簡潔化惡意軟體行為分析

陳嘉玫、王則堯、蔡閎亘
中山大學資訊管理學系

摘要

惡意軟體對電腦使用者產生了極大的安全威脅及不便，除了影響電腦的作業效能外，更直接或間接的造成個人及組織，蒙受難以估計的時間及財務損失。實務上，為了因應惡意軟體的快速增加，自動化分析的平台已成為系統管理人員不可或缺的管理工具。使用者藉由自動化分析系統的協助，可以迅速的執行決策及行動，但由系統產生的分析報告，經常流於細節及繁雜，往往造成使用者額外的負擔及災難。

有鑒於此，本研究提出一種可應用於實務上的惡意軟體自動化分析平台，可以準確的識別惡意軟體外，也提供一份精簡具指導性的摘要式報告，輔助系統管理人員的判斷及決策。我們採取一種獨特的綜合式策略，設計一套惡意軟體分析系統，名為 EDAM (Efficient Dynamic Analyzer for Malware)，紀錄並分析惡意軟體對作業系統之檔案、程序與登錄檔的修改，及網路連線情況。最後由系統提出一份精簡具指導性的摘要式分析報告，提供系統管理人員運用。

關鍵詞：惡意軟體分析、動態分析、模糊化技術

壹、 緒論

惡意軟體是人們對於下列各種令人困擾的軟體的一種泛稱，諸如電腦病毒、蠕蟲、木馬、間諜程式等皆是。惡意軟體對電腦使用者產生了極大的安全威脅及不便，除了造成個人經濟損失的風險，也因其破壞性間接影響電腦的作業效能，讓使用者蒙受難以估計的時間及財物損失。根據資訊安全科技公司 G Data Software(G Data Malware Report ,2011)的調查報告指出，惡意軟體迅速的發展，以 2011 年上半年，就有超過一百二十萬種新的惡意軟體出現，和前一年同期相比，成長幅度超過百分之二十(詳見圖 1)。

為了降低惡意軟體的潛在威脅，對於惡意軟體行為分析的研究益顯重要。在惡意軟體行為分析的相關研究中，主要可區分為靜態分析方法(Christodorescu 2003)與動態分析方法(Bayer 2006)；靜態分析方法針對惡意軟體程式碼進行分析，而動態分析方法則針對惡意軟體的行為進行觀察、識別。靜態分析固然準確，但主要挑戰在於，模糊化技術(如 obfuscation code、Dead-code Insertion、Code Transposition、Subroutine Reordering、Code Integration 等)及惡意程式碼難以取得，經常導致分析的困難與失效。

