Hack Proofing
Your Web Applications

网络与信息安全技术丛书

Web应用程序的黑客防范

"The Only Way to Stop a Hacker Is to Think Like One."
洞悉黑客的思维方式，是阻止黑客进攻的唯一途径

附赠
CD-ROM

(美) Jeff Forristal 等著
耿亦兵 秦凯 等译

机械工业出版社
China Machine Press
SYNGRESS

网络与信息安全技术丛书

Web 应用程序的黑客防范

（美） Jeff Forristal 等著

耿亦兵 秦 凯 等译

战晓苏 审校

机械工业出版社
China Machine Press

本书从黑客技术发展、入侵原因和攻击类型入手，深入介绍了在开发 Web 应用程序时存在的安全隐患，着重从代码磨工和移动代码的角度阐述了产生安全问题的实质；通过对 Java、JavaScript、XML、ActiveX 和 ColdFusion 等常见语言的剖析，强调了可采取的安全预防措施；总结了开发安全 Web 应用程序的方法，并指导读者如何制订全面的安全计划。

本书结构合理、内容详实，既可以作为开发人员开发 Web 安全系统的参考书，又可以为网络管理员、系统管理员在管理网络、预防黑客方面提供有效的安全策略，同时也可以作为网络安全和 Web 安全的普及性读物。本书附有光盘，光盘包含原书（本书英文版）中的主要内容。

本书版权登记号：图字：01-2001-4408

图书在版编目（CIP）数据

Web 应用程序的黑客防范/（美）佛瑞斯特（Forristal, J.）等著；耿亦兵等译．－北京：机械工业出版社，2002.4

（网络与信息安全技术丛书）

书名原文：Hack Proofing Your Web Applications

ISBN 7-111-09948-6

Ⅰ.W... Ⅱ.①佛...②耿... Ⅲ.计算机网络－安全技术 Ⅳ.TP393.08

中国版本图书馆 CIP 数据核字（2002）第 013218 号

机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码 100037）

责任编辑：李 欣

北京市密云县印刷厂印刷·新华书店北京发行所发行

2002 年 4 月第 1 版第 1 次印刷

787mm×1092mm 1/16·20.5 印张

印数：0 001-4 000 册

定价：42.00 元（附光盘）

译 者 序

随着计算机网络的普及，越来越多的人已不再对因特网陌生，人们的工作、学习、生活已经离不开网络。在我国，相信经常使用网络的人对计算机病毒也不会陌生，对付计算机病毒的方法也许只会依靠几个杀毒软件。然而可曾知道计算机病毒攻击仅仅是黑客众多攻击方法中的一种。了解黑客攻击技术，能有效地保护自己的 Web 应用程序的人员更是屈指可数。一方面，因特网在我国的发展时间还比较短；另一方面，有关揭露黑客技术，提供给网络管理人员、系统开发人员和广大用户安全策略的相关书籍也十分缺少。随着屡屡爆发的黑客大战和频频发生的黑客攻击事件，保护网络系统和数据的安全，抵御黑客的袭击迫在眉睫。

本书的出版为广大关心网络安全问题的计算机工作者提供了有价值的参考。本书系统地介绍了 Web 应用程序的安全策略，包括存在的安全隐患、预防及补救措施，开发安全的应用程序的方法和指导原则。

如果读者从未阅读过有关网络安全方面的书籍，或者只是浏览过一些书中的个别章节，那么依照本书章节从头顺序阅读将是一个很好的选择。书中循序渐进的内容将会使读者对 Web 应用程序的安全策略有一个系统的认识；如果你已经了解黑客技术并对网络安全有一定了解，你可以跳过第 1 章，继续后面的阅读；对于系统管理员和网络管理员，你可以跳过第 7、8、9、10 章有关分析代码的章节，但对于开发人员或程序员这些章节就不能不看了；不管你是那一类人员，最后两章的阅读都会对你有所帮助。

