基于文件几何学的恶意软件文件聚类以及利用 R 语言的可视化

非官方中文译本·安天实验室 译注

文档信息			
论文题目	Malware files clustering based on file geometry and visualization using R language		
论文作者	惠普公司		
发布单位	惠普公司		
原文链接/出处	http://h30499.www3.hp.com/hpeb/attachments/hpeb/off-by-on-software-security-blog/335/1/Security%20Briefing%2014%20companion%20report.pdf		
论文发布日期	2014/6	**译文发布日期**	2014/10/8
论文摘要&关键词	**摘要**：本文介绍了一种实用的方法并展示文件分类中使用的基本原则。直观地看，值得注意的是，一个恶意软件家族的变种通常源于一个相似的代码库并展现了相似的文件结构。当通过易用工具和 R 语言中使用的可视化技术应用这一概念时，它能帮助我们分析并标注一组输入文件，也为恶意代码的进一步识别与处理提供了一个良好的基础。 **关键词**：文件几何学；聚类；R 语言；可视化		
译者	安天技术公益翻译组	**校对者**	安天技术公益分析组
免责声明	• 本译文译者为安天实验室工程师，本文系出自个人兴趣在业余时间所译，本文原文来自互联网的公共方式，译者力图忠于所获得之电子版本进行翻译，但受翻译水平和技术水平所限，不能完全保证译文完全与原文含义一致，同时对所获得原文是否存在臆造、或者是否与其原始版本一致未进行可靠性验证和评价。 • 本译文对应原文所有观点亦不受本译文中任何打字、排版、印刷或翻译错误的影响。译者与安天实验室不对译文及原文中包含或引用的信息的真实性、准确性、可靠性、或完整性提供任何明示或暗示的保证。译者与安天实验室亦对原文和译文的任何内容不承担任何责任。翻译本文的行为不代表译者和安天实验室对原文立场持有任何立		

	场和态度。 • 译者与安天实验室均与原作者与原始发布者没有联系，亦未获得相关的版权授权，鉴于译者及安天实验室出于学习参考之目的翻译本文，而无出版、发售译文等任何商业利益意图，因此亦不对任何可能因此导致的版权问题承担责任。 • 本文为安天内部参考文献，主要用于安天实验室内部进行外语和技术学习使用，亦向中国大陆境内的网络安全领域的研究人士进行有限分享。望尊重译者的劳动和意愿，不得以任何方式修改本译文。译者和安天实验室并未授权任何人士和第三方二次分享本译文，因此第三方对本译文的全部或者部分所做的分享、传播、报道、张贴行为，及所带来的后果与译者和安天实验室无关。本译文亦不得用于任何商业目的，基于上述问题产生的法律责任，译者与安天实验室一律不予承担。

HPSR安全简报播客第14期

安全简报

第14期，2014年6月

惠普安全研究（HPSR）

目录

第14期

感谢您订阅惠普安全研究的第 14 期安全简报。在这份简报中，我们探讨了文件几何可视化以及对使用 R 语言的恶意软件的聚类实验。

基于文件几何学的恶意软件文件聚类以及利用 R 语言的可视化

在过去的十多年中，恶意软件的爆炸式增长给许多反病毒企业带来了一个严峻的挑战，即如何有效且精确的处理并标记这么一大批输入文件。随着云服务、大数据以及不断增长的计算能力的出现，许多人将希望寄托于机械学习分类算法。本文将介绍一种实用的方法并展示文件分类中使用的基本原则。直观地看，值得注意的是，一个恶意软件家族的变种通常源于一个相似的代码库并展现了相似的文件结构。当通过易用工具和 R 语言中使用的可视化技术应用这一概念时，它能帮助我们分析并标注一组输入文件。它也为恶意软件的识别与处理的进一步研究与实验，提供了一个好的基础。后文将介绍一种实用的方法，并连同大量有关恶意软件可视化和利用 R 语言及其他工具的聚类。

背景--PE 文件

在 Windows 系统中绝大多数可执行文件都是 PE 格式。PE 文件格式有种常规结构。这一规定是由操作系统下载器及其执行框架所强加的。操作系统下载器的要求可被看作一种多维度网维或过滤器，只能接受符合其预定格式的对象。这一标准为我们创造了机会，能够观察基于几何特性的文件之间的相似性，这种特征能够确保文件的结构符合操作系统下载器的要求。

PE 文件的结构在众多出版物中都是有据可查的，以微软的官方规范为首，但也有许多文章揭露了这些规范的来龙去脉（详细信息可见报告最后的“延伸阅读”部分）。

文件按照数据平流被组织起来。在文件的开头，有一系列被称为标头的结构。第一个结构被称作 MS-DOS 标头，主要源于老版的 DOS。这是一个弊端，如果运行在不支持 PE 文件的模式（真实模式）中，就会退出文件。值得注意的是，PE 文件通常运行于受保护的操作系统模式中，其中每一个内存区域都有相关的关键词来通过运行的程序管制访问权限。MS-DOS 标头后面就是 PE 文件签名——是一个 4 字节的序列，标识了 PE 文件的开端。PE 文件签名后面是 PE 文件标头和一个可选标头。紧接其后就是分区标头和实质的分区主体。这些标头包含了众多在识别文件中可利用的独特信息。

