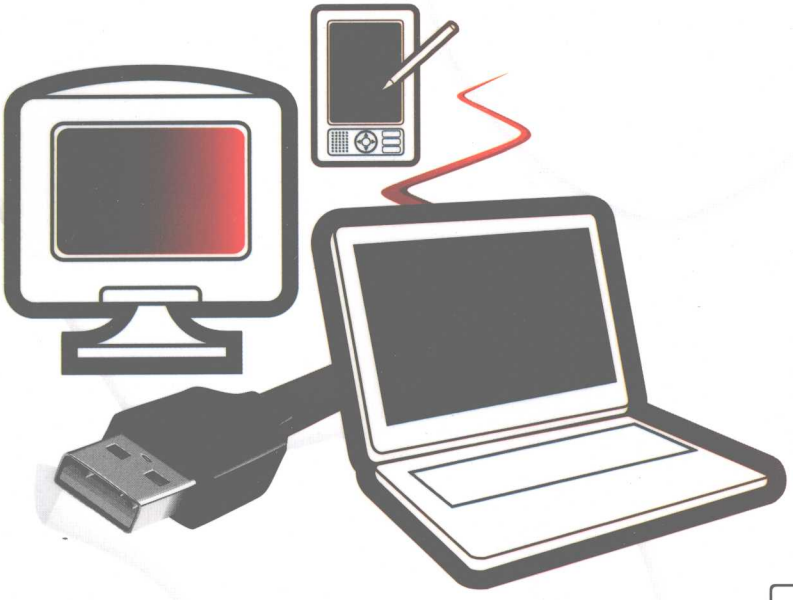
Broadview
www.broadview.com.cn

安全技术大系

ENDPOINT SECURITY

终端安全

【美】 Mark S.Kadrich 著
伍前红 余发江 杨 飏 邹冰玉 等译
张焕国 审校



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

ENDPOINT SECURITY

终端安全

【美】 Mark S.Kadrich 著
伍前红 余发江 杨 飏 邹冰玉 等译
张焕国 审校

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

尽管在安全技术和培训中已经投入了大量人力和资金，但黑客们总是能成功攻击网络中最薄弱的环节——终端。本书作者、顶级安全专家 Mark S.Kadrich 系统地阐述了终端安全是影响信息系统安全的根源这个学术观点，同时提出了以过程控制模型构建网络安全的方法。同时本书也从实际出发，介绍了如何通过过程控制技术来帮助读者保护所有的终端设备，从 Microsoft Windows、Apple OS X、Linux 到 PDA、智能电话、嵌入式设备。在本书中，作者还介绍了许多实际的信息安全技术、软件和工具，对读者有很高的参考和应用价值。

本书特别适合用作信息安全、计算机、通信、电子工程等领域的科技人员的技术参考书，或作为相关专业的教材。

图书在版编目（CIP）数据

终端安全 /（美）卡德里奇（Kadrich, M.S.）著；伍前红，余发江，杨网译. —北京：电子工业出版社，2009.6

（安全技术大系）

书名原文：Endpoint Security

ISBN 978-7-121-08486-7

I. 终… II.①卡… ②伍… ③余… ④杨… III. 计算机网络－终端设备－安全技术 IV. TN915.05

中国版本图书馆 CIP 数据核字（2009）第 035097 号

责任编辑：许 艳

印 刷：北京智力达印刷有限公司

装 订：北京中新伟业印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×980 1/16 印张：18.75 字数：322 千字

印 次：2009 年 6 月第 1 次印刷

印 数：4000 册 定价：49.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

译 者 序

21 世纪是信息的时代。信息成为一种重要的战略资源，信息技术改变着人们的生活和工作方式，信息产业成为世界第一大产业，社会的信息化已成为当今世界发展的潮流，信息的获取、处理和信息安全保障能力成为一个国家综合国力和经济竞争力的重要组成部分。信息安全已经成为影响国家安全、社会稳定和经济发展的决定性因素，如果国家的信息安全受到危害，将会危及国家安全、引起社会混乱，造成重大损失。因此，信息安全保障已成为世人关注的社会问题，并成为信息科学技术领域中的研究热点。

