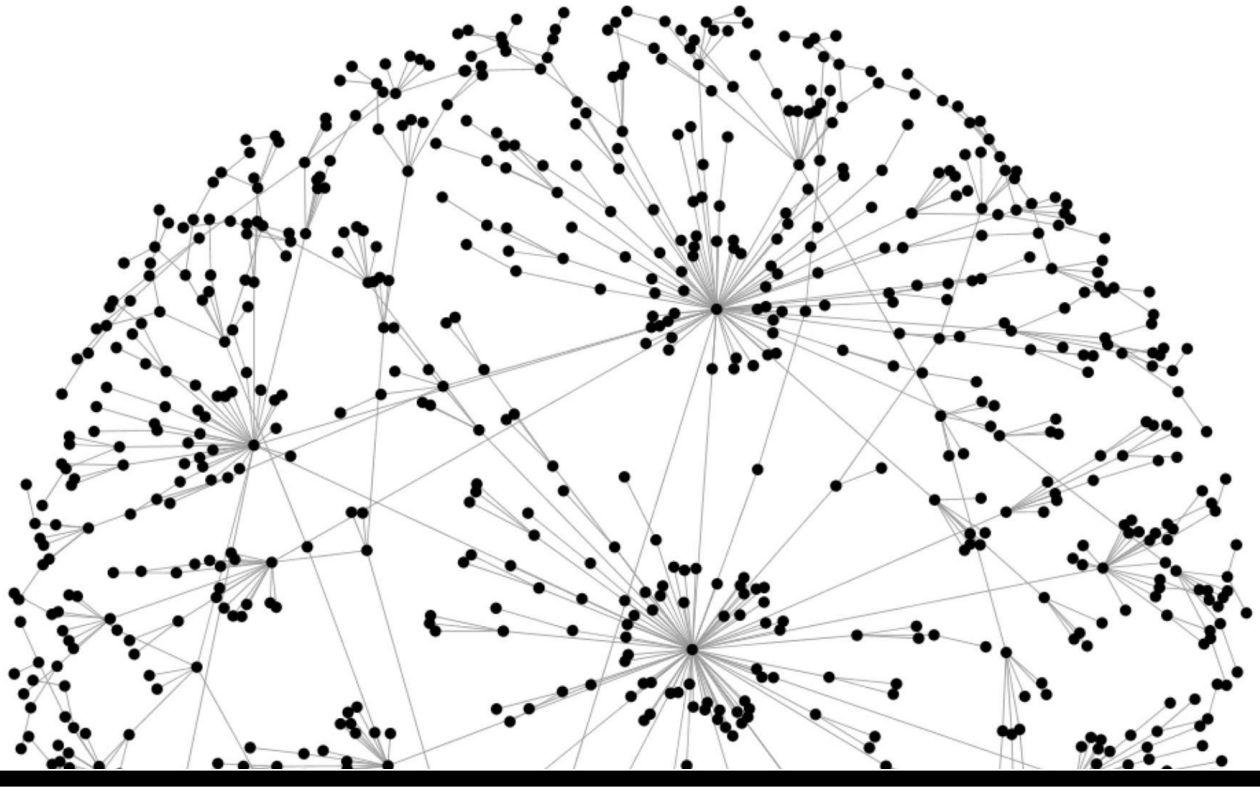
# أساسيات تحليل البرامج الضارة

# Malware Analysis Basics

## Haboob - Team



# Table of Contents

## أهداف تحليل البرمجيات الضارة:

هدف تحليل البرامج الضارة هو فهم جزئ محدد من البرنامج لتكوين ألية دفاع وكشف. يوجد سؤالين جوهرين في ألية تحليل البرامج الضارة، وهيا:

1. كيف أنتشر هذا البرنامج وأصاب الضحايا؟
2. ماذا يفعل بالضبط بعد أصابه الهدف؟

بعد تحديد نوع البرنامج الضار، يجب تحديد أي سؤال أهم في الوضع الحالي.

يمكن تلخيص أهداف تحليل البرامج الخبيثة إلى:

- تطوير التوقيعات والمؤشرات للكشف عن انتشار فايروس في الشبكة.
- استخدام التواقيع لفهم كيفية عمل جزء معين من البرامج الضارة بحيث يمكن بناء الدفاعات لحماية منه.

