資訊安全威脅 03

本章概要 ▶

資訊安全威脅的種類繁多，從來自網路上的駭侵攻擊，到人與人之間所使用的詐騙手法，不一而足。但不論攻擊手法為何，它們都有類似的目的，就是要破壞資訊的機密性、完整性、與可用性。我們將在本章介紹常見的資訊安全威脅，包括惡意程式、網路入侵、與人為詐欺等，在下一章深入地討論某些駭客常用的攻擊手法。

3.1 資訊安全威脅的目的

資訊安全的目的在維護資訊的機密性（Confidentiality）、完整性（Intigrity）、及可用性（Availabilty），這三者常被稱為資訊的「CIA」。機密性在防止未經授權的人或系統存取資訊；完整性在確保被使用的資訊是正確的；而可用性則是確保資訊服務隨時可用。資訊安全威脅破壞資訊的 CIA，目的如下：

- **達到「侵入（Access）」的目的**。讓沒有進入權限的人或系統能夠未經授權地使用他人資源；目的在破壞資訊的機密性。
- **達到「篡改（Modification）」或「否認（Repudiation）」的目的**。讓沒有修改權力的人或系統能夠竄改他人資訊或否認某些事實；目的在破壞資訊的完整性。
- **達到「拒絕服務（Denial-of-service, DoS）」的目的**。讓惡意的人或系統能夠干擾或阻斷他人網路或服務；目的在破壞資訊的可用性。

3.1.1 以侵入為目的

侵入的手法很多，例如有一種實體攻擊方法稱做「垃圾搜尋（Dumpster Diving）」，攻擊者在垃圾箱裡尋找可能含有密碼或機密訊息的廢棄紙張。許多組織為了環保將廢紙集中回收，反而使垃圾搜尋更為便利。機密文件在回收前必須以碎紙機銷毀。在辦公室或其他場合，竊聽與窺視也常見。例如在洗手間或餐廳裡不經意的對話，也可能成為有心人士的竊聽機會。實體文件或是電腦檔案有可能遭窺視或利用，辦公桌上的小紙條也經常是不經意洩密的元凶。

當然侵入攻擊者也常使用科技手法，如電話監聽或在網路上掛線監看（sniffing），或是在網路服務提供者（ISP）那裡裝置側錄功能，以目前盛行的惡意程式而言，其中以竊取受害者資料為目的的側錄軟體，除了一般的使用者之外，對於特權管理者，例如：應用程式管理員、系統管理員等，都成為此類惡意程式的攻擊對象，以獲取管理權限的帳號密碼為主要的目標。

3.1.2 以竄改或否認為目的

篡改是指未經授權的刪除、插入、或更改資訊，並期望別人無法察覺。例如學生竊用老師的帳號密碼更改自己的考試成績，或是駭客侵入銀行網路刪除信用卡消費紀錄等。而否認則是將正確的資訊弄成無效或誤導的狀態。例如惡意者冒名發郵件騷擾他人，或是惡作劇學生侵入系上網站發布放假三天的消息等。

否認的反面為「不可否認（Non-repudiation）」，是指藉由提供原本的證據，使寄件人不能否認曾發出信息，或收件人不能否認曾收到信息。例如在網路上購物，商家有時不只要求消費者提供信用卡號，還要信用卡的 PIN。藉由消費者的身分確認，商家就有不可否認的證據來進行交易。

近年廣為使用的社交網站也成為竄改攻擊的目標，許多來自社群網路平台上的訊息，除了真假難在第一時間辨別外，也可能遭到身分的假冒，造成周遭朋友遭到詐騙等情事發生，我們了解攻擊者的目的未必在於獲取機密，竄改或否認一樣能為攻擊者帶來經濟或政治的利益，加上虛擬貨幣的流通，讓駭客組織或攻擊者更加的運用虛擬貨幣的電子錢包，並且成為資安事件中勒索受害人的方式，過去幾年經常聽到國內外的產業大廠，遭受勒索軟體的威脅，已造成莫大的金錢損失。

3.1.3 以阻斷服務為目的

阻斷服務是讓受害的網路或伺服器忙於處理假的服務要求，而無法處理真的要求；或直接破壞作業系統或硬碟上的資料使資訊服務無法繼續。例如 ping of death 就是一種阻斷服務攻擊，攻擊者利用 ping 工具產生超過 IP 協定所允許的最大封包，使受害者電腦當機。緩衝區溢位（Buffer Overflow）也是一種 DoS 攻擊手段，攻擊者傳送超過緩衝區大小的資料給系統，使它覆蓋其他資料區域，造成系統失敗。

