APT 與偵測方法之分類與分析

吳政穎 林盈達
國立交通大學資訊工程系
Email: cywu.cs02g@nctu.edu.tw; ydlin@cs.nctu.edu.tw
September 30, 2013

摘要

近幾年網路駭客的攻擊已不再使用傳統的攻擊手法，進而演變成了以 APT 的模式作攻擊，APT 的手法複雜、目標明確而且隱匿性高，因此可在受害者電腦端作長時間的潛伏同時作資料擷取的動作，甚至於伺機發動下一波的攻擊。因此如何有效的防範 APT 的攻擊，成了我們必須探討的問題。此實驗收集了六大類共 300 隻的 APT 病毒，並且分別利用靜態、動態以及逆向工程的偵測手法對這些病毒作行為的分析和分類，並且評估哪種偵測手法最能有效的作 APT 攻擊的防範和偵測。在偵測率方面，動態分析（85.3%）以及逆向工程（84%）的偵測率都遠高於靜態比對（35%）；在 APT 攻擊行為分析方面，發現 APT 攻擊手段多為在文件開啟階段即會自動執行程式作軟體弱點的攻擊並取得權限，再透過 Heap spray 的技術執行駭客欲執行的 shellcode，以完成入侵的動作，之後即能開始對感染電腦作遠端的存取控制。

關鍵字：APT、靜態分析、動態分析、逆向工程

1. APT 介紹

什麼是 APT

進階持續性滲透攻擊(Advanced Persistent Threat, APT)[1][2]，不同於傳統的攻擊模式，APT 是個有系統和計畫且針對特定目標組織的攻擊手法，特色在於他的手段更為複雜而且客製化(Advanced)，可以同時針對目標的多個弱點作攻擊，以達到首要任務入侵使用者電腦並取得權限，一旦入侵後，由於其低調且隱匿性高的特性，讓病毒可以作持續性的潛伏攻擊而不被察覺(Persistent)，動輒數個星期甚至於數個月，在這段時間裡，駭客會作機密資料的收集和分析工作，進而慢慢的入侵感染到整個公司或機構的網路系統，最後再將收集到的有利資料作外傳，這對任何一個公司或組織而言，都將會付出慘痛的代價，APT 儼然已成了各大公司機構的一大資安威脅。

APT 攻擊流程

偵查

APT 不同於傳統攻擊手法的一大特點就是鎖定目標，因此第一步即是擬定

計畫設定目標，目標可能是個人電腦金融機構或是政府機關，目標一旦鎖定後，即是對該目標作相關資料的收集，包括公司職員名單、職員電子郵件信箱、聯絡方式等相關資訊，資料蒐集的越充足，越有利於之後的入侵工作進行。

入侵

在入侵階段，駭客會利用社交工程的手法以達到入侵的目的，最常見的就是魚叉式網路釣魚，不同於以往的網路釣魚作大量無特定目標的攻擊，魚叉式網路釣魚攻擊目標更為明確，駭客往往會以他人之名寄發夾帶惡意文件的郵件，以取得收件人的信任並對附帶的文件作下載，一旦點擊文件作開啟，駭客便入侵成功並且安裝後門程式，但由於該惡意文件依舊會正常開啟，收件人絲毫無法察覺已被駭客入侵。

收集資料

入侵之後，駭客便能在有權限的狀態下，開始對感染的電腦作存取，並且竊取有利之相關資料，包括公司網路架構，電腦 IP 網域等重要資訊，甚至於利用這些取得的資訊作更進一步的入侵動作。

感染網路

如前一步驟所說，取得的有利資料，可協助駭客作更進一步的入侵，以作他台主機或伺服器的病毒轉移，期間皆是潛伏於整個網路系統中，不斷的作資料的收集分析和研究，包括使用者帳號密碼瀏覽紀錄，使用者完全無法察覺電腦有絲毫的異常，而這正是 APT 潛伏期長的一大特點。