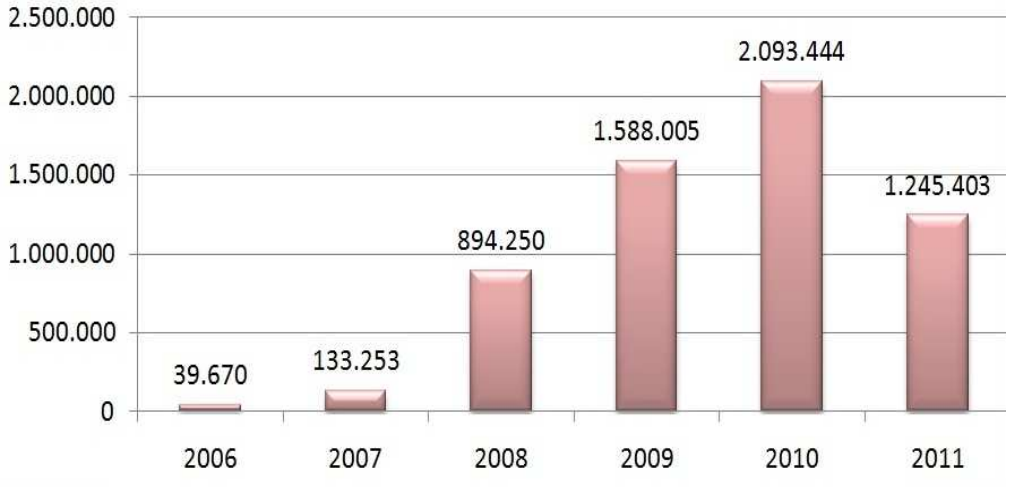

圖 1：2006 至 2011(上半年)惡意軟體統計

動態分析避開靜態分析的弱點，採取直接由惡意軟體的行為分析著手的策略；在惡意軟體的執行期間，藉由各式的分析工具，觀察並記錄下惡意軟體的各項行動，諸如其執行程序、檔案增修、網路連線及登錄檔增修等。但動態分析方法，會產生龐大的分析資料，雖然能清楚且詳細的的了解惡意軟體產生的各項細節，但容易衍生面對龐大資料，卻難以處理的困境。對系統管理人員而言，如何快速準確的得到可靠的分析結果，將是面對日漸快速增加的惡意軟體威脅的一大挑戰。所以若能提出一種快速、準確的惡意軟體自動化分析工具，將可以更有效的，降低惡意軟體可能產生的經濟損失及潛在威脅。

有鑒於此，本研究提出一種可應用於實務上的惡意軟體自動化分析方法，輔助系統管理人員，快速準確的識別惡意軟體，提出有效的對應策略與行動，提高系統的安全係數。本研究中，採取一種獨特的綜合式策略，設計名為 EDAM (Efficient Dynamic Analyzer for Malware)的惡意軟體分析系統，結合 Capture-BAT 工具，紀錄惡意軟體對作業系統之檔案、程序與登錄檔的修改，並以 Tshark 紀錄惡意軟體的網路連線情況。最後，由系統提出一份精簡具指導性的摘要式分析報告，提供系統管理人員運用。

貳、 文獻探討

惡意軟體分析研究，可區分為靜態及動態分析方法兩種。靜態分析是一種白箱 (White Box) 的分析方式，這種分析方式是透過分析惡意軟體的程式碼，進而了解這個惡意軟體的功能。

在 Christodorescu 等人(2003)的研究中，使用靜態分析的方法，試圖解決惡意軟體因運用模糊化技術(obfuscation)而使防毒軟體難以偵測的問題。該研究列出四種主要的模糊化技術，比較和分析經過這些技術處理之後的程式碼，最後建立一套名為 SAFE (static analyzer for executables)的系統，能夠偵測出使用，插入無用程式碼(Dead-code Insertion)、程式碼換位(Code Transposition)、重新指派暫存器(Register Reassignment)、指令替換(Instruction Substitution)，這四種模糊化技術的惡意軟體。

但模糊化技術並非該研究提到的四種而已。因此，惡意軟體若使用其它的模糊化技術，如副程式重新排序(Subroutine Reordering)、程式碼集成(Code Integration)等。未來若出現新的模糊化技術，使用靜態分析系統，將出現無法快速應對的難題。

動態分析是一種黑箱 (Black Box) 的分析方式，這種分析方法是將惡意軟體放置在一個虛擬的環境當中執行，避免惡意軟體破壞真實系統的運作。目前的惡意軟體動態分析研究，其作法主要是將惡意軟體置於一個封閉的作業環境中，將其執行或觸發行為發生，再藉由觀察行為特徵並加以分類辨識。所以此類研究的步驟，首先必須建立一個安全可靠的分析環境。許多研究中，分析環境的建立，常見的有虛擬機器(Virtual Machine, VM)、沙盒系統(Sandbox) (Norman 2006，Willems 2007)兩種。

虛擬機器是安裝在真實系統中的一個模擬系統，其優點是可以模擬出與真實系統相同的環境功能，惡意軟體可以在其中執行運作，卻不會對真實系統造成損害。此外虛擬機器在執行完惡意軟體後，可以輕易、快速地回復到執行前的初始狀態，有利於進行另一次的測試。目前較知名的軟體有 VMware、Virtual PC 及 VirtualBox 等。

執行惡意軟體動態分析，主要監控的行為有：惡意軟體是否更動系統檔案、登錄檔修改、網路的通訊狀態，及惡意軟體執行後，系統環境中程序的變化等。將惡意軟體的行為紀錄後，再對行為紀錄結果進行分析。過去的研究中，使用系統監控工具，如 Capture-BAT (Seiferta 2007)來監控系統的檔案、程序及登錄檔是否遭修改或寫入等，而網路連線狀態，則使用 Wireshark 或類似的網路封包監控軟體，來記錄惡意軟體的網路

活動。雖然上述工具均能有效的執行系統或網路情況監控，但惡意軟體行為複雜，非單一工具監控就能完全掌握其面貌。

參、 系統設計

本研究提出的惡意軟體分析系統名為 EDAM (Efficient Dynamic Analyzer for Malware)，系統架構如圖 2。EDAM 分為二個子系統：系統行為監控、網路連線監控。

