Lean On Me: Mining Internet Service Dependencies From Large-Scale DNS Data

Matteo Dell'Amico
matteo_dellamico@symantec.com
Symantec Research Labs

Leyla Bilge
leyla_bilge@symantec.com
Symantec Research Labs

Ashwin Kayyoor
ashwin_kayyoor@symantec.com
Symantec Research Labs

Petros Efstathopoulos
petros_efstathopoulos@symantec.com
Symantec Research Labs

Pierre-Antoine Vervier
pierre-antoine_vervier@symantec.com
Symantec Research Labs

ABSTRACT

Most websites, services, and applications have come to rely on Internet services (e.g., DNS, CDN, email, WWW, etc.) offered by third parties. Although employing such services generally improves reliability and cost-effectiveness, it also creates dependencies on service providers, which may expose websites to additional risks, such as DDoS attacks or cascading failures. As cloud services are becoming more popular, an increasing percentage of the overall Internet ecosystem relies on a decreasing number of highly popular services. In our general effort to assess the security risk for a given entity, and motivated by the effects of recent service disruptions, we perform a large-scale analysis of passive and active DNS datasets including more than 2.5 trillion queries in order to discover the dependencies between websites and Internet services.

In this paper, we present the findings of our DNS dataset analysis, and attempt to expose important insights about the ecosystem of dependencies. To further understand the nature of dependencies, we perform graph-theoretic analysis on the dependency graph and propose *support power*, a novel power measure that can quantify the amount of dependence websites and other services have on a particular service. Our DNS analysis findings reveal that the current service ecosystem is dominated by a handful of popular service providers—with Amazon being the leader, by far—whose popularity is steadily increasing. These findings are further supported by our graph analysis results, which also reveals a set of less-popular services that many (regional) websites depend on.

1 INTRODUCTION

The Internet was designed to be fault-tolerant, distributed, and resilient. Some of the core services developed on top of it share the same design principles. The DNS, the WWW, and email have been designed so as to provide the assurances of a highly distributed infrastructure, with the ability to isolate failures, contain their effects and, eventually, recover from them.

Maintaining such highly reliable Internet ("layer 7") services is technically challenging, costly, and crucial for business continuity. With more and more of our day-to-day business and personal activity taking place online people have come to rely on the infrastructure being robust and available. Even the smallest amount of downtime may significantly damage a company, both financially and in terms of brand name. The opportunities for failure are numerous: hardware failure, software misconfiguration, service mismanagement, human error, malicious activity (such as intrusions and DoS attacks), and natural disasters are just a few examples of the things that can go wrong. In order for any company to achieve the expected level of availability and capacity, a great amount of resources need to be invested for building and maintaining each and every one of the services needed.

This problem became apparent early on as Internet services became popular and every company was expected to have an Internet presence. Service hosting providers filled the gap of maintaining such services, by providing DNS/WWW/email service hosting for their subscribers. The economy of scale made it possible for such services to be run in datacenters with high (or higher) availability guarantees. Even though some amount of service consolidation was introduced, thus making the overall Internet service landscape less distributed and less fault tolerant, the large number of distinct service providers ensured that failures were somewhat contained within the set of customers of a given provider.

The emergence of cloud services changed the service provider landscape yet again. The SaaS and IaaS cloud offerings enabled customers to shift more and more of their infrastructure and services to third-party cloud providers, capable of catering to the customers' ever increasing demand for high availability, high performance, professional management, and cost efficiency. As the cloud service provider landscape is maturing, we observe that only few "big players" are (1) able to expand their portfolio of services to include more functionality appealing to their customers, while being (2) able to survive the fierce competition involved in running such costly large-scale operations at a profit. Consequently, a small number of service providers own large portions of the market, thus becoming stronger, bigger, which in turn enables them to attract more customers. The sophistication of such service providers may make it less likely for failures to occur, but when they do happen—by mistake or malice—the results can be catastrophic, and their impact can be felt throughout the whole Internet. As more and more companies and individuals depend on a limited number of

ACSAC 2017, San Juan, PR, USA

978-1-4503-5345-8/17/12…$15.00
DOI: 10.1145/3134600.3134637

service providers for critical services, some of the fundamental design properties of Internet services are being put to the test and one cannot help but wonder: *what are the service dependencies within the service ecosystem today*, and *how do they affect the risk to users introduced by relying on such "single points of failure"?*

1.1 When Things Go Wrong—Motivation

Our work is motivated by (1) a number of recent and older incidents that have demonstrated the reliance of Internet users on certain service providers, (2) the increased frequency of such attacks—that are "easy to set up, difficult to stop, and very effective" [1], and (3) our overarching goal of devising methods to assess the security risk associated with such dependencies. In the past, we have witnessed a number of DDoS or other attacks on websites and Web applications that have resulted in partial or complete interruption of the service in question (e.g., daily attacks on the Sony Playstation network [5], Microsoft Xbox [48], major international banks [31], etc.). Similarly, there have been numerous cases of failures due to non-malicious events, such as human error or hardware failures. Such failures, however, are often qualitatively different from failures to global-scale Internet services. If a Web application (such as a social media platform, or a search engine) becomes unavailable, the users of that particular application are inconvenienced, but not deprived of a core Internet service (such as DNS) whose interruption may have cascading effects (by incapacitating additional services) that may cause a widely-felt disruption. As more functionality is consolidated on less service providers, not only do the effects of a potential service interruption become more severe, but the service provider itself becomes a more appealing target for malicious actors and coordinated large-scale attacks. Therefore, assessing the security risks of a particular organization can benefit directly from profiling the services it relies on, their dependency chains, and their place/role in the service ecosystem.

In recent time we have witnessed some alarming examples of the effects failures in major service providers may have. Recent large-scale DDoS attacks on OVH, one of the worlds largest hosting companies [36], reached a rate of nearly 1 Tbps, showing how attackers can direct significant "firepower" towards their victims—enough to seriously disrupt the services provided by the hosting provider to its customers (in OVH's case more than 50,000 customers in North America alone). A good example of the inter-dependencies between services and the cascading effects of such attacks, is the results of similar DDoS attacks to individual websites, such as a security expert's personal blog [29] and BBC [27] (both of which exceeded 600 Gbps of attack traffic). On the surface, one would not expect such attacks to cause Internet-scale service disruption—aside from their effects on the target websites and despite launching hundreds of Gbps of attack traffic. This would be true if it wasn't for the fact that such websites rely on cloud services that can be significantly affected by the targeted attack. For instance, many popular websites rely on Content Distribution Networks (CDNs) for their services which may or may not be able to sustain the collateral damage inflicted on them by the attackers. As such, the original attack may have significant cascading effects to the other services that rely on the same CDN. In the case of [29] Akamai had to stop serving the website under DDoS, despite their effort to mitigate the attack.

More alarmingly, aside from such indirect effects of attacks on services and their users, we have recently witnessed serious DDoS attacks towards core Internet service providers themselves. For instance, the attack on Dyn [30], a popular DNS service provider, caused significant disruption to a wide variety of other websites and services—including Twitter, Spotify, Netflix, Amazon, Tumblr, Reddit, PayPal, and others—thus demonstrating the risk associated with service dependencies. Many websites, services, and users not relying directly on Dyn were affected as a results of the cascading failure effect. Not surprisingly, the Dyn attack affected a major CDN as well: the Cloudflare [40] CDN service, under a certain configuration (using `CNAME` rather than `A` records for its customers) requires to perform a DNS query to resolve the IP address of the origin server. When Dyn was rendered inaccessible, the Cloudflare CDN suffered a cascading failure, observed on (almost) the entire planet [21]. As mentioned in Cloudflare's relevant blog post, "the Internet is very complex and [...] the devil is in the details". These (dependency) details are what we set out to discover, understand, and expose in this work through DNS data analysis.

Another incident that illustrates both the strengths and weaknesses of service dependencies is the attack that was launched against the Spamhaus non-profit anti-spam organization. In 2013, Spamhaus was targeted by more than 300 Gbps of DDoS traffic [39, 46]. Initially, Cloudflare (which provides a DDoS mitigation service) successfully mitigated the first waves of the attack, thus demonstrating the benefits of inter-connected services and how this can help mitigate large scale attacks. In the next phase of the attack, however, attackers targeted networks and ISPs Cloudflare relies upon to operate, resulting in intermittent large-scale Internet outages affecting parts of Western Europe. This demonstrates how service dependencies (1) provide more avenues for attackers to disrupt the operation of their target, and (2) create more opportunity for collateral damage—in this case the rest of Cloudflare's customers, as well as the ISPs' customers.