「分散式阻斷服務（Distributed DoS, DDoS）」則是由駭客的主機控制網路上多台傀儡電腦（Zombies）同時對受害者發動 DoS 攻擊。而發動攻擊的傀儡電腦的使用者其實也是不知情的受害者。在惡意程式的活動中所形成的殭屍網路（Botnet），往往也是駭客經常用來發動分散式阻斷服務攻擊的來源。

病毒及蠕蟲等惡意程式的目的也在破壞資訊的可用性，但不同於駭客所發動的 DoS 或 DDoS 攻擊，惡意程式的撰寫或散佈未必具有針對性，可能只是惡作劇或者炫耀，從過去幾年在網際網路上發生的幾起巨量攻擊事件，就不難看出此類攻擊行為對於網路與系統服務所帶來的影響。

總結來說，資訊安全威脅的目的是為了破壞資訊的機密性、完整性、與可用性，手法可能是技術性的，也可能以詐術為主。

3.2 認識一般的攻擊

一般的攻擊是利用系統潛在的弱點，例如軟體或通訊協定在設計或安裝上的漏洞，來達到侵入、篡改與否認、或拒絕服務等一個或多個目的。近年來透過惡意程式進行的攻擊已大量出現，尤其部分的惡意程式攻擊的對象，已從一般的資通訊設備，轉向物聯網裝置或是工業控制系統（SCADA），大幅的擴展了資安攻擊原本所定義的範疇。

3.2.1 攻擊通關密碼

設定通關密碼是最常見的存取控制方法，因此密碼破解（Password-cracking）也是常見的攻擊類型。破解密碼有幾種可能的方式：一種是直接進行網路監看，另一種是「窮舉攻擊（Brute-force Attack）」，再一種則是所謂的「字典攻擊（Dictionary Attack）」。窮舉攻擊會逐一地嘗試所有可能的密碼組合；而字典攻擊則利用一個預先定義好的檔案（稱為字典），裡面存有較常被使用為密碼的單字。由於幾乎沒有人會設定真正隨機的通關密碼，所以字典攻擊法經常快速有效。

為了防禦密碼猜測攻擊，通關密碼的設定應該要夠長，才能增加窮舉攻擊的難度；通常建議八個字的密碼長度。另外，密碼的選擇要夠冷門，以降低字典攻擊的成功率。不應該使用有意義的單字，而且要夾雜字母與數字為佳。最後，要經常更換通關密碼，以防被破解的帳號遭駭客長期利用。

系統管理員通常擁有較高的權限，可以決定系統的存取授權，因此內部人員管理非常重要，但近年來惡意程式活動的猖獗，更以特權管理者為主要目標，進行特權帳號與密碼竊取，進而假冒管理員的身分進行特權操作，或是變更系統上的組態設備，而嚴重影響了系統本身的安全性。

3.2.2 利用後門

後門攻擊（Backdoor Attack）是指駭客利用程式存在的漏洞進出系統或網路。後門的產生有兩種途徑，一種是軟體開發者原先設計的維護後門（Maintenance Hook）。開發一個複雜的作業系統或應用軟體時，工程師常在程式裡設計後門以利測試與修改，但必須

在產品上市前移除。忘記移除後門常造成嚴重的資訊安全事件，因此若事後才發現維護後門沒有移除，應儘速以補丁修補。

另一種後門由入侵者所植入，為了重新進入。最常見的是木馬程式，可能透過電子郵件或惡意網站進入受害的系統。駭客可以利用木馬程式操控該系統，例如開啟連接埠或關閉防火牆。另外，像 Netcat 之類的工具能夠快速的建立主從架構的服務，讓遠端主機透過網際網路控制另一台電腦，部分公司用它做遠端管理員，但駭客也用它做後門攻擊。

3.2.3 攔截與偽裝

中間人攻擊（Man-in-the-middle Attack）是指在伺服器與客戶端之間放置雙方都無法察覺的軟體，如圖 3-1 所示。它能攔截一方的資料，備份或篡改之後若無其事地傳送給另外一方。隨著無線網路盛行，中間人攻擊更容易成功。一來許多人使用無線網路時並未加密；二來無線網路的中間人攻擊不需要實體掛線，在各種方便的位置都可以進行。

▲ 圖 3-1 中間人攻擊

重放攻擊（Replay Attack）是指攻擊者攔截使用者的登入資料，稍後再正式登入伺服器，如圖 3-2 所示。重放攻擊對 Kerberos 之類的登入系統是有效的。為防禦重放攻擊，加解密過程常會使用會談金鑰（Session Key），這種金鑰在會談完成後就自動失效。

