# Release 15

## TR136

1. Task Background
2. 这是一个tech item，先由tech team来做POC，然后才交给我们做
3. BNP找了一个专业的安全公司来检视我们的CC2的代码，发现有两个主要的漏洞：
4. 可以使用工具来截取AMF请求，再复制这个请求，更改里面的参数信息，如OID等，再执行这个请求，从而可以查看或更改本不应该有权操作的数据
5. 同一个用户可以调用到不同模块的信息。比如一个有权限操作UMT的用户可以截获相应的操作，了解到某个操作的operation是什么，distance是什么，参数有几个。然后换一个本没有权限去操作UMT的用户去登陆CASH模块，找到一个参数个数和类型都相同的请求，并将这个AMF请求改变为UMT那边的操作请求，从而可以实现攻击跨模块的东西和本没有权限查看的信息
6. Task Solution
7. 对于漏洞a，通过缓存OID的形式来处理。即是说每一个操作都会是先查看，然后再进行CURD或者其它的操作，所以在查看的时候，可以对查出来的数据把它们的OID都缓存起来，并且参照flex，根据一些条件把对应的页面可以进行的所有操作build出来（CC2的权限操作都是放在前台去判断的），然后再在进行CRUD或者其它的操作对应的delegate方法里，检查当前操作的对象的OID是否在缓存中，如果在，则把它对应可以做那些操作都取出来，并和当前delegate方法进行的操作判断，看是不是有权限进行这个操作，如果有，则可以进行操作，如果不在，则报Invalid access right错误
8. 对于漏洞b，通过配置文件来实现。即是对CC2里面的每一个菜单定义一个menuId，而每一个页面定义一个pageId，再定义那一个page可以发起那一些请求（即是定义对应AMF的请求路径，如：com.bnp.connexis2.fit.fitDelegateImpl.uploadFile）。从而一个menu下面可以有多个page，每一个page下可以发起多少个请求。而这些配置是在一个叫做ui-configuration.xml这一文件里定义的，并且在用户登陆的时候就把这个文件读进来，从而封装成对应的对象信息，并放在缓存中。
9. 缓存的清除问题。在定义这个ui-configuration.xml文件的时候，也可以在menu这一层定义一个叫cache-limit这样一个属性。例如cache-limit=3，那么在一开始缓存的数据在走完3个界面后就会被清除。
10. Task issues
11. POC的时候，由于tech team业务不熟，或者是当初的scope没有定到那么大，所以有一些情况这个FW是没有办法解决的：
12. 当一个delegate方法有多个参数或者说多个OID作为参数的时候，由于多个参数之间有多种组合的情况，而我们的FW解决不了这个问题，对于这个情况，后来只能是对多个参数分开来检查，组合的情况，暂时不管
13. 有可能在子页面使用到上一个页面没有收集到的OID。比如Batch detail有一个duplicate的操作，在点击overview菜单的时候，调用查找数据的方法，这个时候往往是根据一定的条件来查找的，即是只是查到了一部分数据（如果全部查找出来没有必要且数据一多会有性能问题），而在Batch detail做duplicate操作的时候，可以将duplicate出来的instruction关联到一个已经存在的batch中，并且可以对这个batch根据一些条件来查找出来，而这些条件不一定和点overview菜单的时候的条件一致，并且用popup窗口查找出来的数据与在overview里查找出来的数据不一致，从而可能查找出来前面没有缓存的数据，即是没有能够判断它能不能做某些操作的信息，所以无法build它们的权限。而在duplicate操作完后，会刷新当前页面，即是拿关联到的那个OID来查找instruction的信息，因为前页并没有把这个batch也查找出来，缓存中没有这个OID，刷新页面时调用的那个方法要检查对应的VIEW权限，所以在duplicate的时候就会有报错问题。这个问题的解决方案一开始的时候是在popup查找出来的数据里全部加上view的right，因为刷新的时候只检查它的VIEW权限而已。后来觉得这种方法破坏了我们的FW，并且没有重用现有的ACLBuilder，所以它在ACLBuilder里进行了更改，对在popup页面里缺少的信息来判断的时候，先进行判空处理，从而避免空指针，而每一个进来ACLBuilder都给予VIEW权限，所以临时解决了这个问题。
14. 当两个截然不同的页面调用某个相同的delegate方法的时候，FW也解决不了。比如A页面会BUILD出来ADTO，然后B页面会BUILD出来BDTO，而A,B两个页面都会有可能跑到C页面，而初始化C页面的时候，不知道应该去检查那个DTO。这个问题现在的暂时解决方案是：如果能从参数上去决断得出来当前是从那一个页面进来的，就分别对这种情况来进行处理。比如submitInstruction这个方法有可能从modify进来，也有可能是从data input进来，但这两种情况可以从instruction中是否有oid来区分，从而可以分开来去检查OID。如果不能从参数上去判断的时候，暂时没有任何解决方案。

Summary:

这个task做得比较累，原因是所需要做的调查很多，而HK那边换了一个新的同事，大家第一合作，不熟悉对方的要求，并且他对文档的要求非常高，同时由于上一个Release质量很差，HK那边直接写了一个邮件到大老板那里。要求DESIGN上写的要和代码中完全一致。所以频繁的修改DESIGN。