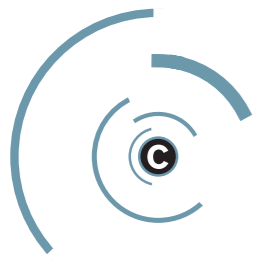


Abusing X.509 certificate features

EuSecWest 2008

Alexander Klink, Cynops GmbH
ak-eusecwest@cynops.de

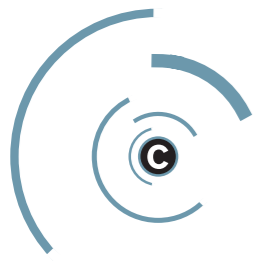


Agenda

... or “what I managed to squeeze into \$timeslot”

- Quick introduction to PKI and X.509
- TLS client certificate user tracking
- Missing hostname binding (Nils Toedtman)
- Why certificate data is untrusted input, too
- HTTP over X.509
- The Debian and OpenSSL debacle



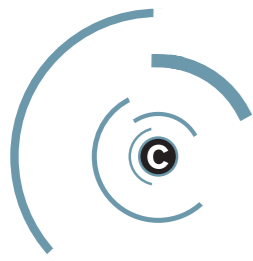


Quick Intro to PKI

RFC 3280 in a nutshell

- PKI = Public Key Infrastructure
- Certificate authorities (CAs) signs binding of information and public key
- X.509 is the format for this block of signed data





Quick Intro to PKI

A basic certificate dump

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 1 (0x1)

Signature Algorithm: sha1WithRSAEncryption

Issuer: C=DE, O=Cynops GmbH, OU=PKI, CN=Cynops CA 1

Validity

Not Before: Nov 9 15:36:06 2006 GMT

Not After : Nov 9 15:36:06 2008 GMT

Subject: C=DE, O=Cynops GmbH, CN=Alexander Klink

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

00:f7:74:5e:84:72:bc:1c:26:5a:89:73:3a:54:87:

[...]

Exponent: 65537 (0x10001)

Signature Algorithm: sha1WithRSAEncryption

93:d1:b8:e0:39:17:05:b4:03:c6:d6:8a:cb:0a:d2:7a:41:bb:

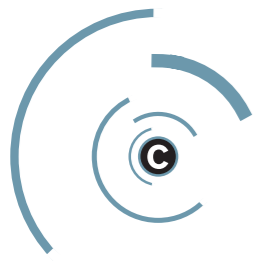
[...]

Subject & Issuer

Public Key

Signature



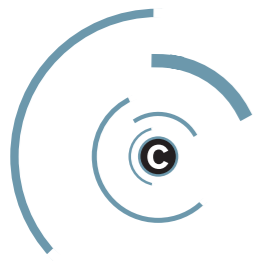


Quick Intro to PKI

Complexity through extensions










- Looks simple?
- Well, that's because it was simplified ...
- “Real” X.509 certificates have extensions
 - Key Usage, Extended Key Usage, Constraints
 - CRL Distribution Points, Authority Info Access
 - subjectAlternativeNames, ...
- most of these are of arbitrary length



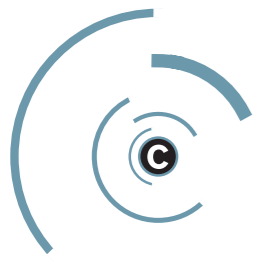


Quick Intro to PKI

Beneath the surface, PKI is everywhere

- “PKI: It’s not dead, just resting” is wrong
- X.509 is in your
 - browser (TLS)   
 - mail client (S/MIME)    
 - office suite (document signatures)  
 - router (IPsec, EAP/TLS, SCEP, ...)
 - ...





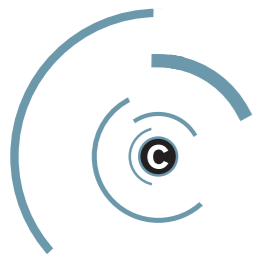
TLS client certificate user tracking

aka cross-domain TLS cookies



- **The feature:** TLS client certificates and their easy generation and installation within a webbrowser
- **The bug:** not letting the user know he is currently using a client certificate and thus sending out private information

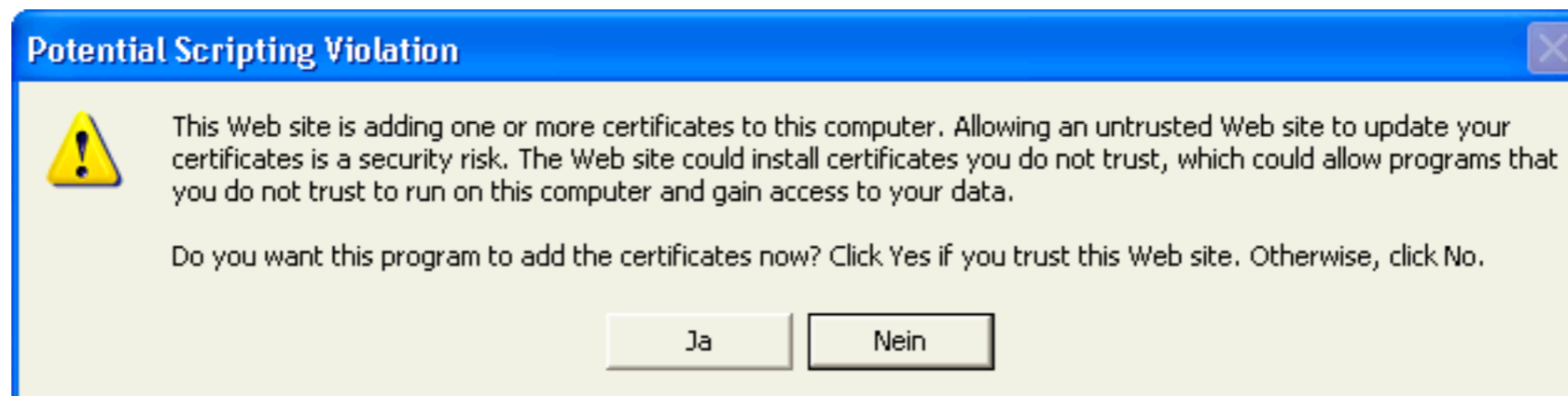
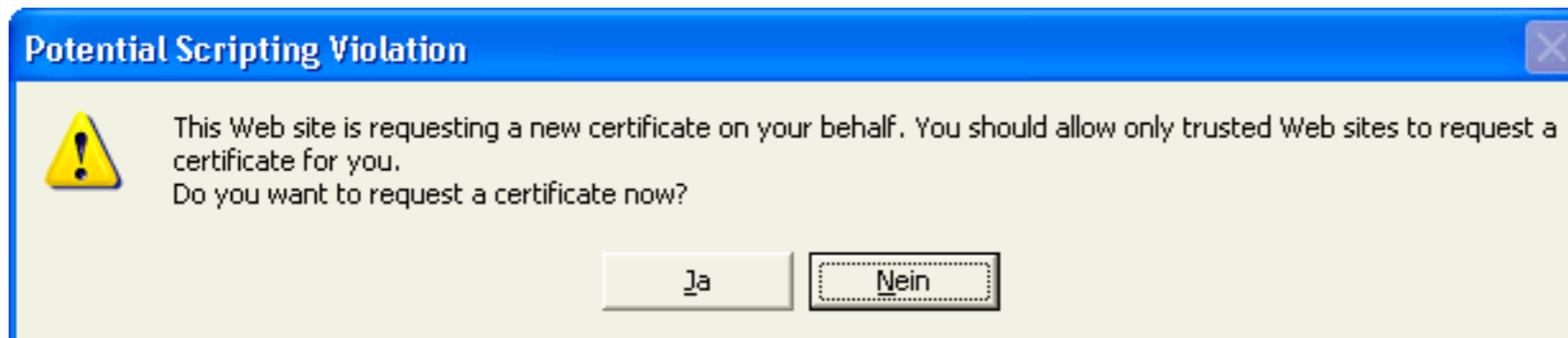