▲ 圖 3-2 重放攻擊

欺騙攻擊（Spoofing Attack）是指攻擊者偽裝成一個熟悉並且可信任的伺服器或網站，藉以騙取登入資料或其他秘密資訊。網路釣魚是一種欺騙攻擊，它可能是一個看似有公信力的惡意網站，或是冒名的電子郵件要受害者連結到惡意網站，再騙取秘密資訊。

歹徒安置假的 ATM 機器也是一種欺騙攻擊，受害者插入磁條提款卡並輸入密碼後即遭電子側錄。提款卡晶片化後，側錄幾乎不可能，但 ATM 欺騙手法又被翻新。中國農業銀行在 2007 年曾張貼以下告示：「不法分子在 ATM 機插卡口處安裝吞卡裝置造成吞卡故障，並在 ATM 機旁張貼假的銀行告示，誘騙持卡人按假告示上的聯系電話與冒充銀行工作人員的不法分子聯繫，不法分子騙持卡人說出銀行卡密碼後，支開持卡人，從 ATM 機取出銀行卡並盜取持卡人銀行卡資金。」可見道高一尺、魔高一丈，資訊安全是永不停止的攻防。

2016 年第一銀行爆發大規模 ATM 盜領事件，全台灣共有 41 台 ATM 自動提款機因遭攻擊者植入惡意程式，而自動吐鈔超過 8 千萬元帶來新的資安思維，因第一銀行內網遭到駭客滲透，並且利用郵件社交工程進行滲透，運用惡意程式遠端控制 ATM 進行吐鈔，而該集團已在全球 100 家以上的銀行盜取了超過 360 億元，讓受到入侵的銀行遭受極大的損失。（取材自天下雜誌 2018/6/27）

3.3 認識軟體弱點的利用

軟體弱點不一定是軟體錯誤，只是攻擊者利用某種軟體「功能」或合併數種功能來達到攻擊之目的。像資料隱碼（SQL Injection）就是一種利用輸入查驗不完整所發動的攻擊。舉例說明，網站設計者將「使用者名稱」這個欄位的輸入資料直接傳送給 SQL Server，這個欄位的輸入應該是文字，但攻擊者輸入之資料中刻意含有某些對資料庫系統有特殊意義的符號或命令時，便可能讓攻擊者有機會對資料庫系統下達指令，而達到入侵的目的。

巨集病毒（Macro Virus）以巨集程式語言來撰寫，這種程式依附在該類型檔案中，並常經由電子郵件被傳播。梅莉莎病毒與 Taiwan No.1 都屬於巨集病毒。巨集程式語言是應用程式（如 Microsoft Word 或 Excel）提供的一項強大功能，有時為了資訊安全考量，而不得不予以限制。

除此之外，電子郵件系統提供的附加功能，如通訊錄，有時會被攻擊者利用做病毒擴大散播的途徑。間諜軟體（Spyware）常經由電子郵件或網站下載等途徑入侵，使用者常在不知情的狀況下自己將間諜軟體載入電腦，它會收集使用者電腦活動或顯示廣告等，大

多為了商業利益。Rootkit 病毒則利用作業系統的一些特性隱藏自己，它在執行時，視窗工作管理員都找不到它的蹤跡。

3.4 認識惡意程式

惡意程式（Malicious Code or Malware）泛指對網路與系統造成威脅的軟體，可被略分為病毒、蠕蟲、木馬、與邏輯炸彈等。已被發現的惡意程式會被公佈在公開網站，讀者可以前往參閱：

- 趨勢科技網站：https://www.trendmicro.com/
- VirusTotal 網站：https://www.virustotal.com/
- TWCERT/CC 惡意檔案檢測服務網站：https://viruscheck.tw/

3.4.1 認識病毒

病毒（Virus）是一個寄居在其他程式上的小軟體，它可能僅只生存在電腦內不造成傷害，也可能刪除磁碟上的資料、破壞作業系統、或傳染別台電腦。電腦病毒的存在主要有兩大目的：第一是傳染給別台電腦；其次是讓受害電腦不能運作。

電腦病毒傳染主要有三個途徑：

- 經由受感染的可移式媒體（Removable Media）如軟碟、CD ROM、USB 碟傳染給其他電腦。
- 經由電子郵件的附件傳染，這類病毒常利用受害者的通訊錄傳送病毒給更多的潛在受害者。
- 附著在別的正常軟體上。尤其越來越多人肆意的從網路上下載軟體，卻未細究該軟體是否已遭病毒感染。

