# Internet Kill Switches Demystified

Benjamin Rothenberger
ETH Zürich
rothenbb@inf.ethz.ch

Daniele E. Asoni
ETH Zürich
daniele.asoni@inf.ethz.ch

David Barrera
ETH Zürich
david.barrera@inf.ethz.ch

Adrian Perrig
ETH Zürich
adrian.perrig@inf.ethz.ch

## Abstract

Internet kill switches are possible in today's Internet, but to date have been locally-scoped and self-inflicted. As more networks move towards centralized key architectures such as DNSSEC and BGPsec, adversarial kill switches become more powerful. We analyze the feasibility of and mechanisms for executing kill switches on remote DNSSEC- or BGPsec-enabled networks, finding that kill switches must be considered in the design of next generation Internet protocols. We also describe recovery procedures and properties intended to evaluate kill switch events, finding that recovering from a compromised key may take up to 48 hours.

## Categories and Subject Descriptors

C.2.0 [**COMPUTER-COMMUNICATION NETWORKS**]: General—*Security and protection*

## Keywords

Kill switches; Centralized key architectures; BGPsec; DNSSEC

## 1. INTRODUCTION

Recent history has seen multiple incidents of country-wide Internet censorship [9, 31]. Egypt, Libya, Syria, and most recently Iraq have all leveraged their control over state-run Internet service providers (ISPs) to shut down Internet connectivity into or out of these countries. These disruptions have become colloquially known as Internet *kill switches*.

Censorship events have sparked debates on political and ethical aspects of kill switches. They also have ignited discussions on a technical level regarding the fragility of current networks. Indeed, a number of different techniques have been used to trigger kill switches, including the withdrawal of Border Gateway Protocol (BGP) routes, Internet Protocol (IP) filtering, domain name system (DNS) hijacking and injection, and even the physical cutting of backbone cables [23].



*EuroSec'17, April 23 2017, Belgrade, Serbia*



DOI: http://dx.doi.org/10.1145/3065913.3065922

Internet kill switches have been thus far geographically-scoped and self-targeted (i.e., nations cutting themselves from the Internet). The possibility of an adversarial use of a killswitch is typically dismissed, both because the safeguarding procedures and technologies are deemed sufficiently secure, and because such an attack would be highly visible. However, a powerful and ruthless adversary may have both the capability and the will to trigger such a killswitch (consider for instance the scenario of cyberwarfare), and more generally, the fact that killswitches are feasible at all is concerning.

This paper investigates the feasibility for an adversary to trigger a kill switch with the objective of *remotely disabling network connectivity for an entire region or country.*

In our analysis, we focus on two emerging secure protocols which can be exploited to build kill switches: DNSSEC [3] and BGPsec [28]. Both schemes were proposed to provide authentication and integrity to core Internet components, but require global trust in a single central authority (Verisign/ICANN for DNSSEC and ARIN for BGPsec). While it is clear that the authorities themselves can impersonate any entity they certify, this paper analyzes the operational steps required for an external adversary to disable connectivity by attacking specific points of these key hierarchies.

We summarize our main findings below.

1. Centralized key infrastructures are ill-suited for Internet-scale infrastructures. These rely on a few roots of trust and give authorities unilateral control over delegated resources, enabling kill switches.

2. Remote Internet kill switches are *feasible*, and *must* be considered especially in the design of secure Internet protocols.

3. Generating, securing, and managing cryptographic keys is known to be challenging [13]. To fill this gap, hosted key management systems have emerged. Hosted PKIs centralize trust in yet another party; this creates high value targets and negates the benefits of independent key management.

