Testing SSL

Dirk Wetter (d0rk)

@drwetter

0. whoami

- **Independent security consultant**
  - pentests / defense+hardening / concepts / training / PM
  - historical strong unix-/networking background
    - programming: past (well, ...)

- **Community involvements**
  - OWASP
  - GUUG

https://drwetter.eu/

- **HowTo do that?**
  - Different tools available
    - Based on Python (sslyze), PHP+Python (ssl-decoder), Perl (o-saft), scripted (cipherscan), SSLlabs (Go), …
    - Coverage: Nmap+LUA, Java (TestSSLServer), Windows EXE (SSLAudit)
  - Some Open Source, some not
  - Privacy
  - Platform availability

- **testssl.sh: what is that?**
  - Blunt:
    - Check's any server's SSL/TLS encryption
  - Cool thing:
    - Plain `/bin/bash` + `openssl` as helper
    - + standard Unix tools, no perl/python etc.
  - compatible:
    - Linux
    - Mac OS X
    - (Free)BSD
    - Windows: MSYS2, Cygwin


```
dirks@laptop:~/testssl|0% ./testssl.sh dev.testssl.sh

###########################################################
    testssl.sh       2.7dev from https://testssl.sh/dev/
    (1.430 2015/12/24 22:00:21)

      This program is free software. Distribution and
             modification under GPLv2 permitted.
      USAGE w/o ANY WARRANTY. USE IT AT YOUR OWN RISK!

       Please file bugs @ https://testssl.sh/bugs/

###########################################################

 Using "OpenSSL 1.0.2-chacha (1.0.2d-dev)" [~181 ciphers]
on trex:$PWD/bin/openssl.Linux.x86_64
 (built: "Jul  6 18:05:33 2015", platform: "linux-x86_64")


 Start 2015-12-26 11:36:32    -->> 81.169.199.25:443 (dev.testssl.sh) <<--

 further IP addresses:   2a01:238:4279:1200:1000:1:e571:51
 rDNS (81.169.199.25):   testssl.sh.
 Service detected:       HTTP


 Testing protocols (via sockets except TLS 1.2 and SPDY/HTTP2)

 SSLv2      not offered (OK)
 SSLv3      not offered (OK)
 TLS 1      offered
 TLS 1.1    offered
 TLS 1.2    offered (OK)
 SPDY/NPN   http/1.1 (advertised)
 HTTP2/ALPN not offered
```

```
 Testing ~standard cipher lists

Null Ciphers                 not offered (OK)
Anonymous NULL Ciphers       offered (NOT ok)
Anonymous DH Ciphers         offered (NOT ok)
40 Bit encryption            offered (NOT ok)
56 Bit encryption            not offered (OK)
Export Ciphers (general)     offered (NOT ok)
Low (<=64 Bit)               not offered (OK)
DES Ciphers                  not offered (OK)
Medium grade encryption      offered (NOT ok)
Triple DES Ciphers           offered (NOT ok)
High grade encryption        not offered (NOT ok)


 Testing (perfect) forward secrecy, (P)FS -- omitting 3DES, RC4 and Null Encryption here

PFS is offered (OK)  DHE-RSA-SEED-SHA


 Testing server preferences

Has server cipher order?     nope (NOT ok)
Negotiated protocol          TLSv1.2
Negotiated cipher            DHE-RSA-SEED-SHA, 999 bit DH (limited sense as client will pick)
Negotiated cipher per proto  (limited sense as client will pick)
    DHE-RSA-SEED-SHA:              TLSv1, TLSv1.1, TLSv1.2
    ECDHE-ECDSA-AES256-GCM-SHA384: http/1.1
No further cipher order check has been done as order is determined by the client
```



```
Testing server defaults (Server Hello)

TLS server extensions (std)  "server name" "renegotiation info" "session ticket" "heartbeat" "next protocol"
Session Tickets RFC 5077     300 seconds (PFS requires session ticket keys to be rotated <= daily)
SSL Session ID support       yes
Server key size              4096 bit
Signature Algorithm          SHA256 with RSA
Fingerprint / Serial         SHA1 AA5FF6B618DB64D962505B4B22F65C21A3560E7F / 053F29F0E45CA1
                             SHA256 FDAB2063E38C2165A0B7471F15D86540CFCDF0D4C5EB2A67F474B2773CDB64C8
Common Name (CN)             "dev.testssl.sh" (CN in response to request w/o SNI: "default.name")
subjectAltName (SAN)         "dev.testssl.sh" "testssl.sh"
Issuer                       "StartCom Class 1 Primary Intermediate Server CA" ("StartCom Ltd." from "IL")
EV cert (experimental)       no
Certificate Expiration       expires < 60 days (56) (2015-02-20 07:51 --> 2016-02-20 20:06 +0100)
# of certificates provided   2
Chain of trust (experim.)    Ok
Certificate Revocation List  http://crl.startssl.com/crt1-crl.crl
OCSP URI                     http://ocsp.startssl.com/sub/class1/server/ca
OCSP stapling                not offered
TLS timestamp                random values, no fingerprinting possible


Testing HTTP header response @ "/"

HTTP Status Code             302 Moved Temporarily, redirecting to "https://github.com/drwetter/testssl.sh/"
HTTP clock skew              -3 sec from localtime
IPv4 address in header       IPv4-test: 10.35.33.7
                             (check if it's your IP address or e.g. a cluster IP)
Strict Transport Security    1169 days=101010101 s, includeSubDomains
Public Key Pinning           --
Server banner                Apache 1.3.37 (Idefix)
Application banner           X-Powered-By: PHP/4.4.42
                             X-Version: seems deliberately borken
Cookie(s)                    2 issued: NONE secure, NONE HttpOnly
Security headers             X-FRAME-OPTIONS: DENY
Reverse Proxy banner         --
```


