Research

Security in Norwegian Internet Banks

14th of November 2011

Juan J. Güelfo
CEO, IT-security consultant and
IT-security researcher at Encripto AS

Table of contents

About Encripto AS

Encripto AS is a Norwegian company which specializes in information security. We mainly work with penetration testing and education within information security. Encripto works with big and small companies, both from public and private sectors. Quality is priority number one for everything we do, and details matter to us.

Encripto AS is committed to information security. We do research to discover trends, new vulnerabilities and better ways to mitigate them. We believe in acting as good internet citizens to the industry, whether you are a provider or a user.

You can read more about us at http://www.encripto.no

Revision history

18th of November 2011. Score adjustment (Bank A from 4.4 to 4.9) due to a false positive.
14th of November 2011. Initial release.

Disclaimer

The material presented on this document is for educational purposes only. Encripto AS cannot be responsible for any loss or damaged carried out by any technique presented in this material. The reader is the only one responsible for applying this knowledge, which is at his / her own risk.

Any of the trademarks, service marks, collective marks, design rights, personality rights or similar rights that are mentioned, used or cited in this document is property of their respective owners.

License

1. Introduction

Online banking is one of the tools that Norwegian society is relaying on for conducting financial operations. A very big number of Norwegian customers manage their accounts or pay invoices from these internet banking applications. But, are they secure?

Encripto AS has carried out a research which analyzes the quality of the SSL/TLS implementation in 10 different Norwegian banks. This whitepaper focuses on configuration problems around SSL/TLS implementations at the banks, which can be discovered without any kind of offensive techniques. Anybody with a web browser and a SSL/TLS client can verify the results.

The goal of this document is to be informative and to increase awareness around information security. Therefore, this document has been written in a low-medium technical level, so the public can understand the situation. A more technical approach has been included at the end of the document (Appendix A), where the methodology used during this study is described.

2. What is SSL/TLS?

SSL and TLS are cryptographic protocols which are widely used on the internet. These protocols are used for protecting communications over a network or the Internet.

In theory, if an attacker intercepts the communication protected by SSL/TLS, it will be very difficult for him to get access to the actual information. So, SSL/TLS is usually used in services that transmit sensitive or important information. It could be said that SSL/TLS is like an armor for the information. However, improper implementation of the protocols can make this armor ineffective.

SSL/TLS is used for protecting a long list of services over the internet, such as web or e-mail. Online banking basically relies on the web and SSL/TLS for ensuring confidentiality.

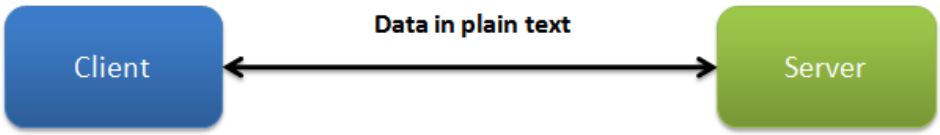


Fig. 1: Client and server transmitting data over an unencrypted channel

Fig. 2: Client and server transmitting data over a SSL/TLS connection

2.1 SSL / TLS is not enough for being secure

SSL/TLS only affects the communication between the server and the client, and it requires a proper configuration for being effective.

Internet bank web applications are another important part in this equation. They must be well built and tested for preventing security problems.

Security does not apply only to servers. Websites (software) and applications can have holes which can compromise security. That's why it is crucial to run security tests against both servers and applications, not only servers.

In addition, the security at the client side should be good in order to prevent access to the information before it is sent to the server.

2.2 The study

The study has analyzed the way that a web browser (SSL/TLS client) negotiates an encrypted connection with the internet banking server, and the quality of such connection. The analysis has not involved offensive testing against the servers.

The tools used to carry out the experimental part of this research were a web browser (Firefox 8.0), OpenSSL 0.9.8r, GnuTLS 2.10.5, SSLscan 1.8.2 and SSL Cipher Check 1.6.

This research was carried out in November 2011 and all banks were made aware of the results before making this document available to the public.

2.3 Banks under study

This research has analyzed a sample with 10 banks (listed alphabetically):

- BN Bank
- DNB Nor
- Fokus Bank
- Landkreditt Bank
- Nordea
- Sparebank1
- Sparebanken Møre
- Sparebank1 Vest
- Storebrand
- YA Bank

The results presented on this report have been anonymized, so the conclusions are marked as Bank A, Bank B, and so on. In addition, the report presents the information sorted by overall quality, from lowest to highest. Therefore, the list above does not have to match the same result position or ranking.

2.4 Categories and parameters taken into account

This research has checked the following elements or categories regarding the SSL/TLS implementation:

- **Category 1: SSL/TLS Certificate quality**
  Certificates are an essential part of an SSL/TLS infrastructure. In a nutshell, a certificate is used for identifying the server (the user can know that it belongs to the bank), and for establishing the SSL/TLS encrypted channel.

  Certificates are issued by a Certificate Authority (CA). The organization applying for it (the bank in this case) will have to follow a process for getting such certificate from the CA.

  There are three types of certificates, each of them with special security features and different application process:

  - ***Domain Validation Certificate (DV)***
    The CA checks the right of the applicant to use a specific domain name (ex: www.example.com).

    No company identity information is checked by the CA, and no identity information is presented to the user when he or she visits a site with this kind of certificate (only encryption).

  - ***Organization Validation Certificate (OV)***
    The CA checks the right of the applicant to use a specific domain name (ex: www.example.com), but this time, the CA will conduct some vetting of the organization.

    This information will be displayed to the user when he or she visits a site which an OV certificate. This allows users to know a little bit about the organization that is associated to such domain.

  - ***Extended Validation Certificate (EV)***
    Extended Validation certificates have the strictest process. The CA will check the right of the applicant to use the domain name (ex: www.example.com), and a thorough vetting of the organization will be conducted.

    EV certificates are recommended for important or valuable web sites.

    For more information about the vetting process for EV certificates, please check the EV Guidelines from the CA/Browser forum[1].

- **Category 2: Certificate information**
  Certificates contain information which is relevant for a satisfactory SSL/TLS configuration.

  - ***Certificates should be issued by a trusted Certificate Authority***
    This will ensure that Man-In-The-Middle (MITM) attacks are detected.

To illustrate this example, imagine a passport which is issued by a shop, or made at home. Would you trust it?

A know Certificate Authority can be compared to the Police or any official governmental institution which provides authenticity to the document (or certificate).

- ***Common name***
  If a certificate does not match the domain name it intends to protect or identify, it will raise suspicions. Can you imagine identifying yourself with somebody else's passport at the airport?

- ***Expiry date***
  Expired certificates do not offer authenticity anymore. To continue the illustration, would an airport border control trust an expired passport?

- ***Use of a not trusted or unknown Certificate Authority***
  Internet banks are published on the internet and they can be used from any part of the world. Therefore, they should use certificates issued by known CAs.

- ***Use of a revoked certificate***
  Certificates can be revoked if needed, which means that they can be "cancelled" and they will not be valid any more. Using a revoked certificate is definitely not an option when it comes to online banking.

- **Category 3: Supported protocols**
  There are many protocol versions when it comes to SSL and TLS. Some versions are affected by weaknesses and they are not secure enough for protecting communications. Therefore, a proper SSL/TLS implementation should not accept these vulnerable protocol versions.

- **Category 4: Key exchange**
  The key exchange between the client and the server will be analyzed. The key exchange is the process where identities are verified (the server identity in this case), and where the secret key for encrypting the data during the rest of the encrypted session is agreed between the client and the server.

  There are two important factors when it comes to key exchange:

  - ***Authentication***
    If the client and the server perform key exchanges without authentication, it will allow attackers to perform Man-In-The-Middle attacks. That means, such attacker would get access to the "protected" communication and the information sent there.

  - ***Certificate key length***
    SSL / TLS use public cryptography for exchanging the keys which are going to be used for encrypting the data during the rest of the session.

The more robust the certificate's key is, the harder it is to break the key exchange phase.

- **Category 5: Cipher suites**
  In order to encrypt the communication, the client (a web browser in this case) and the server need to negotiate the cipher suite. Not all cipher suites used by SSL/TLS are secure. That's why a proper SSL/TLS implementation should not offer them as an alternative for encrypting information. In addition, a proper SSL/TLS implementation should use good cipher strengths (key length).

- **Category 6: Insecure features and vulnerabilities**
  Improper SSL/TLS configuration can affect both the client's and the server's security. This section will analyze if the implementation is vulnerable to the BEAST[2] attack, insecure renegotiations[3] and client initiated renegotiations.

2.5 Score and proper configuration

This section is going to explain the scoring system we have used for evaluating each of the categories.