## 2. BACKGROUND

### 2.1 BGPsec

Securing inter-domain routing requires that IP address blocks and AS numbers used for routing advertisements are valid and that entities disseminating these advertisements are authorized to do so. The *Resource Public Key Infrastructure (RPKI)* [18] provides a trusted mapping from allocated IP prefixes to ASes authorized to originate them in

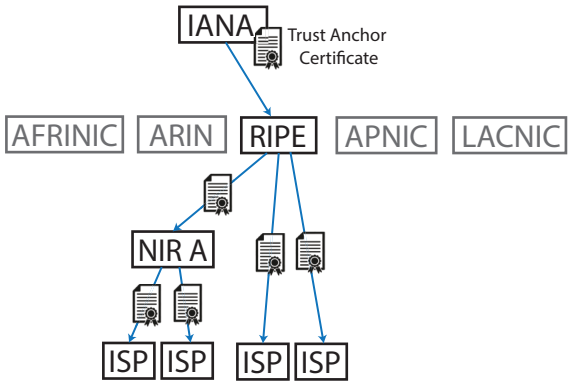


**Figure 1: RPKI resource allocation hierarchy.**

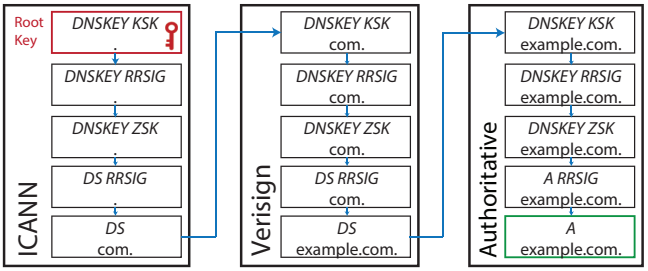


**Figure 2: Chain of trust in DNSSEC for example.com. *DNSKEY* entries contain a public key used to sign an RRset. The signature is then stored in a *RRSIG* record. *DS* records store a hash of a public key of the child zone and is used to link the parent zone to the child zone.**

BGP. RPKI establishes a cryptographic hierarchy of authorities that sub-allocate IP address spaces and authorize their use in BGP (see Fig. 1). Authorities issue signed Route Origin Authorizations (ROAs) allowing an AS to originate one of their prefixes.

The RPKI hierarchy is rooted at the regional Internet registries (RIRs). There is roughly one RIR per continent (RIPE in Europe, ARIN in North America, LACNIC in Latin America, AfriNIC in Africa, and APNIC in Asia Pacific). RPKI does not require any modifications to BGP message formats nor any online cryptography during routing. Each AS regularly syncs its local cache with the public repositories, locally verifies and pushes the resulting whitelist to its border routers.

While RPKI can help authenticate path origins, it cannot fully protect against some classes of attacks such as route leaks or path-shortening attacks [8]. To prevent these attacks, *BGPsec* [28] provides path integrity validation by building on RPKI. BGPsec adds cryptographic signatures to BGP messages and requires each AS to sign its outgoing BGP messages [11]. The signature covers the prefix and AS-level path, the local AS number, the AS number to which the update is being sent and includes all the signed messages received from the previous ASes on the path.

## 2.2 DNSSEC

The *Domain Name System Security Extensions (DNSSEC)* [3] aim to overcome the security limitations of DNS. It provides authentication using digitally signed DNS resource record sets (RRsets). The DNSSEC trust chain is a sequence of records that identify either a public key or a signature of a set of resource records. The root of this chain of trust is the root key, which is managed by the operators of the DNS root. Each record is signed by either a Key Signing Key (KSK) or a Zone Signing Key (ZSK). The former is used to sign DNSKEY records, while the latter is used to sign all other records in the authoritative domain. To validate an RRset, one constructs a chain of trust from the root of trust to the RRset (see Fig. 2). If it succeeds, the information in that RRset is cryptographically authenticated [20]. The root keys themselves are stored on hardware security modules (HSM) in multiple redundant facilities with high physical protection [21]. Smartcards are necessary to activate and decrypt the HSMs, and these cards are stored in physical safe deposit boxes. *Crypto Officers (CO)* are given a physical key to these boxes. In case of an emergency situation where the COs are unable to travel to the facility, the responsible authorities can break into the boxes. If all HSMs become inaccessible, five out of seven *Recovery Key Share Holders (RKSH)* are required to reconstruct the secret backup key used for encryption of the backup HSMs [15].

