COMMERCIAL-IN-CONFIDENCE

SSL GOOD PRACTICE GUIDE

VERSION: 1.4
DATE: 23/09/2015
TASK NUMBER: SSL_Whitepaper

PREPARED FOR

Paul Docherty
Director
Portcullis Computer Security Ltd
The Grange Barn
Pike's End
Pinner
Middlesex
HA5 2EX

Tel: +44 208 868 0098
Email:

PREPARED BY

Mike W. Emery
Researcher
Portcullis Computer Security Limited
Portcullis House, 2 Century Court
Tolpits Lane, Watford
WD18 9RS
United Kingdom

Tel: +44 20 8868 0098
Email: reports@portcullis-security.com

CONTENTS

Version	Author	Role	Date	Comments
0.1	MWE	Researcher	11/12/2012	First Draft of Whitepaper Report
0.2	MRL	Peer Reviewer	28/02/2013	Second Draft of Whitepaper Report
0.3	MRL	Peer Reviewer	28/02/2013	Third Draft of Whitepaper Report
0.4	HJM	Editorial	20/05/2013	Review Of Whitepaper Report
0.5	MWE	Researcher	23/05/2013	Fourth Draft of Whitepaper Report
0.6	TMB	Head Of Research	24/05/2013	Issued for external peer review
0.7	MWE	Researcher	20/09/2013	Fifth Draft of Whitepaper Report
0.8	TMB	Head Of Research	20/09/2013	Sixth Draft of Whitepaper Report
1.0	TMB	Head Of Research	20/09/2013	Publication
1.1	MWE	Researcher	01/04/2014	Publication
1.2	MWE	Researcher	10/04/2014	Publication
1.3	AGD	Researcher	18/09/2015	Draft of Whitepaper Report
1.4	TMB	Head Of Research	23/09/2015	Publication

Please quote Task Number SSL_Whitepaper in all correspondence with Portcullis.

1 Introduction

Secure Sockets Layer (superseded by Transport Layer Security 'TLS' but still commonly referred to as 'SSL') is integral to the modern Internet. First developed in the early 1990s, it was designed to offer a means of securing otherwise unencrypted protocols and, as the Internet grew exponentially, achieved popularity as a means of securing the plaintext protocol HTTP powering the World Wide Web. Today, its use is prevalent in banking, e-commerce and even social media systems. Portcullis Computer Security Ltd is publishing this whitepaper to explain the potential issues associated with SSL, and to offer advice on how to implement SSL securely.

Revelations made public in 2013 shows the interest of nation states in compromising systems and breaking the cryptographic layer of data in transit. The correct deployment of SSL has become more important than ever, offering a greater level of assurance to an end user of the authenticity and confidentiality of their communication with a service from both internal and external threats.

2 SSL Basics

The goal of implementing an SSL solution is to ensure confidentiality, integrity and authenticity of data in transit. In order to assure these cryptographic services, client and server use a combination of asymmetric and symmetric encryption. SSL supports authentication for both endpoints, although the most conventional implementation only authenticates the server. Asymmetric encryption (or public-key cryptography depending on the terminology) is used for authentication and exchange of symmetric encryption keys. Four session keys will be established, two for providing confidentiality, and two for integrity. Confidentiality will be provided through an encryption algorithm established during the session initialization, and integrity will be provided by an integrity function such as a MAC or HMAC algorithm.

SSL relies upon trusted Certificate Authorities to sign public keys (certificates) for each SSL service. These certificates are sent to clients (and servers in some configurations) when they access a service secured by SSL and are verified using the public key of the Certificate Authority that issued the certificate. The "Subject" identified in the certificate is also checked to ensure that it matches the location that clients are actually connected to. If the certificates cannot be verified, the SSL session will not be initiated.

3 Recommendations

Whilst it is acknowledged that there are security concerns associated with the use of SSL, adhering to the guidelines that follow when configuring SSL services will ensure that any risks are minimised. Although some of the issues considered are specific to certain uses of SSL, the advice given below is applicable wherever SSL is in use.