Lastly, one more incident highlighting the cascading failure effect occurred recently, not because of malicious activity, but due to human error. The Amazon S3 failure in January of 2017 [14] briefly demonstrated how many popular services have come to rely on a few big cloud service providers for many of their Internet service and infrastructure needs. Such incidents point to the additional observation that when assessing an organization's exposure to risk it may not be sufficient to assess the organization's direct dependencies (e.g., service or supply chain dependencies), but one may need to dig deeper into that organization's dependency chain.

1.2 Our Goals and Contributions

Motivated by (1) the impact and seriousness of recent attacks to SaaS and IaaS Cloud providers (as well as failures due to non-malicious events), and (2) by the increasing adoption and popularity of such services, we set out to investigate the dependencies between websites and (Cloud) services. Identifying such relationships will provide valuable insights for our overarching effort to assess and quantify cyber risk for organizations. More specifically, our goals include the following:

Identify dependencies between websites and services.

Develop techniques that are broadly applicable, require little or no service-specific knowledge, and scale well.
Analyze discovered dependency information—e.g., popularity analysis, failure effects, temporal trends, etc.—and extract useful/actionable insights.

To the best of our knowledge, this is the first study that 1) attempts to discover such service dependencies through analysis of DNS datasets, and 2) studies the discovered relationships using graph-theoretical analyses.

Non-Goals. We aim to discover dependencies among websites and services using scalable analysis of DNS datasets, to the best of our ability, with no privileged access to internal information. Given these assumptions, we aim to get the best possible coverage. It is not, however, our goal to achieve complete coverage of all dependencies on the Internet, or of the full nature of such relationships. In fact, we are certain that many types of dependencies may not be externally observable without privileged access, or not visible in DNS data. Furthermore, we aim to explore and expose some such dependencies, but, at the moment, we do not aim to propose methods to reduce one's risk due to such dependencies. Lastly, we do not aim to discourage any entity from using services that we find to be of critical role to the ecosystem—we only intend to expose the current dependency structure for the purposes of risk assessment, thus enabling users to make well-informed decisions about service usage (e.g., discover single-points of failure). After all, we all need somebody to lean on.

The effects of the discovered dependencies on risk, and how to mitigate them, is part of ongoing efforts and is left as future work.

Our contributions. We make the following contributions:

We perform a wide-scale analysis of active and passive DNS datasets for the Alexa top 1 million domains in order to identify dependencies on service providers. Our method does not rely on any special knowledge or access to any website or service, and is based entirely on DNS data.
As a result of this analysis we are able to quantify statistics about service dependencies, confirm common intuition in certain cases, and extract new insights in others.
Our results show that service dependencies are converging towards a few very important service providers—most notably, we found evidence that more than half of the top 1M domains and more than 90% of top 1K domains use Amazon services. Furthermore, this phenomenon—whereby the Internet appears to lose part of its decentralized nature—appears to become stronger over time.
We perform graph analysis on Internet service dependency graph data and report top-k important services based on various graph theoretical centrality measures. Results through graph analysis further corroborate with the findings obtained through other analysis methods in this paper.
Finally, we introduce a novel power measure namely *support power*, so as to quantify the extent to which websites and other services are dependent on a particular service.

The rest of the paper is organized as follows. Section 2 presents the datasets analyzed for the purposes of this study, while Section 3 discusses the methods we employed for our analyses. Section 4 presents the findings of both the DNS analysis (4.1) and the graph-based analysis (4.2). Finally, Section 5 presents related work.

2 DATASETS ANALYZED

Our analysis combines DNS data collected both actively and passively with a few additional datasets. The characteristics of these datasets are as follows.

2.1 Non-DNS Data Sources

To avoid including results about irrelevant or marginally used parts of the Internet, we use Alexa's top 1 million domain list:[1] our analysis focuses on the domains included in this list. To map IP addresses to domains, we use the reverse DNS data from Project Sonar [41].[2] The Public Suffix List available at https://publicsuffix.org is a list of domains "under which Internet users can (or historically could) directly register names", such as `.com` or `.co.uk`. We use this list[3] to recognize the entity responsible for a given domain name (e.g., `google.com` or `amazon.co.uk`). The "private domains" section of the Public Suffix List also provides a list of domains that can be obtained through the services of private companies rather than registration authorities (e.g., `*.s3.amazonaws.com`). As we describe in more detail in the following, we use this part of the dataset to infer that some domains pertain to the same service (e.g., `*.s3.amazonaws.com` and `*.s3-us-west-1.amazonaws.com` all pertain to Amazon S3).

2.2 Passively Collected DNS Data

We obtained a very large dataset of passively-collected DNS data, consisting of 2.5 years of DNS query logs from a large service provider, between November 2014 and April 2017, for an average of 85 billion queries per month and a total count of 2.5 trillion DNS query results.

From the record types stored in the DNS datasets, we are interested in data objects having record type `A` (domain to IPv4 address, which we map back to domains using reverse DNS information), `CNAME` (domain to canonical name, used to declare that a domain name is an alias of another one), `MX` (mail exchange, to find out which mail server handles email for a domain), `NS` (delegating a zone to an authoritative name server). `AAAA` (domain to IPv6 address) records are few in our dataset; because of the limited information that can be gleaned from it, and because we did not have access to reverse DNS scans of the IPv6 space, we do not include `AAAA` fields in our analysis.

From our dataset, for each `A`, `CNAME` and `NS` query, we use the query domain and the response; to enable longitudinal analysis we split our dataset in five 6-month periods. After removing duplicates, we ended up with a total of 38.3B records, (36.8B `A`, 1.4B `CNAME` and 167M `NS` records). The predominance of `A` records in this dataset is interesting for two reasons: first, they represent a large majority of the whole dataset (around 96%); hence, being able to use this data for our purposes through reverse DNS information is key. Second, after removing duplicates, the number of ⟨domain, IP address⟩ pairs we

[1] At aws.amazon.com/alexa-top-sites/ there are no more download links, but at the time of writing the list is still available at s3.amazonaws.com/alexa-static/top-1m.csv.zip.
[2] https://scans.io/study/sonar.rdns_v2
[3] https://publicsuffix.org/list/public_suffix_list.dat

have is almost 10 times larger than the size of the IPv4 address space (4.3B). This shows that it is very common for a machine/IP address to respond to multiple FQDNs, hence making this DNS analysis a good mechanism for us to identify such cases of particular interest to our investigation.

Finally, we have observed that the active DNS measurement dataset contains more reliable MX information. In fact, MX records are less commonly queried than other record types. Therefore, passive DNS datasets are likely to provide (too) low MX record coverage. Furthermore, the lower number of MX records, compared to the other DNS record types, makes active measurements more scalable. For these reasons we collected MX information using the methods described in the following.

2.3 Actively Collected DNS Data

The passive DNS dataset is very rich; however, it largely consists of A and CNAME records (NS and MX queries are definitely less); moreover, the number of queries is heavily skewed towards few important domains. For these reasons, we supplemented the data gathered passively with data that we queried actively. In particular, for each domain domain.com in the Alexa top 1M dataset, we used the dig Unix tool to query both domain.com and www.domain.com for the four record types we use. We repeated these queries monthly between January and May 2017.

We found out that this dataset is only marginally useful in learning new data from A and CNAME fields: they are definitely better covered through the passive DNS logs. On the other hand, the information about NS and MX records for less popular domains is rather scarce in the passive DNS data, and in this respect the active measurement shines: in conclusion, both active and passive DNS data are essential for this analysis.

It is noteworthy that to limit the footprint and the burden induced on DNS servers by our active measurements to the lowest possible, we only perform two queries for each domain in the Alexa top 1M list, and we do this only once a month.

3 ANALYSIS AND METHODS USED

Our analysis takes two steps: first, we construct (and analyze) a mapping of services used, based on the intuition that a DNS record mapping x.domain.com to y.service.com is a fairly reliable evidence that domain.com uses the service provided by service.com. Then, we perform a graph-based analysis where domains are nodes and the edges represent relations of service usage (in the previous example, there would be a directed edge from domain.com to service.com).

3.1 DNS Data Analysis

The main input to this analysis is the set of DNS records collected through the active and passive measurement methods described in Section 2.

While MX, CNAME and NS records consist of fully qualified domain names (FQDNs),[4] A records (which, as noted in Section 2.2, are a large majority of our dataset) contain IP addresses. We replace each IP address with the FQDN(s) obtained for that IP through reverse DNS.[5] After this step, for each record type, we obtain a set of $\mathrm{FQDN}_q; \mathrm{FQDN}_r$ mappings, where FQDN_q is the queried domain and FQDN_r is obtained from the response (indirectly through reverse DNS for A records, as described above).

