CTF简单笔记

原创

不做题的浮生 于 2019-10-06 20:26:08 发布 918 收藏 15

分类专栏： CTF 文章标签： CTF 笔记

本文链接：https://blog.csdn.net/qq_23030871/article/details/102249923

CTF简单笔记：

远程开启kali桌面命令：sudo /etc/init.d/xrdp start

CTF：①Misc：杂项 ②Crypto：密码安全 ③Web安全 ④Reverse：逆向工程 ⑤Pwn

#杂项：

```
1、文件操作与隐写

2、图片隐写术

3、压缩文件处理

4、流量取证技术
```

1、文件操作与隐写

#命令就是一个对应的工具，只是这个工具没有对应的图形化界面而已

```
（1）、File：解析文件头
（2）、winhex：十六进制文件编辑器
（3）、010edit notepad++  （看头地址的前四个的值判断文件类型）
```

#文件头缺失：

```
使用场景 文件头部残缺或文件头部字段错误无法打开正常文件（file 文件名 == Data）

  使用010edit，修复文件头
```

#文件分离操作：图片种子 把很多文件融合成一张图片

```
分析文件：binwalk 文件名
分离文件：binwalk -e 文件名  遇到压缩包会进行自动解压
foremost 文件名 -o 输出目录名
使用dd实现文件的手动分离
 格式：dd if=源文件 of=目标文件名 bs=1 count=xx skip=开始分离的字节数
 参数说明：
  if=file  #输入文件名，缺省为标准输入
  of=file  #输出文件名，缺省为标准输出
  bs=bytes #同时设置读写块的大小为bytes，可以代替ibs和obs
  skip=blocks #从输入文件开头跳过blocks个块后再开始复制
  count    #总共要读取count个块
010edit：选中，右键，select，save selection
将16进制字符文件导入保存操作方法：010edit：File 中的Import Hexagon另存为....
```

#文件合并：

```
给一个MD5值计算文件的完整性
1、Linux下的文件合并
 cat 文件1 文件2 文件3 > 文件.xxx
 校验文件的md5值：(命令)md5sum 文件名
2、Windows下的文件合并
 copy /B 文件1+文件2+文件3  文件.xxx
 certutil -hashfile 文件 md5
```

#文件内容隐写：

```
查找功能
```

2、图片隐写术

#题型：

```
细微的颜色差别
GIF图多帧隐藏
 1、颜色通道隐藏
 2、不同帧图信息隐藏
 3、不同帧对比隐写
Exif信息隐藏
图片修复
 1、图片头修复
 2、图片尾修复
 3、CRC校验修复
 4、长宽高修复
最低有效位LSB隐写 8个进制位，将数据藏在最低有效位中
图片加密
 1、Stegdetect
 2、outguess
 3、Jphide
 4、F5
```

