# Lucky 13 Strikes Back

`Lucky13 flush+reload AsiaCCS15`

作者介绍了在云环境中复活Lucky13攻击的方法。

# 摘要

作者在跨虚拟机的情景下，利用从与目标机运行在同一台机器上的另一台虚拟机上观察到的目标机缓存访问次数的不同，检测到只有TLS包的CBC Padding不正确时才会执行的伪函数调用，从而成功地复现了Lucky13攻击。作者相当于发现了一个原来Lucky13攻击中没有考虑到的隐蔽的旁路通道，并且这个新的旁路通道要更加准确，导致的攻击更加有效。作者对主要的密码库进行了测试，发现PolarSSL，GnuTLS和CyaSSL都受此攻击影响，而OpenSSL，Mozilla NSS和MatrixSSL的补丁则不受影响。

# 背景介绍

## Lucky13攻击

由于 MAC-then-encrypt模式与CBC加密结合（MEE-TLS-CBC）时出现的，由于MAC验证实现引入的时间差异从而可以进行Padding Oracle攻击。

假设使用TLS_RSA_WITH_AES_128_CBC_SHA密码套件，有一条待解密消息如下：

ContentType‖ Version ‖ Length ‖ Civ ‖ C1 ‖ C2 ‖ C3 ‖ C4

| ---1字节-- | --2字节-- | --2字节-- | ---------16字节 * 4---------|

服务器收到消息后，将5字节的头消息去掉，解密C1 ‖ C2 ‖ C3 ‖ C4，解密后的消息格式如下：

Message ‖ MAC ‖ Padding

在之前的设定中，为消除时间旁路信息，服务器在检查Padding后，不论Padding是否正确，都进行验证。因此服务器首先检查Padding，如果Padding合法，则去掉相应长度的Padding，否则，认为Padding长度为0。之后去掉20字节的MAC，然后验证MAC，计算如下：

HMAC（Seq[8字节] ‖ ContentType ‖ Version ‖ Length_M ‖ Message）

可见会在Message前面附加13字节的内容后进行消息验证

而HMAC的计算如下：

HMAC(K, m) = H( (K ⊕ opad) ‖ H(K ⊕ ipad) ‖ m )

若使用SHA1作为哈希函数，那么每个压缩函数处理的消息长度是64字节。`(K ⊕ opad)`长64字节，`H(K ⊕ ipad)`也是64字节，如果消息m的长度不是64的倍数，则要先对消息进行填充，填充机制如下：

m ‖ 0x80 ‖ 0x00 … 0x00 ‖ Length(8字节)

可以看出，若消息需要填充，则至少填充9字节。

下面计算HMAC需要调用的压缩函数次数：`H(K ⊕ ipad)`调用一次，若m填充后是64字节，则外层哈希函数调用3次压缩函数，也就是HMAC调用4次；否则更多。因此，当m的长度小于等于64-9=55字节时，Hmac会有4次压缩函数调用，当m长度大于55字节时，Hmac至少有5次调用。

考虑整个攻击，攻击者看到消息

`ContentType || Version || Length || Civ || C1 || C2 || C3 || C4`

将C3的最后两个字节异或某一个值，送给服务器解密，服务器解密后，`Message || MAC || Padding`共64字节，此时检查Padding，

如果Padding不合法（很大可能），则当做Padding长度为0，去掉20字节MAC，剩余44字节Message，因此Hmac的接收到m的长度为44+13=57字节，会有5次压缩函数调用

如果Padding最后一个字节为0x00，那么Message长度为45，m长度为56，仍旧有5次压缩函数调用

而如果Padding最后两个字节都为0x01的话，m的长度就刚好是55字节，只会有四次压缩函数调用

因此，将相差一次压缩函数调用时间作为时间旁路信息，就可以进行常规的Padding Oracle攻击。

## 攻击后修补情况：

修补思想：消除时间旁路信息，也就是对于不同的Padding都用一个恒定的时间处理。
但是不同的库有不同的实现方式：
引入dummy function
引入dummy data，计算最大可能压缩函数调用数目，直接执行压缩函数

# 文章提出的攻击：

## Flush+Reload

以上两种修补方法从网络时间上看，都达到了不泄露时间旁路信息的目的，但是在观察另一种旁路信息时，就会产生差别，这个旁路信息就是缓存访问时间。通过一种叫做`Flush+Reload`的技术，攻击者可以跨虚拟机获得在同一物理机器上的目标机的是否执行了某一函数的信息，从而识别出通过dummy function实现恒定时间情况，而dummy function调用时就表明是Padding不正确的时候，因此这一旁路信息可用于进行Lucky 13攻击。

## 攻击效果

- OpenSSL、Mozilla NSS和MatrixSSL是通过加入dummy data然后执行最大可能压缩函数调用数目，不会在Padding不正确时执行特别的函数，因此无法被Flush+Reload方法攻击
- GnuTLS、PolarSSL会在Padding不正确时调用一个函数额外执行压缩函数，因此可被Flush+Reload方法检测到，而CyaSSL会在Padding正确时执行一个额外的函数，因此可会遭受本文提出的攻击

## 攻击描述

- 函数识别：找到目标函数在库中的偏移，加上由于ASLR导致的用户地址空间的偏移
- 抓包、改包
- 从缓存中清掉目标函数
- 重新加载目标函数并观察加载时间：若加载时间较短，则说明目标函数被包处理程序访问过（表明Padding不正确）

这个旁路信息要比网络时间旁路信息的干扰小，更易分辨，但是要求攻击者与目标运行在同一物理机器上。然而在云环境下，这种攻击还是切实可行的。