We map each FQDN_q to the corresponding domain in the Alexa Top 1M domain list; as argued in Section 2.1 we discard all information regarding other domains.

The last part of this procedure consists in mapping FQDN_r to a service. We use 3 primary fields to describe each service: category, provider domain and service name.[6] For example, Amazon S3 has category "cloud", provider domain amazon.com and service name "Amazon S3". We have five categories: Cloud, CDN, DNS, Email, and ISPs – the ISP category covering mostly hosting and carriers' domains. To perform this mapping, we set up a tree structure mirroring the parts of the DNS hierarchy we have information on: (1) which domains are public (as obtained from the Public Suffix List); (2) which domains correspond to a known service.

The information about known private services is bootstrapped with the private section of the Public Suffix List and a list of known CDN domains;[7] we then update it manually according to the results of a first run of this analysis. We map FQDN_r values to services by matching the longest possible domain suffix with tree elements. If we end up in a node corresponding to a known service, we return it; otherwise, we return as provider domain and service name the shortest private suffix of the name (e.g., x.domain.co.uk is mapped to domain.co.uk), with a null value as category.

A first run of the analysis returns a large set of domain to service mappings where the service category is unknown. We take into account the services that are used by several domains (we use as threshold the services used by at least 0.1% of the domains we are examining), and manually create rules to categorize this. We end up with a set of 243 manually-written rules, which can triage a large set of the customer-to-provider relationships we discover in our study.

Caveats. Reverse DNS configuration errors can sometimes result in erroneous results. In particular, we have found errors in the reverse DNS configuration of some Akamai CDN servers, for which the reverse DNS points to domains like rr.com (an ISP) or even mwsco.com (a welding supply business). We have detected those errors because of (1) an abnormally high number of domains using this service; (2) the fact that we detect essentially all these domains as Akamai customers. We then confirmed that those were in fact Akamai servers by verifying that they indeed serve web pages served by the Akamai network; subsequently, we removed those results from our analysis. Such reverse DNS errors can be explained by the fact that an important part of the value proposition of CDNs is, in fact, placing servers at the edge of the network, in parts that are possibly administered by other actors. Alternatively, such anomalies may simply be due to IP address reuse and improper reverse DNS maintenance (e.g., IP address previously belonging

[4] MX records also contain a priority value that we ignore for our purposes.

[5] IP addresses mapping to multiple FQDNs through reverse DNS records are mapped to all of them.

[6] We also use an optional "subcategory" field, where applicable—e.g., Amazon S3 has subcategory "storage".

[7] github.com/WPO-Foundation/webpagetest/blob/master/agent/wpthook/cdn.h

to one organization being reassigned to a CDN without proper updates to rDNS).

In some cases, reverse DNS does not provide enough granularity to distinguish the particular service given: for example, Google uses `1e100.net` as reverse DNS for all of its services [19]. Obtaining such a result through our reverse DNS information for A records allows us to conclude that a domain is using *some* Google service, but we do not have enough information to know *which one*.

3.2 Graph-Based Analysis

From the service mapping obtained in the previous step, we create a graph $G = (V, E)$ connecting domains, in such a way that a directed edge exists between domains x and y if x uses a service provided by y. We create this graph to enable graph-based analysis that can lead to the discovery of important artifacts—such as chain failures that can be seen as a path in the graph—as well as the application of graph-theoretical tools and epidemic modeling. Indeed, in Section 4.1.4, we have results that suggest that important chain failure events have already been observed in the wild, thus underlining the cascading nature of such failures. Results for our graph-based analysis are in Section 4.2.

To understand the nature of cascading or chain failures, we discuss a set of key metrics. First, we introduce a new metric to quantify the extent to which nodes in the graph are dependent on a particular node. Second, we reason about failure cascades to dependent services using epidemic modelling of the service dependency graph data.

Support Power. In graph theory, the concept of *centrality* and *power* have a peculiar relationship. The traditional degree centrality approach argues that nodes that have more connections are more likely to be powerful because they can directly affect more other nodes. This makes sense, but having the same degree does not necessarily make nodes equally important. If, on the other hand, the set of nodes $V' \subseteq V$ to which a particular node $v \in V$ is connected are not, themselves, well connected, then nodes in V' are dependent on node v. Phillip Bonacich [8] argued that being connected to connected others makes a node *central*, but not powerful. Somewhat ironically, being connected to others that are not well connected makes one *powerful*, because these other nodes are dependent on you – whereas well connected nodes are not. Bonacich proposed that both centrality and power were a function of the connections of the set of nodes in one's neighborhood. The more connections the nodes in node v's neighborhood η_v have, the more central the node v is. Fewer the connections to the set of nodes $V' \subseteq V$ in a node v's neighborhood η_v, the more powerful the node v is. Building on Bonacich's intuition of power for a particular node u, we observe that, in addition to the fewer connection of node u's dependants, node u's lower dependency on other nodes also makes it more powerful. Also, we believe that node u's power further increases when many nodes (recursively) depend on u's dependants. To take these additional observations into consideration, and inspired by the concept of *Bonacich Power*, we introduce a new power metric that we call *support power*.

More intuitively, the *support power* SP_v of a particular node v is higher if nodes that depend on v also depend on few or none other nodes, and/or if nodes depending on v themselves support a large number of other nodes. Here, if there is a directed edge from u to v then node v is said to be supporter of node u and node u is said to be dependent on node v. More formally,

$$SP_u^{1in\ d^0} = \frac{\prod_{v \in \eta_u} d^{\ 1}v^0}{\prod_{v \in \eta_u^+} d^{+1}v^0} \qquad (1)$$

where $d^{\ 1}u^0$ and $d^{+1}u^0$ are the in-degree and out-degree of node u. η_u and η_u^+ are the set of nodes that are incident to and reached from node u. Since the factor $\prod_{v \in \eta_u} d^{\ 1}v^0$ only considers dependency of up to 2-hop neighboring nodes. To consider chains of dependencies greater than 2-hops leading up to a particular node u, instead of in-degree values, we can rather consider the pagerank (pr) or eigenvalue centrality (ec) values of the relevant nodes. More formally, the support power of a node u considering pagerank $SP_u^{1pr^0}$ and eigenvector centrality scores $SP_u^{1ec^0}$ can be given by:

$$SP_u^{1pr^0} = \frac{\prod_{v \in \eta_u} pr^1v^0}{\prod_{v \in \eta_u^+} d^{+1}v^0}; \quad SP_u^{1ec^0} = \frac{\prod_{v \in \eta_u} ec^1v^0}{\prod_{v \in \eta_u^+} d^{+1}v^0}$$

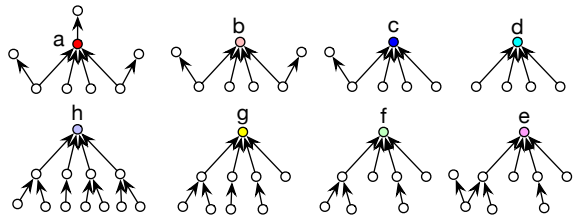

Figure 1: Example showing support chains for different nodes $a; b; c; d; e; f; g$ and h in the graph.

Overall, in the context of this paper, sorting the nodes or services by their support values, in descending order, gives us a ranking of the domains/services that are (1) powerful, and (2) solely support dependency chain of domains/services that, mostly, do not depend on other services. Figure 1 shows examples of support chains for different nodes $a; b; c; d; e; f; g$ and h in the graph. According to the definition of *support power*, we have $SP_h > SP_g > SP_f > SP_e > SP_d > SP_c > SP_b > SP_a$. Here, it is worth noting that node b and a have support chains leading up to them that are similar but node b is more powerful than node a because it is not dependent on any other node unlike a. On the other hand, node c is more powerful than node b, as dependency on c by other nodes is stronger when compared to dependency of other nodes on b, and so on and so forth. Dependency is said to be stronger, if nodes dependent on a particular node are less dependent on other nodes.

Nodes with high support power have high centrality (i.e., they are used by many) *and* they are heavily used by domains with fewer dependencies. Consider nodes b and d in Figure 1, where d has higher support power: similar to the Spamhaus case reported in Section 1, if an attacker wanted to cause a chain failure a node dependent on d, attacking d is their only option. Conversely, b is not the only avenue for attacks if one wants to disrupt one of their dependent nodes. In other words, by virtue of being the single point of failure for many services, nodes with high support power are more in danger of being attacked.

Epidemic Modeling & Degree Distribution. Cascading failures of Internet services can be modeled with tools similar to epidemic modeling. Indeed, epidemics can spread through the human contacts network similarly to the way a failure can spread through the network of dependencies between services: like an infected individual spreads a contagion to people that get close to it, a failure cascades to dependent services.

