SAC105
The DNS and the Internet of Things: Opportunities, Risks, and Challenges

A report from the ICANN Security and Stability Advisory Committee (SSAC)
28 May 2019

Preface

This is a report of the ICANN Security and Stability Advisory Committee (SSAC). The SSAC focuses on matters relating to the security and integrity of the Internet's naming and address allocation systems. This includes operational matters (e.g., pertaining to the correct and reliable operation of the root zone publication system), administrative matters (e.g., pertaining to address allocation and Internet number assignment), and registration matters (e.g., pertaining to registry and registrar services). SSAC engages in ongoing threat assessment and risk analysis of the Internet naming and address allocation services to assess where the principal threats to stability and security lie, and advises the ICANN community accordingly. The SSAC has no authority to regulate, enforce, or adjudicate. Those functions belong to other parties, and the advice offered here should be evaluated on its merits.

Table of contents

Executive Summary

The Internet of Things (IoT) promises to enhance our daily lives by seamlessly and autonomously sensing and acting upon our physical environment through tens of billions of connected devices. While this makes the IoT vastly different from traditional Internet applications like email and web browsing, we expect that a significant number of IoT deployments will use the DNS to locate remote services that they need, for instance to enable telemetry data transmission and collection for monitoring and analysis of sensor data.

In this report, the SSAC provides a discussion on the interplay between the DNS and the IoT, arguing that the IoT represents both an opportunity and a risk to the DNS. It is an opportunity because the DNS provides functions and data that can help make the IoT more secure, stable, and transparent, which is critical given the IoT's interaction with the physical world. It is a risk because various measurement studies suggest that IoT devices may stress the DNS, for instance, because of complex DDoS attacks carried out by botnets that grow to hundreds of thousands or in the future millions of infected IoT devices within hours.

We also identify and discuss five challenges for the DNS and IoT industries (e.g., DNS and IoT operators and software developers) to address these opportunities and risks, for instance by making the DNS's security functions (e.g., response verification and encryption) available on popular IoT operating systems and by developing a shared system that allows different DNS operators to automatically and continually exchange data on IoT botnet activity.

Unlike typical SSAC publications, the aim of this report is to trigger and facilitate dialogue in the broader ICANN community. We therefore provide a tutorial-style discussion that is more forward looking than operational in nature. Our discussion partly falls within ICANN's and SSAC's remit, but also goes beyond it, for instance, because the challenges we identify will take a wider range of players to address. We explicitly do not provide any recommendations and do not solicit any actions from the ICANN community or Board.

1 Introduction

The Internet of Things (IoT) is an emerging Internet application that extends "network connectivity and computing capability to objects, devices, sensors, and items not ordinarily considered to be computers" [1]. The IoT is expected to connect 20-30 billion of such objects to the Internet in the next decade [2], with some analysts even predicting 100 billion connected objects by 2025 [1]. Examples of IoT application areas include smart homes with connected kitchen appliances, toys, lightbulbs, smart cities with connected street lights and environmental sensors, and self-organizing dynamic networks of drones and robots [3].

The key difference between "classical" Internet applications and the IoT is that the former typically enable humans to interact with content and services (e.g., through web browsers), whereas IoT applications typically operate "in the background" as an integral and invisible part of people's lives [1][4]. Conceptually, the IoT accomplishes this by continually interpreting and updating a distributed online representation of physical environments based on data from a wide range of sensors and then using this representation to act upon the real world through actuators, all typically without direct human involvement or awareness. This continual and intelligent sense-interpret-act loop promises to enhance our daily lives, for instance, through a smart wrist watch that automatically opens the front door of our house (perhaps in combination with a voice command) and adds various bio measurements (e.g., movement or heart rate patterns) to our medical file.

Another important difference is that the applications on IoT devices are typically control programs that run on widely heterogeneous hardware, CPU architectures, and operating systems. This is unlike today's (web-based) Internet applications, which typically run on relatively homogenous devices like laptops and mobile phones. For example, some control programs will run on IoT devices with small battery-operated sensors without a user interface and communicate via low-powered radios (e.g., Bluetooth Low Energy (BLE) or Zigbee) that rely on intermediate nodes to connect to the Internet. Others will run on relatively high-end devices with richer interaction facilities (e.g., a connected refrigerator with a touch screen user interface), powerful processing and storage capabilities, WiFi connectivity, and a built-in IP protocol stack, allowing them to directly connect with services on the Internet (e.g., with goods suppliers, maintenance facilities, or energy management functions).

Figure 1 illustrates this device heterogeneity, with examples of high-end general-purpose devices in the top left (low device specificity, high processing power) and low-end special purpose devices in the bottom right (high device specificity, low processing power) [5]. IoT devices are also different because they may be intended, like appliances, to have much longer lifespans (on the order of decades) and their software may be difficult to upgrade [1].

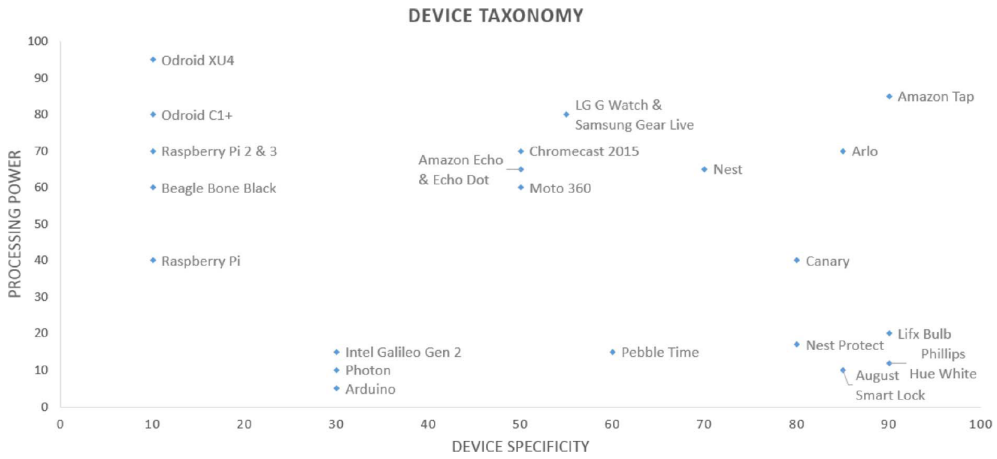

Figure 1: Indicative oT device taxonomy from [5].
y-axis: relative indication of processing power, x-axis: specificity of device functions.

Contemporary IoT devices with an IP stack typically exchange data with one or more remote services hosted on the Internet [6] (e.g., to analyze sensor data) and locate these services using the DNS [5] [7] [8]. As a result, we end up with two co-evolving and interacting ecosystems: the DNS with its resolver operators, authoritative name server operators, and domain registration providers, and the IoT with its device manufacturers, IoT device operators (e.g., swarms of drones or smart street lights), and providers of the remote services these devices interact with.

In this report, we explore the interactions between the DNS and the IoT and argue that the IoT is both an opportunity to further increase the value of the DNS as well as a risk that potentially reduces the value of the DNS.

The IoT is an opportunity because IoT devices sense and act upon physical environments and will therefore have new security, stability, and transparency requirements [9] [10] that the DNS's functions can partly fulfill. For example, the control software of a connected door lock will want to make sure that it talks to the service on the Internet to which users send their unlock instructions to (e.g., using voice control). If the IoT device validates DNSSEC signatures, then it will be able to spot DNSSEC validation errors and decide not to accept unlock instructions from the service that the DNS tells it to connect to. DNSSEC validation errors are typically the result of man-in-the-middle attacks, such as a routing hijack (e.g., the route hijack that targeted Amazon in April of 2018 [11]). Without DNSSEC, devices would not notice such attacks and would send or receive traffic from malicious destinations, which could jeopardize the user's safety and privacy.

The IoT is a risk to the DNS because various measurement studies suggest that IoT devices could stress the DNS infrastructure in ways that we have not seen before. For example, a software update for a popular IP-enabled IoT device that causes the device to use the DNS more frequently (e.g., regularly lookup random domain names to check for network availability) could stress the DNS in individual networks when millions of devices automatically install the update at the same time. While this is a programming error from the perspective of individual devices, it could result in a significant attack vector from the perspective of DNS infrastructure operators. Incidents like this have already occurred on a small scale [12], but they may occur more frequently in the future due

to the growth of heterogeneous IoT devices from manufacturers that equip their IoT devices with controllers that use the DNS.

To take advantage of these opportunities and address the risks, we identify and discuss five challenges for the DNS and IoT industries (e.g., DNS and IoT operators and software developers). One challenge is to make DNS security functions (e.g., response verification and encryption) available on popular IoT operating systems and to develop shared systems that allow different DNS operators to automatically and continually exchange data on IoT botnet activity. The challenges we identify complement and detail more generic IoT security challenges such as the need for secure remote software updates and end-of-life support, which are, for instance, discussed in [1] and [9].

