Attacking Certificate Infrastructures

Rodney Thayer
Canola & Jones

Introduction

Introduction

- Who's Rodney?
- What are we going to talk about?
- Why does this matter?
- Is it bad to be talking about this?
- A word about point of view
- I am not a cryptographer, I'm a crypto plumber

Why should we worry about attacks?

- All sound commerce on the Internet uses certificates in TLS
- Attack vector for all TLS/SSL applications
- It would call into question the trust of the Internet as a vehicle for business
- What if Amazon's certificate weren't trusted?
- What if the Microsoft Windows Update certificate weren't trusted?

Who would be the victims?

- All commerce-based web servers
- SSL and IPSec VPNs
- The majority of sound device management facilities
- Digital signature-based transactions
- Anything else using TLS
- Anything else that's signed (e.g. code, documents)

Threat Model

The Threat Model

- What targets do we expect to be attacked?
- Where do we place our defenses
- What do we do when we're attacked?
- Does this really match the threat model the attackers would use?

Examples of expected attacks

- Compromise of a single certificate (e.g. Amazon.com)
- Compromise of a root (e.g. the VeriSign Class 3 root)
- Obtaining a server certificate fraudulently
- Obtaining a client certificate fraudulently

Defenses

- The CA registration process
- CRLs
- OCSP
- Legal threats
- Customer trust
- CA Reputation
- Expensive cert processing software

Classes of Attacks

How would you attack certificates?

- Certificate Implementations
- Certificate services (CAs, etc.)
- Certificate operations
- Certificate cryptography

Certificate Implementations

- There are relatively few implementations in use
- Lack of genetic diversity
- Essentially all based on OpenSSL or MS (schannel/etc)
- Certificates are hard!
- ASN.1/DER
- Poorly defined
- Complex and arcane standards
- Never fully implemented

Certificate Implementations – what can go wrong?

- Things we know can go wrong because they have already:
  - Forgetting to check the digital signatures (it's happened)
  - Coding the DER implementation wrong (it's happened)
  - Checking expiration dates wrong (it's happened)
  - Poor or missing revocation checks (it's happened)

Certificate Implementations – what can go wrong?

- Things that could go wrong:
  - Buffer overflows from certs with long fields
  - Buffer overflows from CRLs
  - More cert parsing failures
  - Missing private key protection
  - Silicon-based attacks – custom chips may be fast but not correct
  - Fuzzer research – certificates violate “The Fuzzer Theorem”

Certificate Implementations – what can go wrong?

- Sloppy practices:
  - Use of self-signed certificates
  - Training users to ignore certificate errors
  - Poor naming in the issued certs
  - Poor naming in the CA roots
  - Poor root distribution mechanisms
  - Lack of use of status checking
  - Irresponsible private key cloning
  - Poor private key hygiene

Certificate Services – process problems

- We've never really solved the root distribution problem
- "Is the little lock icon there?" is not a sound security check
- "Click fatigue" due to institutionalized use of bad certs
- Poor enforcement of CPS, if it exists at all
- Use of certs in anatomically impossible positions

Certificate Services – infrastructure threats

- CRL server availability
- DoS against the CRL server
- DoS against the OCSP server
- Time attacks
- Expiration apathy
- Reliance on insecure DNS

Certificate Services – trust threats

- There are too many roots
- The retail Certificate Authority business model
- Inconsistent policies among the CAs
- Urban legends spread by the early RSA technology providers
- Identity problems with the certificate authorities
- Too little adoption of private certificate hierarchies

Certificate Services – operational threats

- Theft of private keys
- Does anyone really revoke a certificate?
- Time slew attacks
- Ignoring certificate expiration
- Misuse of certificate technologies
  (i.e. SSL VPNs with no certs)
- Price wars

Certificates - cryptography

- Public key (dual-key) algorithms
- Hash algorithms
- Signature protected areas in cert
- Random number generation
- Formats and infrastructure

Public (dual) key cryptography threats

- Bad seeding – openssl timestamp attack, etc.
- Are those primes really prime?
- Factoring algorithms
- Availability of large scale compute farms that could be weaponized
- The “Irish high school student” problem
- It’s crypto – it’s not known *not* to work
- More exotic attacks
- Little or no attention to alternative algorithms

Digest algorithm threats

- Dobbertin
- Recent MD-5 attacks
- SHA-1 attacks
- Little or no attention to alternative algorithms

What do we do if they break the crypto?

- Tbird's haiku:

SHA-1 has been cracked.

Collisions in the digests:

Oh what shall we do?

15 Feb 2005 – Dr. Tina Bird, *InfoExpress* (and a Shmoo)

Cryptography – signature protection

- Problem:
  A signed object protects the data that's signed
- Make sure all the data to be signed is inside the signed object
- We keep getting this wrong.
- X.509? PKIX? XML? SASL? Whatever's next?

Random number generation

- Needed for key generation
- Uses entropy from environment
- Entropy sources are dodgy
- What if the entropy pool goes dry?
- If the RNG started sucking constants, who would know?
- If the RNG passed out predictable values, who would know?

Cryptography – formats and infrastructure

- The format is also part of the attack surface
- PKCS #1 attack was a surprise
- ASN.1/DER attack was “a surprise”
- What else in the format is an issue?

Conclusion

So why isn't the world falling apart?

- The hackers don't understand crypto
- The users aren't really using certificates
- The flaws we do have are not really visible
- When's the last time you turned off SSL2 and turned on CRL checking in your browser?

Why am I complaining?

- Because these things have slipped by before
- Because the cryptographers and the engineers don't think TOGETHER about threat models
- Because we must assume the hackers are smarter than we are
- Because we still aren't getting the simple stuff right

Recommendations

- Enforce the CPS's
- Test the infrastructure for attacks and confirm the defenses work
- Be more strict about key usage
- Stop using self-signed certificates!
- Make it easier to spin up a new hierarchy
- Stop deploying irrelevant roots
- Use the PKIX technology that's there – key usage fields, etc.

End

- Contact info:

Rodney Thayer

rodney@canola-jones.com

http://www.canola-jones.com