网络安全-CTF取证方法大汇总，建议收藏！

原创

Coisini、 于 2019-05-30 19:53:55 发布 5341 收藏 28

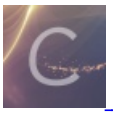
分类专栏： 安全开发 实战篇 文章标签： 网络安全 必须收藏 CTF取证方法

本文链接：https://blog.csdn.net/kclax/article/details/90704289

安全开发 同时被 2 个专栏收录

155 篇文章 10 订阅

订阅专栏



实战篇

168 篇文章 3 订阅

订阅专栏

站在巨人的肩头才会看见更远的世界，这是一篇来自技术牛人的神总结，运用多年实战经验总结的CTF取证方法，全面细致，通俗易懂，掌握了这个技能定会让你在CTF路上少走很多弯路，不看真的会后悔！

本篇文章大约6千字，阅读时间需20分钟，希望大家耐心看完！

取证

在CTF（Capture The Flag，中文一般译作夺旗赛，在网络安全领域中指的是网络安全技术人员之间进行技术竞技的一种比赛形式）中，取证的挑战可能包括文件格式分析，隐写术，内存转储分析或网络数据包捕获分析等。检查和处理静态数据文件，而不是可执行程序或远程服务器的隐藏信息，这其中任何挑战都可以被认为一个取证挑战，除非它涉及密码学，在这种情况下它可能属于Crypto类别。

取证是一个非常宽泛的CTF类别概念，因为这个叫法不能很好对应到安全行业中的特定工作角色，尽管一些挑战模拟了事件响应（IR）中看到的任务种类，即使在IR工作中，计算机取证通常是执法人员为了寻求证据数据和归属而做的事，很少有人是为了预防被攻击或仅仅是为了恢复系统完整性的商业行为。

与大多数CTF取证挑战不同，现实世界的计算机取证任务几乎不会涉及解开巧妙编码的字节、隐藏数据、文件夹中的mastroshka文件或其他复杂的问题。通常人们不会通过仔细重新组装损坏的PNG文件来破坏刑事案件的分析，揭示QR码的照片，该QR码解码为包含NES rom的zip存档的密码。

但是，现实世界的取证通常要求取证人找到间接的恶意证据，比如系统中的攻击者的痕迹或内部威胁行为的痕迹。实际的计算机取证主要在于对日志，内存，文件系统或注册表以及相关的文件和文件系统元数据中找到犯罪的线索。此外，网络（数据包捕获）取证更多的涉及元数据分析，而不是内容分析，因为现在大多数网络会话都在端点之间进行TLS加密。

之所以会在CTF进行这个脱离实践的挑战游戏，就是为了提高此项赛事的观赏性和难度。

为了解决这个挑战，你需要具备以下三个基本技能：

1、了解脚本语言，例如Python；

2、知道如何处理该语言的二进制数据（字节级操作）；

3、识别格式、协议、结构和编码。

当然，像大多数CTF一样，理想的环境是一个Linux系统。如果你愿意使用Windows系统也行，不过不建议用Mac系统。

在Python中处理二进制数据

假设你已经选择了一些Python编程，你仍然可能不知道如何有效的处理二进制数据。像C这样的低级语言可能更适合这个任务。

以下是使用Python中的二进制数据的一些示例。

以二进制方式写入或读取文件：

f = open('Reverseit', "rb")s = f.read()f.close()f = open

('ItsReversed', "wb")f.write(s[::-1])f.close()bytearray类型是一个可变的字节序列，可以在Python 2和3中使用：

你还可以从十六进制表示的Unicode字符串中定义一个bytearray：

bytearray类型具有与Python str或list大致相同的方便方法split()， insert()， reverse()，extend()，pop()，remove()等。

将一个文件读入一个字母进行处理：

data = bytearray(open('challenge.png', 'rb').read())

常见取证概念和工具

文件格式识别和魔术字节

几乎所有的取证挑战都将涉及一个文件，通常会在没有任何上下文的环境中让你猜测这个文件是干什么的。 Filetype作为用户熟知的概念，历史上已被指定为filetype扩展名，例如，MarkDown的readme.md，MIME类型，如Web上Content-Type头文件，或者存储在文件系统中的元数据（as在MacOS中使用mdls命令）。在CTF中，比赛的一部分就是使用启发式方法来自己识别文件。

