“野外”计算机安全漏洞数据量化的初步分析

摘 要

一个由计算机、网络和软件组成的系统存在一定程度的漏洞，其面临黑客的犯罪风险。目前，大多数漏洞研究使用软件供应商的数据和国家漏洞数据库（NVD）。我们提出了另一条前进的道路，将我们的分析建立在操作信息安全社区的数据基础上，即“野外”的漏洞数据。本文提出了一种漏洞数据解析算法，并对漏洞来临和删除过程（又称漏洞出生-死亡过程）进行了单变量和多变量深入分析。我们发现，漏洞到达的最显著特征是对数正态分布，而漏洞删除的最显著特征是指数分布。这些分布可以作为未来贝叶斯分析的先验概率。我们还发现，超过22%的已删除漏洞数据的速率为零，并且到达漏洞数据总是大于零。最后，我们通过二变量散点图和统计观察来量化和可视化漏洞到达和删除之间的依赖性。

关键词：漏洞数据；操作漏洞管理；NESSUS扫描仪；探索性数据分析；重尾分布；单变量分析；双变量分析INTRODUCTION

1.导论

信息安全从业人员在修补网络漏洞方面面临着双重挑战。它们的补丁数量超出了它们的响应能力。网络渗透测试软件输出大量数据，包括常见漏洞枚举（CVE）、常见漏洞评分系统（CVSS）、简要描述和漏洞修补信息。事实上，美国政府和行业的最佳做法是将漏洞风险与CVSS基准分数相关联。然而，越来越多的证据表明，计算CVSS值的方法可能需要重新评估。

为了使漏洞缓解变得更具挑战性，安全工程师在严格的限制下工作。他们的时间、资源和人力有限。在理想情况下，他们会做出有针对性和精确的决定，首先对最关键的机器进行补丁。然而，外部因素，例如一台机器对另一台机器的价值，在当前的漏洞软件风险度量中被完全忽略。因此，系统管理员必须在内部启动项目，并将企业级网络划分为不同关键级别的区域。然后，他们必须将这些区域与漏洞输出进行匹配，以确定应首先解决哪些机器的向下选择过程，形势严峻，许多因素阻碍了管理IT环境中真正具有攻击性的漏洞。

本文对“野外”的漏洞数据进行了初步分析。我们的最终目标是促进解决方案，以解决漏洞管理和数据洞察方面的当前IT挑战。我们通过进行探索性数据分析和量化漏洞出生和死亡过程之间的关系来评估操作IT漏洞暴露的状态，从而实现这一点。在第1.1节中，我们回顾了我们的目标研究成果，并概述了论文的其余部分。

我们在研究中解决了上述问题的多个方面，即：

1. 开发数据解析算法，简化漏洞数据处理；
2. 统计特征化漏洞到达和删除过程；
3. 清楚地描述漏洞到达和删除之间存在哪些依赖关系；
4. 量化单变量和双变量关系，并解释其在“野外”总漏洞暴露方面的含义；
5. 为我们的研究确定对内部和外部有效性的威胁；
6. 总结和概述我们研究的未来工作。

我们认为，上述成果对于更深入地了解当前的操作漏洞管理状况和改进最新技术的机会领域至关重要。

在第2节中，我们对漏洞分析中的相关工作进行了全面的概述。此外，由于本研究侧重于评估使组织面临犯罪剥削风险的整体漏洞暴露，因此我们还为感兴趣的读者提供了不同风险模型和指标的概述。在第3、4和5节中，我们描述了数据、分析数据的方法和结果。在第6节和第7节中，我们将讨论未来工作的机会领域以及我们的结论。

2.相关工作

在本节中，我们概述了有关漏洞模型、风险评估和影响漏洞管理的框架的相关工作。

建模和统计理解漏洞可以为漏洞管理和缓解过程提供更深入的见解。最近，已经提出了两个评估漏洞暴露的指标，即：

1）活动漏洞中位数（MAV）；

2）无漏洞天数（VFD）。这些指标是根据漏洞生命线模型推导出来的，该模型基于测量何时向供应商报告漏洞，以及何时向同一供应商报告漏洞。UE补丁。我们的工作是不同的，因为我们专门研究操作IT数据。基于供应商何时发现新的漏洞的分析，当他们发布补丁时，这为建模漏洞提供了坚实的第一步；但是，它并不能完全捕获漏洞补丁“现实世界”部署的成功和失败。仅仅因为知道了一个漏洞，或者供应商提供了一个补丁，并不一定意味着补丁被有效地使用了。一种解决方案是将来自真实漏洞扫描和企业级渗透测试的现场数据结合起来，以获得更全面的理解。

