这是作者的系列网络安全自学教程，主要是关于网安工具和实践操作的在线笔记，特分享出来与博友共勉，希望您们喜欢，一起进步。前文分享了Web漏洞及端口扫描之Nmap、ThreatScan和DirBuster工具，社会工程学中的IP物理位置定位、IP获取、手机和邮箱查找、文件属性等。本篇文章，作者将分享两篇论文，机器学习是如何运用到恶意代码攻击中的，并谈谈自己的理解，后续深入研究尝试分享相关实验，目前还是小白一只。基础性文章，希望对初学者有帮助，大神请飘过，谢谢各位看官！

百度网盘：https://pan.baidu.com/s/1dsunH8EmOB_tlHYXXguOeA 提取码：izeb

参考文献：

[1] 张东, 张尧, 刘刚, 宋桂香. 基于机器学习算法的主机恶意代码检测技术研究[J]. 网络与信息安全学报, 2017(7): 25-32.
[2] 杨轶, 苏璞睿, 应凌云, 等. 基于行为依赖特征的恶意代码相似性比较方法[J]. 软件学报, 2011, 22(10): 2438-2453.
[3] 杨晔. 基于行为的恶意代码检测方法研究[D]. 西安: 西安电子科技大学, 2015.
[4] 李盼，赵文涛，刘强+，崔建京，殷建平. 机器学习安全性问题及其防御技术研究综述，计算机科学与 探索，2018(12).
[5] 张蕾, 崔勇, 刘静, 江勇, 吴建平. 机器学习在网络空间安全研究中的应用[J]. 计算机学报, 2018(9): 1943-1975.
[6] IMRAN M, AFZAL M T, QADIR M A. Malware classification using dynamic features and hidden markov model[J]. Journal of In telligent & Fuzzy Systems, 2016, 31(2):837-847.
[7] 恶意程序行为分析工具 PeDoll - DBinary
[8] 《恶意代码分析实战》诸葛建伟 姜辉 张光凯
[9] https://www.cnblogs.com/yunji5566/p/4249927.html
[10] 2008年瑞星安全技术大会

文章目录

前文学习：

[网络安全自学篇] 一.入门笔记之看雪Web安全学习及异或解密示例
[网络安全自学篇] 二.Chrome浏览器保留密码功能渗透解析及登录加密入门笔记
[网络安全自学篇] 三.Burp Suite工具安装配置、Proxy基础用法及暴库示例
[网络安全自学篇] 四.实验吧CTF实战之WEB渗透和隐写术解密
[网络安全自学篇] 五.IDA Pro反汇编工具初识及逆向工程解密实战
[网络安全自学篇] 六.OllyDbg动态分析工具基础用法及Crakeme逆向破解
[网络安全自学篇] 七.快手视频下载之Chrome浏览器Network分析及Python爬虫探讨
[网络安全自学篇] 八.Web漏洞及端口扫描之Nmap、ThreatScan和DirBuster工具
[网络安全自学篇] 九.社会工程学之基础概念、IP获取、IP物理定位、文件属性

前文欣赏：

[渗透&攻防] 一.从数据库原理学习网络攻防及防止SQL注入
[渗透&攻防] 二.SQL MAP工具从零解读数据库及基础用法
[渗透&攻防] 三.数据库之差异备份及Caidao利器
[渗透&攻防] 四.详解MySQL数据库攻防及Fiddler神器分析数据包

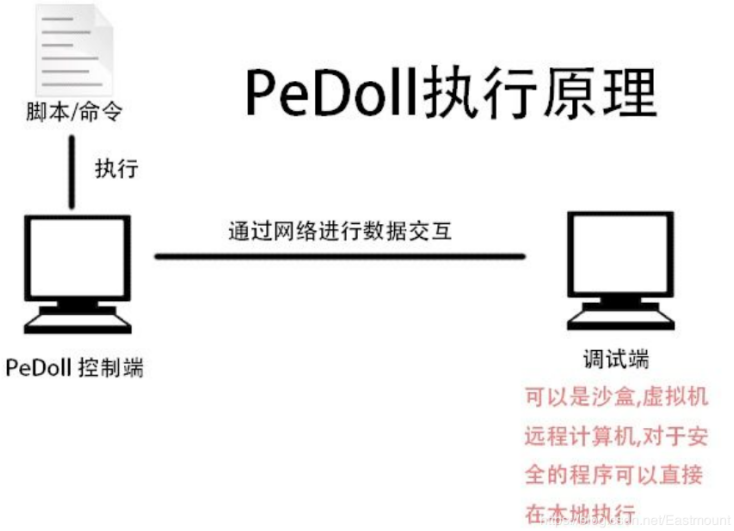
下载地址：https://github.com/eastmountyxz/NetworkSecuritySelf-study

---

一.什么是恶意代码?

