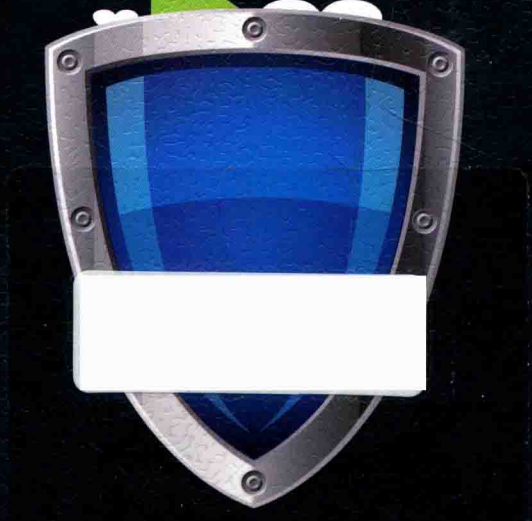
世界著名计算机科学教材

网络技术系列丛书

网络空间和网络安全

George K. Kostopoulos ◎著
赵生伟 ◎译

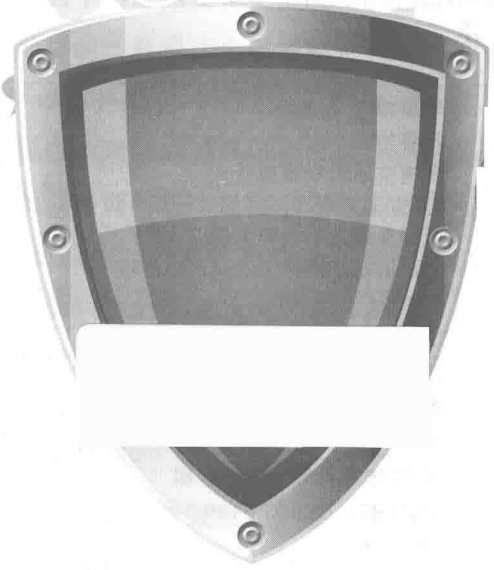


西南交通大学出版社

世界著名计算机科学教材
网络技术系列丛书

网络空间和网络安全

George K. Kostopoulos ◎著
赵生伟 ◎译



西南交通大学出版社
·成都·

四川省版权局
著作权合同登记章
图进字 21-2017-91 号

图书在版编目（CIP）数据

网络空间和网络安全 /（美）乔治·科斯托普洛斯（George K.Kostopoulos）著；赵生伟译. —成都：西南交通大学出版社，2017.5
ISBN 978-7-5643-5199-1
Ⅰ. ①网… Ⅱ. ①乔… ②赵… Ⅲ. ①计算机网络－网络安全 Ⅳ. ①TP393.08
中国版本图书馆 CIP 数据核字（2017）第 010047 号

网络空间和网络安全

[美]乔治·科斯托普洛斯 著
赵生伟 译

责任编辑	宋彦博
封面设计	严春艳
出版发行	西南交通大学出版社 （四川省成都市二环路北一段 111 号 西南交通大学创新大厦 21 楼）
发行部电话	028-87600564　028-87600533
邮政编码	610031
网　　址	http://www.xnjdcbs.com
印　　刷	四川煤田地质制图印刷厂
成品尺寸	210 mm × 235 mm
印　　张	14　　字　　数　222 千
版　　次	2017 年 5 月第 1 版　　印　　次　2017 年 5 月第 1 次
书　　号	ISBN 978-7-5643-5199-1
定　　价	39.00 元

CYBERSPACE
AND
CYBERSECURITY

目录

Cyberspace and Cybersecurity

第 1 章

信息系统中的漏洞

Vulnerabilities in Information Systems

网络空间：从未知之地到无主之地

Cyberspace: From terra incognita to terra nullius

1.1　引言　Introduction

任何系统中的漏洞都可能是由有意或无意的疏忽造成的，或是由一个粗心的设计错误引起的。这个漏洞会直接或间接地危害系统的可用性、完整性和保密性。漏洞可能隐藏于信息安全的各领域：信息访问安全、计算机和存储安全、通信安全以及操作和实体安全。信息系统的主要组成部分是人、硬件和软件，因此必须找出在这三者中存在的漏洞。图 1-1 说明了有助于打造安全网络空间的因素和对网络安全的期望。