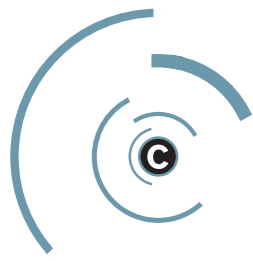


TLS client certificate user tracking

IE and Opera get it right (for once ...)

- Three steps: request, install, use
- You can generate PKCS#10 certificate requests on IE pretty easily, but ...

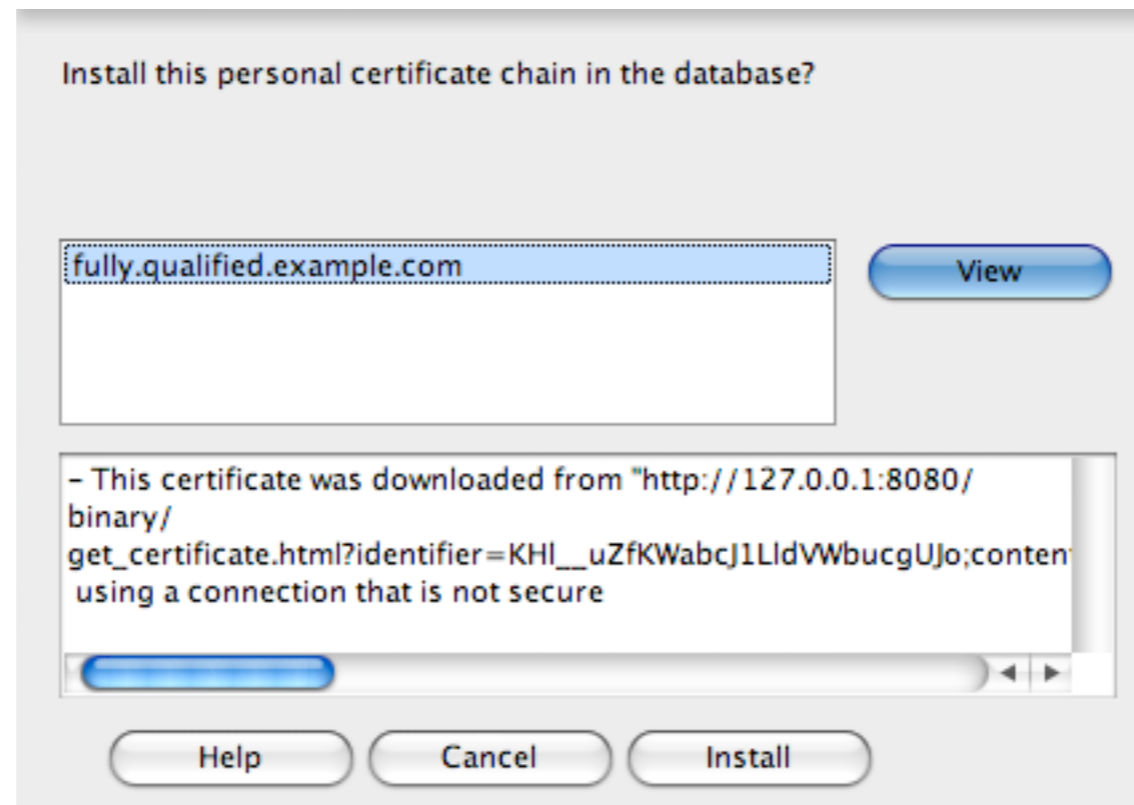


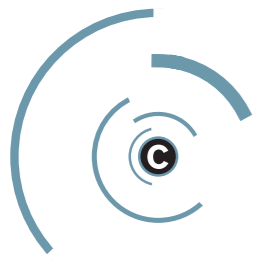


TLS client certificate user tracking

IE and Opera get it right (for once ...)

- Firefox, Opera, Safari, ... use SPKAC
- Opera silently (except for master password input) generates the request, but asks at installation:

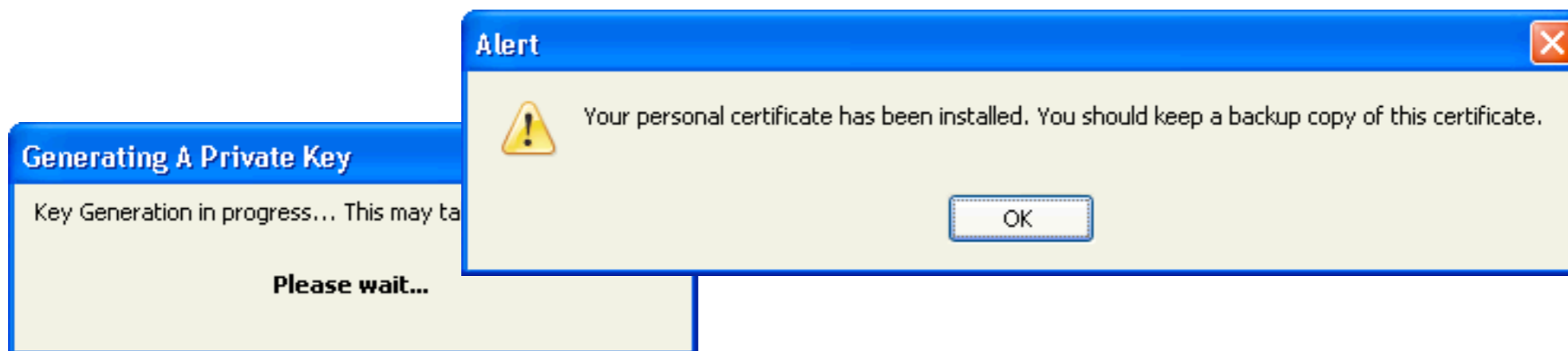


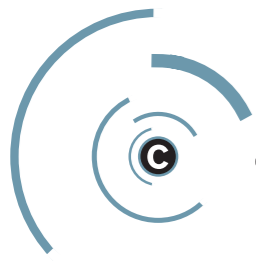


TLS client certificate user tracking

Firefox got it wrong (until recently)

- Key generation just pops up a small dialog which disappears really fast on modern machines
- Installation on Firefox 1.5 is completely silent, 2.x tells the user to make a backup of his certificate ...

