基於知識本體之惡意程式分析平台

*黃獻德　　**李健興　　***莊宗嚴　　*蔡一郎　　*邱敏乘
*財團法人國家實驗研究院國家高速網路與計算中心
**國立臺南大學資訊工程學系
***國立臺南大學數位學習科技學系
TonTon@nchc.org.tw　leecs@mail.nutn.edu.tw　yilang@nchc.org.tw　mcqiu@nchc.org.tw　chuangyen@mail.nutn.edu.tw

摘要

近年來，資訊技術快速發展，網路進入無遠弗屆的時代，個人電腦的運算能力日漸增強，帶來了方便和快速；但木馬、病毒及後門程式等威脅亦相繼衍生；以目前資訊安全防護系統所產出的相關數值報告而言，仍需透過專業人員才可得知所代表的意義，但不同的環境相同的資料值亦代表不同意義；為了避免時間、效力上的浪費，本論文預計透過知識本體（Ontology）提供一個完善的模型來表述，並將資訊安全之惡意程式知識整合模糊標記語言（Fuzzy Makup Language）建置惡意程式知識本體、惡意程式分析平台；期許未來，可以即時提供符合人類思考模式語義，以即時採取適當的處理措施。

關鍵字：惡意程式，行為分析，知識本體

一、前言

隨著網路無遠弗屆時代的來臨，很多安全問題也相繼衍生而出；根據美國喬治亞理工學院資訊安全中心(Georgia Tech Information Security Center, GTISC)所公佈的 2009 年新興網路威脅報告，最具威脅及挑戰的五個網路安全課題分別是惡意程式、殭屍網路、網路戰爭、對網路電話及行動裝置之威脅[1]；而針對上述威脅做進一步的瞭解，並建置相關知識庫，將是防範的最佳辦法之一。

本論文架構如下：第二節探討惡意程式分析以及知識本體相關文獻，第三節說明惡意程式分析平台架構，第四節說明惡意程式知識本體，第五節為結論。

二、文獻探討

惡意程式所衍生出來的安全問題會造成機關或企業的損失，即使有安裝防毒軟體也不例外，因防毒軟體通常是 24 小時內更新一次病毒碼，但是，絕大多數受到惡意程式所感染且破壞通常是在更新的空窗期，加上惡意程式的數量以接近指數在成長；因此，針對惡意程式分析將是不容忽視的議題。

(一) 惡意程式分析

目前最常見惡意程式分析有兩種方法，分別是針對惡意程式碼進行分析的程式碼分析（Code Analysis）；以及主要分析惡意程式對系統、網路等的影響運行結果之行為分析（Behavior Analysis），其中靜態與動態分析是最基本也是最重要的惡意程式分析方法，茲簡述如下[2]：

- 靜態分析
  - 以檔案特徵比對
  - 以黑白名單作比對
  - 以已知特徵對檔案進行分析
  - 以已知規則對檔案進行評估
- 動態分析
  - 監控程式運行，捕捉惡意行為
  - 與正常環境做異動比較

雖然惡意程式分析技術進步快速，但是，反偵測惡意程式分析的技術也快速的

成長，不論是採用靜態分析或是動態分析所採用的虛擬機器（VMware）、沙箱測試（Sandbox）等方法，惡意程式皆有已知的反偵測方法，如：

- Anti-Static Analysis
  - Packed Code
  - Encoded String
  - Anti-Behavior
- Monitor
  - 偵測分析環境
  - 攻擊分析程式
  - 穿越還原系統

其中，動態分析所採用的行為偵測分析，更是近年來防毒業者研發的主力；但是，針對惡意程式行為偵測分析，惡意程式也有可以避開相關監控的技術，如：

- 不呼叫Win32API改呼叫更核心的Native API
- 操作已知未被監控的動作，如RegLoadKey
- Restore Hooking
- 將行為拆開在不同Process間執行

除此之外，根據相關研究，惡意程式為了避開監控，亦可以將其本身常駐於記憶體中，待重新開機後再正式執行其惡意破壞行為[3]；因此，透過Sandbox等相關工具來進行惡意程式分析，取得相關惡意程式行為之結果，其正確性已降低；加上惡意程式產生工具已非常普遍，惡意程式產生以及變種的速度，已遠遠大於病毒碼的更新速度，想要透過分析所得的結果，來加以判斷其行為是否屬於惡意程式，實屬不容易，其結果也充滿不確定性和處於模糊地帶，這也正是市面上之防毒軟體為何常常有誤判現象發生的箇中原因。

正因為惡意程式閃避偵測或監控的技術越來先進，更需使用相關分析工具針對惡意程式執行時監控和比對環境異動，如：Advanced Intrusion Detection Environment（AIDE）做檔案完整性檢查，檢查檔案是否被修改、Regdiff做系統登錄檔值是否遭篡改、Tcpdump與Ngrep監控網路介面之封包所有行為；以及透過記憶體傾印的方法，將其暫存在記憶體或是硬碟快取檔中的資料與原始乾淨的資料做進一步的分析[4]。