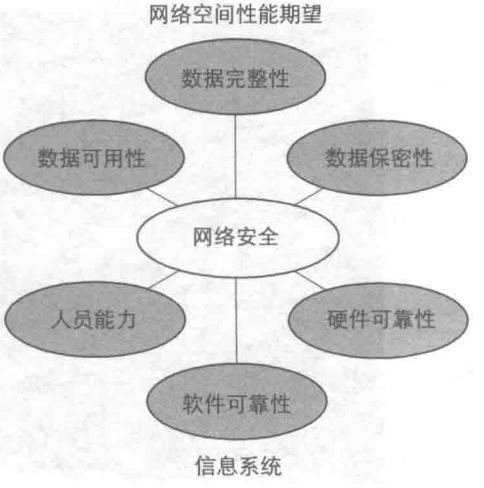


图 1-1　网络安全是网络空间的基础

半个多世纪前，设计师、工程师和科学家们成功地量化了“可靠性”这一概念并将其应用于软硬件的设计和维护中。今天，他们正在努力量化“漏洞”，因为这个抽象的概念适用于信息系统的安全。量化的目的是以可衡量的、标准化的和能理解的方式来诠释安全，并且“通过列举安全数据和能准确传达信息的标准化语言，提高安全的可衡量性，并通过发展知识库来鼓励用户进行信息共享。”[1]

漏洞可以隐藏在数据、代码或者最常见的进程中，不经意地允许了未经授权的访问。然而，入侵不仅会发生在互联网中，也会发生在内网里。内网的安全防范机制通常不完善。通过应用在用户端和服务器端的智能认证机制，可以加强安全。在用户端，如果引进额外的机制，例如一次性密码、动态口令，通过内部或外部额外提供的方法，可以大大强化整个安全认证过程。这些方法可以是生物识别技术、问卷调查，或其他透明的涉及用户设备识别码的参数，比如制造商的序列号——MAC 地址或 IMEI。

MAC 地址：也被称为物理地址，代表媒体访问控制（Media Access Control），是一个 48 位二进制数，以 12 个十六进制数字表示。MAC 地址唯一地标识了计算机的网络接口。网络接口电路可能是可接入网卡，也可能被嵌入计算机的主板中。

图 1-2 显示了如何确定个人计算机的 MAC 地址。

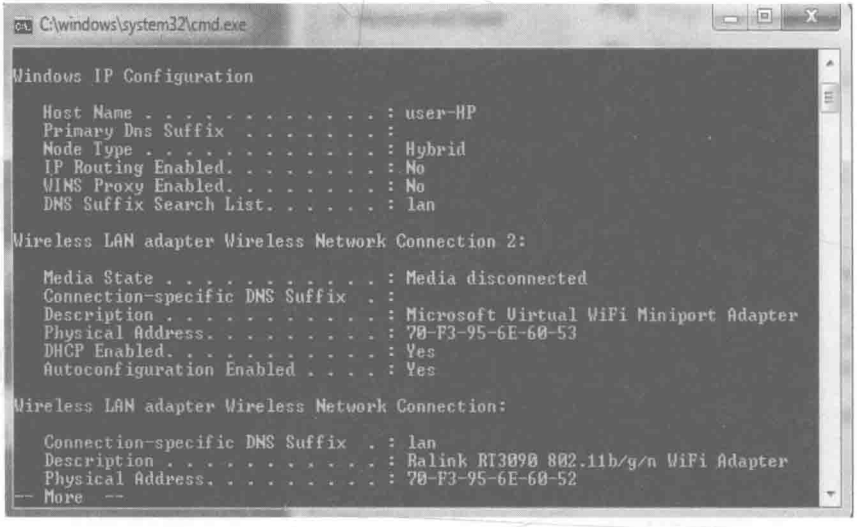


图 1-2 一台个人计算机的 MAC 地址：70-F3-95-6E-60-52

IMEI 的全称是 International Mobile Equipment Identity（国际移动设备识别码），它唯一地标识了移动设备。它通常是一个 13 ~ 15 位的数字。全球移动通信协会分配给每个设备一个 IMEI。图 1-3 显示了手机上的 IMEI。