Testing vulnerabilities

Heartbleed (CVE-2014-0160)	not vulnerable (OK) (timed out)
CCS (CVE-2014-0224)	not vulnerable (OK)
Secure Renegotiation (CVE-2009-3555)	not vulnerable (OK)
Secure Client-Initiated Renegotiation	not vulnerable (OK)
CRIME, TLS (CVE-2012-4929)	not vulnerable (OK)
BREACH (CVE-2013-3587)	no HTTP compression (OK) - only supplied "/" tested
POODLE, SSL (CVE-2014-3566)	not vulnerable (OK)
TLS_FALLBACK_SCSV (RFC 7507), experim.	Downgrade attack prevention supported (OK)
FREAK (CVE-2015-0204)	VULNERABLE (NOT ok), uses EXPORT RSA ciphers
LOGJAM (CVE-2015-4000), experimental	not vulnerable (OK), common primes not checked. See below for any DH ciphers + bit siz
BEAST (CVE-2011-3389)	TLS1: EXP-RC2-CBC-MD5 EXP-DES-CBC-SHA DES-CBC3-SHA EDH-RSA-DES-CBC3-SHA ADH-DES-CBC3-SHA SEED-SHA DHE-RSA-SEED-SHA ADH-SEED-SHA ECDHE-RSA-DES-CBC3-SHA AECDH-DES-CBC3-SHA EXP-RC2-CBC-MD5
	VULNERABLE -- but also supports higher protocols (possible mitigation): TLSv1.1 TLSv1.
RC4 (CVE-2013-2566, CVE-2015-2808)	VULNERABLE (NOT ok): ECDHE-RSA-RC4-SHA AECDH-RC4-SHA ADH-RC4-MD5 RC4-SHA RC4-MD5 RC4-M
EXP-RC4-MD5 EXP-RC4-MD5	

Testing all 181 locally available ciphers against the server, ordered by encryption strength

Hexcode	Cipher Suite Name (OpenSSL)	KeyExch.	Encryption	Bits	Cipher Suite Name (RFC)
x9a	DHE-RSA-SEED-SHA	DH 999	SEED	128	TLS_DHE_RSA_WITH_SEED_CBC_SHA
x9b	ADH-SEED-SHA	DH 999	SEED	128	TLS_DH_anon_WITH_SEED_CBC_SHA
x96	SEED-SHA	RSA	SEED	128	TLS_RSA_WITH_SEED_CBC_SHA
xc011	ECDHE-RSA-RC4-SHA	ECDH 256	RC4	128	TLS_ECDHE_RSA_WITH_RC4_128_SHA
xc016	AECDH-RC4-SHA	ECDH 256	RC4	128	TLS_ECDH_anon_WITH_RC4_128_SHA
x18	ADH-RC4-MD5	DH 999	RC4	128	TLS_DH_anon_WITH_RC4_128_MD5
x05	RC4-SHA	RSA	RC4	128	TLS_RSA_WITH_RC4_128_SHA
x04	RC4-MD5	RSA	RC4	128	TLS_RSA_WITH_RC4_128_MD5
x010080	RC4-MD5	RSA	RC4	128	SSL_CK_RC4_128_WITH_MD5
xc012	ECDHE-RSA-DES-CBC3-SHA	ECDH 256	3DES	168	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
x16	EDH-RSA-DES-CBC3-SHA	DH 999	3DES	168	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
xc017	AECDH-DES-CBC3-SHA	ECDH 256	3DES	168	TLS_ECDH_anon_WITH_3DES_EDE_CBC_SHA
x1b	ADH-DES-CBC3-SHA	DH 999	3DES	168	TLS_DH_anon_WITH_3DES_EDE_CBC_SHA
x0a	DES-CBC3-SHA	RSA	3DES	168	TLS_RSA_WITH_3DES_EDE_CBC_SHA
x08	EXP-DES-CBC-SHA	RSA(512)	DES	40,export	TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
x06	EXP-RC2-CBC-MD5	RSA(512)	RC2	40,export	TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
x040080	EXP-RC2-CBC-MD5	RSA(512)	RC2	40,export	SSL_CK_RC2_128_CBC_EXPORT40_WITH_MD5
x03	EXP-RC4-MD5	RSA(512)	RC4	40,export	TLS_RSA_EXPORT_WITH_RC4_40_MD5
x020080	EXP-RC4-MD5	RSA(512)	RC4	40,export	SSL_CK_RC4_128_EXPORT40_WITH_MD5

