This is the author manuscript accepted for the 4th IEEE European Symposium on Security and Privacy (EuroS&P 2019).
https://www.ieee-security.org/TC/EuroSP2019

Domain Impersonation is Feasible: A Study of CA Domain Validation Vulnerabilities

Lorenz Schwittmann, Matthäus Wander, Torben Weis
University of Duisburg-Essen
{lorenz.schwittmann,matthaeus.wander,torben.weis}@uni-due.de

***Abstract*—Web security relies on the assumption that certificate authorities (CAs) issue certificates to rightful domain owners only. However, we show that CAs expose vulnerabilities which allow an attacker to obtain certificates from major CAs for domains he does not own. We present a measurement method that allows us to check CAs for a list of technical weaknesses during their domain validation procedures. Our results show that all tested CAs are vulnerable in one or even multiple ways, because they rely on a combination of insecure protocols like DNS and HTTP and do not implement existing secure alternatives like DNSSEC and TLS. We have validated our methodology experimentally and disclosed these vulnerabilities to CAs. Based upon our findings we provide recommendations to domain owners and CAs to close this fundamental weakness in web security.**

***Index Terms*—Certificate Authority, Public Key Infrastructure, Initial Validation, Domain Validation, DNSSEC, DANE**

I. Introduction

The security of the WWW relies on cryptography and certificates, which are issued by certificate authorities (CAs). Security-aware users can take to their browsers to learn about the cryptography and key length involved in securing an HTTPS connection. However, the entire cryptography is pointless if the browser trusts in the wrong certificates. Even for a security-aware user it is difficult to judge whether a given certificate should be trusted or not. Therefore, browser vendors like Mozilla or Apple compile lists of CAs, which are considered to be trusted.

Trust in a CA is based upon their commitment to issue certificates to rightful domain owners only. There have been cases like Symantec [10] where a CA has been shown to issue certificates to unauthorized entities. CAs offer different domain validation (DV) procedures, which in turn rely on the security of other protocols and infrastructure like DNSSEC, DANE or HTTP. This poses the following research question: how secure is domain validation and what countermeasures are in use to fend off attackers? Another aspect is the appearance of Let's Encrypt as a new CA, which disrupted the market in 2015 by issuing certificates for free with a fully automated procedure. This raises the additional question whether Let's Encrypt is able to achieve a security level comparable to traditional CAs.

In this paper, we develop a measurement methodology that tests CAs for vulnerabilities in their different DV procedures. We analyze the DV issuance process, identify potential security mitigations and survey their existence while requesting a certificate from 15 CAs, which cover 96% of the certificate market. Our method searches for indications of security measures; an absence reveals conclusively a vulnerability under our threat model.

Our research shows that all major CAs expose weaknesses when validating whether a signing request was issued by the rightful domain owner or not. This is despite the fact that secure countermeasures already exist. CAs either do not employ all available security measures or fail to implement them properly. We confirm the validity of the survey methodology by demonstrating successful man-in-the-middle attacks on three CAs for a test domain under our control.

An attacker can exploit the detected vulnerabilities to falsely convince the CA of owning the domain, resulting in a certificate trusted by all major browsers. Such a certificate can then be used in a man-in-the-middle attack to compromise the authenticity or encryption between browsers and legitimate web servers. For a network-level attacker, attacking the certificate issuance is much easier than breaking HTTPS encryption. This type of attack requires neither to break the cryptography nor a lot of computing power. Thus the security of the web depends substantially on the domain validation practice.

The contributions of this paper are: (1) a security analysis of domain validation, (2) based upon which we develop an approach for the detection and classification of domain validation vulnerabilities, (3) which we applied on major certificate authorities to get insights about their DV practices for the first time.

This paper is organized as follows. Section II describes the certificate issuance process and our threat model. We elaborate on the domain validation methods, their security issues and potential countermeasures. In Section III we present our measurement methodology that detects these countermeasures and classifies vulnerability against attacker types. Section IV lists the certificate authorities that we tested in practice with the results given in Section V. We demonstrate practical attacks in Section VI. Section VII describes the disclosure of results to the CAs and their responses. Section VIII discusses related work. Section IX gives operational recommendations based on our findings.

II. Certificate Issuance

The process of certificate issuance begins with an *applicant* generating an asymmetric key pair. While the private key remains with him, the public key is bundled with the fully qualified domain name (FQDN) in a certificate signing request

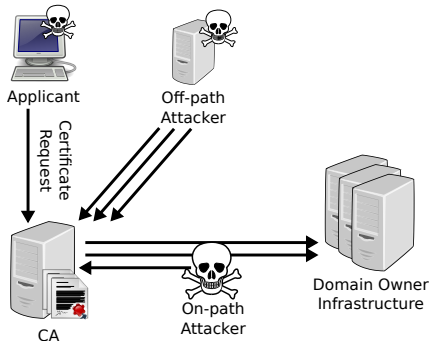


Fig. 1: Threat model: an attacker requests a certificate for a domain owned by someone else.

(CSR) and is sent to a CA. The CA checks whether this request is approved by the *domain owner*, who might be a different entity than the applicant. Domain Validation (DV) is the process of confirming whether the applicant has control over the domain name. Depending on the validation method this involves passing a *challenge* defined by the CA. Once the validation has been completed successfully, the applicant receives a signed certificate.

The actual implementation of these steps depends on the CA and is usually not disclosed to the public. An exception is Let's Encrypt. To fully automate the whole certificate issuance process, the Automatic Certificate Management Environment (ACME) protocol has been developed [3].

Besides DV there are the procedures of Organization Validation (OV) and Extended Validation (EV), which additionally verify and include the name of the domain owner's organization. EV certificates cause web browsers to prominently show that name in the address bar instead of the regular HTTPS indicator, e.g. a green padlock.

In this paper we focus on DV only, as we assume the average user will not be able to tell the difference to an OV or EV certificate. This assumption is backed by a study by Jackson et al. who *"did not find that extended validation provided a significant advantage in identifying the phishing attacks tested in this study"* [28].

A. Threat Model

An attacker attempts to obtain a certificate for a domain name not possessed by him. The attacker acts as the applicant in the certificate issuance (Figure 1), whereas the legitimate domain owner does not intend to interact with the CA. The attacker interferes with the subsequent domain validation to trick the CA into believing that the domain owner approves the certificate issuance.

We consider two types of network-level attacks: *off-path* and *on-path attacker*. Off-path attackers have the capability to spoof IP packets with a source address claiming to originate from the domain owner, but do not see the network traffic between the CA and the domain owner's servers. On-path attackers are capable of passive eavesdropping or performing an active man-in-the-middle attack. The validity of this threat model has been demonstrated by prior work and can be achieved, e.g., by redirecting network traffic via BGP attacks [7]. In any case, the attacker has the capability to choose the CA involved in the domain validation (*CA selection attack*), as he initiates the certificate issuance process.

B. Validation Methods

The *"Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates"* [11] specify the methods allowed for domain validation. They are published by the the CA/Browser Forum which is a consortium of CAs and browser vendors negotiating the set of minimal practices that a publicly trusted CA must employ. For example, the Mozilla Root Store Policy as of version 2.5 [34] explicitly requires CAs to employ a subset of the validation methods defined in the baseline requirements version 1.4.1.

The validation methods include out-of-band validation such as fax, SMS, phone, or postal mail, as well as contacting the domain name registrar. While these methods are valid ones, they are rarely used for domain validation in practice. As we will see in Section IV, none of the considered CAs offered them for domain validation. The following Internet-based validation methods are used in practice:

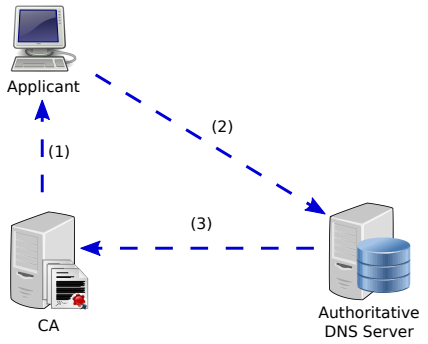
1) **DNS Change.** The CA generates a random token and instructs the applicant to publish it in the DNS zone file as a `TXT`, `CNAME` or `CAA` resource record.
2) **Agreed-Upon Change to Website.** The CA generates a random token and instructs the applicant to publish it under a specific URL within the domain.
3) **TLS Using a Random Number.** The CA generates a random token and instructs the applicant to generate a TLS certificate containing that value and serve it on that domain.
4) **Email to Domain Contact.** The CA sends a random token via email to the email address stored in the domain's WHOIS record. The applicant has to submit this token, usually via a website.
5) **Constructed Email to Domain Contact.** Like (4) but the email address is constructed by using 'admin', 'administrator', 'webmaster', 'hostmaster' or 'postmaster' `@ domain`.