This release of the guide recommends disabling the older SSLv2, SSLv3 and TLSv1.0 protocols. This may cause some compatibility issues on legacy systems that cannot handle the TLSv1.1 or 1.2 protocols.

3.1 Cipher Suites

When selecting a cipher suite, ensure that the symmetric keys are at least 128 bits in length and that asymmetric keys are at least 2048 bits. Use only RC4 under TLSv1 and, wherever possible, ensure that AES-GCM or AES-EAX ciphers are used. The key exchange method should be EDH (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Diffie-Hellman Ephemeral).

3.2 Configuration

SSLv2, SSLv3 and TLSv1.0 should be disabled, and the server configured to use only TLSv1.2/ TLSv1.1 connections. Compression must also be disabled and, if side channel information leakage is a concern, implementing some form of packet padding/size adjustment at the service level should be considered.

3.3 Implementation

It is important to remember to keep your SSL library patched and up-to-date.

If using OpenSSL as your library, you must be certain that it is not between versions 1.0.1 and 1.0.1f, or if it is, that it has been patched against the April 2014 Heartbleed vulnerability. If upgrading the installed version of OpenSSL is not possible, then it is recommended to recompile the existing version from source using the '-DOPENSSL_NO_HEARTBEATS' option, thereby disabling TLS heartbeats.

3.4 Certificates

Ensure that the certificate is current and signed by a trusted Certificate Authority using a suitable hashing algorithm. For more information on issues relating to certificate, please see our SSL Certificate Good Practice Guide. If you have been vulnerable to the Heartbleed vulnerability at any point, it is important to revoke your existing certificates and issue new ones.

3.5 Clients

SSL clients, including mobile applications, should make use of established SSL libraries. Certificate validation is key, as are the ciphers supported. The same rules apply here as on the server, namely that TLSv1.2 with AES-GCM or AES-EAX is used wherever possible and that SSLv2, SSLv3 and TLSv1 are disabled, and ensuring that you are using an up to date version of OpenSSL (if applicable).

Certificate pinning may be possible, such as that implemented in Chrome (http://www.imperialviolet.org/2011/05/04/pinning.html) and Internet Explorer (http://randomoracle.wordpress.com/2013/04/25/certificate-pinning-in-internet-explorer-with-emet/), which ensures that the certificate for a service is provided by a Certificate Authority authorised by the application itself. This ensures that even if a compromised or rogue Certificate Authority issues a valid certificate for a service, clients will reject it.

Examples of suitable configurations for both Apache and IIS are provided in "Sample Implementations" at the end of this document.

4 Areas of Concern

The recommendations above are based on our wealth of experience in testing SSL solutions. Examples of the issues regularly identified are:

- SSL Certificate Not Signed By A Trusted Authority
- SSLv3 Padding Oracle Information Disclosure Vulnerability ('POODLE' Attack)
- SSL Service Supports Anonymous Ciphers

This section provides an overview of the types of issues we typically test for, as well as examples of where attacks simulated during testing have been observed "in the wild".

4.1 Web Specific

Secure Cookies

If cookies issued by a web application are marked 'secure', they will only be passed through an SSL connection. Conversely, if they are not marked as such then they may be sent in plain text, rendering them vulnerable to the various security risks that entails.

Third Party Content

Third party content (which may contain JavaScript for example) may not adhere to the same standards as first party content. If the third party becomes compromised it is possible that the attacker could also compromise clients interacting with the third party content. Further to this, it is possible that information could be disclosed to the third party if the application was susceptible to Cross-Domain Referer Leakage or similar issues.

Non-SSL Content

Despite pages being loaded over SSL, some elements may not be sent securely and this can leave users with a false sense of security, which should be avoided. It is important to ensure that every resource in a secure application is sent using SSL, and this can be achieved by utilising the 'Strict Transport Security' header, which forces browsers to communicate using SSL. In addition to this, implementing a 'Content Security Policy' (CSP headers) will allow to specify the origins of our included web resources.

