科普知识：什么是攻击隐写术

转载

weixin_34121304 于 2017-09-12 15:01:00 发布 363 收藏
文章标签： 运维 嵌入式 网络
原文链接：https://yq.aliyun.com/articles/214886

本文讲的是 **科普知识：什么是攻击隐写术，**

前言

隐写术是以隐藏格式发送数据的做法，因此这些发送的数据都会伪装成各种形式。“隐写”一词是希腊语στεγανό和γράφειν的组合，στεγανό的意思是“覆盖，隐藏或受保护”，而γράφειν的含义是“graphein”，意思是“写作”。

与隐藏秘密消息的密码术不同，隐写术是对传达的消息进行隐藏。隐写术的概念于1499年首次引入，但这个想法的出现其实在古代就有了，比如在罗马帝国，那些需要秘密传送的信息会被写在奴隶的头皮上，具体的做法就是先将他们的头发剃光，再将这些信息以纹身的方式刻在头皮上，等头发长长后，就派他们再去送信，然后接受者再次剃光那些人的头发，并阅读信息。

网络攻击中的隐写术

在20世纪由于技术的发展，隐形术也得到很大地发展，比如隐藏数据的方法或传递的方法等也有了很多的突破。然而，随着网络的兴起，一个危险的新趋势正在出现，隐写术越来越多地被黑客用来创建恶意软件和网络间谍工具。由于目前大多数反恶意软件解决方案都没有提供隐私保护，而有效载荷可以被秘密携带的任何载体构成潜在的威胁。它可能包含由间谍软件渗透的数据，恶意程序与其C&C之间的通信，或新的恶意软件。

在本文中，我们将用到以下一些使定义：

有效载荷：隐藏或秘密发送的信息，或隐藏的数据;

运营商（stego-container）：任何有效载荷秘密嵌入的对象;

Stego系统：用于创建隐藏通道以传达信息的方法;

信道：载波被传送的数据通信信道;

密钥：用于从载波中提取有效载荷的密钥。

目前卡巴斯基实验室的研究人员已经科学地开发和测试了各种隐写方法和算法，他们的总结如下：

1.在LSB隐写中，有效载荷被编码到载波的一个或几个最低有效位中并在其中传送。用于承载有效载荷的位数越少，对原始载波信号的影响越小。

2.离散余弦变换或基于DCT的隐写术是通常应用于JPEG格式载波（即当JPEG图像用于承载有效载荷时）的LSB隐写术的子类型。在该方法中，所传送的数据被秘密地编码成DCT系数。在所有其他因素相同的情况下，该方法提供了较低的数据承载能力,其原因之一是0和1的系数值不能被改变，所以每当系数取这些值时，是不会对数据进行编码的。

3.基于Palette的 image steganography 基本上是LSB隐写术的另一种子类型，其中传送的数据被编码成图像Palette的最低有效位，而不是载体的编码。这种方法的明显缺点是其低数据承载能力。

译者注：Image Steganography是体积小，易于使用的应用程序，用来将隐秘信息保存在保护指定的图像之中。这个小工具非常有意思，你只需待加密的图像到界面，然后输入密码字符串，或者也可以拖动一个密码文件，然后加密。加密后的图像跟以前的图像似乎没什么区别（输入为PNG格式），但是里面隐藏了只有你才知道的奥秘。

4.以数据格式使用服务字段，这是一种相对简单的方法，其中有效载荷被嵌入到运营商头部的服务字段中。不过，该方法存在低数据承载能力和低有效载荷保护的缺点，不过可以使用常规图像查看软件来检测嵌入式有效载荷。

5.有效载荷嵌入是将有效载荷编码到载波中并且在传送时使用双方已知的算法进行解码的方法。可以将多个有效载荷独立地编码到相同的载体中，只要它们的嵌入方法是正交的。

6.宽带方法（Wideband methods）分为以下几种：

6.1 伪随机序列方法，其中秘密载波信号由伪随机信号调制。

6.2 跳频方法，其中载波信号的频率根据具体的伪随机规律变化。

7.覆盖方法，严格来说，这不算是隐写术，而且该方法还是基于一些包含头文件中的数据大小的数据格式，比如基于连接图像文件的着名的RAR / JPEG方法，因此它由JPEG格式部分以及RAR归档部分组成。 JPEG查看器软件程序将读取文件头文件中指定的边界，而RAR存档工具将忽略RAR之前的所有内容。因此，如果在图像文件查看器中打开这样的文件，它将显示图像，如果在RAR归档器中打开，它将显示RAR存档的内容。这种方法的缺点是添加到载体段的叠加层可以在视觉上被分析人员审轻松地识别。

