Rational. software

IBM

利用 IBM Rational AppScan 改进 Web 应用软件开发生命周期的安全性

面向

企业领导者 **RedGuide**

Federik De Keukelaere
Danny Allan
Axel Buecker

- 了解攻击者如何选择目标并将攻击转变为金钱
- 了解自动化 Web 应用安全性测试的价值
- 在您的开发生命周期中部署 IBM Rational Web 应用安全性工具

Redbooks

执行概要

互联网上的黑客已经从单纯为出名而进行的破坏过渡到欺诈，又过渡到了有组织、有利可图的数据和身份窃取。伴随着这样的发展演化，企业领导者必须将其 Web 应用的安全性视为业务成功的重要绩效指标。

在这份 IBM® Redguide™ 出版物中，我们分析了您的组织应如何评估黑客入侵系统的风险。我们还说明了您的组织如何实现安全性测试并整合解决方案，从而改善安全性，保护您的信息资产。

在这份 Redguide 的第一部分中，我们将讨论如何评估您的组织面临的风险。我们将解释您的组织为什么会成为攻击的目标，以及幕后的攻击者。我们将演示成功的攻击可能会给您的组织带来的影响。我们将展示关于 Web 应用漏洞的最新趋势和统计信息以及被窃信息的幕后交易情况。我们还会给出应用可能会受到攻击的领域的技术概览，讨论两种最常见的 Web 应用漏洞。

在这份 Redguide 的下一部分中，我们会介绍 Web 应用的软件开发生命周期，展示如何将安全性纳入这样的生命周期。我们提供了将 Web 应用安全性测试整合到您的软件开发生命周期之中的详尽方法。此外，我们还展示了在软件开发生命周期中利用 IBM Rational® 产品的方法与时机，以根据业务需求改进组织的安全性。

最后，我们以一个业务场景总结了这份指南，在此场景中，一个未使用任何 Web 应用安全性测试的组织逐步转变为交付优质、安全的产品的组织。

揭示 IT 攻击模式的基本要素及其对组织的影响

针对 Web 应用的攻击数量不断增加，这一事实对 IT 企业内的大多数人来说已经不再新鲜。成功的攻击会受到媒体的广泛关注，同样也必然面临着牢狱之灾。对您的组织来说，最重要的是理解未来可能遭遇的风险，以便采取恰当的措施，规避和控制此类风险。

了解以下问题的答案可帮助您实现此目标：

- 我的组织为什么面临着被攻击的可能性？
- 我的组织中存在能被利用的漏洞的几率有多大？
- 如果组织被成功攻击，将会遭受怎样的损失？
- 我怎样才能更好地保护我的组织？

遗憾的是，业余爱好者、大学生或黑客仅仅出于乐趣而入侵企业信息系统的时代已经一去不复返了。如今的攻击者的经济动机更强。他们往往是国际性的犯罪团伙，通过窃取财务信息和身份谋生。当今的威胁比过去的安全威胁更加复杂也更加危险，但从某些方面来说，可预测的程度也更高。业余黑客可能对存在的任何安全漏洞都有兴趣，但真正的计算机犯罪分子对能够提供较高投资回报率的漏洞更感兴趣。简而言之，一切都与金钱有关。

IBM Internet Security Systems X-Force® 研究团队发表的关于 Web 应用安全漏洞的研究显示，在过去几年中，已报告的 Web 应用漏洞在所有报告的安全漏洞中所占的百分比显著增加 [1]。您的组织中存在的这些漏洞被利用的可能性取决于利用漏洞的复杂度。由于现在的攻击者通常是以盈利为目的的组织，因而大多数攻击活动都是自动完成的，这是为了尽可能地降低成本。因而，如果您的 Web 应用能够通过使用自动化的工具轻松攻击，被利用的几率就较高。

在您的组织被成功攻击之后，恶意的攻击者能够进行多种破坏。除了因数据损失而造成的明显损失之外，您还可能面临着信用卡公司征收的高额罚金、通知信用卡持卡人的高昂费用、因负面公开而造成的严重品牌受损，也可能会卷入对您的企业发起的民事诉讼。对于大型企业来说，此类损失很快就会达到数亿美元 [2]。

幸运的是，将 Web 应用安全性提升到一定的水平，使其在经济角度上不再是可行攻击目标，这并非不可能的任务。在这份 Redguide 中，我们将提供这些问题的答案。通过将 Rational AppScan® 产品整合到您的软件开发生命周期之中，我们将为您展示您应如何保护组织免受当前面临的众多威胁的侵害。

了解攻击者

不同类型的攻击者在攻击您的企业时有着不同的动机。第一类攻击者称为脚本顽童（也叫做 H4ck0rZ），如果您的企业十分知名，而且攻击您的企业将提高他们在黑客群体内的名誉，那么他们就可能会选择您的企业作为目标。

第二类攻击者称为有针对性的攻击者，他们可能会出于某些原则、信念、间谍活动或政治动机攻击您的组织。这一组攻击者通常拥有希望达成的明确目标，并相应地选择攻击目标。

第三类攻击者称为有组织的犯罪，他们通过攻击防御薄弱且能将其攻击转变为金钱的任何组织来盈利。他们并非专门关注您的组织，但如果有可能，他们就会利用您的组织为自己谋利。

要了解成功的美国 FBI 取证工作如何取缔买卖用于身份欺诈的信用卡数据的国际性地下互联网论坛，请访问以下地址：

http://news.cnet.com/8301-1009_3-10234872-83.html?part=rss&subj=news&tag=2547-1009 _3-0-20

脚本顽童

第一类攻击者是着迷于吸引媒体注意的黑客。这些黑客攻击知名目标，希望能以此在黑客群体中占据一席之地，也有可能仅仅是出于乐趣而攻击 Web 站点。常见攻击类型是涂改破坏（defacement）和拒绝服务攻击，他们希望通过这样的方式在黑客群体中扬名。

此类攻击中，最著名的 Web 应用攻击就是 2005 年的 MySpace Samy 蠕虫 [3]。这个蠕虫病毒的作者使用跨站点脚本（XSS）攻击来创建蠕虫，在 MySpace 社会网络上传播，在不到 24 小时的时间内，“Samy”就拥有了超过 100 万名“好友”。这是一次精心策划的攻击，如今也已成为 Web 应用安全性领域中著名的一个研究案例。尽管 Samy 蠕虫纯粹是为了乐趣而创建的，但 MySpace 却不得不下线以清除这种蠕虫。

这个例子表明，如果您可能会给这组攻击者带来他们需要的知名度，那么他们就会攻击您的企业。他们会跨越边界，作恶作剧，在此过程中建立自己的知名度。

有针对性的攻击者

第二组攻击者采用的是更有针对性的方式。典型的例子就是为间谍活动（国家、州和企业）、政治或宗教信仰等目的而进行的攻击。有针对性的攻击者往往是由使用互联网作为战争前线的组织聘请的。这些组织通常会窃取特定的数据和智力资产，或试图散布其政治或宗教信仰。

安全社区已经注意到了一些以宣传为目的的战争。例如，在 Gary Warner [4] 的博客中，他写下了一些此类以宣传为目的的电子战争：

> 最初的电子宣传战争是由中国黑客在 2001 年 5 月发起的，是在中国战斗机与美国海军侦察机发生冲突之后。数万个美国网站被中国黑客涂改破坏，为这次事件谴责美国。最近，穆斯林黑客采用了这项技术，首先是在 2006 年 2 月，在关于先知穆罕默德的漫画发表之后，涂改破坏了数以千计的丹麦和美国网站，在 2006 年 8 月轰炸黎巴嫩之后，又开展了针对以色列和美国网站的攻击。

如果您身处政府机构或从某些方面构成了这些黑客幕后的组织的关注点，那么有针对性的黑客就会成为您的组织的真正威胁。他们会投入大量精力，攻击能帮助其达成目的的具体目标。

有组织的犯罪

第三组攻击者称为有组织的犯罪。“如果不存在攻击者对您的企业产生兴趣的具体原因，他们就不会攻击您”，这种常见假设并不适用于这一组攻击者。计算机犯罪是寻求能迅速转化为利润的信息的有组织犯罪的一部分。大体上，这种迅速的投资回收就意味着消费者的信用卡信息和银行帐户访问凭据。

但有些时候，攻击者会找到一些方法，从企业服务器和网络中获得大量此类数据，但直接在用户 PC 上运行的间谍软件也会窃取大量此类信息。具有高级补丁和保护机制的企业能够给这些攻击者设定障碍。然而，缺乏保护、无定期打补丁习惯、缺乏安全性意识的用户依然是易于攻击的目标。有组织的犯罪使用一个组织的服务器作为启动平台，展开对目标用户的攻击。例如，他们可能会利用您的论坛来传播恶意文件，如专门编写的 PDF 文件和多媒体应用（如 Flash），这些文件中包含内嵌的漏洞，可在客户的 PC 上安装恶意软件。

此外，某些类型的企业应用，即自定义构建的软件，如 Web 应用等，也是此类犯罪型攻击者的高利润、低成本的攻击目标。商业和开源 Web 应用中发现的漏洞不计其数，而大部分都没有可用的补丁。这与无数同样脆弱的 Web 应用相结合（但永远不会经历漏洞检测，更不用说补丁了），就成为了企业安全性的致命伤。攻击者依然以 Web 应用的漏洞为目标，特别是 SQL 注入，当毫无察觉的用户访问存在漏洞的网站时，就会安装恶意软件。

可以确信，试图通过攻击计算机系统来盈利的有组织的犯罪并非专门关注您的系统。但若您的应用易于通过使用自动化工具利用，您就很有可能成为攻击的目标。

犯罪经济学 101

为了更好地了解犯罪组织如何通过攻击组织来盈利，这一节内容将分析犯罪经济学。[1]在基本的微观经济学层面上，对计算机犯罪机会的理解源于考虑利用一个漏洞能够带来的收入与利用此漏洞的成本的对比关系。显然，能够以较低的成本带来更高收入的漏洞更受攻击者欢迎。收入（机会）和成本都是由一组复杂的组成部分构成的。部分此类组成部分会被应用的安全性影响。

犯罪机会

利用一个漏洞能带来的实际收入是有存在漏洞的主机的安装群体规模和控制各主机对于攻击者的价值共同决定的。这通常是取决于主机包含的信息和攻击者在黑市上销售这些信息的价格。

当一个漏洞初次被揭露时，存在漏洞的计算机的安装群体规模可能相当大。如果控制此类计算机的价值同样高，攻击者在理论上就有着极高的收入机遇。这样的情况可能会激发安全行业迅速推广补丁、减少安装群体规模的工作。如果安全行业的努力

[1] 关于犯罪经济学的讨论主要摘自 IBM Internet Security Systems “X-Force 2008 Trend & Risk Report” [1]

切实有效，攻击者潜在的攻击总收入就会变得极低，最终导致攻击者不会或者不愿意去实现攻击。同时，也存在另外一种情况，有漏洞的计算机的安装群体规模虽然较大，但控制运行这些机器能带来的价值常并不高，攻击者利用漏洞的动机都不强烈。

犯罪成本

通过利用漏洞创收的成本也是由多种因素构成的。首先是获得漏洞的成本，这取决于可利用的漏洞是否公开。其次是与利用漏洞进行攻击的相关难度。与合法企业的情况相似，犯罪组织也有着围绕可重复的环境和可自动化的任务构建的运作流程。

适合现有流程、可使用现有自动化工具进行利用，进行攻击的漏洞更便于犯罪者赚钱。而需要开发新流程或软件才能发现和利用的漏洞对犯罪者的吸引力就比较低，尤其是在此类漏洞未来重复出现的可能性更低的情况下则更是如此。即使对于犯罪者来说，开发新攻击方法来利用一类新漏洞是有意义的，但是大规模攻击所需的时间也比直接适合现有流程的漏洞更长。所以现有的，公开的漏洞对攻击者的吸引力是比较大的。

赚钱

与其他任何企业相似，犯罪组织必须计算一次可能的攻击的价值，以判断是否值得开展攻击。为了比较攻击价值的计算与创建 Web 应用的价值的计算，我们首先来观察一下 Web 应用的价值。我们要如何确定 Web 应用的价值？

向潜在客户显示信息的价值？	$10
通过客户自助服务降低成本的价值？	$100
在富 UI 中交付业务功能的价值？	$1000
创建富协作式用户社区的价值？	无价！