概括地说，本书与传统的计算机网络安全书籍相比，除了保留系统性、严谨性的风格外，还广泛介绍了黑客简史、黑客攻击过程、各种网络编程语言存在的缺陷等，并将安全计划摆到了突出的位置上。

本书由耿亦兵、秦凯主持翻译。参加翻译工作的有：郭文明、严静东、宋春文、王宁、李景华、赖建荣、苏忠、张少华、李思伟、雷军红、王传芳。全书由战晓苏主审。其他参加翻译和审校工作的还有：张敬松、杨天梁、张志刚、刘世华、邹德新、薛林光、王友军、华书重、刘庶忠、张华勋、徐哲铭、胡向京、武文、夏刚、秦晓周、方林、徐飞、涂卫东、王为聘、李建政。

本书内容新，翻译难度较大，为了使本书中文版尽快与读者见面，译者做了很大努力，但由于翻译时间非常紧迫，加之译者水平有限，错误和疏漏在所难免，恳请广大读者批评指正。

2001 年 11 月

作 者 介 绍

Chris Broomes（MCSE、MCT、MCP + I、CCNA），是 DevonIT（www.devonitnet.com）的一位资深网络分析员，DevonIT 是一家专注于网络安全和 VPN 解决方案的网络服务提供商。Chris 致力于 IT 业八年之久，积累了丰富的技术经验，他是地处 Lansdowne 的网络咨询公司 Infinite Solution 集团公司（www.infinitesols.com）的创始人和首席执行官，同时又是网络设计、系统集成、安全服务、技术编写和培训方面的产品分析员。Chris 精通 Cisco 和 Netscreen 的 VPN 和安全设备，目前，他正在争取 CCDA 和 CCNP 证书。

Jeff Forristal 是 Neohapsis 公司的安全开发主管，Neohapsis 是一家基地设在芝加哥的安全解决方案的咨询公司。Jeff 除了致力于网络安全评估和应用程序安全检查（包括源代码检查）外，同时又是 Security Alert Consensus（安全报警舆论）的核心，这是由 Neohapsis、Network Computing（网络计算）和 SANS 协会出版的有关安全报警通信的周刊。

Drew Simonis（CCNA）是 Fiderus Strategic Security 和 Privacy Services（专用服务）的一位安全顾问。他是信息安全方面的专家，具有安全法则、事故响应、入侵检测和防御、网络与系统管理等方面的经验，精通 TCP/IP 数据网络和 UNIX（特别是 AIX 和 Solaris），熟悉路由、交换和网桥等技术。Drew 曾经参与为 AT&T、IBM 和它们的几个客户开发几个大型 Web 应用程序，其中包括计划和开发了诸如在线银行、自助客户机，以及用于一家大型国家保险公司的在线自适应性保险评估平台。Drew 现在主要是协助客户进行一些网络和应用安全方面的评估，同时也从事着开发方面的研究。Drew 是 MENSA 的一名成员，拥有多种证书，包括 IBM 认证专家、AIX 4.3 系统管理员、AIX 4.3 通信、Sun Microsystems 认证的 Solaris 安全管理员、Checkpoint 认证的安全管理员和安全工程师。他居住在佛罗里达州的 Tampa。

Brian Bagnall（Sun 认证的 Java 程序员和开发员）是《Sun Certified Programmer for Java 2 Study Guide》一书的作者之一。Brian 是加拿大西部的 IdleWorks 公司的首席程序员，IdleWorks 主要是针对大中型企业使用超级计算机，研究分布式处理系统的解决方案。Brian 曾在 IBM 公司开发过客户端应用程序，还是 Lejos（一个 Java 软件开发包）的一名关键程序员。Brian 在此感谢他的家人的支持，尤其是他的父亲 Herb。