我国政府大力支持信息安全技术与产业的发展，先后在成都、上海和武汉建立了国家信息安全产业基地，目前我国的信息安全产业已经具有相当的规模。2001 年武汉大学创建了我国第一个信息安全专业，目前全国建立信息安全专业的高校已经发展到 70 多所。无论是信息安全技术与产业的发展，还是信息安全人才的培养，都迫切需要信息安全技术书籍。因此，引进国外优秀的信息安全书籍，对传播和交流信息安全知识与技术是十分重要的。为此，电子工业出版社组织我国相关技术领域的学者翻译出版了《终端安全》一书。

本书作者 Mark S.Kadrich 先生是美国信息安全界颇有贡献的著名信息安全专家。他擅长于系统级的信息安全设计、策略制定、终端安全和风险管理。在过去的 20 年里，Mark S.Kadrich 先生先后在多个信息安全企业担任总裁或技术负责人，具有丰富的信息安全技术与管理的实践经验。Mark S.Kadrich 先生还是一位热心的作者，曾编写出版了很多书籍，其中《终端安全》一书就是他的一本新作。

在本书中，作者突出阐述了自己的关于终端安全是影响信息系统安全的根源和通过确保终端安全进而确保网络安全的学术观点，提出了以过程控制模型构建网络安全的方法。他强调应当继续加强对信息系统边界的安全保护，强调操作系统的安全机制在信息系统安全中的基础地位，强调网络安全、软件缺陷与恶意代码对信息系统安全的关键作用。在本书中作者介绍了许多实际的信息安全技术、软件和工具。这些不仅构成了本书的显著特色，而且对于读者也有很实际的参考和应用价值。

《终端安全》是一本难得的好书，书中内容简明扼要且实用，讲述深入浅出、生动有趣、通俗易懂。本书特别适合用作信息安全、计算机、通信、电子工程等领域的科技人员的技术参考书，或作为相关专业的教材。

本书第 1~4 章由伍前红翻译，第 5~8 章由杨飏翻译，第 9 章由邹冰玉翻译，第 10~13 章由余发江翻译。全书由张焕国统稿和校审。

由于译者的专业知识和外语水平有限，书中错误在所难免，敬请读者指正，译者在此先致感谢之意。

译者于武汉珞珈山

前　言

为了明天有一个更加安全的计算机空间，今天信息安全必须向前进。前进意味着我们需要不断创新，富于创造性。过去，当人们相信自己有更好的做事方法，认为自己能有所不同的时候，实际上他们已经进行了创新。本书就是关于安全创新的——也就是求新与求异。

如果我们想继续保护关键的基础设施免遭他人破坏，就需要新的工具。今天的安全世界里不但有黑客和信息盗贼，还存在有组织的犯罪分子、间谍甚至“黑客激进主义者”。每天我们关键的计算机基础设施都有某个部分遭到攻击。一些外国政府正在试图盗窃商业与军事机密；有组织的犯罪分子正在不断侵蚀我们的计算机安全防护，企图侵入某个系统，以便有机会实施真正的破坏；有组织犯罪还在将人们的个人信息转化为可以买卖、利用和交易的商品。

如何创新是在这场战斗中的关键，而创新的一个方面即意味着要用与过去不同的方式考虑事情。我们需要先敌不止一步。

我们需要前进，而且要快。

很长时间以来，人们认为信息安全介于神秘的魔术和艺术之间，而它背后的技术则被视为一门工程学科。电路和芯片都是电气工程世界的一部分；软件被普遍认为属于软件工程师掌控的领域。

微软电子百科全书（Encarta）[①]称一个工程师为：

1．在某个专业工程分支中受过训练的人员。

2．擅长修筑，但偶尔也破坏桥梁、要塞以及其他大型建筑的军队成员。

3．规划、检查或者提出某种事物，尤其是一些独创的或巧妙的事物的人员。

我喜欢这样的定义，因为我们是真正的专业人员。但是，除了是专业人员以外，我们还是工程师。我们从不猜测答案是什么，我们分析和测试；当我们认为已经有了答案时，就采取行动。我们是正在建造保护网络的要塞的人；我们致力于摧毁黑客建造的逻辑堡垒——他们躲在这些堡垒后面攻击我们的终端；我们还是人们隐私的监督者和保护人。

信息安全是一门科学和学科，这个事实本身不言而喻。我们正在逐渐看到这反映到了

---

① Encarta World English Dictionary 1999 Microsoft Corporation.微软电子百科全书，微软公司版权所有，由 Bloomsbury 出版公司发行。

