计 算 机 研 究 与 发 展
Journal of Computer Research and Development

DOI: 10.7544/issn1000-1239.202330335
61(9): 2307−2320, 2024

基于目标生成的 IPv6 网络地址扫描综述

侯冰楠　刘　宁　李雄略　周桐庆　陈颖文　蔡志平　卢　凯
（国防科技大学计算机学院　长沙　410073）
（houbingnan19@nudt.edu.cn）

Survey on Target-Generated IPv6 Network Address Scanning

Hou Bingnan, Liu Ning, Li Xionglüe, Zhou Tongqing, Chen Yingwen, Cai Zhiping, and Lu Kai
（*College of Computer Science and Technology*, *National University of Defense Technology*, *Changsha* 410073）

Abstract　With the rapid evolution of IPv6 in recent years, the significance of IPv6 network measurement and security analysis has grown substantially. Obtaining a substantial number of active IPv6 addresses has become a fundamental and critical task in this domain. However, the sheer size of the IPv6 address space and the sparsely distributed nature of active hosts present challenges that render brute-force scanning tools, such as ZMap and MASSCAN. While ZMap can scan the entire IPv4 network in just 5 minutes with a 10-gigabit bandwidth, it would take hundreds of millions of years to scan the entire IPv6 network using similar methods. In response to this challenge and in a bid to enhance the efficiency of IPv6-wide scans, researchers have introduced a series of innovative search strategies tailored to IPv6 scans. These strategies aim to enhance the ability to discover assets and mitigate risks within the IPv6 network. We undertake the task of categorizing, organizing, and summarizing the target generation-based scanning approaches proposed by researchers in this field. We conduct a comprehensive analysis, comparing the hit rate, marginal benefit, and time costs of state-of-the-art solutions through real-network scan experiments. Furthermore, we provide valuable insights into the current landscape and emerging trends in IPv6 target generation scanning techniques. By doing so, we contribute to a deeper understanding of IPv6 network analysis and security, ultimately fostering advancements in this critical area of networking research.

Key words　IPv6 network; network measurement; network scan; target generation algorithm; alias prefix detection

摘　要　随着近年 IPv6 网络的迅猛发展，针对 IPv6 的网络测量和安全分析逐步成为热门的研究课题，这其中一项最基础、最关键的工作是通过网络扫描获取大量的 IPv6 活跃地址. 然而 IPv6 庞大的地址空间和稀疏的活跃主机分布使得传统的暴力扫描工具（如 ZMap 和 MASSCAN 等）无法直接使用. 例如在万兆带宽条件下，可 5 min 内扫描 IPv4 全网的 ZMap 扫描器若对 IPv6 进行全网扫描，仍需要花费上亿年的时间. 针对大规模 IPv6 网络扫描所面临的效率低下问题，研究人员提出了一系列针对 IPv6 网络的扫描方法，提升了当前 IPv6 网络资产发现、识别和风控的能力. 对这些基于目标生成的 IPv6 网络扫描方法进行了分类、梳理和总结，分析了各扫描方法的优缺点及适用场景. 通过实网扫描实验，对比了多种扫描策略的命中率、边际效益和时间花销等性能情况. 最后给出了对 IPv6 网络扫描研究的思考并对未来的研究方向进行了展望.

关键词　IPv6 网络；网络测量；网络扫描；目标生成算法；别名前缀检测

收稿日期：2023-04-24；**修回日期**：2023-11-14

基金项目：中国博士后科学基金项目（2023TQ0089）；湖南省科技创新计划项目（2022RC3061, 2021RC2071）
This work was supported by the China Postdoctoral Science Foundation (2023TQ0089) and the Science and Technology Innovation Program of Hunan Province (2022RC3061, 2021RC2071).

通信作者：蔡志平（zpcai@nudt.edu.cn）

中图法分类号 TP391

互联网络正向IPv6时代演进. 下一代互联网协议IPv6已在全球范围进入加速推广和部署阶段[1]. 从1996年开始, 一系列关于IPv6的定义和规范RFC文档的发表, 逐步完善了IPv6的协议体系. 2012年, 多家知名网站开始永久性支持IPv6网络访问, 如Facebook, Google等公司. 2020年3月亚太互联网络信息中心(Asia-Pacific Network Information Centre, APNIC)统计数据显示网络用户超过1亿的国家中, 印度和美国的IPv6部署率最高, 分别为61.40%和57.28%. 2022年12月国家IPv6发展监测平台显示我国IPv6活跃用户总数已达7.18亿, 全国互联网活跃用户占比达68.28%. 随着IPv6网络的迅猛发展, 针对IPv6网络的资产探测、拓扑发现和安全分析成为了迫切需要解决的研究问题[2].

在传统的IPv4网络中, 随着无状态暴力扫描工具ZMap[3]和MASSCAN[4]的问世, 使得网络扫描可以涵盖整个IPv4网络, 极大地提升了IPv4全网范围的网络测量和安全分析能力. 在此基础上, 针对IPv4全网的资产发现和安全分析等研究快速发展, 例如互联网上层协议安全搜索引擎Shodan[5]和Censys[6]可为IPv4全网范围内的网络设备提供更细粒度的安全分析信息, 网络管理员可以据此评估安全风险, 发现漏洞并构建安全防护手段. 然而, 这些工具在广阔的IPv6地址空间面前都失效了. 这是因为相比于IPv4网络2^{32}的地址空间, IPv6网络含有2^{128}的地址空间. 巨大的IPv6地址空间和其极其稀疏的活跃主机分布使得逐个IPv6地址的网络调查变得不可能. 在万兆链路带宽条件下, 即便最高配置的ZMap扫描器仍需要花费上亿年的时间才能完成对IPv6全网的扫描.

可见, 当前IPv6网络存在全网范围调查能力弱的短板. 为此, 急需开展针对IPv6网络的高效扫描研究, 以实现快速的、大覆盖面的活跃地址发现, 为进一步的全网范围的网络资产发现、拓扑结构分析和安全态势感知等工作打下坚实基础.

1 种子地址获取方法

扫描整个IPv6网络是不可行的, 因此, 当前的IPv6网络扫描采用了早期对IPv4网络扫描的方法, 即在不完全扫描整个网络的情况下使用所收集的IP地址(也被称为地址列表)作为整个地址空间中一个有代表性的子集[7]. 在IPv6地址获取上, 研究者已提出多种有效的手段, 总体上被分为被动采集和主动获取2类方式.

1.1 被动地址获取手段

被动采集的主要手段有: 1)在网络流量中采集, Gasser等人[8]在一个欧洲互联网交换节点和慕尼黑科学网络MWN路由节点进行流量抽样并提取IPv6地址, Song等人[9]在第2代中国教育和科研计算机网CERNET2网络出口镜像流量数据, 并从流量数据中提取IPv6地址. 2)通过众包平台在进行网络测量的同时收集IPv6地址[10-12]. 被动的IPv6地址采集手段往往存在采集数量少、地址覆盖面窄等缺点.

1.2 主动地址获取手段

主动获取地址的手段主要有:

1)利用域名系统(domain name system, DNS)反向解析和从DNS Zone文件中获取地址[13]. Strowes[14]提出了一种通过IPv4地址逆向DNS(rDNS)查询后进行AAAA查询得到IPv6地址的方法, 该研究收集了来自5 531个自治域中的96.5万个IPv6地址, 对这些地址进行主动探测, 响应比例是56.7%. 随后Fiebig等人[15-16]通过进一步利用DNS反向查询(PTR记录)进行IPv6地址收集, 共收集了280万个IPv6地址. Borgolte等人[17]提出了结合域名系统安全扩展(domain name system security extensions, DNSSEC)的区域文件收集IPv6地址的方法. 此外, Gasser等人[8,18]还利用Rapid7[19]所公开的DNS ANY查询记录文件以及TLS证书透明度(CT)日志进行IPv6地址提取, 在这2类文件中其分别提取了250万个和1 620万个地址.

2)利用比特节点API爬取地址. 比特节点API[20]提供比特币网络当前可供连接的节点, Gasser等人[18]利用该方法爬取了2.7万个客户端IPv6地址.

3)从Traceroute数据中提取地址. Gasser等人[18]从欧洲网络测量组织RIPE所提供的开源数据Traceroute中提取出了2 000万个路由接口地址.

表1列出了主要的开源IPv6地址的获取来源及其所获取到地址的属性. 以上3个主动方式所获得的IPv6地址数量多、涵盖面广, 但是其中部分地址由于设置的原因是不响应主动探测报文的.

除了上述主动获取地址的手段外, 研究者还通过主动测量的方式进行种子地址收集. Gasser等人[18]和Rye等人[27]分别利用Traceroute的改进工具Scamper[28]和Yarrpv6[29]进行地址探测, 分别获取了2 590万个

Table 1 Main Sources of Obtaining IPv6 Addresses Using Active Measurement

表 1 使用主动方式获取 IPv6 地址的主要来源

开源的 IPv6 地址获取来源	地址属性
Rapid7 FDNS ANY[19]	服务器
Bitnodes API[20]	客户端
Alexa top websites[21]	服务器
Statvoo websites[22]	服务器
Cisco Umbrella websites[23]	服务器
Zone files[24]	服务器
CAIDA IPv6 DNS names[25]	服务器
RIPE Atlas[26]	路由器

IPv6 路由接口地址和 640 万个客户端设备路由器(customer premises equipment, CPE)的接口 IP. 然而利用 Traceroute 及其改进工具进行 IPv6 活跃地址发现的效率是极其低下的, 因为 Traceroute 类型的扫描将在路径节点上浪费大量的探测包.