針對惡意程式的行為做監控以及分析也許不是最好的方法，但透過瞭解它的行為，以及根據其行為來分類建立相關的惡意程式行為知識庫，將會是目前最有效的方法[5]；然而每天有變化多端的各種惡意程式不斷的在產生，單純建置已知的惡意程式行為知識庫，並無法達到偵測未知的相關行為是否與惡意程式行為有關聯性，且容易有語意以及綱要不同的問題；為了解決這樣的問題，Semantic Web中的Ontology即是用語意來表達物件之間的關係以及屬性，並透過現有的知識規則進行語意推論而得到新知識；因此，使用Ontology來建置網路攻擊與惡意程式知識庫，將會是支援產學研界資安技術發展其中最好的解決方法。

(二) 知識本體（Ontology）

Ontology中文稱「本體論」、「存在論」、「實體論」或「知識本體」，是西元前250年由哲學家亞里斯多德所提出；近年來，被廣為應用在資訊科學界的人工智慧研究領域，也延伸到例如：自然語言處裡、生物醫學資訊系統、電子商務、智慧型代理人、專家系統、地理資訊系統以及軟體工程...領域，其最主要的功能便是「正規化的表達知識」；知識被定義為基於個人任何相關及可行動的經驗[6]，所有的知識工作者分享某些特有的活動。而資料可以依靠架構或知識本體(Ontology)被註釋[7]，其中知識本體在分享知識及表現訊息及語意方面，就是一個非常理想的架構[8]；知識本體也可以將真實世界領域的概念轉變為由實體、特性、關係及定理組合而成人類可認知及機器可讀的格式[9]。

目前，已經有許多學者專家針對Ontology定義出其架構，例如，Lee 等學

者提出應用在新聞摘要的模糊知識本體[10]，以及一個基於知識本體的智慧型決策支持代理人，並將其應用在能力成熟度整合模式(Capability Maturity Model Integration, CMMI)之專案監控領域上[11]；並且開發出各種應用，例如三層式立體本體架構應用[11-13]；而三層式立體架構中，由上而下分別為概念層、關係層及實例層，三層式立體架構主要是加強關係的敘述，以往注重的是概念、屬性、操作概念、實例、物件等，但彼此之間皆有關係值存在，強調關係的存在不單可以將彼此連結起來，更可以知道彼此的關係為何，或是在知識本體中的操作流程。

圖 1 即為三層式立體領域知識本體的結構圖，其中"$C_1, C_2, ..., C_m$"定義為概念，每個概念在這個概念層中被命名成 C_i 並擁有一組屬性$\{A_{C_i1}, A_{C_i2}, ..., A_{C_ip}\}$，一組關聯性的關係$\{R_{C_iC_1}, R_{C_iC_2}, ..., R_{C_iC_m}\}$，以及一組實例關係$\{R_{I_1C_i}, R_{I_2C_i}, ..., R_{I_nC_i}\}$。每一個屬性有他們的名字以及屬性值。在關係層中定義了每項關係表示了領域知識本體中的內部關係，舉例而言，C_1 與 C_4 之間的 association 關係用 $R_{C_1C_4}$ 表示。概念與實例之間的關係則是"instance-of"，相同的，每一個實例在實例層中會有一個名字 I_i，實例 I_1 及概念 C_1 之間的 *Instance-of* 關係可利用 $R_{I_1C_1}$ 表示。

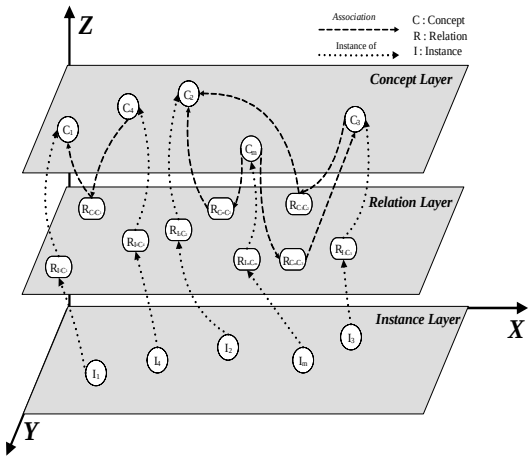

圖 1、Structure of the domain ontology[11-13]

(三) 模糊標記語言(Fuzzy Markup Language，FML)

模糊標記語言(Fuzzy Markup Language，FML)由 Acampora 和 Loia 等學者提出[14-16]，以 XML 為基礎並結合模糊邏輯所定義出來的語言；FML 是用來將人類語意上的模糊轉化為標記式語言，由模糊知識庫、模糊規則庫、推論引擎，模糊系統和解模糊系統所組成。模糊知識庫是根據領域專家的知識所建立的變數，推論引擎是利用模糊知識庫和模糊規則庫進行模糊推論運算，模糊系統和解模糊系統是扮演著控制系統和控制系統之間的橋樑，並且可以透過 XSLT 將其結合為 Java 可以執行的語言；圖 2 為 FML 流程，圖 3 則為 FML 架構。

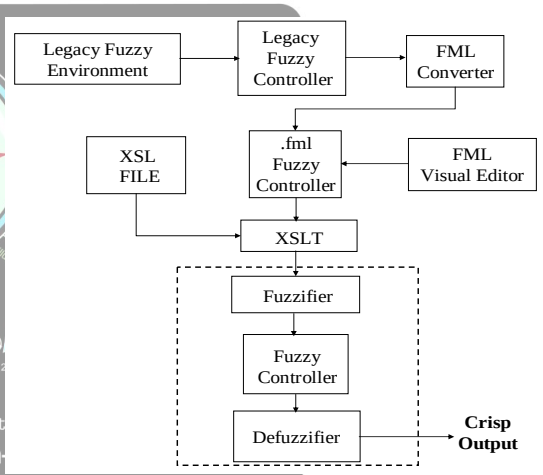

圖 2、FML 流程[14-16]

