信息系统中的漏洞查找与修复

脆弱性评估

和

规避方法

Philip S. Antón

Robert H. Anderson

Richard Mesic

Michael Scheiern

为 (美国国防部)高级防御研究计划局 制备

RAND

国家防御研究院

# 前言

在引导评估者通过对关键风险点的测定和探究合适的减灾方案方面，信息系统的风险测评方法已经成为最薄弱的一块能力短板，而脆弱性评估和加固（Vulnerability Assessment & Mitigation，VAM）理论则填补了这一空白。VAM理论继承和发扬了RAND公司早期用来加固系统的“最小必要信息基础设施”（Minimum Essential Information Infrastructure，MEII）理论，它使用的分类方法相对全面地将导致脆弱性的特性按照自顶向下的方式进行了分类，并将这些脆弱性特性归纳到相对完整的减灾方法列表当中。从横向上看，减灾技术不仅包含人们在正常情况下能够认识到（但事先可能还没有被发现）的一般的、直接的方法，也囊括了那些可以间接降低风险的方法和途径。这会帮助风险评估人员吧那些未知的脆弱性也能考虑周延，同时也丰富了关于那些显在或潜在安全隐患的清单来抵御意外冲击。

这份报告可能会引起一些个人或（独立的或者隶属某个研究单位的）团体组织，这些组织通常涉及评定和规避那些对某对象的功能有决定性影响的风险或脆弱性——这其中也包括发现还未被遇到或利用的弱点。另外，一些关注其他行为（例如利用或攻击漏洞）的人们也会对这份产生兴趣。

这份报告使用了MS EXCEL 2000原型电子表格来编辑某些内容。读者可以在[www.rand.org/publications/MR/MR1601/](http://www.rand.org/publications/MR/MR1601/)下载到这些表格。

本报告的作者还进行了一项应用VAM理论进行的军事战术信息系统的问题的研究，研究成果没有发表。这些理论研究的成果只能对获得政府授权的个人提供，相应地，可以联系笔者Philip Antón([anton@rand.org](mailto:anton@rand.org))或者Robert Anderson ([anderson@rand.org](mailto:anderson@rand.org))。

这项研究得到了隶属于(美国国防部)高级防御研究计划局（Defense Advanced Research Projects Agency）的信息技术办公室（Information Technology Office ，ITO）的赞助，同时由RAND国家防御研究院（National Defense Research Institute）的收购及科技政策研究中心（Acquisition and Technology Policy Center）领导。后者作为一个联邦政府投资研究和发展中心(Federally Funded Research and Development Centers，FFRDC)，由国防部长办公室、联合参谋部和一些统一指挥部、国防机构赞助。

目录

[前言 i](#_Toc364268819)

[序言 ix](#_Toc364268820)

[结论 xv](#_Toc364268821)

[致谢 xvi](#_Toc364268822)

[缩写 xvii](#_Toc364268823)

[第1章 介绍 1](#_Toc364268824)

[VAM理论的使用者 1](#_Toc364268825)

[早期成果 2](#_Toc364268826)

[报告的结构 2](#_Toc364268827)

[第2章 概念和定义 4](#_Toc364268828)

[安全 4](#_Toc364268829)

[信息系统 4](#_Toc364268830)

[系统对象类型 4](#_Toc364268831)

[对于“对象”概念的使用 5](#_Toc364268832)

[哪些属性是导致脆弱性的来源 5](#_Toc364268833)

[安全技术 6](#_Toc364268834)

[第3章 VAM方法论和其他风险评估中的DoD实践 7](#_Toc364268835)

[VAM方法概述 7](#_Toc364268836)

[第1步 识别基本的信息功能 7](#_Toc364268837)

[第2步 识别基本的信息系统 9](#_Toc364268838)

[第3步 识别系统脆弱性 10](#_Toc364268839)

[第4步 从VAMM方法提供的候选条目中寻找合适的安全技术。 12](#_Toc364268840)

[第5步 选择和应用安全技术 13](#_Toc364268841)

[第6步 威胁下的健壮性测试 14](#_Toc364268842)

[其他美国国防部的脆弱性评估方法 15](#_Toc364268843)

[OCTAVE（关键操作风险、资产和脆弱性评估） 15](#_Toc364268844)

[ISO/IEC 15408：通用标准 16](#_Toc364268845)

[ISO/IEC 17799：信息安全管理的实践代码 17](#_Toc364268846)

[操作安全 18](#_Toc364268847)

[操作风险管理 18](#_Toc364268848)

[综合脆弱性评估 19](#_Toc364268849)

[VAM理论技术满足了其他方法中的关键需求 20](#_Toc364268850)

[第4章 系统对象的脆弱属性 20](#_Toc364268851)

[脆弱属性分类 21](#_Toc364268852)

[一个脆弱性用例和实例 21](#_Toc364268853)

[内部威胁 21](#_Toc364268854)

[无法抵御分布式拒绝服务攻击（DDoS） 22](#_Toc364268855)

[IP欺骗 22](#_Toc364268856)

[无法探测到IP网络改变，可能使IP掩藏 25](#_Toc364268857)

[集中网络操作中心 25](#_Toc364268858)

[常规的商业软件和硬件是已知的或者可预知的 25](#_Toc364268859)

[标准化软件 25](#_Toc364268860)

[路由和桌面应用软件漏洞 26](#_Toc364268861)

[电子环境容错 26](#_Toc364268862)

[对于脆弱属性的描述 26](#_Toc364268863)

[设计和结构属性 26](#_Toc364268864)

[行为属性。 28](#_Toc364268865)

[常规属性 29](#_Toc364268866)

[脆弱性如何与通常的威胁结合 30](#_Toc364268867)

[第五章 直接和间接的安全技术 34](#_Toc364268868)

[安全技术分类和举例 34](#_Toc364268869)

[韧性和强度 34](#_Toc364268870)

[ISR（情报获取、监视和侦查）和自校正 38](#_Toc364268871)

[反侦查，屏蔽ISR、目标搜索 38](#_Toc364268872)

[威慑和惩罚。 39](#_Toc364268873)

[安全技术如何和常规的安全方法组合在一起 39](#_Toc364268874)

[第六章 脆弱性的通用安全技术选择 44](#_Toc364268875)

[脆弱性到安全技术的映射 44](#_Toc364268876)

[映射到脆弱性的安全技术 44](#_Toc364268877)

[会引发脆弱性的安全技术 46](#_Toc364268878)

[脆弱属性可以多次推动安全技术 47](#_Toc364268879)

[冲击平衡 47](#_Toc364268880)

[设计和用法的考量 48](#_Toc364268881)

[提炼安全建议 48](#_Toc364268882)

[评估者的工作角色 48](#_Toc364268883)

[攻击模块 50](#_Toc364268884)

[通过评估者工作角色的攻击阶段的关联 51](#_Toc364268885)

[使用VAM产生的安全选择举例 54](#_Toc364268886)

[内部威胁 54](#_Toc364268887)

[无力控制DDoS攻击 55](#_Toc364268888)

[IP欺骗 56](#_Toc364268889)

[不能发现IP网的变化，可能发生IP掩饰 56](#_Toc364268890)

[集中的网络操作中心 57](#_Toc364268891)

[众所周知的一般商用软硬件 58](#_Toc364268892)

[路由器或桌面应用软件的弱点 59](#_Toc364268893)

[电子环境容错 60](#_Toc364268894)

[第七章 自动化执行方法：电子表格工具 61](#_Toc364268895)

[手动执行的原始步骤 61](#_Toc364268896)

[表格指引和记录的脆弱性 61](#_Toc364268897)

[风险评估和减灾选项表格 62](#_Toc364268898)

[说明用户类型和脆弱性并进行分析 62](#_Toc364268899)

[为每一个攻击模块评估风险 65](#_Toc364268900)

[考量和选择规避措施 67](#_Toc364268901)

[评定成本和已被规避的风险 68](#_Toc364268902)

[第八章 下一步以及相关讨论 70](#_Toc364268903)

[未来的挑战与机遇 70](#_Toc364268904)

[引导关键系统和功能的评价 70](#_Toc364268905)

[额外的引导和自动操作：拓展表和基于Web的工具 70](#_Toc364268906)

[安全选项排序 70](#_Toc364268907)

[威胁和风险的量化评估和规避 71](#_Toc364268908)

[将VAM功能整合进其他评估方法中 71](#_Toc364268909)

[使用VAM指导信息攻击 71](#_Toc364268910)

[超越信息系统应用VAM理论 71](#_Toc364268911)

[什么脆弱性会失败或称为下一个被攻击的对象？ 72](#_Toc364268912)

[可用性问题 72](#_Toc364268913)

[为什么执行安全评估？ 72](#_Toc364268914)

[第九章 概要和总结 74](#_Toc364268915)

[附录 脆弱性到规避方法的映射值 75](#_Toc364268916)

[脆弱性映射并积极影响的规避技术 76](#_Toc364268917)

[表A.1 单一性映射的规避技术 76](#_Toc364268918)

[表A.2 唯一性映射的规避技术 77](#_Toc364268919)

[表A.3 集中性映射以及被其促进的规避技术 78](#_Toc364268920)

[表A.4 一致性映射以及被其促进的规避技术 79](#_Toc364268921)

[表A.5 分散性映射以及被其促进的规避技术 80](#_Toc364268922)

[表A.6 逻辑/现实错误、不可靠性映射的规避技术 81](#_Toc364268923)

[表A.7 设计敏感性、薄弱/限制/有限性映射以及被其促进的规避技术 82](#_Toc364268924)

[表A.8 不可恢复性映射的规避技术 84](#_Toc364268925)

[表A.9 行为敏感性/脆弱性映射的规避技术 85](#_Toc364268926)

[表A.10 恶意行为映射的规避技术 87](#_Toc364268927)

[表A.11 强度映射的规避技术 88](#_Toc364268928)

[表A.12 可塑性映射的规避技术 89](#_Toc364268929)

[表A.13 欺骗性映射的规避技术 90](#_Toc364268930)

[表A.14 自满情绪映射的规避技术 91](#_Toc364268931)

[表A.15 易腐蚀性/可控性映射的规避技术 92](#_Toc364268932)

[表A.16 可访问/可侦测/可识别/透明/可截取映射的规避技术 93](#_Toc364268933)

[表A.17 难以控制或管理映射的规避技术 94](#_Toc364268934)

[表A.18 不自知和不可预测性映射的规避技术 95](#_Toc364268935)

[表A.19 可预知性映射以及被其促进的规避技术 96](#_Toc364268936)

[安全技术能引发的脆弱性 97](#_Toc364268937)

[表A.20 异构性可以引起的脆弱性 97](#_Toc364268938)

[表A.21 冗余可以引起的脆弱性 97](#_Toc364268939)

[表A.22 集中可以引起的脆弱性 98](#_Toc364268940)

[表A.23 分散可以引起的脆弱性 98](#_Toc364268941)

[表A.24 验证确认、软硬件工程、评估和测试可以引起的脆弱性 99](#_Toc364268942)

[表A.25 暴露度、访问和输出的控制可以引起的脆弱性 99](#_Toc364268943)

[表A.26 可信学习和执行系统可以引起的脆弱性 100](#_Toc364268944)

[表A.27 抗抵赖可以引起的脆弱性 100](#_Toc364268945)

[表A.28 加固可以引起的脆弱性 100](#_Toc364268946)

[表A.29 错误、不确定性、有效性和质量保证、软着陆可以引起的脆弱性 100](#_Toc364268947)

[表A.30 静态资源分配可以引起的脆弱性 101](#_Toc364268948)

[表A.31 动态资源分配可以引起的脆弱性 102](#_Toc364268949)

[表A.32 一般管理可以引起的脆弱性 102](#_Toc364268950)

[表A.33 应急响应体系与计划可以引起的脆弱性 103](#_Toc364268951)

[表A.34 紧急重建和恢复可以引起的脆弱性 103](#_Toc364268952)

[表A.35 适应和学习可以引起的脆弱性 103](#_Toc364268953)

[表A.36 免疫式防御系统可以引起的脆弱性 104](#_Toc364268954)

[表A.37 防疫可以引起的脆弱性 105](#_Toc364268955)

[表A.38 情报活动可以引起的脆弱性 105](#_Toc364268956)

[表A.39 自校正、监视、评估可以引起的脆弱性 105](#_Toc364268957)

[表A.40 诱骗ISR可以引起的脆弱性 105](#_Toc364268958)

[表A.41 检测攻击、损伤评估、敌我识别可以引起的脆弱性 105](#_Toc364268959)

[表A.42 常规反侦察可以引起的脆弱性 106](#_Toc364268960)

[表A.43 对敌的不可预知可以引起的脆弱性 106](#_Toc364268961)

[表A.44 反侦察欺骗可以引起的脆弱性 107](#_Toc364268962)

[表A.45 威慑可以引起的脆弱性 107](#_Toc364268963)

[表A.46 刑罚与司法保证可以引起的脆弱性 107](#_Toc364268964)

[表A.47 司法执行和民事诉讼可以引起的脆弱性 107](#_Toc364268965)

[参考文献 108](#_Toc364268966)

# 序言

随着信息系统在一些组织团体和社会机构等部门的业务功能方面发挥的作用越来越大，系统的安全和可靠性的问题也变得越来越值得关注。信息交互、资源分享、内外勾结的行为以及设计的缺陷和限制等诸如此类的可能引发风险的问题，都会给系统的安全和操作带来危险。不幸的是，理解这些地方在信息系统的可靠性、系统风险以及如何降低风险，已经成为一个严峻的挑战——尤其是在对那些曾被利用的漏洞不是了解甚至一无所知的时候。

RAND开发和完善了一套理论用来让分析师理解这些关系的理论。它可以拘谨识别系统的风险点，并从技术角度提出合理的加固建议。这就是建立在早期由Anderson等人所做工作基础上的“脆弱性评估和加固方法论/理论”，它填补了一个急需被填补的空白。它以横跨信息系统的所有方面（不仅包括计算机的，也包括物理的、人与社会的、基础设施方面的对象），引导评估者做出全面的评价，并把脆弱性映射到可以定位他们的特别的安全技术上。

VAM理论采用自顶向下的线路和查询方法，解释那些已知的、已被利用或暴露在外的脆弱性，同时也可以发现那种确实存在，但在操作工程中没有遭遇过活被利用的脆弱性。所以，这套理论在规避当前和过去的威胁和弱点的时候，也可以协防抵御来自未来的威胁和系统崩溃。另一方面，狡猾的对手同样也在寻找新的途径去攻击那些尚未被保护起来的资源，也就是信息系统的软肋。因此，这套理论作为i恶意逃处理和平衡当下及未来的威胁方面具有一定的价值。同样，信息系统的复杂性以及他们在结构体系中的功能集成越来越密集，也要求进行额外的考量以确认设计和结构中的缺陷已经被过滤掉了。

**确认组织结构要害功能的安全需求**

**VAM**理论评价方法的六个步骤：

1. 识别组织结构体系中那些基本的信息功能
2. 识别实现这些功能的基本信息资源
3. 识别这些系统中的脆弱性
4. 寻找合适的抵御脆弱性的安全技术
5. 综合考虑现有条件的限制，以及成本和收益的因素，选择和使用最佳的解决办法
6. 测试其在风险下的强度以及现实可行性

有必要时重复第3至6步。

这套方法的指导思想，是一种反向的从要害系统到重要的结构功能的联系，以及对每一个特殊环境下的安全技术的合理性的评价。这种思路不仅引导评估人员筛选极多可能的技术，也会为资源提供其所需的优化、评判、管理（过程）的固化，从何让他人知道需要做些什么以及这样做的原因。

**识别已知的和新的脆弱性**

脆弱点来源于对象的基本属性。VAM理论运用这一事实提出围绕所有对象的相对全面的属性分类方法，并通过这种基于使用属性列表（它运用于物理的、计算机的、人与社会的以及基础设施方面的对象）的分类方法引导评估者。这样的做法避免了评估者仅仅是简单的罗列标准、已知的脆弱性（自底向上的、从历史途径研究得出的），将研究的范围拓展到一般脆弱性识别工作的外围。例如，风险上升不仅源于接入点（例如防火墙的端口），也可源自行为的特性，比如对他人的轻信和一成不变的系统。这些特征都可能会从系统的某个组成部分中表现出来：网络、物理、人与社会、基础设施。

**落实识别和选择减灾办法**

VAM理论找到相对完整的信息安全类别的分类方法以预防、发现和避免信息系统中的危害和弱点。这些技术可以按照以下类型划分：提高系统韧性和强度的技术（加固型技术，译者注，下同。），提高情报获取、监视和侦查（Intelligence, surveillance, and reconnaissance,ISR）能力和自校正能力的技术（感知型技术），反侦查与屏蔽ISR、目标搜索类技术（隐藏型技术），威胁与惩罚类技术（反制型技术）。



**图S1 降低脆弱性影响的技术分类**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **脆弱对象** | | | |
|  |  | 物理 | 计算机 | 人与社会 | 基础设施 |
|  | 属性 | 硬件（数据存储、输入/输出、客户端、服务器）、网络、通信、位置 | 软件、数据、信息、知识 | 员工、任务、管理、制度、规程、培训、认证 | 船、建筑物、电力、水、空气、环境 |
| **结构/设计** | 单一性 |  |  |  |  |
| 唯一性 |  |  |  |  |
| 集中性 |  |  |  |  |
| 一致性 |  |  |  |  |
| 可分离性 |  |  |  |  |
| 逻辑/现实错误、不可靠性 |  |  |  |  |
| 设计敏感性、脆弱/限制/有限性 |  |  |  |  |
| 不可恢复性 |  |  |  |  |
| **行为** | 行为敏感性/脆弱性 |  |  |  |  |
| 恶意行为 |  |  |  |  |
| 强度 |  |  |  |  |
| 可塑性 |  |  |  |  |
| 欺骗性 |  |  |  |  |
| 自满情绪 |  |  |  |  |
| 易腐蚀性/可控性 |  |  |  |  |
| **常规** | 可访问/可侦测/可识别/透明/可截取 |  |  |  |  |
| 难以控制或管理 |  |  |  |  |
| 不自知和不可预测性 |  |  |  |  |
| 可预知性 |  |  |  |  |

**表S2 脆弱性矩阵**

该理论运用了诸多方法来识别那些可以考虑用来解决一倍发现的脆弱点的安全技术。

首先，矩阵罗列了每一个脆弱点所对应的首选或其次的安全避险备选方案。矩阵也能在安全技术遭遇附加风险（当他们发生的时候）发出警告（见图S2和S3）。矩阵也涉及到那些实际上由脆弱性推动的安全技术的情况，这也可以看做一种有益的结果。

其次，用户怀着不一样的意图、职责，也拥有不一样的职权。透过对建立于评估者最初的工作任务：持续运作、不断发展、策略——之上的众多安全技术的筛选过程，这套理论也反映出上述事实（即用户的差异）。这套理论还可以把信息系统的危害分割成为一次攻击或故障的基本组成部分，它们可能与信息、访问、目标漏洞、不回应和评估相关。目标系统的信息用来设计和实现对系统的攻击；访问用来收集信息和在目标漏洞上发起攻击；同时如果没有目标漏洞，攻击首先就是不可能的；不回应（甚至是对最初部分的不回应）的目的是将操作的反作用降低到最小程度；最后，当其他操作依赖一次成功的攻击的时候，评估一次攻击成功与否就很关键。系统意外故障的情况下，只有导致故障的目标漏洞是决定性的因素。

漏洞A

漏洞B

漏洞C

漏洞D

漏洞E

漏洞F

漏洞G

……

漏洞T

方案1 方案2 方案3 方案4

次要

首要

次要

首要

警示

**图S2 连接漏洞与解决方法的思路**

安全技术可以：

2减轻脆弱性（主要的）

1减轻脆弱性（次要的）

0 利用脆弱性

-1脆弱性（主要的）

-2脆弱性（次要的）



**图S3脆弱性与安全技术的关联值举例**

除了更深入地筛选技术，这种划分方式拓展了一种重要的观点：从攻击的角度，抵挡一次攻击的关键组成部分，可以有效预防攻击，且不必发掘出（系统本身）根本性的作为（攻击）目标的漏洞。同时，这种划分方式在评估人员现状和工作角色的基础上，为他们建议更多的选择。比如说，操作员不能重新设计别人架设的信息系统结构，但他们可以自主得限制信息流通和接入系统。

**使用VAM理论下的自动化辅助**

最终，自动化的模型工具（如使用Excel表格）极大地提高该理论的可用性。工具引导评估者完成评估脆弱点，评价风险，检验安全技术的警示和屏蔽功能，选择使用的技术，加固后的二次评估。图S.4展示了用以说明评估者的工作角色的工具的一部分，并从5个攻击模块来估定脆弱性。读者可以从[www.rand.org/publications/MR/MR1601/](http://www.rand.org/publications/MR/MR1601/)获得中国表格的副本。



**图S4 VAM工具中的使用者与攻击组成要素（名义值）**

# 结论

VAM理论提供了相对全面的、自顶向下途径以获得信息系统安全。VAM理论具有新颖的评价和建议产生矩阵以及筛选方法。

脆弱性和安全的分类方法已经很完备。既然脆弱性的内容经常应用于每一种类型的对象，把这些内容看作从系统对象中分离出来的部分已经被证明是检查脆弱性的一种行之有效的方法。同样地，每一种对象的类型都在信息系统中扮演了重要的角色。对基本层面的脆弱性以及它的一些延伸内容明确包含了物理的、人与社会的以及基础设施方面的对象，当然还有网络和计算机硬件对象。这样的脆弱性识别出，同时也包括了信息系统在这几个方面的重要性（就其本身的功能而言）。

VAM填补了一个在查找系统脆弱点方面的清晰的引导和提供相关建议的方法上的空白，它提出针对系统脆弱性的建议，而基于脆弱点、评估者类型、攻击组成要素的筛选方法帮助提高加固建议的现实可用性。

如果能在使用该方法进行评价的过程中借助计算机的帮助，那无疑会极大地提高方法的实用性。而现在的方法能比Anderson等人在1999年开发的那个版本提供更多的加固建议。由于拥有大量安装了Excel软件的PC用户，现在使用的Excel在实用性方面有很大的益处。这个表格同样使用户在生成分析报告，甚至是输入自定义评级算法，以适应本地需求和环境方面变得更加灵活。

该方法能同时服务于个人和团体。个人用户通常专注于他们特殊的环境和责任范围，而团体则能找到多种专业意见来对分析产生影响，以及全面观察一个组织的不同部门。该方法能在不同的部门内平行使用，一次来聚焦他们各自的脆弱性，也能在随后融入一个高水平的评价。一旦每一个部门的理由和对应到组织功能的映射被理解。

# 致谢

DARPA/ITO的Brian Witten曾提议测试早期发布的用于评估计算机风险的RAND MEII理论的效用、完整性、实用性。该理论通过把它应用于真实防御关键信息系统的对象，帮助确认他是否有效。我们感谢他对这项工作的鼓励和建议。

在RAND，我们感谢Scott Gerwehr在信息安全方面使用“欺骗”方法的洞见。Robert Drueckhammer也发起许多有益的讨论，在计算机支持安全实践方面。MSgt Les Dishman（美国空军）提供了我们需要的文档，使我们获益匪浅。最终，我们同样感谢来自RAND的Shari Lawrence Pfleeger 和Steven Bankes的非常有用的建议、问题、观点。他们是这份报告的复审人，我们的报告能更好，是他们深思熟虑去审核的结果。

另外，Claire Antón给予我们ISO和他们使用的标准方面有价值的信息。

# 缩写

|  |  |  |
| --- | --- | --- |
| ATO | 空中任务指令 | air tasking order |
| C2 | 命令与指挥 | command and control |
| C4I | 命令、控制、交流、计算机、智能 | command, control, communications, computers, and intelligence |
| CARVER | 标准、可访问性、可恢复性、脆弱性、影响、可再识别性 | Criticality, Accessibility, Recuperability, Vulnerability, Effect,and Recognizability |
| CC | 信息技术安全通用评估准则 | Common Criteria for Information Technology Security Evaluation |
| CERT | 计算机应急响应小组 | Computer Emergency Response Team |
| CI | 反间谍 | counterintelligence |
| COTS | 集成通信技术 | commercial off-the-shelf |
| DARPA | 高级防御研究计划局 | Defense Advanced Research Projects Agency |
| DDoS | 分布式拒绝服务 | distributed denial-of-service |
| DoD | （美国）国防部 | Department of Defense |
| EMP | 电磁脉冲 | electromagnetic pulse |
| GCCS-M | 全球指挥与控制系统-海事 | Global Command and Control System–Maritime |
| I&W | 标志和警告 | Indications and Warning |
| I/O | 输入/输出 | input/output |
| INFOCON | 信息环境 | Information Conditions |
| IO | 信息操作 | information operations |
| IP | 因特网协议 | Internet Protocol |
| ISO | 国际标准化组织 | International Standards Organization |
| ISR | 情报、监视、侦察 | intelligence, surveillance, and reconnaissance |
| IT | 信息技术 | information technology |
| IVA | 综合脆弱性评估 | Integrated Vulnerability Assessment |
| IW | 信息战 | information warfare |
| JFACC | 联合部队航空分队指挥官 | joint force air component commander |
| LAN | 本地网络域 | local area network |
| MEII | 最小必要信息基础设施 | minimum essential information infrastructure |
| MOU | 谅解备忘录 | memorandum of understanding |
| Nmap | Nmap端口扫描工具 | Network Mapper |
| OCTAVESM | 关键操作风险、资产、脆弱性评价 | Operationally Critical Threat, Asset, and Vulnerability Evaluation |
| OPSEC | 操作安全 | Operations Security |
| ORM | 操作风险管理 | Operational Risk Management |
| PKI | 公钥基础设施 | public key infrastructure |
| PP | 保护概要 | protection profile |
| PsyOps | 心理战 | psychological operations |
| ROM | 只读存储器 | read-only memory |
| SIPRNet | 安全IP路由网 | Secure Internet Protocol Router Network |
| SW/HW | 软件/硬件 | software/hardware |
| TCSEC | 可信计算机系统评估 | Trusted Computer System Evaluation Criteria |
| USAF | 美国空军 | United States Air Force |
| VAM | 脆弱性评估与减灾 | Vulnerability Assessment and Mitigation |
| VV&A | 验证确认 | validation, verification, and accreditation |

# 第1章 介绍

许多组织的关键功能依赖于信息系统载体的核心。保护好这些载体以抵御当下和未来的威胁，这要求对于系统的脆弱性有广泛和客观的观察，如同资源有限时对安全性和稳定性的创造性的思考。信息交互、资源分享、内外勾结的行为以及设计的缺陷和限制等诸如此类的可能引发风险的问题，都会给系统的安全和操作带来危险。不幸的是，理解这些地方在信息系统的可靠性、系统风险以及如何降低风险，已经成为一个严峻的挑战——尤其是在对那些曾被利用的漏洞不是了解甚至一无所知的时候。

RAND开发和完善了一套理论用来让分析师理解这些关系的理论。它可以拘谨识别系统的风险点，并从技术角度提出合理的加固建议。这就是建立在早期由Anderson等人所做工作基础上的“脆弱性评估和加固方法论/理论”，它填补了一个急需被填补的空白。它以横跨信息系统的所有方面，引导评估者做出全面的评价，并把脆弱性映射到可以定位他们的特别的安全技术上。