Hou 等人[30] 提出一种有效的基于已声明的 BGP 前缀领域知识进行主动地址收集的启发式方法, 极大地扩展了地址收集的覆盖面. 该方法主动地在全球已声明的 BGP 前缀下广泛收集种子地址, 通过扩展种子覆盖面的方式提升后续基于目标生成扫描的覆盖面. 文献 [31] 指出, 接口标识符(interface identifier, IID)模式为“低字节”的地址(即 IID 各位的值大部分为 0 的地址)往往在收集到的活跃地址中占比最大, 也就是说低字节模式地址比随机地址具有更高的活跃可能性. 为此, 启发式种子收集选择将各前缀下的 IID 值为::1 的地址作为目标地址. 图 1 显示了其异步设计的启发式种子收集的具体步骤:

步骤 1. 使用 RIPEstat data API[32] 下载国家/地区或自治域(AS)相关联的已声明的 BGP 前缀列表.

步骤 2. 向每个 BGP 前缀的 16 个子前缀下的 IID 值为::1 的地址发送探测报文.

步骤 3. 启发式种子收集在接收线程上使用一个前缀计数器(prefix-counter)来计算前缀下响应地址的数量, 前缀计数器由内存动态分配的桶(bucket)组成, 每个桶中有 2 个变量: 一个用于存储前缀 K_i(K_i 初始值为 none); 另一个变量用于计数 S_i(S_i 初始值为 0), 当活跃地址的应答到达接收线程时, 提取地址前缀并将其散列到前缀计数器桶中. 前缀字符串将存储在桶的 K_i 变量中, 并将 S_i 中的值加 1, 这里 i 表示前缀在前缀计数器中的索引号.

步骤 4. 当前缀下所有活跃地址的应答被散列到前缀计数器后, 遍历每个桶来检查 S_i 的值.

步骤 5. 如果 S_i==16, 则 K_i 被认为是候选的别名前缀, 并将在候选别名前缀下随机生成一个地址进行探测; 如果该地址为活跃地址(即此时 S_i==17), 则认为该候选别名前缀为别名前缀, 随即从前缀列表中删除该别名前缀, 以避免进一步收集别名地址; 如果 S_i<16, 获取应答地址的子前缀并将其添加到前缀列表中, 以便后续迭代能在这些存有活跃地址的非别名前缀下收集种子地址.

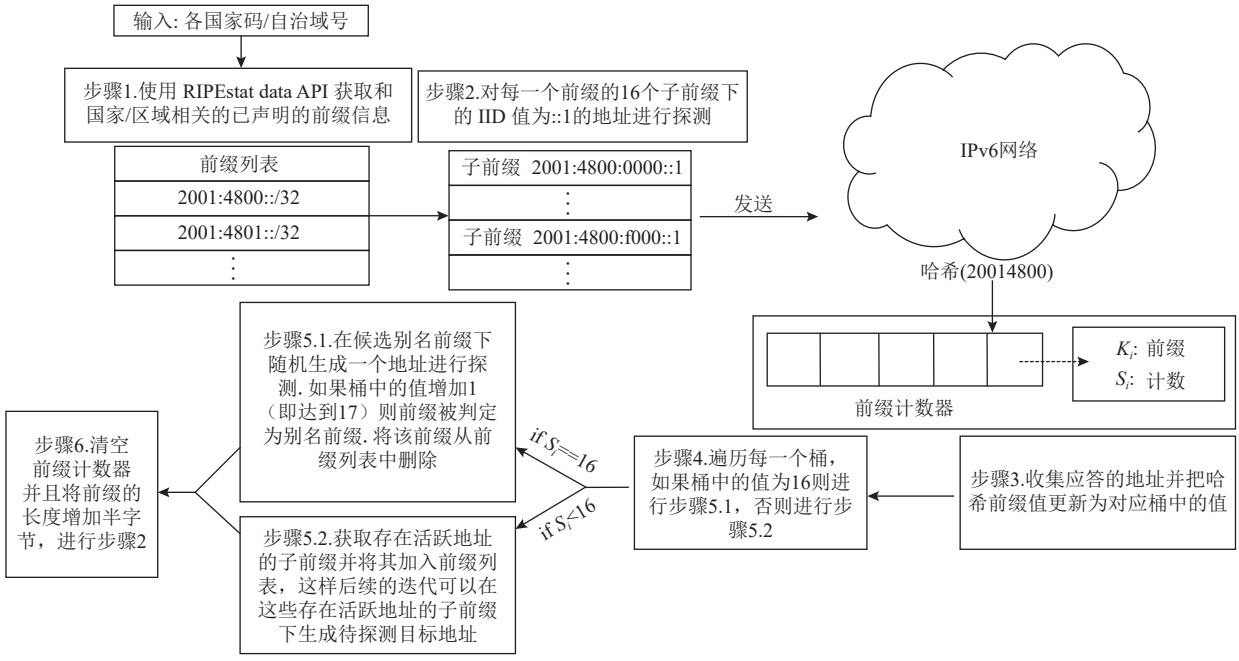


Fig. 1 Heuristic seed collection process

图 1 启发式种子收集流程

步骤 6.清除前缀计数器桶中的值，将前缀的长度增加半字节，转到步骤 2 进行下一轮迭代.

值得注意的是，步骤 2 中是对前缀下的每个子前缀进行探测. 例如给定一个 BGP 前缀 2001:4800::/32，启发式种子收集将在 16 个子前缀下生成的目标地址为 2001:4800:[0~f]000::1.考虑到别名前缀在 IPv6 网络中广泛存在，如在 Linux 系统中通过 IP_FREEBIND 选项可以将大量地址绑定在一台服务器. 这就意味着我们所探测的活性地址中可能很大部分是别名地址，例如一个/32 的别名前缀将包含 2^{96} 个别名地址. 如果不将这些别名前缀标注将会造成探测资源的巨大浪费，且探测结果中过多的别名地址会造成有偏的探测统计.

因此文献 [30] 在步骤 5 中采用了基于 APD[18] 的去别名地址方法. APD 的一个重要假设是在活跃主机分布稀疏的 IPv6 网络下随机生成的目标地址响应的概率是很小的. APD 判定前缀为别名的过程是首先在前缀的每个子前缀下生成一个随机的目标地址，然后向目标地址发送探测包，如果这些地址全部响应，则判定该前缀为别名前缀. 但步骤 5 同 APD 存在的一个区别是 APD 在每个子前缀下生成一个随机地址，而启发式种子收集在每个子前缀下生成一个 IID 值为::1 的固定地址. 已知 IID 值为::1 的地址比随机生成的地址[32] 更有可能是活跃的，为此，文献 [30] 又添加了一个限定条件来判定候选前缀是否为别名前缀，即在候选别名前缀下再随机生成一个地址，只有当该地址为活跃地址时，候选前缀才最终被认定为别名前缀. 此外，启发式种子收集设置当前缀长度超过 100 时迭代结束，设置该结束条件是希望在使用尽可能少的探测预算下完成大覆盖面的种子地址收集. 这种启发式的种子收集算法以及在扫描中进行高效的别名解析方法保证了：1)探测包均匀分布在 BGP 前缀上，使得扫描在相关区域内的地址采样更加均衡；2)受 APD 算法启发的去别名算法有效地避免了因过度扫描别名前缀而造成的探测资源浪费；3)其异步的设计(即在不同线程上发送和接收数据包)以及前缀计数器的使用大大提高了种子收集和别名解析的效率.

2 目标生成扫描研究

IPv6 目标生成扫描是指利用已获取的 IPv6 地址(也称为种子地址)生成更多的目标地址进行扫描的算法. 相关文献 [33] 已证实种子地址能够有效提供寻址方案相关的有价值信息，这些信息将有助于发现更多的活跃地址. 2018 年 Gasser 等人[18] 每周发布其通过主被动手段收集的 IPv6 地址列表(IPv6 hitlist)[34]，极大地促进了目标生成算法的发展，研究者可直接下载其提供的地址列表进行基于种子地址的目标生成研究. 当前 IPv6 目标生成算法可分为两大类，分别是基于种子统计信息目标生成算法和基于种子结构信息的目标生成算法.

2.1 基于种子统计信息的目标生成算法

基于种子统计信息的目标生成算法主要是通过人工智能的方法挖掘种子内在的地址模式规律，从而指导目标地址的生成. Ullrich 等人[35] 根据种子地址统计 IPv6 接口标识符每个比特位为 0 或 1 的概率，并根据每个比特位 0 或 1 概率的大小生成 IID，而后在指定前缀下扫描这些 IID. 随后，Foremski 等人[36] 在 2016 年提出 Entropy/IP 算法，该算法使用香农熵研究种子地址结构，并使用贝叶斯网络模型生成待探测的地址集合. 首先 Entropy/IP 计算 IPv6 种子地址的每个半字节的熵值，这里一个 IPv6 地址共有 32 个半字节(即 128 b)；然后 Entropy/IP 将熵值相近的半字节合并成段，通常用于合并的相近性阈值不能设置过大或者过小，不然会影响分段的效果，最终导致后面的贝叶斯建模效果差；最后根据段与段之间的取值条件概率构建整个地址的贝叶斯网络模型，该模型描述的是每个段的取值以及每个段取值与之前段的条件概率，Entropy/IP 通过随机的方式遍历整个贝斯网络的取值空间，选取概率较大的地址段进行目标生成. 该工作将贝叶斯网络和信息论的知识应用在 IPv6 探测领域. Entropy/IP 存在的缺陷是当种子地址规模较大时，无状态地址分配[37] 的种子比重较高，即 IID 是随机化生成的种子比重高，这样 Entropy/IP 计算 IID 部分的熵值将接近 1，导致 IID 部分无法进行分段，如果 IID 作为一整段处理将严重影响算法效果.

