Ten Years of ZMap

Zakir Durumeric
Stanford University
Stanford, CA, USA

David Adrian
Independent
Researcher
Denver, CO, USA

Phillip Stephens
Stanford University
Stanford, CA, USA

Eric Wustrow
University of
Colorado Boulder
Boulder, CO, USA

J. Alex Halderman
University of
Michigan
Ann Arbor, MI, USA

ABSTRACT

Since ZMap's debut in 2013, networking and security researchers have used the open-source scanner to write hundreds of research papers that study Internet behavior. In addition, ZMap has been adopted by the security industry to build new classes of enterprise security and compliance products. Over the past decade, much of ZMap's behavior—ranging from its pseudorandom IP generation to its packet construction—has evolved as we have learned more about how to scan the Internet. In this work, we quantify ZMap's adoption over the ten years since its release, describe its modern behavior (and the measurements that motivated changes), and offer lessons from releasing and maintaining ZMap for future tools.

CCS CONCEPTS

• **General and reference** → *Measurement*; • **Networks** → **Network protocols**; *Network monitoring*; • **Software and its engineering** → **Software creation and management**; • **Security and privacy** → *Network security*.

KEYWORDS

Internet Measurement, Internet Scanning, ZMap

ACM Reference Format:
Zakir Durumeric, David Adrian, Phillip Stephens, Eric Wustrow, and J. Alex Halderman. 2024. Ten Years of ZMap. In *Proceedings of the 2024 ACM Internet Measurement Conference (IMC '24), November 4–6, 2024, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3646547.3689012

1 INTRODUCTION

In 2013, Durumeric et al. released ZMap [39], an open-source network scanner that made it dramatically easier to scan the entire IPv4 address space. Since then, more than 300 research papers have used ZMap to uncover protocol flaws [2, 14], shed light on the WebPKI [37], reverse engineer mercenary spyware [84], understand headline events like Heartbleed [38] and Mirai [12], and more. Beyond research, security companies have developed products on top of ZMap to continuously monitor organizations' attack surfaces and their third-party dependencies. At the same time, like many security tools, ZMap has also been adopted by attackers to identify vulnerable systems. In aggregate, ZMap now accounts for over one-third of all Internet-wide scan traffic.

IMC '24, November 4–6, 2024, Madrid, Spain

ACM ISBN 979-8-4007-0592-2/24/11
https://doi.org/10.1145/3646547.3689012

Since its initial release, ZMap has evolved as we learned more about how to scan the Internet and better understood researchers' needs. Yet, with the exception of Adrian et al.'s write up of ZMap's 10GbE re-architecture [4], the project team has not documented these changes in the research literature. Of the improvements Adrian et al. introduced, both the lock-free randomization algorithm and fast packet transmission approach have since changed. As we maintained and improved ZMap, we also learned a great deal about how to (and not to) build Internet measurement tools.

Motivated by requests from the community to document what we have learned [28], in this work, we present a retrospective analysis of ZMap. We cover how ZMap has been adopted (§2), advances in Internet scanning that challenged the tool's initial design assumptions (§3), significant changes to ZMap's behavior (§4), and what we learned maintaining and improving ZMap (§5). We hope that by analyzing this endeavor, we can help the researchers who build the next generation of Internet measurement tools.

2 USAGE AND INTERNET LANDSCAPE

Since ZMap's release in 2013, its adoption has steadily increased [9, 36]. Today, over 33% of *all* Internet-wide IPv4 scan traffic can be fingerprinted as coming from ZMap. In this section, we provide a high-level overview of ZMap's adoption by academic researchers, security companies, and malicious actors.

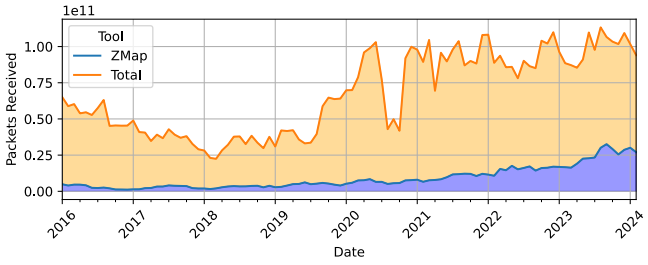


Figure 1: ZMap-Attributed TCP Scan Traffic—ZMap growth has accelerated significantly since 2020. In Q1 2024, 35% of Internet-wide IPv4 TCP scan traffic (by packet) came from ZMap.

2.1 Empirical Analysis

One year after ZMap's release, in 2014, Durumeric et al. found relatively little adoption and that ZMap was primarily used to study academically interesting protocols like HTTP(S) [36]. More recently, in 2021, Anand et al. noted that many "aggressive" scans were from ZMap or Masscan [9]. Using the same methodology as these two studies [9, 36], we measure ZMap adoption by analyzing scans that target at least ten IPs in the ORION Network Telescope [89]. Our analysis is limited to TCP scans, since ORION identifies TCP, ICMP and UDP scanning flows, but only tags scanning tools for TCP flows. In addition, we note that forks of ZMap that remove the static identifying IP ID of 54321 will not be attributed to ZMap.

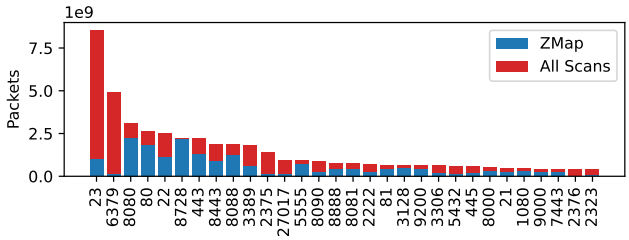


Figure 2: All TCP Scans (Top Ports by Packet)

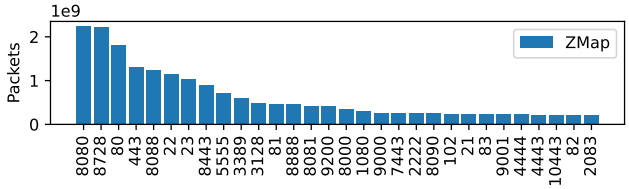


Figure 3: ZMap Scans (Top Ports By Packet)

ZMap usage has increased dramatically over the past four years (Figure 1). From January 1 to March 31, 2024, 35.4% of all Internet-wide IPv4 TCP scan packets originated from ZMap. While it is possible that ICMP and UDP scans use other tools, the vast majority of scan traffic is TCP-based and ZMap still accounts for 33% of all scan packets with only TCP traffic attributed. ZMap scans follow a different distribution of targeted ports than other scans (Figures 2 and 3). For example, ZMap accounts for only 12% of TCP/23 but 69% of TCP/80 and 73% of TCP/8080. In the most extreme case, 99.5% of traffic targeting TCP/8728 (MikroTik router API) is from ZMap, driving it to the sixth most scanned port. There are also dramatic regional differences (Figure 4). For example, while ZMap accounts for more than 66% of scan packets from U.S. hosts, less than 0.5% of Russian scan packets are from ZMap. The outsized U.S. use is driven by its adoption by American security companies (§2.3). In parallel to our work, Griffioen et al. conducted an in-depth analysis of how scanning has evolved over the past decade [49]; the study found that 59% of Internet-wide *scans* in 2024 used ZMap.

2.2 Academic Research

ZMap has been used for a vast range of research purposes, from showing the possible compromise of RSA keys through transient faults [118] to measuring NAT64 deployment [58]. To understand the studies that ZMap has enabled, one author manually analyzed 1,034 papers that cite or reference ZMap through April 2024 via Google Scholar and categorized papers using thematic analysis [23]. We exclude dissertations (since these are often comprised of published papers) as well as studies that used Censys [34]. In total, we identified 307 research papers directly based on ZMap data.

