Rational. software

IBM

利用 IBM Rational AppScan 改善應用程式開發生命週期的安全性

對象

企業領導者 **Redguide**

Federik De Keukelaere
Danny Allan
Axel Buecker

- **瞭解駭客如何選擇目標攻擊並獲得利益**
- **瞭解自動化應用程式安全性測試的價值**
- **在您的開發生命週期中部署 IBM Rational 應用程式安全性工具**

Redbooks

執行概要

Internet 上的駭客已經從單純爲出名而進行的破壞演變到詐欺，又進化到了有組織、有利可圖的資料和身份竊取。伴隨著這樣的發展演化，企業領導者必須將其應用程式的安全性視爲業務成功的重要績效指標。

在這份 IBM® Redguide™ 中，我們分析了您的組織應如何評估駭客入侵系統的風險。我們還說明了您的組織如何實現安全性測試並整合解決方案，進而改善安全性，保護您的資訊資產。

在這份 Redguide 的第一部分中，我們將討論如何評估您的組織面臨的風險。我們將解釋您的組織爲什麼會成爲攻擊的目標，以及幕後的攻擊者。我們將展示成功的攻擊可能會給您的組織帶來的影響。我們將展示關於 Web 應用程式漏洞的最新趨勢和統計資訊，以及被竊資訊的幕後交易情況。我們還會給出應用程式可能會受到攻擊的領域技術概觀，討論兩種最常見的 Web 應用程式漏洞。

在這份 Redguide 的下一部分中，我們會介紹 Web 應用程式的軟體開發生命週期，展示如何將安全性納入這樣的生命週期。我們提供了將 Web 應用程式安全性測試整合到您的軟體開發生命週期之中的詳盡方法。此外，我們還展示了在軟體開發生命週期中利用 IBM Rational® 產品的方法與時機，以根據業務需求改進組織的安全性。

最後，我們以一個業務情節總結了這份指南，在此情節中，一個未使用任何 Web 應用程式安全性測試的組織逐步轉變爲交付優質、安全產品的組織。

揭露 IT 攻擊模式的基本要素及其對組織的影響

針對 Web 應用程式的攻擊數量不斷增加，這一事實對 IT 企業內的大多數人來說已經不再新鮮。對您的組織來說，最重要的是理解未來可能遭遇的風險，以便採取恰當的措施，規避和控制此類風險。

瞭解以下問題的答案可幫助您實現此目標：

- 我的組織爲什麼面臨著被攻擊的可能性？
- 我的組織中存在能被利用的漏洞的機率有多大？
- 如果組織被成功攻擊，將會遭受怎樣的損失？
- 我怎樣才能更好地保護我的組織？

遺憾的是，業餘玩家、學生或駭客僅僅出於樂趣，而入侵企業資訊系統的時代已經過去了。如今攻擊者的獲利動機更強，他們往往是國際性的犯罪集團，透過竊取財務資訊或個人資料謀生。當今的威脅比過去的安全威脅更加複雜也更加危險，但從某些方面來說，可預測的程度也更高。業餘駭客可能對存在的任何安全漏洞都有興趣，但真正的電腦犯罪份子對能夠提供較高投資報酬率的漏洞更感興趣。簡而言之，一切都與金錢有關。

IBM Internet Security Systems X-Force® 研究團隊發表的關於 Web 應用程式安全漏洞的研究顯示，在過去幾年中，已報告的 Web 應用程式漏洞在所有報告的安全漏洞中所占的百分比顯著增加 [1]。您的組織中存在的這些漏洞被利用的可能性，取決於利用漏洞攻擊的困難度。由於現在的攻擊者通常是以盈利爲目的的組織，因而大多數攻擊活動都是自動完成的，這是爲了儘可能地降低成本。因而，如果您的 Web 應用程式能夠透過使用自動化的工具輕鬆攻擊，被利用的機率就較高。

在您的組織被成功攻擊之後，惡意的攻擊者能夠進行多種破壞。除了因資料損失而造成的明顯損失之外，您還可能面臨著高額罰金、通報客戶的高昂費用、因負面消息公開而造成的嚴重商譽損害，也可能會捲入對您的企業發起的民事訴訟。對於大型企業來說，此類損失很快就會達到數億美元 [2]。

幸運的是，將 Web 應用程式安全性提升到一定的水準，使其從經濟角度上不再是值得費力進行攻擊的目標，並非不可能的任務。在這份 Redguide 中，我們將提供這些問題的答案。透過將 Rational AppScan® 產品整合到您的軟體開發生命週期之中，我們將爲您展示您應如何保護組織免受當前面臨的眾多威脅的侵害。

瞭解攻擊者

不同類型的攻擊者在攻擊您的企業時有著不同的動機。第一類攻擊者稱爲腳本頑童（Script Kiddies, 也叫做 H4ck0rZ），如果您的企業十分知名，而且攻擊您的企業將提高他們在駭客社群內的聲望，那麼他們就可能會選擇您的企業作爲目標。

第二類攻擊者爲有針對性的攻擊者，他們可能會出於某些原則、信念、間諜活動或政治動機攻擊您的組織。這一組攻擊者通常擁有希望達成的明確目標，並相應地選擇攻擊目標。

第三類攻擊者爲有組織的犯罪，他們透過攻擊防禦薄弱且能將其攻擊轉變爲金錢的任何組織來盈利。他們並非專門針對您的組織，但如果有可能，他們就會利用您的組織爲自己謀利。

請參考以下美國 FBI 取締買賣信用卡資料的國際性地下網路論壇的新聞：

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

腳本頑童

第一類攻擊者是著迷於吸引媒體注意的駭客。這些駭客攻擊知名目標，希望能以此在駭客社群中佔據一席之地，也有可能僅僅是出於樂趣而攻擊 Web 網站。常見攻擊類型是塗改破壞（defacement）和阻斷服務(DoS)攻擊，他們希望透過這樣的方式在駭客社群中揚名。

此類攻擊中，最著名的 Web 應用程式攻擊就是 2005 年的 MySpace Samy 蠕蟲 [3]。這個蠕蟲病毒的作者使用跨網站腳本（XSS）攻擊來創建蠕蟲，在 MySpace 社交網路上傳播，在不到 24 小時的時間內，“Samy”就擁有了超過 100 萬名“好友”。這是一次精心策劃的攻擊，如今也已成爲 Web 應用程式安全性領域中著名的一個研究案例。儘管 Samy 蠕蟲純粹是爲了樂趣而創建的，但 MySpace 卻不得不下線以清除這種蠕蟲。

這個例子說明，如果您可能會給這組攻擊者帶來他們需要的知名度，那麼他們就會攻擊您的企業。他們會跨越國界惡作劇，在此過程中建立自己的知名度。

有針對性的攻擊者

第二類攻擊者採用的是更有針對性的方式。典型的例子就是爲間諜活動（國家、州和企業）、政治或宗教信仰等目的而進行的攻擊。有針對性的攻擊者往往是由使用網際網路作爲戰爭前線的組織聘請的。這些組織通常會竊取特定的資料和智慧資產，或試圖散佈其政治或宗教信仰。

資安社群已經注意到了一些以宣傳爲目的的戰爭。例如，在 Gary Warner [4] 的部落格中，他寫下了一些此類以宣傳爲目的的電子戰爭：

> 最初的電子宣傳戰爭是由中國駭客在 2001 年 5 月發起的，是在中國戰鬥機與美國海軍偵察機發生衝突之後。數萬個美國網站被中國駭客塗改破壞，爲這次事件譴責美國。最近，穆斯林駭客採用了這項技術，首先是在 2006 年 2 月，在關於先知穆罕默德的漫畫發表之後，塗改破壞了數以千計的丹麥和美國網站，在 2006 年 8 月轟炸黎巴嫩之後，又開展了針對以色列和美國網站的攻擊。

如果您身處政府機構或從某些方面構成了這些駭客幕後組織的關注點，那麼有針對性的駭客就會成爲您的組織的真正威脅。他們會投入大量精力，攻擊能幫助達成目的的具體目標。

有組織的犯罪

第三類攻擊者爲有組織的犯罪。他們尋求能迅速轉化爲利潤的資訊，大致上，這種迅速的投資回報，指的多半是竊取消費者的信用卡資訊和銀行帳戶存取憑證。

有些時候，攻擊者會找到一些方法，從企業伺服器和網路中獲得大量此類資料，直接在用戶 PC 上運行的間諜軟體也會竊取大量此類資訊。具有高水準 patch 和保護機制的企業能夠給這些攻擊者設定障礙。然而，缺乏保護、無定期上 patch 習慣、缺乏安全性意識的用戶依然是易於攻擊的目標。有組織的犯罪使用一個組織的伺服器作爲啓動平台，展開對目標用戶的攻擊。例如，他們可能會利用您的論壇來傳播惡意文件，如精心製作的 PDF 檔和多媒體應用程式（如 Flash），這些檔中包含內嵌的漏洞，可在客戶的 PC 上安裝惡意軟體。

此外，某些類型的企業應用程式，即自行開發創建的軟體，如 Web 應用程式等，也是此類犯罪型攻擊者的高利潤、低成本的攻擊目標。商業性和 Open Source 的 Web 應用程式中發現的漏洞不計其數，而大部分都沒有可用的 patch。這與無數同樣脆弱的 Web 應用程式相結合（但永遠不會經歷漏洞檢測，更不用說修補漏洞了），就成爲了企業安全性的致命傷。攻擊者依然以 Web 應用程式的漏洞爲首要目標，特別是 SQL 注入，除了竊取資訊外，當毫無察覺的用戶前往存在漏洞的網站時，往往也會被植入惡意軟體。

可以確信，試圖透過攻擊電腦系統來盈利的有組織的犯罪並非特別關注您的系統。但若您的應用程式具備易於透過自動化工具發現的漏洞，您就很有可能成爲攻擊的目標。

犯罪經濟學 101

爲了更進一步地瞭解犯罪組織如何透過攻擊組織來盈利，這一節內容將分析犯罪經濟學。[1] 在基本的微觀經濟學層面上，對電腦犯罪機會的理解，源於考慮利用一個漏洞能夠帶來的收入與利用此漏洞的成本的對比關係。顯然，能夠以較低的成本帶來更高收入的漏洞更受攻擊者歡迎。收入（機會）和成本都是由一組複雜的要素構成的。部分此類要素會被應用程式的安全性影響。

犯罪機會

利用一個漏洞能帶來的實際收入是有存在漏洞的主機的安裝群體規模和控制各主機對於攻擊者的價值共同決定的。這通常是取決於主機包含的資訊和攻擊者在黑市上銷售這些資訊的價格。

當一個漏洞初次被揭露時，存在漏洞的電腦的安裝群體規模可能相當大。如果控制此類電腦的價值同樣高，攻擊者在理論上就有著極高的收入機遇。這樣的情況可能會激發安全行業迅速推廣補丁、減少安裝群體規模的工作。如果安全行業的努力

[1] 關於犯罪經濟學的討論主要摘自 IBM Internet Security Systems，“X-Force 2008 Trend & Risk Report” [1]

切實有效，攻擊者潛在的攻擊總收入就會變得極低，最終導致攻擊者不會或者不願意去實現攻擊。同時，也存在另外一種情況，有漏洞的電腦的安裝群體規模雖然較大，但控制運行這些機器能帶來的價值常並不高，攻擊者利用漏洞的動機都不強烈。

犯罪成本

透過利用漏洞創收的成本也是由多種因素構成的。首先是獲得漏洞的成本，這取決於可利用的漏洞是否公開。其次是與利用漏洞進行攻擊的相關難度。與合法企業的情況相似，犯罪組織也有著圍繞可重複的環境和可自動化的任務構建的運作流程。

適合現有流程、可使用現有自動化工具進行利用，進行攻擊的漏洞更便於犯罪者賺錢。而需要開發新流程或軟體才能發現和利用的漏洞對犯罪者的吸引力就比較低，尤其是在此類漏洞未來重複出現的可能性更低的情況下則更是如此。即使對於犯罪者來說，開發新攻擊方法來利用一類新漏洞是有意義的，但是大規模攻擊所需的時間也比直接適合現有流程的漏洞更長。所以現有的，公開的漏洞對攻擊者的吸引力是比較大的。

賺錢

與其他任何企業相似，犯罪組織必須計算一次可能的攻擊的價值，以判斷是否值得開展攻擊。爲了比較攻擊價值的計算與創建 Web 應用程式的價值的計算，我們首先來觀察一下 Web 應用程式的價值。我們要如何確定 Web 應用程式的價值？

向潛在客戶顯示資訊的價值？	$10
透過客戶自助服務降低成本的價值？	$100
在富 UI 中交付業務功能的價值？	$1000
創建富協作式用戶社區的價值？	無價！

透過這樣的計算，可以明確，如果犯罪組織希望在這項業務中盈利，富協作式用戶社區是合理的選擇。
無論憂心忡忡的安全人員存在怎樣的安全性顧慮，都很可能要實現富協作式用戶社區。安全人員將竭力阻礙豐富的互動式環境，因爲此類環境會帶來真正的威脅。然而，必須做出權衡，以正確認識價值。