يوجد تواقيع للكشف عن البرامج الضارة على عدة مستويات:

- مستوى الجهاز الشخصي (المضيف) (Host-Based Signatures)
  - يتم تحديد البرنامج الضار بتحديد الملفات التي تم إنشاؤها أو تعديلها بواسطة البرامج الضارة أو التعديلات الي تتم في الرجيستري (Registry).
- مستوى الشبكة (Network Based Signatures).
  - يتم اكتشاف البرنامج الضار من خلال مراقبة حركة مرور الشبكة (Network Trafficking).

## أنواع البرامج الضارة:

- الباكدور(Backdoor): كود ضار يقوم بتثبيت نفسه على جهاز كمبيوتر للسماح للمهاجم بالوصول إلى الأوامر الموجودة على النظام المحلي وتجاوزها وذلك بتجاوز عناصر تحكم الأمان العادية.
- بوتنت(Botnet) : عدد من أجهزة كمبيوتر الإنترنت المصابة برمز خبيث لتلقي نفس التعليمات من خادم واحد للتحكم والقيادة.
- داونلودر(Downloader): كود هدفه الوحيد تنزيل برامج ضارة على جهاز الضحية.
- روتكت(RootKit) : كود ضار مصمم لإخفاء وجود بعض العمليات أو البرامج (على سبيل المثال، الباكدور) من طرق الكشف العادية ويتيح استمرار الوصول المستمر إلى جهاز الضحية.
- دودة الشبكة (Worm): برنامج ينسخ نفسه ذاتيا وينتشر في الشبكة.
- فايروس (Virus): برنامج ذاتي النسخ يقوم بإعادة إنتاج الشفرة الخاصة به عن طريق إرفاق نسخ في أكواد أخرى قابلة للتنفيذ

## أساليب تحليل البرمجيات الضارة:

يوجد أسلوبين أساسيين لتحليل البرامج الضارة:

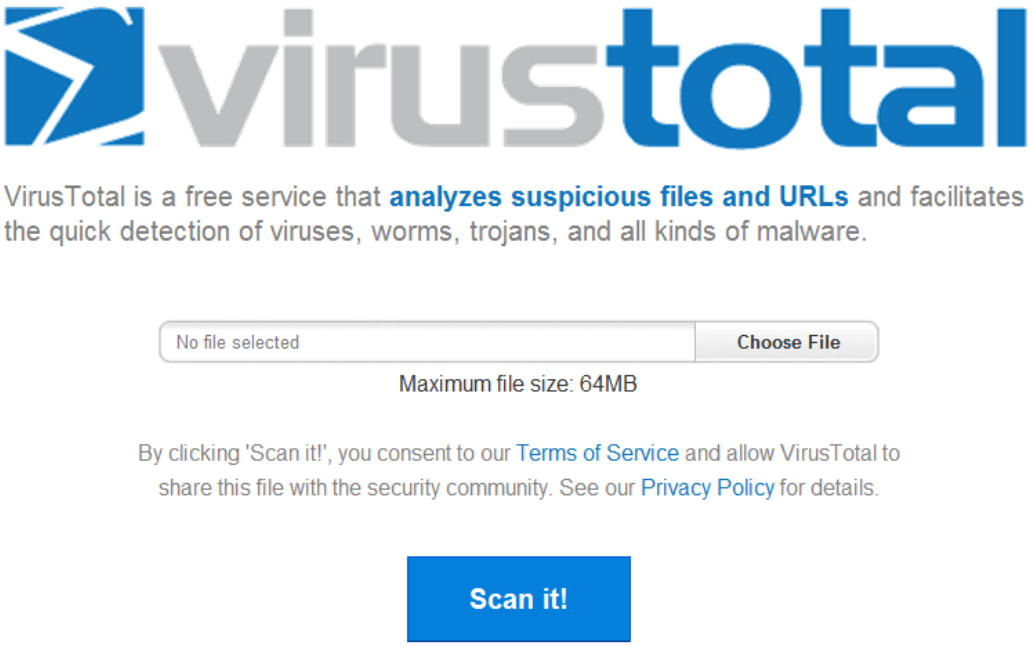
- التحليل الثابت.
  - يتضمن تحليل كود أو هيكل برنامج لتحديد وظيفته بدون تشغيله بمعنى آخر تحليل كامل للبرنامج في وضعيه السكون بدون تشغيل الملف.
- التحليل المتغير (الديناميكي).
  - يتضمن تشغيل البرامج الضارة ومراقبة سلوكها إذا تنشئ اتصالات بسيرفر خارجي او إذا أجرى اي تعديلات بالرجيستري الخ.... يجب إعداد بيئة معزولة تسمح بدراسة البرامج الضارة قيد التشغيل دون الإضرار بالنظام أو الشبكة.