用于在UNIX上识别文件类型的传统启发式是libmagic，它是用于识别所谓的"魔术数字"或"魔术字节"的库，它是文件类型头文件中的唯一标识标记字节。 libmagic 库是文件命令的基础。

$ file screenshot.png screenshot.png: PNG image data, 1920 x 1080, 8-bit/color RGBA, non-interlaced请记住，启发式和使用它们的工具很容易被混淆。因为在比赛中，你可能会看到一个被故意制作来误导的文件。另外，如果一个文件包含一个嵌入其中的其他文件，那么文件命令只能识别包含的文件类型。在这些情况下，你可能需要更仔细的检查文件内容。

TrID是更复杂的文件版本，虽然它是封闭源代码，但它是免费的，可以跨平台运行。它还使用识别启发式，又具有确定的百分比。它的优点是其较大的已知文件类型，包括现实世界中看到的许多专有和晦涩的格式。

File Carving

File Carving是数字取证研究中频繁使用的一种文件恢复技术，它从表面上无差别的二进制数据集，即原始磁盘映象中提取（或者说恢复）文件，而不利用磁盘映象的文件系统类型。这个过程就如同在一块光滑的石头上雕刻出许多图案一样，故称之为"Carving"（雕刻）。

scalpel，现在是SleuthKit的一部分，SleuthKit是File Carving的另一种工具，以前称为Foremost。

要手动提取文件的子部分，可以使用dd命令。许多十六进制编辑器还提供复制字节并将其粘贴为新文件的功能，因此你不需要研究偏移量。

以下是使用dd从文件偏移量1335205处进行File Carving的示例，长度为40668937字节：

$ dd if=./file_with_a_file_in_it.xxx of=./extracted_file.xxx bs=1 skip=1335205 count=40668937尽管上述工具应该足够了，但在某些情况下，你可能还需要使用Python编程方式提取文件的子部分，使用Python的re或regex模块来识别魔术字节，以及zlib模块来提取zlib流。

初始分析

在搜索文件中的所有纯文本字符串时要用到一些有用的命令字符串，比如，grep是用来搜索特定的字符串，bgrep是用来搜索非文本数据模式和hexdump。

以下是使用字符串查找ASCII字符串和文件偏移量的示例：

$ strings -o screenshot.png 12 IHDR 36 $iCCPICC Profile 88 U2EI4HB… 767787 IENDUnicode字符串（如果是UTF-8）可能会显示在搜索ASCII字符串中，但是要搜索其他编码，请参阅-e标志的文档。请注意字符串会存在许多编码陷阱。

以下是在PNG文件中搜索PNG魔术字节的示例：

$ bgrep 89504e47 screenshot.png screenshot.png: 00000000以下是使用hexdump的例子：

hexdump的优点不在于它是最好的十六进制编辑器，而是可以将其他命令的直接输出管道转换为hexdump，或将其输出管道输出到grep又或者使用格式字符串对其输出格式化。

以下是使用hexdump格式字符串将文件的前50个字节作为一个64位整数以十六进制输出：

hexdump命令的其他用途

二进制文本编码

二进制就是1和0，但通常作为文本传输，传输101010101的实际序列将是浪费的，因此首先要使用各种方法对数据进行编码。这就是所谓的二进制到文本编码。当对上述文件进行字符串分析时，你可能会发现编码为文本字符串的二进制数据。

前面已经说过取证最重要的是能够识别编码，有一些可以一目了然地识别，例如Base64编码的内容，可以通过其字母数字字符集和其“=”填充后缀识别。网上有很多Base64编码器或者可以使用base64命令：

$ echo aGVsbG8gd29ybGQh | base64 -Dhello world!ASCII编码的十六进制也可以通过其字符集（0-9，A-F）来标识，ASCII字符本身占用了一定范围的字节（0x00到0x7f，见man ascii），所以如果你正在检查一个文件并找到一个像68 65 6c 6c 6f 20 77 6f 72 6c 64 21这样的字符串，那么请注意这就是ASCII码。在技术上，它是以ASCII（二进制）编码为十六进制编码的文本。