#常用工具：

1、Firework：查看隐写的图片文件
2、Exif：打开属性（给提示）Windows
  exitfool 文件名 Linux
3、Stegsolve（隐写题）：两张图片的信息基本相同
 修改打开方式：bin/javaw.exe 修改注册表：HKEY_CLASSES_ROOT\Applications\javaw.exe\shell\open\command 在中间添加 -jar （需要空格）
 用来筛选最低有效位 Analyse-->Data Extract-->三基色：RGB
4、zsteg需要在线安装（隐写） root@kali:/# gem install zsteg
 detect stegano-hidden data in PNG & BMP
 检测LSB隐写 zsteg xxx.png/bmp
5、wbstego4工具（最低有效位）：针对.bmp/.pdf
 得到的_is文件用notepad++打开
6、画图：图片格式转换
7、python脚本来处理（图片隐写LSB）
8、TweakPNG：文件头正常却无法打开文件，利用TweakPNG修改CRC
 1、CRC校验错误
 2、高度错误导致CRC错误，需要修改高度. 计算脚本
  CRC上一行之后的8个字节，前四个为宽度，后四个为高度（十六进制下的第二行前八位）
 如果修改完CRC还没有获得flag，则为高度或者宽度问题，因为高度和宽度的改变引起了
 计算出来的CRC不正确，则在运行脚本时候的CRC为原本正确的CRC值
9、Bftools：用于解密图片信息
 使用场景：在windows的cmd下，对加密过的图片文件进行解密
 格式：
 Bftools.exe decode braincopter 要解密的图片名称 -output 输出文件名
 Bftools.exe run 上一步输出的文件
10、SlientEye
 使用场景：windows下打开silentEye工具，对加密的图片进行解密
 方法：使用silentEye程序打开目标图片，点击image->decode，点击decode,可以查看隐藏文件，点击保存即可
 如果需要密码，勾选encrypted data，输入密码和确认密码，再点击decode
11、Stegdetect工具探测加密方式：针对jpg
 命令：
  stegdetect xxx.jpg
  stegdetect -s 敏感度 xxx.jpg
 1)、Jphide
 2)、Outguess
  使用场景：Stegdetect识别出来或者题目提示是outguess加密的图片
  该工具需编译使用：./configure && make && make install
  格式：outguess -r 要解密的文件 输出结果的文件名
 3)、F5
  使用场景：Stegdetect识别出来是F5加密的图片或题目提示是F5加密的图片
  进入F5-steganography_F5目录，将图片文件拷贝至该目录下，从CMD进入该目录
  格式：Java Exrtact 要解密的文件名 -p 密码
  运行结果在目录下的output.txt看到

#二维码处理：

1、使用二维码扫描工具CQR.exe打开图片，找到内容字段
2、补全二维码
3、取反

3、压缩文件处理

#伪加密

使用场景：伪加密文件
zip方法：使用winhex打开压缩文件，找到文件头第九第十个字符，将其修改为0000
  使用winhex打开文件搜索16进制504B0102，可以看到每个加密文件的头文件字段
rar方法：使用winhex打开rar文件，找到第24个字节，该字节尾数为4表示加密，0表示无加密

#暴力破解

```
ARCHPR：windows下加密过的rar文件
ziperello：windows下加密过的zip文件
```

#明文攻击

```
使用场景:已知加密的zip?rar部分文件明文内容
方法：
 1、将.txt的明文文件进行压缩，变成.zip
 2、打开archpr，攻击类型选择明文，明文文件路径选择.zip(即将明文文件不加密压缩后的文件)，
 加密的文件
 3、选择要破解的文件，点击开始，破解成功后会获得密码
有时候会跑出加密秘钥
使用该方法需要注意两个关键点：
 1、有一个明文文件，压缩后CRC值与加密压缩包中的文件一致
 2、明文文件的压缩算法需要与加密压缩文件的压缩算法一样（store是默认算法）
```

#RAR文件格式

```
有时候给出的RAR文件的头部各个字块会故意给错导致无法识别
HEAD_CRC    2字节    所有块或块部分的CRC
HEAD_TYPE   1字节    块类型
HEAD_FLAGS  2字节    块标记
HEAD_SIZE   2字节    块大小 #如果快标记的第一位被置1的话，还存在：
 ADD_SIZE 4字节 可选结构-增加块大小
那么，文件块的第三个字节为块类型，也叫头类型
头类型是0x72表示是标记块
头类型是0x73表示是压缩文件头块
头类型是0x74表示是文件头块
头类型是0x75表示是注释快
……
怎么看开始，看前面的到哪里结束
```

4、流量取证技术

```
通常比赛中会提供一个包含流量数据的PCAP文件，有时候需要进行修复或重构传输文件后，再进行分析
 总体把握：
  协议分级
  端点统计
 过滤筛选
  过滤语法
  Host，Protocol，contains，特征值
 发现异常
  特殊字符串
  协议某字段
  flag位于服务器中
 数据提取
  字符串取
  文件提取
 流量分析可以概括为三个方向
  1、流量包修复
  2、协议分析
  3、数据提取
```

