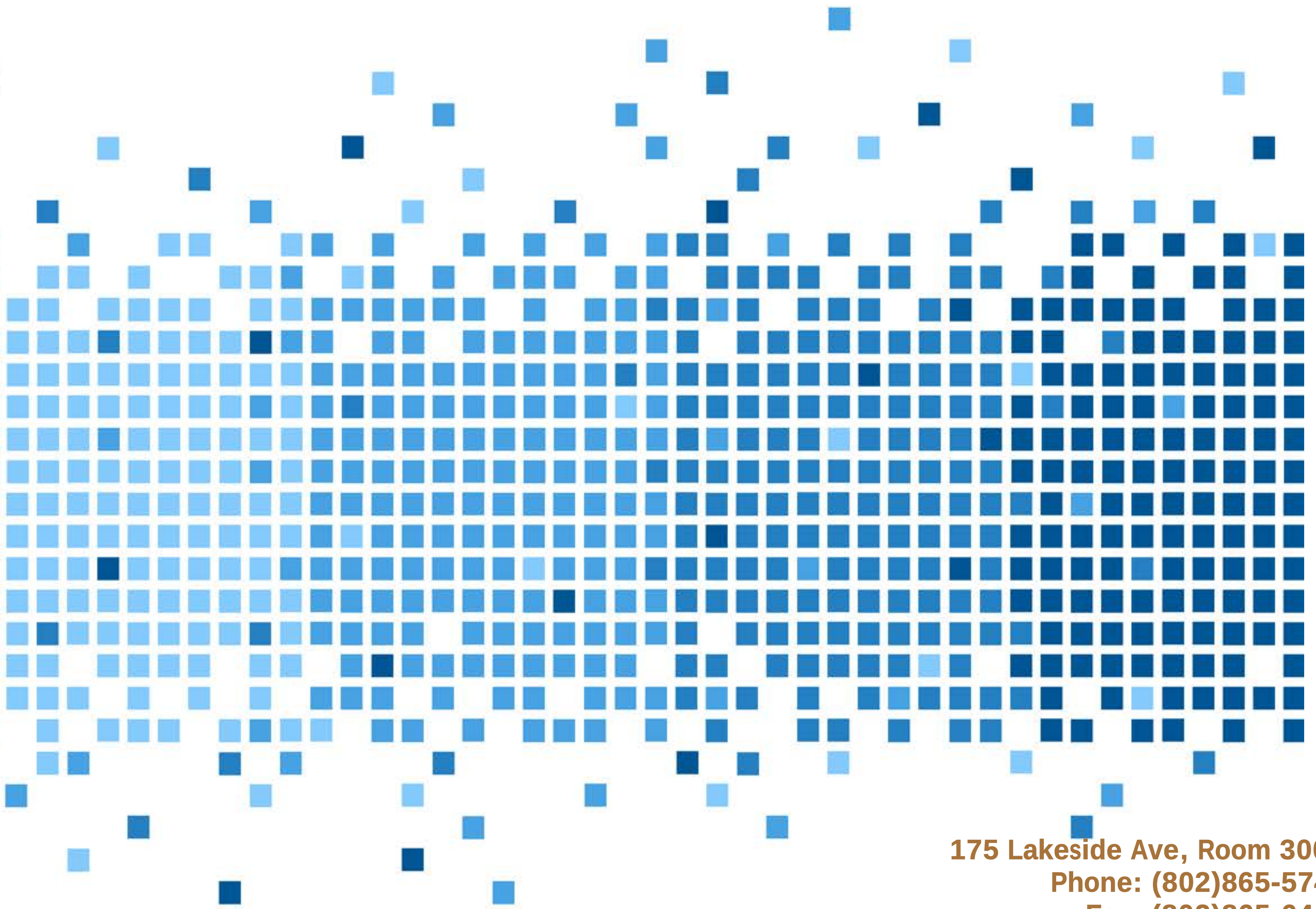
CHAMPLAIN COLLEGE | LCDi Leahy Center for Digital Investigation

[MALWARE ANALYSIS]



175 Lakeside Ave, Room 300A
Phone: (802)865-5744
Fax: (802)865-6446
www.lcdi.champlian.edu

02/04/2016

Contents

Introduction

Malware is the Swiss Army Knife of cybercrime, with capabilities ranging from compromising privileged information to infecting machines with network-crippling time bombs. Having a basic understanding of the varieties of malware and their effects is essential to becoming a competent cyber security professional or digital forensic investigator. The challenge of analyzing malicious software lies in doing so safely and securely, prioritizing repeatability and soundness of procedure. In order to facilitate this, the LCDI's Malware Analysis team has done work on a Cuckoo Sandbox: a closed, sterile system that runs malware in an isolated environment in order to conduct observation and documentation of its impact on a virtual machine. This experimentation will allow aspiring computer security professionals an opportunity to undertake and publish research to the community in a lab setting, opening doors for preliminary research and familiarization.