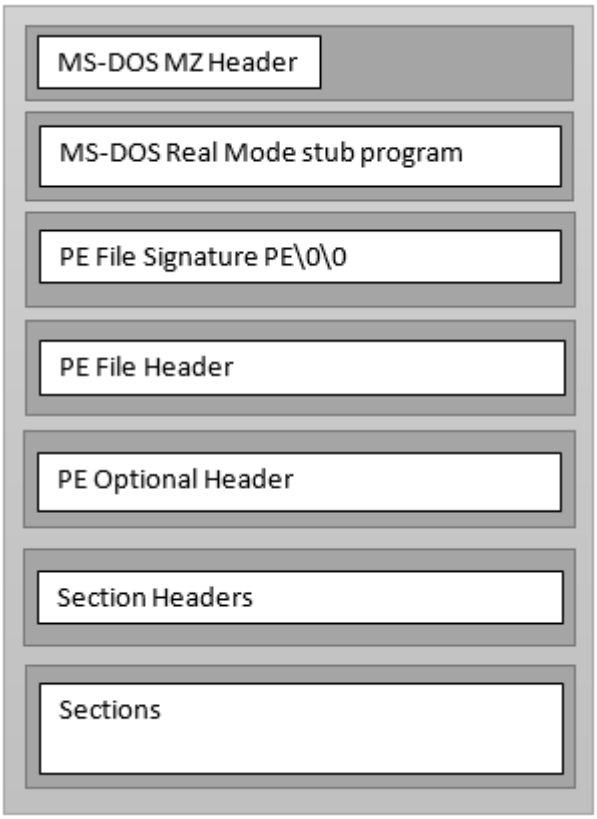

图1：PE文件格式分布草图

该领域的研究

这一领域已经有很多研究了，对于基于不同机器学习算法和文件选择功能的恶意软件分类也有广泛的研究成果。大多数算法都需要相关的培训并测试样本集。若能合理调谐，这些算法能够完全自动化并且具有较高效率。但是，如果它们成为恶意软件作者的目标，则很容易出现故障。本文将探讨设计 PE 文件属性可视化的交互方法。也探讨了无监督学习聚类技术，这一技术能够省略培训集并将人际交往带入分类的过程中，因此在恶意软件分析和分类中生成更大的灵活性和智能保障

我们的研究--恶意文件聚类和可视化

一组文件中，会有哪一种适用于聚类算法的特性吗？如果有，那这一信息稍后可被用来识别检测中新出现或之前没见过的恶意文件。以下文件属性的初始设置被选作参考：

- 入口点地址
- 分区的数量
- 代码长度
- 图像大小
- 第一、二、三分区的虚拟尺寸及原始尺寸

最后决定，最佳的方式是创建一个图形来展示生成的属性，其允许二维空间的数据相互聚类。对于这种情况，平行坐标图更加适用。我们发现，利用 R 语言及其架对可视化有很大的帮助。R 是一种为统计计算和图像服务的语言和环境，它能提供各种各样的统计技巧，例如线性和非线性模式、时间序列分析、分类、聚类等等。R 语言还具有强大的图形可视化功能，并具有高度可拓展性。

方法

将不同的可视化和聚类技术应用于多组已知恶意软件家族文件中，并对有效分组这些文件的功效进行评估。这些分析方法也被应用于一组干净文件中，以作对比。

有关工具的说明：本文研究的意图不仅仅是利用文件几何学和可视化聚类恶意文件，也是想以这种方式与其他研究学者分享这些研究结果，进而对这些实验进行复制研究。因此，选用了这些开源的或是在 GNU 公共许可下的软件工具。

案例研究--Gamarue 与 Ursnif 的外形

Gamarue--解析与绘制实例

在应用可视化与聚类之前，我们要能够解析 PE 文件并提取其属性至可读文件格式。市面上有许多产品具有上述功能。其中一个就是 PeStudio，它能生成含有批命令行模式属性的 XML 文件。这是一种简便方法，因为许多可视化产品都能解析 XML 文件，例如我们选择的 R 语言。作为例子，我们来看看由各种安全产品检测到的 Gamarue。Gamarue 是一种传播相对普遍的恶意软件家族，通过可移动驱动进行传播并允许攻击者远程控制受害者的机器。

在我们的文件集中运行 PeStudio，能够产生一系列包含文件属性的 XML 文件（图 2）。

```
cluster_sets\gamarue>for %i in (*) do c:\Cluster\PeStudio828\PeStudioPrompt.exe -file:%i -xml:xml\%i
```

