TLS Certificates on the Web – The Good, The Bad and The Ugly
TLS Certificates

- TLS Ecosystem is almost 20 years old
- Recently endured three certificate-based migrations:
  - Away from MD2 and MD5 to SHA-1
  - Away from small RSA keys to 2048-bit keys or larger
  - Away from SHA-1 to SHA-256
What’s Driving These Migrations?

- Relentless march of attacks (only getting better)
- CA/Browser Forum
  - Baseline Requirements
  - EV Guidelines
  - Certification Authorities
  - Browser vendors
What’s Slowing These Migrations?

- Use of TLS in non-browser applications
  - Mail, XMPP and other non-web servers
  - POS and other devices
  - Lack of auto-update capabilities
- Institutional inertia
  - Companies wait years to perform a server refresh
TLS Certificates – the Good
Deployment of SHA-2 Certificates

Trajectory of SHA-2 deployment is encouraging (Netcraft)
TLS Certificates – the Good

- 99.98% of certificates contain RSA 2048-bit, ECC 224-bit or larger keys (Netcraft)
- 200K certs with keys >= RSA 4096 bits (Netcraft)
- BR Compliance
  - Responsible for standardizing certificate profiles
- 10.7% of sites use EV (TIM)
TLS Certificates – the Bad
Remaining SHA-1 certs will not work in browsers after 2016:
- 13.3% (Netcraft)
- 11.6% (TIM)

US DOD still issuing SHA-1 certificates

More than 1,000 with < RSA 2048-bit or ECC 224-bit (Netcraft)

Browsers continue to add compliance checks
TLS Certificates – the Bad

- EV violations
  - ~6% of all EV certificates (Netcraft)
  - Most don’t have a valid Subject Business Category (unlikely to cause usability problems)
  - Thousands don’t provide EV treatment in Chrome (customer doesn’t benefit from the extra cost of EV)

- BR violations
  - ~3% of all certificates found (Netcraft)
  - Most are policy violations (CN must appear in SAN, invalid Subject State or Country, etc.) unlikely to cause usability problems
TLS Certificates – the Bad

- Strong keys signed by weaker keys (a dozen or so) don’t provide the cryptographic protection expected by the certificate owner:
  - ECC P-384 signed by ECC P-256
  - ECC P-384 signed by RSA 2048
  - RSA 4096 and 8192 signed by RSA 2048

- Certificate expiration is embarrassing

- Almost 4% of sites serve an incomplete certificate chain (TIM)
  - Most browsers don’t try to fetch missing subordinate CA
TLS Certificates – the Ugly
Invalid Certificates abound

- In Netcraft’s survey, approximately two thirds of all TLS certificates seen are valid, issued by a trusted CA. The remaining one-third are either self-signed, expired, signed by an unknown issuer or contain mismatched names.

- One MD5, 3-year cert issued in 2013 by a public CA (RSA 1024-bit key) it’s got 6 other BR violations

- One 512-bit RSA key used by Government of Korea (South), although it’s signed using SHA-2 it’s got 4 other BR violations

- Browsers block access to such sites
**TLS Certificates – the Ugly**

- Invalid Public Key Exponent: one certificate with an RSA exponent of 1
  - TLS data is sent in cleartext

- Multiple CNs are prohibited, but Netcraft found certificates with up to 24 CNs
  - 2009 study demonstrated attacks on certs with multiple CNs

- EV certs with fewer than the correct number of SCTs
  - Customer doesn’t benefit from the extra cost of EV in Chrome
TLS Certificates – the Ugly, continued

- One cert with RSA 15,424-bit key! (includes 72 SAN values!) It’s an Apache server, but not a web site
  - No harm to the Web
- Ten-year end-entity certificates, issued after the BRs became effective
  - Most browsers block public TLS certs with excessive dates
- Certificates with more than 50 SANs (Netcraft)
  - Nothing illegal, but might cause performance problems
Applying What We’ve Learned
2048-bit RSA with SHA-256 is adequate for now

Keep SANs to a minimum (20 or fewer), and only one CN

Replace all weak, invalid, revoked or soon-to-expire certificates

Generate a new key pair every time you replace a certificate

Make sure your EV certificates have the correct number of SCTs

Test your certificate with all major browsers (don’t forget mobile)

Confirm that your CA has correctly issued the certificate
Check Your Work

- Check TLS certificates and configuration on all servers, not just web servers
  - https://cryptoreport.websecurity.symantec.com/checker/
  - https://www.ssllabs.com/ssltest/

- Consider a discovery tool like Certificate Intelligence Center

- Certlint from Amazon (open source)
  - https://github.com/awslabs/certlint
Data Sets

- Netcraft
  - http://www.netcraft.com

- ICSI
  - https://notary.icsi.berkeley.edu/

- Trustworthy Internet Movement (TIM) SSL Pulse
  - https://www.trustworthyinternet.org/ssl-pulse/

- Comodo’s certificate search tool
  - https://crt.sh
Thank you!