資料擷取

在 APT 攻擊的最後一步驟，就是將收集到有利的相關資料壓縮後傳送到駭客端作存取，而這也是 APT 的最終目的，並且在受害電腦端移除所有的操作紀錄，抹除所以駭客入侵的痕跡。

圖 1-APT 攻擊流程圖

APT 與傳統病毒的比較

APT 與傳統病毒最大的不同在於 APT 是有計畫性且針對性的攻擊模式，加上客製化的攻擊手段，針對目標的弱點設計攻擊手法，再加上 APT 的隱匿性，達到長時間的潛伏以利於機密資料的收集和分析工作。表 1 即為 APT 攻擊與傳統攻擊的比較表，分別針對了病毒的持續性、針對性、隱匿性、是否為計劃性攻擊、攻擊是否客製化、攻擊動機以及常見攻擊目標共七個特性所作的比較。

表 1-APT 與傳統攻擊比較表

	APT Attacks	Traditional Attacks
Persistent	Yes	No
Targeted	Yes	No
Planned	Yes	No
Custom exploits	Yes	No
Hidden	Yes	No
Motivation	Collect benefit information and Exfiltration	Variable
Target	Individuals, Enterprise, or Government Organization	Unspecified

2. 偵測技術介紹

2.1 偵測技術分類

靜態分析（Static Analysis）

靜態分析為無需執行程式的分析方式，而是直接透過程式原始碼和病毒特徵碼資料庫的比對，一旦相符即能認定該程式為惡意程式，優勢在於由於不需要執行程式，因此偵測流程較為簡單而且快速，Overhead 也較低，大多數市售的防毒軟體皆是以這樣的模式作病毒的偵測。

動態分析（Behavior Analysis）

不同於靜態分析，動態分析則是透過在沙盒中執行程式的方式，對惡意程式的行為作監控和分析，優勢在於偵測出的結果較為符合確切的病毒行為，能更有效的判斷是否為惡意程式，但由於必須執行程式作分析，偵測流程上較為複雜，除了需要花費較長的時間，Overhead 也較靜態分析來得高，但病毒若有偵測虛擬環境的機制，動態分析將無法正確的分析出病毒確切的行為動作。

逆向工程（Reverse Engineering）

逆向工程為另一種病毒分析手法，由於許多駭客在開發惡意程式時，會使用程式混淆（Code Obfuscation）的技術來隱匿程式碼，例如以 Encrypt 或 Pack 的

方式作其惡意程式執行流程跟動作的隱藏，如此就能輕易的躲過靜態分析的偵測，因此動態分析的技術在於作程式碼的 Decrypt 或 Unpack，企圖去推導程式碼的執行流程跟架構，並適時作必要的程式執行以驗證推導的正確性。

表 2-APT 偵測技術比較表

Attributes / Methods	Execute File	Fast /Slow	Information	Overhead	Example Tools
Static Analysis	No	Fast	General	Low	ClamAV
Behavior Analysis	Yes	Slow	General	High	ViCheck.ca
Reverse Engineering	Partial	Slow	Detailed	High	XecScan

2.2 工具介紹

ClamAV

ClamAV[3]為靜態分析的代表工具，可疑的文件原始碼與 ClamAV 的病毒特徵碼資料庫進行比對，若有相符及認定為惡意文件，如圖 2 為 CVE-2013-0640 的 APT 病毒樣本經過 ClamAV 的靜態分析後，ClamAV 成功比對出相符的病毒特徵碼，並以此分析結果判斷該文件為 APT 惡意文件。

```
LibClamAV debug: cli_magic_scandesc: returning 1  at line 2449
LibClamAV debug: cli_pdf: returning 1
LibClamAV debug: FP SIGNATURE: 2a42bf17393c3caaa663a6d1dade9c93:828744:PDF.Exploit.CVE_2013_0
640
LibClamAV debug: cli_magic_scandesc: returning 1  at line 2449
02-CVE-2013-0640_PDF_2A42BF17393C3CAAA663A6D1DADE9C93_Mandiant.pdf: PDF.Exploit.CVE_2013_0640
 FOUND
LibClamAV debug: Cleaning up phishcheck
LibClamAV debug: Freeing phishcheck struct
LibClamAV debug: Phishcheck cleaned up
```