### التحليل الثابت (Basic Static Analysis):

أول خطوة في التحليل الثابت هوا جمع معلومات عن البرنامج الضار، ومن الأساليب النموذجية لجمع معلومات عن البرنامج:

1. استخدام مكافح الفيروسات (Anti-Virus) للتأكد إذا البرنامج ضار أم لا.
   - فحص الAntivirus يعتمد على توقيعات الملفات المريبة المعروفة والتحليل السلوكي ومطابقة الأنماط ملفات المشبوهة.
   - تحتوي ملفات التعريف على تواقيع ملفات للبرامج الضارة التي تمت مواجهتها.

نظرًا لأن برامج مكافحة الفيروسات المختلفة تستخدم توقيعات وتوجيهات مختلفة، فمن المفيد تشغيل عدة برامج مختلفة لمكافحة الفيروسات ضد نفس البرنامج المشتبه به. موقع VirusTotal يسمح بمسح البرنامج المشبوه بعدة برامج مكافحة للفيروس.

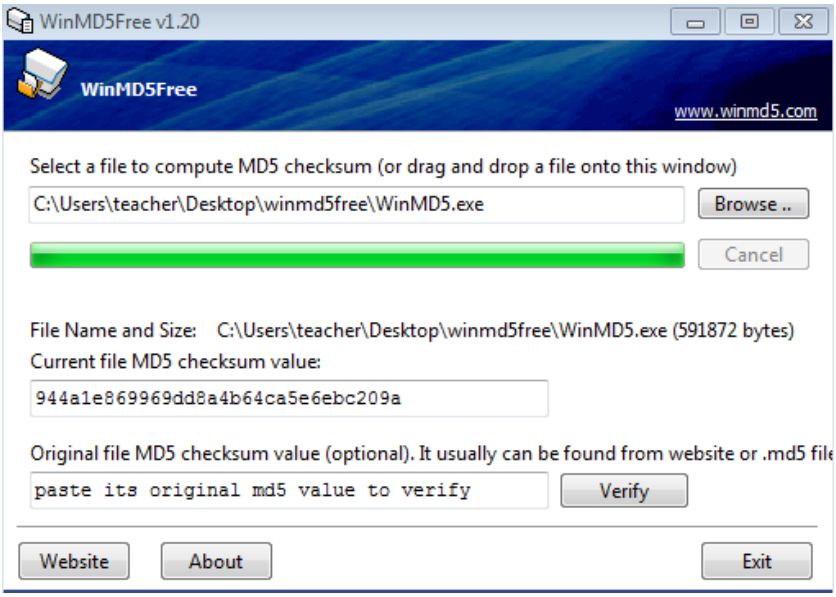


2. استخدام الهاش للحصول على بصمة البرامج الضارة.

- يمكن استخدام الهاش للاستدلال بالبرامج الضارة كبصمة (Fingerprint)، بحيث ان كل برنامج ينتج هاش فريد.
- خوارزمية MD5 تعتبر أشهر خوارزمية لا نتاج هاشات لبرامج مختلفة.

بعد الحصول على الهاش الخاص في البرنامج الضار، يمكن استخدام الهاش كعلامه لتحديد والتعرف على البرامج الضارة أو مشاركته مع محللين أخرين لمساعدتهم بالتعرف على البرنامج او البحث في الأنترنت إذا كان البرنامج الضار تعرف وأنكشف مسبقا.

من الأدوات التي يمكن حساب الهاش فيها WinMD5.