Many epidemic models reason in terms of an *epidemic threshold*, which is a contagion probability that marks a phase transition: below the epidemic threshold epidemics die out quickly; above it, they can reach a constant fraction of the whole population. One key result proven by Pastor-Satorras and Vespignani [37] is that scale free networks —i.e., those whose degree distribution follows a power law—have *no epidemic threshold*: no matter the probability of contagion, all epidemics can become widespread. Pastor-Satorras and Vespignani have shown that this pessimistic result applies to the propagation of malware over the Internet and generalized it to drop assumptions like very large network sizes [38] and absence of connectivity correlations [7]; as we show in Section 4.2, our graph has a power-law degree distribution as well, and hence this result also applies to it.

4 RESULTS

4.1 DNS Analysis Findings

We present the results of our DNS dataset analysis as it pertains to the services used by the Alexa top 1M domain names. While the active DNS dataset was collected since January 2017, our passive DNS dataset covers queries made by tens of millions of real users since November 2014. The 2.5-year-long passive DNS data allowed us to experiment with periods of varying lengths and find out whether there are dramatic differences in the popular services used over time.

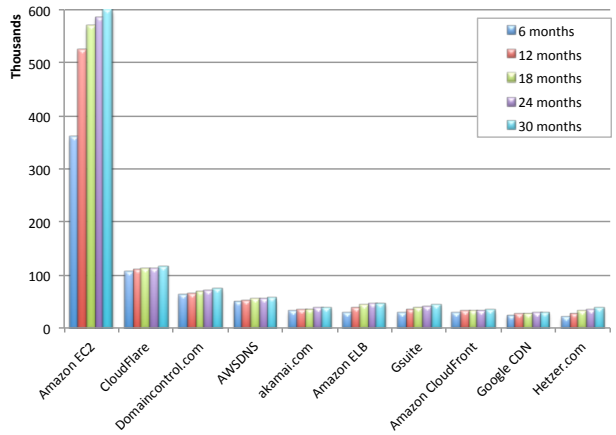

Figure 2: Number of dependencies that can be discovered using data from periods of varying time lengths.

In Figure 2, we compare the results obtained using different lengths of passive DNS traces (selected such that in all cases the traces are collected until April 30 2017). These results suggest that increasing the time window observed provides better coverage in understanding which services are used by companies. While expanding the data coverage does not have a drastic impact on the number of customers for some of the services (e.g., Cloudflare, AWS DNS), for many others we see a great improvement. For example, increasing the data length from 6 months to 30 months lets us discover 41% more Amazon EC2 and 39% more Amazon Elastic Load Balancing customers. This motivates our choice of presenting results that are derived from the more detailed dependency relationship analysis that makes use of the entirety of the passive DNS data.

4.1.1 **Per Service Category Analysis.** In Table 1, we list the top 5 service providers in the cloud, email, ISP, CDN and DNS categories, as derived by their popularity with the Alexa top 1M and 1K domains. We also present results for 961 US government domains (identified by the `.gov` TLD), as they represent a class of domains/services that can be highly sensitive to factors that may increase their risk—such the dependencies we are investigating. Amazon appears to be the top cloud service provider, having customers from 60% of the Alexa top 1M, 94% top 1K and 93% of the US `.gov` domains. These numbers indicate that Amazon is going towards a monopoly on the cloud computing market. This can be seen as scary, as in an unfortunate event of a problem at Amazon, a very big percentage of Internet could be impacted resulting in a catastrophic event.

The most used CDN providers for Alexa top 1M/1K and `.gov` domains are nearly the same while the order is slightly different. Akamai is the top CDN provider, followed closely by Cloudfront, Google CDN and Cloudflare. While among the top 1M domains, these four CDNs have approximately the same amount of customers, among the top 1K almost half of them and among `.gov` domains 20% of them uses Akamai CDN services. On the other hand, the top providers in the managed DNS providers category for Alexa top 1M/1K and `.gov` domains are very different. While Cloudflare is the lead DNS provider for the Alexa top 1M, only 16% of the Alexa top 1K uses Cloudflare, which makes it the third popular DNS provider. Amazon DNS has 30% of the top 1K domains as customer and only 6% of the top 1M.

Gmail is the most used e-mail services according to our data. 10% of the Alexa top 1M and 19% of the Alexa top 1K domains employ google mail. However, `.gov` domains prefer `outlook.com` over it.

4.1.2 **Trends.** The most visible change observed since the end of 2014 is the dramatic increase in the number of Amazon EC2 customers. As clearly witnessed in Figure 3, in April 2017 Amazon EC2 had around twice the number of Alexa top 1M customers than two years before. This finding is interesting since EC2 growth appears to outpace the already optimistic forecasts of 2011 [25], predicting that cloud-related spending would triple in 6 years, by 2017. More broadly, we observe that Amazon appears to be largely dominating the market, in general. In Figure 4 we show the percentage of companies in Alexa top 1M, Alexa top 1K, and the US government domains that use *only* Amazon cloud services. While at the end of 2014 only ~12% of Alexa 1M domains used exclusively Amazon cloud services, that number more than doubled to ~30% by 2017.

There are other interesting changes we see over time:

Table 1: Top 5 services per category in Alexa top 1M, top 1,000, and .gov domains.

#	Cloud		CDN		DNS		Email		ISP	
					Alexa Top 1 Million					
1	Amazon EC2	60%	Akamai	4%	Cloudflare	11%	Gmail	10%	hetzner.com	4%
2	Amazon ELB	5%	CloudFront	3%	DomainControl	7%	Secureserver.net	10%	linode.com	3%
3	GSuite	4%	Google CDN	3%	AWS DNS	6%	outlook.com	4%	endurance.com	2%
4	Office365	2%	Cloudflare	2%	DNSMadeEasy	2%	Yandex	2%	centrulink.com	2%
5	Amazon S3	2%	Incapsula	1%	DNSPod	1%	qq.com	1%	teliacarrier.com	2%
					Alexa Top 1,000					
1	Amazon EC2	94%	Akamai	46%	AWS DNS	30%	Gmail	19%	teliacarrier.com	39%
2	Amazon ELB	35%	CloudFront	30%	dynect.net	17%	sendgrid.net	10%	xo.com	36%
3	GSuite	17%	Google CDN	15%	Cloudflare	16%	secureserver.net	10%	centurylink.com	35%
4	Amazon S3	15%	fastly.net	12%	akadns.net	14%	outlook.com	5%	pccwglobal.com	33%
5	AWS other	11%	edgecast.com	12%	ultradns.com	9%	psmtp.com	2%	verio.com	33%
					.gov domains					
1	Amazon EC2	93%	Akamai	20%	akam.net	8%	outlook.com	18%	centurylink.com	13%
2	Amazon ELB	18%	CloudFront	5%	akadns.net	8%	Gmail	6%	zayo.com	10%
3	Office 365	14%	Cloudflare	5%	AWS DNS	7%	secureserver.net	6%	xo.com	10%
4	lync.com	12%	Google CDN	3%	domaincontrol.com	5%	sendgrid.net	2%	verio.com	10%
5	GSuite	6%	Incapsula	2%	dhhs.gov	4%	pphosted.com	2%	above.net	9%

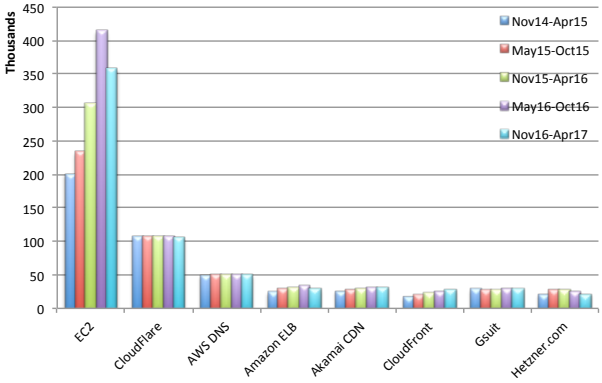

Figure 3: Most popular services and their usage over time.

