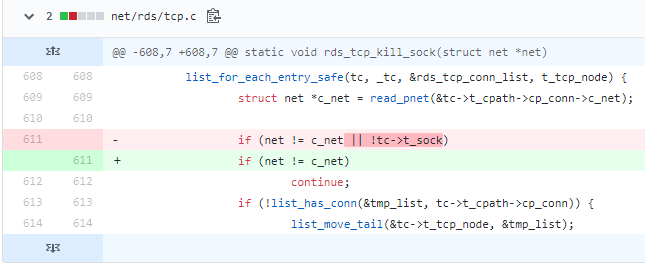
Linux kernel RCE漏洞：Rds use-after-free in rds\_tcp\_kill\_sock（CVE-2019-11815）

1、RDS协议简介：

可靠的数据报套接字（RDS）是一种高性能，低延迟可靠的无连接协议，用于投递数据报。 RDS通过在两个节点之间使用单个可靠传输来提供可靠，有序的数据报传送。 RDS可以构建在提供可靠数据报传输的任何传输上，例如TCP或Infiniband Verbs Reliable Connected连接。

2、漏洞补丁



行611，去掉了tc->t\_sock是否为空判断。不管tc->tc\_sock是否为空，都将释放tc->t\_cpath->cp\_conn对象

（1）具体看一下rds\_tcp\_kill\_sock函数。



行600，通过宏list\_for\_each\_entry\_safe循环遍历rds\_tcp\_conn\_list链表中的rds\_tcp\_connection元素放在tc中，行603，判断tc->t\_capth->cp\_conn->c\_net是否不等于net，或者是tc->t\_sock是否为空。如果其中一个条件成立，那就说明当前遍历到的tc节点不是要被释放的，直接跳过到下一个。如果不成立，行606，放到将tc->t\_tcp\_node放到tmp\_list链表中。跳出循环后，行613，再次循环遍历tmp\_list链表，调用rds\_conn\_destory释放tc->t\_cpath->cp\_conn。

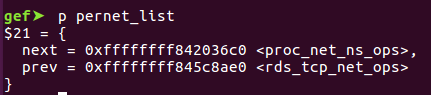
对于rds\_tcp\_kill\_sock函数的调用路径为：

Cleanup\_net-->ops\_exit\_list-->ops->exit->rds\_tcp\_exit\_net->rds->tcp->kill->sock

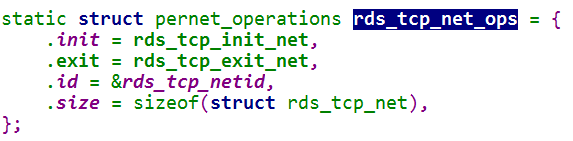
具体看cleanup\_net函数:



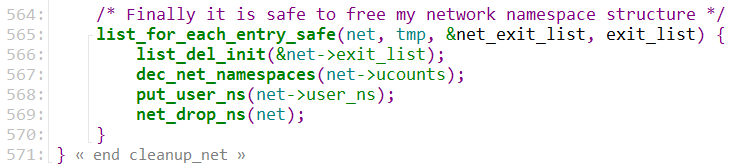
遍历pernet\_list全局链表，获取ops后再调用ops\_exit\_list进行操作。通过调试可知：



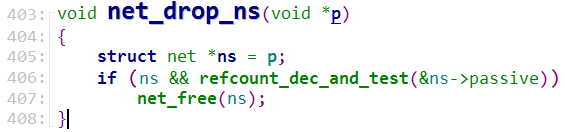
将会获取rds\_tcp\_net\_ops，rds\_tcp\_net\_ops初始化如下：



Ops->exit将调用rds\_tcp\_exit\_net函数。然后cleanup\_net函数中开始循环释放net。



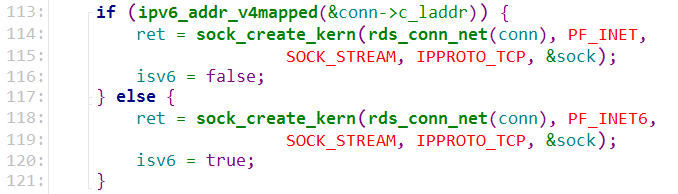
行564，遍历net\_exit\_list链表，依次调用net\_drop\_ns释放net对象。Net\_drop\_ns函数实现如下：