Done 2015-12-26 11:55:44 -->> 81.169.199.25:443 (dev.testssl.sh) <<--

- **testssl.sh**
  - customized runs, see `--help`

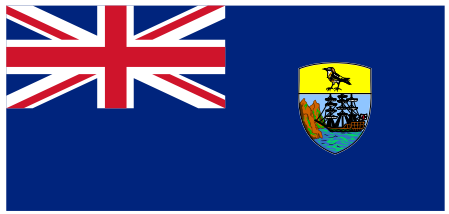

```
dirks@laptop:~|0% testssl.sh

testssl.sh <options>

     -h, --help                    what you're looking at
     -b, --banner                  displays banner + version of testssl.sh
     -v, --version                 same as previous
     -V, --local                   pretty print all local ciphers
     -V, --local <pattern>         which local ciphers with <pattern> are available?
                                   (if pattern not a number: word match)

testssl.sh <options> URI    ("testssl.sh URI" does everything except -E)

     -e, --each-cipher             checks each local cipher remotely
     -E, --cipher-per-proto        checks those per protocol
     -f, --ciphers                 checks common cipher suites
     -p, --protocols               checks TLS/SSL protocols (including SPDY/HTTP2)
     -y, --spdy, --npn             checks for SPDY/NPN
     -Y, --http2, --alpn           checks for HTTP2/ALPN
     -S, --server_defaults         displays the server's default picks and certificate info
     -P, --preference              displays the server's picks: protocol+cipher
     -x, --single-cipher <pattern> tests matched <pattern> of ciphers
                                   (if <pattern> not a number: word match)
     -U, --vulnerable              tests all vulnerabilities
     -B, --heartbleed              tests for heartbleed vulnerability
     -I, --ccs, --ccs-injection    tests for CCS injection vulnerability
     -R, --renegotiation           tests for renegotiation vulnerabilities
     -C, --compression, --crime    tests for CRIME vulnerability
     -T, --breach                  tests for BREACH vulnerability
     -O, --poodle                  tests for POODLE (SSL) vulnerability
     -Z, --tls-fallback            checks TLS_FALLBACK_SCSV mitigation
     -F, --freak                   tests for FREAK vulnerability
     -A, --beast                   tests for BEAST vulnerability
     -J, --logjam                  tests for LOGJAM vulnerability
     -s, --pfs, --fs, --nsa        checks (perfect) forward secrecy settings
     -4, --rc4, --appelbaum        which RC4 ciphers are being offered?
     -H, --header, --headers       tests HSTS, HPKP, server/app banner, security headers,
                                   cookie, reverse proxy, IPv4 address
```

```
  special invocations:

     -t, --starttls <protocol>     does a default run against a STARTTLS enabled <protocol>
     --xmpphost <to_domain>        for STARTTLS enabled XMPP it supplies the XML stream to-'' domain -- sometimes needed
     --mx <domain/host>            tests MX records from high to low priority (STARTTLS, port 25)
     --ip <ip>                     a) tests the supplied <ip> v4 or v6 address instead of resolving host(s) in URI
                                   b) arg "one" means: just test the first DNS returns (useful for multiple IPs)
     --file <fname>                mass testing option: Reads command lines from <fname>, one line per instance.
                                   Comments via # allowed, EOF signals end of <fname>. Implicitly turns on "--warnings batch"
partly mandatory parameters:

     URI                           host|host:port|URL|URL:port   (port 443 is assumed unless otherwise specified)
     pattern                       an ignore case word pattern of cipher hexcode or any other string in the name, kx or bits
     protocol                      is one of ftp,smtp,pop3,imap,xmpp,telnet,ldap (for the latter two you need e.g. the supplie
openssl)

tuning options (can also be preset via environment variables):
     --bugs                        enables the "-bugs" option of s_client, needed e.g. for some buggy F5s
     --assuming-http               if protocol check fails it assumes HTTP protocol and enforces HTTP checks
     --ssl-native                  fallback to checks with OpenSSL where sockets are normally used
     --openssl <PATH>              use this openssl binary (default: look in $PATH, $RUN_DIR of testssl.sh
     --proxy <host>:<port>         connect via the specified HTTP proxy
     -6                            use also IPv6 checks, works only with supporting OpenSSL version and IPv6 connectivity
     --sneaky                      leave less traces in target logs: user agent, referer
     --quiet                       don't output the banner. By doing this you acknowledge usage terms normally appearing in the
banner
     --log, --logging              logs stdout to <NODE-YYYYMMDD-HHMM.log> in current working directory
     --logfile <file>              logs stdout to <file/NODE-YYYYMMDD-HHMM.log> if file is a dir or to specified file
     --wide                        wide output for tests like RC4, BEAST. PFS also with hexcode, kx, strength, RFC name
     --show-each                   for wide outputs: display all ciphers tested -- not only succeeded ones
     --warnings <batch|off|false>  "batch" doesn't wait for keypress, "off" or "false" skips connection warning
     --color <0|1|2>               0: no escape or other codes,  1: b/w escape codes,  2: color (default)
     --debug <0-6>                 1: screen output normal but debug output in temp files.  2-6: see line ~120

All options requiring a value can also be called with '='  e.g. testssl.sh -t=smtp --wide --openssl=/usr/bin/openssl <URI>.
<URI> is always the last parameter.

Need HTML output? Just pipe through "aha" (Ansi HTML Adapter: github.com/theZiz/aha) like

   "testssl.sh <options> <URI> | aha >output.html"

dirks@laptop:~|0%
```