VAM理论采用自顶向下的线路和查询方法，解释那些已知的、已被利用或暴露在外的脆弱性，同时也可以发现那种确实存在，但在操作工程中没有遭遇过活被利用的脆弱性。所以，这套理论在规避当前和过去的威胁和弱点的时候，也可以协防抵御来自未来的威胁和系统崩溃。另一方面，狡猾的对手同样也在寻找新的途径去攻击那些尚未被保护起来的资源，也就是信息系统的软肋。因此，这套理论作为i恶意逃处理和平衡当下及未来的威胁方面具有一定的价值。同样，信息系统的复杂性以及他们在结构体系中的功能集成越来越密集，也要求进行额外的考量以确认设计和结构中的缺陷已经被过滤掉了。

## **VAM理论的使用者**

这份报告可能会引起一些从事风险评估和安全加固的个人和组织的关注。因为它简化了对新的脆弱性的识别过程，可能让一些设计新系统的人员特别感兴趣，同样的也有那些关心高性能和资源丰富的系统攻击者的安全专家，比如企图寻找新的安全问题并通过巧妙的手法将之利用民族主义国家和恐怖分子。VAM理论也使得全面评估已知脆弱性的过程得到简化，因而用户可以描述最严重的问题，并通过合适的路径将其定位。

本理论为脆弱性的源头（不论已知还是未知的）、系统对象、安全选择提供了非常全面的意见，以此帮助避免系统发生偏差。所以，无论外部访问这还是组织内部的人员都能发现它的价值。当然，本方法也要求对陷入泥淖的系统有足够客观的认识和全面的认识，所以外部人员需要访问最它最了解的专家，而内部成员则需要以不太保守的观点去评价自己的系统。

我们也发现，在使用该方法进行评价系统的时候，组织中不同角色的成员也有对其自身有不尽相同同时又有意义的安全需求。这样，设计者、操作人员、策略制定者如果可以有侧重地使用本方法，他们都可以从中获益。

再进一步，我们发现本方法在信息战概念中的价值，在这种条件下，信息系统的弱点和安全响应是首要需要关心的对象。所以，本理论也可能是涉及信息操作的其他方面的人员感兴趣的对象，包括漏洞欧股利用和攻击方面。

## **早期成果**

早在1999年，Anderson等人在RAND发布了Securing the U.S. Defense Information

Infrastructure: A Proposed Approach（保护美国防御基础信息设施：推荐方法），也就是MEII研究。他们最初的研究目标是为美国国防部探索MEII的概念。响应的报告描绘出六个步骤对国防部的要害信息系统进行加固。它的重要贡献是罗列了用来执行命令、控制和情报工作的复杂系统中存在潜在脆弱性的20个基本方面。同时它列出了可以应用在多种途径的减灾安全技术的13个基本方面，提供了颜色和数字标识的矩阵来表现最合理的抵御风险的安全技术。更早期的研究结果仅停留在理论方面，而没有应用到实际的系统上去。

2000年11月，DARPA的Brian Witten提出早期的研究框架可以用来研究一项很具有操作系的美国国防部C2系统来评估理论的在覆盖没有被考虑到的弱点来源的效果，同时来提供相关的安全技术方面的加固建议。于是后续的研究在2001年春天启动，这份报告就是这项研究结果两份文档其中的一份。

研究的过程中，我们描述了早期的方法（映射到一个安全技术列表上的脆弱性列表）很有意义。但是，这份列表需要更新升级，也需要更好的方法来管理生成的拉梁安全建议。这份最新的报告体现了方法的升级和拓展。VAM理论现在确定了以下内容：一个更全面和合理的属性分类，这可以引导脆弱性找到处理它们对应的安全技术；属性和安全技术之间的映射图；备选的安全技术列表中提炼筛选出的最佳办法；不依赖额外信息引导的可以自动生成表格和过滤检测结果的软件工具。

本报告的作者还进行了一项应用VAM理论进行的军事战术信息系统的问题的研究，研究成果没有发表。因为某些研究的细节是比较敏感的信息，上文提到的这份结果只能用于政府授权的个人那里。相关人员可以通过联系笔者Philip Antón([anton@rand.org](mailto:anton@rand.org))或者Robert Anderson ([anderson@rand.org](mailto:anderson@rand.org))获得相应结果。当然，没有涉及敏感信息的部分已经全部包括在下文中。

## 报告的结构

报告剩余部分概要如下：

第二章定明确了是什么构成信息系统。随后它展开了概念性的讨论，指出是什么导致脆弱性，并介绍了一些概念来帮助理解脆弱性、出现在什么地方以及如何被加固。

第三张概括地介绍了VAM的六个步骤，并用一个国家的例子贯穿了介绍内容的始终。这一章也介绍了VAM与其他安全方法相比下的特点，以及如何借助其他方法。既然VAM的核心牵扯到识别脆弱性和对加固安全技术的选择，那么第四章到第七章就详细讨论了VAM如何帮助用户完成这些工作。

第四章深入讲述了信息系统中可以导致脆弱性的对象的属性（六个步骤中的第三步），并例举了一些例子，指出他们如何演化成已知的信息系统的脆弱性。

第五章深入讨论了信息系统安全技术，同样也介绍了它们如何演变为已知的安全解决办法。

第六章讲述了VAM理论如何把第四章的脆弱性映射到第五章讲的安全技术上，明确了如何定位已知脆弱性问题的特殊引导方法。其次，这章介绍了筛选安全技术来提高其适用性——这些技术由矩阵标出，对应的特殊的用户类型和攻击事件。第五章和第六章描述了方法的第四步，也支持选择安全技术（第五步）。最后，这章举了几种特殊安全对策的例子。这些对策可以被使用该方法的评估者用作处理特殊的或者一般的信息系统的脆弱性。

第七章介绍了在VAM使用电子表格来自动查找方法中的信息和说明。

第八章讨论了现在的VAM方法的不足之处，下一步可能进行的计划和一些更一般的讨论。

第九章展示了最终的结果和观点。

附录涵盖了一些详细信息，它们的位置在矩阵当中的“等级”那一栏之后。这些信息为脆弱性和相应的安全技术建立映射的矩阵。

# 第2章 概念和定义

在描绘VAM理论的内容和过程之前，我们需要先探索以下其使用的几个概念和术语。比如说，是什么构成一个信息系统？什么造成系统在被攻击或故障时易于损伤？什么类型的组成部分容易包含脆弱点？

## 安全

是什么导致系统在问题中发生故障？从这个问题出发，我们意识到“安全”的含义就是不同的事情对应不同的人。我们收集了大量而广泛的安全方面的观点，囊括了任意影响系统安全和可靠性能的因素。因而，系统故障的增加不仅仅来自于对手的蓄意攻击，也来自于意外事故、缺陷、故障、性能限制和自然因素。

## 信息系统

我们使用“信息系统”的说法，宽泛的涵盖了所有系统或某个组成不愤怒（无论物理的、网络的、虚拟的、计算机、交流、人还是社会），它在存储、进程、处理或者转换等方面均被涉及到。而信息处理系统的范围则被定义的狭窄得多（比如纯粹是电脑、软件和硬件），我们通常关注与信息相关的功能——组织的或服务于组织的。一个信息系统组成部分中的任何可能导致故障或崩溃的因素，都会对整体的性能和它的任务构成威胁，因此都应当在加固系统时被考虑。

## 系统对象类型

我们围绕是否属于以下四个方面来确切地描绘不同类型的系统组成部分，即：物理、计算机、人与社会、可用基础设施。

**物理。**这些对象包括，例如，硬件设备（比如数据存储介质、输入输出设备、客户端设备、服务器），节点内部和之间的网络和通信（流），处于系统结构内部多个层次的物理位置。

**计算机。**计算机对象包括，例如，软件、数据、信息和知识。通常，它们存在于虚拟的电子世界中，甚至只能做概念上的描述，它们的形态不受它们存在物理模式和媒介（磁盘、纸张、二元交换机）的限制。

**人与社会。**人与社会的对象包括，例如，用户和它们的同事、开发人员、管理者、命令组织、策略、手续、培训、认证。

**可用基础设施。**基础设施包括，例如，物理掩体（如建筑物、交通工具）、电力、水、空气、其他环境条件。

该对象列表的范围允许更全面的检查，而不仅仅是计算机硬件和软件（也就是你通常注意到的那些）。比如，信息被一个组织内部的人来处理和智慧，而不是被电脑和网络。实际上，人对信息的处理是信息系统的关键要素，而且，人和社会系统的弱点必须在全面的风险评估中被定位出来。

### 对于“对象”概念的使用

对于“对象”这个概念的使用，形成了信息科学中一个常识性的理论工具。它允许一个主体代表一个人、一个地点、或者一件事情——当你要阐明它的性能或者“兴趣行为”。信息系统各组成部分可以划分为概念层面上的“对象”，而这样的划分允许我们注意到那些通常在考虑系统安全的时候经常忽略掉的那部分组成物。

“计算机和网络”对象是自动化的、计算机控制的，也可以是软件或者虚拟组件（它们也经常被看成是信息系统的组成部分）。

当然，这些对象通常也要依附于现实的“物理对象”（例如用来实体化虚拟对象的物理设备、设备所处的建筑物以及它们能够影响到的物理范围）。

“人”也作为一个对象，在系统中对信息进行处理。他们使用、管理并且控制系统，以及系统的对象和目标。人存在于众多影响他们行为的社会体系中。

最终，所有的这三种对象类型都依附于“基础设施”——不是信息系统形式上的组成部分而是对系统提供至关重要的支持（例如电力、空气、饮食、温度控制等）。

## 哪些属性是导致脆弱性的来源

脆弱性的出现源自于信息系统对象中可识别的属性。在从本质上做出相对全面的、高水平的脆弱性评估的前提下，VAM理论明确地探索这些脆弱性的起源，同时将它们映射到所有的对象类型上。这样的方法可以引导评估者考察全部的脆弱性——不仅是已知的或者已经被利用的­。也探索了全部系统对象的脆弱性——不仅是计算机网络相关的组成部分。

Anderson等人最先发现了信息系统的属性引起信息系统脆弱性这一概念，我们的工作便建立在这一概念之上——当我们能明确地从属性的角度分离各对象同时扩展引起脆弱性的属性列表。

由系统对象类型分离脆弱性的属性，可以检察潜在的脆弱性。通过遍历或枚举的方法，将一类对象类型到另一类对象类型，应用于与之关联的脆弱属性，则是这一行为的前提。例如，单一性不但可以在计算机软件层面和物理层面涉及到，那些信息系统当中具有特殊操作技能的不可替代的人员也具有这样的属性。

### 安全技术

最后，我们运用大量已经投入使用或正被信息安全团体研究当中的安全技术——它们通常根据减灾途径被分成多类。因此，我们能在方法论的指导下，在抽象概念和描述它们如何把自身和其作用的脆弱性关联起来的过程中，使用这些技术 。每一类安全技术都被罗列在第五章。这里的分类不是对等的，既往而言一些技术得到的关注总是高于其他的。有时候，这不对称是相当合理的，而另一些时候，提供重要支持的新技术在后来得到更多的关注。按照方法类型来考量技术帮助人们在遇到令人头疼的问题的时候能寻找最好的解决办法，而不被这些建议之间的差别分散注意力。

# 第3章 VAM方法论和其他风险评估中的DoD实践

## VAM方法概述

1990年代，RAND发布了用于改善危险系统安全水平的六步法理论（Anderson等，1990），这些步骤是：

1. 识别组织结构体系中那些基本的信息功能
2. 识别基本的信息系统，当第一步中的功能被执行的时候，这些系统也是很基本的
3. 识别这些系统中的脆弱性
4. 寻找合适的抵御脆弱性的安全技术
5. 综合考虑现有条件的限制，以及成本和收益的因素，选择和使用最佳的解决办法
6. 测试其在风险下的强度以及现实可行性

有必要时重复第3至6步。

要特别注意一点：这套理论包含了明确的有脆弱性到安全技术的映射（第四步）。这个映射构成了这套方法的核心，同时在定位脆弱性上为评估者提供了明确的指引。这份报告当下的任务是是增大了这个矩阵的规模和复杂度，以提高该矩阵方法的完整性。

我们从非常全面的角度解释了这六部过程是如何工作的，也用一个理想化的军事案例来实践我们的理论。尽管我们通过一个军事案例来阐明这基本的六个步骤，实际上VAM方法也可以用来被利用到其他关键的商业和政府功能。

在VAM理论中，大部分相关的部分出现在第3和第4步当中（对脆弱性的识别和生成规避风险的安全技术）。第四章到第七章的内容是一些没有包含在内的步骤的其他细节。

### 第1步 识别基本的信息功能

信息系统本身不是所追求的目的。他们被一些个人和单位使用，用来提供不尽相同的功能和操作。在有限的资源条件下，脆弱性会威胁到建立在基本信息上的功能，而这些脆弱性应最先被定为。因而，一个人若要试图发现和整改掉这些问题，首先需要区分什么是基本的功能。

过程——一个可以引导对某组织的基本功能识别的目标进程。

首先，“策略到任务”的分析过程（Lewis和Roll，1993；Thaler，1993；Kent和Simons，1994）能够被执行。在这里，组织的目标和策略可以被优先定义出来，相应的策略也被映射到那些被设计用来执行策略的任务（功能）上。

其次，对这些任务形成支持的特别的信息功能也被识别和分类。

第三，基本措施得以改进，并用来为信息系统进行评级。他们的分类包括：基本的、有价值的、可消耗的。基本的功能是指，一旦它们受到损毁，全面阻止组织执行这些功能的重要任务（由“策略到任务”需求来定义）。有价值的功能是指工作在它们外围的对象是可被识别的。可消耗的功能是指工作在他们外围的对象有着可被接受的可识别的性能代价和风险。

最后，所有的这些已经被识别出来的功能作为一个整体进行信息系统全方位的风险评价。需要特别注意的是应当在这个过程中找到多数或者全部任务的基本和有价值的功能。同样地，支持多数任务的功能的集合或者逻辑组合应该尽可能地被发现。因此，功能性的识别域需要被特别关注。

案例。在一个假设使用该理论的军事组织的例子当中，一位联合部队航空分队指挥官（JFACC）[[1]](#footnote-1)在一次空中战役中执行一系列职能，包括制定和发布空中任务指令（ATO）[[2]](#footnote-2)，分析后勤支持需求、计划燃料的分配、计划医疗措施，以及与其他军事指挥官（见图3.1）的电话会议。在这些功能罗列下来以后，ATO（实体椭圆球中）的制定和部署就可以作为关键功能而被比较和选择了。而这一关键功能必须被附近的团队所支持。其他的功能则对时间要求不那么敏感，而且为ATO的制定（和最后执行）也仅提供次要的支持。因此，我们将ATO的制定和分发选择为JFACC组织中“基本的信息功能”。

### 第2步 识别基本的信息系统

在给出了第一步中所说的与信息关联的基础功能后，这些基本支撑和执行这些功能的信息系统就应该被识别出来了。

过程。第一，用来执行基本的信息功能的信息系统在第一步中被之别出来，这些系统需要被识别出来，然后分类。这些系统构成基本信息系统的备选列表。

此外，对“基本性”的评价在对信息系统中被采用和发展为“基本的”“有价值的”和“可消耗的”。最后，所有被识别出的系统围绕功能成为一个整体，在此基础上，发展出一个全面分级的信息系统。需要特别注意要严格地寻找多数或所有功能的系统。同样的，支撑大量功能的系统集合或逻辑组群应该尽可能地被识别，因此，识别系统的逻辑集群需要格外小心。

案例。在这个不间断的案例中，如果JFACC部署在一条船上，他或她的同事使用大量的信息系统开支持其操作。这些信息系统包括全球指令和控制系统-海洋（GCSS-M）、服务后勤的全球作战支援系统（GCSS）、在许多多用途军事电脑上提供的常规操作环境（COE）、机密互连协议路由网（SIPRNet）、公共交换电话网（见图3.2）。因为第一步识别出ATO的生成与发布，并作为一项基本功能，我们需要选择这个支持这项功能的基本的信息系统。全球指令和控制系统-海洋和机密互连协议路由网（左下黑框中）是支持ATO的基本信息系统。在这两套系统中，从传送信息到JFACC以进行加工的全局流程，机密互连协议路由网可以作为信息交互主干而被识别出来，对于ATO的制定和发布，它是最基本的组成。尽管对于快速制定ATO来说，全球指令和控制系统-海洋一样基础。

### 第3步 识别系统脆弱性

在第二步给出了基本信息系统的优先列表后，我们就可以把注意力集中在发现系统的脆弱性上面。在这一步，VAM理论开始提出建议，而很多其他的方法理论在确定脆弱性的时候缺乏明确的建议。应该注意到的是，一次成功的脆弱性评估需要系统开发和使用者的具备如下描述的洞察力和经验，所以，不论是方法来指导还是经验都很重要。

这里，我们以一个假想的例子来描述第三步中涉及到的步骤。而第四章详细讲述这样的评估如何在一个对象上进行，导致脆弱性的固有属性的自顶向下透视。同样附加的包括介绍了脆弱性表格，特殊的脆弱属性以及系统对象类型的属性差别。第四章到第六章结尾都会包含一些常规脆弱性的具体案例。

过程。VAM理论提供了宽泛的方法来进行脆弱性分析，这需要通过请评估者完成一个矩阵来实现。这个矩阵包含一组相对完整的属性分类，而这些属性又是围绕所有系统对象可以导致脆弱性的属性（详见图解表3.1）。

系统内的脆弱性应该在多方面上被检测。例如，一个计算机或网络的对象的脆弱性可以在全球架构的层面上检查（例如主用系统的相互作用，以及系统全球范围数据交互的系统）；架构中的应用组件（也就是具体的应用分类，即从商业软件到被设计用来解决用户的唯一性需求的定制的应用）；特定的支持软件（例如数据库软件、加密和解密程序、支持库等）；通信级组件（这类软件的借口直接面向通信线路），等等。目标就是检查影响系统的合适和可靠操作的关键组件，不论在哪个层面上，对重要性的判断是非常重要的，不然用户将在大量无关的细节中迷失。

伴随这脆弱性的分类，评估者应当回顾过去关键系统的经验，并回答以下问题：

* 在过去有什么做的不好，为什么？
* 是什么影响了这些失败情况？
* 做过什么纠正的行为？

我们应该在用理论模型[[3]](#footnote-3)解释这些经验上多花些功夫。如果这些经验一直吻合模型，那么评估者应该收集历史经验上的失败数据以帮助发现哪些问题在过去是更严重的。如果模型不充分，那么评估者可以去提炼或者拓展模型，也可以寻找其他模型，可以用来发现故障发生的潜在原因。这些模型不需要特别详细，但是可以帮助人们识别出是哪些脆弱属性一直在导致故障，以及哪些是刚刚出现在系统当中的。

评估者也可以通过检查已经使用的安全技术来发现脆弱性，同样的，考量第四步中的矩阵下部和安全技术联系在一起的脆弱性警告，也可以起到相同的作用。

最后，评估者需要评估何种理论上的脆弱性存在于系统中，而对于这种系统，人们一般没有现实或模拟的经验。评估者需要检查系统的组成部分，使用一张完整的脆弱属性列表，它类似于一张检测表（checklist）。这些属性的出现代表了一类潜在的脆弱性，这类脆弱性需要被更加深入地研究，以判断它们可能有多么严重。再次强调，系统功能的理论模型可以有效地在探索和解释这些属性在潜在的危害和故障中扮演的角色。统计结果可能有用也可能没用，但是理论上存在的威胁和故障的发生“空间”应该被好好检测，以评估影响重要信息系统功能的潜在脆弱性。

案例。考虑到第二步中提及的GCCS-M 和SIPRNet，我们希望知道什么样的严重漏洞是我们需要定位，这样才能对信息系统进行良好的支持（见图表3.3）。不得不提一句，对军用系统的脆弱性评估超出了这份报告讨论的范围，所以我们有选择性地看待脆弱性。理论上讲，我们处理潜在的类型的脆弱性，同时发现GCCS-M包括脆弱性E和F。如果安全技术3已经被使用了，那么用户接下来就硬看看脆弱性T是否出现了（见图表3.4）。记住，我们需要从GCCS-M的多个层面寻找脆弱性，所以我们应该把它看做一个整体，它的原始应用和关键的支持组建（例如SIPRNet）。对于SIPRNet，同样需要对多个层面进行检查，包括所使用的政府和商业软件，通信系统，网络系统和路由器，管理操作员以及物理组件，例如布线和关键的支撑基础设施。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **脆弱对象** | | | |
| 物理 | 计算机 | 人与社会 | 基础设施 |
| **脆弱属性** | **结构/设计** |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **行为** |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **常规** |  |  |  |  |
|  |  |  |  |
|  |  |  |  |

图表3.1

### 第4步 从VAMM方法提供的候选条目中寻找合适的安全技术。

脆弱性识别会是一项困难的任务，但是对如何为其进行定位进行判断则更加令人气恼。VAM理论提供了一种理论上的定位方法。不仅有助于为头脑中自然想到的减灾方法排出一个合理的顺序，也提供一个相对完整的检测其他并不是首先想到的技术的手段。

过程。VAM理论包含一个较大的矩阵，这个矩阵识别与每一项脆弱性有关的常规安全技术。这个矩阵同样识别可以带来新的心得脆弱性的安全技术。一个关于该矩阵的图解包含在下面的例子当中，它描述了这个矩阵如何用来识别潜在的定位与脆弱性相关的安全技术。

GCCS-M（全球命令与指挥系统-M）

潜在脆弱性：

脆弱性A

脆弱性B

脆弱性C

脆弱性D

脆弱性E

脆弱性F

**图3.3 识别哪一些脆弱性作用在关键系统上**

第六章和第七章描述了这个矩阵的细节，围绕着可用性问题和自动甄别候选安全技术的电子表格的实现。

案例。在第三步，脆弱性E和F作为GCCS-M的关键的概念性脆弱性被识别出来。图3.4展示了一个关于VAM表的概念性表格，它将这些脆弱性定位到适当的减灾技术上去。在我们的例子中，技术2和技术4是分别映射到脆弱性E和F的最初的技术。技术2和技术3是备用选项，第二候选技术映射到F。所以，我们首先检查技术2和技术4，看看他们是不是适合GCCS-M的需求。如果不是，我们再考虑技术3。

脆弱性A

脆弱性B

脆弱性C

脆弱性D

脆弱性E

脆弱性F

脆弱性G

脆弱性T

技术1 技术2 技术3 技术4

首要的

次要的

次要的

首要的

警 告

**图3.4 自脆弱性到安全减灾技术的映射概念**

这张图表同样可以识别由于使用了减灾技术而导致的对脆弱性的影响。这里，技术2可能在一些情况下引入脆弱性T，所以一个警告标记出来，看看技术3使用的适合是不是带来了脆弱性T。

既然这个例子如此概念化，读者可能希望看到第六章的结尾来消化安全技术的例子——也就是针对常见的信息系统脆弱性而开发出来的那些安全技术。

### 第5步 选择和应用安全技术

过程。在第四步中识别出的合适的安全技术列表在此时有必要进行一些筛选成为一个集合，它可以被有效地赋予可用的资源和评估者单位的责任。而评估者可以直接应用某些技术，其他的技术可能超出了评估者及其组织的范畴。在随后的例子中，分类中靠谱的方法可以通过评判成为可信的部分。同样，大量被矩阵生成的选项可以作为其他部分的参考，那些部分可能不够直接和明显，不过仍然可以减少系统的脆弱性。比如说，管理、 反间谍活动（CI）、补偿措施可以帮助保护系统和组织攻击——当软件变更或者保护程序不能被用户群体选用的时候。

案例。在GCCS-M的例子当中，我们接下来会应用技术2、3、4来支持GCCS-M（见图3.5）。

### 第6步 威胁下的健壮性测试

简单地应用更多安全技术并不意味着问题就能得到解决。改良的系统应该在真实的或模拟的威胁条件下进行测试，以判断减灾方法是否有效。从这类测试中得到的脆弱性信息可以应用会第三步中，以帮助决定使用或者考虑其他安全选择。

过程。测试系统改良后的效果。对于这样的测试，Red Teaming是一种重要的途径，因为它提供了一种独立的脆弱性和健壮性的检测方法。这些小队不应该仅仅测试攻击已知的问题和修复点，同时应该寻找和发现新的问题（包括任何被新增加的安全技术无意引发的问题）。剩余的关注点则应该放在现实的实践上面（或者在适当的时候，某些情况下的操作设置），以检测相关的规程或者外围的设施。

其他的检测方法可能也是有用的。专门从事安全评估的安全行为的实施者（或独立的机构或公司）同样可以管理执行情况的监控和落实。如果故障和危害统计信息在地3步中被使用，这些值可能通过足够长而且有效的周期与过去执行时的统计数据相比较，来量化减灾的成果。在系统中与计算机网络相关的部分中，自动化攻击或者使用工具可以用来辅助探索系统在遭到模拟攻击时能否良好响应。当然要注意到，很多自动糊工具在常规的、已知的和已经被利用过的漏洞面前，其能力是及其有限的。所以，它们并不是要全面划定系统组件的完整边界，尤其是当物理的、人与社会的以及基础设施方面的组件没有承担太大压力的时候。

全球指挥与控制系统——M

技术1

技术2

技术3

技术4

**图3.5 寻求可考虑的安全技术**

最合适的测试程序将包含一类威胁模型，它可以评估威胁成功损害系统的可能性。这些模型应该足够宽泛，以同时涵盖威胁的效能来发现没有预先被利用的漏洞以及威胁的技术能力以利用漏洞。

测试可能把目光集中在系统中已经修补过的部分，而系统和其他功能组件剩余的第二和第三位的影响也需要被考虑。

最后，由前5步完成后而形成的测试的结果应该被编写进文档并被评价，以觉得是否需要从第三步开始进行附加的工作。

案例。在我们的例子当中，一个（模拟的）威胁应用在GCCS-M中以确定其健壮性（见图3.6）。

全球指挥与控制系统——M

技术1

技术2

技术3

技术4

（模拟）威胁

**图3.6 测试改进系统对抗（模拟）威胁**

## 其他美国国防部的脆弱性评估方法

许多方法和评估技术被运用在商业部门和美国国防部的脆弱性识别和安全行为设计活动当中。我们简短地描述一下后面的常见评估方法和技术，并且讨论一下VAM理论与它们有怎样的依赖关系。

### OCTAVE（关键操作风险、资产和脆弱性评估）

关键操作风险、资产和脆弱性评估SM（OCTAVESM）是一个由Carnegie Mellon大学的软件工程学院开发的框架，用以发现和管理信息安全风险（Alberts等人，1999,2001）[[4]](#footnote-4)。它定义了一套用以识别重要的组织任务、威胁和可能被威胁利用的脆弱性的流程。OCTAVE也包含可以开发保护策略的完整流程，它可以用来见减少这些脆弱性和威胁所造成的风险。这个框架在下面的“进程”中被罗列了出来。

1. 识别企业信息。

2. 识别运行区域内的信息。

3. 识别员工信息。

4. 建立安全需求。

5. 映射高优先权信息资产到信息基础设施。

6. 执行基础设置脆弱性评估。

7. 实施多维风险分析。

8. 拓展保护策略。

OCTAVE具有非常浓重的过程导向色彩，它帮助评估者构建一个工程来分析和规避信息安全风险。这些过程指引方针可以在组织一项活动的过程中扮演一个有价值的角色，但是第6步和第8步没有检测导致脆弱性基本面的系统。同样地，这些步骤没有生成被推荐使用有关脆弱性识别的保护策略。所以，VAM理论补充了OCTAVE框架。这两种方法组合起来使用可能会使评估者受益。