All methods include transfer of a random token whose actual implementation depends on the CA. The baseline requirements define the token as either a randomly generated value of at least 112 bit entropy or as a request token cryptographically derived from the CSR.

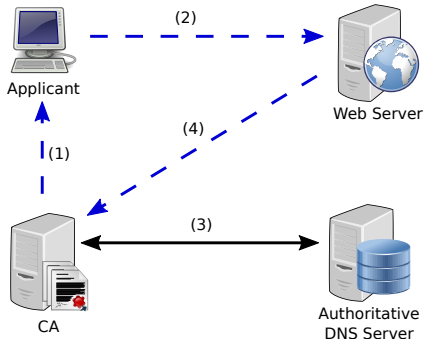
To assess the attack resilience, we have a detailed look at each validation method and discuss potential weaknesses as well as mitigation strategies. An overview of the following discussion is shown in Table I.

C. DNS-based Validation

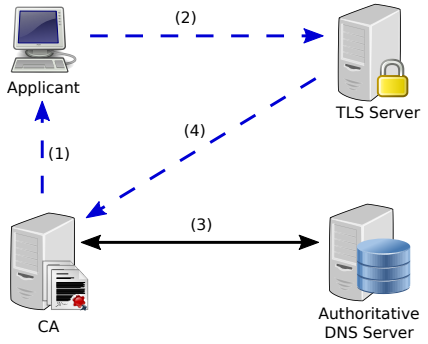
If a CA offers this validation method, it typically prompts the applicant to add a specific `CNAME` or `TXT` record containing the token to the domain's zone file (cf. Figure 2a). After



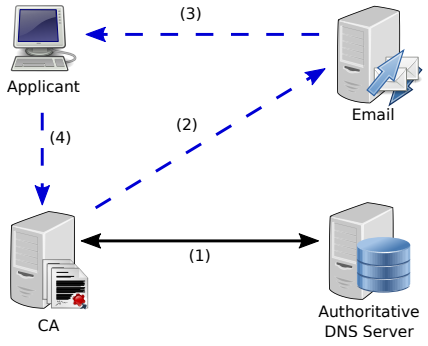
(a) DNS



(b) HTTP



(c) TLS-RND



(d) Email

Fig. 2: Flow of random tokens (dashed lines) and supporting DNS lookups (solid lines) for different validation methods.

TABLE I: Validation methods, attacks and associated countermeasures.

Validation Methods	Attack	Countermeasure
All	Off-Path DNS	Detect mass-spoofing Source port randomization DNS Cookies 0x20 encoding DNS via TCP DNSSEC DNS Multipath DNS Multiserver
All	On-Path DNS	DNSSEC DNS Multipath DNS Multiserver
HTTP	HTTP Active	HTTP Multipath DANE
TLS-RND	TLS Active	TLS Multipath DANE
Email	SMTP Eavesdrop	STARTTLS End-to-end encryption
	SMTP Active	DANE End-to-end encryption MTA-STS

the applicant has made this change to the DNS zone, the CA queries the record using a DNS resolver to verify whether the applicant has control over the domain.

From an attacker's point of view this method is prone to DNS spoofing. Depending on the attacker's capabilities the response has to be spoofed with (on-path attacker) or without (off-path attacker) knowledge of the actual request. DNSSEC protects from both types of attackers, provided that the domain is signed.

If the domain is not signed with DNSSEC, there are several best practices and protocol extensions available to mitigate off-path spoofing by increasing the entropy in DNS requests: DNS Cookies [19], 0x20 encoding [14], source port number randomization [27] or TCP-based transport. Multiple, redundant queries with a consistency check can also be used to decrease likelihood of a successful attack. Off-path spoofing requires a massive amount of forged responses to match the guessed entropy in the request. Another mitigation strategy is to detect spoofing attempts by monitoring the number of incoming responses, thereupon triggering additional countermeasures or human intervention.

Apart from DNSSEC validation there are few effective mitigation strategies against on-path attackers. The ACME specification [3] suggests to send DNS queries from multiple vantage points (*multipath* queries). Similarly a CA could send redundant queries to all authoritative DNS servers for that domain (*multiserver* queries). The idea behind both strategies is that an on-path attacker has the capability to poison a few Internet paths between the CA and domain owner, but probably not every possible one. Sending redundant queries over diverse paths increases the likelihood to receive an untainted response,

which exposes a potential spoofing attempt.

D. Web-based Validation

We sum up the HTTP-based (Agreed-Upon Change to Website) and TLS-RND-based (TLS Using a Random Number) methods as web-based validation because they have similar security properties.

1) HTTP: The CA asks the participant to place a token under a well-known URL with the applied-for domain name (cf. Figure 2b). Once the token has been placed, the CA verifies if the token is indeed in place. Before the HTTP request occurs, the CA has to resolve the domain name to obtain the web server's IP address by querying for `A` or `AAAA` DNS records. This is susceptible to the DNS attacks explained in the previous Section II-C. If the attacker successfully spoofs a forged DNS response, he can redirect the CA to a web server under his control to pass the challenge. Thus, the DNS-specific considerations and countermeasures apply for the web-based validation as well.

The HTTP transaction provides an additional potential target for the attacker, because spoofing either the DNS or the HTTP response suffices to succeed. As the DNS and web server are typically located on different hosts, potentially in different networks, this constitutes another path for on-path spoofing. Again, the CA could employ redundant requests (*HTTP multipath*) to mitigate this vulnerability. A cryptographic proof of authenticity would be required to securely prevent man-in-the-middle attacks on HTTP. In the web context authenticity is usually provided by a trusted certificate—which a CA cannot expect to be available if the applicant is currently applying for one.

Besides this bootstrapping problem, even if the CA attempts an HTTPS connection to the target web server, an on-path attacker can deny HTTPS availability and force a downgrade to cleartext HTTP. To avoid a downgrade attack, the domain owner needs a secure channel to advertise the HTTPS capability along with his identity to the CA. DNS-based Authentication of Named Entities (DANE, [23]) provides such a measure by binding the domain name to a certificate or public key to be used in TLS connections. The certificate used in the HTTPS connection may be a self-signed certificate, if announced as such via DANE. Combined with DNSSEC this approach prevents man-in-the-middle attacks and downgrade attacks.

2) TLS-RND: Presenting a self signed certificate with a CA-defined random number is comparable to HTTP(S)-based validation (cf. Figure 2c). The main difference here is that after DNS resolution and performing a TLS handshake no further application protocol is used. During the specification of ACME, two protocol variants for this approach called *TLS-SNI* and *TLS-ALPN* have been defined.

TLS-SNI is defined in the ACME draft up until version 9 [2]. It uses the Subject Alternative Name (SAN) field to encode the random token as a subdomain of the non-existing `acme.invalid.` zone (SAN A). Likewise, a CSR-specific key is encoded as a second alternative name (SAN B). After deployment of the certificate by the applicant, the CA initiates a TLS handshake with the Server Name Indication (SNI, [18]) set to SAN A and considers the challenge to be passed if a certificate with both SAN A and SAN B is presented.

This approach turned out to be exploitable on hosting providers which allow user-provided certificates to enable HTTPS on their websites[1]. As the Subject Alternative Names have no direct connection to the actual domain names, the providers could not enforce a strict domain separation. Users at the same provider could therefore pass TLS-SNI challenges for any other hosted domain. As a consequence TLS-SNI was deprecated and disabled in Let's Encrypt. TLS-ALPN [37] does not have this issue as the domain name to be validated is kept unchanged as Subject Alternative Name. Instead, the random token is encoded in a newly specified certificate extension. Additionally the Application Level Protocol Negotiation (ALPN, [20]) extension is used during TLS handshake with a fixed identification sequence—although no actual application data is sent over that TLS channel. This should avoid confusion about the semantics of these certificates and effectively make this validation method an opt-in option for hosting providers.

Despite the protocol differences between TLS-SNI and TLS-ALPN our security considerations apply to both of them. As long as the certificate is not trusted—either by a valid CA or by DANE—the CA has no means to assert the authenticity of the domain owner.

E. Email-based Validation

The CA sends an email to the domain contact according to the WHOIS database or to an address constructed from the applied-for domain name (cf. Figure 2d). Some CAs allow the applicant to choose the email address, but only from a small set of addresses and never freely. The email contains a token, which the applicant has to submit on the CA website to prove control over the mailbox.

To send an email—both constructed and WHOIS email—the CA has to query `MX` and `A/AAAA` DNS records to locate the mail server. This is susceptible to the DNS attacks mentioned in Section II-C and the countermeasures discussed there apply as well. If the attacker is able to redirect the SMTP connection to his server, he can easily intercept the token.

