潜伏软件检测小结

## 基于特征码的间谍软件检测方法

目前间谍软件检测的主要方法之一是基于特征码的检测，这种方法维护一个特征库，保存已发现的间谍软件的代码特征。若待检测软件的特征存在于特征库中，则该软件被视为间谍软件。

M. Shaw等通过阻止本机向未知的或未授权的地址发起链接的方法来检测隐藏的间谍软件，但P2P应用也会向未知的地址发起链接，因此，该方法不足以区分正常的应用和间谍软件的区别 。

M. Wu等[8,9]使用预设的规则监控系统所有进程的（注册表、文件、进程、网络）Win API调用，并将重新安装或恢复被删除的自启动扩展点（ASEP）作为关键事件，检测自我恢复的间谍软件进程，并更新规则阻止间谍软件自恢复，最终终止、移除间谍软件进程。不过，该方法需要由第三方反间谍软件配合、提供首次发现间谍软件的信息。

由于间谍软件的特征是由安全专家人工分析获得，基于特征码的间谍软件检测方法具有极高的准确性和效率。然而，该方法对特征不在特征库中的新的或变种的间谍软件视而不见，即无法检测未知的间谍软件。在恶意软件新种和变种产生的数量越来越多、速度越来越快、难度越来越小的大背景下，该方法正在面临越来越多的挑战：

* **不能发现未知的间谍软件**：特征库中不可能有所有间谍软件的二进制码特征，尤其是那些仅针对单个人或单个组织的间谍软件的代码特征（这种情况在商业和国防领域中非常常见）。对于这些代码特征不在特征库中的间谍软件，基于特征码的检测方法无能为力。
* **具有滞后的先天不足**：从新的间谍软件出现，到反间谍软件厂商获得该间谍软件样本，分析出其特征，并更新到特征库，到用户将这个新的特征下载到本地的时间跨度很长。在这段时间中，间谍软件很可能已经造成了巨大的损失。

## 基于行为的间谍软件检测方法

潜伏软件检测的另一种方法是基于行为的检测。该方法不是去寻找特殊的代码特征，而是依据间谍软件的行为进行检测。该方法具有一定的普适性，能够检测部分未知的潜伏软件。但与此同时，这种方法容易被躲避，难免会产生较高的误判率。

Wang等利用间谍软件（随操作系统或多数常用应用程序）自启动的行为特点检测潜伏软件，但该方法是基于单一行为的检测，伪肯定率较高，检测由用户活动触发的间谍软件时效果不理想。

为了解决上述问题，Wang使用“outside the box”的思想，分别从干净系统和感染（rootkit）系统中获取文件快照（利用“dir”命令），对比、分析其中的差异，用来发现存在文件隐藏行为的（file-hiding）软件（如keylogger等）。该方法可以检测部分未知的、通过隐藏文件达到非法目的的间谍软件；但其无法保证在获取参照点时，系统是干净的（即未被感染的），且无法区分正常的软件。

Arastouie介绍了一种windows操作系统下的反间谍软件，该软件通过采集大量系统进程信息，并利用预先设置的静态规则对信息进行分析以发现间谍软件。该方法属于静态方法，缺乏自适应性，且对具有隐藏性的间谍软件检测能力较弱，因此，误报率和漏报率较高。

Wang通过监测作用在数据上的程序动作（复制粘贴和传送），并使用逆向云发生器表达程序的兴趣。如果兴趣集中在敏感数据上，该程序很可能是间谍软件。但该方法依赖于对所检测动作的归纳和分类，而分类要人为进行，存在一定的主观性；同时，该方法的监控动作不宜实施。

N. Lavesson利用学习算法学习软件发行商提供的软件附带的终端用户许可协议（EULA），并根据学习到的分类知识区分合法软件和间谍软件，使得间谍软件无法安装。虽然各主机会定期下载分类知识，但该方法仍对不含EULA的软件无能为力，尤其是未知/零日攻击。