### ISO/IEC 15408：通用标准

国际标准15408——一个通用标准信息技术（简称“CC”）——是一项指引方针，在评估信息技术产品和系统的安全性的时候，它指明系统的哪些方面应当被定位到哪一类进程的分类[[5]](#footnote-5)，[[6]](#footnote-6)。通用标准信息技术着意于信息系统和其组件的用户、开发者、评估者产生关联。通用标准信息技术声明任何安全分析应该检查存放系统设备的物理环境，这些资产需要保护，系统的用途需要被保护（即“目标系统”）。然后授权一份假想、威胁和组织安全策略列表，来引导一套安全对象出现。使用这类对象，一系列安全需求应该被生成，这包括功能和保障需求，如同操作目标系统的环境需求。各种各样的系统中反复出现的需求和设置变成了“保护轮廓”（PP，Protection Profile），它被反复使用来定义系统目标系统的安全需求。“在已经被识别的对象里，包括功能和保障方面的，人们知道这样的需求是有用和有效的。保护轮廓同样包含安全对象和安全需求的基本原理。”[[7]](#footnote-7)包含各种类型的渗透测试在内的凭应该拿出来判断其在多大程度上服从“保护轮廓”。

通用标准信息技术的内容复杂，这体现在它几百页内容的文档上面。过程中的众多脆弱性分析建立在开发者对脆弱性分析的基础上，随后，评估者会对它进行检查以判断其完整性以及“是否有合适的方法可以替代准备环境中的方法，用来防止这里面明显的脆弱性被利用到。”[[8]](#footnote-8)其他表格允许评估者计算针对目标系统的“潜在的攻击”，随着时间的推移它有可能演变成一次真正的攻击。除此之外，表格还允许评估者推断专业意见的需求、系统可用信息、接入许可和环境需求。

我们不能在这里评价通用标准信息技术，我们没有一个意图是想要批评它。我们在公开出版的教材里找不到，当然了，很多给开发者的技术指引和被其他人提到的位于信息系统复杂结构内部**何处**的事物应当查找潜在的脆弱性，**如何**通过一种合理的方法来查找他们，哪一种技术在处理任何已被发现的故障上是最合适的。我们相信VAM理论的概念可以成为通用标准在这些领域中的有力拓展。

### ISO/IEC 17799：信息安全管理的实践代码

国际标准17799[[9]](#footnote-9)起源于关于信息安全管理的英国标准7799。它作为一个内容充实的用例，正在被越来越广泛的应用在确保组织内部的信息安全活动处在合适的地位。它涵盖了信息安全管理的诸多方面，例如下述几条：

* 安全策略（在文档形式的表格中）
* 组织安全（在组织内部、第三方访问安全、程序外包开发）
* 资产分类和控制
* 个人安全，包括合适的职责定义、用户培训、响应程序
* 物理和环境安全
* 沟通和操作管理
* 访问控制，包括监控系统访问和使用，以及移动电脑（例如无线）访问
* 系统开发和维护
* 合规程序

这一套分类从头到尾着实值得褒奖，但是在标准自身当中没有对其每一项都进行详尽解释。标准中的用例是某种“最好的实践”的提示，他是安全的或者不安全的信息系统的经验所导致的结果。但是这个标准没有在理解我们面对的威胁的等级、脆弱性可能潜藏的位置、以及这份指引背后的初衷方面提供太多指导和解释。我们使用这个标准中的安全管理列表，作为我们在研发安全减灾技术的过程中获得的一种资源（见第五章）。

### 操作安全

操作安全（OPSEC）作为一种方法，形成与遇难战争时期。只是一种找出敌人如何获得某些发生在东南亚[[10]](#footnote-10)的战争行为的情报。OPSEC是保护关键信息（见Army Regulation 530-1, Operations Security[[11]](#footnote-11); Joint Doctrine for Operations Security[[12]](#footnote-12); Williams, 1999; and Hamby, 2002）的对策程序。OPSEC包括一些五个步骤：

1.识别需要保护的关键信息。

2.分析威胁。

3.分析脆弱性。

4.评估风险。

5.应用对策程序。

从附加的威胁和风险评估的显著特征方面看，OPSEC的五步大体上平行于VAM理论。然而，OPSEC学说代表性地包含了如何识别脆弱性和选择对策来映射他们的少数指引。这里的技术在VAM理论中也是可以使用的。

### 操作风险管理

围绕所有亟待被规避的危险，操作风险管理（Operational Risk Management，ORM）是另一种以管理风险为目的的减灾程序（例如，包括但不限于信息系统的危险）。[[13]](#footnote-13)操作风险管理是一种被设计用来检测和预测危险的基本面和减少风险到可接受水平的决策程序。操作风险管理的想法原型最初来自于提高开发武器、飞行器、太空运输工具以及核力量过程中的安全性的目的。美国陆军于1991年接受这个规范，用以减少训练和战斗中的损失。ORM包括以下五个步骤：

1. 识别危险。
2. 评估危险。
3. 做出风险决策。
4. 执行控制措施。
5. 监督。

有关ORM的基本概念是管控种可以降低风险的检测，并且提供了这五种可以考虑的实施的步骤，而不是提供了所有类型的危险的细节方法。ORM推荐了可以使用的技术，比如头脑风暴，来生成观点和关系图，把操作分解成各种分类（例如敌方、部队、地形、时间），最终实现在一个时间点上把注意力集中在一个地方的目的。

从制定风险决策的显著特征方面，ORM的五个步骤也像OPSEC一样答题桑平行于VAM理论。ORM学说也包含少量的指引，例如如何识别危险（脆弱性）或者选择控制措施来定位它们。这些技术也可以在VAM方法中得到利用。

### 综合脆弱性评估

海军综合风险评估（Navy Integrated Vulnerability Assessments，IVAs）涉及对系统的用例检测，通过这样的办法可以罗列出最严重的脆弱性。集思广益可以在反馈中执行的安全技术，应执行的命令就会在随后明确了。另外有一种方法，CARVER (Criticality, Accessibility, Recuperability, Vulnerability, Effect, and Recognizability/危害性、易访问性、恢复力、脆弱性、影响和可辨认性)，被当作脆弱性排序的参考指标。CARVER使用很粗略的等级分类。然而，数字分级，尤其是把这些等级分数概括成一个单一的等级的算法是有缺陷的。它的简单的数字评分组合没有准确反映分类之间的重要区别。同样，几乎没有理由让我们相信通过组合问题（例如时间、重要性、物理措施、影响）的不同方面的评级分数可以产生有意义的数字分数。

尽管CARVER有很多问题，下面涉及到的这些基本步骤还是可以使用的：

1. 识别脆弱性。
2. 脆弱性排序。
3. 头脑风暴。
4. 评估风险。

和前面提到的一样，在风险评估的显著特征方面，OPSEC、ORM、CARVER和VAM大体上都是接近的。CARVER包括极少的脆弱性识别的方法，同时所谓的“头脑风暴”看起来似乎也没什么意义。所以，VAM理论中的脆弱性识别方法和探索方法对CARVER的研究来说是非常有借鉴意义的。

### VAM理论技术满足了其他方法中的关键需求

尽管很多这类的方法（包括VAM）使用了相似的哲学理论和指导方法（例如检查关键的功能，识别脆弱性，选择减灾方法，应用安全技术，以及测试威胁条件下的稳定性），但是VAM理论有效地补充了其他方法的不足，例如它提供了一个清晰的机制去帮助一个评估者理解什么导致了脆弱性，什么样的安全技术应用到脆弱性的识别当中，以及什么样的潜在问题可能被安全技术自身的影响而加剧。这些组织为将安全检查制度化做出了积极的努力，这可能使组织采用VAM理论的第三和第四步方法变得有意义，这个过程可以提高其自身的实用价值，同时可以在评估者的交流中为他们提供详细的指导（见图3.7）。

·VAM理论

·ORM（操作风险管理）

1.识别基本信息功能

2.识别基本信息系统

1.识别危险

3.识别系统脆弱性

2.评估危险

3.做出风险决策

4.应用控制

5.应用安全技术

5.监督

6.测试威胁条件下的稳定性

·IVA（整体安全评估）/CARVER

·OPSEC（操作安全）

1.识别脆弱性

1.识别关键信息

2.脆弱性排序

2.分析威胁

3.头脑风暴

3.分析脆弱性

4.评估风险

4.评估风险

5.应用对称程序

4.识别相关安全技术

·VAM匹配矩阵工具

**图3.7 可在其他方法中使用的VAM理论核心**

# 第4章 系统对象的脆弱属性

这里，我们提出脆弱属性列表及其描述，以及他们如何才能在用户表中被映射到系统对象，一些安全问题如何利用这些属性。因而，本章讲述的是VAM理论第3步的细节问题。

## 脆弱属性分类

图4.1罗列了通常可以导致脆弱性的对象内容。脆弱属性包括这些关系到系统的设计结构的内容，也关系到系统发生的动作和行为以及同时出现在结构和行为中的常规属性。这些某种程度上的概念化属性通用应用在许多类型的系统和系统的不同层次上面。

表4.1可以导致脆弱性的属性映射到四种类型的系统对象上面：物理的、计算机和网络的、人与社会的、支持类的基础设施。围绕是否有设计和结构而引发，属性可以被划分成不同的类别，同样的，也可以依据是否有系统对象的行为所引发而划分，亦或这两种情况共同作用的其他类别。

## 一个脆弱性用例和实例

表4.1可以作为一个检测用例来使用，评估人员在检测的过程中可以将其完善。这样，他们可以完整检测正在学习和研究当中的所有对象类型的系统（或子系统）的脆弱属性列表。表4.2展示了通过下面的常见安全问题形成的检测用例。



**图4.1 导致脆弱性的内容**

### 内部威胁

**脆弱属性：**恶意行为

**目标类型：**人与社会

**描述：**人们普遍持有这样一种观点：“内部威胁”（恶意行为由一个可以通过认证访问关键信息系统的受信任的人员发起）是信息系统安全的最大威胁。这样的“内鬼”可能是对组织怀恨在心的一个人，也可能是被敌人或对手敲诈、行贿或者类似的行为而操纵的成员。

### 无法抵御分布式拒绝服务攻击（DDoS）

**脆弱属性：**行为敏感性/脆弱性

**目标类型：**计算机与网络

**描述：**分布式拒绝服务攻击是目前为止最难处理的一种网络攻击，数以百计甚至千记的计算机以数据包或者服务请求来轰击一个特殊的网络节点或部件——通常他们都是些错误的请求，需要通过额外的时间去处理。如果想挫败这种依赖于行为特点和网络敏感性的攻击形式，信息网络必须通过特殊的设计和配置。

### IP欺骗

**脆弱属性：**欺骗性

**目标类型：**计算机与网络

**描述：**用一个数据包或者消息进行IP地址的欺骗，其含义是在IP地址可能出现的位置用一个错误的IP地址对其进行替换。在这样的情况下，这个包或者会话的真实发起者就变得难以被确认，这样许多类型的攻击就可以达成隐藏源地址的目的。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **脆弱对象** | | | |
|  |  | 物理 | 计算机 | 人与社会 | 基础设施 |
|  | 属性 | 硬件（数据存储、输入/输出、客户端、服务器）、网络、通信、位置 | 软件、数据、信息、知识 | 员工、任务、管理、制度、规程、培训、认证 | 船、建筑物、电力、水、空气、环境 |
| **结构/设计** | 单一性 |  |  |  |  |
| 唯一性 |  |  |  |  |
| 集中性 |  |  |  |  |
| 一致性 |  |  |  |  |
| 分散性 |  |  |  |  |
| 逻辑/现实错误、不可靠性 |  |  |  |  |
| 设计敏感性、脆弱/限制/有限性 |  |  |  |  |
| 不可恢复性 |  |  |  |  |
| **行为** | 行为敏感性/脆弱性 |  |  |  |  |
| 恶意行为 |  |  |  |  |
| 不可改变性 |  |  |  |  |
| 可塑性 |  |  |  |  |
| 欺骗性 |  |  |  |  |
| 自满情绪 |  |  |  |  |
| 易腐蚀性/可控性 |  |  |  |  |
| **常规** | 可访问/可侦测/可识别/透明/可截取 |  |  |  |  |
| 难以控制或管理 |  |  |  |  |
| 不自知和不可预测性 |  |  |  |  |
| 可预知性 |  |  |  |  |

**表4.1 脆弱属性和系统对象类型属性**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **脆弱对象** | | | |
|  |  | 物理 | 计算机 | 人与社会 | 基础设施 |
|  | 属性 | 硬件（数据存储、输入/输出、客户端、服务器）、网络、通信、位置 | 软件、数据、信息、知识 | 员工、任务、管理、制度、规程、培训、认证 | 船、建筑物、电力、水、空气、环境 |
| **结构/设计** | 单一性 |  |  |  |  |
| —唯一性 |  |  |  |  |
| —集中性 | 集中网络操作中心 |  |  |  |
| —一致性 |  | 标准化软件 |  |  |
| 分散性 |  |  |  |  |
| 逻辑/现实错误、不可靠性 |  | 路由和桌面应用软件漏洞 |  |  |
| 设计敏感性、脆弱/限制/有限性 | 电子环境容错 |  |  |  |
| 不可恢复性 |  |  |  |  |
| **行为** | 行为敏感性/脆弱性 |  | 无法抵御分布式拒绝服务攻击 |  |  |
| 恶意行为 |  |  | 内部威胁 |  |
| 不可改变性 |  |  |  |  |
| 可塑性 |  |  |  |  |
| 欺骗性 |  | IP欺骗 |  |  |
| 自满情绪 |  |  |  |  |
| 易腐蚀性/可控性 |  |  |  |  |
| **常规** | 可访问/可侦测/可识别/透明/可截取 |  |  |  |  |
| 难以控制或管理 |  |  |  |  |
| 不自知和不可预测性 |  | 无法探测到IP网络改变，可能使IP掩藏 |  |  |
| 可预知性 | 常规商业硬件为已知的 | 常规商业软件为已知的 |  |  |

**表4.2 完成的脆弱性用例实例**

### 无法探测到IP网络改变，可能使IP掩藏

**脆弱属性：**不自知和不可预测性

**目标类型：**计算机与网络

**描述：**如果一个IP网络没有一个主动的监视程序和工具来允许人员确定一个新的主机（IP）从网络中移入或移出，那么可能造成一些人将没有被授权的手提电脑或者其他设备接入网络，并且下载信息到那上面。尤其是在无线网络中，这中影响更严重，在这里，“连接”可以建立在本地到外面的可用的网络端口之间，甚至是组织的建筑物以外。这是一种网络配置“不可自知”的结果，需要在这个操作中改变这种情况。

### 集中网络操作中心

**脆弱属性：**集中性

**目标类型：**物理

**描述：**网络操作中心的中央位置可能包括很多重要的组件（例如关键设备及其备份）。如此一来，某种物理层面的攻击就不单单会破坏主要的路由器和关键通信设备，还会导致他们的备份系统不可用。

### 常规的商业软件和硬件是已知的或者可预知的

**脆弱属性：**可预知性

**目标类型：**物理、计算机和网络

**描述：**个人电脑、工作站、路由器、服务器和其他关键信息系统的组件通常严重依赖于商业产品，例如思科路由软件、Windows NT、Microsoft Outlook、Word、Excel、PowerPoint等。这样一来，脆弱性、组织机构，以及某些情况下软件类型的源代码都是公开的。这样的软件就会存在风险：他们的其他副本存在很大的惯性，他们可以用来测试某种会导致性能故障的情况（例如超出数据库的容量）。

### 标准化软件

**脆弱属性：**一致性

**目标类型：**计算机网络

**描述：**在路由器（例如思科操作系统）、服务器（WinNT）、以及PC和工作站（WinNT和Macintosh OS）上使用标准化软件会产生同型信息和通信系统。任何一个设计上的缺陷都可以在信息系统的范围内广泛复现，同时造成一类普遍的脆弱性。

### 路由和桌面应用软件漏洞

**脆弱属性：**逻辑/现实错误、不可靠性

**目标类型：**计算机网络

**描述：**有许多基本功能设计和实现的缺陷存在于在工作站、路由器或者桌面应用的操作系统上运行的标准软件中。这些缺陷一旦被攻击者获知，那就有可能导致非授权的访问进入或者某种破坏。

### 电子环境容错

**脆弱属性：**设计敏感性、脆弱/限制/有限性

**目标类型：**物理

**描述：**各种各样的商业电子设备对网络通信和计算起着重要的作用，这些设备对于环境影响（例如温度、烟雾、湿度）以及极端的攻击方式（例如电磁脉冲）并没有顽强的抵抗力。

## 对于脆弱属性的描述

下面有一些能导致脆弱性的属性，并对它们做了一些简短的描述。Anderson等人对这些属性进行了额外的距离和讨论。

要注意一点，一些脆弱性可能会一次展示出多个属性，这经常会导致一些列攻击和故障，来达成攻击者的终极目标，或者致使某种意外的系统故障。

### 设计和结构属性

一个系统对象的设计和结构属性可以明确导致脆弱性的结构化特征。这些属性在下面提到的宽范围的分类里是一个组合：

#### **1 单一性**

单一性是一个重要的、宽范围的分类，它可以提供重要目标或者单一脆弱点，这会产生很深远的影响。单一性包括唯一性、集中性和一致性。

* 唯一性。唯一性是指可用性角度的单一性，也就是说一个对象在它的种类中可能是独一无二的，而其他的东西很难替代这个对象。具有唯一性的对象类型可能非常稀少，而且可能已经经过了彻底的测试和优化。例如包括每种只有一个同时不再生产的产品，也可能是掌握特殊知识或者经验的人，而这些都很难转移到其他人身上。
* 集中性。集中性是单一性在位置上的描述，表明脆弱点在一个单一地方相对集中。例如包括决策、数据或者是通过中心节点或进程控制通过行为。
* 一致性。一致性是单一性针对类型而言的结果。通过反应，多个、相同的对象出现共同的故障或漏洞。使用对象的单一类型可以找到共同的目标，如果没有被损坏，这些目标会影响所有它支持的系统功能。

把这三类归纳到“单一性”下面是体现了这些属性以不同的方式体现了单一性。例如，一个单一的脆弱点可以预期是难以替换的（唯一性），单独的区域中的关键节点汇聚到一起（集中性），或者受损的位置广泛分布在系统中——一旦一个常规对象中的弱点暴露出来。

#### **2 分散性**

分散性意味着对象可以轻易地从静止的对象中被分离。可分离的对象可以被分头攻击行为逐个攻克，这些位置的保护信息（例如安全升级补丁）、加固设施或者已修复的部分会被阻断或延时。例如包括可以被分为两个互不联通的子网的网络。

#### 3 逻辑/现实错误、不可靠性

系统对象的逻辑、执行、结构中的错误会直接导致访问权限、可被攻击的目标和“不可归因性”暴露给攻击者。这次错误可以影响系统的可靠性、可用性、易懂性、可维护性以及其他重要的方面。错误和不可靠性可以起源于弱点出现在系统需求中，或者，在更多的基本故障中，存在于需求本身当中。它也可以起源于不充分的认证、确认和鉴定（VV&A）；不足的检测或评估；缺乏严格的系统工程；或来自于技术和科学的落后与不成熟。

#### 4 设计敏感性、脆弱/限制/有限性

相对故障和错误而言，这些属性起因于所有系统的天然局限性。在现实的环境中没有一个系统可以设计得天衣无缝。例子包括环境暴露的脆弱性、变异的输入、不规则的使用以及超载。合理处理这些错误可以减少这样的限制，但是如果没有处理这些错误，脆弱性也会紧跟着出现。

#### 5 不可恢复性

拥有不可替代的组件或信息的对象，如同这些需要过多时间（相对于功能要求而言）和努力（相对于可利用的资源而言）从装张状态中恢复或重置的内容，可以看做一个吸引人的目标——如果它们有关键的功能。例如系统相对于操作响应时间的较长的重启时间，或者系统丢失了关键的状态信息。

### 行为属性。

除它的结构特点以外，对象的行为也可以暴露其易被利用的特征。这里是一些主要的行为特征，它们都可以大致这些脆弱性。

#### 1 行为敏感性/脆弱性

这些属性包含对象如何做出行为或反应，以及系统在改变输入或环境条件的时候其本身的强健程度 。例子包括动作的行为、功能和操作敏感性、配置、设置、输入等。

#### 2 恶意行为

对抗大范围的信息系统及其安全的系统或人可以直接毁坏系统功能，或者被外部的实体利用以加剧它们的恶意行为。

#### 3 不可改变性

不可改变性，或者说缺乏适应性，包含配置、行为或响应在攻击中不易被更改。同样的，缺乏预先计划的程序（例如应急计划或MOUs[[14]](#footnote-14)）会限制对象动作的有用程度，这会使它更加趋于功能上的无效或低效。

#### 4 可塑性

对象可以被轻易修改、操作、改变、插入或者删除对外部或内部的风险造成潜在的脆弱性。

#### 5 欺骗性

具有这些属性的对象可以轻易被欺骗。例子包括可招募的内部人员、不具备操作不明确数据的能力、不充分可信的模型、不能发现自身的偏见以及何时会导致其被欺骗、否认和不充分的认证、被骗做出不合适的响应（例如被设置成为安全状态，这些状态被给予过高或过低的威胁，最终分别导致不足的操作或不足的保护）。

#### 6 自满情绪

缺乏安全意识（例如稀缺的管理程序或不足的监控）和响应能力会导致薄弱的安全状态和应对威胁的软弱表现。

#### 7 易腐蚀性/可控性

这些属性暗含着一个弱点，那就是它们可以被利用来制造一个错误的对象行为或者变成一个恶意代理。例子包括可以被操控或者收买的人员带来的内部威胁、输入、输出、可被改变的记录，以及在单位组件只是缺失的情况下对系统或组织进行控制。

### 常规属性

这些属性可以同时穿插在对象的结构和行为两方面中。

#### 1 可访问/可侦测/可识别/透明/可截取

这些暴露在外的属性适用于结构、行为、适应性、数据等方面，以及将攻击中的关键步骤进行标准化。例如，离开了访问，没人可以攻击系统。

#### 2 难以控制或管理

配置中的困难、控制、维持对象或系统能使其难于查找、修复或预防故障；建立适当的安全保护和响应机制；约束系统或其组件的行为。

#### 3 不自知和不可预测性

正如信息对攻击者非常关键一样，对防御者而言，对自己有充分的了解也是极其重要的。他们需要知道谁和什么构成系统，她们如何相互连接和相互操作，以及系统怎样和何时被损坏。同样地，如果缺乏必要的能力来预测系统配置如何被更改和起作用，就会限制信息对响应问题和攻击的可用性（例如不能发现自己软件中的外来编码）。

#### 4 可预知性

被对手预知到对象的设计、结构、行为会允许他计划和构造远程攻击，以发现对象如何响应，同时把对象设置成他希望的状态或者弄出故障。

## 脆弱性如何与通常的威胁结合

下面的例子证明了脆弱性的形式如何能组合并引发重大的信息安全问题。

首先考虑DDoS攻击，它可以降低系统服务的性能。它通过分散的来源，多重发送看似合理的服务请求来实现。图4.2展示了DDoS利用流量和缺乏弹性的设计上的限制来重定向和阻塞进入的流量，同时通过多个分散的操作实体进行控制的系统通常存在管理上的困难，因为没有单独的权威管理者控制流量。

**结构/设计**

**行为**

**常规**

单一性

行为敏感性/脆

弱性

—唯一性

恶意行为

—集中性

**不可改变性**

**—一致性**

可塑性

难以控制或管理

分散性

欺骗性

逻辑/现实错误

、不可靠性

完备性

设计敏感性、脆

弱/限制/有限性

易腐蚀性可控性

可预知性

不可恢复性

可访问/可侦测/

可识别/透明/可

截取

**不自知**和不可预测性

DDoS攻击

**图 4.2 脆弱性对应DDoS攻击**

其次，考虑对防火墙的渗透导致大量非法的、不受信任的、非认证的访问和请求。图4.3展示了防火墙渗透对同一性有利的全球公认观点：处于市场支配地位和标准化的防火墙、路由器和其他网络组件会使一些漏洞利用起来非常容易。同样，当一个攻击者决定如何从一个常见的防火墙去想一个组织发起渗透的时候，他就能进入整个组织的系统。防火墙渗透依赖于可访问的脆弱性（例如出现在开启的网络上）、防火墙和网络设置的管理困难（例如难于对防火墙进行最初的配置，或者难于对防火墙进行二次配置，以阻止已知的攻击者）以及其不可自知的特点（例如当系统操作员不知道他们的系统是否已经被损坏，谁是渗透者，什么区域被损坏，甚至是他们的系统是如何被配置的所以他们可以调整来抵御更多的渗透）。

**结构/设计**

**行为**

**常规**

单一性

行为敏感性/脆

弱性

—唯一性

恶意行为

—集中性

不可改变性

**—一致性**

可塑性

**难以控制或管理**

分散性

**欺骗性**

逻辑/现实错误

、不可靠性

完备性

设计敏感性、脆

弱/限制/有限性

易腐蚀性**可控性**

可预知性

不可恢复性

**可访问**/可侦测/

可识别/透明/可

截取

**不自知**和不可预测性

防火墙渗透

**图 4.3 脆弱性对应防火墙渗透**

第三，考虑网络拓扑探测（例如使用网络扫描和探测工具），攻击者可以通过这种行为手机目标系统的信息并未下一步的攻击行为做准备。图4.4展示网络拓扑探测可以帮助发现大量的脆弱性。这种“一次购齐”式的收集信息方式，可以使得找到所有系统切入点更加容易。同一性意味着攻击者可以应用其信息跨越大量的系统或者甚至跨越整改组织。不可改变性使得网络配置保持不变，它会一直保持可使用状态，不管敌对方获得什么样的信息。易受骗行允许网络拓扑发现会采用某种欺骗来收集信息（这包括计算机网络方面的调查或者社会工程学）。进入系统会使调查、开源信息收集、社会工程变得容易。网络拓扑探测管理上的困难一直没有被认识到，这会降低防御方抵挡网络探测的能力，以及意识到一个系统何时成为信息收集的目标对象。

**结构/设计**

**行为**

**常规**

单一性

行为敏感性/脆

弱性

—唯一性

恶意行为

—**集中性**

**不可改变性**

**—一致性**

可塑性

**难以控制或管理**

分散性

**欺骗性**

逻辑/现实错误

、不可靠性

完备性

设计敏感性、脆

弱/限制/有限性

易腐蚀性可控性

可预知性

不可恢复性

**可访问**/可侦测/

可识别/透明/可

截取

**不自知**和不可预测性

网络拓扑探测

**图 4.4 脆弱性对应网络拓扑探测**

最后，考虑计算机系统内的特洛伊木马攻击。图4.5展示了特洛伊木马不仅利用容易轻信他人的特点（源自于特洛伊木马故事的最初的概念）也有其他类似的弱点。当过多的数据提交给一个容易被欺骗的软件，特洛伊木马就能进入系统，易受骗的用户会打开一个在有经验的人眼中值得怀疑的附件，或者易受骗的用户从非认证的网站上下载软件。同质性使得把问题聚焦在同一类目标的攻击行为和组织当中的系统的行为。可控性允许木马接管计算机，并且使用它们达成其他的漏洞攻击和利用的目标。不自知和不可预测性阻止用户探测木马本身，以及用户的计算机已被损害和控制的迹象和信号。管理一个系统的难度在某种程度上也意味着它可能难于夺回控制权和发现已经造成感染的木马。最后，可访问性允许木马将自己提交给系统或用户。