恶意代码（Malicious Code） 是指运行在目标主机中，按照攻击者所规定逻辑执行的指令，其类别包括计算机病毒、蠕虫、木马、僵尸网络、勒索软件等。恶意代码攻击可以窃取核心数据和敏感信息，甚至对计算机系统和网络造成破坏，是当今网络安全的最大威胁之一。

恶意代码分析是一种解剖恶意代码的艺术，了解恶意代码是如何工作、如何识别，以及如何战胜或消除它。



现阶段，恶意代码呈现变种数量多、传播速度快、影响范围广的特点。尤其需要注意的是，恶意代码常针对新型漏洞（如零日漏洞）进行设计，是敌手发动 **高级持续性威胁（APT，advanced persistent threat）** 的主要技术手段。

基于行为的恶意代码检测技术 被许多安全厂商用来打造“主动防御”、“启发式查毒”产品。瑞星合理地将该技术应用于本机威胁感知、本机威胁化解及“云安全”中心威胁自动判定分析中，该技术是瑞星“云安全”策略实施的辅助支撑技术之一。

二.恶意代码检测方法

（一）传统的恶意代码检测

传统的恶意代码检测包括基于签名特征码 （ signature ）的检测和基于启发式规则（heuristic）的检测，在应对数量繁多的未知恶意代码时，正面临越来越大的挑战。

1.基于签名特征码的检测

签名特征码检测方法通过维护一个已知的恶意代码库，将待检测代码样本的特征码与恶意代码库中的特征码进行比对，如果特征码出现匹配，则样本为恶意代码。该方法需要耗费大量的人力、物力对恶意代码进行研究并要求用户及时更新恶意代码库，检测效率和效果越来越力不从心，并且很难有效抵御未知恶意代码。

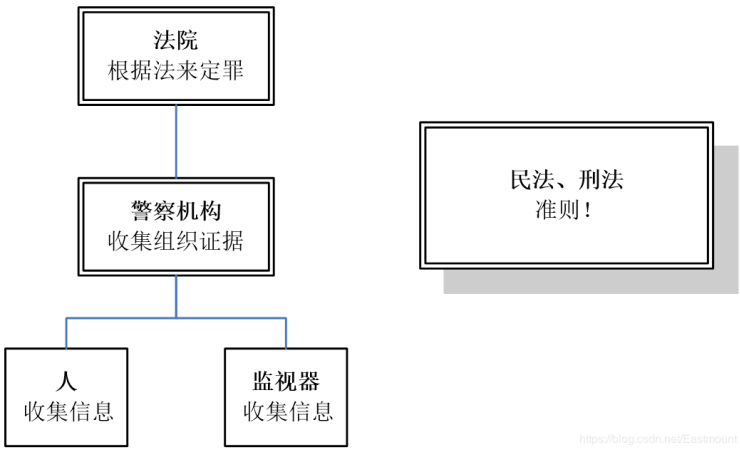
2.基于启发式规则的检测

启发式规则检测方法通过专业的分析人员对现有的恶意代码进行规则提取，并依照提取出的规则对代码样本进行检测。但面对现阶段恶意代码爆炸式的增长趋势，仅依赖人工进行恶意代码分析，在实施上变得愈发困难。

传统的特征检测技术优缺点如下图所示：

那么，什么是特征码技术呢？行为分析又是指什么呢？

人类社会的“特征码”技术是——指纹。截取初犯的指纹放入档案，当再犯时，查对指纹，就可确定谁是犯人。法院可以根据法律和收集的信息来定罪。

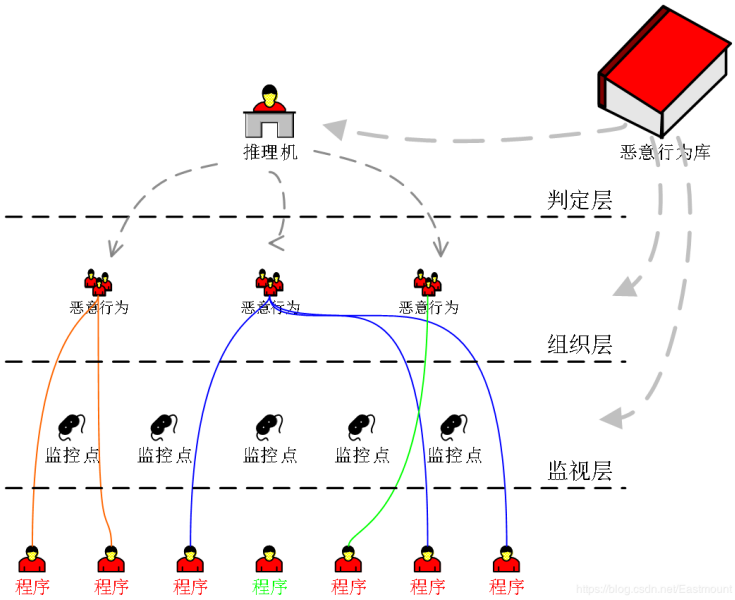


我们又怎么给程序判定罪证呢?
把程序看成“人”，制定适用于这些“人”的“法律”，监视这个“人”的动作，整理、归纳收集到的信息，根据“法律”来判定“人”的好坏，行为分析就这样出现了！

> 行为分析定义为将一系列已经规定好的恶意行为做为规范，根据这些规范，去监视程序做了什么，再结合这个规范来判定程序是否是恶意代码。它并不什么新技术，而是病毒分析专家判定经验的应用。