基于通用漏洞评分系统（CVSS），对漏洞管理进行了研究。CVSS度量提供了分配漏洞严重性的基本分数。不同的级别由卡内基甜瓜大学的CERT、软件供应商和NIST等组织定义。漏洞通常由CVSS基准值进行分类，该基准值不包括时间或环境因素。最近的工作之一是将黑市利用数据等外部因素纳入CVSS基本得分。其他工作还对基于CVSS数据的18个安全评估指标进行了统计分析，并对34个成功攻击进行了时间折中。他们发现，仅使用CVSS数据进行安全建模并不能准确描述系统崩溃的时间。他们还发现，仅基于最严重的CVSS数据的安全模型不如考虑所有漏洞的安全模型可靠，无论其CVSS严重性如何。

相关工作的另一个领域来源于计算和网络风险模型。具体来说，Clark等人通过任务树将一个组织的目标和关系与网络主机联系起来，以模拟风险。开发了4个其他风险模型，以在考虑到漏洞之前考虑到需要保护的网络资产。5此外，对投资效率的风险进行定量评估NT策略（查询）为使用新兴技术进行网络安全风险评估提供了一种强大的跨学科方法。它提供了一种计算方法，用于导航软件和知识产权（IP）等示例中保护最具成本效益的策略。.

其他工作可以通过微软的威胁建模方法找到，这是一种在整个系统开发生命周期中建模网络威胁的方法。它根据Stride对威胁进行分类，Stride代表欺骗、篡改、拒绝、信息披露、拒绝服务、提升权限。该方法跟踪相关的攻击技术、目标和可以执行的补救措施。

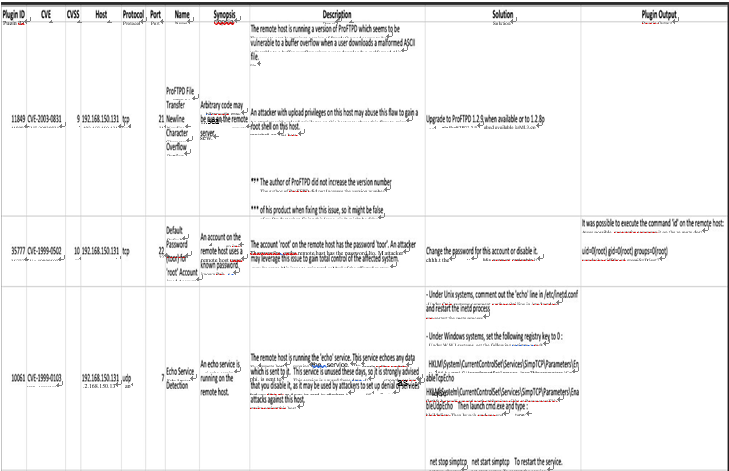
最后，其他三个评估风险的模型是塔拉、莫达和达卡提供了评估风险的方法。tara是威胁评估和修复分析的缩写。它是一种用于应对网络威胁的系统级方法，并使用评分模型和威胁矩阵等工具。它可以在已经部署的系统以及仍在采办生命周期中的系统上执行。它的优势在于，通过调整最适合系统的评估级别，它提供了一种灵活的方法。此外，只需使用给定系统所需的工具。MORDA是面向任务的风险和设计分析的缩写。它是系统级的风险评估框架。它利用了对手、用户和服务提供者的模型。目标是最大化用户价值，无论系统是否受到攻击。这些模型依赖于来自主题专家（SME）的信息，并建立在已知的对手信息和攻击模式的基础上。最后，DACCA是对抗网络攻击的决策分析的缩写。DACCA对攻击影响进行权衡，并对该影响作出可能的响应。它采用主题专家（SME）评估来告知攻击的严重性、攻击的可能性以及对手能力的复杂性。

另一个有趣的方法是由麻省理工学院林肯实验室生产的netspa。它是一个映射计算机网络的软件，最容易受到黑客攻击。它以网络配置信息为输入，输出最坏情况下的场景攻击图。最终状态需要确定为一个目标，以及哪些主机可以被信任，最后是最大递归深度优先搜索大小。