通过这样的计算，可以明确，如果犯罪组织希望在这项业务中盈利，富协作式用户社区是合理的选择。
无论忧心忡忡的安全人员存在怎样的安全性顾虑，都很可能要实现富协作式用户社区。安全人员将竭力阻碍丰富的交互式环境，因为此类环境会带来真正的威胁。然而，必须做出权衡，以正确认识价值。

观察犯罪组织的业务模型，我们可以进行类似计算，了解其价值：

窃取一个信用卡号码的价值？	$0.10
窃取一个电子邮件密码的价值？	$4
窃取一个个人银行帐户凭据的价值？	$10
自动随机攻击系统，收集可出售信息的价值？	无价！

由于犯罪组织往往通过销售攻击 Web 应用获得的信息来盈利，因而我们将使用此类信息的当前交易价值来进行计算，如第 6 页的表 1 所示。2 该表显示了关于不同商品的价值、需求和销售的频率以及其价格范围。

[2] 第 6 页表 1 中的信息是由 Symantec™ 在 Symantec Report on the Underground Economy 中发布的。2008 年 11 月，第 20 页 (http://eval .symantec.com/mktginfo/enterprise/white_papers/ b-whi tepaper_underground_economy_report_1 1-2008-14525717. en-us. pdf). 在许可的情况下转载。

表 1 销售和有需求的商品与服务

销售	需求等级	商品与服务	销售百分比	需求百分比	价格范围
1	1	银行帐户凭据	18%	14%	$0.10 - $1 .000
2	2	带有 CVV2 号码的信用卡信息	16%	13%	$0.50 - $12
3	5	信用卡	13%	8%	$0.10 - $25
4	6	电子邮件地址	6%	7%	$0.30/MB - $40/MB
5	14	电子邮件密码	6%	2%	$4 - $30
6	3	完整的身份	5%	9%	$0.90 - $25
7	4	信用卡套现服务	5%	8%	总价值的 8% - 50%
8	12	代理	4%	3%	$0.30 - $20
9	8	网络诈骗	3%	6%	托管为 $2.50 - $100/周 设计为 $5 - $20
10	7	邮件程序	3%	6%	$1 - $25

由于单位信息的售价相对较低，因而盈利实际上是通过销售大量数据实现的。因而，攻击者必须在攻击和收集数据的成本与此类数据在市场上的价值之间做出权衡。

与其他任何企业相同，自动化也是用于削减成本的一种工具。出于此原因，有组织的犯罪对各类组织的威胁最大，因为有组织的犯罪并不关心从哪里窃取信息，而且他们有能力采用自动化技术来保证较低的成本。

这又将我们引回了利用 Web 应用中现有漏洞的可能性。对照图 1 显示的几条标准，如果您能根据您的组织的情况给每一条标准都打对号，那么就应该考虑到，自动工具入侵您的应用以窃取信息或利用您作为恶意软件传播媒介的可能性是很高的。

图 1 Web 应用中的漏洞被利用的可能性 [3]

图字：OPPORTUNITY：机会
COST：成本
EASY TO EXPLOIT：易于攻击
EASY TO MONETIZE：容易牟利
MANY TARGETS：多个目标
HIGH VALUE：高价值

[3] IBM X-Force 于 2009 年 1 月发布。

对组织的影响

现在，我们已经清晰地了解了犯罪组织选择其他组织（包括您的组织在内）作为攻击目标的动机，我们观察了成功的攻击能给您的组织带来的损害。遗憾的是，成功的攻击给您的组织造成损失的方式是多样化的。数据丢失、品牌受损和无意中助长犯罪，这是成功攻击给您的组织带来的三种最常见的影响。

数据丢失

成功的攻击带来的最直接、最明显的影响就是数据丢失。攻击者入侵您的系统，从您的机器中窃取并删除数据，这将导致数据丢失。在被攻击后使用您的机器时，您的企业无法再使用这些数据，因而您无法再按照之前的方式开展当前的业务。设想一下丢失所有客户记录的影响，想像一下这将给您的企业带来多么严重的损失。

品牌受损

一项更为长期的损失就是攻击者给您的品牌造成的损失。攻击者可能会涂改破坏您的网站，窃取并公布您的大量数据等。所有这些活动都可能受到广泛的关注，并影响您的客户。这会给您的品牌造成负面影响，使您难以保留现有客户或吸引新客户。

无意中助长犯罪

即便犯罪份子并未窃取任何数据或涂改破坏您的网站，他们仍然可以利用您的机器作为宿主，传播其恶意软件。如果这种情况被发现并公开，就可能导致品牌受损，如前一节所述。然而，即便这种情况并未公开，您也在无意识的情况下帮助犯罪份子攻击了您的客户，这将给您的企业带来严重的法律后果。（XSS,跨站点脚本执行就是基于这个原理来助长了攻击和犯罪）。

趋势和统计数据

遗憾的是，由于许多企业更倾向于保密此类信息，因而难以获得关于安全性违规的定量数据。保密的主要动机在于揭露违规细节可能招致的品牌损失。然而，公开报告的信息中已经出现了一些令人担忧的安全趋势和统计数据，在后面的几节中我们将加以讨论。

增加对 **Web** 应用的关注

IBM Internet Security Systems X-Force 研究小组 [1] 发布的研究表明，在过去五年中，已报告的 Web 应用漏洞在所有被报告的安全漏洞中所占的百分比大幅度提高。实际上，自 2006 年以来，所有被发现的漏洞中有 54% 的漏洞都处于商业和开源 Web 应用之中。当今的企业越来越多地通过 Web 广泛实现业务职能，在这一充满挑战的市场中实现增长，因而这样的比例并不令人感到意外。

正因如此，某些 Web 应用供应商被列入 2008 年漏洞披露中情况最严重的十大厂商，而在此之前，这一清单中列出的主要是规模较大的

非 Web 厂商。图 2 展示了 1998 年至 2008 年间所发现的 Web 应用漏洞数量的爆炸式增长。

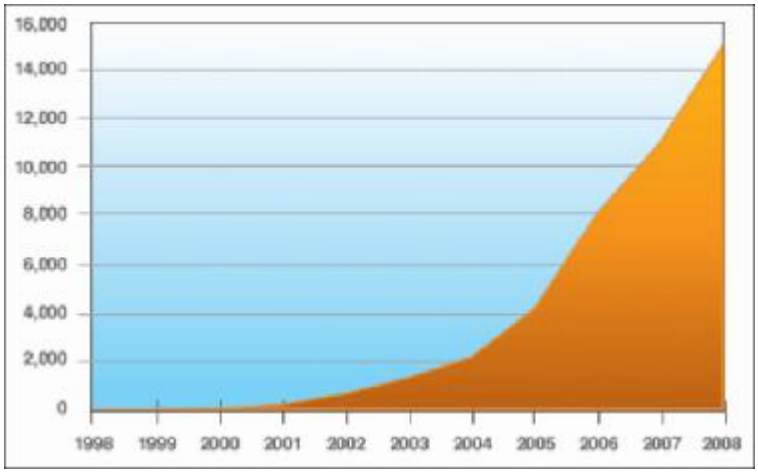


图 2 Web 应用漏洞累计数量[4]

根据全美网络安全局和美国国土安全部发起的已知安全漏洞列表 Mitre Common Vulnerabilities Enumeration (CVE) [5]，公开报告的 Web 应用漏洞总数正在急剧增加。这个数字已经超越了软件中最常见的安全漏洞缓冲区溢出。这样的增加主要源于检测和利用 Web 漏洞更加轻松，加上经验不足的开发人员编写的低级软件应用的普及。这样的增加也是由于创建简单的小 Web 应用更轻松而导致的。尽管出现了这样的简化，但开发安全、高级的 Web 应用仍然是一个复杂的问题。

在 Web 应用安全性方面，有许多反复出现的漏洞。三大最重要的漏洞是 SQL 注入、XSS 和文件包含。图 3 展示了 2004 年至 2008 年间这些漏洞的趋势概览。可以看出，三大漏洞在所有已发现的漏洞中所占的比例高达 70% 至 80%。

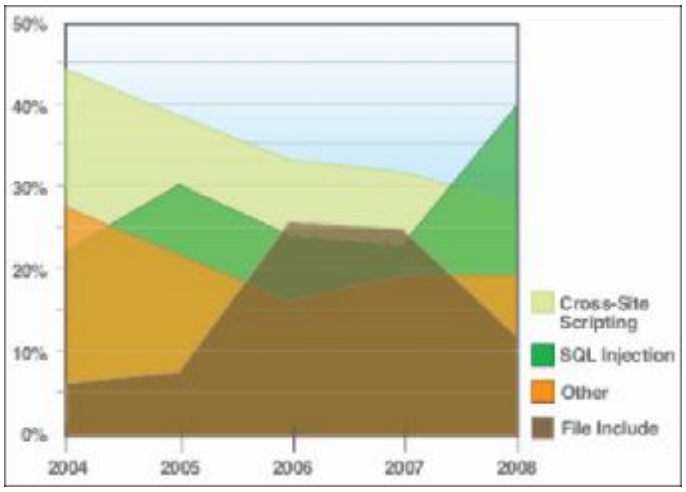


图 3：
Cross-Site Scripting：跨站点脚本
SQL Injection：SQL 注入
Other：其他
File Include：文件包含

图 3 按攻击技术划分的 Web 应用漏洞，2004 - 2008[5]

---

4 *IBM X-Force 于 2009 年 1 月发布。*

5 *Ibid*。

在这份指南中，我们主要讨论 SQL 注入和 XSS，因为它们分别是 2008 年排名第一和第二的漏洞。关于文件包含的更多信息，请参见 IBM Internet Security Systems X-Force 2008 Trend & Risk Report [1]。

攻击的可能性

问题依然存在：在这些漏洞中，有多少比例的漏洞将在一次攻击中被实际利用？恶意攻击者同样注意到了安全问题清单上 Web 应用漏洞的普遍存在，这是一个残酷的事实。Gartner 预计，如今有 75% 的在线攻击关注的是 Web 应用 [6]。这样的统计数据与 大多数安全性支出花费在底层网络而非应用上这一令人担忧的事实相结合，潜在的问题显而易见。Web 应用已经成为攻击的目标——无论是出于财政收入、政治动机还是个人寻找乐趣的目的。

自定义 Web 应用中存在漏洞的可能性

基于上述统计数据，可以明确，所发现和报告的安全性问题往往处于商业打包软件或开源软件中。然而，这对组织理解外包或内部开发应用中存在漏洞的可能性帮助不大。2007 年，Web Application Security Consortium (WASC) [7] 对 32,717 个应用开展的广泛分析表明，其中 96.85% 的应用至少存在一个高危漏洞。58.11% 的应用中存在 XSS 漏洞，25.30% 的应用中存在 SQL 注入漏洞。因而，应该认为，您的自定义 Web 应用中存在漏洞的几率很高。

查明 Web 应用的薄弱环节

在这一届中，我们将介绍关于 Web 应用薄弱环节的技术背景。我们概述 Web 应用可能受到攻击的最常见方法，并提供关于两种最常见的 Web 应用攻击的深入见解。

Web 应用可能发生怎样的问题？

Web 应用中存在一些最有可能发生问题的领域。主要的难点包括不安全的数据传输、服务器端缺乏输入验证、客户端缺乏控制等。

不安全的数据传输

在数据穿越网络时，很有可能被窃。在大多数 Web 应用中，在客户端和服务器之间传输的数据包是通过不受控、不可靠的网络传输的。例如，如果您打开浏览器，请求一个股票交易商的页面，您的请求将拆分成多个信息报，通过公共互联网传输，中间经过无数路由器和交换机。尽管有着可伸缩的优点，但用于互联网传输的模型会使数据易受中间人 (MITM) [8] 攻击的侵扰。在此类攻击中，攻击者处于数据传输的起点和终点之间，监控或修改所传递的数据。例如，攻击者可修改股市新闻的内容，希望影响新闻接收者可能会根据虚假内容做出的决策。

这样的攻击者类似于传统领域中的邮递员。即便您了解并信任每天为您收发邮件的邮递人员，但也不了解在您的邮件离开邮箱到最终抵达目的地这段时间中发生了什么。例如，有人可能会使用蒸汽拆封您的邮件或利用光学成像技术偷窥其内容。同样，其他人也可能会

