Bleichenbacher timing side-channel oracle in PKCS#1 v1.5 decryption

Moderate **tomato42** published **GHSA-wvcv-832q-fjg7** on Dec 18, 2020

Package

tlslite-ng (pypi)

Affected versions	Patched versions
<0.8.0-alpha39, <0.7.6	0.8.0-alpha39, 0.7.6

Description

Impact

The code that performs decryption and padding check in RSA PKCS#1 v1.5 decryption is data dependant.
In particular, code in current (as of 0.8.0-alpha38) master

tlslite-ng/tlslite/utils/rsakey.py
Lines 407 to 441 in 0812ed6

```
407     def decrypt(self, encBytes):
408         """Decrypt the passed-in bytes.
409 
410         This requires the key to have a private component.  It performs
411         PKCS1 decryption of the passed-in data.
412 
413         :type encBytes: bytes-like object
414         :param encBytes: The value which will be decrypted.
415 
416         :rtype: bytearray or None
417         :returns: A PKCS1 decryption of the passed-in data or None if
418             the data is not properly formatted.
```

and code in 0.7.5 branch

tlslite-ng/tlslite/utils/rsakey.py
Lines 394 to 425 in acdde31

```
394     def decrypt(self, encBytes):
395         """Decrypt the passed-in bytes.
396 
397         This requires the key to have a private component.  It performs
398         PKCS1 decryption of the passed-in data.
399 
400         :type encBytes: bytearray
401         :param encBytes: The value which will be decrypted.
402 
403         :rtype: bytearray or None
404         :returns: A PKCS1 decryption of the passed-in data or None if
405             the data is not properly formatted.
```

has multiple ways in which it leaks information (for one, it aborts as soon as the plaintext doesn't start with 0x00, 0x02) about the decrypted ciphertext (both the bit length of the decrypted message as well as where the first unexpected byte lays).

All TLS servers that enable RSA key exchange as well as applications that use the RSA decryption API directly are vulnerable.

All previous versions of tlslite-ng are vulnerable.

Patches

The patches to fix it are proposed in
#438
#439

Note: the patches depend on Python processing the individual bytes in side-channel free manner, this is known to not be the case: https://securitypitfalls.wordpress.com/2018/08/03/constant-time-compare-in-python/
As such, users that require side-channel resistance are recommended to use different TLS implementations, as stated in the security policy of tlslite-ng.

Workarounds

There is no way to workaround this issue.

References

https://securitypitfalls.wordpress.com/2018/08/03/constant-time-compare-in-python/

For more information

If you have any questions or comments about this advisory please open an issue in tlslite-ng.

Severity

Moderate

CVE ID

CVE-2020-26263

Weaknesses

No CWEs

Credits

tomato42