While all validation methods demonstrate control over the domain, email-based validation differs on a conceptual level as the applicant proves ability to read rather than write on the domain. The token thereby becomes a secret, as anyone who can obtain it is allowed to obtain certificates for this domain. This is not the case for the other validation methods where the challenge explicitly consists of publishing the token. Email-based validation is thus the only method where a purely passive attacker can succeed by eavesdropping the SMTP connection. Thus, the email must be encrypted to prevent eavesdropping, and the applicant must not be allowed to choose the key as he is the attacker in our threat model.

[1] https://www.zdnet.com/article/lets-encrypt-disables-tls-sni-01-validation/, Accessed 2018-06-27

One way to achieve this is to upgrade the cleartext SMTP connection to TLS with the STARTTLS extension. Similar to HTTP, a man-in-the-middle attacker can perform a downgrade attack by stripping the STARTTLS signals. Again, using DANE/DNSSEC with a self-signed certificate provides authenticity and prevents downgrade attacks on SMTP connections [16]. Mail Transfer Agent Strict Transport Security (MTA-STS) is another work-in-progress mechanism for DNS-based signaling of STARTTLS support for email domains [32]). Other than DANE, MTA-STS requires a CA-issued certificate for the mail server, which implies the applied-for domain either cannot use it or must rely on a third-party email provider, which already has a certificate. MTA-STS does not require DNSSEC, which makes it easier to deploy but in this case also vulnerable to DNS-induced downgrade attacks.

Another way to hide the secret token is by using end-to-end encryption with S/MIME or OpenPGP. In this case the CA needs a secure way to look up the public key of the domain owner, who is not necessarily identical with the applicant. There are experimental DNS-based approaches to achieve this objective for both, S/MIME [24] and OpenPGP [41].

F. Additional Countermeasures

In addition to the security measures discussed above, there are additional countermeasures that are independent of the chosen domain validation method. While these are not direct countermeasures to the attack sketched in section II-A, they are suited to mitigate it.

Certification Authority Authorization (CAA) [22] is a DNS record that lists the CAs that are permitted to issue certificates for a domain. The domain owner may optionally put such a record into his DNS zone file, thus prohibiting unauthorized CAs from certificate misissuance. Each CA is required to check the existence of the `CAA` record during domain validation according to the baseline requirements. This prevents the CA selection attack, where an adversary choses the CA with the least security measures or iterates through all CAs until the attack succeeds with one of them. As this method relies on the DNS, the attacks and countermeasures discussed for DNS-based domain validation do apply here as well.

Certificate Transparency (CT) [31] is an approach for publishing all certificates issued by trusted CAs in logs with cryptographic proofs of inclusion. Using Merkle Hash Tree the existence of a once-submitted certificate cannot be denied. The goal of this approach is to force CAs to publish every issued certificate as clients will refuse certificates that do not bear a log inclusion proof. When all valid certificates are visible to the public, misbehaving CAs and attacks can be discovered more easily.

Extended Validation (EV) includes a process to verify the domain owner's identity in addition to his control over the domain. Unlike plain DV, this involves human interaction which has the potential to discover an ongoing attack. Still, effectiveness depends on CA-specific realization of this process.

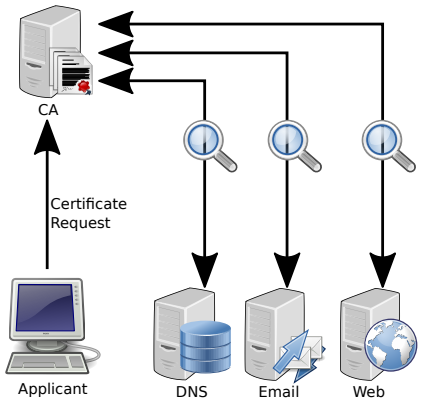


Fig. 3: Measurement setup.

III. Approach

We acquire certificates for a domain under our control from various CAs in the wild. The approach is designed to complete the domain validation successfully while allowing us to observe the security measures that a CA employs. In particular, we do not send invalid responses and do not provoke domain validation errors. All network traffic is recorded to perform a post-mortem analysis after the certificate has been issued.

Design rationale. The rationale for not provoking validation errors is to get a complete view of the domain validation process without endangering the reproducibility or validity of the results. Intentional misconfigurations or simulated attacks may raise suspicion and trigger detection systems, which would influence the outcome of further tests under the same domain and applicant identity. This approach justifies to experiment with productive systems without prior consent from the CAs, which again might endanger the validity of the results. From the CA's point of view our experiments are regular certificate issuances.

Limitations. A drawback of using test domains is that we will not encounter potential additional verification measures of high-risk certificate requests. Another limitation is our optimistic assumption that if there are indications for a security measure, then it will be implemented correctly. Thus we might miss vulnerabilities due to implementations faults. However, the absence of a security measure is undoubtful a vulnerability according to our threat model.

A. Setup

Our test setup is shown in Figure 3. All servers run on one host and serve a newly registered second-level domain D. The authoritative DNS server accepts TCP and UDP-based queries. The domain is DNSSEC-signed using RSA/SHA-1 with a 1024-bit zone signing key (ZSK) signed by a 2048-bit key signing key (KSK). There are two name servers defined in the top-level zone ($ns1.D$, $ns2.D$), each with glue records for distinctive IPv4 and IPv6 addresses. Secure delegation takes

place by DS records with both SHA-1 and SHA-256 digests of the KSK in the parent zone.

An MX record set up for the domain points to a mail server with STARTTLS support. A and AAAA records point to a web server that accepts both HTTP and HTTPS. The mail and web server use self-signed certificates, which are secured by DANE via TLSA records [23]. All servers are reachable via IPv4 and IPv6.

We did not configure MTA-STS, DNS Cookies or DANE-based end-to-end email encryption, but our setup allows us to observe if the CA attempts to use it. Our DNS zone is arranged to ensure the CA will have to send the corresponding DNS queries even if they utilize aggressive negative caching [21], which is a DNS cache optimization that omits queries under certain conditions.

B. Detection of Countermeasures

We start capturing all network traffic in our setup before applying for a certificate. Additionally we retain all log files of involved services as these provide further insights. After passing the CA's challenge and obtaining the certificate we filter obviously unrelated traffic (vulnerability scanning by third parties, other scientific measurements) and automatically analyze the remaining data with a custom software. Analyzing this data consists of two steps: First we determine which countermeasures the CA uses and then we derive which attacks are possible under our threat model.

We classify each countermeasure from Table I as *fully*, *partially* or *not* implemented. As noted above not all measures can be detected passively with certainty. Criteria are defined as follows:

1) DNS: Some of the DNS countermeasures are clearly detectable from one DNS message, including DNS Cookies, 0x20 encoding and TCP transport. We consider them to be *fully* implemented if all relevant queries show them and *partially* if only some show them. The set of relevant queries depends on the validation method. In some cases multiple DNS queries are necessary, which implies that all such queries are relevant and must be protected against spoofing.

Source port randomization, multipath and multiserver queries require an evaluation of more than one query. To accurately recognize source port randomization a large number of queries has to be observed. As these do not appear during a regular issuance we assume source port randomization to be implemented *fully* unless we find two DNS queries with identical source addresses and ports. Detecting multiserver queries is performed by grouping relevant queries by query name, class and type. If each group contains more than one distinct destination IP address this mitigation is *fully* implemented. We classify it as *partially* implemented if only a part of the groups fulfill the requirement.

For detecting multipath queries we perform the same grouping approach but instead of IP addresses we consider autonomous system numbers (ASN). Source IP addresses of queries are mapped to ASNs using the routing history API provided by RIPE NCC[2]. If each group contains more than one distinct ASN, then multipath is *fully* implemented; if only part of the groups fulfills the requirement, we classify it as *partially* implemented. A corner case lies in conflicting prefix announcements by different autonomous systems. If we observe more than one query in a group and one of the source IP addresses cannot be mapped uniquely to an ASN, we consider optimistically this group to fulfill the requirement as well.

We define DNSSEC validation as *fully* implemented if the relevant DNS queries have the DNSSEC OK flag and a DNSKEY query has been observed within *TTL* seconds before or after that query. Otherwise we define DNSSEC validation as *not* implemented. If redundant queries are observed (same name, class and type) we consider the flag as set if at least one query has it set. This definition may result in a misclassification in favor of the CA if the CA uses multiple resolvers, of which only some are validating, or if the CA fetches the DNSKEY but fails to validate correctly.

The countermeasure of mass-spoofing detection cannot be observed passively and thus must be omitted from our analysis.

2) Web: HTTP multipath is considered to be implemented *fully* if we observe redundant requests for the same CA-defined URL from different source IP addresses. DANE support is indicated by a DNSSEC-secured query for TLSA under domain name _443._tcp.D, followed by an HTTPS request.

The time stamp of the last HTTP request allows to refine the DNS classification. As the HTTP request definitely marks completion of the name lookup process, DNS queries performed afterwards are unrelated to it. We therefore consider DNSSEC validation of the lookup part of HTTP-based validation to be *not* implemented if there is no DNSKEY query before the last HTTP request.

