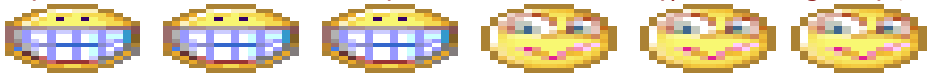
Như các bạn đã biết, hiện nay có rất nhiều malware xuất hiện, do vậy với lượng malware ngày càng nhiều như vậy thì việc phân tích chúng cũng rất mất nhiều thời gian, chưa kể đến những malware phức tạp. Bài viết này với mong muốn giúp các bạn có thể hiểu sơ lược về hoạt động malware và cách thức xây dựng một hệ thống phân tích malware tự động như thế nào ? Theo ý kiến của seamoun thì không tham vọng xây dựng một hệ thống phân tích malware tự động mà có thể phân tích được tất cả malware xuất hiện, bởi vì malware hiện nay xuất hiện với nhiều hình thức khác nhau và hoạt động của chúng thì rất phức tạp (người phân tích soi còn không ra huống gì hệ thống tự động !!! ), thứ hai mà hệ thống có thể phân tích được tất cả các malware xuất hiện thì seamoun chắc nộp đơn xin nghỉ việc, vì mình chẳng có việc gì phải làm  . Nhưng tại sao vẫn phát triển hệ thống phân tích malware tự động ? bởi vì hệ thống này sẽ giúp đỡ cho người phân tích bớt thời gian hơn cho những công đoạn mà người phân tích malware và cũng có cái nhìn tổng quát về những hành động, hành vi ban đầu của malware.

Ví dụ: Khi có một malware mới xuất hiện thì ban đầu để phân tích chúng thì phải chạy malware cho một môi trường độc lập (tức là môi trường khi mà malware thực hiện sẽ không gây hại gì cho hệ thống), môi trường như vậy người ta gọi là sandbox. Tiếp đến người phân tích sử dụng các công cụ phân tích sơ bộ rằng malware thực hiện những hành động gì khi chạy chúng (lưu registry key nào, tạo file gì mới trên hệ thống, thay đổi , tiêm nhiễm gì ....) sau đó sử dụng các công cụ như Ollydbg, IDA, ... để tiến hành phân tích code ... (ngồi đọc code và phân tích mấy con malware này chắc hacnho, mrro, ... các anh em khác trong HVA chắc rành hơn, còn seamoun nhìn thấy là hoa cả mắt ). Như vậy có một số hoạt động phân tích sơ bộ về malware lặp đi lặp lại trong quá trình phân tích malware thì có thể chuyển nó cho một hệ thống phân tích tự động trước khi người phân tích malware tiến hành phân tích sâu hơn.

Ý tưởng và triển khai như sau:
Sẽ có một máy Linux làm master, trên Linux sẽ thực hiện cài đặt một máy ảo (sử dụng VMWare) có cài đặt Windows XP (đây là môi trường để malware chạy và thực hiện phân tích). Linux và máy ảo chỉ kết nối Host-only, tức là chỉ có máy Linux và WMware nói chuyện được với nhau, để khi thực thi malware sẽ không làm ảnh hưởng đến các hệ thống khác. Trên máy Linux sẽ có một kịch bản shell được viết với mục đích thực hiện các phân tích ban đầu đối với malware (sẽ giải thích rõ hơn phần sau), tiếp đến sẽ thực hiện tự động bật VMware lên và đẩy malware vào môi trường VMWare, trên VMware sẽ có sẵn một số công cụ phân tích malware và thực hiện gọi script phân tích (sử dụng AutoIt Script) gọi các công cụ phân tích trên môi trường VMWare để tiến hành phân tích và tất cả các kết quả sau khi phân tích được xuất ra và gửi trở lại cho môi trường Linux và kết thúc quá trình phân tích.

Trên đây mình chỉ giới thiệu sơ bộ, chi tiết các bạn sẽ theo dõi phần sau đây:

I. Các bước xây dựng một Automatic Malware Ananlysis
I.1. Sử dụng một master script, script này sẽ chạy trên môi trường Linux và trên Linux có cài đặt sẵn WMWare có Windows XP SP2 (Sử dụng máy ảo này như một sandbox).