The score model proposed is inspired by the SSL Server Rating Guide[4]. Since this guide does not include the challenges or attacks that SSL/TLS have suffered during the past 2 - 3 years, we have decided to introduce changes.

Each category will have a weight on the final score:

Category	Weight
SSL/TLS Certificate quality	10 %
Certificate information	10 %
Supported protocols	20 %
Key exchange	20 %
Cipher suites	20 %
Insecure features and vulnerabilities	20 %

The score goes from 0 to 10, where 0 is worst and 10 is best.

- **Category 1: SSL/TLS Certificate quality**
  Internet banking is a high value service, and therefore, this report will consider an EV certificate as the proper configuration.

  With the tools we have used in this research, it was not possible to differ between Domain Validation and Organization Validation Certificates. Therefore, we have grouped them together.

The score assigned to this category will be:

Type of certificate	Score
No certificate or no SSL/TLS support	0 points
Domain Validation (DV) or Organization Validation Certificate (OV)	5 points
Extended Validation Certificate (EV)	10 points

- **Category 2: Certificate information**
  When a certificate has problems with any of the parameters or situations we have presented before, the ability to detect Man-In-The-Middle attacks is lost.

  If a web browser detects any problem regarding these certificate information parameters, it will launch a security warning and the user would have to consider whether the connection is trusted or not.

  At this point, security depends on a user decision, but the user has no way to tell if an attacker is intercepting the communications.

  A proper SSL/TLS implementation should always provide correct certificate information, especially when it comes to online banking.

  The score assigned to this category will be:

Certificate information	Score
Errors due to certificate information	0 points
No errors	10 points

- **Category 3: Supported protocols**
  SSL and TLS have different versions, which can be supported by a server simultaneously. Some of these versions are affected by known weaknesses.

  A proper implementation should not use SSL 2.0, or rely only on SSL 3.0.

  Since this category can have several values at once, we have considered the following score for every single protocol:

Protocol	Score
SSL 2.0	2 points
SSL 3.0	4 points
TLS 1.0	6 points
TLS 1.1	8 points
TLS 1.2	10 points

The total score for this category will be calculated based on the formula:

$$Score = \frac{(best\ protocol + worst\ protocol)}{2}$$

- **Category 4: Key exchange**
  A proper SSL/TLS implementation should never allow weak key or anonymous key exchange. In addition, the certificate key length should be at least 2048-bit.

  The US National Institute of Standards and Technology (NIST) announced that 1024-bit RSA keys are no longer viable after 2010. According to them, 2048-bit keys should be viable until 2030.

Key length	Score
Anonymous or weak key exchange	0 points
Less than 512 bits	1 points
Less than 1024 bits (not including 1024)	3 points
Less than 2048 bits (not including 2048)	5 points
Less than 4096 bits (not including 4096)	8 points
More than or equals to 4096 bits	10 points

- **Category 5: Cipher suites**
  The length of the key used for encrypting the information when it is transmitted between the client and the server (and vice versa) will affect the level of security. The longer the key length is, the better security you will achieve.

  Please, note that the BEAST attack (which affects ciphers using CBC) has not been considered in this category. This has been included in the next section.

  The score regarding cipher suites is displayed on the following table:

Key length	Score
0 bits	0 points
Less than 128 bits (not including 128)	1 points
Less than 256 bits (not including 256)	7 points
More than or equals to 256 bits	10 points

  Since a SSL/TLS implementation can support multiple ciphers with different key lengths, the total score will be calculated following this formula:

$$Score = \frac{(strongest\ cipher + weakest\ cipher)}{2}$$

  This study does not take into consideration performance issues or decisions about why an organization chose a specific set of ciphers. It just focuses on security. SSL/TLS are cryptographic protocols and they will always have impact on the server performance, especially for websites with high traffic.

  Performance is usually affected by the key length. Long keys will reduce performance, but they will increase security. However, short keys will provide a better performance, but worse security.

- **Category 6: Insecure features and vulnerabilities**
  This section will focus on some specific features regarding the SSL/TLS configuration.

  - **Insecure renegotiation**
    SSL/TLS renegotiation allows you to establish a new SSL/TLS session over an already established SSL/TLS connection. This might be needed when the client and the server set new encryption keys, or new cipher suites.

    SSL/TLS implementations which support insecure renegotiations are susceptible to Man-In-The-Middle attacks. That means an attacker could insert data into the encrypted data stream.

    If a server supports secure renegotiation, it will announce it during the SSL handshake phase to the client. However, if the server has disabled renegotiation, there will be no indication of their status. Therefore, this is not 100% conclusive. It can be possible that a server which disables renegotiation still runs software vulnerable to insecure renegotiations.

    Insecure renegotiation is definitely a feature which should not be accepted for online banking.

  - **Denial of Service due to client initiated renegotiations**
    If a server supports client initiated renegotiations, a malicious client could be able to stress the server and provoke a Denial of Service attack. That means, the online bank service would be overloaded and it could not be reached by customers.

    For a client, starting a renegotiation has very low performance costs. However, this makes an extra effort for the server.

    Servers which are susceptible to DoS reflect a not robust infrastructure.

  - **BEAST attack**
    This attack allows an attacker to retrieve data from a SSL/TLS encrypted session. This attack affects TLS1.0 and SSL when ciphers use CBC mode.

    Therefore, checking the results from the previous categories (protocol support and cipher suites) will tell us if the SSL/TLS implementation is vulnerable to such attack.

Feature	Score
Insecure renegotiation (vulnerable to MITM)	0 points
Client initiated renegotiations (vulnerable to DoS)	0 points
Vulnerable to BEAST	0 points
No insecure renegotiation, no client initiated renegotiations and not vulnerable to BEAST	10 points

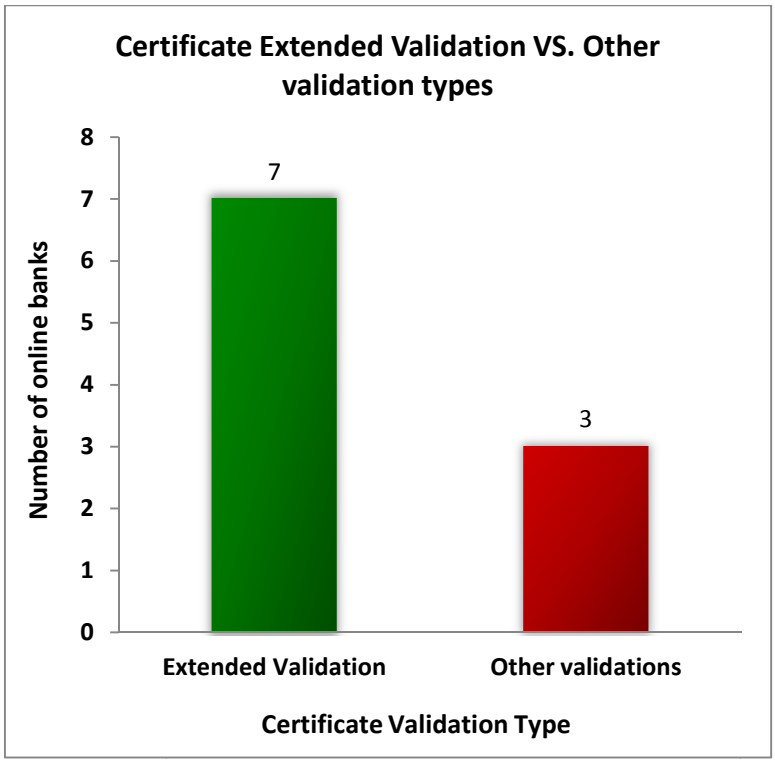
3. Facts, figures and general information

These are some general facts regarding the research and the sample.

3.1 SSL/TLS Certificate quality

According to our criteria, online banking is a high value service. Therefore, complete information about the organization running the service should be provided. The user should know that a thorough vetting has been done by the Certificate Authority before issuing a certificate for a bank. This is achieved only with Extended Validation.

The next graph shows that 7 of 10 internet banks use certificates with Extended Validation, while 3 of 10 use certificates which were issued not following the extensive vetting process agreed by the CA/Browser forum. See reference 1 for more information.