For TLS-RND the same classifications apply.

3) Email: Countermeasures against SMTP attacks revolve around encryption. Whether the sending mail transfer agent requests STARTTLS is determined by analyzing our mail server logfile.

Additional countermeasures are determined by looking for DNS-based queries: DANE queries for TLSA under _25._tcp.D, MTA-STS for TXT under _mta-sts.D and end-to-end encryption by queries for SMIMEA under _smimecert.D or OPENPGPKEY under _openpgpkey.D. For each of these countermeasures DNSSEC validation is checked individually. Similar to HTTP-based validation, delivery of the email provides a time boundary. We consider the DNSKEY query as DNSSEC validation indicator only if we observe it before the email has been sent.

C. Attack Vulnerability

Based on the observed implementation state of each countermeasure, we classify each validation method as either *vulnerable* against an attack, *mitigated* or *no vulnerability*

[2]https://stat.ripe.net/, Accessed 2019-04-02

found. As a result of our classification of countermeasures, *vulnerable* implies that an attack is definitely feasible, because we demonstrated the absence of an appropriate countermeasure. *Mitigated* implies that the attack is potentially feasible, but countermeasures exist that might mitigate it. *No vulnerability* implies that we did not find evidence for a feasible attack, though we cannot rule out further vulnerabilities.

If the CA implements DNSSEC validation then there is *no vulnerability* against DNS off-path attacks. We consider an off-path attack as *mitigated* if the CA implements at least one of the applicable countermeasures (source port randomization, DNS cookies, 0x20 encodig, TCP transport, multipath, multiserver). Otherwise, we consider the CA as *vulnerable* against DNS off-path attacks.

DNSSEC validation also protects against DNS on-path attacks, i.e. there is *no vulnerability*. Most other DNS countermeasures are ineffective against an on-path attack. Only DNS multipath and multiserver have the potential to *mitigate* it, otherwise the CA is *vulnerable*.

If DANE is used then the CA is *not vulnerable* to an active HTTP attack. Employment of HTTP multipath *mitigates* such an attack under certain circumstances. If neither is implemented, the CA is *vulnerable* to this attack. The same countermeasures are effective against an active TLS attacker, i.e. TLS multipath causes a *mitigated* and DANE a *not vulnerable* rating.

To be *not vulnerable* to an SMTP eavesdropper, STARTTLS or end-to-end encryption has to be used. Otherwise the CA is *vulnerable* to this attack. If no countermeasures against an active SMTP attack are implemented, then the CA will be *vulnerable* to man-in-the-middle attacks. Usage of DANE, end-to-end encryption or MTA-STS implies that the CA is *not vulnerable*.

IV. Tested Certificate Authorities

We selected a set of CAs issuing the most certificates on the market. As data sources we used [17] and W3Techs.com. StartCom is included although it recently lost trust by major browser vendors[3]. We attempted but could not test WoSign, because it targets the Chinese market and we were unable to provide one of the payment options supported by WoSign.

As we investigate the DV certificate issuance, we omitted CAs that provide OV or EV certificates only. In one case we inadvertently purchased an OV certificate, which we noticed only after the payment: DigiCert asked for a proof of personal identity and put the personal name as "Organization" attribute in the subject field. We leave DigiCert in the results, as it still provides insights about the domain validation practice. Note however that the applicant had to identify himself in this case, whereas this was not necessary for any of the DV certificates.

Table II lists the 15 CAs considered in our evaluation, their respective validation methods and the price paid. As of January 2018 this list covers 96% of all publicly trusted certificates used on Alexa's TOP 10 million websites[4]. We test all domain validation methods offered by each CA (except where noted). Therefore we need multiple domain names, one for each tested validation method. However, we can reuse the same domain name when acquiring certificates from different CAs, because CAs do not collate issued certificates with each other. The second-level domain names we used for the evaluation consist of two or three randomly chosen words from an English dictionary, registered under `.com` or `.net`.

As shown in Table II, not every entity that we consider as CA issues certificates under a trusted root CA certificate with its name. While there are currently 152 root CA certificates in the Mozilla store[5], this number does not directly relate to the number of trusted CAs. On the one hand CAs use more than one trusted certificate for operational reasons (e.g. DigiCert alone owns 29 trusted root certificates), on the other hand there are companies which sell certificates without being present in root stores. The latter case is possible due to trusted intermediate CA certificates effectively granting that company CA capabilities. In case of Thawte the trusted root CA certificate surprisingly depends on the chosen validation method. We do not differentiate these cases but define a CA as an entity that issues certificates under its own brand.

Amazon is a special case, as it is the only CA that does not support the applicant to generate or retrieve the private key of the certificate. Instead, the private key is deployed on Amazon's TLS load balancers, which forward cleartext requests to cloud instances.

V. Results

We tested the DNS-based, email-based validation in November and December 2017 and the HTTP-based validation in May 2018. The median time it took between submission of a certificate request till receipt of the signed certificate was 7:10 minutes (P_{25} = 4:21 min, P_{75} = 9:22 min).

We present the results of our security evaluation separately for each tested validation method: DNS-based validation in Table III, web-based validation in Table IV and Table V, email-based validation in Table VI. The results are broken down according to the classification from Section III-C as vulnerable (●), mitigated (◐) or not vulnerable (○) against specific attack classes. Detailed lists of detected security measures are given in the appendix.

We consider vulnerability against DNS attacks also for the web and email-based validation, as DNS attacks suffice to undermine any validation method. One might assume the CA would achieve the same security rating against DNS attacks across all validation method, but that is not the case. For example, AlphaSSL, Certum, GoDaddy and Starfield Technologies use DNSSEC validation during HTTP or email-based validation, but strangely not during DNS-based domain validation.

[3]https://blog.mozilla.org/security/2016/10/24/distrusting-new-wosign-and-startcom-certificates/, Accessed 2018-06-20

[4]https://w3techs.com/technologies/overview/ssl_certificate/all, Accessed 2018-01-29

[5]https://wiki.mozilla.org/CA/Included_Certificates, Accessed 2018-01-29

TABLE II: List of tested CAs and their validation methods. Price is minimum of all validation methods (differences due to promotions and exchange rates).

CA	Tested Validation Methods	Trusted Root CA	Price
AlphaSSL	Email, DNS	GlobalSign	17 €
Amazon	Email, DNS	Starfield Technologies	0 €
Certum	Email, DNS, HTTP	Certum	15 €
Comodo	Email, DNS, HTTP	Comodo	0 €†
DigiCert	Email[1] with identity validation	DigiCert	148 €
GeoTrust	Email	GeoTrust	0 €†
GlobalSign	HTTP[2]	GlobalSign	107 €
GoDaddy	Email, DNS, HTTP	Go Daddy Group	54 €
Let's Encrypt	DNS, HTTP, TLS-SNI	IdenTrust	0 €
Network Solutions	Email	USERTRUST	71 €
RapidSSL	HTTP[3]	DigiCert	7 €
SSL.com	Email, DNS, HTTP	USERTRUST	41 €
Starfield Technologies	Email, DNS, HTTP	Starfield Technologies	51 €
StartCom	Email	–	0 €
Thawte	DNS, HTTP	DigiCert	30 €
Thawte	Email	Thawte	30 €

Further available validation methods: [1]HTTP, DNS; [2]DNS, Email; [3]Email
† Obtained free trusted trial certificate.

A. CA Selection Attack

Similarly, we present vulnerabilities of the CAA lookup separately from other DNS lookups, even though CAA is basically a DNS-based mechanism. In most cases the CAA and other DNS ratings are identical, but there are a few discrepancies where CAA is more secure and even a few cases where CAA appears less secure than the other DNS lookups. As a positive note, all CAs perform a CAA lookup as required by the CA/B Forum[6] since September 2017. Despite being strongly recommended [22], not all CAs authenticate the CAA record via DNSSEC. These CAs are vulnerable to a CA selection attack by a DNS on-path attacker, and potentially vulnerable to off-path attackers.

B. On-path DNS Attack

DNSSEC must be used to effectively protect from on-path DNS attacks. Several CAs have shown indications for DNSSEC validation according to our classification criteria, but not all. Without DNSSEC, redundant queries from multiple vantage points (*DNS multipath*) or to multiple authoritative DNS servers (*DNS multiserver*) have a chance to mitigate (◐) an on-path attack. Several CAs show indication for such a countermeasure: Amazon/DNS (two different IPv4 resolvers), Amazon/Email (18 unique IPv4 resolvers querying MX records), GlobalSign/HTTP (6 identical queries by the same GeoTrust IPv4 address, additional lookup via Google Public DNS), Thawte/DNS (usage of both IPv4 and IPv6). However, we cannot confirm with our optimistic approach whether this is actually a security measure, or whether the redundant queries are due to operational reasons or functionality unrelated to the domain validation.