圖 2-ClamAV 偵測結果範例

ViCheck.ca

ViCheck.ca[4]為一款線上動態分析的工具，分析的結果會以報告的方式呈現，報告上除了一些基本的檔案資訊，還有最後針對所有監測到的行為所下的一個 Result，由圖 3 可知紅框處即顯示出 Embedded Executable 所執行的惡意行為，以及箭頭處的 Result。

XecScan

XecScan[5]利用了逆向工程技術，對使用者上傳的可疑文件作偵測的動作，針對有問題且加密的原始碼片段，作程式執行流程的推導，並且顯示出惡意程式執行的流程跟架構，圖 4 顯示出異常的網域連結（C&C 伺服器主機位址）以及不正常的執行程序，由此資訊 XecScan 認定此文件為惡意文件。

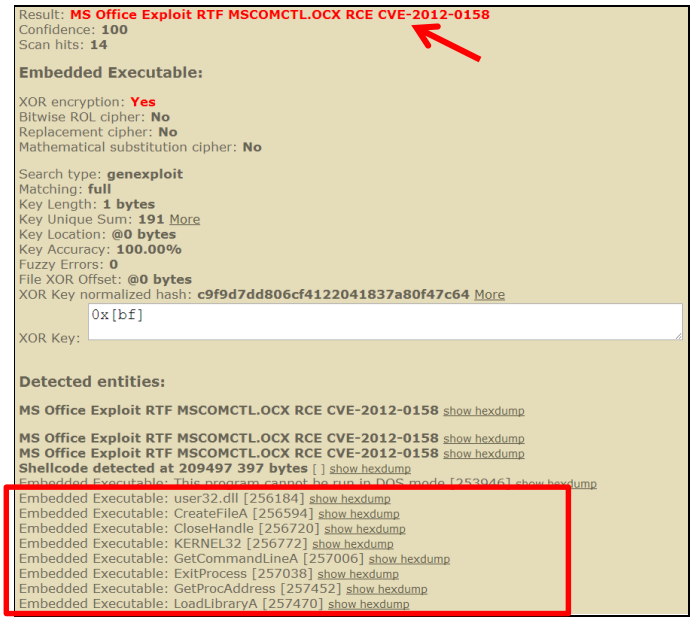


圖 3-ViCheck.ca 偵測報告範例