Many aspects of our discussion are not new, except as they consider new challenges presented by the IoT. For example, making DNSSEC more widely available for "traditional" Internet applications is already challenging, but is even more complex for the IoT because it needs to be supported by many different IoT operating systems (e.g., OpenWRT, RIOT, and Contiki) and must deal with more heterogeneous hardware (e.g., limiting the implementation of cryptographic functions on low-powered hardware).

In terms of scope, we consider the IoT in a "horizontal" way and do not specifically focus on a particular "vertical" application (e.g., home automation, intelligent urban transport systems, energy control, or robotics), except for illustration purposes. The opportunities, risks, and challenges we discuss are similar across these verticals, although the specifics may differ. For example, the inherent safety issues of intelligent transport systems will likely come with tighter security measures, which reduces the probability that devices will be infected with malware used to DDoS a DNS operator. The impact on users (e.g., a malfunctioning online traffic light), however, may be just as severe.

We further focus our discussion by excluding topics like application-level interoperability [10] (e.g., can a service that analyzes heart rate patterns work with sensors from different wrist watch manufacturers), device-specific security mechanisms (e.g., preventing privilege escalation on IoT devices), naming of IoT devices and alternative naming systems, local name discovery mechanisms (e.g., multicast DNS), and liability issues of IoT devices bundled with remote services [6]. We assume the reader has a working knowledge of processes and issues related to the registering of domain names in the public DNS.

Unlike typical SSAC publications, the purpose of this report is to trigger and facilitate dialogue in the broader ICANN community. We therefore provide a tutorial-style discussion that is more forward looking than operational in nature. Our discussion partly falls within ICANN's and SSAC's remit, but also goes beyond it, for instance because the challenges we identify will take a wider range of players to address. We explicitly do not provide any recommendations and do not solicit any actions from the ICANN community or Board.

Our work is also relevant in light of ICANN's recently published draft strategic plan for 2021-2025, which specifically mentions the risks of the IoT for the DNS (strategic objective 1) [13].

In the rest of this report, we first discuss our model of the role that the DNS plays for the IoT (Section 2). We then discuss the opportunities that the DNS offers to increase IoT security and transparency (Section 3), followed by a discussion on how the IoT might pose a risk for the DNS (Section 4). We end with 5 challenges for the DNS and IoT industries (Section 5) and our conclusions and proposed next steps (Section 6).

2 The DNS and the IoT

Our model of the IoT revolves around the concept of an "IoT deployment", which we define as the combination of a group of IoT devices that sense and act upon a physical environment, the remote services on the Internet that the IoT devices use to provide their functions, and the DNS services they interact with. Our definition is inspired by ENISA's more general and loosely defined notion of an IoT deployment [14], which we refined for the specific case of the DNS. We also reuse concepts from [10] (notably their device-to-cloud and device-to-gateway interaction models) and extend them to include DNS interactions for service discovery. We envision that the IoT will consist of many heterogeneous deployments in terms of application areas, interaction facilities, operating systems, and network capabilities.

We discuss how the devices and services in an IoT deployment exchange data (Section 2.1) and how they use the DNS (Section 2.2).

2.1 Data transfer in IoT deployments

Figure 2 shows two examples of IoT deployments (DP1 and DP2) and how they transfer data. DP1 serves users in a smart home and consists of a smart wrist watch (device D1) and a remote service S1. D1 senses a user's movement and heart rate, encodes this information in a stream of IP packets, and sends it to S1 for analysis (device-to-cloud interaction [10]). The traffic flow typically reaches the remote services via the edge network in which the device resides (e.g., an 802.11 network) and several intermediate routers and networks. S1 interprets the measurements using application-specific logic and automatically sends instructions to other devices in the house. For example, it could use biometric information from D1's sensors to identify the user (e.g., by analyzing the person's body movements) and automatically instruct the lock on the front door (device D3) to open when the smart watch is near it. At the same time, S1 could send instructions to turn on the lights in the house (device D2) and interact with other devices in the home (e.g., to turn on the heating).

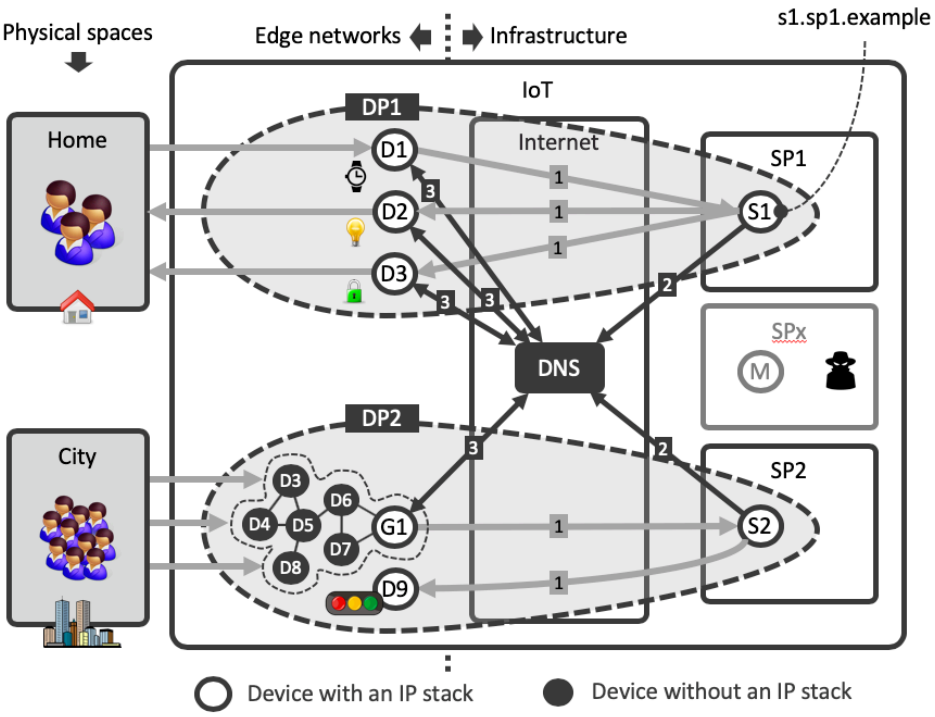

Figure 2: IoT deployment and the DNS.

The devices in deployment DP2 interact with their service (S2) through an intermediate gateway G1 (device-to-gateway interaction [10]). DP2 consists of a set of air pollution sensors distributed across a city (devices D3 through D8) that use a LoraWAN wireless network to connect to the Internet. The sensors are battery-operated and do not have the processing power and memory capabilities to run a full IP stack. Instead, they rely on G1 to put the sensor readings into IP packets and send them to S2. S2 uses the air pollution data along with traffic density data to dynamically guide vehicles away from heavily trafficked and polluted areas in the city by sending instructions to traffic lights (device D9) and roadside traffic signs. Gateways like G1 also bridge other differences between the edge network and the Internet such as protocol translation, encryption and authentication, and storage and retransmission of messages [15]. Gateways may also be mobile devices, for instance to enable an ad-hoc network of in-home IoT devices to connect to a remote service [16].

Remote services may also exchange data with each other [10], but we will not consider this interaction model because it does not involve an IoT device. We also do not consider IoT devices interacting with each other directly [10] (via a computer network or otherwise [17]) because they typically use a local service discovery protocol such as multicast DNS [18] for this purpose instead of the public DNS.

2.2 How IoT deployments use the DNS

The services in an IoT deployment (e.g., S1 and S2 in *Figure 2*) have their domain names registered in the DNS before they exchange data like any discoverable service on the Internet (interaction 2 in *Figure 2*). The difference is that services in the IoT help IoT devices sensing and acting upon a user's physical world, whereas "traditional" Internet applications focus on helping users

interacting with content or services. In addition, the services in an IoT deployment are usually invisible to end-users because the IoT device manufacturer configures them and users typically cannot easily change them. For example, the users in *Figure 2* would typically be unaware that their devices interact with remote services S1 and S2 and that they run in data centers of service providers SP1 and SP2, respectively.

IoT devices and gateways resolve a host name by sending a DNS lookup request to a resolver, the DNS component that "walks through" the DNS to map the host name to an IP address (interaction 3 in *Figure 2*). While an IoT device uses the same DNS protocol as traditional Internet applications, its temporal query pattern might be different. For example, some IoT devices are essentially wrappers around traditional browser applications [7] (e.g., a smart fridge with a touch screen or a device like the Amazon Echo) with a similar DNS query pattern. Other IoT devices will not have a user interface and will autonomously resolve a small static set of host names necessary for correct/secure operation (e.g., a light switch or a sleep tracker [7]). Device-to-gateway deployments use the DNS in the same way, except that the gateway (e.g., G1) handles the DNS interactions.