觀察犯罪組織的業務模型，我們可以進行類似計算，瞭解其價值：

竊取一個信用卡號碼的價值？	$0.10
竊取一個電子郵件密碼的價值？	$4
竊取一個個人銀行帳戶憑據的價值？	$10
自動隨機攻擊系統，收集可出售資訊的價值？	無價！

由於犯罪組織往往透過銷售攻擊 Web 應用程式獲得的資訊來盈利，因而我們將使用此類資訊的當前交易價值來進行計算，如第 6 頁的表 1 所示。2 該表顯示了關於不同商品的價值、需求和銷售的頻率以及其價格範圍。

[2] 第 6 頁表 1 中的資訊是由 Symantec™ 在 Symantec Report on the Underground Economy 中發佈的。2008 年 11 月，第 20 頁 (http://eval .symantec.com/mktginfo/enterprise/white_papers/b-whi tepaper_underground_economy_report_1 1-2008-14525717. en-us. pdf).在許可的情況下轉載。

表 1 銷售和有需求的商品與服務

銷售	需求等級	商品與服務	銷售百分比	需求百分比	價格範圍
1	1	銀行帳戶憑據	18%	14%	$0.10 - $1 .000
2	2	帶有 CVV2 號碼的信用卡資訊	16%	13%	$0.50 - $12
3	5	信用卡	13%	8%	$0.10 - $25
4	6	電子郵件位址	6%	7%	$0.30/MB - $40/MB
5	14	電子郵件密碼	6%	2%	$4 - $30
6	3	完整的身份	5%	9%	$0.90 - $25
7	4	信用卡套現服務	5%	8%	總價值的 8% - 50%
8	12	代理	4%	3%	$0.30 - $20
9	8	網路詐騙	3%	6%	託管爲 $2.50 - $100/周 設計爲 $5 - $20
10	7	郵件程式	3%	6%	$1 - $25

由於單位資訊的售價相對較低，因而盈利實際上是透過銷售大量資料實現的。因而，攻擊者必須在攻擊和收集資料的成本與此類資料在市場上的價值之間做出權衡。

與其他任何企業相同，自動化也是用於削減成本的一種工具。出於此原因，有組織的犯罪對各類組織的威脅最大，因爲有組織的犯罪並不關心從哪里竊取資訊，而且他們有能力採用自動化技術來保證較低的成本。

這又將我們引回了利用 Web 應用程式中現有漏洞的可能性。對照圖 1 顯示的幾條標準，如果您能根據您的組織的情況給每一條標準都打對號，那麼就應該考慮到，自動工具入侵您的應用程式以竊取資訊或利用您作爲惡意軟體傳播媒介的可能性是很高的。

圖 1 Web 應用程式中的漏洞被利用的可能性[3]

圖字：OPPORTUNITY：機會
COST：成本
EASY TO EXPLOIT：易於攻擊
EASY TO MONETIZE：容易牟利
MANY TARGETS：多個目標
HIGH VALUE：高價值

[3] *IBM X-Force 於 2009 年 1 月發佈。*

對組織的影響

現在，我們已經清晰地瞭解了犯罪組織選擇其他組織（包括您的組織在內）作爲攻擊目標的動機，我們觀察了成功的攻擊能給您的組織帶來的損害。遺憾的是，成功的攻擊給您的組織造成損失的方式是多樣化的。資料丟失、品牌受損和無意中助長犯罪，這是成功攻擊給您的組織帶來的三種最常見的影響。

資料丟失

成功的攻擊帶來的最直接、最明顯的影響就是資料丟失。攻擊者入侵您的系統，從您的機器中竊取並刪除資料，這將導致資料丟失。在被攻擊後使用您的機器時，您的企業無法再使用這些資料，因而您無法再按照之前的方式開展當前的業務。設想一下丟失所有客戶記錄的影響，想像一下這將給您的企業帶來多麼嚴重的損失。

品牌受損

一項更爲長期的損失就是攻擊者給您的品牌造成的損失。攻擊者可能會塗改破壞您的網站，竊取並公佈您的大量資料等。所有這些活動都可能受到廣泛的關注，並影響您的客戶。這會給您的品牌造成負面影響，使您難以保留現有客戶或吸引新客戶。

無意中助長犯罪

即便犯罪份子並未竊取任何資料或塗改破壞您的網站，他們仍然可以利用您的機器作爲宿主，傳播其惡意軟體。如果這種情況被發現並公開，就可能導致品牌受損，如前一節所述。然而，即便這種情況並未公開，您也在無意識的情況下幫助犯罪份子攻擊了您的客戶，這將給您的企業帶來嚴重的法律後果。（XSS,跨網站腳本執行就是基於這個原理來助長了攻擊和犯罪）。

趨勢和統計資料

遺憾的是，由於許多企業更傾向於保密此類資訊，因而難以獲得關於安全性違規的定量資料。保密的主要動機在於揭露違規細節可能招致的品牌損失。然而，公開報告的資訊中已經出現了一些令人擔憂的安全趨勢和統計資料，在後面的幾節中我們將加以討論。

增加對 Web 應用程式的關注

IBM Internet Security Systems X-Force 研究小組 [1] 發佈的研究表明，在過去五年中，已報告的 Web 應用程式漏洞在所有被報告的安全漏洞中所占的百分比大幅度提高。實際上，自 2006 年以來，所有被發現的漏洞中有 54% 的漏洞都處於商業和開源 Web 應用程式之中。當今的企業越來越多地透過 Web 廣泛實現業務職能，在這一充滿挑戰的市場中實現增長，因而這樣的比例並不令人感到意外。

正因如此，某些 Web 應用程式供應商被列入 2008 年漏洞披露中情況最嚴重的十大廠商，而在此之前，這一清單中列出的主要是規模較大的

非 Web 廠商。圖 2 展示了 1998 年至 2008 年間所發現的 Web 應用程式漏洞數量的爆炸式增長。

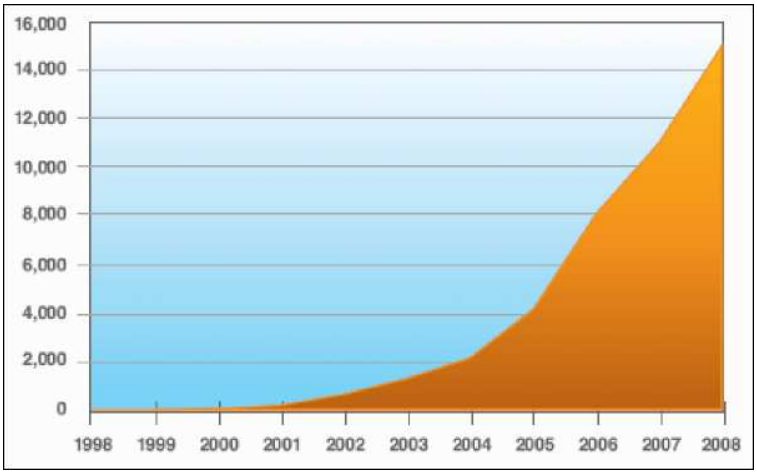


圖 2 Web 應用程式漏洞累計數量[4]

根據全美網路安全局和美國國土安全部發起的已知安全漏洞列表 Mitre Common Vulnerabilities Enumeration (CVE) [5]，公開報告的 Web 應用程式漏洞總數正在急劇增加。這個數位已經超越了軟體中最常見的安全漏洞緩衝區溢出。這樣的增加主要源于檢測和利用 Web 漏洞更加輕鬆，加上經驗不足的開發人員編寫的低級軟體應用程式的普及。這樣的增加也是由於創建簡單的小 Web 應用程式更輕鬆而導致的。儘管出現了這樣的簡化，但開發安全、高級的 Web 應用程式仍然是一個複雜的問題。

在 Web 應用程式安全性方面，有許多反復出現的漏洞。三大最重要的漏洞是 SQL 注入、XSS 和文件包含。圖 3 展示了 2004 年至 2008 年間這些漏洞的趨勢概覽。可以看出，三大漏洞在所有已發現的漏洞中所占的比例高達 70% 至 80%。

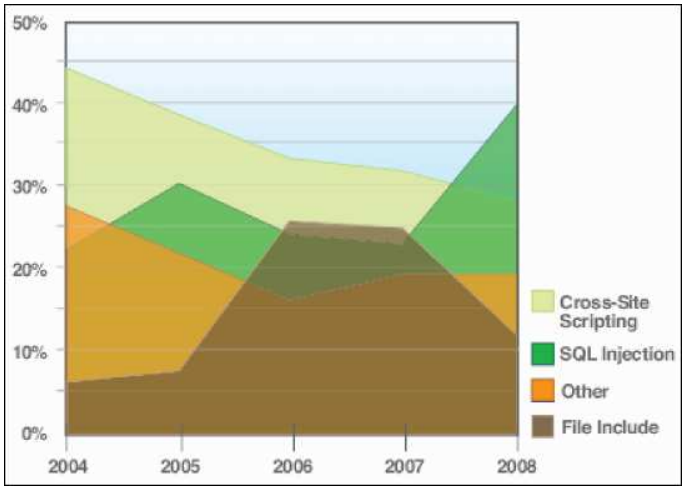


圖 3：
Cross-Site Scripting：跨站點腳本
SQL Injection：SQL 注入
Other：其他
File Include：檔包含

圖 3 按攻擊技術劃分的 Web 應用程式漏洞，2004 - 2008[5]

---

4 IBM X-Force 於 2009 年 1 月發佈。

5 Ibid。

在這份指南中，我們主要討論 SQL 注入和 XSS，因爲它們分別是 2008 年排名第一和第二的漏洞。關於檔包含的更多資訊，請參見 IBM Internet Security Systems X-Force 2008 Trend & Risk Report [1]。

攻擊的可能性

問題依然存在：在這些漏洞中，有多少比例的漏洞將在一次攻擊中被實際利用？惡意攻擊者同樣注意到了安全問題清單上 Web 應用程式漏洞的普遍存在，這是一個殘酷的事實。Gartner 預計，如今有 75% 的線上攻擊關注的是 Web 應用程式 [6]。這樣的統計資料與大多數安全性支出花費在底層網路而非應用程式上這一令人擔憂的事實相結合，潛在的問題顯而易見。Web 應用程式已經成爲攻擊的目標——無論是出於財政收入、政治動機還是個人尋找樂趣的目的。

自定義 Web 應用程式中存在漏洞的可能性

基於上述統計資料，可以明確，所發現和報告的安全性問題往往處於商業打包軟體或開源軟體中。然而，這對組織理解外包或內部開發應用程式中存在漏洞的可能性幫助不大。2007 年，Web Application Security Consortium (WASC) [7] 對 32,717 個應用程式開展的廣泛分析表明，其中 96.85% 的應用程式至少存在一個高危漏洞。58.11% 的應用程式中存在 XSS 漏洞，
25.30% 的應用程式中存在 SQL 注入漏洞。因而，應該認爲，您的自定義 Web 應用程式中存在漏洞的機率很高。

查明 Web 應用程式的薄弱環節

在這一屆中，我們將介紹關於 Web 應用程式薄弱環節的技術背景。我們概述 Web 應用程式可能受到攻擊的最常見方法，並提供關於兩種最常見的 Web 應用程式攻擊的深入見解。

Web 應用程式可能發生怎樣的問題？

Web 應用程式中存在一些最有可能發生問題的領域。主要的難點包括不安全的資料傳輸、伺服器端缺乏輸入驗證、用戶端缺乏控制等。

不安全的資料傳輸

在資料穿越網路時，很有可能被竊。在大多數 Web 應用程式中，在用戶端和伺服器之間傳輸的資料包是透過不受控、不可靠的網路傳輸的。例如，如果您打開流覽器，請求一個股票交易商的頁面，您的請求將拆分成多個資訊報，透過公共 Internet 傳輸，中間經過無數路由器和交換機。儘管有著可伸縮的優點，但用於 Internet 傳輸的模型會使資料易受中間人
(MITM) [8] 攻擊的侵擾。在此類攻擊中，攻擊者處於資料傳輸的起點和終點之間，監控或修改所傳遞的資料。例如，攻擊者可修改股市新聞的內容，希望影響新聞接收者可能會根據虛假內容做出的決策。

這樣的攻擊者類似于傳統領域中的郵遞員。即便您瞭解並信任每天爲您收發郵件的郵遞人員，但也不瞭解在您的郵件離開郵箱到最終抵達目的地這段時間中發生了什麼。例如，有人可能會使用蒸汽拆封您的郵件或利用光學成像技術偷窺其內容。同樣，其他人也可能會