## 2.3 Threat Model and Scope

The scope of this paper is on powerful adversaries (most likely governments conducting cyber-warfare) aiming to disable, either partially or fully, another party's access to the Internet. This can be done by injecting forged routing information into a BGP-enabled network, injecting forged DNS records or responses, or executing denial of service attacks on critical elements of the infrastructure. As recently reported [12], adversaries may coerce organizations (e.g., hardware or software vendors, network operators) to implant back doors, or independently collect data through mass surveillance techniques [10]. Our main focus is on scenarios where adversaries make use of key hierarchies to execute a kill switch.

For space reasons, we assume pervasive and sole deployment of DNSSEC and BGPsec. We do not consider dual DNS/DNSSEC and BGP/BGPsec environments where the victim could switch off the secure variant and fall back to the insecure protocol. Indeed, an adversary powerful enough to trigger a kill switch in these environments, could also execute prefix hijacking or path shortening attacks in BGP, or cache poisoning attacks in DNS respectively, to implement a kill switch.

As of writing DNSSEC is deployed on ∼87.52% of all top-level domains (TLD) [14], and ∼12.63% of all DNS queries are validated [2]. For RPKI only ∼6.32% of advertised IPv4 prefixes have corresponding ROAs (both valid and invalid) [25].

# 3. TRIGGERING KILL SWITCHES IN BGPSEC

RPKI enables administrative control over delegated IP address ranges, but also brings disadvantages. The reliability and redundancy of the Internet is supported by its highly distributed nature. Using RPKI seriously constrains these properties, forcing reliance on a small number of authorities.

Cooper et al. [8] show the risks of misbehaving authorities making targeted manipulations of the RPKI. In the following, we show how these manipulations can be used to trigger a kill switch in BGPsec.

**Certificate Revocation.** RPKI uses conventional Certificate Revocation Lists (CRLs) to revoke certificates that are no longer valid but have not yet expired. An authority can revoke *any* child certificate it issued previously. A revocation may happen for several reasons including key rollover or termination of the resource allocation. Each CA issues a CRL (only) for certificates which the CA itself has issued, and it updates the CRL at declared, regular intervals. Delegated parties must publish their own version of a CRL. All RIRs update their CRL every 24 hours for online and hosted member CAs, and every 3 months for offline CAs.

A certificate revocation triggered by a parent CA causes other BGPsec-enabled networks to reject routing updates from the targeted AS and invalidates the entire subtree of certificates. This effectively makes the prefix invisible.

An attack that results in the revocation and replacement of a key or certificate causes the affected subject to be unable to sign new objects using his private key. An adversary controlling intermediate routers could block propagation of withdrawal messages and thus make its attack more persistent.

**Certificate Forgery.** An adversary might compromise a private key associated with a router or an AS (e.g., by exploiting a software vulnerability). Some regional registries (e.g., RIPE and ARIN) offer their customers "hosted" versions of RPKI, where the registry handles all key management operations as well as generation of ROAs. Hosted RPKI adds yet another trusted party to BGPsec, and makes registrars who offer the service high value targets. Using the compromised key, the adversary can forge updates that appear to have passed through the compromised AS. The forged updates can be injected into neighboring ASes from any adversary-controlled AS. This would effectively transfer the affected address space to the adversary or allow redirection of traffic. If the forgery is used in a remote AS, the neighbors might not accept an incoming update directly from the victim's AS. In this case the adversary can take a legitimate route traversing through the compromised AS and position itself as the next hop to become on-path [16]. An on-path adversary can trivially drop traffic, black-holing the victim's AS.

**Attacks on RPKI Objects.** Certificates, manifests, CRLs, and ROA repositories are critical elements in RPKI. An attacker compromising those repositories or their publication points could remove signed objects, inject invalid objects, or replace existing objects with one for a smaller set of IP addresses. Each of these attacks results in Denial of Service for each relying parties. Because these parties cache the data they acquire, they are able to partially mitigate such attacks by reverting to the cached data, or possibly by accepting stale data.

