Enumerating Active IPv6 Hosts for Large-scale Security Scans via DNSSEC-signed Reverse Zones

Borgolte, Kevin; Hao, Shuang; Fiebig, Tobias; Vigna, Giovanni

DOI
10.1109/SP.2018.00027

Publication date
2018

Document Version
Accepted author manuscript

Published in
Proceedings of 39th IEEE Symposium on Security and Privacy (SP) 2018

Enumerating Active IPv6 Hosts for Large-scale Security Scans via DNSSEC-signed Reverse Zones

Kevin Borgolte*, Shuang Hao†, Tobias Fiebig‡, Giovanni Vigna*
*University of California, Santa Barbara †University of Texas at Dallas ‡Delft University of Technology
*{kevinbo, vigna}@cs.ucsb.edu †shao@utdallas.edu ‡t.fiebig@tudelft.nl

***Abstract*—Security research has made extensive use of exhaustive Internet-wide scans over the recent years, as they can provide significant insights into the overall state of security of the Internet, and ZMap made scanning the entire IPv4 address space practical. However, the IPv4 address space is exhausted, and a switch to IPv6, the only accepted long-term solution, is inevitable. In turn, to better understand the security of devices connected to the Internet, including in particular Internet of Things devices, it is imperative to include IPv6 addresses in security evaluations and scans. Unfortunately, it is practically infeasible to iterate through the entire IPv6 address space, as it is 2^{96} *times* larger than the IPv4 address space. Therefore, enumeration of active hosts prior to scanning is necessary. Without it, we will be unable to investigate the overall security of Internet-connected devices in the future.**

In this paper, we introduce a novel technique to enumerate an active part of the IPv6 address space by walking DNSSEC-signed IPv6 reverse zones. Subsequently, by scanning the enumerated addresses, we uncover significant security problems: the exposure of sensitive data, and incorrectly controlled access to hosts, such as access to routing infrastructure via administrative interfaces, all of which were accessible via IPv6. Furthermore, from our analysis of the differences between accessing dual-stack hosts via IPv6 and IPv4, we hypothesize that the root cause is that machines automatically and by default take on globally routable IPv6 addresses. This is a practice that the affected system administrators appear unaware of, as the respective services are almost always properly protected from unauthorized access via IPv4.

Our findings indicate *(i)* that enumerating active IPv6 hosts is practical without a preferential network position contrary to common belief, *(ii)* that the security of active IPv6 hosts is currently still lagging behind the security state of IPv4 hosts, and *(iii)* that unintended IPv6 connectivity is a major security issue for unaware system administrators.

I. Introduction

There has been a multitude of Internet-wide security challenges of varying severity over the recent years. Heartbleed [1] and SSL related vulnerabilities [2, 3], common misconfigurations of database systems [4], and other issues like protocol amplifiers [5, 6] have been investigated closely. Studying these issues methodologically has only been possible because exhaustive security scans of the Internet Protocol version 4 (IPv4) address space became practical through ZMap in late 2013 [7]. Since then, Internet-wide IPv4 security scans have become an integral part of modern security research.

The total number of IPv4 addresses is, however, limited. For many of those addresses, their use is further restricted through special use arrangements, and because of large allocations to institutions that were early adopters of the Internet. In fact, all addresses managed by the Internet Assigned Numbers Authority (IANA) have been allocated as of September 24, 2015 when the American Registry for Internet Numbers (ARIN) allocated its last IPv4 address [8].

The accepted long-term solution to the IPv4 address exhaustion problem is considered to be the Internet Protocol version 6 (IPv6) [9]. Contrary to the 32-bit wide addresses of IPv4, IPv6 uses 128-bit wide addresses (7.9×10^{28} as many as IPv4) and its adoption would eliminate the need for further address resources for the foreseeable future.

Indeed, IPv6 has gained significant traction in recent years: In August 2016, Google reported that almost 13% of their users accessed their services via IPv6. This number increased by an order of magnitude in just three years from 1.3% as of July 2013 [10]. Similarly, the Internet Society reports that "global IPv6 traffic has grown more than 500% since June 6, 2012." Many other network operators have deployed IPv6 to significant parts of their network since then [11]. In fact, for some networks, up to 97% of all devices use IPv6 (Table I).

Unfortunately, the vast address space of IPv6 threatens to take the important tool of Internet-wide scans away from the security community. Theoretically, for IPv6, up to 2^{128} addresses (approximately 3.4×10^{38}) can be allocated. While scanning all reachable devices is considered to be a solved problem for the IPv4 address space [7], it is practically infeasible to scan the entire IPv6 address space, because it is larger than the IPv4 address space by 2^{96} (28 orders of magnitude). In fact, a sweep over the entire IPv6 address space would take 7.532×10^{23} years with state-of-art tools for Internet-wide scanning.

Due to the Internet's continuing growth and its increasing dependence on IPv6 globally, it is critical to include IPv6-connected devices in future Internet-wide security evaluations, in addition to IPv4. This need is further amplified by the fact that IPv6 traffic is commonly enabled (by default). Often no standard security mechanisms, such as firewalls, have been put in place for IPv6, even though they are already in place for IPv4. In turn, it exposes the respective hosts to attacks from miscreants via IPv6 [12, 13].

At the same time, it remains difficult to perform Internet-wide IPv6 security scans, which leaves a dangerous blind spot. To address this issue, authors have started to suggest various techniques to perform Internet-wide IPv6 security scans, which leverage IPv6 seed sets to scan IPv6 hosts. The most recent of these, 6gen, has been presented by Murdock et al. [14]. However, most existing approaches to collect active IPv6 addresses as seed sets require network vantage points or leverage older, possibly stale, public datasets. For example, some techniques require access to content delivery networks or

Category	Network Operator	Percentage
Wireless Carrier	Digicel Trinidad & Tobago	97.04%
	Verizon Wireless	77.65%
	T-Mobile USA	71.09%
University	University of Twente	79.17%
	Virginia Tech	70.06%
Organization	SPAWAR[1]	74.52%
Broadband Provider	Google Fiber	64.96%
	xs4all[2]	61.75%

Table I: **IPv6 penetration of real-world networks [20].**
[1] United States Space and Naval Warfare Systems Command.
[2] Netherlands.

traffic brokers to observe IPv6 traffic and collect addresses [15, 16]. Others extract IPv6 addresses from historical forward DNS records, in the hope that they are still active [13]. Unfortunately, some techniques to collect these records, such as `ANY` queries, have since been deprecated by the operators of major nameservers to protect from denial of service attacks [17], which renders them impractical for IPv6 address collection. Fiebig et al. [18] recently introduced a different methodology to enumerate IPv6 hosts, namely by exploiting the `NXDOMAIN` semantics in the DNS ecosystem. However, their technique can be mitigated comparatively easily, as they demonstrated on an industry conference in 2016 [19]. Therefore, it is necessary to identify new seed-set collection techniques that allow researchers, who might not have access to network vantage points, to include IPv6-connected devices at scale in Internet-wide security evaluations.

To retain the capabilities of security researchers to conduct Internet-wide scans, in this paper, we introduce a novel IPv6 address enumeration technique that leverages DNSSEC-signed IPv6 reverse zones. We show that our approach enumerates classes of active IPv6 addresses that existing techniques miss, and that prior work has not evaluated. Furthermore, our technique does not depend on any implementation-specific behavior and it is resilient against the mitigation techniques that have been put in place to protect against the enumeration techniques of prior work. Instead, to prevent our enumeration technique, significant changes to the DNSSEC standard are required.

In our evaluation, we discovered that IPv6-connected hosts expose a variety of critical security issues: exposed file sharing, access to interior and exterior routing protocols, remote access to switches and routers, remote monitoring, hosts that can be exploited to launch reflected and amplified denial of service attacks, and, alarmingly, remote system management ports vulnerable to attacks that allow full machine takeover (e.g., IPMI, which provides practically physical access through remote keyboard and video).

In this paper, we make the following contributions:

- We introduce a practical enumeration technique that effectively exploits DNSSEC zone walking to identify active IPv6 hosts by utilizing unique features and the well-structured format of the IPv6 reverse DNS tree. We focus on reverse zones that have deployed `NSEC3` to thwart existing zone-walking attacks. Specifically, we exploit intricacies of how the IPv6 reverse zone is organized to make enumerating active IPv6 addresses in the face of `NSEC3` practical.
- Our methodology is resilient against mitigation techniques, including techniques that are effective against earlier enumeration approaches, and to mitigate it modifications to the DNSSEC standard are required. In fact, we enumerate hosts that have been hidden from established methodology using existing mitigations already.
- Using our methodology, we identify several vulnerabilities and misconfigurations of hosts reachable via IPv6 that were hidden from scans using methodology of prior work. Our results indicate that the exposed IPv6 addresses can cause additional and significant security risks, and network operators are required to take precautions when adding IPv6 addresses into the DNSSEC-signed reverse zones, as it inevitably leaks information about the presence of those hosts to potential attackers.

II. Background

Some background information on the Domain Name System (DNS), DNSSEC, denial of existence records, and the way the IPv6 reverse zone is organized is required for our enumeration technique.

A. Domain Name System and DNSSEC

DNS is a core protocol of the current Internet architecture. It allows using easily identifiable hierarchically organized names instead of IP addresses to access services online. While the basic idea of the DNS is straightforward [21], denials of existence (`NXDOMAIN`) require some attention, as our approach builds upon their equivalent in the scope of DNSSEC (Section III).

In a simplified schema (Figure 1), a client talks to a nameserver to inquire about whether a specific name for a specific resource record (RR) type exists within a zone. If the record does exist, then the nameserver responds with the respective answer (e.g., in case of an `A` record, with the IPv4 address mapping for a name). If the record does not exist, the nameserver generates a `NXDOMAIN` response (`NX` signifying "non-existing").

Unfortunately, however, the DNS protocol does not provide authenticity and it is susceptible to a variety of attacks, including man-in-the-middle attacks, like filtering, redirection, and response spoofing [22, 23]. An intermediate nameserver could (maliciously) hijack `NXDOMAIN` responses and replace them with a record that points to an advertisement website [24, 25]. While the intermediate nameserver is intentionally violating the standard, it is technically able to return bogus responses because they are not authenticated.

