nos serviços BDEA e KickBox, informando domínios já conhecidos como descartáveis.

Restaram uma lista de 827 domínios, que foram submetidos à análise manual, um a um, e devidamente classificados como legítimos ou descartáveis. Para tal classificação manual eram seguidos os seguintes passos:

1. Abrir o domínio no navegador;
2. Caso seja redirecionado a um site de emails descartáveis, é marcado como descartável;

3. Caso seja redirecionado a um site legítimo é marcado como não descartável;
4. Caso seja redirecionado a um site desconfigurado ou a site nenhum, é feita a busca pelo email em motores de busca;
5. Se na busca o domínio foi relacionado à emails descartáveis ou é referenciado em algum provedor de emails descartáveis, é marcado como descartável;
6. Caso não for possível encontrar informações sobre o domínio, é marcado como legítimo.

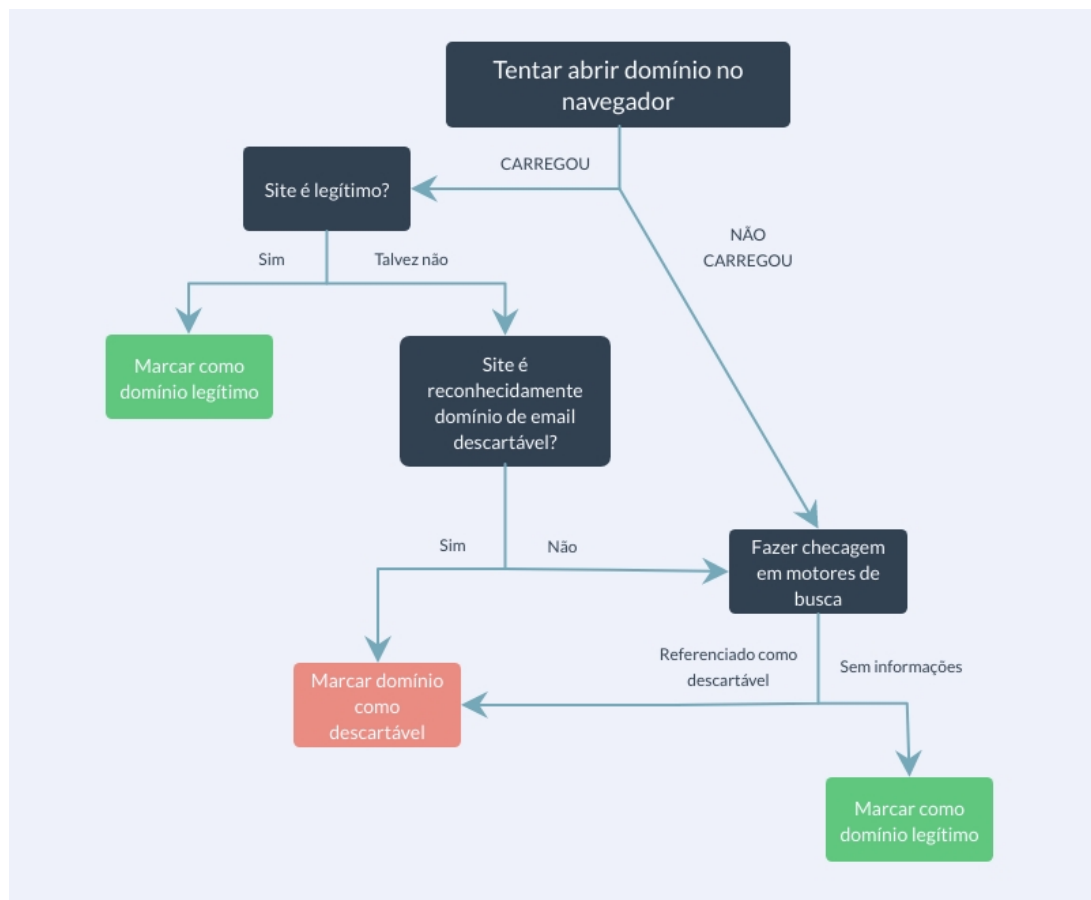


Figura 3 – Fluxograma ilustrando a classificação manual de domínio

Todos os 3144 domínios foram classificados:

Tabela 3 – Número de domínios classificados

DNS Inválido	Legítimo	Descartável	Total
656	1487	1001	3144

Dos 3144 domínios analisados, 1001 foram detectados como sendo de provedores de emails descartáveis, outros 656 foram detectados como domínio que não possuem uma configuração DNS válida e restaram 1487 domínio detectados como provedores de email legítimos.

4.1.2 Identificação entradas DNS válidas

A identificação de entradas de DNS válidas é feita através de chamadas à tabela DNS em busca de entradas do tipo MX. Tal chamada pode ser feita através do terminal com:

```
host -t MX <dominio>
```

Caso haja uma resposta é porque o DNS possui uma configuração adequada e o domínio está apto a receber emails.

4.2 Sistema

Nesta seção será apresentada a construção, uso e administração do sistema proposto.

Como citado anteriormente, foi utilizado o *web framework* Django para a construção do sistema, tratando-se então de um sistema *web*, o acesso pode ser feito apenas utilizando um navegador e o mesmo rodará em nuvem, através do AWS Lambda, utilizando-se das vantagens que a tecnologia *serverless* tem a oferecer.

4.2.1 Modelagem

O primeiro passo para o desenvolvimento é a modelagem dos dados e entidades a serem utilizados pelo *software*.

Foram estabelecidas três entidades que se correlacionam, dando origem a três modelos no *framework* Django:

- Usuário: responsável por guardar os detalhes do usuário, como nome de usuário, email, nome, senha e chave de API;
- Domínio: responsável por guardar domínios e sua classificação;
- Checagem: responsável por correlacionar uma checagem de endereço de email a um domínio e um usuário.

O Django por padrão já oferece um modelo padrão de usuário, porém o mesmo foi estendido para a criação de uma nova propriedade necessária: chave de API, o novo modelo foi chamado de *Profile*. Já os modelos Domínio e Checagem foram chamados respectivamente de *Domain* e *Check*.

O *model Domain* também conta com um campo "Novo", que indica se o domínio foi adicionado recentemente ao banco de dados e ainda não passou por uma verificação manual.