目前已经有几个网站为各种编码提供在线编码解码器，对于本地的转换器，请尝试使用xxd命令。

以下是使用xxd执行text-as-ascii-to-hex编码的示例：

$ echo hello world! | xxd -p68656c6c6f20776f726c64210a

普通文件格式

前面介绍了通用取证任务的基本概念和工具，接下来将更具体介绍一些有挑战的取证方法以及用于分析每个方法中的推荐工具。

对每种可能的数据格式做准备是不可能实现的，但在CTF中有一些是特别受欢迎的，例如你需要准备以下工具：

· 归档文件（ZIP，TGZ）

· 图像文件格式（JPG，GIF，BMP，PNG）

· 文件系统映像（特别是EXT4）

· 数据包捕获（PCAP，PCAPNG）

· 内存转储

· PDF

· 视频（特别是MP4）或音频（尤其是WAV，MP3）

· Microsoft的Office格式（RTF，OLE，OOXML）

分析文件格式时，文件格式感知（a.k.a.模板化）十六进制编辑器，如010编辑器，一个被称为Kaitai的开源产品，此外，Wireshark网络协议分析仪的一个不太知名的功能是能够分析某些媒体文件格式，如GIF，JPG和PNG。然而，所有这些工具都是用于分析未损坏和格式良好的文件，许多CTF挑战会让参赛者根据丢失或清零的格式字段等重建文件的任务。

Zip文件的分析

大多数CTF挑战都包含在zip，7z，rar，tar或tgz文件中，但只有在取证挑战中，存档容器文件才是挑战的一部分。通常，挑战的目标是从损坏的存档中提取文件或者在未使用的字段中找到嵌入的数据（常见的取证挑战），而zip文件是目前最常见的。

有一些zip文件的命令行工具将有助于我们的分析：

· unzip通常会输出有关zip无法解压原因的有用信息。

· zipdetails -v将提供有关格式各个字段中存在的值的深入信息。

· zipinfo列出了有关zip文件内容的信息，而不提取它。

· zip -F input.zip –out output.zip和zip -FF input.zip –out output.zip尝试修复损坏的zip文件。

· fcrackzip brute-force会尝试猜测一个密码小于7个字符的zip密码。

Zip文件格式规范

与密码保护的RAR或7z文件不同，zip文件的一个重要的安全相关注意事项是它们不加密其包含压缩文件的文件名和原始文件大小。

关于zip破解的另一个注意事项是，如果你有加密zip中压缩的任何一个文件的未加密或未压缩副本，你可以执行明文攻击并破解zip。用于密码保护zip文件的新方案（使用AES-256，而不是“ZipCrypto”）并没有这个弱点。

图像文件格式分析

图像文件格式是复杂的，会以许多方式被攻击，这就使得挑战涉及元数据字段，有损和无损压缩，校验和隐写术或视觉数据编码方案。

简单的初步分析步骤是使用exiftool来检查图像文件的元数据字段，如果图像文件的挑战被滥用于CTF，则其EXIF可能会识别原始图像尺寸，相机类型，嵌入的缩略图，注释和版权字符串，GPS位置坐标等。

exiftool输出示例：

特别是，PNG文件在CTF挑战中很受欢迎，可能是因为它们的无损压缩适用于隐藏图像中的非可视数据。可以在Wireshark中解析PNG文件，要验证是否正确或尝试修复损坏的PNG，你可以使用pngcheck。如果你需要深入挖掘PNG，pngtools软件包可能会有用。

利用隐写术在一个不相关的数据中隐藏一些秘密数据的做法在现实中非常罕见，所以在CTF中的另一个受欢迎的取证挑战就是利用隐写术来破解任何类型的数据。隐写术的挑战难点在于，提取隐藏的消息不仅需要使用隐写术的检测，而且还需要用于嵌入隐藏消息准确的隐写工具。如果我们怀疑某文件使用了隐写术，我们至少要检查它是否存在。 Stegsolve通常用于将各种隐写术技术应用于图像文件，以尝试检测和提取隐藏的数据，你也可以试试zsteg。

Gimp提供了改变图像文件的视觉数据的能力，曾经有CTF挑战者使用改变的色相、饱和度、亮度值和颜色通道来隐藏秘密信息。Gimp还有助于确认是否真的是一个图像文件，例如，当你从内存转储或其他地方的显示缓冲区恢复图像数据，但是缺少指定像素格式的图像文件头，图像高度和宽度等，Gimp会将你的数据作为原始图像数据打开，并尝试使用不同的设置。