### 基于网络流量的检测方法

Borders等人提出基于流的方法，根据训练获得的正常的web请求模式，分析主机外发的HTTP流量，发现潜在的间谍软件。本方法对不使用HTTP协议的间谍软件和混淆在用户合法行为中（blend in with legitimate behavior）的间谍软件无能为力。

Borders等利用VM技术向待检测系统中注入honeytoken（已知的网络请求序列），并将这些序列与实际的网络负载进行比较。若实际的网络负载与预期的不符，则产生报警。该方法利用白名单降低伪肯定率。但是，静态的honeytoken容易被间谍软件发现规律，并无视其刺激；且无法发现不使用网络发送敏感信息的间谍软件。

针对间谍软件存在将恶意行为隐藏在合法行为中的特点，Chandrasekaran 等使用一种随机honeytoken方法模拟用户的网络活动，以此诱导间谍软件的网络活动，通过分析这些夹杂异常活动的网络流量与正常网络流量的差异来检测间谍软件。该方法通过诱导间谍软件的行为来进行检测，可以发现部分隐藏的间谍软件，但其同样对不使用网络发送敏感信息的间谍软件则无能为力[18]。

H. Wang等提出一种称为NetSpy的方法，依据用户行为产生的流量信息，检测以浏览器为目标的间谍软件，并引入白名单来过滤正常软件的流量。NetSpy首先用预定输入获取干净系统的外发网络流量；接着，用同样的输入，从感染系统中获取流量信息；然后，采用差别分析方法分离不可信程序生成的流量，并将其内容与用户输入关联，发现间谍软件；最后，方法自动生成间谍软件的网络级别（network level）的特征。但是该文提出的方法仅能检测感染IE（Internet Explorer）浏览器的基于HTTP的间谍软件；且NetSpy在每个程序安装后运行，故无法有效发现多个、时间触发的或特定输入触发的间谍软件；无法保证在获取参照点时，系统是未被感染的。

### 基于虚拟机技术的检测方法

针对间谍软件通常从用户输入窃取隐私和机密信息的特点，Lim使用抗截屏技术的虚拟键盘来对抗间谍软件，但其仅能防御从用户输入设备获取敏感信息的间谍软件，不能防御类似于IE插件的间谍软件。

Provos等通过在虚拟机（VM）中打开被挂马的网页，监测HTTP fetches（利用其他反病毒引擎）及VM状态的变化（如创建进程、注册表及文件改变），检测在用户不知情的情况下，自动安装的基于web的恶意软件。该文提出的方法依赖第三方反病毒引擎，主要是预防基于web的间谍软件感染，对隐蔽性强的间谍软件（如模仿攻击）效果不明显。

### 基于信息流的检测方法

由于污点传播分析法能刻画和分析程序对数据的非法访问和使用，该方法成为基于行为的信息窃取软件检测方法的研究热点。

Kirda等结合动静态行为分析方法，提取、分析待检测组件的控制流，来检测基于IE浏览器辅助对象（BHO）或工具栏接口的间谍软件，其中：1）动态分析方法通过监测浏览器函数调用发现插件是否在收集信息；2）静态分析方法提取可疑插件的控制流，分析其二进制码中是否出现导致信息泄露的库函数调用。该方法检测由用户活动触发的间谍软件时效果不理想，且局限于特殊的环境（基于IE的BHO），方法不具有一般性；同时，静态分析易被混淆技术逃避，且仅能发现是否有信息泄露，不能发现是何信息、泄露向何处 。

Li等提出干扰而非检测的方法，在敏感信息出现（敏感区域）时，通过实施用户定义的安全策略，保证敏感信息流向可信的程序（组件），能防止间谍插件通过约定的接口（如API）获取敏感信息或直接读取主程序的内存以获取敏感信息。但该方法仅能干扰插件式的间谍软件，对独立的间谍软件（如keylogger）无能为力。