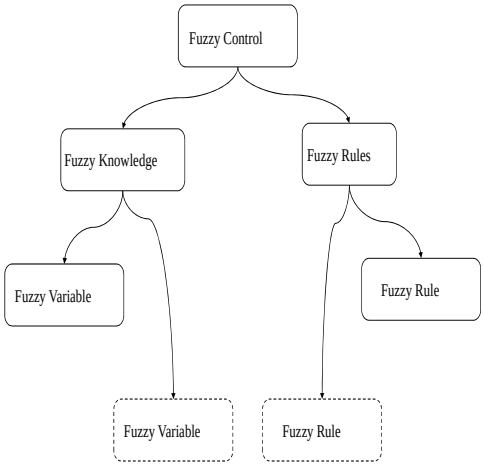

圖 3、FML 架構[14-16]

表 1、FML 的 Fuzzy Term

```
<FUZZYVARIABLE
domainleft = ”aR1 ” domainright = ”cRl ”
ip = ” localhost ” name = ” output ”
scale = ” undef ined ” t y p e = ”OUTPUT”>
<FUZZYTERM name=”R1”>
<TRIANGULARSHAPE
param1 = ”aR1 ”
param2 = ”bR1 ”>
param3 = ”cR1 ”>
</TRIANGULARSHAPE>
</FUZZYTERM>
<FUZZYTERM name=”R2”>
<TRIANGULARSHAPE
param1 = ”aR2 ”
param2 = ”bR2 ”>
param3 = ”cR2 ”>
</TRIANGULARSHAPE>
</FUZZYTERM>
</FUZZYVARIABLE>
```

表 2、FML 的 Fuzzy Rules

```
<RULEBASE>
inferenceengine = ”MINMAXMINMAMDANI”
ip = ”localhost ”
<RULE
connector = ”AND”
ip = ” localhost ”
weight = ”1”>
<ANTECEDENT>
<CLAUSE not = ”FALSE”>
<VARIABLE> input1 </VARIABLE>
<TERM> d1 </TERM>
</CLAUSE>
<CLAUSE not = ”FALSE”>
<VARIABLE> input2 </VARIABLE>
<TERM> d2 </TERM>
</CLAUSE>
<CLAUSE not = ”FALSE”>
<VARIABLE> inputm </VARIABLE>
<TERM> dm </TERM>
</CLAUSE>
</ANTECEDENT>
<CONSEQUENT>
</CONSEQUENT>
</RULE>
</RULEBASE>
```

FML 中根據 Fuzzy Logic 以及 Fuzzy Logic Control 等理論，分別定義了＜FUZZYVARIBLE＞用來表示 Fuzzy Concept，＜FUZZYTERM＞用來表示 Linguistic Term 以表達 Fuzzy Concep，如表 1 所示；FML 中也分別定義了<RULE>表示一條 RULE 以及其連接詞，<ANTECEDENT>用來表示其 RULE 的先行條件，<CONSEQUENT>則用來表示因先行條件所產生的結果，<CLAUSEA>、<CLAUSEC>則表示 RULE 中的每個選項，<VARIABLE>以及<TERM>則分別是用來表示 RULE 中的變數及其值，如表 2 所示。

在資訊安全的領域中，惡意程式的偵測，傳統上是利用資料庫的技術來儲存資料，但是卻無法表達資料間的關係，並且資料格式的不同是難以共享的，資料庫對於問題求解的資料特定且缺乏靈活性，傳統的網路惡意程式資料庫能夠比對、偵測出可能的網路惡意程式，但是在任何有用的意義下都不知道未知網路惡意程式的行為概念。