While ZMap is a general measurement tool, it has most prominently been used by the security community (Appendix B). Notably, ZMap has been used in 38 papers to uncover protocol weaknesses in TLS and underlying cryptographic primitives [2, 14, 18], and to uncover deployment challenges and measure adoption [57, 68, 97]. Collecting X.509 certificates, 28 papers have shed light on the WebPKI prior to the widespread adoption of Certificate Transparency [124]. There is also a large contingent of papers that have measured the exposure of IoT devices (25 papers), ICS (14 papers), and security-relevant services (12 papers). Beyond understanding deployment patterns, a number of papers have been able to identify

US	NL	RU	DE	GB	BG	CN	IN	ZA	HK
66%	33%	0.48%	18%	69%	9%	2%	12%	0.1%	2%

Figure 4: ZMap by Country—The ten countries that emanate the most Internet scan traffic by packet have varied ZMap usage.

real-world attacks [35], to reverse engineering attacker infrastructure [12, 83, 84], and to conduct large-scale notifications using ZMap [38, 72]. We encourage the security community to embrace research that builds Internet measurement tools and techniques since these are frequently used to understand and improve security.

Networking-focused papers cover topics like DNS (24 papers), BGP/RPKI (12 papers), censorship (14 papers), and IP usage and NAT (10 papers). In addition to these studies, 53 other papers reference the recommended practices when conducting measurements.

Despite the plethora of publications, academic networks are not responsible for most ZMap traffic, likely because research experiments do not require continuous long-term scanning of a large number of ports. None of the top 100 ASes that emit the most ZMap traffic belong to universities; rather, most traffic originates from security companies and cloud providers. For example, the provider responsible for—by far—the most ZMap scan traffic is Google Cloud (GCP). Examining the reverse DNS records for scanning IPs, we find that GCP is predominately used by Palo Alto Networks to power their Xpanse Attack Surface Management product.

2.3 Industry Adoption

To understand industry adoption, we categorized the organizations identified by Greynoise as using ZMap and mapped these to broad industry categories of security products (e.g., as defined by Gartner):

Attack Surface Management. With the shift to cloud-based infrastructure and a rise of ransomware attacks against enterprise services (e.g., MoveIt and VMWare ESXi), there has been an increased demand for companies to understand their Internet-facing infrastructure. Palo Alto Xpanse, Microsoft RiskIQ, and Rapid7 insightVM, along with numerous other smaller companies, use ZMap as the basis for providing "attack surface management" products that give enterprises up-to-date data about their Internet-exposed risks and potentially unknown assets.

Third-Party Risk Management. Building on the observation that externally visible security configuration and patching patterns correlate with data breaches and compromise [76, 78, 130], companies such as BitSight and FICO use ZMap to build security ratings that enable companies to understand their supply-chain security.

Internet Intelligence. Several non-profits and companies use ZMap to collect data about IP addresses, threat actors, and Internet services, including BinaryEdge, Censys, IPInfo, and ShadowServer. Using this data, multiple countries proactively monitor for and notify organizations about risks (e.g., U.K. [95]).

2.4 Malicious Use

Most security tools have the potential for both helping defenders and being misused by attackers. ZMap is no exception and there is evidence that ZMap has been used maliciously. While it is difficult to ascertain the intent of network traffic from shared providers without application-layer visibility [54, 63], past darknet analysis

has shown that attackers have used "bulletproof" hosting providers to carry out scans for vulnerable services, including MSSQL, RDP, and Mikrotik's router API [9, 36]. Anecdotally, attackers have repurposed ZMap to carry out DOS attacks [13], and, recently, two IoT botnets incorporated ZMap into their malware: between 2021–2023, variants of the Mirai and Medusa botnet adopted ZMap [99, 122].

3 LESSONS IN INTERNET SCANNING

Subsequent discoveries about Internet scanning have challenged some of ZMap's original design assumptions:

Port Diffusion. Despite IANA assignment of ports to L7 protocols, Bano et al. [15] first noted and Izhikevich et al. [60] more formally showed that protocols run across a long tail of ports: only 3% of HTTP and 6% of TLS services run on ports 80 and 443, respectively [60]. Scanning only assigned ports works well for understanding common user-facing protocols such as HTTP(S) but underestimates the impact of some security phenomenon, such as malware and industrial control system exposure. This has spurred new research into more intelligent Internet scanning approaches [61, 80, 111, 114] and led us to shift ZMap's address generation from being purely "horizontal" to support multiple ports.

L4 vs. L7 Discrepancies. Several studies have noted significant discrepancies between L4 and L7 responsiveness [37, 53, 57, 90, 102, 115]. Izhikevich et al. showed that TCP liveness does not reliably indicate service presence because of pervasive middlebox deployment [60]. ZMap's design also encouraged what Hiesgen et al. term "two-phase scanning" [54], in which L4 service discovery and L7 service interrogation are performed separately. In response, some hosts "shun" two-phase scanners, exacerbating the perceived differences between L4 and L7 results [60]. Sattler et al. later devised a method for more accurately identifying highly L4-but-not-L7 responsive prefixes [112]. These differences fundamentally limit ZMap's utility (as a standalone L4 tool) to discovering *potential* services, requiring most work to be completed in follow-up L7 scans and shifting our focus to downstream tools like LZR [60] and ZGrab [3].

Visibility and Consistency. One challenge of running Internet scans is the lack of ground truth for validation. Wan et al. showed that the ZMap paper slightly overestimated the visibility achieved by the tool, and that a single-probe scan actually misses about 2.7% of HTTP(S) hosts [126]. Figures 1 of Hastings et al. [52] and Chung et al. [27] show that different organizations using ZMap sometimes see different results. However, in even the most egregious cases (e.g., Censys [34]), vantage points miss under 5% of services due to blocking; the bulk of loss is typically driven by a handful of small service and cloud providers [126]. For those who need comprehensive coverage, Wan et al. recommends that the best way to mitigate transient drop is to scan from 2–3 geographically and topologically diverse vantages, rather than to send multiple probes from a single scanner, since both probes are oftentimes lost. Results can also differ across scanning tools: Adrian et al. showed that, despite following a similar high-level approach, Masscan [48] finds notably fewer hosts than ZMap, likely due to biases in its randomization algorithm [4].

4 ZMAP CODEBASE

When we released ZMap, we had little idea what community, if any, would emerge. Over the past ten years, more than 80 individuals have committed code to ZMap—we are deeply thankful to those contributors. Despite the high number of committers, 90% of ZMap code has been written by five individuals. Most contributions to the code have been made by industry and academic involvement has been limited: of the 11 external contributors who changed more than 100 LoC, only one is an academic researcher. Academic groups most frequently contributed probe modules or bugfixes; when academics made improvements to core functionality, improvements tend to be forked and renamed (e.g., XMap [73] and ZMapv6 [43], which implemented the same IPv6 functionality) rather than upstreamed. Funding from the NSF *Internet Measurement Research: Methodologies, Tools, and Infrastructure* (IMR) program [96] has been critical in supporting ZMap's continued development.

Beyond usability improvements and bug fixes, there have been several fundamental changes in how ZMap operates beyond the original paper's description. We describe these changes below.