Another critical point is a compromise of management functions and tools used for RPKI. The adversary could modify the local routing policy to black-hole certain routes or trigger certificate revocations causing router certificates or ROAs to be revoked. By requesting new ROAs, the adversary could re-allocate prefixes allocated to the target. This results in other networks believing that these prefixes no longer originate from the affected network [16].

**Protocol Manipulation.** Protocol manipulation attacks [27] can be used to undermine reachability and route selection by ASes, and are not prevented by authentication and security mechanisms such as those in BGPsec. By manipulating the route selection by the ASes, traffic flow could be influenced to traverse an AS under the control of the adversary. Using route flapping (e.g., announcing and withdrawing routes at high frequency), an adversary can render an AS unreachable.

Furthermore, BGPsec is unable to achieve desired properties such as blackhole-resistance or loop-free routing due to wormhole and mole attacks [19]. These attacks could be used to blackhole traffic or overload network links.

# 4. TRIGGERING KILL SWITCHES IN DNSSEC

Although DNSSEC was designed to defend against adversaries that tamper with DNS responses to block a host, or redirect users to an adversarial host, DNS takedowns are still possible in DNSSEC. This section discusses the necessary steps to adversarially trigger a kill switch on networks that have deployed DNSSEC. We describe operational considerations necessary for the attack and compare them to attacks in BGPsec.

As in BGPsec, control over private keys is essential to execute manipulations in DNSSEC. Powerful adversaries such as governments may (possibly legally) compel network operators to either disclose their private keys or behave according to their intentions. Depending on the country, compromised keys can enable control of a top level domain (e.g., *.ca* in Canada) or even control of the root zone (i.e., the United States). Moreover, many registrars support hosted DNSSEC for registered domains. These services handle all further signing of the zones as well as key roll-over, but create another attack vector as they might allow control over all domains hosted there. Compared to DNS which consists of thirteen different root servers, DNSSEC is built around a single root of trust, which even further centralizes the system.

**Risks during Key Roll-over.** The procedure to roll over a key in DNSSEC depends on the type of key. To establish a new KSK pair, trust in the current key must be ensured; either there exists a signed and validated DS record in the parent zone or a trust anchor has been explicitly configured. In the former case, the parent zone must add a new DS record and remove the obsolete one. In the latter case, there is no superior key to anchor the rolling key. To allow resolvers to build trust in the incoming key, a "hold-down" phase is used in which the incoming key is published but not yet active. After this phase the key is effectively rolled over to the new one [29].

For a ZSK roll-over, DNSSEC-validating resolvers need access to a signature corresponding to a valid ZSK. The key roll conventionally involves introducing a new ZSK signed by the zone's KSK. The new record will eventually propagate as the old cached entries expire. At this point the new ZSK is ready for use and all entries in the zone can be re-signed. The old key is kept to allow old cached entries to be validated and will finally be removed after another TTL period [24].

Roll-over of the Root Zone KSK is particularly precarious. Every validating resolver maintains a local copy of

this key, which is updated through the operating system's software update mechanism or by the resolver software itself by retrieving well-known URLs (e.g., https://data.iana.org/root-anchors/root-anchors.xml). If some resolvers are stranded with the old KSK, they will no longer operate as intended until being reloaded with the new KSK value. Currently, the Root Zone KSK roll-over is in progress. The new key has been generated but not yet published. The complete KSK rollover process is expected to take about two years. Network operators who have enabled DNSSEC validation on their DNS resolvers will need to update their systems with the new root KSK once it has been published. Neglecting to install the new trust anchor after the actual roll-over event, when the new key is used for signing in the root zone, will cause all DNS lookups performed by non-updated resolvers to fail [7].

**Response Forgery.** In order to refresh DNS entries with low time-to-live (TTL) and to support dynamic DNS changes, the ZSK is usually kept online. If an adversary obtains the ZSK of a particular zone, he would be able to modify the Delegation Signer (DS) entries linking to the child zone and validly sign them. DS entries can be crafted such that all clients trying to verify the chain of trust will fail, believing that the KSK of the child zone has been changed. As a consequence, for strictly validating clients, an entire sub-zone might become invisible [5].