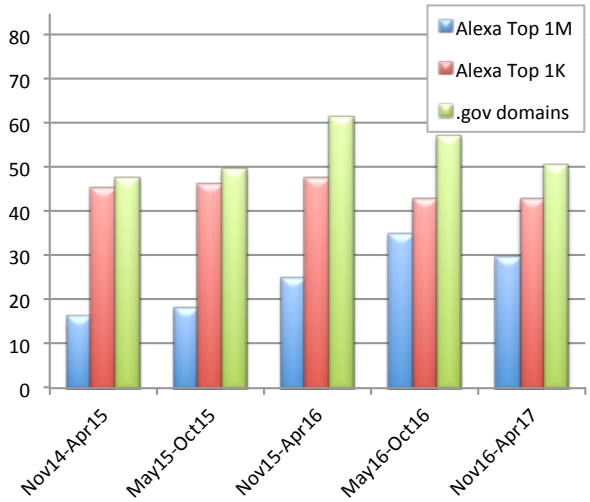

Figure 4: Percentage of companies that use *only* Amazon cloud services

Amazon EC2 customers from Alexa top 1000 increased from 80% to 90%.
The Google CDN gained popularity among the Alexa top 1000 domains (from 9th to 3rd position), but lost popularity in general.
Amazon DNS services increased their customer base in the US government.
For US government domains, the usage of Amazon EC2 increased by 20%, reaching 85% by the beginning of 2017.

4.1.3 **Correlations in Service Usage.** Across our results we notice that there are sets of correlated services – meaning that domains that use a service are more likely to also use another. We have taken into account all services used by at least 2% of the domains in our dataset, and for each of those services we have built a boolean vector where the i^{th} element is true if the i^{th} domain uses the service. We then computed the Pearson correlation coefficient between all services, and the highest and lowest correlation coefficients are reported in Table 2. It is clear that several domains tend to use several services from the same provider at once – for example, there's high correlation between using `domaincontrol.com` and `secureserver.net` which are, respectively, Go Daddy's DNS and email services. We can draw similar conclusions with services provided by Amazon and Google. Albeit correlation coefficients are lower, we can also observe that there are negatively correlated

Table 2: Positively and negatively correlated services.

Category/Service 1	Category/Service 2	Coef.
DNS/domaincontrol.com	Email/secureserver.net	0.57
DNS/AWS DNS	CDN/Amazon CloudFront	0.41
Cloud/Amazon ELB	CDN/Amazon CloudFront	0.28
Cloud/Amazon ELB	DNS/AWS DNS	0.25
Cloud/GSuite	Email/Google Mail	0.22
	...	
Email/Gmail	Email/secureserver.net	-0.06
Email/Gmail	Email/Outlook	-0.06
DNS/Cloudflare	Email/Outlook	-0.07
DNS/Cloudflare	Cloud/Amazon EC2	-0.10
DNS/Cloudflare	Email/Gmail	-0.11

services – for example, users of Cloudflare's DNS solutions are less likely to use services such as Gmail, Outlook or Amazon EC2.

4.1.4 **Case Studies.** As previously mentioned, we have witnessed some recent and massive failure events due to outages at heavily-used services. Two such examples are the Dyn DNS outage in October 2016, and the Amazon S3 outage in February 2017. In addition to being important events that motivate our study, they offer an opportunity for us to investigate the results that our service mapping obtained for these events.

October 2016: Dyn DNS Outage. On October 16, 2016, a very large botnet based on the Mirai malware attacked Dyn, a very popular DNS provider. As a result of the attack, several important Internet services – including Twitter, Spotify and Reddit – were impacted [30]. From the Wikipedia page about the event,[8] we collected a list of 70 services and websites affected by the outage, and compiled a list of their domains. We then found that our mapping discovers that exactly half of them (35) use or used Dyn's DNS. The remaining 35 domains are all in the Alexa top 1M domains, and hence our analysis includes active NS queries issued directly to them as discussed in Section 2.3, which *did not* return Dyn domains. We also verified with historic data that those domains didn't change DNS provider between the incident data and the moment in which we performed data collection. Hence, we speculate that these domains were actually relying on Dyn *indirectly*, by depending on other services that were Dyn's customers. In other words, this was a case of a cascading failure, in which the failure of a service (Dyn) severely impacted *the customers of their customers*. A similar pattern was documented by Cloudflare [16], when failures in a backbone provider caused errors for Cloudflare's customers.

It would be desirable to obtain more insight on *which* services' failure have caused the cascading failures referred to above, but this is made difficult by the fact that 1) by the way our analysis is designed, not all service dependencies are designed, 2) we don't have a comprehensive list of services affected by the outage, and 3) for a popular service like Dyn (cf. Table 1), several paths in the service dependency graph can be used as possible explanation to the failure. We regard this as a potential analysis that can be carried out in future work.

[8] https://en.wikipedia.org/wiki/2016_Dyn_cyberattack

February 2017: Amazon S3 Outage. On February 28, 2017, a configuration error caused a disruption in the service of Amazon S3, leading once again to "partially or fully broken" service on several important websites and services [14]. As Amazon reports, this failure resulted in chain failures on other Amazon services that depend on S3, like the Elastic Compute Cloud (EC2), Elastic Block Store (EBS), and AWS Lambda [4]. Similar to the Dyn event, we found media coverage reporting a list of external services affected by the outage [34], and obtained a list of 74 affected domains.

Unlike DNS, detecting whether a given company is using Amazon S3 is not easy, as usage of the service is not necessarily reflected in DNS data; still, for convenience, companies sometimes insert DNS records for their domains that point – through A and CNAME records – to Amazon S3 hosts, allowing our method to recognize the usage of the S3 service.

In this case, we discovered that 21 (28%) of the domains affected by the outage use Amazon S3, and that other Amazon services were heavily used within the set of affected domains: 70 (95%) use Amazon EC2; 48 (65%) use Amazon Elastic Load Balancing; 41 (55%) use Amazon's AWS DNS solution; 39 (52%) use Amazon's CloudFront CDN. We interpret these results in the following way:

(1) Even for services where our method can't guarantee high coverage, such as S3, we are still able to identify a significant subset of domains using them. This allows us to achieve a 25-30% "direct" detection rate—even without considering the cascading failure effect. If, by extrapolating from the Dyn case study, we assume a 50% cascading failure rate then we could report a failure detection rate of about 50-60%.
(2) Confirming what we learned in Section 4.1.3, companies are likely to use several services at once from the same provider, as demonstrated by the large level of dependency on Amazon for the affected domains. This provides us with a good reason to believe, with high confidence, that if a given domain is, for example, found to be using Amazon's DNS and CDN, it is also more likely to use Amazon's S3 (or other services).

4.2 Graph Analysis Insights

4.2.1 **General Statistics.** We begin our analysis of the service dependency graph according to the method of Section 3.2 by discussing general graph statistics. This graph is built using data collected over a period of two and a half years (Nov 2014 - Apr 2017), is directed and consists of about 1.7 million nodes (services) and 5.2 million edges. This graph has 4; 404 connected components, and the largest or giant component consist of about 1:65 million nodes, which is about 99:52% of the total number of nodes. On the other hand, the largest strongly connected component consists of 23; 251 nodes (1:4%) and 0:2 million edges.

4.2.2 **Centrality.** As a first attempt to understand the graph's properties, we considered various centrality measures and their interpretation. We also present the top-10 services according to each centrality measure. More specifically:

In-Degree Centrality. Count of average number of domains using a given service. In other words, reflects the number of domains that a particular service could break if it goes down/fails.

PageRank. PageRank is somewhat related to in-degree centrality, but unlike in-degree it reflects a measure of connectivity in the whole network—i.e., if many random walks in the graph lead to a particular service u, then u will have high PageRank—since domains used by others use u's services as well. Random walks in the graph follow the same paths that cascading failures would take; hence, high PageRank implies a larger danger of cascading failures.

Betweenness centrality. An important service (node) will lie on a high proportion of dependency/usage paths between other nodes in the network. Removal of high betweenness centrality nodes is known to disconnect graphs in addition to causing cascading failures in the underlying networks.

Eigenvector centrality. Bonacich or eigenvector centrality assigns high importance scores to the services that use other important services and in turn are used by other important services. Eigenvector centrality, in some sense, is a measure of vulnerability to cascading failures – if a particular service or a domain use a lot of external services and are used by a lot of external services then it is potentially vulnerable to cascading failure events.

Table 3 presents the top 10 services for each of the centrality measures discussed above. Perhaps unsurprisingly, Amazon is consistently the most dominant service for all the centrality measures, which indicates that it is the most central and important service provider in today's Internet. This also means that an outage of or attack on Amazon's services can take down many directly as well as indirectly dependant domains and services. Other most important services across different centrality measures include Google, Cloudflare, Barefruit and Akamai. It is worth noting that all the top-10 services across different centrality measures belong to the largest strongly connected component of the graph, which consists of just 1:4% of the nodes in the graph.

The case of Barefruit is quite interesting: it is a little-known advertising company that collaborates with ISPs and takes advantage of non-existant domain errors to display advertising. The results of Barefruit are, hence, likely to be a glitch: they appear in our passive DNS traces because it was the providers —not the queried domains— that used Barefruit's services. This is a useful reminder of the limitations of our approach.