DNSSEC aims to solve these authentication problems through cryptographic signatures for records contained as part of a zone. Authenticating existing records is a straightforward extension of DNS through a signature record type (`RRSIG`) for each original record, which is signed with a zone-signing key (`ZSK`). The public key portion of the `ZSK` is hosted in the zone, while the parent zone provides a hash of the `ZSK` in a `DS` RR. In turn, it solves the problem of distributing public keys in a trustworthy manner through DNS' hierarchical nature and its existing chain of trust from the root zone to the queried zone. Intuitively, signing `NXDOMAIN` RRs would be possible if the zone-signing key is available at the nameserver, so that the generated records can be signed online. However, DNSSEC

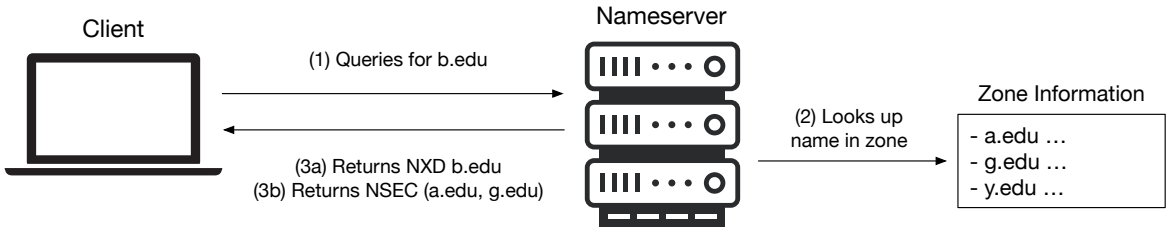

Figure 1: **Example DNS interaction between a client querying a nameserver without and with DNSSEC.** The client queries the nameserver for a record of the domain "b.edu" (1). The nameserver looks up the resource record (RR) in the zone information (2). Here, the queried resource does not exist in the zone file. If DNSSEC is not present, then the nameserver responds with a single `NXDOMAIN` response that is generated online (3a). If DNSSEC is present, then the nameserver responds with an authenticated response. Since DNSSEC discourages online signing, a pre-signed entry must exist. However, pre-signed denials of existence for any possible query are impractical from a space and computational perspective. Therefore, DNSSEC returns pre-signed denials of existence for an entire name range: the previous existing entry with an associated record is "a.edu", the next existing entry with a record is "g.edu" (3b), effectively leaking the existence of those names within the zone.

discourages the use of online signing to prevent denial of service attacks against the nameserver and chosen-plaintext attacks against the zone-signing key [26]. Instead, it strongly encourages to serve zone information that was signed offline. Consequently, authenticating denials of existence naïvely is impractical: all non-existing names would have to be signed and it would require operators to create zones of practically unbounded size. As a solution, a single `NSEC` RR is used to deny the existence of a range of records: it describes the previous existing name and the next existing name. For example, an `NSEC` record might point to "a.edu" as the previous existing name and "g.edu" as the next existing record. Then, any query for a name that is lexically between "a.edu" and "g.edu," e.g., "c.edu" or "foo.edu," would result in the same authenticated `NSEC` response. This is an efficient authenticated denial of existence, satisfying the requirements of DNSSEC.

B. IPv6 and Reverse IPv6 Zones

Contrary to IPv4's quad-dotted decimal representation, IPv6 addresses are represented through 32 hexadecimal digits, which are divided into eight groups of four digits to ease readability, e.g., `2001:0db8:0000:0bad:f00d:feed:cafe:0001`. For convenience, addresses can be abbreviated by removing leading zeroes and replacing the largest consecutive group of zeroes with a double colon, e.g., the above address can be abbreviated to the shorter `2001:db8::bad:f00d:feed:cafe:1`.

Conceptually, reverse zones are like any other standard DNS zone, but they have a specific meaning: They are used to map an address or resource, such as an IPv4 or IPv6 address, to a name instead of the other way around. For IPv6, the designated reverse zone is `ip6.arpa` and it is hierarchically organized at nibble (a nibble is a single hexadecimal digit) boundaries in reverse order. Listing 1 depicts an example reverse zone for `2001:db8::/32` with two entries, one for `2001:db8::bad:f00d:feed:cafe:2` pointing to "h.a.edu" and one for `2001:db8::bad:f00d:feed:cafe:9` pointing to "s.a.edu."

In practice, reverse address zones are used for a variety of reasons. Initially devised for troubleshooting, reverse lookups for forward-confirmed reverse DNS names are nowadays its main use case and considered best operational practice [27]. A forward-confirmed reverse DNS lookup corresponds to looking up the domain name with an address and then looking up the address for that domain name, if they are the same, then the lookup is considered confirmed. Today, most mail transfer agents (MTA) rely on confirming reverse DNS lookups to reduce spam and might reject or bounce incoming mail if the lookup is not forward-confirmed [28]. Consequently, network operators are essentially forced to deploy reverse zones to not degrade the quality of service for the hosts in their network. In practice, reverse zones are regularly populated automatically via DHCP and IPv6 node information queries and the reverse zone information accurately represents an active part of the network [29–31].

Due to DNS' inherent hierarchical design and the IPv6 address space being split into a significant number of sub-networks, it is not possible to simply download the entire reverse zone for IPv6 to enumerate hosts. In fact, the sub-networks are delegated to thousands of different nameservers worldwide, which do not allow to download the respective reverse zones directly. Hence, it motivates the need for an effective IPv6 address enumeration technique.

```
$TTL 1h
@   IN SOA ns1.a.edu. admin.a.edu. (
    2018010101    ; serial
    1h 15m 1w 1h) ; refresh retry copy cache

@   IN  NS      ns1.a.edu.

; IPv6 PTR Entries
2.0.0.0.e.f.a.c.d.e.e.f.d.0.0.f.d.a.b.0.0.0.0.0.0.8.
    ⇝ b.d.0.1.0.0.2.ip6.arpa.  IN  PTR  h.a.edu.
9.0.0.0.e.f.a.c.d.e.e.f.d.0.0.f.d.a.b.0.0.0.0.0.0.8.
    ⇝ b.d.0.1.0.0.2.ip6.arpa.  IN  PTR  s.a.edu.
```

Listing 1: **Example IPv6 reverse zone of 2001:db8::/32.**

Fortunately, the IPv6 reverse zone (`ip6.arpa`) supports DNSSEC since April 2010, which enables our enumeration approach if the respective delegate reverse zones are also DNSSEC-signed. Currently, as of January 2018, already 51 out of 59 delegate IPv6 reverse zones (i.e., zones below `ip6.arpa`) are signed via DNSSEC [32], and, thus, this allows our approach to enumerate IPv6 hosts within those zones, i.e., within those networks. Interestingly, the (still) unsigned reverse zones include the 6-to-4 zone (`2002::/16`), which is an IPv6 transition mechanism and which can be enumerated through traditional IPv4 enumeration techniques.

III. Approach

Following, we describe our approach, which enumerates active IPv6 addresses by walking an IPv6 network's reverse zone. The network can be the entire IPv6 address space or any sub-network that might be of particular interest, for example as part of a security evaluation. Consequently, our approach can be targeted and can be faster than state-of-the-art techniques.

Our enumeration technique requires that the reverse zone for the network is signed via DNSSEC, because it relies on `NSEC` or `NSEC3` responses for non-existing addresses. Nevertheless, it is already practical because over 86% of the top-level delegations in the IPv6 reverse zone are already DNSSEC-signed and it is expected that all zones will support DNSSEC soon [32]. In fact, NIST recommends deploying DNSSEC since September 2013 [33] and adoption has been ever increasing since then [34]. If the records are not signed yet, for example because a large network is partitioned into smaller networks and only some of the zones employ DNSSEC, then we can still enumerate the hosts within networks for which the reverse zones are signed (regardless of whether intermediate zones are signed).

A fundamental difference of our approach to existing techniques that determine an active part of the IPv6 address space through network vantage points or datasets is that our approach can enumerate hosts that do not actively initiate connections, nor does it require that IPv6 addresses appear in a forward zone. At the same time, conventional "brute-force" enumeration attacks known from IPv4 [7] do not scale to the vast IPv6 address space, while our approach can enumerate sparsely populated IPv6 networks without problems.

A. Reverse Zones with NSEC

It is an understood problem that `NSEC` denials of existence allow zone-walking attacks on signed zones because they leak the previous and next existing name of that zone. In case of the IPv6 reverse zone, those leaks correspond to the previous and next IPv6 name pointer (`PTR`) for an address in that reverse zone, or a nameserver (`NS`), if a subdomain (sub-network) is delegated to another nameserver [35]. We modify the existing `NSEC`-based approach and exploit the organization of the IPv6 reverse zone to enumerate addresses more efficiently.

Starting from a target IPv6 reverse zone, e.g., the root zone for the entire IPv6 address space, the steps to enumerate the reverse zone for `NSEC`-based denials of existence records are:

1) *Bootstrapping*: We query for a random string below the target zone, e.g., `foobar.ip6.arpa`, to determine a starting point for address enumeration (seed). Based on the organization of the IPv6 reverse zone (as specified by RFC 5855 [36]), it is guaranteed that a random string that is not a single hexadecimal digit will result in a `NSEC` response. In turn, it removes the requirement to identify a non-existing address in the address space prior enumeration.
2) *Zone Walking*: Starting from the seed, we follow the chain by iteratively querying the next addresses incremented by one, i.e., the next address that might not exist and could yield a denial of existence.

   If we do not receive a `NSEC` response, then we discovered an active address and we keep incrementing the address until we receive a `NSEC` response. Once we receive a `NSEC` response, based on the organization of the IPv6 reverse zone, we can immediately identify if the next entry of a `NSEC` record is an address or a sub-network: if it is not a full-length IPv6 address (32 nibbles), then this sub-part of the reverse zone is delegated, possibly to another DNS server.

   If we encounter a zone delegation, we optionally identify via a random seed whether it is signed at all, and if so, if we can immediately descend into it (`NSEC`) or if it requires further processing (`NSEC3`). If we can descend into it, we optionally add it to a sub-zone queue (i.e., we perform breadth-first search).

   We terminate the zone-walking step if the next address in the returned `NSEC` record points to the seed (we have closed the chain and formed a circle).
3) *Sub-zone Enumeration (optional)*: For each sub-zone that we added to our queue, we may descend into it and recursively apply the same enumeration strategy.

Intuitively, the runtime of our approach to enumerate IPv6 addresses for `NSEC`-based reverse zones is linear and requires $O(n + m)$ DNS queries to nameservers for the reverse zone where n is the number of addresses within the networks and m is the number of zone delegations.

B. Reverse Zones with NSEC3

In an attempt to mitigate the side effect of zone-walking attacks on DNSSEC-signed zones, Laurie et al. proposed `NSEC3` [35]. Instead of listing the previous and next existing name in clear, `NSEC3` uses a cryptographic hash for the names in the zone, sorts the hash values in alphabetical order, and then uses each pair of consecutive hash values in the zone to indicate the denials of existence through a `NSEC3` record.

If the zone is using `NSEC3`, then the nameserver responds to a query for a non-existing name n as follows: it computes its hash value $h(n)$ where h is the cryptographic hash function as specified for the zone, and it then returns the `NSEC3` record with the pre-computed hashes of the existing names n_1 and n_2, such that $h(n_1) < h(n) < h(n_2)$. Note that $n_1 < n < n_2$ does generally not hold because h is not order-preserving. In fact, since the names are ordered by their hash value, and since h is not order-preserving, only the cryptographic hashes of two existing names are exposed, which are considered computationally difficult to reverse.