Michael Dinowitz 是高访问流量的 ColdFusion 邮件列表 CF-Talk 的主管，其订阅数量超过了 Fusion.com。Michael 主要是给 Fusion Authority Weekly New Alert（www.fusionauthority.com/alert）写作并出版论文，著有 ColdFusion 版本的《FuseBox：Methodogy and Techniques》，是非常畅销的《ColdFusion Web Application Construction Kit》的合作作者之一。无论从他研究 ColdFusion 的深度和广度，还是他呈现给读者的著作中，都很明显地显示出他对这种语言的热爱，除 Allaire 外，很少有人会如此致力于某种语言的传播及其团体力量的增强。

Jay D.Dyson 是 OneSecure 公司（一家可信的数字安全服务提供商）的一位资深的安全顾问，同时是美国国家航空航天局（NASA）的兼职安全顾问。其业余工作包括维护 Treachery.Net，是 Attrition.Org 创建人之一。

Joe Dulay（MCSD）是 IT Age 公司的技术副总裁。IT Age 公司是位于亚特兰大的一家项目管理和软件开发企业，专门提供面向客户的业务经营服务和电子商务解决方案。他目前的责任是管理 IT 事业部，掌管技术咨询委员会、软件体系结构、电子商务产品的管理，负责优化开发过程和方法。尽管他目前的主要工作精力放在管理和总体设计方面，他仍然是开发小组的积极参与者。Joe 从 Wisconsin 大学获得计算机科学学士学位。他的工作背景还包括在 Siemens Energy and Automation 的高级开发人员的职位，他也是致力于电子商务开发的独立承包商。Joe 非常感谢他的家庭一直所给予他的帮助。

Michael Cross（MCSE、MCPS、MCP+I、CNA）是微软认证的系统工程师、产品专家、Professional+I 和 Novell 管理员。Michael 现任 Niagara Regional Police Service 的网络管理员、Internet 专家和程序员，他负责网络安全管理、应用程序开发，是 www.nrps.com 的站点管理员。他曾参与和帮助过与计算机和 Internet 相关犯罪案件的调查，是给超过 800 多名国内用户做技术支持的信息技术小组（Information Technology Team）的成员。

Michael 还主管 KnightWare 公司，该公司主要提供咨询、编程、组网、网页设计、计算机培训和其他服务。他曾经是加拿大安大略湖（伦敦）私立大学和技校的一名讲师，他做过几年自由作家，出版过 20 多本著作和文选。Michael 目前同他妻子 fiancée Jennifer 居住在加拿大安大略湖的 Catharines 大街。

Edgar Danielyan（CCNA）目前就职于自己经营的公司，他毕业于 British Institute of Legal Executive（英国法官学院），获得公司法文凭，又在 University of Southern Colorado（美国科罗拉多州南方大学）取得助理律师资格。他曾经做过网络管理员，是亚美尼亚高级经理，也曾经在联合国、军队、国家电信局、银行等部门工作过，还做过律师事务所的助手。他会说四种语言，喜欢喝好茶，是 ACM、IEEE CS、USENIX、CIPS、ISOC 和 IPG 等组织的成员。

David G. Scarbrough 是美国教育网络的一位资深开发人员，在该组织中他担任 ColdFusion 开发小组首席成员。他主要是致力于研究电子商务网站的开发，David 持有 ColdFusion 4.5 认证，在 HTML、JavaScript、PHP、Visual Basic、Active X、Flush 4.0 和 SQL Server 7.0 等语言的使用方面有丰富经验，他同时还是一名程序员和计算机科学家。David 毕业于 Montgomery 的 Troy State University（特洛伊州立大学），获得计算机科学学士学位。他居住在 TN 的 Smyrna。

前　言

如果程序代码中存在许多安全隐患，但又不能等待审核人员，更不能等待客户从中寻找瑕疵或错误，那该怎么办？本书将帮助你从应用程序开发的最早阶段就开始考虑安全问题。想要代码能够完全地避免遭受恶意的攻击，这是不可能的，但是通过本书的指导和建议，肯定将极大地减少受到攻击的可能性，或者减轻受到攻击时破坏性的扩大。

