Análise de métodos de divulgação de endereços IPv6 e suas implicações para a cibersegurança

Luan Teodoro Mundim

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uberlândia
2024

Luan Teodoro Mundim

Análise de métodos de divulgação de endereços IPv6 e suas implicações para a cibersegurança

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Rodrigo Sanches Miani
Coorientador: Ivan da Silva Sendin

Uberlândia
2024

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

M965a
2024

Mundim, Luan Teodoro, 1993-
Análise de métodos de divulgação de endereços IPv6 e suas implicações para a cibersegurança [recurso eletrônico] / Luan Teodoro Mundim. - 2024.

Orientador: Rodrigo Sanches Miani.
Coorientador: Ivan da SilvaSendin.
Dissertação (Mestrado) - Universidade Federal de Uberlândia, Programa de Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
Disponível em: http://doi.org/10.14393/ufu.di.2025.5511
Inclui bibliografia.
Inclui ilustrações.

1. Computação. 2. Sistemas de segurança. 3. Cibernética - Medidas de segurança. I. Miani, Rodrigo Sanches, 1983-, (Orient.). II. Sendin, Ivan da Silva, 1975-, (Coorient.). III. Universidade Federal de Uberlândia. Programa de Pós-graduação em Ciência da Computação. IV. Título.

CDU: 681.3

Rejâne Maria da Silva
Bibliotecária-Documentalista – CRB6/1925

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Computação
Av. João Naves de Ávila, 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpgfacom@ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Dissertação, 48/2024, PPGCO				
Data:	20 de dezembro de 2024	Hora de início:	09:02	Hora de encerramento:	11:25
Matrícula do Discente:	12212CCP014				
Nome do Discente:	Luan Teodoro Mundim				
Título do Trabalho:	Análise de métodos de divulgação de endereços IPv6 e suas implicações para a cibersegurança				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Sistemas de Computação				
Projeto de Pesquisa de vinculação:	Projeto de pesquisa de vinculação: Google Cloud Research Credits Program				

Reuniu-se por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Ivan da Silva Sendin (Coorientador) FACOM/UFU, Diego Nunes Molinos - FACOM/UFU, Bruno Bogaz Zarpelão - CCE - Departamento de Computação/UEL e Rodrigo Sanches Miani - FACOM/UFU, orientador do candidato.

Os examinadores participaram desde as seguintes localidades: Bruno Bogaz Zarpelão Londrina/PR . Os outros membros da banca e o aluno participaram da cidade de Uberlândia.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Rodrigo Sanches Miani, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação da Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir ao candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação

interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Bruno Bogaz Zarpelão**, **Usuário Externo**, em 23/12/2024, às 09:42, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Diego Nunes Molinos**, **Professor(a) do Magistério Superior**, em 23/12/2024, às 11:48, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Ivan da Silva Sendin**, **Professor(a) do Magistério Superior**, em 23/12/2024, às 14:09, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Rodrigo Sanches Miani**, **Professor(a) do Magistério Superior**, em 26/12/2024, às 10:47, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5987672** e o código CRC **0A20AE25**.

Referência: Processo nº 23117.085646/2024-22 SEI nº 5987672

Dedico esse trabalho a todas as pessoas que encontraram o seu próprio propósito de vida.

Agradecimentos

Se passou muito tempo desde o inicio dessa parte da minha vida e quando paro pra pensar tanta coisa aconteceu desde que entrei no mestrado, ele se tornou não só um objetivo cumprido, mas parte de quem eu me tornei e por isso gostaria de agradecer a quem me acompanhou durante essa jornada.

Meus pais Luciano e Iáscara que sempre estiveram presentes e amorosos e foram a base pra eu poder me tornar quem sou, vocês são minha vida! Ao meu irmão Jean que me se tornou também meu grande amigo e que me orgulho tanto de ser quem é, a todos os meus amigos, sei que, as vezes, fui pretérito mas sempre tentei estar presente! Principalmente, Claudinho, Caio, Parru, Isa e Lari que me abrigam em seus corações por todo esse tempo. Ao meu orientador Rodrigo e co-orientador Ivan por terem me apoiado e me dado energia pra continuar e finalizar esse trabalho e finalmente a Universidade Federal de Uberlândia por essa oportunidade de estudo.

“Antes carregava as suas cinzas para as montanhas; agora volta os vales levando fogo!”

(Assim falou Zaratustra, Friedrich Nietzsche)

Resumo

A transição para o *Internet Protocol version 6* (IPv6) é um processo contínuo, impulsionado pelo crescente número de dispositivos conectados à Internet. Desde a publicação da *Request for Comments* (RFC) 2460 em que foi criado o IPv6, no final da década de 1990, o IPv6 oferece um espaço de endereçamento vastamente superior ao *Internet Protocol version 4* (IPv4). No entanto, a escassez de ferramentas eficientes para escanear a vasta gama de endereços IPv6 torna a identificação de hosts ativos um desafio, tanto para atacantes quanto para pesquisadores. Diante desse cenário, este trabalho investigou cinco métodos de divulgação de endereços IPv6 com o objetivo de estudar a detecção de novas ameaças utilizando como infraestrutura *honeypots* implementados em nuvem sob a perspectiva de quem deseja que seus serviços sejam encontrados com o uso deste protocolo. O primeiro método envolveu a utilização dos motores de busca `Censys` e `Shodan` para identificar e armazenar endereços IPv6 em suas bases de dados. O segundo método consistiu na divulgação explícita de endereços em fóruns de cibersegurança. O terceiro método utilizou rastreadores web ou *webcrawlers* para identificar sites que utilizam IPv6 e acessá-los de forma automatizada. O quarto método envolveu a consulta a servidores *Domain System Name* (DNS) para divulgar os endereços IPv6. Por fim, o quinto método explorou a comunicação interna entre as instâncias que hospedam os serviços. Os resultados indicaram que a divulgação de endereços IPv6 não é trivial. Nos dois experimentos realizados, foram configuradas 35 instâncias *honeypots* no total. Destas, uma em cada experimento registrou pelo menos uma tentativa de acesso. Uma das instâncias obteve registros de ataques nas portas 22 e 23 com tentativa de acesso, e a outra instância teve varredura efetuada em 5.060 portas diferentes das 65.535 portas, sendo a primeira utilizando o método de divulgação *Censys* e *Shodan* e a outra sendo uma instância de controle, ou seja, não teve métodos de divulgação aplicados.

Palavras-chave: IPv6. Honeypot. DNS. Métodos de divulgação. Cibersegurança. Segurança da Informação. Ataques Cibernéticos.

Abstract

The transition to IPv6 is an ongoing process, driven by the growing number of devices connected to the Internet. Since the publication of RFC 2460 which created IPv6 in the late 1990s, IPv6 offers a vastly superior addressing space compared to IPv4. However, the scarcity of efficient tools to scan the vast range of IPv6 addresses makes identifying active hosts a challenge for both attackers and researchers. Given this scenario, this work investigated five methods of IPv6 address disclosure with the aim of studying the detection of new threats using cloud-implemented *honeypots* as infrastructure from the perspective of those who want their services to be found using this protocol. The first method involved using the `Censys` and `Shodan` search engines to identify and store IPv6 addresses in their databases. The second method consisted of explicitly disclosing addresses in cybersecurity forums. The third method used web trackers or *webcrawlers* to identify sites that use IPv6 and access them automatically. The fourth method involved querying DNS servers to disclose IPv6 addresses. Finally, the fifth method explored internal communication between the instances hosting the services. The results indicated that IPv6 address disclosure is not trivial. In the two experiments conducted, 35 *honeypot* instances were configured in total. Of these, one in each experiment recorded at least one access attempt. One of the instances recorded attack attempts on ports 22 and 23 with access attempts, and the other instance had scans performed on 5,060 different ports out of the 65,535 ports, the first using the *Censys* and *Shodan* disclosure method and the other being a control instance, meaning it had no disclosure methods applied.

Keywords: IPv6. Honeypot. DNS. Disclosure methods. Cybersecurity. Information Security. Cyber Attacks.

Lista de ilustrações

Lista de tabelas

ACK *Acknowledgment*

AMQP *Advanced Message Queuing Protocol*

API *Application Programming Interface*

ARP *Address Resolution Protocol*

ASN *Autonomous System Number*

AWS *Amazon Web Services*

BSI *British Standards Institution*

CDN *Content Delivery Network*

CGA *Cryptographically Generated Addresses*

CIA *Confidentiality, Integrity and Availability*

CIANA *Confidentiality, Integrity, Availability, Non-Repudiation and Authentication*

CoAP *Constrained Application Protocol*

DoS *Denial-of-Service*

CRC *Cyclic Redundancy Check*

DDoS *Distributed Denial-of-Service*

DHCP *Dynamic Host Configuration Protocol*

DNS *Domain System Name*

GCP *Google Cloud Plataform*

GPS *Global Positioning System*

HTTP *HyperText Transfer Protocol*

HTTPS *HyperText Transfer Protocol Secure*

IaaS *Infrastructure as a Service*

IAM *Identity and Access Management*

ICMPv6 *Internet Control Message Protocol Version 6*

IID *Interface Identification*

IoT *Internet of Things*

IP *Internet Protocol*

IPv4 *Internet Protocol version 4*

IPv6 *Internet Protocol version 6*

ISP *Internet Service Provider*

LAN *Local Area Network*

MAC *Media Access Control*

MFA *Multi Factor Authentication*

MQTT *Message Queuing Telemetry Transport*

NDP *Neighbor Discovery Protocol*

NIST *National Institute of Standards and Technology*

NTP *Network Time Protocol*

OPC UA *Open Platform Communications Unified Architecture*

RFC *Request for Comments*

SAN *Subject Alternative Name*

SHA-128 *Secure Hash Algorithm 128*

SHA-256 *Secure Hash Algorithm 256*

SLAAC *Stateless Address Autoconfiguration*

SMS Short Message Service

SSH *Secure Shell*

SYN *Synchronize*

TCP *Transmission Control Protocol*

Telnet *Teletype Network*

TLS *Transport Layer Security*

TLD *Top-Level Domain*

TOR *The Onion Router*

VM *Virtual Machine*

VPN *Virtual Private Network*

WAN *Wide Area Network*

Sumário

CAPÍTULO 1

Introdução

A evolução tecnológica das últimas décadas trouxe consigo uma dependência cada vez maior de sistemas e redes digitais. Paralelamente a esse avanço, as ameaças cibernéticas se tornaram mais sofisticadas e frequentes, exigindo medidas de segurança cada vez mais robustas. Neste contexto, a segurança da informação e a cibersegurança emergem como tópicos cruciais para proteger dados, sistemas e infraestruturas críticas. A segurança da informação engloba um conjunto de práticas e tecnologias destinadas a garantir a confidencialidade, integridade e disponibilidade da informação, independentemente do formato em que se encontre. Já a cibersegurança, como um subconjunto da segurança da informação, concentra-se especificamente na proteção de sistemas e dados digitais (STALLINGS, 2002; National Institute of Standards and Technology, 2024c; National Institute of Standards and Technology, 2024b).

Uma forma de proteger os dados é utilizar técnicas e ferramentas de teste para identificar possíveis vulnerabilidades que possam comprometer os dados ou o sistema. Uma técnica bastante utilizada é a varredura de vulnerabilidades. Um *scanner* de vulnerabilidades de rede atua como um detector de falhas de segurança, analisando um determinado alvo em busca de pontos fracos exploráveis por atacantes. O processo típico de avaliação de vulnerabilidades consiste em três etapas: mapeamento do alvo, identificação de vulnerabilidades e análise detalhada do risco associado a cada uma delas (MANZUIK; GOLD; GATFORD, 2006). Após o mapeamento do alvo, a identificação de vulnerabilidades e a análise detalhada do risco são realizadas para garantir a efetividade da varredura.

Utilizando o software de varredura de rede `Zmap`[1] conectado a uma interface de 10Gbps por exemplo, é possível varrer todos os 4.294.967.296 endereços IPv4 em apenas cinco minutos. Como forma de solucionar o problema da relativa baixa quantidade de endereços IP do IPv4, o protocolo IPv6 possui endereços de 128 bits, em comparação aos 32 bits do IPv4. Isso significa que, para cada endereço IPv4, existem aproximadamente 2^{98} endereços IPv6.

Essa grande quantidade de endereços IPv6 introduz novos desafios em termos de segurança, por exemplo tornar a tarefa de identificar e rastrear dispositivos na rede significativa-

[1] https://zmap.io

mente mais complexa. A dificuldade de mapear toda a vasta gama de endereços representa um obstáculo para a detecção de atividades maliciosas. Utilizar formas tradicionais de varredura do espaço no IPv6 é computacionalmente inviável devido a quantidade de endereços IPv6 (TANVEER et al., 2023).

Além da questão do tamanho do espaço de endereçamento, a arquitetura do IPv6 introduz novos conceitos e mecanismos que podem impactar a segurança. A autoconfiguração de endereços *stateless*, por exemplo, enquanto simplifica a configuração de redes, pode tornar mais difícil rastrear a origem de tráfego malicioso (BAIG; ADENIYE, 2011; NARTEN; JINMEI; THOMSON, 2007). A fragmentação de pacotes, outro recurso do IPv6, pode ser explorada por atacantes para ocultar assinaturas de ataques (HUMMEN et al., 2013; GONT, 2013). Portanto a segurança em redes IPv6 é um campo de pesquisa ativo, com diversos desafios a serem enfrentados (ASHRAF et al., 2023). A constante evolução das ameaças cibernéticas exigem um esforço contínuo da comunidade de segurança para desenvolver soluções eficientes.

Apesar da varredura de endereços IPv6 de forma tradicional, como é feito para IPv4 ser inviável, existem alguns estudos que buscam outras técnicas para realizar o trabalho de capturar endereços ativos dentro do IPv6.

Uma forma de entender quais os padrões que atacantes usam para fazer varreduras em redes IPv6 é utilizar uma grande faixa de endereços IPv6 como alvo (TANVEER et al., 2023). Outra forma é fazer uso do tráfego de dados que passa por servidores DNS gerado por consultas de endereços IPv6 para entender o monitoramento dessa rede (FUKUDA; HEIDEMANN, 2018). Também é possível utilizar os dados do firewall de uma grande *Content Delivery Network* (CDN) para entender o comportamento, origem do tráfego e serviços mais visados no varredura de endereços IPv6 (RICHTER; GASSER; BERGER, 2022). Alternativamente, é possível documentar os endereços IPv6 ativos através de `hitlists`, que consistem em extensas listas de endereços IPv6 previamente detectados na rede (RYE; LEVIN, 2023).

No entanto, há pouca discussão acerca da dificuldade de um endereço IPv6 tornar-se conhecido na Internet, isto é, como os serviços operantes nesse endereço IPv6 podem despertar interesse. Uma maneira de compreender esse processo é por meio da análise da eficiência dos métodos de divulgação de endereços IPv6.

Com base nos trabalhos previamente citados, este estudo realizou dois experimentos utilizando *honeypots* em ambiente de computação em nuvem como infraestrutura para hospedar as instâncias destinadas à análise de cinco métodos de divulgação de endereços IPv6.

Os métodos de divulgação foram divididos em dois experimentos distintos: o Experimento A – Divulgação em motores de busca e fóruns e o Experimento B – Divulgação *Crawler*, DNS e WAN. O Experimento A empregou dois métodos de divulgação, enquanto o Experimento B utilizou três métodos diferentes.

O Experimento A teve como principio avaliar dois métodos, a divulgação nos motores de busca *Censys* e `Shodan`, que são motores de busca especializados, que diferentemente dos

motores de busca convencionais como o Google, explora a infraestrutura da rede, buscando por servidores, roteadores ou outros dispositivos de redes. Esses motores de busca exploram além da *Surface Web*, a *Deep Web* e *Dark Web*. A segunda forma de divulgação foi publicação dos endereços IPv6 em fóruns de cibersegurança.

No Experimento B, foram utilizados três métodos de divulgação, com o objetivo de aproveitar o próprio tráfego e infraestrutura da Internet. A ideia foi utilizar serviços que já empregam endereços IPv6 como um meio indireto de divulgação.

O primeiro método consistiu em um *Crawler* que simulava a navegação Web de um usuário comum, acessando diversos *websites*. A segunda abordagem envolveu a realização de consultas a servidores DNS, visando divulgar os endereços IP nas bases de dados desses mecanismos. Por fim, foi empregada uma divulgação *Wide Area Network* (WAN), similar ao *Crawler*, porém restrita à comunicação entre as instâncias criadas especificamente para o experimento.

1.1 Objetivos da pesquisa

O objetivo geral deste trabalho é analisar a eficiência de métodos de divulgação de endereços IPv6 a fim de entender o comportamento de atacantes que têm como objetivo varrer e explorar esses endereços. Foram analisados cinco métodos de divulgação, divididos em dois experimentos diferentes.

1.1.1 Objetivos específicos

- ❏ Desenvolver um `framework` para construção de um *honeypot* de baixa interação com suporte a IPv6.
- ❏ Avaliar a eficiência dos métodos de divulgação estudados.

1.2 Hipótese

A aplicação dos métodos de divulgação investigados neste trabalho permite que endereços IPv6, previamente desconhecidos, sejam indexados e, consequentemente, descobertos por usuários da Internet, sejam atacantes ou serviços que possuem acessos aos meios de divulgação utilizados.

1.3 Organização do trabalho

Esta dissertação está estruturada da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica que embasa os conceitos fundamentais para entendimento desta pesquisa. Em seguida, o Capítulo 3 realiza uma revisão da literatura, contextualizando o trabalho no

estado da arte. O Capítulo 4 detalha a metodologia empregada, apresentando as etapas do desenvolvimento da pesquisa. Os resultados obtidos são apresentados e analisados no Capítulo 5. Por fim, o Capítulo 6 sintetiza as principais contribuições da pesquisa, as conclusões e sugere direções para futuras pesquisas.

CAPÍTULO 2

Fundamentação Teórica

Neste capítulo será apresentada a base teórica fundamental para a compreensão deste trabalho. Na Seção 2.1, são apresentados os pilares da segurança da informação, fornecendo um panorama geral dos principais conceitos da área. Na Seção 2.2, são apresentados os tipos e funções dos controles de segurança. Em seguida, a Seção 2.3 explorará a definição e os tipos de *honeypots*, além de suas aplicações, benefícios e limitações. A Seção 2.4 abordará o protocolo IPv6, discutindo suas características e as principais diferenças em relação ao IPv4. A virtualização e a computação em nuvem serão os temas da Seção 2.5, com foco em seus impactos na segurança da informação e na infraestrutura de redes. A Seção 2.6 apresentará o funcionamento do DNS, elucidando seu papel na resolução de nomes de host e na arquitetura da Internet, além de seu papel na cibersegurança. Por fim, na Seção 2.7, será abordado o funcionamento básico do protocolo TCP e o funcionamento do *Synchronize* (SYN) scan, as ferramentas `Shodan` e `Censys`, as camadas da Internet e finalmente redes WAN e *Local Area Network* (LAN).

2.1 Segurança da informação

A segurança da informação se baseia em três pilares fundamentais: Confidencialidade, Integridade e Disponibilidade. Esses pilares, conhecidos pela sigla *Confidentiality, Integrity and Availability* (CIA), servem para garantir a segurança de sistemas e serviços (MICROSOFT, 2024). Além da tríade CIA, é comum encontrar a sigla *Confidentiality, Integrity, Availability, Non-Repudiation and Authentication* (CIANA), que inclui, além dos três pilares iniciais, os conceitos de Não-Repudiação e Autenticação (GRAHAM; OLSON; HOWARD, 2016).

2.1.1 Confidencialidade

A Confidencialidade garante que a informação não será divulgada para nenhuma pessoa, processo ou sistema que não possua autorização (GRAHAM; OLSON; HOWARD, 2016). A informação deve ter propriedades que permitam o acesso somente a pessoas autorizadas.

Por exemplo, uma forma de se obter confidencialidade é cifrar um arquivo, garantindo que somente quem tiver acesso à chave de deciframento poderá decifrar e ler o arquivo.

2.1.2 Integridade

A integridade garante que a informação não sofreu modificação não autorizada (GRAHAM; OLSON; HOWARD, 2016). Uma forma de garantir que os dados não sofreram nenhuma modificação não autorizada é através do uso de funções de hash criptográficas. A função de hash criptográfica possui uma entrada de tamanho variável e retorna uma saída de tamanho fixo chamada de *digest*. As funções de hash criptográficas possuem algumas propriedades que garantem a integridade do dado. Algumas das propriedades mais importantes para uma função de hash criptográfica segura estão descritas abaixo (STALLINGS, 2002):

- ❏ Tamanho de entrada variável: A função pode ser aplicada a um bloco de dados de qualquer tamanho;
- ❏ Tamanho de saída fixo: A função produz uma saída de comprimento fixo;
- ❏ Eficiência: A função é relativamente fácil de computar para qualquer entrada, tornando práticas as implementações tanto em hardware quanto em software;
- ❏ Resistente à pré-imagem: Para qualquer valor hash h dado, é computacionalmente inviável encontrar y tal que $Hash(y) = h$;
- ❏ Resistência à segunda pré-imagem: Para qualquer bloco x dado, é computacionalmente inviável encontrar $y \neq x$ com $Hash(y) = Hash(x)$;
- ❏ Resistente à colisão: É computacionalmente inviável encontrar qualquer par (x, y) com $x \neq y$, tal que $Hash(x) = Hash(y)$;
- ❏ Pseudoaleatoriedade: A saída da função de hash atende aos testes padrão de pseudoaleatoriedade.

Alguns exemplos de algoritmos de hash comumente usados são *Secure Hash Algorithm 128* (SHA-128) e *Secure Hash Algorithm 256* (SHA-256).

2.1.3 Disponibilidade, Não-repúdio e Autenticação

A disponibilidade garante que sistemas, informações ou recursos estejam sempre disponíveis quando solicitados por usuários autorizados (GRAHAM; OLSON; HOWARD, 2016). Um exemplo de ataque que diminui a disponibilidade de sistemas é um ataque *Denial-of-Service* (DoS) (KUROSE; ROSS, 2012), que diminui ou suspende a disponibilidade de um serviço, como um site por exemplo, fazendo com que os usuários tenham pouco ou nenhum acesso à plataforma.

O Não-Repúdio garante que todos os que estiveram envolvidos em uma comunicação não possam negar sua participação (STALLINGS, 2002). Uma forma de implementação do não-repúdio é através do uso de criptografia assimétrica. Na criptografia assimétrica (AYELE, 2021), existem dois tipos de chaves: a chave pública, que é livre para todos conhecerem, e a chave privada, à qual somente o usuário deve ter acesso. Assim, quando uma informação é criptografada usando a chave pública, somente a chave privada consegue descriptografá-la.

A autenticação é uma medida de segurança, que garante que o indivíduo ou entidade é quem clama ser durante uma comunicação ou transação (MOLLIN, 2006). Existem algumas formas de autenticação:

- ❏ Algo que você sabe, como uma senha;
- ❏ Algo que você possui, como um código enviado por Short Message Service (SMS) para o *smartphone*;
- ❏ Algo que você é, como um varredura de impressão digital;
- ❏ Algum lugar que você está, usar informações de *Global Positioning System* (GPS) para habilitar alguma autenticação.

A utilização de pelo menos uma dessas formas de autenticação garante quem foi a pessoa a acessar a informação. Utilizar pelo menos duas formas de autenticação é uma boa prática de segurança, chamada de *Multi Factor Authentication* (MFA).

2.1.4 Ameaças, vulnerabilidades e ataques

As ameaças surgem como forma de comprometer pelo menos um dos pilares da segurança da informação. Ameaças são qualquer ação ou evento que pode prejudicar, causar dano ou perda de dados em um sistema de tecnologia da informação (National Institute of Standards and Technology, 2024a). Alguns exemplos de ameaças são acidentes de causas naturais, ataques cibernéticos, brechas de integridade de dados e vazamento de informação confidencial. As vulnerabilidades são falhas ou fraquezas de um sistema, que podem ser configurações incorretas de *software*, falhas físicas ou falhas em dispositivos físicos da rede, já o ataque é uma ameaça que explora as vulnerabilidades de um sistema, comprometendo-o.

2.2 Controles de segurança

Um controle de segurança é uma contramedida criada para minimizar as chances de uma ameaça explorar uma vulnerabilidade (SWANAGAN, 2023). A Figura 1 mostra os três tipos de controles de segurança (controles técnicos, administrativos e físicos), onde cada controle de segurança pode ser separado por funções.