ImageMagick工具可以合并到脚本中，让你能够快速识别，调整大小，裁剪，修改，转换或以其他方式处理图像文件。它也可以使用比较功能找到两个看似相同的图像之间的视觉和数据差异。

如果你正在编写自定义图像文件格式解析器，请导入Python图像库（PIL），也称为Pillow。它可以让你从动画GIF中提取帧，甚至可以从JPG中提取单个像素，它支持大多数主要图像文件的格式。

如果使用QR码（2D条形码），还可以查看Python的qrtools模块。你可以使用少于5行的Python来解码QR码的图像。当然，如果你只需要解码一个QR码，任何智能手机都可以。

文件系统分析

计算机取证中的分类是指迅速缩小查看内容的能力，以下是安装CD-ROM文件系统映像的示例：

mkdir /mnt/challengemount -t iso9660 challengefile /mnt/challenge一旦安装了文件系统，tree命令会帮你快速查看目录结构，看看是否有任何东西可以进一步分析。

你可能没有在可视文件系统中查找文件，但很有可能是一个隐藏的卷，未分配的空间（不是任何分区的一部分的磁盘空间），已删除的文件或非文件文件系统结构，

如http://www.nirsoft.net/utils/alternate_data_streams.html。对于EXT3和EXT4文件系统，你可以尝试使用extenelete查找已删除的文件。对于其他的，比如TestDisk，恢复丢失的分区表，修复损坏的分区，取消删除FAT或NTFS上的文件等。

Sleuth Kit及其附带的基于Web的用户界面“Autopsy”是用于文件系统分析的强大开源工具包，可以帮助你在整个磁盘映像中搜索关键字或查看未分配的空间等任务。

嵌入式设备文件系统是独有的类别，专门针对固定功能的低资源环境，可以压缩，单文件或只读。 Squashfs是嵌入式设备文件系统的一种流行实现工具。对于嵌入式设备的图像，你最好使用固件模块或二进制解析器进行分析。

数据包捕获（PCAP）文件分析

CTF挑战之一就是提供一个表示一些网络流量的PCAP文件，并挑战播放器恢复或重构传输的文件或传输的秘密。

要进行初步分析，请使用Wireshark的统计信息或对话视图或其capinfos命令对数据包进行高级视图。Wireshark及其命令行版本tshark都支持使用“过滤器”功能，如果你掌握语法，则可以快速减少分析范围。

还有一个名为PacketTotal的在线服务，你可以提交高达50MB的PCAP文件，并在安全连接上以图形方式显示连接的时间线和SSL元数据。此外，它将突出显示文件传输并显示任何“可疑”活动。如果你已经知道要搜索的内容，可以使用ngrep进行grep搜索。

正如File Carving一样，识别和提取文件中嵌入的文件，而“分组式的File Carving”则是用于描述从数据包捕获中提取文件的术语，它是用于从捕获的数据包中恢复文件的昂贵商业工具，但是一个开放源代码的选择是Xplico框架。Wireshark还具有“导出对象”功能，用于从捕获中提取数据，例如，File – > Export Objects – > HTTP – > Save all。除此之外，你可以尝试使用tcpxtract，Network Miner， Foremost或Snort。

如果要编写自己的脚本直接处理PCAP文件，建议使用用于pcap操作的dpkt Python包。你也可以使用Wirepy从你的Python中使用Wireshark。如果尝试修复损坏的PCAP文件，则有一个在线服务来修复名为PCAPfix的PCAP文件。

关于PCAP与PCAPNG的注意事项，有两个版本的PCAP文件格式。你可能需要使用Wireshark或其他兼容工具将文件从PCAPNG转换为PCAP，以便在其他工具中使用它。

内存转储分析

多年来，人们一直把计算机取证与文件系统取证看作是同一回事，但随着攻击越来越复杂，攻击者开始避开磁盘。而且内存快照通常包含在磁盘上无法找到的上下文和线索中，因为它们只存在于运行时，例如操作配置，远程攻击shellcode，密码和加密密钥等。因此，内存快照或内存转储取证已经成为事件响应中的流行做法。

用于内存转储分析的首选开源框架是Volatility，Volatility是用于解析使用外部工具，或通过暂停VM收集的VMware内存映像收集的内存转储的Python脚本。因此，只要知道内存转储文件和相关的配置文件（收集转储的操作系统），Volatility就可以开始识别数据中的结构，运行进程，密码等，它还可以使用插件来提取各种工件类型。

