Identificação e Categorização de Endereços IP Dinâmicos com Dados Públicos

Gabriel Pains de Oliveira Cardoso
Departamento de Ciência da Computação (DCC) - UFMG
gabriel.cardoso@dcc.ufmg.br

Ítalo Fernando Scotá Cunha
Departamento de Ciência da Computação (DCC) - UFMG
cunha@dcc.ufmg.br

***Resumo*—The identification and tracking of devices running services and applications in network vulnerability studies and forensic analysis rely heavily on the assumption of static IP addresses. However, the dynamic allocation of IP addresses, driven by IPv4 exhaustion and cloud computing environments, makes it harder to map behaviors, vulnerabilities, and attack sources to devices. Methods for classifying an IP as static or dynamic exist, but they face limitations such as the reliance on outdated or incomplete Reverse DNS (RDNS) records, or the need for private data. This work expands on DynMap, an algorithm that identifies dynamically-allocated IP blocks from public fingerprint data, previously applied only to Shodan HTTPS certificate scans. We stress and generalize DynMap by applying the same algorithm to two longitudinal Shodan datasets, HTTPS certificates and SSH host keys, plus a shorter, complementary set of zgrab2 TLS scans of the Tranco list, while sweeping its highest-impact parameter, the minimum block size. Our central finding is that the fingerprint type fundamentally shapes what DynMap observes: HTTPS certificate scans are dominated by dynamic addresses, whereas SSH host keys instead reveal shared infrastructure, default keys cloned across customer devices, rather than dynamism. The Tranco set was explored as complementary data but lacked a long enough observation window for meaningful results; since it uses the same fingerprint as the HTTPS set, we expect comparable findings given sufficient longitudinal data. The block-size parameter offers an adjustable trade-off between precision and coverage.**

***Index Terms*—Dynamic IP; IP Identification; Fingerprinting; DynMap; Internet Measurement**

I. Introdução

O estudo de vulnerabilidades, comportamentos e análise forênsica de redes requer a identificação e rastreamento de dispositivos (*hosts*) executando serviços e aplicações. Técnicas que utilizam o endereço IP como identidade de um dispositivo assumem que endereços IP são alocados estaticamente a um dispositivo. No entanto, a exaustão de endereços IPv4 [6] bem como a dinamicidade da alocação e virtualização de recursos em ambientes de computação em nuvem [7] implicam em maior dinamicidade da alocação de endereços IP a múltiplos dispositivos ao longo do tempo.

Endereços IP alocados dinamicamente dificultam a análise de comportamentos, adicionando ambiguidades no mapeamento de comportamentos ou vulnerabilidades para dispositivos. Por exemplo, um comportamento ou serviço identificado em um IP dinâmico e_1 pode não ser encontrado em uma varredura (*scan*) posterior porque o serviço agora está executando no IP e_2, mesmo quando o serviço continua ativo. Ainda, criminosos podem fazer uso de diversos dispositivos atuando em IPs diferentes no decorrer do tempo para realizar ataques do tipo DDoS (*Distributed Denial of Service*) e arquitetar *botnets* para enviar mensagens e e-mails de *phishing*. A natureza dos endereços IP dinâmicos dificulta a prevenção e remediação de tais ataques, pois o rastreamento e identificação da origem dos ataques se torna mais desafiador.

Nesse cenário, não existe uma base de dados consolidada de endereços IP dinâmicos que possa ser usada como referência. Apesar de existirem técnicas para classificar um IP em estático ou dinâmico, elas apresentam limitações que reduzem sua aplicabilidade. Uma técnica comum consiste em procurar o DNS reverso (*Reverse DNS*, RDNS) do IP que se deseja classificar e analisar o nome de domínio associado à procura de indicadores de dinamicidade. No entanto, muitos IPs não possuem RDNS configurado ou podem ter nomes desatualizados, fazendo com que a precisão de análises dos nomes de domínio não seja confiável [8]. Por exemplo, apenas 30–35% dos endereços IPv4 possuem RDNS na base de dados da Rapid7 [4], [20]. Outra abordagem para identificação de IPs dinâmicos é capturar uma miríade de dados a fim de caracterizar o uso de um bloco de endereços IP [3]. No entanto, tais dados podem ser sigilosos ou disponíveis somente em contextos específicos, por exemplo quando os dados necessários são pertinentes a um conjunto limitado de IPs.

Nesse contexto, o algoritmo DynMap [1] se insere como um método de remediar a necessidade de dados privados ou de difícil acesso ao utilizar dados públicos para suas análises. Em seu trabalho original, os autores exploram a aplicação do DynMap nos dados públicos do Shodan [15], encontrando mais de 140 mil IPs dinâmicos dependendo da configuração utilizada, e realizando uma análise qualitativa breve sobre os IPs encontrados.

Assim, o presente trabalho busca expandir e testar o DynMap para além do cenário original, investigando se o método generaliza para outros tipos de *fingerprint* (chaves SSH além de certificados TLS). Para isso, aplicamos o DynMap a três conjuntos de dados, varreduras HTTPS do Shodan, varreduras SSH do Shodan e, como dado complementar, varreduras TLS (zgrab2) da lista Tranco [17], e testamos o parâmetro de maior impacto, o tamanho mínimo de bloco b. Este trabalho se divide em duas etapas do Projeto Orientado em Computação (POC):

Etapa 1 - Projeto Orientado em Computação 1 (POC1):

- Escolher fontes de dados adicionais.

- Coletar IPs alvo de varreduras das fontes adicionais.
- Escanear os IPs alvos ao longo do tempo.
- Organizar varreduras obtidas.

Etapa 2 - Projeto Orientado em Computação 2 (POC2):

- Revisão do progresso do POC1.
- Aplicar o DynMap nos conjuntos resultantes de varreduras.
- Aplicar heurísticas para categorização dos IPs encontrados.
- Comparar as inferências e tipos de IP encontrados.

A proposta inicial (POC1) incluía o PhishTank [18] como conjunto de IPs maliciosos, para contrastar com IPs legítimos do Tranco. No entanto, a varredura de sítios de *phishing* não pôde ser realizada a partir da rede da UFMG por motivos legais e de segurança, e a ideia foi descartada nesta etapa. As principais contribuições deste trabalho são: (i) a demonstração de que o DynMap generaliza para além do Shodan HTTPS; (ii) a evidência de que a escolha do *fingerprint* molda fundamentalmente o resultado; e (iii) uma reprodução dos resultados do trabalho original sobre um conjunto de dados maior.

II. Referencial Teórico e Trabalhos Relacionados

Diversos trabalhos atacaram o desafio de identificar endereços IP alocados dinamicamente. Neste contexto, a riqueza dos dados utilizados determina um compromisso fundamental entre (i) a generalidade do método, *i.e.*, a possibilidade de ser aplicado a diversos contextos, e (ii) a precisão dos resultados obtidos. Métodos que utilizam dados amplamente disponíveis são aplicáveis de forma mais geral, enquanto métodos com dados privados (e potencialmente sensíveis) são aplicáveis apenas pelas entidades de posse dos dados, mas resultam em inferências mais precisas.

No extremo de maior generalidade e menor precisão, temos trabalhos que utilizam, *e.g.*, RDNS para inferência de IPs dinâmicos [4], [8], [9]. Estes trabalhos são amplamente aplicáveis, visto que registros RDNS podem ser facilmente obtidos. Porém, registros RDNS estão disponíveis para uma fração pequena dos endereços IP na Internet [4] e, mesmo quando existem, podem estar desatualizados ou não conter informação que permita a identificação de um endereço IP como dinâmico [8]. Uma melhoria nesse processo é realizar a identificação em blocos, utilizando características sequenciais dos registros RDNS [9], explorando propriedades de IPs e a alocação em blocos, assim como no UDmap e no DynMap. Mesmo assim, essa melhoria está limitada à disponibilidade de registros RDNS.