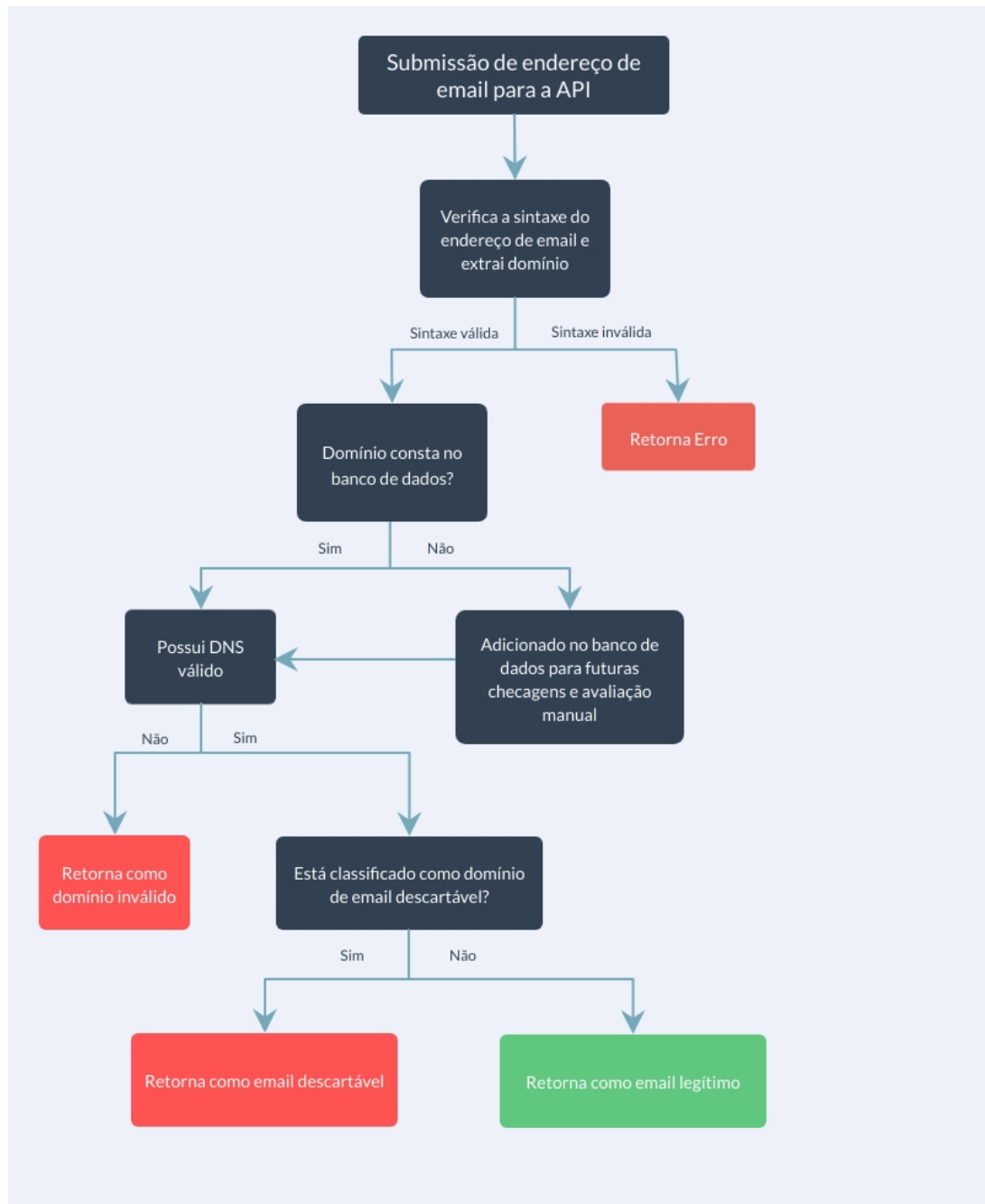


Figura 4 – Fluxograma ilustrando o funcionamento básico da API

4.2.2 Interface

Foi criada uma interface *web* para a utilização do sistema, no domínio principal foi colocado um *hotsite* para a captação de novos usuário para o sistema.

Ao se registrar ou fazer *login* o usuário é redirecionado ao painel de controle, onde pode fazer o controle do seu uso do sistema. O sistema é composto por 3 páginas principais:

- *Dashboard*: mostra os detalhes da conta com algumas estatísticas resumidas;
- *Requets*: mostra em detalhes a utilização e todas as requisições feitas à API;

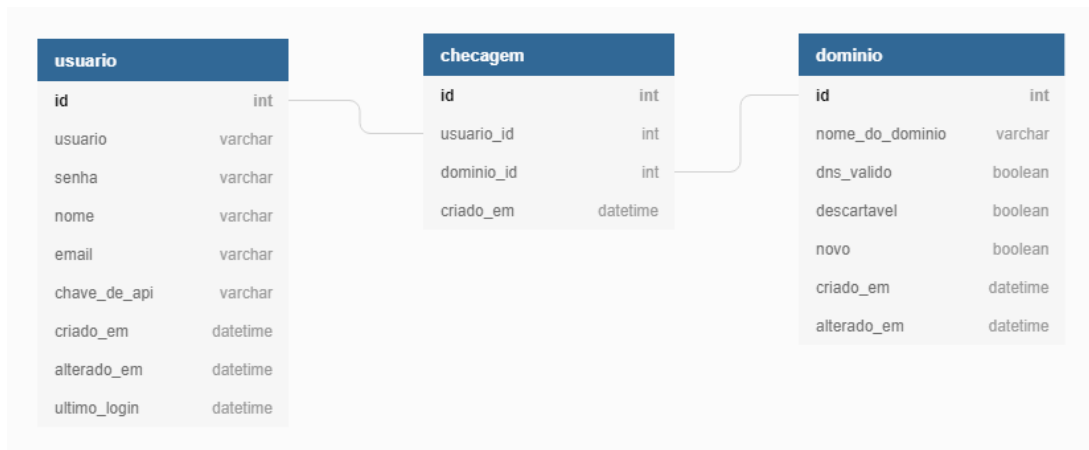
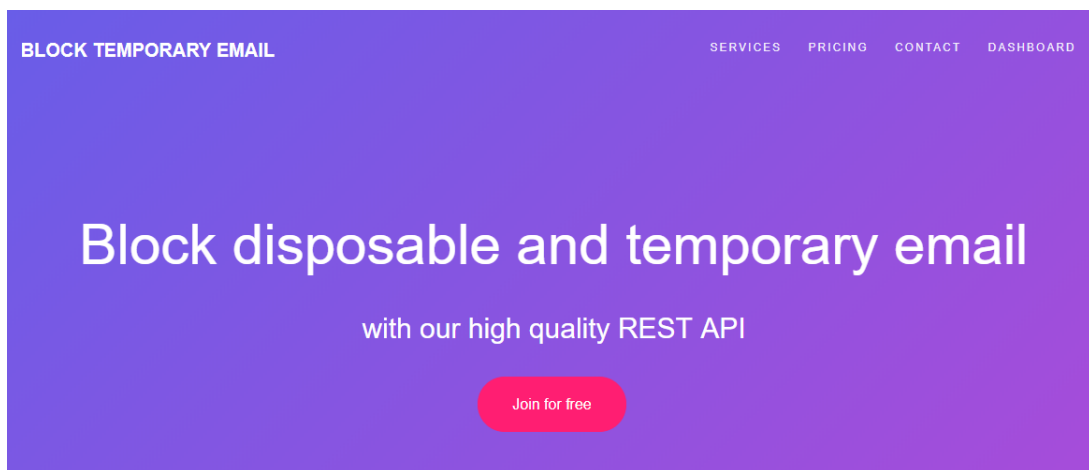


Figura 5 – Diagrama de entidades relacionais (ERD) do sistema

Figura 6 – Imagem do *hotsite*

- *Documentation*: a documentação de como o sistema funciona e como se integrar à API.

4.2.3 API

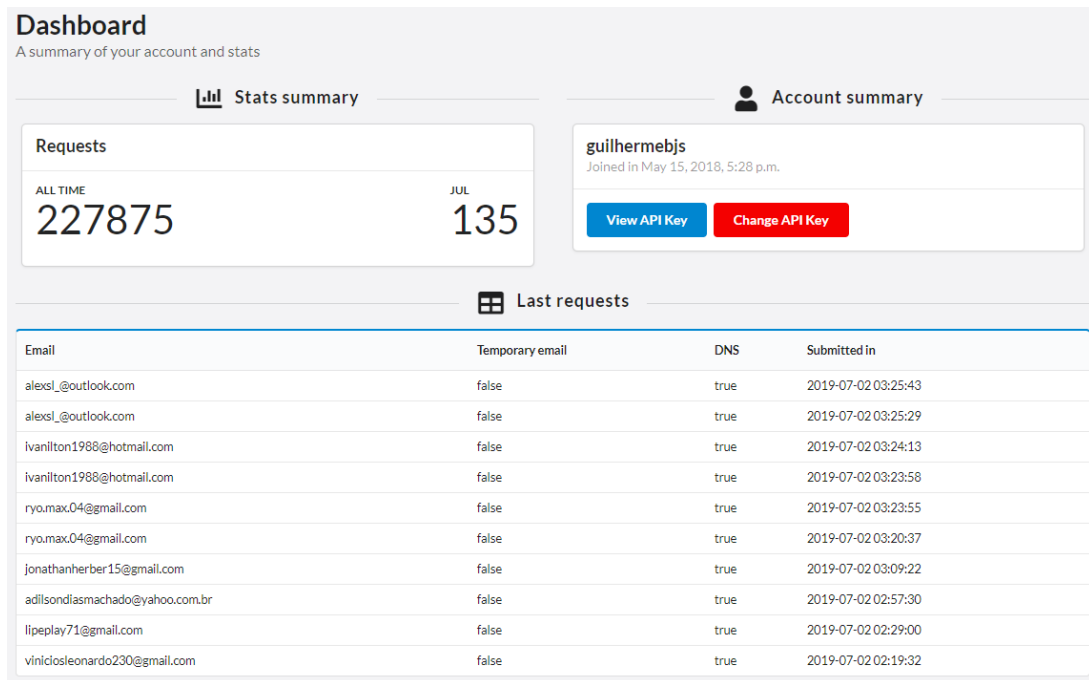
A API é o núcleo do sistema. É ela quem permite a integração do mesmo a outros sistemas.