In the case that a KSK is compromised, the attacker is able to add its own ZSK and sign the zone's keyset using this new ZSK. From that point on the adversary proceeds as in the previous case. Both compromises are externally visible but are indistinguishable from a regular key roll-over (described below). Because KSKs are only used for rolling ZSKs, they should be kept offline. With the exception of the trust anchor in the root zone, establishing a new KSK also requires the interaction of the parent zone as parents need to update their DS records as well. Involving another party in the attack increases complexity and attack visibility.

**Inducing Verification Failures.** Incorrect signatures make DNSSEC resolvers fail during the verification process. Thus, an attacker on the path upstream from a validating resolver can cause signature validations to fail by modifying traversing DNS packets. Because DNSSEC does not provide confidentiality of queries and responses, the attacker can selectively cause responses to fail (e.g., for a specific domain). Of course, if a client can use a different path to a resolver (e.g., through a VPN), the attack will no longer succeed.

**Root key injection.** Resolvers themselves (including running on end-user machines) can be targeted through a root public key injection attack, which would compromise the chain of trust [1]. The attack surface for key injections may grow depending on the root key update mechanism used by the resolver and/or host operating system.

**Misconfiguration.** Even tough DNSSEC provides unforgeable authentication of RRsets, it does not protect against misconfiguration or bogus information on the authoritative name server. As a consequence, if a client repeatedly receives bad information for a particular zone and fails to verify the response, the zone will become invisible.

**Denial of Service and Other Attacks.** As with any network service, DNSSEC servers are vulnerable to Denial of Service (DoS) attacks. While servers in the root zone are highly replicated (anycast) and globally distributed, smaller TLDs or name servers for specific domains may be less resilient to attacks. An attacker can mount a large DoS attack on specific resolvers and take them offline.

DNS responses for a DNSSEC-signed domain are typically larger than those of an unsigned domain, which allows attackers to amplify their attack volume. A recent study [30] shows that the average amplification of DNSSEC exceeds that of regular DNS by a factor of 6-12. In extreme cases amplification in the hundreds is possible. Thus, components of the DNSSEC infrastructure are a valuable target.

## 5. RECOVERING FROM KILL SWITCHES

In both DNSSEC and BGPsec, the general recovery process when a key is compromised involves creating a new key pair and notifying the parent of the key update. At low levels of the trust chain (e.g., for cs.university.edu or for a small ISP), this process can be as simple as submitting the hash of the new key through a web form and waiting for DNS caches to expire or routing tables to converge. However at higher levels, and especially at the DNSSEC/BGPsec root, generating a new key pair and disseminating its existence downstream can require coordination among numerous parties and require an extended time period. Furthermore, the attacker may have taken down networks required for recovery such as payment processors or communication infrastructure (e-mail, support forums, other ISP communication channels).

Concretely, when a non-root DNSSEC KSK is compromised, the victim must generate a fresh key and transmit it to the parent zone for publication as a new DS entry. In the unlikely event of a Root Zone KSK compromise, as per ICANN's emergency KSK roll-over procedures [21] an interim trust anchor will be generated and published within 48 hours. We note that the success of the emergency key roll-over procedure depends on deployment of automatic key roll-over support, as specified in RFC 5011 [29]. Lack of support requires human involvement to update the Trust Anchor, causing further delays while administrators are unavailable. Due to the temporary nature of the interim Trust Anchor, a scheduled key ceremony needs to take place to establish another root KSK according to the root zone DNSSEC practice statement [21].

If a ZSK has been compromised (but the KSK is safe), recovery requires signing and publishing a new ZSK key using the unaffected KSK. As KSKs are kept mostly offline, this delays the recovery. In case of the Root ZSK, VeriSign has procedures for unscheduled roll-over in place, but does not state a specific recovery time [22].

An important factor that limits recovery for DNSSEC is caching: we find that root ZSK records have a TTL of 48 hours, and DS records from the root to the TLDs all have a 24 hour TTL. We also analyzed popular second level domains (SLD)[1]. While fewer than 2% of domains had DS records, half of those had a TTL of 24 hours while the rest had TTLs of around 1 hour.