[6]https://cabforum.org/2017/03/08/ballot-187-make-caa-checking-mandatory/

C. Off-path DNS Attack

Since an on-path attacker is more capable than an off-path attacker, all countermeasures against an on-path DNS attack apply to an off-path attack as well. Thus the off-path attack will exhibit at most the same vulnerabilities as the on-path attack, but not more than that.

In fact, all CAs seem to have appropriate mitigations against off-path attacks in place. None of the CAs showed indication for the lack of source port randomization. Some CAs had countermeasures beyond that in place, e.g. 0x20 encoding by Let's Encrypt and DNS Cookies by Amazon, RapidSSL and Thawte.

D. HTTP Attack

Preventing HTTP attacks requires opportunistic HTTPS and authentication with DANE. Although some CAs use DNSSEC, none attempted to query for our `TLSA` record during HTTP-based validation. Thus, every HTTP-based validation was vulnerable to an active man-in-the-middle attacker (Table IV).

For Certum we observed an anomaly as we were instructed to place our validation token v in a file under `/.well-known/pki-validation/`v`.html`. As *"the Request Token or Random Value MUST NOT appear in the request"* [11, Section 3.2.2.4.6], this is a violation of the baseline requirements.

Comodo and SSL.com allowed the applicant to specify whether HTTP or HTTPS should be used. This choice does not increase security, because the attacker posing as applicant would simply choose HTTP. We do not see a benefit of exposing such security-relevant option to the applicant, since a fall-back approach sketched in Section II-D provides the same flexibility with less potential for misuse.

A potential mitigation (◐) consists of performing multiple HTTP requests from different vantage points (*HTTP multi-path*). We observed indications for this behavior for SSL.com,

TABLE III: Vulnerabilities found for DNS-based validation. Vulnerable (●), mitigated (◐), found no vulnerability (○).

CA	CAA		DNS	
	On-path	Off-path	On-path	Off-path
AlphaSSL	○	○	●	◐
Amazon	●	◐	◐	◐
Certum	○	○	●	◐
Comodo	○	○	○	○
GoDaddy	●	◐	●	◐
Let's Encrypt	○	○	○	○
SSL.com	○	○	○	○
Starfield Technologies	●	◐	●	◐
Thawte	○	○	◐	◐

TABLE IV: Vulnerabilities found for HTTP-based validation.

CA	CAA		DNS		HTTP
	On-path	Off-path	On-path	Off-path	Active
Certum	○	○	○	○	●
Comodo	○	○	●	◐	●
GlobalSign*	○	○	◐	◐	●
GoDaddy	●	◐	○	○	●
Let's Encrypt	○	○	○	○	●
RapidSSL	○	○	○	○	●
SSL.com	○	○	○	○	◐
Starfield Technologies	●	◐	○	○	●
Thawte	○	○	○	○	●

* GlobalSign solved the DNS vulnerabilities in August 2018 after we disclosed our results.

TABLE V: Vulnerabilities found for TLS-SNI-based validation.

CA	CAA		DNS		TLS
	On-path	Off-path	On-path	Off-path	Active
Let's Encrypt	○	○	○	○	●

TABLE VI: Vulnerabilities found for email-based validation.

CA	CAA		DNS		SMTP		
	On-path	Off-path	On-path	Off-path	Passive	Active	TLS version
AlphaSSL	○	○	○	○	○	●	1.2
Amazon	●	◐	●	◐	○	●	1.0
Certum	●	◐	●	◐	○	●	1.0
Comodo	○	○	○	○	○	○	1.2
DigiCert	○	○	○	○	○	●	1.2
GeoTrust	●	◐	●	◐	○	●	1.0
GoDaddy	●	◐	●	◐	○	●	1.2
Network Solutions	○	○	●	◐	○	●	1.2
SSL.com	○	○	●	◐	○	●	1.2
Starfield Technologies	●	◐	○	○	○	●	1.2
StartCom	●	◐	●	◐	●	●	none
Thawte	●	◐	●	◐	○	●	1.0

although it is unclear whether this is a security measure or an operational artifact.

Starfield Technologies queried three URLs one after another: first a file path that the applicant was not asked for to use (`/.well-known/pki-validation/godaddy.html`) over HTTP, followed by HTTPS. Only then in a third HTTP request ([...]`/starfield.html`) the CA was able to obtain the requested token. As Starfield Technologies is a subsidiary of GoDaddy, this indicates a brand-unaware backend software trying multiple well-known paths until one succeeds.

E. TLS-SNI Attack

TLS-SNI (Table V) has only been supported by Let's Encrypt. As Let's Encrypt does not use DANE, this leaves it vulnerable to man-in-the-middle attackers in the same way as HTTP. We tested TLS-SNI in November 2017 before TLS-ALPN was drafted. However, the protocol changes between TLS-SNI and TLS-ALPN do not affect the security assessment under our threat model.

F. Email/SMTP Attack

Most CAs allow the applicant to choose a specific WHOIS or constructed email address. The set of constructed addresses was always restricted to the five well-known local parts. Amazon, DigiCert, Godaddy and Starfield Technologies sent separate emails to all five constructed addresses over separate SMTP connections. This might increase the attacker's chance to intercept the token, but it also increases the chance for the domain owner to notice an unauthorized certificate request.

Email-based validation has the unique property of being vulnerable to passive attackers, which requires encryption to render this attack impossible. All CAs except StartCom used STARTTLS to upgrade SMTP to an encrypted connection. Thus only StartCom is vulnerable against passive attackers (Table VI). For informational purposes we also list the established TLS protocol version. Several CAs negotiated the obsolete TLS 1.0, which is not recommended due to security concerns [36].

An active attacker could impersonate the destination mail transfer agent or deny STARTTLS capabilities to force a downgrade. Unlike with HTTP, we observed support for SMTP with DANE with Comodo, Network Solutions and SSL.com, which could prevent active attacks against SMTP. However, only Comodo queried the `DNSKEY` record in time before sending the mail, which is necessary for DNSSEC validation. Thus, although Network Solutions and SSL.com retrieved the `TLSA` record, the record is unusable due to a lack of validation [23] and the CA remains vulnerable. Other CAs did not use DANE at all and are thus vulnerable against active SMTP attackers.

We did not observe any support for MTA-STS nor support for end-to-end email encryption.

G. Discussion

The inconsistent security ratings raise the question of what causes the diverse results within a single CA. In case of the CAA mechanism, which became mandatory only recently, it makes sense to assume that best current practices are implemented while old processes remain unchanged. But as we have seen, in some cases the CAA validation was less secure than other DNS lookups. The inconsistency of security measures may be the result of a diligent security assessment, which leads under careful consideration of operational realities to diverse security requirements. Or they may be the result of technological legacies, grown infrastructures and ad-hoc implementations with the lack of an overall security strategy.

In a couple of cases the CA relies on Google Public DNS, which is a DNSSEC-validating resolver service. This is a useful addition to increase the number of vantage points, as long as the CA validates DNSSEC signatures additionally by itself. However, there are justifiable doubts about this assumption. Consider for example the following CAs, which showed DNSSEC indications only for a subset of validation methods: Certum (DNS and HTTP), GoDaddy (HTTP) and Starfield Technologies (HTTP and email). In each of these cases the CA relied on Google Public DNS for name resolution, but we never observed a `DNSKEY` query from the resolvers residing in the CA's network. A potential explanation why the other validation methods are not protected by DNSSEC is thus: the CA does not support DNSSEC validation and any indication for DNSSEC support is an artifact of the CA's decision to outsource part of their name lookups and trusting Google Public DNS.

VI. Experimental Validation

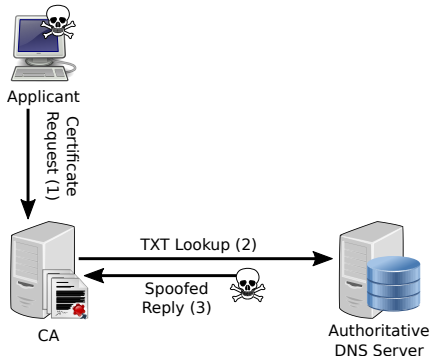
We have found vulnerabilities for all tested CAs despite making optimistic assumptions. To validate our methodology we attempt to obtain certificates by performing a network-level attack on our infrastructure while requesting certificates via different validation methods. We select GoDaddy/DNS, Thawte/HTTP and Network Solutions/Email for these attacks. For our setup we use a dedicated domain name and sign its zone with DNSSEC like a legitimate domain owner would do. DANE records for HTTPs and SMTP with STARTTLS are generated accordingly.

Figure 4 sketches the approaches. A malicious applicant requests a certificate from the CA (1) followed by a validation method-depending attack. For DNS-based validation (Figure 4a) we spoof responses to `TXT` queries using the packet sniffer/generator kamene[7] to complete the challenge. The original query is not modified, which causes the authentic response to eventually reach the CA as well and reveals that an attack has occured.