No extremo oposto temos trabalhos que utilizam dados específicos de um serviço ou rede. Em [3], os autores têm acesso interno à rede, observando o tráfego e coletando métricas internas para a realização das análises. No método UDmap [2], no qual o DynMap se baseia, os autores utilizaram logs privados do Hotmail, que continham várias informações dos usuários, como i) IPs utilizados, ii) horário de acesso e iii) identificador único do usuário. Para a Microsoft, este conjunto de dados é atrativo pois tem grande abrangência do espaço de endereçamento IP e permite mapeamento preciso de qual usuário está utilizando um endereço IP. Estas técnicas trazem resultados precisos, mas são aplicáveis a um conjunto reduzido de endereços IP ou apenas por agentes específicos que possuem acesso aos dados. A dependência de dados sigilosos ou específicos torna tais métodos inviáveis para contextos gerais.

O DynMap [1] é uma solução intermediária entre estes dois extremos, estendendo e complementando as soluções existentes. Ele utiliza apenas informações públicas, mas que trazem mais informações do que o RDNS, permitindo inferências mais precisas por qualquer entidade. Em particular, o DynMap utiliza dados que podem ser obtidos por ferramentas de escaneamento como o ZGrab [19], ou já escaneados, como os dados disponíveis no Shodan e Censys [15], [16]. Para ser capaz de identificar IPs dinâmicos, o DynMap precisa das seguintes informações por endereço IP analisado:

- Endereço IP.
- Porta escaneada.
- Identificador de dispositivo (certificado).
- Nome de domínio associado ao endereço IP.
- *Timestamp* do escaneamento.

Diferente do UDmap, que utiliza um identificador único do usuário para rastrear usuários entre IPs, o DynMap não conta com um identificador preciso para associar um serviço a um endereço IP. Assim, é necessário um metadado que (1) esteja presente em uma quantidade significativa de IPs e (2) cujo valor seja único para cada dispositivo. Em outras palavras, é necessário um metadado que se comporte como um identificador único, para que possa ser usado para rastrear dispositivos e serviços entre IPs ao longo do tempo, a fim de inferir se estes IPs são dinâmicos.

No trabalho original do DynMap é utilizada a assinatura SHA256 de certificados X.509 recebidos de servidores HTTPS como um identificador[1] para rastrear um serviço [10], e por associação a máquina onde está executando, entre diferentes endereços IPs. Essa assinatura é única para cada certificado e, em geral, apresenta o comportamento necessário para o rastreamento de um serviço. Certificados X.509 podem ser capturados em escala global de servidores HTTPS por ferramentas como o ZGrab e estão disponíveis em sistemas de varredura como Shodan e Censys. Apesar disso, *websites* precisam renovar certificados periodicamente e podem compartilhar certificados (apesar desta prática não ser recomendada). Estas ocorrências geram ambiguidades na identificação de serviços e, consequentemente, no mapeamento de serviços para os endereços IP utilizados pelos dispositivos hospedeiros. O DynMap possui medidas para contornar esses casos, discutidas na próxima seção. Por fim, quanto maior o período analisado, mais informações o DynMap possui para identificar IPs; no trabalho original foram utilizadas varreduras de três meses do Shodan.

[1] a partir desta seção, identificador é referido como certificado ou, de forma geral, como *fingerprint*.

III. Fundamentos do DynMap

O DynMap herda suas etapas do UDmap com adaptações para execução em dados públicos. Estas adaptações são necessárias, pois dados públicos, em geral, fornecem menos métricas e granularidade do que dados privados, como os logs do Hotmail utilizados pela Microsoft Research. As subseções a seguir detalham as etapas do DynMap na sua aplicação a um conjunto de IPs qualquer.

A. Seleção de Blocos Candidatos

O DynMap identifica um endereço IP como possivelmente alocado dinamicamente caso este apresente mais de um certificado no período observado. Endereços IP que não são observados no período (*i.e.*, que não estão presentes nos dados), ou que possuem apenas um certificado, são considerados como *lacunas*.

O DynMap inicia identificando blocos de IPs que podem ser dinâmicos, denotados blocos *candidatos*. O algoritmo constrói um bloco candidato de B IPs contíguos com as seguintes propriedades:

- 1. IPs em um bloco candidato devem pertencer ao mesmo AS e possuir o mesmo prefixo BGP [5].
- 2. Um bloco candidato não pode ter lacunas com mais de L IPs consecutivos.
- 3. Cada bloco deve possuir um mínimo de B' IPs, $B \geq B'$.

Figura 1. Ilustração do espaço de endereços IP e os blocos selecionados pelo DynMap.

Pela propriedade (1), o DynMap garante que IPs dentro de um mesmo bloco estão sob um único domínio administrativo e estão topologicamente relacionados. As propriedades (2) e (3) garantem que o algoritmo observe uma fração significativa de IPs dinâmicos em cada bloco. O efeito no espaço de endereços IP da aplicação dessas propriedades pode ser visto na Figura 1. Para a consulta do AS e do prefixo BGP de cada IP, o DynMap utiliza a biblioteca PyASN [12].

Dessa forma, nesta primeira etapa de construção de blocos candidatos, grandes seções contíguas de endereços IP são imediatamente descartadas devido aos seus IPs estarem ausentes dos dados ou apresentarem apenas um certificado. Assim, os blocos candidatos construídos nessa etapa já possuem indícios de comportamento dinâmico e serão filtrados de maneira mais rigorosa nas etapas seguintes.

B. Cálculo da Entropia de Uso por IP

Após construir blocos, o DynMap calcula a *entropia de uso* por IP. Esse cálculo é feito bloco a bloco para os IPs naquele bloco, e captura se um certificado é observado de forma uniforme dentro de um bloco. Para cada IP em um bloco, o DynMap calcula a fração de certificados observados no IP que também foram observados em outros IPs do bloco. A entropia E_i para um IP $i \in B$ calcula se os certificados que aparecem em i têm uma probabilidade igual de serem observados nos outros endereços IP do bloco onde i se encontra.

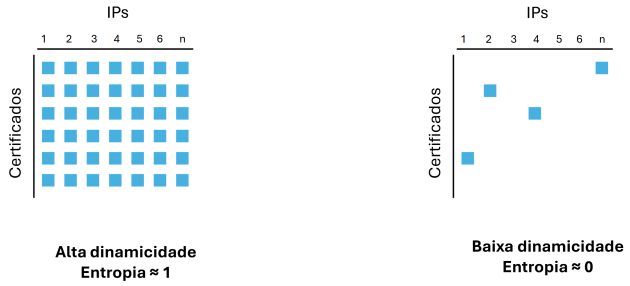

Figura 2. Matriz de correlação *Certificado x IPs* utilizada pelo DynMap.

O DynMap realiza esse cálculo através de uma matriz de correlação *Certificado x IPs* (Figura 2). Uma entropia próxima de 1 indica que a maioria dos certificados aparece na maioria dos IPs do bloco (*i.e.*, o mesmo grupo de certificados está presente em vários IPs diferentes ao longo do tempo), um forte indicativo de alocação dinâmica. Uma entropia próxima de zero indica que a maioria dos certificados fica concentrada em um pequeno número de IPs, indicando baixa dinamicidade. Essa métrica é normalizada na forma da *Normalized Sample IP Usage-Entropy* (NSUE), que varia de 0 a 1 e minimiza o impacto de séries temporais curtas, sendo a métrica utilizada pelo DynMap nas etapas seguintes.