Given a `NSEC3` response, the client can verify herself that the name does indeed not exist in the zone. She verifies that the `NSEC3` response is authentic and then verifies that the queried name, when hashed, falls into the range specified by the `NSEC3` record. To hash the queried name, she uses the parameters specified in the authenticated `NSEC3` record, i.e., hash algorithm (only SHA1 is currently supported), salt, and the number of iterations, which are valid for the entire zone.

Nevertheless, `NSEC3` records still leak two existing records from the zone, even though their names are cryptographically hashed. Therefore, they are technically still vulnerable to zone enumeration through brute-force and dictionary attacks [37, 38]. In fact, the attacks identified by prior work inspired our research. However, existing approaches for forward zones are

ineffective for the IPv6 reverse zone because of the reverse zone's organization: *(i)* existing dictionary attacks, such as `nsec3walker`, are inefficient due to the small alphabet (0-f, one character maximum) and the large height of the zone's hierarchical tree; and, *(ii)* uninformed brute-force attacks are computationally expensive and considerable computational resources are required to successfully launch them, particularly considering the size of the IPv6 address space. Following, for our case, we show the contrary: enumerating IPv6 addresses for `NSEC3`-protected reverse zones is practical and effectively computationally less complex than uninformed brute-force attacks.

Different from `NSEC`-based address enumeration, `NSEC3` requires a two-phased approach. First, we need to collect the `NSEC3` chain for a zone online by actively querying for names. Subsequently, we can unblind the IPv6 addresses offline. Note that the first phase does not necessarily have to be completed before we can launch the second phase. We can launch the second phase as early as the first `NSEC3` record is being observed, which can reduce the time required to enumerate the target network's addresses significantly. Furthermore, even though a network operator could change hash parameters during the collection phase, such as the salt or the iteration count, previously collected `NSEC3` records can still be unblinded and used to enumerate hosts within the zone. Following, we discuss how our approach can efficiently unblind `NSEC3`-protected IPv6 addresses in the reverse zone by exploiting intricate details of the specification and implementation of the IPv6 reverse zone.

C. Online Collection

The design of `NSEC3` makes it computationally impractical to follow its chain to find the next hash. Instead, the core idea is to randomly query for names that do not exist until the full `NSEC3` chain has been recovered. Similar to the `NSEC` case, a complete chain of `NSEC3` records forms a closed circle and, thus, can be verified easily. During the sampling process, any not-yet-discovered `NSEC3` records leave missing "gaps" on the circle. Eventually, the sampling process will fill all gaps (Figure 2). The problem of discovering names whose hashes are inside one of the remaining gaps is similarly embarrassingly parallel as the offline unblinding step and can easily be sped up massively through graphical processing units.

For `NSEC3`-based reverse zones, online collection works as follows:

1) *Bootstrapping*: We query for a random string below the target zone, e.g., `foobar.ip6.arpa`, to determine a starting point for online collection. As in the case for `NSEC`, it is guaranteed that a random string that is not a single hexadecimal digit will result in a `NSEC3` response and it removes the requirement to identify a non-existing address in the address space prior enumeration.

   In addition, we are also interested in the current hash algorithm, salt, and iteration count to fill hash gaps locally as to not query the nameserver unnecessarily or cause suspicion or incur unnecessary load.

2) *Zone-Walking*: We calculate the hash value for a random name under the zone based on salt and iteration count. If the hash value is covered already by a range uncovered from the previously collected `NSEC3` records, then we repeatedly select random names until a hash falls into a gap and is guaranteed to reveal more information about the `NSEC3` chain (Figure 2).

   Intuitively, with the number of hash gaps decreasing, the probability to hit one of the remaining ones decreases too, and the time requirement increases. The average number of required hash calculations is $O(r \log r)$ with r being the number of records in the zone (addresses plus delegated sub-zones).

   Already during the collection phase we can determine whether a hash is a full IPv6 address or a zone delegation: a `NSEC3` record leaks whether the next hashed value is a `PTR` record (full IPv6 address) or a `NS` record (sub-zone delegation) (Listing 2). In fact, this detail allows us to separate addresses and networks into different buckets and unblind them separately later, which reduces computational cost significantly.

   We retain all `NSEC3` records for offline unblinding.

   We repeat the zone-walking step until no more hash gaps exist or in case an exit condition is true, in which case parts of the address space remain unexplored. If we have filled all hash gaps within the `NSEC3` circle, we have successfully collected all hashed IPv6 addresses and sub-zone prefixes.

The runtime of the online collection phase is $O(n + m)$ DNS queries to the nameservers where n is the number of addresses within the target network and m is the number of sub-zone delegations.

To probabilistically enumerate addresses within a zone, one may specify an exit condition that terminates the zone walking step. A trivial condition might be a timeout during which a new gap must be filled. However, a more intelligent solution is to fill in all gaps until at most x gaps of at most size y exist. At that point, at most $x \times y$ hashes of the entire zone will not be collected through our approach (effectively, missing at most $x \times y$ addresses or sub-zones). Here, x and y can be chosen to specific probabilistic requirements, such as "at least 95% of the zone must be enumerated." Additionally, if hashes within those ranges are later discovered during unblinding, the gaps can be filled.

D. Offline Unblinding

Following online collection, the next step to enumerate IPv6 addresses is to unblind the collected hashes offline. Since DNSSEC leverages cryptographically secure hashes, the naïve choice falls to brute-force attacks. Brute-force attacks, however, are impractical because of the large search space for SHA1, which is the only supported hash of DNSSEC, at 2^{160} possible values.

Generally, domain names can be composed of letters, digits, and hyphens [39]. The IPv6 reverse zone, however, follows a well-defined structure: each subdomain is strictly a hexadecimal digit (Section II-B). Practically, by leveraging the organization of the IPv6 reverse zone, we can unblind hashed IPv6 addresses (which we identified as full addresses during online collection) significantly faster through directed search. We exploit the fact that addresses are almost never randomly assigned from a network's range, but instead follow observable

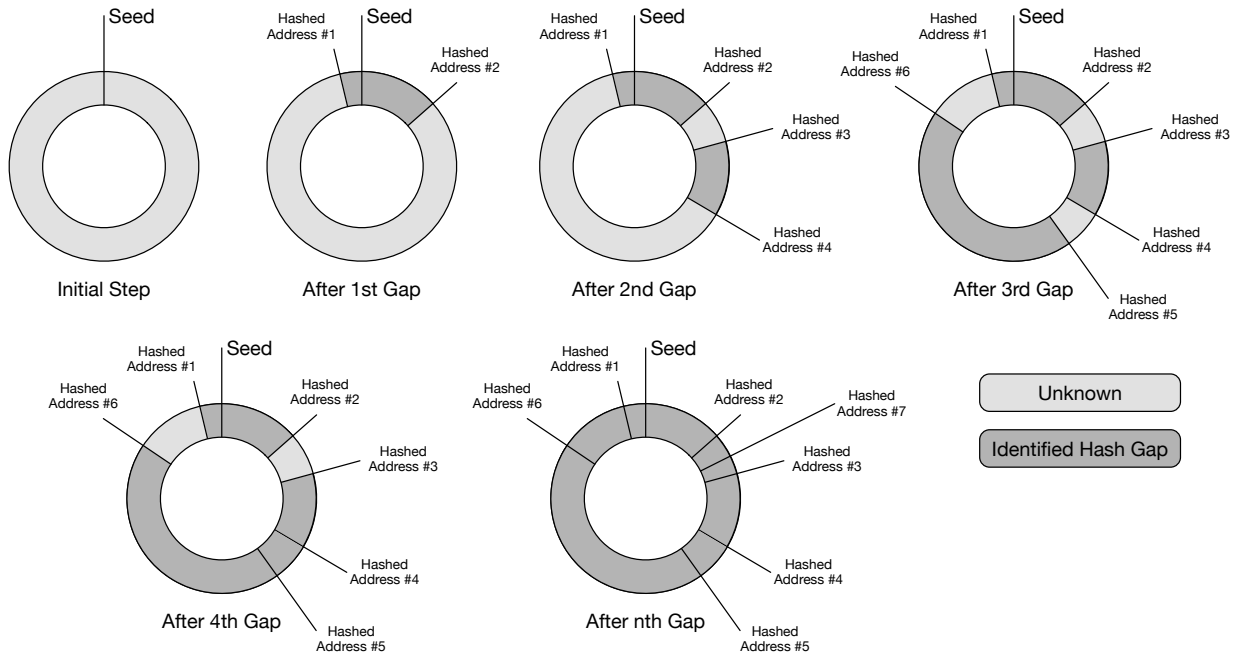

Figure 2: **Online collection and `NSEC3` hash gaps.** During the online collection phase for `NSEC3`-protected zones, we first bootstrap by choosing a random seed that is guaranteed to result in a `NSEC3` response for the zone, which exposes two hashed addresses. Following, we walk the zone randomly and iteratively fill hash gaps to discover more addresses until we have successfully identified all hash gaps.

```
; Reverse IPv6 NSEC Entries
2.0.0.0.e.f.a.c.d.e.e.f.d.0.0.f.d.a.b.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa. IN
    ⇝    NSEC 9.0.0.0.e.f.a.c.d.e.e.f.d.0.0.f.d.a.b.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa. PTR RRSIG

; Reverse IPv6 NSEC3 Entries
1PDJ9FP13S70NCFCJCV35B8LLVT68U5Q.8.b.d.0.1.0.0.2.ip6.arpa. IN
    ⇝    NSEC3 1 0 10 86B3E6B74F0A2C23 G5AL6GMJ6ARLJ9M5F56LL48JPHJ1SGQK PTR RRSIG
```

Listing 2: **Example `NSEC` (top) and `NSEC3` (bottom) records for the reverse IPv6 zone of 2001:db8::/32.** A client querying for a name that is lexically between `2001:db8::bad:f00d:feed:cafe:2` and `2001:db8::bad:f00d:feed:cafe:9` will receive the `NSEC` (top) record from the nameserver. Similarly, for `NSEC3`, if no record exists in the zone whose hash is lexically between `1PDJ9FP13S70NCFCJCV35B8LLVT68U5Q` and `G5AL6GMJ6ARLJ9M5F56LL48JPHJ1SGQK` (base32-encoded SHA1), then the `NSEC3` record will be returned.

patterns. First, addresses are often assigned incrementally through static assignment or via DHCPv6, possibly with gaps at earlier nibbles, such as `2001:db8::1/64`, `2001:db8::2/64`, or, with a gap, `2001:db8::1:1/64`. Second, addresses are also more likely to be assigned through stateless address autoconfiguration (SLAAC) than being randomly picked. With SLAAC, a host commonly assigns itself an IPv6 address based on its MAC address, in which case 12 nibbles (out of 32 nibbles) of the IPv6 address are based on the host's MAC address, which is vendor-based, and additional 4 nibbles are constant across all IPs assigned through SLAAC. For example, a host with MAC address `00:11:22:33:44:55` on the network `2001:db8::/32` would assign itself the IPv6 address `2001:db8::211:22ff:fe33:4455`. As of January 2018, only 24,434 vendor prefixes are officially in use [40], and combined with the constant nibbles, it reduces the search space by a factor of 2^{25}. Inherently, a MAC-based address assignment strategy allows Internet-wide equipment and user tracking, because the MAC is considered universally unique and remains constant across networks. To prevent such tracking, privacy extensions were added to SLAAC, for which temporary addresses may be used instead. These privacy extensions make the enumeration attack more difficult initially due to the addresses' ephemeral nature, however, their effectiveness degrades over time since addresses are generally not reused. Furthermore, their use is commonly limited to end users and they are not used by servers or network equipment.