本书详细地介绍了如下关于防范黑客的关键技术：

- 安全过程的研究、计划、设计和编写都必须具体考虑到各机构，它应该包括网络安全计划、应用程序安全计划和桌面安全计划。所有开发者、管理员和质量保证小组成员都应该参与计划的制定，最终弄清自己在安全过程中扮演的角色。
- 对于应用程序的安全性来说，测试是一个基本的组成部分。通过正反两方面的测试实例，安全性测试应该尽可能地真实，就像真的受到黑客攻击一样。如果安全防御体系十分牢固，黑客费尽时间和精力也达不目的，可使其锐气大减。
- 开发过程中，由于技术快速发展，开发人员必须紧跟上所使用的开发工具的变化和升级，这一点非常重要。很多时候，补丁或新版本工具已经发布，但却迟迟没有使用，这就是因为缺乏忧患意识，或者被其他一些工作给耽搁了。
- 开发人员、Web 网站管理员和网络管理员必须清楚地了解目前存在的来自网络安全方面的威胁，其实，可以通过浏览诸如 www. SecurityFocus. com 或者 www. cert. org 等 Web 站点获得这方面的最新消息。对于开发者来说，这些站点不仅列出了目前关于网络安全一些新的观点，而且还为他们寻求相关安全性建议和解决方案提供了论坛。

安全性应该是多层次的，并且每一层都是很复杂的。在某些情况下，对于一种编程语言是安全的，但对另一种也许就不安全。本书的基本目标是使开发人员了解每一个编程平台所固有的安全问题，并且提供一些合理的解决方法。

第 1 章“黑客方法论”，给出对黑客团体和它的各种动机的基本认识和理解。第 2 章“代码磨工”，讨论了作为一个程序员必须进行“创造性”思考的重要性，解释了没有完全理解代码的用法、功能以至于安全问题就进行开发的危险性。影响创造性和分析性思考常常包括以下因素：由某些管理机构控制和受商业利益（它又会受到体力上和智力上的安全注意点限制）驱动的工作环境、行业标准、对旧技术的依赖，以及成本和最后期限等，这类环境都没有办法进行公开评估和测试。第 3 章“移动代码的危害性”，在用户安全和应用程序前提下，探讨了使用 VBScript、Java Script、ActiveX 控件，以及其他形式的移动代码带来的危险因素。使用这类功能强大的代码时，应用程序的功能和它的真正安全性就会大打折扣。第 4 章“易受攻击的 CGI 脚本”，解释了在 Web HTTP 服务器中使用外部程序而导致的脆弱性。第 5 章“黑客技术

和工具”，展示了在一次成功的攻击过程中，心怀恶意的黑客可能采用的各种不同工具、技术和攻击类型。第 6 章“代码审计与逆向工程”，通过对不同语言环境中的源代码执行逆向跟踪，直至可能产生安全问题的用户输入，针对开发人员怎样才能认识到代码中的安全问题，该章开始了一些实践性的讨论，并提供了一些具体方案。第 7、8、9 和 10 章研究了诸如 Java、Java Script、XML、ActiveX 和 ColdFusion 等语言不同类型的安全问题。第 11 章“开发安全的应用程序”，介绍了 PGP、数字签名、认证服务和为给 Web 应用程序建立可视化安全性的 PKI。最后，第 12 章“制订安全计划指导工作”，指导读者在实施新代码时，如何进行代码复审，它可作为一项保险策略。

——Julie Traxler

目　录

第 1 章　黑客方法论

本章解决以下问题：

- 黑客简史
- 产生黑客的原因
- 当前存在的几种攻击类型
- Web 应用程序面临的安全威胁
- 预防黑客入侵

1.1　概述

你可能听说过 2000 年 2 月，eBay、Yahoo、Amazon 和其他著名的电子商务或非电子商务网站遭受了黑客攻击，那些攻击都是分布式拒绝服务（Distributed Denial of Service，DDoS）攻击，都发生在服务器级别。与此同时，IT 行业和新闻出版单位也受到了同样的攻击。随着黑客事件频繁发生，信息安全专家、项目经理和其他 IT 人士逐步开始意识到问题的严重性，越来越多的公司也开始关注安全问题。同时这又会出现另一种现象，那就是黑客也会变得越来越富有创造性、越来越聪明，相应地，也就增加了网络管理和应用程序开发的难度。