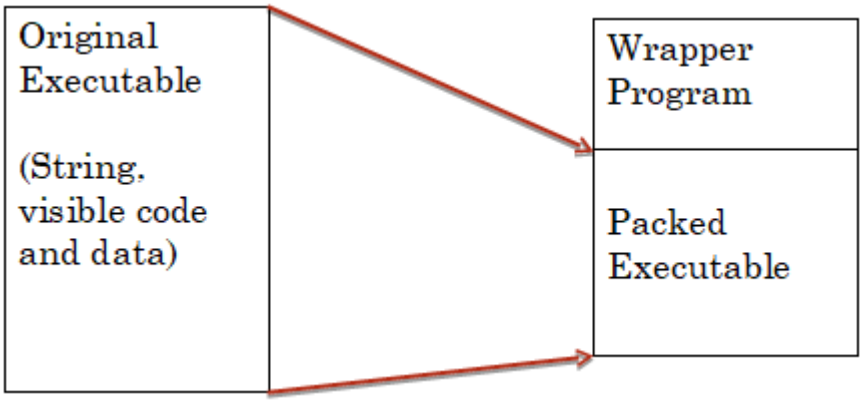


3. فك حزم البرامج الضارة و المضغوطة (Packed Malware).

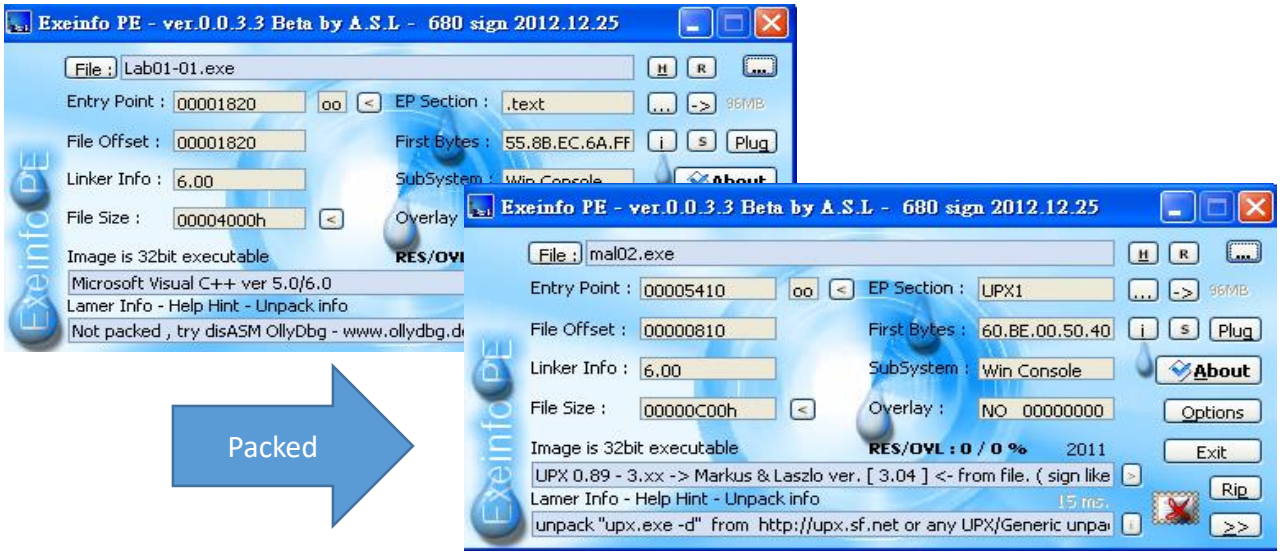
- تستخدم أساليب الضغط والتشويش (Obfuscation) من قبل مبرمجين البرامج الضارة لتفادي الجدران النارية (Firewalls) وال-Anti Virus.
- برامج الضغط (Packers) عبارة عن حزمة برمجية تشفر وتضغط ملفات exe قبل تنفيذها لإخفاء تواقيع وبصمة البرنامج الضار، وتستعيد صورة البرنامج الأصلية بدون اي تعديل قبل تحميلة في الذاكرة.

قبل أجراء أي تحليل على البرنامج، يجب التأكد من أجراء عملية فك الضغط (Unpacking).

عند تشغيل البرنامج الضار المضغوط، يقوم جزء صغير من البرنامج بفك الضغط وتشغيل البرنامج الأصلي.
تحتوي البرامج المضغوطة في الكثير من الأحيان على دوال مثل: Load Library و GetProcAddress




من أشهر البرامج لفك الملفات المضغوطة PEiD و ExeInfo PE



4. جمع معلومات من النصوص الموجودة داخل البرنامج، مثل أسامي الدوال (Headers).

- النصوص (Strings) تكون مخزنة بصيغة ASCI( byte/char) أو صيغة Unicode (2bytes/char).
- قد يحتوي البرنامج على نصوص في إذا قام بطباعة رسالة أو الاتصال بموقع ويب.
- البحث وجرد النصوص التي يحتويها البرنامج قد توحي عن وظائفه.