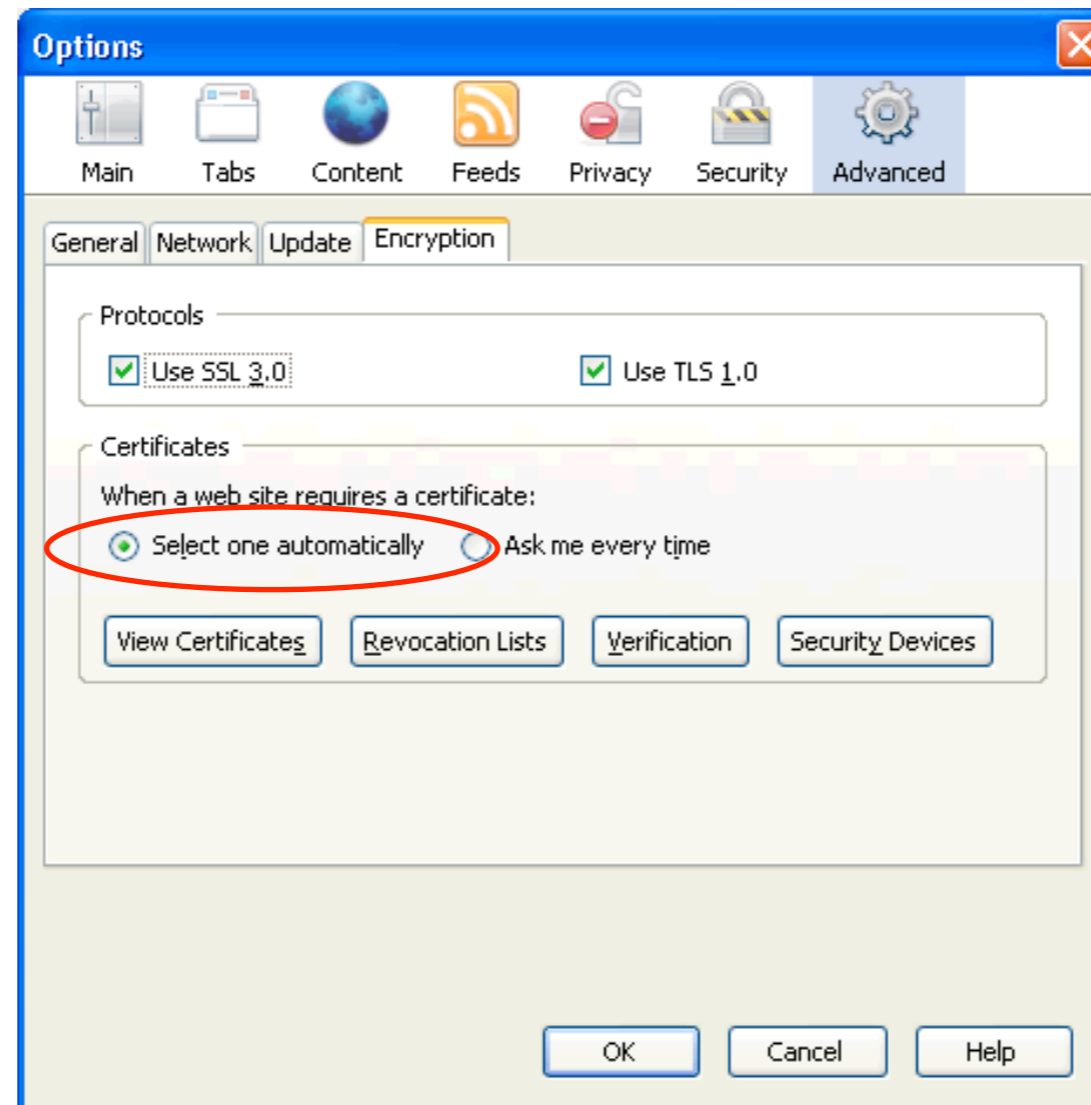




TLS client certificate user tracking

Firefox – the real problem (aka CVE-2007-4879)

- This is suboptimal, but the real problem was here:





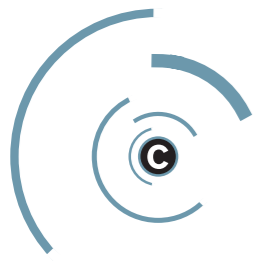
TLS client certificate user tracking

Still works in Safari on Mac OS X

The screenshot shows a Firefox browser window titled "Firefox 2.0.x TLS client certificate tracking POC" with the URL "http://0x90.eu/ff_tls_poc.html". The page content includes a form for a "username" (restricted to alphanumeric characters) with the value "safari" and an "Install certificate" button. Below the form, text instructs the user to visit the "ApacheSSL certificate export" website after installation. In the background, a macOS "Downloads" window shows a file named "issue_cert.cgi" (0.6 KB). In the foreground, the "Schlüsselbundverwaltung" (Keychain) window is open, displaying details for an "Automatisches Ausfüllen" (Automatic Fill) entry under the "Anmeldung" (Login) keychain. The entry details include "Art: Programmkennwort", "Account: Safari", and "Ort: PersonalFormsAutoFillDatabase". A table below lists four objects in the keychain:

Name	Art	Geändert	Verfällt	Schlüsselbund
Automatisches Ausfüllen	Programmkennwort	Heute, 17:14	--	Anmeldung
safari	Zertifikat	--	19.01.2038 00:...	Anmeldung
Schlüssel von 0x90.eu	Öffentlicher Schlüssel	--	--	Anmeldung
Schlüssel von 0x90.eu	Privater Schlüssel	--	--	Anmeldung



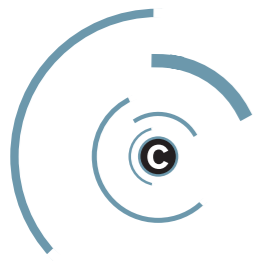


TLS client certificate user tracking

See for yourself

- Proof of concept available at
- http://0x90.eu/ff_tls_poc.html





Missing hostname binding

subjectAlternativeNames considered harmful?