3.2 SSL/TLS Certificate Information

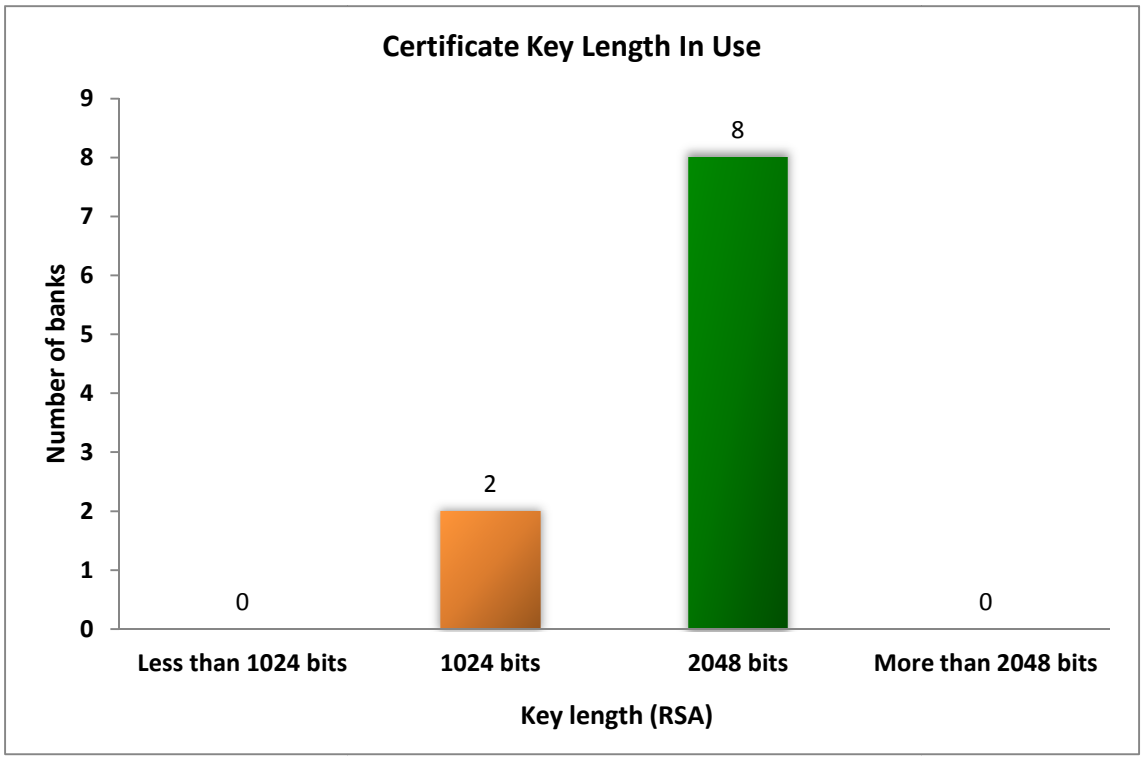
In order to benefit from all the security features provided by certificates, these should not contain any errors in their certificate information.

The results show that 10 of 10 online banks had certificates with no information errors or mismatches.

3.3 Certificate Key Length

The US National Institute of Standards and Technology (NIST) announced that 1024-bit RSA keys are no longer viable after 2010. According to them, 2048-bit keys should be viable until 2030.

The results show that 8 of 10 internet banks use certificates with 2048-bit key (RSA).



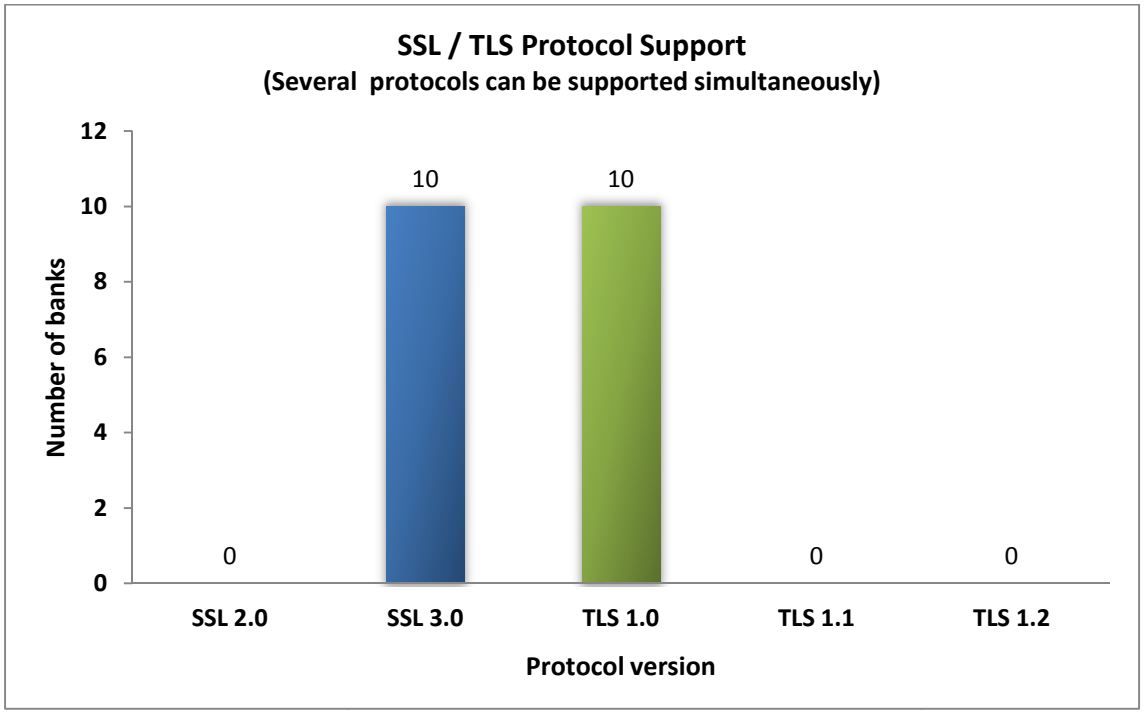
3.4 SSL / TLS Protocol support

SSL and TLS have different versions, which can be supported by a server simultaneously. Some these versions are affected by known weaknesses.

SSL version 2 is known to be insecure. On the other hand, SSL 3.0 and TLS 1.0 are affected by diverse vulnerabilities, which do not affect TLS 1.1 or TLS 1.2. Unfortunately, not too many servers on the internet support TLS 1.1 or higher.

A proper SSL/TLS implementation should never use SSL 2.0, and never rely only on SSL 3.0.

The results show that all internet banks support SSL 3.0 and TLS 1.0.



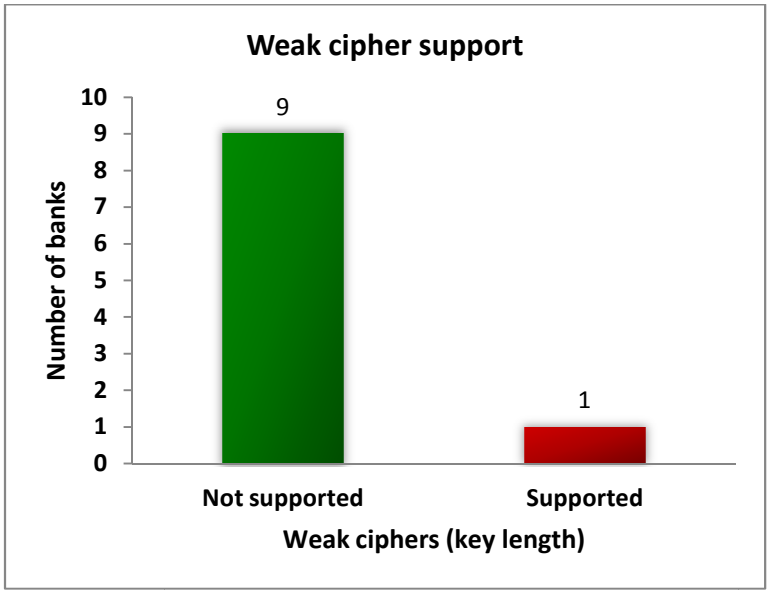
Update:

The study detected that one of the banks supported SSLv2. However, this has been considered a false positive. In practice, when connecting to the server using SSLv2, the server presented a HTML error informing that SSLv2 was not accepted.

3.5 Weak cipher support (based on key length)

Ciphers which use a short key length can be broken (ex: 40-bit / 56-bit keys). That would allow the decryption of communications. When it comes to ciphers suites, a key length is considered strong enough if it is equals to or greater than 128-bits.

The results have shown that 1 of 10 online banks supports weak ciphers. In other words, an attacker would be able to break the encryption in a short time when these weak ciphers are in use.



Update:

The study detected that one of the banks supported weak ciphers, which made a total of 2 banks with weak ciphers. However, this has been considered a false positive. In practice, when trying a connection to the server using a weak cipher, such bank presented a HTML error informing that the cipher was not accepted.

3.6 SSL / TLS insecure features and vulnerabilities

SSL/TLS implementations which support insecure renegotiations are susceptible to Man-In-The-Middle attacks. That means an attacker could insert data into the encrypted data stream between the client and the server.