بعض الأدوات لاستخراج النصوص: Windows Sysinternals Utilities: Strings

```
Strings v2.5
Copyright (C) 1999-2012 Mark Russinovich
Sysinternals - www.sysinternals.com

!This program cannot be run in DOS mode.
Rich
UPX0
UPX1
...
DmM
;0I
PQ6
(23h
MalService
sHGL345
http://w
warean
ysisbook.co
om#Int6net Explo!r 8FEI
.0<
SystemTimeToFile
...
...
KERNEL32.DLL
ADVAPI32.dll
```

```
MSVCRT.dll
WININET.dll
LoadLibraryA
GetProcAddress
VirtualProtect
VirtualAlloc
VirtualFree
ExitProcess
CreateServiceA
exit
InternetOpenA
```

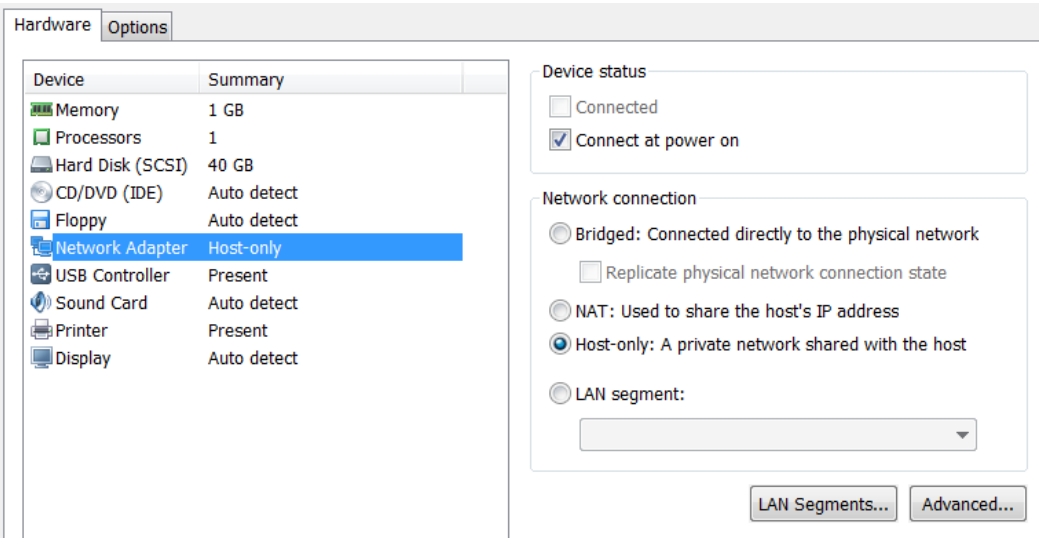
## طريقة التحليل الديناميكي:

المرحلة الأولى: الاستعداد

يجب تجهيز أجهزة مادية او افتراضية مخصصة لفحص البرامج الضارة، وتكون محدثة ومتوفر فيها جميع الخدمات التي يمكن ان يستغلها البرنامج الضار لتخريب او تركيب الBackdoor .

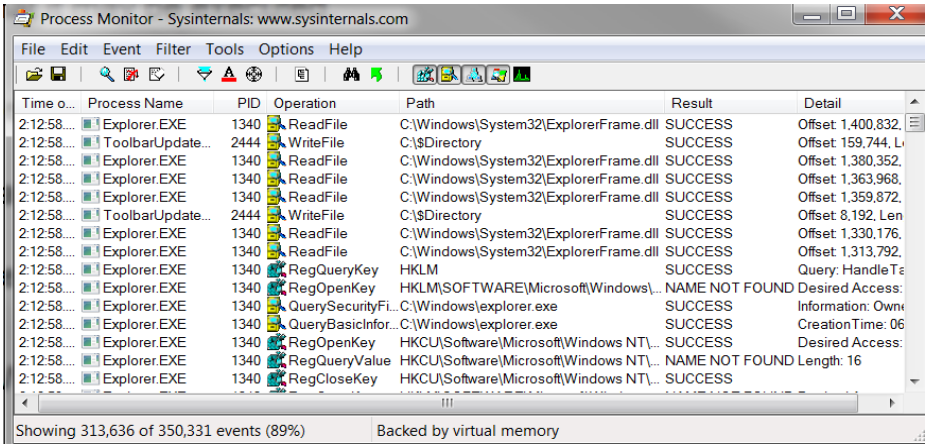
بعد تجهيز الجهاز، يجب عزل البيئة لمنع انتشار البرنامج في الشبكة الداخلية، ويتم ذلك عبر استخدام الشبكات المنعزلة ( Air Gapped Networks). الأفضل استخدام شبكة متصلة في الأنترنت بحيث يمكن تفحص كل خصائص البرنامج.