在本节中，我们概述了相关工作的许多不同方面，从漏洞模型到网络风险管理。在下一节中，我们将回顾我们的数据以及从原始数据到我们分析的最终数据所采用的“清理”过程。

3.数据

在本节中，我们将描述用于分析的数据，以及数据源和软件。在第4节中，我们概述了我们的方法和对数据进行预处理所遵循的步骤。

3.1数据描述

我们分析了Nessus漏洞扫描器的输出，以进行研究。第3.2节描述了Nessus的特定软件信息。我们研究了2013-2014两年期间的17组扫描数据。数据包含近3000个主机，11个不同的主机。图3.1显示了原始漏洞数据的示例。

图3.1：典型Nessus漏洞输出文件的示例。

Nessus漏洞输出通常具有以下11个属性:

1. 插件ID：一个软件代码，每个代码都有一个唯一的ID号，用来识别它正在扫描的属性和漏洞；
2. 常见漏洞枚举（CVE）代码：用唯一可识别号码标记最常见漏洞的系统；
3. 通用漏洞评分系统（CVSS）：如第2节所述，CVSS值提供了分配漏洞严重性的基本分数；
4. 风险等级：用一个词描述漏洞的严重性。软件供应商通常使用“关键”、“高”、“中”、“低”或“信息”等描述符。CVSS值通知大多数软件供应商的风险级别描述；
5. 主机名/IP地址：主机名是标识连接到网络的机器或设备的标签，例如www.dartmouth.edu。IP地址是唯一标识连接到网络的机器或设备的数字字符串，例如172.16.254.1。任何机器或设备都可以同时具有IP地址和主机名，或者仅具有IP地址；
6. 协议：指定主机、机器或设备与另一主机、机器或设备进行“对话”的通信方法和规则集。TCP和UDP是协议的例子；
7. 端口号：主机、机器或设备上通信端点的标识符，由逻辑结构确定；
8. 漏洞名称：漏洞的简短标题。通常少于5个单词，由软件供应商选择；
9. 漏洞概要：对漏洞的一句简短解释；
10. 漏洞描述：对漏洞的扩展和更全面的解释。可以提供当前和背景信息，长度可以是20-100个字。

3.2数据源和软件

生成我们数据的软件是Nessus漏洞扫描程序，版本5.0.1。Nessus由Tenable Security Solutions开发和维护。11漏洞扫描在达特茅斯计算服务数据中心运行。数据中心被称为组织的“大脑”，它集中存储、管理和传播对组织的计算机和网络操作至关重要的数据。它可以作为虚拟服务或物理服务存在。.

3.3最终数据集

数据经过某些关键处理阶段，从原始数据源获取最终数据。在本节中，我们简要概述了最终的数据集，在第4节中，我们更详细地描述了数据解析过程。

表1显示了数据预处理的最终输出。请注意，这些值表示每个类别在时间上的总体速率。它们不代表时间间隔之间的速率，例如从扫描到扫描。作为未来工作，第6节将讨论分析时间间隔之间漏洞数据的变化。

表1：最终漏洞数据示例。.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Host | Arrival Rate | Deletion Rate | Disappearance Rate | Reappearance Rate |
| A | 0.529411765 | 0.529411765 | 0.529411765 | 0 |
| B | 2.882352941 | 2.05882353 | 2.823529412 | 0.764705882 |
| C | 0.352941176 | 0.352941176 | 0.352941176 | 0 |
| D | 0.176470588 | 0 | 0 | 0 |
| E | 1.411764706 | 0.294117647 | 0.294117647 | 0 |

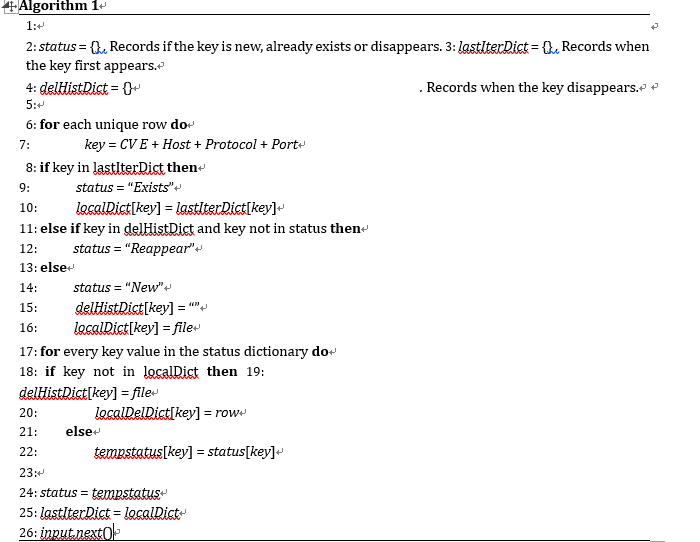
在下一节中，我们将概述数据解析算法和统计分析背后的方法。

4.方法论

在本节中，我们将回顾数据的分析方法。在本研究中，我们分析了漏洞到达过程和漏洞删除过程。首先，我们概述了数据预处理的步骤，以及如何从原始数据中获得可分析和研究的更易于管理的数据。最后，我们描述了我们对数据的假设。