Furthermore, the private component of a key pair might be permanently lost. If the Trust Anchor is permanently lost, this loss will be detected no later than the key ceremony. A new key is established either at the same ceremony or in another ceremony within 48 hours [21].

Recovery after a key compromise in BGPsec is accom-

[1]Alexa top 10 000, www.alexa.com/topsites.

| | Attacker capabilities | Impact | Visibility | Recoverability |
|---|---|---|---|---|
| DNSSEC | Root KSK/ZSK compromise | ● | ○ | >48h |
| | TLD KSK/ZSK compromise | ◑ | ◑ | >24h |
| | SLD KSK/ZSK compromise | ○ | ● | (<24h) |
| | Inject root key into resolvers | ○ | ● | ~ |
| | DoS against resolvers/NSes | ○ | ◑ | ~ |
| BGPsec | Root key compromise | ● | ○ | <48h |
| | RIR key compromise | ◑ | ◑ | <48h |
| | RPKI hosting compromise | ○ | ◑ | <48h |
| | Wormhole/mole attack [19] | ◑ | ○ | ~ |

**Table 1: Properties of different types of kill switches. The qualitative values range from best for security ○ (low impact/high visibility) to worst ● (high impact/low visibility). A "~" for recoverability indicates that the value cannot be estimated in an absolute way.**

plished by establishing a new key and revoking forged certificates through a CRL. After a new key-pair has been created, the affected AS makes the new certificate available to the RPKI global repository and is propagated to RPKI caches within one cache update cycle. The RPKI cache refresh frequency may be chosen by the operator, but typically lies between 1-24 hours [6]. In anticipation of possible key compromise, an operator could pre-provision each router's *next* key in the RPKI to eliminate the propagation delay. From this point in time, BGP updates are signed using the new key and it takes one update cycle to fully propagate. The affected AS also publishes a CRL including the serial number of the old certificate. After the CRL entry has been published (in general every 24 hours [26] or 1 business day [4]), the updated entry must be disseminated. In the worst case, victims would not fetch the updated CRL in a timely fashion, allowing an attack to persist for a full day.

For kill switches without key compromise, recovery is simpler. These events typically do not involve client action, but rather waiting for the attack to conclude or bypassing the affected network path.

# 6. TOWARDS KILL SWITCH EVALUATION

Assessing the effectiveness of a kill switch is challenging since many of the aspects that need to be considered are difficult to quantify. An ideal metric would quantitatively classify an event as a kill switch by comparing it to a disruption threshold.

In this section we propose a set of properties intended to enable reasoning about, evaluating, and comparing kill switches.[2] Table 1 provides a preliminary qualitative comparison of the kill switches discussed in this paper based on our proposed properties. Quantitative values represent a rough estimate based on our findings in Section 5.

[2]Despite its importance, we do not consider the financial cost (to an adversary) of a kill switch because obtaining a precise cost measurement is associated with uncontrollable and unpredictable events, or requires a precise prediction of future events such as software vulnerabilities or human errors.

**Impact.** Intuitively, the *impact* of a kill switch is the amount of damage the kill switch causes. This would include all financial losses, which are difficult to estimate. Instead, we suggest basing impact metrics on network disruption, i.e., the percentage of Internet communications that are blocked by the kill switch. Analogously, network disruption in subnetworks can be compared to a "min-cut" in graph theory. This allows identification of the minimal number of communications that an adversary needs to disrupt such that the largest number of end hosts are affected.

**Visibility.** The *visibility* of a kill switch is directly related to its impact. Visibility should be measured in relation to detection mechanisms designed to raise an alert in case of the activation of a kill switch. Dainotti et al. [9] suggest using Internet Background Radiation-based detection for BGP-based anomalies, while for DNSSEC one could construct a distributed monitoring network that continuously checks DNS infrastructure. Because of the distributed nature of DNS, public views on the system are not consistent, which makes it hard to construct an infrastructure that can detect more targeted attacks [17]. To attain more advanced detection capabilities, a system similar to Certificate Transparency would be required [32].