图2：PeStudioPrompt.exe的批量模式

我们可以利用 XML 浏览器快速查看其属性。它提供了一种简便的方法，能够通过 XML 文件快速导航并查看文件属性。

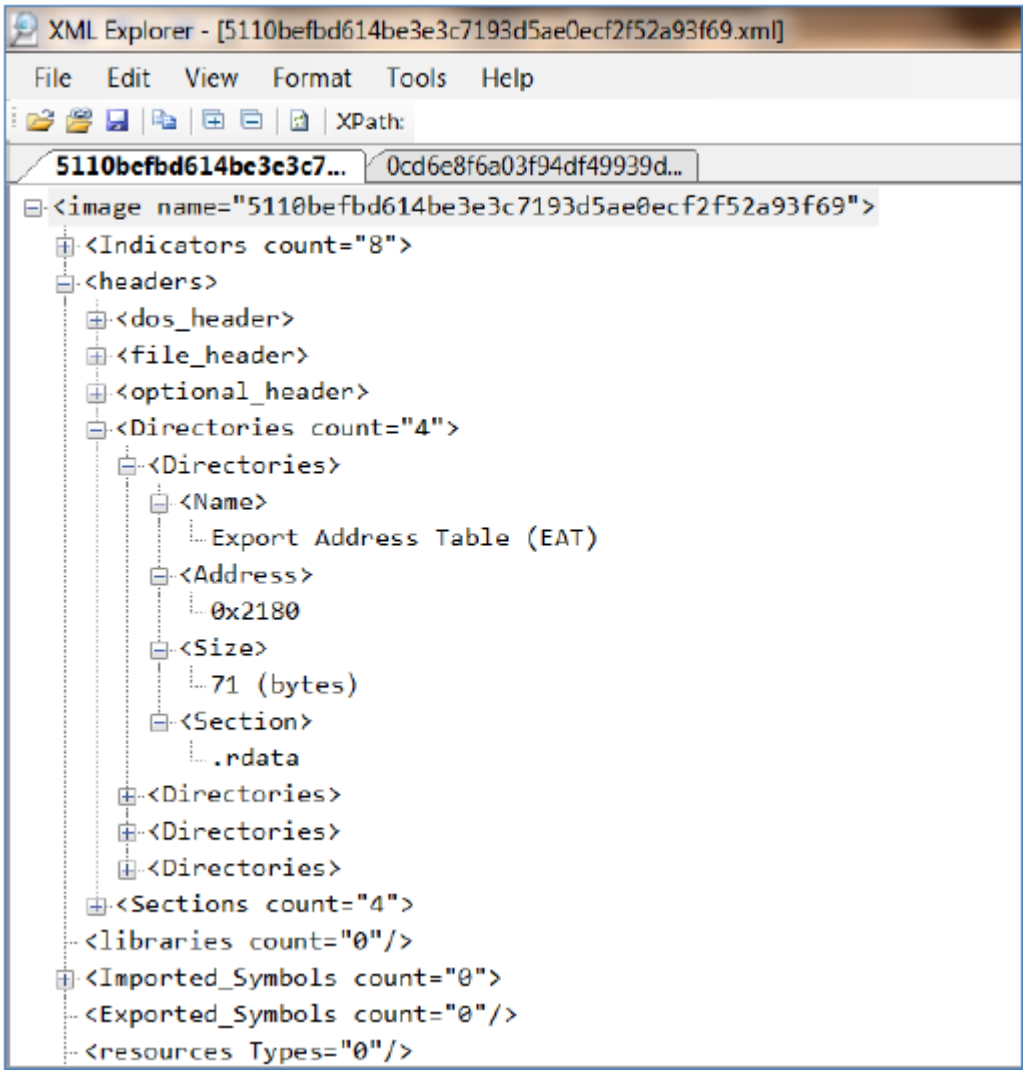

图 3：用 XML 浏览器查看到的 XML 文件数据

利用 R 语言，我们能阅读包含数据框架属性的 XML 文件。由此我们利用这些数据集生成一个平行坐标图。平行坐标图是一种包含众多垂直顶点的图表，其中各顶点代表一个单独的变量。这种可视化使我们能够基于每一顶点所选的属性来估量文件的相似性。选择初始属性集是因为它们能与代码功能紧密耦合，例如入口点地址、区段编号、代码长度、图像大小以及第一、二、三区域的虚拟和原始大小。在多数 PE 文件中，这些属性都是可用的，也不需要额外的复杂处理，例如文件解压或行为分析、或在文件分区中分析数据流。利用至少一种反病毒产品来处理 PE 文件集，检测到 Gamarue 蠕虫，就能生成下面的平行坐标图。

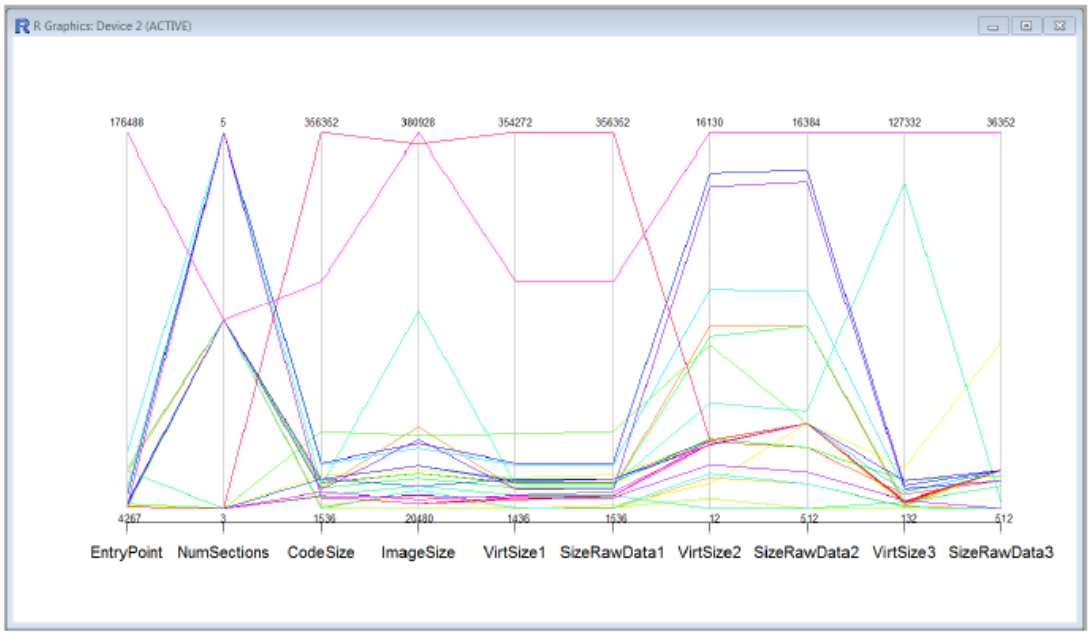

图 4：基于一系列文件属性的平行坐标图--Gamarue 蠕虫家族

图中的每一条线代表着一个文件。每一个顶点代表着一个文件的属性，我们此应用于平行坐标图可视化中。尽管图中显示了基于所选属性的文件聚类，但是这也使得追踪每一个独立的文件变得更加困难——尤其是文件增多的时候。这种情况下，引用交互式平行坐标图就大有帮助了，其中一组线能被手动突出。