瑞星公司的行为分析模型如下图所示，在恶意行为库中，监控层见识程序作了什么，组织层抽象信息，判断模块确定具体判定方式。

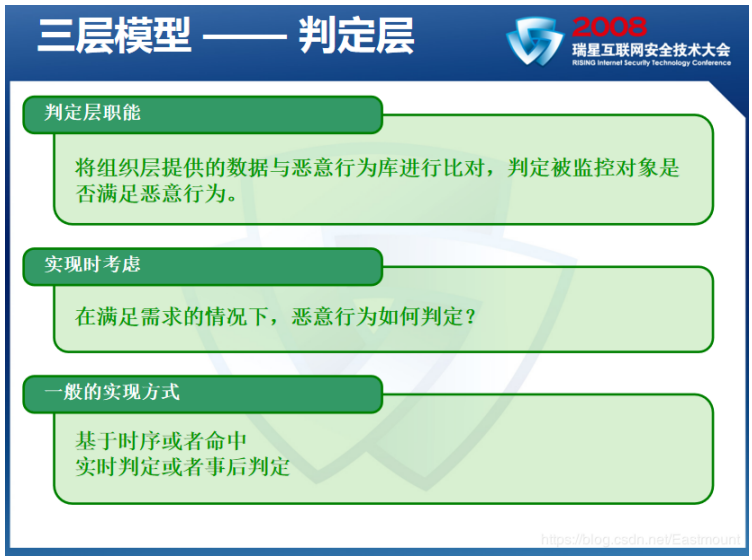
通过推理机和恶意行为库判断恶意行为、恶意程序和正常程序。



恶意行为库是系统设计和实施的重点，直接影响整个系统的设计、实现以及效果。恶意动作、恶意行为要尽可能地区别正常程序与恶意代码，病毒分析经验的运用。除了病毒分析专家之外，没有再合适不过的人选了。

木马行为防御的判定层实现：

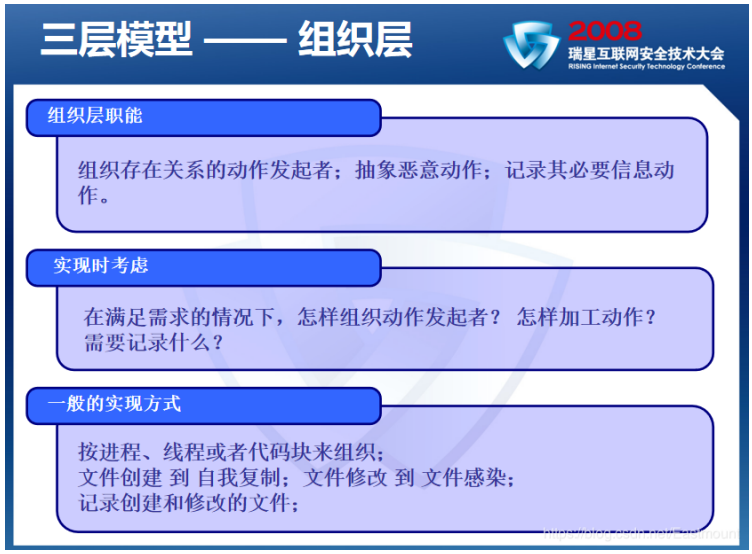
- 针对进程集进行判定。
- 实时比对，为每个进程集合创建并维护恶意行为库的匹配上下文。
- 内置恶意动作发生即可，顺序无关。
- 扩展恶意动作按顺序判定。



木马行为防御的组织层实现：

- 相关进程集合（创建关系，释放关系）。
- 忽略可见进程的程序动作。

- 必要时将程序动作加工成恶意动作。
- 记录程序创建或修改的文件。



木马行为防御的监控层实现：

- 文件监控。
- 进程监控。
- 注册表监控。
- 关键API调用监控。

	实时监控	环境模拟	虚拟机+环境模拟
运行方式	真实运行	真实运行	虚拟运行
运行速度	快	快	慢
执行深度	完全	视环境模拟程度	视环境模拟程度
危险性	危险	较危险	安全
监控粒度	函数级	函数级	指令级,函数级
实现复杂度	简单	视被模拟环境和需求	视被模拟环境和需求
产品化趋势	动态检测与防御	无	静态检测
产品化可行性	高	无	低
代表技术	瑞星木马行为防御	基于Wine的自动分析系统	RS未知DOS病毒检测 RS未知Win95病毒检测

指定恶意行为库：

- 恶意动作：（内置）自我复制，建立自启动关联，挂接全局自释放钩子等；（可扩展）程序动作+约束（自定义特征）。

- 恶意行为：多个不重复内置恶意动作，一组有先后顺序的扩展恶意动作。

未来做什么：

- 快速虚拟机实现
- 更合适规模的模拟环境实现
- 更细粒度的信息组织
- 更多的恶意动作

（二）基于机器学习算法的恶意代码检测

基于机器学习算法的防护技术为实现高准确率、自动化的未知恶意代码检测提供了行之有效的技术途径，已逐渐成为业内研究的热点。根据检测过程中样本数据采集角度的不同，可以将检测分为：**静态分析与动态分析。**