I.2. Trên Windows XP SP2 của máy ảo trong Linux nên chuẩn bị như sau:
Cấu hình IP Address tĩnh và để card mạng chỉ là Host-only
Tạo một thư mục chia sẻ VMWare để giao tiếp giữa Linux và WindowsXP
Đặt user và pass cho máy Windows XP
Firewall của Windows XP phải tắt
Windows XP Simple File Sharing nên tắt

Chuẩn bị công cụ SysInternal's Process Monitor và RegShot, … có thể thêm công cụ phục vụ phân tích tùy thích.
Cài đặt AutoIT để cho kịch bản AutoIt có thể chạy được.

II. Triển khai ứng dụng
II.1. Cài đặt Backtrack trên hard disk
Sử dụng tiện ích ubiquity

II.2. Cài đặt VMWare Workstation trên Linux
Sử dụng phiên bản: VMware-Workstation-6.5.3-185404.i386.bundle
Số serial : MVDUJ-TF4DT-284DV-4W9Z7

II.3. Cấu hình network trên Linux và máy ảo Windows XP
Linux
eth0 ip: 192.168.1.234
vmnet1:192.168.32.1
Windows
Eth0 ip: 192.168.32.2
Gateway: 192.168.32.1

II.4. Cài đặt InetSim trên Linux
II.4.1. Giới thiệu
InetSim là một gói mà chứa các kịch bản Perl được sử dụng để mô phỏng các dịch vụ thông dụng như DNS, HTTP và FTP. Khi chạy, InetSim sẽ đợi những kết nối và log bất kỳ những gì mà nó nhận được trong định dạng log chuẩn, dễ đọc khi phân tích các kết nối ra ngoài của malware.

II.4.2. Cài đặt
Địa chỉ trang Web: http:/www.inetsim.org.
Xem yêu cầu cần thiết trước khi cài đặt InetSim: http://www.inetsim.org/requirements.html
Giải nén và cài đặt tại /usr/local/inetsim.
Tạo nhóm sử dụng cho InetSim
groupadd inetsim
./setup.sh
Cấu hình
File cấu hình sẽ nằm tại thư mục sau: /usr/local/inetsim/conf/
Thiết lập hai tham số service_bind_address và dns_default_ip là địa chỉ của vmnet1.

II.5. Cài đặt Volatility Framework trên Linux
II.5.1. Giới thiệu
Để phục vụ cho việc Memory Analysis thì phải cần phải cài đặt Volatility Framework. Vì sao phải thực hiện Memory Analysis ? Bởi vì nhiều chương trình mã độc sử dụng packer để che dấu malware và làm khó khăn trong việc phân tích chúng. Tuy nhiên, packed malware phải unpacked trong bộ nhớ để thực thi. Bằng việc dumping một chương trình malware trong bộ nhớ, có thể kiểm tra nó mà không cần một packer.
Do chúng ta sử dụng máy ảo cho nên việc ngừng tạm thời hoạt động của máy ảo thì chúng ta có thể phân tích memory của máy ảo thông qua file có phần mở rộng là .vmem.

II.5.2. Cài đặt
Địa chỉ Volatility Framework: https://www.volatilesystems.com/default/volatility

II.6. Download PEiD và WinExe cho Linux

PeiD

Địa chỉ: http://blog.didierstevens.com/

WinExe

winexe thực hiện giống chức năng của chương trình psexec trên Windows, cho phép thực hiện lệnh từ xa.

Địa chỉ: http://eol.ovh.org/winexe/

II.7. Cài đặt AutoIt, Regshot, TCPView trên Windows

Cài đặt AutoIt

Địa chỉ: http://www.autoitscript.com/autoit3/

Cài đặt Regshot

Địa chỉ: http://sourceforge.net/projects/regshot

Cài đặt TCPView

Địa chỉ: http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx

Lưu ý: công cụ Regshot và TCPView được tải về và cài đặt tại cùng một thư mục sau: c:\tools\

Hiện tại demo chỉ hai công cụ, có thể tích hợp nhiều công cụ khác và khi đó chỉ cần thêm kịch bản chạy trong AutoIt.