- **Batch processing**

```
dirks@laptop:/tmp|1% cat demo.txt
--wide facebook.com
--log -t smtp --wide gmail.com:25    # this comment will be ignored
# this one too
--mx facebook.com
# empty lines don't matter either

-H -p --ssl-native testssl.sh
-t pop3 -p --heartbleed --ccs pop.gmx.net:110
-t imap -E --ssl-native imap.gmx.net:143
--wide https://github.com/thecky/testssl.sh/commit/d5e7d14ea9e75f9337514a36afe6e53dbec40026
-t xmpp -p -nsa jabber.ccc.de:5222
-t xmpp -p --rc4 --xmpphost guug.de jabber.guug.de:5222

# full stop after next line
EOF

here you can write any BS

dirks@laptop:/tmp|0% testssl.sh --file demo.txt

###########################################################
    testssl.sh       2.7dev from https://testssl.sh/dev/
    (1.432 2015/12/27 13:51:17)
```


- **testssl.sh**
  - 2005: inhouse testing tool (pentests)
  - Open sourced: ~ 2010
    - 2/2014: domain testssl.sh
    - 4/2014: bitbucket
    - 10/2014: github
    - 3 releases in 2015
  - ~ 5500 LoC



  Flag: https://upload.wikimedia.org/wikipedia/commons/0/00/Flag_of_Saint_Helena.svg from „Patricia Fidi“. Licence public domain

- **testssl.sh**
  - Distributions:
    - Pentesting:
      - BackTrack, BlackArch Linux, Weakerthan Linux
      - hello Kali, anybody out there? ;-)
    - Regular:
      - ArchLinux
      - Debian testing
      - Ubuntu Xenial Xerus (=16.04 LTS)
  - More:
    - Docker images
    - Homebrew

2. Code

- **2015**
  - vulnerabilities

```
-U, --vulnerable                tests all vulnerabilities
-B, --heartbleed                tests for heartbleed vulnerability
-I, --ccs, --ccs-injection      tests for CCS injection vulnerability
-R, --renegotiation             tests for renegotiation vulnerabilities
-C, --compression, --crime      tests for CRIME vulnerability
-T, --breach                    tests for BREACH vulnerability
-O, --poodle                    tests for POODLE (SSL) vulnerability
-Z, --tls-fallback              checks TLS_FALLBACK_SCSV mitigation
-F, --freak                     tests for FREAK vulnerability
-A, --beast                     tests for BEAST vulnerability
-J, --logjam                    tests for LOGJAM vulnerability
-s, --pfs, --fs,--nsa           checks (perfect) forward secrecy settings
-4, --rc4, --appelbaum          which RC4 ciphers are being offered?
```


- **Problem using OpenSSL**
  - new Linux/BSD distributions, Mac OS X: „Fixes“
    - Null, Anonymous Ciphers
    - SSLv2
      - SSLv3 coming
    - export ciphers (FREAK)
    - weak DH ciphers (LOGJAM)
  - Missing new, advanced bits (supplier)