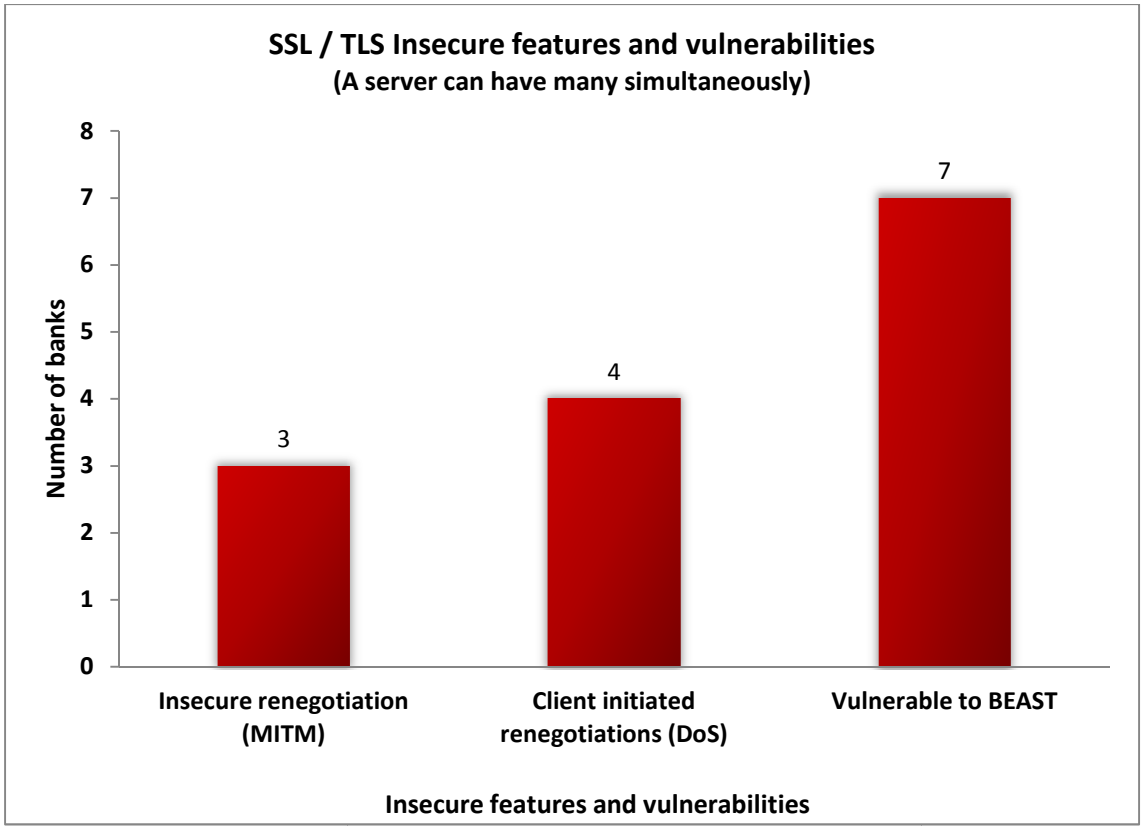
If a server supports client initiated renegotiations, a malicious client could be able to stress the server and provoke a Denial of Service attack. That means, the online bank service would be overloaded and it could not be reached by customers.

The BEAST vulnerability allows an attacker to retrieve data from a SSL/TLS encrypted session. This attack affects TLS1.0 and SSL when ciphers use CBC mode.

The results show that 3 online banks support insecure renegotiations. This means that an attacker could perform a Man-In-The-Middle attack between the customer and the bank.

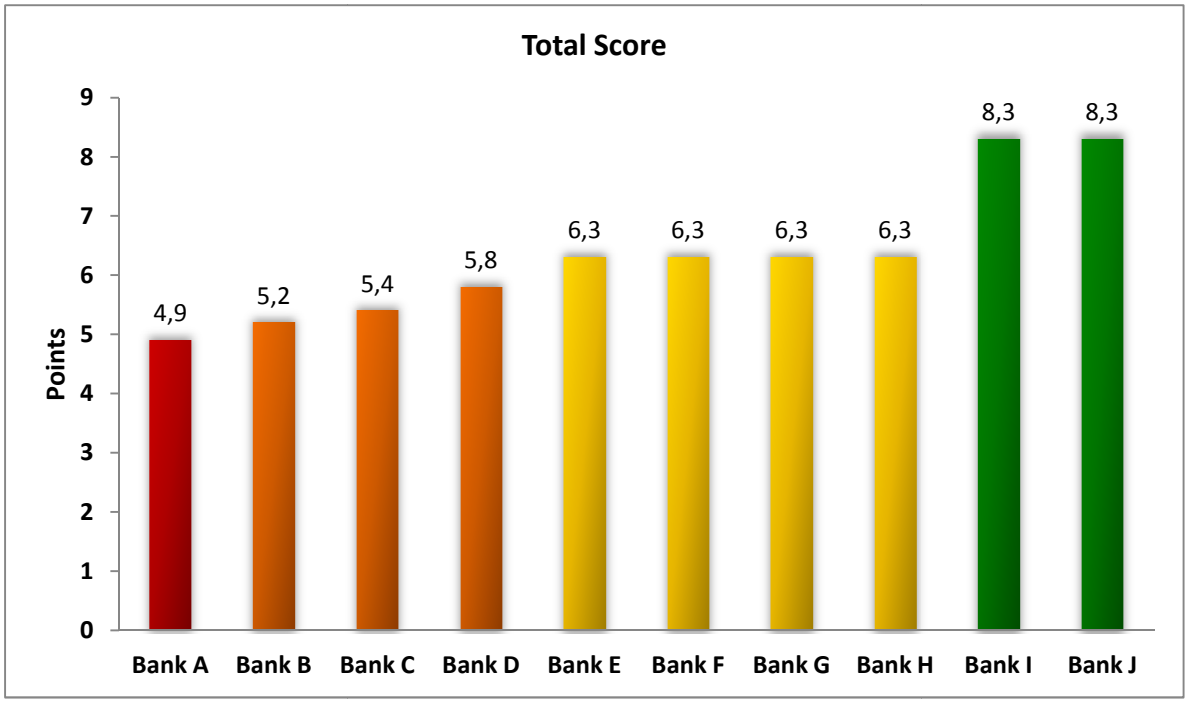
4 internet banks support client initiated renegotiations. This could be used to start a Denial of Service attack against the online bank service. With such attack, the result would be that the online banking server would be overloaded and probably taken out of the internet.

7 online banks are vulnerable to BEAST attack. This means that an attacker could have access to data sent through the encrypted connection between the client and the server.



3.7 Total Score

This is the graphic which shows the total score assigned to each anonymized internet bank, sorted from worst to best.



4. Facts and specific information

This section is going to describe the results found for each online bank. The results have been anonymized. Online banks have been sorted from worst security to best.

The number of insecure features and/or vulnerabilities has been taken into account for those internet banks which have gotten the same score.

The steps followed for testing can be found in the appendix A.

4.1 Bank A

Total Score: 4.9

Insecure features / vulnerabilities: 2.

- **Category 1: SSL/TLS Certificate quality**
  Score: 5
  SSL/TLS enabled. Not Extended Validation certificate.

- **Category 2: Certificate information**
  Score: 10
  No errors detected

- **Category 3: Supported protocols**
  Score: 5
  TLS 1.0, SSL 3.0

- **Category 4: Key exchange**
  Score: 5
  1024-bit certificate key length.

- **Category 5: Cipher suites**
  Score: 7
  The next table shows the cipher suites supported by Bank A.

Protocol	Cipher	Key length (bits)
SSLv3	RC4-SHA	128
TLSv1	RC4-SHA	128

Preferred server ciphers

Protocol	Cipher	Key length (bits)
SSLv3	RC4-SHA	128
TLSv1	RC4-SHA	128

- **Category 6: Insecure features and vulnerabilities**
  Score: 0
  Insecure renegotiation and client initiated renegotiations are supported.

4.2 Bank B

Total Score: 5.2

Insecure features / vulnerabilities: 1.

- **Category 1: SSL/TLS Certificate quality**
  Score: 5
  SSL/TLS enabled. Not Extended Validation certificate.

- **Category 2: Certificate information**
  Score: 10
  No errors detected

- **Category 3: Supported protocols**
  Score: 5
  TLS 1.0, SSL 3.0.

- **Category 4: Key exchange**
  Score: 5
  1024-bit certificate key length.

- **Category 5: Cipher suites**
  Score: 8.5
  The next table shows the cipher suites supported by Bank B.