系統行為監控是為了要紀錄惡意軟體在執行時對作業系統所做的更動，系統行為包括檔案、程序、登錄檔三個項目。因為惡意軟體要靠網路連線來進行傳達命令、下載檔案、攻擊或是傳送竊取資訊等動作，所以必須要監控網路連線。另外為了解決各子系統所使用工具所產生之原始記錄內容過多的問題，本系統會將這些原始記錄檔先經過聚合或擷取所需資訊，再匯入資料庫中以產生較精簡的摘要報告。

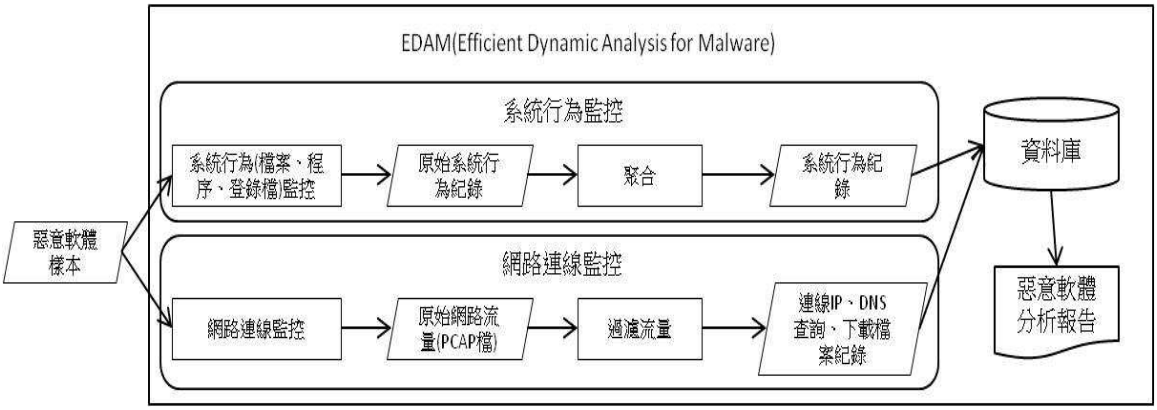

圖 2：EDAM 系統架構

以下將分別對 EDAM 的兩個子系統作詳細說明。

1 系統行為監控

惡意軟體在執行期間會存取或修改作業系統中的檔案、程序、登錄檔；同樣的，作業系統的正常執行期間，也有相同的行為。所以我們將可疑行為與作業系統的正常行為共同紀錄下來。由於原始記錄檔的筆數較多，須先將記錄檔進行處理後再匯入資料庫中。

圖 4 是存取系統檔案行為記錄的處理流程，先將一段時間內連續存取相同檔案的記錄聚合成一筆記錄，並紀錄該行為的起訖時間及存取次數。接著判斷該筆記錄之行為是否有建立新的檔案，若有，則比對該新產生檔案與惡意軟體執行檔的 MD5 值是否相同，依此可找出惡意軟體複製自己至系統其它目錄的行為。

另外，由於原始的系統監控記錄檔可能會在一段時間內紀錄到多次重複的動作。因此本系統在將原始記錄檔匯入資料庫前，會將連續相同的記錄合併為一筆以減少重覆資料，並加入該行為的起迄時間與重複次數。

存取系統程序行為記錄的處理流程，同樣先將一段時間內連續存取相同程序的記錄聚合成一筆記錄，並紀錄該行為的起訖時間及存取次數。因為惡意軟體經常會偽裝成一般正常系統程序的名稱，所以當記錄檔中的程序名稱為正常系統程序名稱時，必須再比對其程序路徑是否和正常的系統程序路徑吻合，若否，則說明該程序很可能是由惡意軟體所偽裝。

經過上述步驟處理過的行為記錄，除了減少重複資料的產生外，另增加了原始記錄檔所沒有提供的行為資訊，如複製檔案、偽裝正常系統程序、登錄檔的功能描述等。

l　網路連線監控

EDAM 除了紀錄惡意軟體對作業系統的存取行為之外，惡意軟體對外的網路流量監控也非常重要，是必須要紀錄的項目。不論是何種惡意軟體，幾乎都必須要靠網路連線來進行傳達命令、下載檔案、攻擊或是傳送竊取資訊等動作。由於短時間的網路流量擷取就可能包含大量的封包資訊，未經過濾的封包資訊將造成判讀上的困難。因此本研究將原始網路流量中，過濾掉與惡意軟體無關的封包資訊，僅保留包括惡意軟體和那些 IP 進行連線，透過 DNS 請求哪些網站，並下載哪些檔案等相關資訊。

