|  |  |  |
| --- | --- | --- |
| Research article | Rethinking SSL development in appified world | Author  Fahl S, Harbach M, et al |
| Abstract | The secure socket layers (SSL) is widely used to secure data transfers on the internet. Previous studies have shown that **the state of non-browser SSL code is catastrophic across a large variety of desktop applications and libraries as well as a large selection of Android apps, leaving users vulnerable to Man-In-the-Middle (MITMAs).** To determine possible causes of SSL problems on all major appified platforms, we extended the analysis to the walled-garden ecosystem of iOS apps. Our **results show that the root causes are not simply careless developers, but also limitations and issues of the current SSL development paradigm**. Based our findings, we derive a proposal to rethink the handling of SSL using Android as a blueprint for other platforms. **Our countermeasures prevent developers from willfully or accidentally breaking SSL certificate validation, offer support for extended features such as SSL Pinning and different SSL validation infrastructures, and protect users.** We evaluated our solution against 13,500 popular Android apps and conducted developer interviews to judge the acceptance of our approach and found that our solution works well for all investigated apps and developers |
| Conclusion/key takeaways | 1. The satet of non-browser SSL code is catastrophic due to not only careless developers but also limitations and issues of the current SSL development paradigm. |
| Research article | Security Flaws Induced by CBC padding – applications to SSL, IPSEC, WTLS | Author  Serge Vaundey |
| Abstract | In many standards, e.g, SSL/TLS, IPSEC, WTLS, messages are first pre-formatted, then encrypted in CBC mode with a block cipher. Decryption needs to check if the format is valid. **Validity of the format is easily leaked from communication protocols in a chosen ciphertext attack since the received usually sends an acknowledgement or an error message**. This is a side channel. **In this paper we show various way to perform an efficient side channel attack. We discuss potential applications, extensions to other padding schemes and various ways to fix the problem** |
| Conclusion | We have shown that **several popular padding schemes which are used in order to transform block ciphers into variable-input-length encryption schemes introduce an important security flaw.** Correctness of the **plaintext format is indeed a hard core bit which easily leaks out from the communication protocol.**  It confirms that **security analysis must not be limited to the block cipher but must rather be considered within the whole environment**: as was raised by Bellovin and Borisov et al. We can really have insecure standards which use unbroken cryptographic primitives. This was already well known in the public key cryptography world. We have demonstrated that the situation of symmetric cryptography is virtual the same. |
| Research article | A framework to secure the development and auditing of SSL Pinning in Mobile Applications: the case of android devices | Author  Ramirez-Lopez F, Varela-Vaca A. |
| Abstract | The use of mobile devices has undergone rapid growth in recent years. However, on some occasions, security has been neglected when developing applications. **SSL/TLS has been used for years to secure communications although it is not a vulnerability-free protocol**. **One of the most common vulnerabilities is SSL pinning bypassing**. This paper first describes some security controls to help protect against SSL pinning bypassing. Subsequently, some existing methods for bypassing are presented and two new methods are defined. **We performed some experiments to check the use of security controls in widely used applications, and applied SSL pinning bypassing methods**. Finally, we created an applicability framework, relating the implemented security controls and the methods that are applicable. This framework provides a guideline for pentesters and app developers. |
| Conclusion | In this paper, we show how SSL/TLS implementation are vulnerable even when using SSL pinning techniques. This way, some measures have to be taken to protect an app from SSL pinning bypassing. In the implementation phase of the app, root detection, debug detection and anti-tampering measures must be implemented. Finally, the code must also be obfuscated.  The main contribution and value of this paper are the introduction of five known methods, the addition of two new methods for circumventing SSL pinning, the presentation of some solutions that developers can use to avoid this and the development of a framework of applicability. This framework can be useful as a guideline for both developers and pentesters.  Five methods are introduced:   1. Change CM in certificate 2. Insert CA in the app 3. Use SSL unpinning 4. Debugging app 5. Modify app executable   New methods to bypass SSL/TLS validation in Android devices:   1. Deals with rooted devices 2. Deals with non-rooted devices   Evaluate proposal with the design of an experiment with a set of apps:   1. Phase 1: checking whether the app includes SSL pinning and contains any security controls 2. Phase 2: checking the seven methods presented in the paper   The result: the framework help simplify the process of developing and auditing android mobile apps, and provide guideline with two main objectives:   1. Learning SSL pinning bypassing methods to check when security controls are implemented 2. Learning the security controls to implement against SSL pinning bypassing methods.   It has been demonstrated how SSL pinning can be circumvented, attacking the integrity of communications. We provide some mechanisms to prevent these attacks |
| Research article |  | Author |
| Abstract |  |
| Conclusion |  |
| Research article |  | Author |
| Abstract |  |
| Conclusion |  |