Figure 3 shows a simplified view of how device D1 (smart watch) locates the service S1, which has domain name s1.sp1.example. For simplicity, we depict a single name server at the root, the TLD, and the second level. D1 sends a DNS query for s1.sp1.example to resolver R3, which first passes the query to a root name server (A1). A1 returns the IP address(es) of the authoritative name server(s) of the .example TLD (A2), which R3 contacts next using the same query. The .example TLD's server returns the IP address(es) of the name server(s) of sp1.example (A3) to one of which R3 sends the query once more. A3 returns the IP address(es) of s1.sp1.example, which R3 sends to D1 so it can use an IP address to connect to S1.

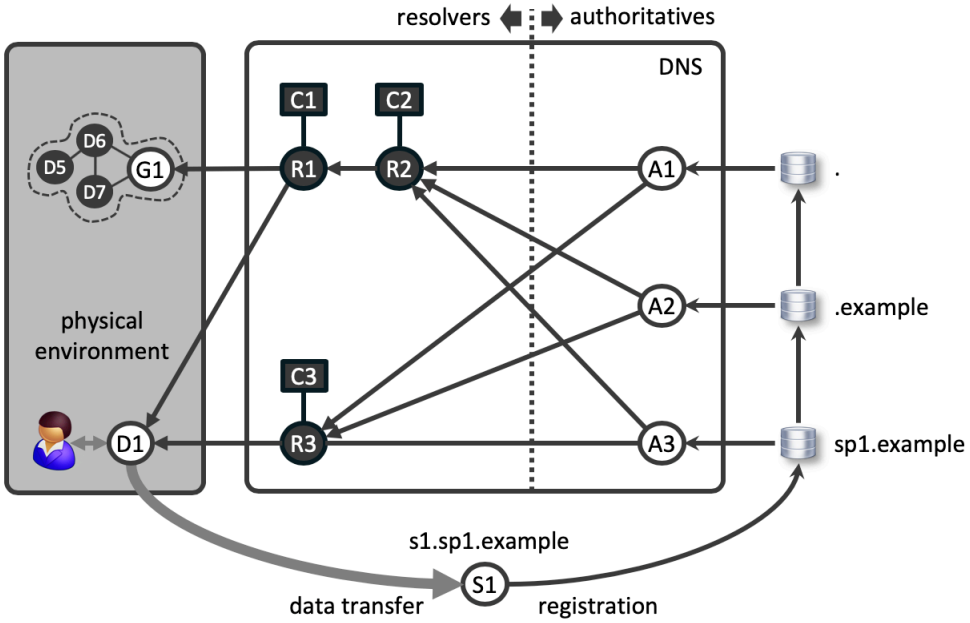

Figure 3: DNS resolution for IoT devices (graph inspired by [19]).

Devices like laptops and mobile phones typically use a resolver operated by the user's ISP, which they often discover dynamically, for instance using DHCP. Like some browsers, IoT devices may

ship with configuration parameters pointing to a preconfigured resolver, such as a public resolver or one that the device manufacturer provides. Domain name lookups may involve multiple layers of resolvers, for instance if R1 is a resolver on the user's home router and R2 is a resolver at the user's ISP.

A resolver has a cache in which it temporarily stores DNS responses, allowing it to reuse query results without making duplicate queries upstream. For example, resolver R3 serves lookups from D1 using its cache (C3). Each cache entry has a time-to-live (TTL), provided by an authoritative name server. TTLs may vary from a few seconds (e.g., for applications to load balance client traffic) to hours or days for TLD operators or servers with addresses that are not expected to change [19]. Caches are a crucial part of the DNS architecture because they reduce the number of DNS queries to authoritative name servers (A1 though A3 in *Figure* 3), which enables the system to scale and reduces lookup latencies.

3 Opportunities for the DNS to protect the physical world and increase IoT transparency

IoT deployments introduce new security, availability, and transparency requirements because they interact with physical space [9] [10], often without human involvement or awareness. This is an opportunity to further increase the value of the DNS because it is a globally pervasive infrastructure that can help in fulfilling these requirements. Specifically, the DNS allows IoT devices to authenticate resolvers and encrypt DNS traffic (Section 3.1), supports message authenticity (Section 3.2), provides additional ways for IoT devices to check the validity of the services they connect to (Section 3.3), and offers an opportunity for registration providers to provide additional security services for IoT deployments (Section 3.4). In addition, DNS datasets (DNS queries) enable a more transparent IoT for end-users (Section 3.5).

These opportunities result in new challenges for DNS software developers (e.g., to develop DNS security libraries for IoT devices), which we will discuss in Section 5.

3.1 Using DoH or DoT to authenticate resolvers of IoT devices and encrypt DNS queries

DNS-over-HTTPS (DoH) [20] and DNS-over-TLS (DoT) [21] are two new protocols that encrypt DNS messages between a DNS client and its resolver, thus hiding domain lookups and responses from on-path inspection and/or alteration between the client and the resolver. For example, DNS-over-HTTPS would encrypt the traffic between device D1 and resolver R3 (*Figure 3*) by sending DNS messages over a secure HTTPS channel, making it much more difficult to monitor the DNS queries D1 transmits.

We consider the key advantage of DoH and DoT to be that they enable IoT devices to verify the identity of their resolver through a digital certificate or, in DoT's case, a DNSSEC-validated chain of trust. As a result, it becomes more difficult to trick IoT devices into using a malicious resolver. For example, with DoH or DoT, device D1 will be able to verify the certificate that resolver R3 presents (see *Figure 3*) and detect malicious resolvers without a valid certificate. Without DoH or DoT, D1 could be redirected to a malicious resolver that lies about the IP address of s1.sp1.example so that D1 connects to a malicious service instead of to the authentic S1. As with

all public key applications, the trustworthiness of the digital certificate that DoH-enabled resolvers present depends on the extent to which the issuer of the certificate validated the identity of the resolver (e.g., no validation, domain validated, or extended validation).

The other potential advantage of DoH and DoT is that they reduce the amount of information that IoT devices reveal about themselves through their DNS queries, which might make it more difficult for miscreants to compromise them. This is important in the IoT because many IoT devices have very specific tasks (e.g., turn on the light, measure sleeping patterns) and will use only a small set of domain names and only for correct operation rather than for a user's interaction with content and services. For example, the sleep monitor studied in [7] uses only six pre-configured domain names, such as hello-audio.s3.amazonaws.com and sense-in.hello.is. The DNS lookups for these names provide on-path adversaries with additional information about the type of device and any vulnerabilities it may have which can then be used, for instance, to compromise the device and infect it with malware. This is an extra risk in the IoT because IoT devices often do not have a user interface, which means that compromises are more likely to stay undetected [5], making these devices attractive to adversaries.

Another advantage of DNS encryption is that it helps to protect the privacy of users when they (unknowingly) interact with devices that have very specific tasks. For example, the smart watch of *Figure 2* (device D1) might reveal when someone is asleep because that is when the traffic between D1 and S1 has a specific pattern, which is a realistic scenario as discussed in [7]. While an IoT device will typically transmit DNS queries at a lower rate than actual data traffic, their pattern might still reveal privacy-sensitive data about the user's physical space.

While both DoH and DoT obscure domain name lookups from on-path observers, IoT devices will not be able to hide the IP address of the services they use subsequent to the resolution [22]. For example, device D1 will be able to encrypt the DNS queries for s1.sp1.example it sends to R3, but it will not be able to hide the IP address of S1 (see *Figure 2*). As a result, some of DoH's and DoT's value will be diminished if on-path adversaries are able to observe the IP headers of the traffic between D1 and S1 and can identify the purpose of S1.

While DoH and DoT provide added value for the IoT, there are some security and scalability issues for the technical community to resolve, such as new processing overhead that both protocols present to the DNS. Further discussion of these topics is however outside the scope of this document.

3.2 Using DNSSEC to detect malicious redirects of IoT devices

DNSSEC provides message integrity in the DNS, which means that resolvers are able to detect adversaries making changes to the content of DNS messages (e.g., by injecting false information into a resolver's cache). This is particularly important in the IoT because manipulated DNS messages can redirect IoT devices to a malicious service, which jeopardizes users' privacy, safety, and well-being. In addition, users may not be aware of such redirects, for instance because devices may be non-interactive (e.g., a tiny sensor without a user interface) and operate autonomously. This is unlike traditional Internet applications where, in the typical case, users explicitly interact with remote content or services (e.g., using a browser or an app on a mobile device).

A particularly relevant type of attack that DNSSEC helps to detect is Border Gateway Protocol (BGP) hijacks, which occur routinely on the Internet [23]. BGP hijacks are unlike DNS cache poisoning attacks that directly target resolvers, and which DNSSEC was originally designed to protect against. DNS cache poisoning attacks are partly being mitigated by modern resolvers using better port randomization. Routing hijacks will likely continue to occur for some time because partial solutions to protect against them (e.g., RPKI [24]) are being deployed, but have not reached ubiquitous deployment yet.