O padrão utilizado foi o REST através de chamadas *web* que trafegam JSON.

A API foi inicialmente versionada como Versão 1, ou seja, a partir do lançamento em produção tal API não pode mais ser modificada ao ponto de causar incompatibilidade com as integrações já existentes, caso precise ser feita uma grande mudança.

A versão 1 possui apenas um *endpoint* que executa a checagem do domínio:

GET <https://block-temporary-email.com/api/v1/check>

Figura 7 – Imagem da página *Dashboard*

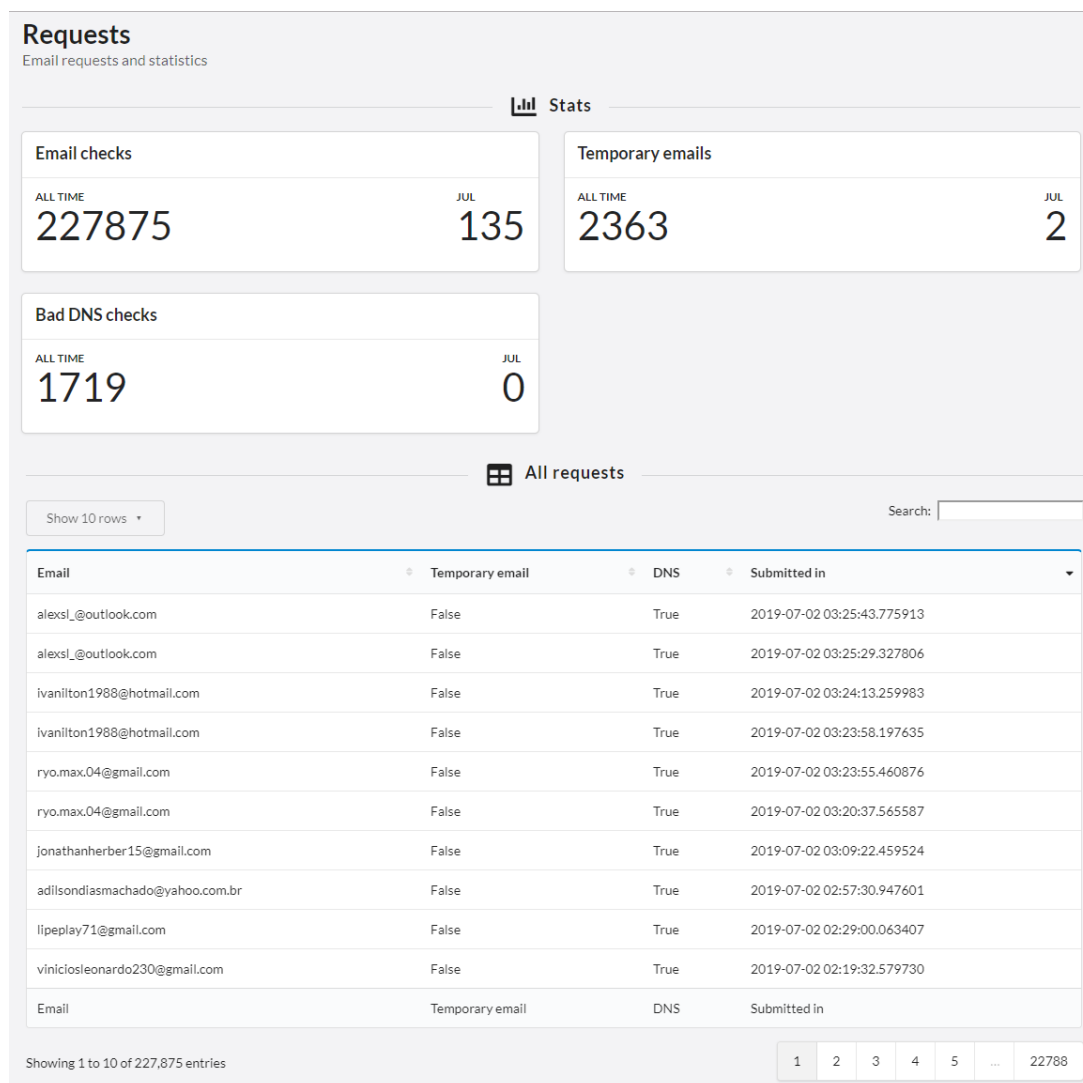
Deverá ser feita uma requisição HTTP do tipo GET à este *endpoint* e deverão ser passadas as seguintes variáveis na URL:

- *username*: nome de usuário;
- *api_key*: chave de API do usuário;
- *email*: endereço de email a ser checado.

A requisição então será respondida com a classificação do domínio:

- *success* (booleano): informa se a requisição foi feita corretamente sem nenhum erro;
- *error* (*string*): informa o erro, caso houver;
- *domain* (*string*): informa o domínio extraído do endereço de email;
- *email* (*string*): informa o email passado para checagem;
- *syntax* (booleano): informa se a sintaxe do email está correta;
- *temporary* (booleano): informa se se trata de um endereço de email descartável;
- *dns* (booleano): informa se o domínio possui entradas DNS válidas para o recebimento de emails;

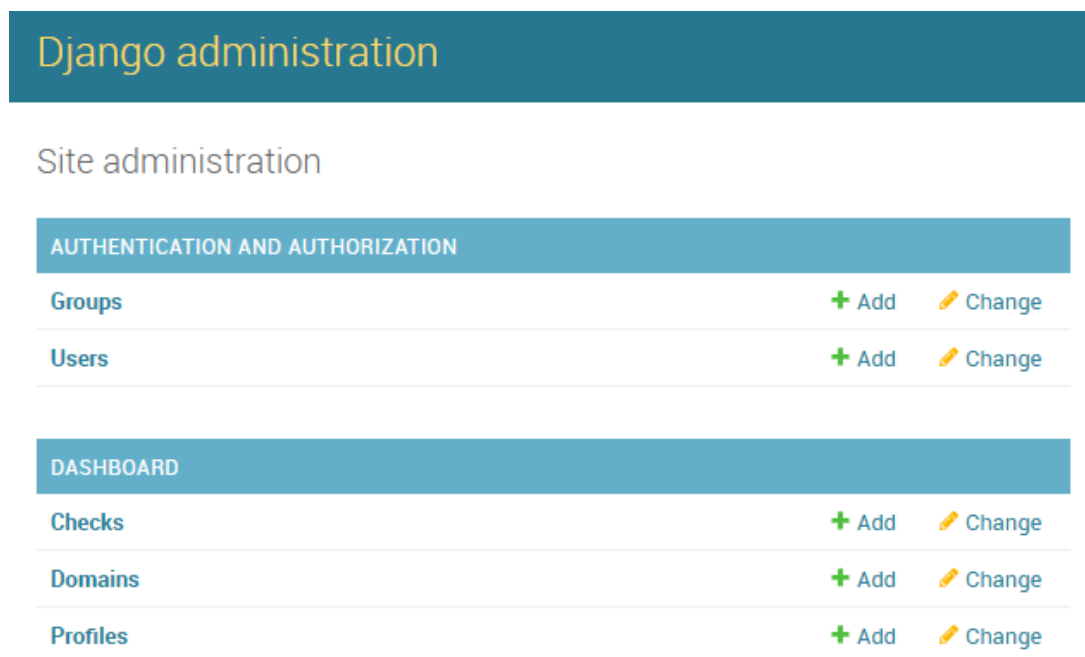
A partir da resposta, em JSON, um sistema acoplado ao *Block Temporary Email* poderá então efetuar ações de acordo com suas necessidades.