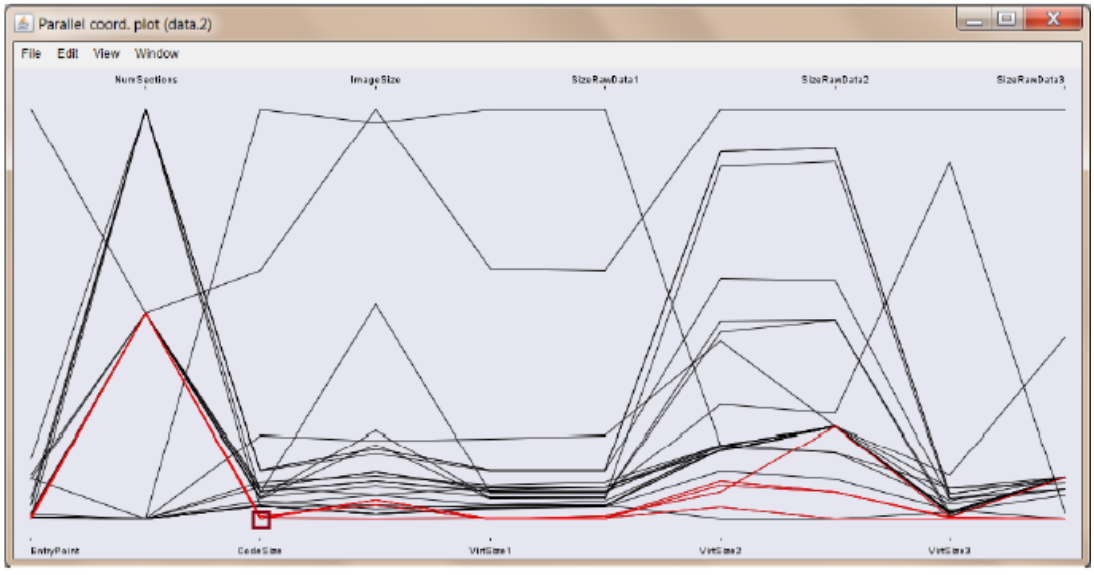

图 5：交互式平行坐标图，选取具有相似 CodeSize 的文件--Gamarue 蠕虫家族

由图可见，CodeSize 的选择标准创建了一个相对紧密耦合的聚类，这也包含了具有相似 EntryPoint 值、VirtualSize1、SizeOfRawData1、VirtualSize3 的文件。但是在 VirtualSize2、SizeOfRawData2 和 SizeOfRawData3 中也存在细微的差别。或者，在第三区域选择具有相似的虚拟大小值的文件时，聚类就有些分散，但也包含大量的文件。

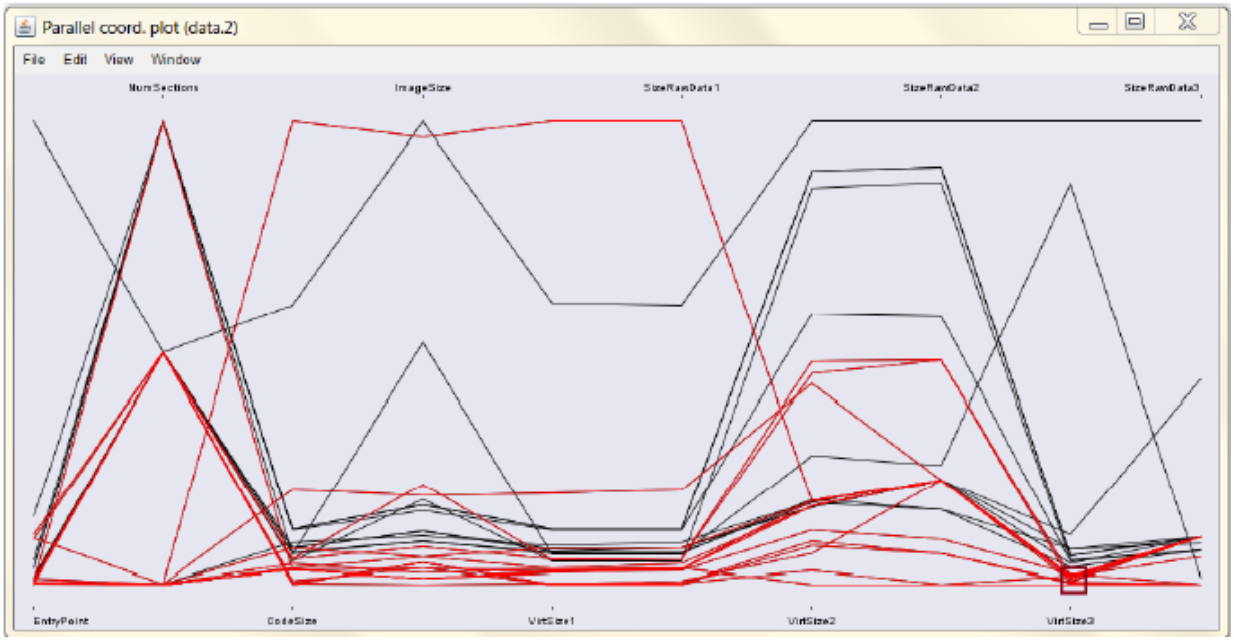

图 6：交互式平行坐标图，选择第三区域中虚拟大小值相似的文件--Gamarue 蠕虫家族

当比对这些结果时，我们应注意：即使所选的文件来自单一的病毒家族（Gamarue），当试图基于文件几何学对其聚类时，这些文件也展示了差异性。

这一实验显示，用于组建平行坐标图的文件属性的选择对于可视化的质量是至关重要的，并且这些属性也可能因所属家族的不同而不同。由文件解析器所提供的输出结果可应用于 XML 或任何其他可读格式，对于这种可视化可检查并利用大量的属性。

Ursnif--目标聚类实例

另一种数据可视化的方式就是应用目标聚类。这是一种无监督学习技术，着眼于独立的属性空间并尝试识别属性的组别和类别。

我们来检查一下 Ursnif 家族的恶意软件。Ursnif 是一种广泛传播的木马，能够窃取敏感信息。该木马能够感染 32 位和 64 位的 Windows 平台，通常在其资源区携带组件（例如 DLL）。这一行为特点能够提供一些有趣的文件几何学知识。通过一组由不同反病毒产品检测到 Ursnif 的 32 位文件，我们能够生成一个展示聚类的交互式平行坐标图。