Cui 等人[38] 将深度学习模型引入 IPv6 目标地址生成领域. 他们提出的 6GCVAE 算法通过门控卷积网络构建变分自编码器来学习种子集的地址特征. 首先 6GCVAE 将 IPv6 地址信息转化为 32 维度的高维坐标；然后通过门控卷积网络(gated CNN)学习种子地址的特征，并将其作为变分自编码器的输入以重建 32 维的地址向量；最后转化为目标地址. 随后 Cui 等人[39-40] 又提出 6VecLM 和 6GAN 这 2 种基于深度学习的模型进行目标地址生成. 6VecLM 使用词集成和语言模型进行目标地址生成. 具体地，采用 IPv62Vec 进行地址到句子的转换，然后 6VecLM 将所

有位置的索引和对应的半字节值组合成一个词向量，IPv6 地址的索引和半字节值被转化为一个句子，而后将这些句子作为输入以训练神经网络. 6VecLM 还将 Transformer 引入到目标地址生成过程中，这样就将目标地址生成问题转化为文本生成问题，实际上 6VecLM 是一种基于种子语义信息的目标生成方法. 6GAN 使用对抗网络进行目标地址生成. 其首先利用手动或自动分类器将种子进行分类；其次，使用长短期记忆网络作为生成器，并利用卷积神经网络作为判别器，不断强化生成遵循固定寻址模式的 IPv6 地址. 值得注意的是 6GAN 实验了 3 种方法进行种子地址分类，分别为基于 RFC 定义的分类、基于信息熵的分类和基于 IPv62Vec 技术的分类. 实验结果表明，基于 RFC 定义的分类效果最好，生成的目标地址为活跃地址的比例最高.

以上基于种子统计信息的目标生成算法较随机探测在目标命中率上有所提升，但算法模型往往利用较复杂的人工智能算法. 当种子地址数量较小时，此类算法不能有效地挖掘出活跃地址的地址模式，造成其扫描命中（活跃地址）率低下；当种子数量较大时，算法过高的复杂度导致训练时间过长，大大降低了整体的扫描效率.

2.2 基于种子结构信息的目标生成算法

基于种子结构信息的目标生成算法本质上是利用种子的结构信息进行待扫描子空间的划分，然后基于种子地址在子空间中的分布进行基于密度驱动式的扫描. Murdock 等人[41]在 2017 年提出将种子用于地址空间划分并结合区域种子密度信息进行目标地址生成的算法 6Gen. 该算法的一个重要假设是种子密度大小表征了区域真实活性密度大小，尽管事实往往并非如此. 图 2 上半部分显示了 6Gen 使用层次聚合聚类（agglomerative hierarchical clustering, AHC）进行地址空间划分的过程. 初始时，每一个种子被当作一个可被聚合的簇，然后所有的簇进行聚合，此时在聚合过程中并不减少簇的个数，即种子可以归属于多个簇，最终所形成的密度最大的多个簇作为待扫描的地址空间. 6Gen 这种基于种子密度驱动的目标生成算法较基于种子统计信息的目标生成算法在

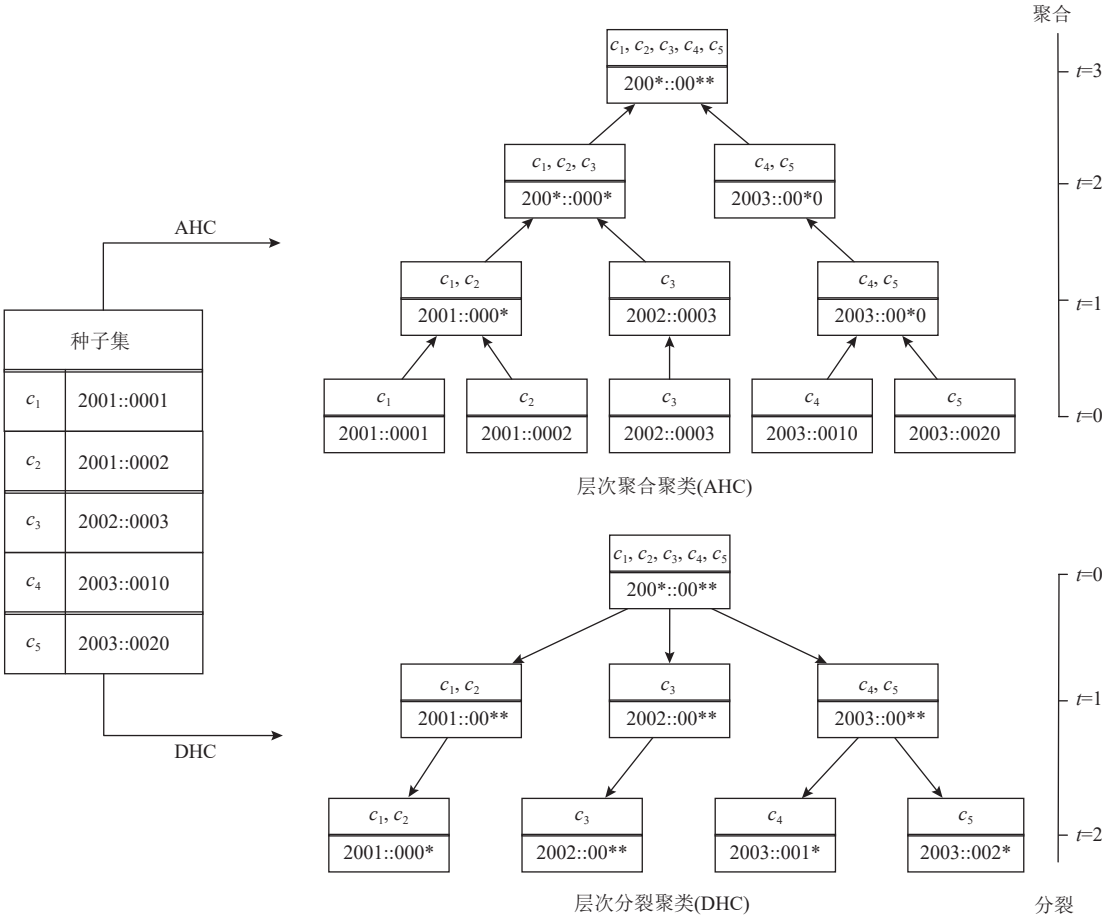


Fig. 2 An example of IPv6 space partitioning using hierarchical clustering algorithms

图 2 使用层次聚类算法进行 IPv6 空间划分的示例

命中率上有大幅度提升,但 6Gen 的 AHC 算法在聚类时由于种子不归属于任意簇,也就是在每一轮聚类时都需要进行复杂的簇与簇之间的距离计算,从而导致了非常高的算法复杂度(即 $O(n^3)$),当处理规模较大的种子集时,算法耗时将呈指数级增长.

2019 年, Liu 等人[42] 提出了 6Tree 算法,同 6Gen 的 AHC 空间划分方法相反,6Tree 使用层次分裂聚类(divisive hierarchical clustering, DHC)算法进行地址空间划分,图 2 下半部分展示了自上而下的层次分裂聚类的空间划分过程. 初始时 6Tree 将能包裹住所有种子地址的子空间作为根节点,然后按地址顺序选取第一个熵值不为 0(存在不同值)的维度对空间进行划分. 以此迭代空间划分过程,直到完成空间树的构建. 而后,6Tree 从空间树的叶子节点开始扫描,并继续上溯扫描命中率最高节点的父节点,直到达到扫描总预算. 同 AHC 空间分裂算法相比,6Tree 提出的 DHC 在算法复杂度上进行了优化(即 $O(n\log n)$),使得其可以处理规模较大的种子集. 同时 DHC 自顶向下的分裂过程将在叶子节点包裹住所有的种子地址,即其种子利用率比 AHC 算法要高,这样,在小规模发包预算时 DHC 算法相比 AHC 算法在命中率上有较大优势.

Song 等人[9] 对 DHC 算法进行了改进并提出了 DET 算法,与 DHC 不同,DET 不按地址顺序进行地址空间的划分,而根据各半字节位上熵值的大小进行地址空间的划分. Yang 等人[43-44] 针对 DHC 算法提出了 2 种改进,即 6Graph 和 6Forest. 6Graph 首先根据 DHC 算法将 IPv6 地址空间划分为不同的区域,然后将每个区域的地址映射为无向图,并进行基于密度的图切割地址聚类形成更精细的地址空间划分.

随后 Zirngibl 等人[45] 对 6Graph, 6Tree, 6GAN, 6VecLM 进行了 2 次验证,证明了在小预算下 6Graph 发现活跃地址数量优于对比方法. 6Forest 使用一种改进的孤立森林算法进行“异常”地址检测,并将“异常”地址剔除在地址划分过程之外,使得所生成的待扫描空间不被“异常”地址割裂,针对一些“异常”地址较多的种子集,6Forest 具有相对的优势. 以上 DET, 6Graph 和 6Forest 三种对 DHC 空间划分方法的改进在当探测预算比较小时命中率相比于 6Tree 有一定的提升,因为其通过算法使空间划分得更加精细,但是在大预算情况下,这 3 种方法并没有太大的优势,因为它们并没有脱离出 DHC 算法划分空间的大方向生成更多有价值的探测空间.