#常用的过滤命令：

```
1、过滤IP，如源IP或者目标x.x.x.x
 ip.src eq x.x.x.x or ip.dst eq x.x.x.x
2、过滤端口
 tcp.port eq 80 or udp.port eq 80
 tcp.dstport == 80 只显tcp协议的目标端口为80
 tcp.srcport == 80 只显tcp协议的源端口为80
 tcp.port >= 1 and tcp.port <= 80
3、过滤协议
 tcp/udp/arp/icmp/http/ftp/dns/ip……
4、过滤MAC
 eth.dst == xx.xx.xx.xx 过滤目标MAC
5、包长度过滤
 udp.length == xx 这个长度是指udp本身固定长度8加上udp下面那块数据包之和
 tcp.len >= 7 指的是ip数据包（tcp下面那块数据），不包括tcp本身
 ip.len == 94 除了以太网头固定长度14，其他都算是ip.len,即从ip本身到最后
 frame.len == 119 整个数据包长度，从eth开始到最后
6、http模式过滤
 http.request.method == "GET"
 http.request.method == "POST"
 http.request.uri == "/img/logo-edu.gif"
 http contains "GET"
 http contains "HTTP/1."
 http.request.method == "GET" && http
 http contains "flag"
 http contains "key"
 tcp contains "flag"
```

#WireShark协议分析：

```
英文版：Statistics -> Protocol Hierarchy
中文版：统计 -> 协议分级
根据数据包特征进行筛选
 右键-->作为过滤器应用-->选中
流汇聚
 右键-->追踪流-->....
 在关注的http数据包或tcp数据包中选择流汇聚，可以将http流或tcp流汇聚或还原成数据，在弹出的框中可以看到数据内容
 常见的http流关键内容：
  1、html中直接包含重要信息
  2、上传或下载文件内容，通常包含文件名，hash值等关键信息
 常用POST请求上传
  3、一句话木马，POST请求，内容包含eval，内容使用base64加密
```

#数据提取：

```
使用wireshark自动提取通过http传输的文件内容
 文件-->导出对象（wireshark自带的文件分离功能）
手动提取文件内容
 右键-->导出分组字节流 或者 点击菜单栏 文件-->导出分组字节流，快捷键ctrl+h
 在弹出的框中将文件保存成二进制文件
```

#无线wifi流量包：

```
kali系统下：aircrack-ng工具进行wifi密码破解
 1、用aircrack-ng检查cap包：aircrack-ng xxx.cap
  bssid：MAC地址    ESSID：wifi名字
 2、用aircrack-ng跑字典进行握手包破解：aircrack-ng xxx.cap -w pass.txt
```

#USB流量：鼠标流量和键盘流量

```
键盘流量：
 右键leftover capture data --> 应用为列..
  在数据包中体现，apply as column
 键盘数据包的数据长度为8个字节，击键信息集中在第3个字节
 Leftover Capture Data中值与具体按键的对应关系，可以参考
  http://www.usb.org/developers/hidpage/Hut1_12v2.pdf
  python脚本
 提出方式
  1、文件-->导出分组解析结果-->保存为一个csv文件（表格） 不好用
  2、使用wireshark提供的命令行工具tshark，可以将Leftover Captuer Data数据单独复制出来
   tshark -r 流量包的名字 -T fields -e usb.capdata>目的文件名.txt
鼠标流量：
 鼠标流量的提取方式和键盘的提取方式相同
 鼠标数据包的数据长度为4个字节
 第一个字节代表按键，当取0x00，代表没有按键；为0x01时，代表按左键；为0x02时，代表当前按键为右键
 第二个字节代表左右偏移
  当值为正时，代表右移多少像素
  当值为负时，代表左移多少像素
 同理，第三个字节代表上下偏移.
 脚本
 用gnuplot工具把坐标画出来
  命令：1、gnuplot
      2、plot"xy.txt"
```

脚本：

https://github.com/WangYihang/UsbMiceDataHacker//鼠标流量

https://github.com/WangYihang/UsbKeyboardDataHacker //键盘流量

摘自王一航的github

#Https流量包文件分析：

```
一般会给key
导入key  Https == Http + TLS
英文版：Preference-->Protocols-->SSL-->edit RSA keys list
中文版：编辑-->首选项-->Protocols-->SSL-->Edit RSA keys list
```