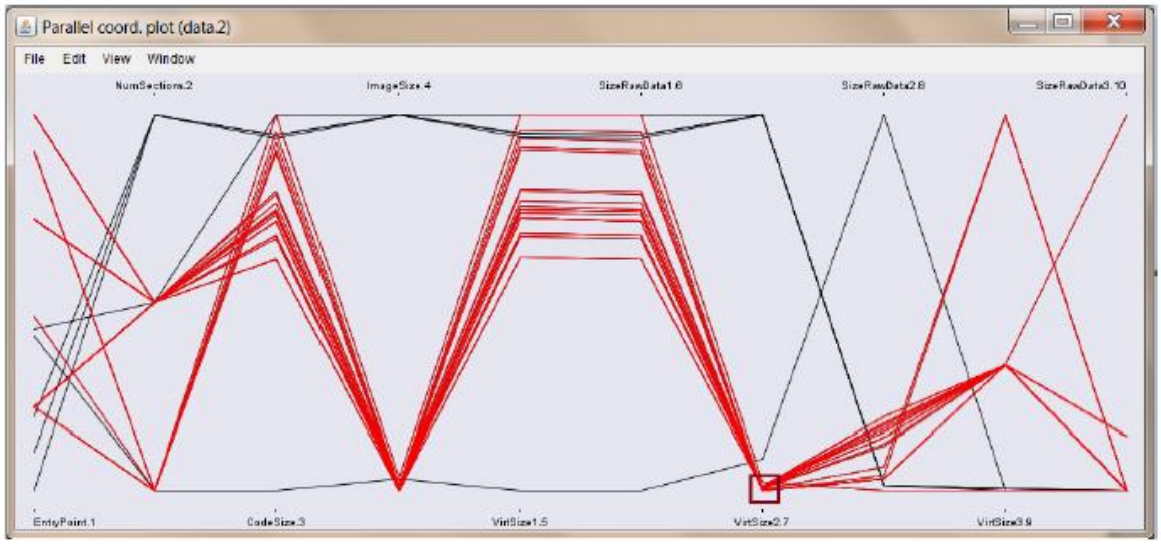

图 7：Ursnif 家族的交互式平行坐标图--突出显示的文件按照分区 2 的 VirtualSize 归类

聚类的基本原则就是把目标分组，使一个组内的属性共性较高、跨组别的属性共性较低。基于不同的聚类方法，有很多 R 语言内可用的算法，例如分区、分层、基于密度和基于网格的方法等。每一种方法都有不同的特点，并且对不同的数据框架实例都有不同的优势。

K 均值聚类是无监督学习分区方法中最普遍也最简单的一种方法。为了解 K 均值聚类算法的工作原理，简单起见，我们假设一个沙盒里有一把弹珠。我们怎样才能发现掩盖了弹珠的聚类呢？首先，我们需要限定预测的聚类数量。K 均值算法需要这个信息作为参数。然后，我们随机选择沙箱中聚类中心点的位置。我们计算出每个弹珠与相应的中心点之间的距离。再之后，我们计算出新的中心点，使其距离的平方误差和减少。一旦计算了新的中心点，我们再选择一组属于新中心点的弹珠并继续计算，直到新计算的中心点与原来一模一样。至此，我们算是发现了聚类的所有中心点。

如前面所述，K 均值算法的一个特点就是，它需要一个预测的聚类数量作为论据。这里，上述的平行坐标图可视化可能会有所帮助。例如，查看 Ursnif 家族的一组文件，平行坐标图能够告知我们所期待的聚类的数量。我们也可能评价用于聚类的 PE 文件属性的质量。由图 7 可见，在 Ursnif 池中，有 2 至 3 个聚类。

应用带有 Ursnif 数据框架的 K 均值聚类，得出：

```
ClusteringResults <- kmeans(ursnif_files_attributes, 3)
```

这里 3 是建议的群体个数。在 ClusteringResults 中，我们能够看到下述的可用构成要素：cluster, centers, totss, withinss, tot.withinss, betweenss, size, iter, ifault

下面是对这些要素在我们的案例中所代表的含义的简要描述：

- cluster – 是整数矢量。在我们的例子中，上述的聚类生成了下面的矢量：

```
> ClusteringResults$cluster
[1] 2 3 3 3 3 3 3 3 1 3 3 2 3 3 3 3 2 3 3 2 3 3 3 3 3 3 3
```

这就意味着元素总数是 27；第一个元素属于聚类 2，第二个元素属于聚类 3，而第九个元素属于聚类 1。类 1 只有一个元素。聚类 2 有 4 个元素，以此类推。

- centers – 是一组中心点的矩阵。上述例子中，中心点如下所示：

	entry_point	num_sections	size_of_code	size_of_image	virtual_size	size_raw_data	virtual_size2	size_raw_data2	virtual_size3	size_raw_data3
1	5359.000	3.000000	2048.00	102400.00	1641.00	2048.00	90112.000	66048.000	20.000	512
2	5045.750	4.750000	64256.00	1134592.00	63989.50	64256.00	1049859.250	1536.000	65.250	512
3	5496.455	3.727273	52410.18	79499.64	52388.55	52410.18	9767.227	8845.864	4655.455	768

图 8：聚类中每一属性空间的中心点

中心点是按照聚类算法而计算的。

- totss – 平方总和

```
> ClusteringResults$totss
[1] 7.472679e+12
```

- withinss –矢量，识别聚类中平方总和

```
> ClusteringResults$withinss
[1] 0  31174032 5696360435
```

注意：0说明第一个聚类中只包含一个元素。

利用聚类平方和，我们能更好地通过计算其总和以及划分图示中的数据来识别一系列聚类。

```
wss <- (nrow(data_cluster)-1)*sum(apply(data_cluster,2,var))
for (i in 2:5) wss[i] <- sum(kmeans(data_cluster, centers=i)$withinss)
plot(1:5, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")
```