參與投遞郵件的這個過程。無論是哪種情況，都必須運用安全技術，保護內容在傳輸過程中的完整性。

伺服器端缺乏輸入過濾和檢查

伺服器端應用程式安全性是一個龐大而複雜的主題，無法在本指南中完全詳述。按照定義，應用程式會接受用戶輸入，並根據該輸入執行操作。這樣的輸入可能是頁面請求、擊鍵操作或表單提交。上述各種輸入都將由伺服器處理。如果輸入未得到安全的處理，就可能無意中破壞或操縱應用程式。

我們可以擴展郵遞員的範例，以包含對郵件基礎設施的攻擊行爲。假設郵遞員未透過 X 光和爆炸物檢查合理過濾傳入的郵包。那麼這位郵遞員就有可能因遞送內有爆炸物的郵包而受到攻擊。可以設想，郵包的不合理過濾可能會對郵遞員產生破壞性的影響。

用戶端缺乏控制

儘管伺服器端一度是流行的攻擊位置，但越來越多的攻擊不再指向伺服器，而是針對用戶端。對於新興的 Web 2.0 應用程式來說，這種情況尤爲嚴重，因爲這些應用程式將更多的邏輯轉入用戶端，以提高應用程式的回應能力。您可能爲應用程式代碼和環境實施了有力的安全控制，但幾乎不可能控制用戶端獨立用戶或用戶端機器的行爲。因此，可能會出現許多問題。

同樣，我們要在用戶端攻擊方面深入探索郵遞員的例子。假設攻擊者已經確定了特定的收件者。他們就可透過製作專門的郵件來攻擊這位收件者，例如包含炭疽病毒的郵件會在郵件抵達目的地時立即攻擊收件者。

Web 應用程式漏洞

在下面的幾節內容中，我們將觀察並分析兩種最常用的 Web 應用程式漏洞，並爲您介紹能在哪里找到更加完整的漏洞清單。

SQL 注入

在 Web 上可以找到大量詳細討論 SQL 注入的文檔 [9-1 1]，所以本文不會涵蓋 SQL 注入的所有方面。但我們將透過範例來展示 SQL 注入的工作原理，解釋它將會對您的組織產生怎樣的影響。

考慮一個帶有自定義登錄表單的應用程式，其中包括用戶名和密碼。系統爲了對用戶進行身份驗證，向資料庫發送這樣一條查詢：“用戶 X 的密碼是否是 Y？”如果此查詢的結果爲真，則用戶 X 就透過了身份驗證。如果結果爲假，用戶將被拒絕。

現在，假設伺服器端並未執行合理的驗證，允許某些人創建以下查詢：“用戶 X 的密碼是否是 Y 或者我能否輸入 X 作爲用戶名？”[6] 此時不是簡單地將 Y 作爲密碼提供，在將此視爲一條查詢時，惡意用戶可確保查詢始終返回真。第 11 頁的範例 1 顯示了與此範例相關的 SQL 查詢。

[6] 爲了更清晰，我們將使用斜體顯示用戶輸入。

範例 1 登錄 Web 應用程式的 SQL 查詢

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

爲成功攻擊此查詢，攻擊者必須確保所提供的輸入能突破當前預定義的 SQL 查詢命令集（SELECT、FROM、WHERE、AND），並插入攻擊者自己的命令。只要攻擊者能夠突破預定義的結構，即可操縱查詢，返回其希望的結果。攻擊者將獲得與執行 SQL 命令的進程相同的許可權，隨後即可對您的資料庫執行各種操作。

現在，假設伺服器不會執行任何驗證，用戶名和密碼欄位的所有輸入都會直接重定向到 X 和 Y。那麼攻擊者即可使用單引號來突破預定義的命令並注入攻擊者自己的命令，進而確保此查詢總是返回真。

在 Y 中注入惡意輸入可能會導致類似於範例 2 所示的查詢，始終保證攻擊者透過身份驗證。

範例 2 破壞 SQL 查詢的惡意輸入

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

攻擊者能透過向 SQL 查詢注入惡意輸入而實現的操作實際上是無限的。基本上，攻擊者具有與執行查詢的進程相同的前往許可權，可以完成此進程能夠執行的任何操作。爲了攻擊您的組織，攻擊者會創建多種輸入值，試圖發現您的查詢中允許使用哪些字元。此後，攻擊者會嘗試瞭解資料庫結構，利用這樣的認識來獲得表名稱、用戶名和密碼。攻擊者甚至可能會破壞整個資料庫。如果您的企業沒有對攻擊者來說有價值的資料，那麼攻擊者可利用 SQL 注入攻擊，將惡意軟體注入您的頁面，利用您的網站作爲傳播平台，攻擊您的網站的前往者。自動化的強大力量和簡便性使此類攻擊成爲 2008 年最常見的一類攻擊方式。

遺憾的是，注入漏洞（injection flaw）不僅限於資料庫，還可能包括檔系統注入、郵件（MX）注入、可擴展標記語言（XML）注入、羽量級目錄前往協定（LDAP）注入和伺服器端包含（SSI）注入，而這些僅僅是其中的一部分。如果伺服器未能正確處理所有應用程式輸入，應用程式環境就可能被入侵，業務邏輯被破壞、受保護的資料被公開。

跨網站腳本

XSS 攻擊十分普遍，Web 上可以找到關於此類攻擊的大量資料 [12, 13]。我們採用與介紹 SQL 注入相同的方法，透過一個範例來解釋攻擊者會如何利用此類漏洞。

設想一個支持論壇。如果此論壇的輸入未得到合理過濾，惡意用戶即可在論壇上張貼任意 HTML。其他用戶前往論壇時，將爲其顯示這些 HTML 代碼。如果 HTML 代碼中包含腳本，每當有一名用戶前往論壇上的這個帖子時，就會執行此腳本。第 12 頁的範例 3 展示了攻擊者如何插入一條彈出消息，在他人前往此帖時顯示彈出消息。

範例 3 論壇上張貼的未經過濾的消息，爲每位元閱讀者顯示一條警報

```
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>
```

初看起來，Web 網站中的 XSS 漏洞似乎並不會給您的用戶造成嚴重影響。但若 XSS 攻擊發揮到最嚴重的程度，就會與 SQL 注入同樣危險，甚至比 SQL 注入更加危險。根據攻擊者的熟練程度不同，攻擊者透過在您的頁面中注入代碼而獲得的對用戶的控制級別也有所不同：

1. 最基本的攻擊在您的 Web 頁面中注入資料片段，以實施塗改破壞。此類攻擊的典型範例就是注入“你已經被入侵”之類的消息，插入圖片等。
2. 在下一個級別中，攻擊者可能會嘗試透過提供錯誤或誤導性的資訊來影響用戶。此類範例包括提供股票走勢的假資訊、僞造線上年度報表等。
3. 第三級的攻擊者會嘗試干擾網站的正常使用。攻擊者可能會利用 XSS 漏洞來注入代碼，使網站以不正常的方式運作甚至完全不可用。此類範例包括生成數以千計的彈出消息、重新定義鏈結的指向位置等。
4. 更熟練的攻擊者可能會利用 XSS 來竊取身份。透過注入在用戶登錄時在頁面上運行的代碼，攻擊者即可透過鍵盤記錄器來截取密碼，竊取用戶的身份。
5. 第 5 級的攻擊者會在用戶使用您的網站時監控這些用戶。高級 XSS 攻擊允許攻擊者監控用戶執行的每一項操作。攻擊者能查看用戶流覽的每一個頁面，獲取用戶輸入的所有資訊（包括其用戶名、密碼、信用卡資訊、位址等）。
6. 最終，攻擊者可全面掌控用戶的環境。透過控制用戶，攻擊者即可自行制定決策、流覽您的網站的各個部分、購買商品等。

其他漏洞

目前 Web 應用程式中存在的漏洞的完整清單過長，無法在本指南中一一詳述。如需瞭解更多資訊，請參見開放 Web 應用程式安全項目（Open Web Application Security Project，
OWASP）[14] 和 Web 應用程式安全聯盟（Web Application Security Consortium，WASC）[15] 的網頁。

保護您的 Web 應用程式免受攻擊

我們必須以一種方式來研究 Web 應用程式的各個方面，進而找出最多漏洞；我們還必須採用類似的方式來檢查軟體開發生命週期的各個方面，進而將我們的 Web 應用程式構建得盡可能安全。

在本節的第一部分，我們概述了軟體開發生命週期的不同難點，介紹了測試軟體安全性的不同方式，並著重指出了兩個最重要的最佳實踐。本節的第二部分首先概括介紹 IBM Rational 工具，這些工具能協助您保證軟體開發生命週期的安全性。然後我們探究 IBM Rational AppScan 產品線的詳細資訊，介紹它們的不同版本，並說明如何將其整合到您的軟體開發生命週期中。借此，我們向您展示如何構建一個開發 Web 應用程式的安全性生態系統。

保證軟體開發生命週期的安全性

軟體開發生命週期包括三個階段：

- 設計階段
- 開發階段
- 交付階段

每個階段都會行銷您的最終產品的整體安全品質，因此，必須從安全性角度考慮它們。

設計階段

軟體開發生命週期中的設計階段包含需求的創建和應用程式架構的設計。爲保證軟體開發生命週期的安全，在執行需求和架構的設計時都必須緊記安全性。如果需求和架構未被明確地設計、規劃和執行，那麼幾乎所有應用程式都可能存在巨大的缺陷。

在水肺潛水人群中，流行這樣一句重要的格言：“爲潛水做計畫，按計劃來潛水。”失敗的計畫可能導致嚴重的後果。人的生命不是常常都會面臨危險，而這一準則對 Web 應用程式而言同樣有效。如果所設計的軟體計畫不健壯和不安全，那麼可能發生最具災難性的軟體故障。擁有適當的需求集合以及滿足這些需求的設計，就能夠創建這樣的計畫。要爲此流程提供幫助，可以使用多種 IBM Rational 工具。

考慮這樣一個與身份驗證相關的普通情節。美國的研究表明 9 個人中就有一個人使用 500 個最常用的密碼中的一個，每 50 個人中就有一個使用最常用的 20 個密碼中的一個。這是一個嚴重的安全問題，因爲駭客很容易暴力破解這 500 個最常用的密碼。

解決這一問題的常用辦法是封鎖那些在短期內過多地進行嘗試而登陸失敗的帳戶。但是，駭客可能使用不同用戶名來嘗試那些最常用的密碼，以避免被封鎖。因爲您不希望封鎖掉所有帳戶，所以您對此無計可施。您也不希望禁止發動攻擊的 IP 位址的前往，因爲擔心這會遮罩來自相同閘道的合法用戶。

因此，安全地設計應用程式至關重要。制定一個需求，要求您的系統不接受容易被猜到的密碼，這可能足以預防此問題。當然，需求必須被正確地實現，這就讓我們進入到軟體開發生命週期的下一個階段，也就是開發階段。

開發階段

開發階段是一個三步流程，在此階段中代碼被編寫、構建和測試。雖然許多軟體開發小組認識到需要安全地開發應用程式，但是經驗證實開發安全應用程式的難度不小。實際上，大多數被報告的漏洞都是開發實踐不佳的結果。惡劣的開發實踐的一個典型範例是，經驗不足的開發人員編寫了一個自定義元件，同時將安全漏洞引入到了應用程式中。較好的實踐是，使用成熟框架中已經經過全面安全漏洞測試的現有的可靠元件。此外，就安全開發實踐對開發人員進行培訓，這在未來將會有所回報。

隨著富 Web 2.0 UI 設計的快速變化，對安全代碼開發的需要也變得尤爲關鍵。不斷變化的 Web 2.0 設計幾乎沒留下進行全面測試的空間。在此之外，爲了最大化交互性，更多應用程式代碼在用戶端流覽器上運行，以便用戶能夠輕鬆查看。組織必須假定，用戶將故意篡改公開的應用程式業務邏輯，
並試圖發掘其自身優勢。[7] 透過將正確的工具整合到開發流程中，您 Web 應用程式的編碼、構建和測試過程中與安全性相關的許多工將實現自動化。

交付階段

最安全地設計和開發的軟體如果被交付到不安全的環境中，也會不再安全。這涉及到（但

不限於）應用程式基礎架構的固化、資料透過網路時的保護、生產環境的防禦以及配套作業系統和元件的修補和更新戰略。

例如，沒有將 Web 伺服器配置爲拒絕前往目錄結構，這會允許惡意用戶直接前往敏感資訊和應用程式代碼。因此，需要有安全的交付階段，在交付環境中最終審核應用程式安全性，然後維持操作環境的安全級別。而且，廣泛的工具可用於使這些任務自動化。

分析的細微差別

在軟體開發生命週期中用於分析 Web 應用程式安全性的工具大致分爲三種不同的類型：

- 白箱分析工具
- 黑箱分析工具
- 灰箱分析工具

這些分析工具根據有關系統和軟體的信息量來區分，這種信息量在進行安全性分析時被這些工具使用。從開發人員的角度分析，白箱分析對信息的前往級別最高，並可以視作逼近安全性。黑箱分析從攻擊者的角度出發進行分析，無需前往資訊。灰箱分析結合了黑箱和白箱分析的特點，以提高精確性和覆蓋範圍。