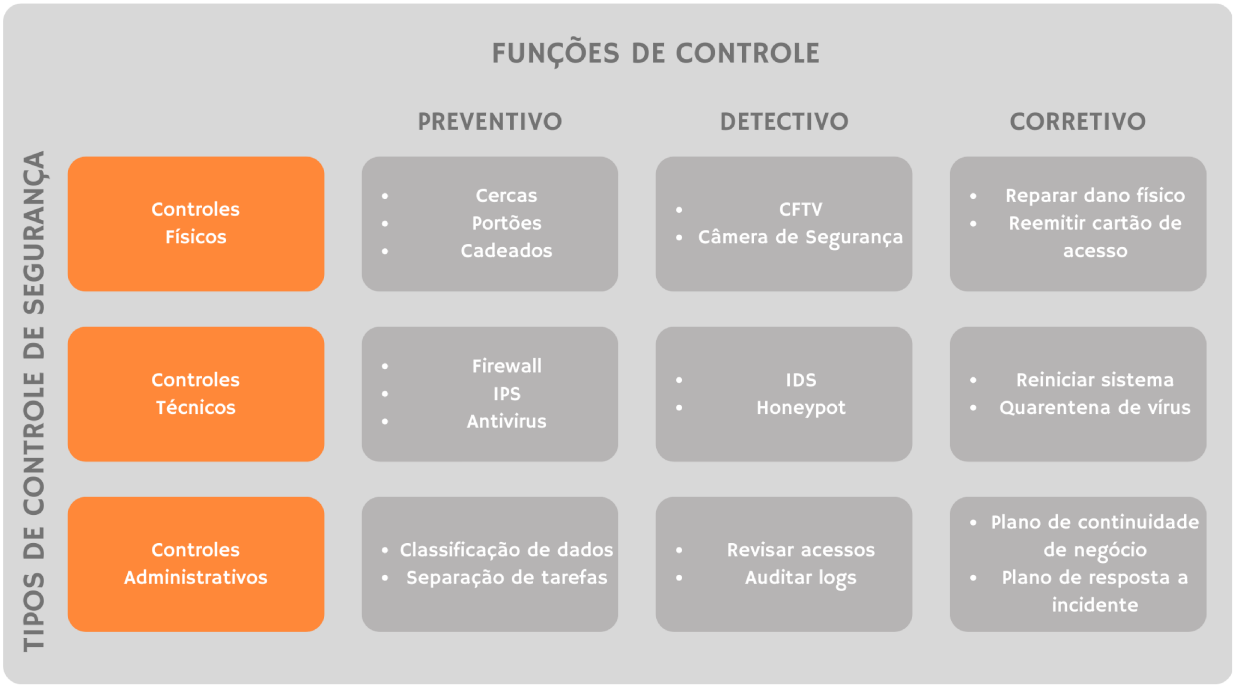


Figura 1 – Funções e tipos de controle de segurança, sendo *honeypot* um tipo de controle técnico que possui uma função detectiva

Fonte: Adaptado de (SWANAGAN, 2023)

Os controles técnicos são tecnologias, dispositivos e softwares implementados para gerenciar e diminuir riscos como o uso de antivírus, varreduras em busca de falhas, *firewalls* ou um sistema de detecção de intrusão.

Controles Administrativos são processos que são implementados pela governança ou time de segurança como forma de controle administrativo, como por exemplo avaliar os riscos de um software novo pra a empresa, renovação a cada período das senhas dos funcionários e associados ou implementar politicas de segurança.

Por fim existem os controles físicos que são criados para impedir ou dificultar a entrada e acesso físicos ao local por exemplo portões, cadeados e patrulha de segurança.

Para implementação das categorias de controles de segurança existem as funções de controle que são três: controles preventivos, controles detectivos e controles corretivos (SWANAGAN, 2023).

Os controles preventivos são realizados de forma proativa a fim de proteção contra falhas ou brechas, que pode ser a implementação de MFA dentro da organização.

Os controles detectivos são formas de monitorar as ações de possíveis atacantes dentro da organização, como por exemplo com o uso de *honeyfiles*, que são arquivos que podem disparar um alarme caso sejam abertos por um atacante, ou *honeypots*, explicado melhor na Seção 2.3.

Controles corretivos são aplicados após a exploração de uma vulnerabilidade de segu-

rança, com o objetivo de restaurar as condições normais do sistema e prevenir a ocorrência de novas explorações semelhantes. Exemplos de controles corretivos incluem a aplicação de *patches* de segurança.

2.3 Honeypots

Honeypots são sistemas ou dispositivos de rede projetados para atrair e enganar atacantes, permitindo que os administradores de segurança monitorem suas atividades e obtenham informações valiosas sobre suas técnicas e intenções. Um *honeypot* é "um recurso de informação cujo valor reside em ser sondado, atacado ou comprometido" (SPITZNER, 2003), muitas vezes possuindo vulnerabilidades de forma proposital, visto que é preciso atrair os atacantes. Os *honeypots* podem ser classificados de várias maneiras, dependendo de sua complexidade, propósito e nível de interação.

A Figura 2 mostra uma topologia que pode ser usada para implantação de um *honeypot*, onde este fica separado de outros serviços que podem funcionar na mesma rede. A outra parte da rede fica em uma rede privada, criando uma segregação com o *honeypot* e aumentando a segurança caso ocorra algum tipo de ataque.

2.3.1 Níveis de interação

Dependendo da complexidade de implementação e funções, os *honeypots* podem ser classificados como um *honeypot* de baixa, média ou alta interação. Os *honeypots* de baixa interação são sistemas que simulam serviços e vulnerabilidades básicas, mas não oferecem um ambiente completo para os atacantes explorarem. Eles são fáceis de configurar e mantêm um risco mínimo, pois limitam a interação do atacante com o sistema. Dessa forma, fornecem informações limitadas sobre as técnicas de ataque. Um exemplo é o `Honeyd` (PROVOS, 2003), que emula serviços de rede em um ambiente controlado.

Os *honeypots* de alta interação estão no outro extremo do espectro de nível de interação e são sistemas completos, que permitem aos atacantes interagir de forma mais profunda com o ambiente. Esses *honeypots* fornecem informações detalhadas sobre as técnicas e ferramentas utilizadas pelos atacantes, mas também apresentam um risco maior, pois os atacantes têm mais liberdade para explorar o sistema. Um exemplo é o projeto `Honeynet` (PROJECT, 2004), que cria redes inteiras de *honeypots* para estudo de ataques.

Entre esses dois tipos de *honeypots* no espectro de nível de interatividade, existem os *honeypots* de média interação, que possuem uma interação, superior aos *honeypots* de baixa interação, mas não a ponto de serem completos como os *honeypots* de alta interação. O `Cowrie`[1], quando configurado com todas as suas funções, pode ser considerado um *honeypot* de média interação.

[1] https://cowrie.readthedocs.io/en/latest/index.html

2.3.2 *Honeypot* de pesquisa e produção

Honeypots de pesquisa são utilizados principalmente por pesquisadores e acadêmicos para estudar o comportamento dos atacantes e desenvolver novas estratégias de defesa. Eles são configurados para coletar o máximo de informações possível sobre os métodos de ataque e são frequentemente usados em ambientes controlados (RAZALI et al., 2018).

Honeypots de produção são implementados em ambientes corporativos para proteger redes e sistemas reais. Eles são usados para detectar e mitigar ataques em tempo real, além de fornecer alertas e logs detalhados para os administradores de segurança (SEIFERT; WELCH; KOMISARCZUK, 2006).

2.3.3 Benefícios e limitações dos *honeypot*

Os *honeypots* oferecem vários benefícios dentro da segurança da informação. Eles ajudam a detectar atividades maliciosas que podem passar despercebidas por outras ferramentas de segurança, fornecem informações valiosas sobre novas ameaças e técnicas de ataque, e podem ser usados para enganar e atrasar atacantes, ganhando tempo para implementar medidas de defesa (SPITZNER, 2003).

Apesar de seus benefícios, os *honeypots* também apresentam algumas limitações. Eles podem ser detectados por atacantes experientes, o que reduz sua eficácia. Além disso, a manutenção de *honeypot* de alta interação pode ser complexa e requer recursos significativos.

2.3.4 Cowrie *honeypot*

Cowrie é um *honeypot*, projetado especificamente para monitorar e analisar ataques aos protocolos *Secure Shell* (SSH) e *Teletype Network* (Telnet). Como sucessor do *honeypot* Kippo, `Cowrie` representa um *honeypot* de interação média que simula efetivamente um ambiente vulnerável para atrair atacantes e coletar informações sobre seus métodos e comportamentos. A arquitetura do `Cowrie` permite que ele registre ataques de força bruta e capture as interações de `shell` dos atacantes. (PRIYA; CHAKKARAVARTHY, 2023; RAGSDALE; BOPPANA, 2023; GRIGORIOU et al., 2022).

A principal função do `Cowrie` consiste em atrair atores maliciosos, desviando-os de sistemas reais. Isso é possível graças à criação de um sistema de arquivos que simula uma instalação `Debian`, levando os atacantes a acreditar que estão interagindo com um servidor legítimo (GRIGORIOU et al., 2022). Essa interação é fundamental, pois permite a coleta de logs detalhados que documentam as ações dos atacantes, incluindo a execução de comandos e tentativas de explorar vulnerabilidades (BASER; GUVEN; AYDIN, 2023; PUTRA, 2022; SETIANTO et al., 2021). A análise desses logs possibilita a identificação de padrões nas metodologias de ataque, o que é essencial para o aprimoramento das medidas de segurança da informação (Setianto, 2021).

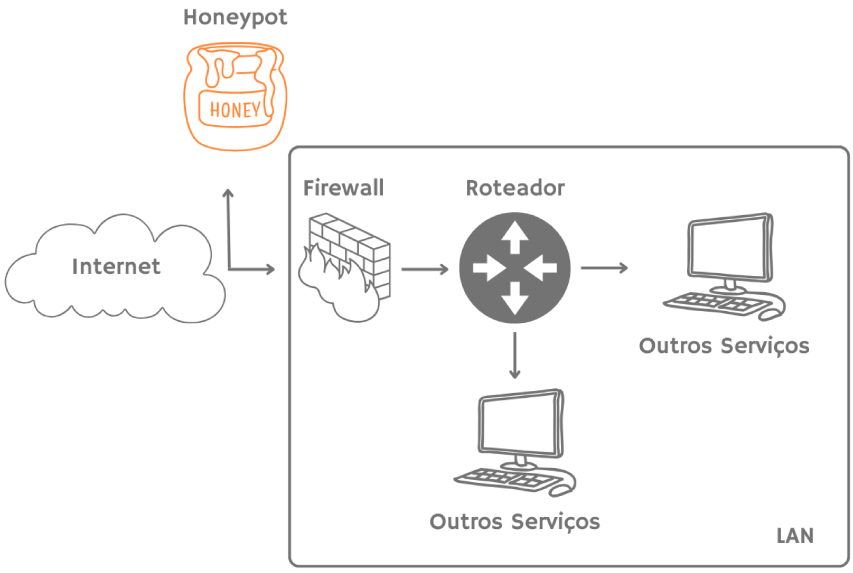


Figura 2 – Topologia de rede com *honeypot*: neste cenário, o *honeypot* é isolado do restante da rede para reforçar a segurança.

Fonte: Autor

2.4 IPv6

O IPv6 foi desenvolvido para superar as limitações do IPv4, como a escassez de endereços, expandindo o tamanho do endereço de 32 para 128 bits. Entretanto, a transição para o IPv6 não está isenta de desafios de segurança. Os novos protocolos introduzidos, como o *Neighbor Discovery Protocol* (NDP), que substitui o *Address Resolution Protocol* (ARP) do IPv4, também trazem vulnerabilidades. O NDP é suscetível a ataques DoS e outras ameaças, o que levanta preocupações sobre a segurança da rede (SETHURAMAN; MATHI, 2018; NAAGAS; GAMILLA, 2022). Além disso, a pesquisa de 2018 indica que 61% dos especialistas em segurança consideram as ameaças de DoS como uma das principais preocupações durante a transição para o IPv6 (NAAGAS; GAMILLA, 2022). A introdução de cabeçalhos de extensão no IPv6 também pode ser explorada por atacantes, aumentando a complexidade da segurança (NAAGAS; GAMILLA, 2022).

O endereço IPv6 é composto por 128 bits e pode ser dividido em três partes, conforme ilustrado na Figura 3: o prefixo de roteamento global, o identificador de subrede e o Interface ID. O tamanho do Interface ID ou *Interface Identification* (IID) é variável; entretanto, é considerada uma boa prática utilizar um comprimento de 64 bits (DEERING; HINDEN, 2006).

O IID, que compõe o endereço IPv6, desempenha um papel crucial na identificação única de dispositivos na rede. O IID é geralmente derivado do endereço *Media Access Control* (MAC) do dispositivo, mas pode ser gerado de forma aleatória para aumentar a privaci-

dade (CHENEAU; LAURENT, 2011).

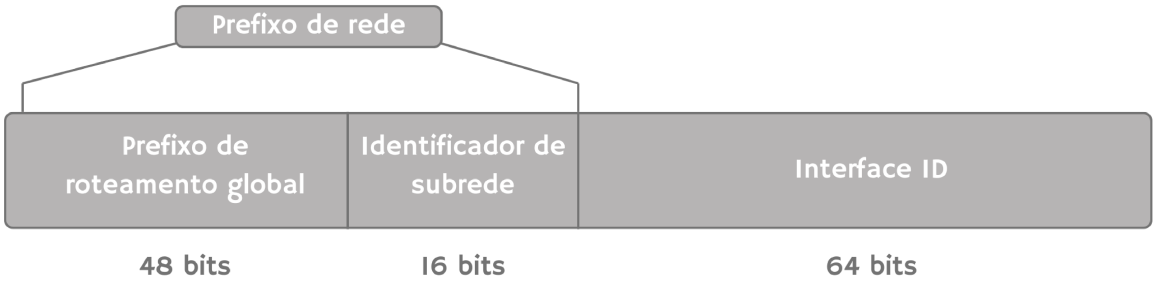


Figura 3 – Exemplo de estrutura do endereço IPv6, sendo 64 bits para o prefixo de rede e 64 bits de interface ID.

Fonte: Adaptado de (KISHIMOTO et al., 2012)

A utilização de endereços gerados criptograficamente, como os *Cryptographically Generated Addresses* (CGA), é uma abordagem que visa proteger a privacidade e a segurança dos dispositivos na rede, dificultando a identificação e o rastreamento (CHENEAU; LAURENT, 2011). Essa técnica é especialmente relevante em ambientes onde a segurança é uma preocupação primordial, como em redes corporativas e na *Internet of Things* (IoT) (LIU et al., 2014/11).

As organizações devem considerar as diretrizes de segurança publicadas por entidades, como o *National Institute of Standards and Technology* (NIST) e a *British Standards Institution* (BSI), que oferecem orientações sobre como abordar as vulnerabilidades específicas do IPv6 (HERMANN; FABIAN, 2014). A falta de suporte adequado para IPv6 em dispositivos de segurança, como *firewalls* e sistemas de detecção de intrusões, pode resultar em lacunas significativas na proteção da rede (CZYZ et al., 2016; HENDRIKS; SPEROTTO; PRAS, 2015). Portanto, é imperativo que as organizações desenvolvam políticas de segurança que considerem as particularidades do IPv6 e as novas ameaças que surgem com sua adoção.

2.4.1 Sub-redes IPv6

Sub-redes IPv6 são divisões lógicas de uma rede IPv6 maior, criando segmentos menores e mais gerenciáveis. Assim como em IPv4, o objetivo é organizar a rede de forma eficiente, otimizar o roteamento e permitir um controle mais granular sobre o tráfego de rede (FALL; STEVENS, 2011).

O IPv6 emprega prefixos para definir o tamanho de uma sub-rede. O prefixo indica o número de bits fixos no início de um endereço IPv6 que identificam a sub-rede. Os bits restantes são utilizados para endereçar os hosts dentro da sub-rede (TANENBAUM; WETHERALL, 2010).

Um prefixo /64 é comumente utilizado para sub-redes IPv6, permitindo 2^{64} endereços IPv6 únicos dentro de uma sub-rede. Esta quantidade é de fato abundante para atender à maioria das redes. Um prefixo /56 permite criar 2^8 sub-redes (256 sub-redes), cada uma com 2^{64} endereços. Da mesma forma, um prefixo /48 possibilita 2^{16} sub-redes, mantendo 2^{64} endereços em cada sub-rede. Para prefixos maiores que /56, como /60, temos 2^4 sub-redes disponíveis, cada uma ainda mantendo 2^{64} endereços. No caso de um /52, são disponibilizadas 2^{12} sub-redes. É importante notar que, independentemente do tamanho do prefixo da rede, o IPv6 mantém o padrão de 2^{64} endereços por sub-rede para compatibilidade com o *Stateless Address Autoconfiguration* (SLAAC), um mecanismo fundamental do protocolo IPv6 que permite que dispositivos de rede configurem automaticamente seus endereços IP sem a necessidade de um servidor centralizado como o *Dynamic Host Configuration Protocol* (DHCP) (GONT, 2014).

2.5 Virtualização e computação em nuvem

A virtualização é uma tecnologia que permite a criação de múltiplas *Virtual Machine* (VM) em um único hardware físico. Cada VM opera como um sistema independente, com seu próprio sistema operacional e aplicativos. A virtualização oferece vários benefícios, incluindo melhor utilização de recursos, isolamento de aplicações e facilidade de gerenciamento (PORTNOY, 2012) o que é uma forma para automatização da criação dos ambientes para *honeypots*.

A computação em nuvem é um modelo de computação que permite o acesso sob demanda a recursos de computação, como servidores, armazenamento e aplicativos, através da Internet. Este modelo oferece várias vantagens, incluindo escalabilidade, flexibilidade e redução de custos (ARMBRUST et al., 2010).

2.5.1 *Google Cloud Platform*

A GCP é uma das principais plataformas de computação em nuvem, oferecendo uma ampla gama de serviços, incluindo computação, armazenamento, banco de dados e aprendizado de máquina. O GCP é projetada para fornecer uma infraestrutura segura e escalável, com várias camadas de segurança integradas (Google Cloud, 2024).

O GCP implementa várias práticas recomendadas de segurança como criptografia de dados em trânsito e em repouso, gerenciamento de *Identity and Access Management* (IAM) e monitoramento contínuo de segurança. Além disso, o GCP oferece ferramentas específicas para ajudar os usuários a gerenciar a segurança de seus ambientes de nuvem, como o *Google Cloud Security Command Center* (Google Cloud, 2024).

2.6 DNS

O DNS é um serviço fundamental da Internet, atuando como um banco de dados hierárquico distribuído que traduz nomes de domínio legíveis por humanos em endereços *Internet Protocol* (IP) legíveis por máquinas. Esse processo permite que os usuários acessem sites e serviços usando nomes fáceis de lembrar, em vez de endereços numéricos (TANENBAUM; WETHERALL, 2010). O DNS opera por meio de uma hierarquia estruturada, onde os nomes de domínio são organizados em níveis, com os domínios de primeiro nível *Top-Level Domain* (TLD)s representando o nível mais alto nessa hierarquia (KHEIR et al., 2014; LU; TSUDIK, 2010; KIM; REEVES, 2020).

O processo de resolução de DNS envolve as seguintes etapas:

1. **Host:** O primeiro ponto de contato é o seu próprio dispositivo (computador, smartphone, etc.). Cada dispositivo possui um arquivo de hosts local, onde você pode manualmente mapear nomes de domínio a endereços IP específicos. No entanto, essa configuração é geralmente usada para fins de teste ou bloqueio de sites e não é a principal fonte de resolução de DNS.

2. **Cache DNS:** Se o nome de domínio não for encontrado no arquivo de hosts local, o sistema operacional consulta seu cache DNS. O cache DNS armazena temporariamente as respostas de consultas DNS recentes, acelerando significativamente o processo de resolução de nomes para sites que você visita com frequência.

3. **Servidor DNS:** Se a resposta não estiver no cache, o sistema operacional envia uma consulta ao seu servidor DNS configurado. Esse servidor pode ser fornecido pelo seu *Internet Service Provider* (ISP) ou configurado manualmente. O servidor DNS irá:

   - ❏ Consultar seu próprio cache: Se a resposta estiver armazenada no cache do servidor DNS, ela será retornada ao seu dispositivo.
   - ❏ Consultar outros servidores DNS: Se a resposta não estiver no cache local, o servidor DNS iniciará um processo recursivo de consulta a outros servidores DNS, como servidores raiz, servidores TLD e servidores autoritativos, até encontrar a resposta correta ou esgotar todas as opções.

4. **Servidores DNS Recursivos e Autoritativos:** Os servidores DNS recursivos são responsáveis por realizar todas as consultas necessárias para encontrar a resposta final, enquanto os servidores DNS autoritativos são os detentores definitivos dos registros de um determinado domínio.

A Figura 4 representa essa hierarquia (DECCIO, 2012; LIAO et al., 2017; ZHANG; LE; XU, 2014).

Os registros DNS são blocos fundamentais, contendo vários tipos de informações associadas a nomes de domínio. Os tipos de registro mais comuns incluem:

1. **Registro A**: Mapeia um nome de domínio para um endereço IPv4, permitindo que os usuários acessem sites usando nomes legíveis por humanos (KHEIR et al., 2014; KIM; REEVES, 2020);

2. **Registro AAAA**: Semelhante ao registro A, mas mapeia um nome de domínio para um endereço IPv6, o que é cada vez mais importante à medida que a Internet faz a transição para IPv6 devido ao esgotamento dos endereços IPv4 (TIAN; WANG, 2014);

3. **Registro CNAME**: Permite que um nome de domínio seja um alias de outro, permitindo que vários nomes de domínio apontem para o mesmo endereço IP sem a necessidade de registros A separados (LU; TSUDIK, 2010);

4. **Registro MX**: Especifica os servidores de troca de e-mail para um domínio, direcionando o tráfego de e-mail para os servidores apropriados (LU; TSUDIK, 2010);

5. **Registro NS**: Indica os servidores de nome autoritativo para um domínio, que são responsáveis por fornecer respostas a consultas sobre esse domínio (ICHISE et al., 2018);

6. **Registro TXT**: Originalmente destinado a texto legível por humanos, esse tipo de registro agora é frequentemente usado para diversos fins de verificação, incluindo SPF (Sender Policy Framework) para autenticação de e-mail (CHIBA et al., 2017).

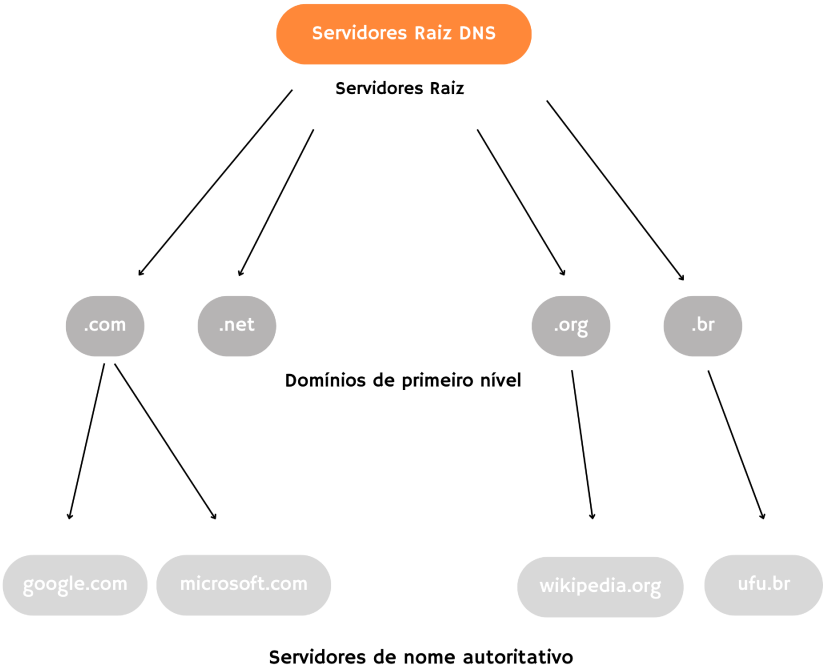


Figura 4 – A hierarquia do sistema DNS apresenta os Servidores Raiz DNS no patamar mais alto. Os servidores de nome autoritativo ocupam posições inferiores nessa cadeia.

Fonte: Adaptado de (JAKAB et al., 2010)

2.7 TCP e SYN scan

O TCP é um dos protocolos fundamentais da Internet, operando na camada de transporte do modelo TCP/IP. Ele é responsável por garantir a entrega confiável de dados entre dispositivos em uma rede, estabelecendo conexões ponto a ponto e ordenando os segmentos de dados (KUROSE; ROSS, 2012).