在这种情况下，我们需要寻找一个独特的曲线，以指出数据框架中最有可能的聚类。

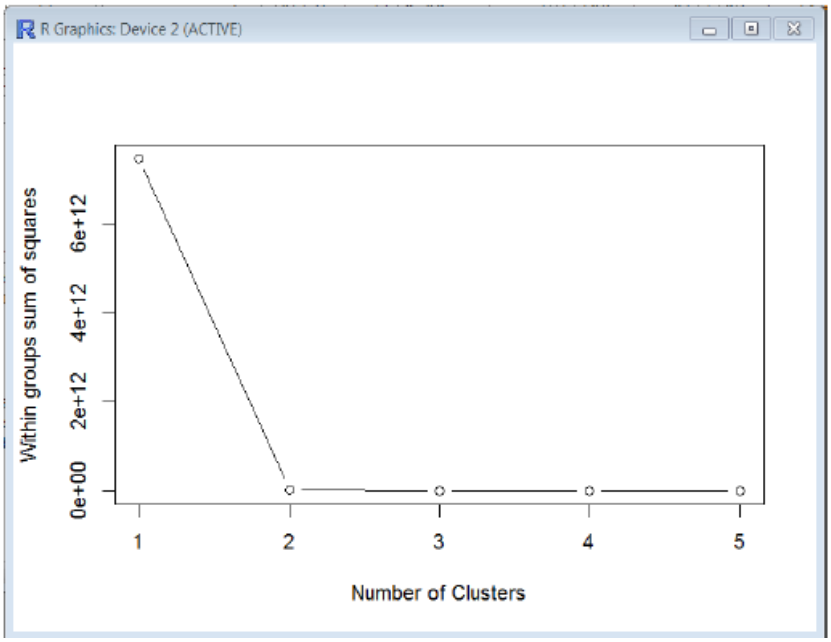

图 9：屏幕划分的平方误差之和

看似，我们需要分析 2 种或至多 3 种聚类。

- tot.withinss – 聚类中的平方总和
- betweenss – 聚类间的平方总和
- size – 每一聚类中的点数

> ClusterResults$size

[1] 1 4 22

- iter - 外迭代的数量
- fault – 表示算法问题的整数

我们可利用R语言可用clusplot()功能将聚类可视化，它能在现有的图形设备中画出二维聚类。

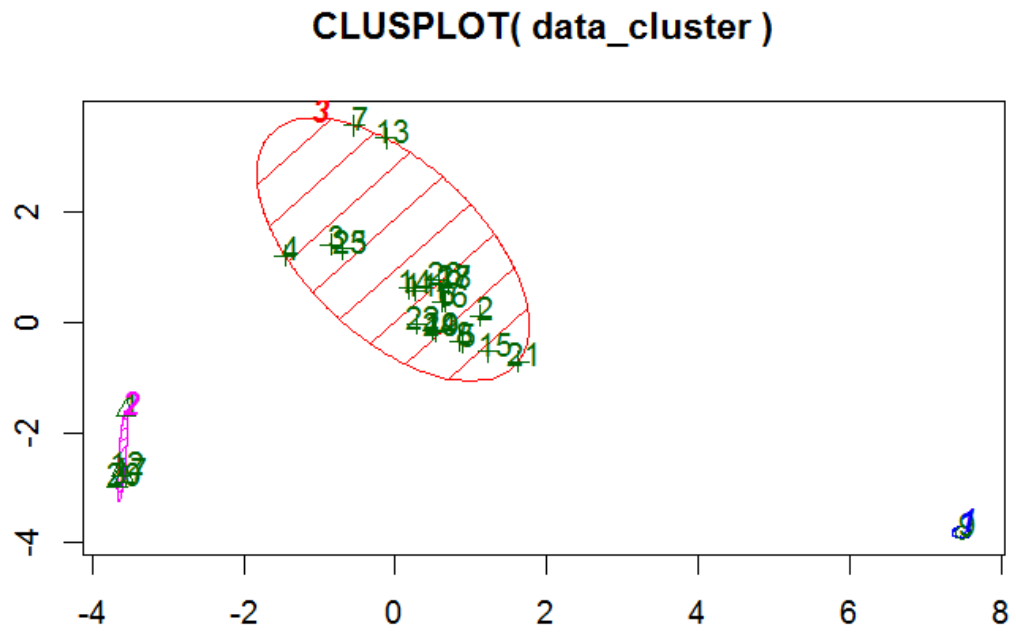

图 10 ：基于Ursnif家族文件属性的聚类划分

由上面的例子可见，文件属性能够为聚类提供坚实的基础。正如前面我们所注意到的一样，文件的几何形状很容易受到各种文件压缩器、压缩包以及自解压文件夹的影响。当我们准备待处理的文件集时，都应该将这些因素考虑在内。我们增加了属性的数量，来看看它是如何影响组别划分的。

增加4个不同的属性后，例如SizeOfInitializedData、CheckSum、SizeOfStackReserve和SizeOfHeapReserve，就会显示更多的可用聚类。单独的曲线将自身延伸到聚类中心点。

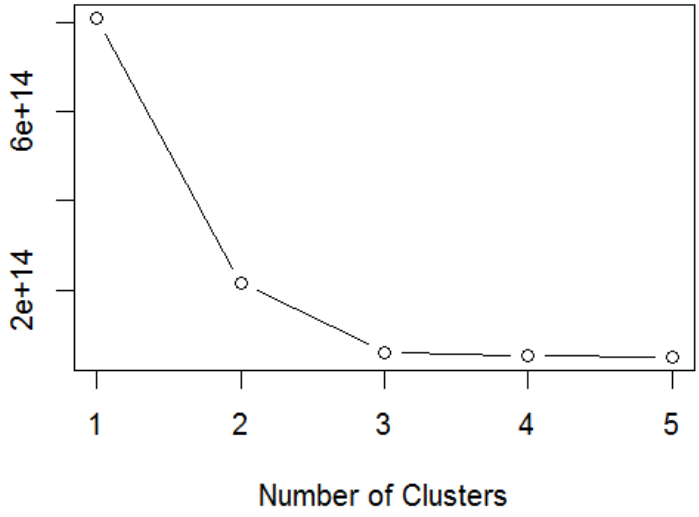

图11：预测的聚类数量

在4个聚类中心点运行K均值分区算法，生成了更加细粒度的集群，也使我们更深入地了解了样本的恶意软件家族分裂。例如，图10中的聚类3（红色阴影的椭圆区域）的分区被分成了2个子群，详见图12。这些结果说明，属性的数量和质量对于聚类的结果至关重要。