4.2.3 **Degree Distribution.** As we discussed in Section 3.2, studying the degree distribution of the graph we are analyzing is key to understanding the spreading of cascading failures over the network: due to their very structure, scale-free graphs—where a few nodes have a very high degree—carry the risk of cascading failures that spread to very large portions of the graph even when the probability of a cascading failure itself is low.

In Figure 5, following the advice of Clauset et al. [12], we show the complementary cumulative distribution function (CCDF) of degree in a double-logarithmic scale[9]: power law degree distributions have a linear shape in the plot, which is true for our graph. In this graph, the in- and out-degrees are different distributions, where

[9]To plot the 0 values, the plot has a linear scale in the »0; 1¼ interval.

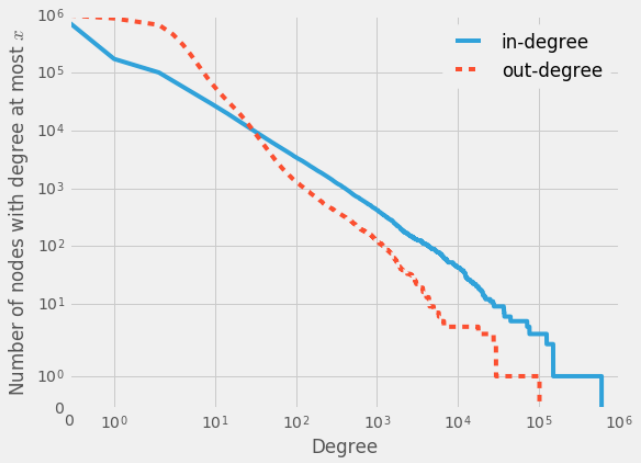

Figure 5: Degree distribution of the domain graph.

the in-degree of nodes like `amazon.com` or `google.com` is definitely larger than the largest out-degrees in the graph.

4.2.4 **Support Power.** Table 4 shows the top-15 domains based on the support power metric discussed in Section 3.2. We present results for in-degree based support power $SP^{1in\ \ d^o}$, PageRank based support power SP^{1pr^o} and eigenvector based support power SP^{1ec^o}. We do not consider betweenness centrality in this context as it is not related to in-degree (dependency) measure as much as pagerank and eigenvector centrality. In-degree based support power just considers 2-hop dependencies, whereas pagerank and eigenvector centrality based power values consider every possible dependency leading up to a particular node (service). Surprisingly, unlike centrality based results, Barefruit appears to be the most powerful service domain across all the variants of the SP metric—followed by HDE Inc Japan, Unbounce, Amazon Europe, Hetzner Online, and PCCW Global. We can observe that service domains that provide a very unique service (or operate in regions where there is a lack of alternatives), are more powerful, with strong support chains.

5 RELATED WORK

To the best of our knowledge, and despite its importance in terms of security, safety and also market research, scientific literature on the problem of deciphering the web of dependencies in the Internet only recently began to gain traction.

In 2012, Nikiforakis et al. [33] crawled a set of popular websites, showing that many of them rely on JavaScript libraries served by third parties; compromising those third parties can enable the attackers to steal data from the page itself and/or from other scripts.

In 2013, He et al. [23] studied the case of two popular cloud providers, namely Amazon EC2 and Microsoft Azure, by the Alexa top 1M website domains. Using a combination of actively collected DNS data and network traffic logs collected at an academic network they report on the popularity of cloud providers, deployment strategies for web services and resilience of cloud infrastructures to failure. Our work extends and goes beyond the work by He et al. by looking at different types of services, i.e., not only cloud

Table 3: Top 10 domains for various centrality metrics.

#	In-Degree Centrality		PageRank		Betweenness Centrality		Eigenvector Centrality	
1	amazon.com	0.36	amazon.com	0.07	amazon.com	0.1B	amazon.com	0.44
2	google.com	0.09	cloudflare.com	0.013	google.com	0.05B	barefruit.co.uk	0.31
3	cloudflare.com	0.075	google.com	0.013	akadns.net	0.04B	akamai.com	0.15
4	secureserver.net	0.05	barefruit.co.uk	0.011	ripe.net	0.038B	rr.com	0.13
5	domaincontrol.com	0.04	secureserver.net	0.005	akamai.com	0.029B	centurylink.com	0.11
6	barefruit.co.uk	0.03	barefruit.com	0.005	microsoft.com	0.026B	google.com	0.11
7	outlook.com	0.02	domaincontrol.com	0.005	ntt.net	0.025B	xo.com	0.10
8	hetzner.com	0.02	hetzner.com	0.003	apple.com	0.021B	zayo.com	0.09
9	akamai.com	0.02	ibm.com	0.003	amazonaws.com	0.02B	pccwglobal.com	0.09
10	linode.com	0.016	akamai.com	0.002	cloudflare.com	0.018B	teliacarrier.com	0.09

Table 4: Top 15 domains based on support power (SP).

#	In-degree based $SP^{1in\ d^o}$		PageRank based SP^{1pr^o}		Eigenvector Centrality based SP^{1ec^o}	
1	barefruit.co.uk	42,538	barefruit.co.uk	7,928	barefruit.co.uk	321,348
2	hdemail.jp	7,083	hdemail.jp	1,349	unbouncepages.com	26,067
3	amazon.eu	4,497	amazon.eu	855	hetzner.com	17510
4	unbouncepages.com	3,129	sixcore.ne.jp	717	pccwglobal.com	14,187
5	pccwglobal.com	2,558	xserver.jp	559	amazon.com	13,775
6	ultradns.com	2,221	hetzner.com	525	vsnl.net.in	9,107
7	vsnl.net.in	2,088	unbouncepages.com	465	hdemail.jp	8,875
8	asahi-net.or.jp	1,922	pccwglobal.com	461	customersaas.com	7,445
9	hetzner.com	1,785	ultradns.com	426	ultradns.com	6,800
10	estore.co.jp	1,712	amazon.com	383	telecomitalia.com	6,569
11	sixcore.ne.jp	1,578	vsnl.net.in	363	zayo.com	6,220
12	telecom.com.ar	1,375	asahi-net.or.jp	349	libguides.com	6,113
13	amazon.com	1,270	xserver.ne.jp	271	verio.com	5,963
14	roaringpenguin.com	1,247	telecom.com.ar	260	amazon.eu	5,611
15	mynet.cn	1,077	estore.co.jp	247	cantv.com.ve	5,189

providers, in order to uncover an as complete as possible picture of web service dependencies. Moreover, by doing so over a period of two and half years we are able to observe trends and the evolution over time.

In 2016, Cangialosi et al. [11] studied the phenomenon of private HTTPS key sharing, confirming —from another angle— our conclusions that the core Internet infrastructure suffers from being overly concentrated in the hands of a few players.

Recently, Simeonovski et al. [44] performed a study that collected data the about Internet topology and used it to model three kinds of attacks (distribution of malicious JavaScript, email sniffing, and DoS against core service providers); among the various datasets Simeonovski et al. used, there is an actively collected DNS dataset, which they employed to collect information on a few types of services (DNS, Web and email servers). Unlike their work, ours focuses specifically on everything that can be obtained with DNS data, including passive measurements, a larger set of services considered, and using several centrality metrics to estimate the influence of different players.

On the industrial side, various companies sell data about what online services each company/domain is using (e.g., SecurityScorecard's Automatic Vendor Detection [43], Datanyze [13], Wappalizer [50], Built With [9], etc.); however, their methods are typically kept as trade secrets, so it is hard to know the technical differences between their approach and the one discussed in this paper. To the best of our knowledge, however, it appears that most of the results from these vendors come from web crawling, rather than DNS data analysis. On a related topic to uncovering web service inter-dependencies, there exists a large corpus of studies seeking to extract and characterize relationships between Autonomous Systems (ASes) in the Internet routing [17, 18, 35, 54]. However, these studies focus exclusively on uncovering the type of relationships that exist between Internet Service Providers (ISPs) and customer networks at the (BGP) routing level. Such measurement study usually requires the use of (BGP) routing data, possibly combined with active measurements such as `traceroute`. While such routing-level service inter-dependency results could complement our the DNS-based results, it is out of the scope of this paper.

