WESTPOINT
internet reconnaissance services

SSL: Paved With Good Intentions

Richard Moore
rich@westpoint.ltd.uk

Why do we need SSL?

- Privacy
- Online shopping
- Online banking
- Identity Protection
- Data Integrity

Early SSL

- First public version was SSLv2
- Developed by Netscape
- Released in November 1994
- No public review prior to release

SSL 2 Basics

- Used X.509 Certificates for identity and key management
- Supported a range of ‘cipher suites’
- No support for extensions
- Protocol was controlled by Netscape

Oops!

- SSL 2 protocol was insecure
- US government forced the ‘Export’ mode where the ciphers were weakened

SSL 3

- Complete rewrite
- New record layer format
- Fixed the security flaws
- Released late 1995
- Still a Netscape protocol

TLS1, Finally…

- Work on this started in 1996
- Intended to be a tidied up version of SSL 3
- 3DES made mandatory
- Designed to be extensible
- Spec ready late 1997

Maybe Not…

- Like SSLv2 and SSLv3 TLS uses X.509 certificates
- X.509 specification was incomplete
- IETF rules means TLS had to wait
- TLS 1 finally released in 1999

TLS 1 Basics

- X.509 certificates used for identity and key management
- Supports a range of cipher suites
- Designed to be extensible
- Not controlled by any single vendor

Certificates

- Certificates are very important
- X.509 standard was not really designed for this
- ASN.1
- Unfortunately complicated

What is in a Certificate?

- Subject
- Issuer
- Public Key
- Extensions

- Simple!

Certificate Authorities

- Certificates should be signed by a CA
- Prevents man-in-the-middle attacks
- Self-signed certificates are bad

Oops We Lost Our Keys

- Keys can be lost or compromised
- We need a way to revoke them
- Certificate Revocation Lists

Except CRLs Don’t Work

- CRLs are too big
- Each CA has their own list
- OCSP is the answer

OCSP

- Online Certificate Status Protocol
- Certificate says where to ask
- Browser checks the OCSP looking for a signed status response

OCSP has Problems Too

- OCSP servers can get overloaded
- CAs don’t update them very well
- Only the leaf certificates are currently checked

OCSP Stapling

- Web server sends the OCSP response as a TLS extension
- Response is signed by the CA so it’s safe
- Only just reaching deployment
- Apache 2.3.3 added support
- Browser support is currently poor

The Story so Far

- TLS 1
- Strong cipher suites
- X.509 Certificates
- Certificate Authorities
- OCSP

- Simple!

What About Virtual Hosting?

- Duplicate elements in Subject and Issuer
- SubjectAltNames
- Wildcards (naturally not specified how they work)
- Server Name Indication

There May be Trouble Ahead

- Now we've had the theory
- The rest is easy…

Ok, So I Lied…

- Subject and issuer actually have a very complex structure
- The Common Name field was used to identify the server
- The RFCs allow certificates to contain arbitrary ASN.1

Getting Silly

- Because X.509 is general it lets you have many fields that are inappropriate for SSL
- Embedded photographs
- Favourite drinks
- Duplicate fields
- Logos