Overall, we can reduce the search space from 2^{128} to as little as 2^{39} for full IPv6 addresses (although the SHA1 search space is 2^{160}, it is reduced to 2^{128} because IPv6 addresses are only 128-bit wide) depending on network prefix and address assignment strategies used. By guiding the address search intelligently, we can further speed up the unblinding process. Specifically, we can exploit that a hash gap (pair of `NSEC3` records) leaks the type of the preceding and following resource record. The type of the resource record indicates the length of the unhashed value (PTR for full addresses, NS and SOA for network prefixes), which, in turn, significantly reduces the complexity of unblinding the hashed value. Practically, we can reduce the search space down to as little as 2^{39} for full addresses and 2^{33} for networks, which renders enumeration practical. Notably, we successfully unblinded various networks of different sizes (/32, /48, and /64) in mere hours, including for reverse zones with high hash iteration count (Section V).

Unblinding zone delegations is practical for similar reasons: First, we accurately identify them as delegated zones during online collection (since the NSEC3 record leaks whether the next hash is a PTR or NS record). Second, we exploit that sub-networks, a common cause for sub-zones being delegated, are commonly assigned and used incrementally rather than randomly from the vast address space. Third, we exploit that networks are allocated at specific nibble boundaries, effectively limiting the search space to $\sum_{0 \leq i \leq 8} 2^{4i}$ ($\leq 2^{33}$).

For example, for the hashes `g5al6gmj6arlj9m5f56ll48jphj1sgqk` and `1pdj9fp13s70ncfcjcv35b8llvt68u5q` (Listing 2), we only need to attempt to unblind full addresses as they are PTR records. Combined with the salt `86b3e6b74f0a2c23`, we can then unblind the hashes to `2.0.0.0.e.f.a.c.d.e.e.f.d.0.0.f.d.a.b.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa` and `9.0.0.0.e.f.a.c.d.e.e.f.d.0.0.f.d.a.b.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa`, i.e., they represent the IPv6 addresses `2001:db8:bad:f00d:feed:cafe:2` and `2001:db8:bad:f00d:feed:cafe:9` respectively.

In summary, our approach can quickly enumerate active IPv6 hosts and networks, even for sparsely populated IPv6 networks, by exploiting the well-defined organization of IPv6 addresses and networks, and by leveraging the structure of the IPv6 reverse zone and the information (record type) leakage of DNSSEC-based denial of existence records (NSEC3).

IV. Ethical Considerations

In our evaluation, we perform active measurements on the enumerated addresses to establish if they are actually active. We also establish a limited set of additional data points on the running software versions and possibly security-sensitive configuration settings. For our data acquisition, we adopt the high and well-accepted ethical standards of prior work conducting Internet-wide active measurements [7, 18, 41]. We further ensured that our measurements do not disrupt or harm evaluation targets, e.g., through unintended resource or bandwidth consumption, and we put a process for a permanent opt-out of our measurements in place.

A. Preventing Disruption

In addition to standard ICMPv6 (Internet Control Message Protocol version 6) echo request to establish host reachability, we performed only basic service and version detection on open service ports. Misconfigurations, such as weak cryptographic keys, were only evaluated based on protocol handshake information. Similar to prior work, our independent evaluation of this measurement procedure yields that it is of negligible risk compared to the benefits provided to the community. This approach prevents misleading findings and reduces false positives, which would cast an incorrectly insecure picture of the evaluated hosts. Examples of such false positives are services listening on non-standard ports or services secured via tcpwrapper, which would also not be detected correctly by a standard port scan.

B. Subject Information and Opt-Out

A network administrator might misjudge our measurements for attacks, due to receiving alerts from an intrusion detection system deployed at the evaluated network. To inform the operators of the measured networks, we follow best practices [7] and provide a "usage notice" website reachable at both the IPv4 and IPv6 addresses of the measurement machine. The notice explains that the measurements are benign in nature, who is conducting them, how to contact the authors, and how to opt out of future measurements. We have not received any opt-out requests or related complaints.

C. Responsible Disclosure

We encountered several vulnerable systems and deployments during our evaluation. With the publication of our methodology, an attacker could use it to enumerate active IPv6 addresses and rediscover vulnerable devices and infrastructure. Therefore, we conducted a responsible disclosure process for our findings, having informed the affected parties. To prevent any possible harm, we contacted the individual parties and the responsible Computer Security Incident Response Teams (CSIRT). The responsible disclosure process has been completed for all our findings.

V. Evaluation

We first evaluate how our technique fares on an Internet-scale. We then look in-depth at various issues IPv6 networks exhibit in the wild, which prior studies have missed, possibly due to being unable to target and enumerate specific IPv6 networks or IPv6-only hosts. Our results underline the need for an active enumeration technique for future IPv6 security studies, instead of being able to rely on data collected at network vantage points.

A. Internet-wide Enumeration

First, we enumerate the entire IPv6 address space using our technique. To enumerate the address space more quickly, we seed our enumeration technique with IPv6 network prefixes that we obtained from aggregating a view on the global routing table (GRT). We aggregate this GRT from Border Gateway Protocol (BGP) dumps available from RIPE RIS [42] and Routeviews [43] following current best practices [44]. In addition, we leverage the enumeration technique of Fiebig et al. to establish a baseline [18].

We find that our technique performs favorably compared to the enumeration technique of Fiebig et al. (Figure 3). Specifically, we perform better than the baseline for large prefixes. For instance, for network prefixes of size /32, the maximum allocation size for IPv6, we identify 3,770 more networks, while for networks of size /48, the general allocation size for IPv6, we find 2,649 more networks [45, 46]. Unfortunately, however, due to the delayed deployment of DNSSEC, our technique currently enumerates fewer different prefixes than Fiebig et al. for more specific nibbles in IPv6 addresses. We expect this behavior to change in the near future as the adoption of DNSSEC is increasing, which, in turn, allows our technique to enumerate even more addresses.

Interestingly, during our study we encounter 316 networks using DNSSEC that have an untrusted path from the root zone. In detail, of these 316 networks, 191 utilize NSEC and 125 have NSEC3 configured. This observation underlines that DNSSEC and DNS zones are not necessarily configured correctly in practice. Following the hierarchical concept of DNSSEC, there

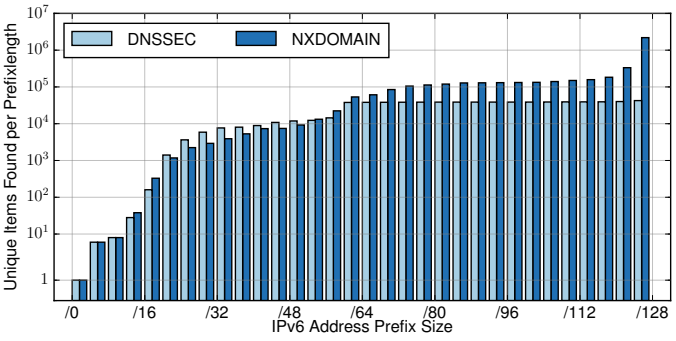

Figure 3: **Records enumerated by our DNSSEC-based technique and the technique by Fiebig et al. [18].** Applied on a global scale, we identify more unique prefixes than the technique by Fiebig et al. for prefix lengths between `/20` and `/56`, e.g., 3,395 more networks with a prefix length of at least `/44`. For networks smaller than a prefix of /60, the number of discovered prefixes increases more slowly, because DNSSEC is not yet being frequently deployed at smaller leaf networks (compared to its wide-spread adoption for zones at the higher level). The deployment of DNSSEC for these smaller networks is expected to increase in the near future.

should not be a zone that is DNSSEC-signed that was not found by enumerating from the reverse zone root (ip6.arpa). DNS is strongly hierarchical by design, and following the tree-based key distribution and verification schema of DNSSEC, there should be no signed zone that is only reachable through intermediate unsigned zones. It further indicates that the seed-based approach we utilized not only reduces the overall runtime, but also discovers networks and enumerates hosts that would otherwise not be found by naïvely enumerating the reverse zone in a top-down fashion. In practice, the time to unblind IPv6 addresses is reduced further by exploiting the knowledge of total addresses in a reverse zone (the number of hashes that we collected online prior to unblinding) and address assignment strategies. Addresses are rarely assigned randomly, but instead follow incremental strategies or use stateless address auto-configuration, and allow us to direct our unblinding process in the search space and reduce its time further.

We also find less records than Fiebig et al. While they enumerated 5.8M unique addresses using their technique in late September 2016 [18], and—with an improved version running on multiple hosts at the same time—over 10M in early 2017 [31], we merely found 2.2M addresses running the published toolchain on a single host (compared to 5.8M). We mostly attribute this to the significantly higher number of necessary requests of their approach compared to our more informed enumeration technique (Figure 2), leading to an increased impact of packet loss. Indeed, especially due to the higher number of requests, their technique can be detected and selectively mitigated with relative ease. Hence, our technique does not only provide more reliability by being harder to mitigate, but also puts less stress on networks. Both features are desirable when conducting large-scale active measurements.

In summary, we find that our technique shows great promise. We easily out-perform existing techniques for zones that are already DNSSEC signed. Our technique is only hampered by the current deployment state of DNSSEC for leaf zones. However, the adoption of DNSSEC is expected to increase even further in the future, as is the adoption of IPv6. Therefore, we expect our approach will be able to enumerate significantly more networks in the near future.

B. Observed Security Issues

Following the demonstration of the large-scale potential of our technique, we utilize it to survey network security issues in current IPv6 deployments around the globe. Specifically, we have scanned 338 different IPv6 networks and we report detailed findings of the security posture of five different networks with different and diverse security requirements: *(i)* a French Internet service provider (ISP), *(ii)* a Ukrainian Local Internet Registry (LIR) and transfer broker (responsible to facilitate IP address space transfers), *(iii)* a European domain registry, *(iv)* a supercomputing facility in the United States, and *(v)* a large German university. The security issues we have uncovered in these networks illustrate that even experienced network operators from a variety of backgrounds might be unaware of the problems that a hasty IPv6 deployment can bring.