في حالة استخدام جهاز افتراضي، يجب التأكد من وضع إعدادات الشبكة الخاصة في الجهاز الى Host – Only لتفادي انتشار البرنامج لأجهزة افتراضية أخرى أو أجهزة حقيقية على نفس الشبكة.

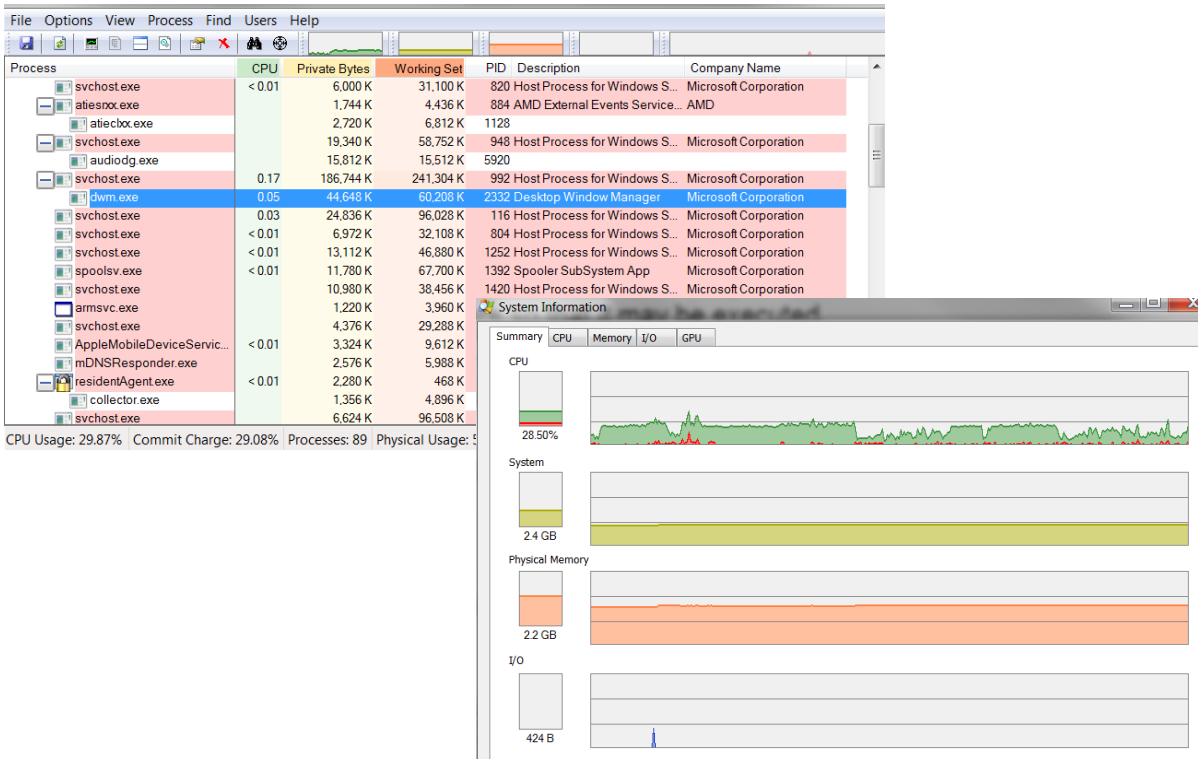



المرحلة الثانية: الأدوات الأساسية

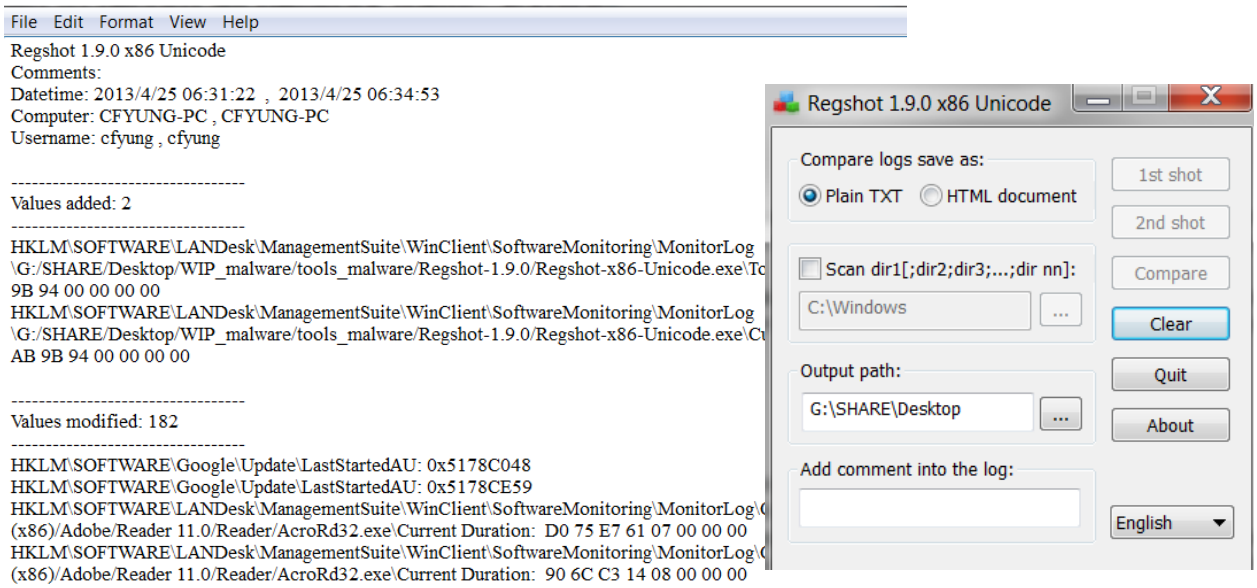
- أداة مراقبة المعالج Processor Monitor (procmon)