電腦一旦感染病毒後常有以下的徵狀：

- 可能系統被控制、螢幕上出現惱人的訊息、甚至硬碟資料被摧毀。
- 可能造成系統上的程式變慢，因為病毒佔用了部分電腦資源。
- 可能有些檔案會消失，因為病毒常刻意刪除重要檔案。

- 可能有些程式的大小會改變，因為病毒附著其上。
- 可能造成系統突然關閉，或磁碟經常做沒有意義的讀寫。
- 可能突然無法使用硬碟或其他電腦周邊。

電腦病毒有時會對自身做加密編碼或壓縮造成變形，企圖躲避掃毒工具的偵測。有的病毒會隱身在硬碟的 Boot Sector，或當防毒軟體掃描時在不同的檔案間移動，目的都在躲避偵測，並延長寄生壽命。

3.4.2 認識蠕蟲

雖然蠕蟲（Worm）與病毒兩個名詞常被混用，但在正式定義上，它們有兩個主要差異：第一、蠕蟲可以自己存在，不需要寄生於別的程式或檔案。第二、蠕蟲可以複製自己，並自行在網際網路上傳播，不需靠人的參與。蠕蟲造成的傷害經常範圍極廣，因為蠕蟲在受害電腦上大量複製，再經由郵件通訊錄上的地址或網路的 IP 位址傳播。

2001 年的紅色警戒（Code Red）蠕蟲一天內癱瘓三十餘萬台電腦。蠕蟲的快速複製與傳播的能力常大規模的占用系統資源，例如記憶體與網路頻寬，導致網站、網路服務、與電腦系統無法正常運作，形成阻斷服務的結果。

3.4.3 木馬與邏輯炸彈

不同於病毒或蠕蟲，木馬（Trojan Horse）程式不會自行複製、傳播或寄生，它進入受害者電腦的管道是靠著使用者錯誤的判斷。木馬程式偽裝成別的程式進入系統或網路，表面上是個使用者想要的程式，例如從網路上下載一個電玩遊戲，安裝之後也能正常操作；但這個程式可能同時在電腦內植入病毒、施放蠕蟲或開啟後門。例如，曾有木馬程式偽裝成「waterfalls.scr」這樣的螢幕保護程式。由於 .scr 為可執行檔且螢幕保護程式執行時防毒功能經常是關閉的，使惡意程式能順利運作。

邏輯炸彈（Logic Bomb）是被放置在受害系統中的軟體程式，被設定在某種條件下啟動一些破壞性的功能。病毒或蠕蟲等惡意程式也常伴隨著邏輯炸彈的設計，在某條件下啟動攻擊。這樣做可以讓程式散布得夠廣之後，才同時爆發。較常見的發作日期是十三日星期五或是四月一日愚人節等。美國 UBS 銀行一位心有不滿的系統管理員 Roger Duronio 被指控使用邏輯炸彈傷害公司的電腦網路。在 2002 年 3 月 4 日當天，公司兩千台伺服器同時受攻擊而停止運作，四百家分行受到影響，造成巨大損失。Duronio 被判刑八年，並須賠償 UBS 三百一十萬美元。

防制病毒或蠕蟲傳播主要靠防毒軟體以及新一代的端點預警系統（Endpoint Detection and Response, EDR），對於端點進行偵測與異常行為的回應，它是安裝在系統上的軟體，能主動地掃描惡意程式，包括了病毒、蠕蟲、或木馬程式等。目前全球有不可計量的已知病毒和其他惡意程式，大部分已知的病毒都被歸納出個別的特徵，防毒軟體按照這些特徵尋找病毒並予消除。然而持續有人製造新的惡意程式，我們要繼續依規定更新病毒碼以防禦新型態的攻擊行為。防禦的軟體應同時存在於伺服器、閘道系統與個人電腦上，會有更好的縱深防禦效果。

3.4.4 勒索軟體

近年來勒索軟體改變了資安事件的型態，也成為攻擊者最喜歡運用的攻擊手法，從早期單純的影響系統運作，轉而對於數位資產進行竊取與加密，配合虛擬貨幣進行威脅與勒索，此類事件大量的出現，攻擊者的目標也從一般的資訊環境，轉向製造業等廠區機台。2018 年台積電遭受勒索軟體攻擊，而機台大當機帶來產線中斷造成出貨延遲以及相關成本的增加，整個事件的損失高達 78 億元，而機台感染到的惡意程式就是名為 WannaCry 的勒索軟體。