参与投递邮件的这个过程。无论是哪种情况，都必须运用安全技术，保护内容在传输过程中的完整性。

服务器端缺乏输入过滤和检查

服务器端应用安全性是一个庞大而复杂的主题，无法在本指南中完全详述。按照定义，应用会接受用户输入，并根据该输入执行操作。这样的输入可能是页面请求、击键操作或表单提交。上述各种输入都将由服务器处理。如果输入未得到安全的处理，就可能无意中破坏或操纵应用。

我们可以扩展邮递员的示例，以包含对邮件基础设施的攻击行为。假设邮递员未通过 X 光和爆炸物检查合理过滤传入的邮包。那么这位邮递员就有可能因递送内有爆炸物的邮包而受到攻击。可以设想，邮包的不合理过滤可能会对邮递员产生破坏性的影响。

客户端缺乏控制

尽管服务器端一度是流行的攻击位置，但越来越多的攻击不再指向服务器，而是针对客户端。对于新兴的 Web 2.0 应用来说，这种情况尤为严重，因为这些应用将更多的逻辑转入客户端，以提高应用的响应能力。您可能为应用代码和环境实施了有力的安全控制，但几乎不可能控制客户端独立用户或客户端机器的行为。因此，可能会出现许多问题。

同样，我们要在客户端攻击方面深入探索邮递员的例子。假设攻击者已经确定了特定的收件者。他们就可通过制作专门的邮件来攻击这位收件者，例如包含炭疽病毒的邮件会在邮件抵达目的地时立即攻击收件者。

Web 应用漏洞

在下面的几节内容中，我们将观察并分析两种最常用的 Web 应用漏洞，并为您介绍能在哪里找到更加完整的漏洞清单。

SQL 注入

在 Web 上可以找到大量详细讨论 SQL 注入的文档 [9-1 1]，所以本文不会涵盖 SQL 注入的所有方面。但我们将通过示例来展示 SQL 注入的工作原理，解释它将会对您的组织产生怎样的影响。

考虑一个带有自定义登录表单的应用，其中包括用户名和密码。系统为了对用户进行身份验证，向数据库发送这样一条查询：“用户 X 的密码是否是 Y？”如果此查询的结果为真，则用户 X 就通过了身份验证。如果结果为假，用户将被拒绝。

现在，假设服务器端并未执行合理的验证，允许某些人创建以下查询：“用户 X 的密码是否是 Y 或者我能否输入 X 作为用户名？”[6] 此时不是简单地将 Y 作为密码提供，在将此视为一条查询时，恶意用户可确保查询始终返回真。第 11 页的示例 1 显示了与此示例相关的 SQL 查询。

[6] 为了更清晰，我们将使用斜体显示用户输入。

示例 1 登录 Web 应用的 SQL 查询

```
SELECT userid, full_name
 FROM members
 WHERE username = 'X' AND password = 'Y'
```

为成功攻击此查询，攻击者必须确保所提供的输入能突破当前预定义的 SQL 查询命令集（SELECT、FROM、WHERE、AND），并插入攻击者自己的命令。只要攻击者能够突破预定义的结构，即可操纵查询，返回其希望的结果。攻击者将获得与执行 SQL 命令的进程相同的权限，随后即可对您的数据库执行各种操作。

现在，假设服务器不会执行任何验证，用户名和密码字段的所有输入都会直接重定向到 X 和 Y。那么攻击者即可使用单引号来突破预定义的命令并注入攻击者自己的命令，从而确保此查询总是返回真。

在 Y 中注入恶意输入可能会导致类似于示例 2 所示的查询，始终保证攻击者通过身份验证。

示例 2 破坏 SQL 查询的恶意输入

```
SELECT userid, full_name
 FROM members
 WHERE username = 'X' AND password = 'Y' OR 'X' = 'X'
```

攻击者能通过向 SQL 查询注入恶意输入而实现的操作实际上是无限的。基本上，攻击者具有与执行查询的进程相同的访问权限，可以完成此进程能够执行的任何操作。为了攻击您的组织，攻击者会创建多种输入值，试图发现您的查询中允许使用哪些字符。此后，攻击者会尝试了解数据库结构，利用这样的认识来获得表名称、用户名和密码。攻击者甚至可能会破坏整个数据库。如果您的企业没有对攻击者来说有价值的数据，那么攻击者可利用 SQL 注入攻击，将恶意软件注入您的页面，利用您的网站作为传播平台，攻击您的网站的访问者。自动化的强大力量和简便性使此类攻击成为 2008 年最常见的一类攻击方式。

遗憾的是，注入漏洞（injection flaw）不仅限于数据库，还可能包括文件系统注入、邮件（MX）注入、可扩展标记语言（XML）注入、轻量级目录访问协议（LDAP）注入和服务器端包含（SSI）注入，而这些仅仅是其中的一部分。如果服务器未能正确处理所有应用输入，应用环境就可能被入侵，业务逻辑被破坏、受保护的数据被公开。

跨站点脚本

XSS 攻击十分普遍，Web 上可以找到关于此类攻击的大量资料 [12, 13]。我们采用与介绍 SQL 注入相同的方法，通过一个示例来解释攻击者会如何利用此类漏洞。

设想一个支持论坛。如果此论坛的输入未得到合理过滤，恶意用户即可在论坛上张贴任意 HTML。其他用户访问论坛时，将为其显示这些 HTML 代码。如果 HTML 代码中包含脚本，每当有一名用户访问论坛上的这个帖子时，就会执行此脚本。第 12 页的示例 3 展示了攻击者如何插入一条弹出消息，在他人访问此帖时显示弹出消息。

示例 3 论坛上张贴的未经过滤的消息，为每位阅读者显示一条警报

Hello everybody,

Thank you very much for helping me out.Due to your suggestions I was able to figu
re out how to solve my problems.

Thanks, Evil Bob

<script> alert(“Don’t you like Evil Bob?”);</script>

初看起来，Web 站点中的 XSS 漏洞似乎并不会给您的用户造成严重影响。但若 XSS 攻击发挥到最严重的程度，就会与 SQL 注入同样危险，甚至比 SQL 注入更加危险。根据攻击者的熟练程度不同，攻击者通过在您的页面中注入代码而获得的对用户的控制级别也有所不同：

1. 最基本的攻击在您的 Web 页面中注入数据片段，以实施涂改破坏。此类攻击的典型示例就是注入“你已经被入侵”之类的消息，插入图片等。
2. 在下一个级别中，攻击者可能会尝试通过提供错误或误导性的信息来影响用户。此类示例包括提供股票走势的假信息、伪造在线年度报表等。
3. 第三级的攻击者会尝试干扰网站的正常使用。攻击者可能会利用 XSS 漏洞来注入代码，使网站以不正常的方式运作甚至完全不可用。此类示例包括生成数以千计的弹出消息、重新定义链接的指向位置等。
4. 更熟练的攻击者可能会利用 XSS 来窃取身份。通过注入在用户登录时在页面上运行的代码，攻击者即可通过键盘记录器来截取密码，窃取用户的身份。
5. 第 5 级的攻击者会在用户使用您的网站时监控这些用户。高级 XSS 攻击允许攻击者监控用户执行的每一项操作。攻击者能查看用户浏览的每一个页面，获取用户输入的所有信息（包括其用户名、密码、信用卡信息、地址等）。
6. 最终，攻击者可全面掌控用户的环境。通过控制用户，攻击者即可自行制定决策、浏览您的网站的各个部分、购买商品等。

其他漏洞

目前 Web 应用中存在的漏洞的完整清单过长，无法在本指南中一一详述。如需了解更多信息，请参见开放 Web 应用安全项目（Open Web Application Security Project，OWASP）[14] 和 Web 应用安全联盟（Web Application Security Consortium，WASC）[15] 的网页。

保护您的 Web 应用免受攻击

我们必须以一种方式来研究 Web 应用的各个方面，从而找出最多漏洞；我们还必须采用类似的方式来检查软件开发生命周期的各个方面，从而将我们的 Web 应用构建得尽可能安全。

在本节的第一部分，我们概述了软件开发生命周期的不同难点，介绍了测试软件安全性的不同方式，并着重指出了两个最重要的最佳实践。本节的第二部分首先概括介绍 IBM Rational 工具，这些工具能协助您保证软件开发生命周期的安全性。然后我们探究 IBM Rational AppScan 产品线的详细信息，介绍它们的不同版本，并说明如何将其整合到您的软件开发生命周期中。借此，我们向您展示如何构建一个开发 Web 应用的安全性生态系统。

保证软件开发生命周期的安全性

软件开发生命周期包括三个阶段：

- 设计阶段
- 开发阶段
- 交付阶段

每个阶段都会营销您的最终产品的整体安全质量，因此，必须从安全性角度考虑它们。

设计阶段

软件开发生命周期中的设计阶段包含需求的创建和应用架构的设计。为保证软件开发生命周期的安全，在执行需求和架构的设计时都必须紧记安全性。如果需求和架构未被明确地设计、规划和执行，那么几乎所有应用都可能存在巨大的缺陷。

在水肺潜水人群中，流行这样一句重要的格言：“为潜水做计划，按计划来潜水。”失败的计划可能导致严重的后果。人的生命不是常常都会面临危险，而这一准则对 Web 应用而言同样有效。如果所设计的软件计划不健壮和不安全，那么可能发生最具灾难性的软件故障。拥有适当的需求集合以及满足这些需求的设计，就能够创建这样的计划。要为此流程提供帮助，可以使用多种 IBM Rational 工具。

考虑这样一个与身份验证相关的普通场景。美国的研究表明 9 个人中就有一个人使用 500 个最常用的密码中的一个，每 50 个人中就有一个使用最常用的 20 个密码中的一个。这是一个严重的安全问题，因为黑客很容易暴力破解这 500 个最常用的密码。

解决这一问题的常用办法是封锁那些在短期内过多地进行尝试而登陆失败的帐户。但是，黑客可能使用不同用户名来尝试那些最常用的密码，以避免被封锁。因为您不希望封锁掉所有帐户，所以您对此无计可施。您也不希望禁止发动攻击的 IP 地址的访问，因为担心这会屏蔽来自相同网关的合法用户。

因此，安全地设计应用至关重要。制定一个需求，要求您的系统不接受容易被猜到的密码，这可能足以预防此问题。当然，需求必须被正确地实现，这就让我们进入到软件开发生命周期的下一个阶段，也就是开发阶段。

开发阶段

开发阶段是一个三步流程，在此阶段中代码被编写、构建和测试。虽然许多软件开发小组认识到需要安全地开发应用，但是经验证实开发安全应用的难度不小。实际上，大多数被报告的漏洞都是开发实践不佳的结果。恶劣的开发实践的一个典型示例是，经验不足的开发人员编写了一个自定义组件，同时将安全漏洞引入到了应用中。较好的实践是，使用成熟框架中已经经过全面安全漏洞测试的现有的可靠组件。此外，就安全开发实践对开发人员进行培训，这在未来将会有所回报。

随着富 Web 2.0 UI 设计的快速变化，对安全代码开发的需要也变得尤为关键。不断变化的 Web 2.0 设计几乎没留下进行全面测试的空间。在此之外，为了最大化交互性，更多应用代码在客户端浏览器上运行，以便用户能够轻松查看。组织必须假定，用户将故意篡改公开的应用业务逻辑，
并试图发掘其自身优势。[7] 通过将正确的工具整合到开发流程中，您 Web 应用的编码、构建和测试过程中与安全性相关的许多任务将实现自动化。

交付阶段

最安全地设计和开发的软件如果被交付到不安全的环境中，也会不再安全。这涉及到（但

不限于）应用基础架构的固化、数据通过网络时的保护、生产环境的防御以及配套操作系统和组件的修补和更新战略。

例如，没有将 Web 服务器配置为拒绝访问目录结构，这会允许恶意用户直接访问敏感信息和应用代码。因此，需要有安全的交付阶段，在交付环境中最终审核应用安全性，然后维持操作环境的安全级别。而且，广泛的工具可用于使这些任务自动化。

分析的细微差别

在软件开发生命周期中用于分析 Web 应用安全性的工具大致分为三种不同的类型：

- 白箱分析工具
- 黑箱分析工具
- 灰箱分析工具

这些分析工具根据有关系统和软件的信息量来区分，这种信息量在进行安全性分析时被这些工具使用。从开发人员的角度分析，白箱分析对信息的访问级别最高，并可以视作逼近安全性。黑箱分析从攻击者的角度出发进行分析，无需访问信息。灰箱分析结合了黑箱和白箱分析的特点，以提高精确性和覆盖范围。