**结构/设计**

**行为**

**常规**

单一性

行为敏感性/脆

弱性

—唯一性

恶意行为

—集中性

不可改变性

**—一致性**

可塑性

**难以控制或管理**

分散性

**欺骗性**

逻辑/现实错误

、不可靠性

完备性

设计敏感性、脆

弱/限制/有限性

易腐蚀性**可控性**

可预知性

不可恢复性

**可访问**/可侦测/

可识别/透明/可

截取

**不自知**和不可预测性

特洛伊木马

**图 4.5 脆弱性对应木马攻击**

# 第五章 直接和间接的安全技术

这一章提供对一种信息系统安全技术的深入介绍，它可以帮助抵御脆弱性。这些技术按照他们使用的基本概念划分成组，他们的分类就是第四步中推荐的矩阵和筛选工具——基于脆弱属性的类型、用户角色以及问题中的攻击行为和故障的发展阶段。

本章最后介绍了一些已知的安全方法如何对应到一个或多个这样分类。

## 安全技术分类和举例

安全领域由大量的使用各类策略达到消除脆弱性的安全技术开辟和发展。一些技术可以使系统对象对攻击和故障事件实施反制，另一些则能积极识别和响应这类事件。此外，还有一些技术则可以在接触对象的过程中限制危险的攻击部分或者故障诱因。更进一步的技术可以在攻击者第一次尝试攻击的时候就将其阻止掉。图5.1罗列了与信息系统对象相关的主要安全技术。他们按照一些原则来分类，包括是否能够提升自身的韧性和强度，是否能够提升对攻击和故障的认知水平，是否能限制或阻止知识和信息被攻击者获得，以及是否能阻止和惩罚攻击者。这些技术中的大部分相互重叠和补充，而这个分类则向人们展示其本身的特征和性能。

### 韧性和强度

第一项安全技术的笼统分类涉及到使系统具备在遭遇攻击的时候更容易恢复和更健壮的性能。

**异构。**异构包括组成部分的类型、操作范围、制造商、专业技术、背景等；随机编译生成的多样性；多媒体；平行异构（例如平行的电子邮件同步程序）。

**冗余。**冗余包括具有相同系统功能的可选系统和/或方法。评估者可以考虑路径的多样性，双路或反向连接，数据库镜像同步，额外的容量和存储空间。

**集中。**集中意味着：信息集散中心、报告、报警、修复、为获得物理系统的常见操作图片而进行升级、质量控制、成本节约等；以及集中的本地管理（或者通过交互的虚拟集中），以提供一致性和协调性。



**图5.1 安全减灾技术的分类**

**分散。**评估者应该考虑分散的控制点，路由过程、备份、配置数据、恢复点、员工；分布式的、移动进程；轮换职责；不同地理位置的荣誉信息。

**验证和确认、软硬件工程、评估和测试。**在信息系统组成部分的严格的设计和工程实现所涉及的广泛领域，包括了质量信息系统产品；程序保证（例如性能成熟度模型[[15]](#footnote-15)）；个人和系统测试、培训、批准、认证；安全程序、检测用例、核查；安全建模、评估、测试；红队（常规安全、入侵物、间隙、访问）；联系（真实的或者模拟和沙盘的）。

**暴露度、访问和输出的控制。**对信息系统边界的控制是信息安全中最普通的注意对象。相关的技术包括密码学、加密技术、公钥基础设施（PKIs）；口令、同步伪随机数生成、生物测量方法、智能卡；防火墙、过滤器、行为限制；防御（入口和出口）；单一路径关口；后门清除；不可复制的媒体和信息；自保护包装；空气隙和离线系统和备份；分类和划分（基于特权、间隙、角色、容量或行为的内部访问）；数据、代码、进程分割；打包可信组件（保护）；打包、隔离、分离不可信组件（控制行为和容纳任意的破坏事件）；I/O检查（错误检查、紧密的类型和范围检查等）；物理安全措施（例如电磁屏蔽、围栏、安全防护、离围栏合适的距离、上锁、正或负气压等）；使用元数据对分类、识别、理论功能进行支撑；传输保密（加密）；完整性认证。

**受信学习和执行系统。**信任是允许访问、梳理数据、使用其他系统组件（例如软件和文件）的基础。特殊方法包括记录学习过的课程，使用一体化（或理解为一致性，译者注）技术，可信和经验性的调查、收集他人分享的经验、使用可信的第三方验证信息和功能组件。

**抗抵赖。**预防抵赖（以及它的早期归属的策划）的技术包括接收和占有的证据；鉴定；PKI；对所有访问的记录、读/写、数据源（登录退出日志、视频监控、访问日志、元数据结构等）。

**加固。**加固保护对象以抵御穿透防御“工事”的攻击行为是最终而重要的守卫。具体的方法包括电子加固、错误纠正代码和软件；坚实的组织和流程架构（甚至只有一个功能组件的子集，提供最小的功能）；电磁脉冲（EMP）、环境的、震动、防浪涌设备；只读（写保护）数据存储、配置、网络等等；只读存储器（ROM）。

**错误、不确定性、有效性和质量保证、柔性降级。**类似于加固，可接受的、缓和的落差允许对象容纳错误、不确定事件、无效事件和较差的质量，这是通过调整和执行可以容纳问题而不导致故障的行为的措施达到的。具体的技术包括可分离性（允许从故障组件中隔离出来）；从设计和方法上预留容错空间；用最简单的方法来操作性能下降的设备（例如在制冷功能失效的空调上开一个风扇，在受保护的模块或磁盘上面仅包括最小拓展和制图能力的最小操作系统上使用CPU等）；有能力从不确定的、部分可靠的、低质量的数据中进行处理和推导；鉴定进入的信息，量化其可靠性或不确定性；可用性评估；来源查证；为数据质量提供元数据；不确定的推论或语义。

**静态资源分配。**按照预先定义的方式分配资源允许时间的高级计划编制、分析变化的后果、为提升系统的安全状况而查看其含义。具体的技术包括限制非重要连接、减少在脆弱单点上的载入、建立和执行依赖于已知的系统敏感性的指导方针（例如在Windows系统中，限制同时开启的应用程序数量；保持对不稳定的应用软件的使用降至最小值，尤其是在一个任务重尽可能将这个次数接近极限）。

**动态资源分配。**作为静态资源分配对应的部分，动态资源分配利用关于威胁和问题的信息以实时或接近实时的方式调节资源，通常这牵扯到复杂的和具有挑战性的响应。具体技术包括：提升衰减的量值（例如被降低的要求、生产量、热力、能量等）；依照主次排序客户或进程（例如基于市场和管理的优先级排序）；从上游更远的地方切断不希望的通信和分发；动态管理的顶位级别；自动或人工的、管理驱动的、基于事件的或搜索的（例如遗传算法或探索模型）动态网络重配置[[16]](#footnote-16)；在一个任务中尽可能接近极限地将不稳定的软硬件的使用次数降到最低；动态变化以确保（自身的某些特性对外界的）不可预知性和迷惑性。

**管理。**有效的管理可以提高安全性，而这是通过报告系统、结构和进程，质量控制，确保默认设置符合安全需求，同侪压力，信息传递和广告，培训，安全活动和提示，警告和威胁，策略提醒和激励因素，红队测试和评估进程和依从性。

**应急响应体系与计划。**信息条件（INFOCONS）和其他预先计划的静态或动态保护措施采用了增长中的信息系统保护措施的一个层面，它被带入到一个响应当中，而这个响应针对于期望的、被观察的攻击威胁水平。其他的方法包括数据和配置的保护和备份；简历备份服务器；基础设施备份；安全计划和谅解备忘录；危机计划和管理；清除和过滤；适应性攻击的适应性反应；资源再分配。

**紧急重建和恢复。**在故障之后恢复和重建的能力差不多和没有故障的影响一样重要——只要关系到系统性能需要的响应的时间足够快。具体技术包括数据保护和恢复；温重启（warm rebooting）；热、冷、温备份服务器；重启时的保留或替换通道；保留能胜任工作的人员（无论是直接的或者是通过谅解备忘录与附近的组织达成的协议）；给每一个网络节点一个重启动“genome”（预定义的指令设置）；基础设施备份和恢复方法；威胁响应计划（例如安全计划、预先协商的谅解备忘录、危险计划和管理）；动态资源分配（例如清除和过滤）；适应性攻击的适应性反应；人工操作和恢复计划；本地的可用复位部件（可能在不同的区域，以提供分散化的目标）；快速复位计划；本地修复功能；检测硬件、软件和数据中最难再生的那一类（因此它们需要被储藏或者设施冗余）。

**适应性和学习。**对手会对目标持续的学习和适应系统的防御变化。因此，一项防御措施必须适应这些变化中的威胁或者管理道高一尺魔高一丈的风险，因为一个富有创造力的攻击者可以轻松绕过防御方的“马其诺防线”[[17]](#footnote-17)。具体的技术包括为未知和适应性攻击建立合适的响应；总结经验教训，创建最佳的安全实践数据库（一个视角下的粗糙的“免疫系统”）；利用集中性提升在攻击中获得的知识的分布效能；采集攻击数据发掘敌对方的行为，开发保护方法和对抗措施；确保充分的监控和信息流跟踪；开发可用的（快速的）培训资料和临界数据资料，尤其是在针对候补人员的快速的关键操作和程序；建立动态的信息条件（INFOCONs）以及其他威胁响应体系和计划。

**免疫式防御系统。**正如从生物学中借鉴的概念，“免疫”系统集成了以下围绕系统和组织的对象：威胁识别、防灾开发、安装启用、宣传扩散。具体的技术包括自动（首选的）或手动的系统以发现威胁、传播警告、安装更新和补丁、颁布安全措施；自动商用现货（COTS）补丁和介绍文件（profile）；记忆、适应和交流（需要报告结构）；全球范围内分享攻击信息，拼接以确认发生的事件以及如何抵御；应用概念开发适应性和被分享的INFOCON流程。

**防疫。**另一个受生物学启发和得到的概念是预谋攻击（deliberate attack）（即“感染”），这种方式的的目的是训练、识别、感知（或者说探测）和为未来的攻击做准备（联合或者脱离正式的“免疫系统”）。通过红队去刺探系统便是这样的途径之一。

### ISR（情报获取、监视和侦查）和自校正

第二个总体的安全技术的分类涉及到关于威胁和某所有者所管理系统的信息收集——某种意义上说，这是“战场上的情报收集的准备。”

**情报活动。**情报收集工作囊括了关于针对敌对势力（目标、推动力、方法、性能等）和信息收集工作的全部范围，同时也包括内部的操作。理念上讲，情报获取工作同时覆盖你的以及你对手的信息系统，因为它们组成了“作战空间”。情报获取不仅仅能发现攻击行为，也能收集显示保护性和反制性的措施的高级信息。

**自校正、监视和评估。**了解你自己的系统，同时可以对其进行监控，以及评估自身条件，经常是关键的一步。这体现在识别和规避攻击和故障事件。具体的技术包括自我监控（针对内部或外部威胁）；安全审计（例如VAM理论和IVA）；红队测试收集信息；网络监控和管理工具；状态和性能监控器；系统状态和配置文档；静态建模和理解（understanding）；对应用程序和人员的行为进行监控和记录（数据访问和改动的、连接的、请求的、执行功能的、访问未遂的等）；在专业人员在集中位置进行评审和监控的背景下提升远程监控的性能。

**诱骗ISR。**欺骗是一种非常有用的（也是非常出乎意料的）ISR技术，通过自然的欺骗影响信息流。具体的技术包括使用蜜罐、间谍联络人、僵尸网、诱饵诱骗、伪装、结构或行为模仿等手段进行刺探或情报收集工作[[18]](#footnote-18)。

**检测攻击、损伤评估、敌我识别。**大量的方法可以用来发现、找出以及分析攻击，也包括评估这些攻击的影响范围。具体的技术包括实时入侵探测；学习系统（神经网络、自我组织映射等）；模式识别（基于案例的、基于规则的、基于模型的相关对象等）；内部系统行为和条件监控；通过欺骗技术进行探测和识别（例如哄骗、特高频噪声、蜜罐）；篡改和启封探测；跟踪和循迹；特殊监控特权；腐败行为识别；使用设计规范限制可接受的硬件、软件和员工行为；预防破坏的外围防护；抗抵赖机制（例如修改记录、收讫和所有凭据、鉴定、PKI）；访问日志、用以帮助分析的全球共享的情报数据。

### 反侦查，屏蔽ISR、目标搜索

第三类安全技术的常规分类涉及对于你的对手的反侦查与屏蔽ISR、目标搜索。这直接影响到你的对手在你的系统中收集对他有用的信息的能力，以此来完成一次攻击。

**常规反侦察。**一些基础的反信息系统侦察技术包括扫描物理监视器、漏洞等；扫描木马软件和监视器；进行安全检查；测谎。

**对敌的不可预知。**让你的系统难以被了解，可以阻止来自一些敌人猜测你方给予工业标准的配置和构成。具体的技术包括伪随机化和非常规话的配置、名称、位置、环境、职责等；极端异构化及分散化；移除文档；自组织集体行为；目标导向行为；特殊化、适应性、基于威胁或基于规则的活动；个体间交流；不可被外部（甚至是内部）预知的有利紧急行为；变化的操作流程（硬件、软件、人员）。

**反侦察欺骗。**正如对于而言ISR一样，欺骗在反情报中也是一种有用的（同样也是出乎意料的）技术，它可以截断去往敌对方的信息流。反侦察的欺骗技术包括隐藏一个条目和它独特的脆弱性；隐藏真实并且暴露虚假的结构、设计和计划信息；误导或使对手混淆。

**屏蔽ISR和目标搜索。**直接拒绝技术包括移动、遮蔽或屏蔽、访问过滤和干扰。

### 威慑和惩罚。

最后一种安全技术的分类涉及到震慑对手，以此来减少他们首次攻击你的系统的意愿。

**威慑。**信息系统的威慑技术有很多，他们包括可靠的威胁；力量的展示；警告、同行施压、心理战（PsyOps）、篡改和损坏显示器件（例如胶带、标签、指示器）；对安装启用障碍物实施适当的管理。

**预防性和报复性的信息和军事活动。**攻击信息活动（Offensive IO[[19]](#footnote-19)）和军事活动可以用来预防和报复来自敌对势力的攻击行为。操作方向包括信息散布、心理战、电子战、物理攻击、信息攻击。

**刑罚与司法保证。**可以使用的技术包括联合；担保；国际协议和合约；利用不可抵赖的数据；攻击和损害的罚款（包括针对内部人员）。

**司法执行和民事诉讼。**最后，法律的强制力也是非常重要的；否则其他的威慑将是空中楼阁。强制的方向包括国际、国内、州或省、本地的当局或法院，利用不可抵赖的数据；合适的后续动作和管理行为。

## 安全技术如何和常规的安全方法组合在一起

以下的例子表明了罗列在下面的基本的减灾技术是如何跟安全方法关联到一起的。

首先，考虑信息条件（INFOCONs）。图5.2描述了信息条件概念是一个运行高级分析和安排的威胁响应（或准备）计划。但是，有效使用信息条件也依赖于监控和评估某个自由系统的的能力。而这个评估的目的，就是理解什么样的威胁才是现实存在的，以及确保给现实的威胁定下的信息条件等级既不太高也不太低。监控和评估方向对于已知的关系非常重要，这种关系意味着信息条件等级可能被定的太高，从而降低系统性能以满足提升安全性的需求。



**信息条件（INFOCONs）**

**图5.2 支持信息条件（INFOCONs）的安全技术**

其次，考虑提供攻击情报的“标志和警告”（I&W）系统。图5.3展示了I&W原来一个完整的ISR集群和自校正技术。信息操作的I&W当前状态主要依赖防御系统内的监控和探测技术（例如入侵检测系统，网络监控、探测技术），更胜于依赖常规的国际互联网中或者敌对方的组织和计算机系统内部的情报操作。

第三，考虑计算机紧急响应小组（consider Computer Emergency Response Teams,CERTs）和其他相关的中心，在遭到攻击的时候，他们可以协同计算机安全、管理脆弱性和进行威胁分析，提供咨询，组织和计划安全响应，以及执行响应（无论是已计划的还是临时的）。[[20]](#footnote-20)图5.4显示CERTs应用集中化来协调监控、安全程序和其他信息、安全响应、管理和交流，以此应对攻击。



**标志和警告（I&W）**

**图5.3 支持标志和警告的安全技术**



**计算机紧急响应小组（CERTs）**

**图5.4 支持计算机紧急响应小组（CERTs）的安全技术**

第四，考虑通过预定义的规则（和多次调试）过滤信息和进入本地网的服务请求的防火墙。图5.5显示防火墙直接采用了控制暴露、访问和信息输出的最原始的方法，但行之有效的防火墙维护要依赖于当前的情报获取工作、威胁评估以及对所有系统发生了何种情况的信息掌握程度。

**防火墙**



**图5.5 防火墙使用的安全技术**

第五，考虑加密和PKI（公钥基础设施）。图5.6现实了他们了关键的技术，这意味着可以通过身份验证控制暴露、访问和输出，也可以在信息传输的过程中控制其暴露。

最后，考虑分立的和物理隔离网络。图5.7显示分立和物理隔离的网络是另一种对暴露度、访问和输出进行控制的安全技术。最关键的信息系统往往使用这些方法，因为电磁过滤、防火墙和加密方案依然能在足够强度的努力下被克服。物理隔离提升了安全等级，所以就意味着敌对方必须采取其他办法（例如想办法进行物理访问、利用内部人员或者人们常说的“chipping”方法使物理设备穿插或修改，以使得未来的访问或破坏进行得更容易。



**加密和PKI**

**图5.6 合并加密和PKI的安全技术**



**分立和物理隔离网络**

**图5.7 分立和物理隔离网络**

# 第六章 脆弱性的通用安全技术选择

这一章讲述VAM理论的第4步中如何将第四章所提出的脆弱性对应到第五章里罗列的安全技术上去，这种映射可以为如何定位已识别的脆弱性提供特殊的指导。下一步，这一章讲述了过滤技术，它可以提升矩阵中识别出来的安全技术对特定的用户类型和攻击阶段的适应性。第五章和第六章描述了理论的第4步，而且支持安全技术（第5步）的选择。最后，这一章提供了一些特别的例子，这些例子描绘了由应用VAM理论的实际评估者为特殊或者一般的信息系统的脆弱性，而识别的多种特殊的安全措施。

## 脆弱性到安全技术的映射

一旦不论针对已知还是未知的脆弱性识别的“经常性挑战（often-challenging）”任务完成了，评估者必须识别究竟第五章中提到的哪些安全技术在评估过程中对应到第四章中提到的脆弱性。VAM理论引导评估者通过矩阵进行目的明确的识别——这个矩阵将安全技术管理到每一个脆弱属性上面——而不是让没有被引导的普通人凭借直觉，或者盲目的头脑风暴来完成这个任务。

### 映射到脆弱性的安全技术

表6.1展示了理论中关联脆弱性的大矩阵属性（见第四章）沿着左列围绕最上部的行排列到安全技术（看第五章）栏里。独立的脆弱属性和安全技术之间的关系被表示为数值（见图6.1）。这些数值由经验和脆弱性大类与安全技术之间的逻辑关系的平衡来决定。每一个复制后面的原因都被写在附录里。

当一种安全技术对应一种脆弱性存在规避其的可能，矩阵则在交叉点处填写2或1的数字。a2表明安全技术是一种首要的减灾备选项，而a1表明安全技术是次要的检查备选项（可回阅图3.4）。因此，当对脆弱性进行识别的时候，他可以看相应的一行，看哪种安全技术对应该脆弱性是首选的，哪种是第二选择的，这通过观察在脆弱性一行中对应的列的交叉处其值是2或者1就能确定。

例如，在放大的图6.1中，评估者在一个“单一性脆弱性”上面首要考虑的是“异构”“冗余”和“分散”；“验证确认、软硬件工程、评估和测试”来帮助减少单一性[[21]](#footnote-21)。集中性则可能是第二考虑的选项——当评估者查看过所以首要的备选项的时候。



**表6.1 脆弱性对应安全技术矩阵**

安全技术可以：

2减轻脆弱性（主要的）

1减轻脆弱性（次要的）

0 利用脆弱性

-1脆弱性（主要的）

-2脆弱性（次要的）



**图6.1 脆弱性与安全技术的关联值**

### 会引发脆弱性的安全技术

有趣的是，当安全技术被执行的时候，他们也可以引发新的脆弱性。这些案例在矩阵中被注明，在交叉点使用了负数-2和-1。a-2指出在安全技术会经常引发脆弱性时的首要警告，而a-1指明次要警告。因此，当考虑任意只用安全技术的时候，他应该观测整个技术列，并查看什么脆弱性警告对于安全技术来说是首要或者次要的。整个识别过程可以忽略安全技术的驱动因素，也可以帮助评估人员在查找隐藏的脆弱性的时候审计存在的安全程序。

例如，在放大的图6.1中，一个评估者考虑应用集中性的影响（或者寻找因为存在的集中性而可能被列出的脆弱性）的过程被给出了（除了别的之外）一个重要警告（-2），即集中的问题可能被引入。[[22]](#footnote-22)评估者也被给出一次次要警告（-1），即一致性可能被引入，因为集中性的影响经常牵扯到对设备、软件、人与社会结构和基础设施的标准化的信任。

### 脆弱属性可以多次推动安全技术

最后，在构造矩阵的时候，我们注意到这样一些例子：“脆弱性”也可能对于安全技术存在有益的影响。这些例子在脆弱属性和安全技术的交叉点处被标记为数字0。

一个显而易见的例子是集中性促进集中化管理，因为这个概念既能从潜在的脆弱性看出来，也能从定位问题的技术上面看出来。一个不太明显的例子是，当一致性可以通过提供统一的系统组件配置来促进静态和动态的资源分配，它们可以更加轻而易举的交换和被授予职责。一致性同时可以促进紧急重建和恢复，因为可交换的部件、通用备件、减少的后勤维护都运行快速复原。在最后的例子当中，可预知性可以被诱骗ISR技术利用以观察敌对方如何针对可预测的情况作出反应，抓住他们使用的工具以及其狡诈行为的线索。

这个矩阵没有识别有促进性的安全技术之间的关系，但它们是客观存在的。回头再看第五章关于安全概念的例子（例如INFOCONs、I&W、CERTs、防火墙），这些概念依赖于不同的安全技术（分别是图5.2、5.3、5.4和5.5）组合起来的影响。

### 冲击平衡

安全技术规避脆弱性以及其引发脆弱性两者之间的相互影响，证明了安全世界的焦点相互矛盾的本性。很多好的事物在安全的世界中就变成了一种有破坏性的事物。有一些通常的平衡一定被冲击了，当

* 在系统的功能模块和安全的对比中权衡投入的时候
* 在应用安全技术的时候的度的掌握
* 在安全中相互矛盾的目标和特征之间
* 在安全措施的益处以及为了最小化或预防额外的脆弱性而使用安全技术所带来的额外开销之间。

举个例子，在放大的图6.1中，一个评估人员试图处理一个单一性问题，应该考虑分散性（在其他事物之中），但是分散性可能带来分散的问题（首要关注的），还有逻辑/实现错误和设计敏感性/脆弱性问题（次要考虑）。评估者需要权衡单一性的风险的开销与分散性的风险开做出决定。那么分散性可以被实施来对应特殊类型的单一性吗？分散性可以被用一种这样的方式实施，来最小化或者预防逻辑和执行错误以及设计敏感性和脆弱性吗？在许多情况当中，对这些警告的意识可以为其设计以及特殊的减灾方法采用的措施提供信息，但是它们应该准确的考虑，这样才能平衡信息系统的全部风险。

### 设计和用法的考量

这些关系没有具体说明系统对象处理脆弱性的类型、对象措施的细节和系统的安全概况。因此，研究中的系统的细节信息和安全选项的调适必须补充矩阵中被反映出来的常识。结果，这个矩阵形成一个瞄准评估者的指引——通过大量的选项空间，而不仅仅是预先留下的惯例。在特殊情况下，脆弱性也可以得益于使用矩阵中不那么有价值的安全技术，所以，有时候可能想要超出矩阵中所明确的安全技术的范围。而缘起于安全研究的新的分类将需要在以后补充在矩阵当中。

## 提炼安全建议

对于每一个脆弱属性来说，表6.1所展示的方法矩阵定义了一个相当规模的有主有次的安全技术，而这些安全技术需要考虑它们潜在的关联。随着脆弱性数量的增长，设置了一个看起来很难被管理的建议。尽管VAM理论比那些没有提出建议的方法有了进步来帮助评估者通过安全问题进行推理，但是仍需要更多帮助来提炼选项程序。同样，很难躲安全建议可能是适当的，除了具体的评估者在使用理论的时候超出了应有的范围和权限——当将这个比较粗糙的矩阵进一步深化的时候。因此，将评估者注意力的焦点置于最相关的安全技术，那么如下的过滤方法就已经被开发出来了，而这些方法就是基于评估者从事安全评估时的职责角色，以及信息系统攻击行为在支持阶段的差异——从主要的攻击（或故障）行为中分离出来的。

### 评估者的工作角色

筛选安全建议的第一条技术利用了一个事实：一个评估者扮演了多个不同的角色；安全建议可以被筛选或者排除，因为受到评估者权力和责任的制约，有些建议可能没什么用。理论目前设置了三种评估人员角色的分类：操作、研发、策略。操作评估者包括系统用户、管理员以及对安全负责或者关注系统安全运行的其他管理者。研发评估者包括研究、开发、测试，还有负责建造和配置信息系统，而不对其进行运维操作的系统工程师。策略评估者说明全部的系统需求、要求、操作程序——通常是在系统大规模使用的环境中。将评估人员的类型罗列出来的列表可以在日后进行扩充或者自定义，但是这些树状类型对于数据而言则一直是有用的。

在表6.2和6.3中的赋值列的第一个树定义了哪些安全技术或强或弱地依赖于这些树状评估者的工作角色。强关联的安全技术被赋值以2，而弱关联的技术被赋值以1。例如，表6.2显示“抗抵赖”“管理”和“应急响应体系与计划”强依赖于（赋值为2）操作评估人员（第一赋值列），而操作的组织和个人可以监控用户和访问、建立和使用管理工具来提升安全性，建立流程和协议以应对威胁和故障。对暴露量、访问和输出的控制属于弱关联，因为执行操作的组织和个人可以执行和控制物理和网络层面的访问，但是他们会被现实，这源于其本身的设计，以及软件和流程的设计和实现，还包括其他人对策略的实施。所以，举例来说，对于一个操作用户，主矩阵（表6.1）向他建议了三种可能的技术——（i）异构性，（ii）抗抵赖，（iii）对暴露、访问和输出的控制，这些技术可能帮助减少脆弱性，而第一应当被考虑的应该是抗抵赖，因为它在表6.2中显示为强关联。对暴露、访问和输出的控制是第二应该被考虑想的，因为在表中它是弱关联。第三个矩阵建议（异构性）应当最后被考虑，因为在表6.2中它没有被赋值。