Figura 8 – Imagem da página *Requests*

4.2.4 Administração

Para a continuidade da eficácia do sistema em classificar domínios corretamente, é preciso que o mesmo esteja em constante evolução. Para isso, torna-se necessária uma boa interface de administração. O Django oferece por padrão o *Django administration*, que trata-se de um painel *web* gerado automaticamente para o gerenciamento de conteúdo dos *models* declarados no sistema. Com o mesmo é possível adicionar, excluir e editar entradas do banco de dados.

Com o painel de administração é possível checar as últimas adições de domínios e fazer edições manuais ao banco de dados a partir das análises manuais do domínio. Também é possível adicionar filtros aos modelos e fazer levantamentos estatísticos.

Figura 9 – Imagem da página *Django administration*

4.3 Análise

Nesta seção serão apresentados os resultados encontrados a partir da avaliação de milhares de requisições feitas ao sistema *Block Temporary Email* e comparações com outros serviços análogos.

4.3.1 Amostra

Para a análise de eficácia da classificação de emails foi retirada uma amostra de dez mil endereços de email escolhida aleatoriamente de uma população de 160 mil endereços de email de usuários reais, cedidos pelo fórum CaveiraTech.

Os dez mil endereços foram submetidos à uma checagem nos seguintes serviços de classificação de emails temporários:

- Block Temporary Email (construído a partir deste trabalho);
- Open KickBox;
- Block Disposable Email Address;

Foram então colhidas as seguintes informações:

- Quantidade de endereços de email identificados como temporários ou descartáveis;
- Quantidade de endereços de email com DNS inválidos;

- Tempo de resposta médio da requisição;
- Quantidade de falhas em requisições.

Todos os testes foram feitos utilizando um *script* em que cada serviço foi testado em condições iguais, utilizando o mesmo poder de processamento e mesma conexão com a Internet. Cada email na lista foi testado através de uma única *thread*, não havendo paralelismo e nem requisições simultâneas.

O tempo médio de resposta é a média entre a soma do tempo total de cada uma das requisições dividido pelo número total de requisições, que neste caso é de dez mil.

4.3.2 Resultados

Após a análise da amostra de 10 mil endereços de emails foram obtidos os seguintes resultados:

Tabela 4 – Serviços de classificação de email

-	BTE	KickBox	BDEA
Descartáveis	267	233	150
DNS Inválidos	85	0	0
Falhas	0	0	531
Tempo médio de resposta (ms)	905	636	2732

Pela tabela de resultados nota-se um grande número de falhas e um alto tempo médio de resposta do serviço *Block Disposable Email Address* que indica problemas com o mesmo quando utilizado em produção.

O serviço KickBox possui um ótimo tempo médio de resposta e não foi encontrada nenhuma falha. O número de emails descartáveis identificados também foi satisfatório, porém o mesmo não faz uma análise de DNS, fazendo com que o sejam aceitos endereços de email inexistentes e disfuncionais.

O serviço Block Temporary Email, fruto deste trabalho, obteve entre os três, a maior taxa de detecção de endereços de email temporários, ao mesmo tempo que também conseguiu identificar 85 domínios com DNS inválidos. Não obteve nenhuma falha de requisição e teve um tempo de resposta satisfatório, com 905 ms, porém 42% mais lento que o serviço KickBox.

5 Conclusão

O objetivo do trabalho, construir e avaliar um sistema capaz de classificar endereços de emails descartáveis e inválidos, foi concluído com sucesso.

Um sistema estável e a nível de produção foi produzido e conseguiu melhores resultados que duas alternativas presentes no mercado.

O maior desafio, a classificação manual de centenas de domínios, foi transposto e agora a manutenção de um banco de dados atualizado pode ser alcançada de maneira sustentável através de poucas classificações manuais semanais.

A taxa de endereços de email bloqueados foi de 352 em uma amostra de dez mil, o que representa 3,52%. Isso fará com que os serviços que utilizarem tal sistema para validação contarão com uma elevada taxa de sucesso de envio de seus emails transacionais e de marketing, diminuindo *bounces* e cortando custos de envio.

Uma possível evolução deste trabalho seria aplicar técnicas de *machine learning* para a identificação de domínios e provedores de emails descartáveis, para que não seja mais necessária a verificação manual. Poderia ser criada uma rede neural através dos dados e amostras já obtidos, tentando assim a identificação de padrões característicos em endereços de emails descartáveis e a possível previsão da classificação em um domínio desconhecido.

Outro grande desafio seria diminuir o tempo de resposta da API, isso pode ser alcançado através da utilização de um banco de dados NoSQL orientado a documentos como o MongoDB ou o DynamoDB, este último oferecido pela AWS como um serviço *serverless* totalmente gerenciado e escalável. Outra abordagem seria a diminuição da complexidade da aplicação, removendo os *models* de usuários e checagem e removendo a necessidade de autenticação, deixando assim de armazenar os emails checados, com isso assim seriam removidas quase que na totalidade as operações de escrita no banco de dados (as mais custosas) e removidas as operações de leitura para se fazer a autenticação. O uso de um *framework* mais leve e simples como o Flask também poderia ser um fator para diminuir o tempo de resposta. Outro fator que pode melhorar o tempo de resposta significativamente é o *deploy* global da API, ou seja, em múltiplos *datacenters* da AWS ao redor do planeta, fazendo com que os tempos de respostas sejam parecidos independentemente do local de origem das requisições.

Referências

- 10 Minute Mail. *About 10 Minute Mail*. 2017. Disponível em: <https://10minutemail.com/10MinuteMail/about.htm>. Citado na página 16.
- ALLMAN, E. et al. *DomainKeys Identified Mail (DKIM) Signatures*. [S.l.], 2007. Disponível em: <https://www.rfc-editor.org/info/rfc4871>. Citado 2 vezes nas páginas 9 e 15.
- ANDROUTSOPOULOS, I. et al. Learning to Filter Spam E-Mail: A Comparison of a Naive Bayesian and a Memory-Based Approach. *Proceedings of the workshop "Machine Learning and Textual Information Access"*, n. September 2000, p. 1–12, 2000. Disponível em: <http://arxiv.org/abs/cs/0009009>. Citado 2 vezes nas páginas 9 e 16.
- BDEA. *About BDEA*. 2017. Disponível em: <https://www.block-disposable-email.com/cms/about/presentation/>. Citado na página 21.
- CHEANG, H. *What is Everything-as-a-Service (XaaS)?* 2014. Disponível em: <http://blog.cimpl.com/what-is-everything-as-a-service-xaas>. Citado na página 17.
- COOK, D. et al. Catching Spam Before it Arrives: Domain Specific Dynamic Blacklists BT - Fourth Australasian Information Security Workshop (Network Security) (AISW 2006). In: *Conferences in Research and Practice in Information Technology Series*. [s.n.], 2006. v. 54, p. 193–202. ISBN 1920682368. ISSN 14451336. Disponível em: <http://crpit.com/confpapers/CRPITV54Cook.pdf>. Citado 2 vezes nas páginas 9 e 15.
- CRISPIN, M. *INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1*. [S.l.], 2003. Disponível em: <https://www.rfc-editor.org/info/rfc3501>. Citado na página 13.
- ICANN. *About ICANN*. 2012. Disponível em: <https://www.icann.org/resources/pages/welcome-2012-02-25-en>. Citado na página 14.
- Internet Engineering Task Force. *RFC 821: Simple Mail Transfer Protocol*. 1982. Disponível em: <http://tools.ietf.org/html/rfc821>. Citado 2 vezes nas páginas 9 e 12.
- Internet Engineering Task Force. *RFC 821: Simple Mail Transfer Protocol*. 1982. Disponível em: <http://tools.ietf.org/html/rfc821https://www.rfc-editor.org/info/rfc2635>. Citado 2 vezes nas páginas 9 e 15.
- JUNQUEIRA, G. *Fórum CaveiraTech*. 2019. Disponível em: <https://caveiratech.com/forum>. Citado na página 24.
- KickBox. *KickBox*. 2019. Disponível em: <https://open.kickbox.io/>. Citado na página 21.
- Klensin J. RFC 7504 - SMTP 521 and 556 Reply Codes. 2015. ISSN 2070-1721. Disponível em: <http://www.rfc-editor.org/info/rfc7504>. Citado 2 vezes nas páginas 9 e 12.