Ethscan用于在内存转储中查找看起来像网络数据包的数据，然后将其解压缩到pcap文件中，以便在Wireshark中查看，用于提取SQL数据库，Chrome历史记录，Firefox历史等的插件。

PDF文件分析

PDF是一个非常复杂的文档文件格式， PDF格式是部分纯文本，如HTML，但内容中包含许多二进制对象。二进制对象可以是压缩或甚至加密的数据，并且包括脚本语言中的内容，如JavaScript或Flash。要显示PDF的结构，你可以使用文本编辑器浏览它，也可以使用PDF感觉文件格式编辑器打开它，如Origami。

qpdf是一个可以用于探索PDF并从中转换或提取信息的工具。另一个是Ruby中的一个框架，叫做Origami。

当探索隐藏数据的PDF内容时，隐藏位置通常指的是以下几个：

· 不可见层

· Adobe的元数据格式“XMP”

· PDF的“增量生成”功能，其中保留先前版本，但对用户不可见

· 在白色背景上的白色文本

· 文字背后的图像

· 重叠图像后面的图像

· 未显示的评论

还有几个Python包用于处理PDF文件格式，如PeepDF，可以让你编写自己的解析脚本。

视频和音频文件分析

与图像文件格式一样，可以使用stegonagraphy在内容数据中嵌入一个秘密消息，也要知道检查文件元数据区域的线索。第一步是使用mediainfo工具或exiftool来查看内容类型并查看其元数据。

Audacity是很流行的开源音频文件和波形查看工具，CTF挑战者喜欢将文本编码成音频波形，尽管一个名为Sonic Visualiser的专用工具特别适合此任，但我还是建议使用spectogram视图查看。Audacity还可以让你减缓，反转和执行其他可能显示隐藏消息的操作，Sox是转换和操作音频文件的另一个有用的命令行工具。

检查秘密消息的最低有效位（LSB）也是常见的。大多数音频和视频媒体格式使用离散方式以便可以流式传输，最低有效位的方法就是偷走某些数据而不会明显影响文件的常见地点。

其他时候，消息可能会被编码为DTMF音调或莫尔斯码。

视频文件格式实际上是容器格式，其中包含音频和视频的单独流，它们被多路复用在一起进行播放。为了分析和处理视频文件格式，建议使用ffmpeg。 ffmpeg –i可以给出文件内容的初步分析。它还可以解复用或回放内容流。

办公文件分析

迄今为止，微软已经创建了数十种Office文档文件格式，其中许多文件格式已经被网络钓鱼和恶意软件作为传播恶意程序的载体，因为它们包含宏（VBA脚本）。Office文档取证分析与PDF文档取证并不相同。

一般来说，Office文件格式有两种类型：OLE格式（RTF，DOC，XLS，PPT等文件扩展名）和“Office Open XML”格式（包括DOCX，XLSX，PPTX的文件扩展名）。两种格式都是结构化的复合文件二进制格式，可以启用链接或嵌入式内容。OOXML文件实际上是zip文件容器，这意味着检查隐藏数据的最简单方法之一是简单地解压缩文档：

你可以看到，一些结构是由文件和文件夹层次结构创建的，其余的在XML文件中指定。

另外，Python工具集存在用于检查和分析OLE和OOXML文档——oletools。对于OOXML文档，OfficeDissector和Python库是一个非常强大的分析框架。有时，对办公文件分析的挑战不是找到隐藏的静态数据，而是分析一个VBA宏来确定其行为。

上述解析器工具可以指示宏是否存在，并可能为你提取数据。 Windows文档中的一个典型的VBA宏会将PowerShell脚本下载到％TEMP％，并尝试执行它，在这种情况下，你可以使用PowerShell脚本分析任务。但恶意的VBA宏不会很复杂，因为VBA通常只是作为一个跳出平台来引导代码执行。

如果宏被模糊化并且具有解压缩程序，则不需要拥有Office许可证来进行调试。你可以使用Libre Office，任何已调试程序的人都会熟悉其界面。你可以设置断点并创建观察变量，并在解压后捕获其值，但在执行任何有效负载行为之前，可以从命令行启动特定文档的宏：

$ soffice path/to/test.docx macro:///.standard.module1.mymacro

关注！