在安全选择当中，开发者似乎是最具有促进性的（例如在赋值列中含有最多“2”），因为他们设计了架构、基本面貌，以及在通常而言比较宽泛的策略约束下，基础设施的依赖性和相关性。然而，开发者不能指示操作用户应如何准确地使用和管理他们的信息系统。因此，每一个工作的类型都含有他自身的责任、权限和这样的促进性的范围。



**2**  强相关

**1**  弱相关

**表6.2 韧性和强度相关技术对于评估者工作角色和攻击构成要素的关系**



**2**  强相关

**1**  弱相关

**表6.3 ISR、CI、威慑技术对于评估者工作角色和攻击构成要素的关系**

### 攻击模块

第二项用来筛选安全建议的技术利用一个事实，那就是当一个故障可能只有一个源脆弱性，对于系统的一次攻击牵扯到有差别的组成部分，而安全技术可以利用于其中。“知识”、“访问”、“目标脆弱性”是任何成功攻击的需求。完全预防这三个组成部分中的任何一个，都可以抵挡一次成功的攻击，围绕这些组成部分的防御措施可以最小化整体风险。其他两项重要的攻击模块——“逃避惩罚”和“评估”一次攻击是否成功的能力——而没有达到攻击成功的临界点对许多攻击者而言是非常重要的，当这些组成部分被阻止，那么攻击就被预防了。

**知识**包括获取和理解目标系统的信息，这其中又包括一般的配置信息、安全状况和防御流程、获得访问系统的方法，目标脆弱性被利用的知识、防御提示和报警系统的知识、防御者定义一次攻击的流程、支持评估攻击行为的信息。

**访问**正在攻击中的系统需要获取知识，对目标执行真实的攻击，评估攻击是否成功。访问可以通过每一种对象达成（物理、计算机网络、人与社会、基础设施），包括：物理接入和接近（例如访问受限域或者电磁访问）；计算机、交流、控制网络；通过内部代理；活跃的基础设施系统。

**目标**脆弱性或者脆弱性在攻击中被利用是由于设计缺陷和行为的敏感性可以被攻击者利用。因为脆弱性在自然发生或意外的状况中产生，目标脆弱性的分类是唯一重要的基准。

**逃避惩罚**，当每一次攻击没有达到成功的临界的时候，对于一些攻击者是很重要的，比如民族主义国家的攻击者，他们不希望自己的信息攻击行为被知道，也有一些阻止，他们担心他们自己的脆弱性遭到报复。

最后，将信息攻击作为最主要业务的组成部分，这样的复杂的组织需要有能力评估其攻击的效果（例如当在不知道攻击是否成功时其他操作不能继续的时候）。

表6.4罗列了一个攻击者完成上述的每一个攻击模块的主要方法（处理目标脆弱性本身，它常常是一个系统本身的性能，而它并不在攻击者视野之下）。这些方法围绕着四个主要的系统对象（物理、计算机网络、人与社会、基础设施）分散开来。表6.5定义了那些脆弱属性在五个攻击模块中可以被利用。

### 通过评估者工作角色的攻击阶段的关联

总体来看，评估者角色和攻击阶段的筛选产生了如下影响，它帮助提炼了安全建议。这些筛选关注在于攻击模块，这其中评估者有更充分的能力来执行保护措施和对策。所以，操作用户通常对系统有比其自身的结构和实现方式更大的控制、知识和访问余地。开发者可以适应硬件和软件以最小化结构和实现方式上的脆弱点，但是少了在使用、知识和访问上的影响。最终，策略的制定者设置了设计和操作中总体的指引和限制，但是没有具体的现实行为去执行那些细节。



**2**  强相关

**1**  弱相关

**表6.5 攻击模块对脆弱性的利用**



**表6.4 完成攻击的每一个组成部分的方法**

## 使用VAM产生的安全选择举例

下述内容现实了这种类型的安全选择可以在产生于营的组织中的评估者——当这些评估者在使用这个方法作为一个分析指引来丁文安全焦点的时候。这些例子涉及到出现在第四章的检测用例矩阵中的常见的安全焦点（见表4.3）。这些点围绕信息系统的计算机网络、物理、人与社会对象，依据不同的脆弱属性赋以数值。例子中的分析较难理解，但是他描述了VAM理论的使用方法，在信息安全领域众所周知的问题和从分析中得出的具体安全策略的类型中，这个理论的功能得以施展。

这些一般的例子不包括依赖于实际脆弱性的具体情况，或者更具体的安全技术例子，即可以定位脆弱性的唯一特征的安全技术。然而，他们帮助证明了VAM理论如何将评估者指引到一个安全技术上面，以及某种被考虑的安全技术的实例化类型。

对于每一个例子，我们都在针对脆弱属性和对象类型后面做出了具体而简短的脆弱性描述。我们将矩阵推荐的安全技术分类标记为一个数值，同时伴随着的是分好类的详细的减灾策略，而这个分类对于某个问题中独有的脆弱性而言也是合适的。这些具体的减灾策略既可以产生于第五章所描述的安全技术的全面列表这个例子，也可以产生于我们重新考虑安全技术分类时一些新生的应对措施。

### 内部威胁

脆弱属性：恶意行为

目标类型：人与社会

众所置信“内部威胁”（一个被信任的人被批准的访问要害信息系统所实施的恶意行为）对于信息系统的安全而言是一个极大的威胁。所谓“内部”可能是抱有怨恨的某个个人，或者地方通过敲诈、贿赂等类似行为指派的某个人选。

潜在的相关规避策略：

暴露度、访问和输出的控制。确保“内部”仅可以通过他人的工作职能上的协助完成网络或者物理域的访问。

抗抵赖。维持访问和不论任何人违反认证程序都可以被检举的许可审计日志。

常规管理。确保流程在起作用（例如安全活动和提示），以警示员工的危险内部威胁（包括他们本人不知情的人员补充）和其他针对要害信息系统的安全威胁。

自校正，监视和评估。（见“攻击探测……”下面。特别要考虑在局域网中使用入侵检测软件。）

诱骗ISR。在要害系统建立“蜜罐”（例如文件内容是伪造的，但是有吸引人的信息），访问这些蜜罐将产生警告说明，这将提示一个单位正在从事可疑行为，而且应该白记录。

检测攻击、损伤评估、敌我识别。考虑使用实时“入侵检测”软件来发现违反预先设定的规则的异常行为，或者现实统计上的异常。审查访问和审计日志以发现可以行为。

对敌的不可预知。限制系统配置信息、关键部件位置、关键系统的依赖性的透露——甚至是对于“可信”的员工。

威慑。使用司法处罚（见下面）对待侵犯者，以警示其他人。

刑法与司法保证。确保司法处罚对内部人员也一视同仁，在适当的时候使用，以此当作一种威慑。

司法执行和民事诉讼。运用司法强制力开惩罚违法行为，这是一种威慑。

### 无力控制DDoS攻击

脆弱属性：行为敏感性/脆弱

目标类型：计算机网络

一种最难进行的网络攻击就是DDoS攻击，数以千计的计算机用数据包或者服务请求攻击一个明确的网络路由或者其他部件。通常它们包含了错误的信息，需要额外的时间来处理。信息网络一定是特别配置和设计的，如果它们打算挫败（或者说达到挫败的程度）这种依赖网络行为特征和敏感性的攻击。

潜在的相关规避策略：

分散。考虑使用平行或备份服务器，这样可以在主服务器由于DDoS攻击响应能力的时候接管访问。

验证确认、软硬件工程、评估和测试。测试网络（例如通过“红队”和其他方法）强度来抵挡DDoS攻击。

暴露度、访问和输出的控制。在网络上可以安装DDoS流量过滤器的多个位置建立控制点。

错误、不确定性、有效性和质量保证、柔性降级。考虑运行网络在DDoS攻击下允许柔性降级的方法（例如通过减少所有其他应用的IP流量）。

动态资源配置。提供快速切断DDoS流量的方法，更进一步是网络（例如在你的网管到互联网服务提供商之间）。

自校正，监视和评估。在危险IP网段的每一级网关提供DDoS攻击的早期警告监控；已经预先计划的流程就绪，当这些监控发现DDoS攻击的时候，马上响应。

### IP欺骗

脆弱属性：欺骗性

目标类型：计算机网络

在一个数据包或者消息中“欺骗”一个IP地址，意味着用一个错误的地址替换掉正确的。这就意味着，难以确保一个信息包或者会话的真实发起者，因此就难于辨别多种隐藏其源地址的攻击形式。

潜在的相关规避策略：

加固和暴露度、访问和输出的控制。强化各类防火墙的预防措施和规则，例如不允许任意不可用源IP网段发出的数据包访问本地网络。

应急响应体系与计划。当系统中的任意主机（计算机）一旦决定由一些发送者使用不可用的IP地址，预先制定的响应计划就要可是警告其他主机阻止来自这个地址的数据内容。

适应和学习。防火墙、路由器和其他操作关键IP网络都是具有适应性的设备，它们对确认为是IP欺骗的攻击响应，可以通过网络快速启动。

防疫。（见前述的“应急响应体系与计划”。）一旦当伪造的IP地址被发现，其他网络中的主机或路由器就应立即对其“免疫”，犹如“免疫防御系统”的基本形式。

自校正，监视和评估。防火墙、路由器和其他更小的设备一定要对伪造的IP时时保持警惕，只有这样上述的补救性步骤才能实施。

### 不能发现IP网的变化，可能发生IP掩饰

脆弱属性：不自知和不可预测性

目标类型：计算机网络

如果一个IP网没有积极的监视程序和工具来允许全体人员确定一个新的主机（IP地址）是否被添加或者移除于网络，那么一些人就可能添加一台没有经过认证的便携计算机或者其他设备，接入到网络中并下载信息到该设备上。这种危险行为在无线网络中尤其常见，这种“连接”可以来自于一个看不见位置的网络端口，或者甚至是某个外部组织的建筑物。这是在网络配置的“自我意识”方面的一个弱点，可以通过重新配置改变它。

潜在的相关规避策略：

集中。通过有能力发现和现实任何网络配置变更的传感器和应用程序，建立中心的、实时的网络监控活动。

自校正，监视和评估。通过合理的监控工具和技术，网络应该意识到其自身的 任何配置的变更，同时在网络操作的显示或信号中突出强调这部分内容——当其一旦发生的时候。

### 集中的网络操作中心

脆弱属性：集中

目标类型：物理

网络操作中心可以在一个中心位置安置许多活动的物理部件（例如关键设备和备份）。这样，一次物理攻击可能同时使得主要和备份的路由器以及关键的通信设备瘫痪。

潜在的相关规避策略：

分散。考虑提供多重的支持中心。不要在主设备的同一个位置存放备份设备。在路由器可以安装的物理局域网线缆处提供多重访问接入点。

暴露度、访问和输出的控制。基于“必要使用”的原则限制物理访问。提供保护掩体或设施（物理的或电磁的）以帮助预防意外的或有意的变化和毁坏等。给仪表面板和/或掩体安装房拆除设施。保持备份设备处于离线状态，以在需要时提供保护。

加固。提供保护掩体或设施（物理的或电磁的）以帮助预防意外的或有意的变化和毁坏等。给仪表面板和/或掩体安装房拆除设施。保持备份设备处于离线状态，以在需要时提供保护。

应急响应体系与计划。为恢复任何集中化的部件制定预先计划流程。

紧急重建和恢复。总体上要计划如何人工重建活动的通信。开发意外事故的复位替换。检查本地修复系统的性能。在不同的地理位置保存备份设备和配置信息，这样就不太可能在同一次物理攻击中同时被毁坏。

### 众所周知的一般商用软硬件

脆弱属性：可预知性

目标类型：物理和计算机网络

个人电脑、工作站、路由器、服务器和其他关键信息系统的组成部分经常依赖于商业产品，例如思科路由软件、Windows NT、Microsoft Outlook、Word、Excel、PowerPoint等。正如这些，脆弱性、组织和某些情况下的程序源代码都是被广泛知晓的。这些程序都是很大程度上可被预知的，因为它们的其他拷贝可以被测试以发现问题。

潜在的相关规避策略：

异构。考虑使用多样化的COTS软件和硬件——例如，网景作为IE的附加，Netscape mail作为微软Outlook的附加。这样当蠕虫或病毒利用一个众所周知的缺陷的时候，不至于一下子感染所有的系统。

验证确认、软硬件工程、评估和测试。在很大程度上，测试重度使用商业软硬件，以在要害信息系统中发现脆弱性。使用“红队”方法做系统测试。使用重要的操作系统和应用的“开源”代码，可以检查并填补缺陷。（注意一点，很多开源社区的用户已经在搜集这样的缺陷，并使用成熟的开源代码来继承他人的劳动成果。）

管理。确保任何可用的补丁和修复在变得可用的时候立即测试并安装。

免疫式防御系统。建立快速分享关于标准商业系统脆弱性和配置的攻击行为信息的协议。

反侦察欺骗。提供欺骗性的文件（例如WIN文件类型在UNIX或Macintosh设备和软件）使得难以分辨使用的软件类型，尤其是通过自动的扫描程序。将系统文件放置在非正规的位置或将他们存储在不同的文件名下（例如不要存储UNIX 二进制文件在/bin文件夹中，不要存储系统文件在“C:WINNT”或者“C:ProgramFiles”目录中，为电子邮件的附件改变默认的文件夹名）。

对敌的不可预知。如果可能的话，删除关于软件类型的信息，无论在内部或是外部，只要系统是可访问的。

标准化软件

脆弱属性：计算机网络

路由器、服务器、PC机/工作站大量使用标准化软件（例如思科操作系统、Windows NT、Windows NT或Macintosh OS）产生了非常一致的信息和通信系统。任何一个存在于这些设计中的缺陷会广泛影响到信息系统，因此会形成整个系统的通病。

潜在的相关规避策略：

异构。考虑有意使用变化的软件（例如Linux、Macintosh OS、SUN Solaris）作为网络或桌面配置的一部分，这样如果任何病毒、蠕虫或者其他网络攻击“拿下”某一类的标准化系统（例如运行了Outlook的Windows NT），而其他系统可能就可以继续运行并且提供紧急服务，直到损害被控制、隔离或者移除。

验证确认、软硬件工程、评估和测试。就某种意义上而言，标准化的（一致的）系统组件被广泛贯穿于重要系统当中，使用额外的测试来确保尽量远离可被利用的缺陷。

加固。依赖标准化软件应该引发额外的措施来确保它对攻击而言是“被加固”的，例如，在一个“硬”（不易被改变）媒体中保存关键操作系统和应用软件的备份拷贝，例如CD-ROM或者DVD-R，用来恢复被攻击之后的系统。

错误、不确定性、有效性和质量保证、柔性降级。确保针对多种类型的攻击，标准软件可以做到“故障容错”和平滑衰减。例如，当一次攻击被发起的时候，软件可能要退出非关键性的应用软件，同时关闭多种防火墙。配置以协助挫败这次攻击。

### 路由器或桌面应用软件的弱点

脆弱属性：逻辑/现实错误、不可靠性

目标类型：计算机网络

应用在操作系统（工作站或路由器）和桌面软件应用上的标准软件的基础设计和工程实现上的缺陷可能是存在的。这些缺陷，如果一旦被攻击者发现，就将发生非授权的访问或者破坏。

潜在的相关规避策略：

异构。使用基于不同软件（例如电子邮件程序）和操作系统（例如UNIX作为Windows NT的补充）的多种服务器部署，尤其是对安全有更高标准的系统。

验证确认、软硬件工程、评估和测试。进行彻底的红队测试和COTs产品测试来揭示其原生的脆弱性；开发安全流程来规避脆弱性。确保适当的边界防火墙安装和维护，就像在本地一样（如果可行）。确保及时更新最新可用的补丁和修复包。

暴露度、访问和输出的控制。基于“必要使用”的原则限制物理访问。提供保护掩体或设施（物理的或电磁的）以帮助预防意外的或有意的变化和毁坏等。

综合管理。确保所有的流程都在避免“root”权限的访问。

免疫式防御系统。采取“免疫式”的防御措施，在这个系统中，识别出一次攻击或者故障，通知其他相似的系统来增加其防御一次类似的攻击或故障。

防疫。用一个中心系统自动化“防疫”全系统的各部分和计算机（通过补丁和修复包，一旦当其通过测试和判定可用后）。需要小心的是这个集中更新流程要被很好的保护以预防变成围绕信息系统拓展攻击的简单目标，另外补丁和修复包应该被很好地测试。

### 电子环境容错

脆弱属性：设计敏感性、脆弱/限制/有限性

目标类型：物理

各种各样的商业电子设备活跃在网络通信和计算中，而他们往往并没有因环境的影响（例如温度、烟尘、湿气）或极端的攻击手段而变得强硬。

潜在的相关规避策略：

异构。考虑使用不同层级的环境冗余，这样完整的性能就不会在某些极端的环境条件下丢失。

冗余。将备份设备存储在密封的容器中，可能包括电磁屏蔽测试。提供本地的、为单一设备提供冗余环境条件的、集中的设备室。

验证确认、软硬件工程、评估和测试。测试哪些关键电子设备可以操作，以此确定不同的环境的层级；尝试确定最小环境敏感性的设备。

暴露度、访问和输出的控制。安装正压力的空气调节设施以保持烟尘、时期或者其他环境威胁远离整个电子环境——尤其是十分特别的电子环境。为重要的设备和终端的某个子集(它们可以在不利的环境条件下提供最基础的性能)安装电磁屏蔽设施。

加固。（见前文关机“为重要的设备和终端的某个子集……安装电磁屏蔽……”和“将备份设备存储在密封的容器中……”的叙述。）

自校正，监视和评估。为所有不利的环境条件安装可以影响电子设备的传感器，尤其是单个或者很集中的设备。原先制定应急计划，当环境条件超过设备操作可以允许的范围。

# 第七章 自动化执行方法：电子表格工具

手工完成这个逐渐形成的巨大的矩阵、评估者筛选工具、攻击构成要素筛选工具，对一个评估者来说需要服务巨大的劳动，并可能彻底或小心地妨碍对VAM理论的使用。此外，通过评估者的教程查阅大量的脆弱性、安全技术和攻击方法的定义也可以让人感到畏惧。因此，一个原型计算化的工具被开发了出来，它的作用是辅助使用这个方法。这个工具采用了使用Visual Basic算法的Microsoft Excel电子表格，它不但用来执行信息查找，同时为基于评估者的输入的脆弱性风险进行简单评分。[[23]](#footnote-23)

有一点很重要，那就是认识到即使有了这个工具，综合的脆弱性评估也无法全部通过自动化的方式来实现。自动的网络和计算机扫描软件和方法可以识别特殊的和一直的脆弱性，比如后门、开放端口、确实的补丁包、吞吐量阀值、操作异常等诸如此类的问题。然而，自动化工具不能执行自顶向下的属性检查，这些属性指的是那些必然会被利用的问题或同时涉及到物理、计算机网络、人与社会、基础设施的配置和行为的问题。它们的精确度极大程度上依赖于它们的威胁和操作模型的宽泛程度、他们产生的输入以及他们观察到的输出。综合检查常常需要人们在信息系统及其操作中经常涉及到的深厚的知识和经验。我们的方法指向评估者，而自动化工具帮助评估者在使用这套方法的过程中处理庞大的信息量。

## 手动执行的原始步骤

方法的第一和第二步（识别关键信息功能和识别支撑这些功能的关键信息系统）可以通过评估者在评估和检查有问题的信息系统的时候用人工的办法执行。虽然它们本身会很复杂，但是这两步经常要求对它们有着系统化的调查和考虑，而这难以用一个小工具去设立和促进。很多手册，像OCTAVE（Alberts等人，1999,2001），都有很多表格和流程可以作为这些步骤的参考，Common Criteria（ISO 15408）具体说明了这些议题和考量点的范围，通过选择出来的过程可以定位这些议题和考量点。

## 表格指引和记录的脆弱性

从第三步开始，VAM工具（见表4.2）开始可以使用工作表。这个工作表应该为每一个检查中的信息系统（或大量子系统）和系统内部多样的结构水平而完善。通过为评估者提供完整的脆弱属性列表，以及罗列信息系统中四个不同的对象类型（物理、计算机网络、人与社会、基础设施）列表，这个表格促进了方法第三步（识别关键信息系统的脆弱性）的执行。使用这个表格有益于确保对围绕各种规模的脆弱性目标进行广泛检查，而不是对仅通过想象得到的或者由某些评价组织提出的脆弱性标准类型的简单记录。

同样要记住一点，当安全技术导致了脆弱性，对应到安全技术矩阵的脆弱性能够识别出警示。评估者可能会发现使用矩阵（手工方法使用表6.1或者下面说的Excel表格）很有用，评估会对找到什么已经出现的脆弱性是因为使用了安全技术而导致的。

## 风险评估和减灾选项表格

在进行上述三步工作之后，这个方法的复杂度会急剧增加。图7.1展示了脆弱性评估和减灾选项拓展表。通过基于评估者提供的方法和信息进行自动化查找和计算，降低了这个表的复杂度。

### 说明用户类型和脆弱性并进行分析

首先，评估者设置一些脆弱性的基本信息，如图7.2所示。评估者的工作角色在第一部分被表现出来，说明了那种评估者的角色被挑选出来应用于表格中进行分析。一个在自由的文本框被包含在第二部分中，它允许由用户在学习和了解之后描述适应性（例如在脆弱性列表中提及的复制和修饰的描述）。第三和第四部分在下拉菜单中具体说明了脆弱属性的类型，以及问题对象的类型。



**图7.1 VAM理论拓展表格工具**



评估者类型

主要脆弱性

属性

对象类型

**图7.2 用户类型和脆弱性并进行分析的说明**

### 为每一个攻击模块评估风险

第二，评估者需要为图个攻击模块评估系统脆弱性的风险——知识、访问、目标脆弱性、逃避惩罚、评估——通过完成图7.3中展示的第5到7步。无意的故障可以通过仅仅完成目标脆弱性行来评估，在这一行中，包括了能导致人们关心的脆弱性。

第五部分允许评估者检查一个敌对方可能达成的四个攻击构成要素（知识、访问、逃避惩罚、评估）的基本方法，而这四个组件支持先前在第三步中评估的目标脆弱性。这里有一个评估者可以选择的下拉菜单，里面是攻击构成要素，同时可以由评估者查看表6.4聊中基于对象类型（在第四部分中有详细说明）的方法。



获得风险分数

查找关于敌对方如何完成每一个攻击构成要素的信息

为每一个攻击构成要素评级和描述风险

**图7.3 为每一个攻击构成要素评估风险**

基于前面的检查步骤和第五部分，评估者在第六步中为五个攻击构成要素的的风险评级使用了下拉菜单。风险评级被设置为：可忽略、低、中和高。

基于这些评级，系统平台通过四个简单计算得出综合的风险评级和分数。风险评级给出简单的五个攻击模块的总体“可忽略、低、中、高”的值，同时分数会取在0到10之间。这个工具使用了四个简单的算法将风险评级综合到一起。

第一个算法，命名为“min 1st3”，使用了三个最关键的攻击模块（知识、访问、目标脆弱性）的评级的最小一个，这三个部分是任何信息操作/信息战的要素。这个算法借鉴了对基本组成部分“最弱链路”评级的思想，这几个部分与减灾策略联系最紧密。而减灾策略试图通过阻止敌对方形成关键攻击模块链条来阻止攻击行为（例如，当应对极简主义风格的攻击者，他们不关心“逃避惩罚”或者“评估”其攻击成功与否）。

第二个算法，命名为“min all”，同样借鉴了“最弱链路”的思想，但是使用了五个攻击模块的最简评级办法。因此，这个值最贴近敌对方同等关注全部五个模块时的情形（例如民族主义国家）以及关于安全技术阻止攻击者形成攻击模块链条的情形。

第三个算法，命名为“min（target，sum 1st 3）”，它不是计算关键的三个攻击模块的综合评级而是选择了目标脆弱性的评级，当然有一个条件，那就是后者的评级要低于前者，也就是三个模块的综合评级。这个算法是非常有用的，当评估者应对那些风格简练的攻击者（他们往往不关注与利用漏洞无关的事情）的时候，他们往往想要得到关键攻击模块的综合值，同时也需要明白目标脆弱性是一个要素（例如，如果有一个地方性的目标脆弱性，即使再大规模的知识获取或访问也不会提高目标最终的可攻击性）。

最后是第四个算法，命名为“min（target，sum all）”，将五个攻击模块组合在一起（例如针对于民族主义国家的攻击行为），也必须意识到必须存在一个脆弱点以发起攻击。

当然，同样可以开发其他的算法整合到评估者提供的风险评级中，但是这四个算法充当了尝试提供对问题中的脆弱性进行整体评价的合理的起点。对“min”函数的使用反映了每一个信息攻击模块的重点内容（有时反映在表达类似“信息操作是三条腿的椅子”的时候），以及作为结果的重要安全着眼点，甚至是在一个用户无法定位脆弱性的领域内（比如说计算机设计上的目标脆弱性），应用在其他领域的技术（比如说拒绝访问或知识获取）在保护系统的时候有重大的积极意义。“sum”函数在整合围绕攻击模块的减灾影响方面同样有效，尤其是当在一个关键区域完整的风险规避无法实现的时候。

分数整合算法还有一个用处，这需要了解对于不同类型的对手的不同类型的攻击权重和方法的实用性和可用性。考虑到评级的主观性，更加复杂的算法可能并不具有太大意义而且仅仅是使优先的风险更难理解。在确定的情况下，它可能是仅仅有益于对比规避前后的风险，而放弃了评分的原理。

工具的使用者将会注意到系统对评级设置了不同的颜色（分别是无色、黄色、橙色和红色）以提升表格中评级的略读性能，同时快速决定如何为一个脆弱性评级（尤其是对其他不同表格的脆弱性评级的比较中），决定规避策略在预期当中多么有影响，以及决定什么攻击模块会受到规避措施的关注。

### 考量和选择规避措施

第三，评估者使用这个工具检查和选择围绕攻击模块的风险规避策略。图7.4展示了图表自动执行矩阵查找的部分，同时为每一种由评估者的角色导出的攻击模块匹配安全技术与脆弱性之间的联系。

记录被选的规避措施



要考虑使用规避技术带来的警告

查看规避措施描述的例子

查看相关的规避方法

**图7.4 考虑和选择规避技术**

第八部分允许评估者查看为每一个攻击构成要素提出的推荐规避措施。矩阵中第一和第二位的推荐建议显示在目录列表里基于评估者的选项当中。关联的技术在五种攻击分类的列中被展示出来。关于这些技术继承于何处的例子和解释可以在建议列表下面的下拉菜单中选择相关技术的时候看到。这个工具也能在矩阵中为指定的安全技术查阅首要和次要的注意事项。

由于有了这样的信息，评估者选择了对其特定的情况而言最好的风险规避途径，并参考注意事项，围绕攻击构成要素的风险预测，围绕该预测的可行技术，关于成本、可行性、权限等执行方面的问题，以及潜在的好处。第九部分提供空白的文本空间，为评估者记录他打算实施的安全技术。

### 评定成本和已被规避的风险

既然评估者已经选择了安全技术来进行进一步的考量和实践，那么这些工具就允许评估者记录他的所消耗的成本、面临的困难，以及每一个攻击构成要素的规避设置的范围。第十部分下面的图7.5说明了这一内容。