大学的课程中，一些高校正在提供信息安全的硕士学位和博士学位的课程规划。更多的人们在学习我们的工程学科，学习我们用于保护计算机空间的方法和工具。

本书解释了安全问题现在还没有完全得到解决的原因，并提供了一种工程框架，解释如何才能更好地解决安全问题，并获得更可预测的结果。在由众多砖瓦构筑的工程学科基础中，本书可谓又一块砖。在保护计算机空间的战斗中，我们有了另外一种有力工具，使我们能够继续享受虚拟空间的乐趣，并从它带给我们的一切中受益。

——Howard A. Schmidt,
信息系统安全认证专家（CISSP）和信息安全经理（CISM），
R&H 安全顾问（LLC）公司总裁兼首席执行官（President & CEO）

序

那是我迄今为止见过的最棒的飞行——直接飞到消灭你的地方。

——电影《壮志凌云》中 Jester 对 Maverick 语

引言

尽管我们非常努力，网络还是会瘫痪，这是最让我苦恼的事情了。多少年来，我们看到，虽然部署到网络上的安全工具数量一直在不断增加，然而，一旦有某种新病毒出现，我们的计算机环境就会崩溃，这让我们感到非常诧异。我们要问："怎么会这样呢？"我们怎么能够一边花费这么多钱来提高安全性，一边还要感受蠕虫带来的痛苦呢？不是一年感受这种痛苦一两次，而**是无时无刻不在感受它**。

要回答这个问题，你需要做的就是把 vulnerability 这个词键入 Google，然后等待结果。我只等了 0.18s，就返回了超过 6900 万个查询结果。再加上 hacker 一词，额外还需要 0.42s，但确实有好处，就是把查询结果降低到了 420 万多一点。仅半秒钟就获得 400 多万条信息，而且还是免费的！这就是它的价值。

然后回到我们的问题，搜索出来的结果差不多就概括了目前的局面。我们为各种各样的安全漏洞所困惑，我们不断努力，试图挣扎出来。修复安全漏洞的问题是如此之大，以至于发展出专门处理这个问题的一个完整行业。分析漏洞和生成补丁的问题也是如此之大，所以微软将 "根据需要"的补丁发布策略改为每个"超级星期二，补丁日"发布。

他们用这些补丁真正想解决什么问题呢？你可能以为它是关于终端保护的，也就是我们所说的终端安全。这是一个很大的话题，如果回到 Google 输入"endpoint security"，我们会得到超过 250 万个查询结果。再输入"solution"一词，我们可以减少这个高得不合理的查询结果数量，然后就会得到一个更易于处理的查询结果，约有 148 万个条目。

那么，这些补丁有什么意义呢？很多人正在谈论这个问题，但他们是从厂商客户关系的角度来谈的：这种关系可以描述为，他们卖给你一些东西，或是一种解决方案，然后你付钱。纯粹的利润动机促使厂商生产可供销售的产品，营销部门竭力弄清楚人们需要什么，如何包装他们的产品，从而使你相信它们能够满足你的需要。有多少次你回去访问厂商的 Web 页，诧异于它们刚好解决了你的问题呢？看看多少厂商从公钥基础设施 PKI 转向单点登录（Single Sign On，SSO），再转向身份管理。为什么呢？由于费用太高，没有人购买公

钥基础设施，所以营销部门决定换个名字，或者重新阐述产品的目的。然后他们的产品就综合了多种用户资格证书，并许诺将简化用户的参与。当这个策略也失败了的时候，营销人员发明了身份管理。这就是我所说的——“他们发明了身份管理”，这样他们就可以再一次和一个失败的市场策略保持距离，让更多的人给他们更多的钱，获得更多的利润。

问一下任何一位首席执行官，他或她的任务是什么，如果这个首席执行官回答的不是“最大化股东的收益”，那么我要向你指出，这个首席执行官很快就要找一份新工作了。一切都是为了获得销量，赢得利润。利润越多，厂商和股东就越高兴。

不要误会我的意思，利润是个好东西，它可以使我们的机制运转，能激励人们。然而，当一种机制能带来利润却不能带来好的解决方案时，我们必然要问：“我们这里要解决的真正问题是什么呢？”我可不想这个答案的内容就是如何最大化股东的收益，我希望答案的内容是，理解一组明确定义的准则，确保企业以及它生成的信息是可靠的、可信的和安全的。

