**Use of compromised or risky cryptographic algorithms**

**SEVERITY: MODERATE**

**VULN. CODE:** CRYP-WK

**AUTH. REQUIRED:**

**VULNERABILITY DESCRIPTION**

The use of cryptography generally helps to increase the level of applications security, however in cases where cryptographic algorithms are used incorrectly or where the cryptographic algorithms used suffer from known vulnerabilities, the use of this technology only feeds a security risk.

During the course of the analysis it was found that the application uses cryptographic algorithms considered compromised or not sufficiently robust by the best practices of reference. The algorithms used should not be used within the application, especially in the management of sensitive processes such as authentication. From the perspective of a threat agent, the inherent vulnerabilities of such algorithms can be used to retrieve encrypted information. It should be considered that the computational complexity required to perform analytical and brute-force crypto attacks tends to decrease significantly over time thanks to the increase in computing power available to a threat agent: both at the private level (through the use of GPUs), and at the distributed level (Cloud Computing, Distributed Cracking, etc..) today it is possible to complete attacks, within a reasonable time, even to encryption algorithms considered robust in the past.

|  |
| --- |
| Output testssl |

**Box 6 – testssl.sh output**

**SUGGESTED SOLUTIONS**

It is recommended to configure or use cryptographic algorithms that offer a high degree of robustness. It is also advisable to select authentication algorithms/schemes, which do not suffer from known vulnerabilities or criticalities from an analytical cryptographic point of view.

In general, today we recommend disabling the SSLv2/SSLv3 and TLSv1/TLSv1.1 encryption protocols, in particular the encryption protocols whose keys are less than 256 bits long. Currently it is recommended to use only the protocols TLSv1.2 and TLSv1.3. We also recommend disabling RC4-based ciphers.

**REFERENCES**

More information regarding the vulnerability and its possible solutions can be found on the following addresses:

**REF 25 – tools.ietf.org/html/rfc7568**

**REF 26 – technet.microsoft.com/it-it/library/cc731109%28v=ws.10%29.aspx**

**REF 27 – www.openssl.org/docs/apps/ciphers.html**

**REF 28 – www.schneier.com/paper-ssl.pdf**

**REF 29 – www.imperialviolet.org/2014/10/14/poodle.html  
REF 30 – www.openssl.org/~bodo/ssl-poodle.pdf  
REF 31 – tools.ietf.org/html/draft-ietf-tls-downgrade-scsv-00**

**REF 32 – msdn.microsoft.com/en-us/library/ff648653.aspx**

**REF 33 – en.wikipedia.org/wiki/Heartbleed**

**REF 34 – heartbleed.com**

**REF 35 – freakattack.com**

**REF 36 – httpd.apache.org/docs/2.2/mod/mod\_ssl.html#sslprotocol**