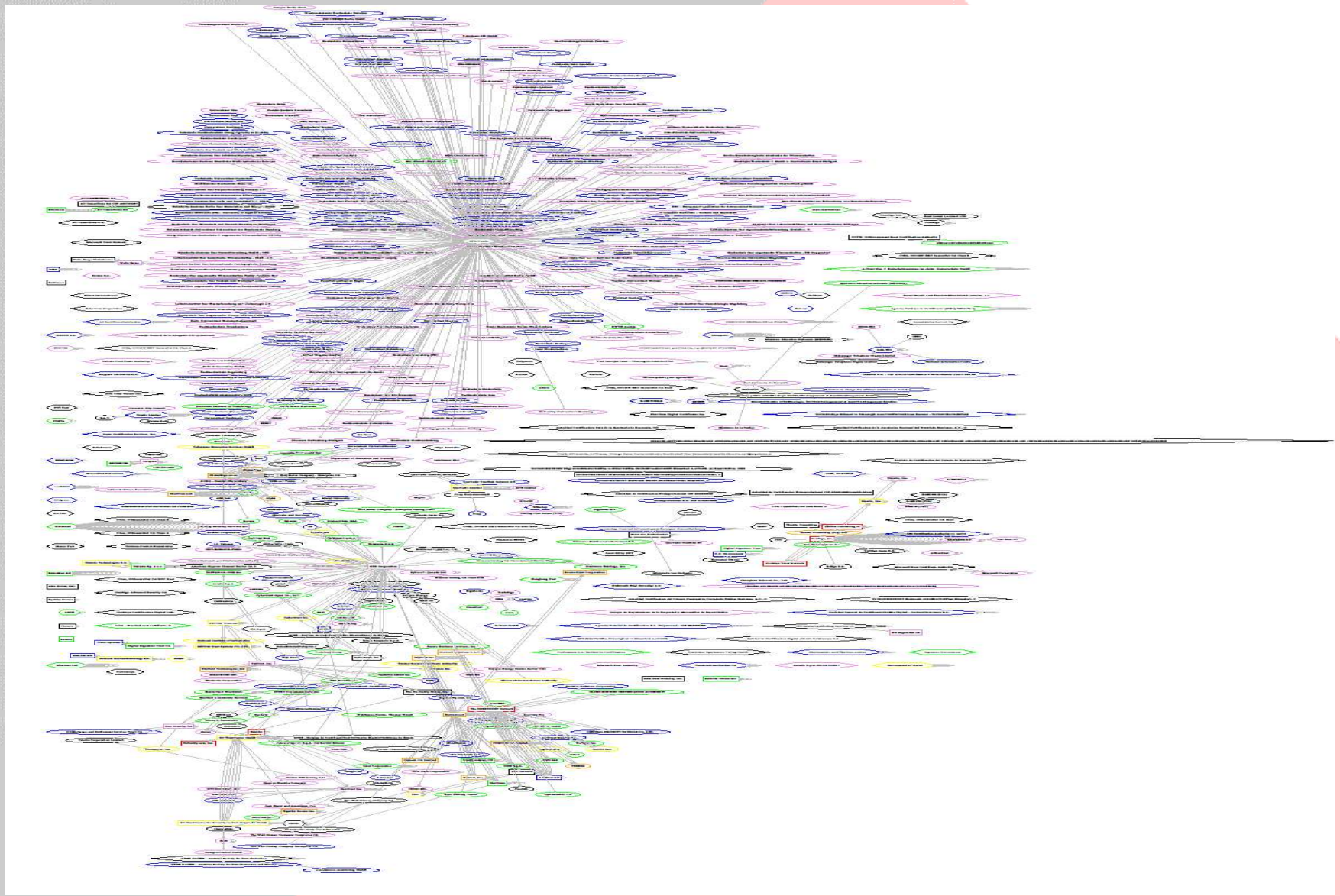
CAs

- Sign anything
- EV certificates (make them do their job)
- Rules for domain validation are only being formulated now
- Get compromised

Who do you Trust?

- People imagine there are few CAs
- Verisign, and a few others…
- The reality is rather different

WESTPOINT
internet reconnaissance services



Random CA Facts

- Any CA can sign a certificate for any domain
- Dozens of German Universities
- Marks and Spencer
- Walt Disney
- 1,482 CA Certificates trustable by Windows or Firefox

Servers Often Misconfigured

- SSL 2 enabled
- Weak ciphers enabled
- NULL ciphers enabled
- Don’t support OCSP pinning
- Don’t support SNI

Certificate Problems

- Lots of default self-signed certs around
- Lots of name mismatches
- Weak certificates due to a bug in Debian's key generation

Bad Practices

- Failing to force users to use HTTPS
- Mixed content
- Content from other sites, especially analytics
- SSL used only for login pages
- Session cookies that aren’t using the secure-only flag

SSL Implementations

- Not checking constraints properly
- ASN.1 problems
- NULs in names
- Shell globs for wildcards

Browsers

- Don’t switch on the security by default
- Poor UI indications for users
- Inconsistent UI
- Even worse on mobile platforms
- Content from more than one HTTPS site are allowed

Users

- Ignore the warning dialogs
- Stick a padlock anywhere and they’re happy
- Don’t even notice if it’s SSL
- So basically, all of the above is somewhat moot!

A World of FAIL

- CAs
- Servers
- Implementations
- Browser
- Users

Summary, SSL is Complex

- A suite of protocols
- All need to be right for real security
- Only as strong as the weakest link in the chain

- Currently the chain has several weak links

Questions?