但是，基于某种奇怪的商业方面的原因，我们似乎看不到厂商会这么做。

本书认为，如果我们多年以来都在做同样的事情，却一直失败，那么我们一定有什么地方做得不对。有关我们在做什么和为什么要做的一些基本假设是不正确的。是的，**不正确**。然而我们还在继续那样做，好像什么问题都没有。痛苦就在这里，更加严重的是，现在这种痛苦无处不在，以至于我们都变得麻木了。像荧光灯发出的嗡嗡声（是的，它们的确发出令人讨厌的声音）或电视暴力一样，我们已经习惯于它在周围，我们甚至还发展出了应付它的处理机制。为什么没有人首先问一问，痛苦从何而来？

本书就做这件事。

本书的特别之处在于，它借用了一种基本科学的思维来理解究竟问题是什么，以及如何解决它。为什么说在解决网络污染和感染问题时，保护终端是你所能做的最明智的事情，本书使用了过程控制模型对此进行解释。同时你还会了解保护终端安全和其他层次安全的不同之处。我们首先讨论每种终端的基本工具和设置；然后转向讨论终端所需要的安全工具，例如反病毒工具；接下来讨论用额外的安全协议和工具升级终端，如 802.1x 协议和客户端认证协议 supplicant，这样就实现了闭环过程控制 (closed-loop process control，CLPC) 模型，实施了最低限度的安全等级。

目标读者

如果你是一个安全部门经理、安全管理员、台式机支持人员，你正在或即将管理、响应、负责网络的安全问题，那么本书适合你。如果你的工作需要的不仅是确保网络“连通”，

而且要网络成为具有生成、共享和存储信息的功能的工具，那么你要阅读此书。如果你因为某个脚本小子[①]接入了首席执行官的笔记本而被解雇，你需要读这本书。如果你担心隔壁小房间的 Barney 下载最新的“免费”视频剪辑软件或很酷的最新聊天客户端程序，你会需要购买本书并赠给你的台式机管理员。

目的

许多书籍讲述了如何利用系统漏洞，或者说如何发现安全漏洞，让系统所有者惊慌失措。如果你要找一本有关黑客攻击的书，本书不是。如果你就是想进行黑客攻击，这本书也不对。将它给你的网络管理员朋友，我肯定在你得到有关黑客攻击的书籍的时候，他会需要这个。所以说，本书不是从“黑客视角”观察安全问题，而是给你一些更有用的东西——信息安全行业实践者的看法。

本书不仅告诉你去探索什么，还告诉你为什么要探索。是的，本书有些地方是一步一步的引导，但是本书遵循格言：“授人以鱼，一日之食；授人以渔，一生之食。”虽然这句话已是陈词滥调，但表达的意思却非常深刻。

本书教你如何配置网络，通过从它的“根”——也就是终端上解决安全问题，使其更安全。

这本书同样也注意了怎样做才有望不会再犯同样的错误。有些批评者可能不喜欢第 2 章，因为我把造成很多困境的部分原因，直接归罪于厂商了，我们的解决方案可是他们精心设计的。是的，还存在一些开放源代码的安全工具，但它们没有推动我们的安全市场。

希望你读完本书后，能够理解为什么我相信闭环过程控制模型可以解决安全问题，明白如何将它应用于你日常的安全解决方案之中。

被忽略的编辑问题：“我们”和“他们”

编辑是一群了不起的人。许多作者讨厌编辑，因为他们更改作者洋洋洒洒的文字，那可是作者花了很多时间构思润色的。他们重新阐述作者所说的东西，改变词序或语气，从而改变向读者表达思想的方式。有一些作者讨厌这样，我不是。我是新手，而且是个懒惰的新手。对一个作者来说，这种情况很糟糕，因此我不介意建设性的批评，通常如此。

“我们”是一个简单的词汇，当一个作者使用它的时候，是想在读者埋首书页时暗示作者和读者之间亲密的关系。当一个作者说“我们”时，应当指的是和书中情节有紧密联系的

① 黑客老手对年轻新手的称呼，译者注。

那部分读者，但有时候“我们”也并不包括读者“你”这个第二人称。例如，作者可能会用“我们”表示不包含读者的一个群体（像在“我们侵入计算机，寻找儿童色情内容的证据”这句话中一样），读者在这时候显然没有包含在“我们”这个群体中。

那么，为什么在一本关于终端安全的书一开始，我就要提出这个问题呢？因为我犯了一个错误，全书很多地方都使用了“**我们**”一词，却没有每次都解释“我们”是谁。我原以为“我们”指谁是显而易见的。