在本文中，我们将仅介绍图像载体和网络通信中隐藏信息的方法。不过，现实情况中，隐写术的应用要多得多。

最近，我们已经看到在以下恶意软件程序和网络间谍工具使用了隐写术：

```
Microcin (AKA six little monkeys);
NetTraveler;
Zberp;
Enfal（其新的加载器称为Zero.T）;
Shamoon;
KinS;
ZeusVM;
Triton（Fibbit）。
```

那么为什么恶意软件的开发者会越来越多地在工具中使用隐写术？我们发现了三个主要原因：

1.不仅隐藏数据本身，而且隐藏数据正在上传和下载的痕迹;

2.它有助于绕过与公司系统相关的DPI系统;

3.使用隐写术可能有助于避免被APT产品检测到，因为APT产品还无法处理所有图像文件（公司网络中包含太多的图像文件，分析算法的费用相当高）。

而对于个人用户来说，检测载波内的有效载荷更是不可能的任务了。举个例子，我们来看看下面的两个图片。一个是空载体，另一个是有载荷的载波。我们将使用标准的测试图像Lenna。

两张不同格式的图片分别为786字节和486字节，然而，你能看出来Lenna_stego.bmp格式的图像包含了纳博科夫小说《洛丽塔》的前10章内容吗？

仔细看看这两个图像。你能看到有什么区别吗？它们在尺寸和外观上都相同。但是，内部包含嵌入消息的载体去不同。

问题很明显：

1.隐写术现在非常受恶意软件和间谍软件作者的欢迎;

2.反恶意软件工具通常对载荷量很大的载体非常难以检测，因为它们看起来像常规的图像文件或其他类型的文件;

3.今天的所有隐写检测程序都是基本的Poc，而且他们的逻辑不能在商业安全工具中实现，因为它们的速度很慢，检测率相当低，有时甚至包含计算错误。

上面列出了一个使用隐写术来隐藏他们通信的恶意程序（尽管不是完整的）。我们来看看这个列表中的一个具体案例，恶意加载器Zero.T。

卡巴斯基的研究人员在2016年底发现了这个装载器，并将其命名为Zero.T，因为这个字符串在其可执行代码中（在通向PBD文件的路径中）：

我们不会在这里讨论恶意加载程序如何绕过受害者系统并驻留在其中，但会讨论它以Bitmap文件的形式加载有效载荷：

然后以特定方式处理它们以获取恶意模块：

从表面上看，这三个BMP文件似乎是图像：

但是，它们不仅仅是我们看到一般图像，它们是被充分载荷的通信载体，在每一副图像中，最低有效位都被有效载荷代替。

那么，有没有办法确定图像是否携带恶意的有效载荷呢？当然是有的，最简单的办法就是视觉冲击（visual attack）。它基于从源图像形成新图像，其中包含不同颜色平面的最低有效位。

让我们看看如何使用史蒂夫•乔布斯照片作为示例图像。

我们对这个图像进行视觉冲击，并以适当的顺序从单独的有效位构建新的图像：

在第二和第三图像中，高熵（高数据密度）区域是显而易见的，这些区域包含嵌入式有效载荷。

听起来很简单，不过这种分析很难自动化。幸运的是，科学家早就开发出了一些基于图像的统计特征来检测载波。然而，所有这些方法的前提都是基于编码的有效载荷具有高熵。这在大多数情况下是可用的，由于容器的容量有限，所以有效载荷在编码之前被压缩或加密，从而增加其熵。

然而，我们的现实生活中的恶意加载器Zero.T在编码之前并没有压缩其恶意模块。相反，它增加了其使用的最低有效位数，其可以是1,2或4。使用较大数量的最低有效位将视觉伪像引入到载体图像中，普通用户可以通过视觉冲击进行检测。但是这个手动过程比较麻烦，所以还是要依靠自动分析。这就引出了一个问题，适用于检测熵低的嵌入式有效载荷的统计方法是什么？