  - المصدر: Microsoft
  - الفائدة: أداة متقدمة في مراقبة الملفات والRegisters الخاصة في نظام التشغيل ونشاط المعالج.



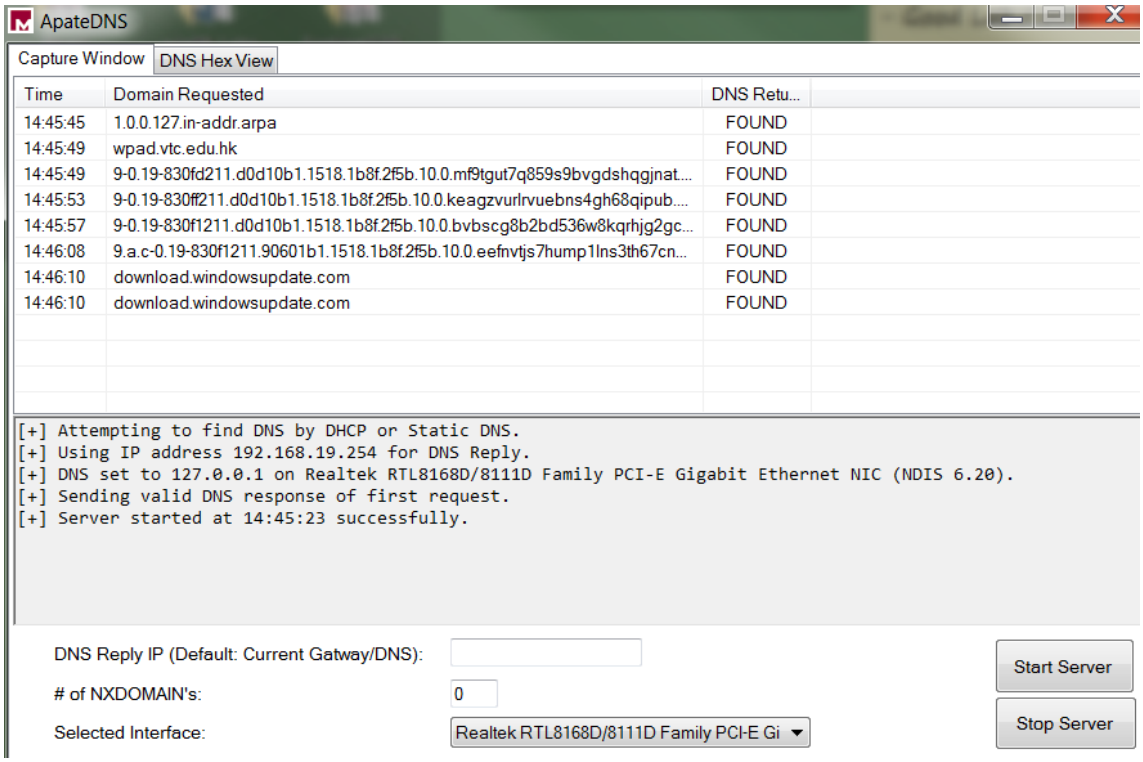
- أداة مستكشف المعالج (Process Explorer):
  - المصدر: Microsoft
  - الفائدة: تدرج جميع الProcesses والDLLs المستدعية من البرامج ومعلومات عن النظام بشكل عام.