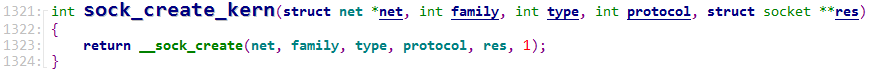
这里释放了net对象。

（二）那到底在哪里进行了释放后使用呢？

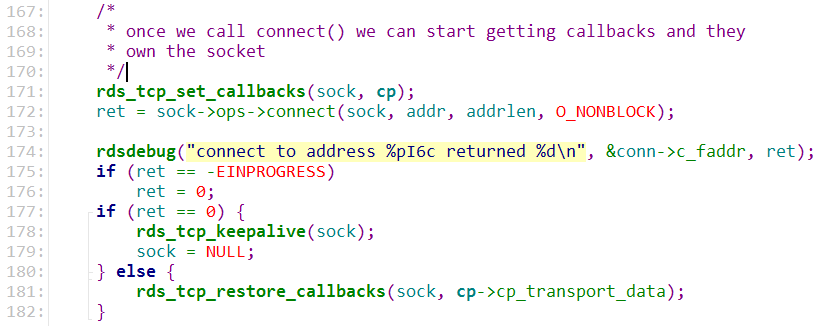
rds\_tcp\_conn\_path\_connect()函数是tds\_tcp套接字对象中.conn\_path\_connect对应的操作函数。关键实现如下：



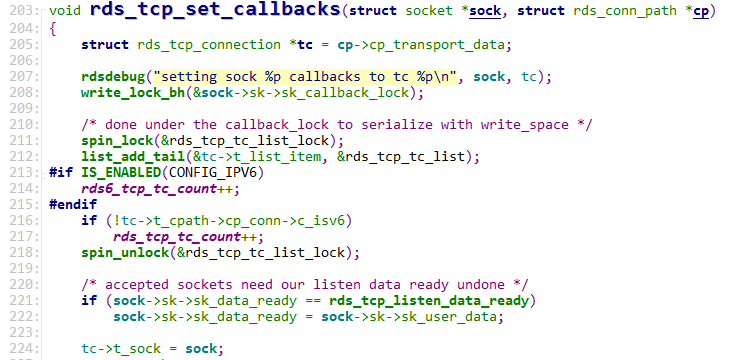
调用sock\_create\_kern()函数。这个函数实现如下：



直接调用\_\_sock\_create函数，而且第一个参数就是net对象！这里会使用net对象。成功创建sock后。开始设置tc。

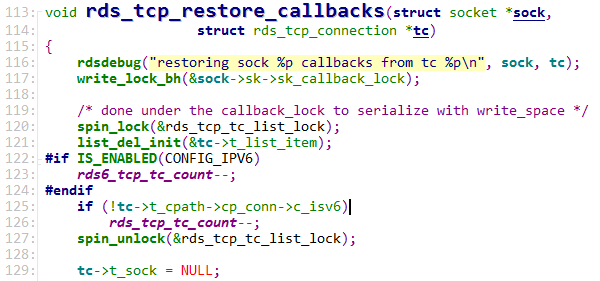


行171，先调用rds\_tcp\_set\_callbacks函数设置sock。



行224，将sock设置给tc->t\_sock，这时tc->t\_sock是非空的。

然后返回到rds\_tcp\_conn\_path\_connect，再调用sock->ops->connect函数。如果connect函数返回错误后，将有机会调用rds\_tcp\_restore\_callbacks函数。该函数关键实现如下：



行129，将tc->t\_sock设置为空。如果一直出现connect失败，那就一会直接循环设置tc->t\_sock为空。

（三）rds\_tcp\_conn\_path\_connect()函数中既可以将tc->t\_sock一直保持为空，又存在使用net对象的调用。而rds\_tcp\_kill\_tcp中因为tc->t\_t\_sock为空，而没有释放conn对象。导致一直存在connect操作（会使用到net对象）,而cleanup\_net函数中将会释放net对象。这特定的条件竞争下，会发生针对net对象的uaf。

四、安全建议：

目前该漏洞补丁已经给出，请及时升级到最新版本。

<https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=cb66ddd156203daefb8d71158036b27b0e2caf63>