统计分析方法：直方图法

这种方法在2000年由Andreas Westfeld和Andreas Pfitzmann提出，也被称为卡方检验方法。下面我们简单介绍一下。

分析整个图像光栅，对于每种颜色来说，在光栅内计数拥有该颜色的点数。为了简单起见，我们以正在处理的一个具有一个颜色平面的图像为例。对于不包含嵌入式有效载荷来说（见下图A），该方法假设拥有两个相邻颜色的像素数量（仅一个最低有效位不同的颜色）。对于具有有效载荷的载体图像，拥有这些颜色的像素数量是相似的（见图B）。

以上是可视化地表示该算法的简单方法。

严格来说，算法由以下步骤组成：必须按顺序执行：

1.有效载荷嵌入图像中颜色i的像素的预期出现频率计算如下：

2.将特定颜色的像素的出现的测量频率确定为：

3.k-1自由度的卡方数为：

4.P是这些条件下分布ni和ni *相等的概率。通过积分密度函数计算：

这几个步骤下来，我们就已经测试了这种方法是否适用于检测填充的容器。以下是检测结果：

p = 0.95和p = 0.99的卡方分布的阈值分别为101.9705929和92.88655838。因此，对于计算出的卡方值低于阈值的区域，我们可以认为相邻颜色具有相似的频率分布，因此我们正在处理具有有效载荷的载波图像。

事实上，如果我们看视觉冲击图像，就可以清楚地看到这些区域包含一个嵌入的有效载荷。因此，该方法适用于高熵有效载荷。

统计分析方法：RS法

检测有效载荷的另一个统计方法是由Jessica Fridrich，MiroslavGoljan和Andreas Pfitzmann在2001年提出。它被称为RS方法，其中RS代表“常规或奇异”。

分析的图像被分成一组像素组，然后对每个组应用特殊的翻转进程（flipping procedure）。基于应用翻转进程之前和之后的判别函数的值，所有组被划分为常规，单数（singular ）和不可用组。

该算法基于一种假设，即原始图像中的常规和特殊像素组的数量必须近似相等，并且在翻转之后的图像中应用。如果这些组的数量在应用翻转之后变化明显，则表示分析的图像是具有有效载荷的载体。

该算法包括以下步骤：

1.原始图像被分为n个像素（x1，…，xn）的组。

2.定义所谓的判别函数，其分配给每个像素组G =（x1，…，xn）实数f（x1，…，xn）∈

3.像素组（x1，…，xn）的判别函数可以定义如下：

4.然后我们定义具有以下属性的翻转函数：

根据应用翻转之前和之后的判别函数的值，所有像素组都分为常规，单数和不可用组：

我们也将这种方法用于测试，并得到以下结果。我们使用与上一次测试相同的空载和有效载荷的载波。

请注意，这种攻击方法在“该特定载体是否包含嵌入式有效载荷”方面没有通过二进制校验，而是确定嵌入式有效载荷的近似长度（以百分比表示）。

从上面的结果可以看出，该方法返回了一个空消息的判定，它填充了小于1％的有效载荷，而对于有载荷载入的载体，它返回了大约44％的填充结果。显然，这些结果略有偏离。我们来看看装满的容器：从视觉冲击来看，超过50％的容器被填满，而RS检测则告诉我们，44％的容器已经装满了。因此，如果我们建立一定的“触发阈值”，我们可以应用这种方法：我们的实验表明10％是足够的可靠性阈值。如果RS检测声称超过10％的容器已满，你可以信任此检测，并将该容器标记为被填满。

现在是时候在真实的条件下，在有载荷具有规则熵的零载波上测试这两种方法了，。

我们进行了适当的测试，结果如下：

我们看到，卡方检测不适用于低熵图像，因为产生不令人满意的或不准确的结果。然而，RS检测却运行良好：在这两种情况下，都检测图像中有隐藏的有效载荷。然而，如果自动分析方法检测不准，该怎么办？

在这种情况下，我们可以应用针对特定恶意软件家族开发的特定程序来提取有效载荷。对于上述的Zero.T加载器，我们已经编写了我们自己的嵌入式有效载荷提取工具，其操作可以理解如下。

