COMMERCIAL-IN-CONFIDENCE

SSL CERTIFICATE GOOD PRACTICE GUIDE

VERSION: 1.2
DATE: 23/09/2015
TASK NUMBER: SSL_Certificate_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

Tim Brown
Head Of Research
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

1	INTRODUCTION	3
2	BACKGROUND TO CERTIFICATE USAGE	4
3	REVIEW CRITERIA	5
4	SUMMARY	6
5	AREAS OF CONCERN	7
6	CONCLUSIONS	12
7	FURTHER DEVELOPMENTS	13
8	GLOSSARY OF TERMS	14
9	REFERENCES	15
	APPENDIX A: ABOUT PORTCULLIS COMPUTER SECURITY LIMITED	16

Version	Author	Role	Date	Comments
0.2	TMB	Head Of Research	15/01/2014	2nd Draft Whitepaper Report - Added Additional Considerations
0.3	LCP	Editorial	03/02/2014	Review of Whitepaper Report
1.0	TMB	Head Of Research	03/02/2014	Publication
1.1	FXB	Researcher	22/09/2015	Draft of Whitepaper Report
1.2	TMB	Head Of Research	23/09/2015	Publication

Please quote Task Number SSL_Certificate_Whitepaper in all correspondence with Portcullis.

1 Introduction

This document is not intended to be a definitive guide, but more of a review of the specific commonly identified issues resulting from the inappropriate deployment of SSL certificates on internal services within a corporate environment.

Whilst this document is not intended to be definitive, Portcullis believes that it should provide a high level summary of the issues that are typically present in such an environment, along with proposals as to how they can be mitigated.

2 Background To Certificate Usage

Certificates can be issued for several uses. These include authentication of users (client certificates), authentication of services (the ubiquitous SSL), secure email, VPNs and code signing, however this guide will focus only on the SSL protocol where Public Key Infrastructure (PKI) using certificates is used to secure data in transit. To do this certificates are generated from the private keys and signed by Certificate Authorities (CA) who certify that a given key is owned by the entity named within the certificate subject. Assuming that third parties trust the signing Certificate Authority, they can utilise these signatures to validate that they are indeed connecting to the intended service. In practice, trusted certificates are most often observed in the context of HTTPS-secured websites where the trust may be indicated through the use of a padlock within the URL bar, which may also be coloured green. In contrast, accessing a website that utilises a certificate signed by an untrusted Certificate Authority may result in a warning being displayed or access to the site in question being blocked. It should be noted that trust paths are bootstrapped through the installation of the certificates for trusted certificate authorities into client systems, typically as part of the operating system or web browser.

3 Review Criteria

As noted above, the Team reviewed issues that result from the inappropriate deployment of certificates within a corporate environment to identify good practice recommendations. The review draws heavily upon Portcullis' real-world experience in performing security testing in corporate environments to focus on those issues which are most frequently observed.

It is important to note that we focus primarily on the risks pertaining to certificate validation by client applications such as browsers, since it is our experience that this is the most common use case for the inappropriately deployed certificates that we identify. We have previously published a Good Practice Guide focusing on the configuration of SSL cipher suites where the risks pertain not to a user connecting to a malicious host, but rather where a malicious party can capture encrypted traffic and then perform cryptanalysis to retrieve the original plaintext.

4 Summary

In the process of reviewing the issues that result from the inappropriate deployment of SSL certificates, the Team identified the following areas of concern:

- Certificate not signed by trusted Certificate Authority
- Certificate signed using a weak hashing algorithm

The remainder of this document will expand on each of these areas of concern to provide real-world examples of attacks that could be performed, as well as appropriate mitigations which will reduce the risks that they pose.

5 Areas Of Concern

5.1 Certificate Not Signed By Trusted Certificate Authority

The most common issue seen by Portcullis in testing corporate environments is where the certificate presented by a service is not signed by a trusted Certificate Authority. Typically, this occurs where the service is configured to use a self-signed certificate, however it can also be the case that the trust path (or certificate hierarchy) includes unknown certificate authorities. In the latter case, the issuer of the certificate needs to create a Certificate Sign Request (CSR), and after some checks (from simple ones to advanced ones) the Certificate Authority will generate a signed certificate, which will be valid to use, as long as the Certificate Authority is trusted. Whilst this can be due to manual configuration (for example on internal web applications that are deemed to accept or present sensitive data), more commonly it is due to the use of embedded devices with exposed management interfaces. Whilst the use of SSL in either of these cases is unquestionably a step in the right direction, too often the lack of a suitable chain of trust inhibits users attempting to validate the certificate with which they have been presented on connection to a service. Whilst most embedded devices offer features to allow new public/private key pairs and thus certificates to be generated, a further complication highlighted by the use of an untrusted Certificate Authority is that the key material is often left unchanged from the time of installation. As such, traffic encrypted using the public key may already be compromised even before the device is ever accessed. The Little Black Box project is an attempt to catalogue such cases.