2021 年, Hou 等人[46] 提出了动态扫描算法 6Hit,该算法将动态性与交互性引入到地址扫描过程中. 通过对强化学习算法的引入,让扫描器在与实际网络的交互中不断了解待扫描 IPv6 地址子空间的活跃度,并根据各子空间活跃度大小动态调整后续扫描方向. 这种动态性与交互性的引入使得扫描方向不断向活跃度更高的子空间倾斜,从而提升了扫描的命中率. 随后, Song 等人[47] 又提出 AddrMiner,其根据种子地址情况将 IPv6 地址空间分为 3 种情形,即没有种子、少量种子和种子充足. 在种子充足的子空间, AddrMiner 同 6Hit 一样使用基于强化学习的目标生成. 在没有种子和少量种子的子空间, AddrMiner 利用一种组织关联策略进行种子地址的挖掘,具体地, AddrMiner 首先使用无向图来表示地址模式之间的相似性;然后使用图社区发现方法来发现模式相似性高的社区,每个社区代表一个地址模式,这些社区构成一个公共的模式库;最后使用组织关联策略将这些公共地址模式迁移到任何已声明的前缀下生成目标地址. 动态性扫描在一定程度上提升了扫描的命中率,特别是在种子质量差,即真实活跃地址分布同种子地址分布拟合较差的情况下. 但动态性扫描对扫描方向的动态调整破坏了扫描的异步性,具体地,动态性扫描在每轮扫描完成后,要将每个响应地址与其所在的子空间做映射以计算每个子空间的奖励大小,且这一过程是耗时的,然后根据每个子空间的奖励大小,分配本轮对每个子空间的探测资源数量后进行扫描,这一耗时的动态调整会极大地降低扫描效率. 为此, Hou 等人[48] 又提出一种基于区域编码的高效 IPv6 活跃地址异步扫描方法 6Scan. 6Scan 在动态性扫描的基础上实现了异步性扫描,即发包与收包过程在各自独立的线程中进行,极大地提升了扫描的效率. 通过区域编码技术,扫描器将目标地址所在的子空间编码标识直接写入所发送的数据包负载中,当扫描器接收到活跃地址应答时,直接将应答包负载中的子空间编码标识进行解码,更新位于内存中的子空间活跃计数器,这样在扫描过程中无需进行过多的计算便可直接进行扫描方向的调整,从而维持了扫描的异步性.

本质上,基于 DHC 的动态性扫描算法并没有生成更多有价值的探测方向,而是在 DHC 算法所划分的待扫描空间中进行探测方向的微调. 当种子地址密度同真实活跃地址密度拟合度较好时或者当探测预算足够多,已涵盖所有 DHC 所生成的有价值探测区间时,动态策略扫描的优势就荡然无存了. 为此, Hou 等人[30] 又提出 HMap6, 希望最大化地利用种子

地址生成更多有价值的探测方向. HMap6 同时采用优化的 AHC 和 DHC 两种算法来进行扫描子空间的生成，这样可以生成更多的待扫描子空间. 此外，HMap6 优化了算法的复杂度. 例如，6Gen 使用了原始的更复杂的 AHC 算法，该算法没有显式地合并距离相近的簇，即每个簇都是独立增长的，允许一个种子地址属于多个簇. 这种特性使 6Gen 能够准确地计算各簇中所涵盖的种子数量，但算法的时间复杂度高达 $O(n^3)$，其中 n 表示种子数量. 6Tree 使用的 DHC 算法最终形成的最小簇只包含 1 个变量维度，而 HMap6 将簇分解成具有指定变量维度的子簇，减少了空间划分的次数和计算开销.

2.3 典型目标生成算法技术对比

2.1~2.2 节阐述了各目标生成算法的原理及优缺点，接下来从多个角度对 8 种典型的目标生成算法进行对比，以帮助我们更好地分析各个生成算法的技术特点和适用场景.

表 2 比较了 8 种经典的目标生成式扫描算法，其中论文中最大种子集数量和最大发包量分别表示相关方法在其论文描述的实网实验中 1 次扫描所使用的最大种子数和最大探测预算量. 可以看到，6Gen 和 6GAN 使用的种子体量很小（小于 10 万个种子地址），这是因为它们的算法复杂度过高，无法处理大规模的种子集，特别是在普通计算能力的测试环境中. 6Hit 和 AddrMiner 使用了一个较小的探测预算，即少于 5 000 万个探测报文，因为其动态搜索策略破坏了扫描的异步性，大大降低了其发包速率.

不同于以上方法在目标生成后直接调用开源扫描工具 ZMapv6[3] 进行地址探测，6Scan 通过自主设计的扫描器将区域标识编码到探测包的负载中，在接收线程直接解码区域标识字段，实现区域活跃度的统计和更新，从而据此调整扫描方向，真正意义上实现了动态的异步扫描，提升了动态策略的扫描效率.

然而，6Scan 同 6Tree 和 6Forest 一样，只适用于中等规模的扫描，这是因为当扫描预算持续增加时，基于 DHC 的目标生成策略性能会出现下降，缘于 DHC 算法仅采用自上而下的空间分裂方式生成待扫描子空间，没有充分地利用种子的结构信息. 而 HMap6 采用层次聚类（hierarchical clustering, HC）对 AHC 和 DHC 算法进行了改进和集成，使其能高效且更充分地利用种子生成更多的待扫描子空间. HMap6 的目标生成算法具有较低的算法复杂度（$O(n\log n)$，其中 n 表示种子数目）使之可以处理大规模种子集，并且其静态扫描策略可以支持异步式的扫描，适用于大规模预算扫描任务.

3 实验评估与分析

3.1 种子集合

实验种子集由 HMap6 通过启发式方法收集的种子[49]（H_{protocal}）和 Gasser[34] 于 2022 年 1 月 15 日发布的地址列表（G_{protocal}）构成，即 $S_{\text{protocal}} = H_{\text{protocal}} \cup G_{\text{protocal}}$. 各种子集数量及涵盖范围如表 3 所示，其中 S_{protocal} 表示种子集是由所有响应该协议的活跃地址构成. 由表 3 可知，实验种子集数量较大且分布广泛.

3.2 对比算法

为对比各算法的实网扫描能力，本节从已开源的相关研究中选择有代表性的 6Gen, 6Tree, 6Forest, 6Scan, HMap6 进行性能对比. 虽然 6GAN 也已开源，但是其过高的算法复杂度在我们没有 GPU 的测试服务器上连最小的种子集（即 $S_{\text{TCP_ACK80}}$）都无法运行成功，可见基于深度学习的方法并不适用于大规模的扫描任务，最起码在普通能力的服务器上是行不通的. 实验选择 6Scan 而非 6Hit 作为其中一种对比方法

Table 2 Technologies Comparison of Typical Target Generation Algorithms

表 2 典型目标生成算法技术对比

扫描算法	目标生成算法	生成算法复杂度	论文中最大种子集数量	论文中最大发包量	是否开源（实现语言）	适用扫描场景
6GAN[40]	GAN	难以估计	5.0×10^4	0.05×10^6	是 (Python)	小规模
6Gen[41]	AHC	$O(n^3)$	10.0×10^4	1.00×10^6	否 (C++)	小规模
6Tree[42]	DHC	$O(n\log n)$	40.0×10^4	$1\,500.00\times10^6$	是 (C++)	中等规模
6Forest[44]	DHC+EL	$\Omega(n\log n)$	140.0×10^4	500.00×10^6	是 (Python)	中等规模
6Hit[46]	DHC+RL	$\Omega(n\log n)$	10.0×10^4	10.00×10^6	是 (Python)	小规模
AddrMiner[47]	DHC+RL	$\Omega(n\log n)$	100.0×10^4	50.00×10^6	否 (Python)	小规模
6Scan[48]	DHC	$O(n\log n)$	367.0×10^4	$1\,500.00\times10^6$	是 (C++)	中等规模
HMap6[30]	HC(AHC+DHC)	$O(n\log n)$	568.9×10^4	$7\,808.00\times10^6$	是 (C++)	大规模

Table 3 Characteristic Messages of Seedsets

表 3 种子集属性信息

种子集	种子数量	涵盖国家/地区数量	涵盖自治域数量
S_{ICMPv6}	569×10^4	200	16 435
S_{UDP53}	258×10^4	197	13 476
$S_{\text{TCP_ACK80}}$	43×10^4	148	4 638

是因为虽然同作为动态搜索策略, 但在种子数量多且质量高(实验中种子集是预扫描后由活动地址构成的, 其质量较高)的情况下, 6Hit 由于探索性强将耗费一定探测预算, 导致命中率低于 6Scan, 且 6Scan 异步性的扫描设计使其实际发包速率远远高于 6Hit.

3.3 评价指标

本节定义 4 个评价指标来衡量各目标生成扫描算法的性能, 具体地:

1)新发现的前缀数量.表示新发现的去别名的前缀长度为 80 b 的前缀数量. 具体来说, 新发现指排除了种子集中列出的地址, 以及 HMap6 所探测到的别名前缀和 Gasser[34] 公开的别名列表中的别名前缀下的地址/前缀.

2)命中率.表示新发现的活跃地址或前缀占总预算的比例.

3)边际效益. 指新增的新发现的地址或前缀与额外的探测预算之比.

4)时间花销. 包含训练时间花销和扫描时间花销, 训练时间花销指的是利用种子生成待扫描子空间的时间开销, 扫描时间花销指的是扫描所有生成的子空间的时间开销.