三、惡意程式分析平台

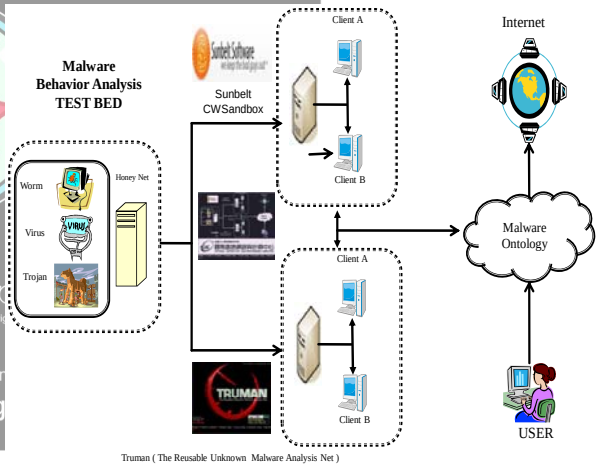

圖 4、惡意程式分析平台架構圖

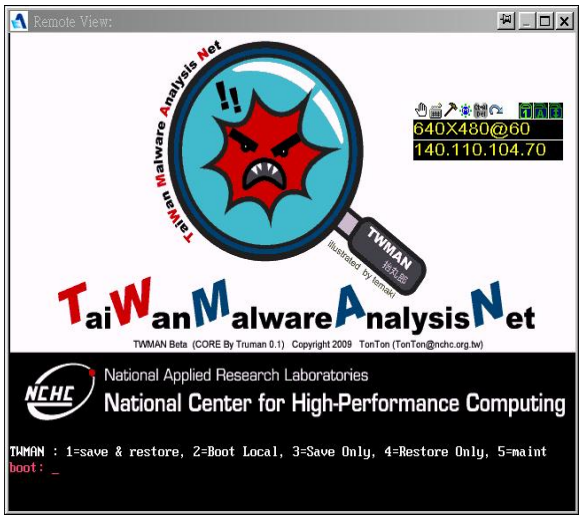

圖 5、惡意程式分析平台開機畫面

圖 4 為惡意程式分析平台之架構，圖 5 為惡意程式分析平台的開機畫面，從圖 4 架構中可得知本論文採用 CWSandBox 進行監控惡意程式之行為[17]；以及透過 Stewart Joe 所開發的 The Reusable Unknown Malware Analysis Net（TRUMAN）[18, 19]以及 Jim Clausing 開發的 Automated Behavioral Analysis Environment[20]所使用之開放源碼工具，進行與正常系統之環境做比對；並將兩者加以整合，以便提升分析所得結果的正確性，並整理所產出的分析報告，修改成欲建置本體所需要的格式。

(一) CWSandBox 分析端

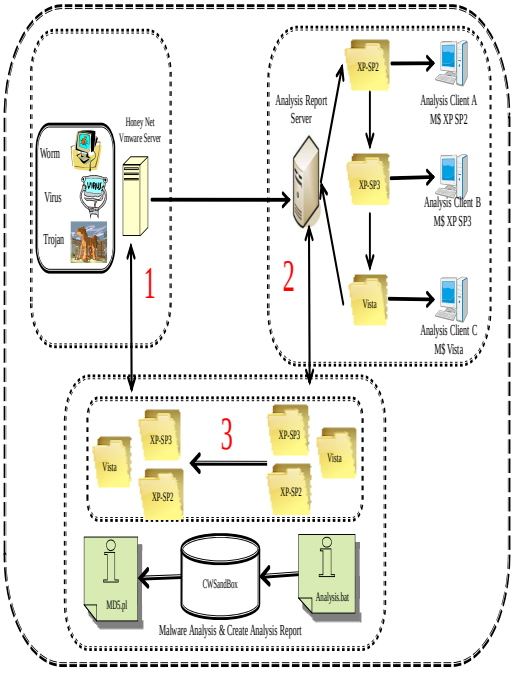

圖 6、CWSandbox 分析流程

圖 6 即為 CWSandBox 進行惡意程式分析運行流程，其中區塊 1 為前端透過 HoneyPot 等惡意程式樣本搜集工具，區塊 2 則是分析平臺會自動將惡意程式下載至 Client 端 PC，並使用 CWSandBox 以及完整的乾淨系統執行惡意程式樣本並進行分析。圖 7 則是圖 4 中區塊 2 當系統進行分析過程時的畫面。

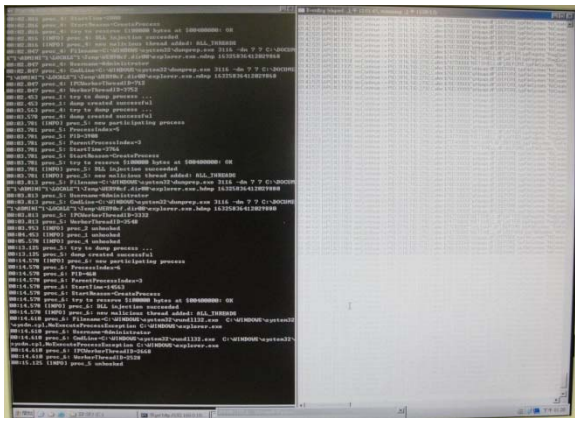

圖 7、CWSandBox 執行圖[17]

圖 8 則是圖 6 中區塊 3 是分析完惡意程式樣本後產出 XML 格式報告，圖 9 則是 XML 格式報告透過 XSLT 將其格式化成 Web 型式，以便進行網頁呈現與 Ontology 相關後續應用推論。

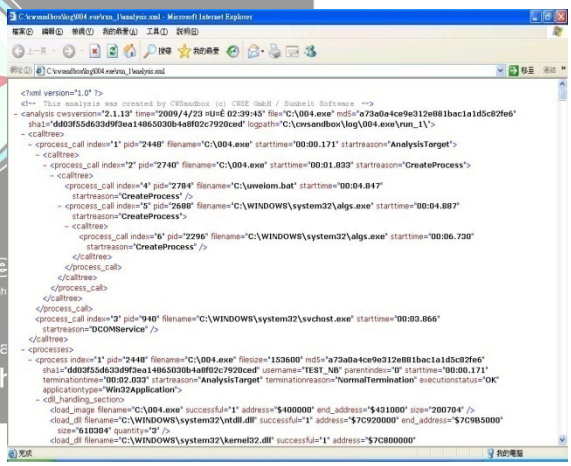

圖 8、XML 格式報告[17]