4.1 Address and Port Generation

One of ZMap's key contributions was its ability to *statelessly* and *pseudorandomly* scan the IPv4 address space. A scanner's randomization approach can dramatically affect results [4] and our randomization algorithm has changed repeatedly since ZMap's initial release. ZMap originally scanned all IPv4 addresses ("horizontal scanning") on a single port by iterating over the cyclic group $(\mathbb{Z}/(2^{32}+15)\mathbb{Z})^{\times}$, and coverting group elements to destination addresses. Soon after, we added additional smaller prime order groups to support efficiently scanning subsets of the address space (e.g., $2^{24}+43$ and $2^{16}+1$). Motivated by the findings of Izhikevich et al. [60], we recently added support for multiple ports by iterating over prime order groups up to size $2^{48}+23$ wherein the top $\lceil \log_2 IPs \rceil$ bits of each group element are used to identify the the target IP address and bottom $\lceil \log_2 Ports \rceil$ bits identify the target port. While conceptually approach, it has several ramifications:

Hosts vs. Targets. ZMap originally tracked statistics by IP address. The new randomization approach selects from a pool of (IP, port) "targets", rather than considering IPs and ports independently. As a result, all data, metadata, and configuration in ZMap are now based on the notion of an (IP, port) target. This enables randomization across the IP–Port space (e.g., rather than rotating scans across ports with each port operating independently), but precludes options like "max hosts" without significant additional state.

Identifying Generators. To create a new permutation of the address space for every scan, ZMap originally identified a random generator (i.e., primitive root) of the appropriate multiplicative group by identifying a generator of $(\mathbb{Z}_{p-1}, +)$ and then mapping it into $(\mathbb{Z}_p^*, \times)$ [39]. This was practical because a generator of the additive group is any integer coprime with $p-1$, which is efficiently testable with the Euclidean algorithm against randomly drawn integers. There are $\phi(2^{32}+14) \approx 10^9$ generators of the additive group, resulting in an average four attempts to identify a generator. Nearly all additive generators could be mapped into usable generators in the multiplicative group, since the only constraint on the multiplicative generators was that they were less than 2^{32} (to ensure safe multiplication using 64-bit arithmetic).

For multiport scans, to support iterating over 2^{48} elements, we need to efficiently find multiplicative generators smaller than 2^{16} in

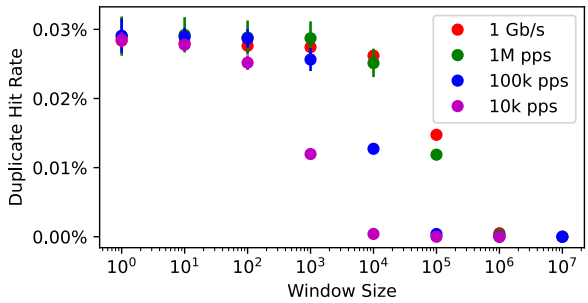


Figure 5: Sliding Window Duplicate Rate—We moved to a sliding window approach for deduplicating responses to support multiple ports. A window size of 10^6 eliminates nearly all duplicates.

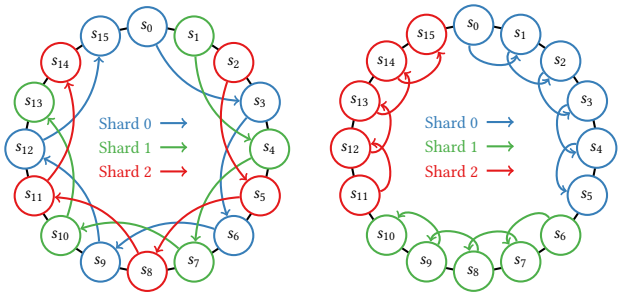
a 2^{48} search space. Unfortunately, because an additive generator can map to a multiplicative generator anywhere in the group, only $1/2^{32}$ candidate additive generators are usable. To address this, we flipped our approach. For each group defined by prime p, we precalculate and store the prime factorization of $p-1 = k_1^{a_1}k_2^{a_2}\dots k_n^{a_n}$ for distinct primes k_i $(1 \le i \le n)$. At runtime, we generate a random candidate generator $g \in [2\,..\,2^{16}-1]$ that is guaranteed to keep future arithmetic within the 64-bit address space. Then we ensure that g is a generator of $\mathbb{Z}_p^*$ by checking that: $g^{(p-1)/k_i} \mod p \neq 1 \,\forall\, i \in [1\,..\,n]$. This requires an average four attempts.

Response Deduplication. Hosts frequently send back repeated responses, in some cases indefinitely. While we initially thought these were due to broken TCP implementations, Goldblatt et al. showed that some hosts will aggressively send tens of thousands of response packets, which they term "blowback" [47]. We originally filtered out duplicate responses using a paged 2^{32}-bit bitmap, which used 512 MB of memory. While this approach guarantees no duplicates, extending it to the 48-bit space of IPs and ports would require 35 TB. Instead, we switch to maintaining a sliding window of the last n IP/Port responses, using a Judy array [16]. As can be seen in Figure 5, a window of 10^6 entries (ZMap default) is effective to filter nearly all duplicate responses, and lower scan rates can make do with smaller window sizes. We found zero duplicates for three trials of 1 Gbps scans targeting TCP/80 in April 2024.

4.2 Scan Sharding

In 2014, Adrian et al. [4] introduced a mutex-free sharding mechanism for ZMap's address generation. This enabled scans to be split across machines and improved performance by allowing multiple send threads on one machine to operate independently. As Mazel et al. noted when they showed that ZMap can be fingerprinted through its IP generation method [86], we shifted the sharding approach in 2017. Given that other work is analyzing ZMap's address generation, we describe the two approaches:

Interleaved Sharding. Sharding was originally implemented with each shard iterating by N steps at a time, offset by one step. For N shards, each shard $n \in [1\,..\,N]$ iterates by multiplying the current element by g^N, but begins iteration at g^n (Figure 6a). With multiple threads in place, each shard n is then further split into T subshards, with each subshard iterating by g^{NT} offset by g^{n+tN}, where n is the shard index and t is the thread index. While conceptually simple, the approach requires calculating the end



(a) Interleaved (Old) **(b)** Pizza (New)

Figure 6: Sharding Approaches—In 2017, ZMap changed sharding approaches. Replicated with permission from [86].

point of each shard to know when to stop iterating. NT is not guaranteed to cleanly divide $p-1$, and so a shard might not repeat its initial value. Unfortunately, the last index of a shard does not have a closed form expression and we found that calculating it is prone to off-by-one errors. After repeated correctness issues, we switched to a simpler mechanism.

Pizza Sharding. Rather than interleaving shards, we divide the multiplicative group into N ranges of values of increasing order, e.g., $[g^0, g^{(p-1)/N})$, $[g^{(p-1)/N}, g^{2(p-1)/N})$, $[g^{2(p-1)/N}, g^{3(p-1)/N})$. For subshards, we similarly slice a single shard into T ranges of values of increasing order. Visually, this is similar to slicing a pizza into N slices, and then subdividing each slice into T subslices (Figure 6b). Because elements are iterated over pseudorandomly in the group, the same randomness guarantees are provided by the second approach while being easier to reason about and implement without off-by-one errors or infinite loops.

4.3 Packet Construction

Striving for the highest send rate, ZMap originally used the smallest possible probes, with no included TCP or IP options. While protocol compliant, we later observed that ZMap would consistently miss some hosts accessible to OS network stacks. By varying TCP options, we found that including any of the Selective ACK (SA), Timestamp (TS), Window Select (WS), or Maximum Segment Size (MSS) TCP options yields a 1.5–2.0% increase in hitrate relative to no options in a scan of TCP/80 (Figure 7). The order of options also affects results: the optimal byte-layout order, minding the TCP 4-byte word boundary, finds only slightly fewer hosts (0.0023%, ≈1.5K hosts in an IPv4 scan of TCP/80) than when sending options using the exact ordering of Linux, BSD, or Windows.