要创建一个安全的防御体系，必须设法弄清楚攻击从哪儿来、攻击者是谁以及为什么要对你进行攻击等问题。通过本书，你将会了解到系统或应用程序随时都可能成为黑客攻击的目标，因此必须对防御体系不断进行测试和完善。如果能够通过效仿黑客的攻击来测试和评估你的系统，这样更容易查出系统的弱点，并有效防止真正的黑客侵入。

黑客也是一个庞大团体，其中既包括一些毫无经验的破坏者（仅仅通过篡改别人的站点来炫耀自己），又包括那些为了获得最大利益而入侵数据库的黑客高手。他们大都臭名远扬。

在互联网世界里，只要提到 Kevin Mitnick，想必都不会陌生。Mitnick 因为黑客犯罪在监狱里服刑数年，虽然成为媒体焦点人物，但在黑客团体中却被认为是替罪羊。

最近，也许是 Mitnick 才使得黑客成为人们关注的焦点，但他肯定不是第一个从事黑客活动的人。目前普遍存在这样一种错误认识，认为黑客攻击是一个相对较新的现象，这种认识在很大程度上是因为最近黑客攻击事件的频繁发生、黑客活动破坏性越来越大、黑客的名声也越来越臭。事实上，黑客随着 Internet 甚至计算机的发明就产生了，正如本章后面讨论的那样，各种破坏代码和电话攻击技术都是黑客最先使用的。

贯穿全书，我们将会提供一些防范黑客的开发工具来保护你的 Web 应用程序，并给出实现安全站点管理、安全代码编写和安全计划实施等目标的基本途径和方法概要，同时为了更好地保证站点可用性、数据保密、数据完整和站点内容等，我们将帮助你“像黑客一样”来思考问题。

名词术语

开始讨论黑客之前，让我们先花几分钟时间理解一下它的含义。有很多不同的术语用来描述黑客，描述的人不同，则其内涵也不同。要了解黑客团体的词汇及其对应的文化背景，可参考术语文件（Jargon File，http：//info.astrian.net/jargon）。

Webster字典恰当地定义了“黑客行为是一种多样性的事物，这类行为包括毁坏性的操作或者投机取巧的方法”；“黑客是一类狂热追求某种作为的人”。同样，IT界也有“黑客”的说法，这些“黑客”并不都心怀恶意，其行为不总是伤害到别人。IT界将他们分为善意的和恶意的二类。定义黑客的一个重要的原则是他们一旦发现弱点后，是否会将它们完全公开地揭露。黑客们也许认为他们自己是白帽子（White hat）黑客，就像好莱坞的“好家伙”牛仔，意思是他们不是恶意的，没必要从事恶意活动；黑帽子（Black hat）黑客们为了个人利益，或者怀有恶意企图，侵入别人的网络或者系统。然而，区别于上述两种，如果从道德规范的意义上（一个人是主观的还是误导的）来定义，又形成了灰帽子（Gray hat）黑客，他们在业内具有强烈的感情色彩，反对将黑客归属为任何一方。不管怎样，黑客们的一个共同特点是，都自认为是“真正的”黑客，都会因为勇于面对新知识的挑战而受到尊敬。其实，“真正的”黑客都有着娴熟的黑客技术，他们认为下面两类人的作为只不过是野蛮人的行为：一是通过使用自己明显不理解的代码从事黑客行为的人（script kiddies）；二是完全为了入侵别人的系统而发动攻击的人（cracker）。