本研究參考當前的線上分析系統有關網路流量的連線紀錄方式，惡意軟體的對外連線，多採取 DNS (port 53)及 HTTP (port 80)的連線，因此由原始流量檔中過濾出這兩種類型的流量。經過 EDAM 過濾後的原始網路流量，僅保留較重要的連線 IP、DNS 查詢、下載檔案三種記錄，可省去解讀大量封包的時間與人力，又能得到惡意軟體在網路連線方面的關鍵行為。

肆、　實驗結果與分析

系統驗證 EDAM 分析結果的正確性，針對四類知名的惡意軟體 Zeus、SpyEye、Nugache、Koobface 進行分析，將分析結果與第三方機構的分析報告進行比對。本階段，我們選擇廣為人知的線上分析平台 CWSandbox，及 Malbed 與 EDAM 的分析結果進行比對。為了確認所取得的樣本，是否為正確的惡意軟體樣本，先將樣本傳送至 VirusTotal 網站進行分析，取得該網站對樣本的分析命名，區分並確認該樣本屬於上述四種惡意軟體。以下是分別對四種惡意軟體的分析，並將 EDAM 分析結果與其他系統的比較：

（一）　Zeus 分析結果

Zeus 是一種後門木馬程式根據 Binsalleeh 等人(2010)所做的研究，Zeus 有三個行為特徵：1.將 Zeus 執行檔複製至系統其它目錄、2.修改 Windows 登錄檔，將 Zeus 執行檔加入開機自動載入清單、3.Injection 程序 winlogon.exe。針對 Zeus 的實驗結果，如下表 1。

表 1：Zeus 分析結果比較

環境 / 行為	EDAM	CWSandbox	Malbed
將 Zeus 執行檔複製至系統其它目錄	O	X	X
修改 Windows 登錄檔，將 Zeus 執行檔加入開機自動載入清單	O	X	O
Injection 程序 winlogon.exe	O	X	X

(註：表中使用符號說明，”O”表示有正確紀錄到該行為、”X”表示未紀錄到該行為。)

Zeus 於 EDAM 進行分析的結果時發現，Zeus 會將其執行檔複製至 C:\WINDOWS\system32\rdpclip.exe ，而 rdpclip.exe 會修改登錄檔 HKCU\Software\Microsoft\Windows\CurrentVersion\Run，來將 Zeus 執行檔加入開機自動載入清單中。同一樣本於 CWSandbox 中進行分析時，樣本並沒有成功執行，因此 CWSandbox 並未紀錄到任何行為。而 Malbed 也只有紀錄到 Zeus 在登錄檔部分的行為特徵。

（二） SpyEye 分析結果

SpyEye 是一種後門木馬程式，根據 Eads(2010)所做的研究 SpyEye 有三個行為特徵：1. 將 SpyEye 執行檔複製至系統其它目錄、2. 修改 Windows 登錄檔，將 SpyEye 執行檔加入開機自動載入清單、3. Injection 程序 explorer.exe。針對 SpyEye 的實驗結果，如下表 2。

表 2：SpyEye 分析結果比較

環境 / 行為	EDAM	CWSandbox	Malbed
將 SpyEye 執行檔複製至系統其它目錄	O	X	—
修改 Windows 登錄檔，將 SpyEye 執行檔加入開機自動載入清單	O	O	—
Injection 程序 explorer.exe	O	X	—

(註：表中使用符號說明，”O”表示有正確紀錄到該行為、”X”表示未紀錄到該行為、”—”表示惡意軟體樣本未正確執行，沒有產生分析報告。)

EDAM 紀錄到 SpyEye 在執行後，利用程序 explorer.exe 將 SpyEye 的執行檔複製到 C:\kereo83.bin\C71313DF12B.exe ，並修改登錄檔 HKCU\Software\Microsoft\Windows\CurrentVersion\Run，將 SpyEye 加入開機自動載入清單。而對執行 SpyEye 前後所建立的記憶體映像檔進行比對後可以發現，explorer.exe 這個系統程序所載入的 DLL 檔，比執行 SpyEye 之前多出九個。