O TCP é orientado a conexão, o que significa que um canal de comunicação é estabelecido entre os dispositivos antes da transmissão de dados. Esse processo é conhecido como apresentação de três vias ou, *three-way handshake*. A Figura 5 ilustra o processo de estabelecimento de conexão de três vias do TCP, conforme descrito a seguir:

- SYN: O cliente envia um segmento com a flag SYN ativada para o servidor, solicitando a abertura de uma conexão;
- SYN/ACK: O servidor responde com um segmento contendo as flags SYN e *Acknowledgment* (ACK), confirmando o recebimento da solicitação e propondo um número de sequência inicial;
- ACK: O cliente envia um segmento com a flag ACK, confirmando o recebimento da resposta do servidor e estabelecendo a conexão.

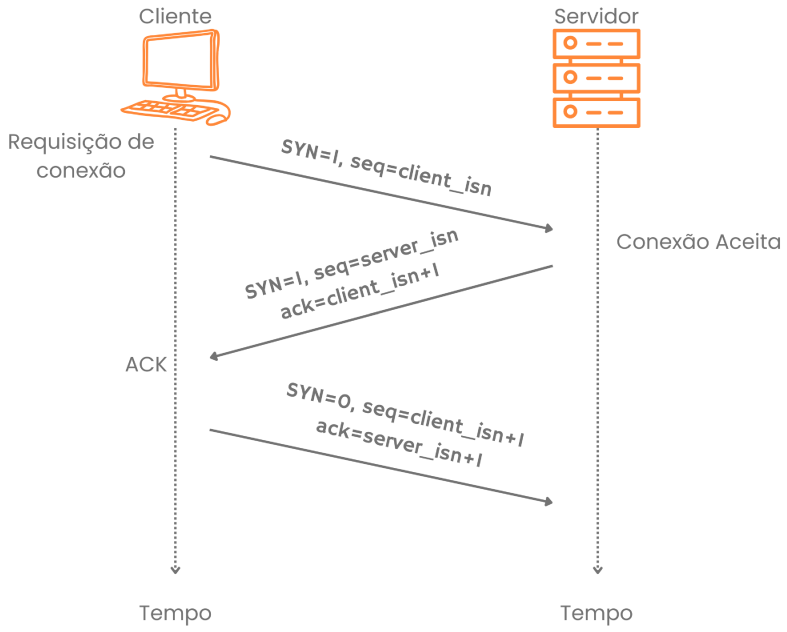


Figura 5 – Apresentação de três vias do TCP entre um cliente e um servidor

Fonte: Adaptado de (KUROSE; ROSS, 2012)

Após a abertura da conexão, os dados são transmitidos em segmentos, cada um com um número de sequência. O receptor confirma o recebimento de cada segmento enviando

um ACK com o número de sequência do próximo segmento esperado. Caso um segmento seja perdido ou danificado, o TCP retransmite o segmento não confirmado (TANENBAUM; WETHERALL, 2010). Algumas características do Protocolo TCP são:

- ❏ Orientada a conexão: Garante a entrega confiável dos dados, estabelecendo uma conexão antes da transmissão, através de um processo de *handshake* que sincroniza os dispositivos e verifica a disponibilidade da rede antes do envio de dados;
- ❏ Orientado a bytes: O TCP divide os dados a serem transmitidos em segmentos de tamanho variável, cada um contendo uma sequência de bytes. Esses segmentos são numerados sequencialmente e enviados pela rede. No destino, o TCP reordena os segmentos e reconstrói os dados originais, garantindo a entrega na ordem correta e sem perdas.
- ❏ Janela deslizante: Atua como um regulador de tráfego, permitindo ao receptor controlar a taxa de envio de dados e evitar sobrecarga na rede. Ao ajustar dinamicamente o tamanho da janela, o receptor indica ao emissor a quantidade de dados que pode processar, garantindo a entrega eficiente e confiável dos pacotes;
- ❏ Controle de erro: Detecta e corrige erros de transmissão através de somas de verificação, como *Cyclic Redundancy Check* (CRC) e *checksum* e retransmissões, garantindo a integridade dos dados em ambientes com ruídos e interferências;

O SYN *scan* é uma técnica de varredura de portas que explora a lógica do protocolo TCP para identificar quais portas de um *host* estão abertas e quais serviços estão em execução. Essa técnica se baseia no fato de que, para estabelecer uma conexão TCP, um cliente envia um pacote SYN ao servidor. Se a porta estiver aberta, o servidor responderá com um pacote SYN-ACK. O cliente, por sua vez, enviará um pacote ACK para finalizar a apresentação de três vias (ERICKSON, 2008).

No caso do SYN *scan*, o cliente envia um pacote SYN para cada porta que deseja verificar, mas não envia o pacote ACK final. Isso faz com que o servidor entre em um estado de espera, abrindo *threads* para cada SYN aceito. Essa técnica é considerada mais discreta do que outras, pois não completa a conexão TCP, o que pode dificultar a detecção pelo sistema de intrusão.

2.8 Camadas da Internet

O conteúdo da Internet pode ser dividido em três camadas (AKHGAR et al., 2021) que são a *Surface Web* a camada onde qualquer pessoa com um navegador convencional possui acesso; a segunda camada é a *Deep Web* que é parte não indexada da *Surface Web*; e a camada mais profunda, a *Dark Web*. Os detalhes de cada camada estão descritos abaixo.

2.8.1 Surface, Dark e Deep Web

A *Surface Web* é a parte da Internet que conhecemos e usamos diariamente. É acessível através de motores de busca tradicionais como `Google`, `Bing` e `Yahoo`. Nela encontramos sites de notícias, redes sociais, lojas online e outros conteúdos indexados e facilmente encontrados(AKHGAR et al., 2021).

A *Deep Web*, por sua vez, engloba um conjunto muito maior de informações que não são indexadas pelos motores de busca tradicionais. Isso inclui conteúdo que reside em bancos de dados privados, intranets corporativas, áreas restritas de sites e outras plataformas que exigem login e senha para acesso (AKHGAR et al., 2021).

A *Dark Web* opera em redes sobrepostas, acessíveis apenas através de softwares específicos como o `Tor` [2]. Nessa camada, a privacidade é elevada ao máximo, o que a torna um ambiente propício para atividades ilícitas, como tráfico de drogas, armas e informações pessoais, além de abrigar comunidades online anônimas (AKHGAR et al., 2021).

2.9 Censys e Shodan

Os motores de busca `Censys`[3] e `Shodan` [4] representam um avanço significativo na forma como é explorada a Internet. Ao mapear a vasta rede de dispositivos conectados, serviços e vulnerabilidades, essas ferramentas oferecem um alto nível de visibilidade sobre a infraestrutura global. No entanto, essa mesma visibilidade que beneficia os pesquisadores de segurança também pode ser explorada por cibercriminosos com intenções maliciosas.

Tanto o `Censys` quanto o `Shodan` funcionam como diretórios extensos da Internet, indexando bilhões de dispositivos e serviços. Essa capacidade de identificar dispositivos específicos, suas configurações e software em execução os torna ferramentas extremamente valiosas para a pesquisa e análise de segurança. Pesquisadores podem utilizar essas plataformas para descobrir vulnerabilidades, mapear a infraestrutura de uma organização e monitorar ameaças emergentes.

[2] https://www.torproject.org/about/history/
[3] https://search.censys.io/
[4] https://www.shodan.io/

CAPÍTULO 3

Trabalhos Relacionados

Este capítulo apresenta uma visão abrangente da pesquisa existente sobre segurança de rede, com ênfase no emprego de *honeypots* e na segurança de redes IPv6. Estrutura-se em duas seções principais: a Seção 3.1 dedica-se à discussão dos *honeypots* implementados em nuvem, enquanto a Seção 3.2 explora as técnicas de varredura do extenso espaço de endereçamento IPv6 utilizadas por atacantes e pesquisadores. Embora existam investigações que inter-relacionam *honeypots* e IPv6, a literatura predominante concentra-se na concepção de sistemas, em detrimento da análise aprofundada de sua implementação e resultados. A Seção 3.3 complementa a exposição com a Tabela 1, consolidando as pesquisas abordadas e o presente trabalho.

3.1 *Honeypot*

A construção do *honeypot* foi elemento fundamental para este trabalho. A compreensão de como estas ferramentas são desenvolvidas e utilizadas no meio acadêmico constituiu parte essencial da pesquisa, razão pela qual os trabalhos relacionados contemplaram seu estudo. A análise focou em implementações em ambiente de nuvem, como demonstrado em Priya e Chakkaravarthy (2023), Kelly et al. (2021), bem como em aplicações com IPv6, conforme apresentado em Schnor e Scheffler (2015).

Em Priya e Chakkaravarthy (2023) foi implementada uma pilha ELK [1] containerizada para melhorar a implantação e análise de *honeypots*. Empregando `Docker`, os pesquisadores conseguiram implantar múltiplos *honeypots* simultaneamente, otimizando a utilização de recursos. A pilha ELK, composta por `Filebeat`, `Logstash` e `Elasticsearch`, facilita a coleta, processamento e análise eficientes de logs. O `Filebeat` coleta logs de várias fontes, incluindo sistemas de *honeypot*, e os encaminha para o `Logstash` para processamento adicional. O `Logstash` limpa, transforma e enriquece os dados antes de indexá-los no `Elasticsearch`. Finalmente, o `Kibana` fornece uma interface para visualizar e analisar os dados indexados.

[1] https://www.elastic.co/elastic-stack

O `framework` foi construído no ambiente de nuvem `Azure` [2]. A infraestrutura consistiu em 12 *honeypots* distribuídos em duas regiões: Índia e Japão, operando durante 10 dias com todas as portas abertas para atrair potenciais atacantes. A análise dos dados focou em ataques DDoS, identificados como o vetor mais prevalente, registrando mais de 2 milhões de tentativas de acesso em um único *honeypot*. O `framework` desenvolvido possibilitou a captura de *malwares*, comandos executados e URL mais requisitadas. A solução demonstrou-se abrangente e eficiente no monitoramento de ataques em tempo real, entretanto, limitou-se ao protocolo IPv4, não contemplando IPv6.

Kelly et al. (2021) examinaram a dinâmica de ameaças em plataformas em nuvem (*Amazon Web Services* (AWS), GCP e Azure) mediante a implementação de *honeypots*. A investigação objetivou compreender as estratégias empregadas pelos cibercriminosos e analisar as particularidades regionais de suas investidas. Mediante o uso de *honeypots* de baixa e média interatividade, simulando variados serviços e protocolos, os pesquisadores mapearam e monitoraram os ataques.

Os achados evidenciaram um significativo volume de ataques, com GCP apresentando-se como o ambiente mais visado. As vulnerabilidades exploradas caracterizaram-se por serem bem conhecidas, como CVE-2006-2369, uma vulnerabilidade no software RealVNC que permite a atacantes contornar a autenticação VNC e CVE-2001-0540, um vazamento de memória em servidores de terminal executando Windows 2000 e Windows NT, permitindo que atacantes remotos causem uma negação de serviço por meio de pacotes RDP malformados propositalmente. Demonstrando que os atacantes continuam aproveitando-se desse tipo de falha. A análise geográfica dos ataques revelou uma origem diversificada, como China, Rússia e Estados Unidos, cuja precisão foi comprometida pelo uso de *Virtual Private Network* (VPN). Adicionalmente, o perfil dos sistemas operacionais dos atacantes sinalizou predominância de versões legadas do Windows e múltiplas distribuições Linux.

As análises revelaram diferenças regionais significativas na atividade dos atacantes. Por exemplo, o *honeypot* Dionaea na Europa recebeu o maior número de ataques, enquanto o `Cowrie` foi mais ativo nas Américas e Ásia. Essas descobertas destacam a importância de monitorar continuamente as ameaças e adaptar as medidas de segurança. A utilização de *honeypots* é uma ferramenta valiosa para as organizações entenderem o cenário de ameaças e fortalecerem sua postura de segurança. Os resultados deste estudo demonstram a necessidade de um gerenciamento proativo de vulnerabilidades e a importância de utilizar inteligência contra ameaças para se proteger contra as táticas em constante evolução dos atacantes.

Schnor e Scheffler (2015) implementaram um *honeypot* configurado para capturar varreduras em IPv6. Utilizou-se uma faixa /34, conduzido durante 15 meses. Mediante análise com `tcpdump`, verificou-se que dos pacotes capturados, 87,56% eram *Internet Control Message Protocol Version 6* (ICMPv6), 12,35% TCP e apenas 0,09% UDP. Observou-se que o trá-

[2] https://azure.microsoft.com/

fego foi majoritariamente composto por varredura TCP SYN, com duas principais modalidades de varredura. A primeira, originada do Instituto de Tecnologia de Karlsruhe, caracterizou-se por varreduras direcionadas exclusivamente a endereços de byte baixo, o byte baixo refere-se ao último byte (2 hexadecimais) do último grupo de um endereço IPv6. A segunda modalidade, desenvolvida por múltiplos atacantes, apresentou padrão mais disperso, mantendo como alvo endereços IPv6 de byte baixo, mas com diferencial de enviar quinze pacotes para cada destino antes de alternar para novos endereços IPv6. Embora tenha sido utilizada uma extensiva faixa de endereços, não se verifica na literatura consultada qualquer referência a procedimentos de divulgação ou comunicação acerca do intervalo de endereços IPv6 selecionado.

3.2 IPv6

O foco das escolhas dos trabalhos relacionados ao IPv6 concentrou-se em metodologias para identificação de endereços. Em Tanveer et al. (2023), a pesquisa direcionou-se à compreensão dos padrões de varredura dos atacantes na rede IPv6. Já em Klein Nijenhuis (2018), Hendriks et al. (2017), Fukuda e Heidemann (2018), a investigação incidiu sobre a infraestrutura de rede DNS, abordando vulnerabilidades, estratégias de localização de servidores DNS e análise de tráfego *backscatter*. Por fim, Dahlmanns et al. (2024) desenvolveu uma abordagem para geração de endereços IPv6 ativos por meio de `scripts` baseados em *hitlists*.

Tanveer et al. (2023) examina as estratégias de varredura de *scanners* IPv6 na Internet. Os pesquisadores conduziram experimentos controlados em uma vasta sub-rede /56 IPv6 não alocados, para analisar o impacto de diferentes atividades de hosts no comportamento de varredura de *scanners* IPv6. A sub-rede /56 foi dividida em 256 sub-redes /64 menores. Destas, 24 foram utilizadas para hospedar diversos serviços que interagiam com a Internet, expondo seus endereços a potenciais varreduras. As demais sub-redes serviram como grupo de controle, isentas de serviços de divulgação.

As 232 sub-redes de controle restantes ficaram completamente sem uso. Os pesquisadores então registraram e analisaram todo o tráfego de varredura de IP e consultas de DNS direcionadas à sub-rede /56.

As principais conclusões deste estudo incluem que as sub-redes que hospedam serviços ativos atraíram 225 vezes mais varreduras de IP e 1,6 vezes mais consultas de DNS reverso em comparação com as sub-redes de controle antes do início dos experimentos. Mesmo após o término da atividade do host, as sub-redes continuaram a receber 426 vezes mais varreduras de IP e 3,7 vezes mais consultas de DNS.

Os serviços com contato direto com servidores voltados para a Internet (navegação na Web, consulta a resolvedores DNS) experimentaram, respectivamente, 50 e 13 vezes mais atividade de varredura em comparação com serviços de contato indireto (ingresso no pool *Network Time Protocol* (NTP), execução de retransmissores *The Onion Router* (TOR)). Os

scanners DNS revelaram-se consideravelmente específicos: os de natureza reversa concentraram sua investigação nas sub-redes que abrigavam serviços ativos, direcionando um número reduzido de varreduras para outras sub-redes. Diferentemente, os *scanners* IP apresentaram um escopo mais abrangente, examinando tanto as sub-redes quanto as de controle.

Em relação as estrategias de varredura, os pesquisadores categorizaram as estratégias de varredura dos *scanners* IPv6 e descobriram que a maioria usou uma das duas abordagens principais – ou varrendo aleatoriamente ou concentrando-se mais em endereços de bytes mais baixos.

Hendriks et al. (2017) destaca a vulnerabilidade de resolvedores DNS abertos em IPv6 a ataques *Distributed Denial-of-Service* (DDoS). Os autores demonstram como o DNS pode ser explorado para amplificar esses ataques, utilizando resolvedores abertos como multiplicadores de tráfego. Para mitigar essa ameaça, os pesquisadores desenvolveram uma metodologia para identificar resolvedores IPv6 abertos. Essa metodologia se baseia na criação de zonas DNS especialmente configuradas, que só podem ser resolvidas por sistemas com capacidade de resolução IPv6 ou por resolvedores IPv4 que também suportem consultas IPv6.

Klein Nijenhuis (2018), investiga métodos para identificar e quantificar resolvedores DNS abertos em redes IPv6. A pesquisa se concentra em duas questões principais: 1) como descobrir resolvedores DNS abertos em IPv6 utilizando técnicas de medição ativa e passiva, além da análise de tráfego de rede; e 2) como avaliar o desempenho e as funcionalidades desses resolvedores.

Para identificar resolvedores abertos que resolvem endereços IPv6, os pesquisadores empregam diversas abordagens. Eles analisam listas de endereços IPv6 provenientes de diversas fontes, como arquivos de zona DNS, listas de domínios Alexa e dados de `traceroute` (OCCUPYTHEWEB, 2018). Além disso, examinam o tráfego de servidores de nomes autorizados para extrair endereços IPv6 de resolvedores DNS.

Os endereços IPv6 identificados são então verificados utilizando a ferramenta `zmap`[3] para identificar aqueles que estão abertos. No entanto, os autores observaram que os resultados do `zmap` nem sempre são precisos, pois muitos resolvedores podem ser reportados como abertos, mesmo que não forneçam serviço recursivo. Para validar os resultados, os pesquisadores realizam uma etapa adicional, consultando um domínio de teste para verificar quais resolvedores são capazes de resolvê-lo corretamente.

Fukuda e Heidemann (2018) concentra-se no uso do *backscatter* de DNS para detectar atividades de varredura em redes IPv6. Os autores adaptam a técnica de *backscatter*, originalmente desenvolvida para IPv4, para o ambiente IPv6, criando regras de classificação para identificar diversos tipos de originadores, como serviços, CDN e servidores de correio.

Os resultados dos experimentos conduzidos pelos autores indicam que hosts IPv6 reagem com menor frequência a varreduras em comparação com hosts IPv4, sugerindo que a vigilância em redes IPv6 é menos intensa. Além disso, a análise de dados do B-Root revelou

[3] https://github.com/zmap/zmap

um aumento gradual na atividade de varredura IPv6 ao longo de seis meses, confirmando a crescente importância dessa ameaça.

O estudo demonstra a eficácia do *backscatter* de DNS como ferramenta para detectar varreduras em IPv6, especialmente considerando a dificuldade de monitorar esse tipo de atividade em redes com um espaço de endereçamento tão vasto. Os autores identificaram diversos tipos de serviços utilizando *backscatter*, incluindo provedores de nuvem, CDN e servidores NTP.

Utilizando três *hitlists* de endereços IPv6 e onze geradores de endereço, a pesquisa de Dahlmanns et al. (2024) identificou 6.658 dispositivos IoT operacionais. A metodologia empregou as referidas *hitlists* como base para os scripts geradores de endereços, a fim de gerar novos endereços IPv6, objetivando ampliar o espectro de possíveis endereços IPv6 ativos, concentrando-se em quatro protocolos característicos de dispositivos IoT: *Advanced Message Queuing Protocol* (AMQP), *Message Queuing Telemetry Transport* (MQTT), *Open Platform Communications Unified Architecture* (OPC UA) e *Constrained Application Protocol* (CoAP). Não obstante a utilização de onze geradores de endereços IPv6, constatou-se que apenas dois, tendo como entrada as três *hitlists*, foram suficientes para gerar 95% dos endereços mapeados, evidenciando-se que somente 6.2% implementaram o protocolo *Transport Layer Security* (TLS) para aprimorar a segurança das aplicações.

3.3 Comparação com trabalhos relacionados

A Tabela 1 sumariza e compara os artigos apresentados nesta seção com o trabalho elaborado na dissertação. A primeira coluna verifica a utilização de endereços IPv6, a segunda avalia a implementação de *honeypot*, a terceira identifica o uso específico do *honeypot* `Cowrie`, a quarta analisa a simulação de serviços Web, a quinta investiga a utilização da GCP e a última examina o emprego de dados DNS.

A implementação da pilha `ELK` em Priya e Chakkaravarthy (2023) foi relevante para compreender as análises dos resultados dos *honeypots*, contribuindo para a construção deste trabalho. Contudo, a pesquisa realizada no artigo não contemplou endereços IPv6 como objetivo de investigação.

O trabalho de Kelly et al. (2021) empregou a pilha `ELK` para análise de dados e realizou uma comparação de diferentes *honeypots* em distintas plataformas de nuvem, o que teve influência na seleção do *honeypot* e da infraestrutura de nuvem neste estudo, contudo não examinou o uso de endereços IPv6.

Schnor e Scheffler (2015) emprega tanto IPv6 quanto *honeypots*, concentrando-se em técnicas de varredura na *darknet*, sem, contudo, abordar as estratégias de divulgação.

O artigo Tanveer et al. (2023) apresenta um estudo abrangente sobre divulgação de endereços IPv6, contudo não emprega *honeypots* e realiza a análise com um intervalo de endereços bastante amplo.

Hendriks et al. (2017) e Klein Nijenhuis (2018) examinam servidores de DNS responsáveis pela resolução de endereços IPv6. As pesquisas forneceram embasamento teórico para a compreensão do funcionamento desses servidores, metodologia que o presente trabalho também adota, embora os estudos anteriores não tenham utilizado *honeypots* nem investigado os servidores como estratégia de divulgação.

Fukuda e Heidemann (2018) evidencia a importância do tráfego *backscatter* na varredura de redes IPv6, configurando-se como um possível método de divulgação, que foi explorado neste trabalho. Contudo, a pesquisa concentra-se exclusivamente nessa forma de obtenção de endereços IPv6 e na análise dos padrões de varreduras.

O trabalho Dahlmanns et al. (2024) propôs a geração de uma *hitlist* de endereços IPv6 para aplicações IoT, o que representa uma estratégia de divulgação de endereços IPv6. No entanto, o escopo do artigo limita-se a essa única abordagem.

Tabela 1 – Tabela comparativa dos trabalhos relacionados citados neste Capitulo e o trabalho aqui desenvolvido

	Utiliza IPv6	Implementa Honeypot	Faz uso do `Cowrie`	Simula serviços Web	Utiliza GCP	Faz uso de DNS
Priya e Chakkaravarthy (2023)	Não	Sim	Não	Sim	Não	Não
Kelly et al. (2021)	Não	Sim	Sim	Sim	Sim	Não
Schnor e Scheffler (2015)	Sim	Sim	Não	Sim	Não	Não
Tanveer et al. (2023)	Sim	Sim	Não	Sim	Não	Sim
Hendriks et al. (2017)	Sim	Não	Não	Não	Não	Sim
Klein Nijenhuis (2018)	Sim	Não	Não	Não	Não	Sim
Fukuda e Heidemann (2018)	Sim	Não	Não	Não	Não	Sim
Dahlmanns et al. (2024)	Sim	Não	Não	Não	Não	Sim
Este estudo	Sim	Sim	Sim	Sim	Sim	Sim

Fonte: Autor

CAPÍTULO 4

Materiais e métodos

Esse Capítulo aborda os métodos e as técnicas utilizadas para a construção do projeto, além de todos os passos que foram utilizados para o desenvolvimento e a coleta de dados. O capítulo foi separado em três Seções.

A Seção 4.1 apresenta os dois experimentos propostos. O primeiro experimento, Experimento A – Divulgação em motores de busca e fóruns e o segundo experimento, Experimento B – Divulgação *Crawler*, DNS e WAN, utilizando a infraestrutura da Internet para fazer a divulgação dos endereços IPv6.

A Seção 4.2 apresenta os oito passos para a construção do experimento: a escolha do provedor de serviço de nuvem (Seção 4.2.1), a configuração das instâncias (Seção 4.2.2), a configuração dos *honeypots* (Seção 4.2.3), a configuração da rede IPv6 (Seção 4.2.4), a configuração das regras de *firewall* (Seção 4.2.5), a configuração dos serviços que irão funcionar nas instâncias (Seção 4.2.6), a configuração do sistema de logs (Seção 4.2.9) e, por fim, o início do projeto (Seção 4.2.10).