我的编辑不喜欢这样，她很礼貌、很委婉、轻描淡写地提到这个问题。尽管如此，她还是不喜欢。

每次编辑校对一章并返还给我时，“**我们**”一词都被突出强调，并附上一个礼貌的注释问：“我们”指的是谁。“Mark，我们是谁？请告诉我‘我们’是谁。”是的，在我用到“**我们**”一词的每个地方，都有一个强调标记和注释。这让我很烦，因为我原以为这是很清楚的。因此，为了努力寻求最终答案，我请了一个权威——我的女友 Michelle——阅读我构思的洋洋洒洒的文字，希望她和我意见一致。我本应该更清楚她的反应。但她也问道：“‘我们’是谁呀？”这并不是我所期待的反应，因此我所能做的就是茫然地看着她，结结巴巴地说：“噢，嗯，我们就是我们！”

我感觉自己像个白痴。她的表情证实了，我就是一个白痴。

但“我们”的确是我们。我们是世界上试图解决一个巨大问题的安全人员。因此，当我在本书中谈到“我们”时，我是在指所有尝试过和正试图创建安全可靠的网络的那些人。

现在，我相信“他们”接着要登场了，因此让我在这里解释一下。“他们”就是他们，是指除我们之外的那些人。厂商是主要的“他们”，是我说“他们”的时候通常所指的人。

所以，“我们”和我们是好人，“他们”和他们，嗯，不是。

我们为什么要这样做

正如我前面说的，如果你做某件事情，不管尝试了多少次，你总是不成功，那么必定有什么做错了，该后退一步，尝试弄清楚为什么不成功。过去的方法不起作用，就应该试一下新的方法。保护终端安全不是什么新思想，实现终端安全的方法也众所周知。但是，我们做了大量的研究，似乎表明如果不把终端作为安全规划的关键组成部分和实施要点，那么你注定要失败。

是的，“注定”似乎有点刺耳，但是如果你被解雇了，原因是某个狡猾的家伙修改了某种病毒的两个字节，攻破了你的网络，用什么词有什么区别？你完了，而完了只不过是注定的过去时。

关于作者

在过去20年里，Mark S.Kadrich是信息安全界颇有贡献的一员。他擅长于系统级设计、策略制定、终端安全和风险管理。Kadrich先生已经出版过很多书籍和刊物，是一个热心的作者。

Kadrich先生目前是The Security Consortium（TSC）公司的总裁兼首席执行官，这是一家私人控股公司，其任务是向客户提供更好的安全产品知识。TSC深入地测试并评价安全产品及其厂商。作为首席执行官和主要推广人员，Kadrich先生负责确保公司能持续增长。

Symantec公司收购了Sygate公司之后，Kadrich先生担任了Symantec公司的网络与终端安全高级经理一职。他的职责是：确保Symantec公司的业务部门在追求创新的技术解决方案的同时，能正确理解安全策略。

在Symantec收购Sygate之前，Kadrich先生是Sygate的资深科学家。在担任资深科学家期间，Kadrich先生负责制定公司策略，把握未来安全趋势，设法让公司规划通过政府的审批并根据需要推动规划。作为Altview公司的创始人之一，在Sygate收购这家创业公司的时候，Kadrich先生加入了Sygate。

Kadrich先生是Altview公司的创始人，作为主要架构师，他负责设计了一个对网络及其终端进行扫描和场境化的系统，并建立了详细的知识库。该系统最终叫做Magellan系统，它能确定网络上有什么终端、网络在如何变化；如果有终端要管理的话，该系统还能确定什么样的终端易于管理。

作为LDT Systems公司的首席技术官（CTO）和首席安全官（CSO），Kadrich先生协助开发和维护了一个基于Web的系统，用于可靠地获取和跟踪器官捐赠者的资料。

Kadrich先生曾是Counterpane互联网安全公司的技术服务主任。他负责拟定方法，以培养和提高Counterpane公司在与客户相关的安全活动方面，部署产品和提供服务支持的能力。

Kadrich先生担任过Conxion公司的安全主任。作为安全主任，其职责是规划Conxion信息安全解决方案的战略方针。

在担任Conxion公司的安全主任之前，他是International Network Service（INS）公司的首席顾问，创建了一种安全评估的方法，并向业界主管解释正确实施安全规划的好处。