於本系統的記錄中還可發現，SpyEye 建立兩個檔案 C:\usxxxxxxxx.exe\config.bin 和 C:\usxxxxxxxx.exe\usxxxxxxxx.exe，並在之後由 explorer.exe 建立程序 usxxxxxxxx.exe。由以上行為可得知 SpyEye 確實利用 explorer.exe 這個系統程序來進行惡意行為。

CWSandbox 僅紀錄 SpyEye 建立 C:\kereo83.bin\這個目錄，但並沒有複製檔案至該目錄，也沒有建立任何檔案。關於程序紀錄部份，CWSandbox 僅提供 dwwin.exe 及 services.exe 另外兩個程序的行為記錄，但並未紀錄到 explorer.exe 這個程序的行為。而 Malbed 在分析 SpyEye 的過程出現錯誤訊息，未成功進行分析。

（三） Nugache 分析結果

Nugache 是一種 P2P Bot，根據 Stover 等人(2007)所提出的 Nugache 分析報告，Nugache 有二個行為特徵：1. 於 C:\WINDOWS\system32\目錄下建立 mstc.exe、mvwatvx.exe、wmipvs.exe 三個檔案其中之一、2. 修改 Windows 登錄檔，將 peer list 加入。針對 Nugache 的實驗結果如表 3。

表 3：Nugache 分析結果比較

環境 / 行為	EDAM	CWSandbox	Malbed
於 C:\WINDOWS\system32\目錄下建立 mstc.exe、mvwatvx.exe、wmipvs.exe 三個檔案其中之一	O	—	X
修改 Windows 登錄檔，將 peer list 加入	O	—	X

(註：表中使用符號說明，”O”表示有正確紀錄到該行為、”X”表示未紀錄到該行為、”—”表示惡意軟體樣本未正確執行，沒有產生分析報告。)

本系統的記錄中發現，Nugache 於系統目錄 C:\WINDOWS\system32\下新增 mstc.exe 這個檔案。mstc.exe 並不屬於 Windows 原本的系統程序，將檔案存放在 C:\WINDOWS\system32\目錄下是為了降低被發現的可能性。而在 Nugache 原始的執行檔建立了 mstc.exe 這個程序後，mstc.exe 程序於 HKCU\Software\GNU\Data 下新增了一百多筆 IP 的鍵值。HKCU\Software\GNU\Data 下所儲存的這些 IP 是 P2P 的 peer list。圖 9 為 Nugache 執行期間的部分流量，和 HKCU\Software\GNU\Data 下所儲存的 IP 也能夠互相對應，說明 Nugache 確實會和這些 P2P peer 進行連線。

（四） Koobface 分析結果

Koobface 是一種以社交網站 facebook 用戶為目標的病毒，感染目的為收集有用的個人資料。根據 SOPHOS 所提供的分析報告，Koobface 有四個行為特徵：1. 將 Koobface 執行檔複製至系統其它目錄、2. 修改 Windows 登錄檔，將 Koobface 執行檔加入開機自動載入清單、3. 建立程序 ld32.exe、4. 建立程序 cmd.exe。針對 Nugache 的實驗結果如表 4。

表 4：Koobface 分析結果比較

環境 / 行為	EDAM	CWSandbox	Malbed
將 Koobface 執行檔複製至系統其它目錄	O	O	—
修改 Windows 登錄檔，將 Koobface 執行檔加入開機自動載入清單	O	O	—
建立程序 ld32.exe	X	X	—
建立程序 cmd.exe	O	O	—

(註：表中使用符號說明，”O”表示有正確紀錄到該行為、”X”表示未紀錄到該行為、”—”表示惡意軟體樣本未正確執行，沒有產生分析報告。)

Koobface 於 EDAM 進行分析的結果，發現 Koobface 會將其執行檔複製至 C:\WINDOWS\ld09.exe，ld09.exe 會修改登錄檔 HKCU\Software\Microsoft\Windows\CurrentVersion\Run，以將 Koobface 執行檔加入開機自動載入清單中。而程序的部分因為 Koobface 所複製的執行檔為 ld09.exe，故所建立的程序名稱也是 ld09.exe，雖與 SOPHOS 所提供的分析報告檔名有所不同，但仍可對照出是屬於相同的行為。此一 Koobface 樣本於 Malbed 並沒有成功分析，出現錯誤訊息且未產生分析報告。