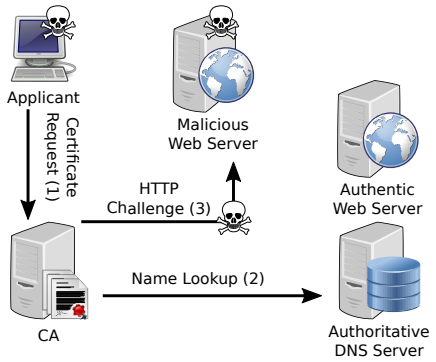
DNS queries of HTTP-based validation (Figure 4b) are not tampered with. Instead all HTTP traffic is tunneled to a malicious web server (3), which responds with the correct token to the CA's request.

As explained in Section V-F, Network Solution performs DANE queries without DNSSEC validation during email-based validation. We exploit this vulnerability by tampering

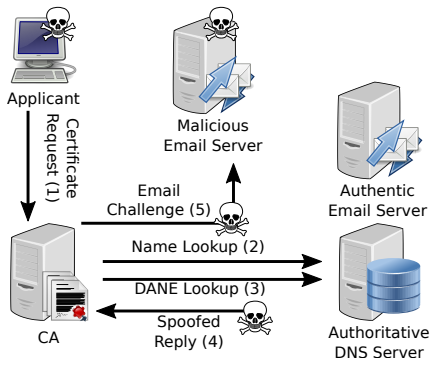
[7]https://github.com/phaethon/kamene



(a) DNS



(b) HTTP



(c) Email with DANE denial

Fig. 4: Attacks performed on validation methods.

with DNS and SMTP (Figure 4c). The attacker actively denies existence of DANE records (4) and tunnels SMTP traffic to a malicious mail transfer agent (5) without STARTTLS support.

All attacks were performed in September 2018 and succeeded. For DNS-based validation we observed a singular `TXT` query, which was spoofed accordingly. HTTP-based validation resulted in one HTTP request for the actual domain and a second one for its `www` subdomain. In both cases the malicious web server served the validation token. Email-based authentication caused a lookup for DANE records which was answered with a spoofed name error response. The subsequent SMTP connection was tunneled to the malicious mail transfer agent and the email was transmitted in plain text.

We obtained trusted certificates after these validation steps in all cases, i.e., all attacks were performed successfully. We did not observe additional countermeasures other than those already revealed by our survey approach, which substantiates the validity of the method.

VII. Disclosure of Results

We disclosed our findings directly to the CAs. We informed AlphaSSL, Comodo, Thawte, SSL.com and Starfield Technologies in April 2018 about inconsistent infrastructure behavior and Certum about its HTTP CA/Browser Forum Baseline Requirements violation. After a refinement of our method we informed the remaining CAs in July 2018 as they were vulnerable via at least one validation method. We reported the successful practical attacks from Section VI to affected CAs in September 2018. Reactions varied greatly between CAs.

Starfield Technologies replied that DNSSEC was not mandated by the CA/B Forum Baseline Requirements and is therefore not supported. Similarly Thawte stated that implementing DNSSEC was not a priority from a security point of view.

Let's Encrypt acknowledged the vulnerabilities, but justified that the DANE approach for securing HTTP and TLS-based validation is too complex. Instead Let's Encrypt favors the restriction of validation methods via the CAA record, which is currently under specification [30].

Certum acknowledged the baseline violation and reported that they deployed a correction in July 2018. Implementing DNSSEC validation was said to be under consideration.

GlobalSign acknowledged the vulnerabilities and announced a new infrastructure with DNSSEC support. They provided voucher codes for us to repeat the analysis. We were able to confirm consistent DNSSEC support for HTTP, email and DNS-based validation in August 2018 which eliminates all vulnerabilities except for active HTTP and SMTP attacks.

The remaining CAs made no factual statements.

VIII. Related Work

Scheitle et al. [35] surveyed the adoption of the CAA record and compliance to it. Compared to our work they examine the CAA mechanism only, but in greater depth as they uncover certificate misissuance with deliberately broken CAA configurations.

Bhargavan et al. [6] formally modelled and verified the ACME protocol used by Let's Encrypt. They discovered a cross-CA attack possible with ACME, where one misbehaving CA forwards a certificate issuance request to another CA and succeeds. While this is a valid attack, we did not consider it in our threat model as the impact is moderate.

Borgolte et al. [8] demonstrated the problem with residual trust in domain names that point to unused IP addresses in the cloud. An attacker can grab the IP address and succeed with domain validation although the domain is not under his control. A similar problem are mistyped nameserver addresses and outdated WHOIS records [39], which an attacker can exploit to pretend control over a domain. This demonstrates the risk

when the attacker can freely choose the validation method and when the CA does not support appropriate countermeasures to harden the validation.

Brand et al. [9] demonstrated that some CAs are vulnerable against an IP fragmentation-based DNS off-path spoofing attack, which lowers the entropy that an attacker must guess. They suggest a DNS multipath approach to protect from DNS man-in-the-middle attacks. As our analysis has shown, the HTTP and SMTP connections must be secured as well, otherwise the attacker can resort to these validation methods.

Certificate studies. Various studies examine the certificates found in the wild [25], their trust relationship to intermediate and root CA certificates [17], and forged certificates encountered [26], [12]. By inspecting a large body of deployed certificates Delignat-Lavaud et al. [15] identified numerous violations of the baseline requirements in 2012–2013. Kumar et al. [29] followed up in 2017 and found that the percentage of misissued certificates decreased to 0.02%, but a long tail of small authorities still issued non-conformant certificates. They developed a certificate linter that checks for errors in certificates but not *"whether the destination domain was correctly validated"* [29], which is the research gap that we address in this paper.

Certificate authority model. Arnbak et al. [1] surveyed the market share and price of DV (avg. $81), OV (avg. $258) and EV (avg. $622) certificates in 2013. They argue that the certificate market is driven by brand reputation or feature bundles, but not security. As the actual CA security is largely unobservable for the potential buyer, she has to make her decision based on the perception of security or other incentives. Our work sheds light on the domain validation practices and discloses weaknesses in that part of the system.

Matsumoto and Reischuk [33] suggested to incentivize CAs for careful identity validation by making them financially accountable. In case of a security incident, an insurance payout should be triggered automatically to the domain owner. Some CAs like Comodo[8], Thawte[9] or GlobalSign[10] in fact offer a warranty bundled with a certificate. However, the security of the system depends on the weakest CA that persists in the browser trust stores, whose security mishaps are not covered by the warranty plan. Several approaches attempt to fix this structural flaw by rethinking the public-key infrastructure, either as addition to the existing CA model [38] or as fundamental alternative [43], [4], [5], [42], [13].

IX. Recommendations

For every tested CA we found at least one attacker model that allows certificate misissuance under at least one domain validation method. One of the oddities in our results are varying DNS countermeasures subject to the validation method. However, secure domain name lookups are required for all Internet-based validation methods, including email and HTTP. The requirement for performing DNSSEC validation should be codified in the baseline requirements. This would provide domain owners with an opt-in way of enhancing security while at the same time maintaining compatibility with non-DNSSEC domains. As DNSSEC signing has not been adopted universally [40], CAs should consider using a combination of additional DNS mitigations listed in Section II-C.
Recommendation: use DNSSEC signing (as domain owner) and DNSSEC validation (as CA).

While DNSSEC support is a necessary prerequisite to prevent attacks on domain validation, it is not sufficient. HTTP, TLS-RND and email-based validations require further measures to provide application layer security. The application layer protocol can benefit from using TLS, but requires a mechanism for downgrade resilience against active attackers. In case of email, Opportunistic DANE [16] prevents downgrade attacks on TLS-secured SMTP connections. DANE could be used in principle to secure HTTP or TLS-RND as well. However, the ACME specification [3] mandates all HTTP-based validation to be performed without HTTPS due to concerns of improperly configured virtual hosts on shared web servers[11]. Similarly we observed CAs to allow applicants (including the attacker according to our threat model) to choose whether HTTP validation should be performed using HTTP or HTTPS.
Recommendation: use a downgrade resilient signaling mechanism like DANE or CAA to choose secured validation channels when available.

Using `CAA` records reduces the potential for certificate misissuance. In its most simple form, the domain owner uses an empty `issue` property to lock the domain from certificate issuance or renewal when not needed. To protect from DNS spoofing, DNSSEC should be used. Only if all CAs performed DNSSEC validation, on-path attackers could be deterred effectively from obtaining illegitimate certificates. This would empower the domain owner to effectively control which CAs may issue certificates for her domain in the presence of attackers. Otherwise restricting a domain to a high-security CA will be moot, if an attacker is able to convince a less secure CA of a false CAA response.
Recommendation: use `CAA` records with DNSSEC.