4.1数据预处理

通常，分析的很大一部分都处于数据预处理阶段。根据数据，它可以吸收多达50%到80%的总间来成分析。在此，我们描述了用于数据的关键处理阶段。从图1提供的示例数据输出中我们知道，Nessus软件提供的输出属性比我们分析所需的更多。我们的数据预处理最关键的部分是准确地存储和识别独特的漏洞。为了引入一些符号，假设我们将所有漏洞的集合表示为v，然后对于每个唯一的漏洞i，我们将关联的漏洞集合表示为v i。然后我们将唯一的漏洞（v i）定义为四个属性集，即：1.）CVE代码；2.）主机名；3.）端口号；和4.）协议。这四个属性是专门选择的，因为一个漏洞的唯一性不能用少于这四个组合属性的任何东西来确定。例如，并非Nessus输出中的所有漏洞都具有CVE代码。因此，仅仅通过CVE代码识别一个独特的漏洞是不够的。此外，每个主机名都可以显示在具有许多不同端口号和/或协议的扫描中。因此，识别唯一漏洞（vi）的唯一方法是根据CVE代码、主机名、端口号和协议的组合键对数据进行解析。结果值是所有可能的唯一漏洞的集合。然后，我们将这组独特的漏洞进行比较，从一次扫描到另一次扫描，以将其状态定义为新的、现有的、消失的或重新出现的。最后的输出是一组文件，用上述标签表示从一次扫描到下一次扫描的差异。在我们的研究中，我们需要速率，因此进一步将数据转换为计数，然后将总计数除以扫描次数。我们在算法1中总结了数据解析器的源代码，它输出新的、消失的或重新出现的唯一漏洞集。如图4.1所示：

图4.1 算法1

4.2假设和关键关系

在本节中，我们将概述数据中的初步假设和关键关系。每个唯一的漏洞都由CVE代码、主机、协议和端口的唯一密钥定义。正如从算法1中可以理解的那样，对数据进行分析的方式使漏洞定义为“新”、“已存在”或“消失”。如下文所述，有一个非常关键的与删除的漏洞相比，消失的漏洞有明显的区别。

从扫描到扫描，漏洞存在于四种可能状态之一，即：

1. 新：如果漏洞不是新的，那么从一次扫描到下一次扫描都会被认为是新的。在以前的扫描中观察到；
2. 消失：漏洞可以消失，原因有很多，包括服务器为了维护和维护，或甚至在服务器主机命名更改的情况下重新组织网络，已解除授权或处于脱机状态；
3. 重新出现：如果漏洞的唯一密钥在扫描中至少出现一次新的漏洞，那么，如果该唯一密钥在任何时间点上连续消失扫描，然后重新出现；
4. 已删除：漏洞修补后可以视为已删除。我们收集了一套通过将漏洞消失率与漏洞重新出现率（如删除率）之差来计算已删除漏洞的总体速率，消失率-再现率定义漏洞比率之间的这种关系删除、消失和再现需要一些简化和假设。我们相信这些简化是建模数据的最佳起点。作为我们模型成熟，我们可以探索合并其他不确定性层的方法，在第6节中进一步说明。

4.3单变量分析

在我们的单变量分析中，我们的目标是通过分布模型描述到达和删除过程。我们评估和比较模型的方法如下:

1. 经验密度和理论密度：我们根据我们认为最适合的理论分布绘制经验数据的直方图。这通常是获得数据直觉的良好起点。Q-Q图（下面描述）提供了对分布是否适合的更深入的了解；
2. 四分位数图（Q-Q图）：Q-Q图比较柱状图的形状和理论密度。如果比较的两个分布相似，则q-q图中的点将近似位于y=x线上。如果分布是线性相关的，则q-q图中的点将近似位于一条线上，但不一定位于y=x线上；
3. 经验和理论CDF：我们比较了经验和理论数据的CDF。我们的目标是寻找一个整体的良好拟合，并尽量减少经验分布和理论分布之间的距离；
4. 概率概率图（P-P图）：P-P图将变量的经验累积分布函数（ecdf）与指定的理论累积分布函数进行比较。这个ECDF定义为从数据中得出的经验累积分布函数。与Q-Q图和概率图一样，P-P图也可以用来确定理论分布对数据分布的建模程度。如果理论CDF在所有方面（包括位置和比例）都合理地模拟了ECDF，则P-P图上的点模式通过原点呈线性，并且具有单位坡度。.

