IEncrypt – a work-in-progress open-source initiative to increase encryption of traffic to and from .ie web sites, starting with newly registered second-level .ie domains.

Developed by Tolerant Networks Limited
Funded by IEDR
Considered important by both:-)

October 2015, ICANN54, Dublin, Ireland
Aside...

- I'm doing this presentation on behalf of IEDR as I did the dev work:-)
- I'm not doing this as IETF security area director nor for Trinity College Dublin
Contents

- Problem
- Initial Goals
- Benefits of Success
- Technical Approach
- Plan
- Proof-of-Concept
- Conclusion
- <Boring Extra Details as Backup>
- <Demo as we go, or later, or earlier>
Problem

• 20 years on, only about 30% of web sites talk https
  – Precise figure not the point but the trajectory in particular for smaller web sites
• Cleartext => larger attack surface
  – For example: Firesheep, great-cannon
  – More attacks => more support/cost/trouble
• Getting certificates for domains and web-sites is too hard for an average registrant or site admin, or they don't care (enough)
  – or they don't even think of it
Initial Goals

- “IEncrypt” check-box for registrants as they create a new .ie domain with associated web server hosting
  - We're providing proof-of-concept for what's behind that checkbox and happy to talk about providing more

- From the very first DNS query and the very first HTTP response, the hosted site will benefit from state of the art security protocols:
  - DNSSEC validating, chaining up to .ie and .
  - Web site gets an “A” from e.g. ssllabs site tester
  - WebPKI leveraging DNSSEC (at issuance time) using Letsencrypt.org

- Aim is medium level security, **reliability and simplicity are more important goals**
  - Opportunistic security design pattern (RFC7435) says that's a valid approach
Benefits of Success

- Site visitors less likely to be hacked via bogus access point attacks (simple cookie theft)
- Site can make better use of “powerful features” that may no longer be available in browser via cleartext
- Fewer browser warnings (e.g. mixed content) to annoy visitors
- Fewer support calls to registrar as sites consider whether/how to setup TLS and as they (try) do that
  - note: that's a guess, feedback /facts welcome
- Common good – helping realise a better Internet [RFC7258]
- SEO ranking - https scores better!
Technical Approach

• Registrant wants a new .ie domain and web-site hosting (e.g. apache via VIP), with all being provided by Registrar

• Either by default or via a checkbox, the “IEncrypt” option is selected
  – An “IEncrypt advanced” could allow client key gen and other options via CLI, with step-by-step guidance (later)

• Registrar uses DNSSEC and letsencrypt.org (LE) CA to get apache running on port 443 from the very start with no browser warnings and no registrant effort
  – Registrar → Registry gets DNSSEC setup
  – Registrar → LE web server certs setup based on DNSSEC signed zone
DNSSEC Setup

• Registrar generates ZSK and KSK and submits DS to registry
  – Extend existing API hosted by Registry
  – Registry signs zone including DS

• Registrar populates zone with DNSSEC RRs

• DNSSEC rollover automation is very important
  – But actually much less so in this case!
  – A DNSSEC rollover-fail will not affect the web site (today)
Web Server Cert Setup

- Registrar generates web server key pair (and initial content)
- Registrar sets up authorization for new domain with LE and is issued with a DNS-challenge
- Registrar includes response to DNS-challenge in signed zonefile for new domain
- Registrar instantiates VM image in hosting
- Registrar runs apache or nginx install with bettercrypto.org recommended settings and key pair
- Web site gets an “A” from ssllabs.com site tester from start
Reliability

• Critical goal: don't make things worse
• Need key rolling for DNSSEC to work seamlessly with no registrar effort
  - dnssec-tools 'rollerd' does this
  - New RFCs coming on automating DS rollover
• Web server cert update will be seamless
  - letsencrypt.org client does this
  - Can be independent of DNSSEC after 1st keys done
Plan

1) IEDR and TN demo a Proof-of-Concept (PoC)
2) Discuss details with Registrars/Hosters
3) Implement DNSSEC authorization with LE
4) Incorporate registrar/LE feedback into code
5) Implement and deploy in registry
6) Registrars who want to play can test
7) All code/tooling will be open-source, BSD license
PoC Status

- https://testbed.ie proof-of-concept
  - Plays the role of the ccTLD in the PoC
  - testbed.ie pretends to be .ie

- PoC allows one to create a new child domain that is DNSSEC signed and with web server cert issued by LE
  - Working now, runs asynchronously (~5min cycle)
  - Screen-shots + details in backup slides