整改后为风险评级

获取新的风险分值并与旧值相比

访问并计算成本

**图7.5评定成本和已被规避的风险**

该图也展示了第十一部分，即评估者在安全技术被实施后，评价针对于每一个针对于攻击构成要素的风险有了怎样的变化。评价已规避的风险使用同样的表格，一如未规避的风险表，同时使用下拉菜单和四组等级复制（极低、低、中、高）。这项工具使用了相同的算法综合风险评级和分数值，也在一个新的对象后面再次展示了未规避的风险评级和分数值以供比较。

除了帮助评估者完成评价和决定继续哪种安全技术，成本、适用性、风险评级帮助帮助为未来的评估记录这些评价。当工作表中的脆弱性的每一种可供参考的安全技术的评估都完成，它们提供了评估专家的评价的可视化再现。同时会提供概述和橡树给管理者和使用者，以进行审批和投资决策。

评估者也能使用工作表中的第十部分和第十一部分来记录应用和测试安全技术到实际系统上的结果（VAM的第五和第六步骤）。这些结果较之评估者在第四步的评估有更强的限定，对于通过记录来判断怎样的附加过程应该被实施在那些被识别的部分（例如重复的4到6）非常重要。同样，这样的记录在未来对于评价为减小风险使用安全技术而产生的实际效益提供参考，也是重要的。

# 第八章 下一步以及相关讨论

在这里，我们提出一些当前VAM理论的缺陷，可能的后续步骤，以及关于这套理论的一些常规的讨论、使用和安全评估的效用。

## 未来的挑战与机遇

当VAM理论推动了用于凭和规避信息系统脆弱性的安全技术的发展，完整的六步理论将受益于额外的自动开发和支持援助。

### 引导关键系统和功能的评价

应用“策略——任务”技术到评估关键信息功能及其支撑系统（第一步和第二步）可能获益于工具中特定的引导功能和工作表，而这个工具可以帮助评估者探索和发现更关键的部分，以及帮助提醒他们在对一个对象的评估中避免标准化的关注点以及先入为主的观点。这些步骤关注于基本的信息功能和支撑这些功能的必要的系统，但是额外的思考和构建可能对于定位和关联可以称之为“有价值的”功能和系统会有一些帮助，正如基本的功能和系统。

### 额外的引导和自动操作：拓展表和基于Web的工具

这个拓展表格工具极大的提高了在理论（尤其是第三和第四步）的运用能力，使用基于Web的工具可以提供大量有意义的建议。基于Web的版本可以通过围绕问答表简历，在这当中系统帮助评估者完成整个评估过程。系统也可以用优先的数据和设置帮助评估者应对在自动填入多个脆弱性的下级表格时候的复杂状况。在线数据库也可以促进评估结果的存储和保留，也可以消除在多次脆弱性评估的过程中将工作表多次复制的需求。然而，该电子表格在我们早期对于VAM理论在美国海军系统的运用中被证明是非常有用的。我们也希望能收到对这个方法感兴趣的人们带给我们的反馈。

### 安全选项排序

额外的可操作的运算，对于决定使用何种安全技术的方法和它们能够对脆弱性空间有多大程度的覆盖，都是有益的。Richard Hillestad和他的同事开发的一个还未发表的方法已经被用来作为对RAND自身组织运用VAM评估脆弱性、生成安全选项、排列这些预算有限制的选项和组织风险耐受性的功能组件。我们希望这个完整的程序方法对于其他组织是有用的，而且可以作为VAM理论的重要补充角色而起作用。

### 威胁和风险的量化评估和规避

对威胁、风险和规避手段的量化评估和赋值，是一个遗留的挑战。现在理论中使用的简单评估依赖于评估者的主观经验，也没有提供一种独立的方法生成量化值（甚至仅仅是定性结果）。主观问题在VAM理论运用的领域中被加重了：当危险仅仅存在于理论上，或者脆弱性已经被对手加以利用。如果没有相关的攻击记录，那么一下几点将是非常困难的：

* 估计一个脆弱性被利用的可能性，
* 为应对威胁的安全投资进行成本收益分析，
* 在多种理论上的威胁和脆弱性之间权衡利弊。

这个问题深刻的证明了在公元2001的“9·11”事件之前证明反恐怖主义基金的合理性的困难程度，以及在2002年的“炭疽热邮件事件”之前计算炭疽热袭击的可能性和严重性。无论如何，“9·11”事件和“炭疽热邮件事件”证明了预先发现和规避未被利用的脆弱性，继而寻找下一个脆弱性，而这个预先被利用的脆弱性一旦被定位，对手将可能发起袭击。除了要有反应能力，我们也应该做到预先控制。

### 将VAM功能整合进其他评估方法中

考虑到很多安全评估方法向VAM方法的六个步骤分享了它们的简单步骤，以及事实上这些方法中很多都缺乏VAM评估的深度和规避建议，那么在执行这些其他已经建立起来的方法的时候借鉴使用VAM的核心步骤（第三和第四步）是有一定好处的。同样的，这些好处也体现在从形式上集成VAM理论的核心脆弱性、矩阵、筛选工具和支撑信息到这些方法中。

### 使用VAM指导信息攻击

方法的首要焦点一直是信息系统的保护，然而脆弱性基本面和攻击阶段的边界评估对于理解IO/IW也是有帮助的。如果我们自己进行的攻击，那么对守护方在攻击过程的前中后阶段采取了什么措施的综合评审，将有助于我们在减小暴露量的时候设计更多有用的IO工具、方法和策略。

### 超越信息系统应用VAM理论

另外，准确处理物理的、人与社会、基础设施系统更可能促进提高VAM理论运用在除信息系统外的其他系统中的可用性。很多类型的信息系统承载着非常重要的社会功能（例如金融、能源、交通运输、农业、水利、医药、司法实施、行政），它们依赖于物理的、人与社会和基础设施对象，同时也对计算机网络产生越来越强的依赖性。RAND没有探索VAM方法应用在这些领域中的相关课题，但这似乎是一个有前景的机遇。脆弱属性列表和许多重要的攻击构成要素可能需要做一些针对于这些领域（例如生物领域）的延伸，但是大部分基本属性和攻击阶段仍然是可以考虑使用的。

## 什么脆弱性会失败或称为下一个被攻击的对象？

VAM理论最有力的能力之一便是可以帮助评估者将思维跳出框架，并寻找一定会引起系统故障或者被攻击者利用的新的脆弱性。当系统很复杂或者攻击者对于防范者定义的系统安全响应和环境是富有创造性、适应性的时候，他们这种评估可以变得尤其重要。例如，随着组织变得越来越依赖复杂的信息系统，建立在冗余或者性能监控上的健壮性非常重要。同样，恐怖组织趋向于寻找简单易用的脆弱性来进行破坏，而现在的安全策略常常关注于从前被利用过的脆弱性。

## 可用性问题

注意一点，即使评估者不能通过运用安全技术定向修复一个脆弱性，评估仍然可以用来向其他有责任和权限的为脆弱性提供补救办法的单位或个人提供一份脆弱性的全面的辩护和说明。这些评估和辩护可能非常重要，它可以向其他人传达安全需求的信息，也可以为编制预算和做出决策提供基础。

同样要注意，这里也包括一些讨论，内容是如何应用特殊的安全技术以及测试这些技术的有效性。类似红队测试，或者真实的攻击演习都很难完成。一个组织时常花精力进行一些演习，拿下那些重要的信息系统。例如，INFOCONs就在攻击演习过程中经常被模拟，而攻击性的红队测试则少有被用来进行这是操作系统的对抗性测试。同样，军事部署的标准认证（例如检查（inspections）、认证（certifications）、评估（assessments）、访查（visits）[ICAV]，例如一个计算机网络的脆弱性评估）通过小型的创新型红队测试贯穿自底向上的脆弱性（补丁、警报、病毒等）中。Anderson等人（1999）在第五和第六步中增括了其他观点。

## 为什么执行安全评估？

进行安全评估能要求在时间和资源上进行有意义的投资，但是你会获得回报。这些投资在某些人看来是不必要的，因为许多脆弱性已经被知晓，也因为有限的资源可能已经预防了面向脆弱性的安全响应在事件发生后的实施。所以，许多人没有发现完整评估或揭示脆弱性的行情，因为那还没有被对手利用。

这种行为是短视的。一个认真的、客观的安全评估可以佐证那些追加的付出是值得的。执行类似VAM的方法将安全投资连接向关键信息功能的脆弱性，同时允许管理部分更好的理解脆弱性的动态意义。因此，对资源需求正当性的说明通过适当的语言和其功能效果的水平被表达出来，而不仅仅是一个组织核心功能下，不明确的愿望清单。

同样，执行这个方法可以帮助平衡有限的资源，确保最重要的脆弱性最先被修复，而从更好的成本收益比的角度出发选择安全技术也不会被忽视。

# 第九章 概要和总结

VAM在已经存在的类似方法理论上填补了意向空白，它在查找系统脆弱性上提供了准确的指导，同时给出了相联系的规避建议。VAM理论给出了完整、自顶向下的方法应用于信息系统安全，它通过筛选工具将一种新颖的评估方法和建议生成矩阵结合起来，而这种筛选工具则用来提炼可供考虑的安全选项。

该理论有助于定义新的脆弱性种类，如同信息系统种已知的脆弱性种类。因此，这套理论通过全面的途径去理解脆弱性，而不是依赖于封装好的扫描工具或者检测用例（尽管这很有用）作为为一点的脆弱性识别的主体。

理论中的脆弱性和安全分类完全公正。观察将脆弱属性从系统对象中剥离出来已经证明了一种有价值的系统脆弱性评估方法，因为这些属性常常应用在每一种其他类型的对象上面。同样，每一种对象类型都在信息系统中扮演着重要的角色。除了计算机网络硬件对象方面，明确考虑物理的、人与社会的、基础设施的脆弱性评估的实现和扩展，确认和适应了这些封面对于信息系统正常运行的重要性。

运用VAM理论进行评估作为一种电子化的协助，提高了方法的可用性，尤其是考虑到当前的方法会生成许多建议，这超出Anderson等人（1999）的早期版本所能做的。

现在在Excel中执行的电子表格对于很多安装了Excel程序的个人电脑用户而言是一个利好。这份电子表格能帮助生成分析报告，甚至可以输入定制的风险评级算法，以适应个性化的需求和环境。

这个方法可以在系统设计阶段和操作过程中提升安全性。决策人也可以使用这个方法，来确定提高信息系统安全性能的步骤。

这套方法对团体和个人而言都可有其价值。个人可以关注他们个人的情况和职责范围，而团体可以引入多种专业意见以左右分析过程，如同同一组织中多个不同的部门多重视角。方法也可以由不同的部门平行使用，关注其各种的脆弱性，一旦每一个部分的需求和解释被理解 以后，就将其整合成一个高水准的评估。VAM方法已经证明了它在真实信息系统分而治之思想方面的价值，现在的方法能够从附加的关于第一和第二步的引导以及工具自动化提炼的某些开发中获益。帮助进行风险分析的识别技术和安全选项的成本收益分析的集成，将会变得有价值，并正在积极进行当中。

我们也有理由相信，理论中涉及的一般方法，是脆弱属性中一个有意义的部分，它也可以应用在其他首要角色并非信息处理的系统中。我们同样在探索这些可能性。

# 附录 脆弱性到规避方法的映射值

我们这套方法的核心是标记了将脆弱属性映射到能规避相关脆弱性（表6.1）的安全技术的赋值矩阵。在下面的表格里，我们罗列并解释为什么可以肯定那些技术对于每一个脆弱属性的规避是行之有效的。我们也提醒大家注意哪些规避手段肯定会引起脆弱性的例子。

每一个表格罗列了看上去与表格中的脆弱属性极其相关的安全技术。这些安全技术列在左边的一列中，而解释为什么每一种安全技术与其相关的原因则被写在右边的一列中。更进一步，安全技术被划分为不同的组别，划分的依据是它们是否被判定为首选和帮助规避脆弱属性具有常规重要性，抑或是第二选择，同时没有那么重要。正如在第六章被描述的那样，表6.1的首选的技术被定义为用数字2来赋值，而第二选择的安全技术被表示为数字1。在这里，一些的表格也包括可以得益于另一些脆弱性的安全技术。这种情况出现的时候，这些技术会被罗列在最后两行，并在表6.1中被表示为数字0。

因而，例如表A.1中所表示的异构性作为单一性这个脆弱属性的首选规避技术，集中性则作为第二安全技术。因此，在表6.1中就有一个“2”出现在单一性和异构性分处的行和咧交叉的地方，而对应于单一性和集中性的交叉处则出现了一个“1”。没有可以被单一性强化的安全技术，所以没有“0”。然而，表A.3中出现了四种可以因集中性脆弱属性产生积极影响的安全技术（集中、适应和学习、诱骗ISR、司法执行和民事诉讼）。每一种技术都在表6.1中集中性那一行对应的位置上被填上一个“0”。

## 脆弱性映射并积极影响的规避技术

表A.1  
单一性映射的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 异构 | 异构为唯一的个体或系统提供替代选择。 |
| 冗余 | 冗余系统可以提供更加健壮的能力。 |
| 分散 | 分散可以直接引起冗余或者让单一的对象变得分散，使它们难于成为目标。 |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以使单一资源更坚固。 |
| 暴露度、访问和输出的控制 | 暴露度、访问和输出控制可以保护单一资源。 |
| 加固 | 加固可以直接保护单一资源，使它难于破坏。 |
| 错误、不确定性、有效性等 | 可在错误和复杂环境下进行操作的单一资源更不容易产生故障。 |
| 应急响应体系与计划 | 许多响应计划都围绕着单一资源准备了备份和意外的开销预算。 |
| 紧急重建和恢复 | 紧急恢复可以减少失去单一资源的影响。 |
| 适应和学习 | 这一项提供用以学习和适应材料、计划，允许其他人或物快速顶替单一资源。 |
| **第二选择** |  |
| 集中 | 集中控制有益于管理访问和保护单一资源。 |
| 可信学习和执行系统 | 对于单一系统，可信系统帮助控制访问和暴露量显得特别重要。 |
| 抗抵赖 | 对于单一系统，抗抵赖性提供对不可信行为的威慑和证明是特别重要的。 |
| 静态资源分配 | 静态资源分配可以帮助预防单一资源超载，并分担压力。 |
| 动态资源分配 | 动态资源分配可以帮助预防单一资源超载，并分担压力。 |
| 综合管理 | 适当的管理流程，比如质量控制、培训、综合安全、流程控制可以帮助保护单一资源。 |
| 情报活动 | 情报可以识别什么单一资源是对手所了解的。 |
| 自校正、监视、评估 | 自校正可以识别单一资源。 |
| 常规反侦察 | 反侦察可以预防对手知道脆弱的单一资源。 |
| 对敌的不可预知 | 反侦察可以预防对手知道脆弱的单一资源。 |
| 反侦察欺骗 | 欺骗可以隐藏单一资源。 |
| 屏蔽ISR和目标搜索 | 屏蔽ISR可以隐藏单一资源。 |

表A.2  
唯一性映射的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 异构 | 异构为唯一的个体或系统提供替代选择。 |
| 冗余 | 冗余系统（即使可能不是唯一的同类型）可以提供更加健壮的能力。 |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以使唯一组件更坚固。 |
| 暴露度、访问和输出的控制 | 暴露度、访问和输出控制可以保护唯一组件。 |
| 加固 | 加固可以直接保护唯一系统，使它难于破坏。 |
| 错误、不确定性、有效性等 | 可在错误和复杂环境下进行操作的唯一组件更不容易产生故障。 |
| 应急响应体系与计划 | 许多响应计划都围绕着单一性准备了备份和意外的开销预算。 |
| 紧急重建和恢复 | 紧急恢复可以减少失去唯一组件的影响。 |
| **第二选择** |  |
| 集中 | 处在中心位置的唯一对象应该被更加有效地监控、维护和维修。 |
| 分散 | 分散可以直接引起冗余或者让唯一的对象变得分散，使它们难于成为目标 |
| 静态资源分配 | 静态资源分配可以帮助预防唯一系统超载，并分担压力。 |
| 动态资源分配 | 动态资源分配可以帮助预防唯一系统超载，并分担压力。 |
| 综合管理 | 适当的管理流程，比如质量控制、培训、综合安全、流程控制可以帮助保护唯一资源。 |
| 情报活动 | 情报可以识别什么唯一系统是对手所了解的。 |
| 自校正、监视、评估 | 自校正可以识别系统的唯一性（资源）。 |
| 常规反侦察 | 反侦察可以预防对手知道脆弱的、唯一组件。 |
| 对敌的不可预知 | 反侦察可以预防对手知道脆弱的、唯一组件。 |
| 反侦察欺骗 | 欺骗可以隐藏唯一组件。 |
| 屏蔽ISR和目标搜索 | 屏蔽ISR可以隐藏唯一组件。 |
| 刑罚与司法保证 | 被担保人和担保服务是一种有用的服务，用以保证唯一对象的性能和稳定性，也可以解决故障（在更长的时间范围）。 |

表A.3  
集中性映射以及被其促进的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 分散 | 分散直接寻址的中心关注点。 |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以使集中系统更坚固。 |
| 暴露度、访问和输出的控制 | 暴露度、访问和输出控制可以保护集中性。 |
| 加固 | 加固可以直接保护集中性，使它难于破坏。 |
| 错误、不确定性、有效性等 | 可在错误和复杂环境下进行操作的集中的资源更不容易产生故障。 |
| 应急响应体系与计划 | 许多响应计划都研发了一些途径来保护和备份集中性性能。 |
| 预防性和报复性的信息和军事活动 | 集中的设施对于敌对方通常是高价值的目标，因此一旦被破坏，确保一种强有力的、侵略性的反击。 |
| **第二选择** |  |
| 异构 | 不可避免的，异构性减小了系统的变化程度，即使他们仍在同一中心站点维护。 |
| 冗余 | 冗余系统在即使集中化的情况下仍可以使其更健壮。 |
| 可信学习和执行系统 | 对于集中系统，可信系统帮助控制访问和暴露量显得特别重要。 |
| 抗抵赖 | 对于集中系统，抗抵赖性提供对不可信行为的威慑和证明是特别重要的。 |
| 综合管理 | 适当的管理流程，比如质量控制、培训、综合安全、流程控制可以帮助保护集中化资源。 |
| 紧急重建和恢复 | 紧急恢复可以减少失去集中化资源的影响。 |
| 免疫式防御系统 | 这些系统通过分散节点分享信息的能力可以避免中心化的原因。 |
| 情报活动 | 情报可以识别什么唯一系统是对手所了解的。 |
| 自校正、监视、评估 | 自校正可以描述我们对于集中化系统的依赖范围。 |
| 常规反侦察 | 反侦察可以预防对手定位和描述我们的集中化资源。 |
| 对敌的不可预知 | 反侦察可以预防对手定位和描述我们的集中化资源。 |
| 反侦察欺骗 | 欺骗可以隐藏集中化资源。 |
| 屏蔽ISR和目标搜索 | 屏蔽ISR可以隐藏集中化资源。 |
| **集中性的促进功能** |  |
| 集中 | 手段集中至最大化（例如质量控制、相容性） |
| 适应和学习 | 集中性可以通过提高自我意识和反馈促进学习和适应性。课程学习的传播同样可以因集中性和得到促进。 |
| 反侦察欺骗 | 集中性可以确保和整合一些欺骗手段，提高反侦察的能力。 |
| 司法执行和民事诉讼 | 集中的项目更易于受到法律强制力的保护。 |

表A.4  
一致性映射以及被其促进的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 异构 | 异构性与一致性是相互对立的，它意图引起一系列不同的可供选择的系统。 |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以更大量的运用于同类系统，并且使它们较之异构系统更加坚固。 |
| 加固 | 加固同类系统可以使它减少被攻击的脆弱性。 |
| 错误、不确定性、有效性等 | 可在错误和复杂环境下进行操作的同类系统更不容易产生故障。 |
| **第二选择** |  |
| 冗余 | 在没有减少一致性的前提下，冗余项目可以使这些系统更加坚固以及更有效的应对故障。 |
| 分散 | 分散同类目标是它们更难于在一次攻击中收到侵害。 |
| 暴露度、访问和输出的控制 | 暴露度、访问和输出控制可以直接保护同类组件。同类系统有利于控制体系的设计。 |
| 综合管理 | 适当的管理流程，比如质量控制、培训、综合安全、流程控制可以帮助保护异构系统。注意，同类系统可以帮助促进信息系统的管理。 |
| 自校正、监视、评估 | 自校正可以决定我们的系统可以变向哪种异构对象。 |
| 常规反侦察 | 反侦察可以防止对手了解我们对哪些系统做的了标准化。 |
| 对敌的不可预知 | 反侦察可以防止对手了解我们对哪些系统做的了标准化。 |
| 反侦察欺骗 | 虚假的异构系统可以隐藏对同类系统的依赖关系。 |
| 屏蔽ISR和目标搜索 | 屏蔽ISR可以防止敌手了解我们对哪些系统做的了标准化。 |
| **异构性的促进功能** |  |
| 静态资源分配 | 注意，异构系统可以帮助促进资源分配。 |
| 动态资源分配 | 注意，异构系统可以帮助促进资源分配。 |
| 综合管理 | 适当的管理流程，比如质量控制、培训、综合安全、流程控制可以帮助保护异构系统。注意，同类系统可以帮助促进信息系统的管理。 |
| 紧急重建和恢复 | 异构性可以因备选系统和部件（也包括关于这些组件的培训和知识学习）的可用性而促进重构和恢复。 |
| 免疫式防御系统 | 异构性可以是从其他节点学习应用课程更加容易。 |
| 防疫 | 异构性可以是从其他节点学习应用课程更加容易。 |

表A.5  
分散性映射以及被其促进的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 集中 | 集中的系统难于被分离。 |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以发现一些使系统组件变得独立的途径，并且找到减少这种脆弱性的方法。 |
| 错误、不确定性、有效性等 | 无论是衰退的环境条件还是不确定性，都可以针对这样的对象进行操作的系统更不容易被分离。 |
| 综合管理 | 适当的管理和协调更有助于确保凝聚和交流。 |
| 自校正、监视、评估 | 监控可以确定信息系统接口处的脱离，促进其复位。评估可以识别在过去发生了怎样的分离情况，报告补救措施。 |
| **第二选择** |  |
| 暴露度、访问和输出的控制 | 暴露度、访问和输出控制可以帮助保护对抗分离情况。 |
| 可信学习和执行系统 | 可信系统可以报告接口控制者以及减少欺骗性分离的可能性。 |
| 加固 | 加固系统接口可以使它们难以被破坏。 |
| 适应和学习 | 适应可以帮助学习并且识别出分隔系统的企图。 |
| 免疫式防御系统 | 信息共享可以阻止需求，以便在系统遭遇攻击时进行隔离，也可以分享关于这些攻击的信息以及如何防御它们。 |
| 防疫 | 模拟攻击可以揭示分离风险，推动风险规避评估。 |
| 情报活动 | 有些信息可以加速重连组件和激活自有分区的效果。 |
| 常规反侦察 | 反侦察可以预防对手了解如何分割系统组件。 |
| 对敌的不可预知 | 反侦察可以预防对手了解如何分割系统组件。 |
| 反侦察欺骗 | 欺骗对手可以是他们更难知道如何分割系统组件。 |
| 屏蔽ISR和目标搜索 | 屏蔽ISR可以减少对手关于如何分割系统组件的认知。 |
| **异构性的促进功能** |  |
| 反侦察欺骗 | 了解分隔可以用在欺骗上面，以确定敌人的大概知识、性能，以及关于我们的特殊知识。 |

表A.6  
逻辑/现实错误、不可靠性映射的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 异构 | 如果具备不同的故障条件，那么一个多样的系统群可以彼此补充。 |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以在推荐解决方案的时候识别错误和不可靠性。 |
| 暴露度、访问和输出的控制 | 暴露度、访问和输出控制可以用来分离系统中没有发生错误和确定没有问题的部分。 |
| 加固 | 加固可以处理错误，降低其不可靠性。 |
| 错误、不确定性、有效性等 | 容错系统在错误发生时具有更好的处理能力。 |
| 综合管理 | 管理评审和质量控制可以帮助识别和避免错误和不可靠性。 |
| 适应和学习 | 性能考核可以帮助定位错误并调整它们。 |
| 免疫式防御系统 | 一些系统自动识别、更新、打补丁或修正错误。 |
| 防疫 | 防疫直接揭示和修复错误。 |
| **第二选择** |  |
| 集中 | 处于集中位置的有缺陷的系统更易于被管理、控制、修复。 |
| 分散 | 当错误是分散的，那么就很难对它们进行理解和利用。 |
| 可信学习和执行系统 | 由于多次访问和关于保护措施的推理，可信学习可以减少不可靠性。 |
| 静态资源分配 | 资源分配可以围绕着错误和不可靠性进行。 |
| 动态资源分配 | 资源分配可以围绕着错误和不可靠性进行。 |
| 应急响应体系与计划 | 许多响应计划都引起了备份和意外开支的预算，来减小不可靠性，或将错误最小化。 |
| 紧急重建和恢复 | 快速恢复能力减少（但通常没有评估）系统组件损失和错误的影响。 |
| 自校正、监视、评估 | 监控是评估可以查找错误和不可靠性。 |
| 常规反侦察 | 反侦察可以减少对手对于系统错误和不可靠性的了解程度。 |
| 对敌的不可预知 | 反侦察可以减少对手对于系统错误和不可靠性的了解程度。 |
| 反侦察欺骗 | 欺骗可以减少对手对于系统错误和不可靠性的了解程度。 |
| 屏蔽ISR和目标搜索 | 屏蔽ISR可以减少对手对于系统错误和不可靠性的了解程度。 |
| 刑罚与司法保证 | 被担保人和约定可以为故障系统提供补救措施，同时激励供应商首先评估问题。 |
| 司法执行和民事诉讼 | 被担保人和约定可以为故障系统提供补救措施。 |