MOCKAPETRIS, P. *Domain names - concepts and facilities*. [S.l.], 1987. 1–55 p. Disponível em: <<https://www.rfc-editor.org/info/rfc1034>>. Citado na página 14.

MYERS, J.; ROSE, M. RFC 1939 - Post Office Protocol - Version 3. *IETF Request For Comments*, p. 1–23, 1996. Citado na página 13.

Oasis Open. *How Email Really Works*. 2017. Disponível em: <https://www.oasis-open.org/khelp/kmlm/user_help/html/how_email_works.html>. Citado 2 vezes nas páginas 4 e 13.

SEIGNEUR, J. M.; JENSEN, C. D. Privacy recovery with disposable Email addresses. *IEEE Security and Privacy*, v. 1, n. 6, p. 35–39, 2003. ISSN 15407993. Citado 3 vezes nas páginas 9, 16 e 21.

Solyd Treinamentos. *Solyd Treinamentos*. 2019. 1–19 p. Disponível em: <<https://solyd.com.br/treinamentos>>. Citado na página 10.

TURNER, M.; BUDGEN, D.; BRERETON, P. Turning software into a service. *Computer*, v. 36, n. 10, p. 38–44, 2003. ISSN 00189162. Citado na página 17.

Apêndices

APÊNDICE A – Documentação completa

Block Temporary Email

O sistema Block Temporary Email pode ser acessado através do endereço: <https://block-temporary-email.com>, o mesmo está em inglês, por isso a documentação anexa encontrada no site e anexada a este trabalho também está em inglês.

A.1 Getting started

Block Temporary Email's API exposes our infrastructure via a standardized programmatic interface. Block Temporary Email's API, you can automatize the process of blocking or categorizing email address in your application.

The Block Temporary Email's API is a RESTful API based on HTTPS requests and JSON responses. If you are registered with Block Temporary Email, you can obtain your API key from the bottom of the "Dashboard" page.

A.2 What is Block Temporary Email

Block Temporary Email is a cloud-based service that helps you to classify and block unwanted emails address in your service. Like temporary/disposable email address.

We can make it by consulting our huge database with thousands of domains that are classified by reputation. We check basically three items:

- Email address syntax;
- MX Records of domain's email address;
- Disposable email address.

A.3 Do's and Don'ts

What can you build with Block Temporary Email APIs?
Anything that's useful and follows the guidelines presented here.

What should you avoid doing with Block Temporary Email APIs?
Do not do any of the following:

- Abuse Block Temporary Email systems or customers;
- Misrepresent Block Temporary Email services as your own.

Abuse: Follow all guidelines, including the rate limits defined below. Your ability to use the Block Temporary Email APIs may be terminated, temporarily or permanently, if our systems are abused. Similarly, anything in an application which goes against the goal of making Block Temporary Email more useful to Block Temporary Email customers or attempts to mistreat customers or their data will be grounds for termination.

Misrepresentation: Draw a clear line between the benefits you provide in your application and those benefits of the Block Temporary Email service that you enable via API. The APIs are not intended for "white labeling" or reselling Block Temporary Email services as your own. Nothing in your service or application should create a false sense of endorsement, sponsorship, or association with Block Temporary Email. You may sell your own application or service which utilizes the Block Temporary Email APIs, but may not sell Block Temporary Email services to customers without a commercial agreement with Block Temporary Email.

A.4 Endpoints

The API is accessed by making HTTPS requests to a specific version endpoint URL. Every endpoint is accessed only via the SSL-enabled HTTPS (port 443) protocol.

Everything (methods, parameters, etc.) is fixed to a version number, and every call must contain one. The latest version is Version 1.

The stable base URL for all Version 1 HTTPS endpoints is: <https://block-temporary-email.com/api/v1/>

A.5 Requests and Resposes

Requests must be sent over HTTPS. All requests need to send username and api_key as parameters.

Each response is a JSON object. The data requested is wrapped in the tag. If you have a response, it will always be within the field.

A.6 All API's actions

A.6.1 Check

GET <https://block-temporary-email.com/api/v1/check>

This action is used to classify an email address. Is used a GET request.

Request parameters:

- username: Your username (where to find my username?);
- api_key: Your API Key (where to find my api key?);
- email: An email address to check if temporary/disposable.

Example:

GET <https://block-temporary-email.com/api/v1/check>

?username= your_username&api_key=your_api_key&email=bob@example.com

Response:

- success (boolean): Return TRUE if the requisition occurs with no errors;
- error (string): Return an error description if success is FALSE;
- domain (string): Return the email address' domain of the requisition;
- email (string): Return the email address of the requisition;
- syntax (boolean): Return TRUE if the email address syntax is correct;
- temporary (boolean): Return TRUE if the email address is temporary or disposable;
- dns (boolean): Return TRUE if the email address' domain has a valid MX entry in DNS;

Errors:

- Invalid arguments: Lack of one or more required arguments;
- Not authorized: Your Username or API Key are invalids;
- Insufficient funds: Lack of funds in your account.

A.7 Rate limiting

The Block Temporary Email API sets no limit for now. All our services are 100% free of charge, but we reserve the right to create plans and limits in the future.

A.8 How to get Username and API Key

Your username and API key are available on your Dashboard. You can query or change the API key at any time, but remember that when you change the API key you will invalidate the previously used key.

A.9 Examples

A.9.1 Python

```
import requests
import json

# Define API URL and Data
url = 'https://block-temporary-email.com/api/v1/check'
your_params = {
    'username': 'yourusername',
    'api_key': 'yourapikey',
    'email': 'bob@example.com'
}

req = requests.get(url, params=your_params) # Make the requisition
req_json = json.loads(req.text) # Transform the json answer in a Dict object

print(req_json) # Print the Dict object
```

APÊNDICE B – Lista de domínios identificados como provedores de email descartáveis

Lista com 1303 domínios identificados como provedores de email temporários/descartáveis extraídas do sistema Block Temporary Email em 9 de setembro de 2019 organizados por ordem alfabética.