静态分析不运行待检测程序，而是通过程序（如反汇编后的代码）进行分析得到数据特征，而动态分析在虚拟机或仿真器中执行程序，并获取程序执行过程中所产生的数据（如行为特征），进行检测和判断。

根据 Cohen 对恶意代码的研究结果，可知恶意代码检测的本质是一个分类问题，即把待检测样本区分成恶意或合法的程序。
其核心步骤为：

1. 采集数量充分的恶意代码样本；（难点）
2. 对样本进行有效的数据处理，提取特征；（难点）
3. 进一步选取用于分类的主要数据特征；
4. 结合机器学习算法的训练，建立分类模型；
5. 通过训练后的分类模型对未知样本进行检测。

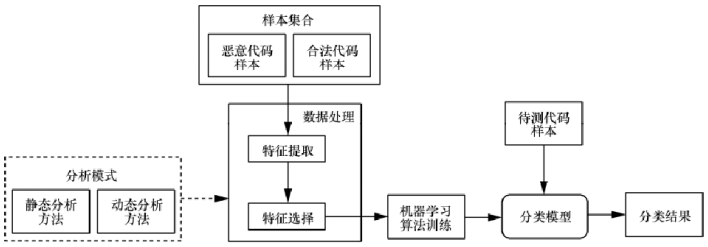


图 1　基于机器学习算法的恶意代码检测步骤

三.恶意代码样本采集

恶意代码样本的有效采集是进行代码分析工作的基础。当结合机器学习算法进行检测时，只有通过充分的样本数据训练，分类模型才能更准确地实现检测功能。一般来讲，恶意代码样本的获取途径有如下几种。

1.用户端采样

这是大多数杀毒软件厂商的主要获取方法，使用杀毒软件的终端用户将恶意代码样本上传至厂商。该方法具有较好的实时性，但安全厂商的样本数据往往选择不对外开放，很难直接获取。

2.公开的网络数据库

如 VirusBulletin、Open Malware、VX Heavens等，相比恶意代码的更新速度，现阶段

公开在线样本系统较有限，且站点存在隐蔽性不足、易遭到攻击的问题。因此，建立威胁情报的共享机制，日益突显出其重要性。

3. 其他技术途径

通过蜜罐（如 Nepenthes蜜罐）等捕获工具进行搜集，即设计一个专门的具有脆弱性的系统，诱导攻击者进行攻击进而得到恶意代码样本。一些木马和网络后门等也可以通过垃圾邮件陷阱或安全论坛（如卡饭论坛）的方式得到。
不过，上述技术途径的捕获样本规模较有限。

> 蜜罐技术本质上是一种对攻击方进行欺骗的技术，通过布置一些作为诱饵的主机、网络服务或者信息，诱使攻击方对它们实施攻击，从而可以对攻击行为进行捕获和分析，了解攻击方所使用的工具与方法，推测攻击意图和动机，能够让防御方清晰地了解他们所面对的安全威胁，并通过技术和管理手段来增强实际系统的安全防护能力。

四.基于机器学习的静态分析方法

提取恶意代码的静态特征，通过对程序代码进行逆向分析。常用的工具包括 IDA Pro、Hopper、OllyDbg 等。

1.样本特征提取

①基于序列的特征类型

该方法在样本特征的提取上应用最为广泛，其代表技术为 N 元语法模型（N-gram）。N-gram 假定 N 个出现的词只与之前出现的 N−1 个词相关，其中，N 代表一个特征序列的长度。如果考虑一个长为 L 的词组集合，则 N元语法模型会通过滑动窗口的形式，将词组划分为 L−N+1 个特征序列。例如，当 3-gram 被应用在词集{PUSH, SUB, SAL, AND, DIV, LDS, POP}上（此时 L=7）时，如图 2 所示，会得到 5 个特征序列，每个序列包含 3 个词元。

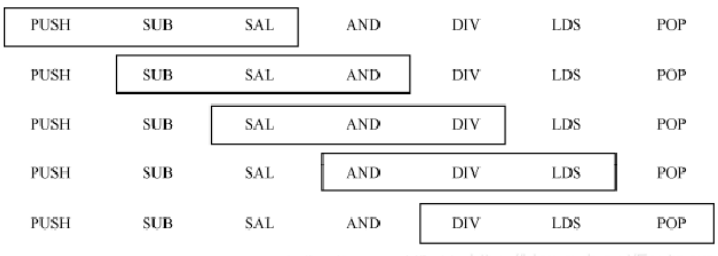


图 2　基于操作码的 3 元语法模型应用示例 http://blog.csdn.net/Eastmount

Abou-Assaleh 等首先提出了基于字节（Byte）序列的特征提取框架，并使用 K 近邻分类方法实现了恶意代码的有效检测。另一类词元选择方式是基于操作码（Opcode）的，Opcode是描述程序执行操作的机器语言指令，相对于字节序列来讲，具有更强的实际意义和可靠性，结合 Opcode 的特征提取可以更好地表征恶意代码。基于Opcode序列能完