本书中，当提到“黑客”时，使用了它通常的意义，指那些未经许可就擅自进入别人系统或者应用程序的人——不管其目的如何。

1.2 黑客简史

在某种意义上，黑客问题应追溯到20世纪40、50年代，有一些狂热的无线电爱好者，喜欢窃听有关政治或军事的无线电信号。大多数情况下，这些“黑客”只是出于好奇，有些“信息瘾”，他们只会寻找有关政府和军事活动中自己感兴趣的信息片段。当然，这种不允许其他人接收的私有信息频道在毫无察觉的情况下被监听的事情，的确有点令人生畏。

黑客和技术的联姻早在60年代就开始了，当时Ma Bell的电话技术可以很容易地被做手脚，黑客发现了使用免费电话的方法，这些将在下一节中讨论。伴随着技术的进步，黑客的手段也在进步。

麻省理工学院首次在计算机领域引入了黑客这个术语，那时，这个词语仅指那些有天赋、精力充沛的程序员，他们多是自行其是的反叛者。MIT工学院的铁路模型俱乐部（Model Railroad Club，MRC）的成员都热衷于创新，他们拒绝使用DEC公司的PDP-10大型计算机的原装软件而决定自己开发，并展示了其基本架构，称为非兼容的分时系统（Incompatible Timesharing System，ITS）。那时，很多黑客都与麻省理工学院的人工智能（AI）实验室有着一定的关系。

然而，在20世纪60年代，第一条横贯大陆的计算机网络ARPANET，第一次真正地将黑

客们召集在一起；ARPANET 使得黑客们能真正在一起工作，作为一个完整群体而不是遍布在整个美国的一个个孤立团体；ARPANET 第一次使得黑客们有机会在一起讨论共同的目标、共同的构想，甚至出版黑客文化方面的著作和网络通信标准（前面提到的术语文件）。

1.2.1 电话系统黑客

John Draper 是著名的电话黑客，别名Cap'n Crunch。他知道如何利用小孩最爱吃的食品完美地发出一个 2600Hz 的音调，并用它来打免费电话。

在 19 世纪 70 年代中期，Steve Wozniak 和 Steve Jobs（正是他们发明了苹果计算机）曾同 Draper 一起工作，他们对 Draper 都有很深的印象。那时，他们正忙于发明一种攻击电话系统的“蓝盒子”（Blue Boxs）设备。Jobs 的昵称是“Berkley Blue”，而 Wozniak 的昵称是“Oak Toebark”，这两人在早期的电话黑客或者“电话窃贼”中扮演了重要的角色。

Draper 和其他的电话窃贼都会参加每晚的“呼叫论坛”，讨论他们在电话系统中发现的漏洞。为了参加讨论，必须通过双音多频（DTMF，dual tone multi-frequency）的方式拨号，就是现在所指的按键式拨号，电话窃贼所要做的就是通过双音多频拨号，通过一个蓝盒子接入电话线。

当呼叫发生时，这个蓝盒子会产生一个 2600Hz 的语音，它将模拟线路空闲信号，等待路由指令，这时，电话窃贼在被呼叫号码的前面或者后面放置一个键控脉冲（Key Pulse，KP）和启动信号音（Start tone，ST），这将会威胁到路由指令，呼叫将被接受并且不需要花钱。能够访问专用线路是基础，也就等价于能够访问贝尔电话系统。

这里详细介绍电话窃取方式（不仅是进行免费通话）的一个目的是，让人讨厌的窃听场所一旦被发现，就会报告给电话公司。事实证明，在 20 世纪 70 年代，John Draper 因参与电话窃取而多次被逮捕，最终只能在监狱里度过。

黑客/电话窃贼得到过最大的一笔奖赏可能是 Kevin Poulsen 赢得无线电竞赛的那一次，Poulsen 成功地入侵到 Pacific Bell 的计算机中，骗过了无线电通信站正在进行的通话。在这次比赛中，Poulsen 利用高超的技术，阻塞了所有的电话线，以至于 102 位对外呼叫者好像都是他。因为那次杰出的表现，Poulsen 赢得了一辆 944-S2 型蓬式保时捷汽车。