4.4二元分析

最终，我们的目标是发现漏洞出生和死亡过程之间的关系，即到达率和删除率之间的关系，并探索如何将其推广到“野外”的其他漏洞数据分析中。

我们分析了双变量散点图的趋势，总结了基础统计数据，并评估了其对未来模型的有用性，以及对分析师和安全工程师的实际应用。

5. 结论

在本节中，我们首先概述了所有四个数据子集(即新的、删除的、重新出现的和消失的)的一些基本统计特征和分析。然后，我们深入分析 一组新的漏洞出生率和一组已删除的漏洞的死亡率。最后，我们评估了双变量生与死过程之间的依赖关系。

5.1汇总统计分析

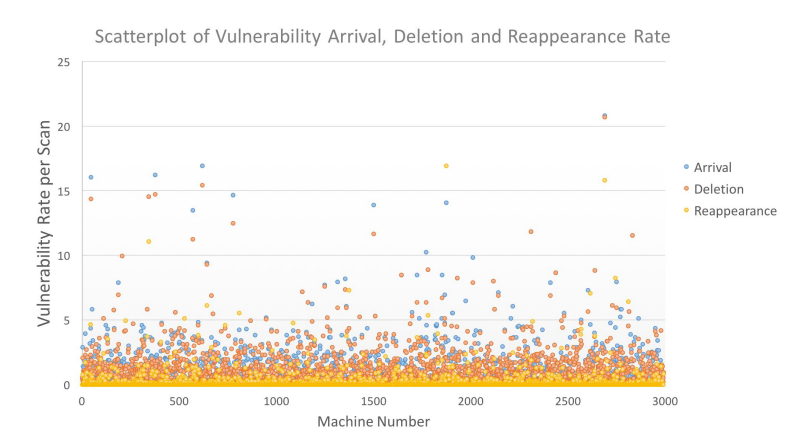
回顾如图1所示的最终数据集，我们提供了一个漏洞到达、删除和重新出现的散点图。我们发现87%的漏洞由于不同的原因消失，包括删除、服务器脱机或退役或扫描策略更改。我们还发现，73%的漏洞通过修补被删除，14%的漏洞在同一台计算机上的稍后时间点重新出现。如图5.1所示：

图5.1 弱点到达、删除和再现速率的散点图

在下一节中，我们将更深入地研究单变量分布和特征。

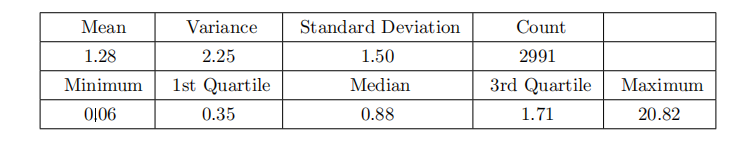
5.2单变量分布和特征

在本节中，我们将概述漏洞到达和删除率分布的近似值，以及统计特征。我们提供了关于漏洞管理的分布适合意味着什么的实际解释。

5.2.1漏洞到达

我们研究了近3000台机器在两年内从17份扫描数据报告中的漏洞到达率。我们将到达率的单变量分布表示为a，这样对于每个主机的单个速率i，我们将每个主机的相关到达率表示为ai。

我们发现每台主机在至少一个扫描报告上至少有一个漏洞计数；因此，每台机器的最终漏洞到达率大于零。这意味着没有一个宿主能免于妥协的风险。我们还发现，漏洞到达数据显示了极端右歪斜的特征。这种偏态分布表明低出生率的高计数和极高出生率的低计数。

表2：漏洞到达率分布的统计摘要

有趣的是，超过一半的机器（54.90%）的漏洞出生率小于或等于1，几乎所有机器（97.89%）都小于或等于5。最大值为20.82，这告诉我们刚好超过2%的数据（具体来说，2.11%）说明了（5，20.82）的重尾值范围。进一步的统计调查总结见表2。