III. Kịch bản trên Linux (analyze.sh)

III.1. Giải thích biến khởi tạo

Code:

Tên biến	Mục dích
ANALYSIS_DIR	Thư mục để chứa các report về malware, mỗi thư mục cấp dưới với md5 của malware là thư mục report về malware đã phân tích.
SHARED_FOLDER	Thư mực sử dụng để chia sẻ malware cần phân tích giữa mỗi trường Linux và Windows. Malware sẽ được copy đến đây và từ đây sẽ map sang môi trường Windows.
INETSIM_DIR	Thư mục cài đặt InetSim
VM_LOAD_TIMEOUT	Biến sử dụng để định thời gian load của máy ảo, tùy thuộc vào khả năng load của máy ảo có thể tăng hoặc giảm thời gian timeout, đơn vị thời gian là giây
MALWARE_RUNTIME	Là thời gian cho phép malware sẽ thực hiện trên máy ảo, đơn vị tính là giây
TIMEOUT	Số lượng thời gian mà script sẽ đợi cho việc dynamic analysis hoàn tất
...	Một số biến khác sẽ rõ trong quá trình giải thích chi tiết kịch bản.

III.2. Giải thích chi tiết kịch bản

III.2.1. Giải thích một số đoạn kịch bản ban đầu

Code:

```
#Kiểm tra nếu như đối số 1 rỗng (tức không chỉ định malware cần phân tích) hoặc file đó không tồn tại và không đọc được thì thông báo và thoát
if [ ! -n "$1" -o ! -r "$1" ]
then
        echo "Usage: 'basename $0' executable"
        exit
fi

if [ ! -d ${SHARED_FOLDER} ]
```

```
then
        mkdir -p ${SHARED_FOLDER}
fi

MALWARE="$1"
MD5=`md5sum ${MALWARE} | awk '{print $1}'`
# Malware sẽ được đặt trong thư mục $ANALYSIS_DIR/$MD5 của nó.
if [ -d ${ANALYSIS_DIR}/${MD5} ] ; then
        echo "${ANALYSIS_DIR}/${MD5} already exists. Exiting."
exit
fi
OUTDIR="${ANALYSIS_DIR}/${MD5}"
echo ${MALWARE} ${MD5} >> ${ANALYSIS_DIR}/records.txt
echo `date +"[%F %T]"` Starting analysis on ${MALWARE}.
echo `date +"[%F %T]"` Results will be placed in ${OUTDIR}
echo
mkdir ${OUTDIR}
REPORT=${OUTDIR}/${REPORT_NAME}
```

III.2.2. Giải thích phần Static Analysis

Trong kịch bản sử dụng dịch vụ của Team Cymru Hash Database. Ý nghĩa của dịch vụ này là gửi MD5 của malware cần phân tích đến dịch vụ và dịch vụ sẽ phản hồi lại: nếu như malware đó được được phát hiện thì nó sẽ hiển thị ngày giờ mà dịch vụ cập nhật và phần trăm phần mềm diệt virus diệt được nó.

Chi tiết về dịch vụ này có thể xem thêm tại:

http://www.team-cymru.org/Services/MHR/

Code:

```
# Static Analysis
echo -e "Analysis of ${MALWARE}\n" > ${REPORT}
echo "MD5 Hash: ${MD5}" >> ${REPORT}
echo "Team Cymru Hash Database:" >> ${REPORT}
whois -h hash.cymru.com ${MD5} >> ${REPORT}
# grab both ASCII and UNIcode strings from the sample
echo `date +"[%F %T]"` Running strings.
(strings -a -t x ${MALWARE}; strings -a -e l -t x ${MALWARE}) \
| sort > ${OUTDIR}/strings.txt
# run pecheck.py
echo `date +"[%F %T]"` Running pecheck.py.
python pecheck.py -d ${PEID_DB} ${MALWARE} > ${OUTDIR}/pecheck.txt
```

Tiếp theo là thực hiện grab các ASCII và UNION string có trong malware. Trong file thực thi của malware khi chúng ta thu thập các string này có thể giúp ích có một số thông tin về malware như : có thể địa chỉ url mà malware có thể kết nối đến, các tài nguyên mà malware lấy, … Tuy nhiên việc thu thập này chỉ là phần phụ, nhỏ bởi hầu như các malware hiện nay đều được packed, do vậy string sẽ xuất hiện dưới dạng string không tường minh.

Đoạn mã thực hiện python pecheck.py, pecheck.py là một chương trình Python được viết bởi Didier Stevens để xuất thông tin PE header từ malware. Thông tin này sẽ rất hữu ích bởi vì nó cho biết malware được biên dịch khi nào, các phân đoạn thực thi và các hàm được import vào. Hơn nữa nó còn so sánh với PeiD database

để biết được malware được pack bởi loại packer nào.