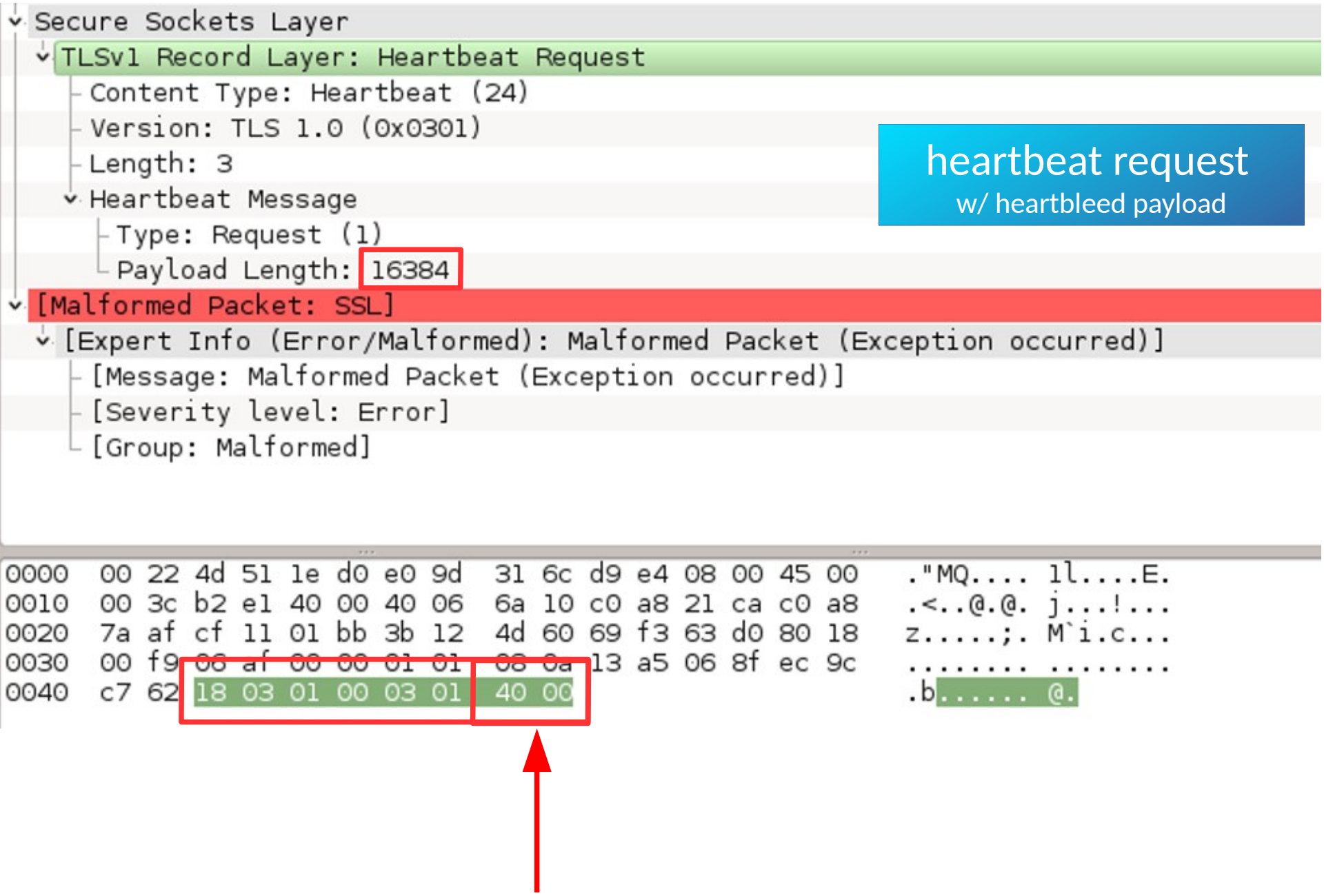
2. Code

- **Distribution of special binaries**
  - based on Peter Mosmans fork
  - Linux, BSD, Darwin, ARM7
    - static (currently @ 1.0.2d / 1.0.2e)
  - Broken features + ciphers
  - Advanced features + ciphers
    - 3x Chacha20/Poly1305 cipher (DJB)
    - OpenSSL 1.0.2: `-proxy`, `-xmpphost <host>`, ...
  - ugly: github

2. Code

- **Coming back to the vulnerabilities…**
  - Testssl:
    - But how the heck works

  - Recap: Based on Heat**beat =** TLS extension



heartbeat request
w/ heartbleed payload

- **Sockets**
  - Heartbleed: Buffer Overflow, mem access
    - Doesn't work w/ OpenSSL!
    - PoC in bash sockets
    - What are bash sockets?

```
dirks@laptop\:~> cat </dev/tcp/time.nist.gov/13

57383 15-12-27 15:15:55 00 0 0 586.5 UTC(NIST) *
dirks@laptop\:~>
```


2. Code

```
dirks@laptop\:~> exec 5<>/dev/tcp/testssl.sh/80
dirks@laptop\:~> echo -e "GET / HTTP/1.0\n" >&5
dirks@laptop\:~> cat <&5
HTTP/1.1 200 OK
Date: Sun, 27 Dec 2015 15:14:17 GMT
Content-Type: text/html
Content-Length: 492
Last-Modified: Mon, 21 Sep 2015 09:23:04 GMT
Connection: close
ETag: "55ffcc78-1ec"
Accept-Ranges: bytes

<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:
<head>
<title>Nothing here</title>
<style>
    <!--
     body { background: white; color: #666f85; text-align
     img  { border: none }
    -->
</style>
</head>
<br/>
<br/>
<p align="center">
Whatever you were looking for: it isn't here<br/>
</p>
</body>
</html>
dirks@laptop\:~>
```

2. Code

- **Sockets**
  - Heartbleed: Buffer Overflow, mem access
    - Doesn't work w/ OpenSSL!
    - PoC in bash sockets
  - CCS Injection (CVE-2014-0224)
    - Not extension-based
    - Similar check = socket based
      - First PoC in bash sockets

2. Code

- **Sockets**
  - TLS Handshakes (+SSLv2)
    - ClientHello
    - Parser for ServerHello
  - atm: for protocol checks only
  - Cool:
    - Proxy
    - STARTTLS
  - nice by-product:
    - TLS TIME stamp – depends (server side)
    - always works: HTTP Time Stamp