成了对恶意代码进化的追踪，Siddiqui等结合操作码序列的方式，通过神经网络、决策树等分类算法，实现了 98.4%的检测准确率。

②基于字符串的特征类型

另一种特征类型的提取方式是基于程序代码中的可输出字符串，因为可输出字符串在某种程度上反映了待测程序的意图。例如，从代码中的“https://…”字符串可以推测程序的网络连接意图；而包含目录路径的字符串则说明程序可能尝试读取用户文档或注册表信息等。文献[18]选取了可执行文件中 100 个可输出字符串，以此为特征训练基于支持向量机的分类器，实现了99.38%的准确率。与基于序列的特征类型相比，代码中的字符串数量有限，因此提取的特征集具有较少的维度，在计算成本上可以实现有效的控制。

③基于 API 调用的特征类型

程序对应用程序编程接口（API，application programming interface）调用也可以作为特征类型。文献[19]对 API 调用进行了讨论，指出程序PE（portable executable）文件头中的 API 信息不具有准确性，因为恶意代码会在 PE 文件头中夹杂错误的 API 信息。Ding等对反汇编后的代码进行 API 调用分析，利用恶意代码和合法代码应用程序编程接口分布的差异性提取了基于 API 调用的程序特征。文献[20]将代码中的 API 调用序列转化为对应的马尔可夫（Markov）链，有向图中边的权重表示调用 API 的状态转移概率，通过基于 Markov 链的特征提取，实现了对未知恶意软件的有效分类。

2.样本特征选取

由于提取的数据特征常包含冗余信息，容易引起过度拟合问题，本节对数据特征选取的主要方法进行介绍，其种类主要包括信息增益（IG，information gain）、增益率（GR，gainratio）、文档频率（DF，document frequency）、主成分分析（PCA，principal component analysis）等。主成分分析也是一类常见的特征选取方法，在静态、动态分析中常被用于实现对样本数据的降维。PCA 通过线性变换，将原始数据投射到新的坐标系下，并通过新空间中最大线性无关组对数据样本进行表达，该线性无关组特征值的空间坐标即 PCA 方法所选取的特征。与 IG、DF 等方法不同，PCA 使用变换后的特征，而非原始特征的子集。

五.基于机器学习的动态分析方法

恶意代码的静态分析技术，在应对代码混淆或加壳等情形时，具有一定的局限性。为了保证代码评估的准确性，动态分析技术利用虚拟机或仿真器执行待测程序，监控并收集程序运行时显现的行为特征，并根据特征数据实现恶意代码的分类。

> 静态分析与动态分析区别：
> 调试逆向分为动态分析技术和静态分析技术。动态分析技术指的是使用调试工具加载程序并运行，随着程序运行，调试者可以随时中断目标的指令流程，以便观察相关计算的结果和当前的设备情况。静态分析技术是相对于动态分析而言的。由于在实际分析中，很多场合不方便运行目标（例如病毒程序，设备不兼容，软件的单独某一模块）。那么这个时候就需要应用静态分

> 析技术。OD（OllyDbg）和IDA Pro这两款工具分别是调试逆向的倚天剑和屠龙刀。虽然两者都兼容动态和静态的调试方式，但就动态调试而言，OD更为灵活和强大，而静态调试工具的王者理所应当是功能极为强大的IDA Pro。

1.行为特征提取

沙箱技术是收集行为特征的重要技术途径，许多安全公司提供了 Web 版的沙箱接口，用以对上传的程序样本进行动态分析，生成行为分析报告。目前常见的沙箱工具有 Anubis、Joe Sandbox、Cuckoo Sandbox、CWSandbox 等。

动态分析的重点是对监控行为的类型进行合理选择。一般来讲，基于行为分析的方案主要考察程序运行过程中所涉及的以下方面：

- 系统文件的改变，如创建或修改文件；
- 注册表键值的操作行为；
- 动态链接库（DLL，dynamic link library）的加载情况；
- 进程访问的情况；
- 系统服务行为，如开启、创建或删除服务；
- 网络访问情况；
- 应用程序编程接口（API）的调用。

此外，一些解决方案还对程序调用函数的数据信息进行分析，这时污点标签设置方法常被结合使用。

文献[22,23]结合行为报告的分析结果，对恶意代码的行为特征进行识别，借助机器学习算法对可执行文件进行分类。杨轶等通过分析污点传播的过程，识别不同的恶意代码行为间控制指令和数据的依赖关系，从而比较恶意代码的相似性。Imran 等通过隐马尔可夫模型对待测样本的动态行为特征进行描述，并借助机器学习算法实现分类。Anderson 等则通过动态方式搜集程序指令序列，进而生成基于马尔可夫链的有向图。

2.行为特征选取

许多沙箱工具，如 Anubis 和 CWSandbox 的输出格式为文本或可扩展标记语言（XML，
extensible markup language），这两类格式更适用于小规模样本的人工分析。具体来说，文本格式报告对行为特征的刻画过于简单，粒度较粗，一些重要的行为不再可见；而 XML 格式下的分析报告表述又过于繁冗，不便于开展自动化分析。为了高效处理行为分析数据，Trinius 等提出基于恶意软件指令集（MIST，malware instruction set）的行为数据描述方法，常被用来对其他格式（如 XML 格式）的行为报告进行转换，从而达到在行为数据中选取主要特征的目的。MIST 将程序行为的监控结果描述为一系列指令，其中每个线程和进程的执行流被分组在一个连续的报告中。每条指令都对应监控到