Certificate authorities can utilize additional CAA parameters to allow restriction to a certain subset of domain validation methods. Combined with DNSSEC this achieves a downgrade-resistant signaling, preventing CA selection attacks as well as fallbacks to insecure protocols. This is especially important, because the validation methods have varying security properties.
Recommendation: use `CAA` records for authorization of the allowed domain validation methods.

[8]https://www.instantssl.com/compare-ssl-certificates.html, Accessed 2018-06-27

[9]https://www.thawte.com/ssl/, Accessed 2018-06-27

[10]https://www.globalsign.com/en/ssl/compare-ssl-certificates/, Accessed 2018-06-27

[11]https://www.ietf.org/mail-archive/web/acme/current/msg00524.html

X. Conclusion

Our results have shown that attacks on domain validation are within reach of a network-level attacker. All tested CAs proved to be vulnerable under our threat model via at least one validation method. In each of these cases a secure countermeasure exists already, but was not supported by the CA. The web-based validation in particular proved to be prone against man-in-the-middle attackers. In one case the CA violated the baseline requirements of the web-based validation. We showed experimentally that the domain validation vulnerabilities found in our analysis can actually be exploited.

Following up on our research question about the security of Let's Encrypt, we can conclude that its domain validation is at least as secure as traditional CAs. Let's Encrypt uses preventive security measures like DNSSEC where a couple of other CAs do not. The HTTP and TLS-SNI validation methods are nevertheless vulnerable to man-in-the-middle attackers.

In general, a higher price for a certificate did not correlate with an increase in deployed security measures. This is however a purely technical view, as we did not consider additional buying incentives like bundled warranty, brand trust or logo availability.

Another core finding is that HTTPS is not enough as sole provider of web security. Before HTTPS can be called into action, DNSSEC is required to secure domain validation and obtain a certificate without the hazard of a man-in-the-middle attack. This applies to all Internet-based validation methods, as they all require a secure domain name lookup. On the other hand, setting up a domain with DNSSEC relies on HTTPS to interact securely with the domain name registrar. Ultimately, both systems complement each other and close their mutual security gaps that exist during setup.

Future work should follow up on our optimistic classification and test whether the indications for a security measure reflect that the measure is actually in use. This could be achieved with deliberate misconfigurations, e.g. invalid DNSSEC signatures or mismatching DANE records. Active attacks like TLS downgrade attacks could provide further insights about the domain validation reality. We did not consider wildcard certificates in our study and leave it for future work. Apart from domain validation, a security assessment of the extended validation processes is also of interest. Furthermore, our focus on the 15 largest CAs has omitted the long tail of small CAs, where future work might discover more vulnerabilities.

References

[1] Arnbak, A., Asghari, H., Van Eeten, M., and Van Eijk, N. Security collapse in the https market. *Commun. ACM 57*, 10 (Sept. 2014), 47–55.

[2] Barnes, R., Hoffman-Andrews, J., McCarney, D., and Kasten, J. Automatic certificate management environment (acme). Internet-Draft draft-ietf-acme-acme-09, IETF Secretariat, December 2017. http://www.ietf.org/internet-drafts/draft-ietf-acme-acme-09.txt.

[3] Barnes, R., Hoffman-Andrews, J., McCarney, D., and Kasten, J. Automatic certificate management environment (acme). RFC 8555, RFC Editor, March 2019.

[4] Basin, D., Cremers, C., Kim, T. H.-J., Perrig, A., Sasse, R., and Szalachowski, P. Arpki: Attack resilient public-key infrastructure. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2014), CCS '14, ACM, pp. 382–393.

[5] Basin, D., Cremers, C., Kim, T. H. J., Perrig, A., Sasse, R., and Szalachowski, P. Design, analysis, and implementation of arpki: An attack-resilient public-key infrastructure. *IEEE Transactions on Dependable and Secure Computing 15*, 3 (May 2018), 393–408.

[6] Bhargavan, K., Delignat-Lavaud, A., and Kobeissi, N. Formal modeling and verification for domain validation and acme. In *Financial Cryptography and Data Security* (Cham, 2017), A. Kiayias, Ed., Springer International Publishing, pp. 561–578.

[7] Birge-Lee, H., Sun, Y., Edmundson, A., Rexford, J., and Mittal, P. Bamboozling certificate authorities with BGP. In *27th USENIX Security Symposium (USENIX Security 18)* (Baltimore, MD, 2018), USENIX Association, pp. 833–849.

[8] Borgolte, K., Fiebig, T., Hao, S., Kruegel, C., and Vigna, G. Cloud strife: mitigating the security risks of domain-validated certificates. In *Proceedings of NDSS'18* (February 2018), Internet Society.

[9] Brandt, M., Dai, T., Klein, A., Shulman, H., and Waidner, M. Domain validation++ for mitm-resilient pki. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (2018), ACM, pp. 2060–2076.

[10] Budington, B. Symantec issues rogue ev certificate for google.com. https://www.eff.org/deeplinks/2015/09/symantec-issues-rogue-ev-certificate-googlecom, Sep 2015. Accessed: 2018-06-27.

[11] CA/B Forum. Baseline requirements documents. https://cabforum.org/baseline-requirements-documents/. Accessed: 2018-06-27.

[12] Cui, M., Cao, Z., and Xiong, G. How is the forged certificates in the wild: Practice on large-scale ssl usage measurement and analysis. In *Computational Science – ICCS 2018* (Cham, 2018), Y. Shi, H. Fu, Y. Tian, V. V. Krzhizhanovskaya, M. H. Lees, J. Dongarra, and P. M. A. Sloot, Eds., Springer International Publishing, pp. 654–667.

[13] Dacosta, I., Ahamad, M., and Traynor, P. Trust no one else: Detecting mitm attacks against ssl/tls without third-parties. In *Computer Security – ESORICS 2012* (Berlin, Heidelberg, 2012), S. Foresti, M. Yung, and F. Martinelli, Eds., Springer Berlin Heidelberg, pp. 199–216.

[14] Dagon, D., Antonakakis, M., Vixie, P., Jinmei, T., and Lee, W. Increased dns forgery resistance through 0x20-bit encoding: Security via leet queries. In *Proceedings of the 15th ACM Conference on Computer and Communications Security* (New York, NY, USA, 2008), CCS '08, ACM, pp. 211–222.

[15] Delignat-Lavaud, A., Abadi, M., Birrell, A., Mironov, I., Wobber, T., and Xie, Y. Web pki: Closing the gap between guidelines and practices. In *Proceedings of NDSS'14* (February 2014), Internet Society.

[16] Dukhovni, V., and Hardaker, W. Smtp security via opportunistic dns-based authentication of named entities (dane) transport layer security (tls). RFC 7672, RFC Editor, October 2015.

[17] Durumeric, Z., Kasten, J., Bailey, M., and Halderman, J. A. Analysis of the https certificate ecosystem. In *Proceedings of the 2013 Conference on Internet Measurement Conference* (New York, NY, USA, 2013), IMC '13, ACM, pp. 291–304.

[18] Eastlake, D. Transport layer security (tls) extensions: Extension definitions. RFC 6066, RFC Editor, January 2011. http://www.rfc-editor.org/rfc/rfc6066.txt.

[19] Eastlake 3rd, D., and Andrews, M. Domain name system (dns) cookies. RFC 7873, RFC Editor, May 2016.

[20] Friedl, S., Popov, A., Langley, A., and Stephan, E. Transport layer security (tls) application-layer protocol negotiation extension. RFC 7301, RFC Editor, July 2014. http://www.rfc-editor.org/rfc/rfc7301.txt.

[21] Fujiwara, K., Kato, A., and Kumari, W. Aggressive use of dnssec-validated cache. RFC 8198, RFC Editor, July 2017.

[22] Hallam-Baker, P., and Stradling, R. Dns certification authority authorization (caa) resource record. RFC 6844, RFC Editor, January 2013.

[23] Hoffman, P., and Schlyter, J. The dns-based authentication of named entities (dane) transport layer security (tls) protocol: Tlsa. RFC 6698, RFC Editor, August 2012. http://www.rfc-editor.org/rfc/rfc6698.txt.

[24] HOFFMAN, P., AND SCHLYTER, J. Using secure dns to associate certificates with domain names for s/mime. RFC 8162, RFC Editor, May 2017.

[25] HOLZ, R., BRAUN, L., KAMMENHUBER, N., AND CARLE, G. The ssl landscape: A thorough analysis of the x.509 pki using active and passive measurements. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference* (New York, NY, USA, 2011), IMC '11, ACM, pp. 427–444.

[26] HUANG, L. S., RICE, A., ELLINGSEN, E., AND JACKSON, C. Analyzing forged ssl certificates in the wild. In *2014 IEEE Symposium on Security and Privacy* (May 2014), pp. 83–97.

[27] HUBERT, A., AND VAN MOOK, R. Measures for making dns more resilient against forged answers. RFC 5452, RFC Editor, January 2009.