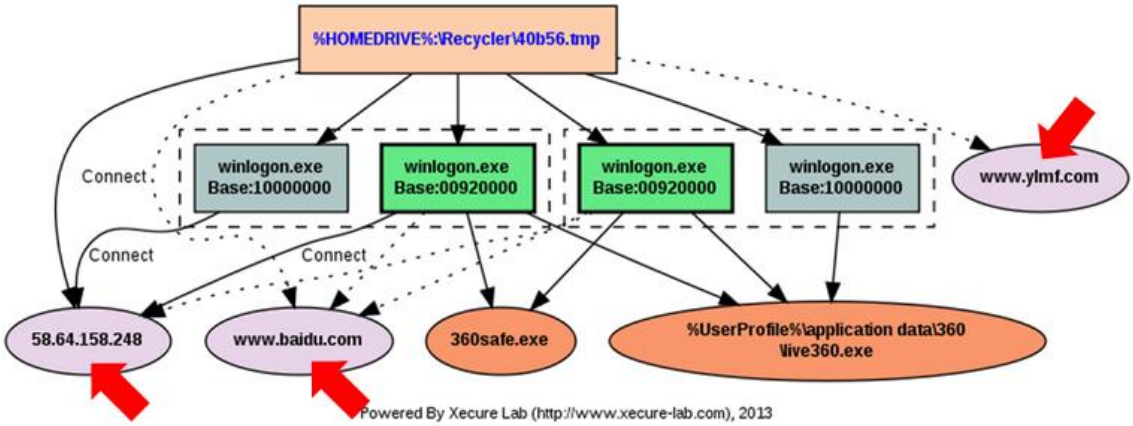


圖 4–XecScan 惡意程式執行架構圖範例

3. 實驗方法

實驗流程

圖 5 為本次實驗流程圖，實驗將 300 隻六大類針對不同軟體弱點作攻擊的 APT 樣本（列於表 3），分別使用 ClamAV（靜態分析）、ViCheck.ca（動態分析）、XecScan（逆向工程）三種不同分析技術的偵測工具作完整的掃描，評估在三種偵測技術下的 APT 偵測率，並透過 ViCheck.ca（動態分析）的行為報告，對這些 APT 樣本作病毒行為的分析和分類，最後再比對三種技術下的偵測數據，作個案的分析。

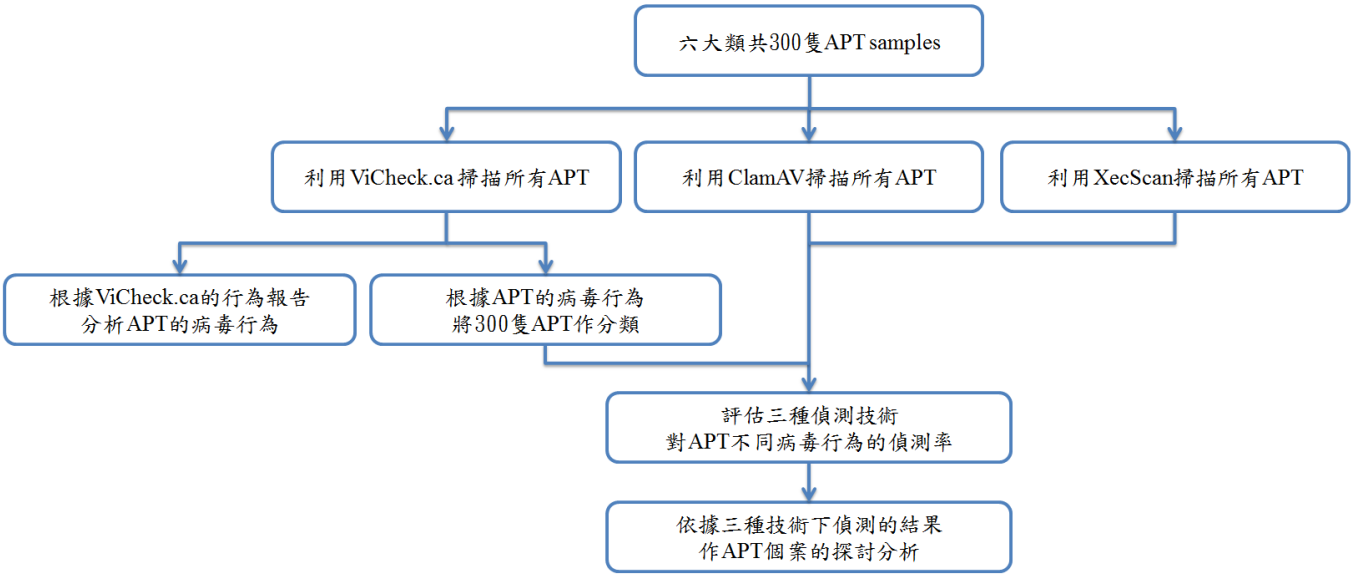


圖 5-實驗流程圖

表 3-各 CVE 病毒代碼的數量及所攻擊的軟體弱點[6][7]