DNS Data Collection and Analysis. The DNS constitutes a critical building block of the Internet and as such plays an ubiquitous role in most Internet activities. Originally designed to translate user-friendly domain names into IP addresses, the DNS nowadays is also (ab)used for many different purposes, such as operating blacklists of spam email senders, tunneling traffic, enabling cybercriminals to operate moving command and control infrastructures, etc. Inspecting DNS data has thus become an ideal way to monitor different aspects of Internet activity, including cyber security aspects. In fact, the security community has been using the DNS extensively to detect Internet abuses [2, 3, 6, 15, 22, 24, 28, 53]. As mentioned earlier, He et al. [23] leverage DNS data to study usage patterns of the two cloud providers Amazon EC2 and Microsoft Azure.

DNS data is usually collected in two ways: (1) by *passively* recording DNS queries and responses that are made to some collector DNS servers [51] or (2) by *actively* querying some domains [28, 47]. While passive DNS data is considered easier to collect and is by far the most common type of DNS data, both approaches appear to be complementary to obtain the best data coverage, in terms of number of domains, possible. On the one hand passively collected DNS datasets tend to be biased towards popular domains likely to be queried at the DNS server collectors. On the other hand active DNS data collection requires a list of seed domains, which will ensure coverage for these domains while potentially reducing the global coverage of the resulting datasets. To overcome these limitations, in this work, we use a combination of passive and active DNS data.

Some publicly available software packages look at `CNAME` fields in order to discover CDNs.[10] This approach bears some similarity with our proposal, but it is limited to active, on-demand, `CNAME` queries, and only recognizes a few specific services for which rules have been written beforehand. Our approach lets us discover previously unknown services of any kind, and exploits passive DNS information in other fields, including `A` fields through reverse DNS.

Graph Analysis. Graph analysis is a powerful tool for discovering valuable information about relationships in complex network data. In this paper, we have tried to analyze the Internet service dependency graph to understand the nature of dependencies, important services, and cascading failures in the presence of service outages. Literature is filled with work where dependencies of some type and kind are analyzed on a variety of graph data related to different problem domains [10, 20, 26, 32, 45, 55]. For instance, Stergiopoulos et al. [45] model cascading critical infrastructure failures using dependency risk graphs; they explore relationships between dependency risk paths and graph centrality measures so as to identify nodes that significantly impact the overall dependency risk. Zimmermann and Nagappan [55] study software dependencies spread across binaries developed by different teams; they compute the complexity of the subsystem's dependency graphs using concepts adapted from classical graph theory, and hypothesize that these complexities correlate with failures. Our work is the very first to use graph-analysis in the domain of DNS and Internet services to assess and analyze the dependency between websites and services.

The problem of measuring node importance through centrality has been studied extensively in past [8, 42, 49, 52]. Our work makes use of several standard graph-theoretical centrality measures to evaluate the importance of services. In addition to centrality, power is also an interesting graph theoretical concept introduced by Bonacich [8]. In this paper, we have identified the limitations in the applicability of Bonacich's power metric in our context, and have introduced a novel power metric—namely *support power*.

[10] See, e.g., https://github.com/WPO-Foundation/webpagetest/

6 CONCLUSION

Even though the Internet was designed to be a highly resilient, decentralized infrastructure, recent incidents that have caused disruption of popular Internet services have underlined the fact that there is a high degree of dependency between websites and service providers.

Motivated by these events and our desire to develop risk assessment mechanisms for organizations, we have analyzed a large corpus of DNS data in order to discover and investigate such dependencies. Our analysis quantifies the degree of dependency of the Alexa top 1M domains to various Internet service providers. Furthermore, our graph analysis reinforces our findings, while the introduction of the *support power* metric attempts to capture more complex relationships that can play an important role in understanding cascading failures.

We believe that the this work is a first step towards establishing a set of actionable metrics that can assist website and service operators in making informed choices about their Internet/cloud service dependencies, so as to mitigate the effects of large-scale incidents, improve resiliency, and minimize overall exposure to risk.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their very valuable comments and suggestions.

REFERENCES

[1] Joshua Abramson. 2016. DDoS Attacks: Bigger, Stronger, Scarier. Symantec Official Blog. (April 2016). Retrieved June 1, 2017 from https://goo.gl/NZDRsD
[2] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. 2010. Building a Dynamic Reputation System for DNS. In *Proceedings of the 19th USENIX Conference on Security (USENIX Security'10)*. USENIX Association, Berkeley, CA, USA, 18–18. http://dl.acm.org/citation.cfm?id=1929820.1929844
[3] Manos Antonakakis, Roberto Perdisci, Wenke Lee, Nikolaos Vasiloglou, II, and David Dagon. 2011. Detecting Malware Domains at the Upper DNS Hierarchy. In *Proceedings of the 20th USENIX Conference on Security (SEC'11)*. USENIX Association, Berkeley, CA, USA, 27–27.
[4] AWS message 2017. Summary of the Amazon S3 Service Disruption in the Northern Virginia (US-EAST-1) Region. AWS message. (March 2017). https://aws.amazon.com/message/41926/
[5] Matt Bertz. 2015. Sony's Yoshida: PSN Is Attacked By Hackers Every Day. Gameinformer. (March 2015). Retrieved June 1, 2017 from https://goo.gl/nrviUG
[6] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. 2011. EXPOSURE : Finding malicious domains using passive DNS analysis. In *NDSS 2011, 18th Annual Network and Distributed System Security Symposium, 6-9 February 2011, San Diego, CA, USA*. Internet Society, San Diego, UNITED STATES, Article 11, 17 pages. http://www.eurecom.fr/publication/3281
[7] Marián Boguná, Romualdo Pastor-Satorras, and Alessandro Vespignani. 2003. Absence of epidemic threshold in scale-free networks with degree correlations. *Physical review letters* 90, 2 (2003), 028701.
[8] Phillip Bonacich. 1987. Power and Centrality: A Family of Measures. *Amer. J. Sociology* 92, 5 (1987), 1170–1182. http://www.jstor.org/stable/2780000
[9] Built With 2017. Built With. Website. (2017). https://builtwith.com/
[10] G. Candea, M. Delgado, M. Chen, and A. Fox. 2003. Automatic failure-path inference: a generic introspection technique for Internet applications. In *Proceedings the Third IEEE Workshop on Internet Applications. WIAPP 2003.* IEEE, San Jose, CA, USA, 132–141. DOI: http://dx.doi.org/10.1109/WIAPP.2003.1210298
[11] Frank Cangialosi, Taejoong Chung, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, and Christo Wilson. 2016. Measurement and analysis of private

key sharing in the https ecosystem. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Vienna, Austria, 628–640.

[12] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. 2009. Power-law distributions in empirical data. *SIAM review* 51, 4 (2009), 661–703.

[13] Datanyze 2017. Datanyze. Website. (2017). https://www.datanyze.com/

[14] Darrell Etherington. 2017. Amazon AWS S3 outage is breaking things for a lot of websites and apps. TechCrunch. (February 2017). Retrieved June 1, 2017 from https://techcrunch.com/2017/02/28/amazon-aws-s3-outage-is-breaking-things-for-a-lot-of-websites-and-apps/

[15] Mark Felegyhazi, Christian Kreibich, and Vern Paxson. 2010. On the Potential of Proactive Domain Blacklisting. In *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More (LEET'10)*. USENIX Association, Berkeley, CA, USA, Article 6, 8 pages.

[16] Jérôme Fleury. 2016. A Post Mortem on this Morning's Incident. Cloudflare blog. (June 2016). https://blog.cloudflare.com/a-post-mortem-on-this-mornings-incident/

[17] Lixin Gao. 2001. On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Trans. Netw.* 9, 6 (Dec. 2001), 733–745.

[18] Vasileios Giotsas, Matthew Luckie, Bradley Huffaker, and kc claffy. 2014. Inferring Complex AS Relationships. In *Proceedings of the 2014 Internet Measurement Conference (IMC '14)*. ACM/IEEE, Vancouver, BC, Canada, 23–30.

[19] Google Help 2017. What is 1e100.net? Google Help. (2017). Retrieved June 2, 2017 from https://support.google.com/faqs/answer/174717?hl=en

[20] Li Gou, Bo Wei, Rehan Sadiq, Yong Sadiq, and Yong Deng. 2016. Topological Vulnerability Evaluation Model Based on Fractal Dimension of Complex Networks. *PLOS ONE* 11, 1 (01 2016), 1–11.

[21] John Graham-Cumming. 2016. How the Dyn outage affected Cloudflare. Cloudflare blog. (October 2016). Retrieved June 1, 2017 from https://blog.cloudflare.com/how-the-dyn-outage-affected-cloudflare/

[22] Chris Grier, Lucas Ballard, Juan Caballero, Neha Chachra, Christian J. Dietrich, Kirill Levchenko, Panayiotis Mavrommatis, Damon McCoy, Antonio Nappa, Andreas Pitsillidis, Niels Provos, M. Zubair Rafique, Moheeb Abu Rajab, Christian Rossow, Kurt Thomas, Vern Paxson, Stefan Savage, and Geoffrey M. Voelker. 2012. Manufacturing Compromise: The Emergence of Exploit-as-a-service. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS '12)*. ACM, Raleigh, North Carolina, USA, 821–832.