7 黑客根据可视的客户端逻辑逆向工程业务逻辑的一个很好的例子就是 2007 攻击，它使黑客获得通行 Macworld 2007 的免费 VIP 折扣代码 [16]。

白箱分析

使用白箱分析，关于系统或软件的所有相关信息是已知的，并可被测试人员使用。该分析方法在质量保证领域（在该领域中，负责软件的人员既能访问设计文档，又能访问源代码）中得到普遍应用。除开源软件外，这种信息通常不会对限定人群以外的个人公开。

给定相关信息，白箱分析比较全面。这种方法可以快速揭示外行人看不到、潜在的、模糊的相互关系。 白箱分析可以快速确定整个攻击面并创建一套必要的测试。

在白箱分析中，重要的是不要过于强调设计规范或者源代码。过于强调设计规范可能导致测试人员失去在该文档之外构建的或者尚未正确实现的功能。过于强调源代码可能导致测试人员遗漏与架构相关的关键漏洞或系统设计的方式。

白箱分析包括以下三种主要的技术：

- 架构分析，通常称为威胁建模，试图枚举在软件内攻击者所攻击的目标并提出针对每种威胁的对策
- 源代码分析， 扫描应用的源代码并跟踪用户输入，以查找漏洞和错误编码实践
- 静态二进制分析，与源代码分析的操作类似，但只在二进制级别上分析，允许查找上下文风险和特定于平台的问题。

白箱分析提供了如下优点：

- 测试人员可利用所有的相关信息。
- 可以确定逻辑分析的缺陷。
- 能够轻松、快速地归档整个攻击面。
- 能有效查找编程和实现错误。

白箱分析面临如下挑战：

- 不良的编码实践可能作为漏洞被错误地检测到。
- 软件不容易在分布式环境中被远程测试。
- 不可能始终访问设计规范和源代码。

白箱分析的一个例子就是在开发期间运行源代码扫描器。

黑箱分析

黑箱分析涉及在事先不了解环境的情况下检查软件或系统。这种分析与外部攻击者可能从事的分析类似。使用自动化工具和手工技术，这种分析首先检测攻击面和探查相关信息的系统。

黑箱分析的基本概念是在针对安全性问题进行实际测试前尽可能全面地了解系统。信息披露和配置不正确的系统在建立这种认知的过程中尤其有用。在测试阶段中，对软件或系统的这种基本了解使测试人员的工作效率更高。

当一个机构要分析外部人员的威胁时，黑箱分析经常是首选的分析方法。从这种分析和测试中得出的风险结果经常要区分优先次序，因为它能更准确地反应外部人员所构成的直接风险。

黑箱分析包括以下两项基本技术：

- 漏洞扫描，通过利用已知漏洞的大型数据库以及尝试识别应用中的已知漏洞来实现。 这既可以是被动的，例如通过搜索 Web 页面中的版本号来进行漏洞扫描，也可以是主动的，例如通过尝试利用一种漏洞以及搜索已知的利用结果来进行漏洞扫描。

- 动态分析，通常称为 Web 应用扫描。这种分析试图自动扫描并记录攻击面，借助故障注入的手段来测试应用以及根据响应来确定漏洞的存在。这种分析和漏洞扫描之间的主要区别在于动态分析可以发现未知的漏洞，而漏洞扫描只针对已知的漏洞进行测试。

黑箱分析包括如下优点：

- 测试已部署的软件会生成高度可信的结果。
- 无须访问设计规范或源代码。
- 可以跨网络轻松测试软件。
- 在已部署的环境中测试支持环境分析。

黑箱分析面临以下挑战：

- 不可能确定代码覆盖范围；不明显或隐蔽的功能可能被遗漏。
- 逻辑设计缺陷不易被检测到。

黑箱分析的一个例子就是在交付阶段中针对所部署的应用运行 Web 应用扫描器。

灰箱分析

白箱和黑箱分析均可以揭示可能的软件风险和潜在的漏洞。白箱分析保证了代码覆盖范围，但存在难以估量的风险。黑箱分析能保证识别真实问题，但其代价是代码覆盖范围未知。灰箱技术以一种强大的方式将白箱和黑箱分析方法结合起来。可以获得这两种方法的优势，同时最小化遗漏重要问题的可能性。

部署灰箱分析的挑战是，这一过程通常需要使软件开发生命周期的不同阶段中所得到的结果集相互关联起来。在没有企业文化转型的情况下可能难以采取成功的灰箱分析策略。当灰箱分析成为软件开发生命周期的一个重要部分，允许在整个生命周期内平稳地整合所获得的不同测试结果时，最有可能取得成功，

注意：使用这些术语的一种更常用的方式是区分源代码访问（白箱）和非源代码访问（黑箱）。尽管从严格意义上讲这种命名并不准确，但它通常是流行的命名法。

在开发过程中使用安全性分析技术

当考虑更广泛的软件开发生命周期时，我们就会明白每种安全性分析技术在软件开发生命周期中具有各自的功能。要获得完整的覆盖范围，您必须将白箱和黑箱测试结合到一项安全性测试解决方案中，作为软件开发生命周期的一个组成部分。

图 4 显示了在整个设计、开发和交付阶段中如何使用并组合架构分析、源代码分析、动态分析、二进制分析和漏洞扫描。 安全性测试不是只能在软件开发生命周期结束到产品发布这段时间内才能做的事情，认识到这一点很重要。尽管在交付阶段进行全面的安全性审计适当保护软件的一个重要方面，但必须将安全性整合到开发生命周期的每一步中。

架构安全性分析应该作为设计阶段的一个组成部分。应该将源代码分析整合到开发阶段中。动态分析在开发阶段开始进行并一直持续到交付阶段。二进制分析通常在交付阶段的审计期间针对软件的每次新建进行一次。最后，漏洞扫描是一项重复性任务，只要软件运行，就应该定期进行。

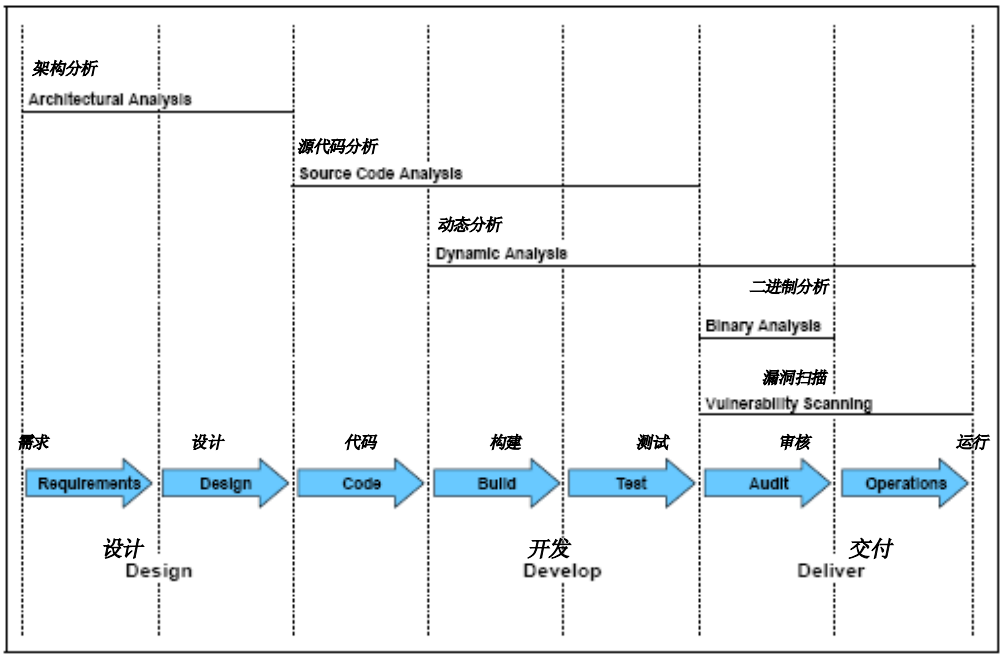


图 4 在整个软件开发生命周期内安全性分析的不同类型

两种重要的最佳实践

下面让我们更深入地了解两种重要的最佳实践：

- 定义明确的安全需求
- 最大化地实现对安全需求的自动化测试

定义明确的安全需求

要定义明确的安全需求，重要的是确切了解什么是安全的需求。尽管安全需求与安全特性用于根本不同的目的，但它们经常被混淆。使用以下两个定义可以明确这种差异：

- *安全需求是必须完成的任务。* 例如，所有密码必须加密存储。安全需求通常不会指示如何做，例如指示用哪个软件用来加密。

► *安全特性是必须可用的功能*。 例如，用户管理必须能够在 Web 界面进行。

与功能要求或性能要求类似，安全需求需要确保从一开始就将安全性构建到应用中。安全需求定义要求何种新安全特性和如何改变现有特性以包括必要的安全属性。设定安全需求的目标是确保应用可以预防和抵挡攻击。当构建 Web 应用时，您必须考虑九类安全需求，下列章节将一一解释。

审计和日志记录

尽管人们通常依赖网络和包日志来进行取证分析，同时，应用程序内部的日志记录也是同等重要的；应用程序应该对软件的保密性、可用性和完整性至关重要的事件进行内部日志记录。

例如，应用就需要有审计日志。审计日志来记录日志的事件必须包括当前会话令牌（如果有）、IP 地址和时间。必须记录日志的事件还包括帐户验证尝试、帐户锁定、应用错误和与规定的验证例程不匹配的输入值。

身份验证

由于大多数应用具备访问控制限制，以防止机密性泄漏，因此，确保这些访问控制机制不能被破解或操作以允许未经授权的访问，这一点非常重要。

例如，要求强密码。任何身份验证凭证必须包含适当的强度，其中包括大写字母、小写字母和数字字符，而且在长度上不能小于 8 个字符。

会话管理

HTTP 协议最初的设计使得难以在整个应用会话持续期间跟踪会话。这推动了 HTTP 协议之上的会话管理功能的构建。

例如，一个安全需求是合法用户自始至终可以保持正常访问；远程会话的所有资源利用必须加以监控和限制，以防止或减轻对应用可用性的攻击。

输入验证和输出编码

尽管在建模和架构阶段大多数设计级安全性缺陷都被发现，但大多数开发和交付安全性问题是因为不良的输入验证和输出编码而引入的。重要的是用户提供的数据要通过适当的验证。

例如，所有输入必须通过集中的验证控制来加以验证。

异常处理

从严格意义上讲，一个应用不可能完全安全。 但是，常常最可接受的是应用“足够安全”。隐藏详细的应用异常或过于具体的错误消息能够延长攻击应用所需的时间。

举例来说，此上下文中的一个安全需求是应将所有错误消息捕获并记录在安全性审计日志中。

加密技术

安全的加密算法极难创建和实现。一个机构选择一种满足业务需要、受行业支持的算法极其重要。

安全需求的一个例子是，应用内所使用的所有加密算法必须经过联邦信息处理标准（Federal Information Processing Standards ， FIPS）[17]批准且与之兼容。

静止数据

尽管所有应用都试图保护后端存储库中的数据，但是最好假设该数据存储将来在某种意义下会被泄漏。深度防护规定任何敏感数据都要针对这种可能性进行加密。

安全需求的一个例子是，如果应用包含必须为了法规遵从性而加以保护的敏感用户信息，则必须使用功能强大的加密技术来保护姓名（名称）、用户名、地址和财务数据等敏感的用户信息。

活动数据

只要软件应用被实现，就存在对要求跨网络或系统传输数据的软件架构的已知攻击。当应用数据跨越开放和封闭的网络和系统时，对其妥善保护是至关重要的。

对这类系统的安全需求可能包括：如果应用在不可信或不安全的网络间传输敏感的用户信息，那么所有通信渠道必须予以加密。

配置管理

新的漏洞每天都会涌现，这凸显了普通基础设施组件中的弱点。尽管其中一些问题可通过打补丁的方式加以纠正，但有时要求这些选项能够用于进行特定部署的用户。

通过既满足应用业务需求又保护了应用和基础设施的方式来平衡底层系统，这至关重要。

安全需求的一个例子是，所有管理界面必须从非管理界面分离出来。

最大化对这些需求的自动测试

在定义了安全需求的明确集合后，具备可靠的流程以检查这些需求是否在整个开发生命周期内得到正确使用非常重要。如果您回想攻击者的动机以及他们操作的方式，就会想到他们试图通过自动化攻击您的 Web 应用来削减成本。