A Seção 4.3, detalha os métodos utilizados para a divulgação dos endereços IPv6 das instâncias, sendo eles: divulgação motores de busca 4.3.1, divulgação Fóruns 4.3.2, divulgação *Crawler* (Seção 4.3.3), divulgação DNS (Seção 4.3.4) e por fim divulgação WAN (Seção 4.3.5).

4.1 Descrição dos experimentos

Este trabalho apresenta dois experimentos complementares que investigam métodos para a divulgação de endereços IPv6 na Internet. O objetivo é estudar se os métodos testados são eficientes como formas de divulgação.

O Experimento A – Divulgação em motores de busca e fóruns, contemplou duas estratégias de divulgação: a publicação explícita dos endereços IPv6 em fóruns da Internet e a tentativa de inserção dos endereços IPv6 nas bases de dados dos motores de busca `Censys` e `Shodan`, para isso foram criados três grupos de divulgação, sendo dois grupos para cada métodos de divulgação, motores de busca e fóruns, respectivamente e um grupo de controle

conforme mostrado na Figura 6. Maiores detalhes do Experimento A se encontram na Seção 4.2.6.

Utilizando o GCP foram criadas 15 instâncias distribuídas em cinco regiões geográficas distintas, conforme a Figura 6. Cada região continha duas instâncias, cada uma com um método de divulgação específico e uma instância controle que não teve endereço IPv6 divulgado. Cada instância operava com três serviços distintos: o Serviço de divulgação, o Serviço de *honeypot* e o Serviço de log, conforme detalhado na Seção 4.2.6 e apresentado na Tabela 2.

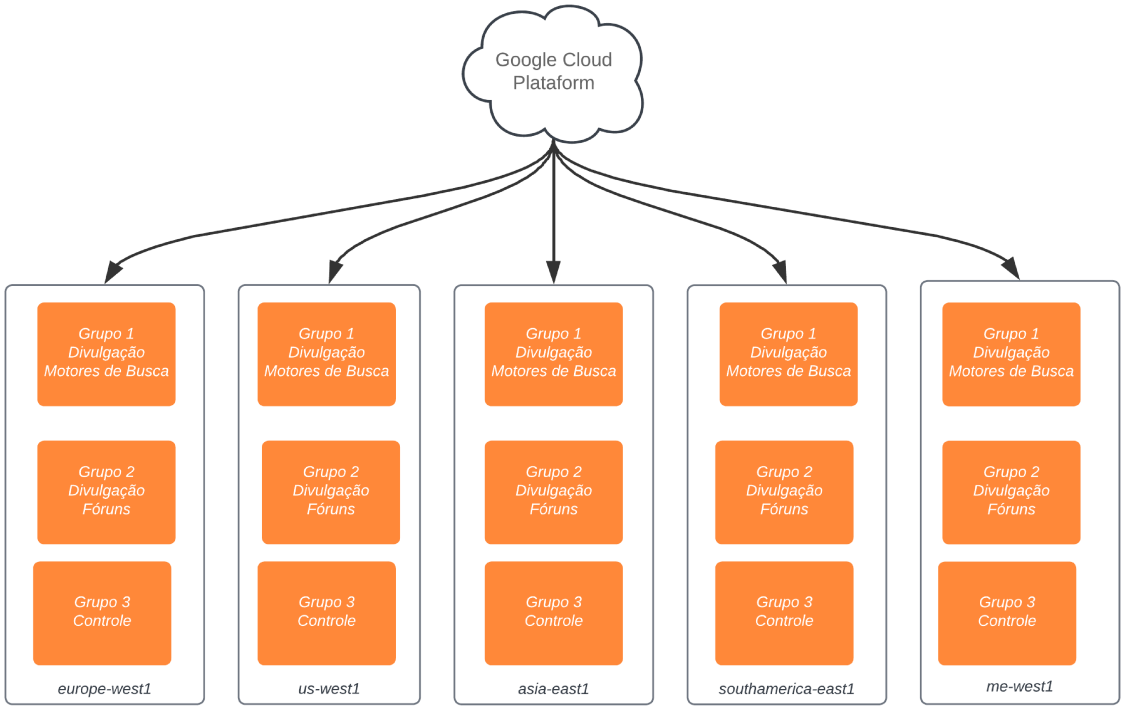


Figura 6 – Diagrama de montagem das instâncias separadas por Grupos e por regiões dentro do GCP para o Experimento A.

Fonte: Autor

O Experimento B – Divulgação *Crawler*, DNS e WAN, assim como o anterior, também objetivou estudar formas eficientes de divulgação de endereços IPv6 na Internet. As formas de divulgação analisadas foram *Crawler*, DNS e WAN, sendo assim foram criados quatro grupos de divulgação, sendo três para os métodos de divulgação citados e o quarto grupo sendo de controle conforme Figura 7. Mais detalhes do Experimento estão na Seção 4.2.6. Para o Experimento B, foram criadas 20 instâncias distribuídas nas mesmas cinco regiões, contendo em cada região uma instância para cada método de divulgação e uma instância de controle, conforme a Figura 7, ou seja quatro instâncias por região.

Para cada instância do Experimento B, foram utilizados um serviço de divulgação, dois serviços *honeypot*, o `Cowrie` e outro *honeypot* desenvolvido em `Python` para simulação de

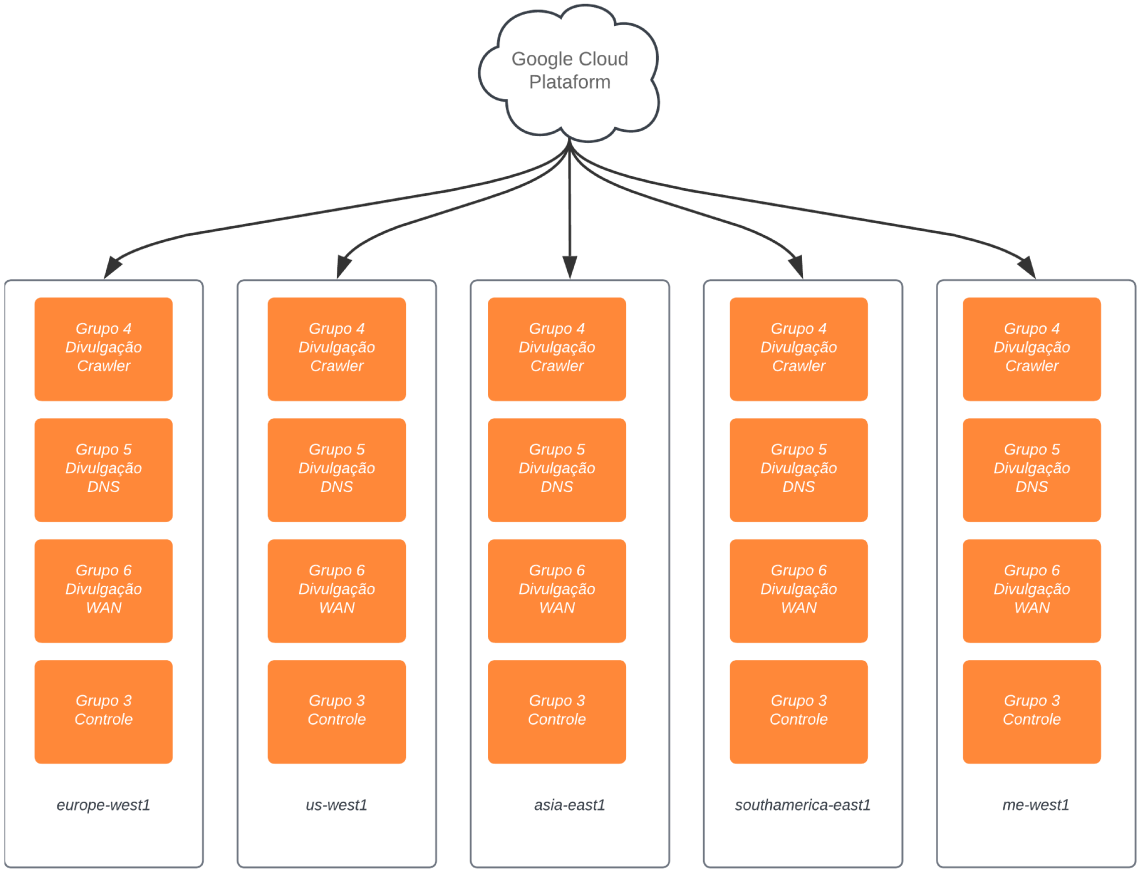


Figura 7 – Diagrama de montagem das instâncias separadas por grupos e por regiões dentro do GCP para o Experimento B.

Fonte: Autor

serviços Web - além do serviço de log, conforme detalhado na Seção 4.2.6 e apresentado na Tabela 2. O código-fonte se encontra no Github (MUNDIM, 2024). Os dois experimentos foram executados no ambiente GCP, e suas descrições detalhadas encontram-se na Seção 4.2.

O *framework* criado para o Experimento B foi chamado de `Trigona` que é o nome científico de uma abelha `Trigona spinipes`. Detalhes sobre o funcionamento do framework podem ser encontrados no Github[1] do projeto.

4.2 Elaboração do ambiente experimental

A Figura 8 apresenta o processo de criação da infraestrutura do sistema, dividido em oito etapas, que foi comum aos dois Experimentos. Após a fase inicial, os experimentos se diferenciaram pelos seus respectivos métodos de divulgação. A implementação da infraestrutura abrangeu desde a seleção do provedor de serviço em nuvem até o início do experimento.

[1] https://github.com/Luanmundim/trigona

Toda a configuração foi executada de forma automatizada por meio de `scripts` em `Python`, os quais estabeleciam conexão com os servidores da Google utilizando `Google Cloud CLI` [2].

4.2.1 Passo 1 - Escolha do provedor de serviço na nuvem

A infraestrutura necessária para a implantação das instâncias baseou-se em uma plataforma de nuvem, utilizando um serviço *Infrastructure as a Service* (IaaS). Esta escolha eliminou preocupações com aspectos físicos do sistema, permitindo concentrar esforços exclusivamente na implementação dos serviços.

O provedor de serviço em nuvem escolhido foi o Google Cloud Platform [3] por dois motivos. Primeiro, entre os provedores analisados, como a AWS da Amazon [4] e o Azure da Microsoft [5], o Google era o único que possuía a opção de segregar o tráfego IPv4 e IPv6, fator crítico para a criação deste projeto. Os outros provedores apenas faziam o balanceamento de tráfego entre os protocolos de Internet, o que tornava possível encontrar as instâncias virtuais através do IPv4. O segundo motivo foi o projeto Google Cloud for Education [6], que disponibiliza créditos a professores com o objetivo de incentivar a pesquisa. Esses créditos concedidos foram um incentivo para que o projeto se mantivesse no ar por mais tempo.

4.2.2 Passo 2 - Configuração das instâncias

Após a escolha da plataforma de nuvem, o próximo passo foi a configuração das instâncias virtuais para o funcionamento do experimento. Após alguns testes de desempenho, o modelo de instância escolhido foi o *E2-small*, que possui 2 GB de memória RAM e 40 GB de disco. A imagem utilizada foi *debian-11-bullseye-v20231113*. Em relação ao processador, as instâncias escolhidas possuíam vCPU compartilhado, que poderia variar desde 0,5 até 2 vCPU, dependendo do consumo da instância.

No Experimento A, realizou-se o redirecionamento das portas 22 e 23 para 2222 e 2223, respectivamente. Já no Experimento B, além dessas alterações, as portas 80 e 443 foram redirecionadas para 8080 e 4443. Tais modificações na tabela de roteamento foram implementadas devido às limitações do sistema operacional em estabelecer novos serviços nas portas padrão[7].

As instâncias foram distribuídas em cinco regiões geográficas, conforme ilustrado nas Figuras 6 e 7. A seleção abrangeu Estados Unidos, Europa, Ásia, América do Sul e Oriente Médio, visando uma cobertura global significativa. A nomenclatura das instâncias seguiu o padrão "região-ipv6-tipo de divulgação", exemplificado por `europe-west1-ipv6-control`.

[2] https://cloud.google.com/sdk/docs/install
[3] https://cloud.google.com
[4] https://aws.amazon.com
[5] https://azure.microsoft.com
[6] https://cloud.google.com/edu/faculty
[7] https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt

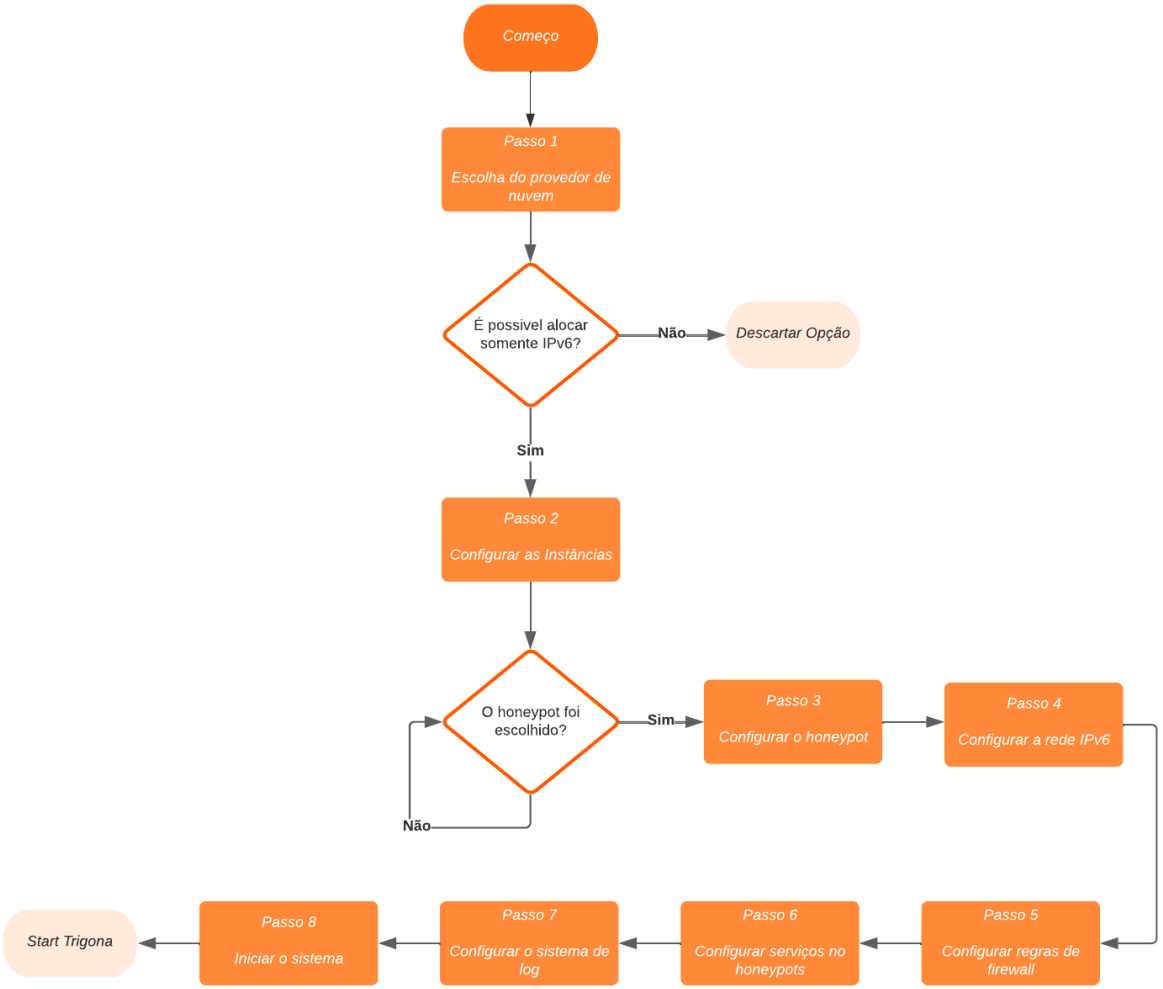


Figura 8 – Fluxograma de criação da infraestrutura que foi comum para a realização dos Experimentos A e B até o passo oito, início do sistema de *honeypot* apelidado de `trigona`.

Fonte: Autor

4.2.3 Passo 3 - Escolha do *honeypot*

O terceiro passo consistiu na seleção do *honeypot*. A primeira ferramenta avaliada foi o `T-pot` [8], uma solução robusta e completa que disponibiliza mais de 20 *honeypots* nativamente integrados, com suporte para Linux, MacOS e Windows. No entanto, devido aos seus requisitos computacionais elevados - mínimo de 8GB de memória RAM e 128GB de armazenamento - esta opção foi descartada.

O segundo *honeypot* analisado foi o `Qeeqbox` [9], uma ferramenta ágil e eficiente desenvolvida em `Python`, capaz de simular 30 serviços de baixa, média e alta interação. No entanto,

[8] https://github.com/telekom-security/tpotce
[9] https://github.com/qeeqbox/honeypots

devido à ausência de suporte a IPv6, esta opção também foi descartada. A escolha final recaiu sobre o `Cowrie` 2.3.4, ferramenta leve e de fácil configuração que oferece suporte a IPv6. Como o `Cowrie` limita-se à simulação de serviços de conexão remota SSH e Telnet, e havia a necessidade de simular serviços Web para o Experimento B, desenvolveu-se um *honeypot* complementar em `Python` para emulação de páginas *HyperText Transfer Protocol* (HTTP) e *HyperText Transfer Protocol Secure* (HTTPS). Os `scripts` desenvolvidos encontram-se disponíveis no repositório (MUNDIM, 2024).

4.2.4 Passo 4 - Configuração da rede IPv6

O passo quatro foi a configuração da rede IPv6. Foram alocadas 15 sub-redes /64 para o Experimento A e 20 sub-rede /64 para o Experimento B, sendo uma para cada instância.

Os endereços de sub-rede foram gerados de forma automática pelo `Google Cloud` e cada instância recebia o primeiro endereço válido dentro de cada sub-rede, que era o objetivo visto que como apresentado no artigo Tanveer et al. (2023), os dois principais padrões de varredura realizados pelos atacantes é ou fazendo uma busca aleatória dentro do espaço analisado ou buscando os endereços de menor IID.

4.2.5 Passo 5 - Configuração das regras de *firewall*

Foram criadas duas regras de *firewall* para acesso às instâncias. A primeira regra, referente aos endereços IPv6, foi configurada sem nenhum tipo de filtro, permitindo o acesso a qualquer porta das instâncias a partir de qualquer endereço IPv6.

A segunda regra de *firewall*, por sua vez, diz respeito aos endereços IPv4. Os filtros estabelecidos nessa regra permitiam o acesso às instâncias somente a partir do endereço IPv4 da máquina responsável pela coleta de logs, bloqueando qualquer outro tipo de acesso.

4.2.6 Passo 6 - Configuração de serviços nos *honeypots*

Com as instâncias criadas e configuradas, o próximo passo foi a configuração dos serviços. Os serviços criados foram separados em dois tipos: i) Serviços de divulgação e ii) Serviços de *honeypot*.

Os serviços de divulgação foram desenvolvidos utilizando `scripts` em `Python` e `shell` [10]. A implementação foi estruturada em três componentes principais. Um agendamento na `crontab` do sistema operacional executava o `script` orquestrador desenvolvido em `Python`. O orquestrador acionava o `script shell`, que por sua vez era responsável por iniciar os serviços de divulgação e *honeypot* construídos tambem em `Python`, conforme ilustrado na Figura 9. A Tabela 2 apresenta os serviços em execução nas instâncias, seus tipos (Serviço de

[10] https://www.shellscript.sh

divulgação ou *honeypot*), seus respectivos grupos, conforme Figuras 6 e 7, e os experimentos em que foram utilizados.

Tabela 2 – Serviços iniciados funcionando nas instâncias agrupados por experimento. O Grupo 3 oculto da tabela é o grupo das instâncias controle.

Serviço	Tipo	Grupo	Experimento
Divulgação motores de busca	Serviço de Divulgação	Grupo 1	Experimento A
Divulgação Fóruns	Serviço de Divulgação	Grupo 2	Experimento A
Divulgação *Crawler*	Serviço de Divulgação	Grupo 4	Experimento B
Divulgação DNS	Serviço de Divulgação	Grupo 5	Experimento B
Divulgação WAN	Serviço de Divulgação	Grupo 6	Experimento B
Serviço Web	Serviço honeypot	-	Experimento B
Serviço Cowrie	Serviço honeypot	-	Experimento A/B

Fonte: Autor

4.2.7 Serviços de divulgação

O Grupo 1 foi responsável pela divulgação em motores de busca (`Shodan` e `Censys`), operando duas vezes ao dia. O Grupo 2 realizou a divulgação em fóruns uma única vez, no início dos experimentos. Os Grupos 4, 5 e 6 executaram os serviços *Crawler*, DNS e WAN, respectivamente, com frequência de duas vezes ao dia.

4.2.8 Serviços de *honeypot*

Foram configurados dois tipos de serviços de *honeypot*. Primeiramente, foram implementados serviços de SSH e Telnet utilizando o *honeypot* `Cowrie`. Estes serviços estiveram disponíveis e acessíveis a todos os grupos.

Em segundo lugar, foram desenvolvidos *honeypots* de serviço Web. Para tanto, foi criado um `script` em `Python` que gera uma página Web, simulando um servidor Web nas portas 80 e 443 (HTTP e HTTPS, respectivamente). A página apresenta campos para nome de usuário e senha, sendo que, independentemente dos valores inseridos, o usuário é redirecionado para uma página de erro, indicando credenciais inválidas. A Figura 10 mostra a página de login do sistema, o *honeypot* Web, só foi implementado no Experimento B.

4.2.9 Passo 7 - Configuração do sistema de log

Durante todo o funcionamento do experimento, foram realizadas coletas de logs diárias às 00:00 para todos os serviços citados na Seção 4.2.6. Para os serviços de divulgação motores de busca do Experimento A, foram coletados os resultados de saída das consultas realizadas em cada um dos motores de busca. Para o serviço de divulgação nos Fóruns, também do

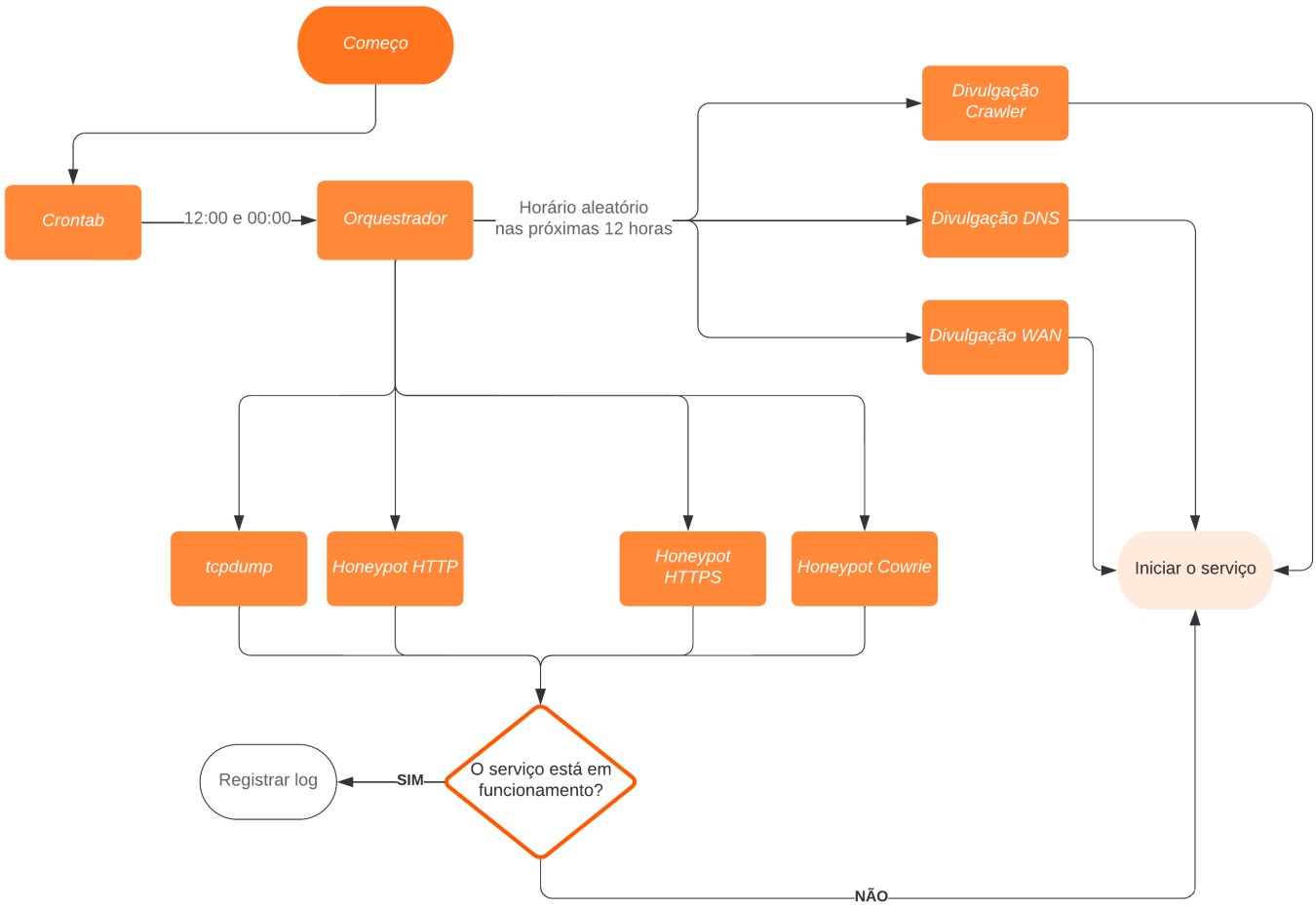


Figura 9 – Fluxograma do `script` de serviços de divulgação, serviços de *honeypot* e coleta de logs.

