Considerations for using short-term certificates

draft-nir-saag-star

Yoav Nir
Thomas Fossati
Yaron Sheffer
Toerless Eckert
Why are we doing this?

• Lots of interest in short-term certificates
  • In the standards process (ACME, STIR, ANIMA).
  • Datacenters.
  • Deployed multi-node systems (NSFs, Storage).

• The interest is in avoiding revocation checking.

• Need a document that:
  • Tells people this is OK, and
  • Tells people what needs to be done to make it OK.
Why Avoid Revocation

• Revocation makes RPs and EEs more complex.
  • Introduces many modes of failure.

• Revocation makes protocols more complex.
  • Stapling or online fetching of CRL or OCSP response.

• Revocation makes the system more expensive.
  • Need a revocation server (CRL DP or OCSP Responder) that is always available.

• Revocation makes connection start-up slower.

• Revocation doesn’t make sense any more.
  • Why sign BLOB A, and then sign BLOB B to say that the signature on BLOB A is still valid?
Short-Term Certificates

• We define short-term certificates as certificates with a short period of time between issuance and the NotAfter date.
  • Issuance date is not necessarily the same as NotBefore date.
• No hard definition of “short”, but we propose a few guidelines
  • Regular web certificates are typically issued for 1-2 years.
  • Regular ACME certificates are for three months.
  • Revocation is still necessary. We can’t have a compromised certificate keep being valid for months.
  • To justifiably forego revocation, we need the certificate to be valid for much less time.
Short-Term Auto-Renewal Certificates

• You have hundreds or thousands of end entities and you renew them every week, every day, every hour.
• There is no way to do this manually without failing all the time.
• It’s fine to have some manual intervention in issuing the first certificate, but renewal MUST be automatic.
• Standard and proprietary protocols:
  • ACME
  • Vendor-specific (CPMI, SCP, many more)

• Note that this is a different definition to the acme-star draft, where a certificate is only STAR if it uses ACME.
Revocation

• Revocation is just non-renewal.
• The time for revocation to apply is limited by certificate lifetime.
• RPs MUST take expiry seriously.
  • No 72-hour ”grace period”
Is This For The Web?

• No reason why not, but...

• The Web is different.
  • The web is a huge investment by multiple bodies.
  • Commercial and non-commercial CAs with expensive, redundant infrastructure to handle issuance and revocation.
  • Many millions of end entities with no prior relationship to any CA.

• What is unaffordable to some datacenter may be affordable for the big web.

• So while short-term certificates may work for the web, it is not the focus of this document.
Sample Use Case - NSFs

• Multiple NSFs:
  • Firewalls
  • Deep inspection firewalls
  • Malware detonation chamber
  • VPN gateways
  • IDS/IPS

• All communicating with each other (SNMP or proprietary)
• VPN gateways using certificates for IKE.
Sample Use Case - Datacenter

- Datacenters are too big and heterogenic to consider them a "safe space".
- Regulatory compliance requires that certain data is never transmitted in the clear: national security, financial, health, other private.
- Possible solution is to have host-to-host IPsec between all pairs of nodes.
  - I2NSF is working on this now
- Still need credentials for these hosts.
- Can add the PKI within the datacenter to manage these certificates.
Sample Use Case – Proprietary

• Virtual block storage example:
  • 100s of data clients run as a driver on application servers.
  • 100s or 1000s of data servers serve data using local block storage devices (magnetic or SSD) and synchronize changes with secondary copies.
  • 2-3 meta-data managers manage volume allocation, volume mapping, balancing, recovery, snapshots, replication, and much more.
  • 2-3 management servers install, configure and manage the other services.
  • Any two or more of the above may be co-located on the same machine.

• Need authentication between these components.
• (Possibly) need data encryption between them.
• The management server is a natural candidate to also run the CA.
Use case and automatic renewal system

• ANIMA-WG “ANI” infra
  • Use-case: “ACP/ANI”
    • Fully autonomicomically created, cert secured virtual network infrastructure
    • Cert to be used to secure any network service/protocol wanting to use it
  • Fully automated Cert system: BRSKI – draft-ietf-anima-bootstrapping-keyinfra)
    • EST (RFC7030) based PKI. Solves manual step left in EST enrolment:
    • New Vendor credential to authenticate network to pledge (“voucher” draft-ietf-anima-voucher)
    • Automatic connectivity (via proxy/ACP) to permit pledge to enrol without routed IP addr
    • Can use classical revocation – but overall design simpler with short lived cert
  • Simple extension to leverage BRKI to make short lived certs more resilient
    • Expired Cert -> only EST server needs to accept expired cert for its TLS connection
    • Any service utilizing short lived cert fails untei nothin else
    • Aka: simple change of semantic of expiry time in cert on EST server
Benefits

• Simpler PKI
  • No need for CDP or OCSP Responder.
  • Can do “fire and forget” CA
    • But then how do you revoke?
  • Easier to integrate with some “management function”.

• Faster start-up and connection establishment
  • EE doesn’t need to get OCSP response for stapling
  • RP doesn’t have to perform revocation checks.

• Simpler code for RPs
  • No need for both a signed blob, and a second signed blob saying that the first signed blob is still OK.
Operational Challenges

• Clock skew always causes failures.
  • With STAR certificates it causes more failures sooner.
  • Yes, it’s still an issue in 2018.

• A down CA will give you down nodes sooner
  • Can be mitigated by re-issuing early (4 day certificate, 2 day renewal)
    • Above example can still fail over the weekend.
  • Need to tune lifetime and re-issuance time to strike the proper balance between robustness and responsiveness to compromise.

• Spamming Certificate Transparency
Security Properties

• Regular certificates with the RP checking an OCSP server when the EE presents its certificates (and no caching!) is the best security.

• Many applications cannot live with the latency. Specifically browsers refuse to do online revocation checking.

• So second best is OCSP Stapling: the EE gets an OCSP response valid for some time (hours, days, weeks) and presents that with the certificate.
  • This can be enforced with Must-Staple.

• Our claim is that STAR is roughly equivalent to stapled OCSP
  • As long as certificate issuance is as frequent as OCSP validity.
Security Challenges

• Can’t just revoke. The certificate is valid until expiration.
  • Mitigation by short lifetime.
  • Even with revocation, both CRLs and OCSP responses are cached.
  • Revocation is never immediate.

• No differentiation between system issues that made renewal late (clock skew, network issues) and real the-bad-guys-have-my-private-key revocation.
  • So you have to treat expiry the same as revocation.
  • No “grace period”.
Summary

• We don’t need signedBlob2 to validate that signedBlob1 is still valid.

• Revocation checks made sense when issuing CRLs was easy while issuing certificates was hard.
  • ACME and proprietary protocols make re-issuance easy enough.

• Ideally, the lifetime of a certificate should be the same as the “lifetime” or a CRL or OCSP response.
  • But processing rules are different: A CRL or OCSP Response is valid past its nextUpdate time, but a certificate is not.
  • Clock skew is a bigger problem.

• The target is a BCP document that says that STAR certificates are secure enough, and how to deploy them so that they are.
Please Adopt Me!