Portcullis' generic recommendation for such cases is that certificates should always be signed by a Certificate Authority that browsers belonging to normal system users will trust. Whilst this is good advice, there is an argument to be made that it does not go far enough. A typical internal corporate system may have two or three services that utilise SSL, and multiplying up the cost to account for all such systems can unfortunately lead to a requirement for hundreds of SSL certificates to be purchased. Clearly if an organisation was to have all of their certificates signed by a commercial Certificate Authority the costs of doing so would soon mount up. Instead, for such a system, an internal Certificate Authority should be utilised. For Windows only estates this can easily be achieved through the use of Windows Public Key Infrastructure Certificates where the Certificate Authority certificate can be deployed to client systems through Group Policy, however for mixed estates the process of creating a Certificate Authority and distributing the resultant certificate is more involved. As such, we will leave the process of implementing an internal Certificate Authority for now, although we hope to revisit it in more detail in a future Good Practice Guide. For now, it is recommended that implementors examine the Baseline Requirements published by the CA/Browser Forum.

At a minimum however, you should ensure that newly created certificates (signed by your internal Certificate Authority) are managed so that:

- Services are not left configured with expired certificates

- The Common Name configured within the certificate subject matches the accepted hostname of the system

Utilising an internal Certificate Authority is considered good practice as it allows discrete security monitoring where management is able to intercept all communications from the entire company even when these are encrypted through SSL and allows users to stay away from the chaotic "public" Public Key Infrastructure.

Whilst the solution as described has some clear improvements, sadly, utilising an internal Certificate Authority will only protect users providing the following holds:

- That the internal Certificate Authority is not compromised in a manner that allows attackers to generate their own trusted certificates
- That the internal Certificate Authority only signs certificates that have been appropriately generated
- That the internal Certificate Authority's certificate is correctly distributed to all client systems
- That the users are given suitable training to avoid trusting untrusted certificates by simply clicking OK when warned by their client application (web browser)

Leaving aside the first caveat which is largely outside of the scope of this review, and the second which is examined later in this document, is there anything that can be done to give assurance with respect to the third and forth caveats?

Initial analysis suggests that the third caveat is rather easy to solve, at least, in a corporate environment. The answer is to bake the internal Certificate Authority's certificate into client systems either through the process of producing a "gold" build or by utilising existing patch/ management infrastructure such as the aforementioned Group Policy which comes as part of Microsoft's Active Directory. Simply push the certificate out when the client system is provisioned or as part of routine maintenance.

With respect to the final caveat, the solution is more complicated since by default it is left to the user's discretion as to whether they trust the untrusted certificate (some, such as Chrome, even support hardcoded certificate pinning). Whilst some client applications such as browsers may make attempts to limit this, this practice is far from universal and different browser vendors and versions make it difficult to issue generic advice. Instead, we have come up with the following alternative way of implementing an internal Public Key Infrastructure that centralises control by handling all traffic through a proxy that, in turn, will reach the desired internal and external services and provide the required connectivity to its clients.

Essentially, whilst not all browsers support the blocking of untrusted certificates, they do all support proxies and, in particular, the use of PAC files as a mechanism to automatically configure browsers. Essentially, the Team identified that this can be used to force all requests for internal websites through an SSL proxy. By signing the proxy's own certificate with your internal Certificate Authority and baking the internal Certificate Authority's certificate into client systems, it then becomes possible to offer assurance that users will only connect to a single internal service that has a trusted certificate. You can then configure the SSL proxy

with a single valid Certificate Authority (yours) and apply ACLs to prevent it connecting to any website utilising a certificate signed by an unknown Certificate Authority. Furthermore, this same approach can also be taken with respect to externally accessed services that utilise SSL.

Once this is functioning, you can reconfigure client systems and your internal proxy to limit access only to websites utilising a certificate signed by your internal Certificate Authority, and configure your external SSL proxy to only include those certificates which are expected to be encountered. For example, whilst browser vendors such as Mozilla may have accepted the China Internet Network Information Center as a trusted Certificate Authority, unless you expect your users to need to access Chinese systems, you should remove it from the list of trusted certificate authorities on your external SSL proxy.