C. Dinamicidade de Domínios por IP

Como discutido na Seção II, ambiguidades acontecem quando o uso de certificados em servidores HTTPS não permite a identificação de um serviço e o mapeamento para o endereço IP do hospedeiro. Para contornar estes casos, o DynMap calcula uma segunda métrica para caracterizar blocos de endereços IP que chamamos *dinamicidade de domínio*, que captura se um domínio é observado de forma uniforme em múltiplas assinaturas.

Para cada porta com um serviço HTTPS sondado em cada IP de um bloco, calculamos a razão entre o número de domínios e o número de assinaturas que aparecem ao longo do tempo. Se a razão é próxima de 1, todas as assinaturas possuem um domínio distinto. Neste caso, temos um mapeamento entre domínios e assinaturas: a mudança de assinatura indica que o IP é utilizado por diferentes serviços e possivelmente dinâmico (*e.g.*, servidores de um provedor de computação em nuvem alocados a diferentes clientes ao longo do tempo). Por outro lado, se a razão é próxima de zero, então poucos domínios (possivelmente um) possuem várias assinaturas (*e.g.*, devido a mudança ou renovação de certificados), indicando que o endereço IP pode ser estático e que é utilizado por poucos serviços. Este cálculo precisa ser realizado para cada porta separadamente, pois é possível que uma máquina execute vários serviços em diferentes portas. Por exemplo, um dispositivo com um endereço IP estático executando múltiplos serviços

em portas diferentes apresenta múltiplas assinaturas e poderia ser confundido com um endereço IP dinâmico.

Após calcular a razão do número de domínios e assinaturas para cada porta, selecionamos o mínimo entre todas as portas como a dinamicidade de domínio de cada endereço IP, para diminuir a influência de cenários anômalos. Por exemplo, para um IP estático a maioria das portas têm razão próxima de zero, mas uma porta executando um ambiente de testes pode ter razão próxima de 1; utilizar o mínimo evita inferir endereços IP estáticos como dinâmicos.

D. Análise de Comportamento e Classificação de Blocos

Nesta etapa combinamos a entropia de uso e a dinamicidade de domínios para reduzir erros e refinar a inferência de blocos de IP dinâmicos. A Figura 3 mostra os histogramas da entropia de uso e da dinamicidade de domínios para os IPs dos blocos candidatos. As distribuições de ambas as métricas são bimodais, isto é, as métricas indicam fortemente que os endereços IP são estáticos ou dinâmicos, com poucos IPs apresentando valores intermediários. Como as distribuições são bimodais, temos cinco combinações comuns de métricas que cobrem o comportamento de um endereço IP. Aplicamos as seguintes regras em cada IP para combinar as duas métricas em uma única métrica final, que será utilizada na etapa de identificação de sub-blocos de IPs dinâmicos:

- **Caso 1.** Se as duas métricas são próximas de zero, então ambas indicam que o endereço IP é estático. Para estes IPs calculamos a média das duas métricas, o que resulta em uma métrica final próxima de zero. Estes IPs tendem a ser descartados na etapa de construção de sub-blocos.
- **Caso 2.** Se as duas métricas são próximas de 1, então ambas indicam que o endereço IP é dinâmico. Novamente calculamos a média das duas métricas, resultando em uma métrica final alta.
- **Caso 3.** Se a entropia de uso é próxima de zero e a dinamicidade de domínio é próxima de 1, este IP não compartilhou muitas assinaturas com outros endereços IP do bloco, mas, devido à mudança de domínio na maioria ou em todas as suas portas, é provável que seja dinâmico. A métrica final é definida como a dinamicidade de domínio, próxima de 1.
- **Caso 4.** Se a entropia de uso é próxima de 1 e a dinamicidade de domínio é próxima de zero, utilizamos a entropia de uso como métrica final e estendemos a análise. Calculamos as razões (i) entre o número de assinaturas do IP e a quantidade de IPs no bloco (F_{ass}) e (ii) entre o número de domínios do IP e a quantidade de IPs no bloco (F_{dom}). Se F_{ass} for menor que 0,5, temos poucas assinaturas que aparecem em muitos endereços IP do bloco ao longo do tempo. Nesse caso, pode ser que este endereço IP seja utilizado por um proxy reverso ou balanceador de carga [11], e marcamos este IP como *proxy*. Se F_{ass} for maior que 0,5 e F_{dom} for menor que 0,5, então temos muitas assinaturas e poucos domínios, indicando que uma grande quantidade de certificados é utilizada por vários serviços; neste caso marcamos o IP como sendo de um *servidor compartilhado*. Se ambas F_{ass} e F_{dom} são maiores que 0,5, temos muitos domínios e muitas assinaturas e não fazemos nenhuma classificação especial, mantendo o IP marcado simplesmente como *dinâmico*. A classificação em proxy e servidores compartilhados resolve as duas limitações de falsos positivos discutidas para o UDmap; os estudos de caso que motivaram estas classes são apresentados na Seção VII.
- **Caso 5.** Se alguma métrica não é próxima de zero nem de 1, então calculamos a métrica final como a média da entropia de uso e da dinamicidade de domínio. É possível que esse IP seja um caso de borda (*outlier*); sua inclusão ou descarte dependerá das métricas de seus vizinhos e da saída do filtro de mediana.

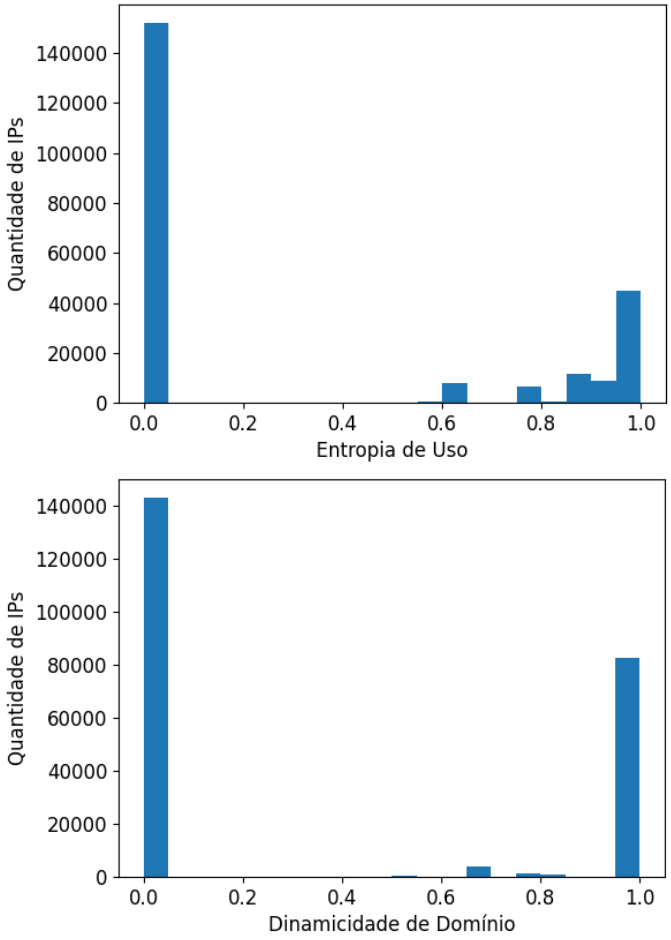

Figura 3. Entropia de uso (acima) e dinamicidade de domínio (abaixo) para os endereços IP em blocos candidatos (antes da aplicação dos filtros). Ambas as distribuições são bimodais.

