Essay by Steven Leego
“CVE-2015-0701 vulnerability”

Supervisor: Meelis Roos
Tartu 2016
Introduction
The vulnerability found its way to life when OpenSSL 1.0.2 version was released on 19.03.2015. The release suffered from a previously known attack "Key Recovery Attack on Diffie-Hellman small subgroups" [3].

Beforehand OpenSSL usually only ever generated DH parameters based on "safe" primes. In the 1.0.2 version, support was provided for generating X9.42 style parameter files such as those required for RFC 5114 support. The primes used in such files might not use a "safe" prime. An attacker could use this fact to find the victim’s private DH exponent. This attack requires that the attacker complete multiple handshakes in which the victim uses the same private DH exponent. This component could be used for a man-in-the-middle attack on SSL/TLS connections. [4][5][6]

Background
Diffie-Hellman key exchange
The Diffie-Hellman algorithm provides the capability for two communicating parties to agree upon a shared secret between them. Its an agreement scheme because both parties add material used to derive the key (as opposed to transport, where one party selects the key). The shared secret can then be used as the basis for some encryption key to be used for further communication. [2][9]

Alice
- Common point
- Secret colours
- Public transport
  (assume that mixture separation is expensive)
- Secret colours
- Common secret

Bob
- Common point
- Secret colours
- Common secret

Illustration [11]

Safe prime
A safe prime is a prime number of the form 2p + 1, where p is also a prime. Safe primes are important in Diffie-Hellman key exchange, because the multiplicative
group of numbers modulo $2p + 1$ has a subgroup of large prime order. It is usually this prime-order subgroup that is desirable, and the reason for using safe primes is so that the modulus is as small as possible relative to $p$. [1]

**The vulnerability**

**Parameters**
I will use these parameters in explaining the vulnerability:

- $p$ - the prime number
- $g$ - the generator with order
- $q$ - the size of the prime-order subgroup generated by $g$
- $Y$ - public key
- $X$ - private key

**Prerequisites**

1. **$p$ is a non-safe prime.**

   Non-safe primes are generated with:
   
   - Genpkey when dh_rfc15114 option is enabled
   - dhparam when the -dsaparam option is enabled.

2. **$g$ is reused.**

   $g$ is reused when:
   
   - SSL_CTX_set_tmp_dh() / SSL_set_tmp_dh() is used and SL_OP_SINGLE_DH_USE is not set.
   - SSL_CTX_set_tmp_dh_callback() / SSL_set_tmp_dh_callback() is used, and both the parameters and the key are set and SSL_OP_SINGLE_DH_USE is not used.
   - Static DH ciphersuites are used.

**Attack concept**

The server chooses his private key to be $X^b$ and then transmits $Y^b = g^X^b \pmod{p}$.

The attacker chooses $b$ where order($b$) is small (and is equal to one of the small factors of $p-1$).

The attacker then chooses $X^a$ and calculates $Y^a = g^*X^a \pmod{p} * b$

With the received $Y^b$ the attacker tries with multiple handshake sessions $Y^b*X^a * b^j \pmod{p}$. He does it $j$-times where $0 < j < \text{order}(b)$.

At this point the attacker found $j = X^b \pmod{\text{order}(b)}$.

Once this is done the attacker repeats the same steps with a different computationally feasible $b'$ where order($b'$) is small.
The resulting partial secrets can then be combined using the Chinese Remainder Theorem. And for the remaining bits Shanks's method or Pollard's lambda method can be used. [4][12]

The fix
The fix adds an additional check where the "q" parameter is available. This stops from disclosing that q and is the only possible defence for static DH ciphersuites.

Additionally the SSL_OP_SINGLE_DH_USE option has been switched on by default which prevents the g from being reused. The option cannot be disabled anymore either, although it has some impact on performance.

The patch 1.0.2f, which provided this fix went live on 28-01-2016. [4][6][10]

References:
3. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.5296 (Research, that explains the attack principle)
5. https://access.redhat.com/security/cve/cve-2016-0701
10. https://git.openssl.org/?p=openssl.git;a=commit;h=ffaef3f1526ed87a46f82fa4924d5b08f2a2e631 (Fix git commit)