For each identified network of hosts, we perform the following two steps:

1) We look up the hostnames for the enumerated IPv6 addresses within the reverse zone, and then forward look up the hostnames to obtain the corresponding IPv4 addresses. If a hostname maps to a single IPv4 address, then we assume that IPv4 and IPv6 address point to the same physical host and compare open ports and available services through access via IPv4 and IPv6, respectively. If a hostname maps to multiple IPv4 addresses, we do not further evaluate its security as it would skew the comparative analysis because it is uncertain which IPv6 and IPv4 addresses correspond to each other.
2) We evaluate the enumerate hosts with `nmap` and we specify the command-line arguments *-Pn -O -sV –nsock-engine=epool -p1-10000 -sS -sU –max-retries 1* to identify potential security issues.

We responsibly disclosed our findings to the network operators for all networks that we have evaluated in the course of this paper. We hope that our findings motivate network operators to evaluate the security of IPv6-connected devices on their network.

The different networks that we investigated in-depth vary in the way they deploy DNSSEC: one network deploys `NSEC3` and the remaining four deploy `NSEC`. They also differ in size as the number of active hosts ranges from 235 to 70,818, with between 28 and 4,619 hosts classified as IPv6-only. We classify a host as IPv6-only if we were unable to confirm that its hostname, which we obtained from the reverse IPv6 zone, points to exactly a single IPv4 address in the forward zone.

An IPv6 network might be split into various sub-networks for specific purposes or regions. Unsurprisingly, the number of sub-networks differs quite a lot per network type: the French Internet service provider's network has 43 sub-networks, which likely correspond to different regions where they provide their services; the Ukrainian LIR delegates the most networks (611), most likely to its customers, some of which are government and law enforcement entities; the European domain registry and the German university do not have any sub-networks, possibly because of a central network operations center; and the United States supercomputing facility uses one sub-network, possibly for users of the computing cluster or the cluster itself. While we did not include the sub-networks

Network	Hosts			Sub-Networks
	IPv6-only	Dual-Stack	Total	
French Internet Service Provider	2,069	66,545	70,818[*]	43
Ukrainian LIR	4,619	245[†]	4,864	611
European Domain Registry	130	119[‡]	249	0
United States Supercomputing Facility	28	1,343	1,371	1
German University	138	97[§]	235	0

Table II: **Number of IPv6 hosts enumerated and sub-networks identified for in-depth analysis.**

[*] We successfully unblinded 68,614 addresses within our timeout of 12 hours. Only 2,204 hosts remain blinded, or a 96.90% success rate.
[†] Two (2) hosts leak private IPv4 addresses via forward DNS lookups from two networks, and two (2) hosts point to IPv4 localhost addresses.
[‡] Five (5) hosts leak private IPv4 addresses via forward DNS lookups. [§] Sixteen (16) hosts leak private IPv4 addresses via forward DNS lookups.

in our evaluation, our technique can enumerate them readily as they are also DNSSEC-signed.

We are focusing our efforts on the following problems and discuss them separately: *(i)* for IPv4 and IPv6 dual-stack hosts, we look at all ports accessible via IPv6 but not via IPv4 and vice-versa; *(ii)* for IPv6-only hosts, we look at all services that can be accessed externally and which could be a security risk; *(iii)* potential privacy concerns for names in the reverse zone. Particularly, we investigate more closely:

- **Remote access protocols:** Secure Shell (SSH), Telnet, and remote desktop sharing.
- **File sharing:** Apple Filing Protocol (AFP, Apple macOS), FTP, HTTP, Server Message Block/Common Internet File system (SMB/CIFS, Microsoft Windows), and WebDAV.
- **Monitoring and system management:** Nagios Remote Plugin Executor (NRPE), Simple Network Management Protocol (SNMP), Intelligent Platform Management Interface (IPMI), and management interfaces for machine virtualization (Hyper-V, VMware).
- **Network management via routing protocols:** Open Shortest Path First (OSPF) as an interior gateway protocol, and the Border Gateway Protocol (both iBGP and BGP).

C. Dual-Stack Analysis: IPv4 vs. IPv6

We contrast the security deployment of IPv4 and IPv6 by taking an in-depth look into some existing networks. In total, we investigate more closely the differences in security measures of accessing 68,349 hosts through IPv6 compared to through IPv4. The hosts are part of the networks of five different institutions with varying security requirements.

The infrastructure network of the French Internet service provider is the most populous network, of the ones we have investigated more closely, with 66,545 dual-stack hosts. Fortunately, most hosts are secured appropriately. In fact, much to our surprise, hosts following incremental IPv6 address assignment pattern exhibit the same or better security, i.e., the same or less exposed ports through IPv6 than via IPv4. This might be the case because the services are configured to listen on their respective IPv4 address only, instead of the default to listen on all available addresses (IPv4 and IPv6) or interfaces, and, thus, no access via IPv6 is possible. On the other hand, hosts who have taken on globally routable addresses via stateless address autoconfiguration (SLAAC) do exhibit worse security. Alongside world-readable Apple file sharing we discovered open ports for access to management interfaces of Cisco switches via Telnet, access to Hewlett Packard StoreFabric network storage devices (both client and management interface ports), as well as read-only SNMP access for various networking devices (access might not be restricted to read-only, but without potentially disrupting infrastructure, we are unable to confirm whether access is read-write; therefore, we report all SNMP access as read-only).

Different is the network of the supercomputing facility in the United States, for which we enumerated 1,371 IPv6-capable hosts, with 1,343 of them being dual-stack. Although a significant amount of services, like HTTP(s), FTP(s), IMAP(s), SMTP(s), POP3(s), are available on the network, almost all of them are accessible via IPv4 and IPv6 and we consider them as intentionally open and without additional security risk. Of all 1,371 hosts, 828 hosts assigned themselves IPv6 addresses through SLAAC, while the remaining 543 hosts have IPv6 addresses assigned incrementally with gaps due to jumps at earlier nibble boundaries, confirming that guided search for enumeration has substantial benefits. There was no difference in security for incrementally assigned addresses and automatically assigned address through SLAAC, but hosts remained more open to attackers via IPv6 than IPv4. Specifically, we still encountered services accessible via IPv6 that are likely unintentionally accessible as they are security-sensitive, including, but not limited, to BGP (secured via tcpwrapper for some hosts only), Telnet access to Cisco routers, and access to Microsoft's Active Directory.

Similar to the supercomputing facility, a variety of IPv6 hosts on the German University's network expose SSH, HTTP, and FTP. Again, we observed the same ports being publicly accessible via IPv4. Since universities often provide HTTP and FTP mirrors of open-source software, and SSH is generally considered secure, we do not consider them potential security problems. Alarmingly, however, we still determined a plethora of potentially critical security problems. In particular, publicly accessible via IPv6 but not IPv4 are: interior BGP and exterior BGP for 57 hosts, old SSH versions on 2 Cisco switches, SNMP on 35 hosts, Nagios Remote Plugin Executor for 38 hosts, a portmap version on 38 hosts that can be exploited to launch reflected and amplified denial of service attacks [47], and fingerd on one host. Especially concerning are the exposure of BGP, portmap, and SSH access on the two Cisco switches, which used weak host keys (512-bit RSA).

We observed no significant differences in security for dual-stack hosts for the European domain registry or the Ukrainian LIR. Yet, over all networks, the security of hosts

whose addresses appear assigned through SLAAC, i.e., automatically based on the hosts' MAC addresses, is worse than those for which the address is assigned incrementally.

D. Security Posture of IPv6-only Hosts

We also enumerated hosts that are single-stack and thus are only reachable via IPv6. Interestingly, some early proponents of IPv6 without prior experience operating IPv4 networks exhibited the worst security measures and exposed administrative, infrastructure, and network management interfaces through IPv6 to the world. Most likely, they assume more secure defaults and might not know better given a lack of experience.

Unfortunately, although experience helps to mitigate some issues, it is not a silver bullet. An example is the infrastructure network of a major LIR in the Ukraine of which almost all hosts (4,619 of 4,864 hosts) are reachable only through IPv6. However, since the network operator has extensive experience operating an IPv4 network, we were expecting a relatively secure network. Regardless of prior operating experience, we discovered critical security issues on two IPv6-only hosts, both of which do not have an entry in the forward zone. Both hosts expose the Quagga routing software's management port as well as BGP via IPv6 and could be used to control routing for all of the LIR's sub-networks, which include law enforcement and government entities. Although already concerning, we detected an old version of Quagga (0.99.22.1) at a core network router, which is potentially vulnerable to a remote code execution and a denial-of-service attack [48, 49]. Unfortunately, the critical security issues did not stop there, and, even more alarming, we discovered a vulnerable version of SuperMicro IPMI at an IPv6 address that was assigned automatically (via stateless address autoconfiguration), which not only allows full remote execution, but it allows an attacker to gain practically physical access to the machine remotely.

We manually confirmed that all vulnerable hosts were not part of any public dataset used by Czyz et al. [13], which further emphasizes the need for practical IPv6 address enumeration techniques, and it illustrates that existing datasets might in fact cast a skewed result on the security state of IPv6-connected devices. Considering that Czyz et al. collected their dataset from `ANY` records on the forward zone, it is clear why prior work did not include it: the hosts' IPv6 addresses do not appear in the forward zone at all, but only appear in the reverse zone.

As in the case for dual-stack hosts, we reach the conclusion that the security posture of IPv6-only hosts varies in the way addresses are assigned. For devices who leverage SLAAC security is worse than for those who have addresses assigned manually or via DHCPv6.

E. Privacy Issues

A possible security and fundamental privacy issue we discovered is the leakage of meaningful hostnames through the automatic population of the reverse zone.

In case of the European NIC, regardless of the deployed security measures at those hosts, the respective hostnames leaked information about their use case: configuration management and deployment, system and network monitoring, logging, version control, bug tracking, as well as registry internal infrastructure (authentication, transfers, validation). Although not a security issue necessarily, it opens an avenue for reconnaissance for attackers and it might provide the extra information that is necessary to circumvent security measures that have been put in place.

Similarly, for the French ISP, stateless autoconfigured IPv6 addresses leaked that Apple, Cisco, and Hewlett Packard devices are on the network. From reverse DNS entries, we further determined that the Apple devices are laptops and based on a combination of reverse DNS, MAC address, and service and version detection on open ports, we can determine that the Hewlett Packard devices are HP StoreFabric storage devices, while the Cisco devices are top-of-rack switches. Additionally, based on hostnames themselves and routes taken to hosts, we believe that we have enumerated hosts in four datacenters or office buildings: two in Paris, one in Lyon, and one in Toulouse.

Significantly more concerning is the case of the United States supercomputing facility though. The way the reverse zone is used and populated allows us to track employees' devices and even their location. Specifically, we were able to track 13 phones and 10 laptops of employees over time and we correlated their working hours, and their presence across two buildings. Of the ten laptops, three laptops are connected via Ethernet and Wi-Fi, allowing higher fidelity tracking, and one person is using two laptops. From reverse zone information, we can also determine that four people work in the main complex, while another nine work in an adjacent and affiliated research center. We manually verified this to be true through its website.