TCP options affect packet size and therefore scan rate. However, including the MSS option alone finds the vast majority of services (over 99.99% of services on TCP/80) and remains under the minimum size of an Ethernet frame, continuing to support the maximum 1.488 Mpps line rate of a 1 GbE link. Using the Windows or Linux packet layouts finds slightly more services but reduces send rates to 1.389 and 1.276 Mpps, respectively. While filtering packets based on TCP options could be due to defensive mechanisms attempting to block scanning, removing the easily identified static IP ID value of ZMap probes appears to have no impact on scan hit rates. We performed three scans of 10% of IPv4 on TCP/80 in April 2024 with a static IP ID and with a random per-packet IP ID and find that

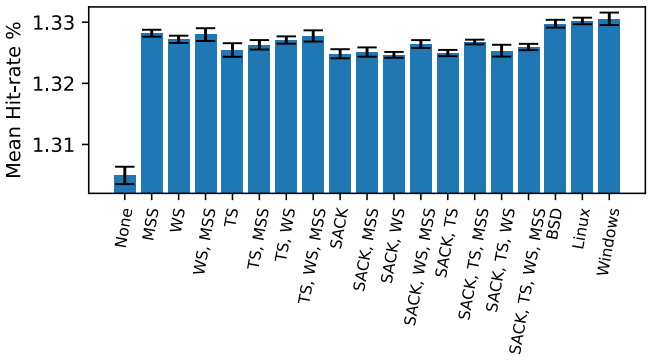


Figure 7: Hitrates for Varying TCP Options—SYN probes without any TCP options, as originally sent by ZMap, find 1.5–2.0% fewer services on TCP/80 than probes that include options. Mimicking common OSes maximizes coverage. Note truncated y axis.

the difference in hit-rate between the random and static IP IDs is not statistically significant. In early 2024, ZMap changed its default behavior to use random per-probe IP IDs.

5 LESSONS AND RECOMMENDATIONS

Based on our experiences, we offer several lessons and recommendations for researchers building future Internet measurement tools. These lessons are derived from decisions that we revisited, approached differently in subsequent tools like ZGrab and ZDNS, would make differently if we were to build ZMap today, or believe were fundamental to ZMap's success. As such, these recommendations are not comprehensive and are inherently opinionated, focusing on decisions where the right choice for ZMap was non-obvious to us at the time. They are, however, the starting point for how we would architect future measurement tools ourselves.

Tools Not Frameworks. First documented in 1978 by McIlroy et al. [87] and more concisely captured by Salus in 1994 [110], the Unix philosophy is to write programs that:

(1) do one thing and do it well;
(2) work together; and
(3) handle text streams (a universal interface).

Nearly 50 years later, we cannot agree more with this guidance. It is difficult to predict how researchers will use measurement tools or the environments in which they will operate. ZMap was originally envisioned as a framework where researchers would build customized *Scan Modules* and *Output Modules* for service follow up. In practice, the vast majority of researchers use ZMap for service discovery and pipe results to secondary tools for investigation or storage. The output modules ZMap included for connecting to specific databases (e.g., Redis) became liabilities, requiring upkeep and complicating testing and packaging. In time, we removed them, opting to support only Text, CSV, and JSON Lines output.

Recommendation: Build small, simple, easy-to-understand, easy-to-use, and easy-to-test measurement tools that can be creatively assembled, rather than complex applications or software frameworks. Build applications that do *one* thing well. Continuously output results on a per-record/per-line basis when possible. Avoid proprietary formats and standardize output on well-worn interfaces like CSV, JSON Lines [127], BSON, and Apache Avro. Carefully consider whether binary formats are worth the cost of direct interoperability with existing data processing toolkits and command-line tools.

Usability. ZMap was not the first fast, asynchronous network scanner: Unicornscan [69], Scanrand [65], and IRLscanner [71] were released years prior. IRLscanner was published at IMC; though Unicornscan and Scanrand were both unknown to the team. ZMap was likely more successful than prior tools due to its greater ease of use: it enabled researchers to scan the IPv4 Internet from a single machine by running a single command.

Recommendation: Obsess over ease of installation, usage, and troubleshooting as well as documentation. It is better to have a tool that is easy to use and less full featured than vice versa.

Library and Command Line Wrapper. It is natural to build an application where the command-line interface, application configuration, and operation are intermingled, since most tools are first used via the CLI and this is the path of least resistance. However, this design will limit a tool's potential to be integrated into larger systems. Automated or continuous measurements, such as the scans that power services like Censys [34], are cumbersome to control from the CLI and more suited to a library interface.

Recommendation: Structure tools with two major components: a backend library and a simple command line interface that wraps the library. This investment is relatively small and will enable the tool to be used in larger systems.

Data, Metadata, and Logs. Given the amount of raw data collected by many measurement tools, it is difficult to tell whether experiments are operating as expected without analyzing metadata in real time. In addition, tracking as much information about the execution (e.g., time, version of software, configuration parameters, and environment) helps to later interpret, troubleshoot, or reproduce results. Logs are helpful for human debugging, but they are not inherently designed to be machine-parsable, which is needed for monitoring long-running experiments. Ultimately, we extended ZMap to produce four output streams: (1) data, (2) logs, (3) real-time updates (e.g., packets sent, received, dropped per second), and (4) machine-readable metadata at completion.

Recommendation: Design measurement tools to produce separate streams of data, metadata, and logs. Do not cross these streams, since this complicates downstream processing. Be liberal in what environment and execution information is included in scan metadata, as it is difficult to know *a priori* what will be useful. Adopt a logging library that supports multiple log levels, and use debug-level logging liberally to enable future troubleshooting. In slight contrast to SoMeta [113], we recommend that metadata collection should be built *into* measurement tools to maximize ease of use.

Static Types and Output Schema. JSON and CSV provide considerable flexibility for encoding data. For example, JSON objects can have dynamic keys and value types across records. However, downstream applications/databases often do not support this flexibility, and it is easy to create valid but painful to process records.

Recommendation: Define a schema for the data you output. Ensure that each field uses a single, well defined type and that the type of one field does not depend on the value of another field. Avoid maps with dynamic keys, and instead use lists of a static document type.

Consider using a tool like JSON Schema [133] or ZSchema [33] to document the structure of output data and metadata.

Versioning and Releases. We released ZMap far too infrequently, repeatedly wanting to include one more feature or bugfix in each release. In particular, ZMap 3.0 was released nearly eight years after the previous release of ZMap 2.1.1. Unfortunately, this meant that most users were either using long out-of-date releases or unversioned code. This made debugging difficult and prevented users from describing the ZMap version they used when publishing.

Recommendation: Follow the Semantic Versioning Specification [105] religiously. Focus on making regular, stable, versioned releases rather than trying to finish a preset amount of work.

Language Choice. When we wrote ZMap, C/C++ were the only practical high-performance systems languages available. It is easy to convince oneself that it is possible to safely write C code; empirical evidence overwhelmingly says the opposite [44, 121]. Network parsers are particularly hard to implement safely and must protect against attacker-controlled input [1]. ZMap has had multiple regressions that caused incomplete measurements and memory safety bugs, which could have been avoided (e.g., [19, 32, 92]). We also found that memory safety concerns make it harder to review external contributions for safety and correctness, which has reduced the rate at which we merge improvements. If were to implement ZMap today, we would do so in Rust.

Recommendation: Develop tools in modern, memory-safe languages like Rust and Go. While Rust has a relatively steep learning curve, its safety and performance make it ideal for performance-critical applications. Go's simple syntax and parallelism-oriented architecture make it particularly suited for quickly developing high-performance measurement tools (e.g., ZGrab [3] and ZDNS [59]).

6 INTERNET CITIZENSHIP

Our work provides an opportunity to revisit our original best practices for conducting scans [39]. Overall, we believe that the 2013 recommendations remain a sound set of considerations. However, we encourage the measurement community to treat these recommendations as good practices for *most* research, not as a set of requirements nor as the basis for having conducted research ethically. For example, there may be situations when scans are better performed unattributed or when opting out specific networks could invalidate results (e.g., tracking a specific threat actor). We additionally recommend that researchers:

(1) Investigate whether existing datasets suffice. Often, these provide better coverage and reduce aggregate bandwidth.
(2) Publish newly collected datasets for other researchers.
(3) Deploy WHOIS entries that identify how to contact you.
(4) Validate how handshakes will appear in logs. For example, some benign SSH handshakes inadvertently show up as failed authentication attempts that concern operators.