表A.7  
设计敏感性、薄弱/限制/有限性映射以及被其促进的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 异构 | 如果具备不同的敏感性、薄弱性、操作范围和规模限制，那么一个多样的系统群可以彼此补充。 |
| 冗余 | 冗余系统可以提供可靠性能或帮助在性能达到上限是分离负载。 |
| 分散 | 当被分散以后，薄弱性就难于被发现和利用，并且限制在系统当中。如果信息处理过程可以被保护，分散性也可以引起更高的性能供利用（分布式计算）。分散系统可以用来作为系统性能达到上限时候的可供利用的对象。 |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以识别和解决性能达到上限时的问题。 |
| 暴露度、访问和输出的控制 | 控制措施可以帮助保护薄弱系统原理恶劣环境或超负载攻击。 |
| 加固 | 加固可以减少设计上的问题。 |
| 错误、不确定性、有效性等 | 容错系统在故障、系统敏感性问题、性能限制问题发生时具有更好的处理能力。 |
| 静态资源分配 | 资源分配可以平衡负载预防故障和分工协作，避免敏感性问题。 |
| 动态资源分配 | 资源分配可以平衡负载预防故障和分工协作，避免敏感性问题。 |
| 综合管理 | 注意管理可以避免系统超载和薄弱系统组件的压力过大。 |
| 自校正、监视、评估 | 状态监控可以帮助管理预防系统发生交叉限制，避免过分敏感等问题。评估可以继续识别位置的薄弱点和限制点。 |
| **第二选择** |  |
| 集中 | 如果是集中管理的，薄弱和有线的系统将更易于控制他们的负载和输入。 |
| 应急响应体系与计划 | 响应计划可以提供额外的资源以最小化设计敏感性的影响和限制（如果事先知道它们）。 |
| 紧急重建和恢复 | 快速恢复能力减少（但通常没有评估）系统组件损失和薄弱、限制等的影响。 |
| 适应和学习 | 如果进行了有针对的开发，性能考核可以帮助定位薄弱点和限制。 |
| 免疫式防御系统 | 一些系统自动识别、阻断、更新、打补丁或修正敏感性。 |
| 防疫 | 防疫直接揭示薄弱性和错误。有一些可能被直接修复了，而另一些可能在未来避免。 |
| 情报活动 | 有关于指向目标的限制和敏感性的信息可以被用来计划和执行对抗措施。 |
| 检测攻击、损伤评估、敌我识别 | 行为和环境监控可以帮助描述敏感性、限制、薄弱点。 |
| 常规反侦察 | 反侦察可以减少对手对于系统敏感性和限制的了解程度。 |
| 对敌的不可预知 | 反侦察可以减少对手对于系统敏感性和限制的了解程度。 |
| 反侦察欺骗 | 欺骗可以减少对手对于系统敏感性和限制的了解程度。 |
| 屏蔽ISR和目标搜索 | 屏蔽ISR可以减少对手对于系统敏感性和限制的了解程度。 |
| 刑罚与司法保证 | 被担保人和约定可以为薄弱系统提供补救措施，同时激励供应商首先评估问题。 |
| 司法执行和民事诉讼 | 被担保人和约定可以为薄弱系统提供补救措施。 |
| **设计敏感性、薄弱/限制/有限性的促进功能** | |
| 反侦察欺骗 | 了解敏感性可以用在欺骗上面，以确定敌人的大概知识、性能，以及关于我们的特殊知识。 |

表A.8  
不可恢复性映射的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 异构 | 不同的系统可能在不同的时候发生故障，这样可以避免系统全部陷入故障当中。 |
| 冗余 | 一旦系统故障，冗余系统可以提供可靠性能。 |
| 分散 | 一旦系统故障，分布式操作可以提供补充的性能。 |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以识别为什么系统是不可恢复的，以及推荐补救措施。 |
| 加固 | 加固可以减少系统第一次发生故障的可能性。 |
| 错误、不确定性、有效性等 | 容错系统第一次发生故障的可能性更小。 |
| 自校正、监视、评估 | 早先发现不可恢复的故障可以加速恢复流程的执行、报告如何在未来避免故障，以及报告能令系统更具首次故障恢复能力的方法。 |
| **第二选择** |  |
| 集中 | 如果是集中管理的，不可恢复系统将更易于预防首次故障。 |
| 暴露度、访问和输出的控制 | 分区和隔离可以帮助一个不可恢复的组件限制损坏范围。 |
| 静态资源分配 | 资源分配有事可以围绕不可恢复的故障工作。 |
| 动态资源分配 | 资源分配有事可以围绕不可恢复的故障工作。 |
| 综合管理 | 管理可以帮助避免首次故障的发生。 |
| 应急响应体系与计划 | 响应计划可以提供备份和意外开销的预算来应对不可恢复故障。 |
| 适应和学习 | 学习可以帮助避免未来发生不可恢复故障的条件。 |
| 免疫式防御系统 | 不可恢复性可能是当攻击被识别和理解的时候，预先从其他系统上获得的。 |
| 防疫 | 攻击可以明确不可恢复的能力，可能激发规避的想法。 |
| 情报活动 | ISR可以向我们报告攻击，提示不可恢复的资产，以及发现特殊攻击，并从冲过滤出来。 |
| 检测攻击、损伤评估、敌我识别 | 更理想的故障分析可以帮助理解是什么引起故障，在未来避免其形成条件，以及最先纠正错误模式。 |
| 常规反侦察 | 反侦察可以减少对手对于系统中不可恢复部分的了解程度。 |
| 对敌的不可预知 | 反侦察可以减少对手对于系统中不可恢复部分的了解程度。 |
| 反侦察欺骗 | 故意欺骗可以减少对手对于系统中不可恢复部分的了解程度。 |
| 屏蔽ISR和目标搜索 | 屏蔽ISR可以减少对手对于系统中不可恢复部分的了解程度。 |
| 刑罚与司法保证 | 被担保人和约定可以为故障系统提供补救措施，同时激励供应商首先评估问题。 |
| 司法执行和民事诉讼 | 被担保人和约定可以为故障系统提供补救措施。 |

表A.9  
行为敏感性/脆弱性映射的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 异构 | 异构系统含有不同的敏感性和薄弱性，可以提供补充的性能。 |
| 冗余 | 冗余系统可以帮助弥补行为敏感性和薄弱性。 |
| 分散 | 当被分散以后，薄弱性就难于被发现和利用，并且限制在系统当中。同样的，自主分散系统、异构实体可以提供更坚固的保护。 |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以识别行为的敏感性和薄弱性，以及推荐补救措施。 |
| 暴露度、访问和输出的控制 | 控制可以帮助保护薄弱系统远离恶劣环境或超载攻击。 |
| 加固 | 加固可以减少行为的薄弱和敏感。 |
| 错误、不确定性、有效性等 | 容错系统可以更好的处理薄弱、敏感和限制，当他们发生的时候。 |
| 静态资源分配 | 资源分配可以平衡负载，预防故障和分工协作避免敏感性。 |
| 动态资源分配 | 资源分配可以平衡负载，预防故障和分工协作避免敏感性。 |
| 综合管理 | 专心的管理可以避免压迫薄弱的组件，以及控制行为敏感性。 |
| 自校正、监视、评估 | 状态监控可以帮助管理预防系统交叉进入敏感操作环境。评估可以继续识别位置的薄弱和敏感性。 |
| **第二选择** |  |
| 集中 | 如果是集中管理的，行为敏感性和薄弱性是易于观察和管理的。 |
| 应急响应体系与计划 | 响应计划提供额外的资源以最小化现实和设计敏感性的影响。 |
| 紧急重建和恢复 | 快速恢复能力减少（但通常没有评估）因系统薄弱和交叉限制而造成的系统组件损失的影响和薄弱、限制等的影响。 |
| 适应和学习 | 当开发了相应的方法，性能检测可以帮助定位薄弱环节和限制。 |
| 免疫式防御系统 | 一些系统自动报警、阻断、识别、更新、打补丁以及纠正敏感性。 |
| 防疫 | 防疫直接揭示薄弱和限制点。有些能直接纠正，有些在未来避免。 |
| 情报活动 | 指向目标薄弱环节和敏感性的信息可以用来计划、执行对抗措施。 |
| 检测攻击、损伤评估、敌我识别 | 行为和环境监控可以帮助描述敏感性、限制和薄弱环节。 |
| 常规反侦察 | 反侦察可以减少对手对于系统敏感性和薄弱环节的了解程度。 |
| 对敌的不可预知 | 反侦察可以减少对手对于系统敏感性和薄弱环节的了解程度。 |
| 反侦察欺骗 | 故意欺骗可以减少对手对于系统敏感性和薄弱环节的了解程度。 |
| 屏蔽ISR和目标搜索 | 屏蔽ISR可以减少对手对于系统敏感性和薄弱环节的了解程度。 |
| 刑罚与司法保证 | 被担保人和约定可以为故障系统提供补救措施，同时激励供应商首先评估问题。 |
| 司法执行和民事诉讼 | 被担保人和约定可以为故障系统提供补救措施。 |

表A.10  
恶意行为映射的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以识别恶意行为和解决恶意动态。 |
| 暴露度、访问和输出的控制 | 控制可以帮助趋利或封闭恶意组件，隔离关键系统，在关键区域针对恶意行为执行更深入的检查。 |
| 可信学习和执行系统 | 可信学习和执行系统能帮助识别和控制恶意行为和实体。 |
| 抗抵赖 | 抗抵赖可以给恶意行为找到源信息，并对恶意实体进行威慑。 |
| 综合管理 | 管理可以主动监控恶意成员。 |
| 情报活动 | 情报信息可以专门查找恶意成员。 |
| 自校正、监视、评估 | 监控可以在早期发现恶意成员。 |
| 诱骗ISR | 欺骗是一种揪出恶意成员的有用手段。 |
| 检测攻击、损伤评估、敌我识别 | 监控和评估能直接起到查找恶意成员的作用。 |
| 常规反侦察 | 反侦察寻找提供系统情报的内部恶意人员。 |
| 对敌的不可预知 | 反侦察寻找提供系统情报的内部恶意人员。 |
| 反侦察欺骗 | 欺骗寻找提供系统情报的内部恶意人员。 |
| 屏蔽ISR和目标搜索 | 屏蔽ISR寻找提供系统情报的内部恶意人员。 |
| 威慑 | 威慑可以抑制恶意动态。 |
| 预防性和报复性的信息和军事活动 | 积极的行动可以排除或牵制恶意成员。 |
| 刑罚与司法保证 | 惩罚措施可以阻止恶意成员，或者主动抑制已捕获的行为。 |
| 司法执行和民事诉讼 | 强制力可以限制恶意成员。 |
| **第二选择** |  |
| 异构 | 不同的系统可能有不同的恶意趋向、弱点，甚至完全不存在这种趋向，以及让系统脱离恶意的风险回避方法也是不一样的。 |
| 冗余 | 冗余可以减小故障系统的不良影响。 |
| 分散 | 恶意实体在控制和处理过程处于分散状态时的影响会减小，因为它要求在不同系统的更宽范围付出更多努力和取得更广的视野。 |
| 应急响应体系与计划 | 开发良好的计划可以减少恶意人员的访问和损坏。 |
| 免疫式防御系统 | 监控和分享将减少恶意实体隐藏维持访问后门或跳到新系统而不被发现。 |
| 防疫 | 模拟攻击可以使系统对于恶意行为变得敏感。 |

表A.11  
强度映射的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以识别强度，以及推荐补救措施。 |
| 可信学习和执行系统 | 可信系统会适应系统的访问和信息，这些信息曾持续改变行为和新实体。 |
| 错误、不确定性、有效性等 | 容错系统更加宽容，并且可以在更广的输入范围内操作。 |
| 动态资源分配 | 动态分配应该调整以适应当前条件。 |
| 综合管理 | 专心的管理可以对新的问题有反应，并尽快解决。 |
| 应急响应体系与计划 | 计划可以适应当前的状况，尤其是当它们提供一般的周边环境，布置和本地响应者可以工作的选择对象。 |
| 紧急重建和恢复 | 应急重建和恢复能在失去响应能力的时候提供弹性空间。 |
| 适应和学习 | 动态适应和学习可以调整系统配置以匹配当前需求。 |
| 免疫式防御系统 | 这些系统寻找新的威胁和解决办法，一旦找到，它们分享信息，并提供快速更新来改变现状。 |
| 防疫 | 防疫解释哪里的系统需要被整顿。 |
| **第二选择** |  |
| 异构 | 不同的系统在不同的方式下强度不同，它们的差异可能突出了原生系统的强度。 |
| 分散 | 分散系统趋向于更易于调整、更灵活、更适应本地条件。 |
| 静态资源分配 | 静态分配可以为当前环境引起不同的响应等级。 |
| 自校正、监视、评估 | 监控和评估可以发现强度。 |
| 检测攻击、损伤评估、敌我识别 | 理解攻击行为常常引起内部变化，提高安全性。 |
| 常规反侦察 | 反侦察可以减少对手对于系统强度的了解程度。 |
| 对敌的不可预知 | 反侦察可以减少对手对于系统强度的了解程度。 |
| 反侦察欺骗 | 故意欺骗可以减少对手对于系统强度的了解程度。 |
| 屏蔽ISR和目标搜索 | 屏蔽ISR可以减少对手对于系统强度的了解程度。 |

表A.12  
可塑性映射的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以识别系统何处具有可塑性，并推荐解决办法。 |
| 可信学习和执行系统 | 可信系统引起更多约束和监管，以此使控制和操纵信息系统更难。 |
| 加固 | 加固可以使系统不易变化无常。 |
| 综合管理 | 管理监督可以监控不期望的变更和操作。 |
| 应急响应体系与计划 | 计划为操作和意外事故提供框架，减少系统被操纵的可能性。 |
| 自校正、监视、评估 | 如果系统具备自我矫正能力，那么他们就不容易被操纵，而当你试图操纵它的时候，行为也是可见的。 |
| 威慑 | 威慑可以对方更谨慎，客观上降低他们的控制力。 |
| **第二选择** |  |
| 异构 | 不同的系统有不同的可塑性。 |
| 集中 | 集中系统可以被更有效的控制，因此于己而言更难操纵。 |
| 分散 | 分布式系统较之集中情况下更不易在没有认证的情况下被改变。 |
| 暴露度、访问和输出的控制 | 控制使其更不倾向于在系统没有认证的前提下被更改。 |
| 抗抵赖 | 如果一直明确源信息，那么系统更不倾向于被操控。 |
| 静态资源分配 | 预先计划的分别可以预防分配配置被操控。 |
| 适应和学习 | 如果积极地进行性能检查并调节适应当前环境，那么系统更不倾向于被操控。 |
| 检测攻击、损伤评估、敌我识别 | 对攻击行为的理解和认知能使主题更不容易被控制，而且能够为未来纠正错误识别没有保护的控制点。 |
| 常规反侦察 | 反侦察可以减小对手理解系统的控制点和操控方法。 |
| 对敌的不可预知 | 反侦察可以减小对手理解系统的控制点和操控方法。 |
| 反侦察欺骗 | 欺骗可以减小对手理解系统的控制点和操控方法。 |
| 屏蔽ISR和目标搜索 | 屏蔽ISR可以减小对手理解系统的控制点和操控方法。 |
| 刑罚与司法保证 | 惩罚的存在使得威慑更有实效。 |

表A.13  
欺骗性映射的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以检查系统是否容易被欺骗，同时可以推荐解决办法。 |
| 可信学习和执行系统 | 可信系统引起更多约束和监管，以此想愚弄信息系统更难。 |
| 加固 | 加固可以使系统对可靠的信息源有更强的认知能力，也更坚定。 |
| 综合管理 | 管理监督可以监控期望之外的变化，分享威胁知识和提供建议。 |
| 应急响应体系与计划 | 计划为操作和意外事故提供框架，减少系统被盲目操作的可能性。 |
| 适应和学习 | 注意力和适应性在当前环境下可以减少盲目行为。 |
| 防疫 | 模拟攻击可以使系统对于恶意行为变得敏感，也更不易受欺骗。 |
| 情报活动 | 情报可以提供关于我们对手和他们的技术水平的信息。 |
| 自校正、监视、评估 | 如果系统具有自我矫正能力那么它将难以被控制，同时如果被控制了，那么也是可以被发现的。 |
| **第二选择** |  |
| 暴露度、访问和输出的控制 | 控制往往在有意义的长远考虑下使用，它可以避免很多幼稚的地方。 |
| 抗抵赖 | 如果提供了源信息，那么他就不容易被欺骗。 |
| 静态资源分配 | 静态分配通常在预先被很好的设计。 |
| 免疫式防御系统 | 这些系统的适应、分享、自动运维的性质是的他们难以攻击简历在不听从或不知晓最新信息的系统的某部分。 |
| 检测攻击、损伤评估、敌我识别 | 理解和认识攻击可以使实体不易被欺骗，也不会在同一个诡计前显得无知。 |
| 常规反侦察 | 反侦察能减少对手了解系统的偏差和操作，使其更难控制。 |
| 对敌的不可预知 | 反侦察能减少对手了解系统的偏差和操作，使其更难控制。 |
| 反侦察欺骗 | 欺骗反侦察能减少对手了解系统的偏差和操作，使其更难控制。 |
| 屏蔽ISR和目标搜索 | 屏蔽ISR反侦察能减少对手了解系统的偏差和操作，使其更难控制。 |

表A.14  
自满情绪映射的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以帮助系统持续识别系统弱点，保证适当的流程。 |
| 可信学习和执行系统 | 可信系统会适应系统的访问和信息，这些信息曾持续改变行为和新实体。 |
| 动态资源分配 | 动态资源分配把注意力集中在当前环境上。 |
| 综合管理 | 主动管理可以持续查找和适应新的威胁。 |
| 应急响应体系与计划 | 计划的制定吸引人们主动考虑脆弱性，并建立意外事故系统来促进响应。 |
| 适应和学习 | 注意力和适应性以及系统性能在当前环境下可以减少自满情绪。 |
| 免疫式防御系统 | 这些系统时刻保持对可疑活动和问题的警惕。 |
| 情报活动 | 当下的和细致的对敌人活动的理解可以激励我们避免自满情绪。 |
| 自校正、监视、评估 | 完整的攻击和活动的知识可以激励人们主动行动。 |
| 威慑 | 警告和惩罚可以阻止成员变得自满。 |
| **第二选择** |  |
| 集中 | 集中可以引起有规律的预订的安全检查和流程，因此减少了自满情绪。 |
| 暴露度、访问和输出的控制 | 如果他们主动维持和提高，额外对控制的注意可以减少自满情绪。 |
| 防疫 | 模拟攻击可以使系统对于恶意行为变得敏感并保持警惕。 |
| 检测攻击、损伤评估、敌我识别 | 基于对先前的攻击产生的实际危险的认知将会减少自满的情绪。自动分析系统可以捆绑在保护系统上也能起到同样的效果。 |
| 常规反侦察 | 情报风险的认知可以激励人们在安全上投入更多注意力。 |
| 对敌的不可预知 | 情报风险的认知可以激励人们在安全上投入更多注意力。 |
| 反侦察欺骗 | 情报风险的认知可以激励人们在安全上投入更多注意力。 |
| 刑罚与司法保证 | 惩罚措施能是警告和威慑更令人谨慎。 |

表A.15  
易腐蚀性/可控性映射的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以识别和补救可以被利用的弱点。 |
| 暴露度、访问和输出的控制 | 控制过滤器可以保护以防御一般控制问题和脆弱性。 |
| 可信学习和执行系统 | 可信系统引起更多约束和监管，以此使控制和操纵信息系统更难。 |
| 加固 | 加固可以使系统具有更强的认知能力，并坚持可靠的信息源。 |
| 综合管理 | 管理监督可以监控不期望的变更和操作。 |
| 免疫式防御系统 | 使用最新最好的安全知识和流程将使系统更难被直接攻击和控制。 |
| 情报活动 | 情报可以直接监控腐蚀（腐败）和识别敌对势力及其活动，如果有迹象表明他要控制和访问你的系统。 |
| 自校正、监视、评估 | 自我监控可以识别腐败行为，评估可以识别可控制点和腐败的危险信号。 |
| 威慑 | 威慑可以减少成员的腐败和控制。 |
| **第二选择** |  |
| 异构 | 不同的系统有不同的易腐蚀和易控制的弱点，或又一下弱点在不同的区域，所以他们能帮助补偿其他弱点。 |
| 集中 | 集中控制可以帮助监控和处理腐败和篡夺控制。 |
| 抗抵赖 | 一些系统保护信息量不受腐蚀，或者确定系统更新的来源。 |
| 防疫 | 模拟攻击可以使系统对于恶意行为变得敏感并警惕腐败行为。 |
| 检测攻击、损伤评估、敌我识别 | 对攻击行为的理解和认知能使主题更不容易被控制，而且能够为未来纠正错误识别没有保护的控制点。 |
| 常规反侦察 | 评估可以识别可被控制的点和外界腐蚀的危险信号（例如私人问题，无意行为）。 |
| 对敌的不可预知 | 评估可以识别可被控制的点和外界腐蚀的危险信号（例如私人问题，无意行为）。 |
| 反侦察欺骗 | 欺骗可以减小对手理解系统的控制点和易腐蚀点。 |
| 刑罚与司法保证 | 惩罚的存在使得威慑更有实效。 |

表A.16  
可访问/可侦测/可识别/透明/可截取映射的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 分散 | 分布式系统难于在整体上发现、识别、跟踪、访问和截断。 |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以识别和补救可以被利用的弱点。 |
| 暴露度、访问和输出的控制 | 这些控制直接被设计用来减少可访问性、可探测性和可截断性。 |
| 可信学习和执行系统 | 可信系统可以适应系统限制，访问和暴露于可信赖的实体。 |
| 加固 | 如果访问是有危害的，加固可以使系统减少可访问性、可拦截性以及被损毁的可能。 |
| 错误、不确定性、有效性等 | 容错系统在访问有危害的时候可以减小危害的可能性。 |
| 应急响应体系与计划 | 响应计划能减少基于（感知）威胁和环境的可见性和暴露量。 |
| 常规反侦察 | 反侦察工作直接将对手的发现、识别、访问和截断系统组件的能力最小化。 |
| 对敌的不可预知 | 反侦察工作直接将对手的发现、识别、访问和截断系统组件的能力最小化。 |
| 反侦察欺骗 | 欺骗可以直接干扰探测、识别和透视。 |
| 屏蔽ISR和目标搜索 | 屏蔽ISR可以直接干扰探测、识别和透视。 |
| 威慑 | 当监控和拦截行为被公开，威慑将增大它们的成本。 |
| 预防性和报复性的信息和军事活动 | 主动报复能保固访问点，增加监控和拦截的成本，使危害更明显。 |
| **第二选择** |  |
| 异构 | 不同级别的系统类型将难以被跟踪、识别和访问。 |
| 冗余 | 多路系统更难被识别和跟踪。 |
| 综合管理 | 积极的和计划良好的管理行为可以帮助最小化暴露量和截断行为。 |
| 免疫式防御系统 | 防疫和自动分享能将暴露控制在最高水平。 |
| 防疫 | 暴露控制下的攻击能强化其弱点，识别错误配置和激励行为。 |
| 自校正、监视、评估 | 关于对手传感器性能的情报可以向我们提供对策设计和操作流程提供有用信息。 |
| 检测攻击、损伤评估、敌我识别 | 可能在应对当前的攻击时提供攻击阻断机制的信息时发现和取证能识别弱点。 |
| 刑罚与司法保证 | 惩罚措施可以使警告和威慑更有实效。 |
| 司法执行和民事诉讼 | 强制力可以在访问点上提供物理保护。 |

表A.17  
难以控制或管理映射的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 集中 | 集中化便于管理和控制操作。 |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以检查系统为何难于管理和控制，同时推荐如何提高系统性能的办法。 |
| 暴露度、访问和输出的控制 | 暴露量、访问和输出控制架构可以帮助在管理和控制进入、流出以及在信息系统内部的信息流。 |
| 可信学习和执行系统 | 可信系统可以引起更多限制和支持以进行系统管理。尤其是当环境中包含可靠性位置的实体。 |
| 静态资源分配 | 资源分配计划引起额外的管理控制架构。 |
| 动态资源分配 | 资源分配计划引起额外的管理控制架构。 |
| 综合管理 | 管理架构和控制点上更多的注意力可以帮助系统回到可控状态。 |
| 应急响应体系与计划 | 计划和意外开销提供更对的方法来管理和控制系统。 |
| 自校正、监视、评估 | 自我信息对于有效的管理和控制是一个重要的先决条件。 |
| **第二选择** |  |
| 免疫式防御系统 | 系统的自动化特性促进管理，尤其是对于分布式和多样化的系统。 |
| 检测攻击、损伤评估、敌我识别 | 提高的弱点和操作的理解水平能提高管理水平。 |
| 威慑 | 威慑使用管理工具，帮助控制成员的行为。 |
| 刑罚与司法保证 | 惩罚措施可以使警告和管理行为更有实效。 |
| 司法执行和民事诉讼 | 强制力表明不对行为加以管理将会导致现实的惩罚。 |

表A.18  
不自知和不可预测性映射的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 集中 | 集中化便于监控和控制操作。 |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以识别和解决自校正和不可预知性方面的限制。 |
| 可信学习和执行系统 | 可信系统加入监控，这使它更加意识到系统正在发生的状况。 |
| 免疫式防御系统 | 系统的自我监控组件帮助提供全系统状态和行为的洞察力。 |
| 防疫 | 模拟攻击将在信息系统和它们在压力下的操作方面提供更丰富的视角和信息。 |
| 自校正、监视、评估 | 新的科技将我们自己的系统信息收集到一起，这可以直接定位这些缺陷。 |
| 检测攻击、损伤评估、敌我识别 | 监控和分析将能提高对信息系统的认知水平。 |
| **第二选择** |  |
| 静态资源分配 | 资源分配为信息系统和它本身的操作提供状态信息。 |
| 动态资源分配 | 资源分配为信息系统和它本身的操作提供状态信息。 |
| 综合管理 | 自我认识在建立管理框架和控制的时候是非常重要的一步。 |
| 应急响应体系与计划 | 计划往往为减少缺陷而引起自身系统和控制体系的新的信息源。 |
| 常规反侦察 | 反侦察往往需要对我们自己的系统有作为情报目标一样彻底的认识。 |
| 对敌的不可预知 | 反侦察往往需要对我们自己的系统有作为情报目标一样彻底的认识。 |
| 反侦察欺骗 | 欺骗往往需要对我们自己的系统有作为情报目标一样彻底的认识。 |

表A.19  
可预知性映射以及被其促进的规避技术

|  |  |
| --- | --- |
| **首选** |  |
| 异构 | 一系列不同的系统类型将要求更多的资源来了解和预告它们将如何运作，尤其是如果它们交叉冲突产生突发行为。 |
| 验证确认、软硬件工程、评估和测试 | 工程、验证确认、评估、测试可以识别和解决系统中过多的可预测性。 |
| 动态资源分配 | 动态分配能减少可预测性，因为它们依赖当前的环境。 |
| 适应和学习 | 适应性提供一个活动的目标，让对手去了解。 |
| 免疫式防御系统 | 围绕系统的快速修补能力能使它更难让对手维持一种一般的信息系统及其配置的操作图。 |
| 常规反侦察 | 一个主要的反侦察目标是降低对手预测我方系统如何工作的预测能力。 |
| 对敌的不可预知 | 一个主要的反侦察目标是降低对手预测我方系统如何工作的预测能力。 |
| 反侦察欺骗 | 欺骗可以使信息系统更难被理解和推断。 |
| 屏蔽ISR和目标搜索 | 屏蔽敌人的ISR干扰敌人预测信息系统架构和功能的能力。 |
| **第二选择** |  |
| 分散 | 分布式系统经常包含一定程度的自理能力和异构性，这使得它们更难被推测。 |
| 暴露度、访问和输出的控制 | 控制措施可以可以是对手更难预测在内部保护的区域系统如何配置的。 |
| 综合管理 | 主动的和计划良好的管理行为能帮助将信息系统的情报传播最小化。 |
| 应急响应体系与计划 | 计划能引起有适应性的备选对象和资源，这使得系统有更少的可预测性。 |
| 防疫 | 重复的红队测试可以帮助系统持续维护，也使其有更少的可预测性。 |
| 检测攻击、损伤评估、敌我识别 | 探测和便利能识别可预测需要修正措施的弱点。 |
| **设计敏感性、薄弱/限制/有限性的促进功能** | |
| 诱骗ISR | 可预测性是一种欺骗攻击者或探测器的杠杆，利用它我们可以知道它们的行为表现和他们知道多少内容。 |