然而，Poulsen 并非只为金钱才进行黑客行动，他涉嫌侵入过 FBI 系统，也被指控侵入过其他政府机构的计算机系统。Poulsen 进入 FBI 系统，通过同对方巧妙周旋掌握了对方跟踪监视的方法。Poulsen 是第一位触犯美国间谍法而被告上法庭的黑客。

1.2.2 计算机黑客

就像前面提到的，在 20 世纪 50 年代，随着第一台网络计算机的出现，计算机黑客活动就开始了。1969 年的美国国家高级研究计划署网络（ARPANET）和此后的美国国家科学基金会网络（NSFNet）的引入，加快了计算机网络应用的步伐，最先通过 ARPANET 连接 4 个站点的分别是洛杉矶的加利福尼亚大学、斯坦福大学、芭芭拉的加利福尼亚大学和犹他州大学，这 4 个站点无意中给黑客们创造了一个更加有组织的合作机会。在 ARPANET 之前，黑客仅能与

位于同一建筑中工作的另外一个人直接通信。不平常的事还不只这些，因为多数计算机的狂热者聚集在大学里。

每一次计算机、网络和 Internet 等新技术的发展，黑客技术也随之进步。其实，推动技术进步的人正是那些从事黑客活动的人，他们了解不同系统的最有效的工作方法。那时，MIT（麻省理工学院）、Carnegie-Mellon 大学和斯坦福大学都站在了的人工智能（AI）领域研究的最前沿，大学里面使用的计算机，常常是美国数字设备公司（DEC）的 PDP 微型计算机系列，在人工智能研究盛行的当时扮演着关键的角色。DEC 公司——商用交互计算和时分操作系统的先锋，提供给各大学机器，不仅功能强大、能够灵活配置，且在那时相对较便宜，这也足以使大多数学校有能力使用他们的计算机。

ARPANET 网络时期，网络上使用的都是 DEC 的机器，其中最为普遍的是 1967 年开始推出的 PDP-10。在将近 15 年里，PDP-10 是黑客们的首选机器。它的操作系统 PDPS-10 和它的汇编器 MACRO-10 回想起来仍旧惹人喜爱。虽然多数大学在考虑计算机配置问题上互相效仿，MIT 却探索出自己的道路，他们使用 PDP-10 系列的机器，但是他们却没有选择 DEC 公司相应的软件，MIT 决定开发一个适合自己使用的操作系统，这就是非兼容分时系统（ITS）。在以后很长时间内，ITS 都一直作为分时系统使用，ITS 虽然是用汇编语言编写的，但是许多 ITS 项目却使用了 LISP 语言。LISP 语言在那时比任何其他语言的功能都强大，更具有伸缩性。MIT 在防止黑客的入侵方面之所以成功，大部分是 LISP 语言的功劳。

到 1978 年，黑客世界中最重要的一件事是一次虚拟聚会。如果黑客们不能聚集在同一个地方，这些最优秀的、最成功的黑客们将如何交流呢？1978 年，Randy Sousa 和 Ward Christiansen 创建了第一个个人电脑电子公告板系统（BBS），这个系统至今仍在使用，它是联合世界范围内的黑客所不可缺少的工具。

然而，第一个单机系统（包括一个独占的 CPU、软件、内存和存储单元）直到 1981 年才由 IBM 发布，称之为个人计算机（Personal Computer，PC）。进入 80 年代后，情况开始有所改变：ARPANET 慢慢地开始演变为 Internet，BBS 一下子盛行了起来。

80 年代末，Kevin Mitnick 首次因为计算机犯罪而被判刑，他秘密地监视 MCI 和 DEC 安全官员的电子邮件但被发现，被判处一年徒刑。与此同时，芝加哥第一国家银行发生了价值 7 亿美元的计算机盗窃案；还有，古罗马军团风暴（LOD）也于此时成立，但其中有一位非常聪明的成员却因同别人争斗而被逐出组织，于是，他决定建立自己的黑客组织——黑客主人（MOD），继而在两大组织之间发生了为期两年的冲突，直到官方干预才宣告平息，MOD 的成员被叛入狱。