Institutions have adopted different practices for validating opt-out requests [11]. We have found it necessary to verify the authenticity of exclusion requests. Given that IP address ownership changes over time, it likely makes sense to eventually expire opt-out requests and it may not make sense for institutions to share blocklists. Our team expires requests after 1–2 years and found that in the vast majority of exclusions are not re-requested. We offer the following, updated set of best practices as a *recommended starting point* when conducting active measurements:

(1) **Minimize Internet Impact.** While Internet scanning is a powerful research methodology, it can also affect systems and create work for operators. Consider whether existing open source datasets provide the data you need. If you do perform scans, conduct scans no larger or more frequent than necessary and at the minimum scan rate needed for your research objectives. Publish any scan data you collect.
(2) **Signal Intent.** When possible, publish reverse DNS entries, IP WHOIS records, and a website that describes the scans. Ensure that operators can easily contact the research team.
(3) **Provide An Opt-Out Mechanism.** Provide a simple mechanism for operators to request exclusion from future scans. Indicate the IP ranges you use for scanning so that operators can drop research traffic themselves.
(4) **Proactively Investigate Effects.** Run newly developed scanning code against your own systems to ensure that you understand how scans might affect devices and appear in logs. Start with small experiments before completing full scans in case your scanner causes unexpected problems.
(5) **Coordinate Locally.** Coordinate with your local IT and security teams to reduce the risk of overwhelming local networks, as well as to ensure that they know how to handle any inbound inquiries from operators.
(6) **Disclose Results.** When appropriate, consider how you can improve the security of the systems you have scanned. Responsibly disclose security problems you uncover and consider notifying vulnerable system owners.

7 CONCLUSION

The most exciting aspect of building ZMap has been watching how other researchers have used it in unpredicted but valuable ways to meaningfully improve the Internet. We are sincerely thankful to everyone who has contributed, pushing the tool close to feature completion. While the development of new ZMap features has slowed, we are excited to continue to expand the ecosystem of tools that work with ZMap (e.g., ZDNS [59] and ZGrab [3]) and to enable an even broader range of measurement uses. Many of the lessons we learned from maintaining ZMap may seem obvious today, but were not obvious at the time. We hope that our retrospective analysis will help the community build an even richer and more reliable ecosystem of Internet measurement tools moving forward.

ACKNOWLEDGMENTS

We thank all of the contributors to ZMap, the IT and Security teams at the University of Michigan and Stanford University, Michael Bailey, Jeff Cody, and Liz Izhikevich. We thank the reviewers at IMC for their suggestions and feedback. This work was supported by the National Science Foundation under grants CNS-2223360, CNS-1518888, and CNS-2319080, as well as a Sloan Research Fellowship. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of their employers or the sponsors.

REFERENCES

[1] The rule of 2, 2024. https://chromium.googlesource.com/chromium/src/+/master/docs/security/rule-of-2.md.

[2] Adrian, D., Bhargavan, K., Durumeric, Z., Gaudry, P., Green, M., Halderman, J. A., Heninger, N., Springall, D., Thomé, E., Valenta, L., et al. Imperfect forward secrecy: How Diffie-Hellman fails in practice. In *ACM Conference on Computer and Communications Security* (2015).

[3] Adrian, D., and Durumeric, Z. https://github.com/zmap/zgrab2.

[4] Adrian, D., Durumeric, Z., Singh, G., and Halderman, J. A. Zippier ZMap: Internet-wide scanning at 10 Gbps. In *USENIX Workshop on Offensive Technologies* (2014).

[5] Alaraj, A., Bock, K., Levin, D., and Wustrow, E. A global measurement of routing loops on the Internet. In *International Conference on Passive and Active Network Measurement* (2023).

[6] Albakour, T., Gasser, O., Beverly, R., and Smaragdakis, G. Third time's not a charm: exploiting SNMPv3 for router fingerprinting. In *ACM Internet Measurement Conference* (2021).

[7] Albakour, T., Gasser, O., and Smaragdakis, G. Pushing alias resolution to the limit. In *ACM Internet Measurement Conference* (2023).

[8] Alt, L., Beverly, R., and Dainotti, A. Uncovering network tarpits with degreaser. In *30th Annual Computer Security Applications Conference* (2014).

[9] Anand, A., Kallitsis, M., Sippe, J., and Dainotti, A. Aggressive Internet-wide scanners: Network impact and longitudinal characterization. In *Conference on emerging Networking EXperiments and Technologies* (2023).

[10] Anderson, S., Bell, T., Egan, P., Weinshenker, N., and Barford, P. Powerping: Measuring the impact of power outages on Internet hosts in the U.S. *International Conference on Critical Infrastructure Protection.*

[11] Anonymous. Can't stop the scan: An empirical study on opting out of internet-wide scanning. https://csl.nict.go.jp/pdf/paper_draft.pdf.

[12] Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., and Zhou, Y. Understanding the Mirai botnet. In *USENIX Security Symposium* (2017).

[13] Arise, H. Clogging, saturating and DoSing Russia's Internet with ZMap. https://www.hackers-arise.com/post/clogging-and-dosing-russia-s-internet-with-zmap.

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

[15] Bano, S., Richter, P., Javed, M., Sundaresan, S., Durumeric, Z., Murdoch, S. J., Mortier, R., and Paxson, V. Scanning the Internet for liveness. *ACM SIGCOMM Computer Communication Review* (2018).

[16] Baskins, D. The Judy array implementation, 2000.

[17] Bayat, N., Mahajan, K., Denton, S., Misra, V., and Rubenstein, D. Down for failure: Active power status monitoring. *Future Generation Computer Systems* (2021).

[18] Beurdouche, B., Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Kohlweiss, M., Pironti, A., Strub, P.-Y., and Zinzindohoue, J. K. A messy state of the union: Taming the composite state machines of TLS.

[19] Bieri, L. Fixed missing null termination. https://github.com/zmap/zmap/pull/118.

[20] Bock, K., Alaraj, A., Fax, Y., Hurley, K., Wustrow, E., and Levin, D. Weaponizing middleboxes for TCP reflected amplification. In *USENIX Security Symposium* (2021).

[21] Bonkoski, A., Bielawski, R., and Halderman, J. A. Illuminating the security issues surrounding Lights-Out server management. In *USENIX Workshop on Offensive Technologies* (2013).

[22] Bos, J. W., Halderman, J. A., Heninger, N., Moore, J., Naehrig, M., and Wustrow, E. Elliptic curve cryptography in practice. In *Financial Cryptography and Data Security* (2014).

[23] Braun, V., and Clarke, V. *Thematic analysis.* American Psychological Association, 2012.

[24] Brinkmann, M., Dresen, C., Merget, R., Poddebniak, D., Müller, J., Somorovsky, J., Schwenk, J., and Schinzel, S. ALPACA: Application layer protocol confusion-analyzing and mitigating cracks in TLS authentication. In *USENIX Security Symposium* (2021).

[25] Brubaker, C., Jana, S., Ray, B., Khurshid, S., and Shmatikov, V. Using frankencerts for automated adversarial testing of certificate validation in SSL/TLS implementations. In *IEEE Symposium on Security and Privacy* (2014).

[26] Bushart, J., and Rossow, C. DNS unchained: Amplified application-layer DoS attacks against DNS authoritatives. In *Research in Attacks, Intrusions, and Defenses (RAID)* (2018).