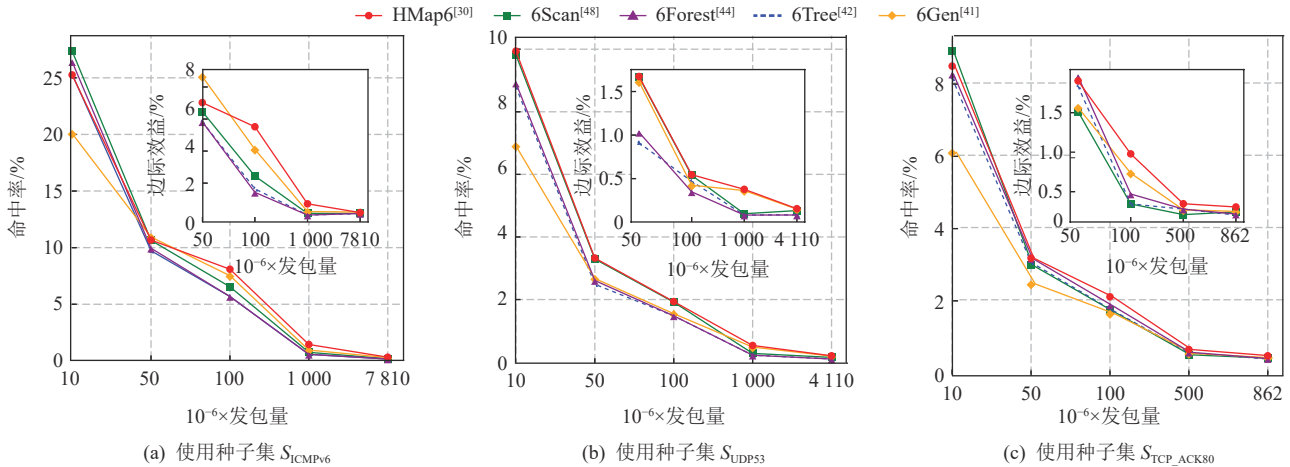
3.4 实网测试

本节比较 HMap6 和基准方法在实网环境下的扫描性能. 扫描时各方法设置最高 100 kpps 的探测速率, 之所以进行速率限制是因为: 1)过高的探测速率容易触发本地网关上的入流量限制; 2)该速率也是相关网络测量研究中[50-51] 常设置的上限速率.

3.4.1 命中率和边际效益对比

图 3 显示了分别使用 3 个种子集时各目标生成扫描算法在命中率和边际效益上的对比. 可以看到, DHC 算法(应用于 HMap6, 6Scan, 6Forest, 6Tree)在使用小预算(小于 5 000 万个)时具有明显的优势, 因为该算法通过分裂聚类的方式试图在生成较少子空间下涵盖尽可能多的种子地址, 这里, 扫描子空间涵盖种子地址越多, 区域活跃度高的可能性就越大. 然而, 随着预算的增加, DHC 的这种优势会减少, 因为 AHC 生成的子空间也逐渐覆盖所有种子地址. 同时, 在中小规模预算下, 6Scan 比 6Tree 和 6Forest 命中率性能更优, 说明其对搜索方向的动态调整提高了一定的命中率. 同样, 随着预算的增加, 扫描覆盖面越来越广, 动态搜索策略不能提供或者生成更多有效的扫描子空间, 其优势也逐渐消失. 在较大的探测预算(大于 10 亿个)下, AHC 的优势逐渐显现出来, 因为它可以比 DHC 在同等子空间大小下生成更多的子空间. 例如, 当使用种子集 S_{ICMPv6} 时, AHC 生成 11.06 万个维度为 4 的子空间, 而 DHC 仅生成 1.72 万个维度为 4 的子空间. 可见, 在不同的扫描预算下, AHC 和 DHC 各有优劣. HMap6 正是通过集成 AHC 和 DHC, 使得其无论在小预算还是大预算下都能取得相对较高的命中率.

边际效益同样随着预算的增加急剧下降. 如图 3 所示在使用种子集 S_{ICMPv6} 时, 当探测预算从 10 亿个增加到 78.1 亿个时, 所有扫描策略的边际效益都下



(a) 使用种子集 S_{ICMPv6} (b) 使用种子集 S_{UDP53} (c) 使用种子集 $S_{\text{TCP_ACK80}}$

Fig. 3 Comparison of hit rate and marginal benefit of each target generation scanning algorithm with various probing budget

图 3 不同发包量下各目标生成扫描方法命中率和边际效益对比

降到每增加100个探测包新发现活动地址数不足0.17个，这表明随着扫描区域的增大，种子指引高效扫描的能力变弱. 与其他算法相比，当预算大于1亿个时，HMap6可以更好地利用种子，从而减缓边际效益值的下降速度. 例如，在使用种子集S_{ICMPv6}的扫描中，HMap6可以生成维度为4的子空间11.9万个，这将消耗78.1亿个探测预算，而其他基准方法则需要扫描维度数为5的子空间才能消耗掉78.1亿个的探测预算，但扫描更大的子空间命中率会出现大幅度的下降.

3.4.2 种子集利用率对比

表4显示了在不同种子集下各子空间生成算法所生成的不同维度数的子空间的数量. 这里子空间的维度数表示地址空间向量的可变维度个数，子空间的可变维度数越多表示子空间范围越大. 例如子空间2001::00**的可变维度数是2，则扫描该子空间将消耗256个探测预算. 由表4可知，DHC算法在每个维数所生成的子空间的数量都是最少的，这是因为其自顶向下的层次分裂聚类模式试图在尽可能少的子空间下覆盖尽可能多的种子. 这也解释了为什么DHC算法会在小预算情况下表现良好，因为其在小预算下生成的子空间覆盖了更多的种子，即待扫描子空间种子密度高，而基于密度驱动的目标生成算法种子密度越高其真实活跃度高的概率越大(在种子质量良好的情况下). 而当扫描预增大时，AHC算法的优点得以突出. 例如，当使用种子集S_{ICMPv6}时，AHC算法生成11.06万个维度数为4的子空间，扫描这些子空间将消耗72.5亿个探测预算，而DHC算法只产生了1.72万个4维子空间，只能消耗11.8亿个探测预算. 换句话说，当扫描设定的探测预算超过11.8亿个时，基于DHC的搜索策略(如6Tree和6Scan)则要在维度数为5的子空间进行扫描，而空间搜索范围16倍指数级增长将大大减少空间的种子密度，同时导致命中率的急剧下降. HMap6所使用的HC算法在各维度数中生成了最多的子空间，因为其同时利用(改进的)AHC和DHC算法进行扫描子空间的生成. 换句话说，HMap6的HC算法达到了最大的种子利用率，使其能够无论在大预算还是小预算下都能保持较高的命中率.

为了验证上述分析，我们运行10亿个及以上规模预算扫描，比较各扫描策略新发现/80前缀的总数和命中率，统计结果如表5所示. 实验选择这个规模的预算作为终止条件是因为各扫描策略在这个体量预算下边际效益值已降低至每增加100个探测报文新增活动地址发现数量不足0.46个. 换句话说，种子地址已不能继续有效地指导更大规模的扫描. 具体在各种子集上，各扫描策略至少完成了所有其生成的维度数为4及以下维数子空间的扫描. 表5可以看到，HMap6在各种子集上发现了最多具有活动地址的前缀. 对各种子集下发现的地址进行合并并删除重复的地址，HMap6总共发现了2 939万个具有活动地址的前缀，取得了最高的命中率，即0.23%，比最先进的其他方法如6Gen提高了11.88%.

3.4.3 总体时间开销对比

使用3个种子集评估各扫描策略完成扫描预算时的总时间开销，统计结果如表6所示，其中扫描设置最大发包速率为100 kpps. 同时在我们的实验平台(无GPU)还对基于深度学习的扫描策略6GAN进行了测试，遗憾的是即便使用最小的种子集($S_{\text{TCP_ACK80}}$，含43万个种子地址)也无法运行成功. 原因是6GAN过于复杂的算法导致了过高的训练成本，可见像6GAN这类基于深度学习的方法不太适用于大规模扫描任务，至少在普通性能的计算机上行不通. 6Gen需要最多的子空间生成时间开销，特别是当种子数量较大时，AHC算法过高的时间复杂度($O(n^3)$)导致时间开销的爆炸式增长. 而HMap6采用改进的AHC和DHC算法，将时间复杂度降低到$O(n\log n)$，完成百万级种子的训练(子空间生成)仅需数秒.

在探测阶段，静态搜索策略(即HMap6, 6Gen, 6Forest, 6Tree)几乎可以达到最大的探测速率(即100 kpps)，因为静态搜索策略所有的目标地址都是根据种子一次性确定的，从而可以直接进行异步无

Table 4 Subspace Numbers with Different Dimension Numbers Generated by Target Generation Algorithms with Different Seedsets

表4 目标生成算法在不同种子集下生成不同维度数的子空间数量

生成算法	S_{ICMPv6}				S_{UDP53}				$S_{\text{TCP_ACK80}}$			
	1维	2维	3维	4维	1维	2维	3维	4维	1维	2维	3维	4维
AHC	47.54×10^4	39.83×10^4	18.09×10^4	11.06×10^4	17.54×10^4	13.77×10^4	7.49×10^4	5.59×10^4	0.61×10^4	1.06×10^4	1.57×10^4	1.21×10^4
DHC	3.32×10^4	3.21×10^4	2.62×10^4	1.72×10^4	1.88×10^4	1.54×10^4	1.83×10^4	1.22×10^4	0.49×10^4	0.22×10^4	0.34×10^4	0.22×10^4
HC	47.71×10^4	40.42×10^4	19.56×10^4	11.92×10^4	17.60×10^4	14.22×10^4	8.72×10^4	6.27×10^4	0.66×10^4	1.13×10^4	1.66×10^4	1.32×10^4