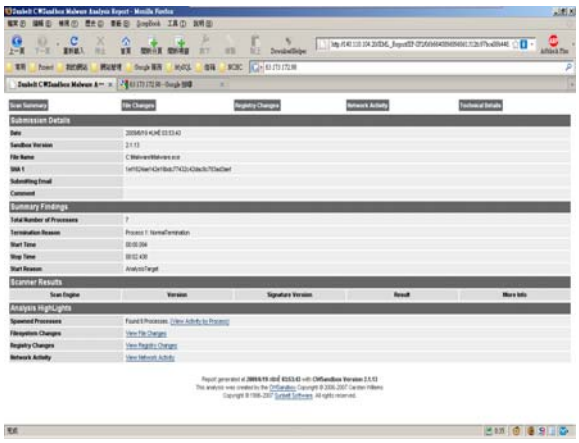

圖 9、XSLT 格式化後報告[17]

當 Client 端產出惡意程式樣本分析報告上傳 Server 端進行處理後，透過由國家高速網路與計算中心自由軟體實驗室所開發的再生龍（Clonezilla），進行還原功能，它使用 partimage 還原乾淨的映像檔至系統上，以便繼續進行下一個分析。表 3 為 CWSandBox 在分析完惡意程式樣本的行為後所產出的 XML 格式報告，從該報告範例中可以看出，CWSandBox 會針對惡意程式對系統所做的 dll handling、filesystem、mutex、registry、process 以及 stored created files 等行為做記錄；表 4 則是整個 XML 格式報告中關於 stored created files 的記錄；從表 4 中可以觀察出該惡意程式會自行在 C 槽建立 ab3.bat 以及 index.dat 檔，並且會將該建立出來的檔案之各相關資訊如檔案大小以及存取路徑保存起來[17]。

表 3、惡意程式行為分析 XML 範例[17]

```
<process>
<dll_handling_section>
<filesystem_section>
<mutex_section>
<registry_section>
<process_section>
<system_section>
<system_info_section>
<stored_created_files_section>
<process>
```

表 4、惡意程式行為分析 XML 範例[17]

```
<stored_created_file srcfile="C:\Documents and
Settings\Administrator\Local Settings\Temporary Internet
Files\Content.IE5\index.dat"
dstfile="e65b2507b7bee965fc1a2dcc0637adf9.dat"
filesize="32768" />
 <stored_created_file srcfile="C:\Documents and
Settings\Administrator\Cookies\index.dat"
dstfile="d7a950fefd60dbaa01df2d85fefb3862.dat"
filesize="16384" />
 <stored_created_file srcfile="C:\Documents and
Settings\Administrator\Local
Settings\History\History.IE5\index.dat"
dstfile="c6012e7e33fc7f7b6ff418a0e164467b.dat"
filesize="32768" />
</stored_created_files_section>
```

(二) Truman 分析端

圖 10 為 Truman 端的分析流程，茲簡述如下：

1. 乾淨 Client 端 建立系統映象檔
2. Client 端 PC 下載惡意程式樣本
3. Client 端 PC 運行 10～15 分鐘
4. Client 端 PC 立系統感染後映象檔
5. 與乾淨的系統進行比對產出報告

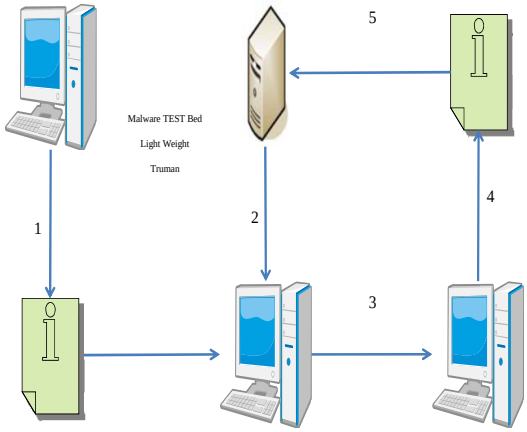

圖 10、Truman 分析流程

從圖 10 可以得知 Truman 透過 PXE-BOOT 開機，使用 dd 儲存感染執行惡意程式後的系統映像檔，以及還原乾淨的映像檔至系統上，以便繼續進行下一個分析。圖 11 則為 Truman 的 Client 端下載惡意程式進行分析時的畫面，當執行惡意程式後為了使其對系統開始做相關惡意破壞行為，Truman 使用了 Windows Server 2003 Resource Kit Tools 中的 Sleep 功能，它可以使系統在執行時設定呈現欲靜止多久時間的狀態；另外，Truman 並設計了 Fauxservers 提供了 IRC、SMTP、HTTP、FTP 等相關虛擬 Services，並且將其相關封包資訊截取下來。

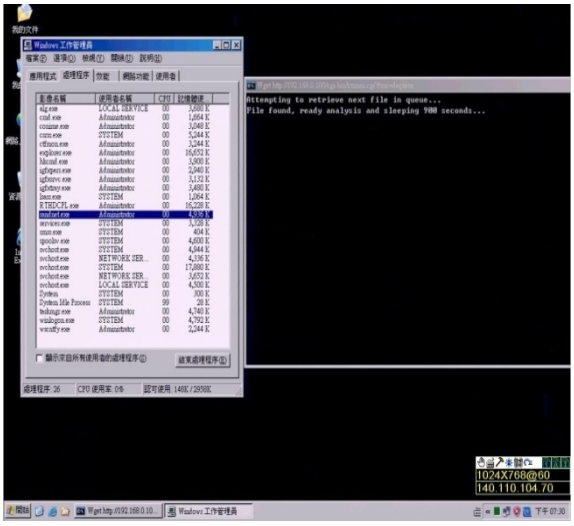

圖 11、Truman 分析畫面[18, 19]