0-mail.com	2.tebwinsoi.ooo
0815.ru	20boxme.org
0815.su	20email.eu
0clickemail.com	20mail.eu
0mail.org	20mail.it
0mixmail.info	20minute.email
0wnd.net	20minutemail.com
0wnd.org	20mm.eu
10g.pl	2anom.com
10host.top	2ether.net
10mail.org	2odem.com
10minutemail.co.za	2prong.com
10minutemail.com	3.emailfake.ml
10minutemail.davidxia.com	30minutemail.com
10minutemail.net	30wave.com
10minutesmail.com	33mail.com
10vpn.info	3d-painting.com
12hosting.net	3ntongm4il.ga
12storage.com	4.emailfreedom.ml
163.com	4.kadokawa.top
1dmedical.com	4easyemail.com
1mail.x24hr.com	4simpleemail.com
1qpatglchm1.ga	4warding.com
1rentcar.top	4warding.net
1shivom.com	4warding.org
1webmail.info	55hosting.net
2.sexymail.ooo	5music.info

5music.top	alarmists.com
6.emailfake.ml	alienware13.com
60-minuten-mail.de	alivance.com
60minutemail.com	allfamus.com
675hosting.com	alltempmail.com
675hosting.net	amail.club
675hosting.org	amilegit.com
69postix.info	amiri.net
6en9mail2.ga	amiriindustries.com
6monthsbr.33mail.com	anappfor.com
6paq.com	anappthat.com
6url.com	anonbox.net
75hosting.com	anonymbox.com
75hosting.net	antichief.com
75hosting.org	antichief.net
7rent.top	antispam.de
7tags.com	apkmd.com
88clean.pro	appixie.com
8wkkrizxpphbm3c.ml	armymail.com
99pubblicita.com	armyspy.com
9me.site	arsed.com
9ox.net	arur01.tk
9y.com	arurimport.ml
a-bc.net	asdasd.ru
aa5zy64.com	asorent.com
aaaf.ru	ass.pp.ua
abyssmail.com	astromail.com
acentri.com	attnetwork.com
add3000.pp.ua	averdov.com
adiazq.com	awsoo.com
aditus.info	ax80mail.com
advantimo.com	axon7zte.com
aegde.com	axsup.net
afrobacon.com	b2bx.net
air2token.com	b5safaria.com
airmail.cc	balanc3r.com
ajaxapp.net	banit.club
akgq701.com	bareed.ws
akkadian.com	barefooted.com

baxomale.ht.cx	brutes.com
bcaoo.com	bsnow.net
beddly.com	btcmil.pw
beefmilk.com	budaya-tionghoa.com
beevee.gq	budayationghoa.com
bellybuster.com	bugmenot.com
bemused.com	bumpymail.com
bessa.esy.es	bungee.net
bestlucky.pw	burjnet.ru
bestvpn.top	burstmail.info
bigisbeautiful.com	buttonpushers.com
bigisbetter.com	buy003.com
bigprofessor.so	byom.de
bigsecret.com	c.andreihusanu.ro
binka.me	cloramn.com
binkmail.com	cano.cl
binnary.com	car101.pro
bio-muesli.net	carbtc.net
bissabiss.com	carins.io
bit-degree.com	cars2.club
bitwhites.top	cartelera.org
bladesmail.net	casualdx.com
blag.com	cbair.com
blahdeblah.com	cdcmail.date
blogrtui.ru	cdnqa.com
blowitup.com	ce.mintemail.com
boardmaster.com	centermail.com
bobmail.info	centermail.net
bocps.biz	chalky.com
bodhi.lawlita.com	chammy.info
bofthew.com	changeplace.com
bootkp8fnp6t7dh.cf	charlies.com
boster.com	chasing.com
box2temp.com.br	cheaphub.net
boximail.com	cherrycola.com
boxtemp.com.br	chewies.com
boxtwos.com	chocolatejunkies.com
brefmail.com	chogmail.com
broadbandninja.com	choicemail1.com

cialiscouponss.com	cutout.club
cialisy.info	cuvox.de
cjpeg.com	d.pics.mu
ckiso.com	d1yun.com
ckoie.com	dacoollest.com
clay.xyz	daily-email.com
click-email.com	daintly.com
cloud99.pro	dandikmail.com
cloud99.top	darkcorner.com
clrmail.com	darkplace.com
clubfever.com	datum2.com
cmail.club	daydreamer.com
cnm.coms.hk	dayrep.com
cobin2hood.com	dc-business.com
cock.li	deadaddress.com
cocovpn.com	deadspam.com
coieo.com	deepdesire.com
coldemail.info	desilver.com
cool.fr.nf	desoz.com
coreclip.com	despam.it
correo.blogos.net	despammed.com
cosmorph.com	devnullmail.com
courriel.fr.nf	deyom.com
courrieltemporaire.com	dfgh.net
crankmails.com	dfremails.com
creazionisa.com	dianhabis.ml
crotslep.ml	digitalsanctuary.com
crowd-mail.com	digow.33mail.com
crumbly.com	disaq.com
crusthost.com	disbox.net
crymail2.com	disbox.org
cryp.email	discardmail.com
crypemail.info	discardmail.de
cs5xugkcirf07jk.ga	discos4.com
cubiclink.com	disposableaddress.com
currymonster.com	disposeamail.com
curryworld.de	disposemail.com
cust.in	dispostable.com
cutemail.com	divermail.com

dividendxk.com	e.xxi2.com
divismail.ru	e4ward.com
diwaq.com	e4wfnv7ay0hawl3rz.cf
djsuperstars.com	eb609s25w.com
dlemail.ru	ebtukukxnn.cf
dm.w3internet.co.ukexample.com	edutom.info
dndent.com	eelmail.com
dnsdeer.com	efremails.com
dobramama.pl	einrot.com
dodgeit.com	elitists.com
dodgemail.de	email-lab.com
dodgit.com	email60.com
dodgit.org	emailapps.in
doiea.com	emaildienst.de
doleoffice.com	emailias.com
donemail.ru	emailigo.de
dontreg.com	emailinfive.com
dontsendmespam.de	emailko.in
dotters.com	emailmenow.info
downboy.com	emailmiser.com
dr69.site	emailna.co
drdrb.com	emailna.life
drdrb.net	emailo.pro
drisd.com	emailportal.info
drivetagdev.com	emailrtg.org
dropjar.com	emailsecurer.com
droplar.com	emailsensei.com
dropmail.me	emailszy.info
dsiaay.com	emailtemporario.com.br
duck2.club	emailtex.com
ducktail.com	emailto.de
dumoac.net	emailure.net
dump-email.info	emailwarden.com
dumpandjunk.com	emailx.at.hm
dumpmail.de	emailxfer.com
dummyemail.com	emergencymail.com
dwse.edu.pl	eml.pp.ua
dx.ez.lv	emlhub.com
e.wupics.com	emlpro.com

emltmp.com	fastmazda.com
emz.net	fastmitsubishi.com
endrix.org	fastnissan.com
enterto.com	fastsubaru.com
envy17.com	fastsuzuki.com
ephemail.net	fasttoyota.com
est.une.victime.ninja	fastyamaha.com
eth2btc.info	figjs.com
ether123.net	filzmail.com
ethereum1.top	findemail.info
ethersports.org	firelord.com
ethersportz.info	firemail.cc
etlgr.com	fizmail.com
etoic.com	fleckens.hu
etranquil.com	flemail.ru
etranquil.net	flurred.com
etranquil.org	foxja.com
euaqa.com	fr33mail.info
europe.com	frapmail.com
evopo.com	freeletter.me
evyush.com	freemail.tweakly.net
ew.eu	freemymail.org
exceptmail.com	freemails.com
explodemail.com	freundin.ru
extremail.ru	from.onmypc.info
eyeemail.com	front14.org
fairyqueen.com	fulvie.com
fake-email.pp.ua	fun64.com
fakeinbox.com	furusato.tokyo
fakeinformation.com	fux0ringduh.com
fammix.com	fw6m0bd.com
fantasyforce.com	fxprix.com
fast-email.info	fynns.com
fastacura.com	g14l71lb.com
fastbowler.com	gafy.net
fastchevy.com	galaxyarmy.tech
fastchrysler.com	gameaddict.com
fastkawasaki.com	gamgling.com
fastmailforyou.net	garage46.com