Fonte: Autor

Experimento A, foram coletadas as respostas dos questionamentos dos fóruns no final do experimento com o objetivo de ter evidência de possível acesso as instâncias.

Para os serviços de divulgação *Crawler* e WAN do Experimento B, foram coletados os códigos de status da requisição HTTP para garantir que as requisições foram bem-sucedidas. Por fim nos serviços de divulgação DNS do Experimento B, foram registrados os endereços IP das consultas DNS, também para garantir que as consultas foram realizadas.

No caso dos serviços de *honeypot*, para o `Cowrie`, foram coletados os logs gerados pelo próprio *honeypot*[11] e nos *honeypots* de serviço Web, foram coletados os logs conforme mostra a Lista 4.2.9. A Figura 11 mostra a captura de tela de um log registrado do serviço de *honeypot* Web.

Lista dos logs gerados no serviço de *honeypot* Web:

[11] https://adhdproject.github.io/#!Tools/Annoyance/Cowrie.md

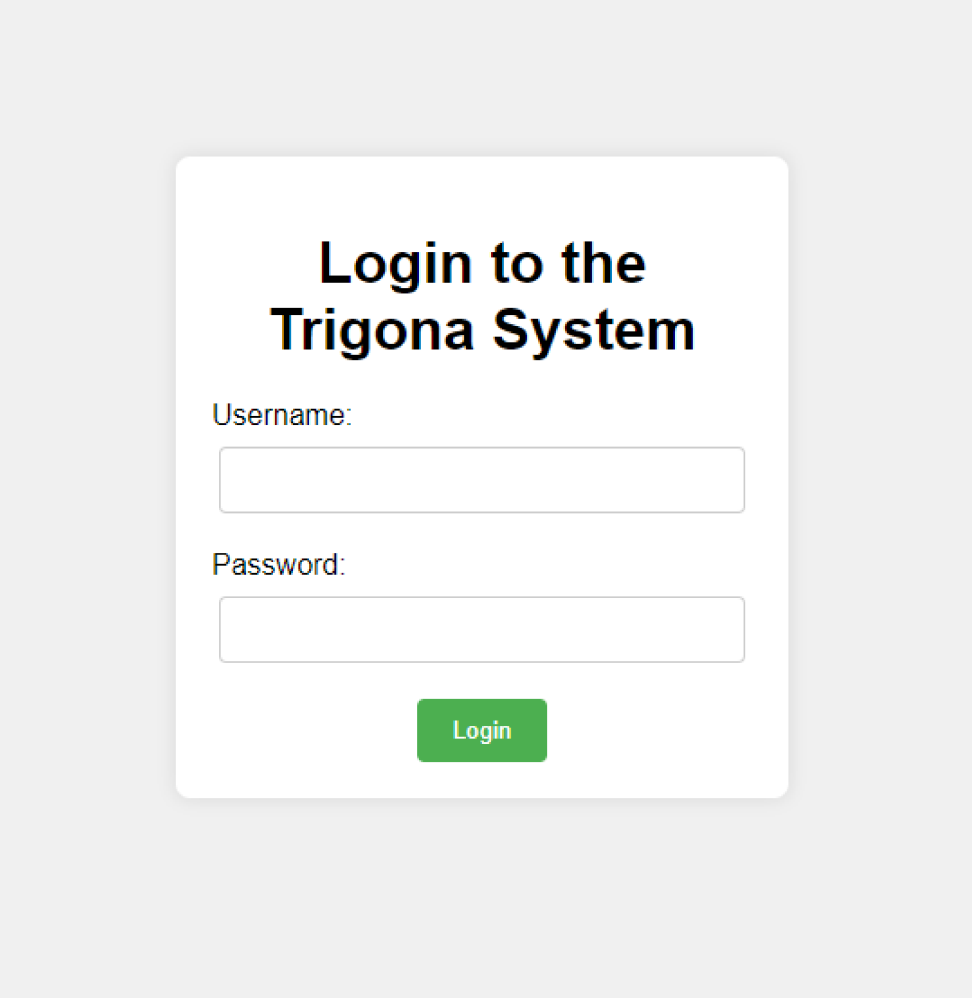


Figura 10 – Página de login criado para o serviços Web para o Experimento B

Fonte: Autor

1. Usuário;
2. Senha;
3. User-Agent do navegador;
4. Método HTTP;
5. Endereço IP de origem;
6. Endereço IP de destino;
7. Porta de origem;
8. Porta de destino;
9. URL;

10. Timestamp.

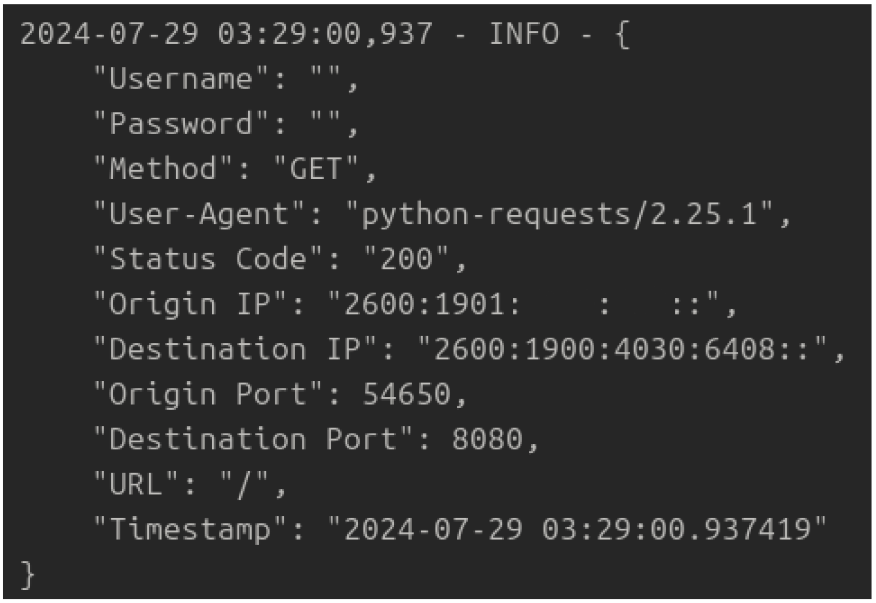


Figura 11 – Exemplo de log de serviço *honeypot* Web do Experimento B recolhido com endereço de IP de origem anonimizado

Fonte: Autor

Além dos logs citados acima, também foi usado o `tcpdump` [12], ferramenta que captura e analisa todos os pacotes de rede recebidos e transmitidos em uma conexão e os salva em um arquivo `pcap`. Dessa forma todos os pacotes que trafegaram pela interface IPv6 das instâncias foram coletados e posteriormente tratados usando o `Wireshark` [13] e `Python`.

4.2.10 Passo 8 - Inicio do experimento

O Experimento A teve início em 06/10/2023 e finalizou em 11/11/2023, totalizando 37 dias de duração. Por sua vez, o Experimento B funcionou por dez semanas, desde 14/07/2024 até 25/09/2024. Todas as instâncias foram criadas e mantidas em modo de suspensão até o início dos respectivos experimentos.Durante todo o período em que o Experimento A esteve em execução, os serviços de divulgação por meio de motores de busca estiveram em funcionamento. Já no Experimento B, os serviços de divulgação operaram apenas nas cinco primeiras semanas, realizando a divulgação dos sistemas duas vezes ao dia. Após esse período, os serviços de divulgação foram encerrados em 22/08/2024, porém as instâncias permaneceram ativas por mais 5 semanas, sendo finalizadas em 25/09/2024. Assim, as durações dos experimentos podem ser resumidas da seguinte forma:

[12] https://www.tcpdump.org/manpages/tcpdump.1.html
[13] https://www.wireshark.org

- ❏ `Experimento A:` Início em 06/10/2023, Término em 11/11/2023, Duração de 37 dias.
- ❏ `Experimento B:` Início em 14/07/2024, Término em 25/09/2024, Duração de 74 dias.

4.3 Métodos de divulgação

A diferença fundamental entre os Experimentos A e B residiu no método de divulgação empregado em cada um deles. No Experimento A, foram constituídos três grupos de instâncias, sendo dois grupos de divulgação e um grupo de controle. Já no Experimento B, foram estabelecidos quatro grupos, sendo três grupos de divulgação e um grupo de controle. Em ambos os casos, um grupo era composto por uma instância em uma região distinta, conforme ilustrado nas Figuras 6 e 7, referentes aos Experimentos A e B, respectivamente.

4.3.1 Divulgação motores de busca

Para o método de divulgação de motores de busca foram realizadas consultas dos endereços IPv6 das instâncias do Grupo 1 utilizando as *Application Programming Interface* (API)'s dos motores de busca `Cencys` [14] e `Shodan` [15]. As buscas eram realizadas através de um `script` que fazia uma consulta para cada endereço IPv6 que deveria ser divulgado. O mecanismo de consulta dos dois motores de busca são extremamente parecidos, o código que foi utilizado para o `Shodan` e `Censys` estão nos Apêndices C e D respectivamente. A expectativa era que os endereços IPv6 consultados fossem indexados nos bancos de dados desses serviços, expondo-os à comunidade *hacker* ou qualquer pessoa/serviço que tenha interessa em encontrar esse tipo de dispositivo.

4.3.2 Divulgação fóruns

No método de divulgação Fóruns os endereços IPv6 foram divulgados em fóruns de *surface Web* e *dark Web*, com o objetivo de atrair a atenção de atacantes. Foram realizadas perguntas relacionadas a endereços IPv6 em fóruns. Três perguntas foram publicadas na darknet (`Hidden Answers`, `Raddle` e `Deep Answers`), e outras três perguntas em fóruns da *Surface Web* (`Nulled`, `Black Hat World` e `Hacknology`). Os fóruns foram escolhidos por serem movimentados e possuírem tópicos relacionados a cibersegurança. Todas as perguntas que foram realizadas nos fóruns estão no Apêndice E.

4.3.3 Divulgação *crawler*

A partir da lista dos 1 milhão de sites mais acessados, compilada pela Cisco Umbrella [16], foram coletados os primeiros 2000 sites com acesso IPv6. Essa coleta foi realizada em uma

[14] https://search.censys.io/api
[15] https://developer.shodan.io
[16] https://s3-us-west-1.amazonaws.com/umbrella-static/index.html

instância de testes para gerar a lista de divulgação. A proposta consiste em simular a navegação de um usuário por diversos *websites*, objetivando a divulgação de endereços IPv6, os quais, quando acessados, ficam registrados nos cabeçalhos dos pacotes e, subsequentemente, nos bancos de dados dos servidores, que podem eventualmente compartilhar essa informação.

Posteriormente, foi desenvolvido um `script` em `Python` [17] para realizar requisições HTTP duas vezes ao dia, uma vez no início e outra no final do dia, em horários aleatórios. A ferramenta `wget` [18] foi utilizada para receber arquivos HTTP durante essas requisições. Todos os `scripts` estão disponíveis no repositório (MUNDIM, 2024).

4.3.4 Divulgação DNS

Para o serviço de divulgação DNS, foi compilada, através dos sites public-dns.info [19] e publicdns.xyz [20], uma lista de todos os servidores DNS abertos. Utilizando a instância de testes, foi criada uma nova lista contendo 2000 servidores DNS que resolviam uma consulta "AAAA" para o site google.com, ou seja, quais são os endereços IPv6 associados ao endereço google.com. A proposta de divulgação visa gerar tráfego nos servidores DNS, de modo que o endereço IPv6 responsável pela consulta seja registrado em bancos de dados ou identificado como tráfego *backscatter* (FUKUDA; HEIDEMANN; QADEER, 2017).

Da mesma forma que o serviço de divulgação *Crawler*, foi desenvolvido um `script` em `Python` para fazer consultas DNS "AAAA" para o endereço google.com duas vezes ao dia, uma vez nas primeiras 12 horas do dia e outra vez nas últimas 12 horas, de forma aleatória dentro desse intervalo de tempo.

4.3.5 Divulgação WAN

O serviço de divulgação WAN teve comunicação exclusivamente com as instâncias pertencentes ao grupo 3, simulando uma *Wide Area Network* [21]. Seu modo de funcionamento é similar ao método de divulgação *Crawler*, porém a lista de divulgação era composta apenas pelos endereços IP das outras quatro instâncias pertencentes ao Grupo 3, uma vez que todas as instâncias desse grupo possuíam os serviços de *honeypot* HTTP e HTTPS funcionando. Este método de divulgação visa deixar vestígios do endereço IP nos cabeçalhos dos pacotes TCP, ao percorrer diversos enlaces de rede, com o propósito de validar o potencial compartilhamento desses dados.

Assim como o serviço de divulgação Web e o serviço de divulgação *Crawler*, a consulta era realizada duas vezes ao dia: uma vez nas primeiras 12 horas e outra vez nas 12 horas

[17] https://www.python.org/doc/
[18] https://www.gnu.org/software/wget/
[19] https://public-dns.info
[20] https://www.publicdns.xyz
[21] https://www.cisco.com/c/en/us/products/switches/what-is-a-wan-wide-area-network.html

restantes, em um horário aleatório dentro desse intervalo.

4.3.6 Grupo controle IPv6

Com o intuito de controle, foi criado também um grupo com instâncias IPv6 cujos endereços IP não foram divulgados. A finalidade de criar o Grupo de Controle foi servir como comparação em relação aos grupos de divulgação, visto que se esperava que este grupo não fosse detectado. Assim como os grupos de divulgação, o grupo de controle possuía os serviços *honeypot* e serviço log em funcionamento, sendo que apenas os serviços de divulgação não estavam ativos neste grupo.

CAPÍTULO 5

Resultados

Neste capítulo serão apresentados os resultados de ambos os experimentos, acompanhados de análise e discussão. A estrutura do capítulo divide-se em quatro seções. Na seção 5.1, é apresentada uma visão geral dos experimentos A e B. Em seguida, os dados do experimento A são analisados na seção 5.2, seguidos da análise dos dados do experimento B na seção 5.3. Por fim, na seção 5.4, é realizada uma discussão abrangente de todos os resultados obtidos.

5.1 Visão geral dos resultados

Nos dois experimentos realizados, foram configuradas 35 instâncias ao todo. Dentre essas, uma em cada experimento registrou pelo menos uma tentativa de acesso. Ambos os métodos utilizados nas instâncias não obtiveram êxito na divulgação de seu endereço IPv6, ou seja a instância foi encontrada por atacantes, porém, não foi encontrada pelo método de divulgação que foi utilizado demonstrando a dificuldade da referida tarefa.

5.1.1 Experimento A - divulgação em motores de busca e fóruns

No Experimento A, apenas uma instância de todas as 15 recebeu algum tipo de tentativa de acesso, por exemplo uma tentativa de estabelecimento de conexão em uma determinada porta. Caso o atacante inserisse um usuário e senha nos serviços disponibilizados, essa tentativa será chamada de conexão bem-sucedida.

Somente a instância `asia-east1-ipv6-MotorBusca` do Grupo 1 de divulgação em motores de Busca foi acessada. Foram realizadas 106 tentativas de acesso nas portas 22 e 23, porém, consultando os logs gerados, nenhum dos dois mecanismos de busca, `Shodan` e `Censys`, retornou informações sobre os serviços que estavam funcionando nos endereços IP, o que confirma o insucesso da tentativa de inseri-los artificialmente em seus bancos de dados. Os detalhes dos resultados do Experimento A estão dentro da Seção 5.2.

5.1.2 Experimento B - Divulgação *crawler*, DNS e WAN

No Experimento B, de forma similar, das 20 instâncias implementadas, apenas uma obteve algum tipo de tentativa de acesso. A instância `europe-west1-ipv6-control` do grupo de controle 4.3.6 teve tentativa de acesso em todos os serviços, com o `Cowrie` nas portas 22 e 23 e o *honeypot* Web pelas portas 80 e 443. Além disso, outras tentativas de acesso foram capturadas pelo `tcpdump` 4.2.6. Os detalhes dos resultados obtidos no Experimento B estão na Seção 5.3.

5.2 Resultados - Experimento A

Os resultados obtidos durante o Experimento A foram organizados em duas Seções. A Seção 5.2.1 apresentou os resultados das tentativas de acesso, quando o atacante não tentou fazer login no `Cowrie`. A Seção 5.2.2 apresentou os dados das conexões bem-sucedidas, quando o atacante de fato fez login no `Cowrie`.

5.2.1 Tentativas de acesso

Somente a instância `asia-east1-ipv6-MotorBusca` foi alvo de tentativas de acesso dentre todas as 15 instâncias que foram colocadas em operação para o experimento. Embora a instância acessada fizesse parte do Grupo 1 - Divulgação de motores de Busca, de acordo com os logs que foram gerados, tanto o `Shodan` quanto o `Censys` falharam em fornecer dados sobre o endereço IP dentro da ferramenta, ou seja, nenhuma das consultas dos `scripts` de divulgação de motores de Busca gerou um resultado positivo com relação aos serviços que estavam funcionando na instância.

A Figura 12 ilustra as tentativas de acesso durante o experimento. A primeira tentativa aconteceu no dia 07/10/2023, um dia após o inicio do experimento uma tentativa de acesso Telnet na porta 23 realizada pela “Hurricane Electric LLC”. ilustra como essas tentativas de acesso estão distribuídas ao longo do experimento.

Foram realizadas 106 tentativas de acesso. Dessas, 63 tiveram origem no protocolo Telnet e as outras 43 no protocolo SSH, conforme representado na Figura 13. As tentativas de acesso partiram de 52 endereços IP diferentes, pertencentes a cinco organizações em quatro países distintos. A “Alibaba Private Limited” opera a partir de Singapura. A “Hurricane Electric LLC” dos Estados Unidos, enquanto que tanto a “Aliyun Computing Co. Ltd” quanto a “China Next Generation Internet CERNET2” são originárias da China. Por fim, a “Constant Company LLC” está situada na Coreia do Sul. Notavelmente, todas essas empresas fornecem serviços de computação em nuvem. A “Hurricane Electric LLC” inclusive oferece como serviço ferramentas para varredura de rede [1].

[1] https://networktools.he.net

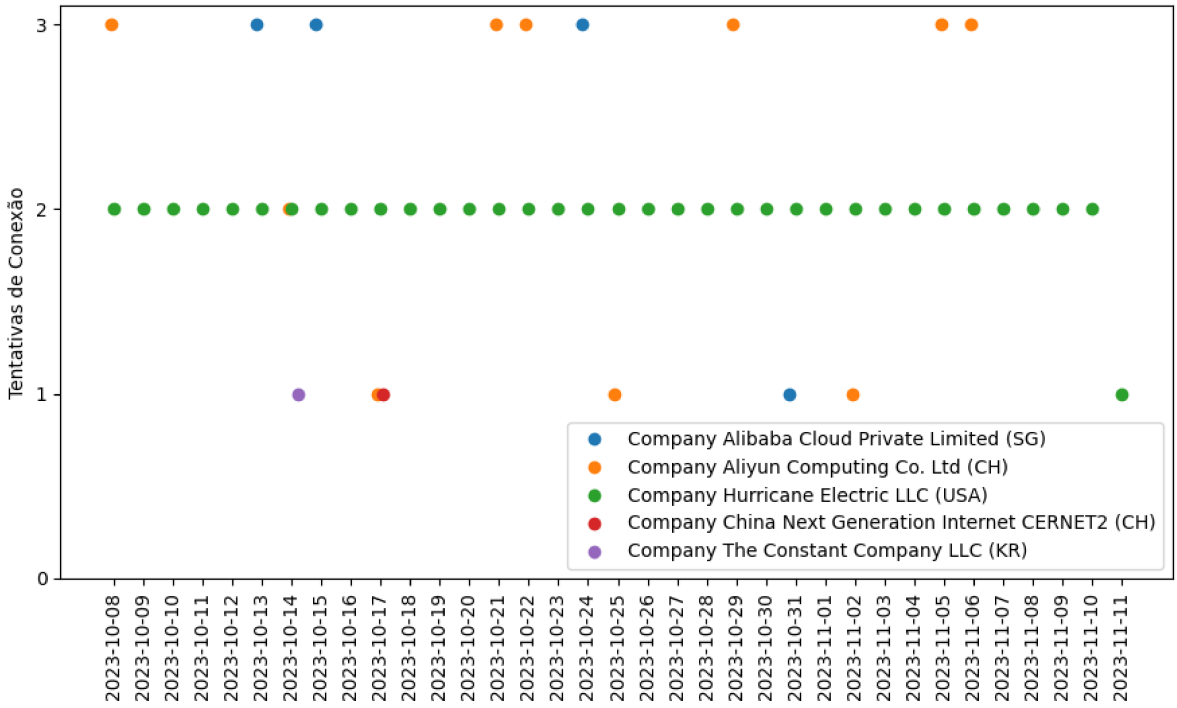


Figura 12 – Tentativa de acessos únicas ao longo do experimento A.

Fonte: Autor

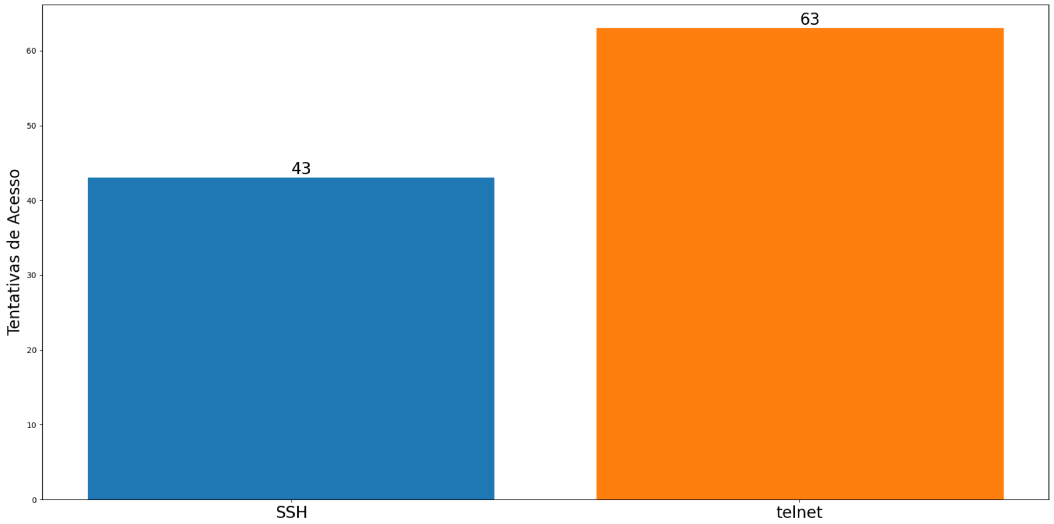


Figura 13 – Número de tentativas de acesso ao `Cowrie` por protocolo no Experimento A

Fonte: Autor

Posteriormente, todos os dias até o fim do experimento, houve pelo menos uma tentativa de acesso. A maior parte das tentativas de acesso, vieram dos Estados Unidos como mostra a Figura 14.