Tracking is made possible due to the automatic populating of the reverse zone. To track working hours, regular liveness probes are sufficient (e.g., via ICMPv6). On the other hand, tracking users across buildings is possible in two different ways. First, through liveness probes over multiple network prefixes, since the remaining nibbles of the address stay constant (due to SLAAC), and, second, through forward DNS lookups on the hostname under a different subdomain (the subdomain used for Wi-Fi access in the buildings is different). More fine-grained location tracking, up to floors and even rooms, is sometimes possible through tracing the route to the host and investigating intermediate router hostnames more closely. The privacy implications of automatically populating the reverse zone are further amplified by host and node information, such as names in "jane-iphone" or "doe-notebook" (with only one person with the first name Jane or last name Doe working at the facility).

F. Discussion

From our evaluation it is apparent that IPv6 hosts can be, and sometimes are, secured in the same manner and to the same level as IPv4 hosts. However, as of today, IPv6-connected hosts still lag behind in regard of security when compared to IPv4 hosts, and their improvement progress must be monitored and evaluated closely as to not relive the "Wild West" days of the Internet from the 1990s.

Furthermore, we discovered that stateless address configuration can be a significant security problem if network-based firewalls are not deployed. Our findings show that devices take on global IPv6 addresses automatically if they are advertised an IPv6 route, regardless of whether they are secured appropriately. Since some networks are secured appropriately and since the self-assigned IPv6 addresses do not fit into

the networks' address assignment pattern, we suspect that the devices with self-assigned addresses have worse security because the network operators are unaware of their behavior and might assume that they do not support IPv6 yet, possibly because support might have been added with a software update after deployment. We believe that we encountered these cases because IPv6 is sometimes enabled by default in newer firmware versions of switches and routers, which might be installed for part of a datacenter only, e.g., through a staggered deployment, and because laptops might normally connect to IPv4-only networks exclusively, but sometimes connect to a network where an IPv6 route is advertised. For them, host-based firewall rules might not be configured for IPv6 yet, thus exposing the machine completely to the rest of the Internet.

VI. Mitigation

In response to zone-walking attacks against DNSSEC, a variety of defenses have been proposed. Some of these approaches would also prevent enumerating IPv6 addresses from the reverse zone. However, the proposed defenses have significant shortcomings and some require to fully trust the nameserver with the authoritative zone-signing keys, a practice that DNSSEC strongly discourages. We discuss how those techniques would impact our approach and, if adopted, what other issues they bear.

A. Reverse Zone Modifications

A straightforward solution to prevent IPv6 addresses from being enumerated via DNSSEC on the reverse zone is to drop the reverse zone completely or to not deploy DNSSEC on it. Not keeping any reverse zone information for IPv6 addresses has significant problems though, which would render the affected IPv6 addresses almost entirely useless in practice. Nowadays, reverse zones are used to protect against spam and other inconveniences and the lack of a reverse entry for an address is considered a lack of trust and "sign of trouble." For instance, almost all incoming email servers (SMTP) are configured to look up the reverse name and reject incoming mail from IP addresses that do not hold a valid reverse DNS record. Therefore, not keeping a specific IP address in a reverse zone immediately limits the use of that address. For instance, in the case of a hosting or access Internet service provider, it would effectively prevent its customers from sending email.

Alternatively to dropping the reverse zone entirely, one could choose not deploy DNSSEC for it. However, similarly as to verifying that an IP address has a reverse entry, some SMTP servers are trusting signed and valid reverse entries more and service them quicker (e.g., no greylisting). In turn, the decision to not sign the reverse zone can degrade the overall quality of service but it would not prevent the service to be used at all. In addition, this technique exposes the reverse zone to the known problems of DNS that have been solved by DNSSEC. For example, by effectively removing any authenticity on a zone one enables malicious nameservers to return bogus responses (again).

In both cases, the respective authority for the reverse zone needs to decide on the trade-off: whether she prefers to degrade quality of service, or whether she wants to prevent zone-walking and protect the privacy of addresses on her network. It is understandable that network operators prefer to guarantee a high quality of service over preventing zone-walking attacks, particularly considering that IP addresses will become public during communication with other hosts anyways. Thus, hiding them is merely a misguided attempt at security through obscurity. Furthermore, security management of the hosts that could be enumerated is often outside of the responsibilities of the network operator herself (instead, a system administrator is often responsible) while the quality of service is her métier.

B. Minimally Covering NSEC Records

An alternative approach to preventing zone-walking attacks via already existing DNSSEC record types, such as `NSEC3`, was proposed by Weiler et al. [26]. Instead of signing the zone offline and thus, by requirement, introducing large spans for `NSEC3` records, Weiler et al. suggest to sign records online and to return *minimally covering* `NSEC3` *records* on demand. For instance, a minimal covering `NSEC3` record for a non-existing domain n with hashed name h_n would fake the previous existing hash as $h_n - 1$ and next existing hash as $h_n + 1$. For proving the denial of existence for n, it is irrelevant whether $h_n \pm 1$ actually exist, if they do not exist the denial record is considered a "white lie."

Minimally covering `NSEC3` records prevent zone-walking attacks effectively. However, this approach requires online signing and thus requires the full zone-signing secret key to be available at the nameserver. If the zone-signing key is deployed to the authoritative nameservers, then any single compromised authoritative nameserver results in a complete zone compromise, and any bogus and possibly malicious responses can be signed and returned. This would be a direct contradiction to the goals of DNSSEC and its operational practices [50]. Given the computational overhead of online signing DNS responses and its potential security risks, minimally covering `NSEC` records have so far been adopted only hesitantly.

C. NSEC4

A separate attempt to revolutionize DNSSEC's denial of existence records was the proposal of `NSEC4` by Gieben et al. [51]. However, the respective Internet-Draft does not propose any techniques that would prevent zone-walking, and thus cannot be considered a mitigation technique. Instead, it introduces performance optimizations for denials of existence of wildcard records and the opt-out flag. The draft has expired in January 2013 and has not been renewed. The optimizations have been integrated into `NSEC5`.

D. NSEC5

Goldberg et al. [52] introduce `NSEC5` as a solution to provably preventing zone enumeration attacks. The adoption of `NSEC5` would prevent enumeration of active IPv6 addresses through the reverse zone, but, it comes at the significant cost of requiring additional online asymmetric cryptography operations. In fact, the additionally incurred cost for online signing when deploying DNSSEC renders nameservers subject to denial of service attacks and chosen-plaintext attacks [26], which is why it might have been rejected by industry leaders in favor of signing zones offline. Specifically, denial of service attacks due to asymmetric cryptography can be abused in many more ways for DNSSEC over similarly authenticated protocols, like TLS, because it uses UDP for the transport protocol instead

of TCP. The latter are less impacted because they normally do not perform any cryptographic operations prior completion of the TCP handshake, which acts as a way to ensure that the connection between server and client is intended. On the contrary, in the case of DNSSEC, no such protection exists and cryptographic operations must be performed when receiving the first and only packet. Furthermore, it is more prone to abuse because of reflection and spoofed addresses. Nonetheless, we support the authors' effort to have NSEC5 become an Internet standard. The additional computational cost incurred on the nameserver and the increased risk of denial of service attacks might be a reason why the Internet-Draft remains a work in progress, and had to be renewed by the authors prior to expiration five times already [53]. Without sufficient industry interest and without an implementation except for the reference implementation for Knot DNS being available (although NSEC5 solves a known problem and was published in mid 2014 [54], no implementation for the BIND nameserver exists), wide adoption of NSEC5 in the (near) future appears highly unlikely, allowing our approach to be used in practice.

If NSEC5 would be deployed for a zone, an attacker who is trying to enumerate that zone would need to obtain the NSEC5-signing-key. Once the attacker has obtained the key, she can degrade NSEC5's security guarantees to those of NSEC3, walk the zone, and, in turn, enumerate IPv6 addresses.

VII. Related Work

We discuss related work in the areas of Internet-wide security scanning, enumerating active IPv6 addresses, and privacy issues with respect to DNSSEC and zone enumeration.

A. IPv4 Security Scanning

Internet-wide scans have become an important tool for applied security research. They are imperative to identify and understand the impact of new vulnerabilities or common misconfigurations, like Heartbleed or DROWN. Heninger et al. scanned the IPv4 address space for weak cryptographic keys used by TLS and SSH servers [41]. Alarmingly, they discovered shared secret keys due to a lack of entropy during key generation, and they were even able to recover secret keys. Aviram et al. discovered DROWN, a new attack that exploits flaws in SSLv2. To determine its practical impact, they scanned the entire IPv4 address space and identified that 33% of all HTTPS servers were vulnerable [55].

These discoveries have been made possible by various advances around Internet-wide scanning. Heidemann et al. performed one of the first Internet-wide scans by sending ICMP messages to all allocated IPv4 addresses to identify reachable hosts [56]. Although enumerating all reachable hosts took multiple months to complete, the study clearly indicated the potential and benefits of large-scale probing. In 2013, Durumeric et al. developed ZMap [7], a fast scanning tool that can scan the entire IPv4 address space in under 5 minutes given the right conditions. They further discuss guidelines and best practices in using this tool to perform Internet-wide scans. We support their guidelines and took similar precautions to minimize the impact of our measurements.

B. Enumerating/Scanning IPv6 Addresses

While Internet-wide scans have become a common tool in the IPv4 world, measurements for IPv6 are still lagging behind. Specifically, three distinct research directions have been pursued: prefix-based measurements, client-centric vantage point based studies, and, the most neglected, server-centric and security motivated studies.

Monitoring and measuring the IPv6 deployment has been of growing interest ever since the IPv6 standard was introduced. Large service providers and vendors, such as Cisco or Google, have since been tracking the use of IPv6 [10, 20, 57]. Similarly, Dhamdhere et al. analyzed historical BGP data to determine IPv6 deployment at the autonomous system (AS) level, for which they were able to determine that it was lagging behind at edge networks [58]. While some publicly accessible resources exist about the allocated IPv6 prefixes, e.g., prefix assignments from IANA [59], those resources only provide a high-level view and do not allow exact measurements. However, considering that the smallest recommended end-user allocation for IPv6 networks is a /64 network (2^{32} *times* the size of the entire IPv4 address space), it is impossible to tell which part of an announced prefix is allocated or in active use. Therefore, it is impossible to provide insights into IPv6 address utilization from prefix information alone, and efficiently enumerating active IPv6 addresses remains a challenge.

To characterize IPv6 adoption by end-user systems, Colitti et al. included web resources from a dual-stack host and from an IPv4-only host on the Google landing-page, so that its visitors' browsers would attempt to access the dual-stack hosted resources via IPv6 first. Due to possible browser or DNS incompatibilities in respect to IPv6 however, the reported numbers are lower bounds [60].