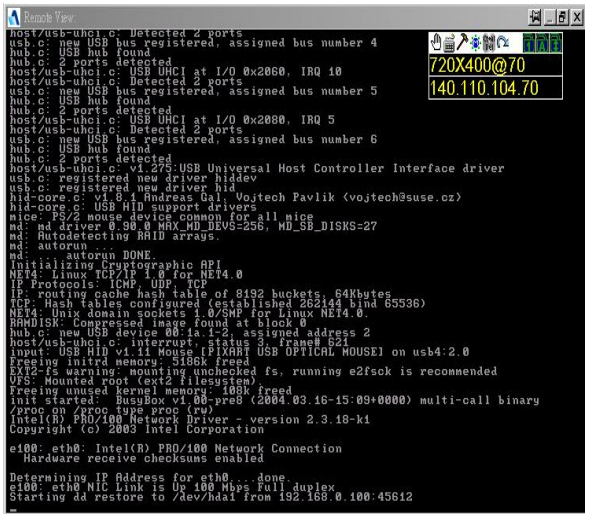

圖 14、還原系統

圖 12 則為 Truman 的所設計之虛擬網路 Service 的執行畫面，其所截取到的網路封包行為 log 檔，茲整理如表 5 所示。

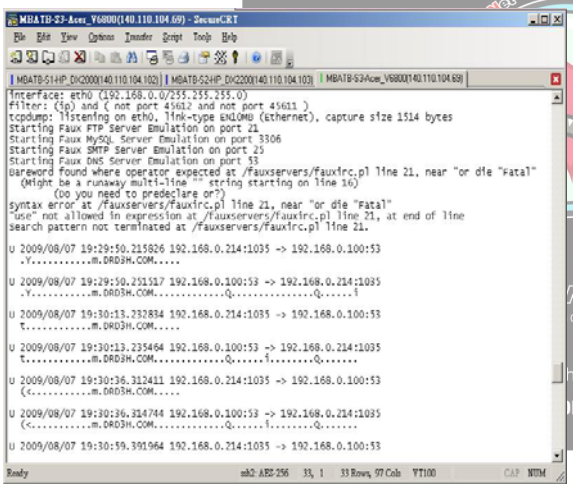

圖 12、Truman 截取網路封包畫面[18, 19]

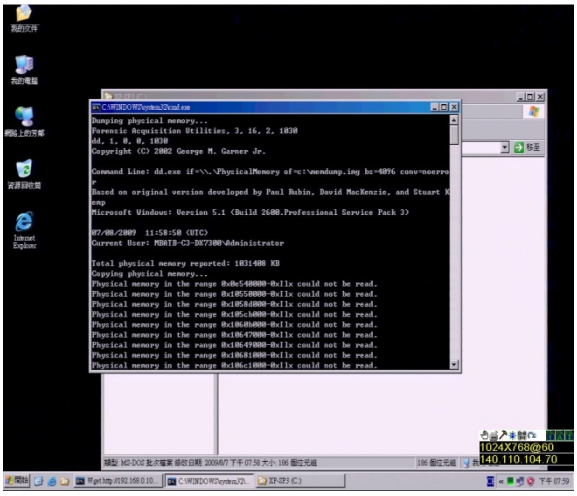

圖 13、Dump 記憶體

圖 13 為 Client 端執行惡意程式後，為了深入分析惡意程式是否將其相關行為暫存在記憶體或是硬碟暫存檔，而將記憶體內之資料 Dump 成一映像檔；圖 14 則是 Truman 將乾淨的系統映像檔還原至系統以便進行下一個分析；最後，當感染過惡意程式之映像檔已回存至 Server 端時，便可以進行相關分析程序，並產出其相關 Report，如表 6 所示。

表 5、虛擬服務所產生之範例 log 檔[18, 19]

```
Starting Faux IRC Server Emulation on port 6667
Starting Faux FTP Server Emulation on port 21
Starting Faux SMTP Server Emulation on port 25
tcpdump: listening on eth0, link-type EN10MB (Ethernet),
capture size 1514 bytes
Starting Faux DNS Server Emulation on port 53U
2009/08/07 23:36:08.901798 192.168.0.214:138 ->
 192.168.0.255:138
U 2009/08/07 23:36:40.450763 192.168.0.214:138 ->
 192.168.0.255:138
DHDDDADAAA. ABACFPFPENFDECFCEPFHFDEFFP
U 2009/08/25 22:02:07.735116 192.168.0.100:38406 ->
U 2009/08/25 22:02:07.735519 192.168.0.109:57100 ->
192.168.0.100:38406
U 2009/08/25 22:04:56.584138 192.168.0.110:137 ->
192.168.0.255:137
FHEPFCELEHFCEPFFFACACACACACACABL.. ..
```

表 6、Truman 所產生之 Report 範例[18, 19]