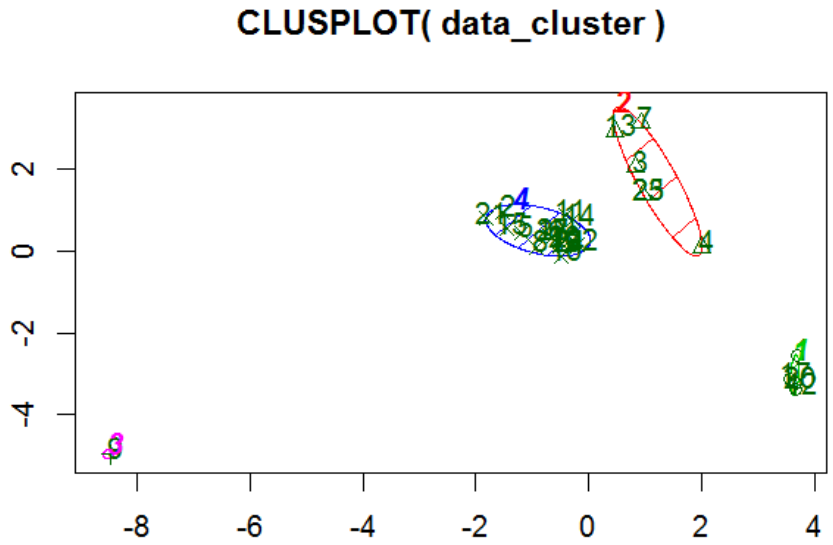

图12：基于Ursnif病毒家族扩展的属性集的聚类划分

两个恶意软件家族的文件对比实例

基于实例中的两个恶意软件家族——Ursnif 和 Gamarue，我们来探讨 K-mean 文件集聚类。平方误差划分的总和展示了约 10 个聚类中的独特曲线。

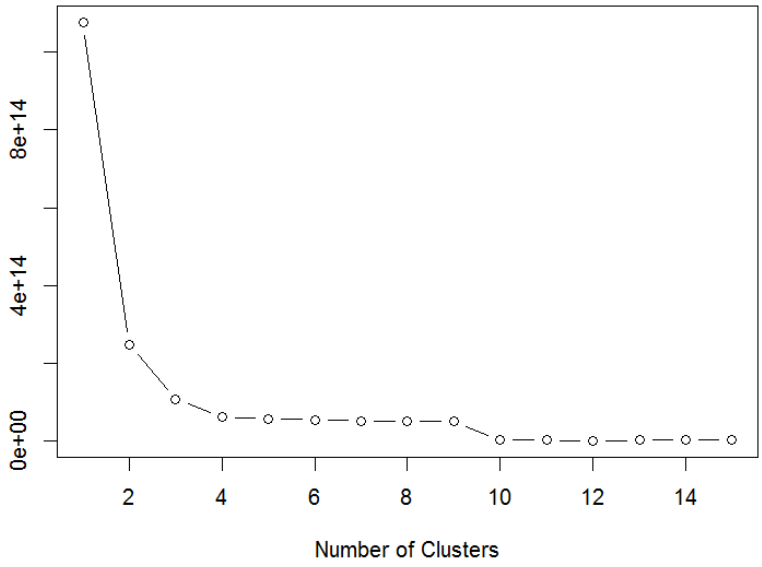

图13：基于误差平方和预测的Gamarue与Ursnif的聚类

在这10个聚类中运行聚类算法展示了聚类间的隔离。请看图14。

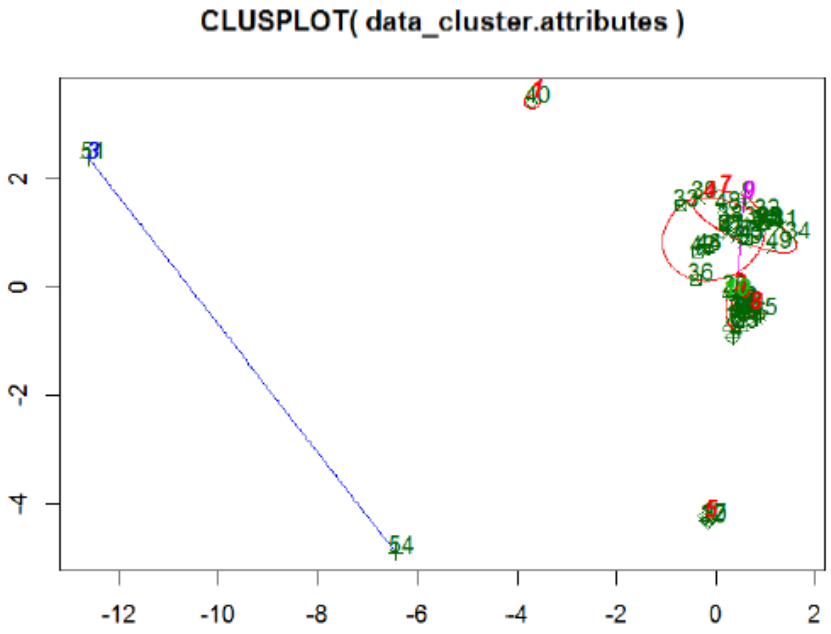

图14：两大恶意软件家族Gamarue和 Ursnif的聚类划分

聚类尺寸显示，有10个聚类含有目标1、13、2、13、4、2、11、4、2和2。

> ClusterResults$size

[1] 1 13 2 13 4 2 11 4 2 2

我们来看看不同的恶意软件家族是如何在创建的聚类中下跌的。首先，我们创建一个表格，其中待分析的恶意软件家族名称与每一个文件代表的属性相联系。然后，我们将该信息应用于之前创建的聚类中。

```
> table(data_cluster$Name, clus_results$cluster)
```

表 1： Gamarue和Ursnif聚类

	1	2	3	4	5	6	7	8	9	1
Gam	1	0	2	13	0	0	11	0	0	0
Ursn	0	13	0	0	4	2	0	4	2	2

由上表可见，我们列出10个聚类，其中Gamarue家族文件在聚类4和7之间分布良好。大多数Ursnif文件位于聚类2、5和8中，并与Gamarue家族相分离。Ursnif家族所占据的聚类之多可能意味着恶意软件所代表的文件集受到其几何形状的严重影响并包含了不同的变种。另一方面，Gamarue家族同文件集和聚类形成紧密耦合。