4.2 Root causes for a faulty SSL implementation.

- Weak Ciphers: If the encryption used for the SSL connection is not strong enough, an attack could result in the encrypted communications being stored and "cracked" at

a later date, exposing the entire contents of the communications. Implementations including inappropriate ciphers are vulnerable to attacks against weak encryption.
- Protocol Weaknesses: Design and implementation flaws are discovered by vendors and security researchers. Shortcomings such as renegotiation and possibility to downgrade an algorithm used are widespread and are cause of several well known issues. Portcullis recommends keeping an up to date SSL library in order to obtain fixes to publicly known issues in a timely manner. An improper patching policy allows protocol weaknesses to remain unfixed.

4.3 Known Attacks and SSL issues.

RC4 Bias Attacks

RC4 has known bias in its output, meaning that statistical analysis of a large number of identical plaintexts can result in partial decryption of traffic. This requires the same content to be sent a very large number of times. This is a published attack (http://www.isg.rhul.ac.uk/tls/) that currently renders it impossible to offer SSLv3 or TLSv1 without a security issue being present.

Export ciphers: 'Logjam' and 'FREAK' attacks

During the 90s very strict export regulations regarding cryptography were present in the United States of America. Due to this issue, some SSL implementations have deliberately weakened ciphers which would comply with the American export laws. All these ciphers are tagged as EXPORT ciphers or export-grade cryptography, which nowadays is considered obsolete. Solutions exist at the moment to provide higher confidentiality and integrity levels for messages, and these export regulations are no longer present.

The Logjam attack exploits the acceptance on both clients and servers of export-grade ciphers using Diffie-Hellman as their key exchange protocol. To exploit this vulnerability, the attacker performs a downgrade attack on the victim, forcing them to use a weak key exchange protocol. A "Man-In-The-Middle" can force TLS clients to use export strength DH with any server that allows DHE_EXPORT ciphers. Then, by computing the discrete log using techniques such as precomputation attacks against known primes, the attacker can learn the session key and arbitrarily read or modify the contents of the data transmitted between both parties.

The FREAK attack exploits the presence of export-grade cryptography RSA key exchange. As with the Logjam attack, the target connection is deliberately weakened by forcing the use of an export-grade key exchange. To exploit this vulnerability, an attacker downgrades a regular RSA key exchange to one that uses export-grade ephemeral RSA keys, relying on a bug in several TLS client implementations. The attacker then factors the ephemeral key to hijack future connections that use the same key. At the time the vulnerability was first published, it cost around $100 (USD) to factor the weak RSA key in 8 hours on a cluster of systems running GPUs. Once the RSA key has been factored, an attacker would be able to decrypt the RSA-protected key-exchange, recover the symmetric encryption key and decrypt all SSL traffic captured.

3DES "Meet-In-The-Middle" attack

3DES in an EDE construction (as used in SSL/TLS) is vulnerable to a "Meet-In-The-Middle" attack, which allows an attacker to perform a time/space trade-off during a brute-force attack on the key, lessening the effective security of the cipher to that of a 112-bit key. "Meet-In-The-Middle" attacks may be performed offline.

CRIME (Compression Ratio Info-leak Made Easy)

The CRIME attack, reported in September 2012 (http://arstechnica.com/security/2012/09/crime-hijacks-https-sessions/) by the same people responsible for BEAST, takes advantage of the compression feature of SSL allowing brute-force decryption of data (for example, a session cookie), enabling the attacker to hijack a 'secure' SSL session. Like its predecessor, this attack is also eminently practicable.

BREACH (Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext)

The BREACH attack is similar in concept to the CRIME attack, revealed in July 2013. However, BREACH targets HTTP response bodies (and the gzip compression many sites use to deliver them), which sees much more widespread use than the compression targeted by its CRIME predecessor.

