Using least privileged design principals to improve trust in the online marketplace

Ryan Hurst, GlobalSign
ryan.hurst@globalsign.com
@rmhrisk

Presented at the NIST Workshop on Improving Trust in the Online Marketplace
April 10-11, 2013

Only the minimum amount of access necessary to perform an operation should be granted, and that access should be granted only for the minimum amount of time necessary.

Have we been doing this?

I don’t think so. At least not as much as we can.

Lots of tail risk from Certificate Authorities that most users will never legitimately encounter a certificate from. – Brad Hill

SSL Observatory found 1,482 “CAs”: do all of these need the ability to issuing for any name space? Do they all need to issue Server SSL certificates?

Some observations.

- Limited usage of the BasicConstraints extension’s pathLenConstraint.
- Limited use of some clients ability to restrict CA usages to subsets via EKU.
- Almost zero usage of Name Constraints outside the Federal Bridge.
- No observed adoption of automatic issuance auditing for these CAs.
- Often single PKI used for internal and external certificates even though internal often doesn’t need public trust.

But why is this the case?

- These mostly represent CAs have legitimate need to represent their community of users.
- It’s hard to imagine all the possible use cases in the beginning of a project.
- It’ also hard to change a the CAs once the are setup.
- Why add additional complexity this is hard enough.

Do they need to be unconstrained?

- In some cases yes.
- But In most they serve only a few namespaces.
- In some they are also limited to a few use cases.
- We need to remember though that requirements change.

Where to start reducing exposure?

- First we need to understand their needs:
  - Who do they serve?
  - What are the certificates used for?
  - What clients rely on the certificates?
  - How many namespaces are involved?
  - How frequently does the list of namespaces change?

- Find ways to respond to changed needs more quickly.
- Look at how we can use the tools we have available to restrict permissions.

What tools do we have to work with?

- Hosted “managed” CAs where TTP manage keys and restrict access to pre-vetted namespaces.
- Subordinating “issuing CAs” with pathLength to 0 to ensure root is always aware of new issuing CAs.
- Using separate CA infrastructure for internal and external scenarios.
- Delegate CAs specific sets of usage via Extended Key Usage and rely on the clients enforcement of nesting rules (86%+ clients support)
- Adopting Name Constraints restricting to a limited set of namespaces.

What have we found implementing this.

- Client and library support for Name Constraints (NC) is very good, exception is OSX/IOS (11.19%); still workable means needs to be marked Non-Critical.
- XP’s Support of NC more restrictive than other platforms; still workable but requires planning.
- Effective enforcement of NC requires specifying all namespaces, not just those used.
- Client support of “nested EKUs” better than expected.
- Many customers who were hosting their own CAs can actually move hosted services quite easily /w API support.
- For many customers applying name constraints for “core name spaces” is acceptable if paired with managed services for more dynamic list of names.

But what about the problems?

- As we know complex mechanisms are hard to deploy; they are:
  - Hard to model, Miss-understood, and misconfigured.
- Name Constraints is complicated complicated:
  - DN ordering issues, inconsistent naming issues.
- One of the largest delays is associated with these organizations getting grasp on minimal namespaces.

Can we do more?

- Yes.
- Adoption of Certificate Transparency of (CT) within our own operations.
- Making publication of an authoritative list of subordinate CAs standard.
- Leveraging CT as a means to reduce reliance on post issuance audits.
  - Many of these CAs use EJBCA and the MS Certificate Authority; can implement CT as an extension.
- Work to close the gap on client support for Name Constraints so we can ultimately mark the extension critical.
- Adopt Name Constraints for hosted CAs as well

Why do I care?

- A simple changes that can greatly reduce the surface area of attack.
- Can potentially increase adoption of SSL as a more flexible alternatives to wildcards.
  - Improve key management practices.
  - Move to shorter lived certificates.
  - Fewer barriers to adoption.
- Other technology efforts to reduce risk such efforts such as CT, CAA and the variants of pinning are great but don’t address all risk – approach discussed here is complimentary.

A Band-Aid, not a Panacea

- Top 8-10 global CAs can’t apply this to most of their infrastructure – these CAs responsible >95% of SSL certificates.

- Large organizations with very complicated namespaces and distributed IT are likely unable to adopt many of these approaches.

- Those clients that do not understand these restrictions do not benefit.

- Significant, long-term attack surface reduction for what is arguably the weakest link.

QUESTIONS