除了 MAC 地址和 IMEI，设备序列号和网络参数也可以用于认证，如内网和互联网地址。上述方法适用于客户端向服务器端进行身份验证。

图 1-3　手机的 IMEI

在服务器端，证书、IP 限制、数据封装可以大大加强身份认证及安全性。传输过程中的数据可以用哈希码保护，如 CRC（循环冗余码）和私钥/公钥加密机制。

信息系统中的漏洞可能由各种各样的原因造成：从防火墙侵入和木马攻击到静态资源的分配。最常见的漏洞出现在系统正在升级或适应新操作环境时。

1.2　测量漏洞　Measuring Vulnerability

NIST（美国国家标准技术研究所）已开发出一种在软件系统里为安全内容提供标准化分类和评估的协议。该协议旨在“规范安全漏洞和配置信息的标识和编目。”[3]。该协议被命名为 S'CAP（读作“es-kæp”，

意为安全内容自动化协议），由以下六个部分组成：

1. 通用漏洞披露①

这是用来记录已知信息安全漏洞的数据库，其中的每个漏洞都有其独特的标识码。最初，一个新出现的“漏洞”被定义为疑似漏洞，如果为其在 MITRE CVE[4]列表中注册一个条目并最终在 NVD[5]（国家漏洞数据库）中注册，该疑似漏洞就成为正式的了。截止 2010 年夏末，NVD 包含 43 163 个 CVE 漏洞，并以每天 11 个漏洞的速度在增加。在该数据库中，可以发现许多软件（包括著名的操作系统和网页浏览器）的安全漏洞。

2. 通用配置计数器

这是一个类似的数据库，但存储的是在系统配置中发现的安全漏洞和接口不一致信息。这些信息可以帮助系统遵守合规性，确定适当的互操作性和记录核查。其提供的信息是以叙事形式发现存在的问题并通常会提供相应的解决方案[6]。

3. 通用平台计数器②

该协议涉及软件的适当命名并提供一个层次结构。这样可使软件被明确定义，从而大大方便了软件的库存管理[7]。

4. 通用漏洞评估系统③

这是一种和系统软件的开发及使用有关的参数的算法，它提供了一个分数来体现其安全性[8]。由于其所提供的算法没有使用成本，因而被执行风险分析和系统规划的系统设计者和安全分析师们广泛使用。网上有利用通用漏洞评估系统实现开发的算法[9]。

---

① 通用漏洞披露（Common Vulnerabilities and Exposures，CVE）：在国际范围内免费公开使用，是一个公开的信息安全漏洞和披露的目录。http://cve.mitre.org/

② 通用平台计数器（Common Platform Enumerator，CPE）：是一种用于信息技术系统、平台和软件包的结构化命名方案。基于统一资源标识符（URI）的通用语法，通用平台计数器包括正式的命名格式，用于描述复杂平台的语言，用于根据系统检查命名的方法，以及用于将文本和测试绑定到名称的描述格式。http://cpe.mitre.org/

③ 通用漏洞评估系统（Common Vulnerability Scoring System，CVSS）：提供了一个开放的框架来传递漏洞的特征和影响。系统的定量模型确保可重复的精确测量，同时使用户能够看到用于生成分数的潜在漏洞特征。http://nvd.nist.gov/ cvss.cfm?version = 2

5. 扩展配置清单描述格式[①]

它是XML模板，以便于编制标准化的安全指导性文件。这些指导性文件“通过自动化安全工具的规范化的配置内容”[10]介绍软件、特定配置或使用软件的漏洞或安全问题。

6. 开放漏洞评估语言[②]

它“横跨整个信息安全工具和服务（和）标准化评估过程的三个主要步骤”。换句话说，开放漏洞评估语言是系统信息的代表，描述特定机器状态和信息系统。开放漏洞评估语言被企业使用在各种各样的关键功能中，包括漏洞评估、配置管理、补丁管理、政策遵守、基准文件和安全内容自动化[11]。

“安全内容自动化协议（S'CAP）是来自共同体思想的可互操作规范的综合体”[③]

图 1-4 展示了协议的各个部分。在 S'CAP 之外，其他措施在其他方面提供标准化，如图 1-5 所示，其中有通用弱点计数器[④]、通用恶意软件计数器[⑤]、通用弱点评估系统[⑥]、恶意软件属性枚举描