Plonka and Berger passively measured which and how clients connected to a large content delivery network's IPv6-capable servers and inferred patterns from it, like the stability and density of active IPv6 addresses [16]. Foremski et al. develop Entropy/IP, which is an approach that leverages machine learning to predict likely active IPv6 addresses, based on a seed set of active addresses observed in the past [61]. Murdock et al. introduced a more generic approach (6Gen) to determine potential IPv6 addresses from seed sets [14]. In both cases, addresses that are not in use might be generated, and, hence, the generated addresses are subjected to subsequent liveness verification. Unfortunately, these prior studies depend on existing and comprehensive seed sets, which are difficult to collect without the visibility that a network vantage point provides, such as a large Internet service provider or network operator. However, due to their inherent privacy concerns, these vantage points are heavily guarded and generally not accessible to third parties, such as academic researchers. In contrast, we presented an approach to enumerate an assigned part of the IPv6 Internet that does not depend on a privileged network position. Furthermore, network vantage points can miss certain hosts. For example, for the content delivery networks any host that does not initiate any connections to it, e.g., servers, is missed. These hosts, however, are still discovered by our approach (Section V). Ultimately, the dataset that our approach collects can be readily used as input for generative algorithms, such as Entropy/IP [61] or 6Gen [14].

Czyz et al. aim to evaluate the general filtering policy applied to dual-stack servers (IPv6 and IPv4; less than 20 ports) [13]. As a source for dual-stack hosts, i.e., hosts with IPv4 and IPv6 addresses, they rely on hostnames with both A and AAAA records in the Rapid7 DNS ANY dataset [62]. Consequently, the security posture of IPv6-only hosts is not evaluated, a gap we fill in this paper. As our findings confirm, their results indicate that dual-stack enabled servers have more permissive IPv6 firewall policies compared to IPv4, e.g., SSH, Telnet, and SNMP are more than twice as open for IPv6-capable routers as they are for their IPv4 counterparts. However, their work exhibits limitations that our technique does not have. Specifically, due to their focus on dual-stack hosts Czyz et al. have missed IPv6-only hosts as well as systems lacking forward-zone A and/or AAAA records. We overcome these limitations by presenting a technique to identify active IPv6 hosts in specific networks instead of relying on network vantage points or public, possibly stale, datasets. Hence, we can survey so-far neglected IPv6-only systems, which exhibit critical security issues. Furthermore, contrary to Czyz et al., we pinpoint a possible root cause of the differences in firewall policing between IPv4 and IPv6: Stateless address autoconfiguration (SLAAC).

Fiebig et al. also utilize reverse DNS entries to obtain a view on assigned IPv6 addresses [18]. Specifically, they exploit semantic differences in the type of the response of a nameserver [63] to enumerate reverse zones. However, their work does not include a security evaluation of the identified hosts. We leverage their work as a baseline for our evaluation and we find that our technique performs better for large prefixes, due to the already high deployment rate of DNSSEC in their respective reverse zones. Furthermore, we find that the usefulness of their technique has limitations. After Fiebig et al. presented their findings in late 2016 [19], mitigation technique have been adopted by network operators. Furthermore, their technique generates a significant request volume, which can be mitigated similarly. In contrast, mitigations for our enumeration technique require significant changes to the DNSSEC standard, which we hypothesize industry is unlikely going to adopt in the near future due to deployment concerns (Section VI). Furthermore, our technique is more economical in generated requests, putting less of a strain on networks and rendering network-based detection more difficult.

C. DNSSEC Privacy Issues

DNSSEC-signed zones that leverage NSEC-based denial of existence are known to be vulnerable to zone enumeration attacks [35]. Although NSEC3 renders it more difficult, as a hash-based approach, it remains possible to enumerate the zone through a brute-force attack. Goldberg et al. presented variants of NSEC3 and showed that the modified schemes would still be vulnerable to zone enumeration through brute-force attacks [37]. To break the hashed names, Wander et al. leveraged a GPU to launch a dictionary attack against the ".com" zone and successfully unblinded 64% of the zone [64]. We discussed prior work related to preventing zone-walking attacks on DNSSEC in Section VI, which is why we omit it here in the pursuit of brevity.

Previous work hints at the potential of information leakage through reverse DNS zones [65–67]. However, they only provide preliminary insight, and do not discuss or leverage any information leaks (e.g., resource record types and their meaning for IPv6 reverse zones) nor do they conduct any empirical study on the real-world significance of such leaks. Contrary to prior work, our approach transfers the challenge of unblinding NSEC3 into a new domain. There, we leverage various intricate details, which have not yet received any attention, to considerably reduce the effort to unblind IPv6 addresses from the NSEC3 chain. Specifically, we utilize the way reverse zones are organized, the well-defined structure of IPv6, and the insight that NSEC3 still leaks the record types, which have a specific meaning for reverse zones.

VIII. Conclusion

In this paper, we introduced a technique to enumerate part of the active IPv6 address space as a starting point to evaluate the security state of IPv6-connected hosts. Our approach leverages DNSSEC-signed reverse DNS zones to enumerate active IPv6 addresses that can later be scanned through readily available tools, such as nmap. Although NSEC3 should protect from zone-walking attacks, the combination of the well-defined structure of IPv6 addresses in the reverse zone, and the implications of the disclosure of the record types for the previous and next hashes in the NSEC3 chain counteract its protective impact. In turn, it reduces the search space needed to break the hashed addresses to as little as 2^{64}, with additional reductions in practice through intelligent search due to incremental (e.g., manual or via DHCPv6) and MAC address-based (stateless address autoconfiguration) address assignment schemes. Exploiting these intricacies, we successfully demonstrate that it is practical to enumerate active IPv6 addresses at scale in the face of NSEC3. Furthermore, to the best of our knowledge, we are the first to introduce systematic and practical methodology to enumerate IPv6 addresses through NSEC and NSEC3 based DNSSEC-signed reverse zones by exploiting previously ignored subtleties in the interplay of reverse zones and DNSSEC.

Based on the enumerated address set, we evaluated the state of security of IPv6 hosts and we have shown that many are insufficiently secured. Specifically, IPv6-enabled systems often expose critical infrastructure or sensitive and privacy-concerning information to the outside. For instance, we discovered various routers exposing unsecured Telnet access, or internal file shares being exposed via IPv6, and that the analysis of hostnames in the reverse zone can leak employees' working hours and locations. Furthermore, from our comparative analysis of scanning dual-stack hosts via IPv6 and IPv4, we conclude that one main cause is that globally routable IPv6 addresses are assigned automatically to the machines. It appears that hosts assigning themselves a globally routable IPv6 address is a practice some system administrators are unaware of, as the respective hosts are almost always properly protected from unauthorized access via IPv4.

Finally, we discussed mitigation mechanisms that can be employed to protect against zone-walking in the presence of DNSSEC and, in turn, could prevent IPv6 address enumeration attacks through DNSSEC-signed reverse zones. Ultimately, we reach the conclusion that the proposed defenses suffer from shortcomings that will prevent them from being adopted in practice in the (near) future. Therefore, we expect our approach to continue being a viable IPv6 address enumeration technique and to further improve with the continued deployment of DNSSEC.

IX. Acknowledgements

We thank the anonymous reviewers for their helpful suggestions to improve the paper. We also thank Bruce Maggs for his valuable feedback.

This material is based on research sponsored by the Defense Advanced Research Projects Agency (DARPA) under agreement number FA8750-15-2-0084, the Office of Naval Research (ONR) under grant N00014-17-1-2011 and N00014-15-1-2948, the National Science Foundation (NSF) under grant DGE-1623246, CNS-1704253 and CNS-1408632, and a Security, Privacy and Anti-Abuse Award by Google to Giovanni Vigna.

The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

Any views, opinions, findings, recommendations, or conclusions contained or expressed herein are those of the authors, and do not necessarily reflect the position, official policies, or endorsements, either expressed or implied, the U.S. Government, DARPA, ONR, NSF, or Google.

References

[1] Z. Durumeric, J. Kasten, D. Adrian, J. A. Halderman, M. Bailey, F. Li, N. Weaver, J. Amann, J. Beekman, M. Payer, et al. "The Matter of Heartbleed". In: *Proc. ACM Internet Measurement Conference (IMC)*. Nov. 2014.

[2] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman. "Analysis of the HTTPS Certificate Ecosystem". In: *Proc. ACM Internet Measurement Conference (IMC)*. ACM. Oct. 2013.

[3] F. Cangialosi, T. Chung, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. "Measurement and Analysis of Private Key Sharing in the HTTPS Ecosystem". In: *Proc. ACM Conference on Computer and Communications Security (CCS)*. ACM. Oct. 2016.

[4] T. Fiebig, A. Feldmann, and M. Petschick. "A One-Year Perspective on Exposed In-memory Key-Value Stores". In: *Proc. ACM Workshop on Automated Decision Making for Active Cyber Defense (SafeConf)*. ACM. Oct. 2016.

[5] C. Rossow. "Amplification Hell: Revisiting Network Protocols for DDoS Abuse". In: *Proc. Symposium on Network and Distributed System Security (NDSS)*. Internet Society. Feb. 2014.

[6] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir. "Taming the 800 Pound Gorilla: The Rise and Decline of NTP DDoS Attacks". In: *Proc. ACM Internet Measurement Conference (IMC)*. Nov. 2014.

[7] Z. Durumeric, E. Wustrow, and J. A. Halderman. "ZMap: Fast Internet-wide Scanning and its Security Applications". In: *Proc. USENIX Security Symposium*. USENIX. Aug. 2013.

[8] J. Curran. *ARIN IPv4 Free Pool Reaches Zero*. Sept. 2015. URL: https://www.arin.net/vault/announcements/2015/20150924.html.

[9] S. E. Deering and R. M. Hinden. *Internet Protocol, version 6 (IPv6) Specification*. RFC 2460. Dec. 1998. URL: https://tools.ietf.org/html/rfc2460.

[10] Google Inc. *IPv6 - Google*. Aug. 2016. URL: https://www.google.com/intl/en/ipv6/statistics.html.

[11] Internet Society. *World IPv6 Launch*. Aug. 2016. URL: http://www.worldipv6launch.org/.

[12] M. Kaeo, E. Vyncke, and K. K. Chittimaneni. *Operational Security Considerations for IPv6 Networks*. Internet-Draft draft-ietf-opsec-v6-09. Work in Progress. Internet Engineering Task Force, July 2016. URL: https://tools.ietf.org/html/draft-ietf-opsec-v6-09.

[13] J. Czyz, M. Luckie, M. Allman, and M. Bailey. "Don't Forget to Lock the Back Door! A Characterization of IPv6 Network Security Policy". In: *Proc. Symposium on Network and Distributed System Security (NDSS)*. Internet Society. Feb. 2016.

[14] A. Murdock, F. Li, P. Bramsen, Z. Durumeric, and V. Paxson. "Target Generation for Internet-wide IPv6 Scanning". In: *Proc. ACM Internet Measurement Conference (IMC)*. Nov. 2017.