TIME (Timing Info-leak Made Easy)

The TIME attack is another refinement of CRIME style attacks, using a timing attack on compressed data to reveal the plaintext character by character. This attack is notable, as unlike similar previous attacks, TIME attacks allow the attacker to be anywhere (rather than in a privileged network position), and use javascript on the client machine to initiate the attack.

Lucky Thirteen

The Lucky Thirteen attack is effectively a timing attack based on brute-force decryption of TLS traffic and, as such, is an advanced form of padding oracle attack. It is only effective against block ciphers and relies on encrypted information being padded by a fixed amount, typically 13 bytes, hence the name of the attack. This real-world weakness has been reported at http://www.isg.rhul.ac.uk/tls/Lucky13.html.

Perfect Forward Secrecy

If perfect forward secrecy is not properly implemented, an attacker managing to obtain a private SSL key and traffic associated with that key could decrypt all such traffic. Whilst this is not a short-term concern it is important to consider who could be capturing and potentially storing secure traffic with the aim of obtaining private keys at a later date.

This exploit hinges on the key exchange mechanism used: RSA and ADH based key exchanges do not offer perfect forward secrecy, however, EDH based key exchanges do. The weakness was reported to have been exploited as early as 2005 (http://www.indymedia.org.uk/en/2005/06/315042.html).

SSL Version 2

SSL Version 2 contains a known security flaw that renders it vulnerable to "Man-In-The-Middle" attacks and should not be used in a production environment (http://sce.uhcl.edu/yang/teaching/csci5931webSecuritySpr04/secure%20Sockets%20Layer%20%28SSL%29%20Man-in-the-middle%20Attack.htm).

STARTTLS Command Injection

In some implementations of STARTTLS it is possible to inject commands into the plaintext handshake before SSL is employed, which are then executed as part of the subsequent secured connection. Typical protocols that employ STARTTLS include SMTP, POP3, IMAP and FTP.

This vulnerability affected a number of common applications, including Postfix (http://www.postfix.org/CVE-2011-0411.html) and should have subsequently been patched.

BEAST (Browser Exploit Against SSL/TLS)

The BEAST attack, disclosed in September 2011, allows an attacker to effectively decrypt 'secure' data by injecting a known string into it, which is then used to deduce the unencrypted data from the blocks of encrypted traffic the data is divided into (which is where the weakness exists).

POODLE (Padding Oracle On Downgraded Legacy Encryption)

The Padding Oracle On Downgraded Legacy Encryption (POODLE), disclosed in October 15, 2014, is an attack on SSLv3 cipher suites using block ciphers using CBC mode. SSLv3 uses a MAC-then-encrypt construction, which doesn't authenticate the padding as it is applied on the plaintext message before padding or encryption are applied. This gives rise to a padding oracle bug, which although similar to BEAST, is more feasible to perform.

Downgrade and Renegotiation Attacks

These issues exist where a client is able to renegotiate an existing SSL connection. Renegotiating to a less secure SSL cipher than the one initially negotiated weakens the security of a connection, and repeatedly attempting to renegotiate a number of existing connections can result in a Denial of Service condition. In addition, in some SSL implementations it is possible to exploit the TLS authentication gap to facilitate "Man-In-The-Middle" attacks.

Anonymous Diffie-Hellman Key Exchange

Anonymous Diffie-Hellman key exchange ciphers are vulnerable to "Man-In-The-Middle" attacks, potentially resulting in the interception of the entire communication. Performing an anonymous key exchange will not verify client/server certificates, and does not provide any authenticity to the session.

Side Channel Attacks

Despite the information transmitted being properly encrypted, the encrypted traffic itself may still allow some information to be gathered. For example, the length of a request may allow an attacker to infer which resource has been requested, given that there are a limited number of known options. This becomes more serious in the context of forms, where it may be possible to discern some of the information supplied by a user.