7 駭客根據可視的用戶端邏輯逆向工程業務邏輯的一個很好的例子就是 2007 攻擊，它使駭客獲得通行 Macworld 2007 的免費 VIP 折扣代碼 [16]。

白箱分析

使用白箱分析，關於系統或軟體的所有相關資訊是已知的，並可被測試人員使用。該分析方法在品質保證領域（在該領域中，負責軟體的人員既能前往設計文檔，又能前往原始碼）中得到普遍應用。除開源軟體外，這種資訊通常不會對限定人群以外的個人公開。

給定相關資訊，白箱分析比較全面。這種方法可以快速揭示外行人看不到、潛在的、模糊的相互關係。 白箱分析可以快速確定整個攻擊面並創建一套必要的測試。

在白箱分析中，重要的是不要過於強調設計規範或者原始碼。過於強調設計規範可能導致測試人員失去在該文檔之外構建的或者尚未正確實現的功能。過於強調原始碼可能導致測試人員遺漏與架構相關的關鍵漏洞或系統設計的方式。

白箱分析包括以下三種主要的技術：

- 架構分析，通常稱爲威脅建模，試圖枚舉在軟體內攻擊者所攻擊的目標並提出針對每種威脅的對策
- 原始碼分析， 掃描應用程式的原始碼並跟蹤用戶輸入，以查找漏洞和錯誤編碼實踐
- 靜態二進位分析，與原始碼分析的操作類似，但只在二進位級別上分析，允許查找上下文風險和特定於平台的問題。

白箱分析提供了如下優點：

- 測試人員可利用所有的相關資訊。
- 可以確定邏輯分析的缺陷。
- 能夠輕鬆、快速地歸檔整個攻擊面。
- 能有效查找編程和實現錯誤。

白箱分析面臨如下挑戰：

- 不良的編碼實踐可能作爲漏洞被錯誤地檢測到。
- 軟體不容易在分散式環境中被遠端測試。
- 不可能始終前往設計規範和原始碼。

白箱分析的一個例子就是在開發期間運行原始碼掃描器。

黑箱分析

黑箱分析涉及在事先不瞭解環境的情況下檢查軟體或系統。這種分析與外部攻擊者可能從事的分析類似。使用自動化工具和手工技術，這種分析首先檢測攻擊面和探查相關資訊的系統。

黑箱分析的基本概念是在針對安全性問題進行實際測試前盡可能全面地瞭解系統。資訊披露和配置不正確的系統在建立這種認知的過程中尤其有用。在測試階段中，對軟體或系統的這種基本瞭解使測試人員的工作效率更高。

當一個機構要分析外部人員的威脅時，黑箱分析經常是首選的分析方法。從這種分析和測試中得出的風險結果經常要區分優先次序，因爲它能更準確地反應外部人員所構成的直接風險。

黑箱分析包括以下兩項基本技術：

- 漏洞掃描，透過利用已知漏洞的大型資料庫以及嘗試識別應用程式中的已知漏洞來實現。 這既可以是被動的，例如透過搜索 Web 頁面中的版本號來進行漏洞掃描，也可以是主動的，例如透過嘗試利用一種漏洞以及搜索已知的利用結果來進行漏洞掃描。

- 動態分析，通常稱爲 Web 應用程式掃描。這種分析試圖自動掃描並記錄攻擊面，借助故障注入的手段來測試應用程式以及根據回應來確定漏洞的存在。這種分析和漏洞掃描之間的主要區別在於動態分析可以發現未知的漏洞，而漏洞掃描只針對已知的漏洞進行測試。

黑箱分析包括如下優點：

- 測試已部署的軟體會生成高度可信的結果。
- 無須前往設計規範或原始碼。
- 可以跨網路輕鬆測試軟體。
- 在已部署的環境中測試支援環境分析。

黑箱分析面臨以下挑戰：

- 不可能確定代碼覆蓋範圍；不明顯或隱蔽的功能可能被遺漏。
- 邏輯設計缺陷不易被檢測到。

黑箱分析的一個例子就是在交付階段中針對所部署的應用程式運行 Web 應用程式掃描器。

灰箱分析

白箱和黑箱分析均可以揭示可能的軟體風險和潛在的漏洞。白箱分析保證了代碼覆蓋範圍，但存在難以估量的風險。黑箱分析能保證識別真實問題，但其代價是代碼覆蓋範圍未知。灰箱技術以一種強大的方式將白箱和黑箱分析方法結合起來。可以獲得這兩種方法的優勢，同時最小化遺漏重要問題的可能性。

部署灰箱分析的挑戰是，這一過程通常需要使軟體開發生命週期的不同階段中所得到的結果集相互關聯起來。在沒有企業文化轉型的情況下可能難以採取成功的灰箱分析策略。當灰箱分析成爲軟體開發生命週期的一個重要部分，允許在整個生命週期內平穩地整合所獲得的不同測試結果時，最有可能取得成功，

> 注意：使用這些術語的一種更常用的方式是區分原始碼前往（白箱）和非原始碼前往（黑箱）。儘管從嚴格意義上講這種命名並不準確，但它通常是流行的命名法。

在開發過程中使用安全性分析技術

當考慮更廣泛的軟體開發生命週期時，我們就會明白每種安全性分析技術在軟體開發生命週期中具有各自的功能。要獲得完整的覆蓋範圍，您必須將白箱和黑箱測試結合到一項安全性測試解決方案中，作爲軟體開發生命週期的一個組成部分。

圖 4 顯示了在整個設計、開發和交付階段中如何使用並組合架構分析、原始碼分析、動態分析、二進位分析和漏洞掃描。 安全性測試不是只能在軟體開發生命週期結束到產品發佈這段時間內才能做的事情，認識到這一點很重要。儘管在交付階段進行全面的安全性稽核適當保護軟體的一個重要方面，但必須將安全性整合到開發生命週期的每一步中。

架構安全性分析應該作爲設計階段的一個組成部分。應該將原始碼分析整合到開發階段中。動態分析在開發階段開始進行並一直持續到交付階段。二進位分析通常在交付階段的稽核期間針對軟體的每次新建進行一次。最後，漏洞掃描是一項重複性任務，只要軟體運行，就應該定期進行。

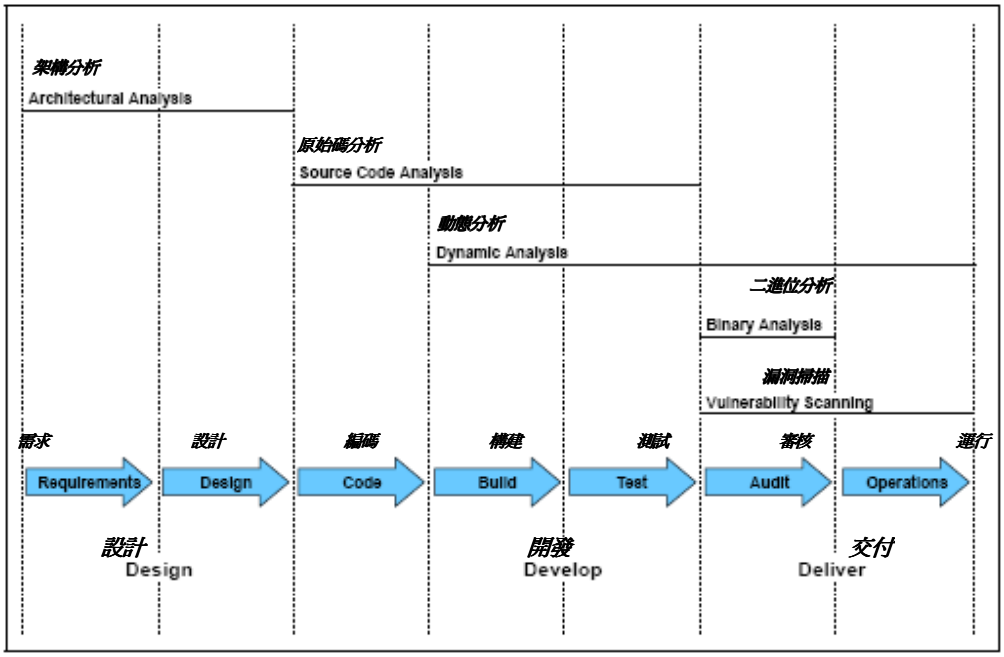


圖 4 在整個軟體開發生命週期內安全性分析的不同類型

兩種重要的最佳實踐

下面讓我們更深入地瞭解兩種重要的最佳實踐：

- 定義明確的安全需求
- 最大化地實現對安全需求的自動化測試

定義明確的安全需求

要定義明確的安全需求，重要的是確切瞭解什麼是安全的需求。儘管安全需求與安全特性用於根本不同的目的，但它們經常被混淆。使用以下兩個定義可以明確這種差異：

- *安全需求是必須完成的任務。* 例如，所有密碼必須加密存儲。安全需求通常不會指示如何做，例如指示用哪個軟體用來加密。

► *安全特性是必須可用的功能*。 例如，用戶管理必須能夠在 Web 介面進行。

與功能要求或性能要求類似，安全需求需要確保從一開始就將安全性構建到應用程式中。安全需求定義要求何種新安全特性和如何改變現有特性以包括必要的安全屬性。設定安全需求的目標是確保應用程式可以預防和抵擋攻擊。當構建 Web 應用程式時，您必須考慮九類安全需求，下列章節將一一解釋。

稽核和日誌記錄

儘管人們通常依賴網路和包日誌來進行取證分析，同時，應用程式內部的日誌記錄也是同等重要的；應用程式應該對軟體的保密性、可用性和完整性至關重要的事件進行內部日誌記錄。

例如，應用程式就需要有稽核日誌。稽核日誌來記錄日誌的事件必須包括當前會話權杖（如果有）、IP 地址和時間。必須記錄日誌的事件還包括帳戶驗證嘗試、帳戶鎖定、應用程式錯誤和與規定的驗證常式不匹配的輸入值。

身份驗證

由於大多數應用程式具備前往控制限制，以防止機密性洩漏，因此，確保這些前往控制機制不能被破解或操作以允許未經授權的前往，這一點非常重要。

例如，要求強密碼。任何身份驗證憑證必須包含適當的強度，其中包括大寫字母、小寫字母和數位字元，而且在長度上不能小於 8 個字元。

會話(session)管理

HTTP 協議最初的設計使得難以在整個應用程式會話持續期間跟蹤會話。這推動了 HTTP 協議之上的會話管理功能的構建。

例如，一個安全需求是合法用戶自始至終可以保持正常前往；遠端會話的所有資源利用必須加以監控和限制，以防止或減輕對應用程式可用性的攻擊。

輸入驗證和輸出編碼

儘管在建模和架構階段大多數設計級安全性缺陷都被發現，但大多數開發和交付安全性問題是因爲不良的輸入驗證和輸出編碼而引入的。重要的是用戶提供的資料要透過適當的驗證。

例如，所有輸入必須透過集中的驗證控制來加以驗證。

異常處理

從嚴格意義上講，一個應用程式不可能完全安全。 但是，常常最可接受的是應用程式“足夠安全”。隱藏詳細的應用程式異常或過於具體的錯誤消息能夠延長攻擊應用程式所需的時間。

舉例來說，此上下文中的一個安全需求是應將所有錯誤消息捕獲並記錄在安全性稽核日誌中。

加密技術

安全的加密演算法極難創建和實現。一個機構選擇一種滿足業務需要、受行業支援的演算法極其重要。

安全需求的一個例子是，應用程式內所使用的所有加密演算法必須經過聯邦資訊處理標準（Federal Information Processing Standards ，FIPS）[17]批准且與之相容。

靜止資料

儘管所有應用程式都試圖保護後端存儲庫中的資料，但是最好假設該資料存儲將來在某種意義下會被洩漏。深度防護規定任何敏感資料都要針對這種可能性進行加密。

安全需求的一個例子是，如果應用程式包含必須爲了法規遵從性而加以保護的敏感用戶資訊，則必須使用功能強大的加密技術來保護姓名（名稱）、用戶名、位址和財務資料等敏感的用戶資訊。

活動資料

只要軟體應用程式被實現，就存在對要求跨網路或系統傳輸資料的軟體架構的已知攻擊。當應用程式資料跨越開放和封閉的網路和系統時，對其妥善保護是至關重要的。

對這類系統的安全需求可能包括：如果應用程式在不可信或不安全的網路間傳輸敏感的用戶資訊，那麼所有通信管道必須予以加密。

配置管理

新的漏洞每天都會湧現，這凸顯了普通基礎設施元件中的弱點。儘管其中一些問題可透過打補丁的方式加以糾正，但有時要求這些選項能夠用於進行特定部署的用戶。

透過既滿足應用程式業務需求又保護了應用程式和基礎設施的方式來平衡底層系統，這至關重要。

安全需求的一個例子是，所有管理介面必須從非管理介面分離出來。

最大化對這些需求的自動測試

在定義了安全需求的明確集合後，具備可靠的流程以檢查這些需求是否在整個開發生命週期內得到正確使用非常重要。如果您回想攻擊者的動機以及他們操作的方式，就會想到他們試圖透過自動化攻擊您的 Web 應用程式來削減成本。