As an example, consider an adversary redirecting device D1 (smart wrist watch) to a malicious service (M) through a route hijack (see *Figure 2*). The attacker accomplishes this by injecting malicious announcements into the Internet's routing system claiming that it owns the IP address range of S1's network, which results in the Internet routing D1's DNS queries and all other traffic to the adversary's network. To catch any DNS queries from D1 after the hijack, the adversary also sets up a malicious name server that answers to lookups for s1.sp1.example with the IP address of M instead of with the IP address of S1, thus effectively poisoning the cache of the resolver that D1 uses (cf. [11]). As a result, D1 connects to M when it queries for S1, which allows the malicious service to analyze or decrypt D1's sensor information, or even send instructions to D3 (the door lock) that cause it to act upon the user's physical environment.

DNSSEC works by digitally signing DNS records, which enables resolvers to validate these signatures using DNSSEC's chain of trust. Validation errors indicate that the response of an authoritative name server may have been tampered with, for instance as a result of a route hijack or man in the middle. In the example above, the DNSSEC-enabled resolver of D1 (R3) detects the redirect attempt because it cannot validate the DNS records received from the adversary's name server, for instance because they are unsigned or because the signatures were not valid. As a result, R3 returns an error to D1 to prevent transmitting D1's measurements to M.

3.3 DNS protocols to double-check the authenticity of IoT services

There also exist several DNS-based security protocols that provide IoT devices with additional means to check the authenticity of the certificates they receive from services after DNS resolution. Similar to protecting the integrity of DNS messages (Section 3.2), this is important to avoid IoT devices talking to malicious services that could impact the privacy and safety of users.

One example is DNS-based Authentication of Named Entities (DANE) [25], which enables IoT devices and gateways to validate the Transport Layer Security (TLS) certificates that they receive, such as when they connect to a remote service through HTTPS. For example, if device D1 uses HTTPS to connect to S1, it can look up S1's certificate in the DNS (using the name _443._tcp.s1.sp1.example) and check that the certificate it received through the HTTPS connection is (1) a valid certificate and (2) bound to S1. DANE is an application of DNSSEC and requires DNSSEC validation to work.

Another example is DNS Certification Authority Authorization (CAA) [26], which is a special DNS record through which a service indicates which certificate authorities are allowed to issue certificates for it. This avoids situations in which other authorities issue a certificate for the service either erroneously or because they were compromised.

3.4 Protecting IoT devices against domain registration hijacks

Registries and registrars are in a unique position to provide IoT-specific services that contribute to the safety, privacy, and transparency of the IoT for end-users. For example, the registrar of sp1.example (*Figure 2*) could provide multifactor authentication services for the administrative panel that service provider SP1 uses to update S1's DNS settings (e.g., a combination of a certificate, a one-time password, and an interactive check of SP1's identity through a human operator). This lowers the probability of domain hijacking, which occurs when someone makes unauthorized changes to S1's domain registration information, such as DNS records, resulting in IoT devices querying for S1 and ending up at a different (malicious) service. The effect of domain registration hijacking is similar to the effect of manipulating DNS transactions (Section 3.1), except that the hijack occurs on the registration side and can modify DNSSEC signing keys.

We speculate that domain registration hijacking campaigns will become more attractive for attackers in the future because it allows them to compromise users' physical environments (in addition to their IT systems). For example, the data that services collect about people's physical environments (e.g., door lock controllers, video monitoring) might make them high-value targets that result in attackers using more advanced spear phishing campaigns to obtain the credentials to a service's administrative panel. A high-profile domain hijacking campaign was reported by security company FireEye and others in January of 2019 [27]. [28] lists examples of domain registration hijacks that took place in the past.

3.5 Using DNS datasets to increase IoT transparency

DNS queries can act as a data source to provide users with more insight into the services that their IoT devices opaquely use and that potentially process their personal data (in line with the transparency guideline suggested by [10]). An example where this would provide added value is the light switch analyzed in [7], which users can turn on and off through an app on their phone. While this appears like a direct interaction between the light switch and the app via the local network, the researchers discovered that the on/off instructions from the phone actually travel to the switch via a remote service on the Internet.

One possibility to provide this insight is to visualize the DNS queries and other network traffic of a user's devices (e.g., the light switch and others) and enrich it with information on the geolocation and governing legal jurisdictions of the services that receive their data. An application like this needs to be able to co-exist with DoH/DoT (Section 3.1), making DNS queries only available for the legitimate user of the IoT devices and keep them DoH/DoT-encrypted for others. If IoT devices do not use DoH/DoT, then the application could also work by simply inspecting the DNS traffic on the path to its local resolver (e.g., in a home network).

There is a similar opportunity to provide users with more insight into, and control over, the DNS resolvers that their IoT devices use (see recommendation 1 in [29]). This is important because IoT device manufacturers could ship their products with a 3rd party resolver operator as their default, a model that browser manufacturers are currently exploring. The effect would be that the DNS queries of IoT devices would no longer go through the resolver of the user's local ISP, which traditionally provides this service as part of the user's Internet subscription. This is a potential problem because the DNS queries of IoT devices might reveal information about the device's type (e.g., if it is a light switch, a sleep tracker, a smart watch, or an autonomous vehicle) to the 3rd

party resolver operator that the user has no relationship with. In addition, the 3rd party resolver might fall under a different legal regime (e.g., a legal regime in the USA with the IoT devices based in Asia), which might make it more difficult for users to acquire rights—for example, in the case of a car accident caused by an outage or compromise of the resolver service. Another complicating factor is that IoT devices might not facilitate or even allow users to change their resolver settings, possibly because the device is a sensor that does not have the capabilities for a (remote) user configuration interface. In such cases, users would have to setup a proxy to redirect the queries, but this is typically a non-trivial task for most users.

4 Risks to the DNS from the IoT

We envision three ways in which the IoT can cause stress on the DNS: DNS-unfriendly programming at IoT-scale (Section 4.1), increased size and complexity of DDoS attacks powered by IoT botnets (Section 4.2), and an increased number of open DNS resolvers (Section 4.3). We identified these risks based on several published measurement studies (e.g., [12], [30], and [31]) and we speculate how their findings may evolve in the future. We consider them a risk because they might stress the DNS to the extent that it reduces the value of the DNS as a reliable service.

While these risks are not new, we expect that the IoT's characteristics (e.g., interaction with a user's physical world, a scale of tens of billions of widely heterogeneous deployments, and more autonomous operation of IoT devices) will pose new challenges for DNS operators and other infrastructure operators, as we will discuss in Section 5.

4.1 DNS unfriendly programming at IoT scale

A potential cause for an additional load on the DNS is IoT device engineers using the DNS naively. For example, after an update to iOS 6.0 in November 2012 [12], the TuneIn music app (a "classical" Internet application), started transmitting one DNS query per second for domains of the form www.<random-string>.com, perhaps to regularly check for network connectivity. The mobile network operator who observed the event reported around 1,000 of these queries per second from around 700 iPhones. The result was that the operator's DNS resolver's cache grew to about 5 million entries (normally around 400K) and its memory consumption increased to around 10 GB (normally around 4 GB), leading the operator to classify the event as a DDoS attack on its resolver. The network operator was unable to block the traffic because the devices were also making normal queries and instead had to wait until the new version of the app came out, which was about three weeks later.

In the IoT, incidents like the TuneIn app can have DNS-wide effects on resolvers. For example, a certain type of IoT device with a large installed base across many different networks and resolvers exhibiting TuneIn-like behavior may cause stress on the local DNS resolvers in those networks because they fill up their caches and run out of memory, resulting in packet drops or increased response latency. A similar event would be a large number of IoT devices coming back online after a power outage and all trying to locate their remote services almost simultaneously. The actual effects will however be difficult to estimate in advance because they depend on various operational and site-specific factors such as concentrations of IoT devices across networks, how often they resolve a domain name (e.g., depending on real-world triggers), and the TTL of domain names of IoT services. The impact that these incidents have on the authoritative side of the DNS

would likely be limited because resolvers typically use negative caching (i.e., they cache responses for domain names that do not exist) with a default TTL between one and three hours [32].

From a long-term perspective, another risk is that a large percentage of IoT devices continue to be shipped without a timely software update function. This means that TuneIn-like DNS traffic patterns might persist for an extended period of time because IoT devices often operate unattended and have much longer lifespans than today's PCs and laptops, perhaps on the order of decades. This could result in a steadily increasing number of queries for non-existing domain names that resolver operators and TLD operators would need to handle ("DNS background noise"). This load would come on top of the already significant existing load of queries for non-existing domain names. For example, the percentage of queries for non-existing domains that the .nl operator handles increased from around 5% in 2014 to 10% in 2019 [33]. Similar effects could surface when manufacturers discontinue remote services or software updates, for instance when they stop selling a certain IoT product or when they go out of business.

A possible root cause is that IoT device engineers rely on open source stacks (Linux variants) that hide the details of networking functions from them. As a result, they are less familiar with how the DNS works and the Internet-scale effects of "DNS-unfriendly" programming. Another cause may be simple programming errors in IoT client software or software developers making false assumptions about domain names [34].