CVE Number	File Type	# Samples	Product	Vulnerability
CVE-2010-0188	PDF	48	Acrobat Reader	Adobe Reader PDF LibTiff Integer Overflow
CVE-2010-2883	PDF	24	Acrobat & Acrobat Reader	Adobe CoolType SING Table Stack Overflow
CVE-2010-3333	RTF	52	Microsoft Office	MS Office 2010 RTF Header Stack Overflow
CVE-2011-2462	PDF	25	Acrobat & Acrobat Reader	Adobe Reader U3D Memory Corruption
CVE-2012-0158	RTF	131	Microsoft Office	Stack Buffer Overflow in MSCOMCTL.OCX
CVE-2013-0640	PDF	20	Acrobat & Acrobat Reader	Adobe Reader Unspecified Buffer Overflow

4. 實驗結果

4.1 APT 行為分析

依實驗的數據以及結果，得知 APT 的攻擊手段多為針對特定版本的軟體弱點作攻擊，而這些弱點多為緩衝區溢位（Buffer Overflow Error）的問題，一旦造成溢位成功即能透過 Heap spray 的技術執行駭客欲執行的 shellcode，以完成入侵的動作並啟動惡意程式檔，進而跟 C&C server 建立連線。由於檔案格式的不同，除了病毒碼放置位置相異，攻擊模式也有些微的不同：

PDF 格式

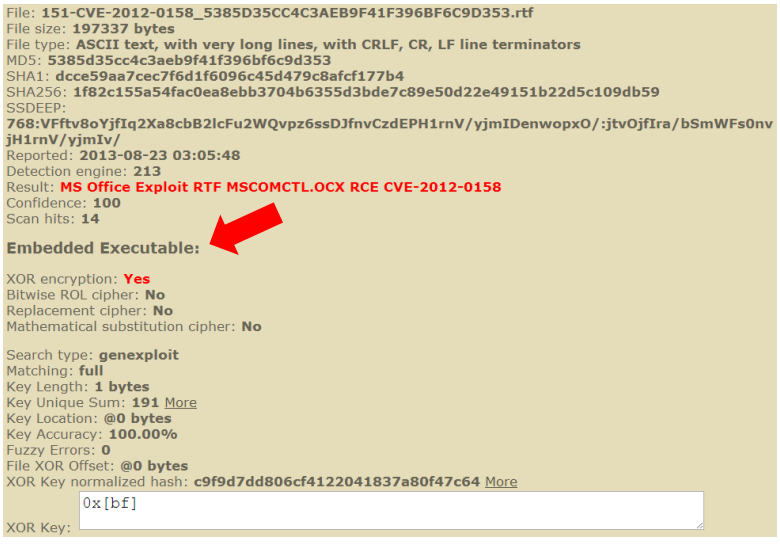
PDF 為一由物件架構組成的檔案格式，在開啟 PDF 惡意文件、循序執行各物件內容的過程中，即會自動的執行到駭客所嵌入在物件中的 JavaScript 程式碼，利用 JavaScript 針對軟體弱點產生溢位並利用 Heap spray 執行 JavaScript 內預先寫好的 shellcode，shellcode 即能在有權限的狀態下執行欲執行的惡意行為。

RTF 格式

RTF 檔的組成架構中，存在著一個 OLE（Object Linking and Embedding)的物件，此物件原用意是提供使用者作跨平台的文件編輯使用，而駭客的手法則是將嵌有惡意執行檔的 OLE 嵌入 RTF 檔中，當使用者執行惡意的 RTF 文件，即會自動執行 OLE objects 內的 shellcode 以及 Embedded Executable，而完成入侵動作。