Table 5 Comparison of Scanning Results of Each Target Generation Algorithm

表 5 各目标生成算法扫描结果对比

种子集	探测预算个数	目标生成算法	发现/80前缀总数量	命中率/%
S_{ICMPv6}	78.1 亿	HMap6	25.47×10^6	0.33
		6Gen	22.76×10^6	0.29
		6Scan	15.08×10^6	0.19
		6Forest	12.98×10^6	0.17
		6Tree	12.91×10^6	0.17
S_{UDP53}	41.1 亿	HMap6	8.35×10^6	0.20
		6Gen	7.89×10^6	0.19
		6Scan	6.24×10^6	0.15
		6Forest	4.11×10^6	0.10
		6Tree	4.19×10^6	0.10
S_{TCP_ACK80}	8.60 亿	HMap6	4.14×10^6	0.48
		6Gen	3.65×10^6	0.42
		6Scan	3.65×10^6	0.42
		6Forest	3.47×10^6	0.40
		6Tree	3.37×10^6	0.39
合计	127.8 亿	HMap6	29.39×10^6	0.23
		6Gen	26.27×10^6	0.21
		6Scan	17.40×10^6	0.14
		6Forest	14.98×10^6	0.12
		6Tree	14.90×10^6	0.12

Table 6 Time Cost of Each Target Generation Scanning Algorithm

表 6 各目标生成扫描算法的时间开销

种子集	目标生成算法	子空间生成时间花销	扫描时间花销	总时间花销
S_{ICMPv6}	HMap6	43 s	22 h50 min10 s	22 h50 min53 s
	6Gen	56 h24 min36 s	22 h48 min15 s	79 h12 min51 s
	6Scan	23 s	22 h50 min11 s	22 h50 min34 s
	6Forest	30 s	22 h49 min15 s	22 h49 min45 s
	6Tree	23 s	22 h49 min13 s	22 h49 min36 s
S_{UDP53}	HMap6	19 s	12 h8 min43 s	12 h9 min2 s
	6Gen	5 h43 min12 s	12 h5 min51 s	17 h49 min3 s
	6Scan	10 s	12 h9 min21 s	12 h9 min31 s
	6Forest	11 s	12 h10 min	12 h10 min11 s
	6Tree	10 s	12h10 min21 s	12h10 min31 s
S_{TCP_ACK80}	HMap6	7 s	2 h34 min7 s	2 h34 min14 s
	6Gen	1 min35 s	2 h30 min54 s	2 h32 min29 s
	6Scan	4 s	2 h33 min51 s	2 h33 min55 s
	6Forest	4 s	2 h32 min12 s	2 h32 min16 s
	6Tree	4 s	2 h35 min3 s	2 h35 min7 s

状态扫描, 即在各自独立的线程中发送和接收数据包. 动态搜索策略 6Scan 因其区域编码技术的应用也可以实现异步性扫描, 达到静态搜索策略的发包速率. 综上, HMap6 在 10 亿个发包量规模的扫描中使用较少的时间找到了最多的活跃前缀.

4 总结与展望

4.1 相关研究总结

IPv6 超级庞大的网络地址空间使得研究者面临的一个基本事实是以当前的算力和带宽无法完成对 IPv6 网络的全网暴力扫描, 从而在 IPv4 网络中发展而来的暴力扫描工具(如 ZMap[3] 等)无法直接使用. 这导致基于网络扫描的嗅探资产发现这一互联网基础测量行为功能受阻, 进而后续的 IPv6 互联网范围的拓扑发现与安全分析等研究也无法开展. 基于目标生成的扫描方法以种子地址为抓手, 试图解决当下 IPv6 扫描所面临的效率低下问题, 在不同应用场景下提升 IPv6 网络扫描的能力.

本文从实用性角度对比分析各目标生成算法进行 IPv6 全网大规模扫描探测的能力. 尽管文献 [2] 对比了部分目标生成算法的性能, 但仅局限于早期提出的静态目标生成算法, 缺少对当前新提出的动态扫描算法(如 6Hit 和 6Scan)的分析. 且其实网验证环节仅使用小种子集进行小规模(10 万个种子地址生成 10 万个目标地址)测量分析, 无法评估各算法在大规模(如本文进行百亿发包量探测)场景下的实用性能. 在各算法性能表现差异上, 文献 [2] 缺少深入的对比分析, 而本文从边界收益、种子利用率等多个角度分析各算法性能表现差异的原因.

分析发现, 以 6GAN 为代表的基于深度学习的目标生成技术, 在小规模预算下能够取得较高的命中率, 但这些需要付出较大的算力代价, 且其过高的算法复杂度往往无法训练种子体量较大的种子集, 进而无法胜任大规模扫描任务.

以 6Hit 和 Addrminer 为代表的基于强化学习的动态扫描技术, 在扫描过程中增强了与实际网络的交互, 使得扫描对于初始种子地址质量有较好的弹性, 但动态的方向调整使得扫描器无法以完全异步的方式进行扫描, 极大地降低了扫描的速率. 尽管 6Scan 在扫描器内部设计了简单的区域计数器并利用区域编码技术实现了实时的扫描方向的调整, 维持了扫描的异步性. 但动态的方向调整也仅是在种子地址确定扫描大方向的基础上进行的搜索方向微

调，随着探测预算的持续性增大，这些微调的作用就不显著了，因此动态的目标生成方法较适用于中等规模的扫描任务.

在大规模大覆盖面扫描上，HMap6结合了一种启发式扫描算法利用已声明的BGP前缀信息，进行大覆盖面的种子地址收集，增加了后续目标生成扫描的覆盖面. 同时在目标生成上，HMap6从边际效益的角度出发，最大化利用种子地址生成目标地址以进行探测. 在实网100亿个规模预算的扫描测量中，HMap6发现了2 939万个含有活跃地址的/80网段[49]，这一数量比现有公开的IPv6地址列表[34]提升了12.64倍.

4.2 未来工作展望

后续的研究工作可以以目标生成扫描所提供的地址列表为基础开展针对IPv6互联网络级别的各项研究调查，这其中针对IPv6互联网的拓扑发现、异常检测和安全分析等与互联网基础资产密切相关的研究课题值得引起进一步的探讨和关注：

1）在IPv6互联网拓扑发现方面，基于目标生成扫描已获得大量且分布广泛的活跃地址，接下来可利用高效Traceroute主动探测工具，如Yarrpv6[29]或FlashRoute[51]，对目标地址进行探测从而获得路径上的拓扑关系. 然后结合BGP等信息可进一步构建自治域级和国家级网络拓扑关系[52]. 最后还可通过调用MaxMind[53]或IP2Location[54]等IP地址到地理坐标的映射接口来标注IP地址的位置信息，从而构建网络路径信息与地理位置的映射地图.

2）在IPv6互联网异常检测方面，相关研究[55-57]表明仅通过简单的网络测量手段（如Ping和Traceroute）结合一些人工智能算法对网络性能（如RTT）的推断就能诊断网络异常问题. 这些技术可应用在目标生成式扫描对IPv6网络长期扫描所伴随产生的性能（即RTT）测量数据中，从而实现针对IPv6网络的大覆盖网络异常监测和中断故障定位. 这类研究的意义在于即便在非合作条件下，即不需要得到运营商的允许，或不需要对其主干网络流量等数据进行分析，仍能对大范围的网络进行连通性和故障性诊断.

3）在IPv6互联网安全分析方面，相关研究已关注到IPv6协议所带来的新的安全问题，如文献[58]指出了双栈服务器可能在IPv6网络侧存在更多的安全隐患；文献[59]分析了IPv6网络客户端路由设备（CPE）存在的安全风险. 但这些IPv6互联网范围的研究存在目标主机少、分布范围小且测量协议少等缺陷，仍存在大量常用的且存在安全隐患的应用需要测量和分析，如HTTPS证书漏洞[60]、SSH协议漏洞[61]、SMTP协议漏洞[62-63]、DNS劫持测量[64-65]、FTP服务器可写性隐患[66]、工控系统[67]和IoT设备[68-69]、相关协议漏洞以及基于IPID设备指纹信息的侧信道攻击[70-72]检测等. 而在目标生成式扫描所提供的IPv6地址列表基础上结合高效的协议扫描工具（如Zgrab[6]等）可完成上述IPv6互联网级协议安全的测量与分析.

4）在IPv6互联网高效测量方面，与传统的网络扫描探针通常部署在普通运算能力的服务器甚至笔记本电脑上且探测速率受限于网络带宽和防DDoS攻击等网关安全策略条件的限制不同. 文献[73]将交换机转变为高速率的网络扫描器，其设计的IMap扫描器基于可编程交换机实现了更高效的网络探测. IMap克服了可编程交换机计算和内存资源的限制，在可编程交换机上实现了扫描地址随机化和速率自适应的探测数据包生成机制，以及高效的响应数据包处理方法. IMap可在8 min内探测完成高达250亿个地址. 未来可以进一步扩展IMap，使其能够针对IPv6网络完成更大规模和范围的测量任务，以适应IPv6网络不断增长的复杂性和数据量.

综上，基于目标生成的IPv6网络的高效扫描工具以及发布的分布广泛的活跃地址列表将极大地促进IPv6互联网级测量与调查研究工作的开展.

5 结束语

IPv6网络不同于传统的IPv4网络，其能极大地扩展地址空间的范围，改变了地址分配模式，并允许直接进行端到端的互联网通信. 且随着近年IPv6网络的快速发展和大力部署，现阶段研究IPv6网络的结构性和安全性问题至关重要.