这些结果又一次突出了文件属性选择的问题。其中显示，仅依赖文件的几何实体是不足够的，人们需要对恶意软件特点挖掘的更深。

干净文件对比实例

现在，我们将Windows系统中的一些干净文件混入其中。运行的误差划分平方和显示出，在聚类12和13之间有一个特别的曲线。

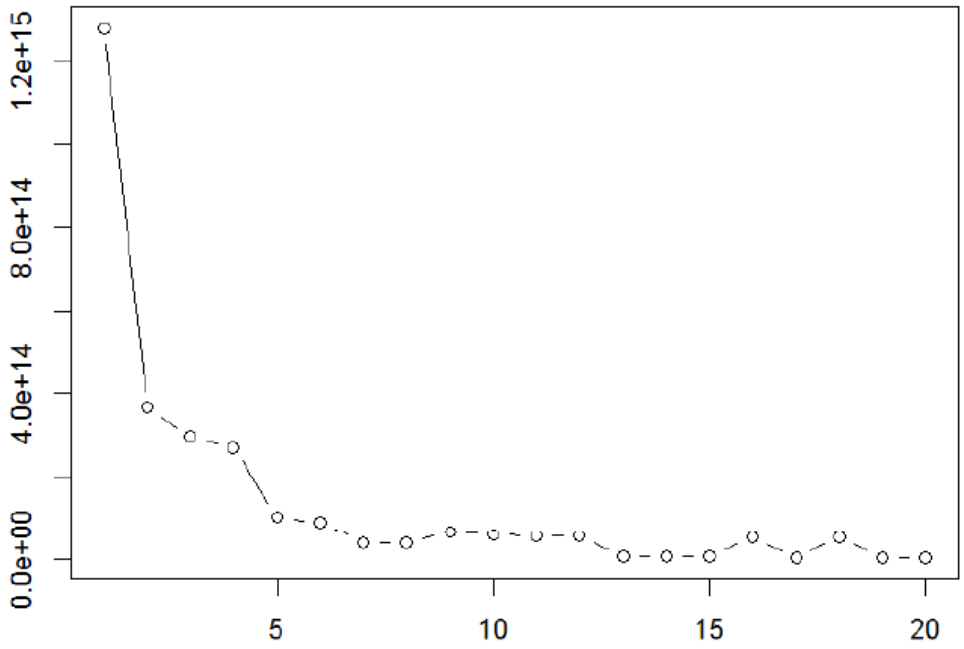

图 15： Ursnif、Gamarue以及干净文件集的误差平方和划分

在12个中心点运行的K均值聚类算法为我们提供了一个相对密集的群体分布。

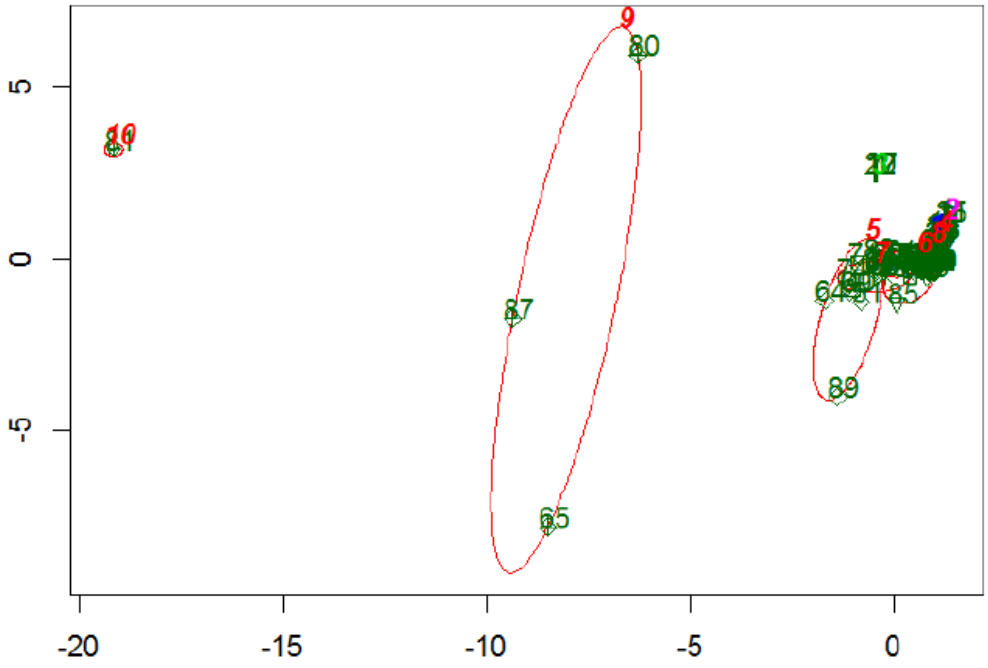

图 16 ：Ursnif、 Gamarue以及干净文件的混合聚类划分

从中可见，Gamarue家族与Ursnif家族隔离甚远。然而，干净文件集--虽然与Ursnif相分离--与Gamarue重叠较多。

表 2 : Gamarue、Ursnif以及干净文件聚类

	1	2	3	4	5	6	7	8	9	1	1	1
干净	9	0	0	2	1	1	0	1	0	0	4	8
Gamarue	0	0	0	0	0	0	0	2	0	1	0	2
Ursnif	2	4	4	0	0	0	1	0	4	0	0	0

结论

我们在这份研究报告中想要展示的是，利用可视化技术和在线免费工具，一种交互式的恶意软件分析方法能够为机械学习算法和自动文件处理的文件属性甄选提供大大的帮助。我们向大家展示了，只要充分考虑文件属性空间，K 均值无监督学习聚类算法就能够应用于文件分组。我们发现，甄选的属性足以将两个恶意软件家族分离成聚类，但当引入一个干净文件集时，属性就会减少。干净文件集与 Gamarue 恶意软件家族相互重叠，但却与 Ursnif 家族聚类分隔开来--这可能是因为 Ursnif 文件几何形状特殊并且在干净文件中不常见。它也表明，在对比 Gamarue 和 Ursnif 文件属性的平行坐标图时（参见图 5 和图 7），Gamarue 属性与选定文件集的耦合度不高，并且覆盖了大部分的数值空间，这种情况也出现在了干净文件中。这表明，文件几何学自身不足以产生精确的组别分类，还需要考虑和发掘其他属性集。这种属性可能是如下这些结果：静态和代码行为分析、分区分均信息量、导入和导出的 API 以及上述各项的组合。但是，这里所提供的方法可能对恶意软件家族的文件属性聚类的选择、测试及评估有所帮助。

延伸阅读

Microsoft PE and COFF Specification

Peering Inside the PE: A Tour of the Win32 Portable Executable File Format

An In-Depth Look into the Win32 Portable Executable File Format

欲了解更多信息，请访问 hp.com/go/hpsr。