我们发现漏洞到达率满足对数正态分布。图3显示了四个有助于评估分布适合性的图。通过比较漏洞到达的经验密度和理论密度，验证了对数正态拟合的可靠性。为了更深入的了解，我们发现Q-Q图中的数据非常接近直线，除了几个最右边的圆。当我们用理论CDF来观察经验CDF时，我们看到了一个整体的很好的拟合。最后，经验数据点与P-P图中的直线吻合较好。数据的前半部分有一条温和的曲线，表明经验概率和理论概率之间存在轻微的差异，但总的来说，这是一个很好的拟合。

当我们从对数正态模型的其他领域观察到实例时，对数正态分布是直观的，例如流行病学研究中的病毒生长率、13和机械结构中的疲劳应力失效。14对数正态模型通常最能代表自然现象，其值均大于零。

5.2.1漏洞删除

使用与第5.2.1节中使用的方法相似的方法，我们研究了两年内17份扫描数据报告中几乎3,000台机器的漏洞删除率。我们表示Univari。 将缺失分布为D，因此对于每个主机的每一个速率I，我们将每个主机的相关联的删除率表示为di。

我们发现漏洞删除率集合与到达率集合不同，超过22%的机器在两年的时间内完全没有被删除的漏洞。TH 表示几乎三分之一的机器存在漏洞，而这些漏洞没有以任何方式得到解决。与漏洞到达相似，漏洞删除率数据极为右偏。在删除漏洞的情况下，偏度的一个原因可能来自所谓的“启动效应”，如我们在可靠性分析中可能会发现的。由于许多不同的PO 可能的原因是，在漏洞补救过程开始时可能会出现大量的“失败”。

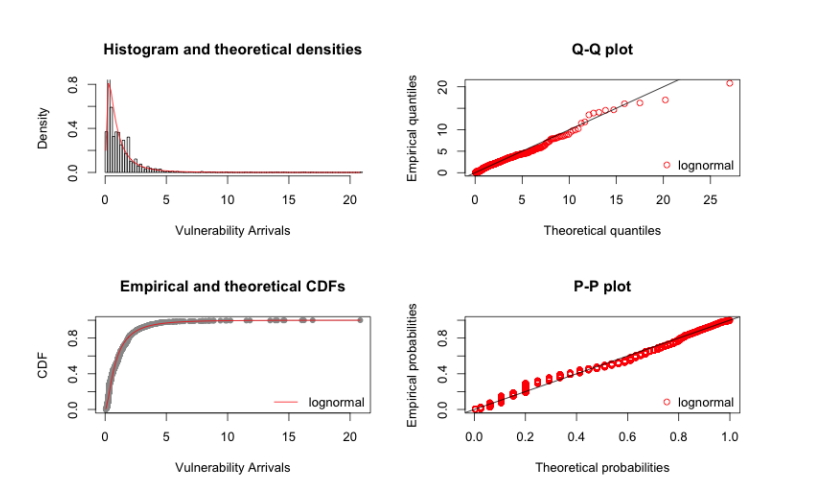
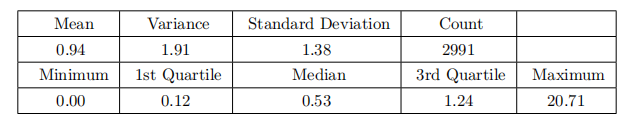


图3：日志正态分布为漏洞到达过程提供了一个可能的模型。

毫不奇怪，相当多的机器(67.20%)的漏洞死亡率低于或等于一台，几乎所有机器(98.36%)都小于或等于五台。W 最大值为20.71，这告诉我们，只有不到2%的数据(特别是1.64%)占重尾值范围(5，20.71)。表3总结了进一步的统计探索。

表3：删除漏洞率分布的汇总统计数据

我们还发现，被删除的漏洞集可以用指数分布来建模。图4显示了四幅图，以帮助评估最佳发行版的适配度。我们发现埃佩里卡 L和理论密度几乎完全吻合。q-q图在线的左上方显示出轻微的向上弯曲，这表明底层的proc可能有更多的活动。 Ess比我们目前能够捕获的†更多。P-P图在左下角区域也显示了一些轻微的偏差，但是总的来说，我们发现指数分布的概率值是ar的。 E非常类似于从前的经验。