4.2 Increased size and complexity of IoT botnets targeting the DNS

The second risk we identify is IoT botnets that are able to hit the DNS (and other types of Internet infrastructure) with large coordinated DDoS attacks. An IoT botnet consists of a set of IoT devices (e.g., IP cameras and DVRs) that have been compromised by malicious software. The devices in a botnet (individually referred to as "bots") are controlled by a botnet master, which can, for instance, instruct them to simultaneously send traffic to a specific target (e.g., a DNS operator [30]), thus together carrying out a DDoS attack. The devices in an IoT botnet can be IoT devices (device-to-cloud IoT deployments) or gateways (device-to-gateway deployments).

Bots infect IoT devices over a network by logging on to them (e.g., using Telnet, predefined or easy-to-guess credentials [30]) or by exploiting software vulnerabilities (e.g., Hajime bypassed the authentication mechanisms of some routers by appending "?images" to any of the URLs that the routers used [35]). They then copy the bot malware (binary) onto the compromised device, for instance from a special download server [30]. The bot master controls its botnet through a central Command and Control (C&C) server, through a peer-to-peer system, or through a combination of these [36] [37]. We do not consider other ways of infecting IoT devices, such as through a physical interaction.

IoT botnets behave similarly to traditional botnets, which exploit PCs, laptops, and mobile phones [36]. For example, both types of botnets scan for vulnerable devices, infect them, and launch DDoS attacks, and some are available as a service for a few tens of Euros per hour [38].

The difference between the two, however, is that IoT botnets operate in a different environment:

- *IoT botnets can run on a wider range of devices.* IoT devices use a range of CPU architectures, hardware, and operating systems [35]. As a result, removing IoT botnets may

require device-specific (and perhaps manual) cleanup procedures, which makes it more difficult to reduce the number of infected IoT devices in an IoT botnet quickly and at scale.
- *Vulnerable IoT devices can be more difficult to fix quickly at scale.* For example, because they interact with people's physical environment, they may require more tedious and more time-consuming repair for safety reasons. The study of the Hajime botnet [35] suggests that Hajime bots had longer lifetimes than bots previously installed in other ways.
- *An IoT bot may have multiple ways of infecting another IoT device.* Infection could occur via the Internet or via a direct link-level connection (proximity infection).
- *Infections of IoT devices often stay undetected longer.* IoT devices often operate "in the background" and may not be salient to end-users (e.g., pressure sensors in floor tiles).
- *IoT devices are more likely to always be online than traditional devices like laptops.* This means they can be compromised and misused unceasingly.
- *IoT device engineers may have limited expertise with networking (Section 4.1) and with device and network security in particular.* As a result, IoT devices are potentially more vulnerable than traditional Internet devices, for instance, because they use libraries that are vulnerable to exploits for which mitigations are known but not installed.

We speculate that these differences make IoT botnets a new challenge for DNS operators and other Internet infrastructure operators. Specifically, they will have to deal with:
- Larger DDoS attacks, partly because IoT bots are more difficult to eradicate. Current botnet sizes are on the order of hundreds of thousands. The most well-known example is the Mirai botnet [30], which involved 400K (steady-state) to 600K (peak) infected IoT devices and whose DDoS traffic reduced the services of DNS operator Dyn and hosting provider OVH, amongst others. The Hajime botnet hovers around 400K infected IoT devices, but has not launched any DDoS attacks yet [35]. With the growth of the IoT, these attacks may grow to involve millions of bots and as a result larger DDoS attacks.
- Higher DDoS complexity, because the set of IP addresses that an IoT botnet uses may be large (hundreds of thousands) and may change quickly. For example, the Hajime botnet has a churn of around 2K bots (bots recruited into and leaving the botnet) per 20-minute interval [35]. This makes it extremely challenging to filter out botnet traffic in incoming DDoS traffic based on IP addresses because the target observes a large and continuously changing set of DDoS sources. Blocking the wrong IP addresses may affect other non-infected IoT devices, for instance if the blocked device is a gateway.
- Higher propagation rates, for instance because botnet developers are designing their bots to quickly infect devices through newly found vulnerabilities. For example, the Hajime botnet received an update to its software to exploit and infect Gigabyte Passive Optical Network (GPON) routers through a vulnerability that was published only 10 days earlier [35]. Propagation speeds to 100K infections are currently taking on the order of days. For example, the number of Hajime infections jumped from around 42K bots to 71K bots in an hour and then to 93K in the next 24 hours [35]. Similarly, Mirai reached around 100K active bots in about 36 hours [30].
- Reduced advantage of DDoS localization through anycast when an IoT botnet manages to infect a large number of device manufacturers across regions. For example, a perfectly evenly distributed DDoS attack of 1 Tbps would hit 1,000 name servers with 1 Gbps each, which some of them might not be able to handle [39]. The propagation of IoT botnets across networks has been uneven until now. For example, Mirai infections were

> concentrated in South America and South-East Asia, such as in Brazil (15%), Colombia (14%), and Vietnam (12.5%). Also, the top 100 infected Autonomous Systems (ASs) accounted for 78.6% of all infections and the top 10 ASs accounted for 44.3% of all infections.

As a result, we expect that potential targets such as DNS operators will need to enhance their anti-DDoS capabilities to quickly scale up their DDoS mitigation services (e.g., DDoS scrubbing services or traffic filtering rules), and proactively search for DDoS-for-hire sites in DNS zones, as we will discuss in Section 5.

4.3 Increased DDoS amplification through open DNS resolvers

"Open" resolvers have been misconfigured to accept DNS queries from any client on the Internet, rather than restricting access to clients within the domain that they are intended to serve (e.g., an ISP network or a home network). Attackers can take advantage of such misconfigurations by sending many DNS requests to an open resolver with the query's source IP addresses set to spoof a victim's IP address. As a result, the resolver will send any responses to the victim instead of to the attacker, adding an amplification factor because DNS responses are usually larger than DNS requests.

The number of open resolvers on the Internet is on the order of millions, with [31] estimating 23-25 million open resolvers in 2014 and Shadowserver reporting over 3 million open resolvers based on their active scanning system (Dec 2018) [40]. Rossow [41] reported amplification factors between 29 and 64 (i.e., DNS responses are 29 to 64 times the size of requests) for open resolvers in the wild.

While open resolvers are a longtime problem [42], they represent an additional risk to the IoT. This is because Mirai has demonstrated that a botnet of several 100K bots can launch direct DDoS attacks on DNS operators that can lead to large-scale service outages (Section 4.2), which would potentially be tens of times higher if they were amplified through a set of open resolvers. There is anecdotal evidence that there are IoT botnets, such as the Reaper botnet [43], that are capable of exploiting open resolvers.

Another risk is that IoT device engineers may have limited expertise with networking (Section 4.1) and with device/network security (Section 4.2), and as a result introduce vulnerabilities that enable adversaries to install an open resolver on their devices, similar to how bots infect other IoT devices (Section 4.2). Similarly, IoT device engineers might link a DNS resolver into their IoT software (e.g., through a third-party module) and then accidentally misconfigure it. While both risks also exist with laptops and smart phones, open resolvers may be more persistent on IoT devices because IoT devices are typically harder to upgrade and because they often operate unattended. This could result in a gradual increase in the number of open resolvers running on IoT devices that only disappear when devices disconnect or reach their end of life.

5 Challenges for the DNS and IoT industries

In this section, we discuss 5 challenges we have identified for the DNS and IoT industries to take advantage of the opportunities we discussed in Section 3 and reduce the risks discussed in Section

4. We also provide starting points for these challenges. Our challenges complement and detail more generic IoT security challenges (e.g., those discussed in [1] and [9]), specifically focusing on the DNS.

The challenges we identified are:

- Developing a DNS library for IoT devices (Section 5.1) that makes the DNS's security functions (e.g., DNSSEC validation and DoH/DoT) available for device control applications and that uses DNS query data to make IoT deployments more transparent for users (Sections 3.1, 3.2, and 3.4).
- Training IoT and DNS professionals (Section 5.2) to help DNS players such as registrars and registrants understand the implications of providing services for domain names that act as a backend for IoT devices rather than as a means for making content available to humans (Section 3.3), and to help IoT device manufacturers understand how to use the DNS and how to configure resolvers (Sections 4.1 and 4.3).
- Developing a shared system that enables different DNS operators to automatically and continually share information on IoT botnets (Section 5.3), allowing them to more quickly respond to rapidly growing botnets and the DDoS attacks they generate (Section 4.2).
- Developing systems that enable DNS operators to share DDoS handling capacity and that stop attacks in an early stage in edge networks (Section 5.4), so DNS operators can better handle very large (amplified) IoT-powered DDoS attacks (Sections 4.2 and 4.3).
- Developing a system that enables DNS operators to measure how the IoT uses the DNS (Section 5.5), to better understand how IoT risks evolve (Section 4)—for instance to develop new domain name policy or for incident response purposes.