Heartbleed

Whilst not being a cryptographic issue in TLS itself, this catastrophic flaw exposes a weakness in OpenSSL's implementation of TLS heartbeats, allowing an attacker to read a 64KB block of server process memory at an unpredictable address. By submitting multiple requests, the attacker is able to map out almost the entire process memory on the server which may contain session tokens, usernames and passwords, or even private keys. This flaw lay dormant in the code for two years between versions 1.0.1 and 1.0.1f, before being widely publicised (http://heartbleed.com/) and fixed in version 1.0.1g. Reports (http://arstechnica.com/security/2014/04/heartbleed-vulnerability-may-have-been-exploited-months-before-patch/) have indicated that this attack may have been exploited in the wild as far back as November 2013.

The ramifications of this weakness are far reaching. Due to the possibility that private keys may have been compromised, all services should revoke their existing certificates and procure new ones. The revocation is an important step as without it the old, potentially compromised certificate will continue to be accepted alongside the new secure one.

4.4 Certificate Authorities

Compromised/Rogue Certificate Authority

As Certificate Authorities are able to generate trusted keys for services, they present a threat to SSL services in that if a Certificate Authority was compromised, or had malicious intentions, it is possible that they could issue new certificates allowing "Man-In-The-Middle" attacks.

Certificate Authorities have been comprised a number of times in recent years, for example Comodo (http://www.pcworld.com/article/223147/google_skype_yahoo_targeted_by_rogue_comodo_ssl_certificates.html) and Diginotar (http://www.f-secure.com/weblog/archives/00002228.html).

Certificate Revocation Lists and the Online Certificate Status Protocol

Both of these mechanisms for identifying revoked certificates present security issues. The former periodically publishes lists of revoked certificates, however, invalid certificates may erroneously appear valid for some time after their revocation. The Online Certificate Status Protocol (OCSP) raises concerns as it requires a separate request to the Certificate Authority each time a SSL service is accessed to check that the certificate is valid, which provides the identities of all users of that service. The behaviour of the client when the certificate cannot be validated is also important. These issues can be mitigated by using OCSP stapling, where the request to the Certificate Authority is sent via the service being accessed, alleviating privacy and performance concerns.

4.5 Certificate Validation

Server Name Indication and Subject Alternate Names

Server Name Indication allows for multiple virtual hosts on a single server, each using their own SSL certificate. Conversely, Subject Alternate Names allows one certificate to cover multiple unrelated domains. Both of these extensions to SSL facilitate easier enumeration of associated hosts.

NULL Byte Attacks

It is possible to insert NULL characters when registering for certificates, affecting the way CommonName validation occurs and the address contacted by the Certificate Authority for registration. It is possible to generate valid certificates for any domain via these methods and Null Byte attacks have been previously exploited, as in the case of PayPal (http://www.theregister.co.uk/2009/10/05/fraudulent_paypay_certificate_published/).

Weak Hashing Algorithms

If the SSL Certificate is signed using a weak algorithm, namely MD5, it may be possible to generate a certificate where the hash collides with the original, allowing it to pass a hash check. This weakness was famously exploited by malware that used MD5 collisions to generate a fraudulent certificate that passed a hash check (http://arstechnica.com/security/2012/06/flame-malware-was-signed-by-rogue-microsoft-certificate/`flame').

4.6 Mobile Devices

SSL is often used in mobile devices to secure the communications between apps and the web services that power them, and the same standards of security that apply in a traditional web application scenario should be observed. However, a large number of android apps do not properly validate SSL certificates, resulting in "Man-In-The-Middle" attacks (http://android-ssl.org/Why_Eve_and_Mallory_Love_Android__An_Analysis_of_Android_SSL_%28In%29Security/android-ssl.org.html).

