

# Defensive PKI

## (What happens when PKI fails?)

Kelvin Yiu

Steve Whitlock

Tim Polk

Carl Ellison

# The Problem

- Dec 2008: Exploitation of MD5 collisions
- May 2008: Debian (OpenSSL) RNG error
- What's next?
  - Hash 2<sup>nd</sup> pre-image attack?
  - Dead key length?
  - Dead PK algorithm?
- The infrastructure is “too big to fail”
- What do we do about it?

# Defensive PKI

## (What happens when PKI fails?)

Kelvin Yiu

Steve Whitlock

Tim Polk

Carl Ellison

# Microsoft's Response Options

1. Remove affected root CA certificates (nuclear option #1)
2. Disable MD5 in certificate validation (nuclear option #2)
  - Also break CAs that use non-random serial numbers
3. Work with affected CAs to update their infrastructure before attack is practical to others
  - Trust Sotirov and team, and the security of their PS3 cluster

## Constraints

- Cannot break millions of users (without sufficient justification)
  - Certificate error UX in latest browsers is very effective in stopping users
  - Previously issued certificates were not vulnerable
  - Did not affect all CAs that use MD5
  - Many long lived subordinate CA certificates use MD5

# How Microsoft responded to the MD5 collision attack

- Microsoft immediately contacted affected CAs to assess situation
  - CAs were cooperative, but needed more time than usual because attack was announced over the holidays
  - Quick engineering fix, but long QA cycle
- Asked all CAs in our “Root CA Program” to provide information on crypto algorithms in use
  - 52 out of 80 CAs responded. The rest needed more time to gather information
  - Request includes the CA’s use of 1024 bit RSA and MD5 in their hierarchy
  - Most newer CAs have issued with SHA1 only
  - Most have already switched to SHA1

# Some Observations...

- Revocation was not designed to handle pre-image attacks against hash algorithms
  - Old expired certificates are vulnerable
  - Rogue certs will not contain a CDP
  - CAs revoke by specifying a serial number, but serial number could be changed
  - Path validation code cannot require CDP since CDP is not present in many intermediate CA certs
- Subordinate CA may be signing new certificates with SHA-1, but its own certificate may have been issued a lot time ago and still uses MD5
  - Reissuing a subordinate CA certificate may trigger audit
  - Distribution of new root and subordinate CA certificates is very difficult and time consuming
  - Not scalable for MS to distribute sub-CA certs through Windows Update
- From the experience with the Debian bug, replacing certs would be a very slow process. <http://www.eweek.com/c/a/Security/CAs-Not-Getting-Big-Response-to-Debian-Encryption-Flaw/>

# Defensive PKI

## (What happens when PKI fails?)

Kelvin Yiu

Steve Whitlock

Tim Polk

Carl Ellison

# The Minor Issues

- Usability that degrades trust
  - Should I accept the NIST certificate?
  - Now, where did I write down the password protecting my private key?
- Fragility
  - Oops, my root CA certificate expired...
  - My applications were blocked because their server certificates expired
- Our business partners assumed that authentication == authorization
- How do I upgrade the hash algorithms (as opposed to how do I get a good one – see Majors)

# Defensive PKI

## (What happens when PKI fails?)

Kelvin Yiu

Steve Whitlock

Tim Polk

Carl Ellison

# Defense Begins at *Home*

- Relying Parties have ultimate responsibility to ensure a certificate is acceptable
- Acceptability decisions might be based on
  - Policy
  - Certificate status
  - Trust anchor
- But these tools are not enough!
  - Lifecycle issues, crypto issues not adequately addressed

# Cryptographic Lifecycle

- Cryptographic Migration is part of the Lifecycle of a system, but is always an afterthought in the implementation
  - This is not unique to PKIs! Think about DES...
- Migration timelines may vary by application
  - E.g., NIST SP 800-78-1 requires use of
    - 2048 bit RSA for signatures after 12/31/2010
    - 2048 bit RSA for authentication after 12/31/2013

# Overreliance on Policy Leaves Relying Party at Risk

- Relying parties depend on policies (implicitly or explicitly) to ensure that key sizes and hash algorithms provide acceptable levels of security
  - This is not agile, and may be inexact on details that matter to your application!
    - policy mapping can be more abstract, ignoring small discontinuities
  - To increase security and agility, relying parties need crypto based acceptance controls
    - Algorithm, key length, parameters

# Defensive PKI

## (What happens when PKI fails?)

Kelvin Yiu

Steve Whitlock

Tim Polk

Carl Ellison

# It's a fault tolerance problem

- We know how to do fault tolerance.
  - Keep running in spite of failures!
- The failures we need to address are:
  - Bad specific key(s)
  - Bad key length
  - Bad algorithm
- Revocation
  - Flawed
  - Not fault tolerant

# Straw-man Solutions

- Enroll not for 1 certificate but for a binding – and get multiple certificates, with different algorithms and keys, as they come available, during the lifetime of the binding.
- Get not one timestamp but a living sequence of timestamps, each with a newer, better algorithm or key (and sacrifice blindness).
- Fix revocation
  - CDP today in the attacked certificate
  - Revoke keys, algorithms, key lengths; not just certs
    - We need to choose authorities and channels for those

**Q & A**