与攻击者自动化其任务的方式相同，您也可以自动测试安全需求并验证其实现的正确性。事实上，如果您确信以自动化方式轻松找到的这些漏洞再也不会出现在您的 Web 应用中，那么您就已经消除了高比例的可能攻击。

因此，确保您使用了自动测试来保护 Web 应用是一种不错的实践。由于自动测试不能涵盖所有漏洞，所以如有必要，这种测试可以根据您的业务需求通过防御攻击和手动代码修正来扩大。在下一节中，我们向您展示如何使用 IBM Rational AppScan 产品在开发生命周期内自动化安全性测试。

IBM Rational AppScan 软件套件

IBM Rational AppScan 是一个 Web 应用安全性测试产品套件，用于自动化应用扫描和漏洞识别。Rational AppScan 产品针对大量 Web 应用漏洞进行扫描和测试，其中包括 Web 应用安全协会（Web Application Security Consortium，WASC）威胁分类和开放 Web 应用安全项目（Open Web Application. Security Project，OWASP）所确定的漏洞。Rational AppScan 产品线包含大量产品，每种产品适用于特定用户的要求。

在本节的剩余部分中，我们按照软件开发生命周期中产品出现的顺序讨论主要的版本。要了解整套 Rational AppScan 产品的相关信息，请参见下列地址：

http://www.ibm.com/software/awdtools/appscan/

Rational AppScan Source Edition

Rational AppScan 产品线的首要目标用户是开发人员。预防应用安全漏洞的最有效方式是从头开始安全地构建软件。其挑战在于，大多数开发人员并非安全专家，且编写安全代码并非总是他们的头等大事。因此，在应用安全性流程中从事开发的最佳方式是向其提供在其开发环境中工作且以他们理解的语言生成结果的工具。

IBM Rational AppScan Source Edition 的设计理念在于使开发人员能够从其开发环境内进行应用安全性测试。它能解决可能在代码中存在的大量安全性问题、流线化开发生命周期工作流，并帮助减少在发布周期结束时可能出现的安全性测试瓶颈。Rational AppScan Source Edition 使用大量的分析技术来准确确认应用中的安全性问题，其中包括静态代码分析、运行时分析和字符串分析。

当进行安全性测试时，开发人员的要求与安全审计人员的要求有很大不同。Rational AppScan Source Edition 旨在用作一种开发工具。因此，其焦点集中在易用性以及易于整合到开发流程中。最大限度减少错误肯定并提供易于理解的结果的特性具有比增加扫描范围（可能使安全性测试复杂化）的特性更高的优先级。配置和结果的协作和共享是该产品的核心部分，扫描配置的重用能够帮助在每个应用上提供一致、可重复的扫描。

开发人员版本在设计时充分采用了自动化理念。为实现更出色的易用性和精确性而进行的自动代码分析配置是通过使用字符串分析来实现的，字符串分析是在与 IBM Research 协作的过程中发明的新技术。 通过帮助解决困扰当前安全代码扫描解决方案的最大挑战，也就是错误肯定，字符串分析是静态代码分析领域中的一大突破。迄今，分析代码安全性的最先进技术——污点分析能够在代码流入系统时跟踪输入值。但是，它依靠开发人员来确定数据是否通过标记清理功能进行了适当清理。

因此，开发人员必须知道何谓十分简单的清理，以及必须能对分析工具进行详细配置，使其精确。这些内在的限制导致了大量错误肯定现象，经常需要修改代码来支持扫描工具，并且需要安全专家参与。字符串分析自动做出这些决定，帮助消除错误肯定并支持不同的输入处理方法。尽管污点分析能够衡量输入是否被污染，但字符串分析可以准确确定输入如何被被污染的，从而将静态代码分析提升到一个全新的精确性水平。

除了字符串分析的自助服务优点外，Rational AppScan Source Edition 还提供了内建的培训，准确和区分优先次序的结果（直接指向有疑问的代码行），以及带有代码样例的详细补救建议。这些直观、易用的特性使开发人员能够在 Web 应用安全性测试的日常处理中充满自信。

图 5 显示了一份 Rational AppScan Source Edition 扫描报告，说明了它如何将白箱问题与受影响的代码行联系起来，从而快速定位和轻松缓解问题。Rational AppScan Source Edition 还具备与其他产品整合的各种能力，详见下一节。

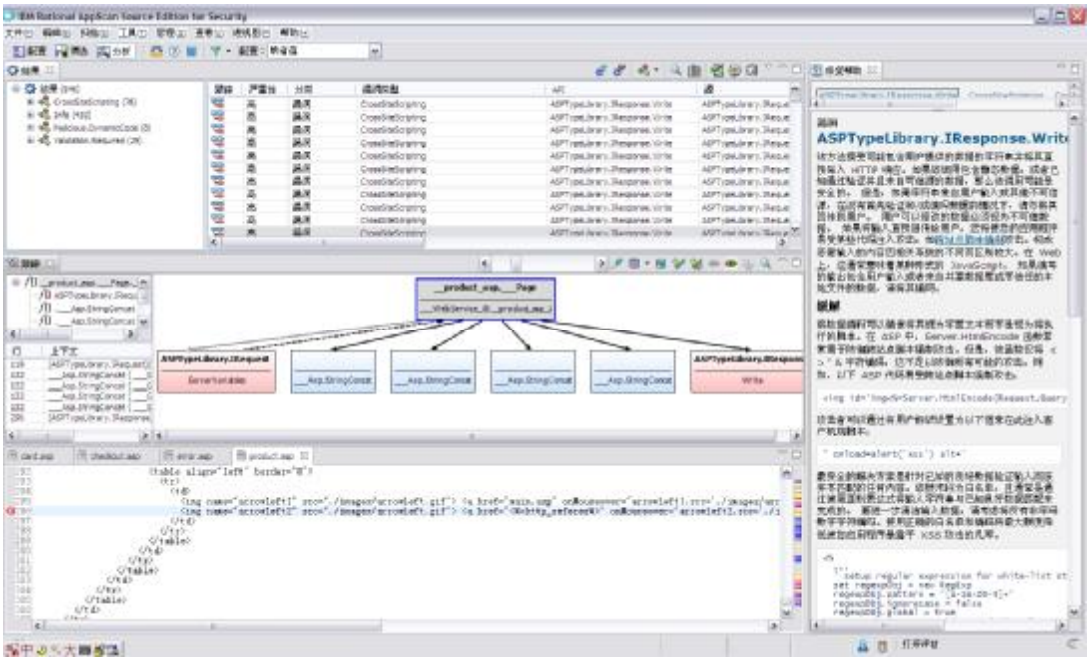


图 5 Rational AppScan Source Edition 中显示了白箱问题和受影响的代码行的完整扫描报告

要了解有关 Rational AppScan Source Edition 的更多信息，请参见下列网址：

http://www-01.ibm.com/software/rational/products/appscan/source/

Rational AppScan Build Edition

使用 Rational AppScan Source Edition 漏洞扫描引擎的漏洞检测功能的另一种方式是在构建期间执行自动扫描。通过使用 IBM Rational AppScan Build Edition，可以将安全性整合到构建阶段。通过与 IBM Rational Build Forge® 软件等多个构建管理系统整合，Rational AppScan Build Edition 提供了针对计划构建的安全性测试覆盖范围。它包括与 Rational AppScan Source Edition 相同的分析技术集，提供了高水平的精确性和代码覆盖范围，能够帮助确定哪些代码经过测试。

扫描后，Rational AppScan Build Edition 将结果通过 IBM Rational ClearQuest® 软件等缺陷跟踪解决方案或通过 Rational AppScan Enterprise Edition 或 Rational AppScan Reporting Console 等安全报告解决方案发送给开发人员。Rational AppScan Build Edition 还包括一个应用编程接口（API）和各种其他结果格式，支持将扫描结果传播到其他存储库。

要了解有关 Rational AppScan Build Edition 的更多相关信息，请参见下列网址：http://www.ibm.com/software/awdtools/appscan/

Rational AppScan Standard Edition

IBM Rational AppScan 产品考虑的下一类用户是安全审计人员。为了帮助这类用户，我们发布了 IBM Rational AppScan Standard Edition。为了使安全审计人员能够自动化对最新技术的测试，Rational AppScan Standard Edition 支持最新的 Web 2.0 技术；JavaScriptTM 和 Adobe® Flash 应用的解析和执行；异步 JavaScript XML（AJAX）和与 Adobe Flex 相关的协议，如 JavaScript Object Notation（JSON）、Action Message Format（AMF）和 SOAP；精细的面向服务架构（SOA）环境；以及针对 mashup 和流程驱动应用的自定义配置和的报告功能。

通过自动化许多重复性任务，Rational AppScan Standard Edition 降低了与手动漏洞测试相关的成本。无论是外包您的漏洞测试工作，还是在组织内部手工执行漏洞测试，Rational AppScan Express 都可以显著减少对应用执行全面的漏洞评估所需的时间。这使您能不断对 Web 安全状态进行评估，而不是每季度或每年审计一次，从而获得更高的安全性水平并实现可控的成本。

Rational AppScan Standard Edition 扫描引擎为您提供了高水平的扫描精确性并显著限制了错误肯定。为了进一步提高精确性和性能，该引擎包括了智能模拟人类逻辑的自适应测试流程，以适应针对个别应用的测试阶段。Rational AppScan Standard Edition 了解应用的信息，一直深入到每个特定参数，并进行调节，以便只执行相关的测试。为了帮助确保免受最新威胁，Rational AppScan Standard Edition 在软件每次启动时都会检查来自 IBM 安全研究专家团队的攻击签名更新。

Web 漏洞扫描的最重要的方面之一是问题的快速修补。Rational AppScan Standard Edition 提供了每次扫描发现的漏洞的完整列表，这些漏洞按照优先次序排列，使高优先级的问题首先得到修复，帮助机构从安全性角度将精力集中在最要紧的问题上。每个漏洞结果包括漏洞如何工作和潜在原因的详尽描述。综合的、基于 Web 的培训提供了直接来自用户界面的短期培训模块。软件的修补视图然后解释修补该问题所要求的步骤，其中包括安全和非安全代码的样例。

图 6 显示了 Rational AppScan Standard Edition 应用的一个窗口。

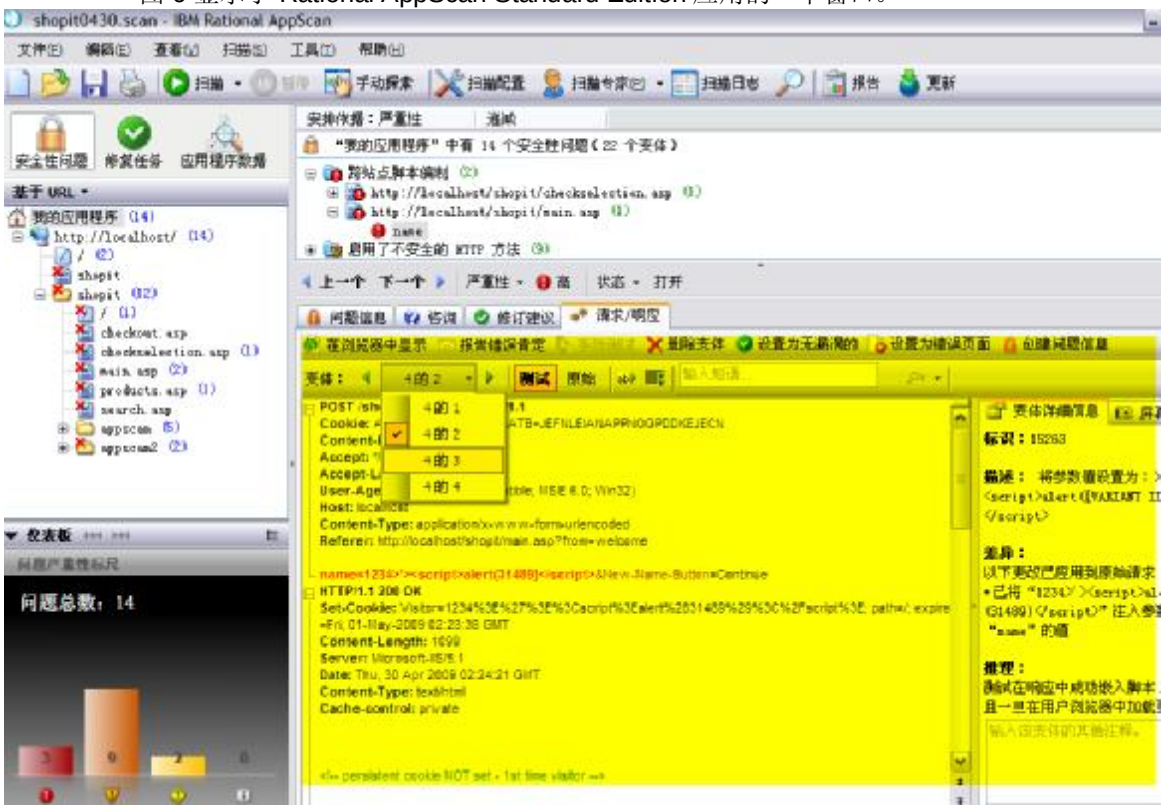


图 6 Rational AppScan Standard Edition 帮助用户快速识别、理解、区分优先次序和修复重要的 Web 漏洞

Rational AppScan Standard Edition 还可以通过提供一种方式来支持现行应用安全性级别来帮助机构解决支付卡行业数据安全标准（Payment Card Industry Data Security Standard，PCI DSS）等重要的遵从性要求。IBM 是一个授权的扫描服务商（Approved Scanning Vendor，
ASV），Rational AppScan Standard Edition 产品提供了这项资格，使该软件成为解决围绕 PCI DSS 提出的应用安全需求的完美选择。