的一个系统调用（system call）及其调用到的参数（argument），指令以短数值的方式予以标识。此外，系统调用的具体参数被分隔在不同等级的块中，反映不同程度的行为粒度。

MIST 报告可以进一步通过向量空间模型（VSM，vector space model）进行向量化，生成可用于机器学习算法分类的数据。在特征项和特征项权重的计算上，可运用词袋模型（BOW，bag of words）。

词袋模型的示例如下，假设有下述 2 个文件。

1. Samuel detected a malware. I detected the malware too.
2. The malware was detected by us.

基于上述 2 个文件，可以构建一个词汇表。
词汇表={1．“Samuel”，2．“detected”，3．“a”， 4．“malware”，5．“I”，6．“the”，7．“too”，8．“was”， 9．“by”，10．“us”}。

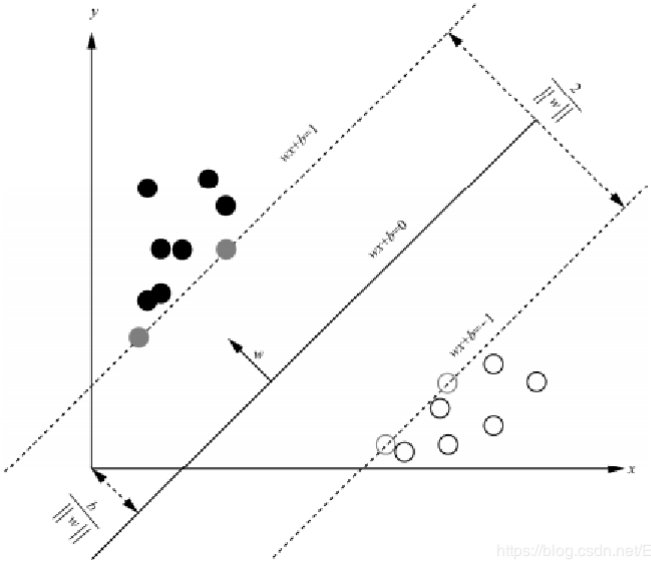
这个词汇表一共包含 10个不同的单词，利用索引号，上面 2 个文件可分别用 10 维向量表示（向量中元素为词表单词在文件中出现的频率）。

```
[1，2，1，2，1，1，1，0，0，0]
[0，l，0，1，0，1，0，1，1，1]
```

利用词袋模型，经过 MIST 处理后的指令语句将作为 VSM 模型中的特征项，指令的出现频率即为特征项的权重，以建立恶意代码的向量空间数据，这样就可以利用机器学习算法（如支持向量机）进行恶意代码的分类。

六.恶意代码分类算法

恶意代码进行静态、动态分析后得到的特征数据，可以作为机器学习算法训练的输入，产生
相应的恶意代码分类器。常见的算法如 K近邻（KNN，k nearest neighbor）、支持向量机
（SVM，support vector machine）、朴素贝叶斯（Naïve Bayes）、决策树（DT，decision tree）、随机森林（RF，randomforest），以及深度学习算法，如卷积神经网络（CNN，convolutional neural network）等。



下图展示了一种投毒攻击的示意图，以及机器学习训练过程中安全威胁及防御措施。

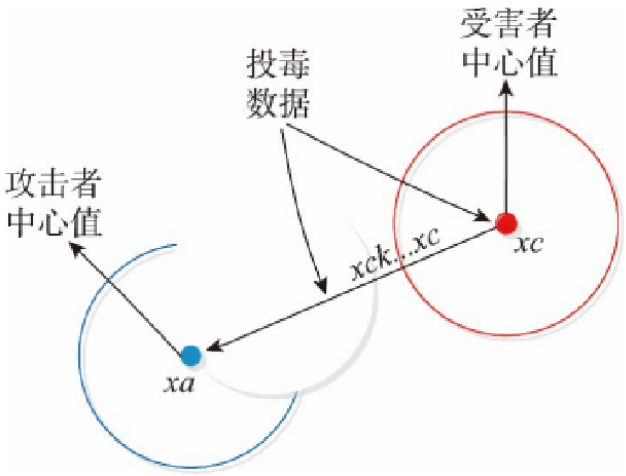


Fig.2 Illustration of a poisoning attack

图2 一种投毒攻击示意图

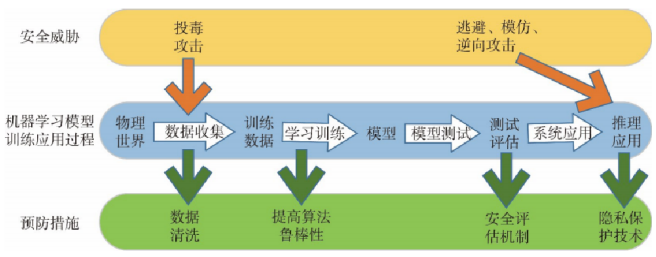


Fig.3 Security threats and their countermeasures in machine learning model training and application