許多產業都曾經遭受到各種類型的勒索軟體攻擊，不論是一般的資訊環境，或是產業環境中的工廠設備，都已有許多的案例發生，一旦遭到了勒索軟體攻擊，最重要的一件事就是如何快速的讓受到攻擊的資通訊設備、設備機台儘快的回復運作，以確保企業仍然可以持續營運與發展。

3.5 認識網路攻擊

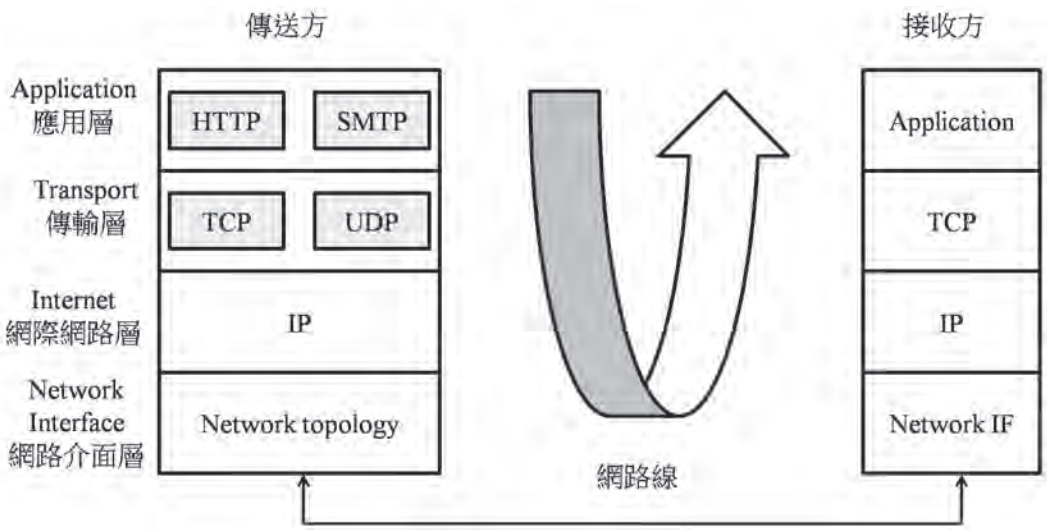
過去駭客多藉由網路或可移式媒體，以無目標的方式散播病毒或蠕蟲，但這種攻擊方式很難為攻擊者帶來實際利益。除此之外，作業系統的安全功能被強化，防火牆及防毒軟體更普及使用，這種攻擊方式已經不容易再造成本世紀初的 Code Red 或 Nimda 蠕蟲的殺傷力。然而這並不代表網路的威脅降低了，駭客開始鎖定特定對象進行攻擊。這種攻擊的成功機率更高，而且具有經濟價值。駭客挑選特定對象之後，可能按照以下步驟進行網路攻擊：

- **偵察**：駭客會先送出各式的探測封包，以獲得特定對象的網路資訊。
- **測試**：依據這些資訊，駭客會找出可以從外部入侵內部的弱點。

- **侵入**：駭客為方便侵入系統會先弱化其安全防禦功能，例如掌控其防火牆。如果已經由欺騙手法植入木馬，侵入就更容易了。
- **控制**：駭客控制特定對象電腦的方式之一，就是在他的系統啟動程式中插入控制碼，為入侵者建立一個遠端控制的入口。
- **利用**：控制特定對象的電腦之後，駭客就可以恣意地使用它的資源，包括分享機密資訊。
- **轉戰**：駭客會使用其所控制的電腦，攻擊其他網路或系統；前述被用來發動 DDoS 攻擊的傀儡電腦就是例子。駭客也可能利用被控制的電子信箱發更多的釣魚郵件給受害者認識的人。

3.5.1 認識 TCP/IP 協定

幾乎所有大型網路，包括網際網路都使用 TCP/IP 協定，它的彈性和使用便利性卻造成安全疑慮。TCP/IP 協定可分為四層，如圖 3-3 所示。



▲ **圖 3-3** TCP/IP 協定的四層

「應用層（Application Layer）」是 TCP/IP 的最上一層，讓應用程式透過服務與協定來交換資料，大部分程式包括瀏覽器都在這一層與 TCP/IP 互動。常見的應用層協定包括：HTTP 是在 www 傳送檔案的規則，FTP 是主機與網際網路間傳輸資料的協定，SMTP 協定用來傳送、接收、轉寄電子郵件，POP 是一種主從式協定，定義伺服器如何為使用者接收、儲存電子郵件，Telnet 是一種互動式終端機的模擬協定，還有 DNS 在網際網路上將網域名稱轉換為 IP 位址。