Protocol	Cipher	Key length (bits)
SSLv3	AES256-SHA	256
SSLv3	AES128-SHA	128
SSLv3	RC4-SHA	128
SSLv3	RC4-MD5	128
TLSv1	AES256-SHA	256
TLSv1	AES128-SHA	128
TLSv1	RC4-SHA	128
TLSv1	RC4-MD5	128

Preferred server ciphers

Protocol	Cipher	Key length (bits)
SSLv3	AES256-SHA	256
TLSv1	AES256-SHA	256

- **Category 6: Insecure features and vulnerabilities**
  Score: 0
  The server prefers ciphers which use CBC mode. That means it is vulnerable to BEAST.

4.3 Bank C

Total Score: 5.4

Insecure features / vulnerabilities: 3.

- **Category 1: SSL/TLS Certificate quality**
  Score: 10
  SSL/TLS enabled with Extended Validation certificate.

- **Category 2: Certificate information**
  Score: 10
  No errors detected

- **Category 3: Supported protocols**
  Score: 5
  TLS 1.0, SSL 3.0.

- **Category 4: Key exchange**
  Score: 8
  2048-bit certificate key length.

- **Category 5: Cipher suites**
  Score: 4
  The next table shows the cipher suites supported by Bank C. Note that weak ciphers based on key length are marked in red.

Protocol	Cipher	Key length (bits)
SSLv3	DES-CBC3-SHA	168
SSLv3	DES-CBC-SHA	56
SSLv3	RC4-SHA	128
SSLv3	RC4-MD5	128
TLSv1	DES-CBC3-SHA	168
TLSv1	DES-CBC-SHA	56
TLSv1	RC4-SHA	128
TLSv1	RC4-MD5	128

Preferred server ciphers

Protocol	Cipher	Key length (bits)
SSLv3	DES-CBC3-SHA	168
TLSv1	DES-CBC3-SHA	168

- **Category 6: Insecure features and vulnerabilities**
  Score: 0
  The server prefers ciphers which use CBC mode. That means it is vulnerable to BEAST. In addition, it supports insecure renegotiations and client initiated renegotiations.

4.4 Bank D

Total Score: 5.8

Insecure features / vulnerabilities: 1.

- **Category 1: SSL/TLS Certificate quality**
  Score: 5
  SSL/TLS enabled. Not Extended Validation certificate.

- **Category 2: Certificate information**
  Score: 10
  No errors detected

- **Category 3: Supported protocols**
  Score: 5
  TLS 1.0, SSL 3.0.

- **Category 4: Key exchange**
  Score: 8
  2048-bit certificate key length.

- **Category 5: Cipher suites**
  Score: 8.5
  The next table shows the cipher suites supported by Bank D.

Protocol	Cipher	Key length (bits)
SSLv3	AES256-SHA	256
SSLv3	AES128-SHA	128
SSLv3	RC4-SHA	128
SSLv3	RC4-MD5	128
TLSv1	AES256-SHA	256
TLSv1	AES128-SHA	128
TLSv1	RC4-SHA	128
TLSv1	RC4-MD5	128

Preferred server ciphers

Protocol	Cipher	Key length (bits)
SSLv3	AES256-SHA	256
TLSv1	AES256-SHA	256

- **Category 6: Insecure features and vulnerabilities**
  Score: 0
  The server prefers ciphers which use CBC mode. That means it is vulnerable to BEAST.

4.5 Bank E

Total Score: 6.3

Insecure features / vulnerabilities: 3.

- **Category 1: SSL/TLS Certificate quality**
  Score: 10
  SSL/TLS enabled with Extended Validation certificate.

- **Category 2: Certificate information**
  Score: 10
  No errors detected

- **Category 3: Supported protocols**
  Score: 5
  TLS 1.0, SSL 3.0.

- **Category 4: Key exchange**
  Score: 8
  2048-bit certificate key length.

- **Category 5: Cipher suites**
  Score: 8.5
  The next table shows the cipher suites supported by Bank E.

Protocol	Cipher	Key length (bits)
SSLv3	DHE-RSA-AES256-SHA	256
SSLv3	AES256-SHA	256
SSLv3	DHE-RSA-AES128-SHA	128
SSLv3	AES128-SHA	128
SSLv3	EDH-RSA-DES-CBC3-SHA	168
SSLv3	DES-CBC3-SHA	168
SSLv3	RC4-SHA	128
SSLv3	RC4-MD5	128
TLSv1	DHE-RSA-AES256-SHA	256
TLSv1	AES256-SHA	256
TLSv1	DHE-RSA-AES128-SHA	128
TLSv1	AES128-SHA	128
TLSv1	EDH-RSA-DES-CBC3-SHA	168
TLSv1	DES-CBC3-SHA	168
TLSv1	RC4-SHA	128
TLSv1	RC4-MD5	128

Preferred server ciphers

Protocol	Cipher	Key length (bits)
SSLv3	DHE-RSA-AES256-SHA	256
TLSv1	DHE-RSA-AES256-SHA	256

- **Category 6: Insecure features and vulnerabilities**
  Score: 0
  The server prefers ciphers which use CBC mode. That means it is vulnerable to BEAST. In addition, it supports insecure renegotiations and client initiated renegotiations.

4.6 Bank F

Total Score: 6.3

Insecure features / vulnerabilities: 2.

- **Category 1: SSL/TLS Certificate quality**
  Score: 10
  SSL/TLS enabled with Extended Validation certificate.

- **Category 2: Certificate information**
  Score: 10
  No errors detected

- **Category 3: Supported protocols**
  Score: 5
  TLS 1.0, SSL 3.0.

- **Category 4: Key exchange**
  Score: 8
  2048-bit certificate key length.

- **Category 5: Cipher suites**
  Score: 8.5
  The next table shows the cipher suites supported by Bank F.

Protocol	Cipher	Key length (bits)
SSLv3	AES256-SHA	256
SSLv3	AES128-SHA	128
SSLv3	DES-CBC3-SHA	168
SSLv3	RC4-SHA	128
SSLv3	RC4-MD5	128
TLSv1	AES256-SHA	256
TLSv1	AES128-SHA	128
TLSv1	DES-CBC3-SHA	168
TLSv1	RC4-SHA	128
TLSv1	RC4-MD5	128

Preferred server ciphers

Protocol	Cipher	Key length (bits)
SSLv3	AES256-SHA	256
TLSv1	AES256-SHA	256

- **Category 6: Insecure features and vulnerabilities**
  Score: 0
  The server prefers ciphers which use CBC mode. That means it is vulnerable to BEAST.
  The server supports secure renegotiations, but client initiated renegotiations are enabled.

4.7 Bank G

Total Score: 6.3

Insecure features / vulnerabilities: 1.

- **Category 1: SSL/TLS Certificate quality**
  Score: 10
  SSL/TLS enabled with Extended Validation certificate.

- **Category 2: Certificate information**
  Score: 10
  No errors detected

- **Category 3: Supported protocols**
  Score: 5
  TLS 1.0, SSL 3.0.

- **Category 4: Key exchange**
  Score: 8
  2048-bit certificate key length.

- **Category 5: Cipher suites**
  Score: 8.5
  The next table shows the cipher suites supported by Bank G.

Protocol	Cipher	Key length (bits)
SSLv3	AES256-SHA	256
SSLv3	AES128-SHA	128
SSLv3	RC4-SHA	128
SSLv3	RC4-MD5	128
TLSv1	AES256-SHA	256
TLSv1	AES128-SHA	128
TLSv1	RC4-SHA	128
TLSv1	RC4-MD5	128

Preferred server ciphers

Protocol	Cipher	Key length (bits)
SSLv3	AES256-SHA	256
TLSv1	AES256-SHA	256

- **Category 6: Insecure features and vulnerabilities**
  Score: 0
  The server prefers ciphers which use CBC mode. That means it is vulnerable to BEAST.

4.8 Bank H

Total Score: 6.3

Insecure features / vulnerabilities: 1.

- **Category 1: SSL/TLS Certificate quality**
  Score: 10
  SSL/TLS enabled with Extended Validation certificate.

- **Category 2: Certificate information**
  Score: 10
  No errors detected

- **Category 3: Supported protocols**
  Score: 5
  TLS 1.0, SSL 3.0.

- **Category 4: Key exchange**
  Score: 8
  2048-bit certificate key length.

- **Category 5: Cipher suites**
  Score: 8.5
  The next table shows the cipher suites supported by Bank H.

Protocol	Cipher	Key length (bits)
SSLv3	AES256-SHA	256
SSLv3	AES128-SHA	128
SSLv3	RC4-SHA	128
SSLv3	RC4-MD5	128
TLSv1	AES256-SHA	256
TLSv1	AES128-SHA	128
TLSv1	RC4-SHA	128
TLSv1	RC4-MD5	128

Preferred server ciphers

Protocol	Cipher	Key length (bits)
SSLv3	AES256-SHA	256
TLSv1	AES256-SHA	256

- **Category 6: Insecure features and vulnerabilities**
  Score: 0
  The server prefers ciphers which use CBC mode. That means it is vulnerable to BEAST.