图3 机器学习训练应用过程中安全威胁及防御措施

在考虑敌手视角时，如果攻击者也通过机器学习技术优化恶意代码的设计，对攻击目标画像并实现精准攻击，该如何应对？同时，又该如何保证机器学习引擎不被攻击者“投毒”，防止出现干扰项致使训练出错产生误判，这些都是需要进一步研究和思考的问题。

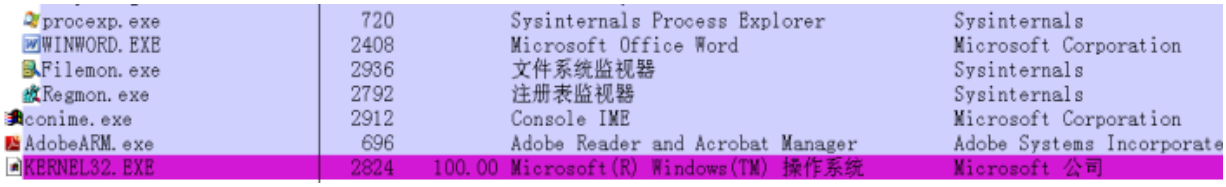
七.恶意代码检测实战知识

下面简单举一个示例——冰河木马分析与检测。后续希望自己能深入学习，学会这些实例分析，加油!

冰河开发的最初原因是为了开发一个功能强大的远程控制软件。但一经推出就成了黑客们的入侵工具。2006年以前冰河一直是国内不动摇的领军木马。功能有自动跟踪目标机屏幕变化、记录各种口令信息、获取系统信息、限制系统功能、远程文件操作、远程文件操作等。下面从以下几个不同方面分析冰河木马。

1.进程检测

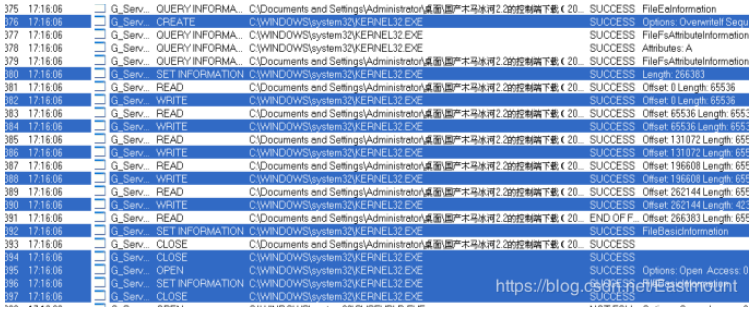
从Procexp软件可以明显的看到，有一个KERNEL32.EXE进程（能否进一步确定该进程调用的模块，进一步找准木马程序）。这个明显是假装系统进程的木马进程，CPU使用率达到了99%!



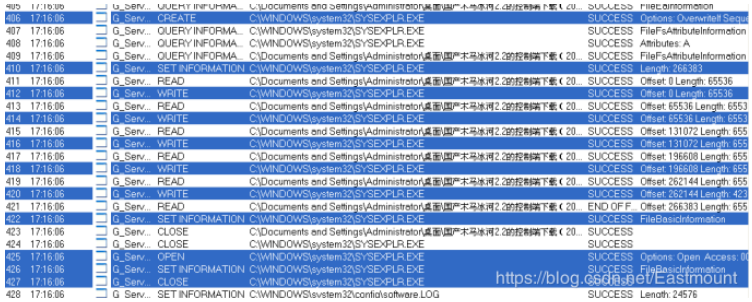
procexp.exe	720		Sysinternals Process Explorer	Sysinternals
WINWORD.EXE	2408		Microsoft Office Word	Microsoft Corporation
Filemon.exe	2936		文件系统监视器	Sysinternals
Regmon.exe	2792		注册表监视器	Sysinternals
conime.exe	2912		Console IME	Microsoft Corporation
AdobeARM.exe	696		Adobe Reader and Acrobat Manager	Adobe Systems Incorporate
KERNEL32.EXE	2824	100.00	Microsoft(R) Windows(TM) 操作系统	Microsoft 公司

2.文件监测

用Filemon监测到，样本先在c:\Windows\system32 目录创建了一个KERNEL32.EXE文件，并往其中写入了大量与自身运行有关的数据。如下图所示:



然后又在C:\Windows\system32目录下创建一个名SYSEXPLR.EXE的文件，随后又把查看了电脑文件目录信并把它们写入这两个文件。

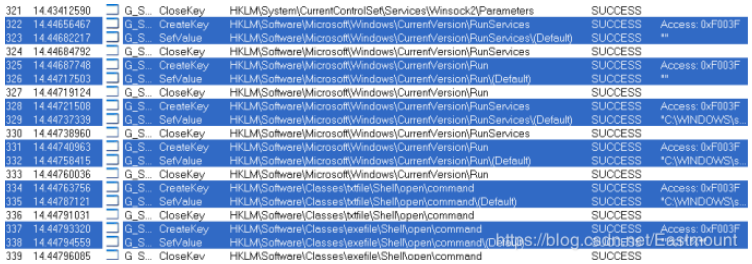


3.注册表监测

从Regmon我们可以看出，木马把KERNEL32.EXE注册成了服务。并把KERNEL32.EXE注册为开机启动。

HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices(Default) SUCCESS “C:\WINDOWS\system32\KERNEL32.EXE”