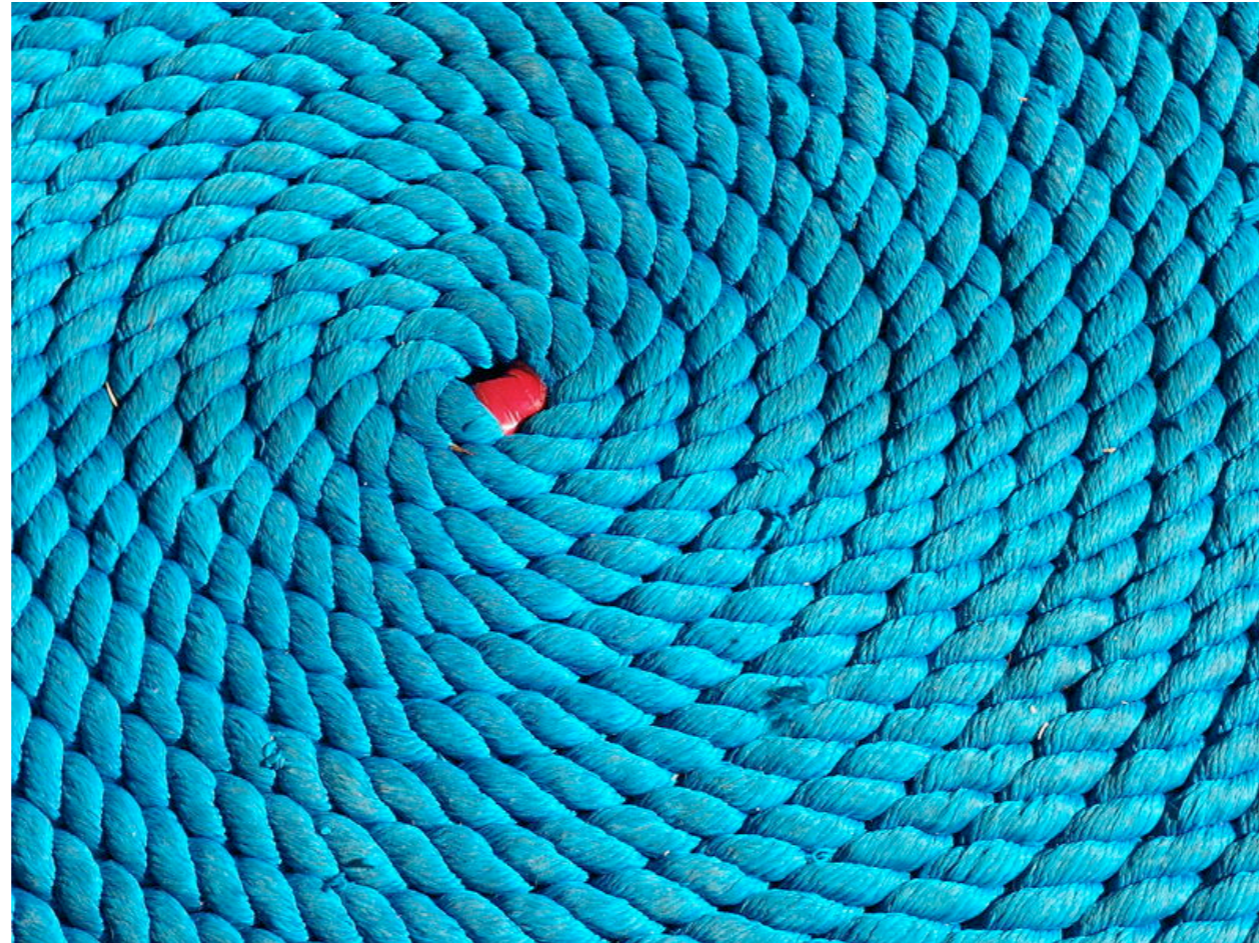


Image: Nevit Dilmen (GNU FDL)

- For a live demo, please go to <https://eusecwest.klink.name> – now!

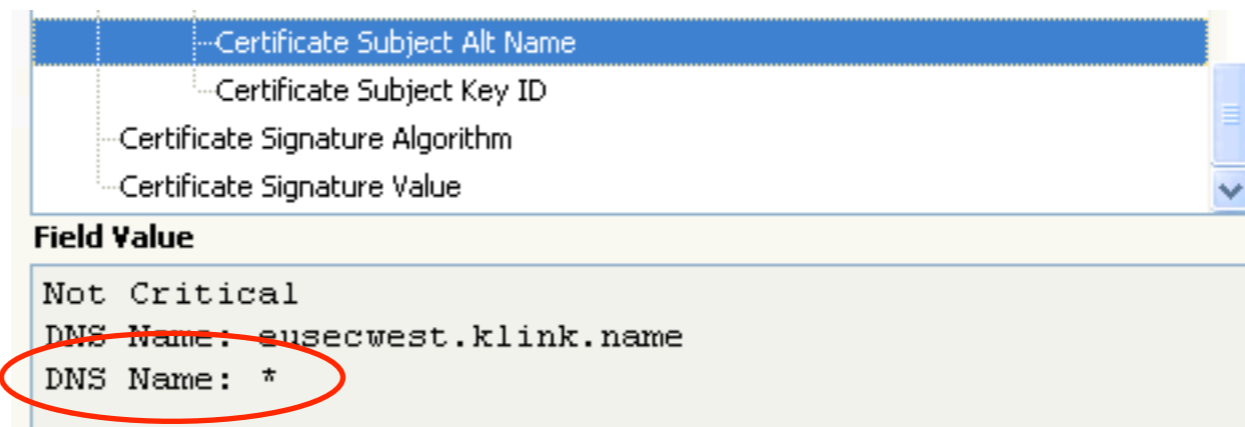


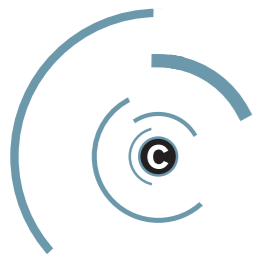


Missing hostname binding

Anybody noticed something suspicious?

- Something suspicious going on?
- OK, an untrusted certificate, but we just want to view the content, right?
- Anybody noticed the subjectAlternativeName?



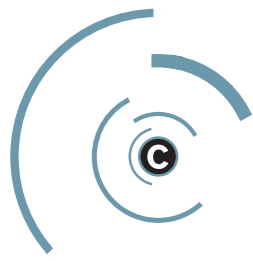


Missing hostname binding

... waiting for Server Name Indication support

- The features:
 - Wildcard matching for lazy sysadmins
 - Accepting untrusted certificates temporarily
- The bugs:
 - subjectAltNames are not shown
 - Wildcard '*' matches anything – '*.com' a lot
 - Trust decision is not bound to the DNS name