開發維運安全 13

本章概要 ▶

13.5 應用程式安全檢測

對於應用程式的安全檢測，是目前經常用來評估可能潛在資安風險的方式，從應用程式開發走向服務平台維運的過程，資訊安全可以扮演重要的角色，建構一個以資訊安全為前提所發展出來的應用程式，對於後續的維運將可以降低潛在的資安風險，以目前常見的應用程式安全檢測，在程式開發的階段可以採用不同的作法，包括原始碼檢測、沙箱測試以及逆向工程。以加強應用程式在開發期間的安全，例如：使用存在資安漏洞的開發套件、編譯工具等，這些都可能因此而影響到應用程式本身的安全問題，另外在政府資訊服務委外的作業中，也不斷的要求應用程式的安全檢測作業，以確定委外開發的應用程式避免已知的資安風險，也降低未來因為應用程式的安全問題，所帶來的資安風險。

13.5.1 原始碼檢測

原始碼檢測（Source Code Review），對於所開發的應用程式碼進行測試，發掘可能潛在的資安風險，例如：運算邏輯的問題、引用存在資安風險的函式庫等，或是開發人員在應用程式開發撰寫階段，為方便應用程式的除錯或其他的目的，所留存未來不需使用到的程式片斷等，這些問題都有機會透過原始碼檢測的作業，發現存在的問題。

應用程式原始碼檢測，主要包括了以下的內容：

- **已知漏洞掃描**：目前每天被揭露的應用程式弱點相當多的，對於已被揭露的漏洞進行測試，避免已知的問題仍然存在現有的應用程式中，造成資安風險的產生。
- **檢測工具掃描**：透過自動化的工具，可以協助程式開發人員減輕原始程式碼的檢測工作，由分析的報告找到可能潛在的問題。
- **資料保護處理**：對於應用程式所處理的資料，進行資料安全的測試，目前許多的應用程式都會配合檔案、資料庫等方式，進行資料的收集與後續的處理。
- **存權控制與權限檢查**：對於應用程式的存取，以及資料在不同的程式間進行交換時，都需要考量存取權限的管制，以避免未經授權的使用者存取到不屬於權限範圍內的資料。
- **運作環境的搭配**：進行程式碼的檢測時，有時候會參考到未來規劃運作的環境，是否能夠與應用程式本身搭配，也需要考量未來部署後的管理機制。

原始碼檢測是屬於應用程式安全的其中一環，並非通過原始碼檢測就沒有資安的風險，仍然需要考量應用程式與運作環境的關係，以及配合風險管控的機制，能否有效的保護應用程式本身與運作的環境遭到攻擊者惡意入侵的可能性。

13.5.2 沙箱測試

沙箱測試（Sandbox Test）是一種經常應用於資訊安全領域的測試方式，透過所設計出來一個受到管控的環境，再進行應用程式的行為分析，整個測試過程並不會對於實際的環境造成影響，而且大多數的沙箱測試，會採用一個完全隔離的環境，但會參考實際環境的特徵、參數、組成進行設計，因此能夠在一個可受到控制的環境中，對於想要測試的項目進行功能測試、行為測試以及安全面向的檢測作業，並透過測試後的結果進行應用程式的調校。整體而言，沙箱測試對於應用程式的運作而言，可以提供一個有效的安全測試方式，透過風險的評估找出潛在的問題，並降低對於真實環境所帶來的影響。

以下是沙箱測試環境的特點：

- **隔離的環境**：透過隔離的環境進行受測目標的檢測，以確定不會受到外來的干擾與影響。
- **擬真的環境**：考量未來需要部署到真實的環境，一般而言會在沙箱中採用接真實的環境參考，例如：未來即會使用的系統環境、資料庫系統或是相關的網路防護架構等。
- **完整的安全測試**：透過自動化工具或是人工的方式進行測試，並不需要考量是否會影響到真實的環境，因此能夠不受限制的進行各種測試項目。
- **整體的安全評估**：沙箱可以依據安全評估的面向進行環境的設計，能夠涵蓋完整的評估項目，或是參考現有的資安標準進行評估。

13.5.3 逆向工程

逆向工程（Reverse Engineering）屬於進階的資安檢測技術，對於應用程式進行分析，並且觀察與追蹤應用程式執行的過程，以發現應用程式的執行流程、運作的結構以及所提供的功能等項目，進行逆向工程的分析，多數無法取得完整的原始程式碼，而是已經編譯後的執行檔或是高階的應用程式的形式，因此需要對於應用程式在環境中執行的邏輯、演算法以及處理資料的結構有深入的瞭解，才能掌握應用程式是否有潛存的資料風險。