- Implementation available, all BSD license
  - https://basil.dsg.cs.tcd.ie/code/tcd/iencrypt
    - Mercurial repo, bogus TLS cert:-)
  - May move to github, soon's I get a chance
    - If so, look below https://github.com/sftcd/
PoC Hosts

https://testbed.ie
https://<foo>.testbed.ie
(hoba.ie)

request staging
web server virtual hosts
web server config & keygen (webcfg)
LE client (after DNSSEC done)

NS1
(jell.ie)

request staging
Child - zone signing, KSK & ZSK generation
Parent – add child to named.conf.local, add DS to zonefile, zone signing

NS2
(down.dsg.cs.tcd.ie)

request staging
Re-configure named.conf.local to add slaves
Secondary DNS server
PoC Pictures
PoC Pictures
Conclusion

- It is entirely possible to make DNSSEC useful and easy (actually invisible) to help more web sites use HTTPS today automatically and for free
- Invisible security like this should become the norm
- Once ubiquitous, similar automation can be done for other things (SMTP/DANE)
- Registrars who are hosters and (esp. ccTLD) registries are well positioned to help and be key to success
Thanks!
Questions?

Contact
stephen@tolerantnetworks.com
Backup Slides
Future Goals

- Handle more kinds of hosting
- Help existing domains to use TLS at renewal time
- SMTP/STARTTLS with DANE
PoC Software

• Off-the-shelf:
  – Ubuntu 14.04, Bind (9.9.5), Apache (2.4.7)
  – dnssec-tools (2.0.0, zonesigner, rollerd)
  – letsencrypt client (0.1)
  – openssl (1.01f), curl (7.35.0), php (5.5.9), bash (4.3.11)

• Chewing gum and string:
  – IEncrypt scripts, some via cron, some as root
PoC repos

• Chewing gum, string and docs (this mainly)
  – https://basil.dsg.cs.tcd.ie/code/tcd/ienrcrypt

• Letsencrypt client
  – https://github.com/letsencrypt/letsencrypt
PoC Workflow - Registrant

- Registrant requests foo.testbed.ie at testbed.ie
  - If invalid, error
  - If being processed – say to wait
    else foo.testbed.ie added to “inwork” list
- If not ready, return estimated seconds until ready
  - If ready, return link to https://foo.testbed.ie
- Non-error HTML response pages autorefresh every N seconds
  - N = uniform random between 5 and 15
PoC Workflow - DNSSEC

- (every 5 mins) NS1/children grabs list of new children from testbed.ie
  - Via mutually-authenticated (client-cert) TLS and “hidden” SNI
  - If valid, generates new zonefile, KSK/ZSK and DS
  - Signs Zonefile

- (every 5 mins) NS1/parent grabs list of new children (via file system)
  - Adds DS to parent zone and re-signs
  - Add children to named.conf.local
  - Pushes child to NS2/parent via mutually-authenticated (client-cert) TLS and “hidden” SNI
    - Ready to add new slave
  - Pushes child to testbed.ie via mutually-authenticated (client-cert) TLS and “hidden” SNI
    - Ready to start webcfg client processing (next slide)
  - Re-starts BIND

- (every 5 mins) NS2/parent grabs list of new children from file system
  - Via mutually-authenticated (client-cert) TLS and “hidden” SNI
  - Add children as new slaves to named.conf.local
  - Re-starts BIND
PoC Workflow - ACME

- ACME is the protocol used between LE client and CA service, implemented by letsencrypt client, so once DNSSEC is done...
- (Every 5 minutes) webcfg checks what children to process
- LE client generates key pair for authorization and account handling (for foo.testbed.ie)
- LE client authorizes itself to LE service for foo.testbed.ie
  - Currently via “standalone” option
    - Requires IEncrypt briefly stopping apache on testbed.ie
  - LE client generates new key pair for foo.testbed.ie web server and requests certificate
    - LE service issues certificate
- IEncrypt re-starts apache and sets status of foo.testbed.ie to ready
- Registrant
PoC Restrictions

- LE service today uses fake CA, “happy hacker fake CA”

- Standalone authorization used
  - No DNS, or DNSSEC, DNS is on the way from LE though
  - We'll be signing anyway, we may need to help them verify that the DNS challenge response is from a signed zone
  - Means testbed.ie web server is done now and then for a few seconds

- No port 80 for testbed.ie or <foo>.testbed.ie just due to sharing the same apache install with hoba.ie, hence no HSTS etc. PoC children only ever run on 443
### PoC pictures