由本階段的測試得知，EDAM 對上述四類惡意軟體的分析結果，比另外兩種分析系統，較能如實紀錄樣本的行為特徵。檢驗 EDAM 系統提出一份具指導性的完整摘要式報告的效能。

伍、 結論

本研究提出的自動化惡意軟體動態分析環境，可紀錄各類惡意軟體執行期間所產生的行為，並針對分析結果，提出一份精簡具指導性的摘要式報告。在惡意軟體動態分析的研究發展而言，我們的研究改善了當前惡意軟體動態分析平台的缺點，提供一份精簡具指導性的摘要式報告，取代龐大、複雜的分析報告內容。

先前的研究皆著重在如何設計一個良好及可靠的動態分析環境，卻忽略分析後產生的大量資料，應如何處理的問題。當考慮人類精力與注意力是極其有限的情況下，處理大量資料產生的錯誤，將嚴重影響決策品質，導致系統出錯的機會大增。當前的分析平台，產生的分析報告內容過於龐雜，很難在短時間內獲得具指導性的資訊，有鑒於此，我們提出摘要式報告的想法，嘗試改善這個缺點。

參考文獻

1. Capture-BAT, https://honeynet.org/node/315 .
2. GFI Sandbox, Automated malware analysis tool http://www.gfi.com/malware-analysis-tool, 2010.
3. C. Seiferta, R. Steensona, I. Welcha, P. Komisarczuka and B. Endicott-Popovskyb, “Capture – A behavioral analysis tool for applications and documents ”, DFRWS Conference 2007,volume 4,p.23-30, 2007.
4. G Data Malware Report - Half-yearly report January–June 2011. http://www.gdatasoftware.com/uploads/media/G_Data_MalwareReport_H1_2011_EN.pdf
5. H. Binsalleeh, T. Ormerod, A. Boukhtouta, P. Sinha, A. Youssef, M. Debbabi, and L. Wang,“On the Analysis of the Zeus Botnet Crimeware Toolkit,”2010.
6. J. Eads, “EtherAnnotate: a transparent malware analysis tool for integrating dynamic and static examination”, 2010.
7. M. Christodorescu, and S. Jha,“Static Analysis of Executables to Detect Malicious Patterns”,2003.
8. Norman. Normal Sandbox. http://sandbox.norman.no/, 2006.
9. QEMU, http://wiki.qemu.org/Main_Page.
10. Rootkit, http://en.wikipedia.org/wiki/Rootkit.
11. Sober.Y, http://www.f-secure.com/v-descs/sober_y.shtml ,2009.
12. S. Stover, D. Dittrich, J. Hernandez, and S. Dietrich, “analysis of the Stormand Nugache trojans: P2P is here,”2007.
13. U. Bayer, A. Moser, C. Kruegel and E. Kirda,“Dynamic analysis of malicious code”, Journal in computer virology ,Volume 2, Number 1, 67-77 , 2006.
14. W32/Koobface-AZ, http://www.sophos.com/en-us/threat-center/threat-analyses/viruses-and-spyware/W32~Koobface-AZ/detailed-analysis.aspx .
15. Wireshark, http://www.wireshark.org/ .
16. Willems, C.Holz, T.Freiling, F. , “Toward Automated Dynamic Malware Analysis Using CWSandbox“, IEEE security & privacy, Volume 5 Issue 2, March 2007.

Concise Analysis of Malware Behavior

Chia-Mei Chen, Tse-Yao Wang, Hung-Shiuan Tsai
Department of Information Management, National Sun Yat-sen University
chiamei.chen@gmail.com, harry0716@gmail.com, finch319@gmail.com

Abstract

Malware analysis becomes important for attack detection, as increasing attacks based on various malware and detection or forensics requires the behaviors to identify the attacks. Current malware analysis tools often produce a comprehensive and lengthy report which is a burden for administrator to understand the key behaviors of the target malware. This study proposes an efficient dynamic analyzer of malware (EDAM) which generates a concise report summarizing the key behaviors as well as a complete one for further analysis. EDAM records and analyzes the malware behaviors on different aspects including processes, files, registries, network connections. The experiments show that the summary report is efficient and precise and indicate the applicability of the proposed system.

Keywords: Malware analysis, dynamic analysis, obfuscation