**Recoverability.** This metric considers how easily or quickly the network can recover. As with impact, we propose the use of a simplified metric that does not consider financial costs of recovery, but solely considers the time to recovery. Depending on the analysis, it could be useful to consider the range between the best-case and worst-case recovery time. In cases where partial recovery can be performed quickly, but full recovery takes longer (e.g., client software updates are needed), a more useful metric would be the time to 90% recovery.

**Precision and Effectiveness.** While the previous properties concern a specific instance of the activation of a kill switch, properties relative to the goals of the adversary can be considered as well. These properties specify the degree of control that the adversary has over the kill switch prior to its execution. The first such property is the *precision* of a kill switch with respect to its target area. Intuitively, high precision implies low collateral damage (i.e., low damage outside the target area). Precision could be quantified as the ratio between the impact within the target area and the overall impact. The second property is *effectiveness*. In context of kill switches this can be seen as the fraction of the target area that is actually affected by a kill switch, i.e. the fraction of entities that remain online after a kill switch event has taken place. Effectiveness could be quantified as ratio between the actual impact in the target area, and the maximum possible impact in the target area.

## 6.1 Discussion

There appears to be an inherent relationship between attack feasibility, visibility, and recoverability. For example, the execution of a kill switch high up at the DNS root would require successfully bypassing several layers of physical and digital access control, or expending large amounts of resources on denial of service of root servers. However, if the root KSK were replaced (e.g., to execute a global DNSSEC kill switch), it is reasonable to expect many resolver operators would notice malfunction (e.g., through support calls or log messages). Recovery from such an attack could be time

consuming, in particular if it requires coordination amongst several remote parties.

## 7. CONCLUDING REMARKS

This paper has highlighted the perils of requiring global trust in a single central authority for fundamental components of online communications. Not only are Internet kill switches feasible, they have potentially devastating impact due to the time-consuming processes needed for recovery. Moreover, attacks simultaneously targeting BGPsec and DNSSEC may create circular dependencies, further delaying restoration of service. Even when a kill switch can be triggered only once, the risk of such an attack is high.

To alleviate kill switches and reduce the impact of powerful adversaries, newly designed security infrastructures must be *decentralized* and must resist using existing centralized ones for new protocols. Although distributed security solutions are more challenging, they are technically feasible.

Since neither DNSSEC nor BGPsec is, as of writing, fully deployed, now is a good time to evaluate kill switches enabled by these and other proposals, ultimately arriving at a technically informed solution.

## 8. ACKNOWLEDGMENTS

We would like to thank Elizabeth Stobert and the anonymous reviewers for their insightful feedback and suggestions. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement 617605. This work was also supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.R0190-16-2011, Development of Vulnerability Discovery Technologies for IoT Software Security). We also gratefully acknowledge support from ETH Zurich and from the Zurich Information Security and Privacy Center (ZISC).

## 9. REFERENCES

[1] A. Alsaid and C. J. Mitchell. *Revised Selected Papers of EuroPKI 2005*, pages 227–239. Springer, 2005.

[2] APNIC. Use of dnssec validation for world. http://stats.labs.apnic.net/dnssec/XA?c=XA&x=1&g=1&r=1&w=7&g=0.

[3] R. Arends et al. DNS Security Introduction and Requirements. RFC 4033 (Proposed Standard), 2005.

[4] ARIN. ARIN CPS for resource certification. https://www.arin.net/resources/rpki/cps.pdf, 2012.

[5] S. Ariyapperuma and C. J. Mitchell. Security vulnerabilities in DNS and DNSSEC. In *ARES*, 2007.

[6] R. Bush. Origin Validation Operation Based on the Resource Public Key Infrastructure (RPKI). RFC 7115 (Best Current Practice), 2014.

[7] D. Conrad. Ksk rollover operations begin. https://www.icann.org/news/blog/ksk-rollover-operations-begin, 2016.