These challenges address some of the generic IoT security requirements outlined by organizations such as the European Telecommunications Standards Institute (ETSI) [44] and the European Union Agency for Network and Information Security (ENISA) [14], but specifically for the DNS. For example, a shared system that enables DNS operators to exchange information on the characteristics of IoT botnets (Section 5.3) helps fulfill ENISA's requirement to "[p]articipate in information sharing platforms to report vulnerabilities and receive timely and critical information about current cyber threats and vulnerabilities from public and private partners. Information sharing is a critical tool in ensuring stakeholders are aware of threats as they arise" [14].

We emphasize that our challenges go beyond ICANN's and SSAC's remit, which is why they are not SSAC recommendations.

5.1 Developing a DNS security and transparency library for IoT devices

The first challenge is developing and maintaining an open source DNS security library that makes functions like DNSSEC validation, DANE, and DoH/DoT available on IoT devices. The library would need to include transparent support for routine root KSK rollovers, such as automatically changing keys and notifying the DNS of which KSK an IoT device is currently using [45]. This will likely be more challenging than with traditional resolvers, partly because it may not be possible to remotely update IoT devices, and many IoT devices lack the non-volatile storage necessary to store keying material.

The library would need to work with the most popular IoT operating systems (e.g., OpenWRT and RIOT) and CPU architectures, and for the more limited resources of IoT devices (e.g., limited battery power, CPU power, or capacity to perform cryptographic operations). It would also need to offer an API that enables IoT device engineers to easily include and use the library. Potential starting points for developing the DNS security library might be Danish [46] (which makes HTTPS DANE available on OpenWRT) or DNSSEC Trigger [47] (which provides DNSSEC signature validation for laptops and phones).

Another challenge is to make the DNS security features of IoT deployments more visible to and configurable by end-users, for instance by including user-level tools in the DNS security library. For example, users may want to know if an IoT device or gateway they just bought supports DNS encryption and which resolver operator it uses to provide these functions (e.g., a public resolver operator or the resolver of the user's ISP). While most users are unaware of DNS security and mostly cannot influence it today, the IoT may make this a stronger requirement so that users stay in control of how and where IoT deployments send information about their physical environment.

Similarly, the library could also increase IoT transparency for users by making use of the DNS queries that IoT devices transmit (cf. the challenge of users verifying device behavior described in [9], Section 5.6). One way to accomplish this would be to extend shared resolvers in the user's local network with a "DNS transparency module" that enables users to register the IP addresses of their IoT devices with the resolver, and then obtain regular reports from the resolver on the queries it received from these devices. The advantage of co-locating the service with a shared local resolver is that the service can monitor multiple IoT devices and still allow clients to encrypt their DNS traffic (e.g., using DNS over HTTPS). A potential starting point might be the Security and Privacy for In-home Networks (SPIN) visualizer, which renders the DNS query patterns of IoT devices in real-time [48].

The DNS library may also include other helpful functions, such as support for multiple recursive resolvers so that an IoT device can automatically failover to a secondary recursive DNS operator, for instance to switch to a public resolver when the resolvers of a user's ISP are temporarily unavailable due to a DDoS attack [49]. This would enable IoT devices to better survive DNS outages, which would reduce the risk of a device (e.g., a door lock) failing to function as the user expects because it cannot locate its remote service. The library would also need to be intelligent enough to switch back to the primary resolver when it is available again, which avoids IoT devices continuing to use the secondary resolver when the primary one is back online (e.g., when the user configured the secondary manually at the time the primary became unavailable and then forgot about it).

A related and ongoing challenge is increasing the deployments of DNSSEC (signing and validation) and DANE, which are currently limited.

5.2 Training IoT and DNS professionals

Another challenge is to provide multilingual (online) training on the interactions between the IoT and the DNS for IoT and DNS professionals, such as IoT product managers, IoT device engineers, and product managers of registries and registrars.

IoT product managers would benefit from the training in that they get a deeper understanding of Internet security topics (e.g., IoT botnets and open resolvers) and the potential effects of insecure IoT deployments on their customers and on the security and stability of the Internet. The training would also discuss IoT security measures (e.g., the DNS's security functions and other security protocols) as well as guidelines to calculate the costs of developing and maintaining them.

IoT device engineers would benefit in that they would get a better understanding of what constitutes "DNS friendly" programming (e.g., avoid many IoT devices aggressively sending DNS requests), DNS programming for long-living devices (e.g., adding an "application lifetime" so that they at some point stop querying the DNS), and "programming for network failure" (e.g., devices having the ability to dynamically switch between a primary and a secondary resolver). This would include an increased understanding of device and network security, such as how DNSSEC works (e.g., related to Section 5.1).

Product managers of registries and registrars would learn how IoT deployments use the DNS and what this means for the registration services they provide to registrants. For example, the training could stimulate them to further increase the security of their registration services, perhaps by providing multifactor authentication for registrants who run services for IoT devices, by adding more advanced monitoring of name server changes to detect anomalies (e.g., name server changes or transfers), or by delaying the release of a domain name when a device manufacturer with a large installed base of IoT devices goes out of business to avoid domain speculators re-registering the domain name and picking up the DNS traffic. An important message to convey is that the IoT might be changing the traditional model of how humans register meaningful names to deliver content or services to other humans and that this puts new requirements on registration services.

To develop the training, both the DNS and the IoT communities need to understand how they currently perceive IoT security for the DNS, how they use it, and what they require from it. RFC4367 [50] and PowerDNS's "Hello DNS" site [51] might provide good starting points to develop the training for IoT engineers.

With both the IoT and the DNS co-evolving quickly and new people joining both industries, this type of training will also need continual updating.

5.3 Developing a system to share information on IoT botnets

Our third challenge is to develop a shared system that enables a DNS operator who gets attacked by an IoT-powered DDoS to automatically share the characteristics of the DDoS traffic (e.g., volumetric and TCP state exhaustion attacks [30]) with other DNS operators, perhaps including information from the victim on how they handled the attack (e.g., in terms of filtering rules). This helps the other DNS operators to more quickly write their filtering rules or set up other measures in case the attack targets them (Section 5.5). As a result, they can proactively prepare for DDoS attacks, which is particularly important in the IoT because IoT botnets can grow quickly in size and can (quickly) vary the types of DDoS traffic they generate (Section 4.2). In the end, this type of information sharing would enable DNS operators to provide a better service to their customers because writing filtering rules during an attack usually takes place under intense pressure and is therefore more error prone than doing it before the attack.

Conceptually, we envision DNS operators sharing IoT botnet information through a joint database that contains an entry for each IoT botnet that generated a "considerable" amount of DDoS traffic (e.g., 500Gbps or more), with each entry describing the attributes of the botnet and the rules that DNS operators have used to filter the botnet's DDoS traffic (e.g., for different router platforms). DNS operators and vetted security researchers could potentially enrich this information over time by automatically and continually adding measurements on a botnet's deployment and behavior. Examples include longitudinal measurements of bot concentrations across autonomous systems, "fingerprints" of the DDoS traffic they generate (e.g., in terms of port numbers, domain names used, and packet lengths), booter sites that use the botnet, and open resolvers that the botnet uses for amplification. DNS operators could also provide DNS message sequences of a botnet's C&C similar to how the D-root operator mapped Hajime infections [35].

A few prototypes of shared systems for exchanging DDoS information across multiple collaborating players are under development and are potential starting points for a shared system for DNS operators. An example is 3DCoP (DDoS Defense for a Community of Peers) [52], a prototype peer-to-peer system to detect DDoS attacks and share that information among network operators. Another example is the Dutch national DDoS clearing house [53], which is currently being developed by several ISPs, banks, and government agencies in the Netherlands. It enables groups of service providers to automatically and continually create and share "fingerprints" of DDoS attacks based on packet captures.

Sources that may enrich the botnet information in the shared database include:

- IoT-Pot by Saarland University and Yokohama National University [54], which is a honeypot to analyze the behavior of IoT botnets for various IoT CPU architectures.
- Shadowserver's Open Resolver Scanning Project [40], which could help to identify resolvers that IoT botnets have used or could use for reflection attacks.

The system should preferably be implemented in a fully distributed way (like 3DCoP) because it may become a target for DDoS attacks itself.

5.4 Proactive and flexible mitigation of IoT-powered DDoS traffic

We envision different complementary DDoS mitigation systems and measures that help protect DNS operators and other Internet infrastructure operators against IoT-powered DDoS attacks in a more proactive and flexible way. This is important because IoT botnets enable complex and amplified DDoS attacks that can grow to several hundreds of thousands of infected devices within hours and in the future perhaps to millions.

On the DNS side, we envision a system that enables DNS operators to flexibly share DDoS mitigation capacity. This is helpful when an individual operator can no longer handle the DDoS traffic on its own and needs to quickly scale up its mitigation capacity through one or more third parties (e.g., their upstream transit providers). Scenarios like these are likely with the increasing size, complexity, and propagation speed of IoT botnets (Section 4.2).

