我们提出了一个通用的方法来搜索回避变体，并使用我们针对两个PDF恶意软件分类器PDFrate和Hidost的技术进行实验。我们的方法能够为我们研究中的所有500个恶意种子的两个分类器自动找到规避的变体。

贡献

1.我们提出了一种通用的方法来自动找出目标分类器的回避变体。该方法不依赖于任何特定的分类算法或假设详细的特征提取知识，但只需要生成的变种的分类评分反馈和分类器使用的可能特征的粗略知识（第二节）

2.我们实现了一个原型系统，该系统能够自动发现可以逃避基于结构特征的PDF恶意软件分类的变体。这涉及设计在PDF文件上执行随机操作的操作符，确定生成的变体是否保留恶意的预言器，在进化过程中促进有希望的变体的选择机制以及每个目标分类器的适合度函数（第四节）。

3.我们评估了我们的系统的有效性，以回避两个最近的PDF恶意软件分类器：PDFrate [25]和Hidost [28]，这是一个以明确拒绝回避企图为目标的分类器。我们的系统在500个恶意软件样本种子的实验中发现了两个分类器的回避变体，实现了100％的成功率。对每个分类器的特征空间中发现的规避变体的分析表明，分类中使用的许多非鲁棒特征有助于逃避攻击（第V和第VI部分）。我们在第二部分提供机器学习分类器的背景知识，第三部分提供有关PDF恶意软件的背景知识。第八节讨论了有关逃逸攻击的工作。

A.机器学习分类器

训练机器学习系统的努力始于特征提取。由于大多数机器学习算法不能在高度结构化的数据上操作，所以数据样本通常在特别设计的特征空间中表示。例如，恶意软件分类器可以将文件大小和函数调用跟踪提取为特征。每个特征都是特征空间中的一个维度;因此，每个样本都被表示为一个向量。当分类算法的特征数量太大时，可以执行特征选择的额外步骤以减少特征的数量。

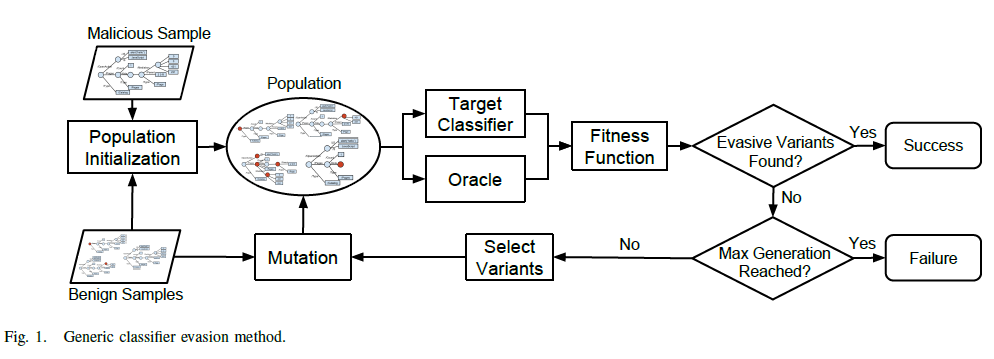
恶意软件分类器通常使用静态属性来预测恶意，如文件结构（the ﬁle structure），文件大小，元数据（metadata），令牌克数（grams of tokens）或系统调用（system calls）。尽管这种方法在验证测试中经常达到很高的准确性，但是分类器可能学习的是属性是训练数据的超痕迹，而不是与恶意软件本身相关的属性。这是因为训练数据中的恶意软件样本可能与良性样本在许多方面有所不同，而这些样本对其恶意行为并不重要。

B.威胁模型

假设攻击者可以对目标分类器进行黑箱访问，并且可以向该分类器提交许多变体。对于每个提交的变体，攻击者学习其分类分数。分类评分是一个数字（通常是介于0和1之间的实数），表示分类器对恶意的预测，其中高于某个阈值（比如0.5）的值被认为是恶意的，而分类评分低的样本被认为是良性的。我们不假设攻击者有任何有关分类器的内部信息，只是它可以用它作为黑盒来输出输入样本的分类得分。我们假设分类运算符不会将分类器适配到提交的变体（如果攻击者能够访问分类器，则必须是这种情况）。