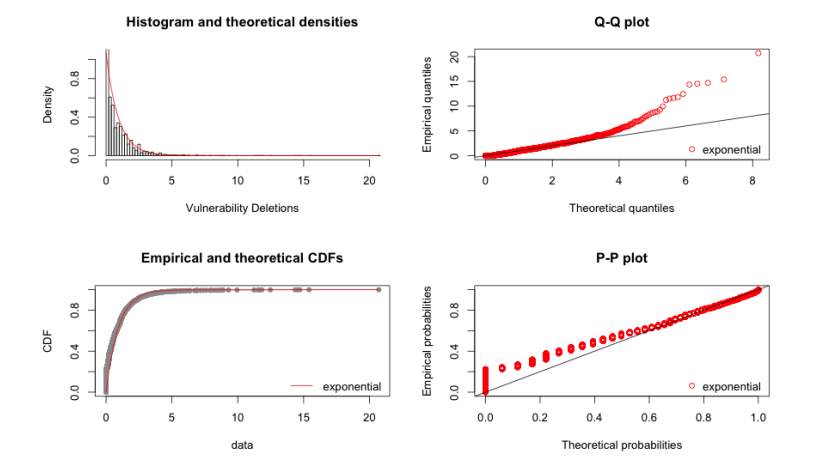
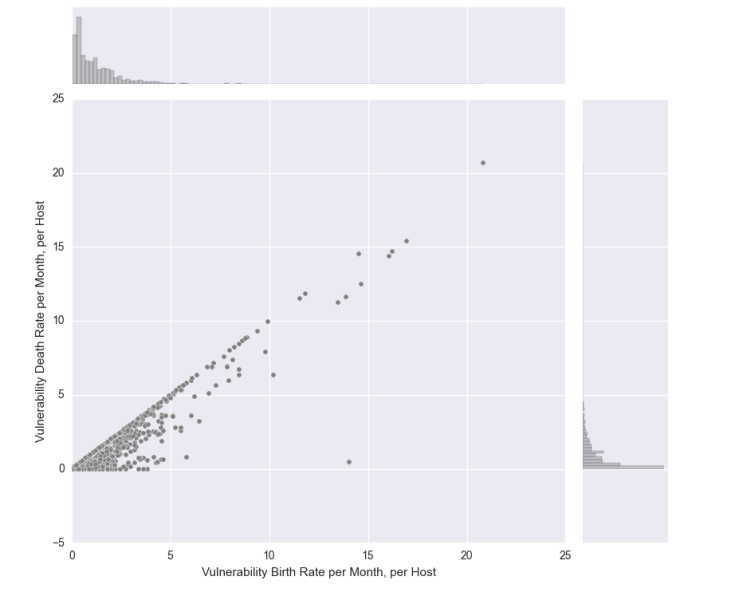


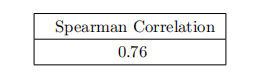
图4：指数分布为漏洞删除过程提供了一个可能的模型

我们怀疑指数分布最适合最好的原因主要是由于数据中有大量的零。正如我们在第6节中所概述的，未来的工作是探索多久的秃鹫 可测性随着时间的推移而持续，它们的到达率和删除率在扫描之间是如何变化的。根据结果，漏洞生命线可以作为可靠性理论检验来研究。

5.3二元关系

在本节中，我们旨在通过双变量散射图和相关性来指定关系。统计的。我们知道，漏洞删除在某种程度上依赖于漏洞到达。在建模这种依赖关系时，一个自然的选择是出生率(对于易受伤害的到达)和死亡率(用于漏洞删除)。请注意，我们使用的这种语言的互换性贯穿本节的分析。

图5：漏洞、出生和死亡率的双变量散点图。

表4：出生率与死亡率二元关系的Spearman相关统计。

在图5中，我们看到漏洞出生率与死亡率之间存在着很强的正相关关系。表4显示了进一步的相关证据，其中Spearman相关统计量i 很好，离一个很近。我们包括双变量图的X轴和Y轴的外边缘上的边缘（单变量）分布，以辅助进一步分析。

我们发现，当漏洞死亡率与漏洞出生率相等时，最好的案例场景是。考虑了漏洞出生率和死亡率符合性的点 平衡点。请注意，有些数据点偏离x=y线。这些数据表明机器具有比他们的死亡率高得多的漏洞出生率。

这个二元图的有用之处在于它不仅为模型和联合概率分布提供了基础，而且还可以为分析人员和securi提供即时反馈。 TY工程师关于其漏洞管理的当前状态。X-Y线上的数据群集越多，我们就知道在24个月的Perio上正在修补更积极的机器。 D.xy线下方的数据点群集越多，漏洞被修补得越小，并且它们就越需要维修。综上所述，双变量散点图为。 有助于深入了解漏洞管理的历史。它们还可以根据过去的网络行为对未来的漏洞指标进行预测分析。

