安卓手机和iphone手机特殊机制导致AC-IMC双因素802.1x，每隔一段时间终端因再次认证失败下线

802.1X wlan接入 **谭奇伟** 2022-04-10 发表

组网及说明

AC-旁挂-核心交换机-IMC，本地转发， IMC配置双因素802.1x认证（EAP中继模式）：账户名+密码+IMC推送给终端的短信认证码 现场使用多台安卓和iphone手机终端，因业务需要，要求24 h不间断接入WiFi 在终端侧安装并运行了不熄屏/不休眠软件，AC侧配置了终端保活策略，以避免终端因为和AP之间一段时间没有报文交互被踢下线

告警信息

无告警信息

问题描述

故障现象为：

1. 安卓手机每隔8小时24分，即30240 s下线，需要手动连接Wifi，输入用户名+密码+随机验证码后才能上线，上线后再间隔8小时24分后再次下线； 2. iphone手机每隔12小时下线一次，同样需要手动连接Wifi，输入用户名+密码+随机验证码后才能上线，上线后再间隔12小时后再次下线；

3. 配置没有认证的clear服务模板，安卓和iphone手机终端连接后能保证24 h不间断在线；

4. 配置没有IMC推送随机验证码，只需用户名+密码的802.1x认证服务模板，安卓和iphone手机终端连接后同样能保证24 h不间断在线；

5. 以上故障现象稳定复现，安卓和iphone手机故障发生时的在线时长分别为8小时24分和12小时，时间精确至1 s以内，在IMC侧查看终端接入明细，其中安卓手机的接入时长为8小时24分钟0秒，iphone手机的接入时长是12小时0分0秒；

6. 已确认AC或者IMC上没有设置定时让终端进行重认证的配置。

过程分析

考虑到终端连接clear服务模板或者常规的（用户名+密码）的服务模板，终端连接后能避免故障发生，因此推测是到达特定时长后，某种原因触发了安卓或者iphone手机的双因素802.1x认证，且认证失败导致的终端下线，因此：

(1) 对安卓手机终端在接入SSID后8小时24分左右，对于iphone终端在接入12小时左右，在AC上提前开启如下debug命令：

debugging port-security all

debugging wlan access-security all

debugging radius packet

debugging wlan client all

debugging wlan usersec all

结果显示，终端下线原因为密码错误，安卓手机上线8小时24分后，debug捕捉到AC向IMC发起Access Request请求后，IMC回复Access Reject，原因是“incorrect password”，iphone手机在上线12小时后捕捉到相同现象。

*Dec 25 17:19:47:563 2021 BJDXHB30B1_DT_AC_01_02 RADIUS/7/PACKET:

EAP-Message=0x040a0004 Reply-Message="E63023: Incorrect password."

Message-Authenticator=0x8d6258bc380a71b0ef6d91bc1ebc13dd

*Dec 25 17:19:47:563 2021 BJDXHB30B1_DT_AC_01_02 RADIUS/7/PACKET:

03 38 00 49 70 b0 92 ae dc 18 72 d2 da b4 92 74 9b 31 4a 11 4f 06 04 0a 00 04 12 1d 45 36 33 30 32 33 3a 20 49 6e 63 6f 72 72 65 63 74 20 70 61 73 73 77 6f 72 64 2e 50 12 8d 62 58 bc 38 0a 71 b0 ef 6d 91 bc 1e bc 13 dd

由于IMC通过短信推送给手机的随机验证码由于有效期的限制（有效期较短），因此再次认证失败的原因是终端仍旧使用首次登录时输入的用户名+密码+随机验证码再次进行双因素802.1x认证，而随机验证码由于超过了有效期，被IMC判定为密码错误而导致认证失败。

(2) 为了排除随机验证码的干扰，配置了仅使用账户名+密码的普通802.1x认证进行对比测试，结果发现安卓和iphone终端同样分别在接入SSID 8小时24分和12小时后进行了再次认证，但由于没有随机验证码的存在，认证通过后终端保持继续在线。

由此可以确定是随机验证码导致终端再次认证失败后下线。

那么问题来了，在AC和IMC侧都没有进行特殊设置的前提下，如何确定这个定时进行的802.1x重认证，到底是由终端、AC还是IMC侧主动发起的呢？ 这里分别进行了两个实验：

(3) 实验一：让安卓手机（与现场不相同的品牌），iphone手机和MacBook分别接入普通的（用户名+密码） AC-IMC 802.1x认证环境，然后等待接入8小时23分左右，在AC上开启debug，发现安卓手机在8小时24分0秒准时产生了802.1x认证的radius过程完整交互，而不仅仅是终端和AC之间进行四次EAPOL-KEY交互。iphone和MacBook在此时没有发生认证或密钥交互，而在接入12小时后，iphone终端准时产生了EAP认证的完整radius交互，而MacBook终端则仅与AC进行了四次EAPOL-KEY的密钥交互，没有进行EAP中继认证中的radius交互。

(4) 实验二：现场开启安卓手机（HUWEI）的debug功能，测试发现，在手机接入8小时24分后，手机侧准时发起了EAPOL-Start：

01-11 01:02:15.598 29875 29875 D wpa_supplicant: EAPOL: txStart // 终端发送EAPOL-start，主动发起EAP认证

01-11 01:02:15.598 29875 29875 D wpa_supplicant: TX EAPOL: dst=84:**:**:**:58:11

01-11 01:02:15.613 29875 29875 D wpa_supplicant: l2_packet_receive: src=84:**:**:**:58:11 len=9

01-11 01:02:15.613 29875 29875 D wpa_supplicant: wlan0: RX EAPOL from 84:**:**:**:58:11

01-11 01:02:15.613 29875 29875 D wpa_supplicant: EAPOL: Received EAP-Packet frame

01-11 01:02:15.613 29875 29875 D wpa_supplicant: EAP: maintaining EAP method data for fast reauthentication

01-11 01:02:15.613 29875 29875 D wpa_supplicant: EAP: Received EAP-Request id=11 method=1 vendor=0 vendorMethod=0

01-11 01:02:15.613 29875 29875 I wpa_supplicant: eap auth started,set wifi mtu=1500 01-11 01:02:15.614 29875 29875 D wpa_supplicant: set wlan mtu to 1500

01-11 01:02:15.614 29875 29875 I wpa_supplicant: wlan0: CTRL-EVENT-EAP-STARTED EAP authentication started.

由此可知，终端主动向AC发送EAPOL-Start报文，再次触发了与IMC之间的802.1x双因素认证，由于随机验证码存在有效期的原因，终端重认证失败下线。

安卓终端上线8小时24分后主动发起802.1x再次认证，iphone手机上线后12小时主动发起802.1x再次认证属于两种终端的特殊机制，

使用MacOS系统的MacBook不存在这一特征。

解决方法

在IMC侧去掉双因素认证中的短信随机验证码的设置，改为普通的（账户名+密码）的802.1x认证。