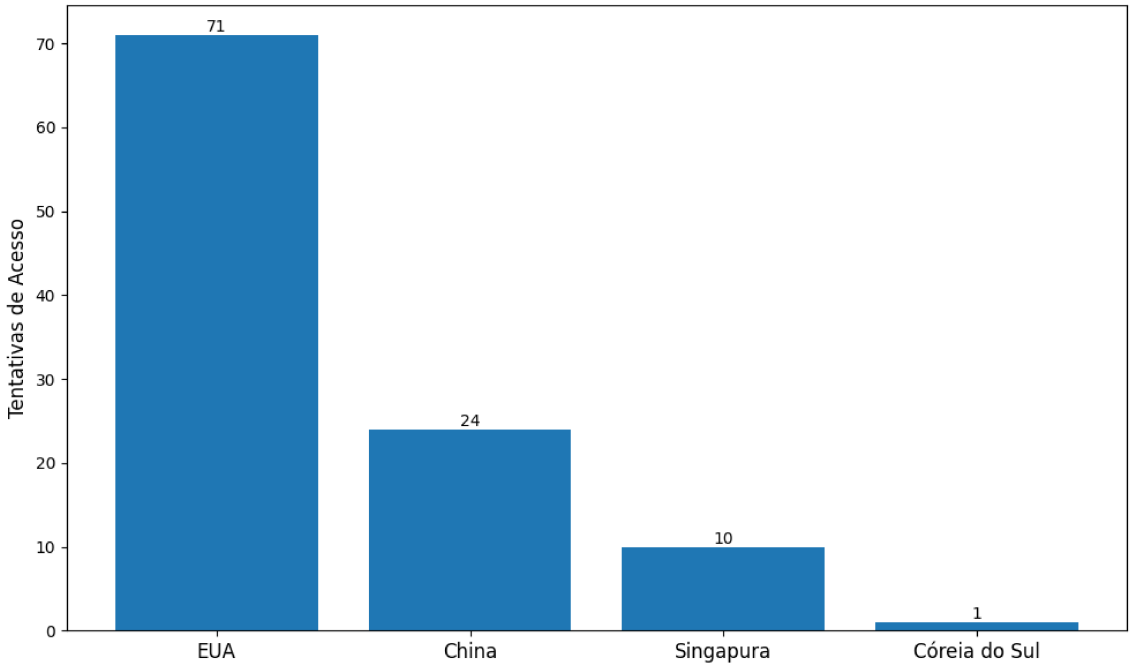


Figura 14 – Número de tentativas de acesso por país no Experimento A

Fonte: Autor

5.2.2 Tentativas de conexões bem-sucedidas

Diferentemente do Experimento B, no Experimento A os atacantes não só verificaram os serviços disponíveis, mas também fizeram tentativas de conexões bem sucedidas nos serviços do `Cowrie` na porta 23. Foram estabelecidas nove conexões bem-sucedidas, todas através do protocolo Telnet, com cada sessão durando menos de 0,5 segundos. A Tabela 3 fornece informações detalhadas sobre a data, endereço IP, localização e ASN de cada ataque.

Tabela 3 – Endereços IP, País de origem e ASN de cada conexão bem-sucedida realizada no Experimento A

ID	Data	Endereço IP de Origem	Localização	ASN
1	2023-10-08	x:x::9d20:678e	China	Aliyun Computing
2	2023-10-13	x:x::3ad7:2824	Singapore	Alibaba Cloud
3	2023-10-15	x:x::3ad7:2824	Singapore	Alibaba Cloud
4	2023-10-21	x:x::9d20:678c	China	Aliyun Computing
5	2023-10-22	x:x::9d20:6788	China	Aliyun Computing
6	2023-10-24	x:x::74a5:8b60	Singapore	Alibaba Cloud
7	2023-10-29	x:x::9d20:6789	China	Aliyun Computing
8	2023-11-05	x:x::9d20:678e	China	Aliyun Computing
9	2023-11-06	x:x::9d20:6787	China	Aliyun Computing

Fonte: Autor

Os atacantes fizeram uso de três combinações distintas de nomes de usuário e senhas.

A Tabela 4 expõe os nomes de usuário e senhas, cada um vinculado a uma credencial ID de login específico. Contudo, conforme se pode verificar, tais credenciais compõem o cabeçalho de uma requisição GET e, à vista do contexto, não parece terem sido utilizadas com o propósito de efetuar login no sistema `Cowrie`.

O endereço IPv6 encontrado na senha associada ao ID de login 1 corresponde ao endereço anonimizado da instância `asia-east1-ipv6-MotorBusca`, e a senha relacionada ao ID de login 3 foi deixada em branco pelos atacantes.

Tabela 4 – Nome de usuário e senha utilizado pelos atacantes no acesso a instância do Experimento A

Usuário	Senha	ID de login
GET / HTTP/1.1	Host: [x:x:x:x:x:x:x:x]:23	1
User-Agent: Mozilla/5.0 (Windows NT 10.0 Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.0 Safari/537.36	Accept: */*	2
Accept-Encoding: gzip		3

Fonte: Autor

O atacante usou os ID de Login 1, 2 e 3 na primeira conexão bem-sucedida. A terceira tentativa acessou a instância com sucesso, conforme mostrado na Tabela 5.

Em todas as outras conexões bem-sucedidas, o atacante tentou se conectar usando a credencial 1 e teve a conexão bem-sucedida ao tentar se conectar pela segunda vez usando a credencial 2. Lembrando que se o atacante tentar fazer login entre duas tentativas de login consecutivas com a mesma credencial, o segundo login será bloqueado.

Tabela 5 – Endereços IP, País de origem e ASN de cada conexão bem-sucedida realizada no Experimento A

IP ID	Data e Hora	Endereço IP do atacante	ID de Login
1	2023-10-08 03:28:23	x:x::9d20:678e	1,2,3
2	2023-10-13 10:08:36	x:x::3ad7:2824	1,2
3	2023-10-15 06:55:32	x:x::3ad7:2824	1,2
4	2023-10-21 18:14:17	x:x::9d20:678c	1,2
5	2023-10-22 03:17:21	x:x::9d20:6788	1,2
6	2023-10-24 08:17:13	x:x::74a5:8b60	1,2
7	2023-10-29 19:21:21	x:x::9d20:6789	1,2
8	2023-11-05 02:09:33	x:x::9d20:678e	1,2
9	2023-11-06 09:57:11	x:x::9d20:6787	1,2

Fonte: Autor

5.3 Resultados - Experimento B

Os resultados do Experimento B serão organizados por serviços a fim de facilitar a análise e o entendimento dos dados. Inicialmente, foi realizada a análise dos logs do `Cowrie`, conforme apresentado na Seção 5.3.1. Em seguida, na Seção 5.3.2, foram analisados os dados gerados pelo serviço Web. Por fim, na Seção 5.3.3, foram examinados os logs produzidos pelo `tcpdump`.

5.3.1 Cowrie

A única instância que teve ao menos uma tentativa de acesso nas portas 22 ou 23 foi a máquina `europe-west1-ipv6-control` que era a insância de controle. Foram 72 tentativas de conexão no total, sendo 48 tentativas para a porta 22 SSH e 24 para a porta 23 Telnet como mostrado na Figura 15. As tentativas de conexão foram originadas de 41 endereços IPv6 diferentes da faixa 2a06:4880:: conforme a Figura 16. Todas as conexões foram encerradas em menos de dois segundos, ou seja o atacante apenas tentou se conectar nas referidas portas e se desconectou logo em seguida.

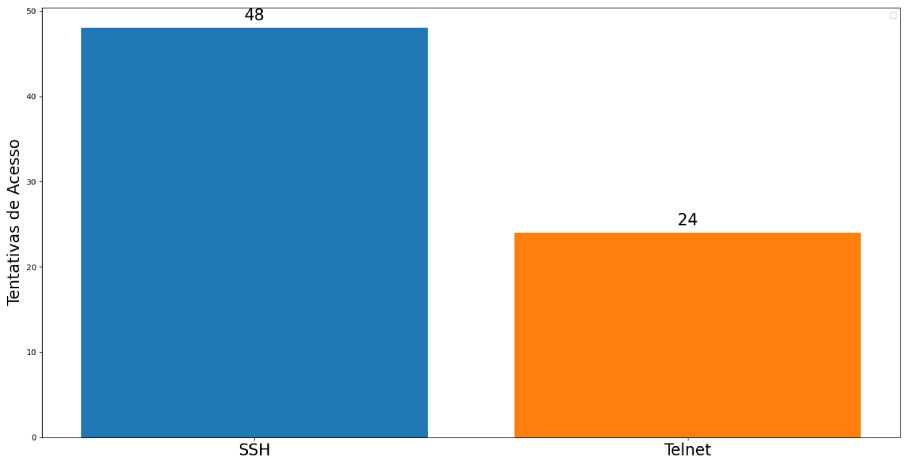


Figura 15 – Número de tentativas de acesso ao `Cowrie` por protocolo no Experimento B

Fonte: Autor

Embora tenham sido identificados 41 endereços IP distintos, todos pertencem à mesma organização: a empresa de cibersegurança `Constantine Cybersecurity Ltd`, sediada no Reino Unido. Não obstante a descrição no `whois` [2] indicar: "*This IP range is not attacking your network. Visit https://internet-measurement.com for more details.*", verificou-se que a

[2] https://search.arin.net/rdap/

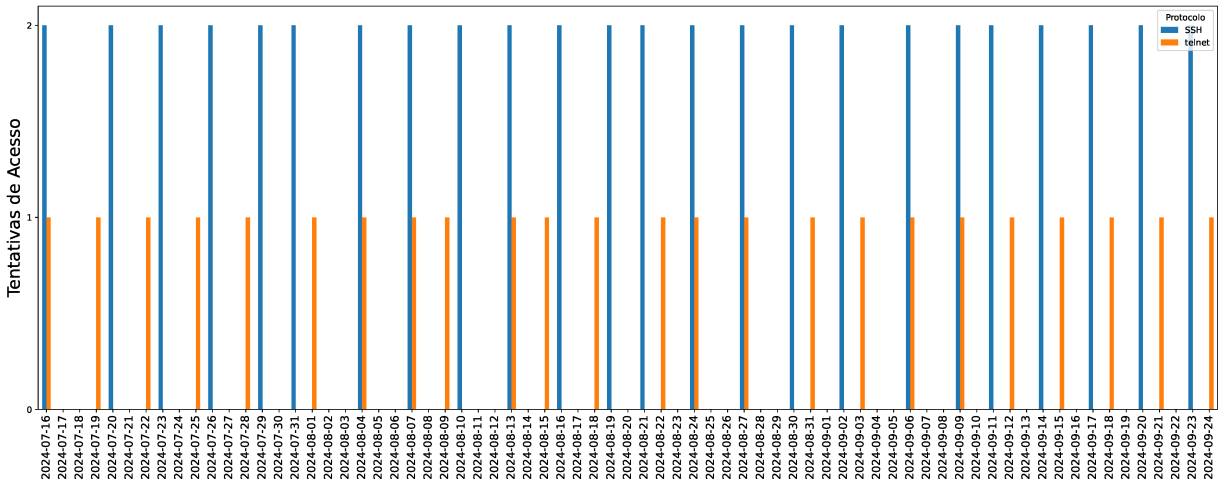


Figura 16 – Número de tentativas de acesso ao `Cowrie` por protocolo ao longo do tempo no Experimento B

Fonte: Autor

empresa estava registrada em duas (`blacklists`) como fonte de spam: s5h.net e UCEPROTECTL3 (MXTOOLBOX, 2024). A descrição disponível no *website* da empresa encontra-se no Apêndice B.

Com o intuito de compreender as razões pelas quais a máquina de controle foi alvo de acessos, foi enviado um e-mail questionando como foi realizada a descoberta da única instância cujo endereço não havia sido divulgado, esse questionamento é relevante, pois de acordo coma empresa eles não realizam nenhum tipo de *Web Crawler*. A resposta fornecida foi a seguinte: “Descobriu-se que esses endereços IP estavam sendo utilizados como *Subject Alternative Name* (SAN) em certificados. Ou seja, havia uma VM do GCP em execução que possuía endereços IPv4 e IPv6 associados e estava apresentando um certificado com ambos os IP no SAN.” A captura de tela da resposta do e-mail está disponível no Apêndice A.

5.3.2 Serviço web

Em relação aos serviços Web que foram criados nas portas 80 e 443, HTTP e HTTPS respectivamente, somente a instância `europe-west1-ipv6-control` foi alvo de acessos. Foram 24 tentativas de conexão na porta 80 e 25 tentativas na porta 443 totalizando 49 tentativas de acesso. Da mesma forma que as tentativas de acesso ao `Cowrie`, as instâncias não tiveram interação além de receberem um GET na página inicial. A Figura 17 ilustra esse fluxo de pacotes, onde a conexão se inicia, o atacante envia um pacote com método GET e a conexão é encerrada. Nenhum Usuário/Senha foi testado e nenhuma página além da página inicial foi visitada pelo atacante. Na Figura 18 são mostrados detalhes dos dados do método de requisição utilizado.

Todas as tentativas de acesso tiveram origem em 48 endereços IP diferentes, sendo todos

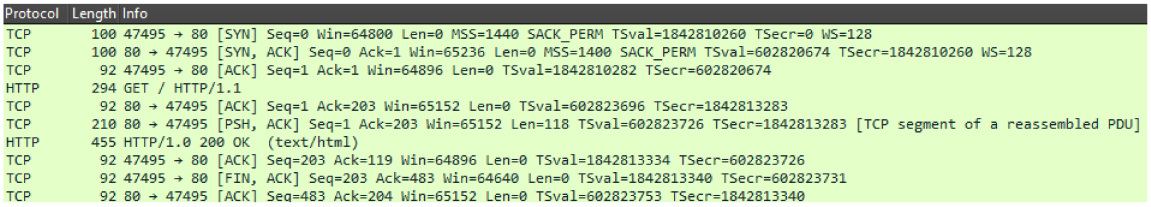


Figura 17 – Captura de tela do fluxo de pacotes retirados do `wireshark` para uma conexão na porta 80.

Fonte: Autor

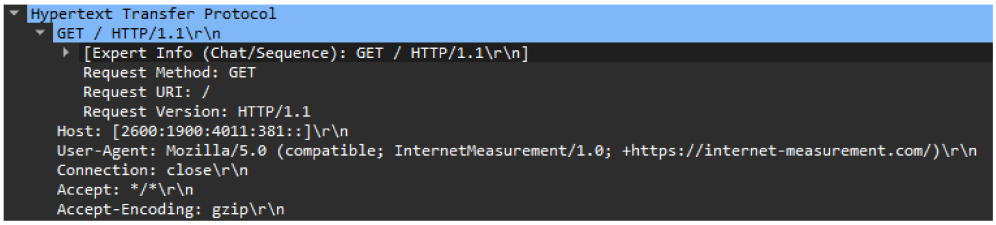


Figura 18 – Captura de tela dos detalhes do fluxo de pacotes retirados do `wireshark` para uma conexão na porta 80.

Fonte: Autor

pertencentes a mesma ASN 'Constantine Cybersecurity Ltd', todos tendo origem também na Grã-Bretanha.

A Figura 19 mostra a quantidade de acesso aos sistemas Web ao longo do tempo mostrando como a quantidade de tentativas de acesso estão bem distribuídas, implicando na utilização de algum `script` automatizado para varredura dos serviços.

5.3.3 Tcpdump

A principal melhoria do Experimento A para o Experimento B foi a implementação da captura dos pacotes de rede. Durante as análises do Experimento A, observou-se que o atacante poderia realizar varreduras de rede do dispositivo sem necessariamente acessar seus serviços, especificamente o `Cowrie`. Esta implementação possibilitou a coleta de qualquer interação do atacante com o *honeypot*, independentemente dos serviços em execução, proporcionando novos dados para compreensão dos padrões de ataque.

Os pacotes então enviados e recebidos pela interface de rede IPv6 das instâncias foram capturados pelo `tcpdump`, que salva os dados em formato `pcap`. Em virtude do elevado volume de pacotes capturados, foram realizados diversos pré-processamentos para filtrar as informações relevantes para análise.

O primeiro pré-processamento realizado foi a coleta de alguns dados de cabeçalho que estavam dentro dos arquivos `pcap`, que foram:

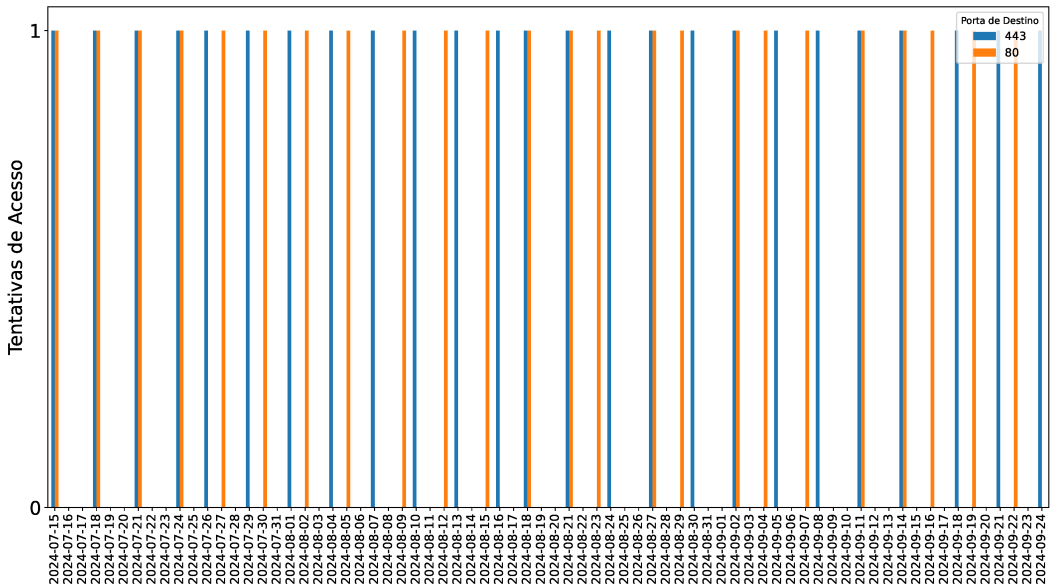


Figura 19 – Número de tentativa de acesso aos serviços Web por protocolo ao longo do tempo no Experimento B

Fonte: Autor

- ❏ *Timestamp*;
- ❏ Data;
- ❏ Número do pacote;
- ❏ Índice de *Stream*;
- ❏ IP de origem;
- ❏ IP de destino;
- ❏ Porta de origem;
- ❏ Porta de Destino.

Os pacotes contendo todas as informações necessárias foram selecionados e armazenados em formato `json`, sendo posteriormente inseridos em um banco de dados `MySQL`. O *timestamp* serviu para identificar dia e horário de recebimento/envio dos pacotes, criando-se uma coluna específica de Data para otimizar a filtragem. A numeração dos pacotes, essencial para eventual consulta no arquivo `pcap`, em conjunto com o índice de *stream* - número atribuído pelo `Wireshark` a cada conexão estabelecida [3] - possibilitou a identificação das tentativas de acesso únicas. Foram registrados também os dados de Protocolo, IPs e Portas de origem e destino.

[3] https://www.wireshark.org/docs/wsug_html_chunked/ChAdvFollowStreamSection.html

Com o auxilio do banco de dados foi possível constatar que somente a instância `europe-west1-control` foi alvo assim como nos serviços de `Cowrie` e serviços Web.

Foram colhidos então 2.035.139 pacotes após o primeiro passo do pré-processamento. No segundo passo foram filtrados os endereços IP pertencentes ao Google que são utilizados pelo próprio `Google Cloud` para coletar log das instâncias.

O terceiro passo de pré-processamento foi filtrar as conexões únicas utilizando os dados de Data, Índice de *Stream*, endereços IP de origem e destino e também portas de origem e destino, gerando assim 755.453 tentativas de acessos únicas ao longo de toda duração do experimento como mostra a Figura 20.

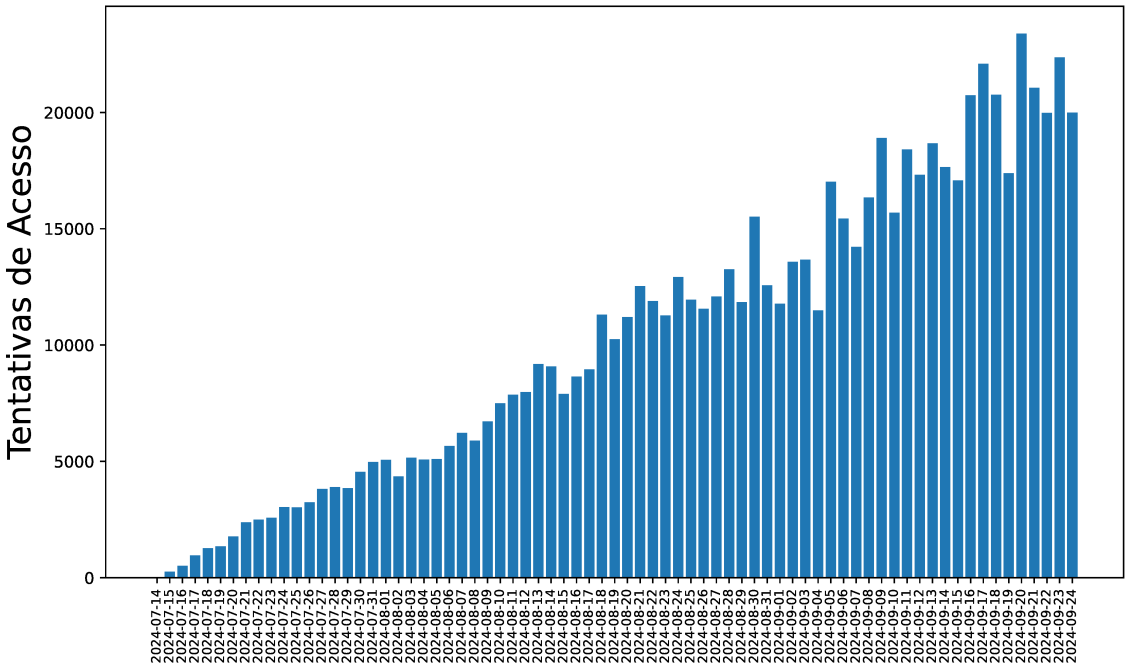


Figura 20 – Número de tentativa de acesso ao longo do tempo no experimento B.

Fonte: Autor

A notável discrepância entre o número de tentativas de acesso capturados pelo `tcpdump` se deve ao fato de o atacante não completar o TCP *handshake* de três vias para acessar as portas de serviço que estavam abertas. Dessa forma, os serviços do `Cowrie` e Web não registraram os logs dessas tentativas de acesso, sendo então somente registrado pelos arquivos `pcap`.

Como todas as tentativas de acesso originaram-se da 'Constantine Cybersecurity Ltd', infere-se a existência de um único atacante, o qual utilizou 290 endereços IP distintos durante o período do experimento, conforme ilustrado na Figura 21. Durante o experimento, o número de tentativas de acesso por endereço IP variou entre 2 e 12.890, apresentando tendência crescente ao longo do período.

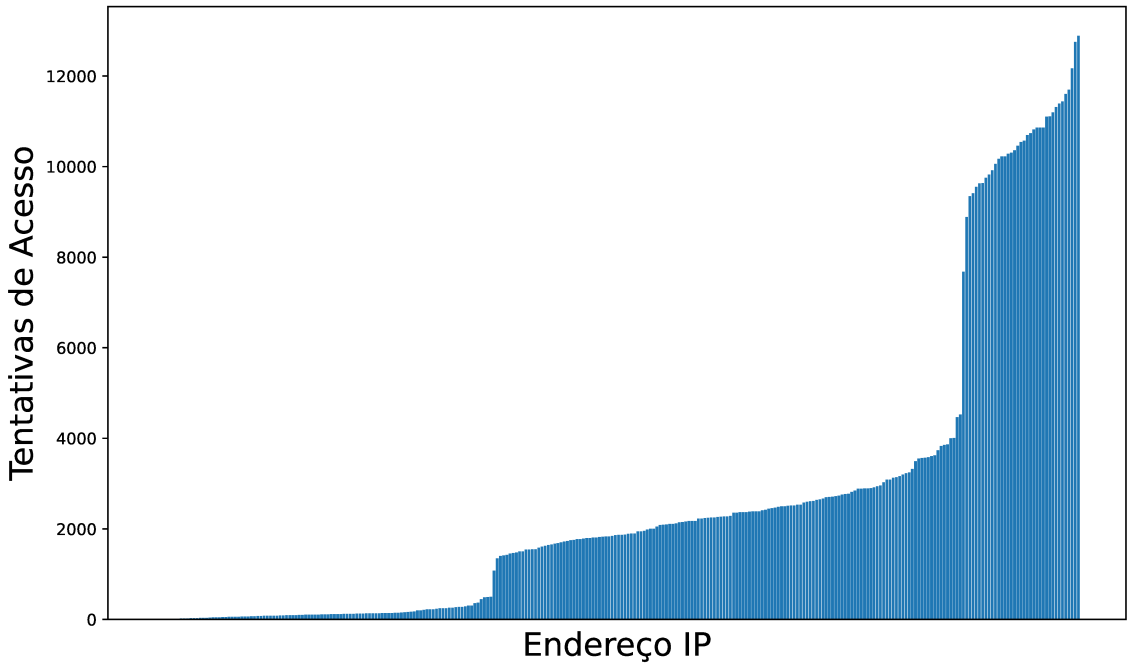


Figura 21 – Número de tentativa de acesso para cada um dos 290 endereços IPv6 (eixo x) que foram encontrados no Experimento B.