[27] Chung, T., Liu, Y., Choffnes, D., Levin, D., Maggs, B. M., Mislove, A., and Wilson, C. Measuring and applying invalid SSL certificates: The silent majority. In *ACM Internet Measurement Conference* (2016).

[28] Claffy, K., and Clark, D. The 11th Workshop on Active Internet Measurements (AIMS-11) workshop report. *ACM SIGCOMM Computer Communication Review* (2019).

[29] Costin, A., Zaddach, J., Francillon, A., and Balzarotti, D. A Large-scale analysis of the security of embedded firmwares. In *USENIX Security Symposium* (2014).

[30] Dahlmanns, M., Lohmöller, J., Fink, I. B., Pennekamp, J., Wehrle, K., and Henze, M. Easing the conscience with OPC UA: An Internet-wide study on insecure deployments. In *ACM Internet Measurement Conference* (2020).

[31] Dahlmanns, M., Lohmöller, J., Pennekamp, J., Bodenhausen, J., Wehrle, K., and Henze, M. Missed opportunities: measuring the untapped TLS support in the industrial Internet of things. In *ACM Asia Conference on Computer and Communications Security* (2022).

[32] Durumeric, Z. Fix use-after-free's in ipip probe module. https://github.com/zmap/zmap/pull/815.

[33] Durumeric, Z., and Adrian, D. Zschema. https://github.com/zmap/zschema.

[34] Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., and Halderman, J. A. A search engine backed by Internet-wide scanning. In *ACM Conference on Computer and Communications Security* (2015).

[35] Durumeric, Z., Adrian, D., Mirian, A., Kasten, J., Bursztein, E., Lidzborski, N., Thomas, K., Eranti, V., Bailey, M., and Halderman, J. A. Neither snow nor rain nor MITM… an empirical analysis of email delivery security. In *ACM Internet Measurement Conference* (2015).

[36] Durumeric, Z., Bailey, M., and Halderman, J. A. An Internet-wide view of Internet-wide scanning. In *USENIX Security Symposium* (2014).

[37] Durumeric, Z., Kasten, J., Bailey, M., and Halderman, J. A. Analysis of the HTTPS certificate ecosystem. In *ACM Internet Measurement Conference* (2013).

[38] Durumeric, Z., Li, F., Kasten, J., Amann, J., Beekman, J., Payer, M., Weaver, N., Adrian, D., Paxson, V., Bailey, M., and Halderman, J. A. The matter of Heartbleed. In *ACM Internet Measurement Conference* (2014).

[39] Durumeric, Z., Wustrow, E., and Halderman, J. A. ZMap: Fast Internet-wide scanning and its security applications. In *USENIX Security Symposium* (2013).

[40] Fiebig, T., Feldmann, A., and Petschick, M. A one-year perspective on exposed in-memory key-value stores. In *ACM Workshop on Automated Decision Making for Active Cyber Defense* (2016).

[41] Franzen, F., Steger, L., Zirngibl, J., and Sattler, P. Looking for honey once again: Detecting RDP and SMB honeypots on the Internet. In *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* (2022).

[42] Frolov, S., Wampler, J., and Wustrow, E. Detecting probe-resistant proxies. In *Network and Distributed System Security Symposium* (2020).

[43] Gasser, O., Scheitle, Q., Gebhard, S., and Carle, G. Scanning the IPv6 Internet: Towards a comprehensive hitlist. *arXiv preprint arXiv:1607.05179* (2016).

[44] Gaynor, A. What science can tell us about C and C++'s security, 2020. https://alexgaynor.net/2020/may/27/science-on-memory-unsafety-and-security/.

[45] Giotsas, V., Smaragdakis, G., Dietzel, C., Richter, P., Feldmann, A., and Berger, A. Inferring BGP blackholing activity in the Internet. In *ACM Conference on Computer and Communications Security* (2017).

[46] Giotsas, V., Smaragdakis, G., Huffaker, B., Luckie, M., and Claffy, K. Mapping peering interconnections to a facility. In *11th ACM Conference on Emerging Networking Experiments and Technologies* (2015).

[47] Goldblatt, D., Vuong, C., and Rabinovich, M. On blowback traffic on the Internet. *arXiv preprint arXiv:2305.04434* (2023).

[48] Graham, R. D. Masscan: Mass IP port scanner. *URL: https://github.com/robertdavidgraham/masscan* (2014).

[49] Griffioen, H., Koursiounis, G., Smaragdakis, G., and Doerr, C. Have you SYN me? characterizing ten years of Internet scanning. In *ACM Internet Measurement Conference* (2024).

[50] Guo, H., and Heidemann, J. Detecting ICMP rate limiting in the Internet. In *Passive and Active Measurement* (2018).

[51] Guo, R., Chen, J., Liu, B., Zhang, J., Zhang, C., Duan, H., Wan, T., Jiang, J., Hao, S., and Jia, Y. Abusing CDNs for fun and profit: Security issues in cdns' origin validation. In *IEEE Symposium on Reliable Distributed Systems* (2018).

[52] Hastings, M., Fried, J., and Heninger, N. Weak keys remain widespread in network devices. In *ACM Internet Measurement Conference* (2016).

[53] Heninger, N., Durumeric, Z., Wustrow, E., and Halderman, J. A. Mining your Ps and Qs: Detection of widespread weak keys in network devices. In *USENIX Security Symposium* (2012).

[54] Hiesgen, R., Nawrocki, M., King, A., Dainotti, A., Schmidt, T. C., and Wählisch, M. Spoki: Unveiling a new wave of scanners through a reactive network telescope. In *USENIX Security Symposium* (2022).

[55] Hilts, A., and Parsons, C. Half baked: The opportunity to secure cookie-based identifiers from passive surveillance. In *5th USENIX Workshop on Free and Open Communications on the Internet* (2015).

[56] Hlavacek, T., Cunha, I., Gilad, Y., Herzberg, A., Katz-Bassett, E., Schapira, M., and Shulman, H. Disco: Sidestepping rpki's deployment barriers. In *Network and Distributed System Security Symposium* (2020).

[57] Holz, R., Amann, J., Mehani, O., Wachs, M., and Kaafar, M. A. TLS in the wild: An Internet-wide analysis of TLS-based protocols for electronic communication.

arXiv preprint arXiv:1511.00341 (2015).

[58] Hsu, A., Li, F., Pearce, P., and Gasser, O. A first look at NAT64 deployment in-the-wild. In *Conference on Passive and Active Network Measurement* (2024).

[59] Izhikevich, L., Akiwate, G., Berger, B., Drakontaidis, S., Ascheman, A., Pearce, P., Adrian, D., and Durumeric, Z. ZDNS: a fast DNS toolkit for Internet measurement. In *ACM Internet Measurement Conference* (2022).

[60] Izhikevich, L., Teixeira, R., and Durumeric, Z. LZR: Identifying unexpected Internet services. In *USENIX Security Symposium* (2021).

[61] Izhikevich, L., Teixeira, R., and Durumeric, Z. Predicting IPv4 services across all ports. In *ACM SIGCOMM Conference* (2022).

[62] Izhikevich, L., Tran, M., Izhikevich, K., Akiwate, G., and Durumeric, Z. Democratizing leo satellite network measurement. *ACM SIGMETRICS* (2024).

[63] Izhikevich, L., Tran, M., Kallitsis, M., Fass, A., and Durumeric, Z. Cloud watching: Understanding attacks against cloud-hosted services. In *ACM Internet Measurement Conference* (2023).

[64] Janovsky, A., Nemec, M., Svenda, P., Sekan, P., and Matyas, V. Biased rsa private keys: Origin attribution of gcd-factorable keys. In *25th European Symposium on Research in Computer Security* (2020).

[65] Kaminsky, D. Paketto simplified. https://dankaminsky.com/2002/11/18/77/.