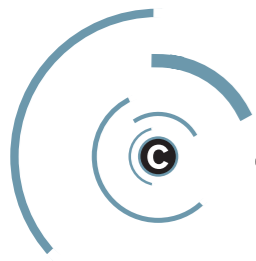


Missing hostname binding

... has been broken for years in some browsers

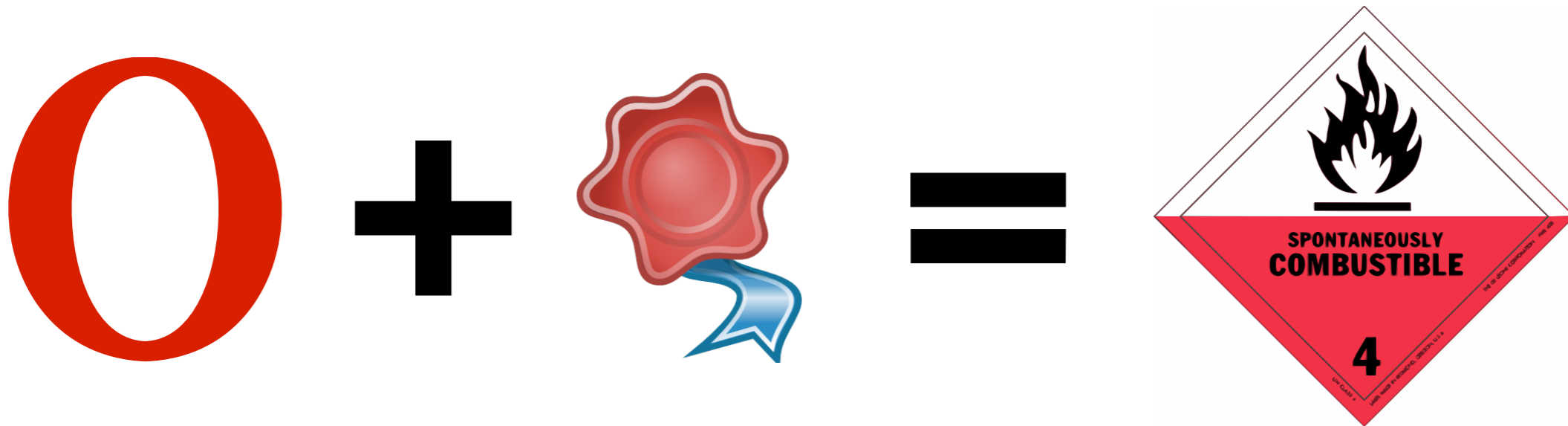
- Yes, that means you now possibly trust me for anything (for this session) ...
- Pretty useful for MITM attacks ...
 - Nils Toedtmann's TODO list:
 - Find a braindead major-browser-accredited CA which signs my certificate request with hidden TLD-wildcard subjectAltName. Take over the internet.
- This has been reported to Mozilla in 2004(!)
- recently raised again by Nils Toedtmann
- Test your browser at <http://test.eonis.net>





Certificate data is untrusted, too

... repeat after me: it is user input



- **The feature:** X.509 certificates can have subjectAlternativeNames of arbitrary length
- **The bug:** Assuming that they are of fixed length – to quote Ilja: “The 90’s called, they want their bugs back :-P”





Certificate data is untrusted, too a heap buffer overflow in Opera (CVE-2007-6521)

CPU - main thread, module Opera_1

Address	Hex dump	Disassembly	Comment
67CDC9A9	FF53 14	CALL DWORD PTR DS:[EBX+14]	
67CDC9AC	57	PUSH EDI	
67CDC9AD	8D85 78FDFFFF	LEA EAX, DWORD PTR SS:[EBP-288]	
67CDC9B3	50	PUSH EAX	
67CDC9B4	68 DCF6F667	PUSH Opera_1.67F6FADC	UNICODE "%s %s"
67CDC9B9	FF75 08	PUSH DWORD PTR SS:[EBP+8]	
67CDC9BC	E8 0A82EFFF	CALL Opera_1.67BD48CB	
67CDC9C1	83C4 10	ADD ESP, 10	
67CDC9C4	837D FC 00	CMP DWORD PTR SS:[EBP-4], 0	
67CDC9C8	74 6A	JE SHORT Opera_1.67C9A34	
67CDC9CA	8B0D E01F967	MOV ECX, DWORD PTR DS:[67F901E0]	
67CDC9D0	6A 01	PUSH 1	
67CDC9D2	FF75 14	PUSH DWORD PTR SS:[EBP+14]	
67CDC9D5	8D95 38FDFFFF	LEA EDX, DWORD PTR SS:[EBP-2C8]	
67CDC9DB	8B01	MOV EAX, DWORD PTR DS:[ECX]	
67CDC9DD	FF75 10	PUSH DWORD PTR SS:[EBP+10]	
67CDC9E0	6A 1F	PUSH 1F	
67CDC9E2	52	PUSH EDX	
67CDC9E3	FF50 0C	CALL DWORD PTR DS:[EAX+C]	
67CDC9E6	8D85 38FDFFFF	LEA EAX, DWORD PTR SS:[EBP-2C8]	
67CDC9EC	68 2C39F967	PUSH Opera_1.67F9392C	UNICODE "bytes"
67CDC9F1	50	PUSH EAX	

Registers (FPU)

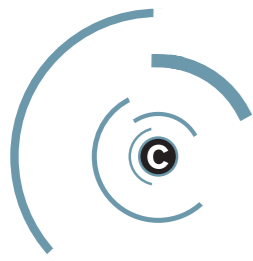
EAX	0012EBA8	
EAX	00A6A190	UNICODE "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
ECX	00000000	
EDX	00000000	
EBX	00010061	
ESP	002EB4C	
EBP	001EE30	
ESI	00A6A190	UNICODE "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
EDI	67F93924	Opera_1.67F93924
EIP	67CDC9A9	Opera_1.67CDC9A9

Address violation: DS:[00610075]=???

Address	Stack	Procedure / arguments	Called from
0012EE34	67DEFDFC	? Opera_1.67CDC4B3	Opera_1.67DEF
0012F000	67DEF74C	Opera_1.67DEF957	Opera_1.67DEF
0012F018	67AB8121	Includes Opera_1.67DEF74C	Opera_1.67AB8
0012F024	67AB7502	Includes Opera_1.67AB8121	Opera_1.67AB7
0012F04C	67AB69F0	? Opera_1.67AB746F	Opera_1.67AB6
0012F0B8	67AB7600	Opera_1.67AB6882	Opera_1.67AB7
0012F0D8	67AB7F06	Opera_1.67AB75B8	Opera_1.67AB7
0012F0E0	67E61A7E	Opera_1.67AB7EEE	Opera_1.67E61
0012F0EC	67E6186A	Opera_1.67E61A59	Opera_1.67E61
0012F0F8	7E368734	Includes Opera_1.67E6186A	user32.7E368
0012F124	7E368816	? user32.7E36870C	user32.7E368
0012F18C	7E3689CD	? user32.7E36875F	user32.7E368
0012F1EC	7E368A10	? user32.7E3688F1	user32.7E368
0012F1FC	67E618F8	? Opera_1.DispatchMessageW	Opera_1.67E61
0012F200	0012F210	pMsg = WM_USER hw = 3007F	
0012F230	67E6A5F9	Opera_1.67E61870	Opera_1.67E6A
0012F2DC	67E18111	Includes Opera_1.67E6A5F9	Opera_1.67E18
0012F334	67D8700E	Opera_1.67E17D18	Opera_1.67D87
0012F35C	67D86294	Opera_1.67D86FA9	Opera_1.67D86
0012F400	67D7C22A	Opera_1.67D85A2B	Opera_1.67D7C
0012F454	67D54733	Opera_1.67D7C104	Opera_1.67D54