Fonte: Autor

Com base nos dados apresentados na Figura 21, foi elaborado um mapa de calor, ilustrado na Figura 22. O gráfico apresenta no eixo Y os endereços IP, no eixo X os dias, e a intensidade das tentativas de acesso é representada por um gradiente de cores do roxo ao azul, conforme indicado na barra lateral direita.

A análise revela um padrão de varredura caracterizado por duas particularidades: a alternância periódica entre diferentes blocos de endereços IP e a intensificação das tentativas de acesso nas portas que apresentavam resultados positivos durante a varredura com o passar do tempo.

Com relação às portas de destino, identificou-se 5.060 portas distintas como alvo, com variação de acessos entre 1 e 9.560. As portas mais requisitadas foram a 443, seguida das portas 22 e 80, conforme apresentado na Tabela 6, que exibe as 10 portas com maior número de acessos.

Embora as quatro portas mais acessadas, conforme a Tabela 6, pertençam à categoria portas baixas (portas inferiores a 1023), a análise segregada demonstra que as portas superiores a 1023 foram os principais alvos de tentativas de acesso, com essa tendência se intensificando ao longo do experimento. Adicionalmente, na Figura 23, observou-se um aumento progressivo do desvio padrão relacionado às portas altas, evidenciando dois perfis distintos de comportamento para as portas de destino.

A Figura 24 apresenta uma análise das tentativas de acesso às portas de destino, seguindo

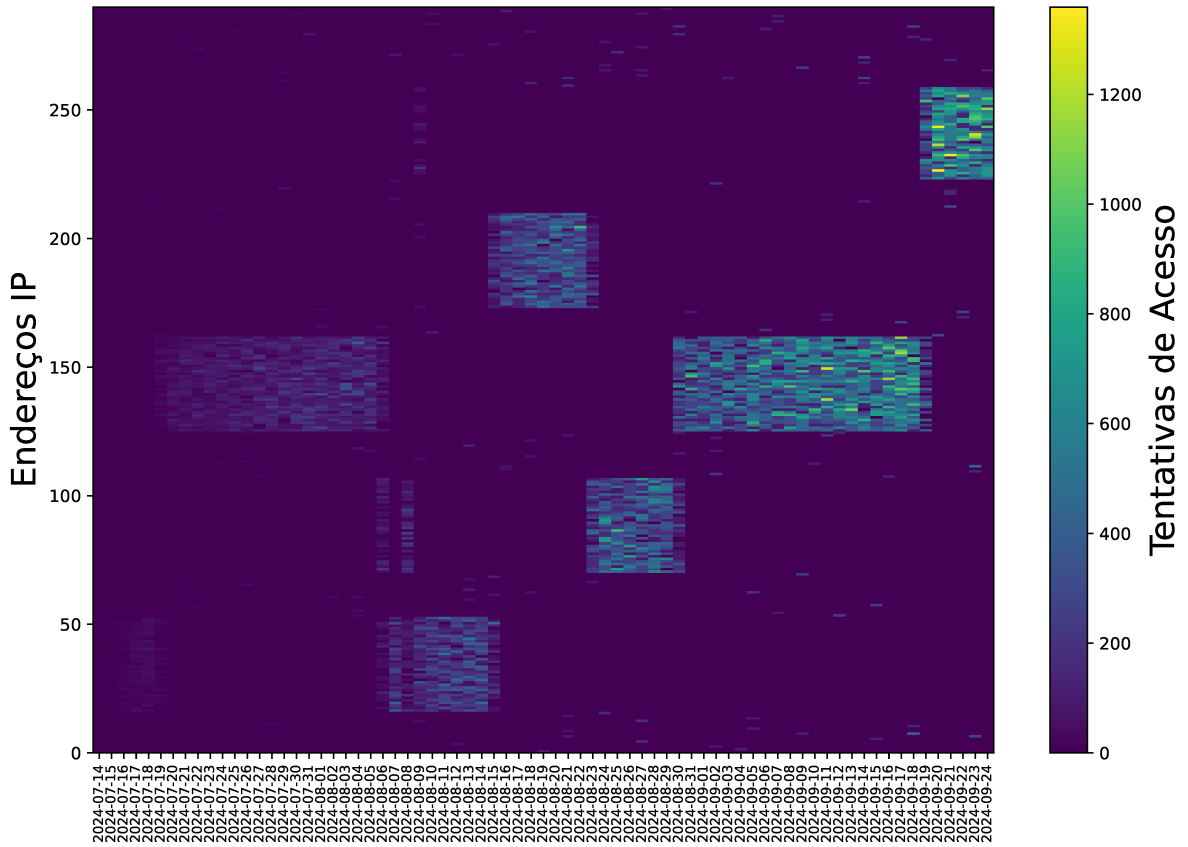


Figura 22 – Número de tentativas de acesso ao longo do experimento B. As datas são representadas no eixo X e os endereços IP, no eixo Y. A intensidade da cor da barra gradiente indica a quantidade de tentativas de acesso em cada ponto.

Fonte: Autor

Tabela 6 – As 10 portas de destino com maior quantidade de acesso no Experimento B

Porta de Destino	Quantidade Total
443	9560
22	8520
80	6785
25	5083
8443	5083
139	5083
8080	5079
135	5072
3306	5033
7547	5010

Fonte: Autor

metodologia similar à Figura 22. Observa-se um aumento gradual no volume de tentativas ao longo do tempo, mantendo um padrão consistente de crescimento. Entretanto, diferen-

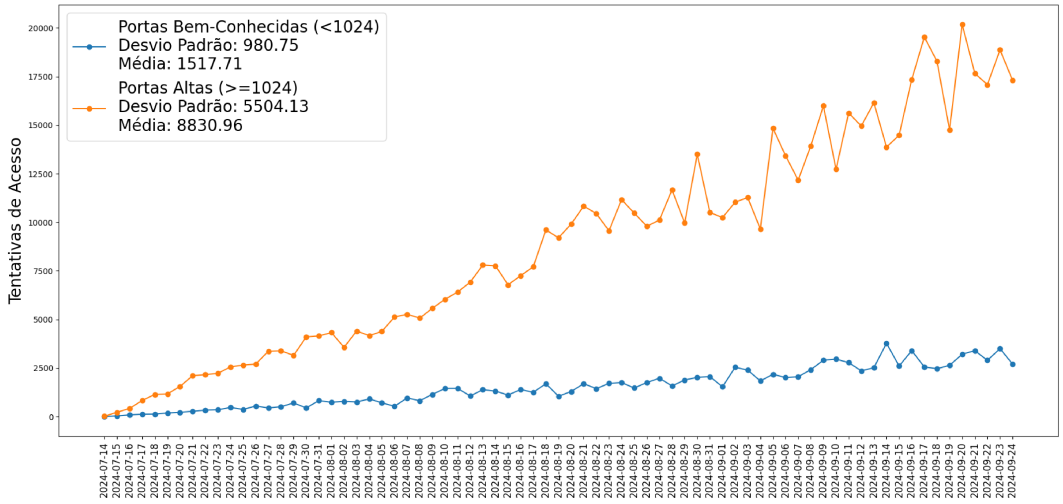


Figura 23 – Tentativas de acesso separados por portas Bem-Conhecidas (menores que 1023) e o restante das portas, junto dos seus valores de desvio padrão.

Fonte: Autor

temente da análise dos endereços IP, as portas de destino não demonstram um padrão lógico em sua seleção.

5.4 Discussão dos resultados

A aplicação dos métodos estudados para a divulgação de endereços IPv6 na Internet, mostrou-se mais desafiadora do que o esperado. No Experimento A, apesar de a instância que foi encontrada ter sido divulgada, o método que o atacante utilizou para a encontrar não estava relacionada aos métodos de divulgação utilizados, que foram os motores de busca `Censys` e `Shodan`. Da mesma forma em que o Experimento B, em que a única instância que foi acessada, foi uma instância de controle que não teve seu endereço IP divulgado. O caso do experimento B houve uma justificativa para isso ter acontecido. O atacante possuía o endereço IPv6 previamente salvo em seu banco de dados, o que poderia se aplicar também a instância `asia-east1-ipv6-MotorBusca` que foi encontrada no Experimento A em menos de 24 horas após o inicio do experimento por uma empresa que oferece serviços de varredura de rede [4].

A análise dos experimentos evidencia a complexidade na divulgação e indexação de endereços IPv6 na Internet. A ausência de investimento significativo por parte das empresas na localização destes endereços sugere um baixo interesse nessa área, além disso também demonstra que a rede IPv6 pode constituir um ambiente relativamente seguro para hospe-

[4] https://networktools.he.net

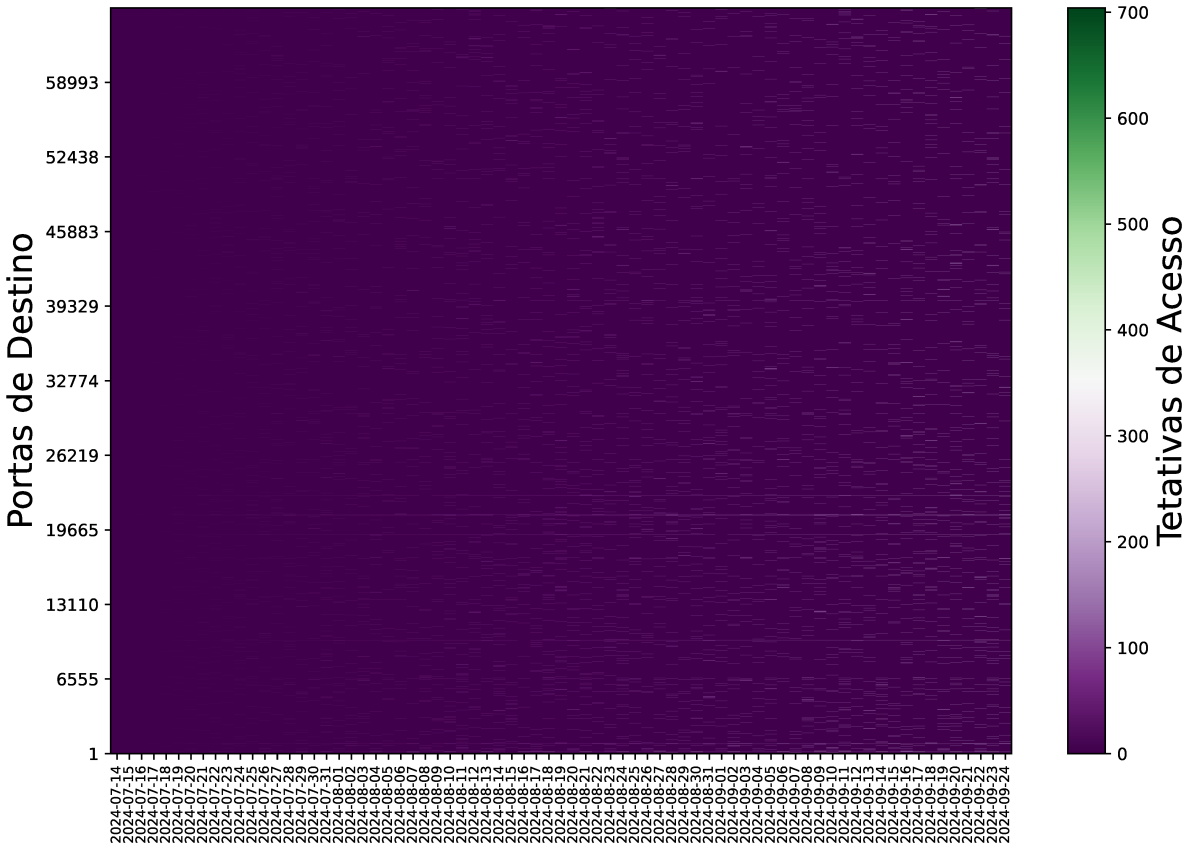


Figura 24 – Número de tentativas de acesso ao longo do experimento B. As datas são representadas no eixo X e as portas de destino, no eixo Y. A intensidade da cor da barra gradiente indica a quantidade de tentativas de acesso em cada ponto.

Fonte: Autor

dagem de serviços que não tem como interesse serem encontradas.

No Experimento B, foram registradas 755.453 tentativas de conexão através do `tpdump`. Em nenhuma dessas tentativas os atacantes procuraram efetuar login no `Cowrie` ou navegar além da página inicial do *honeypot* Web. Embora alguns dados de usuário/senha tenham sido utilizados, não houve demonstração de interesse efetivo em acessar o dispositivo, indicando que o objetivo dos invasores limitava-se a verificar a atividade dos endereços IPv6 e identificar portas abertas nos endereços acessíveis.

Evidencia-se que, apesar da extensa gama de endereços IPv6 pertencentes ao Google, a empresa os reutiliza em suas aplicações, especialmente no endereçamento de instâncias do GCP. Ademais, observa-se que a instância `europe-west1-ipv6-control` do Experimento B foi identificada no mesmo dia de início do experimento, demonstrando que a empresa 'Constantine Cybersecurity Ltd' mantinha esse endereço em seu banco de dados. Tal fato ressalta a importância de preservar e monitorar endereços IPv6, mesmo inativos, considerando a possibilidade de sua reativação futura. O mesmo pode se esperar do Experimento

A que também possuiu endereço encontrado um dia após o experimento por uma empresa que vende serviços de varredura de rede.

CAPÍTULO 6

Conclusão

Este trabalho propõe-se a analisar a eficiência de cinco métodos de divulgação de endereços IPv6 através de *honeypots* instanciados em nuvem. O Capítulo 2 apresentou uma base teórica para entendimento desse trabalho, incluindo os temas de Segurança cibernética, *honeypots*, IPv6, virtualização e computação em nuvem, DNS, SYN, SYN Scan e a Surface Deep e Dark Web. O Capítulo 3 foi necessário para uma contextualização desse trabalho dentro da produção acadêmica, onde foram abordados os artigos relevantes que foram escritos sobre *honeypots* e IPv6. No Capítulo 4 foram abordados todos os procedimentos feitos para a criação dos dois experimentos realizados durante a execução desse trabalho. Os resultados então foram mostrados e discutidos no Capítulo 5 onde foi constatado que apesar dos esforços empregados nos Experimentos A e B, utilizando ferramentas como `Censys` e `Shodan`, divulgação via fóruns, divulgação utilizando DNS, *Crawler* e WAN, a descoberta por atacantes de instâncias com endereços IPv6 divulgados foi mais difícil do que o previsto.

Essa dificuldade em fazer os endereços IPv6 que foram divulgados se tornarem conhecidos, pode ser explicada pela falta de interesse em investir nesse tipo de atividade. Como consequência, a rede IPv6 pode se tornar um ambiente propício para hospedar serviços que não devem ser encontrados, inclusive aqueles considerados ilícitos, uma vez que são menos visíveis e rastreáveis.

Os resultados dos experimentos também indicam que o interesse dos atacantes não está necessariamente ligado à exploração de vulnerabilidades ou à identificação de serviços específicos, mas sim à descoberta de portas abertas em endereços IPv6 já conhecidos.

Além disso, a pesquisa revelou que grandes empresas como o Google reutilizam seus endereços IPv6, o que pode dificultar ainda mais a rastreabilidade de suas atividades na rede. Por outro lado, a descoberta da instância `europe-west1-ipv6-control` demonstra a importância de manter um banco de dados com endereços IPv6, mesmo que inativos, pois eles podem ser reutilizados com frequência no futuro.

Em resumo, a divulgação de endereços IPv6 na Internet se mostrou um desafio maior do que o esperado, com implicações importantes para a segurança da rede e para as estratégias de ataque e defesa cibernética.

6.1 Contribuições bibliográficas e tecnológicas

A produção deste trabalho trouxe duas contribuições para a academia, sendo uma produção cientifica e uma produção tecnológica. A produção cientifica foi a publicação do artigo *Evaluating Attacks on IPv6 Devices Using Honeypots: A Preliminary Analysis* na conferência IEEE CCNC 2025 [1].

A principal contribuição tecnológica deste trabalho é o desenvolvimento de um *honeypot* híbrido de média interação que utiliza o `Cowrie` para emulação de serviços de conexão remota além da emulação de serviços Web utilizando `scripts` em `Python`. Todo o código utilizado para a construção do *honeypot* `Trigona` encontra-se hospedado e disponível para reproduções no `Github` (MUNDIM, 2024).

6.2 Trabalhos futuros

Foram testados cinco métodos de divulgação de endereços IPv6 utilizando endereços fornecidos por meio de um serviço de nuvem que esteve no ar por pouco mais de um mês no Experimento A e dois meses e meio no Experimento B. Os itens descritos abaixo podem ser utilizados como guia para trabalhos futuros:

- ❏ Testar outros métodos de divulgação como hospedagem de serviços NTP ou rede P2P muito utilizada em blockchain/criptomoedas;
- ❏ Empregar endereços IPv6 que não foram usados;
- ❏ Monitorar os endereços IPv6 por um período maior;
- ❏ Utilizar uma faixa maior de endereços IP para divulgação;
- ❏ Analisar se após descoberta de um endereço IP, existe interesse em encontrar os vizinhos desse endereços IP também.

[1] https://ccnc2025.ieee-ccnc.org

Referências

AKHGAR, B. et al. **Dark Web Investigation**. Springer International Publishing, 2021. (Security Informatics and Law Enforcement). ISBN 9783030553432. Disponível em: <https://doi.org/10.1007/978-3-030-55343-2>.

ARMBRUST, M. et al. A view of cloud computing. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 53, n. 4, p. 50–58, abr. 2010. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/1721654.1721672>.

ASHRAF, Z. et al. Challenges and mitigation strategies for transition from ipv4 network to virtualized next-generation ipv6 network. **Int. Arab J. Inf. Technol.**, v. 20, n. 1, 2023. Disponível em: <https://doi.org/10.34028/iajit/20/1/9>.

AYELE, W. Y. **Non-repudiation mechanisms for IoT applications : A systematic literature review**. 2021. 59 p.

BAIG, Z. A.; ADENIYE, S. C. A trust-based mechanism for protecting ipv6 networks against stateless address auto-configuration attacks. In: **2011 17th IEEE International Conference on Networks**. Singapore: IEEE, 2011. p. 171–176. Disponível em: <https://doi.org/10.1109/ICON.2011.6168470>.

BASER, M.; GUVEN, E. Y.; AYDIN, M. A. **Analysis of Malicious Files Gathering via Honeypot Trap System and Benchmark of Anti-Virus Software**. Istanbul, Turkey: Research Square Platform LLC, 2023. Disponível em: <https://doi.org/10.21203/rs.3.rs-3733937>.

CHENEAU, T.; LAURENT, M. Using send signature algorithm agility and multiple-key cga to secure proxy neighbor discovery and anycast addressing. In: **2011 Conference on Network and Information Systems Security**. La Rochelle, France: IEEE, 2011. p. 1–7. Disponível em: <https://doi.org/10.1109/SAR-SSI.2011.5931376>.

CHIBA, D. et al. Domainprofiler: Toward accurate and early discovery of domain names abused in future. **International Journal of Information Security**, 2017. Disponível em: <https://doi.org/10.1007/s10207-017-0396-7>.

CZYZ, J. et al. Don't forget to lock the back door! a characterization of ipv6 network security policy. In: **Network and Distributed System Security Symposium**. California, USA: Network and Distributed Systems Security (NDSS), 2016. Disponível em: <https://doi.org/10.14722/NDSS.2016.23047>.

DAHLMANNS, M. et al. Unconsidered installations: Discovering iot deployments in the ipv6 internet. In: **NOMS 2024-2024 IEEE Network Operations and Management Symposium**. Seoul, Republic of Korea: IEEE, 2024. p. 1–8. Disponível em: <https://doi.org/10.1109/NOMS59830.2024.10574963>.

DECCIO, C. Maintenance, mishaps and mending in deployments of the domain name system security extensions (dnssec). **International Journal of Critical Infrastructure Protection**, 2012. Disponível em: <https://doi.org/10.1016/j.ijcip.2012.05.002>.

DEERING, D. S. E.; HINDEN, B. **IP Version 6 Addressing Architecture**. RFC Editor, 2006. RFC 4291. (Request for Comments, 4291). Disponível em: <https://doi.org/10.17487/RFC4291>.

ERICKSON, J. **Hacking: the art of exploitation, 2nd edition**. Second. USA: No Starch Press, 2008. ISBN 9781593271442.

FALL, K.; STEVENS, W. **TCP/IP Illustrated, Volume 1: The Protocols, Volume 1**. Pearson Education, 2011. (Addison-Wesley Professional Computing Series). ISBN 9780132808187. Disponível em: <https://books.google.com.br/books?id=a23OAn5i8R0C>.

FUKUDA, K.; HEIDEMANN, J. Who knocks at the ipv6 door? detecting ipv6 scanning. In: **Proceedings of the Internet Measurement Conference 2018**. New York, NY, USA: Association for Computing Machinery, 2018. (IMC '18), p. 231–237. ISBN 9781450356190. Disponível em: <https://doi.org/10.1145/3278532.3278553>.

FUKUDA, K.; HEIDEMANN, J.; QADEER, A. Detecting malicious activity with dns backscatter over time. **IEEE/ACM Transactions on Networking**, v. 25, n. 5, p. 3203–3218, 2017. Disponível em: <https://doi.org/10.1109/TNET.2017.2724506>.

GONT, F. **Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery**. RFC Editor, 2013. RFC 6980. (Request for Comments, 6980). Disponível em: <https://doi.org/10.17487/RFC6980>.

____. **A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)**. Provincia de Buenos Aires,Argentina, 2014. Disponível em: <https://doi.org/10.17487/RFC7217>.

Google Cloud. **Google Cloud Security Overview**. 2024. Disponível em: <https://cloud.google.com/docs/security/overview/whitepaper?hl=pt-br>.

GRAHAM, J.; OLSON, R.; HOWARD, R. **Cyber Security Essentials**. CRC Press, 2016. ISBN 9781439851265. Disponível em: <https://doi.org/10.1201/b10485>.

GRIGORIOU, E. et al. Protecting iec 60870-5-104 ics/scada systems with honeypots. In: **2022 IEEE International Conference on Cyber Security and Resilience (CSR)**. Rhodes, Greece: IEEE, 2022. p. 345–350. Disponível em: <https://doi.org/10.1109/CSR54599.2022.9850329>.

HENDRIKS, L. et al. On the potential of ipv6 open resolvers for ddos attacks. In: KAAFAR, M. A.; UHLIG, S.; AMANN, J. (Ed.). **Passive and Active Measurement**. Cham: Springer International Publishing, 2017. p. 17–29. ISBN 978-3-319-54328-4. Disponível em: <https://doi.org/10.1007/978-3-319-54328-4_2>.

HENDRIKS, L.; SPEROTTO, A.; PRAS, A. Characterizing the ipv6 security landscape by large-scale measurements. In: LATRÉ, S. et al. (Ed.). **Intelligent Mechanisms for Network Configuration and Security**. Cham: Springer International Publishing, 2015. p. 145–149. ISBN 978-3-319-20034-7. Disponível em: <https://doi.org/10.1007/978-3-319-20034-7_16>.

HERMANN, S.; FABIAN, B. A comparison of internet protocol (ipv6) security guidelines. **Future Internet**, v. 6, n. 1, p. 1–60, 2014. ISSN 1999-5903. Disponível em: <https://doi.org/10.3390/fi6010001>.

HUMMEN, R. et al. 6lowpan fragmentation attacks and mitigation mechanisms. In: **Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks**. New York, NY, USA: Association for Computing Machinery, 2013. (WiSec '13), p. 55–66. ISBN 9781450319980. Disponível em: <https://doi.org/10.1145/2462096.2462107>.

ICHISE, H. et al. Detection and blocking of anomaly dns traffic by analyzing achieved ns record history. In: **2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)**. Hawaii, USA: IEEE, 2018. p. 1586–1590. Disponível em: <https://doi.org/10.23919/APSIPA.2018.8659739>.

JAKAB, L. et al. Lisp-tree: A dns hierarchy to support the lisp mapping system. **IEEE Journal on Selected Areas in Communications**, v. 28, n. 8, p. 1332–1343, 2010. Disponível em: <https://doi.org/10.1109/JSAC.2010.101011>.

KELLY, C. et al. A comparative analysis of honeypots on different cloud platforms. **Sensors**, v. 21, n. 7, 2021. ISSN 1424-8220. Disponível em: <https://doi.org/10.3390/s21072433>.