研究人员当前已普遍意识到IPv6网络庞大的地址空间无法进行全覆盖的暴力扫描. 虽然早期研究引入了一些技术来主动或被动地进行地址收集，但这些方法仍然存在效率低下且覆盖面窄的限制. 目标生成式扫描技术的提出，极大地提升了IPv6网络扫描的效率和覆盖面. 特别是在大规模大覆盖扫描方面，以HMap6为例，其能在较短时间发现大量IPv6网络空间中的活跃地址，活跃地址范围覆盖可达198个国家/地区、1.47万个自治域以及2 939万个/80网段. 鉴于IPv6全网暴力扫描研究的不切实际，IPv6全网范围的研究调查只能回归到早期对IPv4互联网使用的方法，即使用地址列表开展研究. 这样，地址列

表就作为IPv6整个网络地址空间的一个有代表性的子集. 以HMap6为代表的目标生成式扫描能在短时间内获取大覆盖且都是响应地址构成的地址列表, 从数量和质量上提升了现有公开IPv6地址列表的水平, 为互联网级IPv6网络研究提供了基础.

作者贡献声明: 侯冰楠提出了算法思路、实验方案, 并对相关工作进行了梳理总结和撰写论文; 刘宁和李雄略负责完成实验; 周桐庆和陈颖文修订论文; 蔡志平和卢凯提出修改意见并指导论文撰写.

参 考 文 献

[1] Dhamdhere A, Luckie M, Huffaker B, et al. Measuring the deployment of IPv6: Topology, routing and performance[C]//Proc of the 12th ACM Internet Measurement Conf (IMC). New York: ACM, 2012: 537−550

[2] Li Guo, He Lin, Song Guanglei, et al. Survey of IPv6 address probing technology based on seed address[J]. Telecom Science, 2019, 35(12): 24−37 (in Chinese)
(李果, 何林, 宋光磊, 等. 基于种子地址的IPv6地址探测技术综述[J]. 电信科学, 2019, 35(12): 24−37)

[3] Durumeric Z, Wustrow E, Halderman J A, et al. ZMap: Fast Internet-wide scanning and its security applications[C]//Proc of the 22nd USENIX Security Symp (USENIX Security). Berkeley, CA: USENIX Association, 2013: 605–620

[4] Graham R D. MASSCAN: Mass IP port scanner[CP/OL]. [2023-09-20].https://github.com/robertdavidgraham/masscan

[5] Matherly J. Shodan: The search engine for Internet-connected devices[EB/OL]. [2023-09-20].https://www.shodan.io

[6] Durumeric Z, Adrian D, Mirian A, et al. A search engine backed by Internet-wide scanning[C]//Proc of the 22nd ACM SIGSAC Conf on Computer and Communications Security (CCS). New York: ACM, 2015: 542–553

[7] Fan Xun, Heidemann J. Selecting representative IP addresses for Internet topology studies[C]//Proc of the 10th ACM Internet Measurement Conf (IMC). New York: ACM, 2010: 411–423

[8] Gasser O, Scheitle Q, Gebhard S, et al. Scanning the IPv6 Internet: Towards a comprehensive hitlist [J]. arXiv preprint, arXiv: 1607.05179, 2016

[9] Song Guanglei, He Lin, Wang Zhiliang, et al. Towards the construction of global IPv6 Hitlist and efficient probing of IPv6 address space[C]//Proc of the 28th IEEE/ACM Int Symp on Quality of Service (IWQoS). Piscataway, NJ: IEEE, 2020: 11–20

[10] Huz G, Bauer S, Claffy K, et al. Experience in using MTURK for network measurement[C]//Proc of the 15th ACM SIGCOMM Workshop on Crowdsourcing and Crowdsharing of Big (Internet) Data. New York: ACM, 2015: 27–32

[11] Lone Q, Luckie M, Korczynski M, et al. Using crowdsourcing marketplaces for network measurements: The case of spoofer[C]//Proc of the 18th Network Traffic Measurement and Analysis Conf (TMA). Piscataway, NJ: IEEE, 2018: 10–18

[12] Varvello M, Blackburn J, Naylor D, et al. EYORG: A platform for crowdsourcing Web quality of experience measurements[C]//Proc of the 12th Int Conf on Emerging Networking Experiments and Technologies (CoNEXT). New York: ACM, 2016: 399–412

[13] Gont F, Chown T. RFC 7707 Network Reconnaissance in IPv6 Networks[S/OL]. 2016 [2023-09-20].https://datatracker.ietf.org/doc/rfc7707/

[14] Strowes S D. Bootstrapping active IPv6 measurement with IPv4 and public DNS [J]. arXiv preprint, arXiv: 1710.08536, 2017

[15] Fiebig T, Borgolte K, Hao Shuang, et al. Something from nothing (there): Collecting global IPv6 datasets from DNS[C]//Proc of the 17th Passive and Active Measurement Conf (PAM). Berlin: Springer, 2017: 30–43

[16] Fiebig T, Borgolte K, Hao Shuang, et al. In rDNS we trust: Revisiting a common data-source's reliability[C]//Proc of the 18th Passive and Active Measurement Conf (PAM). Berlin: Springer, 2018: 131–145

[17] Borgolte K, Hao S, Fiebig T, et al. Enumerating active IPv6 hosts for largescale security scans via DNSSEC-signed reverse zones[C]//Proc of the 18th IEEE Symp on Security and Privacy (S&P). Piscataway, NJ: IEEE, 2018: 770–784

[18] Gasser O, Scheitle Q, Foremski P, et al. Clusters in the expanse: Understanding and unbiasing IPv6 Hitlists[C]//Proc of the 18th ACM Internet Measurement Conf (IMC). New York: ACM, 2018: 364–378

[19] Rapid7. DNS records (ANY)[EB/OL]. [2023-09-20].https://scans.io/study/sonar.fdns

[20] Addy Y. Bitnodes API[EB/OL]. [2023-09-20].https://bitnodes.earn.com/

[21] Alexa. The top sites on the Web[EB/OL]. [2023-09-20].https://www.alexa.com/topsites

[22] Statvoo. Website analytics and reviews[EB/OL]. [2023-09-20].https://statvoo.com

[23] Cisco. Cisco umbrella[EB/OL]. [2023-09-20].https://umbrella.cisco.com

[24] PremiumDrops. Domain zone file and zone changes downloads [EB/OL]. [2023-09-20].https://www.premiumdrops.com/zones.html

[25] CAIDA. IPv6 topology[EB/OL]. [2023-09-20]. http://www.caida.org/data/active/ipv6_allpref_topology_dataset.xml

[26] RIPE. RIPE NCC Atlas[EB/OL]. [2023-09-20].https://atlas.ripe.net

[27] Rye E C, Beverly R. Discovering the IPv6 network periphery[C]//Proc of the 20th Passive and Active Measurement Conf (PAM). Berlin: Springer, 2020: 3–18

[28] Luckie M. Scamper: A scalable and extensible packet prober for active measurement of the Internet[C]//Proc of the 10th ACM Internet Measurement Conf (IMC). New York: ACM, 2010: 239–245

[29] Beverly R, Durairajan R, Plonka D, et al. In the IP of the beholder: Strategies for active IPv6 topology discovery[C]//Proc of the 18th ACM Internet Measurement Conf (IMC). New York: ACM, 2018: 308–321

[30] Hou Bingnan, Cai Zhiping, Wu Kui, et al. Search in the expanse: Towards active and global IPv6 hitlists[C]//Proc of the 23rd IEEE Int Conf on Computer Communications (INFOCOM). Piscataway, NJ: IEEE, 2023: 10–20

[31] Hu Qinwen, Brownlee N. How interface ID allocation mechanisms are performed in IPv6[C]//Proc of the 14th ACM Int Conf on Emerging Networking Experiments and Technologies Student Workshop (CoNEXT Workshop). New York: ACM, 2014: 26–27

[32] RIPE. RIPEstat data API[EB/OL]. 2021 [2023-09-20].https://stat.ripe.net/docs/data_api/

[33] Barnes R, Altmann R, Kerr D. Mapping the great void smarter scanning for IPv6 [R/OL]. 2012 [2023-09-20].https://www.caida.org/workshops/isma/1202/slides/aims1202_rbarnes.pdf

[34] Gasser. IPv6 Hitlist service[EB/OL]. [2023-09-20].https://ipv6hitlist.github.io/

[35] Ullrich J, Kieseberg P, Krombholz K, et al. On reconnaissance with IPv6: A pattern-based scanning approach[C]//Proc of the 10th IEEE Int Conf on Availability, Reliability and Security (ARES). Piscataway, NJ: IEEE, 2015: 186–192

[36] Foremski P, Plonka D, Berger A. Entropy/IP: Uncovering structure in IPv6 addresses[C]//Proc of the 16th ACM Internet Measurement Conf (IMC). New York: ACM, 2016: 167–181

[37] Narten T, Jinmei T, Thomson S. RFC 4862 IPv6 stateless address autoconfiguration [S/OL]. 2007 [2023-09-23].https://www.rfc-editor.org/rfc/rfc4862

[38] Cui Tianyu, Gou Gaopeng, Xiong Gang. 6GCVAE: Gated convolutional variational autoencoder for IPv6 target generation[C]// Proc of the 24th Pacific-Asia Conf on Knowledge Discovery and Data Mining. Berlin: Springer, 2020: 609–622

[39] Cui Tianyu, Xiong Gang, Gou Gaopeng, et al. 6VecLM: Language modeling in vector space for IPv6 target generation[C]//Proc of the 21st European Conf on Machine Learning and Knowledge Discovery in Databases. Berlin: Springer, 2021: 192–207

[40] Cui Tianyu, Gou Gaopeng, Xiong Gang, et al. 6GAN: IPv6 multi-pattern target generation via generative adversarial nets with reinforcement learning[C]//Proc of the 21st IEEE Int Conf on Computer Communications (INFOCOM). Piscataway, NJ: IEEE, 2021: 11–20