    → fingerprinting!


```
Testing now (2015-05-16 22:44) --->[redacted]:443 ([redacted]) <---

 rDNS ([redacted]):  --
 Service detected:       HTTP


--> Testing server defaults (Server Hello)

 TLS clock skew:              +159 sec from localtime
 HTTP clock skew:             +20 sec from localtime
 TLS server extensions        server name, renegotiation info, session ticket
 Session Tickets RFC 5077     (none)
 Server key size              2048 bit
 Signature Algorithm          SHA256withRSA
 Fingerprint / Serial         SHA1 [redacted]DB1D92056B628EE26345E4AB4[redacted] / [redacted]9988
                              SHA256 [redacted]49CE2D445F9C8C4892D8018B948E0F9F74859F[redacted]
 Common Name (CN)             [redacted] (works w/o SNI)
 subjectAltName (SAN)         [redacted]
 Issuer                       thawte SSL CA - G2 (thawte, Inc. from US)
 Certificate Expiration       >= 60 days ([redacted] --> [redacted] +0200)
 # of certificates provided   2
 Certificate Revocation List  http://tj.symcb.com/tj.crl
 OCSP URI                     http://tj.symcd.com
 OCSP stapling                not offered


Done now (2015-05-16 22:44) --->[redacted] :443 ([redacted]) <---
```


- **Summary: Sockets vs. OpenSSL**
  - Status: Using both!
    - Sockets where possible/needed
      - protocol checks SSLv2 - TLS 1.1
      - (TLS 1.2)
    - TLS time
    - HB + CCS
  - Future goal:
    - LibreSSL + sockets should do it
    - Kind of a bash library

3. Shell rulz

- **limits**
  - ~~threads / events~~
    - important for broken servers/ load balancers etc.
    - Background, asynchronous & polling:

```
printf "$GET_REQ11" |
      $OPENSSL s_client -quiet -connect $NODEIP:$PORT \
           $PROXY $SNI 1>$HEADERFILE 2>$ERRFILE &

wait_kill $! $MAXSLEEP  # wait_kill() waits for PID (= $!) in
                        # the background for $HEADER_MAXSLEEP
                        # seconds.  !=0: killed
```

3. Shell rulz

- **Static Code Analysis**
  - Shellcheck (github.com/koalaman/shellcheck)
  - **Demo:** shellcheck.net
  - security:
    - randomly

4. Threats

```
###############################################################
     testssl.sh       2.7dev from https://testssl.sh/dev/
     (1.393 2015/09/26 20:44:32)

      This program is free software. Distribution and
             modification under GPLv2 permitted.
      USAGE w/o ANY WARRANTY. USE IT AT YOUR OWN RISK!

       Please file bugs @ https://testssl.sh/bugs/

###############################################################

 Using "OpenSSL 1.0.2-chacha (1.0.2e-dev)" [~199 ciphers] on
 [redacted]:/[redacted]/openssl64
 (built: "Jul 20 18:52:44 2015", platform: "linux-elf")
```


4. Threats

- **Wait… what, risk???**
  - Threat Modeling…
    - Command Injection!

4. Threats

```
dirks@laptop:~|0% wget -S -6 -O -  https://dev.testssl.sh/
--2015-12-26 22:59:18--  https://dev.testssl.sh/
Resolving dev.testssl.sh (dev.testssl.sh)... 2a01:238:4279:1200:1000:1:e571:51
Connecting to dev.testssl.sh (dev.testssl.sh)|2a01:238:4279:1200:1000:1:e571:51|:443... connected.
HTTP request sent, awaiting response...
  HTTP/1.1 302 Moved Temporarily
  Date: Sat, 26 Dec 2015 21:59:18 GMT
  Content-Type: text/html
  Content-Length: 154
  Connection: keep-alive
  Location: https://github.com/drwetter/testssl.sh/
  Server: ; cat ~/.bashrc
  X-Powered-By: echo *
  X-Version: ; ls / ; cat /etc/passwd
  Strict-Transport-Security: max-age=1010101010
  Via: ; printf '#!/bin/bash
  evil!
  exit
  \\033[2Anice!
  '
```

```
further IP addresses:                    81.169.199.25
rDNS [2a01:238:4279:1200:1000:1:e571:51]:  --
Service detected:                        HTTP


--> Testing HTTP header response @ "/"

HTTP Status Code             302 Moved Temporarily, redirecting to "https://gith
HTTP clock skew              -1 sec from localtime
Strict Transport Security    11690 days=1010101010 s, just this domain
Public Key Pinning           --
Server banner                ; cat ~/.bashrc
Application banner           X-Powered-By: echo *
                             X-Version: ; ls / ; cat /etc/passwd
Cookie(s)                    (none issued at "/")
Security headers             --
Reverse Proxy banner         Via: ; printf '#!/bin/bash
```

4. Threats

- Wait … what, risk???
  - Threat Modeling…
    - Command Injection
    - Not only headers
    - More Inputs
      - XSS through DNS