Rational AppScan Standard Edition 可以生成定制的安全报告，并包括将哪些数据点选入每项报告的能力。用户还可以从 40 多种预定义的报告和地图扫描结果以及关键的行业和法规遵从性标准中进行选择。这些选项包括国家安全技术局专刊（National Institute of Standards and Technology Special Publication，NIST SP）800-5、10 大开放 Web 应用安全项目（Open Web Application Security Project ,OWASP）、PCI DSS、萨班斯-奥克斯利法案（Sarbanes-Oxley）、金融服务现代化法案（Gramm-Leach-Bliley Act，GLBA）、健康保险责任法案（Health Insurance Portability and Accountability Act， HIPAA)）、家庭教育权利和隐私法（Family Educational Rights and Privacy Act，FERPA）、自由信息和隐私保护令（Freedom of Information and Protection of Privacy Act，FIPPA）和支付应用最佳实践（Payment Application Best Practices，PABP）。

为了定制和扩展测试以进行更强的控制，Rational AppScan Standard Edition 包括了一组功能强大的定制功能。IBM Rational AppScan 软件开发工具箱（SDK）提供了一套功能强大的接口，支持对 Rational AppScan Standard Edition 中的每项操作（从长期扫描的执行到单独定制测试的提交）进行可定制调用。

该平台能够轻松整合到现有系统上，支持 Rational AppScan 引擎的高级定制使用，并为 Rational AppScan eXtensions Framework 和 Pyscan 提供了基础。

通过自动化 Web 应用测试流程来帮助安全审计人员和渗透测试人员快速、有效地从事他们的工作，Rational AppScan Standard Edition 这种可用作一个桌面应用或用作一项软件即服务（Software as a Service，SaaS）的软件有助于显著提高攻击者的攻击成本，从而使您的机构不再是有价值的目标。

要了解有关 Rational AppScan Standard Edition 的更多信息，请参见下列网址：

http://www.ibm.com/software/awdtools/appscan/

Rational AppScan Tester Edition

让我们从交付流程倒退一步，进入开发流程并查看一下我们如何能够将 Rational AppScan 产品整合到测试阶段。通过使用与 Rational AppScan Standard Edition 相同的漏洞检测功能，IBM Rational AppScan Tester Edition 可用作桌面应用，它提供了各项功能来帮助质量保证（QA）团队将安全性测试整合到现有的质量管理流程中，从而减轻安全专业人员的负担。由于 Rational AppScan Tester Edition 与领先的测试系统整合在一起，所以 QA 专业人员可以在测试脚本中使用其功能。他们可以在熟悉的测试环境中进行安全性检查，方便了安全性测试与功能和性能测试的同时采用。

假定 QA 机构已经知道如何区分缺陷的优先顺序、使测试可重复并报告测试覆盖范围以及部署准备程度，那么这些团队完全适合进行安全性测试。他们能在应用交付流程中帮助更早地调整安全性测试，查找和修补安全漏洞。在针对功能和性能问题进行测试和在发布中防止高成本的延迟时，他们可以这样做。
要了解有关 Rational AppScan Tester Edition 的更多信息，请参见下列网址： http://www.ibm.com/software/awdtools/appscan/tester/

Rational AppScan Enterprise Edition

除了先进的应用扫描功能以外，Rational AppScan Enterprise Edition 还提供了完善的报告和修补功能，以及与 AppScan 桌面版 Rational AppScan Standard Edition 的无缝整合。IBM Rational AppScan Enterprise Edition 是一个基于 Web 的多用户 Web 应用漏洞测试和报告解决方案，适用于需要在整个机构中执行应用扫描，同时又要保持对漏洞数据的集中控制的机构。Rational AppScan Enterprise Edition 包括 QuickScan、一个“傻瓜式”测试工具和基于计算机的综合培训，以促进在整个软件开发生命周期中采用安全性测试。

通过生成实用的安全指标、指示板和重要法规遵从性报告，Rational AppScan Enterprise Edition 有助于用户了解整体安全状态。

Rational AppScan 的扫描引擎遍历一个 Web 应用，分析和测试该应用的安全性和遵从性问题，并生成包含修复建议的可操作报告，以简化修补流程。这些高级修复建议为开发人员和安全审计人员提供了无与伦比的精确性和效率，能够帮助解决和修补扫描发现的漏洞。与领先的 QA 测试工具（包括 IBM Rational ClearQuest）、开发环境和代码扫描设备的无缝整合进一步简化了由 QA 和开发团队进行的安全性测试和修补。

使用基于 Web 的架构，Rational AppScan Enterprise Edition 的设计旨在帮助各个机构在多个利益相关者之间分配安全性测试的责任。Rational AppScan Enterprise Edition 用于需要以集中的方式进行 Web 应用安全评估并提供充分整合的解决方案集的团队。图 7 显示了 Rational AppScan Enterprise Edition 指示板。

图 7 IBM Rational AppScan Enterprise Edition 指示板视图

要了解有关 Rational AppScan Enterprise Edition 的更多信息，请参见下列网址：http://www.ibm.com/software/awdtools/appscan/enterprise/

IBM Rational 技术与全面、综合的安全平台结合

从上一节我们可以看出，Rational AppScan 产品线本身并没有覆盖整个软件开发生命周期。要创建全面、综合的安全开发平台，Rational AppScan 产品要与其他产品整合。例如，到现在为止，我们既没有包含各项要求和设计阶段，又没有过多地探讨各种跟踪和报告 AppScan 产品所发现的漏洞的产品。

在本节中，我们将说明其他 IBM 产品如何与 Rational AppScan 产品套件整合来创建完整的安全开发生命周期。虽然我们介绍了 IBM Rational 软件产品，但竞争性技术也同样能帮助达到相同的最终目标。因为并非所有创建的技术都是等同的，所以我们将各项技术分组为四个重要性层次。

安全软件开发生命周期所需的第一个技术层次等级应该不会令大家吃惊：带有源代码控制和更改请求管理的集成开发环境。几乎每个开发团队都会认识到代码规定和跟踪缺陷的重要性。在没有中央存储库来管理代码和缺陷的情况下，软件的构建可能会是一个完全临时的过程。像 Rational Application Developer for WebSphere® Software、Rational ClearQuest 和 Rational ClearCase® 之类的技术为可重复、可度量的软件创建建立一个基线。

► IBM Rational Application Developer for WebSphere Software 的设计旨在帮助开发人员快速构建高质量的 JavaTM、Java Platform, Enterprise Edition（Java EE）Web、Web 服务、门户，以及 SOA 解决方案。集成开发环境（IDE）有助于快速设计、开发、装配、测试和部署这些应用。软件的可视工具通过抽象 Java EE 编程模型来帮助减少手工编码。他们使您更轻松、更快速地完成开发项目，并让您将精力集中在创造性的软件解决方案上。

要了解有关 Rational Application Developer for WebSphere Software 的更多信息，请参见下列网址：
http://www.ibm.com/software/awdtools/appscan/tester/

► IBM Rational ClearCase 是一个行业领先的解决方案，它提供了完善的版本控制、工作区管理、平行开发支持以及版本审计，从而能够提高生产率。这个全面的软件配置管理产品提供了一个强化的集中部署模型，甚至使全球性团队更易于协同工作。当开发团队继续面临交付比以前质量更高、速度更快的软件的更大压力时， Rational ClearCase 能帮助简化软件交付流程并提高生产率。

要了解有关 Rational ClearCase 的更多信息，请参见下列网址：http://www.ibm.com/software/awdtools/clearcase/

► IBM Rational ClearQuest 提供了变更跟踪、流程自动化、报告和生命周期可追溯性的特性，以获得对软件开发生命周期的更好的可视性和控制。其设计旨在帮助更有效地管理软件生命周期。它使您能访问作出更佳决策所需的信息。它帮助您更有效地管理任务和计划，并快速响应客户需求。Rational ClearQuest 的自动化工作流能帮助您控制和实施开发流程，并帮助改善团队交流、生产率和质量。

要了解有关 Rational ClearQuest 的更多信息，请参见下列网址：http://www.ibm.com/software/awdtools/clearquest/

第二层组件涉及另外两类：需求和测试管理。Rational DOORS® 等产品使业务专业人员和架构师能够在软件内定义清晰、可度量的要求。

Rational AppScan 产品线的自动安全性测试产品使自动化在不同的软件开发生命周期阶段有效地得到使用，从而针对定义的安全漏洞进行测试。
所有这些功能与 Rational Quality Manager 这类将信息都收集到一处的产品相结合，创建了一个能够跟踪测试、测试结果和缺陷以及回答最后和最终的大问题的系统。这个大问题是：
我们准备好发布了吗？以上两个顶级组件对于交付安全软件来说是最低的要求。

► IBM Rational DOORS 专为那些想管理安全需求、编写优良的用例、提高可追溯性、加强协作、减少项目风险和提高质量的项目团队而设计。它为分布式团队提供了可伸缩而又快速的 Web 界面，并为企业部署提供了增强的高度安全的模型。因此，对于业务分析师、架构师、设计人员、开发人员和测试人员来说，它支持定制的需求访问，同时允

许在安全软件开发生命周期的多个阶段中进行整合。

要了解有关 Rational DOORS 的更多信息，请参见下列网址：http://www-01.ibm.com/software/awdtools/doors/

- IBM Rational Quality Manager 是一个基于 Web 的集中测试管理环境。它适用于业务、系统和 IT 决策者和质量专业人士。他们寻求一种用于测试计划、工作流控制、跟踪和指标报告的协作和可定制的解决方案，这个解决方案能够定量分析项目决策和交付成果如何影响业务目标以及如何与业务目标保持一致。其设计旨在通过允许团队无缝共享信息、借助自动化加速项目计划，以及报告项目指标以便做出明智的发布决策，从而帮助他们进行协作。

  要了解有关 Rational Quality Manager 的更多信息，请参见下列网址：http://www.ibm.com/software/awdtools/rqm/

近年来，传统的“预先大量设计（big design up front，BDUF）”瀑布式方法的开发已经转变为迭代式和增量式的敏捷方法。这种转换使企业能够通过回答有关软件成功和早期生存能力的重要问题来实现实际成本的节约。它还使企业能够对客户建议、与软件相联系的需求做出更快速的反应。但是，这还意味着创建更多的构造。

Rational Build Forge 等产品提供了采用这种新方式构建软件的解决方案。它使开发团队能够维护自动化的构建管理系统，该系统提供了这种新开发方法所需的及时反馈。将这一点看作构建安全软件的重要组件的原因在于，与 Rational AppScan 的整合允许安全分析在每个构建版本上迭代完成。它提供了早期且有价值的反馈，降低了软件开发生命周期后期的成本和风险。