Após processar os IPs de um bloco seguindo as regras acima, o tipo de cada bloco é definido como o tipo prevalente entre seus IPs (*i.e.*, dinâmico, proxy ou servidores compartilhados). Em geral, o tipo mais prevalente é unânime e captura o comportamento dos IPs do bloco como um todo; *e.g.*, 80% dos blocos candidatos com $B = 8$ têm prevalência maior que 70%, e 90% dos blocos candidatos com $B = 128$ têm prevalência

Tabela I
CONJUNTOS DE DADOS ANALISADOS COM O DYNMAP.

Conjunto	Fonte	*Fingerprint*	Domínio	IPs de entrada
Shodan-HTTPS	Varreduras HTTPS (Shodan)	SHA-256 do cert. X.509	*common name*	2 328 491
Shodan-SSH	Varreduras SSH (Shodan)	chave de *host* SSH	DNS reverso	707 130
Tranco-TLS	TLS (zgrab2) da lista Tranco	SHA-256 do cert. X.509	*common name*	47 591

maior que 90%, indicando que a unanimidade aumenta com o tamanho do bloco.

E. Identificação de Sub-blocos de IP Dinâmicos

A métrica de dinamicidade calculada acima é utilizada em um processo de suavização com uma janela deslizante (filtro de mediana), com a alteração, em relação ao UDmap, de que tomamos sempre o máximo entre o valor da janela e a métrica de dinamicidade de um endereço IP. Isto faz com que a janela deslizante possa aumentar a métrica de dinamicidade de um IP (*e.g.*, que faça parte de uma lacuna), mas nunca possa reduzir a dinamicidade de um IP observado no conjunto de dados de entrada. Como a janela deslizante atualiza a métrica de um IP que não foi observado utilizando as métricas de seus vizinhos, esta etapa transfere a inferência dos endereços IP presentes para as lacunas, melhorando a cobertura do método.

Após a aplicação do filtro, o DynMap segmenta cada bloco candidato em sub-blocos, descartando as "quedas"remanescentes, *i.e.*, IPs cuja métrica de dinamicidade permanece abaixo de um limiar T. As seções *contíguas* de IPs restantes formam os sub-blocos, e cada sub-bloco herda o tipo do bloco candidato em que se encontra. Por fim, os IPs contidos nos sub-blocos são coletados por tipo nas listas de IPs dinâmicos, proxy e servidores compartilhados; IPs classificados como *outlier* são descartados.

IV. METODOLOGIA E CONJUNTOS DE DADOS

A. Conjuntos de Dados

Avaliamos o DynMap sobre três conjuntos de dados, resumidos na Tabela I. O *pipeline* de execução é independente da fonte de dados: cada fonte é convertida, em uma etapa de pré-processamento, para um formato comum de entrada, e o *mesmo* algoritmo é executado sobre os três conjuntos, mudando apenas o *fingerprint* de entrada e seu domínio associado.

O Shodan-HTTPS reproduz e estende o cenário do trabalho original, agora sobre um conjunto consideravelmente maior (mais de 2,3 milhões de IPs de entrada) ao longo de 18 meses de escaneamentos. O Shodan-SSH troca o *fingerprint* de certificado X.509 pela chave de *host* SSH, utilizando como domínio o nome obtido por DNS reverso (já incluído na varredura fornecida pelo Shodan). Este conjunto testa a generalização do método para um *fingerprint* de natureza distinta, também ao longo de 18 meses. O Tranco-TLS fornece IPs dos *websites* mais visitados da Internet [17], contando com mecanismos contra manipulação de ranqueamentos, e é explorado neste trabalho como dado complementar. Suas varreduras cobrem aproximadamente um mês, janela temporal bem mais curta que os 18 meses dos conjuntos do Shodan.

Como mencionado na introdução, a proposta inicial incluía o PhishTank [18] como fonte de IPs maliciosos. A varredura de *websites* de *phishing* não pôde ser realizada a partir da rede da UFMG por motivos legais e de segurança, e a fonte foi descartada nesta etapa.

B. Coleta e Pré-processamento

Para o Shodan, as varreduras HTTPS e SSH coletam e exportam o certificado X.509 (incluindo a assinatura SHA-256 e o *common name*) ou a chave de *host* SSH quando a porta correspondente é sondada. Para o Tranco, utilizamos uma máquina Linux executando o ZGrab [19] periodicamente; entre os dados obtidos estão o endereço IP, a porta escaneada, o estado do escaneamento, o certificado, o nome de domínio (se houver) e a data do escaneamento, com serviços executando primariamente nas portas 80, 443, 8080 e 4433.

Cada fonte é então convertida para o formato comum de entrada do DynMap, que relaciona, para cada IP, o conjunto de *fingerprints* observados e a respectiva série temporal de observações (*timestamp*, *fingerprint*, porta e domínio). Essa separação entre pré-processamento e análise permite adicionar novas fontes de dados sem alterar o algoritmo.

C. Parametrização

O DynMap possui quatro parâmetros: B (tamanho mínimo de bloco candidato), L (maior lacuna permitida), T (limiar de dinamicidade para a construção de sub-blocos) e J (tamanho da janela do filtro de mediana). O tamanho mínimo do bloco B é um dos parâmetros com maior impacto nos resultados: quanto maior B, menor a quantidade de blocos candidatos e de endereços identificados, e, como os IPs de um bloco devem pertencer ao mesmo AS e prefixo BGP, ASes com poucos IPs deixam de produzir blocos. Em outras palavras, B define um compromisso entre precisão e cobertura. O tamanho máximo de lacunas L atua de forma oposta, flexibilizando a presença de lacunas e permitindo blocos maiores e mais granulares.

Seguindo o trabalho original, focamos a avaliação na varredura do parâmetro de maior impacto, fixando os demais. Aplicamos o DynMap a cada conjunto variando $b \in \{8, 16, 32, 64, 128, 256\}$, com $g = 8$, $t = 0,75$ e $w = 5$.

V. RESULTADOS

A. Comportamento

Observamos que, com a mesma configuração ($b = 8$), o *mesmo* algoritmo produz perfis de categoria bastante distintos conforme o tipo de *fingerprint*, como mostram a Tabela II e a Figura 4. O Shodan-HTTPS é dominado por IPs dinâmicos (79,9%), enquanto o Shodan-SSH é dominado por servidores compartilhados (que, somados aos blocos proxy, representam 95,9% da saída, Seção VII). O Tranco-TLS, explorado como dado complementar, produziu pouquíssimos IPs (0,7% de cobertura), insuficientes para conclusões, seus valores constam nas tabelas apenas para completude.

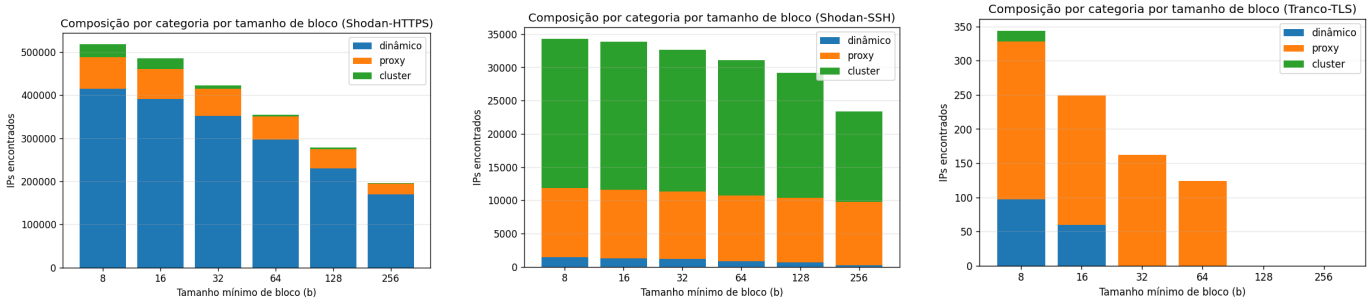

Figura 4. Composição por categoria (dinâmico/proxy/servidores compartilhados) por tamanho de bloco, da esquerda para a direita: Shodan-HTTPS, Shodan-SSH e Tranco-TLS.