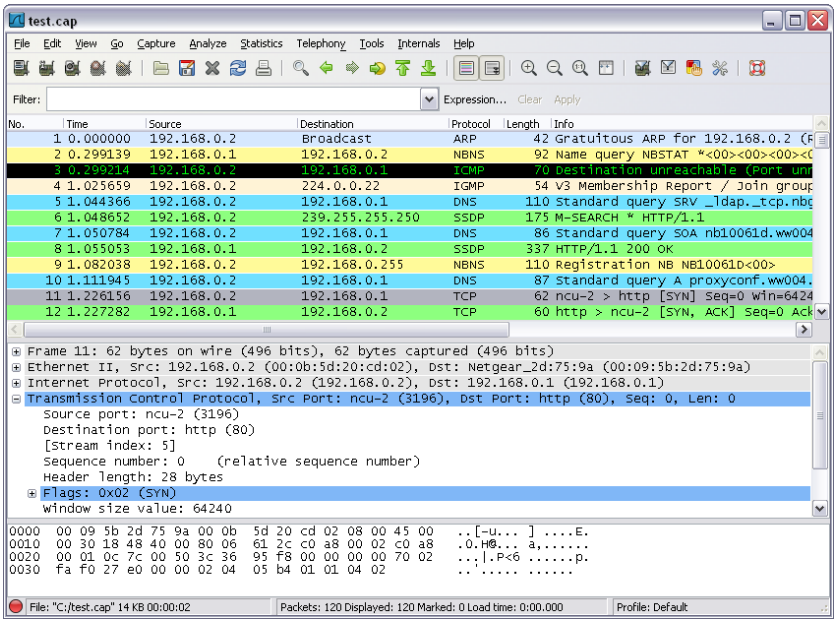

- أداة Regshot
  - المصدر: مفتوحة المصدر
  - الفائدة: تسمح بالتقاط وتسجيل حالة Registry قبل تشغيل البرنامج ومن ثم مقارنتها بحالتها بعد تشغيل البرنامج إذا كانت هناك أي تعديلات.



- أداة ApateDNS:
  - المصدر: Mandiant
  - الفائدة: تلتقط الأداة أي طلب خاص في الDNS من البرنامج الضار.




- أداة Wireshark:
  - المصدر: مصدر مفتوح
  - الفائدة: متنصت على مستوى الشبكة يعترض ويسجل جميع البيانات المارة في الشبكة.



المرحلة الثالثة: الخطوات العملية

التحليل الديناميكي يساعد ويؤكد النتائج التي تم الحصول عليها من التحليل الثابت.

- الخطوة الأولى: المعطيات الأساسية
  - قبل تشغيل البرنامج الضار المراد اختباره، يجب أخذ صورة من الRegistry باستخدام أداة Regshot.
- الخطوة الثانية: متابعة حالة النظام
  - خلال فترت تشغيل البرنامج الضار، تستخدم اداتين Process Monitor و Process Explorer لتسجيل اية تحركات.
    وتغيرات للبرنامج الضار خلال فترت عملة.
- الخطوة الثالثة: متابعة الشبكة
  - تستخدم أدوات ApateDNS و Wireshark لتسجيل البينات التي تصدر الى الشبكة من البرنامج الضار و متابعة طبيعة هاذي البينات.
- الخطوة الرابعة: المقارنة
  - انتظار البرنامج الضار بقيام أي تغير على النظام، ومن ثم نقوم بالتقاط حالة النظام باستخدام Regshot ومقارنة التغيرات مع الحالة الأولى.