Further Considerations

Whilst the proxy solution will ensure that your users only connect to legitimate services, there are some further questions that you will need to consider:

- What will be done for isolated networks such as those that might be physically located in a data centre? Using a proxy will not be possible in all cases (for example on isolated networks)
- Are you happy to add an additional single point of failure? If the proxy fails, the entire communications affected of the corporation both internal and external fail. Usually companies prefer security solutions that can fail without affecting communications (i.e. business). Likewise, if the proxy is compromised, you provide an easy way for an attacker to access all communications in clear text and modify them. You also provide a way of bypassing all network segregations that may be in place because this proxy needs to have access to everything
- Will this raise compliance issues? Intercepting all communications within a corporation like this may have legal implications at least in some countries and may also affect PCI compliance
- How will you enforce proxy usage? The whole advantage of preventing users from accepting untrusted certificates relies on users using the proxy and not modifying the PAC file. This means that network connectivity must be blocked to users by default and managed by this proxy if they are authorised to talk to the required service. This consideration is important to ensure that there is no easy way to bypass the whole thing
- How will you deploy the certificate authority and PAC file? You will need a central mechanism to push out your internal Certificate Authority and reconfigure your users' browsers to utilise the PAC file
- Are you happy to break the end-to-end security that SSL provides? With the proxy you are breaking this so you need to ensure that the proxy itself uses a certificate signed with your internal Certificate Authority which is baked into client systems
- How are you going to handle the high performance demands? The proxy will also need to be maintained so that the rules of "who is able to access what" do not stand in the way of business requirements. These things can be achieved but it might be worth mentioning them for the costs involved
- How will you tackle remaining services that do not support the use of this proxy? Bear in mind that the solution only works by blocking all traffic that does not go through the proxy

It is key, therefore, that suitable consideration is given for how these questions can be resolved.

As an aside, in the event that you are actually using an internal Certificate Authority and that you have correctly configured your client applications to trust it, Portcullis would recommend that you make information about your Certificate Authority (including the legitimate hash for certificate you use for signing) available to your penetration testing service provider. By doing so, you may significantly reduce the number of false positives that are present in reports, allowing those services where certificate issues remain to be more easily identified. Remember, as a tester, it is impossible to discount an issue with the trust path of a certificate even if the subject and issuer looks legitimate; it is very easy for an attacker to create their own certificate where these and other non-cryptographic properties accurately reflect the values that were entered when the legitimate public/private key and associated certificate were generated.

5.2 Certificate Pinning With EMET

The Enhanced Mitigation Experience Toolkit (EMET) is a Microsoft security tool that can be used to prevent successful exploitation of security vulnerabilities on user defined applications. Data Execution Prevention, Structured Exception Handler Overwrite Protection and Address Space Layout Randomization are some of the methods used by the tool to make traditional exploitation harder.

Since version 4.0, EMET implements a new security feature known as Certificate Trust, which uses Certificate Pinning methods to detect and block Man-In-The-Middle attacks that can occur when trusted certificate authorities are compromised. In order to achieve this result, trust relationships are defined between domains and their expected SSL certificates. Users can create their custom set of certificate pinning rules, associating SSL certificate's subject names to one (one-to-one) or more (one-to-many) root certificate authorities. The user will be alerted when a difference is detected between the Root Certificate Authorities specified in the pinning rules and the Root Certificate Authority of the certificate presented to the user when accessing a website. As an example, let's assume that a pinning rule links the domain 'pinning.example.com' to the certificate authority X. When a user browses the Internet and accesses 'pinning.example.com', EMET will check if the certificate presented is signed by X. If not, the difference will be detected and reported to the user. If certificate authority Y was compromised, an attacker could start a man in the middle attack forging a valid SSL certificate for 'pinning.example.com' and signing it with the private key of Y. The browser will not alert the user because Y is within its set of trusted Certificate Authorities. Nevertheless an alert will be raised by EMET after noticing a discrepancy between the CA defined in the pinning rule for 'pinning.example.com' and the one that signed the certificate presented by the attacker.

For a centralized management of EMET, System Center Configuration Manage (SCCM) and Group Policy can be used to deploy and configure EMET in a Windows domain environment.

5.3 Certificate Signed Using A Weak Hashing Algorithm

Another issue frequently seen by Teams from Portcullis when testing corporate environments is where a certificate is signed using a weak hashing algorithm. Hashing algorithms are used in a number of places within the SSL protocol, however in this specific instance, we are talking about the hash that is included within the certificate when it is created from the private key. The purpose of this hash is to allow the client application to easily determine that none of the properties of the certificate has changed since the Certificate Authority signed it.

A number of cryptographic flaws, including hash collisions, have been identified within the hashing algorithms commonly used to verify certificates, and thus some of these have fallen out of favour. Whilst difficult, it may be computationally feasible for an attacker to generate a custom certificate that will pass the signature hashing check dependent on the hashing algorithm that was used when it was created. By doing so, an attacker may be able to impersonate the legitimate service without the client application detecting that this is the case.