C.找到回避样本

我们的方法使用遗传编程技术来执行对可能样本空间的定向搜索，以找到逃避分类器的样本，同时保留所需的恶意行为。

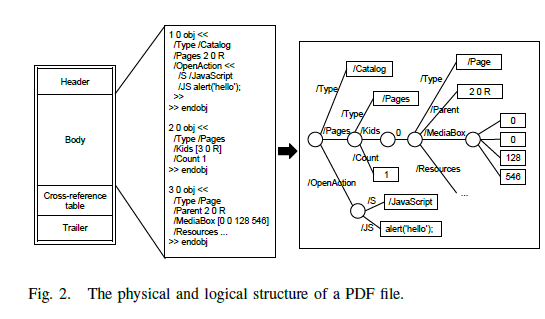


2图1.通用分类器逃避方法。包括修复遗留软件错误[17]，软件逆向工程[13]和软件重新设计[23]。方法。我们的过程如图1所示。它从一个展现恶意行为的种子样本开始，被目标分类器分类为恶意行为。我们的方法旨在发现一个避开恶意行为但被目标分类器错误分类为良性的回避样本。

首先，我们通过对恶意种子执行随机操作来初始化变体种群。

然后，每个变体都由一个目标分类器和一个预言器进行评估。

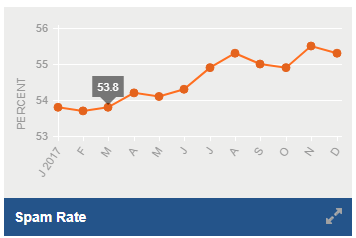
接下来，所选择的变体被变异算子随机操纵以产生下一代种群。这个过程一直持续到找到一个回避的样本或达到了一个世代的阈值。

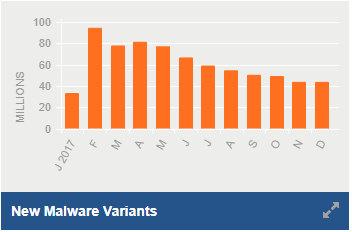


一个PDF文件由四部分组成：标题，正文，交叉引用表（CRT）和预告片。标题包含PDF幻数和格式版本指示符。正文是一组包含文件内容的PDF对象，而CRT则是索引正文中的对象。预告片指定如何找到CRT和其他特殊对象，如根对象。因此，PDF阅读器通常从文件末尾开始阅读PDF以提高效率。

主体是最重要的。它包含八种基本类型的对象，即布尔值namely Booleans，数字numbers，字符串strings，名称names，数组arrays，字典dictionaries，流streams和空对象the null objects。如图2右侧所示。这个树状结构非常适合基因编程技术，因为它很容易改变和移动子树来生成新的变体。

根据最近的互联网安全威胁报告[30]，2014年，PDF在鱼叉钓鱼电子邮件中排名前7位。我们预计PDF恶意软件攻击将继续存在，因为Acrobat Reader中的128个新漏洞已经在CVE到2015年（到12月8日）为止，几乎是2014年的三倍[8]。





PDF恶意软件通常包含利用特定PDF阅读器（通常为Adobe Acrobat）的漏洞的JavaScript对象或其他对象的漏洞。

III. PDF MALWARE AND CLASSIFIERS

根据Wepawet的结果，这500个恶意软件样本利用Acrobat Readers中的两个不同漏洞：其中333个使用CVE-2007-5659中报告的多个缓冲区溢出漏洞，另外167个使用CVE-2009-0927报告的基于堆栈的缓冲区溢出漏洞。这两个漏洞都可以被利用来执行任意代码。总之，500个样本中的有效载荷访问255个不同的主机，从互联网上下载额外的恶意软件。

着眼于恶意软件样本的网络行为，我们可以提取由Cuckoo报告的各种网络行为作为签名，如DNS查询，HTTP URL请求和网络目的地