III.2.3. Khởi động InetSim

Code:

```
echo `date +"[%F %T]"` Starting InetSim.
CWD=`pwd`
mkdir -p ${OUTDIR}/inetsim
cd ${INETSIM_DIR}
sudo ./inetsim --session inetsim --config ${INETSIM_DIR}/conf/inetsim.conf \
--log-dir ${OUTDIR}/inetsim --report-dir ${OUTDIR} > /dev/null & cd ${CWD}
```

InetSim phải được chạy dưới quyền root, do vậy sử dụng lệnh sudo. Và ý nghĩa của các tham số như sau:
-session: chỉ định tên cho phiên làm việc.
-config: chỉ định nơi cấu hình được nạp.
-log-dir: chỉ định nơi log.
-report-dir: chỉ định nơi xuất ra report.
Sẽ có ba file log được tạo ra đó là debug.log, service.log và main.log. Ý nghĩa của main.log chỉ đề cập đến hoạt động của InetSim (những vấn đề liên quan đến hoạt động của InetSim). Phần chúng ta quan tâm đó là phần mà malware kết nối ra ngoài sẽ được lưu tại service.log.

III.2.4. Khởi động Tcpdump

Code:

```
echo `date +"[%F %T]"` Starting tcpdump.
sudo tcpdump -i vmnet1 -n -s 0 -w ${OUTDIR}/tcpdump.pcap &
TCPPID=`jobs -l | grep "sudo tcpdump" | awk '{ print $2 }'`
```

Tác dụng của TCPDump là một sniffer trên Linux, sử dụng nó để quan sát và ghi lại những hoạt động trao đổi của malware với môi trường ngoài, mà ở đây là Linux, do ta cấu hình Linux và Windows XP là Host-only, nên ta thực hiện sniff trên card vmnet1.

III.2.5. Khởi động VMWare

Code:

```
# Start up VMWare
# First we revert to our base snapshot
vmrun revertToSnapshot "/root/vmware/MalwareAnalysis/sandbox.vmx" base
# Then we start VMWare running
echo `date +"[%F %T]"` Starting VMWare.
vmrun start "/root/vmware/MalwareAnalysis/sandbox.vmx"
sleep ${VM_LOAD_TIMEOUT}
# Move the malware over to the sandbox
cp ${MALWARE} ${SHARED_FOLDER}/malware.exe
# Set up the share and execute the AutoIT script
echo `date +"[%F %T]"` Setting up network share.
winexe -U WORKGROUP/analysis%analysis --interactive=1 --system //192.168.32.2 'cmd /c net use z: \\192.168.32.1\malware'
echo `date +"[%F %T]"` Starting dynamic analysis script.
```

```
winexe -U WORKGROUP/analysis%analysis --interactive=1 --system //192.168.32.2 "c:\progra~1\autoit3\autoit3.exe
c:\tools\scripts\analyze.au3  z:\malware.exe z:\ ${MALWARE_RUNTIME}"
sleep ${MALWARE_RUNTIME}
LOOP=0
echo `date +"[%F %T]"` Starting check for finished file.
# Check for finished file - if not there, wait
while [ ! -f ${SHARED_FOLDER}/_analysis_finished ] ; do
        echo Checking...
        sleep ${TIMEOUT}
        LOOP=$(( $LOOP + 1 ))
        if [ ${LOOP} -gt 5 ] ; then
                echo `date +"[%F %T]"` ERROR: Sandbox is hung.
                break;
        fi
done
# Remove the share
echo `date +"[%F %T]"` Removing network share.
winexe -U WORKGROUP/analysis%analysis --interactive=1 --system //192.168.32.2 'cmd /c net use z: /delete'
# Stop the VMWare Image
echo `date +"[%F %T]"` Suspending VMWare.
vmrun suspend "/root/vmware/MalwareAnalysis/sandbox.vmx"
```

1) Trước khi thực hiện chạy kịch bản ta cần phải Take SnapShot VMWare 1 lần và đặt tên nó là base.

2) vmware revertToSnapShot "/root/vmware/MalwareAnalysis/sandbox.vmx" base sẽ đưa VMWare về trạng thái SnapShot mà chúng ta đã lưu. Cần phải làm như vậy bởi vì VMWare phải luôn đặt trong tình trạng chưa có bất kỳ malware nào thực hiện trước khi chép malware vào máy ảo phân tích.