[66] Kim, S. K., Ma, Z., Murali, S., Mason, J., Miller, A., and Bailey, M. Measuring ethereum network peers. In *ACM Internet Measurement Conference* (2018).

[67] Kosek, M., Schumann, L., Marx, R., Doan, T. V., and Bajpai, V. Dns privacy with speed? evaluating dns over quic and its impact on web performance. In *ACM Internet Measurement Conference* (2022).

[68] Kranch, M., and Bonneau, J. Upgrading https in mid-air. In *Network and Distributed System Security Symposium* (2015).

[69] Lee, R. E., and Louis, J. C. Introducing unicornscan. https://defcon.org/images/defcon-13/dc13-presentations/DC_13-Lee.pdf.

[70] Lee, Y., and Spring, N. Identifying and aggregating homogeneous ipv4/24 blocks with hobbit. In *ACM Internet Measurement Conference* (2016).

[71] Leonard, D., and Loguinov, D. Demystifying service discovery: implementing an Internet-wide scanner. In *ACM Internet Measurement Conference* (2010).

[72] Li, F., Durumeric, Z., Czyz, J., Karami, M., Bailey, M., McCoy, D., Savage, S., and Paxson, V. You've got vulnerability: Exploring effective vulnerability notifications. In *USENIX Security Symposium* (Aug. 2016).

[73] Li, X., Liu, B., Zheng, X., Duan, H., Li, Q., and Huang, Y. Fast IPv6 network periphery discovery and security implications. In *IEEE/IFIP International Conference on Dependable Systems and Networks* (2021).

[74] Lichtblau, F., Streibelt, F., Krüger, T., Richter, P., and Feldmann, A. Detection, classification, and analysis of inter-domain traffic with spoofed source IP addresses. In *ACM Internet Measurement Conference* (2017).

[75] Liu, D., Hao, S., and Wang, H. All your dns records point to us: Understanding the security threats of dangling dns records. In *ACM Conference on Computer and Communications Security* (2016).

[76] Liu, Y., Sarabi, A., Zhang, J., Naghizadeh, P., Karir, M., Bailey, M., and Liu, M. Cloudy with a chance of breach: Forecasting cyber security incidents. In *USENIX Security Symposium* (2015).

[77] Liu, Y., Tome, W., Zhang, L., Choffnes, D., Levin, D., Maggs, B., Mislove, A., Schulman, A., and Wilson, C. An end-to-end measurement of certificate revocation in the web's pki. In *ACM Internet Measurement Conference* (2015).

[78] Liu, Y., Zhang, J., Sarabi, A., Liu, M., Karir, M., and Bailey, M. Predicting cyber security incidents using feature-based characterization of network-level malicious activities. In *ACM International Workshop on Security and Privacy Analytics* (2015).

[79] Lu, C., Liu, B., Li, Z., Hao, S., Duan, H., Zhang, M., Leng, C., Liu, Y., Zhang, Z., and Wu, J. An end-to-end, large-scale measurement of dns-over-encryption: How far have we come? In *ACM Internet Measurement Conference* (2019).

[80] Luo, Y., Li, C., Wang, Z., and Yang, J. IPREDS: efficient prediction system for Internet-wide port and service scanning. *Proceedings of the ACM on Networking*, CoNEXT1.

[81] Malhotra, A., Cohen, I. E., Brakke, E., and Goldberg, S. Attacking the network time protocol. *Cryptology ePrint Archive* (2015).

[82] Mangino, A., Pour, M. S., and Bou-Harb, E. Internet-scale insecurity of consumer Internet of things: An empirical measurements perspective. *ACM Transactions on Management Information Systems (TMIS)* (2020).

[83] Marczak, B., Scott-Railton, J., Senft, A., Poetranto, I., and McKune, S. Pay no attention to the server behind the proxy: Mapping finfisher's continuing proliferation. Tech. rep., 2015.

[84] Marczak, W. R., Scott-Railton, J., Marquis-Boire, M., and Paxson, V. When governments hack opponents: A look at actors and technology. In *USENIX Security Symposium* (2014).

[85] Maroofi, S., Korczyński, M., Hölzel, A., and Duda, A. Adoption of email anti-spoofing schemes: a large scale analysis. *IEEE Transactions on Network and Service Management* (2021).

[86] Mazel, J., and Strullu, R. Identifying and characterizing zmap scans: a cryptanalytic approach. *arXiv preprint arXiv:1908.04193* (2019).

[87] McIlroy, M., Pinson, E., and Tague, B. Unix time-sharing system: Forward. *The Bell system technical journal 57*, 6 (1978), 1899–1904.

[88] Mehani, O., Holz, R., Ferlin, S., and Boreli, R. An early look at multipath TCP deployment in the wild. In *International workshop on hot topics in planet-scale measurement* (2015).

[89] Merit Network. ORION network telescope. https://www.merit.edu/initiatives/orion-network-telescope/.

[90] Mirian, A., Ma, Z., Adrian, D., Tischer, M., Chuenchujit, T., Yardley, T., Berthier, R., Mason, J., Durumeric, Z., Halderman, J. A., et al. An Internet-wide view of ics devices. In *14th Annual Conference on Privacy, Security and Trust* (2016), IEEE.

[91] Moon, S.-J., Yin, Y., Sharma, R. A., Yuan, Y., Spring, J. M., and Sekar, V. Accurately measuring global risk of amplification attacks using {AmpMap}. In *USENIX Security Symposium* (2021).

[92] Moore, H. Fix a segfault when udp->uh_ulen is less than 8. https://github.com/zmap/zmap/pull/155.

[93] Morishita, S., Hoizumi, T., Ueno, W., Tanabe, R., Gañán, C., Van Eeten, M. J., Yoshioka, K., and Matsumoto, T. Detect me if you… oh wait. an Internet-wide view of self-revealing honeypots. In *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)* (2019).

[94] Moura, G. C., Ganán, C., Lone, Q., Poursaied, P., Asghari, H., and van Eeten, M. How dynamic is the ISPs address space? towards Internet-wide DHCP churn estimation. In *IFIP Networking Conference* (2015).

[95] National Cyber Security Centre. Active cyber defence: The sixth year. https://www.ncsc.gov.uk/files/ACD6-full-report.pdf.

[96] National Science Foundation. Internet measurement research: Methodologies, tools, and infrastructure (imr). https://new.nsf.gov/funding/opportunities/internet-measurement-research-methodologies-tools.

[97] Nemec, M., Klinec, D., Svenda, P., Sekan, P., and Matyas, V. Measuring popularity of cryptographic libraries in Internet-wide scans. In *Proceedings of the 33rd Annual Computer Security Applications Conference* (2017), pp. 162–175.

[98] Nguyen, T. T., Backes, M., and Stock, B. Freely given consent? studying consent notice of third-party tracking and its violations of gdpr in android apps. In *ACM Conference on Computer and Communications Security* (2022).

[99] Palo Alto Unit 42. New Mirai variant targeting network security devices. https://unit42.paloaltonetworks.com/mirai-variant-iot-vulnerabilities/.

[100] Pearce, P., Ensafi, R., Li, F., Feamster, N., and Paxson, V. Augur: Internet-wide detection of connectivity disruptions. In *IEEE Symposium on Security and Privacy* (2017).

[101] Pearce, P., Jones, B., Li, F., Ensafi, R., Feamster, N., Weaver, N., and Paxson, V. Global measurement of DNS manipulation. In *USENIX Security Symposium* (2017).

[102] Perino, D., Varvello, M., and Soriente, C. Proxytorrent: Untangling the free HTTP(S) proxy ecosystem. In *World Wide Web Conference* (2018).

[103] Perl, H., Fahl, S., and Smith, M. You won't be needing these any more: On removing unused certificates from trust stores. In *Financial Cryptography and Data Security* (2014).