- IBM Rational Build Forge 是一个自适应的流程执行框架。该框架自动化、编排、管理和跟踪软件开发的每种装配线内每次传送之间的所有流程，同时创建一个自动化软件工厂。Rational Build Forge 整合到您的当前环境中，并支持主流开发语言、脚本、工具和平台。它使您能够继续使用现有的投资，同时在流程自动化、加速、通知和计划方面添加了有价值的功能。

  要了解有关 Rational Build Forge 的更多信息，请参见下列网址：http://www.ibm.com/software/awdtools/buildforge/

与安全性直接相关的下一层组件是架构和资产管理系统。Rational Software Architect for WebSphere Software 等架构管理软件使开发团队能够快速设计和重用成熟的、能实现组件间安全交互的安全设计。Rational Asset Manager 等资产管理系统使开发团队能够维护经过认证的安全组件库。开发人员可以转向这个组件库来执行身份验证、授权、输入验证、日志记录、审计等普通任务。如果实现错误，该组件库还可能危及系统的安全。

当这种成熟的资产库不符合要求且必须编写一个定制组件时，安全团队会被召集起来进行评估。

- IBM Rational Software Architect for WebSphere Software 是一个功能强大、综合的设计和开发环境。它能帮助 IT 架构师和开发人员跨团队、跨不同的技术专家领域以及在全球了解、设计、管理和发展解决方案。其抽象、分析和报告功能旨在使交流和协作更高效。此外，其自动化和智能编辑工具能够帮助提高生产率、增强架构控制并使 Java 和 Java，J2EE、Web 服务、SOA 和 Web 2.0 应用从设计到代码的体验变得轻松。

  要了解有关 Rational Software Architect for WebSphere Software 的更多信息，请参见下列网址：

  http://www.ibm.com/software/awdtools/swarchitect/websphere/

- IBM Rational Asset Manager 通过促进与软件开发相关的所有类型的资产重用来降低软件开发成本并提高质量。它是一个协作性软件开发资产管理解决方案，能帮助定位、共享和跟踪跨业务和部署团队的资产。Rational Asset Manager 使组织级分布式开发团队能

够确定、管理和治理软件资产（包括作为 SOA 计划组成部分的服务）的设计、开发和消费。 该软件推动了协作软件资产的开发、部署和使用，帮助 IT 机构交付创新的 IT 解决方案，同时控制了成本、降低了应用延迟并提高了业务的灵活性和响应能力。

要了解有关 Rational Asset Manager 的更多信息，请参见下列网址：http://www.ibm.com/software/awdtools/ram/

当您在操作环境内达到部署阶段后，最后一层产品能使您保持所交付的产品及其数据的安全性。IBM OptimTM Data Privacy Solution 能确保您的客户数据在发生泄漏的情况下得到充分的保护和屏蔽。为了自动化保持企业内网的安全性的任务，可以将 Proventia® Network Enterprise Scanner 和 Proventia Network MFS 纳入到您的网络基础设施中。为了保持服务器和桌面得到充分修补，IBM Tivoli® Security Compliance Manager 提供了一个跟踪机器安全的集中解决方案。

- IBM Optim Data Privacy Solution 能够保护客户数据的隐私。取消机密数据的标识是保护隐私并支持 HIPAA、DPP、PIPEDA、PCI DSS 等法规的遵从性的一种最佳方法。Optim Data Privacy Solution 交付了强大的数据转换功能，可屏蔽机密企业数据，使您能够安全地将其用于应用测试。您可以通过应用简单的数据屏蔽技术保护易受攻击的测试环境，也可应用预先打包的转换算法来处理复杂的数据元素，如信用卡号码、电子邮件地址和身份证号码。

  要了解有关 Optim Data Privacy Solution 的更多信息，请参见下列网址：

  http://www.ibm.com/software/data/data-management/optim/data-privacy-solution/

- IBM Proventia Network Enterprise Scanner 使您能够了解您的网络上正在传输的数据以及潜在问题位于何处。该解决方案对确定和管理风险具有重大意义。遵守安全法规并简略说明修补措施可能是代价高昂、劳动力密集型的。Proventia Network Enterprise Scanner 使您能在管理网络漏洞方面节省成本和时间。Proventia Network Enterprise Scanner 能够帮助确保创收服务的可用性并通过确定风险并区分其优先次序、分配保护活动和报告结果来保护您的企业数据。

  要了解有关 Proventia Network Enterprise Scanner 的更多信息，请参见下列网址：

  http://www.ibm.com/services/us/index.wss/offering/iss/a1027216

- IBM Proventia Network Multi-Function Security（MFS）是一种统一威胁管理（UTM）设备，在网关和网络级提供保护，同时不会影响网络带宽或可用性。它可以一次性防御多种威胁，比如未授权的访问、网络攻击、恶意代码、混合型威胁、基于内容的攻击、间谍软件和网络钓鱼攻击。Proventia Network MFS 将这些业界最佳的（best-of-breed）安全模块整合到一个单独的高性能和易用的 UTM 设备中：防火墙/VPN、入侵预防、防病毒、防垃圾邮件、Web/URL 过滤器和应用保护。

  要了解有关 Proventia Network Multi-Function Security 的更多信息，请参见下列网址：

  http://www.ibm.com/services/us/index.wss/offering/iss/a1027111

- IBM Tivoli Security Compliance Manager 确定安全漏洞和安全策略违规。它通过定义一致的安全策略和检测这些限定的安全策略的遵从性来保护您的业务，避免攻击者凭借易受攻击的软件配置进行的攻击。它自动扫描服务器和桌面系统，能帮助减少与手动安全性检查相关的成本和时间。

  Tivoli Security Compliance Manager 向安全官员和遵从性审计人员报告详细的业务安全状态信息，以便他们能采取适当的步骤使各个系统和部门合规。它在安全事件造成重大损失前识别软件安全漏洞，改善业务经营并通过自动化和集中方式帮助提高效率。Tivoli Security Compliance Manager 通过自动化遵从性任务、监测通信、减少人为错误和降低遵从性成本来帮助解决法规和标准遵从性问题。

要了解有关 Tivoli Security Compliance Manager 的更多信息，请参见下列网址：

http://www.ibm.com/software/tivoli/products/security-compliance-mgr/

图 8 显示了如何将所有这些技术结合到一个完整的安全开发生命周期中。通过将与安全相关的 IBM 产品整合到开发生命周期的各个步骤中，您的最终产品的安全性可以得到显著增强。

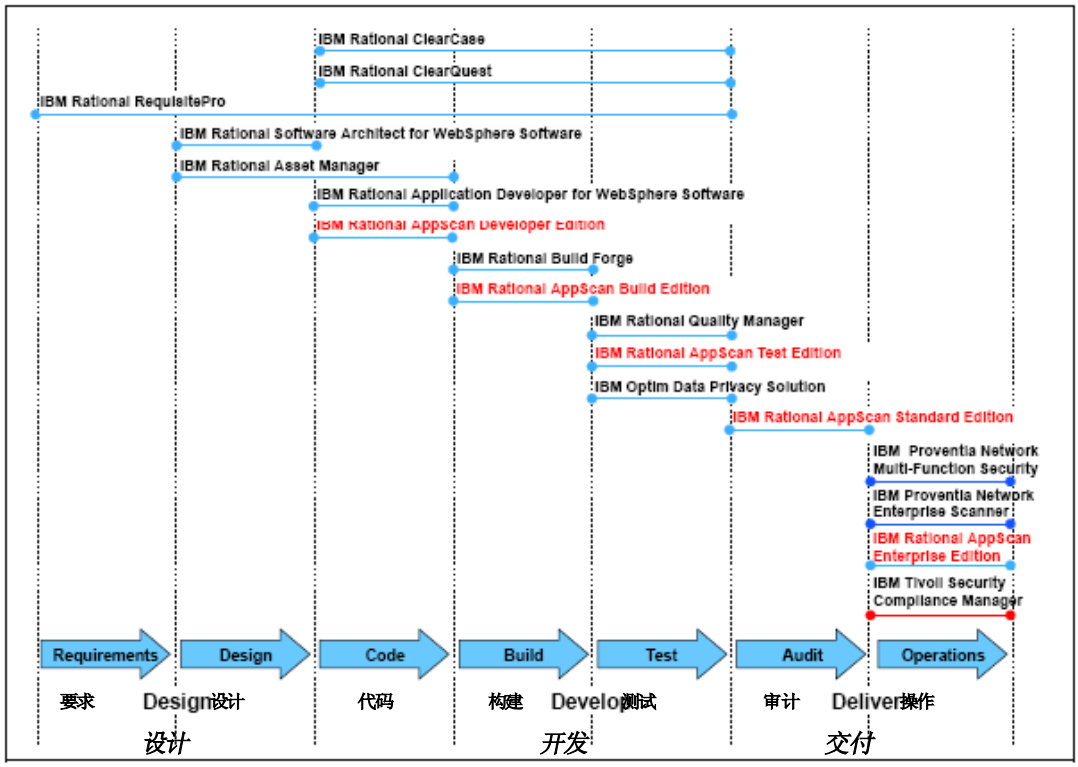


图 8 结合 IBM 技术来创建全面、综合的安全平台

业务场景：实现 Web 应用安全的分步方法

在本节中，我们研究这样一种业务场景：一家原本没有安全测试的组织在其软件开发生命周期中整合了最新的安全测试。在本场景中，获悉比它规模更大的竞争对手出现了一些安全问题之后，这家虚构的公司决定考虑 IBM 产品来提高安全性，以避免其竞争对手遭受的负面效应，并提供高质量、安全且基于 Web 的产品以使自己脱颖而出。

与 IBM 商议之后，该公司决定逐步将其现有的不安全的软件开发生命周期转化为具有安全意识的生命周期，以生产高质量、安全且基于 Web 的产品。因为该公司希望快速取得成果，即直接了解其现时安全状况，所以他们最初选择通过 Rational AppScan On Demand Production Site Monitoring 与外包审计团队合作。此服务允许他们获得直接反馈，同时准备将安全性深入整合到系统中。借此，他们获得了正确的资源和正确的专家，可进一步开发其特有安全方案。在执行此工作的同时，他们开始

针对安全性培训所有开发生命周期相关人员。此培训依靠基于 Web 的嵌入式培训模块来推行，这些模块有助于说明漏洞、展示 Rational AppScan On Demand 所实现的业绩。

该公司培训了一批自己的员工，完全可以不依赖于服务，而能够自己执行安全测试，他们获得了 Rational AppScan Standard Edition 的必要产品许可，构建了自己的内部安全审计团队。使用 Rational AppScan Standard Edition 的自动扫描功能，内部审计人员在开发生命周期结束时发现了漏洞。他们将漏洞报告给开发人员，以便后者在产品发布前进行修复。

由于持续对整家公司进行安全性培训，而且内部安全审计团队向开发人员提交了反馈，所以该公司的开发人员很快就能够承担更大的安全责任了。为了在开发生命周期之初就引入安全测试，开发人员配备了 Rational AppScan Source Edition。这允许他们在编码时自动发现漏洞，并使他们了解必须避免的、与安全相关的、不好的开发实践。通过在开发生命周期中较早地捕获漏洞，安全审计团队的资源得到了释放，他们可以关注更复杂的漏洞，因此进一步增强了公司的安全水平。

接着，更多安全测试被自动化，并整合到构建和测试流程中。Rational AppScan Build Edition 和 AppScan Test Edition 用于实现这一目的。这时，公司已经在整个软件开发生命周期中分布了安全测试。

