WHY JOHNNY THE DEVELOPER CAN’T WORK WITH PUBLIC KEY CERTIFICATES

Martin Ukrop
Usable security researcher
Centre for Research on Cryptography and Security, Masaryk University
Red Hat research cooperation

Vashek Matyas
Head of the security laboratory
Centre for Research on Cryptography and Security, Masaryk University
File does not exist.

File /etc/ssl/certs/certificate1.pem does not exist.

Clicking OK will solve everything and create a secure solution with no bugs or vulnerabilities.
Annoying Security Control

**Something happened** and you need to click OK to get on with doing things.

Security error:
Permitted subtree violation

Certificate mismatch security identification administrator communication intercept liliputian snotweasel foxtrot omegafource.

Show more technical crap

Cancel  OK
Is it really that bad?

- Let’s find out!
- What is the most used tool for generating certificates?

Q: Have you ever used OpenSSL?

[xukrop@styx ~]$ openssl version
OpenSSL 1.1.0g-fips 2 Nov 2017
[xukrop@styx ~]$ openssl x509 -help
Usage: x509 [options]
Valid options are:
  -help           Display this summary
  -inform format  Input format - default PEM (one of DER, NET or PEM)
  -in infile      Input file - default stdin
  -outform format Output format - default PEM (one of DER, NET or PEM)
  -out outfile    Output file - default stdout
  -keyform PEM|DER Private key format - default PEM
  -passin val     Private key password/pass-phrase source
  -serial         Print serial number value
  -subject_hash   Print subject hash value
  -issuer_hash    Print issuer hash value
  -hash           Synonym for -subject_hash
  -subject        Print subject DN
  -issuer         Print issuer DN
  -email          Print email address(es)
  -startdate      Set notBefore field
Empirical experiment in usability

- 87 participants (👨) of DEVCONF.cz
  (developer conference by Red Hat Czech)

Task:

You are a software tester.
Use command line OpenSSL (v1.0.2g)

1. ... to issue a self-signed certificate.
2. ... to validate 4 given certificates.
“Interacting with OpenSSL voluntarily?”

“Sorry, not even for research.”
“We all know OpenSSL sucks.”

“But finally, there is someone collecting empirical evidence.”
Task success

**Task 1** (certificate generation)

- Succeeded: 46% (39/87)
- Did not succeed: 10% (8/87)
- Thought that they succeeded but did not: 44% (37/87)

**Task 2** (certificate validation)

- Success: 19% (14/72)
- Incorrect: 10% (7/72)
- Ignoring OS cert store: 71% (51/72)
Created certificates

- 4096-bit key: 20% (16/82)
- 2048-bit key: 42% (34/82)
- 1024-bit key: 38% (30/82)

- Organization = “Internet Widgits Pty Ltd.”: 42% certificates (27/65)
- cca 260 000 such certificates online (Censys.io dataset 2018-02-28)
Used resources

Task 1 (generating certificates)
- Online only: 32
- Online & manual page: 47
- Local manual page only: 6
- CLI help only: 2

Task 2 (validating certificates)
- Online only: 25
- Online & manual page: 33
- Local manual page only: 10
- CLI help only: 4
How to create a self-signed certificate with openssl?

I'm adding https support to an embedded linux device. I have tried to generate a self-signed certificate with these steps:

```
openssl req -new > cert.csr
openssl rsa -in privkey.pem -out key.pem
openssl x509 -in cert.csr -out cert.pem -req -signkey key.pem -days 1001
cat key.pem>>cert.pem
```

This works, but I get some errors with, for example, google chrome:

This is probably not the site you are looking for!
The site's security certificate is not trusted!

Am I missing something? Is this the correct way to build a self-signed certificate?
WISC knowledge base: 40% people

Verifying that a Certificate is issued by a CA

How to use OpenSSL on the command line to verify that a certificate was issued by a specific CA, given that CA's certificate:

```bash
$ openssl verify -verbose -CAfile cacert.pem server.crt
server.crt: OK
```

If you get any other message, the certificate was not issued by that CA.

See Also:

- How to turn a X509 Certificate in to a Certificate Signing Request
- Verifying that a Private Key Matches a Certificate
Web pages used

- Describing security implications: 23% web pages
- Explaining individual parameters: 27% web pages
- Changes after copy-paste: 9% people (8/87 😊)
Q: How to display manual page for this?

`openssl verify -CAfile ca.pem cert.pem`

1. `man openssl`  ??%  (/?/53)
2. `man openssl verify`  28%  (15/53)
3. `man openssl.verify`  2%  (1/53)
4. `man openssl-verify`  8%  (4/53)
5. `man verify`  100%  (53/53)
OPENSSL manual page

NAME
openssl - OpenSSL command line tool

SYNOPSIS
openssl command [ command_opts ] [ command_args ]
openssl list [ standard-commands | digest-commands | cipher-commands | cipher-algorithms | digest-algorithms | public-key-algorithms]
openssl no-XXX [ arbitrary options ]

DESCRIPTION
OpenSSL is a cryptography toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) network protocols and related cryptography standards required by them.

The openssl program is a command line tool for using the various cryptography functions of OpenSSL's crypto library from the shell. It can be used for...
Has the world moved on?

Our work

- `man openssl verify` now works
- Fixed URLs for online documentation
- Research into better error messages

OpenSSL team

- High-level `help` command
- `-help` argument for each command
- Improved defaults (key lengths, ...)

18
OpenSSL usability is poor.

(But better than other tools.)

Improvement is possible!
People may not know they failed if the tool does not tell them.
Takeaways III.

Documentation (manuals, tutorials, Q&A forums, ...) matters a lot.

Stack Overflow is a seriously used resource.
What should you do next?

**USING security products?**

- Ask your developers what they find unusable.
- Investigate past vulnerabilities: Caused by tool unusability?
- Report usability issues back to developers.

**DEVELOPING security products?**

- In your project, strive for good “developer experience” (DX).
- Ask users on usability feedback.
- Organize a usability lab study to improve your product.
CARE FOR YOUR DEVELOPERS.
DEVELOPER EXPERIENCE (DX) MATTERS.

Martin Ukrop
mukrop@mail.muni.cz

Centre for Research on Cryptography and Security
Masaryk University, Brno, Czech Republic