[104] Perta, V. C., Barbera, M. V., and Mei, A. Exploiting delay patterns for user ips identification in cellular networks. In *Privacy Enhancing Technologies* (2014).

[105] Preston-Werner, T. The semantic versioning specification. *semver.org* (2011).

[106] Ramesh, R., Raman, R. S., Bernhard, M., Ongkowijaya, V., Evdokimov, L., Edmundson, A., Sprecher, S., Ikram, M., and Ensafi, R. Decentralized control: A case study of Russia. In *Network and Distributed System Security Symposium* (2020).

[107] Rodday, N., Kaltenbach, L., Cunha, I., Bush, R., Katz-Bassett, E., Rodosek, G. D., Schmidt, T. C., and Wählisch, M. On the deployment of default routes in inter-domain routing. In *ACM SIGCOMM 2021 Workshop on Technologies, Applications, and Uses of a Responsible Internet* (2021), pp. 14–20.

[108] Rüth, J., Poese, I., Dietzel, C., and Hohlfeld, O. A first look at quic in the wild. In *Passive and Active Measurement* (2018).

[109] Rye, E. C., and Beverly, R. Sundials in the shade: An Internet-wide perspective on icmp timestamps. In *Passive and Active Measurement* (2019).

[110] Salus, P. H. *A quarter century of UNIX*. ACM Press/Addison-Wesley Publishing Co., 1994.

[111] Sarabi, A., Jin, K., and Liu, M. Smart Internet probing: Scanning using adaptive machine learning. *Game Theory and Machine Learning for Cyber Security* (2021).

[112] Sattler, P., Zirngibl, J., Jonker, M., Gasser, O., Carle, G., and Holz, R. Packed to the brim: Investigating the impact of highly responsive prefixes on Internet-wide measurement campaigns. *Proceedings of the ACM on Networking*, CoNEXT3 (2023).

[113] Sommers, J., Durairajan, R., and Barford, P. Automatic metadata generation for active measurement. In *ACM Internet Measurement Conference* (2017).

[114] Song, G., He, L., Zhao, T., Luo, Y., Wu, Y., Fan, L., Li, C., Wang, Z., and Yang, J. Which doors are open: Reinforcement learning-based Internet-wide port scanning. In *IEEE/ACM 31st International Symposium on Quality of Service* (2023).

[115] Springall, D., Durumeric, Z., and Halderman, J. A. FTP: the forgotten cloud. In *IEEE/IFIP International Conference on Dependable Systems and Networks* (2016).

[116] Springall, D., Durumeric, Z., and Halderman, J. A. Measuring the security harm of TLS crypto shortcuts. In *ACM Internet Measurement Conference* (2016).

[117] Srinivasa, S., Pedersen, J. M., and Vasilomanolakis, E. Open for hire: Attack trends and misconfiguration pitfalls of IoT devices. In *ACM Internet Measurement Conference* (2021).
[118] Sullivan, G. A., Sippe, J., Heninger, N., and Wustrow, E. Open to a fault: On the passive compromise of TLS keys via transient errors. In *31st USENIX Security Symposium* (2022).
[119] Švenda, P., Nemec, M., Sekan, P., Kvašňovský, R., Formánek, D., Komárek, D., and Matyáš, V. The Million-Key question: Investigating the origins of RSA public keys. In *USENIX Security Symposium* (2016).
[120] Szurdi, J., and Christin, N. Email typosquatting. In *ACM Internet Measurement Conference* (2017).
[121] The White House. Back to the building blocks: A path toward secure and measurable software. Tech. rep., 2024. https://www.whitehouse.gov/wp-content/uploads/2024/02/Final-ONCD-Technical-Report.pdf.
[122] Toulas, B. Medusa botnet returns as a Mirai-based variant with ransomware sting. https://www.bleepingcomputer.com/news/security/medusa-botnet-returns-as-a-mirai-based-variant-with-ransomware-sting/.
[123] van der Toorn, O., van Rijswijk-Deij, R., Sommese, R., Sperotto, A., and Jonker, M. Saving brian's privacy: the perils of privacy exposure through reverse dns. In *ACM Internet Measurement Conference* (2022).
[124] VanderSloot, B., Amann, J., Bernhard, M., Durumeric, Z., Bailey, M., and Halderman, J. A. Towards a complete view of the certificate ecosystem. In *ACM Internet Measurement Conference* (2016).
[125] Vissers, T., Van Goethem, T., Joosen, W., and Nikiforakis, N. Maneuvering around clouds: Bypassing cloud-based security providers. In *ACM Conference on Computer and Communications Security* (2015).
[126] Wan, G., Izhikevich, L., Adrian, D., Yoshioka, K., Holz, R., Rossow, C., and Durumeric, Z. On the origin of scanning: The impact of location on Internet-wide scans. In *ACM Internet Measurement Conference* (2020).
[127] Ward, I. JSON lines. https://jsonlines.org/.
[128] Xu, Z., Wang, H., and Wu, Z. A measurement study on co-residence threat inside the cloud. In *USENIX Security Symposium* (2015).
[129] Yang, K., Li, Q., and Sun, L. Towards automatic fingerprinting of IoT devices in the cyberspace. *Computer networks* (2019).
[130] Zhang, J., Durumeric, Z., Bailey, M., Liu, M., and Karir, M. On the mismanagement and maliciousness of networks. In *Network and Distributed System Security Symposium* (2014).
[131] Zhu, L., Wessels, D., Mankin, A., and Heidemann, J. Measuring dane TLSA deployment. In *Traffic Monitoring and Analysis* (2015).
[132] Zirngibl, J., Buschmann, P., Sattler, P., Jaeger, B., Aulbach, J., and Carle, G. It's over 9000: Analyzing early quic deployments with the standardization on the horizon. In *ACM Internet Measurement Conference* (2021).
[133] Zyp, K. JSON schema. https://json-schema.org/.

A ETHICS

Our work is primarily a metareview of prior studies and a presentation of lessons learned from ZMap. When conducting our own experiments to validate changes made to ZMap, we followed the original guidelines set forth by Durumeric et al. in 2013 [39]. We provide updated recommendations on how to best conduct Internet scanning in Section 6.

B ACADEMIC USAGE OF ZMAP

Topic	Papers	Examples
Censorship and Anonymity	14	[42, 100, 101, 106]
Cryptography and Key Generation	17	[22, 52, 64, 119]
Denial of Service (DoS)	15	[20, 26, 45]
DNS and Naming	24	[75, 79, 131]
Email and Spam	8	[35, 57, 85, 120]
Exposure, Hygiene, and Patching	12	[21, 38, 40, 130]
Honeypots, Telescopes, and Attacks	9	[8, 41, 93]
IP Usage, DHCP Churn, Nand AT	10	[58, 70, 94]
Industrial Control Systems (ICS)	14	[30, 31, 90]
Internet of Things (IoT)	25	[12, 29, 82, 117]
Systems and Network Security	19	[51, 81, 125, 128]
PKI, Certificates, Revocation	28	[25, 37, 77, 103]
Power Outages and Grid Monitoring	4	[10, 17]
Privacy	5	[55, 104, 123]
QUIC	7	[67, 108, 132]
Routing, BGP, and RPKI	12	[5, 46, 56, 107]
Scanning and Device Identification	25	[6, 7, 112, 129]
TLS, HTTPS, and SSH	38	[2, 24, 57, 116]
Understanding Threat Actors	4	[83, 84]
Other Internet Measurement Topics	26	[50, 74, 88, 109]
Ethics Guidance Only (No ZMap Use)	53	[62, 66, 91, 98]

Figure 8: Academic Papers Built on ZMap Data—We manually investigated papers that cited ZMap or ZMap-derived datasets to understand what types of research studies have used ZMap data.