[15] J. Czyz, M. Allman, J. Zhang, S. Iekel-Johnson, E. Osterweil, and M. Bailey. "Measuring IPv6 Adoption". In: *Proc. ACM SIGCOMM*. ACM. Aug. 2014.

[16] D. Plonka and A. Berger. "Temporal and Spatial Classification of Active IPv6 Addresses". In: *Proc. ACM Internet Measurement Conference (IMC)*. ACM. Nov. 2015.

[17] Cloudflare Inc. *Deprecating the DNS ANY meta-query type*. Mar. 2015. URL: https://blog.cloudflare.com/deprecating-dns-any-meta-query-type/.

[18] T. Fiebig, K. Borgolte, S. Hao, C. Kruegel, and G. Vigna. "Something From Nothing (There): Collecting Global IPv6 Datasets From DNS". In: *Proc. Passive and Active Measurement (PAM)*. Springer. Mar. 2017.

[19] T. Fiebig, K. Borgolte, S. Hao, C. Kruegel, and G. Vigna. "You can -j REJECT but you can not hide: Global scanning of the IPv6 Internet". In: *Chaos Communication Congress (CCC)*. Chaos Computer Club. Dec. 2016.

[20] Internet Society. *Measurements, World IPv6 Launch*. July 2016. URL: http://www.worldipv6launch.org/measurements/.

[21] P. Mockapetris. *Domain Names - Implementation and Specification*. RFC 1035. Nov. 1987. URL: https://rfc-editor.org/rfc/rfc1035.txt.

[22] D. Dagon, M. Antonakakis, P. Vixie, T. Jinmei, and W. Lee. "Increased DNS Forgery Resistance Through 0x20-bit Encoding: Security via Leet Queries". In: *Proc. ACM Conference on Computer and Communications Security (CCS)*. ACM. Oct. 2008.

[23] D. Dagon, M. Antonakakis, K. Day, X. Luo, C. P.Lee, and W. Lee. "Recursive DNS Architectures and Vulnerability Implications". In: *Proc. Symposium on Network and Distributed System Security (NDSS)*. Internet Society. Feb. 2009.

[24] B. Adler. *Who Stole My Web Browser?* Oct. 2009. URL: http://infiniteedge.blogspot.com/2009/10/who-stole-my-web-browser.html.

[25] C. Metz. *Comcast trials DNS hijacker*. July 2009. URL: http://www.theregister.co.uk/2009/07/28/comcast_dns_hijacker/.

[26] S. Weiler and J. Ihren. *Minimally Covering NSEC Records and DNSSEC On-line Signing*. RFC 4470. Oct. 2015. URL: https://rfc-editor.org/rfc/rfc4470.txt.

[27] D. Barr. *Common DNS Operational and Configuration Errors*. RFC 1912. Mar. 2013. URL: https://rfc-editor.org/rfc/rfc1912.txt.

[28] D. J. C. Klensin. *Simple Mail Transfer Protocol*. RFC 2821. Mar. 2013. URL: https://rfc-editor.org/rfc/rfc2821.txt.

[29] Y. Rekhter, B. Volz, and M. Stapp. *The Dynamic Host Configuration Protocol (DHCP) Client Fully Qualified Domain Name (FQDN) Option*. RFC 4702. Mar. 2013. URL: https://rfc-editor.org/rfc/rfc4702.txt.

[30] M. Crawford and B. Haberman. *IPv6 Node Information Queries*. RFC 4620. Oct. 2015. URL: https://rfc-editor.org/rfc/rfc4620.txt.

[31] T. Fiebig, K. Borgolte, S. Hao, C. Kruegel, G. Vigna, and A. Feldmann. "In rDNS We Trust: Revisiting a Common Data-Source's Reliability". In: *Proc. Passive and Active Measurement (PAM)*. Springer. Mar. 2018.

[32] M. Terry. *IP6.ARPA DNSSEC Report*. Feb. 2017. URL: http://stats.research.icann.org/dns/ip6_report/.

[33] R. Chandramouli and S. Rose. *Secure Domain Name System (DNS) Deployment Guide*. NIST, Sept. 2013.

[34] ICANN. *DNSSEC Deployment Report*. Nov. 2017. URL: http://dnssec-deployment.icann.org/dctld/.

[35] B. Laurie, G. Sisson, and R. Arends. *DNS Security (DNSSEC) Hashed Authenticated Denial of Existence*. RFC 5155. Mar. 2008. URL: https://rfc-editor.org/rfc/rfc5155.txt.

[36] J. Abley and T. Manderson. *Nameservers for IPv4 and IPv6 Reverse Zones*. RFC 5855. Oct. 2015. URL: https://rfc-editor.org/rfc/rfc5855.txt.

[37] S. Goldberg, M. Naor, D. Papadopoulos, L. Reyzin, S. Vasant, and A. Ziv. *Stretching NSEC3 to the Limit: Efficient Zone Enumeration Attacks on NSEC3 Variants*. Tech. rep. Boston University, Feb. 2015.

[38] Daniel J. Bernstein. *The nsec3walker tool*. Jan. 2011. URL: http://dnscurve.org/nsec3walker.html.

[39] P. Mockapetris. *Domain names - concepts and facilities*. RFC 1034. Nov. 1987. URL: https://rfc-editor.org/rfc/rfc1034.txt.

[40] IEEE. *IEEE Organizationally Unique Identifier*. Jan. 2018. URL: http://standards-oui.ieee.org/oui.txt.

[41] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman. "Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices". In: *Proc. USENIX Security Symposium*. USENIX. Aug. 2012.

[42] Ripe NCC. *RIPE Routing Information Service (RIS)*. URL: https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris.

[43] University of Oregon. *Route Views Project*. URL: http://bgplay.routeviews.org.

[44] B. Zhang, R. Liu, D. Massey, and L. Zhang. "Collecting the Internet AS-level Topology". In: *ACM Computer Communication Review* 35.1 (Jan. 2005).

[45] ARIN. *ARIN Number Resource Policy Manual*. July 2016. URL: https://www.arin.net/policy/nrpm.html.

[46] G. Huston and D. T. Narten. *IPv6 Address Assignment to End Sites*. RFC 6177. Mar. 2011. URL: https://rfc-editor.org/rfc/rfc6177.txt.

[47] Level 3 Communications. *A New DDoS Reflection Attack: Portmapper; An Early Warning to the Industry*. Aug. 2016. URL: http://blog.level3.com/security/a-new-ddos-reflection-attack-portmapper-an-early-warning-to-the-industry/.

[48] *Quagga bgpd with BGP peers enabled for VPnv4 contains a buffer overflow vulnerability*. Nov. 2015. URL: http://www.kb.cert.org/vuls/id/270232.

[49] *Denial of Service Vulnerability in Quagga BGP Routing Daemon (bgpd)*. Nov. 2015. URL: https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-4049.

[50] O. M. Kolkman. *DNSSEC Operational Practices*. RFC 4641. Mar. 2013. URL: https://rfc-editor.org/rfc/rfc4641.txt.

[51] M. Gieben and M. Mekking. *DNS Security (DNSSEC) Authenticated Denial of Existence*. Internet-Draft draft-gieben-nsec4-01. Work in Progress (Expired). Internet Engineering Task Force, Jan. 2013. URL: https://tools.ietf.org/html/draft-gieben-nsec4-01.

[52] S. Goldberg, M. Naor, D. Papadopoulos, S. Vasant, and A. Ziv. "NSEC5: Provably Preventing DNSSEC Zone Enumeration". In: *Proc. Symposium on Network and Distributed System Security (NDSS)*. Internet Society. Feb. 2015.

[53] J. Vcelak, S. Goldberg, and D. Papadopoulos. *NSEC5, DNSSEC Authenticated Denial of Existence*. Internet-Draft draft-vcelak-nsec5-05. Work in Progress. Internet Engineering Task Force, July 2017. URL: https://tools.ietf.org/html/draft-vcelak-nsec5-05.

[54] S. Goldberg, M. Naor, D. Papadopoulos, L. Reyzin, S. Vasant, and A. Ziv. "NSEC5: Provably Preventing DNSSEC Zone Enumeration". In: *IACR Cryptology ePrint Archive* (July 2014).

[55] N. Aviram, S. Schinzel, J. Somorovsky, N. Heninger, M. Dankel, J. Steube, L. Valenta, D. Adrian, J. A. Halderman, V. Dukhovni, E. Kasper, S. Cohney, S. Engels, C. Paar, and Y. Shavitt. "DROWN: Breaking TLS using SSLv2". In: *Proc. USENIX Security Symposium*. USENIX. Aug. 2016.

[56] J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, G. Bartlett, and J. Bannister. "Census and Survey of the Visible Internet". In: *Proc. ACM Internet Measurement Conference (IMC)*. ACM. Oct. 2008.

[57] Cisco Systems, Inc. *Monitoring IPv6 adoption*. Aug. 2016. URL: http://6lab.cisco.com/stats/.

[58] A. Dhamdhere, M. Luckie, B. Huffaker, A. Elmokashfi, E. Aben, and K. Claffy. "Measuring the Deployment of IPv6: Topology, Routing, and Performance". In: *Proc. ACM Internet Measurement Conference (IMC)*. ACM. Nov. 2012.

[59] IANA. *IPv6 Global Unicast Address Assignments*. Jan. 2016. URL: http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml.

[60] L. Colitti, S. H. Gunderson, E. Kline, and T. Refice. "Evaluating IPv6 Adoption in the Internet". In: *Proc. Passive and Active Measurement (PAM)*. Springer. Apr. 2010.

[61] P. Foremski, D. Plonka, and A. Berger. "Entropy/IP: Uncovering Structure in IPv6 Addresses". In: *Proc. ACM Internet Measurement Conference (IMC)*. ACM. Nov. 2016.

[62] Rapid7. *DNS Records (ANY) Datasets*. 2015. URL: https://scans.io/study/sonar.fdns.

[63] S. Bortzmeyer and S. Huque. *NXDOMAIN: There Really Is Nothing Underneath*. RFC 8020 (Proposed Standard). Internet Engineering Task Force, Nov. 2016. URL: http://www.ietf.org/rfc/rfc8020.txt.

[64] M. Wander, L. Schwittmann, C. Boelmann, and T. Weis. "GPU-based NSEC3 Hash Breaking". In: *Proc. IEEE International Symposium on Network Computing and Applications (NCA)*. IEEE. Aug. 2014.

[65] S. Rose and A. Nakassis. "Minimizing Information Leakage in the DNS". In: *IEEE Network* 22.2 (2008).

[66] R. Arends and P. Koch. "DNS for Fun and Profit". In: *Proc. DFN-CERT Workshop*. Mar. 2005.

[67] S. Rose, R. Chandramouli, and A. Nakassis. "Information Leakage through the Domain Name System". In: *Proc. Cybersecurity Applications & Technology Conference for Homeland Security*. Mar. 2009.