5.4对内部和外部有效性的挑战

在本节中，我们对内部和外部有效性提出的一些已确定的挑战进行了审查。认识和减轻这些挑战对于产生和处理数据是至关重要的。 取得了可靠的结果和彻底的科学分析。尽管如此，我们认识到，在我们的研究中，内部和外部的有效性都面临着一些挑战。例子包括 t不限于Nessus软件中的假阳性率或假阴性率。第5.4.1节和第5.4.2节全面概述了这些挑战。

5.4.1内部有效性

内部效度是指实验进行得有多好。支持良好内部有效性的因素是尽可能少的混淆变量。作为独立变量的数目 ES同时作用也增加了内部有效性的挑战。

本研究的内部有效性挑战包括：

删除率假设我们将删除率定义为消失漏洞集和重新出现漏洞率集合之间的差异。可能还有别的 在消失的漏洞和再次出现的漏洞之间的关系中混淆变量。尽管如此，我们也不想把数据过大，相信我们的准确性是相当高的。 h计算一组漏洞删除率。

扫描策略更改 企业级网络经历周期性更改，例如永久添加和删除服务器。发生这种情况时，需要添加或删除主机名 漏洞扫描列表。在这两年的漏洞数据中，有三次扫描策略的变化。

数据空白 24个月中有6个缺少扫描结果。我们将这些数据的结果与缺少几个月的不同级别的假设进行了比较，发现没有最优的策略 为这个特定的数据集连接缺少的几个月。为了避免在数据中引入伪影或无意中歪曲结果，我们决定将这些数据作为“扫描”鼠。

Nessus扫描 本身的软件缺陷问题可能导致数据不完美。目前有一批旨在解决漏洞软件错误和错误的研究。我们引导好奇的读者参考3，15和16。

5.4.2外部有效性

外部有效性是指科学成果在类似情况下的推广程度。我们知道，一项科学研究越少推广到其他研究，那么它所面临的外在效度挑战就越大。

本研究的外部有效性挑战包括：

数据 可能对外部有效性的最大挑战是数据。我们正在处理一个来源的业务数据。从多个来源获得数据是理想的。因为这项工作是公共关系的一部分 通过排除研究，我们将在第6节中扩大对更多数据源的分析。作为最大限度地减少这一挑战的努力，这也是我们对所发现的假设和概括做出的努力的一部分。 第4.2节，以避免对数据进行过度拟合。

6.未来工作

本文主要研究了大型漏洞数据解析过程的初步工作，探讨了漏洞到达和删除过程的统计特征。在一个操作设置中，我们的研究是独一无二的，因为它是第一次从“野生”中获取漏洞数据，随之而来的是大量的安全研究问题，这些问题可以进一步帮助漏洞缓解和管理进程。具体而言，我们提出以下研究问题：我们能量化运营漏洞生命线吗？净预计成本是多少 基于他们目前的漏洞修补过程？我们如何衡量或建模风险，以便更有效地进行漏洞分类？

为了解决这些问题，我们建议今后的工作是我们的结果：

收集更多的信息技术业务漏洞扫描数据，以更好地估计“野生”地区漏洞修补的现状；

将属性时间(T)包含到我们的模型中，方法是测量漏洞持续存在的时间长度，以及时间间隔之间的生灭过程的变化；

将随机变量-时间(T)、漏洞到达(A)和漏洞删除(D)-纳入改进漏洞分类的风险模型；

根据如何实时管理漏洞来估算组织的预期成本净额。

7.结束语

最后，本文概述了研究来自运营IT环境的漏洞数据的必要性。给出了一种用于在LACE中进行争论的数据解析算法和难以控制的漏洞数据集。然后，我们提供了一个深入的单变量和二元分析，提供了洞察漏洞到达和删除过程之间的关系。 我们用来描述单变量的分布可以作为未来研究的先验分布，例如关于漏洞管理和缓解的贝叶斯预测。未来工作包括：

1.)推广和自动化我们的漏洞分析算法，以便在任何环境中使用，从学术到操作；

2)通过MEA量化随机变量时间(T) 从它们在时间上的持久性以及时间间隔之间的变化来分析漏洞；

3)基于三个随机变量到达(A)，风险计算 (D)和时间(T)。我们的研究结果也自然有助于为网络保险等领域的无数风险评估和精算研究提供信息。

谢 辞

这项工作得到AROMURI授予W911NF-13-1-0421的支持。作者们感谢国防部为学生提供资助的智能奖学金、BenPriest提供的反馈，以及提供漏洞数据的达特茅斯计算服务。