M. Egele 等 利用动态污点传播分析的方法监控敏感信息的流动，当有插件试图将敏感信息泄露到浏览器地址空间之外时，则认为该插件为间谍软件，同时对敏感信息进行追踪以获取其去向。但该方法仅针对基于IE浏览器辅助对象（BHO）的间谍软件，具有较大的环境局限性。

JiaJing Li等利用静态污点分析法来检测具有信息窃取行为的间谍软件，该方法能够发现隐藏的恶意行为。不足：

李佳静等从软件中的隐式流的角度，使用静态污点传播分析方法，检测截获用户按键信息的间谍软件，实验表明该方法比显式流敏感的方法有更低的漏报率，并能有效地发现需要特定条件触发的信息窃取行为，但静态污点分析不能对抗混淆技术。

### 基于诱导的检测方法

Saroiu提出一种SpySaver方法，利用VM模拟用户上网活动，生成大量虚假信息，来降低间谍软件所窃取信息的价值，达到混淆的目的。但此方法容易被间谍软件识破，且产生大量虚假信息，导致系统负载过大。

Han 等利用被称为“HoneyID”的诱导技术来实现对未知间谍软件的检测，通过多角度地伪造用户活动（如键盘、鼠标、网络活动），引诱间谍软件实施恶意行为。进而基于引诱阶段和非引诱阶段动作的不同，计算伪造事件的响应比，以此发现潜在的间谍软件。该方法可以检测到利用用户或主机信息的会话间谍软件（dialog spyware），但是，正常的软件亦会有键盘、鼠标、网络等行为，而该方法采用的白名单方式不足以区分未知的软件是否属于间谍软件（且该方法必须有待检测间谍软件敏感的行为的先验知识），从而限制了该方法的应用范围。

## AIS检测方法

基于行为的间谍软件检测方法取得了一定的效果，但是，仅依赖单一行为（设置钩子等行为）的检测方法的伪肯定率较高；而多行为（击键跟踪、文件访问和网络通信等）关联的方法由于使用固定的关联时间窗口和较简单的关联算法，使得检测容易被躲避。

J. Fu 使用人工免疫系统（AIS）中的树突状细胞算法（Dendritic Cell Algorithm，DCA），对键盘记录器击键跟踪、文件访问和网络通信三种行为进行智能的关联，发现被感染主机中的键盘记录器恶意软件。由于使用了可变的关联时间窗口，并考虑了上述三种行为的时序关系，该方法在一定程度上提高了键盘记录程序的检测率，且伪肯定率较低。但攻击者也可以利用DCA关联行为的特点，通过降低恶意行为频率的方法逃避DCA对其的检测，在用户输入的句子较长时，逃避效果尤其明显。

为了解决上述问题，针对键盘记录程序具有隐藏恶意行为的特点，J. Fu 在文献的基础上，分析了键盘记录程序逃避检测的机制，并引入诱导机制，通过模拟用户击键，提高按键（包括回车键等特殊键）的频率，诱导键盘记录器程序表现更明显的恶意行为，并使用树突状细胞算法对“放大”后的行为进行智能关联。初步实验结果显示，该方法具有一定的可行性，使键盘记录程序逃避检测更加困难。