Kadrich先生是一个信息系统安全认证专家（CISSP），持有Phoenix大学管理信息系统的学士学位，并有计算机工程和电气工程学位（Memphis，1979年）。他的论著发表在《TCP内幕》、《出版杂志》、《全球信息技术》、《RSA》、《CSI》，以及《黑帽简报》等杂志和出版物上。

致　谢

我首先要感谢的是 Jessica Goldstein，感谢她倾听一个可能听起来就像疯子一样的人谈论信息安全和安全理论。她的学识、帮助和指导对完成本书具有不可估量的作用。

我要感谢校阅小组的朋友们，你们应该得到感谢，你们从百忙之中抽出时间仔细阅读初稿、增加注释并更正错误。我有一个很棒的小组，包括 Dan Geer、Curtis Coleman、Rodney Thayer、Debra Radcliff、Joe Knape、Kirby Kuehl、Jean Pawluc，Kevin Kenan 和 Harry Bing-You。当我开始慷慨激昂的时候，Dan 和 Rodney 就提醒我要注意了。Kirby 和 Joe 担负起了技术编辑的工作，确保我提供的事实（和我关于 NetBIOS 的文献）正确无误。Curtis 和 Deb 提供了商业角度的见解，提出了一些经理们可能想知道答案的问题。Jean、Kevin 和 Harry 重新仔细检查了他们的工作成果并给出了最终意见。

我还要感谢 Addison Wesley 出版社的编辑们：Sheri Cain（做了大量最初的编辑工作）、Jana Jones、Kristen Weinberger、Romny French、Karen Gettman、Andrew Beaster 和 Gina Kanouse。Kristen 和 Romny 主要负责管理我和项目，确保我完成项目的时候不会超过截止时间太远。不能忘了 Keith Cline，他是这个星球上最好的文字编辑。由于他的原因，看来好像我确实通过了语法课。

我特别要感谢 Howard Schmidt 为本书写下前言。当 Howard 回到家里时，发现由于冬季暴风雪他家停电了，他证明了手持设备能做的不只是接听电话和播放音乐。

最后我想感谢我深爱的女友——Michelle Reid。她忍受着我一边咒骂一边工作，还在我确实想放弃的时候安慰我。当我需要用通俗易懂的话解释一些技术性东西的时候，Michelle 作为顾问的价值也是无法衡量的。

我向这里的每一个人都表示由衷的感谢。由于他们的贡献和启发，使本书增色不少。没有他们，我不可能做得到。

Mark S.Kadrich

漫谈Fuzz测试技术

——软件安全专家王清谈Fuzz技术与软件安全性测试

Part 1 Security Testing Overview

我相信大家对测试不陌生，但是对安全测试可能有一些疑问。安全测试关心的问题是不一样的。这种测试也被人叫做工具测试、渗透测试、攻击测试、测试产品安全性。

这种测试需要首先知道各种各样的攻击的方式和原理，在攻击产生之前尽量多的找到产品的漏洞，尽量多的发现产品里面的问题，再努力把它解决掉。

What is the difference between security testing and traditional function based testing ?

这两个之间的差别我们可以用一个简单的东西来说明，Targeted towards making sure that the app does what it is supposed to do。

For Functionality testing：我设计这个产品是1、2、3、4、5，我得让它能干到1、2、3、4、5才行，不是说是1、2、3、4，少1份干不了。这是一般的出来的结果也是一二三四五。

For Security testing：是说我设计了1、2、3、4、5，我得确定你干出来的不是1、2、3、4、5、6、7，还多出来2个。大家都知道你用word去解析一个文件，看一个文档的时候，是一个代码，具体怎么发生大家可能比较清楚的。

一般来说，重点是通用的漏洞、Memory问题、堆溢出、栈溢出，还有各种各样的溢出，SQL、XSS和各种各样的validation。被黑客熟悉的除了我们耳熟能详的漏洞之外，还要关注其他的一些东西，比如说Access Control、information、加密方面的问题、认证方面的问题。一些软件安全工程师经常用加密技术，但是用错或者是设计上有一些东西出现错误。

Security Testing一般的方法就是以下这几种：

1.White BOX：我们一般会用静态的源码扫描工具。通过对源代码的扫描，我们可以把源代码的某一个函数，某一个文件，某一行用了哪些不安全的东西，有哪些漏洞，有哪些缓存区域的漏洞等等，这些东西都能扫描出来。通过扫描把这些漏洞全部做出来以后，从源代码方面能够保证基本上保证常见的漏洞不会出现。

2.Black BOX：首先要灵活，会有很多的方法，比较好的有Injection Fuzzer和Dumb Fuzzer，后面会介绍它们之间的区别。

3.Gray BOX

Part 2 Smart Fuzz Test

Fuzz这个名词来自于Professor Barton Miller。在1989年一个风雨交加的夜晚，他登陆一台自己的主机，不知道怎么回事，信号通过猫传到主机上，雷电一闪，把里面的高位变低位，低位至高位了，结果到了主机以后改变了。他突发奇想，把这种方式作为一种测试的方式来做。

1、到底什么是Fuzz Test?