garliclife.com	great-host.in
gmail.top	greensloth.com
geekforex.com	grr.la
geronra.com	gsaseoemail.com
get.pp.ua	gsrv.co.uk
get1mail.com	guarchibao-fest.ru
get2mail.fr	guccibagshare.com
getairmail.com	guerillamail.biz
getapet.net	guerillamail.com
getnada.com	guerillamail.net
getonemail.com	guerillamail.org
getonemail.net	guerrillamail.biz
getsimpleemail.com	guerrillamail.com
ghosttexter.de	guerrillamail.de
giftto12.com	guerrillamail.info
girlsundertheinfluence.com	guerrillamail.net
gishpuppy.com	guerrillamail.org
givmail.com	guerrillamailblock.com
giyam.com	gustr.com
gmailbox.org	h.mintemail.com
gmailertyq.com	h8s.org
gmatch.org	haltospam.com
gmex.com	happymail.guru
go.jwork.ru	haqed.com
go2usa.info	haribu.com
go2vpn.net	haribu.net
gobby.com	hartbot.de
golemico.com	hatespam.org
gowikibooks.com	hawkdown77.e4ward.com
gowikicampus.com	haydoo.com
gowikicars.com	hi2.in
gowikifilms.com	hidemail.de
gowikigames.com	himail.online
gowikimusic.com	hitbts.com
gowikinetwork.com	hitler.rocks
gowikitavel.com	hmamail.com
gowikitv.com	hochsitze.com
grandmamail.com	honor-8.com
grandmasmail.com	hostcalls.com

hostlaba.com	iwi.net
hothatches.com	jackmailer.com
hotmailproduct.com	jajxz.com
hotpop.com	janproz.com
housat.com	jetable.com
hu4ht.com	jetable.fr.nf
hubii-network.com	jetable.net
hulapla.de	jetable.org
hurify1.com	jetable.pp.ua
i6.cloudns.cc	jmail.ovh
i6.cloudns.cx	jnxjn.com
iaoss.com	jokenaka.press
ieatspam.eu	jourrapide.com
ieatspam.info	jumonji.tk
ihateyoualot.info	junk1e.com
iheartspam.org	just-email.com
il3nexus.esy.es	kasmail.com
imails.info	kaspop.com
imgof.com	katztube.com
inaby.com	keepmymail.com
inboxalias.com	kekita.com
inboxbear.com	kickedout.com
inboxclean.com	killmail.com
inboxclean.org	killmail.net
inboxdesign.me	kiois.com
inboxstore.me	kir.ch.tc
incognitomail.com	kismail.ru
incognitomail.net	klassmaster.com
incognitomail.org	klassmaster.net
insorg-mail.info	klipschx12.com
intempmail.com	kloap.com
intolerant.trillianpro.com	klzlk.com
iozak.com	kmhow.com
ip4.pp.ua	knol-power.nl
ip6.pp.ua	konno.tk
ipdeer.com	kopqi.com
ipoo.org	kosiarszkont.com
irish2me.com	koszmail.pl
itis0k.org	kpoa.com

krgyui7svgomjhso.cf	lopl.co.cc
kulturbetrieb.info	lortemail.dk
kumail8.info	lovenest.co.uk
kurzepost.de	lovesecrets.com
kyois.com	lr78.com
l0real.net	luckjob.pw
lackmail.net	luckster.com
lackmail.ru	lucys.com
lajoska.pe.hu	lucyu.com
lakqs.com	lzfkvktj5arne.cf
landmail.co	lzoaq.com
laoeq.com	m4ilweb.info
laoho.com	maboard.com
larisia.com	macr2.com
larjem.com	madder.com
lastmail.co	mail-on.us
ldaholce.com	mail-temporaire.fr
leeching.net	mail.by
lek nawypadaniewlosow.pl	mail.mezimages.net
lemonmail.com	mail.oyu.kr
lenovog4.com	mail.ru
letsmail9.com	mail.ticket-please.ga
letthemeatspam.com	mail1.ftp1.biz
lgxscreen.com	mail2rss.org
lhsdv.com	mail333.com
lifebyfood.com	mail4-us.org
lillemap.net	mail4trash.com
lilylee.com	mail707.com
link2mail.net	mailbidon.com
liquidlunch.com	mailblocks.com
litedrop.com	mailblog.biz
loan101.pro	mailbox52.ga
loaoa.com	mailbox72.com
loapq.com	mailbox80.biz
loh.pp.ua	mailcat.biz
loketa.com	mailcatch.com
lol.ovpn.to	mailconn.com
lolekemail.net	maildrop.cc
lookugly.com	maildx.com

maileater.com	mailtemp.info
mailed.ro	mailtothis.com
maileme101.com	mailtrix.net
mailexpire.com	mailxcdn.com
mailf5.com	mailzi.ru
mailfnmng.org	mailzilla.com
mailforspam.com	mailzilla.org
mailfreeonline.com	makethebreak.com
mailfs.com	mallinco.com
mailgov.info	manmachine.com
mailhazard.com	matchpol.net
mailimate.com	max88.club
mailin8r.com	mbx.cc
mailinater.com	mcdonald.cf
mailinator.com	mciek.com
mailinator.net	md5hashing.net
mailinator2.com	mega.zik.dj
mailincubator.com	megsars.ru
mailismagic.com	meinspamschutz.de
mailme.ir	meltmail.com
mailme.lv	messagebeamer.de
mailmetal.com	miauj.com
mailmetrash.com	micsocks.net
mailmoat.com	mierdamail.com
mailna.biz	mihep.com
mailna.co	milk.gage.ga
mailna.in	minex-coin.com
mailna.me	mintemail.com
mailnator.com	misssporty.com
mailnesia.com	mistersporty.com
mailnull.com	ml00.gmailmirror.com
mailpick.biz	ml01.gmailmirror.com
mailpick.info	moakt.co
mailplus.pl	moakt.com
mailrock.biz	moakt.ws
mailshell.com	mobilevpn.top
mailsiphon.com	moburl.com
mailslite.com	mohmal.com
mailtechx.com	mohmal.im

mohmal.in	nagging.com
mohmal.tech	naseem.com
mojxpoczta.poczta.lolekemail.net	neomailbox.com
molms.com	nepwk.com
moncourrier.fr.nf	nervmich.net
monemail.fr.nf	nervtmich.net
monmail.fr.nf	netmails.com
morfeu.esy.es	netmails.net
morsin.com	netzidiot.de
moscowmail.com	neverbox.com
mox.pp.ua	nextemail.in
mozej.com	nezdiro.org
mrlottery.com	nicked.com
mrossi.ml	nickkrizos.com
mrsporty.com	niketexanshome.com
msa.minsmail.com	nincsmail.com
msgos.com	nincsmail.hu
mswork.ru	no-spam.ws
mt2009.com	nobulk.com
mt2014.com	noclickemail.com
mt2015.com	nogmailspam.info
muimail.com	noicd.com
mvrht.com	noipmail.com
mvrht.net	nomail.xl.cx
mwp4wcqnh7t.cf	nomail2me.com
mx0.ew.eu	nomorespamemails.com
my10minutemail.com	norih.com
mycleaninbox.net	nospam.ze.tc
mypartyclip.de	nospam4.us
myphantomemail.com	nospamfor.us
myspaceinc.com	nospamthanks.info
myspaceinc.net	not0k.com
myspaceinc.org	notmailinator.com
myspacepimpedup.com	notsharingmy.info
myspamless.com	now.mefound.com
mystvpn.com	nowmymail.com
mytrashmail.com	nurfuerspam.de
mziqo.com	nus.edu.sg
nada.ltd	nutpa.net

nwldx.com	pdold.com
nwytg.com	pecinan.com
nwytg.net	pecinan.net
nypato.com	pecinan.org
o.s01.cc	penisuzvetseni.com
o3enzyme.com	pfui.ru
oalisp.com	piloq.com
objectmail.com	pimpedupmyspace.com
obobbo.com	pisls.com
oepia.com	pjjkp.com
office.microsoft.com.bel.kr	poh.pp.ua
oing.cf	pokemail.net
olypmall.ru	politikercclub.de
one.haddo.eu	polyfaust.com
one2mail.info	pooae.com
oneoffemail.com	poofy.org
onewaymail.com	pookmail.com
online.ms	poreglot.ru
oopi.org	porsh.net
oou.us	posdz.com
opayq.com	posta.store
openmail.cc	postonline.me
ordinaryamerican.net	prayersa3.com
otherinbox.com	privacy.net
otvetinavoprosi.com	proxymail.eu
ourklips.com	prtnx.com
outlawspam.com	psles.com
ovpn.to	psoxs.com
owlpic.com	pullmail.info
ownplace.com	punkass.com
ozozwd2p.com	putthisinyourspamdatabase.com
p33.org	pw.epac.to
pammy.com	pwrby.com
pancakemail.com	q.awatum.de
paplease.com	qasti.com
pavilionx2.com	qisdo.com
pay-mon.com	qisoa.com
payperex2.com	qtum-ico.com
pcmylife.com	qualitymail.com