---

① 扩展配置清单描述格式（Extensible Configuration Checklist Description Format，XCCDF）：是用于编写安全检查表、基准测试和相关类型文档的规范语言。扩展配置清单描述格式表示某些目标系统集合的安全配置规则的结构化集合。该规范旨在支持信息交换、文档生成、组织调整、自动化合规测试和合规评分。http://scap.nist.gov/specifications/xccdf/

② 开放漏洞评估语言（Open Vulnerability and Assessment Language，OVAL）：是国际化的、可免费供公众使用的，是信息安全社区努力规范如何评估和报告计算机系统状态的成果。开放漏洞评估语言包括编码系统的语言和整个社区内各种内容存储库。http://oval.mitre.org/

③ 安全内容自动化协议是从社区想法中不断发展出来的互操作规范的集合，旨在满足不断变化的社区需求。http://scap.nist.gov/

④ 通用弱点枚举器（Common Weakness Enumerator，CWE）：提供一组统一的、可衡量的软件弱点，以便更有效地讨论、描述、选择和使用可以在源代码和操作系统中找到这些缺陷的软件安全工具和服务，并更好地了解和管理与架构和设计相关的软件弱点。http://cwe.mitre.org/

⑤ 通用恶意软件计数器（Common Malware Enumerator，CME）：为新的病毒威胁提供唯一的常见标识符，并减少公众对恶意事件的混淆。http://cme.mitre.org/

⑥ 通用弱点评估系统（Common Weakness Scoring System，CWSS）：就像上面的通用恶意软件计数器一样，这个工具是基于一个全面的弱点算法，它考虑到了许多可能会导致该软件出现漏洞的因素。http://cwe.mitre.org/cwss/index.html

述[①]和通用攻击模式枚举分类[②]。

上述标准为信息系统漏洞的评估和报告形成了一个非常强大的基础，因此可以用一个准确、标准、量化和明确的方式来描述漏洞、弱点和恶意软件。使用 S'CAP 技术，可以用一种标准和最终被全行业都接受的方式对信息系统进行安全级别的评估。

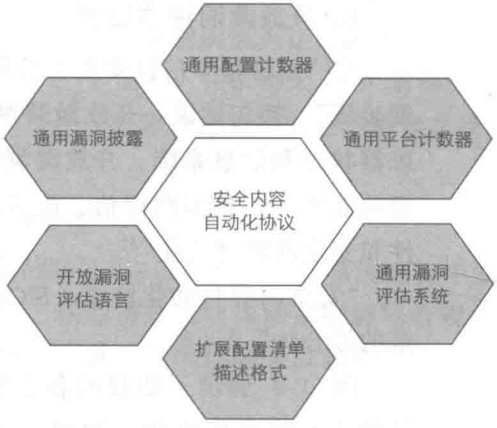


图 1-4　安全内容自动化协议的组成部分（http://scap.nist.gov/）

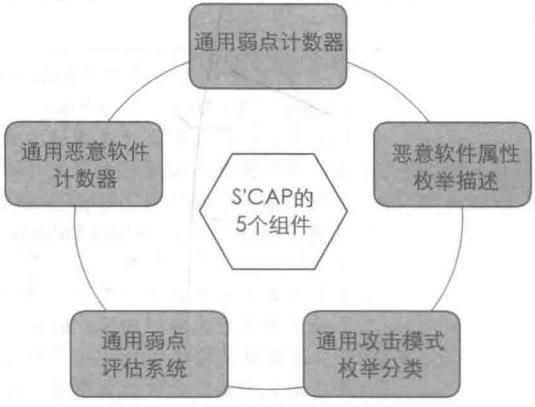


图 1-5　除 S'CAPE 外的五个标准化领域

① 恶意软件属性枚举描述（Malware Attribute Enumeration and Characterization，MAEC）：是一种标准化语言，根据诸如行为和攻击模式等属性，传递关于恶意软件的高保真信息。http://maec.mitre.org/

② 通用攻击模式枚举分类（Common Attack Pattern Enumeration and Classification，CAPEC）：是由国土安全部负责的，是国家网络安全部软件保障战略计划的一部分。这项工作的目标是公开提供攻击模式目录以及全面的模式和分类。http://capec.mitre.org/