與攻擊者自動化其任務的方式相同，您也可以自動測試安全需求並驗證其實現的正確性。事實上，如果您確信以自動化方式輕鬆找到的這些漏洞再也不會出現在您的 Web 應用程式中，那麼您就已經消除了高比例的可能攻擊。

因此，確保您使用了自動測試來保護 Web 應用程式是一種不錯的實踐。由於自動測試不能涵蓋所有漏洞，所以如有必要，這種測試可以根據您的業務需求透過防禦攻擊和手動代碼修正來擴大。在下一節中，我們向您展示如何使用 IBM Rational AppScan 產品在開發生命週期內自動化安全性測試。

IBM Rational AppScan 軟體套件

IBM Rational AppScan 是一個 Web 應用程式安全性測試產品套件，用於自動化應用程式掃描和漏洞識別。Rational AppScan 產品針對大量 Web 應用程式漏洞進行掃描和測試，其中包括 Web 應用程式安全協會（Web Application Security Consortium，WASC）威脅分類和開放 Web 應用程式安全項目（Open Web Application. Security Project，OWASP）所確定的漏洞。Rational AppScan 產品線包含大量產品，每種產品適用於特定用戶的要求。

在本節的剩餘部分中，我們按照軟體開發生命週期中產品出現的順序討論主要的版本。要瞭解整套 Rational AppScan 產品的相關資訊，請參見下列位址：

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

Rational AppScan Source Edition

Rational AppScan 產品線的首要目標用戶是開發人員。預防應用程式安全漏洞的最有效方式是從頭開始安全地構建軟體。其挑戰在於，大多數開發人員並非安全專家，且編寫安全代碼並非總是他們的頭等大事。因此，在應用程式安全性流程中從事開發的最佳方式是向其提供在其開發環境中工作且以他們理解的語言生成結果的工具。

IBM Rational AppScan Source Edition 的設計理念在於使開發人員能夠從其開發環境內進行應用程式安全性測試。它能解決可能在代碼中存在的大量安全性問題、流線化開發生命週期工作流，並幫助減少在發佈週期結束時可能出現的安全性測試瓶頸。Rational AppScan Source Edition 使用大量的分析技術來準確確認應用程式中的安全性問題，其中包括靜態代碼分析、運行時分析和字串分析。

當進行安全性測試時，開發人員的要求與安全稽核人員的要求有很大不同。Rational AppScan Source Edition 旨在用作一種開發工具。因此，其焦點集中在易用性以及易於整合到開發流程中。最大限度減少錯誤肯定並提供易於理解的結果的特性具有比增加掃描範圍（可能使安全性測試複雜化）的特性更高的優先順序。配置和結果的協作和共用是該產品的核心部分，掃描配置的重用能夠幫助在每個應用程式上提供一致、可重複的掃描。

開發人員版本在設計時充分採用了自動化理念。爲實現更出色的易用性和精確性而進行的自動代碼分析配置是透過使用字串分析來實現的，字串分析是在與 IBM Research 協作的過程中發明的新技術。 透過幫助解決困擾當前安全代碼掃描解決方案的最大挑戰，也就是錯誤肯定，字串分析是靜態代碼分析領域中的一大突破。迄今，分析代碼安全性的最先進技術——污點分析能夠在代碼流入系統時跟蹤輸入值。但是，它依靠開發人員來確定資料是否透過標記清理功能進行了適當清理。

因此，開發人員必須知道何謂十分簡單的清理，以及必須能對分析工具進行詳細配置，使其精確。這些內在的限制導致了大量錯誤肯定現象，經常需要修改代碼來支援掃描工具，並且需要安全專家參與。字串分析自動做出這些決定，幫助消除錯誤肯定並支援不同的輸入處理方法。儘管污點分析能夠衡量輸入是否被污染，但字串分析可以準確確定輸入如何被被污染的，進而將靜態代碼分析提升到一個全新的精確性水準。

除了字串分析的自助服務優點外，Rational AppScan Source Edition 還提供了內建的培訓，準確和區分優先次序的結果（直接指向有疑問的代碼行），以及帶有代碼樣例的詳細補救建議。這些直觀、易用的特性使開發人員能夠在 Web 應用程式安全性測試的日常處理中充滿自信。

圖 5 顯示了一份 Rational AppScan Source Edition 掃描報告，說明了它如何將白箱問題與受影響的代碼行聯繫起來，進而快速定位和輕鬆緩解問題。Rational AppScan Source Edition 還具備與其他產品整合的各種能力，詳見下一節。

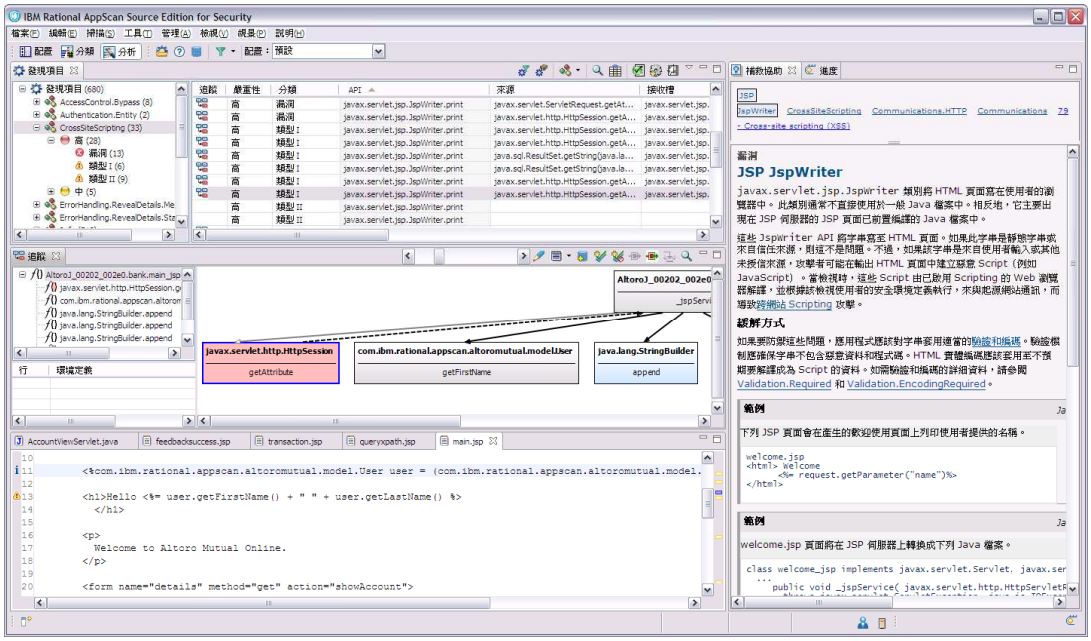


圖 5 Rational AppScan Source Edition 中顯示了白箱問題和受影響的代碼行的完整掃描報告

要瞭解有關 Rational AppScan Source Edition 的更多資訊，請參見下列網址：

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

Rational AppScan Build Edition

使用 Rational AppScan Source Edition 漏洞掃描引擎的漏洞檢測功能的另一種方式是在構建期間執行自動掃描。透過使用 IBM Rational AppScan Build Edition，可以將安全性整合到構建階段。透過與 IBM Rational Build Forge® 軟體等多個構建管理系統整合，Rational AppScan Build Edition 提供了針對計畫構建的安全性測試覆蓋範圍。它包括與 Rational AppScan Source Edition 相同的分析技術集，提供了高水準的精確性和代碼覆蓋範圍，能夠幫助確定哪些代碼經過測試。

掃描後，Rational AppScan Build Edition 將結果透過 IBM Rational ClearQuest® 軟體等缺陷跟蹤解決方案或透過 Rational AppScan Enterprise Edition 或 Rational AppScan Reporting Console 等安全報告解決方案發送給開發人員。Rational AppScan Build Edition 還包括一個應用程式編程介面（API）和各種其他結果格式，支援將掃描結果傳播到其他存儲庫。

要瞭解有關 Rational AppScan Build Edition 的更多相關資訊，請參見下列網址：http://www.ibm.com/software/awdtools/appscan/

Rational AppScan Standard Edition

IBM Rational AppScan 產品考慮的下一類用戶是安全稽核人員。爲了幫助這類用戶，我們發佈了 IBM Rational AppScan Standard Edition。爲了使安全稽核人員能夠自動化對最新技術的測試，Rational AppScan Standard Edition 支援最新的 Web 2.0 技術；JavaScriptTM 和 Adobe® Flash 應用程式的解析和執行；非同步 JavaScript XML （AJAX）和與 Adobe Flex 相關的協議 ，如 JavaScript Object Notation （JSON）、Action Message Format （AMF）和 SOAP；精細的面向服務架構（SOA）環境；以及針對 mashup 和流程驅動應用程式的自定義配置和的報告功能。

透過自動化許多重複性任務，Rational AppScan Standard Edition 降低了與手動漏洞測試相關的成本。無論是外包您的漏洞測試工作，還是在組織內部手工執行漏洞測試，Rational AppScan Express 都可以顯著減少對應用程式執行全面的漏洞評估所需的時間。這使您能不斷對 Web 安全狀態進行評估，而不是每季度或每年稽核一次，進而獲得更高的安全性水準並實現可控的成本。

Rational AppScan Standard Edition 掃描引擎爲您提供了高水準的掃描精確性並顯著限制了錯誤肯定。爲了進一步提高精確性和性能，該引擎包括了智慧類比人類邏輯的自適應測試流程，以適應針對個別應用程式的測試階段。Rational AppScan Standard Edition 瞭解應用程式的資訊，一直深入到每個特定參數，並進行調節，以便只執行相關的測試。爲了幫助確保免受最新威脅，Rational AppScan Standard Edition 在軟體每次啓動時都會檢查來自 IBM 安全研究專家團隊的攻擊簽名更新。

Web 漏洞掃描的最重要的方面之一是問題的快速修補。Rational AppScan Standard Edition 提供了每次掃描發現的漏洞的完整列表，這些漏洞按照優先次序排列，使高優先順序的問題首先得到修復，幫助機構從安全性角度將精力集中在最要緊的問題上。每個漏洞結果包括漏洞如何工作和潛在原因的詳盡描述。綜合的、基於 Web 的培訓提供了直接來自用戶介面的短期培訓模組。軟體的修補視圖然後解釋修補該問題所要求的步驟，其中包括安全和非安全代碼的樣例。

圖 6 顯示了 Rational AppScan Standard Edition 應用程式的一個視窗。

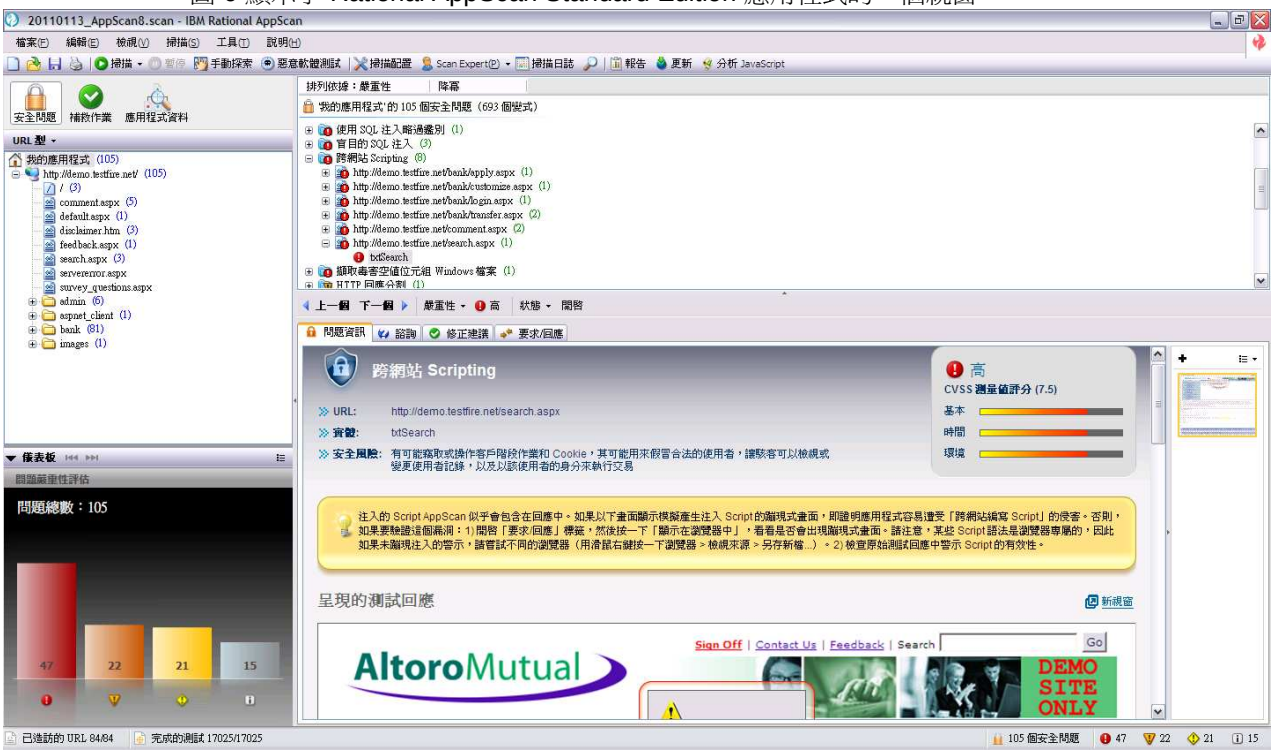


圖 6 Rational AppScan Standard Edition 幫助用戶快速識別、理解、區分優先次序和修復重要的 Web 漏洞

Rational AppScan Standard Edition 還可以透過提供一種方式來支援現行應用程式安全性級別來幫助機構解決支付卡行業資料安全標準（Payment Card Industry Data Security Standard，PCI DSS）等重要的遵從性要求。IBM 是一個授權的掃描服務商（Approved Scanning Vendor，
ASV），Rational AppScan Standard Edition 產品提供了這項資格，使該軟體成爲解決圍繞 PCI DSS 提出的應用程式安全需求的完美選擇。