[28] JACKSON, C., SIMON, D. R., TAN, D. S., AND BARTH, A. An evaluation of extended validation and picture-in-picture phishing attacks. In *Financial Cryptography and Data Security* (Berlin, Heidelberg, 2007), S. Dietrich and R. Dhamija, Eds., Springer Berlin Heidelberg, pp. 281–293.

[29] KUMAR, D., WANG, Z., HYDER, M., DICKINSON, J., BECK, G., ADRIAN, D., MASON, J., DURUMERIC, Z., HALDERMAN, J. A., AND BAILEY, M. Tracking certificate misissuance in the wild. In *2018 IEEE Symposium on Security and Privacy (SP)* (2018), vol. 00, pp. 288–301.

[30] LANDAU, H. Caa record extensions for account uri and acme method binding. Internet-Draft draft-ietf-acme-caa-06, IETF Secretariat, January 2019. http://www.ietf.org/internet-drafts/draft-ietf-acme-caa-06.txt.

[31] LAURIE, B., LANGLEY, A., AND KASPER, E. Certificate transparency. RFC 6962, RFC Editor, June 2013.

[32] MARGOLIS, D., RISHER, M., RAMAKRISHNAN, B., BROTMAN, A., AND JONES, J. Smtp mta strict transport security (mta-sts). RFC 8461, RFC Editor, September 2018.

[33] MATSUMOTO, S., AND REISCHUK, R. M. Certificates-as-an-insurance: Incentivizing accountability in ssl/tls. In *SENT'15* (2015), Internet Society.

[34] MOZILLA FOUNDATION. Mozilla root store policy. https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/policy/, June 2017. Accessed: 2018-06-27.

[35] SCHEITLE, Q., CHUNG, T., HILLER, J., GASSER, O., NAAB, J., VAN RIJSWIJK-DEIJ, R., HOHLFELD, O., HOLZ, R., CHOFFNES, D., MISLOVE, A., AND CARLE, G. A first look at certification authority authorization (caa). *SIGCOMM Comput. Commun. Rev. 48*, 2 (May 2018), 10–23.

[36] SHEFFER, Y., HOLZ, R., AND SAINT-ANDRE, P. Recommendations for secure use of transport layer security (tls) and datagram transport layer security (dtls). RFC 7525, RFC Editor, May 2015. http://www.rfc-editor.org/rfc/rfc7525.txt.

[37] SHOEMAKER, R. Acme tls alpn challenge extension. Internet-Draft draft-ietf-acme-tls-alpn-05, IETF Secretariat, August 2018. http://www.ietf.org/internet-drafts/draft-ietf-acme-tls-alpn-05.txt.

[38] SYTA, E., TAMAS, I., VISHER, D., WOLINSKY, D. I., JOVANOVIC, P., GASSER, L., GAILLY, N., KHOFFI, I., AND FORD, B. Keeping authorities "honest or bust" with decentralized witness cosigning. In *2016 IEEE Symposium on Security and Privacy (SP)* (May 2016), pp. 526–545.

[39] VISSERS, T., BARRON, T., VAN GOETHEM, T., JOOSEN, W., AND NIKIFORAKIS, N. The wolf of name street: Hijacking domains through their nameservers. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2017), CCS '17, ACM, pp. 957–970.

[40] WANDER, M. Measurement survey of server-side dnssec adoption. In *2017 Network Traffic Measurement and Analysis Conference (TMA)* (June 2017).

[41] WOUTERS, P. Dns-based authentication of named entities (dane) bindings for openpgp. RFC 7929, RFC Editor, August 2016.

[42] YU, J., CHEVAL, V., AND RYAN, M. Dtki: A new formalized pki with verifiable trusted parties. *The Computer Journal 59*, 11 (2016), 1695–1713.

[43] ZHU, W. T., AND LIN, J. Generating correlated digital certificates: Framework and applications. *IEEE Transactions on Information Forensics and Security 11*, 6 (June 2016), 1117–1127.

Countermeasure	AlphaSSL	Amazon	Certum	Comodo	GoDaddy	Let's Encrypt	SSL.com	Starfield Technologies	Thawte
DnsBit0x20	No	No	No	No	No	Full	No	No	No
DnsBit0x20CAA	No	No	No	No	No	Full	No	No	No
DnsCAADNSSEC	Full	Partial	Full	Full	Partial	Full	Full	Partial	Full
DnsDNSCookie	No	No	No	No	No	No	No	No	No
DnsDNSCookieCAA	No	Full	No	No	No	No	No	No	Partial
DnsDnskey	Full	No	Full	Full	Full	Full	Full	No	Full
DnsMultiServer	Partial	Full	No	Partial	No	No	Full	No	Full
DnsMultiServerCAA	No	No	No	Full	No	No	Full	No	Partial
DnsMultipath	No	Full	No	No	No	No	Full	No	Full
DnsMultipathCAA	No	No	No	Full	No	No	Full	No	Partial
DnsRelevantDNSSEC	No	No	No	Full	No	Full	Full	No	No
DnsTcp	No	No	No	No	No	No	No	No	No
DnsTcpCAA	No	Partial	No	No	No	No	No	No	No

TABLE VII: Raw list of countermeasures for DNS-based validation.

Countermeasure	Certum	Comodo	GlobalSign	GoDaddy	Let's Encrypt	RapidSSL	SSL.com	Starfield Technologies	Thawte
DaneTls443	No	No	No	No	No	No	No	No	No
DnsBit0x20	No	No	No	No	Full	No	No	No	No
DnsBit0x20CAA	No	No	No	No	Full	No	No	No	No
DnsCAADNSSEC	Full	Full	Full	Partial	Full	Full	Full	Partial	Full
DnsDNSCookie	No	No	No	No	No	Partial	No	No	No
DnsDNSCookieCAA	No	No	No	No	No	Full	No	No	Full
DnsDnskey	Full	Full	Full	Full	Full	Full	Full	Full	Full
DnsMultiServer	Partial	No	Full	Full	Partial	Partial	Partial	Partial	Partial
DnsMultiServerCAA	No	Full	No	No	No	No	Full	No	No
DnsMultipath	Full	No	No	Full	No	Partial	Full	Full	Partial
DnsMultipathCAA	No	Full	No	No	No	No	Full	No	No
DnsRelevantDNSSEC	Full	No	No	Full	Full	Full	Full	Full	Full
DnsTcp	No	No	No	No	No	No	No	No	No
DnsTcpCAA	No	No	No	No	No	Partial	No	No	Partial
HttpMultipath	No	No	No	No	No	No	Full	No	No

TABLE VIII: Raw list of countermeasures for HTTP-based validation.

Countermeasure	Let's Encrypt
DaneTls443	No
DnsBit0x20	Full
DnsBit0x20CAA	Full
DnsCAADNSSEC	Full
DnsDNSCookie	No
DnsDNSCookieCAA	No
DnsDnskey	Full
DnsMultiServer	Partial
DnsMultiServerCAA	No
DnsMultipath	No
DnsMultipathCAA	No
DnsRelevantDNSSEC	Full
DnsTcp	No
DnsTcpCAA	No
TlsMultipath	No

TABLE IX: Raw list of countermeasures for TLS-SNI-based validation.

Countermeasure	AlphaSSL	Amazon	Certum	Comodo	DigiCert	GeoTrust	GoDaddy	Network Solutions	SSL.com	Starfield Technologies	StartCom	Thawte
DaneTls25	No	No	No	Full	No	No	No	Partial	Partial	No	No	No
DnsBit0x20	No	No	No	No	No	No	No	No	No	No	No	No
DnsBit0x20CAA	No	No	No	No	No	No	No	No	No	No	No	No
DnsCAADNSSEC	Full	Partial	Partial	Full	Full	Partial	Partial	Full	Full	Partial	Partial	Partial
DnsDNSCookie	No	No	No	No	No	No	No	No	No	No	No	No
DnsDNSCookieCAA	No	No	No	No	No	No	No	No	No	No	No	No
DnsDnskey	Full	Full	No	Full	Full	No	No	Full	Full	Full	No	No
DnsMultiServer	Partial	Partial	No	Partial	Partial	No	No	No	No	No	No	No
DnsMultiServerCAA	No	No	No	No	Partial	No	No	Full	No	No	No	No
DnsMultipath	No	Partial	No	Full	Partial	No	No	No	No	No	No	No
DnsMultipathCAA	No	No	No	No	No	No	No	Full	No	No	No	No
DnsRelevantDNSSEC	Full	No	No	Full	Full	No	No	No	No	Full	No	No
DnsTcp	No	No	No	Partial	No	No	No	No	No	No	No	No
DnsTcpCAA	No	No	No	No	No	Partial	No	No	No	No	No	Partial
TlsSmtp	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full	No	Full

TABLE X: Raw list of countermeasures for email-based validation.