One possible way to accomplish this would be through a "DDoS mitigation broker", a to be developed automated system which would enable DNS operators that get hit by a DDoS attack to dynamically locate other DNS operators or 3rd party scrubbing services that are willing to help in

mitigating the attack (e.g., through additional scrubbing capacity). Such scenarios would become possible with the DDoS Open Threat Signaling (DOTS) protocol [55], which is currently being developed in the IETF and enables a network to signal to another organization that it needs additional DDoS mitigation capacity.

Another DNS-side measure would be to develop a "playbook" for DNS operators on how to deal with the unique properties of IoT-powered DDoS attacks (e.g., in terms of propagation rate and bot churn). This would enable them to proactively adapt the design of their DNS anycast services and their operational procedures, perhaps in collaboration with the ICANN TLD-OPS Standing Committee [56], the incident response community for ccTLDs.

In parallel, we envision security systems in edge networks (e.g., on home gateways) that help mitigate IoT-powered DDoS attacks by automatically blocking traffic from devices that are part of a botnet. This would proactively stop IoT-powered DDoS attacks close to the source, thereby reducing the amount of DDoS traffic that DNS operators would have to handle. Traffic blocking requires functions like distributed traffic measurements in edge networks, advanced anomaly detection, interactions with users when a security system has temporarily blocked one of their devices, and coexistence of and interoperability among different security systems in the same network.

One way to accomplish this is through a Manufacturer Usage Description (MUD) [57], an IETF standard for describing a device's expected network behavior,—for example, by describing what domain names and protocols are used. Security systems in edge networks could use MUDs to whitelist a device's normal behavior and block all other traffic, such as outbound DDoS traffic. Device manufacturers could ship a MUD with their devices, but security systems could also create MUDs themselves based on network measurements of baseline behavior [58]. Examples of security systems for edge networks are SIDN Labs' SPIN system [48], CIRA's secure home gateway [59], and IIT/CNR's NTOP system [60].

5.5 Developing a system to measure how the IoT uses the DNS

Another challenge we identified is to develop and operate a system that measures how IoT deployments use the DNS and how this evolves over time. This is important for domain name policymaking at ICANN and elsewhere, for instance, if and how domain names that IoT devices use should be treated differently when they may impact users' safety (e.g., they might need more secure registration services). It is also important for incident response purposes, for example to understand the concentrations of IoT devices that are part of a botnet across networks.

Conceptually, the system would consist of a (distributed) device-to-domain name database containing an entry per known type of IoT device and the domain names it uses (e.g., hello-audio.s3.amazonaws.com and sense-in.hello.is for the sleep tracker in [7]). The system could for instance automatically collect publicly available Manufacturer Usage Descriptions (MUDs) [57] of IoT devices and use the domain names in these descriptions to fill the database.

Each entry in the database could also contain an indication of how often the domain names were being queried (e.g., the number of queries for hello-audio.s3.amazonaws.com, perhaps expressed as a range), which is information that DNS operators could collaboratively make available. For

example, a public resolver operator could provide the total number of queries for domain name sense-in.hello.is that it handled in a particular month.

DNS operators could also provide more granular statistics, such as the autonomous systems where DNS queries come from, which provides an indication of device concentrations across the Internet. This could help in estimating the effects of events such as a routing hijack that affected IoT devices.

6 Conclusions and future work

The IoT is an emerging distributed application that is expected to further ease our daily lives and make our society safer and more sustainable by autonomously and seamlessly interacting with our physical world through billions of connected sensors and actuators. A significant number of IoT devices will likely be IP enabled and will use the DNS to locate the remote services they require to perform their functions. As a result, the DNS will continue to play the same crucial role for the IoT that it has for traditional applications that enable human users to interact with services and content. The role of the DNS might become even more crucial from a security and stability perspective with IoT devices interacting with people's physical environment.

The contribution of this report for the ICANN community is that we provide a tutorial-style overview of the DNS and the IoT as two co-evolving and interacting ecosystems, with the purpose of triggering and facilitating dialogue in the broader ICANN community. We motivated how the IoT might form an opportunity for the DNS from a security and stability perspective (e.g., the DNS security functions to protect users' safety and privacy), and used various measurement studies to illustrate IoT-related risks that might stress the DNS (e.g., complex DDoS attacks launched from large and quickly propagating IoT botnets). We identified 5 challenges to seize this opportunity and address the risks and provided starting points for these challenges.

Our next step will be to discuss our report with the ICANN community, for instance in the form of a workshop or a public webinar. Depending on community feedback, we may also prioritize the challenges of Section 5 and flesh out a few of them through a separate project like a multidisciplinary feasibility analysis (e.g., technical, legal, regulatory, and business).

7 Acknowledgments, Statements of Interests, and Dissents and Withdrawals

In the interest of transparency, these sections provide the reader with information about four aspects of the SSAC process. The Acknowledgments section lists the SSAC members, outside experts, and ICANN staff who co-authored or contributed directly to this particular document (Contributors) or who provided reviews (Reviewers). The Statements of Interest section points to the biographies of all SSAC members, which disclose any interests that might represent a conflict—real, apparent, or potential—with a member's participation in the preparation of this Report. The Dissents section provides a place for individual members to describe any disagreement that they may have with the content of this document or the process for preparing it. The Withdrawals section identifies individual members who have recused themselves from discussion of the topic with which this Report is concerned. Except for members listed in the Dissents and Withdrawals section, this document has the consensus approval of all of the members of SSAC.

7.1 Acknowledgments

The committee wishes to thank the following SSAC members for their time, contributions, and review in producing this report.

Contributors
Tim April
Lyman Chapin
kc claffy
Cristian Hesselman (work party chair)
Merike Kaeo
Jacques Latour
Danny McPherson
Dave Piscitello
Rod Rasmussen
Mark Seiden

Reviewers
Jaap Akkerhuis
Paul Ebersman
Patrik Fältström
James Galvin
Robert Guerra
Julie Hammer
Geoff Huston
Andrei Kolesnikov
Warren Kumari
John Levine
Tara Whalen

External Reviewers
Elmer Lastdrager
Caspar Schutijser

ICANN staff
Andrew McConachie (editor)
Kathy Schnitt
Steve Sheng

7.2 Statements of Interest

SSAC member biographical information and Statements of Interest are available at:
https://www.icann.org/resources/pages/ssac-biographies-2019-01-08-en

7.3 Dissents and Withdrawals

There were no dissents or withdrawals.

References

1. K. Rose, S. Eldridge, and L. Chapin, "The Internet of Things: an Overview", ISOC, Oct. 2015, http://www.internetsociety.org/sites/default/files/ISOC-IoT-Overview-20151221-en.pdf
2. A. Nordrum, “Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated”, https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated
3. N. Asokan, F. Brasser, A. Ibrahim, A.-R. Sadeghi, M. Schunter, G. Tsudik, and C. Wachsmann, “SEDA: Scalable Embedded Device Attestation”, CCS’15, Denver, Colorado, USA, October 2015
4. M. Weiser, "The Computer for the 21st Century", Scientific American Special Issue on Communications, Computers, and Networks, September 1991
5. Javid Habibi, Daniele Midi, Anand Mudgerikar, and Elisa Bertino, “Heimdall: Mitigating the Internet of Insecure Things”, IEEE Internet of Things Journal, Vol. 4, No. 4, Aug 2017
6. Eireann Leverett, Richard Clayton & Ross Anderson, “Standardisation and Certification of the `Internet of Things'", 16th Annual Workshop on the Economics of Information Security (WEIS2017), USA, June 2017, https://www.cl.cam.ac.uk/~rja14/Papers/weis2017.pdf
7. N. Apthorpe, D. Reisman, N. Feamster, “A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic”, Workshop on Data and Algorithmic Transparency (DAT '16), New York University Law School, November 2016, https://arxiv.org/abs/1705.06805
8. “Students give SIDN Labs course thumbs up”. https://www.sidnlabs.nl/a/weblog/students-give-sidn-labs-course-thumbs-up?language_id=2
9. O. Garcia-Morchon, S. Kumar, and M. Sethi, “Internet of Things (IoT) Security: State of the Art and Challenges”, RFC8576, April 2019, https://datatracker.ietf.org/doc/rfc8576/
10. H. Tschofenig, J. Arkko, and D. McPherson, “Architectural Considerations in Smart Object Networking”, RFC7452, March 2015, https://www.rfc-editor.org/rfc/rfc7452.txt
11. “BGP Leaks and Crypto Currencies”, Blog, April 2018, https://blog.cloudflare.com/bgp-leaks-and-crypto-currencies/
12. G. Choules, “Cache Attacks”, DNS-OARC Spring Workshop, Dublin, May 2013, https://indico.dns-oarc.net/event/0/contributions/3/
13. Draft ICANN Strategic Plan for Fiscal Years 2021-2025, Dec 20, 2018, https://www.icann.org/en/system/files/files/strategic-plan-2021-2025-draft-20dec18-en.pdf
14. ENISA, “Baseline Security Recommendations for IoT”, November 2017, https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot
15. Ala Al-Fuqaha, Abdallah Khreishah, Mohsen Guizani, Ammar Rayes, and Mehdi Mohammadi , “Toward Better Horizontal Integration Among IoT Services,” IEEE Communications Magazine, Communications Standards Supplement, September 2015, http://homepages.dcc.ufmg.br/~mmvieira/cc/papers/Toward%20better%20horizontal%20integration%20among%20IoT%20services.pdf
16. R. Want, B. N. Schilit, and S. Jenson, “Enabling the Internet of Things,” Computer, no. 1, pp. 28–35, January 2015, https://www.computer.org/cms/Computer.org/ComputingNow/issues/2015/07/T-mco2015010028.pdf