Rational AppScan Standard Edition 可以生成定制的安全報告，並包括將哪些資料點選入每項報告的能力。用戶還可以從 40 多種預定義的報告和地圖掃描結果以及關鍵的行業和法規遵從性標準中進行選擇。這些選項包括國家安全技術局專刊（National Institute of Standards and Technology Special Publication，NIST SP）800-5、10 大開放 Web 應用程式安全項目（Open
Web Application Security Project ,OWASP）、PCI DSS、薩班斯-奧克斯利法案（Sarbanes-Oxley）、金融服務現代化法案（Gramm-Leach-Bliley Act，GLBA）、健康保險責任法案（Health Insurance Portability and Accountability Act， HIPAA)）、家庭教育權利和隱私法（Family Educational Rights and Privacy Act，FERPA）、自由資訊和隱私保護令（Freedom of Information and Protection of Privacy Act，FIPPA）和支付應用程式最佳實踐（Payment Application Best Practices，PABP）。

爲了定制和擴展測試以進行更強的控制，Rational AppScan Standard Edition 包括了一組功能強大的定制功能。IBM Rational AppScan 軟體開發工具箱（SDK）提供了一套功能強大的介面，支援對 Rational AppScan Standard Edition 中的每項操作（從長期掃描的執行到單獨定制測試的提交）進行可定制調用。

該平台能夠輕鬆整合到現有系統上，支援 Rational AppScan 引擎的高級定制使用，並爲 Rational AppScan eXtensions Framework 和 Pyscan 提供了基礎。

透過自動化 Web 應用程式測試流程來幫助安全稽核人員和滲透測試人員快速、有效地從事他們的工作，Rational AppScan Standard Edition 這種可用作一個桌面應用程式或用作一項軟體即服務（Software as a Service，SaaS）的軟體有助於顯著提高攻擊者的攻擊成本，進而使您的機構不再是有價值的目標。

要瞭解有關 Rational AppScan Standard Edition 的更多資訊，請參見下列網址：

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

Rational AppScan Tester Edition

讓我們從交付流程倒退一步，進入開發流程並查看一下我們如何能夠將 Rational AppScan 產品整合到測試階段。透過使用與 Rational AppScan Standard Edition 相同的漏洞檢測功能，IBM Rational AppScan Tester Edition 可用作桌面應用程式，它提供了各項功能來幫助品質保證（QA）團隊將安全性測試整合到現有的品質管制流程中，進而減輕安全專業人員的負擔。由於 Rational AppScan Tester Edition 與領先的測試系統整合在一起，所以 QA 專業人員可以在測試腳本中使用其功能。他們可以在熟悉的測試環境中進行安全性檢查，方便了安全性測試與功能和性能測試的同時採用。

假定 QA 機構已經知道如何區分缺陷的優先順序、使測試可重複並報告測試覆蓋範圍以及部署準備程度，那麼這些團隊完全適合進行安全性測試。他們能在應用程式交付流程中幫助更早地調整安全性測試，查找和修補安全漏洞。在針對功能和性能問題進行測試和在發佈中防止高成本的延遲時，他們可以這樣做。
要瞭解有關 Rational AppScan Tester Edition 的更多資訊，請參見下列網址： http://www.ibm.com/software/awdtools/appscan/tester/

Rational AppScan Enterprise Edition

除了先進的應用程式掃描功能以外，Rational AppScan Enterprise Edition 還提供了完善的報告和修補功能，以及與 AppScan 桌面版 Rational AppScan Standard Edition 的無縫整合。IBM
Rational AppScan Enterprise Edition 是一個基於 Web 的多用戶 Web 應用程式漏洞測試和報告解決方案，適用於需要在整個機構中執行應用程式掃描，同時又要保持對漏洞資料的集中控制的機構。Rational AppScan Enterprise Edition 包括 QuickScan、一個“傻瓜式”測試工具和基於電腦的綜合培訓，以促進在整個軟體開發生命週期中採用安全性測試。

透過生成實用的安全指標、儀表板和重要法規遵從性報告，Rational AppScan Enterprise Edition 有助於用戶瞭解整體安全狀態。

Rational AppScan 的掃描引擎檢測一個 Web 應用程式，分析和測試該應用程式的安全性和遵從性問題，並生成包含修復建議的可操作報告，以簡化修補流程。這些高級修復建議爲開發人員和安全稽核人員提供了無與倫比的精確性和效率，能夠幫助解決和修補掃描發現的漏洞。與領先的 QA 測試工具（包括 IBM Rational ClearQuest）、開發環境和代碼掃描設備的無縫整合進一步簡化了由 QA 和開發團隊進行的安全性測試和修補。

使用基於 Web 的架構，Rational AppScan Enterprise Edition 的設計旨在幫助各個機構在多個利益相關者之間分配安全性測試的責任。Rational AppScan Enterprise Edition 用於需要以集中的方式進行 Web 應用程式安全評估並提供充分整合的解決方案集的團隊。圖 7 顯示了 Rational AppScan Enterprise Edition 儀表板。

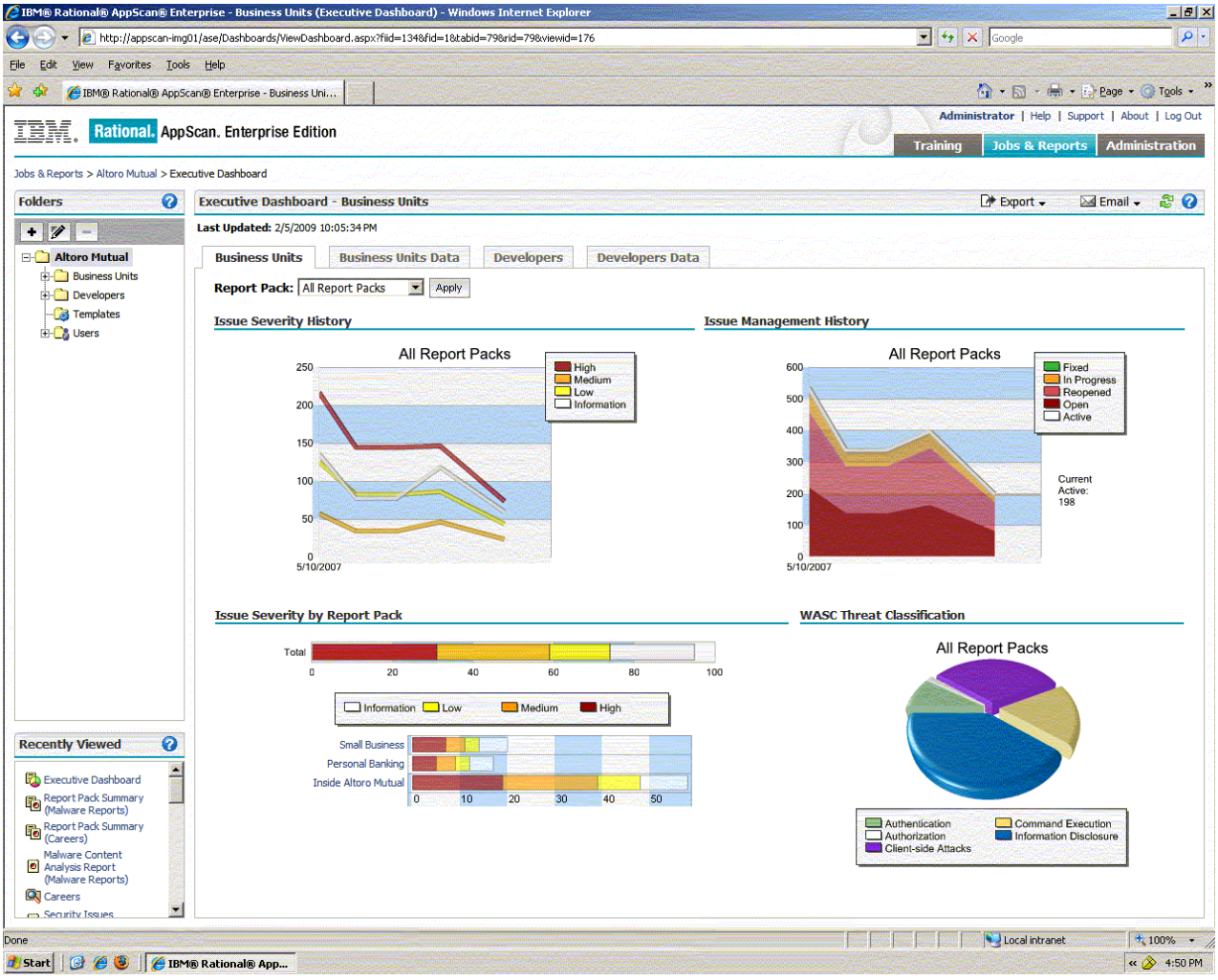


圖 7 IBM Rational AppScan Enterprise Edition 儀表板視圖

要瞭解有關 Rational AppScan Enterprise Edition 的更多資訊，請參見下列網址：http://www.ibm.com/software/awdtools/appscan/enterprise/

IBM Rational 技術與全面、綜合的安全平台結合

從上一節我們可以看出，Rational AppScan 產品線本身並沒有覆蓋整個軟體開發生命週期。要創建全面、綜合的安全開發平台，Rational AppScan 產品要與其他產品整合。例如，到現在爲止，我們既沒有包含各項要求和設計階段，又沒有過多地探討各種跟蹤和報告 AppScan 產品所發現的漏洞的產品。

在本節中，我們將說明其他 IBM 產品如何與 Rational AppScan 產品套件整合來創建完整的安全開發生命週期。雖然我們介紹了 IBM Rational 軟體產品，但競爭性技術也同樣能幫助達到相同的最終目標。因爲並非所有創建的技術都是等同的，所以我們將各項技術分組爲四個重要性層次。

安全軟體開發生命週期所需的第一個技術層次等級應該不會令大家吃驚：帶有原始碼控制和更改請求管理的集成開發環境。幾乎每個開發團隊都會認識到代碼規定和跟蹤缺陷的重要性。在沒有中央存儲庫來管理代碼和缺陷的情況下，軟體的構建可能會是一個完全臨時的過程。像 Rational Application Developer for WebSphere® Software、Rational ClearQuest 和 Rational ClearCase® 之類的技術爲可重複、可度量的軟體創建建立一個基線。

- IBM Rational Application Developer for WebSphere Software 的設計旨在幫助開發人員快速構建高品質的 JavaTM、Java Platform, Enterprise Edition（Java EE）Web、Web 服務、門戶，以及 SOA 解決方案。集成開發環境（IDE）有助於快速設計、開發、裝配、測試和部署這些應用程式。軟體的可視工具透過抽象 Java EE 編程模型來幫助減少手工編碼。他們使您更輕鬆、更快速地完成開發專案，並讓您將精力集中在創造性的軟體解決方案上。

  要瞭解有關 Rational Application Developer for WebSphere Software 的更多資訊，請參見下列網址：
  http://www.ibm.com/software/awdtools/appscan/tester/

- IBM Rational ClearCase 是一個行業領先的解決方案，它提供了完善的版本控制、工作區管理、平行開發支援以及版本稽核，進而能夠提高生產率。這個全面的軟體配置管理產品提供了一個強化的集中部署模型，甚至使全球性團隊更易於協同工作。當開發團隊繼續面臨交付比以前品質更高、速度更快的軟體的更大壓力時， Rational ClearCase 能幫助簡化軟體交付流程並提高生產率。

  要瞭解有關 Rational ClearCase 的更多資訊，請參見下列網址：http://www.ibm.com/software/awdtools/clearcase/