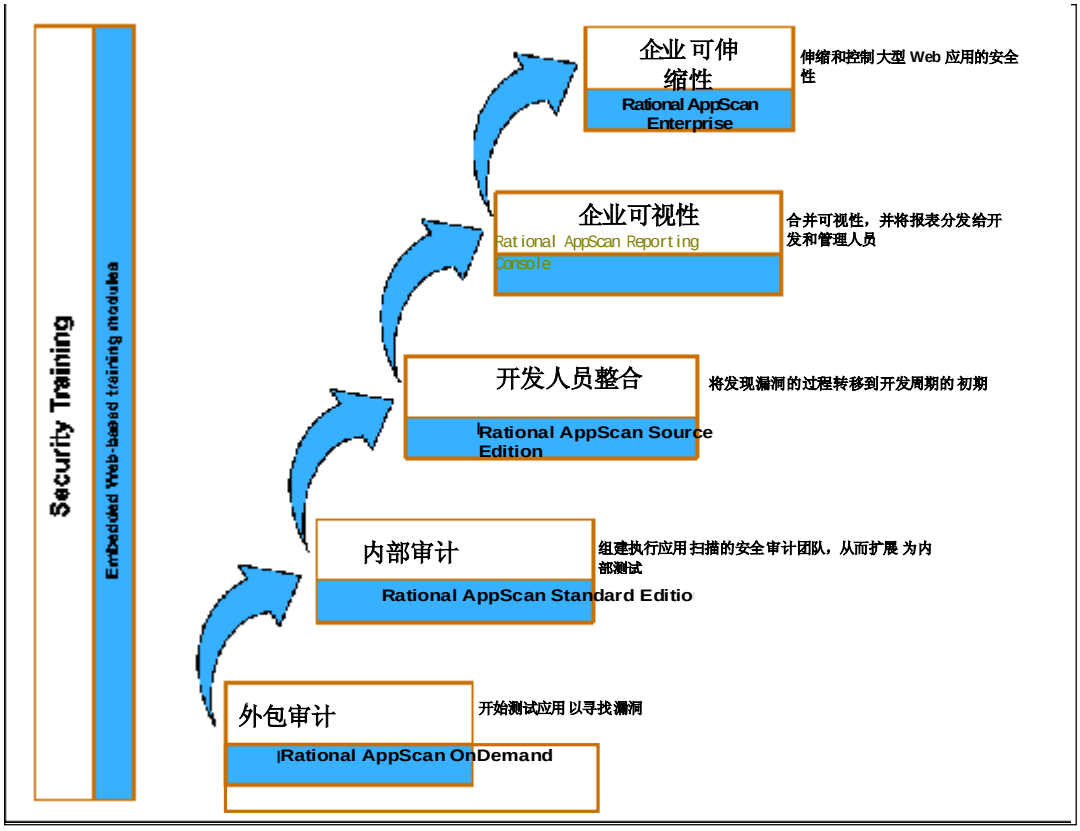
为了让开发和管理人员更明确地了解开发项目进展中的安全进程，该公司开始将 Rational AppScan Reporting Console 用于集中收集安全数据。利用 Rational AppScan Reporting Console，他们能够纵览其软件开发生命周期中使用的不同软件工具创建的所有安全报表，从而提高了可视性，并进一步增强了安全性。

集中所有的漏洞扫描报表，这使该公司已对安全问题充满自信，同时保持其安全性在掌控之中。该公司在 Web 应用领域将其安全性作为关键卖点，展示自身优于竞争对手的不同，从而吸引了市场的注意，开始迅速成长。这导致该公司整体需要充分可伸缩的安全测试解决方案。

为了适应其迅速发展的 Web 环境的规模，该公司将 Rational AppScan Enterprise Edition 引入到软件开发生命周期中。这种可伸缩的企业架构使他们能运用分布式扫描代理，持续扫描大量的 Web 应用。他们能从 Rational AppScan Enterprise Edition 中的一个集中报告点控制那些扫描代理。这样他们就能持续快速发展，获得无限扩充的安全扫描和审计能力。

图 9 概述了该公司如何逐步将安全性整合到软件开发生命周期中，转变成为拥有最新安全测试的快速成长的公司。

图 9 逐步实现 Web 应用安全的方法



图字：
Security Training：安全性培训
Embedded Web-based training modules：基于 Web 的嵌入式培训模块

结束语

我们看到互联网上有这样的变化，从渴求出名的黑客执行的破坏活动，发展到有组织的数据和身份窃贼为获利而进行的欺诈，因此企业领导必须将 Web 应用安全性视为业务成功的关键指标。

这份 IBM Redguide 从攻击者的角度研究了 Web 应用安全性。通过窥探攻击者的动机和操作方式，我们示范了他们如何依靠攻击互联网上的组织而获取金钱。我们解释了对他们而言为何攻击 Web 应用是有利可图的生意，以及他们如何使用自动化来削减成本、获取更大利益。

接着，我们向您展示了如何采用与他们进行自动化攻击相同的方式，在软件开发生命周期中依靠自动化来保护您的组织。我们介绍了 IBM Rational AppScan 产品线，说明了您如何将此产品线整合至您的软件开发生命周期中，从而借助最新的安全测试，改善您的 Web 应用的整体安全状况。

最后，我们展示了这样一个场景，一家没有 Web 应用安全测试或知识的公司，转变为一家将最新 Web 应用安全测试以一种可伸缩且可控制的方式充分整合到其软件开发生命周期中、具备安

全意识的公司。我们说明了交付高质量的安全 Web 应用如何使一家公司在竞争对手中脱颖而出，并增强其市场地位。

其他资源中的更多信息

参考本指南中已引用的以下资料，了解更多信息：

1. IBM Internet Security Systems X-Force 2008 Trend & Risk Report
   http://www.ibm.com/services/us/iss/xforce/trendreports/xforce-2008-annual-report . pdf
2. Sharon Gaudin，“Estimates Put T.J. Maxx Security Fiasco At $4.5 Billion，”Information Week，2007 年 5 月 2 日
   http://www.informationweek.com/news/security/showArticle.jhtml?articleID=199203277
3. Nate Mook，“Cross-Site Scripting Worm Hits MySpace，”Betanews，2005 年 10 月 13 日 *http://www.betanews.com/article/CrossSite-Scripting-Worm-Hits-MySpace/1129232391*
4. Gary Warner，“Radical Muslim Hackers Declare CyberWar on Israel，”CyberCrime & Doing Time blog，2008 年 12 月 30 日
   http://garwarner.blogspot.com/2008/12/muslim-hackers-declare-cyberwar-on.html
5. The MITRE Corporation，“Common Vulnerabilities and Exposures”
   http://cve.mitre.org/
6. Dan Verton，“Airline Web sites seen as riddled with security holes，”Computer World，2002 年 2 月 4 日
   http://www.computerworld.com/action/article.do?command=viewArticleBasic&article Id=67973
7. Web 应用安全协会，“Web Application Security Statistics”
   http://www.webappsec.org/projects/statistics/
8. Roi Saltzman 和 Adi Sharabani，Active Man in the Middle Attacks:A Security Advisory，IBM Rational Application Security Group，2009 年 2 月 27 日
   http://blog.watchfire.com/AMitM.pdf
9. 开放 Web 应用安全项目（OWASP），“SQL Injection”
   http://www.owasp.org/index.php/SQL_injection
10. Cisco Systems，“Understanding SQL Injection”
    http://www.cisco.com/web/about/security/intelligence/sql_injection.html
11. Microsoft® SQL Server® 2008 Books Online，“SQL Injection”
    http://msdn.microsoft.com/en-us/library/ms161953.aspx
12. RSnake，“XSS (Cross-Site Scripting) Cheat Sheet”
    http://ha.ckers.org/xss.html
13. Cgisecurity.com，“The Cross-Site Scripting (XSS) FAQ”

http://www.cgisecurity.com/xss-faq.html

14.OWASP

http://www.tpc.org.

1 5.Web Application Security Consortium (home page)

http://www.webappsec.org/

1 6.Joris Evers，“Macworld crack offers VIP passes, hacker says，”CNET News，2007 年 1 月 12 日

http://news.cnet.com/2100-1002_3-6149994.html?part=rss&tag=2547-1_3-0-5&subj=news

17. Federal Information Processing Standards Publications（FIPS 主页），Information Te chnology Laboratory

http://www.itl.nist.gov/fipspubs/

本文创作团队

本文由国际技术支持组织（International Technical Support Organization，ITSO）中来自世界各地的专家组成的团队编写。

Frederik De Keukelaere，IBM Research 日本东京研究实验室研究员。他是 Security and Web Platform 小组的成员，从事下一代 Web 安全模型研究。他目前的研究兴趣在于可用的 Web 安全性。他在 Web 应用安全性、Web 以及多媒体技术方面有超过 7 年的研究经验。他曾积极参与了多个标准组织，例如 OpenAjax Alliance 和 Moving Picture Experts Group (MPEG)，并为之做出了贡献。他是数个 ISO/IEC 标准的编者，并拥有 5 项 ISO/IEC MPEG-21 杰出技术贡献奖。在 2006 年加入 IBM 之前，他在比利时多媒体实验室，是 BroadBand Technology 的交叉学科（Interdisciplinary）协会成员。在此期间，他获得了比利时根特大学计算机科学工程专业博士学位。

Danny Allan，IBM Rational 安全性研究主管。由于 2007 年 7 月对 Web 应用安全和遵从性领域的领袖 Watchfire 的收购，Danny 来到了 Rational。他具有超过 8 年的业务和安全技术相关经验，包括曾为加拿大最大的大学之一从事渗透测试和内部系统补救。在安全性研究员这一角色上，他密切参与企业的全球客户部署，研究和评估技术，并协助定义和推荐战略方向。Dann y 担任着数个面向客户的关键职位，包括团队领导、咨询服务以及销售工程师。他已发表多篇白皮书和文章，并加入了行业工作组。他还经常在安全性活动中发言，接受包括 Associated Pr ess、Bloomberg 以及 Wall Street Journal 在内的重要媒体的拜访，抒发关于 Web 应用安全性的观点。Danny 拥有卡尔顿大学信息系统专业的商学学士学位。

Axel Buecker，德克萨斯州奥斯汀 ITSO 的认证咨询软件 IT 专家（Certified Consulting Sof tware IT Specialist）。他就软件安全架构和网络计算技术领域撰写了大量 IBM 课程，并在全球教授。他拥有德国不莱梅大学计算机科学专业学位。他在与工作站和系统管理、网络计算以及电子商务解决方案相关的多个领域拥有 22 年的经验。在 2000 年 3 月加入 ITSO 之前，Axel 在德国 IBM 担任软件安全架构方面的高级 IT 专家。

感谢以下人员对本项目的贡献：

Emma Jacobs，ITSO，IBM U.S.
Gary Vincent，IBM U.S.

注意事项

本信息适用于在美国提供的产品和服务。

IBM 可能不在其他国家/地区提供本文档讨论的产品、服务或功能。咨询您本地的 IBM 代表，了解有关本地区当前可用产品和服务的信息。对 IBM 产品、程序或服务的任何引用不声明或暗示只可以使用该 IBM 产品、程序或服务。也可以使用任何不破坏 IBM 知识产权的类似产品、程序或服务。然而，评估或验证任何非 IBM 产品、程序或服务属于用户自己的责任。

对于本文档中描述的主题内容，IBM 可能具有专利或正在申请专利。本文档的描述未赋予您针对这些专利的任何许可。您可以以书面形式将许可查询发送给：
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

以下内容不适用于英国或者其他与本地法律不一致的国家：国际商业机器公司根据“现状”提供本出版物，不提供任何明确或隐含的担保，包括但不限于关于非侵权、适销性、符合特定用途的适用性所有隐含担保。某些国家在某些事务中不允许明确或隐含担保的免责声明，因此，此声明可能不适合您。

本信息可能包含技术错误或排版错误。这里的信息会定期变更，这些变更将合并到本出版物的新版本中。IBM 可能随时对产品和/或程序做出改进和/或变更，恕不通知。

本信息中对非 IBM Web 站点的引用仅出于方便考虑，不能以任何方式看作对这些 Web 站点的认可。这些 Web 站点上的内容不是本 IBM 产品资源的一部分，使用这些 Web 站点时风险自负。

IBM 可能以它自己认为合适的方式使用或分发您提供的信息，而不会承担对您的任何责任。

有关非 IBM 产品的信息是通过这些产品的提供商、他们发布的公告或其他公共可用的来源获得的。IBM 没有测试过这些产品，不能确认与非 IBM 产品相关的性能、兼容性或任何其他声明的准确性。关于非 IBM 产品功能的问题应该由这些产品的提供商解决。

本信息包含日常业务运营中使用的数据和报告的示例。为了尽可能完整的阐释它们，这些示例包括个人、公司、商标和产品的名称。所有这些名称都是虚构的，如果同实际企业使用的名称和地址雷同，纯属巧合。

版权许可：

本文包含使用源语言的样本应用程序，演示了在多种操作平台上的编程技巧。为了开发、使用、推广或分发符合操作平台应用编程接口（样本程序正是为之编写）的应用程序，您可以以任何形式复制、修改和分发这些样本程序，而无须向 IBM 支付费用。这些示例未在所有环境中经过彻底测试。因此，IBM 不能保证或暗示这些程序的可靠性、有效性或功能性。

本文档，REDP-4530-00，创建或更新于 2010 年 12 月 12 日。

红皮书®

商标