1.3　通过安全编码避免漏洞
Avoiding Vulnerabilities throuth Secure Coding

以前，软件开发的设计目标是开发出代码行最少的高效程序，或是执行时间最短的程序，或者是两种的结合。由于内存的高成本和处理器的低速，以前有价值的是内存空间和处理速度。数据完整性的容错率往往是设计师考虑的。因为所有的系统都相互分离，没有联系，所以安全性从来没有得到过关注。

在重要软件的设计中，遵循操作系统设计的原则，通过把代码和数据分配到常驻区和暂存区可使漏洞最少。也就是说，当应用程序被调用时，程序和数据被按需实时调用，而不是调用整个应用程序。通过这种方式，如果有恶意软件入侵，损害将会被限制在已经下载好的（载入到内存的）部分程序中。

今天，一般来说，存储空间和处理速度都不再是开发程序的约束条件。那个时代已经一去不复返了，取而代之的是高速互联的世界。互联网将每一个可连接的设备——从手机到超级计算机——连接起来。这种先进的方式提高了生产力，但往往掩盖了附带的安全风险。专家们坦率地承认："互联网是一个不友善的环境，因此你必须（能够）抵御攻击（以生存）。"[12]因而互联网往往被称为"狂野的"网络。

设计在网络上使用的代码时要考虑到被恶意攻击的可能性，就像设计和建设抵御地震的建筑一样。至于什么是好代码，其标准已经改变，即已从具有最少行数的代码变为有最少漏洞的代码。

希腊有一句很流行的谚语："与其花时间寻找你失去的驴，不如花时间保护你有的。"这个谚语直接适用于互联网软件的设计：与其费心费力地处理被入侵后的后果——昂贵的补丁、负面的影响等，不如多花时间设计安全的软件。换句话说，今天，软件质量和软件安全的概念已深入到各个方面。因此，如果软件需要保护，那么相应的安全机制必须到位。对于网络应用程序来说，现在的趋势是依靠自己的防火

墙而不是仅仅依靠主机系统的防火墙。

如果考虑漏洞所产生的无形损失，修复漏洞的成本通常是很高的。修复期间，许多资源会十分紧迫地从其他任务撤出来并致力于这一修复任务。如果以货币的形式来衡量，修复漏洞的成本以美元计是 5 到 7 位数，具体取决于漏洞的隐蔽性。通常采取以下步骤来纠正漏洞，图 1-6 也说明了这些步骤。

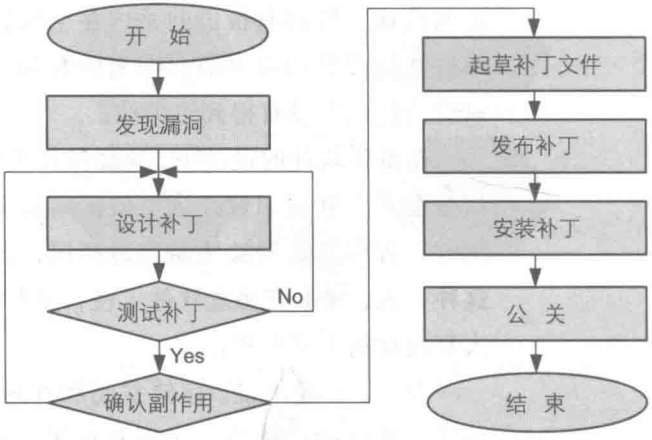


图 1-6 纠正一个软件漏洞的必要步骤

（1）定位漏洞的起源。

（2）设计一个补丁来加强代码和消除漏洞。

（3）应用和测试补丁。

（4）确认是否有副作用。

（5）起草补丁文件。

（6）为每个受影响的客户安装补丁。

（7）安装补丁程序。

（8）确认补丁的有效性。展开宣传活动以抵消之前的负面影响。

除非明确找出了漏洞的源头，否则任何补丁都可能只是掩盖了问题而没有真正消除漏洞的根源。最常见的由不安全代码造成的安全漏洞是：

- 缓冲区溢出。程序给新创建的数据分配有限的空间。除非该程序采取预防措施[13]，否则过多的数据会填满缓冲区，从而造