- IBM Rational ClearQuest 提供了變更跟蹤、流程自動化、報告和生命週期可追溯性的特性，以獲得對軟體開發生命週期的更好的可視性和控制。其設計旨在幫助更有效地管理軟體生命週期。它使您能前往作出更佳決策所需的資訊。它幫助您更有效地管理任務和計畫，並快速回應客戶需求。Rational ClearQuest 的自動化工作流能幫助您控制和實施開發流程，並幫助改善團隊交流、生產率和品質。

  要瞭解有關 Rational ClearQuest 的更多資訊，請參見下列網址：http://www.ibm.com/software/awdtools/clearquest/

第二層元件涉及另外兩類：需求和測試管理。Rational DOORS® 等產品使業務專業人員和架構師能夠在軟體內定義清晰、可度量的要求。

Rational AppScan 產品線的自動安全性測試產品使自動化在不同的軟體開發生命週期階段有效地得到使用，進而針對定義的安全漏洞進行測試。
所有這些功能與 Rational Quality Manager 這類將資訊都收集到一處的產品相結合，創建了一個能夠跟蹤測試、測試結果和缺陷以及回答最後和最終的大問題的系統。這個大問題是：我們準備好發佈了嗎？以上兩個頂級元件對於交付安全軟體來說是最低的要求。

- IBM Rational DOORS 專爲那些想管理安全需求、編寫優良的用例、提高可追溯性、加強協作、減少專案風險和提高品質的專案團隊而設計。它爲分散式團隊提供了可伸縮而又快速的 Web 介面，並爲企業部署提供了增強的高度安全的模型。因此，對於業務分

析師、架構師、設計人員、開發人員和測試人員來說，它支援定制的需求前往，同時允許在安全軟體開發生命週期的多個階段中進行整合。

要瞭解有關 Rational DOORS 的更多資訊，請參見下列網址：http://www-01.ibm.com/software/awdtools/doors/

- IBM Rational Quality Manager 是一個基於 Web 的集中測試管理環境。它適用於業務、系統和 IT 決策者和品質專業人士。他們尋求一種用於測試計畫、工作流控制、跟蹤和指標報告的協作和可定制的解決方案，這個解決方案能夠定量分析專案決策和交付成果如何影響業務目標以及如何與業務目標保持一致。其設計旨在透過允許團隊無縫共用資訊、借助自動化加速專案計畫，以及報告項目指標以便做出明智的發佈決策，進而幫助他們進行協作。

  要瞭解有關 Rational Quality Manager 的更多資訊，請參見下列網址：http://www.ibm.com/software/awdtools/rqm/

近年來，傳統的“預先大量設計（big design up front，BDUF）”瀑布式方法的開發已經轉變爲迭代式和增量式的敏捷方法。這種轉換使企業能夠透過回答有關軟體成功和早期生存能力的重要問題來實現實際成本的節約。它還使企業能夠對客戶建議、與軟體相聯繫的需求做出更快速的反應。但是，這還意味著創建更多的構造。

Rational Build Forge 等產品提供了採用這種新方式構建軟體的解決方案。它使開發團隊能夠維護自動化的構建管理系統，該系統提供了這種新開發方法所需的及時回饋。將這一點看作構建安全軟體的重要元件的原因在於，與 Rational AppScan 的整合允許安全分析在每個構建版本上迭代完成。它提供了早期且有價值的回饋，降低了軟體開發生命週期後期的成本和風險。

- IBM Rational Build Forge 是一個自適應的流程執行框架。該框架自動化、編排、管理和跟蹤軟體開發的每種裝配線內每次傳送之間的所有流程，同時創建一個自動化軟體工廠。Rational Build Forge 整合到您的當前環境中，並支援主流開發語言、腳本、工具和平台。它使您能夠繼續使用現有的投資，同時在流程自動化、加速、通知和計畫方面添加了有價值的功能。

  要瞭解有關 Rational Build Forge 的更多資訊，請參見下列網址：http://www.ibm.com/software/awdtools/buildforge/

與安全性直接相關的下一層元件是架構和資產管理系統。Rational Software Architect for Web Sphere Software 等架構管理軟體使開發團隊能夠快速設計和重用成熟的、能實現元件間安全交互的安全設計。Rational Asset Manager 等資產管理系統使開發團隊能夠維護經過認證的安全元件庫。開發人員可以轉向這個元件庫來執行身份驗證、授權、輸入驗證、日誌記錄、稽核等普通任務。如果實現錯誤，該元件庫還可能危及系統的安全。

當這種成熟的資產庫不符合要求且必須編寫一個定制元件時，安全團隊會被召集起來進行評估。

- IBM Rational Software Architect for WebSphere Software 是一個功能強大、綜合的設計和開發環境。它能幫助 IT 架構師和開發人員跨團隊、跨不同的技術專家領域以及在全球瞭解、設計、管理和發展解決方案。其抽象、分析和報告功能旨在使交流和協作更高效。此外，其自動化和智慧編輯工具能夠幫助提高生產率、增強架構控制並使 Java 和 Java ，J2EE、Web 服務、SOA 和 Web 2.0 應用程式從設計到代碼的體驗變得輕鬆。

  要瞭解有關 Rational Software Architect for WebSphere Software 的更多資訊，請參見下列網址：

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

- IBM Rational Asset Manager 透過促進與軟體開發相關的所有類型的資產重用來降低

軟體開發成本並提高品質。它是一個協作性軟體開發資產管理解決方案，能幫助定位、共用和跟蹤跨業務和部署團隊的資產。Rational Asset Manager 使組織級分散式開發團隊能夠確定、管理和治理軟體資產（包括作爲 SOA 計畫組成部分的服務）的設計、開發和消費。 該軟體推動了協作軟體資產的開發、部署和使用，幫助 IT 機構交付創新的 IT 解決方案，同時控制了成本、降低了應用程式延遲並提高了業務的靈活性和回應能力。

要瞭解有關 Rational Asset Manager 的更多資訊，請參見下列網址：http://www.ibm.com/software/awdtools/ram/

當您在操作環境內達到部署階段後，最後一層產品能使您保持所交付的產品及其資料的安全性。IBM OptimTM Data Privacy Solution 能確保您的客戶資料在發生洩漏的情況下得到充分的保護和遮罩。爲了自動化保持企業內網的安全性的任務，可以將 Proventia® Network Enterprise Scanner 和 Proventia Network MFS 納入到您的網路基礎設施中。爲了保持伺服器和桌面得到充分修補，IBM Tivoli® Security Compliance Manager 提供了一個跟蹤機器安全的集中解決方案。

- IBM Optim Data Privacy Solution 能夠保護客戶資料的隱私。取消機密資料的標識是保護隱私並支援 HIPAA、DPP、PIPEDA、PCI DSS 等法規的遵從性的一種最佳方法。Optim Data Privacy Solution 交付了強大的資料轉換功能，可遮罩機密企業資料，使您能夠安全地將其用於應用程式測試。您可以透過應用簡單的資料遮罩技術保護易受攻擊的測試環境，也可應用預先打包的轉換演算法來處理複雜的資料元素，如信用卡號碼、電子郵件位址和身份證號碼。

  要瞭解有關 Optim Data Privacy Solution 的更多資訊，請參見下列網址：

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

- IBM Proventia Network Enterprise Scanner 使您能夠瞭解您的網路上正在傳輸的資料以及潛在問題位於何處。該解決方案對確定和管理風險具有重大意義。遵守安全法規並簡略說明修補措施可能是代價高昂、勞動力密集型的。Proventia Network Enterprise Scanner 使您能在管理網路漏洞方面節省成本和時間。Proventia Network Enterprise Scanner 能夠幫助確保創收服務的可用性並透過確定風險並區分其優先次序、分配保護活動和報告結果來保護您的企業資料。

  要瞭解有關 Proventia Network Enterprise Scanner 的更多資訊，請參見下列網址：

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

- IBM Proventia Network Multi-Function Security（MFS）是一種統一威脅管理（UTM）設備，在閘道和網路級提供保護，同時不會影響網路帶寬或可用性。它可以一次性防禦多種威脅，比如未授權的前往、網路攻擊、惡意代碼、混合型威脅、基於內容的攻擊、間諜軟體和網路釣魚攻擊。Proventia Network MFS 將這些業界最佳的（best-of-breed）安全模組整合到一個單獨的高性能和易用的 UTM 設備中：防火牆/VPN、入侵預防、防病毒、防垃圾郵件、Web/URL 篩檢程式和應用程式保護。

  要瞭解有關 Proventia Network Multi-Function Security 的更多資訊，請參見下列網址：

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

- IBM Tivoli Security Compliance Manager 確定安全漏洞和安全策略違規。它透過定義一致的安全策略和檢測這些限定的安全策略的遵從性來保護您的業務，避免攻擊者憑藉易受攻擊的軟體配置進行的攻擊。它自動掃描伺服器和桌面系統，能幫助減少與手動安全性檢查相關的成本和時間。

  Tivoli Security Compliance Manager 向安全官員和遵從性稽核人員報告詳細的業務安全狀態資訊，以便他們能採取適當的步驟使各個系統和部門合規。它在安全事件造成重大損失前識別軟體安全漏洞，改善業務經營並透過自動化和集中方式幫助提高效率。Tivoli Securi

ty Compliance Manager 透過自動化遵從性任務、監測通信、減少人爲錯誤和降低遵從性成本來幫助解決法規和標準遵從性問題。

要瞭解有關 Tivoli Security Compliance Manager 的更多資訊，請參見下列網址：

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

圖 8 顯示了如何將所有這些技術結合到一個完整的安全開發生命週期中。透過將與安全相關的 IBM 產品整合到開發生命週期的各個步驟中，您的最終產品的安全性可以得到顯著增強。

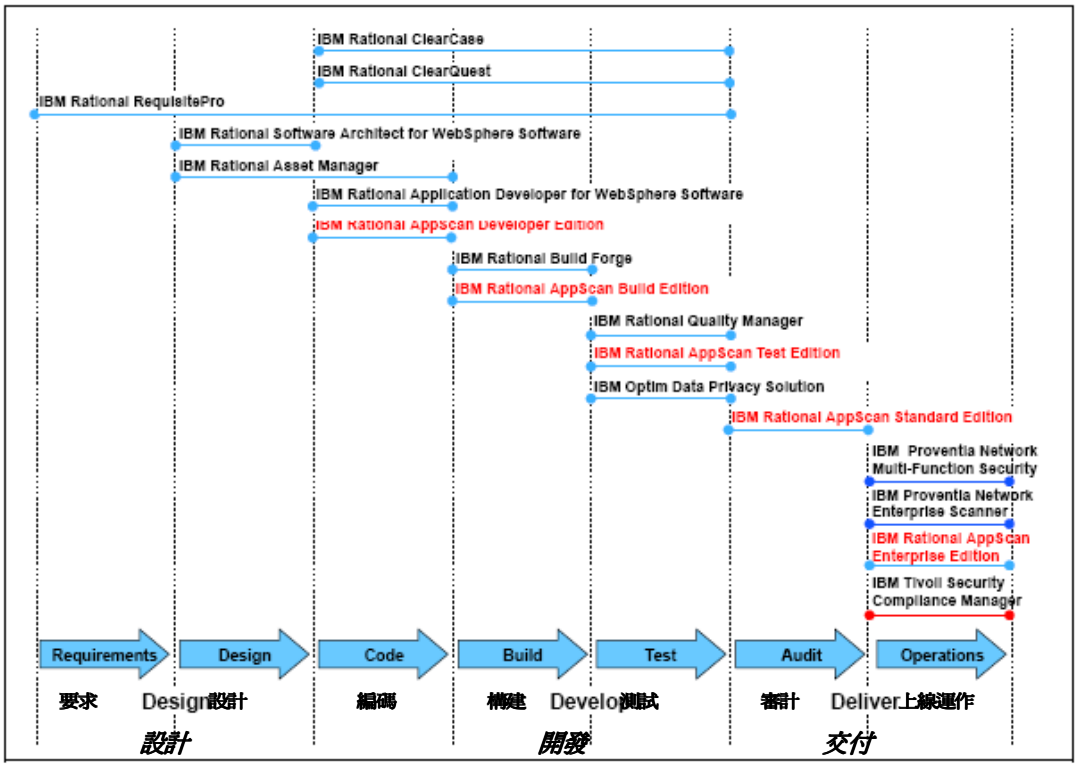


圖 8 結合 IBM 技術來創建全面、綜合的安全平台

業務情節：實現 Web 應用程式安全的逐步實行方法

在本節中，我們研究這樣一種業務情節：一家原本沒有安全測試的組織在其軟體開發生命週期中整合了最新的安全測試。在本情節中，獲悉比它規模更大的競爭對手出現了一些安全問題之後，這家虛構的公司決定考慮 IBM 產品來提高安全性，以避免其競爭對手遭受的負面效應，並提供高品質、安全且基於 Web 的產品以使自己脫穎而出。