Tabela II
COMPOSIÇÃO DA SAÍDA POR CONJUNTO ($b = 8$).

Conjunto	Saída	Cobertura	din.	proxy	comp.	ASes
Shodan-HTTPS	518 578	22,3%	79,9%	14,2%	5,9%	1 976
Shodan-SSH	34 308	4,9%	4,1%	30,3%	65,6%	101
Tranco-TLS	344	0,7%	28,2%	67,2%	4,7%	14

Tabela III
VARREDURA DO TAMANHO MÍNIMO DE BLOCO b PARA OS TRÊS CONJUNTOS.

Conjunto	b	Saída	Cob.	din.	proxy	comp.	ASes	Blocos candidatos
Shodan-HTTPS	8	518 578	22,3%	414 218	73 862	30 498	1 976	48 153
Shodan-HTTPS	16	485 150	20,8%	390 623	70 404	24 123	1 403	24 247
Shodan-HTTPS	32	422 079	18,1%	351 341	63 291	7 447	463	10 589
Shodan-HTTPS	64	355 046	15,2%	296 452	53 864	4 730	238	4 522
Shodan-HTTPS	128	277 919	11,9%	230 239	44 229	3 451	127	1 716
Shodan-HTTPS	256	195 611	8,4%	169 186	25 279	1 146	42	558
Shodan-SSH	8	34 308	4,9%	1 417	10 383	22 508	101	6 607
Shodan-SSH	16	33 851	4,8%	1 297	10 307	22 247	64	3 810
Shodan-SSH	32	32 645	4,6%	1 151	10 141	21 353	46	1 754
Shodan-SSH	64	31 114	4,4%	869	9 825	20 420	30	592
Shodan-SSH	128	29 208	4,1%	656	9 707	18 845	17	216
Shodan-SSH	256	23 390	3,3%	251	9 469	13 670	5	74
Tranco-TLS	8	344	0,7%	97	231	16	14	90
Tranco-TLS	16	249	0,5%	60	189	0	4	11
Tranco-TLS	32	162	0,3%	0	162	0	2	2
Tranco-TLS	64	124	0,3%	0	124	0	1	1
Tranco-TLS	128	0	0,0%	0	0	0	0	0
Tranco-TLS	256	0	0,0%	0	0	0	0	0

B. Precisão e Cobertura

À medida que b cresce, exigimos blocos maiores, então ASes pequenos são descartados, a saída e a cobertura caem, e a inferência se concentra em poucos ASes grandes. A Tabela III apresenta a varredura completa para os três conjuntos. Os conjuntos menores zeram no topo da varredura (o Tranco-TLS não produz nenhuma saída a partir de $b = 128$). As Figuras 5 e 6 mostram, respectivamente, a queda da cobertura e a concentração em poucos ASes (de 1.976 para 42 no HTTPS, e de 101 para 5 no SSH).

A concentração em ASes grandes pode ser observada na Figura 7, que mostra a distribuição do número de IPs encontrados por AS no Shodan-HTTPS, e na Figura 8, que mostra a CCDF do tamanho dos ASes. Para valores menores de b, ASes de organizações menores (*e.g.*, ISPs regionais) estão mais presentes; com tamanhos de bloco maiores e mais restritivos, esses ASes são descartados e a inferência se concentra em ASes de grande porte.

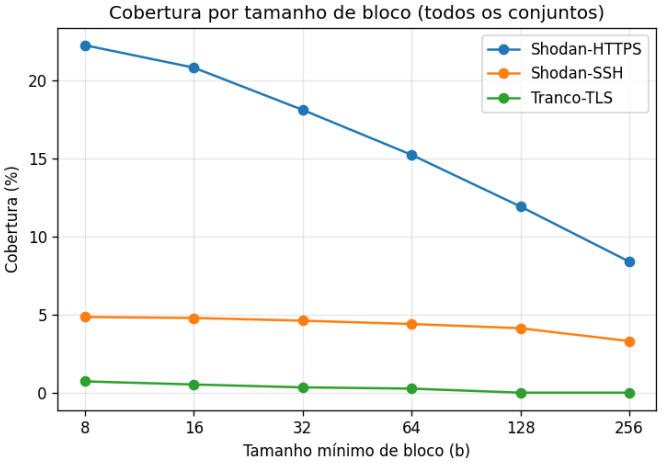

Figura 5. Cobertura (% da entrada) em função de b, uma linha por conjunto.

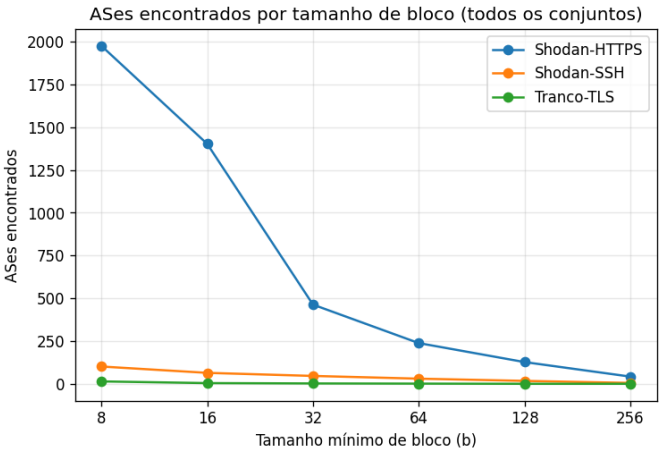

Figura 6. Número de ASes com endereços identificados em função de b.

C. ASes mais Frequentes

A Tabela IV resume os maiores ASes por número de IPs encontrados em cada conjunto, com a categoria predominante. As Tabelas V a IX apresentam, para referência, os 10 maiores ASes de cada conjunto, com a contagem detalhada por categoria.

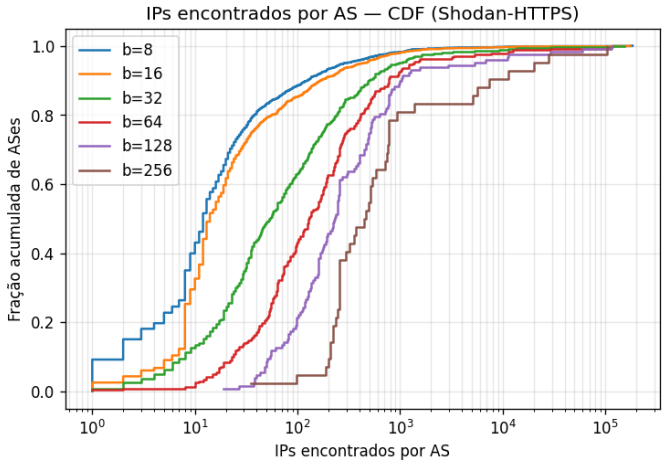

Figura 7. Distribuição (CDF) do número de IPs encontrados por AS no Shodan-HTTPS, uma linha por valor de b.