Access violation when reading [00610075] - use Shift+F7/F8/F9 to pass exception to program





Certificate data is untrusted, too

release cycles for mobile software suck

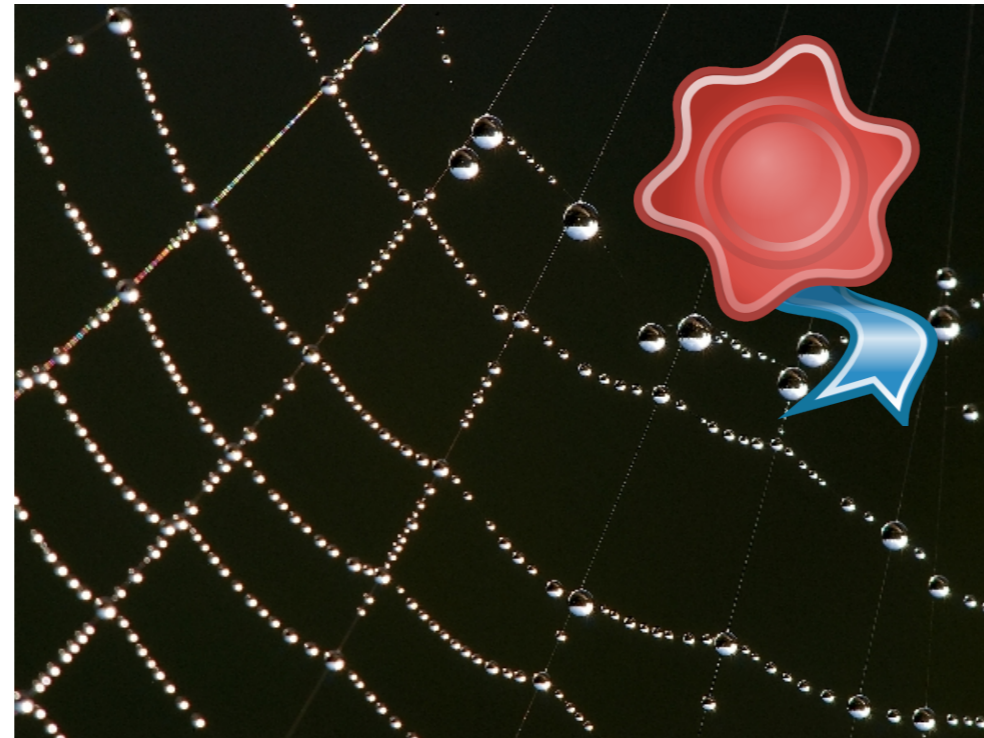
- Reported to Opera on October 5th, 2007
- Fixed on desktop with Opera 9.25 (December 19th, 2008)
- but Opera also has browsers for mobile devices (Opera Mini, Mobile, Devices ...)
- fixed for that (wouldn't tell me for what exactly) nowish





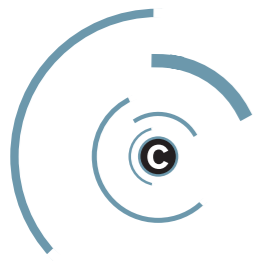
HTTP over X.509

triggering HTTP requests using X.509 extensions



- **The feature:** issuer certificate URIs can be specified within a certificate
- **The bug:** Automatically fetching those “certificates” from any location specified



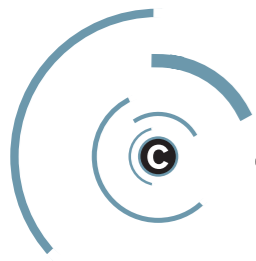


HTTP over X.509

URIs in X.509 extensions

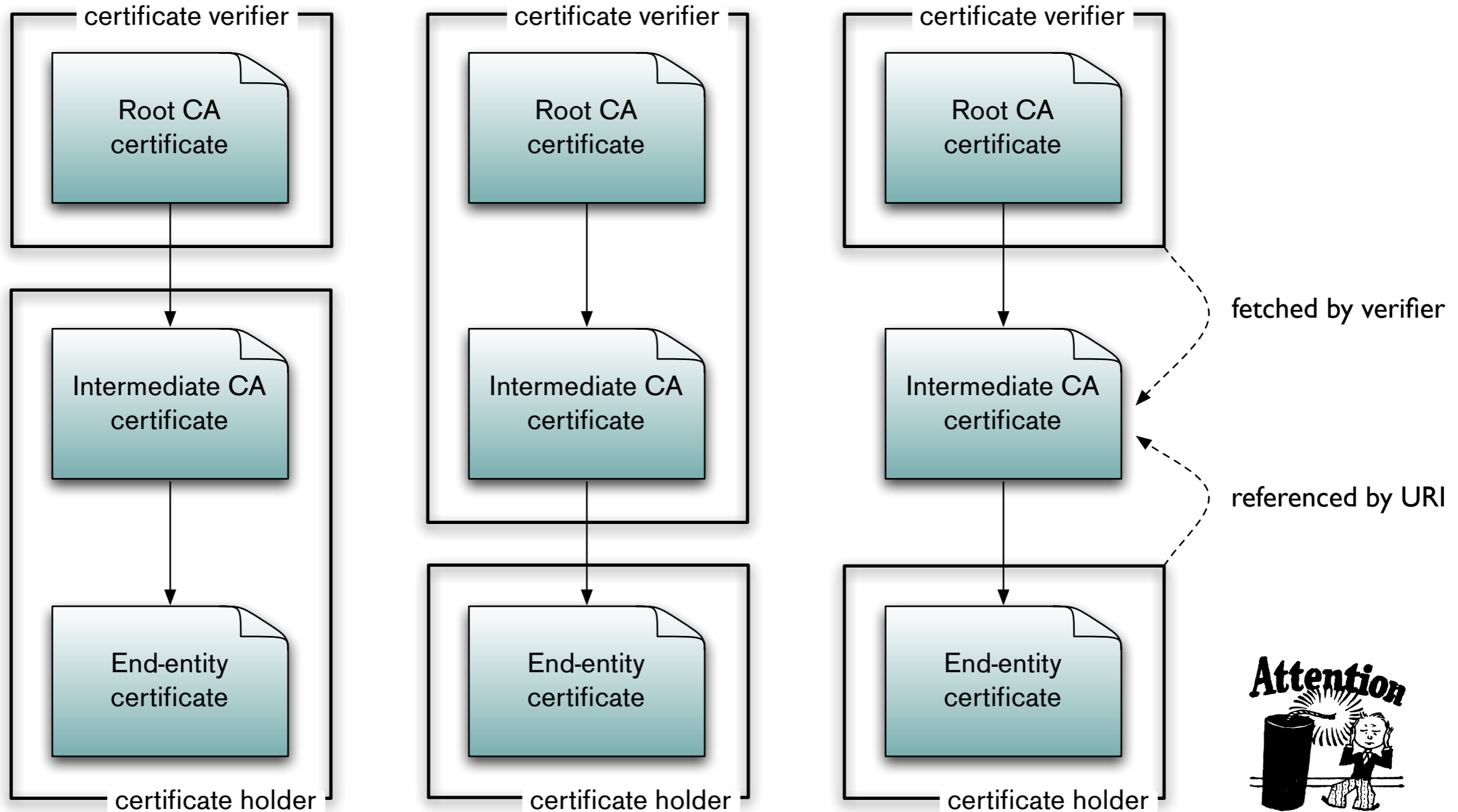
- There are URIs in quite a number of certificate extensions
 - CRL Distribution Point
 - CPS Pointer
 - OCSP server
- authorityInformationAccess caIssuers extension