Summary report for
cb35f063c0b96110f75051ea0827a56e created at
9 日 8 月 07:39:28 CST 2009
Host file changes>>>
Registry Run Key changes>>>
Persistence|C:\WINDOWS\system32\igfxpers.exe
PHIME2002A|C:\WINDOWS\system32\IME\TINTLGNT
\TINTSETP.EXE /IMEName
IMJPMIG8.1|"C:\WINDOWS\IME\imjp8_1\IMJPMIG.EXE"
/Spoil /RemAdvDef /Migration32
Registry Service Key changes>>>
-SwPrv|MS Software Shadow Copy Provider|
C:\WINDOWS\system32\dllhost.exe
/Processid:{DB99892F-6778-4E3E-AB71-15D040B76A0A}
|Own_Process|Manual|
+SwPrv|MS Software Shadow Copy Provider
|HTTP>>>
IP traffic>>>
06:51:18.835960 arp reply 192.168.0.100 is-at
00:16:17:e4:e2:13
1460,nop,nop,sackOK>
06:51:18.836382 IP 192.168.0.214.1035 > 192.168.0.100.80: .
ack 1 win 65535
AIDE>>>
Start timestamp: 2009-08-09 07:33:06
Total number of files: 11218
Added files: 35
Removed files: 7
Added files:
added: /WINDOWS/system32/dllcache/hidusb.sys
added: /WINDOWS/system32/dllcache/mouhid.sys
added: /WINDOWS/system32/drivers/hidusb.sys
added: /mnt/images/WINDOWS/system32/drivers/mouhid.sys
Removed files:
removed: /WINDOWS/pchealth/helpctr/BATCH/hscsp_w3.cab
removed: /WINDOWS/pchealth/helpctr/System/DFS/privacy.htm
removed:/WINDOWS/pchealth/helpctr/System/DFS
/upldrvinfo.htm
removed:/WINDOWS/pchealth/helpctr/System/DFS/xmldialog.htm
removed:/WINDOWS/pchealth/helpctr/System/DFS/xmldisplay.xsl

四、惡意程式知識本體

根據分析平台所產出的 XML 報告，依照相關惡意程式之行為，如：修改 Registry、網路 Connect、修改 Files 等，再透過三層式立體知識本體架構來繪製其概念圖，如圖 15 所示；以及使用 Stanford 大學所開發的 Protégé 來建置知識本體庫，以供未來進行後續語義推論，如圖 16 所示。

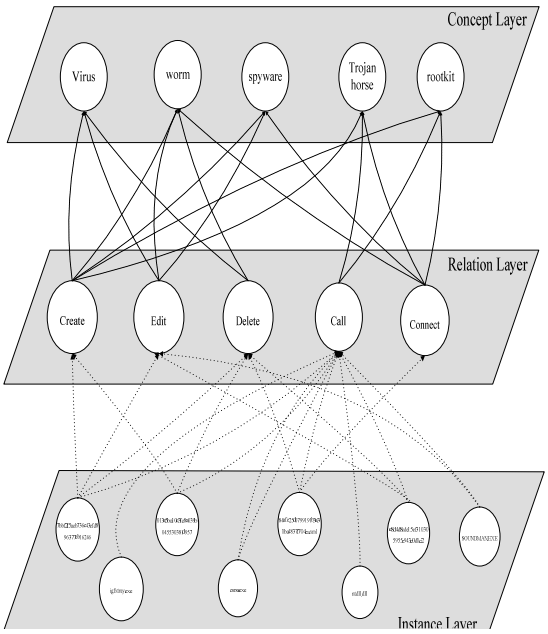

圖 15、惡意程式行為三層式立體知識本體

圖 16、惡意程式行為知識本體

除此之外，為了降低知識庫語義推論的誤判率，將所建置的本體知識庫，根據其相關行為結合 FML，依據惡意程式所修改的 Files、Registry 以及 Connect 建置出相關 Fuzzy Term、Fuzzy Varible 以及 Fuzzy Rules 等知識庫，如表 7 所示；透過所建置之 Ontology 以及 FML，來進行惡意程式行為語義推論，將可減少其誤判之機率。

表 7、FML 範例

```
<!DOCTYPE FUZZYCONTROL SYSTEM ”
fml.dtd ”>
<FUZZYCONTROL defuzzifymethod
= ”CENTROID”
ip = ” localhost ” type = ”MAMDANI”>
<KNOWLEDGEBASE IP = ” localhost ”>
<FUZZYVARIABLE
domainleft = ”0” doinright = ”1”
ip = ” localhost ” name = ”Luminosity”
scale = ”Lux” type = ”INPUT”>
<FUZZYTERM name=”low”>
<PISHAPE
param1 = ” 0.0 ”
param2 = ” 0.45 ”>
</PISHAPE>
</FUZZYTERM>
<FUZZYTERM name=” medium”>
<PISHAPE
param1 = ” 0.49999999999999994 ”
param2 = ” 0.44999999999999996 ”>
</PISHAPE>
</FUZZYTERM>
<FUZZYTERM name=”HIGH”>
<PISHAPE
param1 = ” 0.5501 ”
param2 = ”1”>
</PISHAPE>
</FUZZYTERM>
</FUZZYVARIABLE>
</KNOWLEDGEBASE>
<RULEBASE
inferenceengine = ”MINMAXMINMAMDANI”
ip = ” localhost ”>
<RULE connector = ”AND” ip = ” localhost ”
weight = ”1”>
<ANTECEDENT>
<CLAUSE not = ”FALSE”>
<VARIABLE>Connect</VARIABLE>
<TERM>38.49.185.12.445</TERM>
</CLAUSE>
<CLAUSE not = ”FALSE”>
<VARIABLE>Registry Changes</VARIABLE>
<TERM>PHIME2002ASync</TERM>
</CLAUSE>
</ANTECEDENT>
<CONSEQUENT>
<CLAUSE not = ”FALSE”>
<VARIABLE>dimmer</VARIABLE>
<TERM>medium</TERM>
</CLAUSE>
</CONSEQUENT>
</RULE>
</RULEBASE>
</FUZZYCONTROL>
```