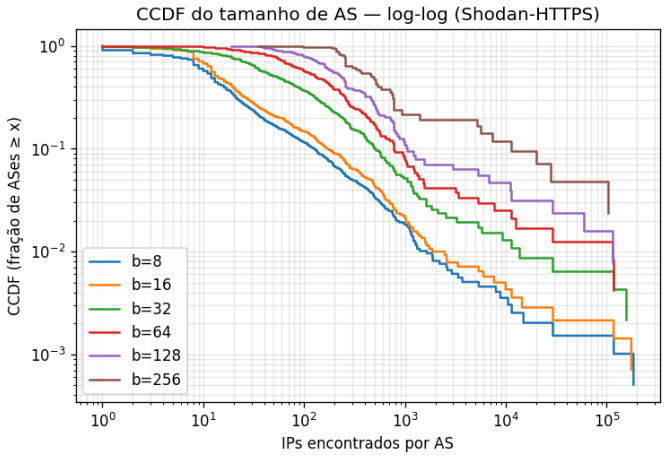

Figura 8. CCDF do tamanho dos ASes com endereços identificados no Shodan-HTTPS.

VI. Análise

A. Shodan-HTTPS

Os maiores ASes do Shodan-HTTPS são provedores de computação em nuvem, hospedagem e ISPs (Amazon, Alares, Akamai, Locaweb, Microsoft, Google), exatamente o perfil esperado de IPs dinâmicos e consistente com o trabalho original [1]. Observamos que os blocos da Microsoft (AS8075) são majoritariamente proxy (cerca de 7,1 mil de 10,5 mil IPs em $b = 8$), comportamento consistente com um *front-end* reverso centralizado. A redução drástica do número de IPs da Amazon ao aumentar b (de 184 mil para 60 mil) indica que existem vários blocos pequenos e fragmentados da Amazon no espaço de endereços, ignorados quando o tamanho de bloco cresce e o método foca em blocos grandes e uniformes.

B. Shodan-SSH

Ao usar a *fingerprint* de chave de *host* SSH, os IPs dinâmicos são quase eliminados (4,1%) e a saída se torna

Tabela IV
Maiores ASes por conjunto (% da saída) e categoria predominante. Quando aplicável, mostra-se b=8 → b=128.

Conjunto	AS	Organização	b=8	b=128	Predominante
Shodan-HTTPS	AS16509	Amazon	35,5%	21,4%	din.
Shodan-HTTPS	AS28220	Alares (ISP)	22,7%	41,1%	din.
Shodan-HTTPS	AS16625	Akamai	5,6%	10,5%	din.
Shodan-HTTPS	AS27715	Locaweb	2,9%	4,0%	proxy
Shodan-HTTPS	AS8075	Microsoft	2,0%	2,4%	proxy
Shodan-SSH	AS26615	TIM S/A (BR)	76,1%	83,0%	comp.
Shodan-SSH	AS263317	VoxTech (BR)	3,6%	4,3%	proxy
Shodan-SSH	AS202422	G-Core Labs	2,2%	1,4%	din.
Tranco-TLS	AS2635	Automattic	39,0%	–	proxy
Tranco-TLS	AS7979	Servers.com	22,4%	–	din.
Tranco-TLS	AS8075	Microsoft	18,3%	–	proxy

Tabela V
Shodan-HTTPS, $b = 8$: 10 maiores ASes por IPs encontrados.

AS	Nome do AS	IPs	%	din.	proxy	comp.
AS16509	Amazon.com, US	184 316	35,5%	176 931	6 026	1 359
AS28220	Alares Cabo, BR	117 552	22,7%	102 231	15 188	133
AS16625	Akamai, US	29 289	5,6%	29 289	0	0
AS27715	Locaweb, BR	14 941	2,9%	5 922	7 924	1 095
AS28270	(sem nome)	11 438	2,2%	11 398	0	40
AS8075	Microsoft, US	10 533	2,0%	2 406	7 089	1 038
AS396982	Google, US	8 828	1,7%	8 780	15	33
AS20940	Akamai Intl., NL	7 972	1,5%	757	7 030	185
AS28667	VERO S.A, BR	5 383	1,0%	5 337	0	46
AS4230	CLARO S.A., BR	3 704	0,7%	634	2 116	954

quase inteiramente servidores compartilhados, dominada por um único AS (TIM, 76–83%). Para o SSH, agrupamos as classes proxy e servidores compartilhados em uma única categoria (Seção VII), que então corresponde a 95,9% da saída em $b = 8$ e 97,8% em $b = 128$. Chaves SSH são estáveis por dispositivo e frequentemente compartilhadas ou clonadas (CPEs com chave padrão, CGNAT), por isso não revelam realocação dinâmica, o que é esperado, mas sim anomalias de compartilhamento de chave. Assim, observamos que executar o DynMap para SSH é útil para detectar infraestrutura compartilhada.

C. Tranco-TLS

O conjunto Tranco-TLS foi explorado como dado complementar. Sua janela de observação (cerca de um mês, contra 18 meses dos conjuntos do Shodan) é curta demais para o DynMap, que depende de séries temporais longas para acumular evidências de dinamicidade. Assim, a cobertura é muito baixa (0,7% em $b = 8$) e nula a partir de $b \geq 128$, e os poucos IPs encontrados não permitem conclusões significativas. Como o Tranco-TLS utiliza o mesmo *fingerprint* (certificado X.509) que o Shodan-HTTPS, esperamos que, com dados longitudinais suficientes, suas inferências se aproximem das observadas no Shodan-HTTPS. Uma avaliação conclusiva do Tranco fica como trabalho futuro.

VII. Estudos de Caso

As classificações de proxy e servidores compartilhados, que refinam a inferência de blocos dinâmicos (Seção III-D), surgiram da inspeção manual de blocos com métricas peculiares

Tabela VI
SHODAN-HTTPS, $b = 128$: 10 MAIORES ASES POR IPS ENCONTRADOS.

AS	Nome do AS	IPs	%	din.	proxy	comp.
AS28220	Alares Cabo, BR	114 132	41,1%	99 254	14 878	0
AS16509	Amazon.com, US	59 601	21,4%	54 276	4 408	917
AS16625	Akamai, US	29 070	10,5%	29 070	0	0
AS28270	(sem nome)	11 273	4,1%	11 273	0	0
AS27715	Locaweb, BR	11 175	4,0%	4 561	6 089	525
AS8075	Microsoft, US	6 727	2,4%	1 266	5 461	0
AS28667	VERO S.A, BR	5 273	1,9%	5 273	0	0
AS28299	LWSA S/A, BR	2 961	1,1%	1 750	1 211	0
AS263600	Radio Link, BR	1 540	0,6%	1 540	0	0
AS8069	Microsoft, US	1 265	0,5%	0	1 265	0

Tabela VII
SHODAN-SSH, $b = 8$: 10 MAIORES ASES POR IPS ENCONTRADOS.

AS	Nome do AS	IPs	%	din.	proxy	comp.
AS26615	TIM S/A, BR	26 117	76,1%	15	8 230	17 872
AS263317	VoxTech, BR	1 249	3,6%	0	928	321
AS202422	G-Core Labs, LU	761	2,2%	761	0	0
AS268050	SUPRINET, BR	596	1,7%	0	0	596
AS270545	Digit@l Inf., BR	508	1,5%	0	0	508
AS262744	ICENET, BR	395	1,2%	1	0	394
AS268468	MXCONECT, BR	361	1,1%	0	361	0
AS53185	zago & zago, BR	353	1,0%	0	0	353
AS53080	VERO S.A, BR	318	0,9%	0	238	80
AS263110	Louvetel, BR	262	0,8%	0	0	262

Tabela VIII
SHODAN-SSH, $b = 128$: 10 MAIORES ASES POR IPS ENCONTRADOS.