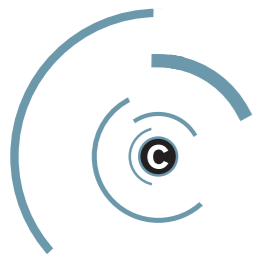




HTTP over X.509

Handling intermediate CA certificates



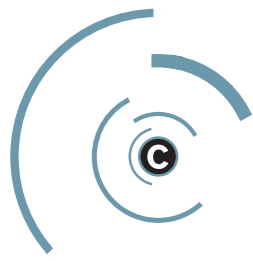


HTTP over X.509

Microsoft actually implements the RFC :-)

- Yes, this URI is completely attacker controlled
- This is actually a bug in RFC 3280
- It has not been widely implemented though
- Actually, the only productive implementation I know of is in Microsoft's CryptoAPI
- Known to be vulnerable to this issue:
 - Microsoft Outlook, Windows (Live) Mail
 - Office 2007



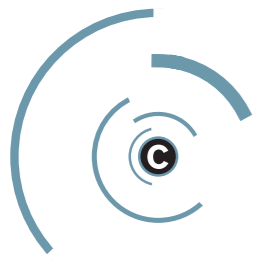


HTTP over X.509

Impact

- For S/MIME, this means:
 - spam filter testing
 - read receipt + IP geolocation
- For Office 2007 documents:
 - read receipt (when and how often)
- Generally:
 - the ability to access any host reachable from the client/server (blind)



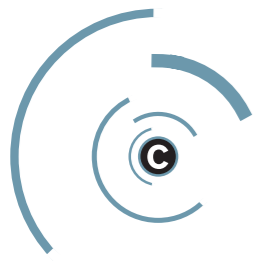


HTTP over X.509

Other potential vectors

- Other S/MIME clients / gateways
- IPSec (tests were negative on Windows 2003 Server and Cisco ASA 5540 running IOS 7.2.3)
- TLS client certificates (IIS seems not to be vulnerable)
- EAP/TLS
- Smartcard logon (not under Windows, apparently)
- ...



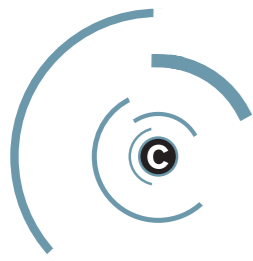


HTTP over X.509

HTTP in related standards

- Client Certificate URL extension (RFC 3546)
- OCSP ServiceLocator extension (RFC 2560)
- Logotypes in X.509 certificates (RFC 3709)
- Qualified Certificates Profile – Biometric Information Extension (RFC 3739)



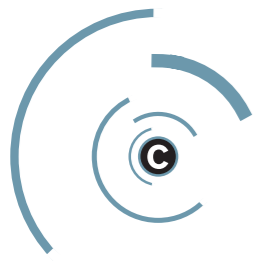


HTTP over X.509

Mitigation & Proof of Concepts

- Mitigation
 - Wait for Microsoft to fix it ... – no idea when
 - Configure (application level) firewalls, proxies accordingly / incorrectly
- Proof of concepts
 - `smime-http@klink.name`
 - `http://www.klink.name/security/HTTP_over_Office_2007_PoC.docx`





Debian & OpenSSL

No need to break it if the vendor already broke it ...

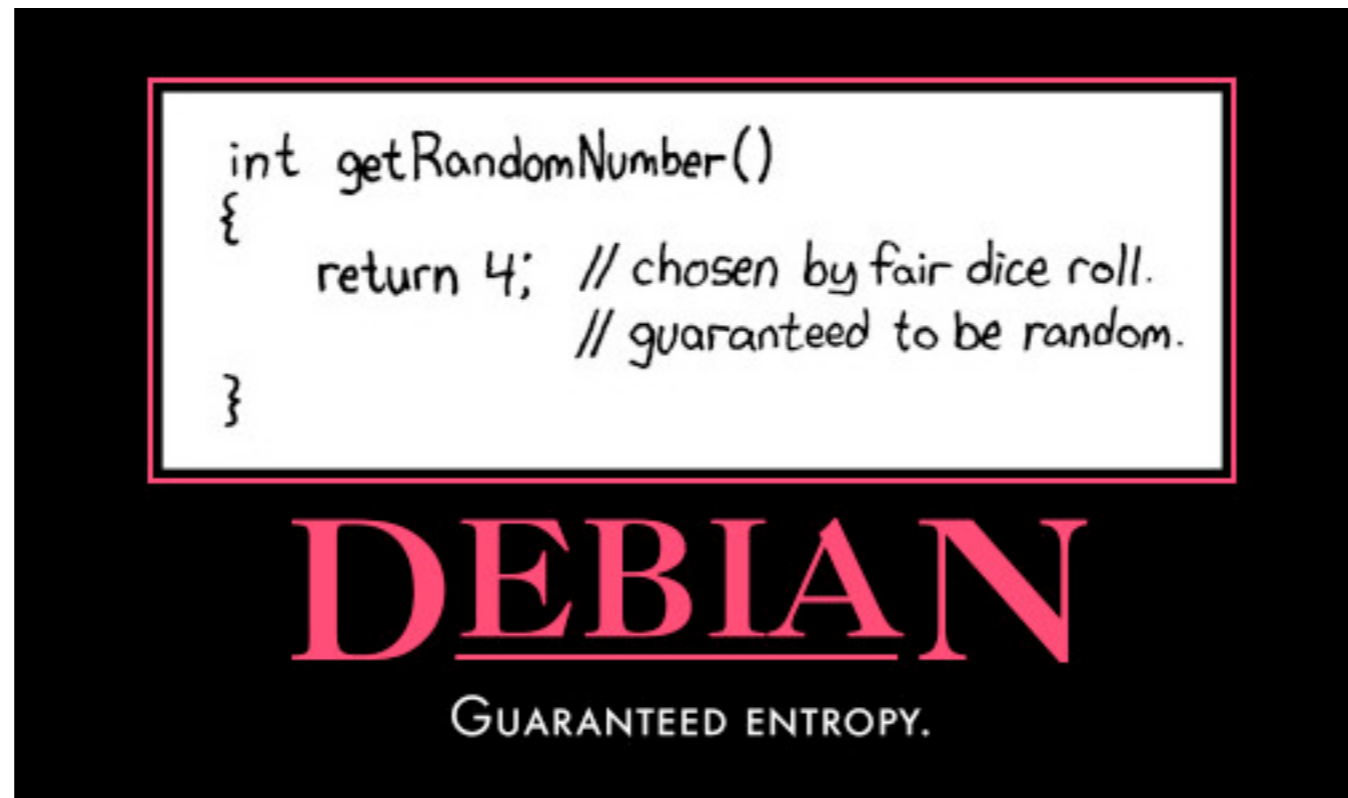
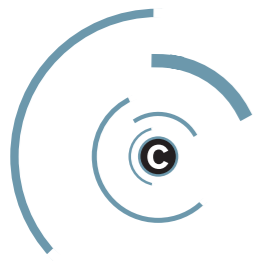