4.2 APT 分類

從 ViCheck.ca 的偵測結果，分析 APT 樣本的行為報告，依據報告內容是否有偵測到執行檔的惡意行為亦即表示該惡意文件內嵌惡意執行檔，圖 6 即為 APT 攻擊文件內嵌有惡意執行檔的動態分析報告，依此報告內容可將 APT 樣本分成 Embedded Executable 以及 Download Executable 兩大類別，差別在於部分惡意文件會將惡意執行檔直接嵌入至文件當中，而有些卻是必須透過 shellcode 才能作下載惡意執行檔的動作，相較於從網路上下載執行檔，執行檔的嵌入提供了更多的行為資訊供偵測上的惡意與否判斷。


File: **151-CVE-2012-0158_5385D35CC4C3AEB9F41F396BF6C9D353.rtf**
File size: **197337 bytes**
File type: **ASCII text, with very long lines, with CRLF, CR, LF line terminators**
MD5: **5385d35cc4c3aeb9f41f396bf6c9d353**
SHA1: **dcce59aa7cec7f6d1f6096c45d479c8afcf177b4**
SHA256: **1f82c155a54fac0ea8ebb3704b6355d3bde7c89e50d22e49151b22d5c109db59**
SSDEEP:
768:VFftv8oYjfIq2Xa8cbB2lcFu2WQvpz6ssDJfnvCzdEPH1rnV/yjmIDenwopxO/:jtvOjfIra/bSmWFs0nvjH1rnV/yjmIv/
Reported: **2013-08-23 03:05:48**
Detection engine: **213**
Result: **MS Office Exploit RTF MSCOMCTL.OCX RCE CVE-2012-0158**
Confidence: **100**
Scan hits: **14**

Embedded Executable:

XOR encryption: **Yes**
Bitwise ROL cipher: **No**
Replacement cipher: **No**
Mathematical substitution cipher: **No**

Search type: **genexploit**
Matching: **full**
Key Length: **1 bytes**
Key Unique Sum: **191** More
Key Location: **@0 bytes**
Key Accuracy: **100.00%**
Fuzzy Errors: **0**
File XOR Offset: **@0 bytes**
XOR Key normalized hash: **c9f9d7dd806cf4122041837a80f47c64** More
XOR Key: 0x[bf]

圖 6-內嵌惡意執行檔的 APT 病毒文件動態分析報告

4.3 偵測率

依據圖 7 顯示，三種偵測技術對於 APT 惡意文件的偵測率，動態分析(85.3%)

以及逆向工程（84%）的偵測率都遠高於靜態分析（35%），由此可知，由於 APT 大多利用了 Code Obfuscation 的隱匿技術，甚至於是多重式的加密，不僅增加了偵測上的難易度，也造成靜態分析的低偵測率，在 300 隻的 APT 樣本中，靜態分析僅偵測到 105 隻的惡意程式，相較於其他偵測技術，動態分析以及逆向工程皆有五成以上的偵測率。

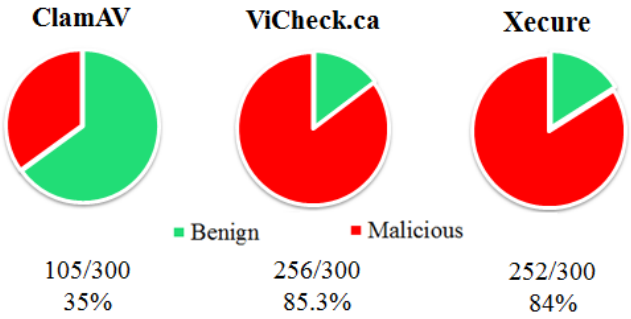


圖 7-三種技術對 APT 病毒的偵測率

由於不同的檔案格式，病毒的行為模式也不完全相同，因此根據不同的檔案類型作 APT 樣本的掃描，由表 5 得知，相較於 ViCheck.ca 和 XecScan 五成以上的偵測率，ClamAV 不管對於 PDF 或 RTF 的檔案格式，偵測率都偏低。

表 5-三種技術對不同檔案格式 PDF 跟 RTF 的偵測率

Tools / File format	ClamAV	ViCheck.ca	XecScan
PDF File	49.6%	81.2%	62.4%
RTF File	25.7%	88.0%	97.8%