AS	Nome do AS	IPs	%	din.	proxy	comp.
AS26615	TIM S/A, BR	24 249	83,0%	0	8 180	16 069
AS263317	VoxTech, BR	1 249	4,3%	0	928	321
AS270545	Digit@l Inf., BR	508	1,7%	0	0	508
AS268050	SUPRINET, BR	474	1,6%	0	0	474
AS202422	G-Core Labs, LU	405	1,4%	405	0	0
AS268468	MXCONECT, BR	361	1,2%	0	361	0
AS53185	zago & zago, BR	353	1,2%	0	0	353
AS263110	Louvetel, BR	253	0,9%	0	0	253
AS52717	Teleuno, BR	251	0,9%	251	0	0
AS53080	VERO S.A, BR	238	0,8%	0	238	0

Tabela IX
TRANCO-TLS, $b = 8$: 10 MAIORES ASES POR IPS ENCONTRADOS.
($b = 128$ FOI OMITIDO POIS NÃO GEROU SAÍDA)

AS	Nome do AS	IPs	%	din.	proxy	comp.
AS2635	Automattic, US	134	39,0%	0	134	0
AS7979	Servers.com, US	77	22,4%	69	8	0
AS8075	Microsoft, US	63	18,3%	0	63	0
AS133618	Trellian, AU	13	3,8%	13	0	0
AS24940	Hetzner, DE	10	2,9%	0	10	0
AS52130	ARTCOMPANYCZ, CZ	9	2,6%	0	0	9
AS396982	Google, US	8	2,3%	0	8	0
AS198610	Beget, RU	8	2,3%	0	8	0
AS49981	WorldStream, NL	7	2,0%	0	0	7
AS16509	Amazon.com, US	5	1,5%	5	0	0

de entropia de uso e dinamicidade de domínio. Nesta seção apresentamos os estudos de caso que motivaram cada categoria e ilustram como esse comportamento se manifesta em blocos reais.

A. Proxy: AMX Internet e Microsoft

A AMX Internet (AS271229) é um ISP onde o DynMap identificou 1024 endereços IP dinâmicos. O DNS reverso destes IPs segue um padrão que inclui o próprio endereço IP e o nome da empresa, como em dynamic.200-49-28-32.amxinternet.com.br. Inspecionando as métricas manualmente, observamos que a dinamicidade de domínio para os IPs do bloco é muito baixa: existem apenas cinco domínios (dois deles sendo 0.0.0.0 e localhost.localdomain), compartilhando sete assinaturas concentradas nas portas 8081 e 8443. A entropia de uso, por sua vez, é maior que 0,95, pois mais de 970 dos 1024 endereços IP observam as mesmas sete assinaturas. A inspeção do conteúdo HTTP indica que as páginas são servidas por um comutador de borda Ubiquiti, provavelmente configurado para responder a requisições nestas portas em todos os IPs do bloco.

De forma similar, uma grande quantidade de endereços IP dinâmicos foi encontrada na Microsoft (AS8075). Os IPs da Microsoft possuem comportamento análogo aos da AMX Internet: entropia de uso maior que 0,95 e dinamicidade de domínio baixa. Os IPs da Microsoft não possuem DNS reverso configurado, mas todos utilizam o mesmo certificado, com *common name* *.mp.skype.com, que foi renovado uma única vez durante o período analisado (duas assinaturas em intervalos disjuntos).

O grande número de endereços IP compartilhando um número muito pequeno de domínios e assinaturas indica que estes IPs podem ser utilizados por um único serviço centralizado em um proxy reverso, um *front-end* Web ou um balanceador de carga, o que motivou a criação desta classe de refinamento. Esse comportamento é coerente com os resultados agregados (Seção VI), em que os blocos da Microsoft permanecem majoritariamente proxy mesmo para tamanhos de bloco grandes.

B. Servidores Compartilhados: Facebook

No Facebook (AS32934), o DynMap encontrou 128 endereços IP divididos em blocos de tamanho pequeno (8 a 32 IPs), onde a entropia de uso é maior que 0,95 e a dinamicidade de domínio é baixa (cada bloco possui cerca de dez domínios). Porém, cada bloco apresenta 265 assinaturas distintas. Como os blocos são pequenos, isso significa que um IP arbitrário do bloco apresenta várias assinaturas diferentes ao longo do tempo para um número pequeno de domínios. Esse comportamento indica que os blocos do Facebook são servidores compartilhados, hospedando diversos tipos de conteúdo e serviço, o que se alinha com os *common names* dos certificados (*e.g.*, *.facebook.com, *.facebookvirtualassistant.com e *.fbinfra.net). Esse comportamento motivou a definição da subclassificação de servidores compartilhados.

C. SSH: Compartilhamento de Chaves de Host

No Shodan-SSH, o *fingerprint* é a chave de *host* SSH e o domínio é o nome obtido por DNS reverso. Diferente dos certificados HTTPS, o DNS reverso atribui um único sufixo (registro PTR) a todo um bloco de endereços de um provedor,

de modo que a dinamicidade de domínio é praticamente nula: cerca de 92% dos blocos candidatos possuem um único domínio, e a métrica de dinamicidade de domínio é zero para essencialmente todos os blocos de proxy e de servidores compartilhados. As poucas exceções, com muitos domínios distintos, são provedores de nuvem (vultrusercontent.com, linodeusercontent.com, googleusercontent.com), justamente os blocos classificados como dinâmicos.

Como a dimensão de domínio é degenerada, a distinção entre proxy e servidores compartilhados, que no caso HTTPS depende justamente dos domínios (Seção III-D), perde o significado. Ambos os tipos recaem no mesmo caso de classificação (entropia de uso alta e dinamicidade de domínio nula), e a separação entre eles passa a depender apenas de o bloco ter menos chaves do que hosts (rotulado proxy) ou mais chaves do que hosts (rotulado servidores compartilhados), o que tende ao acaso, sem significado semântico. Por isso, no SSH agrupamos as duas classes sob a única categoria de servidores compartilhados, que então corresponde a 95,9% da saída em $b = 8$ e 97,8% em $b = 128$, restando apenas 4,1% e 2,2% de IPs dinâmicos.

Observamos então o compartilhamento massivo de chaves de host SSH. Das 478.952 chaves observadas, 15% aparecem em mais de um endereço IP, e 15 chaves aparecem em 500 ou mais IPs. A chave mais compartilhada aparece em 107.219 endereços IP espalhados por marcas da Telefônica/Vivo (vivozap.com.br, telesp.net.br, gvt.net.br); a segunda, em 58.808 IPs da V.tal/Oi e da TIM (v-tal.net.br, timbrasil.com.br); e a terceira, em 21.582 IPs da Algar Telecom e de ISPs regionais. Essas chaves cobrem praticamente todo o espaço de endereços e cruzam múltiplos provedores, o que indica chaves de host padrão embutidas no *firmware* de equipamentos de cliente (CPEs, modems e ONTs), o mesmo modelo de equipamento implantado em milhões de assinantes, e até em provedores distintos que reutilizam o mesmo SDK de fabricante, além do efeito de CGNAT. O reuso de chaves criptográficas embutidas em dispositivos é um problema documentado em larga escala [21], [22].