Certificate pinning is available for mobile applications, which would allow the application to authenticate the server without relying on certificate authorities. It should be noted that simply bundling the server's public SSL certificate chain inside the application and validating against the server's presented certificate chain could help to prevent "Man-In-The-Middle" attacks. It should be noted that simply relying on mobile API's to do the certificate checks is not enough as it can be easily hooked by an attacker. Also, the whole certificate chain should be checked in addition to checking the validity of the server's certificate.

4.7 Weak Private Keys

A weakness (https://security-tracker.debian.org/tracker/DSA-1571-1) existed within the Debian OpenSSL package causing it to generate predictable private keys, resulting in the decryption of traffic and "Man-In-The-Middle" attacks.

5 Sample Implementations

5.1 Apache

These settings were devised after consulting the documentation for Apache's mod_ssl extension (http://httpd.apache.org/docs/current/mod/mod_ssl.html). Please note: At a minimum, they require Apache 2.4 and OpenSSL 0.9.8.

```
SSLProtocol -ALL +TLSv1.2 +TLSv1.1
SSLHonorCipherOrder On
SSLCipherSuite ALL:!SSLv2:!SSLv3:!aNULL:!DH:!kRSA:!MD5:!PSK
SSLCompression off
Strict-Transport-Security: max-age=15768000 ; includeSubDomains
```

5.2 IIS

These settings were devised after consulting the documentation in Microsoft's Knowledge Base article 245030 (http://support.microsoft.com/kb/245030). Please Note: At a minimum, they require IIS 7 and Windows Server 2008 R2.

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\PCT 1
.0\Server]
"Enabled"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 2
.0\Server]
"Enabled"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 3
.0\Server]
"Enabled"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1
.0\Server]
"Enabled"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC2 128
/128]
"Enabled"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4 64/
128]
"Enabled"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4 56/
128]
"Enabled"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC2 56/
128]
"Enabled"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4 40/
128]
"Enabled"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC2 40/
128]
"Enabled"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\NULL]
"Enabled"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Hashes\MD5]
"Enabled"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\Triple
DES 168/168]
"Enabled"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC2 56/
56]
"Enabled"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\DES 56/
56]
"Enabled"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\KeyExchangeAlgo
rithms\Diffie-Hellman]
"Enabled"=dword:ffffffff
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\KeyExchangeAlgo
rithms\PKCS]
"Enabled"=dword:00000000
```

Appendix A About Portcullis Computer Security Limited

Since our formation in 1986 Portcullis has developed into a widely recognized and respected provider of Information Security services with the strong foundation that comes from being an independent, mature and financially stable Company.

Portcullis' revered reputation stems from our Security Testing Service, launched back in 1996, which flourished into the professional and high quality service that our Clients benefit from today. This is further endorsed by Portcullis' array of industry accreditations and the numerous accredited CHECK Team Leaders / Members and CREST Application / Infrastructure Consultants we have, which stands testament to the investment Portcullis makes in its staff, training and R&D.

Over the years Portcullis has also expanded its key portfolio of services, which now fall into 4 main disciplines - security testing, digital forensics, cyber defence and security consultancy services. The most recent addition to our range of specialist services has been the launch of our Cyber Threat Analysis and Detection Service (CTADS®) and eDisclosure Service. These specialist IT security services not only broaden Portcullis' offering to its Clients but they also enhance and compliment each other, enabling us to deliver comprehensive solutions to our Clients as a trusted security advisor and dependable security partner.

Today, Portcullis is in the proud position of employing one of the largest multidiscipline information security resources in the UK across two locations, in Watford (Hertfordshire) and Cheltenham (Gloucestershire), and has extended this capability further with international offices in San Francisco (USA) and Madrid (Spain).

With a client base encompassing Central and Local Government, Banks, Manufacturing, Charities, Telecoms, Utilities, Insurance, Retail, Healthcare, Energy, Education, Fast Moving Consumer Goods, Technology, Financial Services, Media and many international Blue Chip clients operating in EMEA and the Americas Portcullis' breadth of expertise and experience is second to none.