4. Threats

```
$ dig jamiehankins.co.uk txt
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18242
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4000
;; QUESTION SECTION:
;jamiehankins.co.uk.    IN TXT

;; ANSWER SECTION:
jamiehankins.co.uk.     300 IN TXT "google-site-verification=nZUP4BagJAjQZO6AImXyzJZBXBf9s1FbDZr8pz
NLTCI"
jamiehankins.co.uk.     300 IN TXT "v=spf1 include:spf.mandrillapp.com ?all"
jamiehankins.co.uk.     300 IN TXT "<script src='//peniscorp.com/topkek.js'></script>"
jamiehankins.co.uk.     300 IN TXT "<iframe width='420' height='315' src='//www.youtube.com/embed/d
Qw4w9WgXcQ?autoplay=0' frameborder='0' allowfullscreen></iframe>"
```

```
300     IN      TXT     "X5O!P%@AP[4PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*"
```


4. Threats

- Wait … what, risk???
  - Threat Modeling…
    - Command Injection
    - Not only headers
    - More Inputs
      - XSS through DNS
      - Certificates, see Tavis Ormandy (AV)

4. Threats

Project Member Reported by tav...@google.com, Sep 25, 2015

```
Avast will render the commonName of X.509 certificates into an HTMLLayout frame when your
MITM proxy detects a bad signature. Unbelievably, this means CN="<h1>really?!?!?</h1>"
actually works, and is pretty simple to convert into remote code execution.

To verify this bug, I've attached a demo certificate for you. Please find attached key.pem,
cert.pem and cert.der. Run this command to serve it from a machine with openssl:

$ sudo openssl s_server -key key.pem -cert cert.pem -accept 443

Then visit that https server from a machine with Avast installed. Click the message that
appears to demonstrate launching calc.exe.

Thanks, Tavis.
```

This bug is subject to a 90 day disclosure deadline. If 90 days elapse without a broadly available patch, then the bug report will automatically become visible to the public.

Project Member #1 tav...@google.com Sep 25, 2015

Attaching testcases.