Usando o maior bloco rotulado proxy como exemplo, na TIM (timbrasil.com.br, AS26615), o DynMap identificou um bloco de 7.812 endereços IP compartilhando 3.263 chaves sob um único domínio. A entropia de uso é alta (as chaves se repetem entre os hosts), mas a dinamicidade de domínio é nula, o mesmo comportamento que, em outros blocos com ainda menos chaves distintas, recebe o rótulo de servidores compartilhados. Em todos os casos, trata-se da mesma anomalia: um conjunto contíguo de máquinas de um provedor compartilhando chaves de host SSH. Concluímos que, para o SSH, o DynMap não revela realocação dinâmica de endereços, mas é eficaz para evidenciar infraestrutura de chaves compartilhadas.

VIII. Conclusão

Neste trabalho aplicamos e estendemos o DynMap, um sistema de identificação de endereços IP dinâmicos a partir de dados públicos, sobre três conjuntos de dados que vão além do cenário original de varreduras HTTPS do Shodan. Demonstramos que o DynMap generaliza para além do Shodan HTTPS, mas que a escolha do *fingerprint* molda o resultado: no Shodan-HTTPS, os certificados X.509 revelam dinamismo de nuvem, hospedagem e ISPs, enquanto as chaves de *host* SSH revelam compartilhamento de infraestrutura. O parâmetro b oferece um compromisso ajustável entre precisão e cobertura, concentrando a inferência em ASes grandes conforme cresce. O conjunto Tranco-TLS, explorado como dado complementar, não dispôs de janela temporal suficiente para resultados conclusivos, ficando como exploração futura. As inferências do DynMap são uma fonte adicional de informação que pode ser integrada em ferramentas e processos utilizados por analistas de segurança e operadores de rede.

IX. Trabalhos Futuros

Como trabalhos futuros, pretendemos: (i) revisitar uma fonte de IPs maliciosos (PhishTank ou *phishing* em geral) com uma infraestrutura de varredura segura e isolada, (ii)ç explorar outros *fingerprints* (*e.g.*, JARM, *banners* de serviço) e a combinação de múltiplos tipos por IP; (iii) realizar validação externa das inferências, relacionando os blocos identificados ao tipo do AS (PeeringDB [13]), ao porte do AS (relações entre ASes da CAIDA [14]) e ao DNS reverso (Rapid7 [20]); e (iv) avaliar o Tranco-TLS sobre uma janela longitudinal mais longa, comparável aos 18 meses dos conjuntos do Shodan, para verificar se suas inferências se aproximam das do Shodan-HTTPS.

Referências

[1] Cardoso, Gabriel and Oliveira, Leonardo and Cunha, Ítalo, *Identificação de Endereços IP Dinâmicos com Dados Públicos*, Anais do XXIV Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, SBC, São José dos Campos/SP, 2024, pp. 822–828, doi: 10.5753/sbseg.2024.241681, URL: https://sol.sbc.org.br/index.php/sbseg/article/view/30074.

[2] Xie, Yinglian and Yu, Fang and Achan, Kannan and Gillum, Eliot and Goldszmidt, Moises and Wobber, Ted, *How Dynamic are IP Addresses?*, Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Association for Computing Machinery, 2007, pp. 301–312, doi: 10.1145/1282380.1282415.

[3] Jin, Yu and Sharafuddin, Esam and Zhang, Zhi Li, *Identifying dynamic IP address blocks serendipitously through background scanning traffic*, Proceedings of the 2007 ACM CoNEXT Conference, Association for Computing Machinery, 2007, pp. 1–12, doi: 10.1145/1364654.1364659.

[4] Dan, Ovidiu and Parikh, Vaibhav and Davison, Brian D., *IP Geolocation through Reverse DNS*, ACM Trans. Internet Technol., Association for Computing Machinery, 2021, vol. 22, pp. 1–29, doi: 10.1145/3457611.

[5] Marder, Alexander and Luckie, Matthew and Dhamdhere, Amogh and Huffaker, Bradley and claffy, kc and Smith, Jonathan M., *Pushing the Boundaries with bdrmapIT: Mapping Router Ownership at Internet Scale*, Proceedings of the Internet Measurement Conference 2018, Association for Computing Machinery, 2018, pp. 56–69, doi: 10.1145/3278532.3278538.

[6] Richter, Philipp and Smaragdakis, Georgios and Plonka, David and Berger, Arthur, *Beyond Counting: New Perspectives on the Active IPv4 Address Space*, Proceedings of the Internet Measurement Conference 2016, Association for Computing Machinery, 2016, pp. 135–149, doi: 10.1145/2987443.2987473.

[7] Xiao, Zhen and Song, Weijia and Chen, Qi, *Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment*, IEEE Transactions on Parallel and Distributed Systems, 2013, vol. 24, no. 6, pp. 1107–1117, doi: 10.1109/TPDS.2012.283.

[8] Fiebig, Tobias and Borgolte, Kevin and Hao, Shuang and Kruegel, Christopher and Vigna, Giovanni and Feldmann, Anja, *In rDNS We Trust: Revisiting a Common Data-Source's Reliability*, Passive and Active Measurement, 2018, Springer International Publishing, pp. 131–145, doi: 10.1007/978-3-319-76481-8_10.

[9] Nakamori, Tomofumi and Chiba, Daiki and Akiyama, Mitsuaki and Goto, Shigeki, *Detecting Dynamic IP Addresses and Cloud Blocks Using the Sequential Characteristics of PTR Records*, Journal of Information Processing, 2019, vol. 27, pp. 525–535, doi: 10.2197/ipsjjip.27.525.

[10] Husák, Martin and Čermák, Milan and Jirsík, Tomáš and Čeleda, Pavel, *HTTPS traffic analysis and client identification using passive SSL/TLS fingerprinting*, EURASIP Journal on Information Security, 2016, doi: 10.1186/s13635-016-0030-7.

[11] Sankar, Vedha and Bharanikumar, Varsha and Swaminathan, Lakshmi, *Dynamic Load Balancing and Resource Optimization Algorithm for Reverse Proxy Servers*, 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS), 2024, doi: 10.1109/ICC-ROBINS60238.2024.10533888.

[12] PyASN, *IP address to Autonomous System Number lookup module*, 2024, URL: https://github.com/hadiasghari/pyasn.

[13] PeeringDB, *The Interconnection Database*, 2024, URL: https://www.peeringdb.com.

[14] The CAIDA UCSD, *AS Relationships Dataset*, 2024, URL: https://catalog.caida.org/dataset/as_relationships_serial_1.

[15] Shodan, *Shodan Search Engine*, 2025, URL: https://www.shodan.io/.

[16] Censys, *Attack Surface Management and Threat Hunting Solutions*, 2025, URL: https://censys.com.

[17] Tranco, *A Research-Oriented Top Sites Ranking Hardened Against Manipulation*, 2025, URL: https://tranco-list.eu/.

[18] PhishTank, *Out of the Net, into the Tank*, 2025, URL: https://www.phishtank.com/.

[19] ZGrab, *Application-Layer Network Scanner*, 2025, URL: https://github.com/zmap/zgrab2.

[20] Rapid7 Labs, *Reverse DNS (RDNS)*, 2025, URL: https://opendata.rapid7.com/sonar.rdns_v2.

[21] Heninger, Nadia and Durumeric, Zakir and Wustrow, Eric and Halderman, J. Alex, *Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices*, Proceedings of the 21st USENIX Security Symposium, 2012, pp. 205–220, URL: https://factorable.net/weakkeys12.conference.pdf.

[22] SEC Consult, *House of Keys: Industry-Wide HTTPS Certificate and SSH Key Reuse Endangers Millions of Devices Worldwide*, 2015, URL: https://sec-consult.com/blog/detail/house-of-keys-industry-wide-https/.