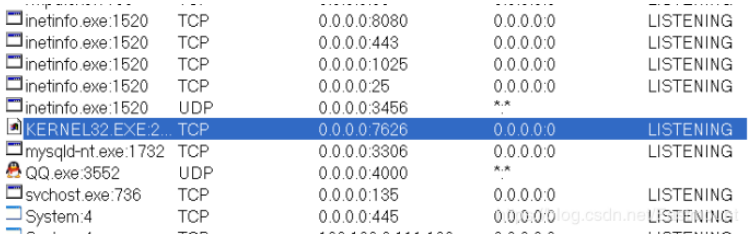
如下图所示：



另外，木马还修改了 .TXT 文件的关联， sysexplr.exe和TXT文件关联。即使删除了Kernel32.exe，但只要你打开 TXT文件，sysexplr.exe就会被激活，它将再次生成Kernel32.exe，于是冰河又回来了。

4.系统通信端口监测

通过TCPView监测 KERNEL32.EXE开启了TCP7626端口。如下图所示：



inetinfo.exe:1520	TCP	0.0.0.0:8080	0.0.0.0:0	LISTENING
inetinfo.exe:1520	TCP	0.0.0.0:443	0.0.0.0:0	LISTENING
inetinfo.exe:1520	TCP	0.0.0.0:1025	0.0.0.0:0	LISTENING
inetinfo.exe:1520	TCP	0.0.0.0:25	0.0.0.0:0	LISTENING
inetinfo.exe:1520	UDP	0.0.0.0:3456	*:*	
KERNEL32.EXE:2...	TCP	0.0.0.0:7626	0.0.0.0:0	LISTENING
mysqld-nt.exe:1732	TCP	0.0.0.0:3306	0.0.0.0:0	LISTENING
QQ.exe:3552	UDP	0.0.0.0:4000	*:*	
svchost.exe:736	TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
System:4	TCP	0.0.0.0:445	0.0.0.0:0	LISTENING

接着需要分析木马样本外部特征，包括文件特征、注册表特征、进程特征、端口特征等。

文件特征：

创建文件	文件路径	文件类型
KERNEL32.EXE	C:\Windows\system32	只读
SYSEXPLR.EXE	C:\Windows\system32	只读

注册表特征：

路径	键名	文件类型
HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices\	Default	C:\WINDOWS\system32\KERNEL32.EXE
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\	Default	C:\WINDOWS\system32\ KERNEL32.EXE

进程特征：

进程名
KERNEL32.EXE

开启端口：

端口类型	端口号
TCP	7626

该木马的清除方法如下：

- 删除C:Windows\system下的Kernel32.exe和Sysexplr.exe文件。
- 删除注册表HKEY_LOCAL_MACHINE/software/microsoft/windows/ CurrentVersionRun下键值为C:/windows/system/Kernel32.exe。
- 删除注册表HKEY_LOCAL_MACHINE/software/microsoft/windows/ CurrentVersion/Runservices下键值为C:/windows/system/ Kernel32.exe。
- 最后改注册表HKEY_CLASSES_ROOT/txtfile/shell/ open/command 下的默认值，由中木马后的C: /windows/system/Sysexplr.exe %1改为 正常情况下的C:/windows /notepad.exe %1，即可恢复TXT文件关联功能。

八.总结

在网络攻击日益复杂、恶意代码层出不穷的今天，机器学习算法在恶意代码检测中的应用逐渐受到学术界和众多安全厂商的重视。本文对基于机器学习算法恶意代码检测的技术方法和主流方案进行了梳理和讨论，这一工作将为新型主机恶意代码检测技术的设计和实现提供重要参考。但该领域仍属于发展阶段，还存在着许多未来工作和挑战，对其归纳如下。

- 静态分析检测速度快、系统资源占用少，但随着代码混淆、加壳等反检测技术的发展，静态分析的准确性受到一定程度的限制。动态分析技术需要运行被测代码，在效率上存在局限性。一个主流的发展方向是将静态、动态分析技术进行有效结合，全方位地对待测代码进行评估。
- 机器学习算法可以提供高准确率的恶意代码分类，但分类器一般作为黑盒机制被加以使用，安全人员缺乏对结果的理解。结果往往在不质疑分类器性质的情况下直接被使用，因此分类结果受经验阈值和数据特征的影响，出现一定倾向性。研究传统

量化分析（如准确率、误报率）之外的统计学方法，如可信度（credibility），科学地评价和比较底层的机器学习算法，是未来一项重要的研究工作。

最后希望基础性文章对您有所帮助，作者也是这个领域的菜鸟一枚，希望与您共同进步。同时，明天是教师节，感谢自己所有老师的教育与栽培，也祝自己节日快乐，哈哈！第四个教师节。

(By:Eastmount 2019-09-09 晚上10点 http://blog.csdn.net/eastmount/)