For a real-world example of this, one should look no further than the Flame malware that was discovered by Iran in 2012 to have infected a significant number of their key networks. In this instance, the attacker, likely a state actor, utilised a weakness in the MD5 hashing algorithm to produce their own certificates for the purpose of signing code. The Flame binary was signed with a fraudulent certificate purportedly from the Microsoft Enforced Licensing Intermediate PCA Certificate Authority which was at the time trusted by Windows as a legitimate Certificate Authority for the signing of code. The developers of the Flame malware identified that Microsoft Terminal Server Licensing Service certificates were enabled for code signing and that they still used the weak MD5 hashing algorithm. They then produced a certificate which shared the same MD5 hash and used it to sign the malware to make it appear to have originated within Microsoft. Whilst it was already known (since 2008) that collision attacks were possible against the MD5 hashing algorithm when used in this manner, the developers of Flame were found to have implemented their own variation of the chosen-prefix collision attack.

Whilst this attack was performed by what many believe to be a nation state, the trust that can be placed in any cryptographic primitive will only degrade with time as the methods by which it can be broken become more sophisticated and accessible. The public attack documented in 2008 utilised a cluster of more than 200 PlayStations but as costs of computation fall, there may come a time where, in combination with advances in the attacks it is possible to perform this attack on a simple home PC. As a result, from around 2010, following advice from NIST, commercial certificate authorities and browser vendors moved in to eliminate the use of MD5 hashes. As such, current good practice is to utilise the SHA2-256 hash for this function.

6 Conclusions

Portcullis considers that the specific issues resulting from the inappropriate deployment of SSL certificates on internal services should be mitigated. Whilst doing so will pose challenges, Portcullis does not believe that these are insurmountable. However, it is recommended that the issues identified in this document should be considered as part of the deployment of new systems and services in line with good practice.

Based on the previous observations, Portcullis recommends that the existing processes and procedures for the deployment of SSL should be reviewed and updated as necessary. In addition, Portcullis believes that new policies and procedures may be required to cover the following aspects of the process:

- Use of an internal Certificate Authority
- Removal of any unused Certificate Authorities from client systems
- Distribution of information about your internal Certificate Authority to penetration testing service provider
- Ensuring that you only sign certificates using the SHA2-256 hashing algorithm
- Regularly auditing internal SSL certificates that are in use
- Configuration of EMET to support Certificate Trust

Whilst it is perfectly acceptable for the risks outlined in this review to be accepted, Portcullis suggests that corporate environments should be operated in line with good practice. The nature of security is that new vulnerabilities will be reported and without a satisfactory process for upkeep, new vulnerabilities may come to light. Portcullis strongly recommends that once implemented the solution should be re-tested on an annual basis for continued assurance that it remains resilient and to ensure that any of the above areas of concern are adequately resolved.

7 Further Developments

Having analysed the specific issues resulting from inappropriate deployment of SSL certificates on internal services in some depth, Portcullis would strongly recommend other areas for consideration. These include:

- Implications of the increase on the number of the Certificate Authorities that are trusted by default, and what happens when one is compromised
- How external SSL proxies can most appropriately be configured for deep packet inspection
- How key management for the internal Certificate Authority and service specific certificates ought to be performed
- How more subtle problems with certificate properties can be identified and avoided
- How to securely implement certificate revocation
- How to ensure that thick client applications are not subject to similar issues

8 Glossary Of Terms

- Certificate Authority (CA) - These entities certify that a given certificate derived from a private key is owned by the entity named within the certificate subject
- Certificate hierarchy - Representation of the trust path by which a given certificate derived from a private key can be validated as being owned by the entity named within the certificate subject
- Certificate pinning - Pinning uses an out of band mechanism to reduce the scope for an arbitrary Certificate Authority to certify a malicious (spoofed) certificate. Support varies but both Google's Chrome and Microsoft's EMET security extension can enforce pinned certificates
- Certificate Sign Request (CSR) - Process by which the entity named within the certificate subject requests that their certificate is signed by a Certificate Authority
- Common Name - Section of certificate subject that specifies the hostname for which the certificate is valid
- Cryptanlysis - Analysis of the mathematical properties of a ciphertext to gain as much information as possible about the original plaintext
- PAC file - Proxy auto-config (PAC) files define how web browsers automatically choose the appropriate proxy server for fetching a given URL
- Public Key Infrastructure (PKI) - Collective term for trusted certificate authorities, specifically their signing and revocation services

9 References

- https://code.google.com/p/littleblackbox/ - Project to catalogue known, default public/private key pairs
- https://cabforum.org/baseline-requirements-documents/ - Baseline Requirements published by the CA/Browser Forum
- http://csrc.nist.gov/groups/ST/key_mgmt/ - Project operated by NIST to define good practice in the field of key management
- http://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security - Standards based mechanism to enable certificate pinning
- https://support.microsoft.com/en-gb/kb/2458544 - EMET - Enhanced Mitigation Experience Toolkit

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.