[41] Murdock A, Li F, Bramsen P, et al. Target generation for Internet-wide IPv6 scanning[C]//Proc of the 17th ACM Internet Measurement Conf (IMC). New York: ACM, 2017: 242–253

[42] Liu Zhizhu, Xiong Yinqiao, Liu Xie, et al. 6Tree: Efficient dynamic discovery of active addresses in the IPv6 address space[J]. Computer Networks, 2019, 155(1): 31−46

[43] Yang Tao, Hou Bingnan, Cai Zhiping, et al. 6Graph: A graph-theoretic approach to address pattern mining for Internet-wide IPv6 scanning[J]. Computer Networks, 2022, 203(1): 108−130

[44] Yang Tao, Hou Bingnan, Zhou Tongqing, et al. 6Forest: An ensemble learning-based approach to target generation for Internet-wide IPv6 scanning[C]//Proc of the 22nd IEEE Int Conf on Computer Communications (INFOCOM). Piscataway, NJ: IEEE, 2022: 1679–1688

[45] Zirngibl J, Steger L, Sattler P, et al. Rusty clusters? Dusting an IPv6 research foundation[C]//Proc of the 22nd ACM Internet Measurement Conf (IMC). New York: ACM, 2022: 395–409

[46] Hou Bingnan, Cai Zhiping, Wu Kui, et al. 6Hit: A reinforcement learning-based approach to target generation for Internet-wide IPv6 scanning[C]//Proc of the 21st IEEE Int Conf on Computer Communications (INFOCOM). Piscataway, NJ: IEEE, 2021: 11–20

[47] Song Guanglei, Yang Jiahai, He Lin, et al. AddrMiner: A comprehensive global active IPv6 address discovery system[C]//Proc of the 22nd USENIX Annual Technical Conf (ATC). Berkeley, CA: USENIX Association, 2022: 309–326

[48] Hou Bingnan, Cai Zhiping, Wu Kui, et al. 6Scan: A high-efficiency dynamic Internet-wide IPv6 scanner with regional encoding[J]. IEEE/ACM Transactions on Networking, 2023, 99(1): 1−16

[49] Hou Bingnan. HMap6 IPv6 Hitlists[EB/OL]. [2023-09-20].https://hbn1987.github.io/HMap6.github.io/

[50] Beverly R. Yarrp'ing the Internet: Randomized high-speed active topology discovery[C]//Proc of the 16th ACM Internet Measurement Conf (IMC). New York: ACM, 2016: 413–420

[51] Huang Yuchen, Rabinovich M, Al-Dalky R. FlashRoute: Efficient traceroute on a massive scale[C]//Proc of the 20th ACM Internet Measurement Conf (IMC). New York: ACM, 2020: 443–455

[52] Spring N, Mahajan R, Wetherall D. Measuring ISP topologies with rocketfuel[J]. ACM SIGCOMM Computer Communication Review, 2002, 32(4): 133−145

[53] MaxMind. Geolite API[EB/OL]. [2023-09-20]. http://dev.maxmind.com/geoip/legacy/geolite/

[54] MaxMind. IP2Location[EB/OL]. [2023-09-20]. http://lite.ip2location.com

[55] Fontugne R, Pelsser C, Aben E, et al. Pinpointing delay and forwarding anomalies using large-scale traceroute measurements[C]// Proc of the 17th ACM Internet Measurement Conf (IMC). New York: ACM, 2017: 15–28

[56] Padmanabhan R, Schulman A, Dainotti A, et al. How to find correlated Internet failures[C]//Proc of the 19th Passive and Active Measurement Conf (PAM). Berlin: Springer, 2019: 210–227

[57] Hou Bingnan, Hou Changshen, Zhou Tongqing, et al. Detection and characterization of network anomalies in large-scale RTT time series[J]. IEEE Transactions on Network and Service Management, 2021, 18(1): 793−806

[58] Czyz J, Luckie M, Allman M, et al. Don't forget to lock the back door! A characterization of IPv6 network security policy[C]//Proc of the 16th Network and Distributed System Security Symp (NDSS). Berkeley, CA: USENIX Association, 2016: 13–25

[59] Li Xiang, Liu Baojun, Zheng Xiaofeng, et al. Fast IPv6 network periphery discovery and security implications[C]//Proc of the 51st IEEE/IFIP Int Conf on Dependable Systems and Networks (DSN). Piscataway, NJ: IEEE, 2021: 88–100

[60] Durumeric Z, Kasten J, Bailey M, et al. Analysis of the HTTPS certificate ecosystem[C]//Proc of the 13th ACM Internet

Measurement Conf (IMC). New York: ACM, 2013: 291–304
[61] Durumeric Z, Li F, Kasten J, et al. The matter of heartbleed[C]//Proc of the 14th ACM Internet Measurement Conf (IMC). New York: ACM, 2014: 475–488
[62] Durumeric Z, Adrian D, Mirian A, et al. Neither snow nor rain nor MITM...: An empirical analysis of email delivery security[C]//Proc of the 15th ACM Internet Measurement Conf (IMC). New York: ACM, 2015: 27–39
[63] Holz R, Amann J, Mehani O, et al. TLS in the wild: An Internet-wide analysis of TLS-based protocols for electronic communication [C]//Proc of the 16th Network and Distributed System Security Symp (NDSS). Berkeley, CA: USENIX Association, 2016: 13–25
[64] Kuhrer M, Hupperich T, Bushart J, et al. Going wild: Large-scale classification of open DNS resolvers[C]//Proc of the 15th ACM Internet Measurement Conf (IMC). New York: ACM, 2015: 355–368
[65] Pearce P, Jones B, Li F, et al. Global measurement of DNS manipulation[C]//Proc of the 26th USENIX Security Symp (USENIX Security). Berkeley, CA: USENIX Association, 2017: 307–323
[66] Springall D, Durumeric Z, Halderman J A. FTP: The forgotten cloud[C]//Proc of the 46th IEEE/IFIP Int Conf on Dependable Systems and Networks (DSN). Piscataway, NJ: IEEE, 2016: 503–513
[67] Li F, Durumeric Z, Czyz J, et al. You've got vulnerability: Exploring effective vulnerability notifications[C]//Proc of the 25th USENIX Security Symp (USENIX Security). Berkeley, CA: USENIX Association, 2016: 1033–1050
[68] Feng Xuan, Li Qiang, Wang Haining, et al. Acquisitional rule-based engine for discovering Internet-of-thing devices[C]//Proc of the 27th USENIX Security Symp (USENIX Security). Berkeley, CA: USENIX Association, 2018: 327–341
[69] Li Zhenyu, Ding Yong, Yuan Fang, et al. An integrated protection method of moving target defense and access control based on IPv6 network[J]. Journal of Computer Research and Development, 2022, 59(5): 1105−1119 (in Chinese)
(李振宇, 丁勇, 袁方, 等. 基于 IPv6 网络的移动目标防御与访问控制融合防护方法[J]. 计算机研究与发展, 2022, 59(5): 1105−1119)
[70] Geoffrey A, Antonio M E, Jedidiah R C. Detecting TCP/IP connections via IPID Hash collisions[J]. Proceedings on Privacy Enhancing Technologies, 2019, 4(1): 311−328
[71] Feng Xuan, Fu Chuangpu, Li Qi, et al. Off-path TCP exploits of the mixed IPID assignment[C]//Proc of the 20th ACM SIGSAC Conf on Computer and Communications Security (CCS). New York: ACM, 2020: 1323–1335
[72] Feng Xuewei, Li Qi, Sun Kun, et al. PMTUD is not panacea: Revisiting IP fragmentation attacks against TCP[C]//Proc of the 22nd Network and Distributed System Security Symp (NDSS). Berkeley, CA: USENIX Association, 2022: 12–25
[73] Li Guangyu, Zhang Menghao, Guo Chen, et al. IMap: Fast and scalable in-network scanning with programmable switches[C]//Proc of the 19th USENIX Symp on Networked Systems Design and Implementation (NSDI). Berkeley, CA: USENIX Association, 2022: 667–681



Hou Bingnan, born in 1987. PhD, postdoc. His main research interests include network measurement and network security.
侯冰楠, 1987 年生. 博士, 博士后. 主要研究方向为网络测量、网络安全.



Liu Ning, born in 1998. Master candidate. His main research interest includes network measurement.
刘　宁, 1998 年生. 硕士研究生. 主要研究方向为网络测量.



Li Xionglüe, born in 1990. PhD candidate. His main research interests include network measurement and artificial intelligence.
李雄略, 1990 年生. 博士研究生. 主要研究方向为网络测量、人工智能.



Zhou Tongqing, born in 1991. PhD, assistant professor. His main research interests include privacy preserving and edge computing.
周桐庆, 1991 年生. 博士, 助理研究员. 主要研究方向为隐私保护、边缘计算.



Chen Yingwen, born in 1979. PhD, professor. Senior member of CCF. His main research interests include network protocols and AIoT.
陈颖文, 1979 年生. 博士, 教授. CCF 高级会员. 主要研究方向为网络协议、人工智能物联网.



Cai Zhiping, born in 1975. PhD, professor, PhD supervisor. Distinguished member of CCF. His main research interests include network measurement, network security, and artificial intelligence.
蔡志平, 1975 年生. 博士, 教授, 博士生导师. CCF 杰出会员. 主要研究方向为网络测量、网络安全、人工智能.



Lu Kai, born in 1973. PhD, professor, PhD supervisor. Standing director of CCF. His main research interests include high performance computing, intelligent computing, and network security.
卢　凯, 1973 年生. 博士, 研究员, 博士生导师. CCF 常务理事. 主要研究方向为高性能计算、智能计算、网络安全.