一般而言，進行逆向工程需要處理以下幾個重要的項目：

- **應用程式的運作**：利用分析工具或是拆解編譯後的程式，以掌握應用程式的運作架構，包括了程式本身的結構、執行邏輯、資料的處理方式等，有助於由其中找到可能的資安風險。
- **漏洞與弱點的分析**：發掘潛在的應用程式安全漏洞，分析程式碼以確定是否在可被攻擊者運作的已知漏洞，或是程式執行過程可能產生的弱點。
- **破解保護的機制**：目前大多數正式發佈的應用程式都會採行保護的措施，因此在進行逆向工程的作業中，可能會需要對於應用程式所使用的保護措施進要破解，但此技術也被應用於非法的侵權、再製等不當的使用上。
- **解析智財權**：解析應用程式所使用的技術，或是檢測是否有公司及個人的智慧財產權問題，這部分也經常與開放原始碼等自由軟體的授權方式進行合規的分析。
- **跨平台的運作環境**：在無法取得應用程式原始程式碼的情況下，透過逆向工程的技術，有機會實現將現有的應用程式轉移到其他的系統環境，或是作業平台執行的可能性。

13.6 營運環境的安全檢測

除了應用程式本身在開發過程需要注意的資安風險外，當進行程式的部署以及營運環境的調校，將會進入正式運作與開始提供應用服務的階段，此階段除了留意原本的應用程式碼的安全性外，最重要的是營運環境可能對於應用程式本身帶來的資安風險，需要透過環境上的安全檢測機制進行持續的關注，一般而言，多數的企業或組織，對於營運環境的安全檢測會定期進行，以確保營運的資安風險，可以藉由定期的進行技術面的檢測，發掘可能的資安風險再加以防護。

目前應用程式的營運環境，多數透過網路以及網站應用服務的方式來提供，因此應用程式部署到營運環境後，上線提供服務前將會進行部署的資訊安全檢測作業，包括了弱點掃描、滲透測試等基本的資安檢測作業，如果對於資安風險的管控想要透過擬模駭客攻擊的方向進行驗證，也可以採用紅隊演練的方式進行，而所有的資安檢測作業都是為了讓應用服務在營運過程，能夠持續的掌握資訊安全風險，並且持續不斷的進行修補與降低可能因為資安風險所帶來的資訊外洩或是營運中斷的風險。

安全檢測的過程，會對於受測的系統進行存取權限的確認，以確保使用者對於應用程式本身的存取管控機制，防止未經授權的存取，由應用程式所提供的操作介面，進行資料庫安全與資料處理機制的風險評估，而部署應用程式到營運環境後，往往需要對於營運的環境，配合應用程式的需要進行調校，以符合營運所需的安全管理，如果有組態配置不當等問題，可能在此階段進行修正，另外在營運階段應用程式提供使用者操作的過程中，應用程式與系統環境的日誌紀錄，對於當下狀況的掌握，或是日後對於事件的追蹤都是相當重要的資料來源，因此在營運的環境中需要考量如何有效的建立日誌紀錄的收集與分析，以及對於可能對於營運環境帶來的資安風險進行掌控。最後，在營運階段的資安風險處理，也包括了對於資安事件發生時的應處，以及資安事件發生後的應變，對於營運環境的復原或是資料救援，都需要一套受到實證的作業流程，或是業務持續營運計畫、災難復原計畫等，這些都是有備而無患，且每隔一段時間或是有新型態的資安威脅出現時，就需要同步的對於應變計畫進行調整，以確定仍然可以確保企業與組織能夠持續營運。

13.6.1 弱點掃描

弱點掃描（Vulnerability Scanning）是資安技術檢測中常見的測試方法，一般會透過自動化的工具軟體，對於受測的目標依據弱點特徵資料庫中的項目進行驗測，以發掘存在的安全弱點與漏洞，主要用來識別可能受到攻擊者運用的漏洞，透過掃描工具的分析，進行風險的評分以及提供改善的建議與對應的處理措施。