[23] Keqiang He, Alexis Fisher, Liang Wang, Aaron Gember, Aditya Akella, and Thomas Ristenpart. 2013. Next Stop, the Cloud: Understanding Modern Web Service Deployment in EC2 and Azure. In *Proceedings of the 2013 Internet Measurement Conference (IMC '13)*. ACM, Barcelona, Spain, 177–190.

[24] Thorsten Holz, Christian Gorecki, Konrad Rieck, and Felix C. Freiling. 2008. Measuring and Detecting Fast-Flux Service Networks.. In *NDSS* (2009-06-18). The Internet Society, San Diego, CA, USA, Article 16, 12 pages.

[25] IHS Markit 2011. Cloud- Related Spending by Businesses to Triple from 2011 to 2017. IHS Markit News Releases. (2011). http://news.ihsmarkit.com/press-release/design-supply-chain/cloud-related-spending-businesses-triple-2011-2017

[26] M. Jalili. 2015. Resiliency of cortical neural networks against cascaded failures. *Neuroreport* 26, 12 (Aug 2015), 718–722.

[27] Swati Khandelwal. 2016. 602 Gbps! This May Have Been the Largest DDoS Attack in History. The Hacker News. (January 2016). Retrieved June 1, 2017 from http://thehackernews.com/2016/01/biggest-ddos-attack.html

[28] Athanasios Kountouras, Panagiotis Kintis, Chaz Lever, Yizheng Chen, Yacin Nadji, David Dagon, Manos Antonakakis, and Rodney Joffe. 2016. *Enabling Network Security Through Active DNS Datasets*. Springer International Publishing, Cham, 188–208.

[29] Edward Kovacs. 2016. Brian Krebs' Blog Hit by 665 Gbps DDoS Attack. SecurityWeek. (February 2016). Retrieved June 1, 2017 from http://www.securityweek.com/brian-krebs-blog-hit-665-gbps-ddos-attack

[30] Brian Krebs. 2016. DDoS on Dyn Impacts Twitter, Spotify, Reddit. Krebs on Security. (October 2016). Retrieved June 1, 2017 from https://krebsonsecurity.com/2016/10/ddos-on-dyn-impacts-twitter-spotify-reddit/

[31] Mike Lennon. 2016. Britain's HSBC Recovers from Massive DDoS Attack. Security Week. (January 2016). Retrieved June 1, 2017 from https://goo.gl/haZbz8

[32] Igor Mishkovski, Mario Biey, and Ljupco Kocarev. 2011. Vulnerability of complex networks. *Communications in Nonlinear Science and Numerical Simulation* 16, 1 (2011), 341–349.

[33] Nick Nikiforakis, Luca Invernizzi, Alexandros Kapravelos, Steven Van Acker, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. 2012. You are what you include: large-scale evaluation of remote javascript inclusions. In *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, Raleigh, NC, USA, 736–747.

[34] Jordan Novet. 2017. AWS is investigating S3 issues, affecting Quora, Slack, Trello. VentureBeat. (February 2017). Retrieved June 1, 2017 from https://venturebeat.com/2017/02/28/aws-is-investigating-s3-issues-affecting-quora-slack-trello/

[35] Ricardo Oliveira, Dan Pei, Walter Willinger, Beichuan Zhang, and Lixia Zhang. 2010. The (in)Completeness of the Observed Internet AS-level Structure. *IEEE/ACM Trans. Netw.* 18, 1 (Feb. 2010), 109–122. DOI: http://dx.doi.org/10.1109/TNET.2009.2020798

[36] Pierluigi Paganini. 2016. OVH hosting hit by 1Tbps DDoS attack, the largest one ever seen. Security Affairs. (September 2016). Retrieved June 1, 2017 from http://securityaffairs.co/wordpress/51640/cyber-crime/tbps-ddos-attack.html

[37] Romualdo Pastor-Satorras and Alessandro Vespignani. 2001. Epidemic spreading in scale-free networks. *Physical review letters* 86, 14 (2001), 3200.

[38] Romualdo Pastor-Satorras and Alessandro Vespignani. 2002. Epidemic dynamics in finite size scale-free networks. *Physical Review E* 65, 3 (2002), 035108.

[39] Matthew Prince. 2013. The DDoS That Knocked Spamhaus Offline (And How We Mitigated It). BGPmon.net. (March 2013). https://blog.cloudflare.com/the-ddos-that-knocked-spamhaus-offline-and-ho/

[40] Matthew Prince. 2014. Technical Details Behind a 400 Gbps NTP Amplification DDoS Attack. Cloudflare blog. (February 2014). Retrieved June 1, 2017 from https://goo.gl/K1EJHF

[41] Rapid7. 2013. Project Sonar. (2013). Retrieved June 1, 2017 from https://sonar.labs.rapid7.com/

[42] Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda. 2016. Super mediator—A new centrality measure of node importance for information diffusion over social network. *Information Sciences* 329 (2016), 985 – 1000. DOI: http://dx.doi.org/10.1016/j.ins.2015.03.034 Special issue on Discovery Science.

[43] Security ScoreCard. 2016. Automatic Vendor Detection – Do You Know Who Your Vendors Are? Blog post. (2016). http://blog.securityscorecard.com/2016/02/15/automatic-vendor-detection-know-who-your-vendors-are/

[44] Milivoj Simeonovski, Giancarlo Pellegrino, Christian Rossow, and Michael Backes. 2017. Who Controls the Internet?: Analyzing Global Threats using Property Graph Traversals. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, Perth, Australia, 647–656.

[45] George Stergiopoulos, Marianthi Theocharidou, Panayiotis Kotzanikolaou, and Dimitris Gritzalis. 2015. *Using Centrality Measures in Dependency Risk Graphs for Efficient Risk Mitigation*. Springer International Publishing, Arlington, VA, USA, 299–314. DOI: http://dx.doi.org/10.1007/978-3-319-26567-4_18

[46] Andree Toonk. 2013. Looking at the Spamaus DDOS from a BGP perspective. BGPmon.net. (March 2013). https://bgpmon.net/looking-at-the-spamhouse-ddos-from-a-bgp-perspective/

[47] Roland van Rijswijk-Deij, Mattijs Jonker, Anna Sperotto, and Aiko Pras. 2015. The Internet of Names: A DNS Big Dataset. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM '15)*. ACM, London, United Kingdom, 91–92.

[48] Kimberley Wallace. 2014. Xbox Live And PSN Issues Plague Christmas, Some Still Ongoing. Gameinformer. (December 2014). Retrieved June 1, 2017 from https://goo.gl/c00RH2

[49] Shasha Wang, Yuxian Du, and Yong Deng. 2017. A new measure of identifying influential nodes: Efficiency centrality. *Communications in Nonlinear Science and Numerical Simulation* 47 (2017), 151 – 163. DOI: http://dx.doi.org/10.1016/j.cnsns.2016.11.008

[50] Wappalyzer 2017. Wappalyzer. Website. (2017). https://wappalyzer.com/

[51] Florian Weimer. 2005. Passive DNS replication. In *17th Annual FIRST Conference on Computer Security*. FIRST, Singapore, 98.

[52] Sheng Wen, Jiaojiao Jiang, Bo Liu, Yang Xiang, and Wanlei Zhou. 2017. Using epidemic betweenness to measure the influence of users in complex networks. *Journal of Network and Computer Applications* 78 (2017), 288 – 299. DOI: http://dx.doi.org/10.1016/j.jnca.2016.10.018

[53] Sandeep Yadav, Ashwath Kumar Krishna Reddy, A.L. Narasimha Reddy, and Supranamaya Ranjan. 2010. Detecting Algorithmically Generated Malicious Domain Names. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC '10)*. ACM, Melbourne, Australia, 48–61.

[54] Beichuan Zhang, Raymond Liu, Daniel Massey, and Lixia Zhang. 2005. Collecting the internet AS-level topology. *SIGCOMM CCR* 35 (January 2005), 53–61.

[55] T. Zimmermann and N. Nagappan. 2007. Predicting Subsystem Failures using Dependency Graph Complexities. In *The 18th IEEE International Symposium on Software Reliability (ISSRE '07)*. IEEE, Trollhattan, Sweden, 227–236. DOI: http://dx.doi.org/10.1109/ISSRE.2007.19