## 安全技术能引发的脆弱性

以下的安全技术没有出现脆弱性警告：

* 屏蔽ISR和目标搜索
* 预防性和报复性的信息和军事活动

表A.20  
异构性可以引起的脆弱性

|  |  |
| --- | --- |
| **首要警告** |  |
| 难于管理和控制 | 各种各样不同的系统类型难以管理、维护和交互操作。 |
| 不自知和不可预测性 | 各种各样不同的系统类型难以监控和预测它们在如何相互影响和操作。 |
| **次要警告** |  |
| 设计敏感性、脆弱/限制/有限性 | 一些异构系统可能会引起设计上的问题或者最小公分母（译者注，是指不可再拆分的最小因素）限制。 |
| 行为敏感性/薄弱性 | 一些异构系统可能引起行为的敏感性或弱点，这源于它们操作上的不同或管理上的挑战。 |

表A.21  
冗余可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 分散 | 冗余系统（尤其是如果分布于不同的地方）可能会被分离或者分别攻击。 |
| 行为敏感性/薄弱性 | 冗余、异构系统可能在“最优”决定没有达成的情况下引起投票悖论（例如委员会形成的决议往往难以折衷）。 |
| 难于管理和控制 | 如果合法程序没有生效以控制他们的交互行为和强制执行合法决策，冗余系统可能不易于管理。 |

表A.22  
集中可以引起的脆弱性

|  |  |
| --- | --- |
| **首要警告** |  |
| 集中 | 按照定义，集中化直接引起集中性问题，所以一定要谨慎执行。 |
| 强度 | 集中系统能变得更坚固和强硬，因为他们倾向于减少创造性的探索和使用各种备用方法。 |
| 可访问/可侦测/可识别/透明/可截取 | 集中使对上更容易定位、探测和识别操作。 |
| **次要警告** |  |
| 单一性 | 集中可能以节约成本的名义造成单一性问题。 |
| 一致性 | 集中化的努力可能有使系统趋于一致的趋向，以此简化管理和节约资金。 |
| 自满情绪 | 一些集中系统变得轻敌，因为它们总是被看做更健壮的。 |
| 易腐蚀性/可控性 | 集中系统可能有被篡夺的控制逻辑和途径。 |
| 可预知性 | 集中化操作趋向于更稳定、可预测、可预定义和较少革新。 |

表A.23  
分散可以引起的脆弱性

|  |  |
| --- | --- |
| **首要警告** |  |
| 分散 | 分散项目更容易分离和分别攻击。 |
| 难于管理和控制 | 分散和分布式系统更难管理和控制，因为它们需要大量C4I协调系统。 |
| 不自知和不可预测性 | 分布式系统的操作难于理解和回溯。 |
| **次要警告** |  |
| 逻辑/现实错误、不可靠性 | 分布式系统中的逻辑和交互式组件会使系统更复杂更易出错。 |
| 设计敏感性、脆弱/限制/有限性 | 分布式系统中的逻辑和交互式组件会使系统更复杂更易出错。 |
| 行为敏感性/薄弱性 | 分布式系统（尤其当它们变得更复杂的时候）会出现行为异常。 |
| 可塑性 | 去中心化和去结构化控制的分布式的创新节点可能需要更严格的管理，因此也更容改变。 |
| 欺骗性 | 去中心化和去结构化控制的分布式的创新节点可能需要更严格的管理，因此也更容易绕过。 |

表A.24  
验证确认、软硬件工程、评估和测试可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 自满情绪 | 的存在能是一个系统的好似用者和管理者感到它已经掌控了关键的脆弱性，因此将变得轻敌，尤其是对新型的威胁。 |
| 可预知性 | 使用标准化的验证确认、软硬件工程、评估和测试（以及他们的报告和文件）能在系统操作上被对少找到规律。 |

表A.25  
暴露度、访问和输出的控制可以引起的脆弱性

|  |  |
| --- | --- |
| **首要警告** |  |
| 分散 | 这些控制往往引起分散性，而且可能被利用以分割系统其它方面的功能部分。这种分散会在改善系统的时候降低整体的性能。 |
| 强度 | 控制能使系统在总体上更坚固而难于快速改变。 |
| **次要警告** |  |
| 集中 | 控制通常是集中的，可能引起其它脆弱点。 |
| 设计敏感性、脆弱/限制/有限性 | 控制能引起限制和敏感性，因为它们的过滤器不是完美的，能干扰正常的通信。 |
| 不可恢复性 | 因为恢复的目的，受限制的通信能是它更难监控和快速访问系统。 |
| 行为敏感性/薄弱性 | 控制能引起限制和敏感性，因为它们的过滤器不是完美的，能干扰正常的通信。 |
| 欺骗性 | 任何控制都依赖使用一种偏离的功能去过滤接口，如果被理解了，那么这种偏离可以被利用来欺骗控制。 |
| 自满情绪 | 被大规模控制的系统通常被认为是安全的，这可能会导致他们忽视自身并非完美无缺。 |
| 易腐蚀性/可控性 | 额外的控制体系通常导致另一个潜在的易被控制和腐蚀的点。 |
| 难以控制或管理 | 复杂的控制体系会难于控制和管理，需要大量培训、联系和相关知识。 |
| 不自知和不可预测性 | 受约束的网文和控制能使监控交互系统环境更困难，并预测到系统如何运行。 |
| 可预知性 | 一些控制系统在行业中是标准化的，因为他们有可预知的限定范围和默认的配置。 |

表A.26  
可信学习和执行系统可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 分散 | 一些信任模式能被分离可信实体虚假信息操作。 |
| 可塑性 | 一些信任模式能被分离可信实体虚假信息操作。 |
| 欺骗性 | 如果偏离功能被对手知道，测量可信行为的模式可能被欺骗。 |
| 自满情绪 | 如果可信系统的限制没有被识别出来并带入脆弱性评估，那么它的使用能引起自满情绪。 |

表A.27  
抗抵赖可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 自满情绪 | 严格的抗抵赖能提供有意义的安全保护，但是信息必须在这个过程中起作用以最大化其价值。 |

表A.28  
加固可以引起的脆弱性

|  |  |
| --- | --- |
| **首要警告** |  |
| 强度 | 加固是系统强度更大。 |
| **次要警告** |  |
| 设计敏感性、脆弱/限制/有限性 | 某些加固的结果是以牺牲性能为代价的。 |
| 自满情绪 | 被加固过的系统可能让人觉得是没有问题的。 |
| 难以控制或管理 | 坚固的、被加固过的系统会难以被控制和管理，尤其是改变环境的时候。 |
| 不自知和不可预测性 | 一些加固方法能使监控和理解系统下一步的发展动态和如何应对它们更加困难。 |
| 可预知性 | 坚固的、被加固过的系统能更容易一个信息丰富的对手被预测出来。 |

表A.29  
错误、不确定性、有效性和质量保证、软着陆可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 设计敏感性、脆弱/限制/有限性 | 有时候系统以一种退化的方法降低故障，这样其他系统就可以很好的操作了。 |
| 自满情绪 | 容错系统可能被看成是无错系统。 |
| 不自知和不可预测性 | 一些容错系统和故障处理方法，很难让人们理解它们是如何工作的。 |

表A.30  
静态资源分配可以引起的脆弱性

|  |  |
| --- | --- |
| **首要警告** |  |
| 分散 | 资源分配可以被利用来攻击或淹没为特殊问题而设立的分区。 |
| 强度 | 静态分配可能对于当前状态不再合适。 |
| 欺骗性 | 对手可能把系统配置改的不是我们想要的。静态分配可能不适应当前环境。 |
| 可预知性 | 如果对方知道，静态分配计划就会被预测到。 |
| **次要警告** |  |
| 集中 | 静态资源分配可能需要集中监控和控制。 |
| 可塑性 | 动态分配触发器可能被主动操作，将系统引导向一个并不期待的配置环境中。 |
| 自满情绪 | 分配计划的存在可能让人的安全感过度。 |

表A.31  
动态资源分配可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 集中 | 动态资源分配可能需要集中控制和监控。 |
| 分散 | 一些分配方法可能被利用来切断某部分与整体之间的联系。 |
| 行为敏感性/薄弱性 | 一些动态资源分配可以和行为敏感性共同出现。 |
| 可塑性 | 动态分配触发器可能被主动操作，将系统引导向一个并不期待的配置环境中。 |
| 欺骗性 | 动态分配可能将系统引导向一个并不期待的配置环境中。 |
| 自满情绪 | 分配计划的存在可能让人的安全感过度。 |
| 易腐蚀性/可控性 | 动态分配控制体系可能被利用。 |
| 难以控制或管理 | 随着选项的增加，动态分配会难于管理。 |
| 不自知和不可预测性 | 在不同的分配方式下，可能不好推测系统如何运转。同样，当分配是自动和快速进行的时候，监控系统状态也不容易。 |
| 可预知性 | 一旦决策标准被知道，动态分配就能被预测。 |

表A.32  
一般管理可以引起的脆弱性

|  |  |
| --- | --- |
| **首要警告** |  |
| 集中 | 许多管理组织有很强的集中性。 |
| 一致性 | 高度管理的组织趋向于一致化，对于要更多消耗管理成本和精力的备用方法、系统和设计并不愿接纳。 |
| **次要警告** |  |
| 唯一性 | 关键管理功能有时跟唯一组件或个人是共生的。 |
| 设计敏感性、脆弱/限制/有限性 | 管理控制能在性能上造成限制和问题。 |
| 强度 | 管理系统比较僵化，难以适应新的环境。 |
| 自满情绪 | 详细的管理流程可能导致人们以为系统被保护得很充分。 |
| 可预知性 | 高度体系化和微型管理的系统会使用许多众所周知的方法。关于这些管理体系的文件如果被盗用，则可能泄露信息。 |

表A.33  
应急响应体系与计划可以引起的脆弱性

|  |  |
| --- | --- |
| **首要警告** |  |
| 分散 | 一些响应架构将系统分割以在高威胁的环境下保护它们。 |
| 强度 | 计划可能过于严谨和僵硬，尤其是如果他们在广泛的地方应用，就不大可能适应每一个场景。 |
| 欺骗性 | 过于严谨和僵硬的计划可能会推动系统进入过度保护的状态，在降低成本的同时降低了性能。 |
| **次要警告** |  |
| 集中 | 一些响应体系和计划运用了集中监控、决策和执行等方法。 |
| 一致性 | 计划可能在广泛的区域命令对象统一响应，而不仅仅在本地有所不同。 |
| 逻辑/现实错误、不可靠性 | 许多计划可能在现实世界中从不完整练习，因而可能包含一些不能预估的错误。 |
| 设计敏感性、脆弱/限制/有限性 | 一些响应行为可能会限制执行一些性能，当他们寻找并保护关键性能的时候。 |
| 行为敏感性/薄弱性 | 许多计划可能在现实世界中从不完整练习，因而可能包含一些不能预估的错误。 |
| 自满情绪 | 有了响应计划就可能导致自满和轻敌的意识，除非经常复查和完善。 |
| 可访问/可侦测/可识别/透明/可截取 | 如果不加小心，计划中的行为可能相当明显，这会传递出系统的信息。 |
| 不自知和不可预测性 | 许多计划可能在现实世界中从不完整练习，因而可能包含一些不能预估的错误。 |
| 可预知性 | 如果响应计划被泄露，那么别人就能轻而易举地预测系统在遭遇威胁和破坏的时候会做出怎样的反应。 |

表A.34  
紧急重建和恢复可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 自满情绪 | 快速恢复和重建初试系统状态的能力（例如重启）能使我们放松对故障和系统危害的警惕，这给我们一个对于系统运行性能的假象。 |

表A.35  
适应和学习可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 行为敏感性/薄弱性 | 对参数的适当探索能临时地引发一些故障，可能在调试好之前会降低一些性能。 |
| 可塑性 | 适应算法，如果被对手知道，可能会被利用误导系统。 |
| 欺骗性 | 适应算法，如果被对手知道，可能会被利用误导系统。 |
| 难以控制或管理 | 如果独立了，适应性良好的系统更难被控制。 |
| 不自知和不可预测性 | 一些适应算法究竟如何工作，对于人而言是难以理解的。 |

表A.36  
免疫式防御系统可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 集中 | 一些免疫系统以来集中化信息和配位点。分散的、点对点结构可以解决这个问题。 |
| 一致性 | 使用这种方法使系统一致化很更容易，它的应用也驱动管理配置更加一致。 |
| 可塑性 | 自动更新补丁为大面积操作信息系统提供一种新的手段，而这必须被高度保护。 |
| 自满情绪 | 尽管有价值，而且看起来很坚固，这些系统并不完美，而且一定会导致在其他安全域内放松警惕。 |
| 易腐蚀性/可控性 | 自动更新补丁为大面积操作信息系统提供一种新的手段，而这必须被高度保护。 |
| 可预知性 | 分享通道可能为对手的情报活动提供了一种途径。 |

表A.37  
防疫可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 一致性 | 使用这种方法使系统一致化很更容易，它的应用也驱动管理配置更加一致。 |
| 可塑性 | 注意模拟攻击不能造成无法挽回的损失，这会提示你的对手。 |
| 易腐蚀性/可控性 | 注意模拟攻击不能损耗你的系统。 |
| 可预知性 | 注意模拟攻击不能提示对手怎么攻击你的系统会更有效。 |

表A.38  
情报活动可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 集中 | 情报信息往往集中协调，统一利用。 |
| 分散 | 情报活动使得个体加深彼此间的怀疑。 |
| 自满情绪 | 情报能力的存在使我们感到比承诺的还要安全。 |

表A.39  
自校正、监视、评估可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 集中 | 监控整个系统需要集中融合和开发能力。 |
| 自满情绪 | 大量无意义的信息和长时间的误导会使得人们麻木。 |
| 可访问/可侦测/可识别/透明/可截取 | 我们的监控可能被我们的对手所利用。 |

表A.40  
诱骗ISR可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 集中 | 有效的探测经常需要协调规划。 |
| 难以控制或管理 | 如果没有识别出来，探测我们自己的系统会迷惑管理者。 |
| 不自知和不可预测性 | 如果没有识别出来，探测我们自己的系统会迷惑管理者和组件。 |

表A.41  
检测攻击、损伤评估、敌我识别可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 集中 | 这些评估需要集中的信息源来促进融合和其他分析。 |
| 分散 | 不确定或错误的探测或结论会导致内部怀疑、分裂、拒绝信息交换。 |

表A.42  
常规反侦察可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 分散 | 过多的恐慌和警告会让人们猜忌，最终导致孤立。 |
| 行为敏感性/薄弱性 | 过多担心毁坏和入侵会使系统陷入偏执。 |
| 欺骗性 | 甚至反侦察的努力是可以被操作的。 |
| 难以控制或管理 | 常规的管理功能和控制也会干涉反侦察的努力结果。 |

表A.43  
对敌的不可预知可以引起的脆弱性

|  |  |
| --- | --- |
| **首要警告** |  |
| 不自知和不可预测性 | 如果没有被识别出来，不可预测性和复杂性会迷惑我们自己的管理制度和组件。 |
| **次要警告** |  |
| 分散 | 过多的恐慌和警告会让人们猜忌，最终导致孤立。 |
| 行为敏感性/薄弱性 | 过多担心毁坏和入侵会使系统陷入偏执。 |
| 欺骗性 | 甚至反侦察的努力是可以被操作的。 |
| 难以控制或管理 | 常规的管理功能和控制也会干涉反侦察的努力结果。 |

表A.44  
反侦察欺骗可以引起的脆弱性

|  |  |
| --- | --- |
| **首要警告** |  |
| 不自知和不可预测性 | 如果没有被识别出来，欺骗行为会迷惑我们自己的管理制度和组件。 |
| **次要警告** |  |
| 分散 | 过多的欺骗会让实体难于判断真伪，引起人们相互怀疑，最终导致孤立。 |
| 行为敏感性/薄弱性 | 当合法的使用者没有意识到的时候，过多的欺骗会引发行为反常。 |
| 难以控制或管理 | 常规的管理功能和控制也会干涉反侦察的努力结果。 |

表A.45  
威慑可以引起的脆弱性

|  |  |
| --- | --- |
| **首要警告** |  |
| 分散 | 强大的威胁和惩罚措施能使系统保守、僵化、谨慎。 |
| 自满情绪 | 强大的威慑力量可能简单的让人们感到系统是安全的。 |
| 可预知性 | 强大的威胁和惩罚措施能耗死系统保守、僵化、谨慎，因此就变得容易推断。 |

表A.46  
刑罚与司法保证可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 自满情绪 | 强大的刑罚与司法保证能引起不切实际的安全感。 |

表A.47  
司法执行和民事诉讼可以引起的脆弱性

|  |  |
| --- | --- |
| **次要警告** |  |
| 自满情绪 | 强大的法律强制力能引起不切实际的安全感。 |

# 参考文献

Alberts, Christopher, and Audrey Dorofee, *OCTAVESM Threat Profiles*, Pittsburgh, Pa.:Carnegie Mellon University, Software Engineering Institute, n.d., www.cert.org/archive/pdf/OCTAVEthreatProfiles.pdf (accessed June 2003).

Alberts, Christopher J., Sandra G. Behrens, Richard D. Pethia, and William R. Wilson,*Operationally Critical Threat, Asset, and Vulnerability EvaluationSM (OCTAVESM) Framework*, Version 1.0, Pittsburgh, Pa.: Carnegie Mellon University, Software Engineering Institute, CMU/SEI-99-TR-017, June 1999.

Alberts, Christopher J., Audrey J. Dorofee, and Julia H. Allen, *OCTAVESM Catalog of Practices*, Version 2.0, Pittsburgh, Pa.: Carnegie Mellon University, Software Engineering Institute, CMU/SEI-2001-TR-020, October, 2001.

Anderson, Robert H., Phillip M. Feldman, Scott Gerwehr, Brian K. Houghton, Richard Mesic, John Pinder, Jeff Rothenberg, and James R. Chiesa, *Securing the U.S.Defense Information Infrastructure: A Proposed Approach*, Santa Monica, Calif.:RAND Corporation, MR-993-OSD/NSA/DARPA, 1999.

Common Criteria, *Common Criteria for Information Technology Security Evaluation—Part 1: Introduction and General Model*, CCIMB-99-031, Version 2.1, August 1999a.

\_\_\_\_\_\_\_, *Common Criteria for Information Technology Security Evaluation—Part 2:Security Function Requirements*, CCIMB-99-032, Version 2.1, August 1999b.

\_\_\_\_\_\_\_, *Common Criteria for Information Technology Security Evaluation—Part 3:Security Assurance Requirements,* CCIMB-99-033, Version 2.1, August 1999c.

\_\_\_\_\_\_\_, *Common Criteria for Information Technology Security Evaluation: User Guide,* October 1999d.

\_\_\_\_\_\_\_, *Common Methodology for Information Technology Security Evaluation, Part 2: Evaluation Methodology*, CEM-99/045, Version 1.0, August 1999e.

Dutch Ministry of Transport, Public Works, and Water Management, and Dutch Ministry of Economic Affairs, *Internet Vulnerability,* July 2001, www.dgtp.nl/docs/intvul.pdf (accessed June 2003).

Gerwehr, Scott, and Russell W. Glenn, *The Art of Darkness: Deception and Urban Operations*, Santa Monica, Calif.: RAND Corporation, MR-1132-A, 2000.

Hamby, Zhi, “What the Heck Is OPSEC?” 2002, at the OPSEC Professionals Society webpage, www.opsec.org/who (accessed June 2003).

International Organization for Standardization (ISO), *Information Technology: Code of Practice for Information Security Management*, ISO/IEC 17799:2000(E), first edition,Geneva, Switzerland, December 1, 2000.

Joint Chiefs of Staff, *Command and Control for Joint Air Operations*, Joint Publication 3-56.1, November 14, 1994, www.adtdl.army.mil/cgi-bin/atdl.dll/jt/3-56\_1/3-56\_1 toc.htm (accessed June 2003).

\_\_\_\_\_\_\_, *Joint Doctrine for Operations Security*, Joint Publication 3-54, January 24,1997.

\_\_\_\_\_\_\_, *DoD Dictionary of Military and Associated Terms*, Joint Publication 1-02,June 5, 2003 (last update), http://www.dtic.mil/doctrine/jel/doddict/.

Kent, Glenn A., and William E. Simons, “Objective-Based Planning,” in Paul K. Davis,ed., *New Challenges for Defense Planning: Rethinking How Much Is Enough*, Santa Monica, Calif.: RAND Corporation, MR-400-RC, 1994, pp. 59–71.

Lewis, Leslie, and C. Robert Roll, *Strategy-to-Tasks: A Methodology for Resource Allocationand Management*, Santa Monica, Calif.: RAND Corporation, P-7839, 1993.Minehart, Robert F., Jr., “Information Warfare Tutorial,” Army War College, 1998,

at http://carlisle-www.army.mil/usacsl/divisions/std/branches/iw/tutorial/intro.htm (accessed June 2003).

Thaler, David E., *Strategies to Tasks: A Framework for Linking Means and Ends*, SantaMonica, Calif.: RAND Corporation, MR-300-AF, 1993.

U.S. Army Communications Electronics Command, *OPSEC Primer*, Fort Monmouth,N.J.: Software Engineering Center (SEC) Security Office, June 27, 1999.

U.S. Department of the Air Force, “Operational Risk Management,” Air ForceInstruction 90-901, April 1, 2000a.

\_\_\_\_\_\_\_, “Operational Risk Management,” Air Force Policy Directive 90-9, April 1,2000b.

\_\_\_\_\_\_\_, “Operational Risk Management (ORM) Guidelines and Tools,” Air ForcePamphlet 90-902, December 14, 2000c.

U.S. Department of the Army, Headquarters, *Army Regulation 530-1, OperationsSecurity (OPSEC)*, Washington, D.C.: U.S. Government Printing Office, unclassified,distribution limited, March 3, 1995.

U.S. Naval Safety Center, “Operational Risk Management (ORM),” OPNAV Instruction3500.39A/Marine Corps Order 3500.27A, July 1997.

\_\_\_\_\_\_\_, “Introduction to Operational Risk Management,” Naval Safety Center, n.d.,www.safetycenter.navy.mil/orm/generalorm/introduction/default.htm (accessedJune 2003).

\_\_\_\_\_\_\_, “Operational Risk Management” (webpage), www.safetycenter.navy.mil/orm/default.htm (accessed June 2003).

Williams, Gary, “Operations Security (OPSEC),” Ft. Leavenworth, Kan.: Center forArmy Lessons Learned, 1999, http://call.army.mil/products/trngqtr/tq3-99/opsec.htm (accessed June 2003).

1. 美国国防部这样定义JFACC：

   指挥官统一指挥, 下属统一命令，联合特遣部队负责建立指挥中心，制定关于合理地调度被指定的、附属的使用派遣空军的建议；计划和协调空中作业；或者在被委派后完成这些任务的执行。统合航空部队指挥官被给予必要的授权去完成指挥官分配的任务。(参谋长联席会议 [2003]) [↑](#footnote-ref-1)
2. 在军事操作中，ATO制定哪一个航空运输工具分配给哪一项任务（例如指挥巡逻编队、指定地点投放物资、为小队或供给品提供运输措施）。 [↑](#footnote-ref-2)
3. 例如，一些入侵检测系统使用“常规”通信行为模型来寻找在软件或设备的特别部位的大规模通信，而这些地方原本可能只有很小的通信量。另一个模型可能就如预期的组件故障率曲线一般简单，这里的数据可以被收集起来以确定非常规的漏洞等级。也有一些其他的模型可以是员工的安全轮廓模型，这种模型可以用来进行背景调查，以帮助识别可能存在的可能导致弱点或问题的员工危害和行为模式。 [↑](#footnote-ref-3)
4. 参见OCTAVE网站[www.cert.org/octave/](http://www.cert.org/octave/)。 [↑](#footnote-ref-4)
5. 细节请在www.commoncriteria.org参见标准和其历史版本 [↑](#footnote-ref-5)
6. 通用标准（CC）是由1980年在美国开发可信计算机系统评估标准（Trusted Computer System Evaluation Criteria ，TCSEC）演化而来。在20世纪90年代早期，欧洲也开发了信息技术安全评估标准（Information Technology Security Evaluation Criteria，ITSEC），它建立在TCSEC的概念之上。1990年，国际标准化组织（ISO）力图开发一套通用的国际统一的评估标准。CC项目开始于1993年，它带来了这些（也包括其他的）成功汇总到一个单独的国际IT安全评估标准。1990年，ISO正式接受通用标准信息技术为国际标准，编号为15408。 [↑](#footnote-ref-6)
7. 参见通用标准（1999a，28页） [↑](#footnote-ref-7)
8. 参见通用标准（1999a，365页） [↑](#footnote-ref-8)
9. 2000年12月12日，第一版 [↑](#footnote-ref-9)
10. 见美国陆军通信电子命令（U.S. Army Communications Electronics Command）（1999）。 [↑](#footnote-ref-10)
11. 美国陆军部（U.S. Department of the Army）（1995）。 [↑](#footnote-ref-11)
12. 整合员工的领导者（Joint Chiefs of Staff）（1997）。 [↑](#footnote-ref-12)
13. 例如，参见美国空军部（2000a，b，c）；美国海军安全中心（1997）；美国海军安全中心“操作风险管理”（网页）。 [↑](#footnote-ref-13)
14. 谅解备忘录。 [↑](#footnote-ref-14)
15. See www.sei.cmu.edu/cmm/. [↑](#footnote-ref-15)
16. 例如，网络可以被重配置以适应新的负载，绕开非认证的通信或者促进基于可用网络管理信息的优先通信。新的配置可以通过启用启发式规则、遍历搜索优先案例、搜索模拟运行模型的全部空间来建立（例如使用探索模型或者遗传算法），用以找到当前条件下良好的在配置方式。 [↑](#footnote-ref-16)
17. 马其诺防线曾经是一战后法国的一项武装防御工事和支持通道，它被设计用来挫败延续到德国边境的潜在物理攻击。由于采取迂回战术绕过防线，并且攻击了其无防御的死角，这个概念证明它是个废物。 [↑](#footnote-ref-17)
18. 查阅Gerwehr and Glenn（2000，Chapter3）回顾常规的欺骗技术。 [↑](#footnote-ref-18)
19. 信息操作也可能在信息站（Information Warfare,IW）中被提及。 [↑](#footnote-ref-19)
20. 例如CERTs包括“CERT 协调中心”（CERT CC）(www.cert.org)，DoD-CERT(www.cert.mil)，陆军计算机紧急响应小组(ACERT)，空军计算机紧急响应小组(AFCERT)。有关的中心包括联邦计算机事件应急中心(FedCIRC) (www.fedcirc.gov)，国家基础设施保护中心(NIPC) (www.nipc.gov)，海军计算机紧急响应小组 (NAVCIRT)，NASA事件响应中心(NASIRC) (www-nasirc.nasa.gov)。 [↑](#footnote-ref-20)
21. 其他技术也被评价为首要使用2，但是没有显示在放大的图中。 [↑](#footnote-ref-21)
22. 集中性既是一种脆弱性，也是一种积极的安全技术。 [↑](#footnote-ref-22)
23. 没有内在的Excel函数被要求在这个原型工具中使用。它简单得被改变，就像一个方便和一直有效的电子表格应用软件，在这当中，要求的算法可以实施。基于Web的工具可能是另一个很有用的执行该工具的平台，它使得可访问性和可用性都更高了。 [↑](#footnote-ref-23)