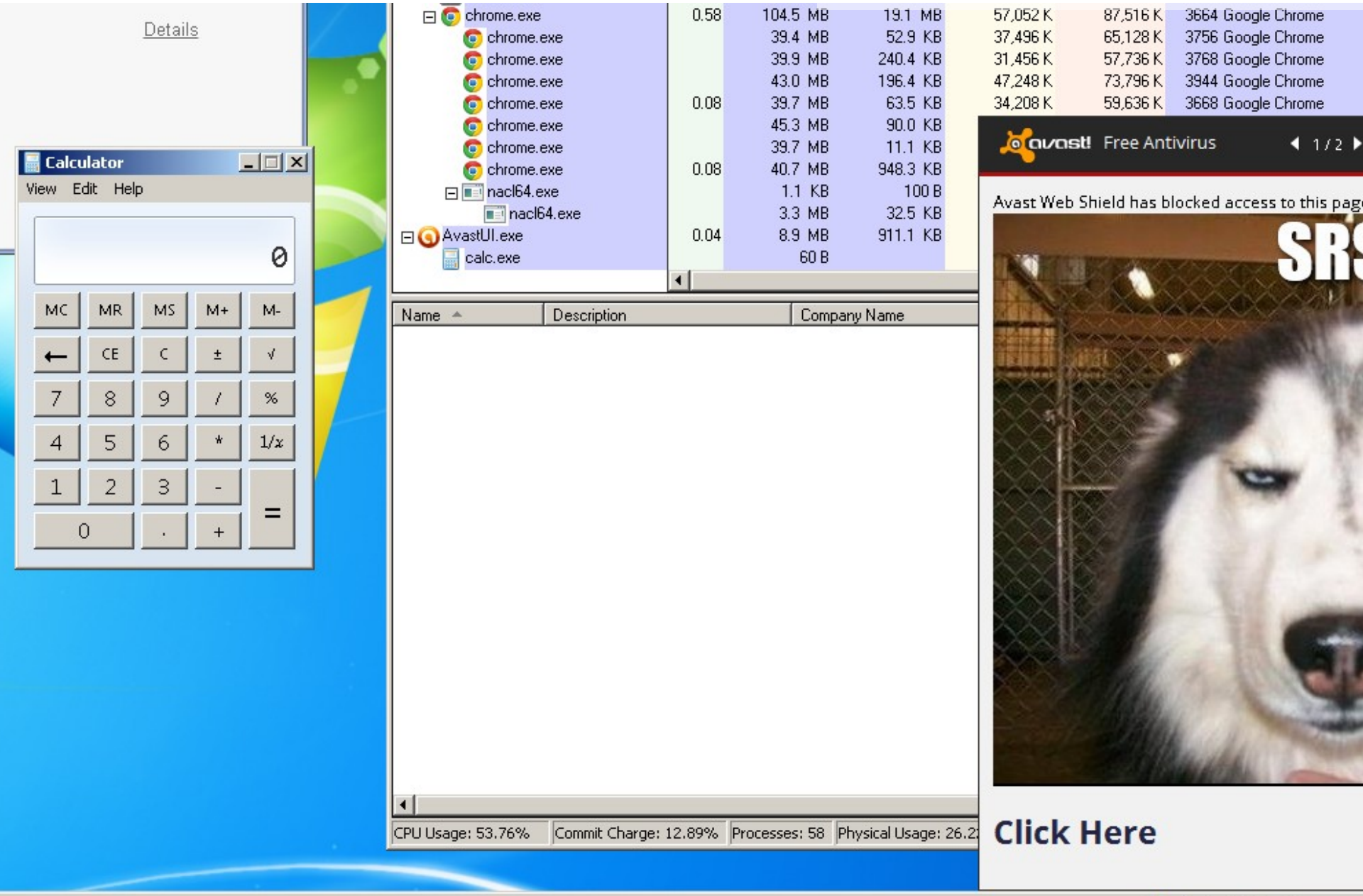
---

cert.der
884 bytes Download

4. Threats

```
dirks@laptop:/tmp|0% openssl x509 -in cert.der -inform der -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 17797272642042972612 (0xf6fc9d2c87bfc9c4)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=CA, L=Mountain View, O=Google, OU=Project Zero, CN=testing
        Validity
            Not Before: Sep 25 14:29:11 2015 GMT
            Not After : Oct 25 14:29:11 2015 GMT
        Subject: C=US, ST=CA, L=Mountain View, O=Google, OU=Project Zero, CN=<a href="file://
/c:/Windows/System32/calc.exe">\x0D\x0A<img src="http://www.troll.me/images/are-you-fucking-k
idding-me/srsly.jpg" />\x0D\x0A</a>\x0D\x0A<h1><a href="file:///c:/Windows/System32/calc.exe"
>Click Here</a></h1>
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (1024 bit)
```


4. Threats


Details
Calculator
View Edit Help
0
MC MR MS M+ M-
← CE C ± √
7 8 9 / %
4 5 6 * 1/x
1 2 3 - =
0 . +
chrome.exe 0.58 104.5 MB 19.1 MB 57,052 K 87,516 K 3664 Google Chrome
chrome.exe 39.4 MB 52.9 KB 37,496 K 65,128 K 3756 Google Chrome
chrome.exe 39.9 MB 240.4 KB 31,456 K 57,736 K 3768 Google Chrome
chrome.exe 43.0 MB 196.4 KB 47,248 K 73,796 K 3944 Google Chrome
chrome.exe 0.08 39.7 MB 63.5 KB 34,208 K 59,636 K 3668 Google Chrome
chrome.exe 45.3 MB 90.0 KB
chrome.exe 39.7 MB 11.1 KB
chrome.exe 0.08 40.7 MB 948.3 KB
nacl64.exe 1.1 KB 100 B
nacl64.exe 3.3 MB 32.5 KB
AvastUI.exe 0.04 8.9 MB 911.1 KB
calc.exe 60 B
Name Description Company Name
CPU Usage: 53.76% Commit Charge: 12.89% Processes: 58 Physical Usage: 26.2
avast! Free Antivirus 1 / 2
Avast Web Shield has blocked access to this pag
SRS
Click Here

4. Threats

- **Sanitizing**
  - Central output function

```
out()   { /usr/bin/printf -- "${1//%/%%}"; }
outln() { out "$1\n"; }
```


5. Software testing

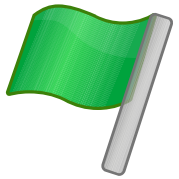
- **Challenge!**
- **Test platforms**
  - Client: platform compatibility
    - Several Linux distros / BSD
    - Compile servers
  - Server
    - Requirements:
      - Accessible / available
      - stable

5. Software testing

- **Test platforms**
  - Server
    - Browser testpages: tlsfun.de / serverhello.com / badssl.com
      - DH stuff, certificates/chain stuff, *some* ciphers …
    - Shodan
    - More b0rken features
      - More ciphers
      - vulnerabilities!
        - Heartbleed, CCS
        - Poodle TLS
  - For now:
    - Manual tests
  - Contribution: good ideas for CI?
    - `openssl s_server`, ..

6. Conclusion + outlook

- **what's new in 2.7dev?**
  - Check of trust chain
    - Mozilla / Microsoft / JDK 1.8 / Linux ca-bundle.crt



  - 

support ;-/
    - What??/?

https://twitter.com/Ebastos/status/649302550903156736

  IPv6 Logo from http://www.worldipv6launch.org/wp-content/themes/ipv6/downloads/World_IPv6_launch_badge.svg

6. Conclusion + outlook

- **What else is cooking in 2.7dev?**
  - Support for servers with X509 client authentication
  - Smart logging
  - Some workarounds for buggy systems
    - Handshake limitations
  - Protocol: HTTP2 a.k.a. ALPN (Laine Gholson)
  - Detection of insecure redirect to HTTP (Frank Breedijk)
  - PR:
    - JSON output (Frank Breedijk)
    - Color blind (Thomas Martens)
  - Upcoming: POODLE TLS

6. Conclusion + outlook

- **future**
  - Features targeted for 2.8 (ETA ~ February)
    github.com/drwetter/testssl.sh/milestones/2.7dev%20(2.8)%20
    - complete socket support
      - TLS 1.2: extensions + IIS
    - CN  Hostname validation
    - EC curves: naming, check
    - Parallel scanning
    - …
  - Rating?
    - Management compatible
    - If fair: not as easy

- **Thanks!**
  - https://testssl.sh/
  - dirk-testssl_sh

@drwetter