It’s All About Virus!

資訊二　b95902086　陳冠妤

財金二　b95701213　孫采萱

動機

基於多方考量之下，在給定的報告主題方向中，我們決定選擇電腦病毒做爲研究對象，概述電腦病毒的起源與種類，並以組合語言的方式探討病毒的基本架構——其如何自體複製與感染其他程式。希望能藉此機會運用上課所學及各種參考資料的內容，在關於電腦病毒的這一個領域能有更深入的了解。

病毒簡介

電腦病毒是具有感染其他磁碟(或磁片)及自體複製性質之程式，其撰寫的目的多數是爲了破壞及惡作劇，其中大部分病毒是否能發揮作用，則取決於其感染他人電腦的方式。而電腦病毒本身，也可以有很多種分類方法，如(1)開機型及檔案型；(2)常駐型及非常駐型；(3)編碼型及非編碼型等等……

而電腦病毒之起源，在一九四九年電腦的先驅者約翰.范紐曼(John Von Neumann)在他所提出的一篇論文 [複雜自動裝置的理論及組織的進行]裏,即已把病毒程式的藍圖勾勒出來，直到十年之後，在美國電話電報公司(AT&T) 的貝爾(Bell)實驗室中，這些概念在三個程式人員道格拉斯麥耀萊(H, Douglas McIlroy)，維特・維索斯基(Victor Vysottsky)以及羅伯・莫里斯(Robert T. Morris)——所想出的電子遊戲「磁蕊大戰」(core war)中成形。

磁蕊大戰的玩法以雙方各寫一套程式，並令其程式在電腦記憶系統內互相追殺。有時它們會放下一些關卡，有時會停下來修理(重新寫)被對方破壞的幾行指令；當它被困時，也可以把自己複製一次，逃離險境，因爲它們都在電腦的記憶磁蕊中遊走，因此得到了磁蕊大戰之名。當時電腦間互相獨立沒有連線，若某部電腦受到感染，只需把它關閉即可。但是當電腦連線逐漸成爲社會結構的一部份之後，一個或自我複製的病毒程式便很可能帶來無窮的禍害了。故當時這些電腦工作者選擇不公開其程式內容，直至一九八三年科恩・湯普遜(Ken Thompson)在授獎典禮上公開其存在。而於一九八五年杜特尼討論磁蕊大戰的文章中，第一次提出了「病毒」一詞——義大利的羅勃吐・歇魯帝(Roberto Cerruti)和馬高・麼魯顧帝(Marco Morocutti)所提出讓病毒先感染磁碟，而以電腦爲媒介，藉此使病毒在磁碟間互相傳染。

開機型病毒

所謂開機型的病毒(Boot-type virus) 是界定爲在電腦開機時，搶先作業系統進入記憶體的程式。正常我們由軟碟開機的程序如下

開電源	→	POST	→	BIOS	→	IO.SYS	→	MSDOS.SYS	→	SHELL
		程序		載入		載入		載入		程式

由於病毒必須取得磁碟讀寫的控制權（這樣才能達成感染的目的），因此開機型病毒本身會存在於開機磁區(Boot Area) ，以便在載入 OS 時會先 OS 載入以取得絕對控制權。因此感染（中毒）後開機的程序變成了下面這樣：

開電源 → POST → BIOS → IO.SYS → MSDOS.SYS → SHELL
程序 載入 ↑ 載入 載入 程式
※病毒載入※

我們看到了病毒在 DOS 載入前載入，這樣便可以利用讀寫磁片的機會（如 dir 指令）進行感染。而硬式磁碟的感染，就是比軟碟多了一項硬碟分割表的檢查程序，而開機型病毒便可藏身於開機磁區或是硬碟分割表中，多了一種可能。而一般開機型病毒可分爲兩種：傳統開機型病毒及隱形開機型病毒。

檔案型病毒

所謂檔案型的病毒(File-type virus) 是介定爲在檔案執行時，在原檔案之前執行的程式。病毒本體寄居於可執行檔案中，當此檔案被執行時，便侵入作業系統取得絕對控制權；當然也有不常駐而僅在執行時感染其它檔案的病毒。檔案型病毒主要是感染 .COM 檔或是 .EXE 檔。近年來，檔案型病毒的感染目標又擴及到 .OVL 檔和.BIN 檔。

而病毒要如何去取得控制權呢？大體而言病毒都是朝 BIOS 呼叫及 DOS 呼叫兩方面著手，以取得中斷進入點。方式則千奇百怪，如早期的正常方式(Int21h & 25h & 35h)，中期的單步中斷（MacGyver 1.0) 及之後流行的字串比對法（ MacGyver4.0 & T4-Virion)(但此法對 BIOS 會有不相容的情形。)

通常正常的中斷呼叫程序爲：

中斷產生 → DOS 處理 → BIOS 處理 → 硬體 I/O

而當病毒試圖去入侵記憶體時，它可能會有兩種侵入的方式：

中斷產生 → DOS 處理 → BIOS 處理 → 硬體 I/O
↑ ↑
※病毒攔截※ ※病毒攔截※

其中(1)的方式就是取得 DOS 的進入點（當然還有分末端進入點還是原始進入點），而(2)的方式就是取得 BIOS 的原始進入點。當病毒侵入記憶體後，便是和開機型病毒相同，藉由磁碟的動作來達到複製的目的。

常駐型病毒

病毒依佔用主記憶體方式可分爲兩種:

1.非常駐性病毒（TRANSIENT）

非常駐性程式執行時被 DOS 載入主記憶體中，但執行完畢則在主記憶體之（該）程式碼將被 DOS 刪除。大部分程式都屬於非常駐性，如此可減少佔用主記憶體之空間，如 DISKCOPY.COM。

對於非常駐性病毒而言，它隨載體程式被放入 RAM 時機執行，因此必須在程式結束之前進行干擾，如 VIENA(維也納）、CARTIER....等屬於非常駐性病毒。

2.常駐性病毒

所謂常駐（Terminate and stay resident，即 TSR），乃是當一程式執行後回 DOS，該程式仍駐留在主記憶體中，換言之它會佔用主記憶體空間；一般常見之常駐性程式如中文系統、病毒偵測軟體、DOS 核心程式。正常之常駐性式如中文系統，其常駐目的是希望隨時爲使用者服務，譬如說在文書軟體下可使用中文輸入法輸入中文字（中文輸入法並非文書軟體所提供之功能）。而常駐性病毒之目的亦希望常駐 RAM 中，隨時窺視系統之活動以進行干擾、傳染或破壞。大部分之病毒都屬於常駐性病毒，如黑色星期五、紅色九月、石頭病毒・・・等。『常駐型病毒』當您在執行到被感染病毒程式的時候，這個帶毒的程式會將它自己常駐在記憶體之中，等到下一個程式要執行的時候，躲藏在記憶體中的電腦病毒便會去感染目前所要執行的程式。

對於這一類型的電腦病毒會去攔截中斷向量來加以感染，相對的此時用 CHKDSK 去檢查記憶體會少了幾 k。既然病毒會去攔截中斷向量，所以病毒程式只要利用 INT 21h 功能就可以去修改檔案的屬性。換句話來說，不管您的檔案是否有設定成唯讀的屬性，病毒都有辦法修改成可讀寫的屬性，然後再來感染檔案。

病毒與組語

當我們寫出一個病毒後，對於這個病毒還可以做幾種加強，以增進這個病毒的可用性。

一、多形

多形現象，所指的就是病毒在每次感染時都會改變自己的形態，讓它保持原有的功能但看起來卻完全不同。這個目的可以利用結果相同但不同的指令來達成。爲了達到多形的目的，但是有因爲要在解碼之前執行，並不可以將其編碼，所以代換的方法，在程式中 b_replace 到 e_replace 中這段，就是在感染別的檔案時，會被置換掉的。以下有四種用來置換的程式碼範例：

```
way1:
    mov ax,offset set
    sub  ax,bp
    neg  ax
    nop
    xchg bp,ax
way2:
```

```
        mov bx,offset set
        sub  bx,bp
        neg  bx
        xchg bx,bp
way3:
        mov cx,offset set
        sub  cx,bp
        neg  cx
        xchg bp,cx
way4:
        mov dx,offset set
        sub  dx,bp
        neg  dx
        xchg dx,bp
```

或加入些垃圾指令。因爲所用的暫存器只有固定幾個，所以剩下的暫存器填什麼並不會影響病毒程式。例：

```
 mov            cx,10
 mov            di,1234              ;trash
 and       ax,[di+1234]              ;trash
 cld                                 ;trash
 mov            si,encoded_data
 test      [di+1234],bi              ;trash
 or        al,cl                     ;trash
 main_loop:    add  di,di            ;trash
 xor       ax,1234              ;trash
 xor       byte ptr [si],55
 sub       di,123               ;trash
 inc       si
 test      dx,1234              ;trash
 and       al,[bp+1234]         ;trash
 dec       cx
 nop
 xor       ax,dx                ;trash
 sbb       ax,[di+1234]         ;trash
 and       cx,cx                ;trash
 jnz       main_loop
```

二、避免重複感染

以感染.com 型的病毒爲例，當病毒感染檔案後，你會發現了檔案一直在長大，如此很容易被人發現，因爲磁碟空間會一直縮小，所以避免重覆感染是很重要的。要如何避免呢？最簡單的方法就是在被感染過的檔案內作記號，下次要感染之前先檢查記號是否存在，如果是就不感染，反之則感染之。

病毒實做

選擇最基礎的感染.com 檔做起，因爲.com 檔的結構比.exe 簡單，因此這算是入門基本題。另外，在這隻病毒中加入上述避免重複感染的檢查機制，以增加他的功用性。

感想

藉由這次的報告，我們更深入的了解到了病毒的型態與其傳染途徑，以及傳染過程中與軟體、硬體間的交互作用關係，並以組合語言之型態深入探討至病毒在記憶體內運用及隱藏的方式。也同時能正視資訊安全之議題，希望未來除了研發更專業的防毒工具外，資訊工作者們也能以正確的態度及道德觀來運用此技術。組合語言所寫的病毒主要活躍於 DOS 時代，然而現在因爲網路的發達、各種程式語言的發展，使得撰寫病毒不再是很困難的事情。包括三、四年前的 Bagel 程式碼公開事件等，只要我們越來越依靠資訊、電腦，相信寫病毒的人就會越來越多。因此，瞭解了病毒的基本運作原理後，更該擁有同等的防毒知識與概念，才能應付這個資訊時代。

參考來源

http://www.csie.ntu.edu.tw/~cyy/courses/assembly/07fall/news/
http://www.csie.ntu.edu.tw/~wcchen/asm98/