[8] D. Cooper, E. Heilman, K. Brogle, L. Reyzin, and S. Goldberg. On the risk of misbehaving RPKI authorities. In *ACM HotNets*, 2013.

[9] A. Dainotti et al. Analysis of country-wide internet outages caused by censorship. In *ACM SIGCOMM*, 2011.

[10] S. Farrell and H. Tschofenig. Pervasive Monitoring Is an Attack. RFC 7258 (Best Current Practice), 2014.

[11] S. Goldberg. Why is it taking so long to secure internet routing? *Communications of the ACM*, 57(10):56–63, 2014.

[12] G. Greenwald. *No place to hide: Edward Snowden, the NSA, and the US surveillance state*. Macmillan, 2014.

[13] P. Gutmann. PKI: It's Not Dead, Just Resting. *IEEE Computer*, 35(8):41–49, 2002.

[14] ICANN. TLD DNSSEC report. http://stats.research.icann.org/dns/tld_report/.

[15] ICANN. Trusted community representatives - proposed approach to root key management. http://www.root-dnssec.org/wp-content/uploads/2010/04/ICANN-TCR-Proposal-20100408.pdf, 2010.

[16] S. Kent and A. Chi. Threat Model for BGP Path Security. RFC 7132 (Informational), 2014.

[17] B. Laurie. Certificate Transparency. *ACM Queue*, 12(8), 2014.

[18] M. Lepinski and S. Kent. An Infrastructure to Support Secure Internet Routing. RFC 6480 (Informational), 2012.

[19] Q. Li, Y.-C. Hu, and X. Zhang. Even rockets cannot make pigs fly sustainably: Can BGP be secured with BGPsec? In *NDSS Workshop on Security of Emerging Networking Technologies (SENT)*, 2014.

[20] W. Lian, E. Rescorla, H. Shacham, and S. Savage. Measuring the Practical Impact of DNSSEC Deployment. In *USENIX Security*, 2013.

[21] F. Ljunggren, T. Okubo, R. Lamb, and J. Schlyter. DNSSEC practice statement for the root zone KSK operator. https://www.iana.org/dnssec/icann-dps.txt, 2010.

[22] F. Ljunggren, T. Okubo, R. Lamb, and J. Schlyter. DNSSEC practice statement for the root zone ZSK operator. http://www.root-dnssec.org/wp-content/uploads/2010/06/vrsn-dps-00.txt, 2010.

[23] A. Mamiit. FBI searches for suspects in new fiber optics cable attack in California. https://perma.cc/S7V7-QZGG, 2015.

[24] S. Morris, J. Ihren, J. Dickinson, and W. Mekking. DNSSEC Key Rollover Timing Considerations. RFC 7583 (Informational), 2015.

[25] NIST. Global prefix/origin validation using RPKI. http://rpki-monitor.antd.nist.gov/.

[26] RIPE. RIPE NCC RPKI CPS. https://www.ripe.net/publications/docs/ripe-549, 2012.

[27] Y. Song, A. Venkataramani, and L. Gao. Identifying and addressing protocol manipulation attacks in" secure" bgp. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2013.

[28] K. Sriram and M. Lepinski. BGPsec Protocol Specification. https://tools.ietf.org/html/draft-ietf-sidr-bgpsec-protocol-15, 2016.

[29] M. StJohns. Automated Updates of DNS Security (DNSSEC) Trust Anchors. RFC 5011 (Internet Standard), 2007.

[30] R. van Rigswijk-Deij, A. Sperotto, and A. Pras. DNSSEC and its potential for DDoS attacks: a comprehensive measurement study. In *ACM Internet Measurement Conference (IMC)*, 2014.

[31] X. Xu, Z. M. Mao, and J. A. Halderman. Internet censorship in China: Where does the filtering occur? In *PAM*, 2011.

[32] D. Zhang, D. K. Gillmor, D. He, B. Sarikaya, and N. Kong. Certificate transparency for domain name system security extensions. https://tools.ietf.org/html/draft-zhang-trans-ct-dnssec-03, 2016.