Image: H D Moore (license unknown), xkcd.com (CC-BY-NC)

- would be worth a complete talk
- Luckily, no CAs affected so far. We did have the webserver key of a large german financial institution for a few hours, though ... :-)



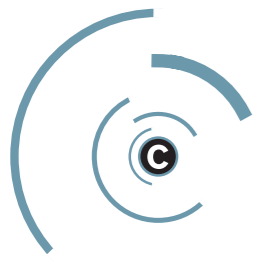


Debian & OpenSSL

Consider your passwords broken, too

- Everybody only talks about broken keys
- But: the Diffie-Hellmann key exchange uses random numbers too (g^x / g^y with x, y random)
- Did someone sniff your SSH traffic with either a compromised client or server (think “conference”)?
- Tool release: reads a PCAP file and tries to break the key exchange (client only for now)
- http://www.cynops.de/download/check_weak_dh_ssl.pl.bz2





Conclusions

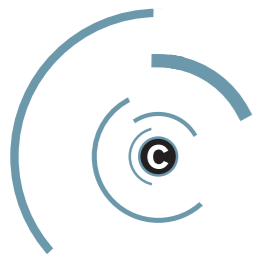
so, what do I do now?



Image: Miuet (GFDL)

- PKI and X.509 certificates are a valid technology to secure your applications and services
- Still, they are tricky in the details, consider them when implementing a solution



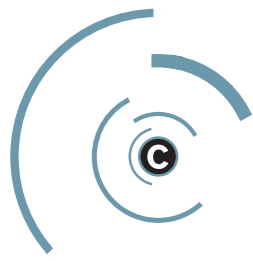


Thanks & Acknowledgements

Credit where credit is due

- FX of Phenoelit for helping with the Opera bug
- Jaromir Likavec, Alexander Opel and Alexander Nouak at Fraunhofer Institute for Computer Graphics Research (IGD) for setting up a Windows & Cisco test infrastructure
- Nils Toedtman for his interesting research
- Ralf-Phillip Weinmann for finding out that the TLS Client tracking PoC works under Safari



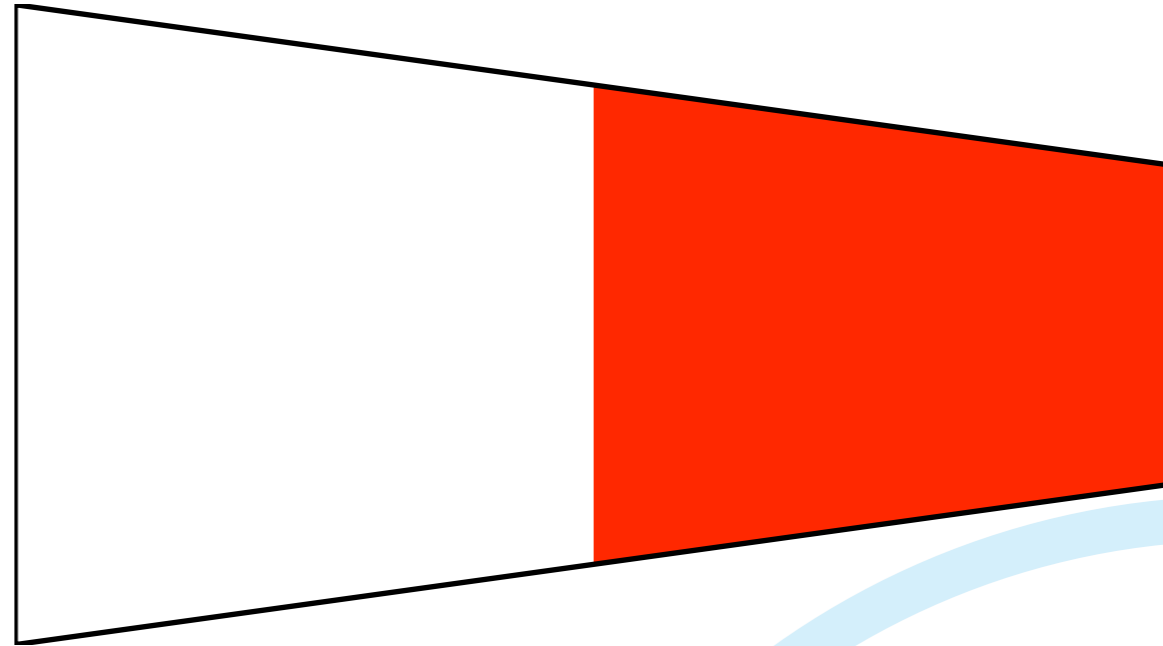
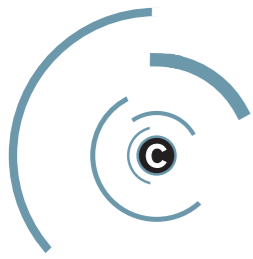


Further reading

all the gory details ...

- “Firefox 2.0.x: tracking unsuspecting users using TLS client certificates”, Alexander Klink, <http://permalink.gmane.org/gmane.comp.security.full-disclosure/55354>
- “Phishing for Confirmations. Certificate spoofing with subjectAltName and domain name wildcards”, Nils Toedtmann, <http://nils.toedtmann.net/pub/subjectAltName.txt>
- “Opera – heap-based buffer overflow”, Alexander Klink, <https://www.cynops.de/advisories/CVE-2007-6521.txt>
- “HTTP over X.509 – a whitepaper”, Alexander Klink, http://www.cynops.de/techzone/http_over_x509.html
- “PKI: It’s Not Dead, Just Resting”, Peter Gutmann, <http://www.cs.auckland.ac.nz/~pgut001/pubs/notdead.pdf>
- “Debian OpenSSL Predicatable PRNG Toys”, H D Moore, <http://metasploit.com/users/hdm/tools/debian-openssl/>





Q & A