依據是否內嵌惡意執行檔的分類去進行 APT 文件的掃瞄，見表 6 所示，ClamAV 的偵測率依舊低於其他兩樣工具，而就觀察 ViCheck.ca 以及 XecScan 這兩樣工具的數據，發現有內嵌惡意執行檔的文件偵測率皆高於無內嵌的文件，原因可能在於偵測到的線索變多，有更多的資訊去判斷該文件是否為惡意程式，而造成這樣的結果。

表 6-三種技術對有無內嵌惡意執行檔的病毒樣本偵測率

Tools / Executable	ClamAV	ViCheck.ca	XecScan
Embedded Executable	26.6%	88.9%	96.1%
Download Executable	53.8%	77.4%	89.2%

此外，在 CVE-2011-2462 的 APT 樣本中，發現不少除了 ViCheck.ca 能成功偵測外，其餘兩種技術皆顯示非惡意程式的情況，比對了同一 CVE-2011-2462 類別下其他樣本的 ViCheck.ca 報告結果，發現原因可能在於這些 APT 都使用了兩種的加密方法（FlateDecode | ASCIIHexDecode）(圖 8)，這不僅對 ClamAV 來說是一大偵測阻礙，也增加了 XecScan 在推導程式流程上的困難度，因此無法正確辨識。