五、結論

以人工智慧的角度而言，知識和推理同樣扮演著至為重要的角色，基於知識的代理人能夠將知識和當前的感知結合起來，從而在選擇行動之前推導出當前狀態的隱藏部份；每天有變化多端的各種惡意程式不斷在產生，未知的相關行為是否與惡意程式行為有關聯性並無法偵測，針對惡意程式的行為做監控與分析也許不是最好的方法，但使用 Ontology 來建置網路攻擊與惡意程式知識庫，而採用 FML（Fuzzy Makeup Language）[11-13]將可表達其語意上的模糊，除了解決傳統上知識表達的正規化，更可降低誤判的機率。如果可以從資安專家中學習更多的知識及推論出惡意程式可能的行為，那網路惡意程式偵測將可獲得革命性的進展。

參考文獻

[1] G. T. I. S. Center, "Emerging Cyber Threats Report for 2009," October 15 2008.

[2] S. Software, "CWSandbox User Guide v 2.1.13," 2007.

[3] B. Dolan-Gavitt, "Forensic analysis of the Windows registry in memory," *Digital Investigation*, vol. 5, pp. S26-S32, September, 2008.

[4] R. B. v. Baar, W. Alink, and A. R. v. Ballegooij, "Forensic memory analysis: Files mapped in memory," *Digital Investigation*, vol. 5, pp. S52-S57, September 2008.

[5] T. Holz, C. Willems, K. Rieck, P. Düssel, and P. Laskov, "Learning and Classification of Malware Behavior," in *Fifth Conference*

on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 08), 2008.

[6] B. Ghosh and J. E. Scott, "Comparing knowledge management in health-care and technical support organizations," *IEEE Transactions on Information Technology in Biomedicine*, vol. 9, pp. 162-168, 2005.

[7] P. Warren, "Knowledge management and the semantic web: from scenario to technology," *IEEE Intelligent Systems*, vol. 21, pp. 53-59, 2006.

[8] M. Reformat and C. Ly, "Ontological approach to development of computing with words based systems," *International Journal of Approximate Reasoning*, vol. 50, pp. 72-94, 2009.

[9] Q. T. Tho, S. C. Hu, A. C. M. Fong, and T. H. Cao, "Automatic fuzzy ontology generation for semantic web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, pp. 842-856, 2006.

[10] C. S. Lee, Z. W. Jian, and L. K. Huang, "A fuzzy ontology and its application to news summarization," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 35, pp. 859-880, 2005.

[11] C.S. Lee, M.H. Wang, and J.J. Chen, "Ontology-based Intelligent Decision Support Agent for CMMI Project Monitoring and Control," *International Journal of Approximate Reasoning*, vol. 48, pp. 62-76, 2008.

[12] C.S. Lee, M.H. Wang, Z.R. Yan, C.F. Lo, H.H. Chuang, and Y.C. Lin, "Intelligent estimation agent based on CMMI ontology for project planning," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC 2008)* Singapore, 2008.

[13] C.S. Lee, M.H. Wang, W.C. Sun, and Y.C. Chang, "Intelligent healthcare agent for food recommendation at Tainan City," in *Systems, Man and Cybernetics, 2008* Singapore, 2008, pp. 1465-1470.

[14] Giovanni Acampora and Vincenzo Loia, "A Proposal of an Open Ubiquitous Fuzzy Computing System for Ambient Intelligence," *Computational Intelligence for Agent-based Systems*, vol. 72, pp. 1-27, 2007.

[15] Giovanni Acampora and Vincenzo Loia, "Fuzzy Control Interoperability and Scalability for Adaptive Domotic Framework," *IEEE Trans. Industrial Informatics*, vol. 1, pp. 97-111, May 2005.

[16] Giovanni Acampora and Vincenzo Loia, "Using FML and Fuzzy Technology in Ambient Intelligent Environments," *International Journal of Computational Intelligence Research*, vol. 1, pp. 171-182, 2005.

[17] C. Willems, T. Holz, and F. Freiling, "Toward Automated Dynamic Malware Analysis Using CWSandbox," *IEEE Security and Privacy*, vol. 5, pp. 32-39, March 2007.

[18] Stewart Joe, "TRUMAN - The Reusable Unknown Malware Analysis Net (Version 0.1)," 2005.

[19] Stewart Joe, "Behavioural malware analysis using Sandnets," *Computer Fraud & Security*, vol. 2006, pp. 4-6, December, 2006 2006.

[20] J. Clausing and C. Hornat, "Building an Automated Behavioral Malware Analysis Environment using Open Source Software," in SANS Institute Reading Room, 2009.