3) Khởi động VMWare và copy malware cần phân tích vào thư mục shared trên Linux.

4) Sau khi khởi động xong máy ảo Windows XP SP2 (trước đó chúng ta đã tạo một user và pass là analysis) chúng ta sẽ thực hiện map thư mục chia sẻ trên Linux vào máy Windows XP. Lưu ý nếu trên Linux không thể chia sẻ được có thể do chưa cài dịch vụ Samba, do vậy cần phải cài dịch vụ Samba để Linux có thể chia sẻ file.

5) Khởi động script analysis.au3 trên Windows XP để chạy kịch bản phân tích malware tự động trên Windows XP.

6) Chờ đợi kết quả từ máy ảo Windows XP, việc chờ đợi sẽ kết thúc khi máy ảo Windows XP xuất file _analysis_finished.

7) Sau khi kết thúc kịch bản analysis.au3 trên Windows XP thì sẽ gỡ bỏ map ổ đĩa trên Windows XP và suppend máy ảo.

III.2.6. Thực hiện Volatility trên Memory

Code:

```
# Run Volatility on memory
echo `date +"[%F %T]"` Starting Volatility psscan2.
```

```
python /usr/local/src/Volatility-1.3_Beta/volatility psscan2 -f "/root/vmware/MalwareAnalysis/sandbox.vmem" \
> ${OUTDIR}/volatility-psscan.txt
echo `date +"[%F %T]"` Starting Volatility connscan2.
python /usr/local/src/Volatility-1.3_Beta/volatility connscan2 -f "/root/vmware/MalwareAnalysis/sandbox.vmem" \
> ${OUTDIR}/volatility-connscan2.txt
echo `date +"[%F %T]"` Starting Volatility dlllist.
python /usr/local/src/Volatility-1.3_Beta/volatility dlllist -f "/root/vmware/MalwareAnalysis/sandbox.vmem" \
> ${OUTDIR}/volatility-dlllist.txt
echo `date +"[%F %T]"` Starting Volatility modscan2.
python /usr/local/src/Volatility-1.3_Beta/volatility modscan2 -f "/root/vmware/MalwareAnalysis/sandbox.vmem" \
> ${OUTDIR}/volatility-modscan2.txt
# Move Results
echo `date +"[%F %T]"` Cleaning up.
mv ${SHARED_FOLDER}/* ${OUTDIR}
```

1) Sẽ thực hiện psscan2 module. Điều này rất hữu ích để phân tích các rootkit thường ẩn các tiến trình của malware trên hệ thống. Bằng việc truy vấn trực tiếp vào các danh sách tiến trình từ bộ nhớ, rootkit không thể ẩn tiến trình của chúng và người phân tích có thể quan sát một cách trực quan các tiến trình chạy trên hệ thống đã nhiễm malware.

2) Tiếp đến sẽ thực hiện truy vấn các kết nối trên hệ thống bị nhiễm với connscan2 module. Bởi vì những kết nối cũng có thể ẩn bởi rootkit, do vậy truy vấn trực tiếp lấy những kết nối mạng từ bộ nhớ sẽ cho phép người phân tích thấy được những kết nối trong hệ thống bị nhiễm malware.

3) Cuối cùng sẽ thực hiện xem thử những dll nào được nạp trong quá trình malware thực thi bằng cách sử dụng modscan2, bởi vì malware có thể tự nó nhiễm qua một tiến trình khác như là một DLL hoặc tự nó nạp, ...

III.2.7. Kịch bản kết thúc quá trình phân tích

Code:

```
# Stop tcpdump. Since its running as root we need to sudo to kill it
if [ ! -z ${TCPPID} ]; then
        sudo kill ${TCPPID}
fi
# Stop InetSim
if [ -f /var/run/inetsim.pid ] ; then
        INETPID=`cat /var/run/inetsim.pid`
        sudo kill ${INETPID} > /dev/null
        wait ${INETPID}
fi
# check to see if malware.exe is in the outdir - if so, delete it
if [ -f ${OUTDIR}/malware.exe ]; then
        rm -f ${OUTDIR}/malware.exe
fi
# Reset permissions on the files
sudo chown -R ${WHOAMI} ${OUTDIR}
echo `date +"[%F %T]"` Analysis finished.
```