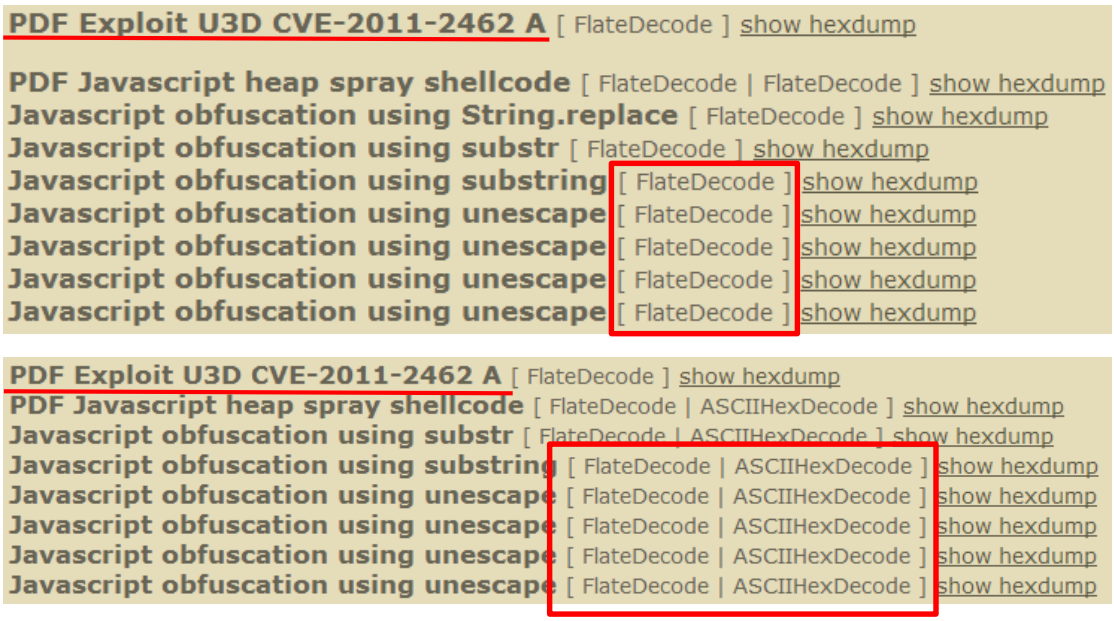


圖 8- CVE-2011-2462 的 ViCheck 偵測報告

最後，在 CVE-2013-0640 類別的偵測上發現，ViCheck.ca 無法明確的指出病毒代號即為 CVE-2013-0640，而是依程式碼有加密的行為來推定該文件為可疑或惡意的文件，研判這樣的成因在於 APT 的攻擊都是針對特定軟體的特定版本作弱點攻擊，因此在入侵的初期，APT 的惡意程式碼都會有軟體版本的判斷，若版本不符也就不會有後續造成溢位的動作，以此觀點，經查證 ViCheck.ca 官網，發現最近一次的更新為 2012 年八月，而 Adobe Acrobat 11 發行日期是在 2012 年 10 月，加上 CVE-2013-0640 是專對 Adobe 10 之後版本所作的弱點攻擊，因此更能研判是由於偵測環境的軟體版本不符，造成在動態分析的過程中，程式並無執行攻擊的動作，ViCheck.ca 也才無法明確的辨識出是 CVE-2013-0640 的 APT 攻擊病毒。

5. 結論

由以上圖表，我們可以歸納以下結論：

1. 在不同 APT 檔案格式的分析上，發現不同的 APT 檔案格式也會有不一樣的病毒內嵌位置，攻擊手法也會有不同，而在偵測率方面，發現不管是 PDF 或 RTF，靜態分析技術的偵測率都遠低於其他兩樣技術，而動

態分析技術對 PDF 格式的偵測率最高（81.2%），而 RTF 格式則以逆向工程技術偵測率最好（97.8%）。

2. 在 APT 有無內嵌惡意執行檔的偵測率分析上，發現不管是無內嵌或內嵌有惡意執行檔的 APT 文件，靜態分析技術的偵測率依舊低於其他兩樣技術，而相反的在三種偵測技術中，逆向工程偵測技術則都有最高的偵測率，分別為內嵌惡意執行檔文件的 96.1%以及無內嵌文件的 89.2%。
3. 總體而言，三種偵測技術在 300 隻 APT 樣本的防護上，動態分析技術有 85.3%的偵測率；逆向工程為 84%；而靜態分析為 35%，相較於靜態分析，動態分析以及逆向工程都有八成以上的偵測率，若能作技術上的整合，相信對於 APT 病毒會有更好的偵測效率。
4. 市售的防毒軟體多採用靜態分析的技術，由於 APT 的隱匿技術，讓 APT 躲過了許多防毒軟體的的偵測，若能針對此一問題作技術上的改良，相信能提高靜態分析對 APT 的防範效果。
5. 縱使再好的偵測技術，都還是會有漏網之魚，最好的防範 APT 策略還是要養成良好的網路使用習慣， 不下載來路不明的檔案，才是對 APT 的因應之道。

參考文獻

[1] I. Jeun, Y. Lee, D. Won, “A Practical Study on Advanced Persistent Threats,” International Conferences, SecTech, CA, CES3 2012, vol. 339, pp. 144–152, Nov. 28-Dec. 2, 2012.

[2] Colin Tankard, “Advanced Persistent threats and how to monitor and deter them, ” Network Security, Vol. 2011, No. 8, pp. 16-19, August 2011.

[3] ClamAV, http://www.clamav.net/lang/en/

[4] ViCheck.ca - Find embedded malware in documents, PDFs or emails, https://www.vicheck.ca

[5] XecScan - Xecure Lab, http://scan.xecure-lab.com/

[6] National Vulnerability Database, http://nvd.nist.gov/home.cfm

[7] CVE Details, http://www.cvedetails.com

[8] The Wikipedia - Advanced persistent threat, http://en.wikipedia.org/wiki/Advanced_persistent_threat

[9] Command Five Pty Ltd, “Advanced Persistent Threats:A Decade in Review”, June 2011.

[10] Aditaya K Sood, Richard J. Enbody, “Targeted Cyberattacks: A Superset of Advanced Persistent Threats, ” Security & Privacy, IEEE （Volume:11, Issue : 1）, pp.54 – 61, July 2012.