一般而言，弱點掃描作業包括以下幾個主要的項目：

- **識別目標**：對於受測目標進行識別，以分辨出目標的屬性，例如：網路設備、伺服器、作業系統、應用程式環境等資訊。
- **系統分析**：進行系統環境的探測，以識別所使用的作業系統、開放的網路服務、應用程式的版本等資訊。
- **弱點探測**：針對已知的漏洞與安全弱點進行測試，透過掃描作業確認受測目標是否存在相同的問題，收集弱點的資訊、系統環境或是組態上的問題，也可能在探測階段發現其他的應用程式存取點。
- **風險評估**：依據掃描後的結果，進行弱點風險的評估做為修補的參考，而評估的結果可能受到網路架構的影響，例如：在外網測試的結果，可能與在內網測試的結果不同。

- **分析報告與建議**：整體資安風險分析報告，依據發掘的弱點提供改善的建議，例如：升級應用程式或是函式庫的版本，或是在資安防護的設備上加上特定的資安規則等作法。
- **定期執行與差異比對**：透過定期進行弱點掃描作業，並且分析歷次的結果，可以做為企業與組織長期掌握受測目標可能遭受的資安風險，透過掃描結果的差異比對，掌握應用程式與系統環境的變化。

弱點掃描作業，可以透過掃描後的結果，掌握受測目標可能存在的資安風險，提供企業與組織做為參考，減少目標遭受攻擊時可能發生的資安問題，也可以降低營運資料外洩的可能性。

13.6.2 滲透測試

滲透測試（Penetration Testing）是一種模擬攻擊的測試手法，可用來評估受測目標的系統、應用程式以及網路架構中的安全性，透過模擬攻擊者的行為，可以協助發現可能的資安風險，一般而言滲透測試的作業會與弱點掃描的作業協同進行，在完成弱點掃描的作業後，視需求與情況評估是否進行滲透測試的作業，相對於弱點掃描作業，滲透測試將會更完整的對於受測目標進行評估，例如：發掘的資安弱點可被利用的程度，另外滲透測試的作業有較完整的流程，我們主要可以分成以下幾個階段進行：

- **準備階段**：擬定與規劃測試範圍、目標、時程等計畫，確定後再執行相關的檢測作業，並且在此階段將會盡量掌握應用程式的功能、系統的營運架構、網路防禦的機制等。
- **收集階段**：進行主動與被動的資訊收集，例如：系統曾經發生過的漏洞、所採用的資訊技術等。
- **分析階段**：透過自動化工具或是手動的進行分析，評估系統上可能存在的問題，例如：弱密碼、未更新的軟體版本、組態設備的問題等。
- **測試與攻擊階段**：進行實際的模擬攻擊，識別在資訊收集與分析階段評估可能存在的資安風險是否可被運用，進一步的測試該漏洞所帶來的影響範圍評估。
- **報告撰寫階段**：依據檢測的結果進行報告的整理與撰寫，其中需要包括測試的項目、結果以及風險的等級，並且提供發現的漏洞、成功進行攻擊的路徑、帶來的潛在風險等問題，再依據資安風險提供改善的建議措施。

目前許多的資安法規，皆會要求需要進行資安檢測作業，實際的進行檢測或是模擬攻擊，是現行發掘潛在資安風險的有效方法，另外在進行滲透測試時，需要留意合法性、風險管控、通報以及留下關鍵紀錄的事項，考量進行的過程可能會對於營運中的系統帶來影響，因此必須清楚的通知受測範圍相關的人員，並且留存佐證的紀錄，以詳細的記載整個檢測的過程，並且避免因為進行滲透測試造成無法營運的現象。

13.6.3 紅隊演練

紅隊演練（Red Team Exercise）與滲透測試都是資安檢測中的主要方法，都是透過模擬攻擊者的手法進行資安的檢測，不過紅隊演練比起滲透測試，更強調以下的重點：

- 紅隊演練主要模擬攻擊者可能的攻擊行為，評估整個企業與組織的安全性，涵蓋了應用服務、系統平台、網路架構、資料安全等面向，在演練的過程也可以配合事件應變能力的測試。
- 範圍擴大不限特定目標，只要是企業與組織相關的目標與人員，都會是演練的範圍，屬於全面性的資安檢測。
- 採用全面性實戰的原則，包括多維度的攻擊、社交工程的運用、資安防護機制的驗證或繞過，發掘所有可能造成資安風險的原因。
- 檢測團隊須具備攻擊者思維，才能夠找出非正規或預設情況的資安風險。
- 演練報告需詳實記錄檢測軌跡，以進行實際環境的比對。
- 無預警的攻擊測試，多數的紅隊演練並不會預先通知參與的人員或組織實施的日期，以貼近真實的資安威脅測試，同時可以驗證事件應變的能力。
- 綜合風險的評估，不局限於特定的系統與目標，以整體企業或組織的風險為評估。