quickemail.info	secmail.pro
quickinbox.com	seekjobs4u.com
r-a-v-e.com	selfdestructingmail.com
r31s4fo.com	send22u.info
raddled.com	sendspamhere.com
radioslayer.esy.es	sezet.com
raiasu.ml	seznam.cz
raimu.cf	sfamo.com
rcasd.com	sfrty.ru
rcpt.at	shalar.net
re-gister.com	sharklasers.com
recode.me	shearer.com
recursor.net	shiftmail.com
reftoken.net	shinnemo.com
refurhost.com	shitmail.me
regbypass.com	shoeonlineblog.com
regbypass.comsafe-mail.net	shortmail.net
rejectmail.com	shotmail.ru
rhyta.com	sibmail.com
ribber.com	sina.com
riski.cf	sjuaq.com
rklips.com	skeefmail.com
rmqkr.net	slaskpost.se
rnzcomesth.com	slopsbox.com
robot2.me	smap.4nmv.ru
rootfest.net	smaretboy.pw
rppkn.com	smashmail.de
rtrtr.com	smcleaningbydesign.com
ruggedinbox.com	smellfear.com
ruru.be	snakemail.com
s0ny.net	sneakemail.com
s1xssanlgkgc.ml	sofimail.com
safe-mail.net	sofort-mail.de
safersignup.de	sogetthis.com
safetymail.info	sohai.ml
safetypost.de	soioa.com
sandelf.de	soisz.com
sawoe.com	solvemail.info
saynotospams.com	soodonims.com

spam.2012-2016.ru	spamkill.info
spam.la	spaml.com
spam.su	spaml.de
spam4.me	spammotel.com
spamavert.com	spamobox.com
spambob.com	spamoff.de
spambob.net	spamslicer.com
spambob.org	spamspot.com
spambog.com	spamthis.co.uk
spambog.de	spamthisplease.com
spambog.ru	spamtrail.com
spambox.info	speed.ls.fr
spambox.irishspringrealty.com	spikio.com
spambox.me	spoofer.com
spambox.us	spybox.de
spamcannon.com	sqoi.com
spamcannon.net	ss.hi5.si
spamcero.com	ssoia.com
spamcon.org	staircraft5.com
spamcorptastic.com	stalkers.com
spamcowboy.com	starpower.space
spamcowboy.net	startkeys.com
spamcowboy.org	stelliteop.info
spamday.com	stexsy.com
spamex.com	sthelens.com
spamfree24.com	stop-my-spam.pp.ua
spamfree24.de	storiqax.com
spamfree24.eu	storiqax.top
spamfree24.info	storj99.com
spamfree24.net	storj99.top
spamfree24.org	stromox.com
spangourmet.com	stubby.com
spangourmet.net	suioe.com
spangourmet.org	sunster.com
spamherelots.com	supergreatmail.com
spamhereplease.com	supermailer.jp
spamhole.com	superrito.com
spamify.com	suremail.info
spaminator.de	svk.jp

sweetxxx.de	thankyou2010.com
syujob.accountants	thearchway.com
tacomail.com	thebrewer.com
tafmail.com	thecloudindex.com
taphear.com	thecutest.com
taureans.com	theeasymail.com
taylorventuresllc.com	thelostworld.com
te.budayationghoa.com	thepsoft.org
te.caseedu.tk	thisisnotmyrealemail.com
te.pecinan.org	thraml.com
tech69.com	thrma.com
techgroup.me	throam.com
teentoday.every1.net	throwam.com
teewars.org	throwawayemailaddress.com
teleosaurs.xyz	throya.com
teleworm.com	thunderbolt.science
teleworm.us	tiggy.com
tempalias.com	tilien.com
tempe-mail.com	tinoza.org
tempemail.biz	tipsb.com
tempemail.com	tizzi.com
tempemail.net	tm2mail.com
tempinbox.co.uk	tmail.ws
tempinbox.com	tmailinator.com
tempmail.it	tmails.net
tempmail.us	tmpmail.net
tempmail2.com	tmpmail.org
tempmailapp.com	tool.pp.ua
tempomail.fr	toomail.biz
temporarily.de	top1mail.ru
temporarioemail.com.br	top1post.ru
temporaryemail.net	topikt.com
temporaryforwarding.com	tormail.org
temporaryinbox.com	tosser.com
tempr.email	tqoai.com
tempsky.com	tqosi.com
tenderkiss.com	tradermail.info
tfnw.gf	trash-amil.com
thanksnospam.info	trash-mail.at

trash-mail.com	uphomail.ga
trash-mail.de	upliftnow.com
trash2009.com	uplipht.com
trashemail.de	urchatz.ga
trashmail.at	urhen.com
trashmail.com	urx7.com
trashmail.de	uscaves.com
trashmail.me	ushijima1129.gq
trashmail.net	ux.dob.jp
trashmail.org	uyhip.com
trashmail.ws	v.0v.ro
trashmailer.com	valemail.net
trashymail.com	vektik.com
trashymail.net	venompen.com
travala10.com	veryrealemail.com
trbvm.com	vfemail.net
trbvn.com	viditag.com
trbvo.com	viewcastmedia.com
trickmail.net	viewcastmedia.net
trilby.com	viewcastmedia.org
trillianpro.com	vipepe.com
trimsj.com	virtualemail.info
tryalert.com	vkcode.ru
tulnl.xyz	vlstwoclbfqip.ga
turual.com	vmani.com
tutye.com	voidbay.com
tverya.com	vomoto.com
twinmail.de	vpk.gmeil.me
twocowmail.net	vpn33.top
tyldd.com	vps30.com
u.0u.ro	vps911.net
u.10x.es	vpslists.com
u.mef.ru	vpsorg.pro
u03.gmailmirror.com	vpsorg.top
ubismail.net	vpstraffic.com
ucylu.com	vr5gpowerv.com
uemail99.com	w.0w.ro
uggsrock.com	walkmail.net
ugimail.net	walkmail.ru

wat.emailfake.ml	xzsok.com
web-mail.pp.ua	yahooproduct.net
web2mailco.com	yandex.com
webm4il.info	yandex.ru
wegwerfadresse.de	yandex.com
wegwerfemail.de	yeah.net
wegwerfmail.de	yep.it
wegwerfmail.net	yhcaturkl79jk.tk
wegwerfmail.org	yhg.biz
werrmai.com	yk20.com
wesmailer.com	yogamaven.com
wetrainbayarea.com	yomail.info
wetrainbayarea.org	yopmail.biz.st
wh4f.org	yopmail.com
whatsaas.com	yopmail.fr
whyspam.me	yopmail.net
wickmail.net	yopmail.pp.ua
wikidocuslava.ru	youzend.net
willselfdestruct.com	ypmail.webarnak.fr.eu.org
wimsg.com	yroid.com
winemaven.info	yuurok.com
wla9c4em.com	zain.site
wmail.club	zainmax.net
works.usa.cc	zaktouni.fr
wronghead.com	zamge.com
wscu73sazlccqsir.ml	zasod.com
wuzup.net	zehnminutenmail.de
wuzupmail.net	zep-hyr.com
xagloo.com	zetmail.com
xemaps.com	zhorachu.com
xents.com	zipcad.com
xmaily.com	zippiex.com
xmlfour.ru	zippymail.info
xmlone.ru	ziyap.com
xmlthree.ru	zoaxe.com
xmltwo.ru	zoemail.org
xoixa.com	zombee.33mail.com
xoxy.net	zukmail.ga
xperiae5.com	zwwnhmmcec57ziwux.cf
xww.ro	