4.9 Bank I

Total Score: 8.3

Insecure features / vulnerabilities: 0.

- **Category 1: SSL/TLS Certificate quality**
  Score: 10
  SSL/TLS enabled with Extended Validation certificate.

- **Category 2: Certificate information**
  Score: 10
  No errors detected

- **Category 3: Supported protocols**
  Score: 5
  TLS 1.0, SSL 3.0.

- **Category 4: Key exchange**
  Score: 8
  2048-bit certificate key length.

- **Category 5: Cipher suites**
  Score: 8.5
  The next table shows the cipher suites supported by Bank I.

Protocol	Cipher	Key length (bits)
SSLv3	AES256-SHA	256
SSLv3	AES128-SHA	128
SSLv3	DES-CBC3-SHA	168
SSLv3	RC4-SHA	128
SSLv3	RC4-MD5	128
TLSv1	AES256-SHA	256
TLSv1	AES128-SHA	128
TLSv1	DES-CBC3-SHA	168
TLSv1	RC4-SHA	128
TLSv1	RC4-MD5	128

Preferred server ciphers

Protocol	Cipher	Key length (bits)
SSLv3	RC4-MD5	128
TLSv1	RC4-MD5	128

- **Category 6: Insecure features and vulnerabilities**
  Score: 10
  No insecure features / vulnerabilities were detected.

4.10 Bank J

Total Score: 8.3

Insecure features / vulnerabilities: 0.

- **Category 1: SSL/TLS Certificate quality**
  Score: 10
  SSL/TLS enabled with Extended Validation certificate.

- **Category 2: Certificate information**
  Score: 10
  No errors detected

- **Category 3: Supported protocols**
  Score: 5
  TLS 1.0, SSL 3.0.

- **Category 4: Key exchange**
  Score: 8
  2048-bit certificate key length.

- **Category 5: Cipher suites**
  Score: 8.5
  The next table shows the cipher suites supported by Bank J.

Protocol	Cipher	Key length (bits)
SSLv3	AES256-SHA	256
SSLv3	AES128-SHA	128
SSLv3	DES-CBC3-SHA	168
SSLv3	RC4-SHA	128
TLSv1	AES256-SHA	256
TLSv1	AES128-SHA	128
TLSv1	DES-CBC3-SHA	168
TLSv1	RC4-SHA	128

Preferred server ciphers

Protocol	Cipher	Key length (bits)
SSLv3	RC4-SHA	128
TLSv1	RC4-SHA	128

- **Category 6: Insecure features and vulnerabilities**
  Score: 10
  No insecure features / vulnerabilities were detected.

5. Conclusions

The results of this study have shown that only two banks have proper SSL/TLS implementations. As stated at the beginning of this document, SSL/TLS is not enough for being secure, but it has a lot to say. Communications between clients and servers depend on these protocols for protecting the information.

Keep in mind that security involves other factors as well, such as the internet bank application itself, server configurations, networks, etc. However, the overall security will be only as strong as the weakest link in the chain.

Organizations should minimize risks and never consider security as a one time job. They should never leave security to users, because not everyone has computers or software up to date, and not everybody knows the technology which is in the background. Plus, many attacks can be difficult to detect. Remember that these attacks happen even if users do not notice or do not report them.

Security is dynamic and new vulnerabilities are found every day. Organizations should evolve as quickly as security does, especially banks.

Appendix A - Methodology

The next example is going to describe the methodology followed for checking the implementation of SSL/TLS. We are going to use ***www.paypal.com*** as test site, so we do not reveal the identity of the online bank services while showing screenshots or commands.

- **Category 1: SSL/TLS Certificate quality**
  Certificates with Extended Validation will always show detailed information about the organization behind the website.

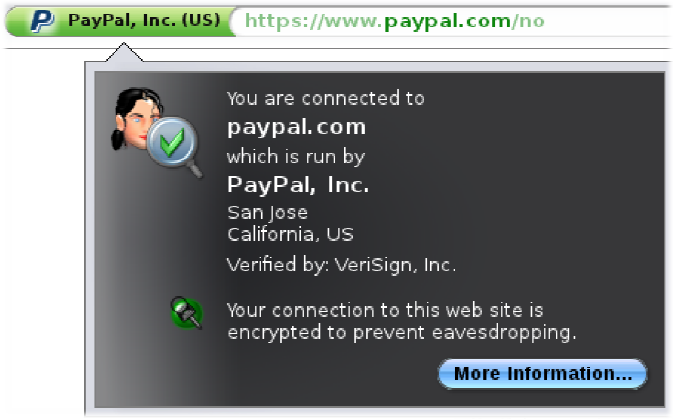


Fig. 3: EV certificate in Firefox

Studying the certificate parameters with Firefox (Certificate viewer), we find that the type is correct.

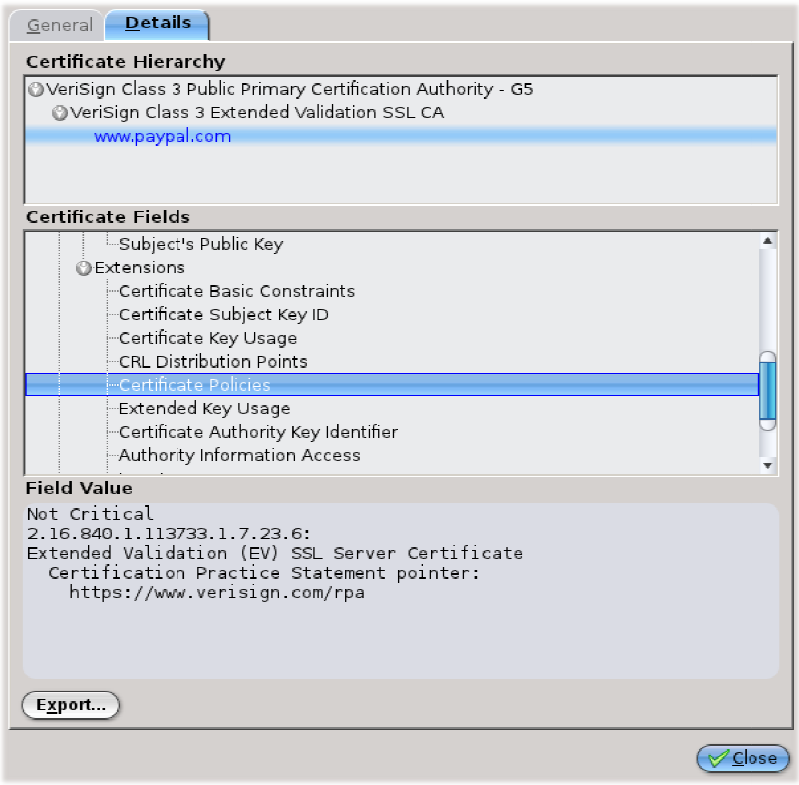


Fig. 4: Extended Validation server certificate

- **Category 2: Certificate information**
  Browsing other parameters from the certificate viewer, we can see that the certificate information matches the website. Another good symptom is that the browser has not complained about the certificate yet:

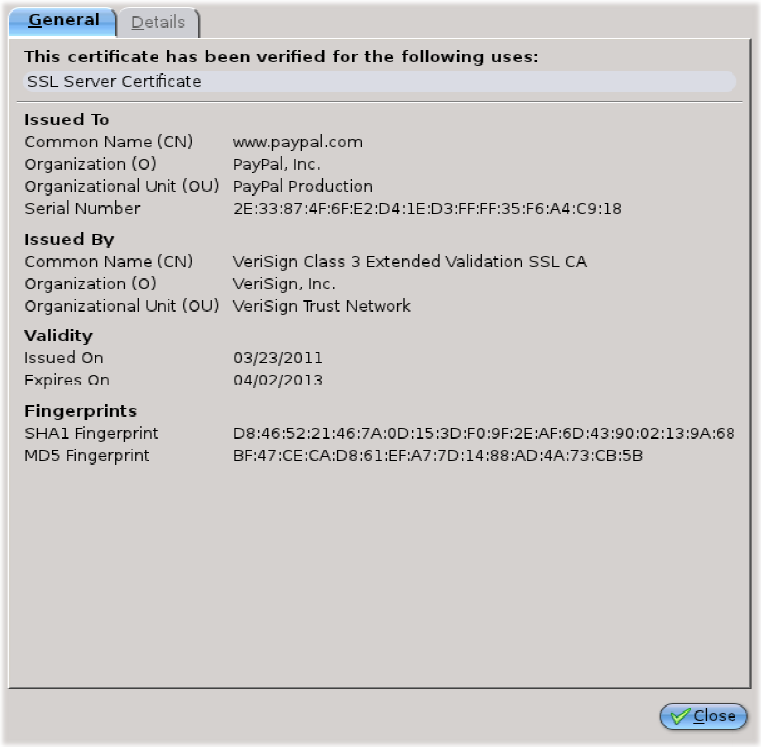


Fig. 5: The certificate has correct information regarding the common name, expiry date, trusted CA.