Generally speaking fuzz is a brute force method which used to break software，就是用大量的测试用例一个一个试，尽可能多的找出有可能出问题的地方。

2、Fuzz怎么工作?

现在有无数有名的Fuzz工具，有很多人很多还在写，一般包括四个部分。

（1）Generate lots of malformed data as test cases，要生成大量的测试用例。这个测试用力是malformed的，一个软件首先要找到输入点，然后把数据丢进去，这个数据有可能是一个文件，有可能是一个数据包，有可能是测试表里面的一个项，有可能是临时文件里面的一个东西，总之是一种数据，要定义malformed这种非正常的数据。

（2）Drop the test cases into product，把它丢进去，看这个产品怎么反应。

（3）Monitor and log any crash/exception triggered by malicious input.

（4）Review the test log, investigated deeply.

3、Injection Fuzzer和Dumb Fuzzer的区别

下面我们说一下Dumb Fuzzer和Intelligent Fuzzer之间的区别。

Dumb就是哑的，就是笨的意思。刚开始Fuzz的时候用的都是这种东西，直接把非法数据丢到软件里边去，这种东西很难真正测试出问题，因为很难把问题放到缓冲区去。比如丢一个Word，不能随便写一个文件，这就很难测试到真正问题，必须基本符合Word本身的文件格式，才有可能测试到结果。要考虑到软件本身执行的流程，你的case放进去，能够放到多深，逻辑放到多深，你要考虑这个问题。你要写这种程序的话，就要非常了解要测试的文档结构。

什么东西可以被Fuzzed？文件格式可能出问题，数据包也可能出问题。因为数据包在解析数据包的时候也是一个状态机，放进去以后再看下一步到哪里去。抗组件这些东西也是相当容易出问题的。这么多东西，都是可以被Fuzz的。当然有一些东西也是可以被Fuzz的，我们现在说的集中是软件产品，硬件也可能有。

4、几个Fuzz工具

简单介绍几个Fuzz工具，因为Fuzz的工具特别多，下面结合我在工作中的用到的一些东西，用最简单的方式介绍一下。

（1）Com Raidor

对于ActiveX，一般来说，我们用的都是Com Raidor，它是非常简单，而且是免费的，它是Idenfense的产品。你只要选中，它里面有很多的项目。因为很多问题是针对IE的，它就会生成测试用例，会把脚本拿出来，一个一个的丢到IE里跑，接下来一直是这个状态。很简单，相信点几下鼠标就可以做。大家如果在测试的时候，其实跑一下也不费什么力。

（2）Fuzz网络协议——SPIKE

SPIKE第一个提出格式化Fuzz的概念，而且提供了免费的、开源的工具，它发现了很多著名的漏洞。如果我要用SPIKE　Fuzz一个文件的话就不是很简单了。Fuzz的时候，你要保证数据结构基本是正确的，不用完全对，完全对的话就不用Fuzz了。改一些小的地方，一次改一点点，一次改一点点，一次改一个地方或者几个地方，不要改太多，这样一丢，才能测试到东西。如果改的太多，最基本的条件没有满足，进去以后它就会走到其他分值区。这是一个结构化的最基本的原理。是要注重数据之间的结构，在这种情况下才能比较准确的找到问题。

协议一般都有状态，第二个包过来，第一个包回去，第三个包过来，第四个包回去，可能是第三次组合的东西某一项才会触发漏洞。但是如果你从第一个包就开始胡乱的发，也许根本进行不到第三个包去了。所以说，协议Fuzz的文件要更麻烦一点。你要想把它送的深，送的远，送的逻辑远，覆盖的逻辑深，把东西丢到程序里面深一点，你得思考程序到底是怎么样执行的，你得想它控制的东西不是一个重复的。我想它已经不是一个纯黑盒测试了，要考虑程序里面是怎么运行的，还要考虑到程序的具体逻辑。