There's worse than SSL

András Veres-Szentkirályi
vsza@silentsignal.hu

Camp++
2015

Scenario

- Executable used as client for exchanging financial info
  - (including on-line banking credentials)
- Server only available as a service
  - (no executable)
- No source code available
- Proprietary protocol
  - (allegedly encrypted)

Network traffic analysis

- *Wireshark* for capturing network traffic and stream reconstruction
- *Flow tools* for differential analysis
  `https://github.com/silentsignal/flowtools`

What we know so far

- Client sends two numbers
- Server sends one number
- Client sends one number
- This is followed by frames of $n \times 16$ bytes

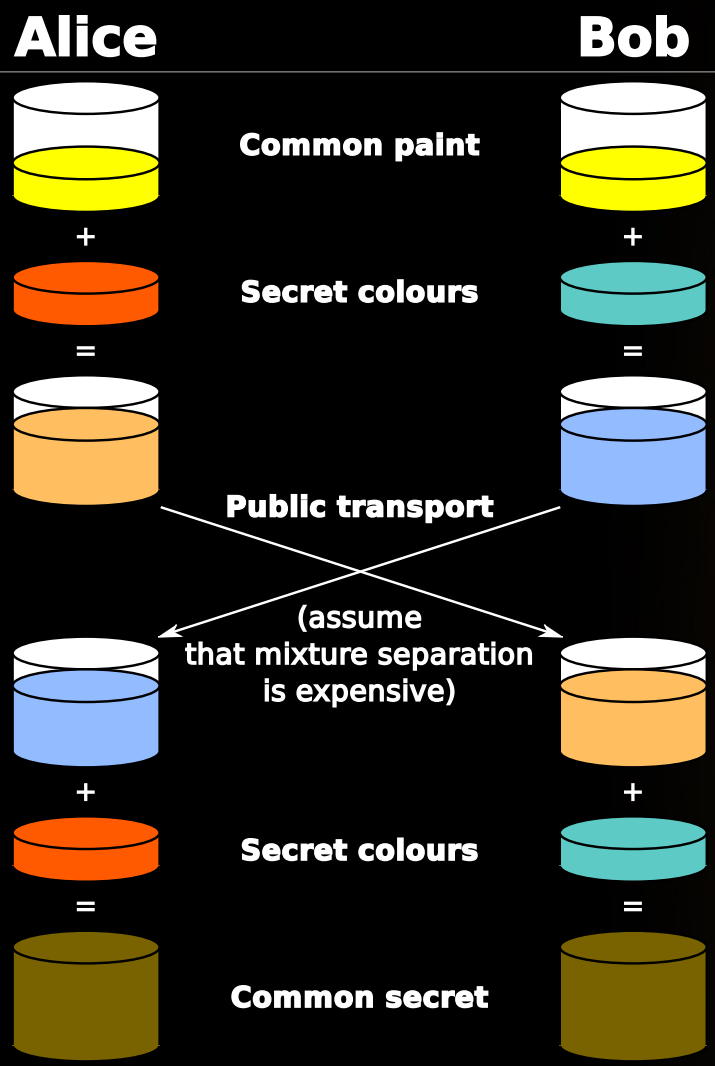
What we can deduce

- Client sends two numbers
- Server sends one number
- Client sends one number
  - Diffie-Hellman key exchange?
- This is followed by frames of $n \times 16$ bytes
  - Matches AES block size!

Diffie-Hellman Key Exchange

- Alice and Bob agree to use a prime number $p = 23$ and base $g = 5$ (which is a primitive root modulo 23).
- Alice chooses secret $a = 6$, sends $A = g^a \mod p$
  - $A = 5^6 \mod 23 = 8$
- Bob chooses secret $b = 15$, sends $B = g^b \mod p$
  - $B = 5^{15} \mod 23 = 19$
- Alice computes $s = B^a \mod p$
  - $s = 19^6 \mod 23 = 2$
- Bob computes $s = A^b \mod p$
  - $s = 8^{15} \mod 23 = 2$
- Alice and Bob now share a secret (the number 2).

Source: https://en.wikipedia.org/wiki/Diffie-Hellman


Alice
Bob
Common paint
+
Secret colours
=
Public transport
(assume
that mixture separation
is expensive)
+
Secret colours
=
Common secret

Static binary analysis of client

- Find out if Diffie-Hellman is used
- Find out how the AES key is derived from DH secret
- Disassemble and partially decompile using *reverse* `https://github.com/joelpx/reverse`

System V AMD64 ABI

The calling convention of the System V AMD64 ABI is followed on Solaris, **Linux**, FreeBSD, Mac OS X, and other UNIX-like or POSIX-compliant operating systems. The **first six integer or pointer arguments** are passed in registers **RDI, RSI, RDX, RCX, R8**, and **R9**.

Source: `https://en.wikipedia.org/wiki/X86_calling_conventions#System_V_AMD64_ABI`

Are we there yet?

- We managed to mount a MitM attack against the service – great, let's go celebrate!
- What if we cannot modify the traffic?
- Diffie-Hellman key exchange is pretty robust
- Look for the weakest link
- Where do the DH parameters come from?

What we know so far

- Diffie-Hellman parameters are 80 digit numbers
- Digits are filled from left to right using the least significant decimal digit of the return value of rand_r
  - The seed is the OS process ID
  - Getting the next number is fast
  - There are $2^{16} = 32768$ PIDs on a Linux system by default
  - OpenSSL, anyone?
- If it starts with zero, replace with 1

Decrypting captured data

- Regenerate DH secret from OS PID value
- Read and dissect packets from PCAP
- Extract public DH params and encrypted data
- Derive AES key and decrypt payload

`https://jon.oberheide.org/blog/2008/10/15/dpkt-tutorial-2-parsing-a-pcap-file/`

Conclusion

- You do not roll your own crypto
- You DO NOT roll your own crypto
- Secure random is essential
  - (can be tricky on the go and in VMs or the cloud)
- Authentication is essential
- Security by obscurity doesn't work


Thanks for your attention!
Facebook
vsza@silentsignal.hu
web
e-mail
SILENT SIGNAL
VÉSZJELZÉS HELYETT...
WWW.SILENTSIGNAL.HU