Fig. 6: Certificate with RSA key and signed with SHA-1

- **Category 3: Supported protocols**

In order to test what protocols are supported by the server, we will use OpenSSL and GnuTLS as a client. We will use both tools during this section.

With this command, OpenSSL will establish a SSL connection just like a regular client or a browser does. However, we will specify that only SSLv2 should be used.

Since, SSLv2 is not accepted by the server, we will get an error.

```
bash-4.1# openssl s_client -no_tls1 -no_ssl3 -connect www.paypal.com:443
CONNECTED(00000003)
6237:error:140770FC:SSL routines:SSL23_GET_SERVER_HELLO:unknown
protocol:s23_clnt.c:607:
bash-4.1#
```

We will try the same operation with SSLv3. This time, the connection will succeed because the protocol is supported by the server.

```
bash-4.1# openssl s_client -no_tls1 -connect www.paypal.com:443
CONNECTED(00000003)
depth=2 /C=US/O=VeriSign, Inc./OU=VeriSign Trust Network/OU=(c) 2006 VeriSign,
Inc. - For authorized use only/CN=VeriSign Class 3 Public Primary Certification
Authority - G5
verify error:num=20:unable to get local issuer certificate
verify return:0
---
Certificate chain
 0
s:/1.3.6.1.4.1.311.60.2.1.3=US/1.3.6.1.4.1.311.60.2.1.2=Delaware/businessCategory=
Private Organization/serialNumber=3014267/C=US/postalCode=95131-
2021/ST=California/L=San Jose/street=2211 N 1st St/O=PayPal, Inc./OU=PayPal
Production/CN=www.paypal.com
   i:/C=US/O=VeriSign, Inc./OU=VeriSign Trust Network/OU=Terms of use at
https://www.verisign.com/rpa (c)06/CN=VeriSign Class 3 Extended Validation SSL CA
 1 s:/C=US/O=VeriSign, Inc./OU=VeriSign Trust Network/OU=Terms of use at
https://www.verisign.com/rpa (c)06/CN=VeriSign Class 3 Extended Validation SSL CA
   i:/C=US/O=VeriSign, Inc./OU=VeriSign Trust Network/OU=(c) 2006 VeriSign, Inc. -
For authorized use only/CN=VeriSign Class 3 Public Primary Certification Authority
- G5
 2 s:/C=US/O=VeriSign, Inc./OU=VeriSign Trust Network/OU=(c) 2006 VeriSign, Inc. -
For authorized use only/CN=VeriSign Class 3 Public Primary Certification Authority
- G5
   i:/C=US/O=VeriSign, Inc./OU=Class 3 Public Primary Certification Authority
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIGSzCCBTOgAwIBAgIQLjOHT2/i1B7T//819qTJGDANBgkqhkiG9w0BAQUFADCB
ujELMAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMR8wHQYDVQQL
ExZWZXJpU2lnbiBUcnVzdCBOZXR3b3JrMTswOQYDVQQLEzJUZXJtcyBvZiB1c2Ug
YXQgaHR0cHM6Ly93d3cudmVyaXNpZ24uY29tL3JwYSAoYykwNjE0MDIGA1UEAxMr
VmVyaVNpZ24gQ2xhc3MgMyBFeHRlbmRlZCBWYWxpZGF0aW9uIFNTTCBDQTAeFw0x
MTAzMjMwMDAwMDBaFw0xMzA0MDEyMzU5NTlaMIIBDzETMBEGCysGAQQBgjc8AgED
EwJVUzEZMBcGCysGAQQBgjc8AgECEwhEZWxhd2FyZTEdMBsGA1UEDxMUUHJpdmF0
ZSBPcmdhbml6YXRpb24xEDAOBgNVBAUTBzMwMTQyNjcxCzAJBgNVBAYTAlVTMRMw
EQYDVQQRFAo5NTEzMS0yMDIxMRMwEQYDVQQIEwpDYWxpZm9ybmlhMREwDwYDVQQH
FAhTYW4gSm9zZTEWMBQGA1UECRQNMjIxMSBOIDFzdCBTdDEVMBMGA1UEChQMUGF5
UGFsLCBJbmMuMRowGAYDVQQLFBFQYXlQYWwgUHJvZHVjdGlvbjEXMBUGA1UEAxQO
d3d3LnBheXBhbC5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCd
szetUP2zRUbaN1vHuX9WV2mMq0IIVQ5NX2kpFCwBYc4vwW/CHiMr+dgs8lDduCfH
5uxhyRxKtJa6GElIIiP8qFB5HFWf1uUgoDPC1he4HaxUkowCnVEqjEowOy9R9Cr4
yyrmqmMEDccVsx4d3dOY2JF1FrLDHT7qH/GCBnyYw+nZJ88ci6HqnVJiNM+NX/D/
```

```
d7Y3r3V1bp7y1DaJwK/z/uMwNCC+lcM59w+nwAvLutgCW6WitFHMB+HpSsOSJlIZ
ydpj0Ox+javRR1FIdhRUFMK4wwcbD8PvULi1gM+sYsJIzP0mHDlhWRIDImG1zmy2
x7ZLp0HA5WayjI5aSn9fAgMBAAGjggHzMIIB7zAJBgNVHRMEAjAAMB0GA1UdDgQW
BBQxqt0MVbSO4lWE5aB52xc8nEq5RTALBgNVHQ8EBAMCBaAwQgYDVR0fBDswOTA3
oDWgM4YxaHR0cDovL0VWU2VjdXJlLWNybC52ZXJpc2lnbi5jb20vRVZTZWN1cmUy
MDA2LmNybDBEBgNVHSAEPTA7MDkGC2CGSAGG+EUBBxcGMCowKAYIKwYBBQUHAgEW
HGh0dHBzOi8vd3d3LnZlcmlzaWduLmNvbS9ycGEwHQYDVR0lBBYwFAYIKwYBBQUH
AwEGCCsGAQUFBwMCMB8GA1UdIwQYMBaAFPyKULqeuSVae1WFT5UAY4/pWGtDMHwG
CCsGAQUFBwEBBHAwbjAtBggrBgEFBQcwAYYhaHR0cDovL0VWU2VjdXJlLW9jc3Au
dmVyaXNpZ24uY29tMD0GCCsGAQUFBzAChjFodHRwOi8vRVZTZWN1cmUtYWlhLnZl
cmlzaWduLmNvbS9FVlNlY3VyZTIwMDYuY2VyMG4GCCsGAQUFBwEMBGIwYKFeoFww
WjBYMFYWCWltYWdlL2dpZjAhMB8wBwYFKw4DAhoEFEtruSiWBgy70FI4mymsSweL
IQUYMCYWJGh0dHA6Ly9sb2dvLnZlcmlzaWduLmNvbS92c2xvZ28xLmdpZjANBgkq
hkiG9w0BAQUFAAOCAQEAisdjAvky8ehg4A0J3ED6+yR0BU77cqtrLUKqzaLcLL/B
wuj8gErM8LLaWMGM/FJcoNEUgSkMI3/Qr1YXtXFvdqo3urqMhi/SsuUJU85Gnoxr
Vr0rWoBqOOnmcsVEgtYeusK0sQbxq5JlE1eq9xqYZrKuOuA/8JS1V7Ss1iFrtA5i
pwotaEK3k5NEJOQh9/Zm+fy1vZfUyyX+iVSlmyFHC4bzu2DlzZln3UzjBJeXoEfe
YjQyLpdUhUhuPslV1qs+Bmi6O+e6htDHvD05wUaRzk6vsPcEQ3EqsPbdpLgejb5p
9YDR12XLZeQjO1uiunCsJkDIf9/5Mqpu57pw8v1QNA==
-----END CERTIFICATE-----
subject=/1.3.6.1.4.1.311.60.2.1.3=US/1.3.6.1.4.1.311.60.2.1.2=Delaware/businessCat
egory=Private Organization/serialNumber=3014267/C=US/postalCode=95131-
2021/ST=California/L=San Jose/street=2211 N 1st St/O=PayPal, Inc./OU=PayPal
Production/CN=www.paypal.com
issuer=/C=US/O=VeriSign, Inc./OU=VeriSign Trust Network/OU=Terms of use at
https://www.verisign.com/rpa (c)06/CN=VeriSign Class 3 Extended Validation SSL CA
---
No client certificate CA names sent
---
SSL handshake has read 4547 bytes and written 461 bytes
---
New, TLSv1/SSLv3, Cipher is AES256-SHA
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol  : SSLv3
    Cipher    : AES256-SHA
    Session-ID: 187A00ECC377A4367B6957EC0A4DE1CA8CBD359AA1D025C1415D600D25C5C446
    Session-ID-ctx:
    Master-Key:
D37AFBC36005FA805B25F8E45B8D0024A096004AF4FA488E1160EC311C2C173D5F53F5DF6A506AE264
2D5A22B40B1792
    Key-Arg   : None
    Start Time: 1321045509
    Timeout   : 300 (sec)
    Verify return code: 20 (unable to get local issuer certificate)
```