1. M. Shaw, S. D. Gribble. Reverse Firewalls in Denali, 2002
2. Wang et al. [Gatekeeper: Monitoring Auto-Start Extensibility Points (ASEPs) for Spyware Management]((2004)%20Gatekeeper：Monitoring%20Auto-Start%20Extensibility%20Points%20(ASEPs)%20for%20Spyware%20Management.pdf),2004
3. Wang et al. [Strider GhostBuster: Why It’s a Bad Idea for Stealth Software to Hide Files]((2004-T)%20Strider%20GhostBuster：Why%20It's%20a%20Bad%20Idea%20for%20Stealth%20Software%20to%20Hide%20Files.pdf), 2004-T
4. Borders et al. [Web Tap：Detecting Covert Web Traffic]((2004)%20Web%20Tap：Detecting%20Covert%20Web%20Traffic.pdf), 2004
5. Kirda et al. [Behavior-based Spyware Detection]((2006)%20Behavior-based%20Spyware%20Detection.pdf), 2006
6. Borders et al. [Siren: Catching Evasive Malware]((2006)%20Siren：Catching%20Evasive%20Malware.pdf), 2006
7. H. Wang, S. Jha et al. [NetSpy: Automatic Generation of Spyware Signatures for NIDS]((2006)%20NetSpy：Automatic%20Generation%20of%20Spyware%20Signatures%20for%20NIDS.pdf), 2006
8. M. Wu, Y. Huang, Y. Wang et al. [A Stateful Approach to Spyware Detection and Removal]((2006)%20A%20Stateful%20Approach%20to%20Spyware%20Detection%20and%20Removal.pdf), 2006
9. M. Wu, Y. Wang et al. [Self-Healing Spyware: Detection, and Remediation]((2007-J)%20Self-Healing%20Spyware：Detection,%20and%20Remediation.pdf),2007
10. Lim, [Defeat Spyware with Anti-screen Capture Technology using Visual Persistence]((2007)%20Defeat%20Spyware%20with%20Anti-screen%20Capture%20Technology%20using%20Visual%20Persistence.pdf), 2007
11. Provos et al. [The Ghost in the Browser-Analysis of Web-Based Malware]((2007)%20The%20Ghost%20in%20the%20Browser%20Analysis%20of%20Web-Based%20Malware.pdf),2007
12. Li et al. [SpyShield: Preserving Privacy from Spy Add-Ons]((2007)%20Spyshield：Preserving%20Privacy%20from%20Spy%20Add-Ons.pdf),2007
13. Chandrasekaran et al. [SpyCon: Emulating User Activities to Detect Evasive Spyware]((2007)%20SpyCon：Emulating%20User%20Activities%20to%20Detect%20Evasive%20Spyware.pdf),2007
14. M. Egele, C. Kruegel, E. Kirda et al. [Dynamic Spyware Analysis]((2007)%20Dynamic%20Spyware%20Analysis.pdf),2007
15. J. Jensen. Detection of Hidden Software Functionality,2007
16. Saroiu et al. [SpySaver: Using Incentives to Address Spyware]((2008)%20SpySaver：Using%20Incentives%20to%20Address%20Spyware.pdf),2008
17. Arastouie et al. [Hunter: An Anti spyware for windows Operating System]((2008)%20Hunter：An%20Anti%20spyware%20for%20windows%20Operating%20System.pdf),2008
18. Han et al. [HoneyID：Unveiling Hidden Spywares by Generating Bogus Events]((2008)%20HoneyID：Unveiling%20Hidden%20Spywares%20by%20Generating%20Bogus%20Events.pdf),2008
19. Wang et al. [Interests-based Spyware Detection]((2009)%20Interests-Based%20Spyware%20Detection.pdf),2009
20. Li et al. [A Static Method for Detection of Information Theft Malware]((2009)%20A%20Static%20Method%20for%20Detection%20of%20Information%20Theft%20Malware.pdf),2009
21. N. Lavesson, M. Boldt et al. [Learning to Detect Spyware using End User License Agreements]((2010-J)%20Learning%20to%20Detect%20Spyware%20using%20End%20User%20License%20Agreements.pdf),2010
22. 李佳静 et al. [一种隐式流敏感的木马间谍程序检测方法]((2010-J)%20一种隐式流敏感的木马间谍程序检测方法.pdf),2010
23. Mykola Krasnostup, Dennis Kudin, Anti-Spyware: Efficiency of the Means of Defense, <http://www.keylogger.org/articles/mykola-krasnostup/anti-spyware-efficiency-of-the-means-of-defense-13.html>
24. J. Fu, Y.W. Liang, C.Y. Tan. Detecting Software Keyloggers with Dendritic Cell Algorithm, International Conference on Communications and Mobile Computing 2010, IEEE.
25. J. Fu, Y.W. Liang, C.Y. Tan et al. Mimicking User Keystrokes to Detect Keyloggers with Dendritic Cell Algorithm