17. Tianlong Yuy, Vyas Sekary, Srinivasan Seshany, Yuvraj Agarwaly, Chenren Xuz, “Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the Internet-of-Things”, HotNets’15, November 2015, Philadelphia, USA
18. S. Cheshire and M. Krochmal, “Multicast DNS”, RFC6762, Feb 2013, https://tools.ietf.org/rfc/rfc6762.txt
19. G. Moura, J. Heidemann, M. Müller, R. de O. Schmidt, M. Davids, “When the Dike Breaks: Dissecting DNS Defenses During DDoS”, ACM Internet Measurement Conference 2018, Boston, USA, Oct-Nov 2018
20. P. Hoffman, P. McManus, “DNS Queries over HTTPS (DoH)”, RFC8484, Oct 2018, https://tools.ietf.org/html/rfc8484
21. S. Dickinson, D. Gillmor, and T. Reddy, “Usage Profiles for DNS over TLS and DNS over DTLS”, RFC8310, March 2018, https://tools.ietf.org/html/rfc8310
22. B. Hubert, “Opinion: DNS privacy debate”, blog, Feb 2019, https://blog.apnic.net/2019/02/08/opinion-dns-privacy-debate/
23. BGPmon Twitter feed, https://twitter.com/bgpmon
24. M. Lepinski and S. Kent, “An Infrastructure to Support Secure Internet Routing”, RFC6480, Feb 2012, https://tools.ietf.org/html/rfc6480
25. P. Hoffman, J. Schlyter, “The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA”, RFC6698, Aug 2012, https://tools.ietf.org/html/rfc6698
26. P. Hallam-Baker and R. Stradling, "DNS Certification Authority Authorization (CAA) Resource Record", RFC6844, January 2013, https://datatracker.ietf.org/doc/rfc6844/
27. M. Hirani, S. Jones, B. Read, “Global DNS Hijacking Campaign: DNS Record Manipulation at Scale”, blog, Jan 2019, https://www.fireeye.com/blog/threat-research/2019/01/global-dns-hijacking-campaign-dns-record-manipulation-at-scale.html
28. “SSAC Advisory on Registrant Protection: Best Practices for Preserving Security and Stability in the Credential Management Lifecycle”, November 2015, https://www.icann.org/en/system/files/files/sac-074-en.pdf
29. J. Livingood, M. Antonakakis, B. Sleigh, and A. Winfield, “Centralized DNS over HTTPS (DoH) Implementation Issues and Risks”, Internet Draft, March 2019, https://www.ietf.org/id/draft-livingood-doh-implementation-risks-issues-03.txt
30. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z., Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, “Understanding the Mirai Botnet”, 26th USENIX Security Symposium, 2017, https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-antonakakis.pdf
31. M. Kührer, T. Hupperich, C. Rossow, T. Holz, “Exit from Hell? Reducing the Impact of Amplification DDoS Attacks”, 23rd USENIX Security Symposium, USENIX Sec 2014, San Diego, USA
32. M. Andrews, “Negative Caching of DNS Queries (DNS NCACHE)”, RFC2308, March 1998, https://tools.ietf.org/html/rfc2308
33. .nl stats and data, graph “Response Codes”, https://stats.sidnlabs.nl/en/dns.html#response%20code
34. J. Rosenberg, “What's in a Name: False Assumptions about DNS Names”, RFC4367, February 2006, https://tools.ietf.org/html/rfc4367

35. S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, “Measurement and Analysis of Hajime, a Peer-to-peer IoT Botnet”, Network and Distributed Systems Security (NDSS) Symposium 2019, San Diego, CA, USA, February 2019
36. J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon, “Peer-to-peer botnets: Overview and case study”, USENIX HotBots’07, 2007
37. G. Vormayr, T. Zseby, and J. Fabin, “Botnet Communication Patterns”, IEEE Communications Surveys & Tutorials, Vol. 19, No. 4 , Fourth quarter 2017
38. “DDoS-er caught -- how the attacker’s cover was blown” (in Dutch), Tweakers, February 2018, https://tweakers.net/reviews/6031/een-ddoser-betrapt-hoe-de-aanvaller-tegen-de-lamp-liep.html
39. D. Conrad, “Thoughts on DDOS and the Root Server System”, ICANN63, Barcelona, October 2018, https://static.ptbl.co/static/attachments/192183/1540457869.pdf?1540457869
40. Open Resolver Scanning Project, https://dnsscan.shadowserver.org/
41. C. Rossow, “Amplification Hell: Revisiting Network Protocols for DDoS Abuse”, Network and Distributed System Security Symposium, NDSS 2014, San Diego, USA
42. “SSAC Advisory on DDoS Attacks Leveraging DNS Infrastructure”, Feb 2014, https://www.icann.org/en/groups/ssac/documents/sac-065-en.pdf
43. “IoT_reaper: A Rappid Spreading New IoT Botnet”, Oct 2017, http://blog.netlab.360.com/iot_reaper-a-rappid-spreading-new-iot-botnet-en/
44. “CYBER; Cyber Security for Consumer Internet of Things”, ETSI TS 103 645 V1.1.1, Feb 2019, https://www.etsi.org/deliver/etsi_ts/103600_103699/103645/01.01.01_60/ts_103645v010101p.pdf
45. D. Wessels, W. Kumari, and P. Hoffman, “Signaling Trust Anchor Knowledge in DNS Security Extensions (DNSSEC)”, RFC8145, April 2017, https://www.rfc-editor.org/rfc/rfc8145.txt
46. https://github.com/smutt/danish
47. DNSSEC Trigger Homepage, https://nlnetlabs.nl/projects/dnssec-trigger/about/
48. SPIN homepage, https://spin.sidnlabs.nl/en/
49. “Ziggo solves second outage by fighting off DDoS attack on its DNS” (in Dutch), August 2015, https://tweakers.net/nieuws/104853/ziggo-lost-tweede-storing-op-door-ddos-aanval-op-dns-af-te-weren.html
50. J. Rosenberg (Ed.), “What's in a Name: False Assumptions about DNS Names”, RFC4367, Feb 2006, https://www.rfc-editor.org/rfc/rfc4367.txt
51. “Hello DNS”, PowerDNS, https://powerdns.org/hello-dns/
52. J. Berkes and A. Wick, “DDoS Defense for a Community of Peers”, Presentation, FloCon 2017, San Diego, USA, January 2017, https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=497970
53. C. Hesselman, “Collaboratively increasing the resilience of critical services in the Netherlands through a national DDoS clearing house”, Triple-I Security Day at APRICOT2019, Feb 2019, https://www.sidnlabs.nl/downloads/presentations/collaborative%20ddos%20mitigation.pdf
54. Yin Minn Pa Pa, Shogo Suzuki, Katsunari Yoshioka, Tsutomu Matsumoto, Takahiro Kasama, Christian Rossow. “IoTPOT: Analysing the Rise of IoT Compromises”. 9th USENIX Workshop on Offensive Technologies (co-located with USENIX Sec '15), WOOT '15, Washington, DC, http://www.christian-rossow.de/publications.php

55. Dobbins, D. Migault, S. Fouant, R. Moskowitz, N. Teague, L. Xia, and K. Nishizuka, “Use cases for DDoS Open Threat Signaling”, Internet Draft, draft-ietf-dots-use-cases-16, July 2018, https://www.ietf.org/id/draft-ietf-dots-use-cases-16.txt
56. TLD-OPS Homepage, https://ccnso.icann.org/en/resources/tld-ops-secure-communication.htm
57. E. Lear, R. Droms, and D. Romascanu, “Manufacturer Usage Description Specification”, IETF Internet Draft, April 2018, https://www.ietf.org/id/draft-ietf-opsawg-mud-20.txt
58. C. Schutijser, “Towards automated DDoS abuse protection using MUD device profiles”, M.Sc. Thesis, University of Twente, August 2018
59. CIRA Secure Home Gateway Homepage, https://cira.ca/cira-secure-home-gateway
60. NTOP Homepage, https://www.ntop.org/