为了防止将来类似的事件发生，1986 年议会通过一项名为《联邦计算机诈骗和滥用行为》的法律。在法律颁布不久，政府就对一起重大的黑客案件提起了诉讼。1988 年，Robert Morris 因为编写互联网蠕虫而遭到指控，Morris 的蠕虫破坏了超过 6000 台联网计算机，但 Morris 坚持认为他编写的程序是无害的，只是不知为何失去了控制。此后，黑客攻击事件迅速升温，因为计算机欺诈活动而受到指控的人们被四处围攻，而正好在那时，Kevin Poulsen 撞上了枪口，因篡改电话计费系统而遭到指控。其实，在被捕之前，他已经逍遥法外长达 17 个月。

不论是每天的晚报，还是互联网上的故事，我们都可以看到黑客攻击技术和手段的飞速发展，据计算机安全协会估计，世界财富500强企业中，90%的公司对计算机攻击的忧患大于去年，20%～30%的机密数据受到入侵者的威胁。随着黑客工具和公开实用技术的不断涌现，黑客攻击如此盛行以至于成为商业遇到无法抵御的危险，黑客们因此而感到得意。公司研究防御策略不但可以防止自己成为黑客的目标，而且也保护了其客户，因为对于Web应用程序的很多威胁将会涉及到最终的用户。

1.3 产生黑客的原因

扬名、挑战、无聊和报复仅仅是黑客的一些动机，从事黑客行为本是无可厚非的。其实，大多数黑客只是出于好奇，他们只是想试试他们能看什么，或者能做什么，他们也许没有认识到自己所做事情的后果，但是随着时间的推移，熟练程度不断提高，他们开始认识到自己的作为会带来潜在价值。如果说，从事黑客活动并不全是为了个人利益，但至少离不开个人利益。

但很多时候也不完全如此，黑客入侵大都是为了能够证明他的能力。黑客积聚的知识是力量和声望的组合，在黑客群体里，对于大多数黑客来说出名是最重要的（他们上过法庭之后通常都会扬名）。

对知识的挑战是黑客行为的另一个原因。发现弱点、研究踪迹、查找没有人能发现的漏洞——这些都是技术方面的智力锻炼。在由黑客组织和软件公司等组织的黑客竞赛活动日益盛行时，对于一个渴望挑战的程序员来说，当黑客的吸引力也是显而易见的。

无聊也是黑客行为的另一重大原因。黑客们常常只是四处搜索，想知道他们能访问哪些内容，而找到一个目标又常常会是出于偶然发现一个弱点，这在一个特定的地点之外是不能发现的。

报复性黑客行为则大大不同，这大都因为在某地因某种原因错怪了某人，例如对于被开除的或临时被解雇的员工，为了向从前的老板证明解雇他们是愚蠢的选择。对于大多数公司来说，报复性黑客行为是最危险的黑客攻击方式，因为老员工可能非常熟悉公司的网络和系统的源代码。作为老板，绝对不能等到一名网络工程师或开发人员离开之后才开始担心有人会入侵计算机系统，而在此之前就应该有一个长期的安全计划。

1.3.1 黑客的两面性

试着问问任一开发人员是否有过黑客行为，再问问自己是否曾经作过黑客，得到的答案很可能就是Yes。我们曾经就会因为这个或者那个原因，有过黑客行为：管理员围绕配置故障寻找缺点；专业的安全人员无意（或者有意）中通过后门进入应用程序或者数据库，他们甚至企图采用各种方法来摧毁系统；安全专家模仿黑客入侵网络和应用程序，找出其弱点，然后报告给老板。当然，他们进行的都是善意的黑客行为，他们同意将所有发现反馈给老板，他们还签订了保密协议，用以保证决不将信息泄漏给其他人，但是并非一定要雇佣一个安全专家。在对自己或与别人合作编写的代码进行“极限性测试”时，也会出现善意黑客行为，善意黑客行为的目的就是企图阻止恶意黑客的攻击而取得成功。