显然，如果我们得到一个有效的结果（是一个可执行文件），那么源图像中就有一个嵌入的有效载荷。

DNS隧道也是隐写术吗？

我们可以考虑使用DNS隧道进行隐写术的子类型，首先，让我们回顾一下DNS隧道的工作原理。

从封闭网络中的用户计算机发送请求以解析域，例如发送域 wL8nd3DdINcGYAAj7Hh0H56a8nd3DdINcGYAlFDHBurWzMt[.]imbadguy[.]com 到一个IP地址，（在此URL中，二级域名无意义）。本地DNS服务器会将此请求转发到外部DNS服务器。而外部DNS服务器又不知道第三级域名，所以它会向前传递这个请求。因此，此DNS请求遵循从一个DNS服务器到另一个DNS服务器的重定向链，并到达域imbadguy[.]com的DNS服务器。

恶意软件的开发者可以通过解码其第一部分来解决DNS服务器上的DNS请求，而不是从接收到的域名中提取所需的信息。例如，可以以这种方式传送关于用户系统的信息。作为回应，恶意软件的开发者的DNS服务器还以解码格式发送一些信息，将其放入第三或更高级别的域名。

这意味着每个DNS分辨率的攻击者具有255个字符的预留位置，子域名最多为63个字符。每个DNS请求发送63个字符的数据，并返回63个字符作为响应等等。这样就使得它成为一个常规的数据通信渠道。最重要的是，它是隐藏的通信渠道，肉眼看不到任何额外的数据。

熟悉网络协议的专家，特别是DNS隧道的专家，才会对包含这种通信的流量转储很敏感，这可以通过包含太多的长域被发现。比如，我们正在查看由Trojan Backdoor.Win32.Denis生成的流量的现实示例，其使用DNS隧道作为隐藏通道与C&C进行通信。

借助任何流行的入侵检测（IDS）工具（如Snort，Suiricata或BRO IDS），都可以检测DNS隧道。这可以使用各种方法完成，例如，最常用的就是使用在DNS解析过程中发送的域名在隧道期间比通常要长得多。

alertudp any any -> any 53 (msg:"Large DNS Query, possible cover channel"; content:"|01 00 00 01 00 00 00 00 00 00|"; depth:10; offset:2; dsize:>40; sid:1235467;)

这也是一个相当原始的方法：

Alert udp $HOME_NET and -> any 53 (msg: "Large DNS Query"; dsize: >100; sid:1234567;)

在这里有很多的方法可以找出虚假数量和检测实际DNS隧道实例之间的平衡。

除了可疑的长域名外，还有哪些其他因素可能有用呢？比如域名的异常语法。典型的域名通常包含字母和数字。但如果一个域名包含Base64字符，它会看起来很可疑，不是吗？如果这样的域名也很长，那么肯定有猫腻。

其实还有很多的异常因素，正则表达式对检测它们有很大的帮助。

我们注意到，目前巴斯基实验室的恶意软件分析工具中，已经把DNS隧道检测的方法加进去了，并检测到使用DNS隧道作为C&C通信的隐蔽通道的几个新的、以前未知的后门。

总结

我们看到恶意软件开发人员使用隐写术的趋势已经出现并呈现强进的上升势头，包括隐藏C&C通信和下载恶意模块。对于恶意软件的开发者来说，这是一种实现隐写术有效的方法，因为考虑到用户配备有效载荷检测工具是很昂贵的一笔支出，所以这意味着大多数安全解决方案目前还无法处理可能包含隐写术有效载荷的所有对象。

然而，确实存在有效的解决方案，它们基于不同的分析方法，及时的预检测，潜在有效载荷的元数据分析等的组合。目前，这种解决方案在卡巴斯基实验室的反目标攻击（Anti-Targeted Attack，KATA）解决方案中就实现了。在部署KATA的情况下，信息安全人员可以及时了解对受保护范围所遭受的攻击或数据正在被渗透的证据。

原文发布时间为：2017年8月14日
本文作者：luochicun
本文来自云栖社区合作伙伴嘶吼，了解相关信息可以关注嘶吼网站。
原文链接