The same will happen with TLS1.0. This time, we will suppress the output:

```
bash-4.1# openssl s_client -no_ssl3 -connect www.paypal.com:443
CONNECTED(00000003)
depth=2 /C=US/O=VeriSign, Inc./OU=VeriSign Trust Network/OU=(c) 2006 VeriSign,
Inc. - For authorized use only/CN=VeriSign Class 3 Public Primary Certification
Authority - G5
```

[…OUTPUT SUPPRESSED…]

```
---
SSL handshake has read 4531 bytes and written 447 bytes
---
New, TLSv1/SSLv3, Cipher is AES256-SHA
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
```

```
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol  : TLSv1
    Cipher    : AES256-SHA
    Session-ID: 187A00ECC377E4007B6957EC0A4DE2CA00266CEA35663FD0415D600D25C5C326
    Session-ID-ctx:
    Master-Key:
59AAEEF0B4B84DD085171ECBD9A39C6DD91DDB8200247EF57620EACF181C8FAC3464C46150BCCD279C
B94271F1A2DACE
    Key-Arg   : None
    Start Time: 1321045797
    Timeout   : 300 (sec)
    Verify return code: 20 (unable to get local issuer certificate)
---
```

In order to test TLS1.1 and TLS1.2, we are going to use GnuTLS from the command line (gnutls-cli). This time, we will disable TLS1.0 and SSLv3. If the server supports any other protocol, it will be able to connect. Otherwise, the protocols are not supported.

```
bash-4.1# gnutls-cli www.paypal.com --priority "NORMAL:%COMPAT:-VERS-TLS1.0:-VERS-
SSL3.0"
Resolving 'www.paypal.com'...
Connecting to '66.211.169.74:443'...
*** Fatal error: A record packet with illegal version was received.
*** Handshake has failed
GnuTLS error: A record packet with illegal version was received.
bash-4.1#
```

- **Category 4: Key exchange**
  From the output we got from the previous section, it is possible to see that the certificate uses a 2048-bit key:

```
---
New, TLSv1/SSLv3, Cipher is AES256-SHA
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol  : TLSv1
    Cipher    : AES256-SHA
    Session-ID: 187A00ECC377E4007B6957EC0A4DE2CA00266CEA35663FD0415D600D25C5C326
    Session-ID-ctx:
    Master-Key:
59AAEEF0B4B84DD085171ECBD9A39C6DD91DDB8200247EF57620EACF181C8FAC3464C46150BCCD279C
B94271F1A2DACE
    Key-Arg   : None
    Start Time: 1321045797
    Timeout   : 300 (sec)
    Verify return code: 20 (unable to get local issuer certificate)
---
```

- **Category 5: Cipher suites**
  Finding what cipher suites are supported by the server can be painful when running in manual mode (there are many possible ciphers). That's why some of automation is always handy.

Tools like sslscan[5] (written by Ian Ventura-Whiting) or the script ssl-cipher-check[6] (written by Lee Heath) will use OpenSSL / GnuTLS to establish a connection and ask the server to use a cipher suite. If the server does not support the cipher suite, it will reject it. Otherwise, it will be accepted.

Running sslscan will return the list of ciphers accepted by the server. Note that the output is very long, so we have suppressed it. It is always recommended to make a script to extract exactly what you need.

```
bash-4.1# sslscan www.paypal.com:443

[…OUTPUT SUPPRESSED – ONLY SUPPORTED CIPHERS ARE DISPLAYED…]
  Supported Server Cipher(s):

    Accepted  SSLv3  256 bits  AES256-SHA
    Accepted  SSLv3  128 bits  AES128-SHA
    Accepted  SSLv3  168 bits  DES-CBC3-SHA
    Accepted  SSLv3  128 bits  RC4-SHA
    Accepted  SSLv3  128 bits  RC4-MD5
    Accepted  TLSv1  256 bits  AES256-SHA
    Accepted  TLSv1  128 bits  AES128-SHA
    Accepted  TLSv1  168 bits  DES-CBC3-SHA
    Accepted  TLSv1  128 bits  RC4-SHA
    Accepted  TLSv1  128 bits  RC4-MD5

  Prefered Server Cipher(s):
    SSLv3  256 bits  AES256-SHA
    TLSv1  256 bits  AES256-SHA
bash-4.1#
```

Testing the cipher support with ssl-cipher-check will return a similar list. The output has been suppressed as well:

```
bash-4.1# perl ssl-cipher-check.pl -v www.paypal.com

[…OUTPUT SUPPRESSED – ONLY SUPPORTED CIPHERS ARE DISPLAYED…]

   SSLv3:RC4-MD5 - ENABLED - STRONG 128 bits
   SSLv3:DES-CBC3-SHA - ENABLED - STRONG 168 bits
   SSLv3:RC4-SHA - ENABLED - STRONG 128 bits
   SSLv3:AES128-SHA - ENABLED - STRONG 128 bits
   SSLv3:AES256-SHA - ENABLED - STRONG 256 bits
  Error 20: unable to get local issuer certificate

   TLSv1:RC4-MD5 - ENABLED - STRONG 128 bits
   TLSv1:DES-CBC3-SHA - ENABLED - STRONG 168 bits
   TLSv1:RC4-SHA - ENABLED - STRONG 128 bits
   TLSv1:AES128-SHA - ENABLED - STRONG 128 bits
   TLSv1:AES256-SHA - ENABLED - STRONG 256 bits
  Error 20: unable to get local issuer certificate


Default:
   TLSv1/SSLv3, Cipher is AES256-SHA
```

- **Category 6: Insecure features and vulnerabilities**

Insecure renegotiation and DoS due to client initiated renegotiations

Just by connecting to the server with OpenSSL as a regular client and asking the server to start a renegotiation, it will show if the server supports secure or insecure renegotiations. Also, it will determine if the server accepts client initiated renegotiations.

Note that we use grep for filtering the output from OpenSSL.

```
bash-4.1# echo R | openssl s_client -connect www.paypal.com:443 | grep -E
"Secure Renegotiation"
depth=2 /C=US/O=VeriSign, Inc./OU=VeriSign Trust Network/OU=(c) 2006 VeriSign,
Inc. - For authorized use only/CN=VeriSign Class 3 Public Primary
Certification Authority - G5
verify error:num=20:unable to get local issuer certificate
verify return:0
RENEGOTIATING
Secure Renegotiation IS NOT supported
4425:error:14094410:SSL routines:SSL3_READ_BYTES:sslv3 alert handshake
failure:s3_pkt.c:1102:SSL alert number 40
4425:error:1409E0E5:SSL routines:SSL3_WRITE_BYTES:ssl handshake
failure:s3_pkt.c:539:
bash-4.1#
```

BEAST attack

Cipher suites which use CBC mode are vulnerable to BEAST. Since the server uses AES256 (CBC mode) as preferred cipher suite, it is vulnerable to BEAST.

References

[1] CA/Browser forum. Overview of the Extended Validation SSL Certificate Vetting Process.
http://cabforum.org/vetting.html

[2] Browser Exploit Against SSL/TLS (BEAST), by Thai Duong and Juliano Rizzo.
http://www.netifera.com/research/beast/

The "BEAST" SSL/TLS issue, by Sigbjørn Vik
http://my.opera.com/securitygroup/blog/2011/09/28/the-beast-ssl-tls-issue

[3] TLS renegotiation vulnerability (CVE-2009-3555)
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3555

[4] Qualys SSL Labs - SSL Server Rating Guide (2009)
https://www.ssllabs.com/downloads/SSL_Server_Rating_Guide_2009.pdf

[5] Sslscan, by Ian Ventura-Whiting
http://www.titania.co.uk

[6] SSL Cipher Check, by Lee "MadHat" Heath
http://www.unspecific.com/ssl/