與 IBM 商議之後，該公司決定逐步將其現有的不安全的軟體開發生命週期轉化爲具有安全意識的生命週期，以生產高品質、安全且基於 Web 的產品。因爲該公司希望快速取得成果，即直接瞭解其現時安全狀況，所以他們最初選擇透過 Rational AppScan On Demand Production Site Monitoring 與外包稽核團隊合作。此服務允許他們獲得直接回饋，同時準備將安全性深入整合到系統中。借此，他們獲得了正確的資源和正確的專家，可進一步開發其特有安全方案。在執行此工作的同時，他們開始

針對安全性培訓所有開發生命週期相關人員。此培訓依靠基於 Web 的嵌入式培訓模組來推行，這些模組有助於說明漏洞、展示 Rational AppScan On Demand 所實現的業績。

該公司培訓了一批自己的員工，完全可以不依賴於服務，而能夠自己執行安全測試，他們獲得了 Rational AppScan Standard Edition 的必要產品許可，構建了自己的內部安全稽核團隊。使用 Rational AppScan Standard Edition 的自動掃描功能，內部稽核人員在開發生命週期結束時發現了漏洞。他們將漏洞報告給開發人員，以便後者在產品發佈前進行修復。

由於持續對整家公司進行安全性培訓，而且內部安全稽核團隊向開發人員提交了回饋，所以該公司的開發人員很快就能夠承擔更大的安全責任了。爲了在開發生命週期之初就引入安全測試，開發人員配備了 Rational AppScan Source Edition。這允許他們在編碼時自動發現漏洞，並使他們瞭解必須避免的、與安全相關的、不好的開發實踐。透過在開發生命週期中較早地捕獲漏洞，安全稽核團隊的資源得到了釋放，他們可以關注更複雜的漏洞，因此進一步增強了公司的安全水準。

接著，更多安全測試被自動化，並整合到構建和測試流程中。Rational AppScan Build Edition 和 AppScan Test Edition 用於實現這一目的。這時，公司已經在整個軟體開發生命週期中分佈了安全測試。

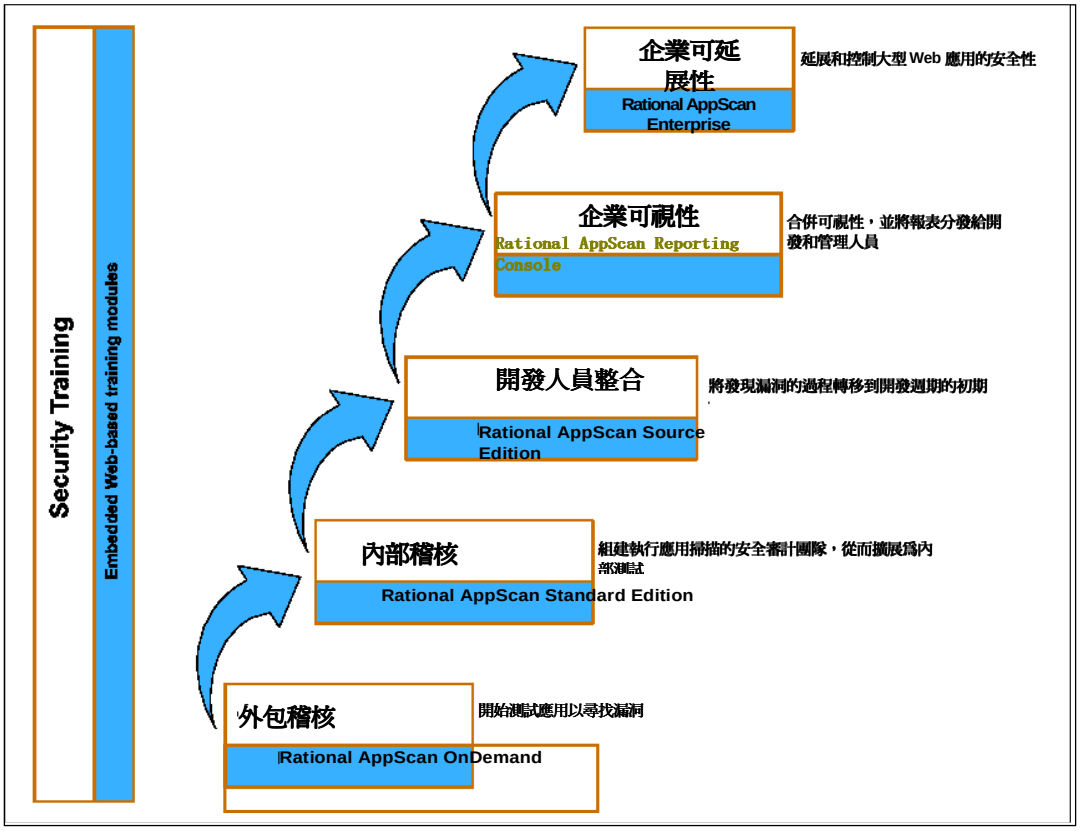
爲了讓開發和管理人員更明確地瞭解開發專案進展中的安全進程，該公司開始將 Rational AppScan Reporting Console 用於集中收集安全資料。利用 Rational AppScan Reporting Console，他們能夠縱覽其軟體開發生命週期中使用的不同軟體工具創建的所有安全報表，進而提高了可視性，並進一步增強了安全性。

集中所有的漏洞掃描報表，這使該公司已對安全問題充滿自信，同時保持其安全性在掌控之中。該公司在 Web 應用程式領域將其安全性作爲關鍵賣點，展示自身優於競爭對手的不同，進而吸引了市場的注意，開始迅速成長。這導致該公司整體需要充分可伸縮的安全測試解決方案。

爲了適應其迅速發展的 Web 環境的規模，該公司將 Rational AppScan Enterprise Edition 引入到軟體開發生命週期中。這種可伸縮的企業架構使他們能運用分散式掃描代理，持續掃描大量的 Web 應用程式。他們能從 Rational AppScan Enterprise Edition 中的一個集中報告點控制那些掃描代理。這樣他們就能持續快速發展，獲得無限擴充的安全掃描和稽核能力。

圖 9 概述了該公司如何逐步將安全性整合到軟體開發生命週期中，轉變成爲擁有最新安全測試的快速成長的公司。

圖 9 逐步實現 Web 應用程式安全的方法



圖字：
Security Training：安全性培訓
Embedded Web-based training modules：基於 Web 的嵌入式培訓模組

結語

我們看到 Internet 上有這樣的變化，從渴求出名的駭客執行的破壞活動，發展到有組織的資料和身份竊賊爲獲利而進行的詐欺，因此企業領導必須將 Web 應用程式安全性視爲業務成功的關鍵指標。

這份 IBM Redguide 從攻擊者的角度研究了 Web 應用程式安全性。透過窺探攻擊者的動機和操作方式，我們示範了他們如何依靠攻擊 Internet 上的組織而獲取金錢。我們解釋了對他們而言爲何攻擊 Web 應用程式是有利可圖的生意，以及他們如何使用自動化來削減成本、獲取更大利益。

接著，我們向您展示了如何採用與他們進行自動化攻擊相同的方式，在軟體開發生命週期中依靠自動化來保護您的組織。我們介紹了 IBM Rational AppScan 產品線，說明了您如何將此產品線整合至您的軟體開發生命週期中，進而借助最新的安全測試，改善您的 Web 應用程式的整體安全狀況。

最後，我們展示了這樣一個情節，一家沒有 Web 應用程式安全測試或知識的公司，轉變爲一家

將最新 Web 應用程式安全測試以一種可伸縮且可控制的方式充分整合到其軟體開發生命週期中、具備安全意識的公司。我們說明了交付高品質的安全 Web 應用程式如何使一家公司在競爭對手中脫穎而出，並增強其市場地位。

其他資源中的更多資訊

參考本指南中已引用的以下資料，瞭解更多資訊：

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 應用程式安全協會， “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. <\0 開放 Web 應用程式安全項目（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 主頁），Information Technology Laboratory

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

本文創作團隊

本文由國際技術支援組織（International Technical Support Organization，ITSO）中來自世界各地的專家組成的團隊編寫。

Frederik De Keukelaere，IBM Research 日本東京研究實驗室研究員。他是 Security and Web Platform 小組的成員，從事下一代 Web 安全模型研究。他目前的研究興趣在於可用的 Web 安全性。他在 Web 應用程式安全性、Web 以及多媒體技術方面有超過 7 年的研究經驗。他曾積極參與了多個標準組織，例如 OpenAjax Alliance 和 Moving Picture Experts Group (MPEG)，並爲之做出了貢獻。他是數個 ISO/IEC 標準的編者，並擁有 5 項 ISO/IEC MPEG-21 傑出技術貢獻獎。在 2006 年加入 IBM 之前，他在比利時多媒體實驗室，是 BroadBand Technology 的交叉學科（Interdisciplinary）協會成員。在此期間，他獲得了比利時根特大學電腦科學工程專業博士學位。

Danny Allan，IBM Rational 安全性研究主管。由於 2007 年 7 月對 Web 應用程式安全和遵從性領域的領袖 Watchfire 的收購，Danny 來到了 Rational。他具有超過 8 年的業務和安全技術相關經驗，包括曾爲加拿大最大的大學之一從事滲透測試和內部系統補救。在安全性研究員這一角色上，他密切參與企業的全球客戶部署，研究和評估技術，並協助定義和推薦戰略方向。Danny 擔任著數個面向客戶的關鍵職位，包括團隊領導、諮詢服務以及銷售工程師。他已發表多篇白皮書和文章，並加入了行業工作組。他還經常在安全性活動中發言，接受包括 Associated Press、Bloomberg 以及 Wall Street Journal 在內的重要媒體的拜訪，抒發關於 Web 應用程式安全性的觀點。Danny 擁有卡爾頓大學資訊系統專業的商學學士學位。

Axel Buecker，德克薩斯州奧斯丁 ITSO 的認證諮詢軟體 IT 專家（Certified Consulting Software IT Specialist）。他就軟體安全架構和網路計算技術領域撰寫了大量 IBM 課程，並在全球教授。他擁有德國不萊梅大學電腦科學專業學位。他在與工作站和系統管理、網路計算以及電子商務解決方案相關的多個領域擁有 22 年的經驗。在 2000 年 3 月加入 ITSO 之前，Axel 在德國 IBM 擔任軟體安全架構方面的高級 IT 專家。

感謝以下人員對本專案的貢獻：

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

注意事項

本資訊適用於在美國提供的產品和服務。

IBM 可能不在其他國家/地區提供本文檔討論的產品、服務或功能。諮詢您本地的 IBM 代表，瞭解有關本地區當前可用產品和服務的資訊。對 IBM 產品、程式或服務的任何引用不聲明或暗示只可以使用該 IBM 產品、程式或服務。也可以使用任何不破壞 IBM 知識產權的類似產品、程式或服務。然而，評估或驗證任何非 IBM 產品、程式或服務屬於用戶自己的責任。

對於本文檔中描述的主題內容，IBM 可能具有專利或正在申請專利。本文檔的描述未賦予您針對這些專利的任何許可。您可以以書面形式將許可查詢發送給：
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

以下內容不適用於英國或者其他與本地法律不一致的國家：國際商業機器公司根據“現狀”提供本出版物，不提供任何明確或隱含的擔保，包括但不限於關於非侵權、適銷性、符合特定用途的適用性所有隱含擔保。某些國家在某些事務中不允許明確或隱含擔保的免責聲明，因此，此聲明可能不適合您。

本資訊可能包含技術錯誤或排版錯誤。這裏的資訊會定期變更，這些變更將合併到本出版物的新版本中。IBM 可能隨時對產品和/或程式做出改進和/或變更，恕不通知。

本資訊中對非 IBM Web 網站的引用僅出於方便考慮，不能以任何方式看作對這些 Web 網站的認可。這些 Web 網站上的內容不是本 IBM 產品資源的一部分，使用這些 Web 網站時風險自負。

IBM 可能以它自己認爲合適的方式使用或分發您提供的資訊，而不會承擔對您的任何責任。

有關非 IBM 產品的資訊是透過這些產品的提供商、他們發佈的公告或其他公共可用的來源獲得的。IBM 沒有測試過這些產品，不能確認與非 IBM 產品相關的性能、相容性或任何其他聲明的準確性。關於非 IBM 產品功能的問題應該由這些產品的提供商解決。

本資訊包含日常業務運營中使用的資料和報告的範例。爲了盡可能完整的闡釋它們，這些範例包括個人、公司、商標和產品的名稱。所有這些名稱都是虛構的，如果同實際企業使用的名稱和位址雷同，純屬巧合。

版權許可：

本文包含使用源語言的樣本應用程式，展示了在多種操作平台上的編程技巧。爲了開發、使用、推廣或分發符合操作平台應用編程介面（樣本程式正是爲之編寫）的應用程式，您可以以任何形式複製、修改和分發這些樣本程式，而無須向 IBM 支付費用。這些範例未在所有環境中經過徹底測試。因此，IBM 不能保證或暗示這些程式的可靠性、有效性或功能性。

本文檔，REDP-4530-00，創建或更新於 2010 年 12 月 12 日。

紅皮書®

商標