Proof of static DCV security

Henry Birge-Lee
Princeton University
https://henrybirgelee.com/

Proof strategy

- Security proofs are hard: difficult to prove security against every potential attack
- One strategy to cope with this is to utilize proof by contradiction to show that a newly-proposed protocol/system is as secure as an existing system
- Argument goes as follows:
  - Assume there exists an adversary (let's refer to this adversary as Malary) that can compromise the new protocol
  - Can we construct an agent (let’s call this agent Jafar) that is able to use Malary as a building block to compromise a known existing protocol
  - If the answer is yes we can conclude the logical statement:
    - IF new protocol is compromised THEN old protocol is compromised
  - From here, we also know the contrapositive is true:
    - IF old protocol is not compromised THEN new protocol is not compromised
  - With this statement we can be sure that the new protocol is as secure as the old protocol

A note on this proof

The PKI exists within an ecosystem, not in a sterile theory environment as these proofs are typically used. The following proof is scoped only to whether an adversary capable of fooling one of these new methods would be able to fool an existing method.

Introducing these methods has additional ecosystem change (e.g., more domains may start using them). This itself has security implications. One notable feature of static DCV is that it is vulnerable to account compromise which is another attack surface beyond just bypassing DCV (i.e., out of scope of this proofs). I feel there is not a significant risk added given that current regulations allow for 1y validation reuse and when an applicant gets a certificate is public knowledge due to CT. Thus, there is little existing protecting against account compromise, but this is an attack vector. Static DCV could be introduced with additional account security (e.g., 2FA or limited-scope API keys) to reduce this risk.

Comparison of 3.2.2.4.14 to Aaron’s 12/12/24 proposal

- Aaron’s proposal in email today most closely matches up to 3.2.2.4.14
- 3.2.2.4.14 will be the old existing protocol
- Aaron’s proposal will be the new protocol
- Let’s assume Malary is an adversary capable of fooling the new protocol
  - Malary can make a one-time change to the TXT record of a domain using a given value
  - Because Malary is only capable of compromising a static DCV method, Malary can only ever change the domain once and must wait an arbitrarily long amount of time before requesting a certificate

Proof

- Jafar asks Malary to put in the email jafar.badguy@example.com into victim-domain.com IN TXT
- Malary completes this task, but takes an arbitrarily long amount of time during which Jafar cannot request a certificate
- Jafar now approaches Trusted CA and asks the CA to use method 3.2.2.4.14 to validate victim-domain.com
- The CA lookups victim-domain.com IN TXT and finds jafar.badguy@example.com
- The CA emails jafar.badguy@example.com with the random token
- Jafar gets the token by checking his email and presents it to the CA
- Jafar has now passed domain control validation using 3.2.2.4.14

Contrapositive

- In the previous proof we showed:
    - IF there is an adversary that can compromise static DCV implemented in Aaron's proposal
    - THEN there is an adversary that can compromise 3.2.2.4.14
- Given this is true, we know the contrapositive is also true:
    - Contrapositive of IF p THEN q is IF not q THEN not p
    - IF there is no adversary that can compromise 3.2.2.4.14
    - THEN there is no adversary that can compromise static DCV implemented in Aaron's proposal

Additional relations between static DCV and existing DCV

- Static DCV via CNAME at underscore-prefixes subdomain
  - Method: applicant must upload CA-specified value to underscore-prefixed subdomain
  - Probably as secure as dns-01 or 3.2.2.4.7
- Static DCV via token in CAA record
  - Method: applicant must upload CA-specified value to CAA record set
  - Probably as secure as 3.2.2.4.13
- Static DCV via 302 or 301 redirect of /.well-known
  - Method: applicant must upload CA-specified value to the location header of a 301 or 302 response for /.well-known or /.well-known/acme-challenge or /.well-known/pki-validation
  - Probably as secure as http-01 or agreed-upon change to website v2

Some close but not quite proposals

- The below proposals are similar to the previous proposals but cannot be formally proved to have comparable security
- Static DCV via individual TXT record at underscore-prefixed subdomain
    - Since the applicant can only upload a single TXT record and cannot “redirect” the CA’s request via a CNAME, this method is not necessarily a subset of the dns-01 or 3.2.2.4.7 attack surface
- Static DCV via individual file within /.well-known/acme-challenge
    - If it is not a redirect code, a single file with a value will only complete a single request and after the applicant can no longer update this file, http-01 will fail. Only showing control of a redirect code is strictly a subset of http-01