KHEIR, N. et al. Mentor: Positive dns reputation to skim-off benign domains in botnet c&c blacklists. In: **29th IFIP International Information Security Conference (SEC)**. Marrakech, Morocco: Springer, 2014. p. 1–14. Disponível em: <https://doi.org/10.1007/978-3-642-55415-5_1>.

KIM, T. H.; REEVES, D. A survey of domain name system vulnerabilities and attacks. **Journal of Surveillance, Security and Safety**, v. 1, n. 1, 2020. ISSN 2694-1015. Disponível em: <https://doi.org/10.20517/jsss.2020.14>.

KISHIMOTO, K. et al. An adaptive honeypot system to capture ipv6 address scans. In: **2012 International Conference on Cyber Security**. Alexandria, USA: IEEE, 2012. p. 165–172. Disponível em: <https://doi.org/10.1109/CyberSecurity.2012.28>.

Klein Nijenhuis, T. **Discovery and Quantification of Open DNS Resolvers on IPv6**. 2018. Disponível em: <http://essay.utwente.nl/74575/>.

KUROSE, J. F.; ROSS, K. W. **Computer Networking: A Top-Down Approach (6th Edition)**. 6th. ed. [S.l.]: Pearson, 2012. ISBN 0132856204.

LIAO, Y. et al. A site selection method of dns using the particle swarm optimization algorithm. **Transactions in GIS**, Wiley Online Library, v. 21, n. 5, p. 969–983, 2017. Disponível em: <https://doi.org/10.1111/tgis.12244>.

LIU, Z. et al. Security and privacy protection of smart home based on ipv6. In: **Proceedings of the 2014 International Conference on Mechatronics, Electronic, Industrial and Control Engineering**. Atlantis Press, 2014/11. p. 517–520. ISBN 978-94-62520-42-4. ISSN 2352-5401. Disponível em: <https://doi.org/10.2991/meic-14.2014.116>.

LU, Y.; TSUDIK, G. Towards plugging privacy leaks in the domain name system. In: **2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P)**. Delft, Netherlands: IEEE, 2010. p. 1–10. Disponível em: <https://doi.org/10.1109/P2P.2010.5569976>.

MANZUIK, S.; GOLD, A.; GATFORD, C. **Network Security Assessment: From Vulnerability to Patch by Steve Man**. 1. ed. Dutch: Syngress, 2006.

MICROSOFT. **O que é Segurança Cibernética**. 2024. Disponível em: <https://support.microsoft.com/pt-br/topic/o-que-%C3%A9-seguran%C3%A7a-cibern%C3%A9tica-8b6efd59-41ff-4743-87c8-0850a352a390>.

MOLLIN, R. A. **An Introduction to Cryptography, Second Edition (Discrete Mathematics and Its Applications)**. USA: Chapman & Hall/CRC, 2006. ISBN 1584886188.

MUNDIM, L. trigona. 9 2024. Disponível em: <https://figshare.com/articles/software/trigona/26973181>.

MXTOOLBOX. **Email Blacklist Check**. 2024. Accessed on 2024-10-21. Disponível em: <https://mxtoolbox.com/blacklists.aspx>.

NAAGAS, M. A.; GAMILLA, A. P. Denial of service attack: An analysis to ipv6 extension headers security nightmares. **International Journal of Electrical and Computer Engineering (Ijece)**, 2022. Disponível em: <https://doi.org/10.11591/ijece.v12i3.pp2922-2930>.

NARTEN, D. T.; JINMEI, T.; THOMSON, D. S. **IPv6 Stateless Address Autoconfiguration**. RFC Editor, 2007. RFC 4862. (Request for Comments, 4862). Disponível em: <https://doi.org/10.17487/RFC4862>.

National Institute of Standards and Technology. **Cyber Threat - NIST Glossary**. 2024. Disponível em: <https://csrc.nist.gov/glossary/term/cyber_threat>.

____. **Cybersecurity - NIST Glossary**. 2024. Disponível em: <https://csrc.nist.gov/glossary/term/cybersecurity>.

____. **Information Security - NIST Glossary**. 2024. Disponível em: <https://csrc.nist.gov/glossary/term/information_security>.

OCCUPYTHEWEB. **Linux Basics for Hackers: Getting Started with Networking, Scripting, and Security in Kali**. No Starch Press, 2018. ISBN 9781593278557. Disponível em: <https://books.google.com.br/books?id=HkFYswEACAAJ>.

PORTNOY, M. **Virtualization Essentials**. 1st. ed. USA: SYBEX Inc., 2012. ISBN 1118176715.

PRIYA, V. D.; CHAKKARAVARTHY, S. S. Containerized cloud-based honeypot deception for tracking attackers. **Scientific Reports**, Nature Publishing Group UK London, v. 13, n. 1, p. 1437, 2023. Disponível em: <https://doi.org/10.1038/s41598-023-28613-0>.

PROJECT, H. **Know Your Enemy: Learning about Security Threats**. USA: Addison-Wesley, 2004. ISBN 9780321166463. Disponível em: <https://books.google.com.br/books?id=2LJgPgAACAAJ>.

PROVOS, N. Honeyd - a virtual honeypot daemon. In: **12th USENIX Security Symposium**. Washington, D.C.: USENIX Association, 2003.

PUTRA, W. R. A. Infrastructure as code for security automation and network infrastructure monitoring. **Matrik Jurnal Manajemen Teknik Informatika Dan Rekayasa Komputer**, 2022. Disponível em: <https://doi.org/10.30812/matrik.v22i1.2471>.

RAGSDALE, J.; BOPPANA, R. V. On designing low-risk honeypots using generative pre-trained transformer models with curated inputs. **IEEE Access**, v. 11, p. 117528–117545, 2023. Disponível em: <https://doi.org/10.1109/ACCESS.2023.3326104>.

RAZALI, M. F. et al. Iot honeypot: A review from researcher's perspective. In: **2018 IEEE Conference on Application, Information and Network Security (AINS)**. Langkawi, Malaysia: IEEE, 2018. p. 93–98. Disponível em: <https://doi.org/10.1109/AINS.2018.8631494>.

RICHTER, P.; GASSER, O.; BERGER, A. Illuminating large-scale ipv6 scanning in the internet. In: **Proceedings of the 22nd ACM Internet Measurement Conference**. New York, NY, USA: Association for Computing Machinery, 2022. (IMC '22), p. 410–418. ISBN 9781450392594. Disponível em: <https://doi.org/10.1145/3517745.3561452>.

RYE, E.; LEVIN, D. Ipv6 hitlists at scale: Be careful what you wish for. In: **Proceedings of the ACM SIGCOMM 2023 Conference**. New York, NY, USA: Association for Computing Machinery, 2023. (ACM SIGCOMM '23), p. 904–916. ISBN 9798400702365. Disponível em: <https://doi.org/10.1145/3603269.3604829>.

SCHNOR, B.; SCHEFFLER, T. Hyhoneydv 6 : A hybrid honeypot architecture for ipv 6 networks. **International Journal of Intelligent Computing Research (IJICR)**, v. 6, n. 2, June 2015. Disponível em: <https://api.semanticscholar.org/CorpusID:28483583>.

SEIFERT, C.; WELCH, I.; KOMISARCZUK, P. **Taxonomy of honeypots**. Wellington, New Zealand: Citeseer, 2006.

SETHURAMAN, M.; MATHI, S. Preventing denial-of-service in next generation internet protocol mobility. **Indonesian Journal of Electrical Engineering and Computer Science**, 2018. Disponível em: <https://doi.org/10.11591/ijeecs.v12.i1.pp137-146>.

SETIANTO, F. et al. Gpt-2c: A gpt-2 parser for cowrie honeypot logs. **ArXiv**, abs/2109.06595, 2021. Disponível em: <https://doi.org/10.1145/3487351.3492723>.

SPITZNER, L. **Honeypots: Tracking Hackers**. Addison-Wesley, 2003. ISBN 9780321108951. Disponível em: <https://books.google.com.br/books?id=oNFQAAAAMAAJ>.

STALLINGS, W. **Cryptography and Network Security: Principles and Practice**. 3rd. ed. Harlow, England: Pearson Education, 2002. ISBN 0130914290.

SWANAGAN, M. **The 3 Types Of Security Controls (Expert Explains)**. 2023. Acessado em agosto/2024. Disponível em: <https://purplesec.us/learn/security-controls/>.

TANENBAUM, A. S.; WETHERALL, D. J. **Computer Networks**. 5th. ed. USA: Prentice Hall Press, 2010. ISBN 0132126958.

TANVEER, H. B. et al. Glowing in the dark: Uncovering IPv6 address discovery and scanning strategies in the wild. In: **32nd USENIX Security Symposium (USENIX Security 23)**. Anaheim, CA: USENIX Association, 2023. p. 6221–6237. ISBN 978-1-939133-37-3. Disponível em: <https://www.usenix.org/conference/usenixsecurity23/presentation/bin-tanveer>.

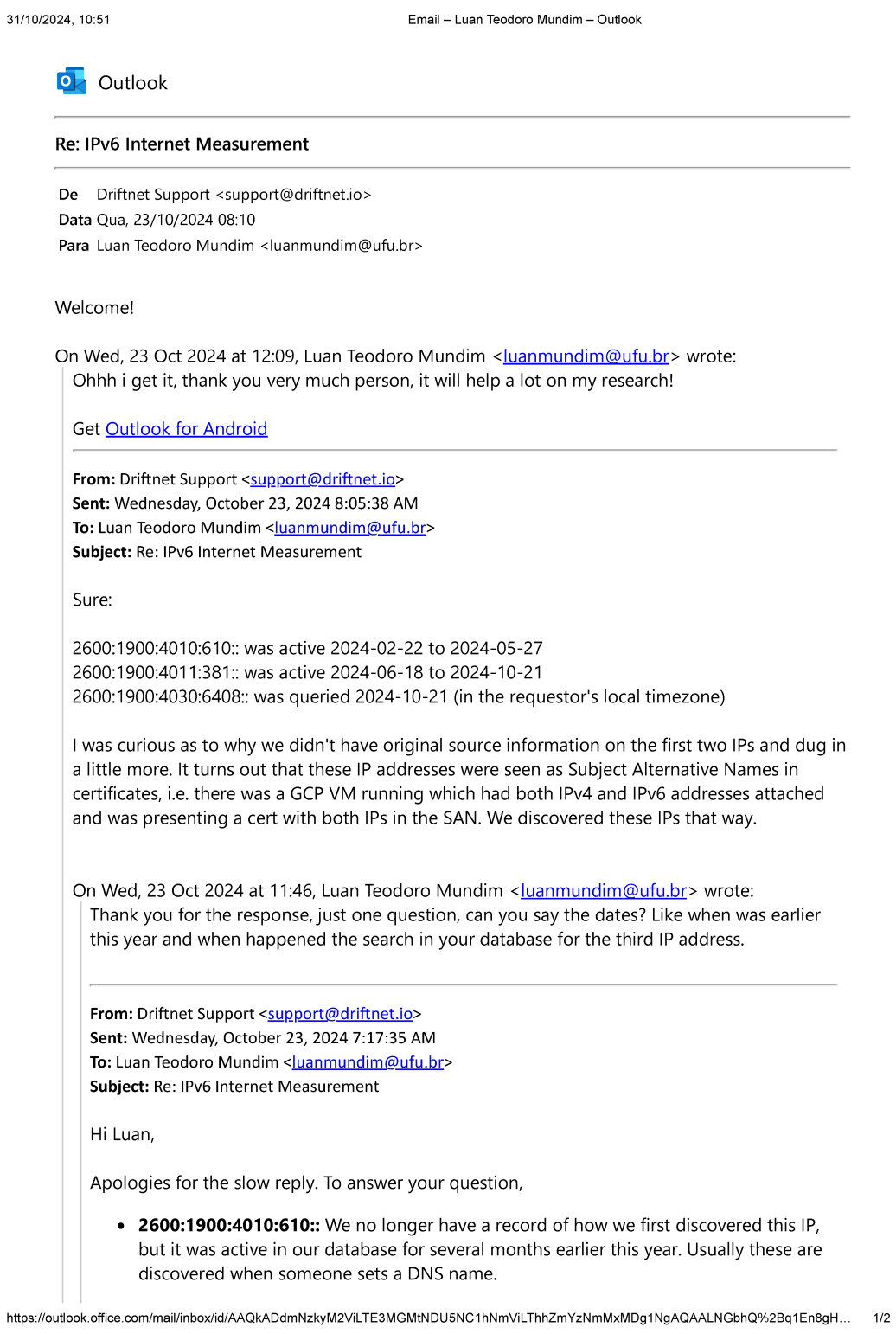
TIAN, H. C.; WANG, H. Deployment and exploration of domain name system based on ipv6. **Applied Mechanics and Materials**, 2014. Disponível em: <https://doi.org/10.4028/www.scientific.net/amm.668-669.1247>.

ZHANG, N.; LE, J.; XU, H. Y. Domain names based on fault tree analysis. **Applied Mechanics and Materials**, 2014. Disponível em: <https://doi.org/10.4028/www.scientific.net/amm.667.143>.

Apêndices

APÊNDICE **A**

Emails da driftnet.io (Constantine Cybersecurity Ltd)



31/10/2024, 10:51 Email – Luan Teodoro Mundim – Outlook

Outlook

Re: IPv6 Internet Measurement

De Driftnet Support <support@driftnet.io>
Data Qua, 23/10/2024 08:10
Para Luan Teodoro Mundim <luanmundim@ufu.br>

Welcome!

On Wed, 23 Oct 2024 at 12:09, Luan Teodoro Mundim <luanmundim@ufu.br> wrote:
> Ohhh i get it, thank you very much person, it will help a lot on my research!
>
> Get Outlook for Android
>
> **From:** Driftnet Support <support@driftnet.io>
> **Sent:** Wednesday, October 23, 2024 8:05:38 AM
> **To:** Luan Teodoro Mundim <luanmundim@ufu.br>
> **Subject:** Re: IPv6 Internet Measurement
>
> Sure:
>
> 2600:1900:4010:610:: was active 2024-02-22 to 2024-05-27
> 2600:1900:4011:381:: was active 2024-06-18 to 2024-10-21
> 2600:1900:4030:6408:: was queried 2024-10-21 (in the requestor's local timezone)
>
> I was curious as to why we didn't have original source information on the first two IPs and dug in a little more. It turns out that these IP addresses were seen as Subject Alternative Names in certificates, i.e. there was a GCP VM running which had both IPv4 and IPv6 addresses attached and was presenting a cert with both IPs in the SAN. We discovered these IPs that way.
>
> On Wed, 23 Oct 2024 at 11:46, Luan Teodoro Mundim <luanmundim@ufu.br> wrote:
> > Thank you for the response, just one question, can you say the dates? Like when was earlier this year and when happened the search in your database for the third IP address.
> >
> > **From:** Driftnet Support <support@driftnet.io>
> > **Sent:** Wednesday, October 23, 2024 7:17:35 AM
> > **To:** Luan Teodoro Mundim <luanmundim@ufu.br>
> > **Subject:** Re: IPv6 Internet Measurement
> >
> > Hi Luan,
> >
> > Apologies for the slow reply. To answer your question,
> >
> > - **2600:1900:4010:610::** We no longer have a record of how we first discovered this IP, but it was active in our database for several months earlier this year. Usually these are discovered when someone sets a DNS name.

https://outlook.office.com/mail/inbox/id/AAQkADdmNzkyM2ViLTE3MGMtNDU5NC1hNmViLThhZmYzNmMxMDg1NgAQAALNGbhQ%2Bq1En8gH… 1/2

Figura 25 – Captura de tela da resposta das trocas de e-mail com a driftnet.io (Constantine Cybersecurity Ltd) p.1

Fonte: Autor

- **2600:1900:4011:381::** As above.
- **2600:1900:4030:6408::** We saw someone query this IP on our site.

Sorry there's not more info to share here.

All the best,

-- driftnet.io

On Tue, 22 Oct 2024 at 01:11, Luan Teodoro Mundim <luanmundim@ufu.br> wrote:

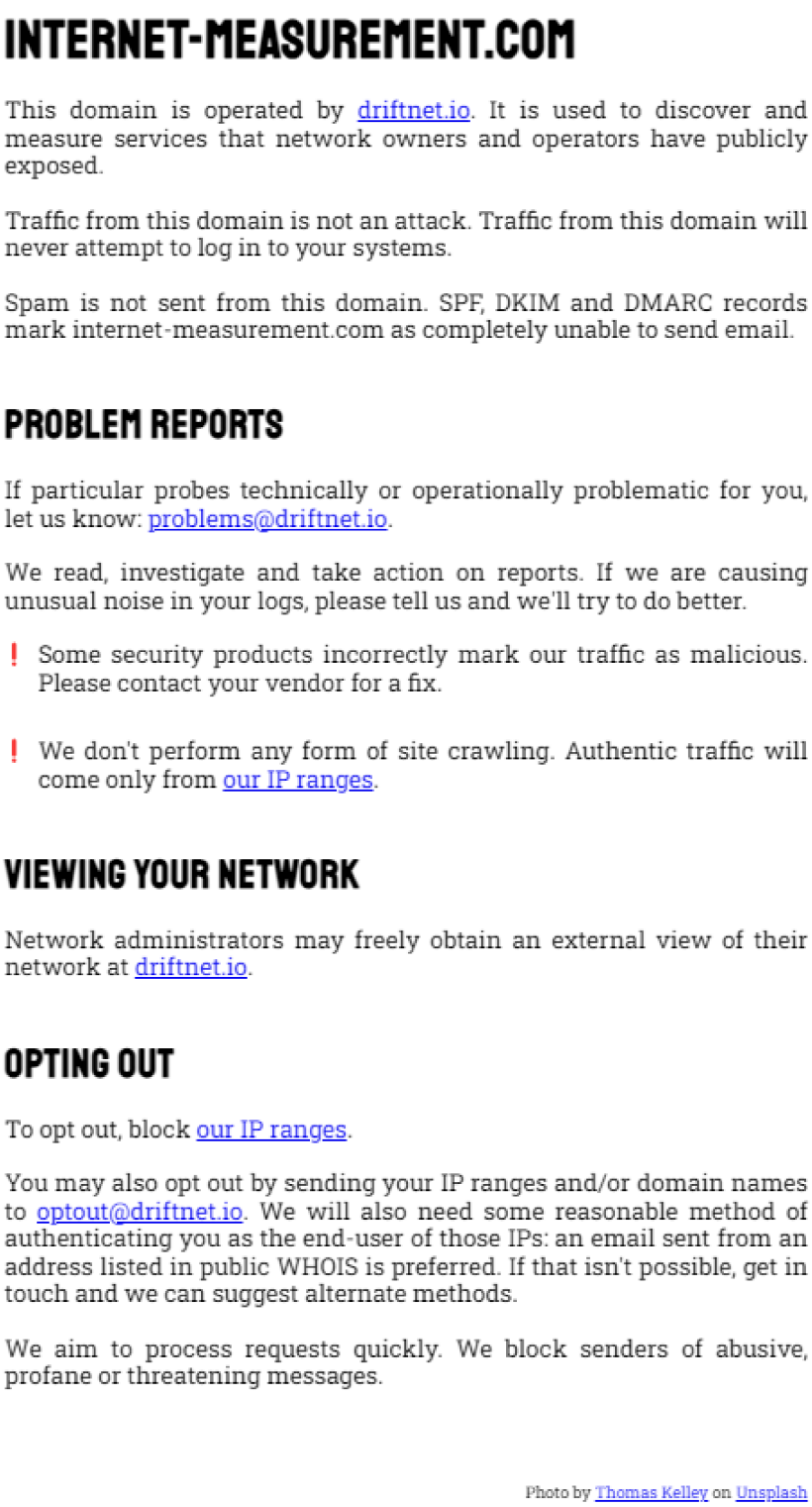
> Shure, it was these six addresses:
>
> 2600:1900:4030:6408:0:0:0:0
> 2600:1900:4011:381:0:0:0:0
> 2600:1900:4010:610:0:0:0:0
> 2600:1901:8160:335:0:0:0:0
> 2600:1900:40f0:a4de:0:1:0:0
> 2600:1900:4041:8c3:0:1:0:0
>
> ---
>
> **De:** Driftnet Support <support@driftnet.io>
> **Enviado:** segunda-feira, 21 de outubro de 2024 20:40
> **Para:** Luan Teodoro Mundim <luanmundim@ufu.br>
> **Assunto:** Re: IPv6 Internet Measurement
>
> Hi Luan,
>
> If you share your IPv6 address we'll let you know which method we used.
>
> -- driftnet.io
>
> On Mon, 21 Oct 2024 at 23:57, Luan Teodoro Mundim <luanmundim@ufu.br> wrote:
>
> > Hello and greetings. My name is Luan, and I'm a master's degree student in cybersecurity. I was testing some IPv6 addresses, and your IPv6 knocked on my door. Despite your website saying that you don't perform any type of web crawling, how could you find my IPv6 address?

Figura 26 – Resposta das trocas de e-mail com a driftnet.io (Constantine Cybersecurity Ltd) p.2

Fonte: Autor

APÊNDICE B

Página web da driftnet.io (Constantine Cybersecurity Ltd)



INTERNET-MEASUREMENT.COM

This domain is operated by driftnet.io. It is used to discover and measure services that network owners and operators have publicly exposed.

Traffic from this domain is not an attack. Traffic from this domain will never attempt to log in to your systems.

Spam is not sent from this domain. SPF, DKIM and DMARC records mark internet-measurement.com as completely unable to send email.

PROBLEM REPORTS

If particular probes technically or operationally problematic for you, let us know: problems@driftnet.io.

We read, investigate and take action on reports. If we are causing unusual noise in your logs, please tell us and we'll try to do better.

! Some security products incorrectly mark our traffic as malicious. Please contact your vendor for a fix.

! We don't perform any form of site crawling. Authentic traffic will come only from our IP ranges.

VIEWING YOUR NETWORK

Network administrators may freely obtain an external view of their network at driftnet.io.

OPTING OUT

To opt out, block our IP ranges.

You may also opt out by sending your IP ranges and/or domain names to optout@driftnet.io. We will also need some reasonable method of authenticating you as the end-user of those IPs: an email sent from an address listed in public WHOIS is preferred. If that isn't possible, get in touch and we can suggest alternate methods.

We aim to process requests quickly. We block senders of abusive, profane or threatening messages.

Photo by Thomas Kelley on Unsplash

Figura 27 – Captura de tela do website internet-measurement.com (Constantine Cybersecurity Ltd)

Fonte: internet-measurement.com

APÊNDICE C

Script de consulta da API do Shodan

```
from shodan import Shodan
from datetime import date
import json

instancesIP = ['...']
today = date.today()
API_KEY = ''
output_filename = f'shodan-{today}.json'
api = Shodan(API_KEY)
json_list = []

for ip in instancesIP:
        try:
                info = api.host(ip)
                print(info)

                json_list.append(info)

        except Exception as e:
                print ("\nIP_not_found: "+ ip + "\n")

json_file = open(output_filename, "a")
json_file.write(json.dumps(json_list))
json_file.close()
```

APÊNDICE D

Script de consulta da API do Censys

```
from censys.search import CensysHosts
from datetime import date
import json

API_ID = ''
API_SECRET = ''
api = CensysHosts(API_ID, API_SECRET)

json_list = []
today = date.today()
output_filename = f'censys-{today}.json'

instancesIP = ['...']

for ip in instancesIP:
        result = api.view(ip)
        json_list.append(result)
        print(result)

json_file = open(output_filename, "a")
json_file.write(json.dumps(json_list))
json_file.close()
```

APÊNDICE E

Perguntas realizadas nos fóruns

Tabela 7 – Perguntas que foram realizadas dentro dos fóruns de cibersegurança com endereços IPv6 anonimizados.

Fórum	Pergunta
Hidden Answers	I got a problem, someone using this IP Address: x:x:x:x:x:x tried to hack me, can someone help me?
Raddle	how can i find the hostname for this IP address x:x:x:x:x:x ?
Deep Answers	why this scan its not working? nmap -Ss x:x:x:x:x:x
Nulled	Can a link-local address in IPv6 communicate with other network interface addresses? Assuming I have a network interface named 'int1': *35: int1: <BROADCAST,NOARP,UP,LOWER_UP>mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000 inet6 x:x:x:x:x:x/128 scope global valid_lft forever preferred_lft forever** Is it possible for the link-local address of int1 to communicate with others addresses?"
Black Hat World	how can i scan this IP Address x:x:x:x:x:x ?
Hacknology	i am seeing unusual network traffic from x:x:x:x:x:x, isthis something to be concerned about?

Fonte: Autor