Background:

This project was initiated to foster a more detailed understanding of how malware operates. The team began by conducting Internet research on terms related to "malware analysis tools," "malware analysis sandbox," and "automated malware analysis." The searches lead us to free online sources like malwr.com, anubis.iseclab.org, zeltser.com, and cuckoosandbox.org. Subsequent searches for "cuckoo sandbox" lead the team to Cuckoo Sandbox, an open-source analysis environment, and more specifically Cuckoo Modified, which contains numerous bug-fixes and additional features. Prior to our study of Cuckoo, the LCDI was limited in how its analysts could safely and securely analyze malware. Our intent was to research, develop, and employ a local malware analysis environment at the LCDI in order to provide our analysts a known and tested malware analysis solution.

Purpose and Scope:

The primary purpose of the malware analysis project was to identify an investigative solution that could be used for future LCDI projects. The scope of the project was to ascertain whether a malware analysis system could be developed with the LCDI's existing equipment and infrastructure. Further, the team intended to explore a possible standard for the construction of such a system, as well as gathering a baseline level of information that could be determined from examining malware samples and other possible applications for this system. After reading through the documentation, blog material, and FAQs on sites such as Lenny Zeltser , FireEye Malware Analysis, and Cuckoo, addressing security concerns also became a focal point of the testing.

Research Questions:

1. Can a malware analysis system such as the Cuckoo Sandbox be safely and securely deployed on current computing assets?
2. Given a piece of malware, what type of information can be discovered using malware analysis?
3. Is the information that can be gleaned from Cuckoo Sandbox relevant in understanding malware?
4. Is there malware that the Cuckoo sandbox cannot analyze? How do we analyze it?

Terminology:

Sandbox – an isolated computing environment in which a program or file can be executed without affecting the application in which it runs.

Static Analysis – a term referring to when computer code is examined without executing the program in order to gain an understanding of the content and capability of the code. When static analysis is done by an automated tool, the code is parsed and identifiable content is reported in a human readable format.

Dynamic Analysis – testing and evaluating a computer program by executing it in real time in a controlled test environment. Executing the code allows the analyst to examine any visual and ephemeral effects caused by the code on the test environment.

Cuckoo – (refers to Cuckoo Sandbox and Cuckoo in general) an open-source automated malware analysis tool suite to be used in tandem with a virtual machine environment to analyze malware and output analyses into a database.

Cuckoo-Modified – A modified version of the Cuckoo sandbox that fixes a few inherent areas in which the program is lacking. A full debrief on this version can be found here: Accuvant.

Methodology and Methods

An isolated computer system was set up with forensic analysis tools such as Volatility (an open-source memory dump software suite), TCPDump (a command-line packet examiner), VirtualBox (a freeware VM host), Cuckoo, Yara (a tool designed to outline and classify malware), and WireShark (a cross-platform network protocol analyzer). Team members familiarized themselves with the tools as well as methodologies regarding malware analysis, such as memory analysis, static analyses like import and string analysis, and dynamic analyses like network packet sniffing. Some of the malware analysis tools and test environments explored included Malwr, Comodo Instant Malware Analysis, Anubis, and those listed at Lenny Zeltser. A methodology for investigating malware was developed both based on similarities in the above tools and focusing on the use of the Cuckoo Sandbox due to its versatility and accessibility. Each team member then tested malware using those methods in an effort to refine the malware investigative process.

Data pertaining to setting up a Cuckoo malware analysis environment, as well as data for creating a step-by-step guide on how to use the system, was collected. How-to guides for forensically wiping external media, setting up the required virtual machines, running the system, and backing up / resetting the system were also developed. The guides were developed with the LCDI lab in mind and are intended to standardize the methodologies used when investigating malware once it has been isolated. The data collected in these reports can be used in future malware research conducted by the LCDI.

Equipment Used

Table 1: Hardware requirements for our Cuckoo malware analysis system

Item	Identifier	Size/Specification
Host OS	Ubuntu	Ver 14.04 64-bit
CPU	i 7 – 3770k	8 cores @ 3.50GHz
Memory	DDR-3	32GB
Graphics	NVIDIA	GTX 650 TI
Hard Drive	SATA	1x 500GB, 3x 1TB (RAID 0)
Virtualization Software	VirtualBox	Ver 5.0.6
Coding Language	Python	Ver 2.7.6, 3.4.0
Remote Network Utility	SSH	Ver 0.91-ubuntu1
Analysis Software	Cuckoo Modified (Accuvant)	Ver 1.3

Memory Forensic Framework	Volatility	Ver 2.4
Malware Pattern Matcher	YARA	Ver 3.4.0
Network Analysis Tool	Wireshark	Ver 1.10.6 64-bit

**** Note: Working with malware on a production machine or a production network is dangerous and should only be conducted with permission from a relevant authority directly. No exceptions.****

Data Collection:

The suggested hardware requirements can be found in Appendix A. The recommended system specifications were determined with future malware analysis efforts in mind; specifically, a system with the ability to run multiple simultaneous Virtual Machines, separate storage for samples and results, and multiple analysis processes at once were taken into consideration.

Malware analysis data, at the time of publication, is collected by the Cuckoo system itself. An analysis report template for gathering analysis data is provided in Appendix B. The data fields of the report were determined by finding similarities between malware samples tested in Cuckoo. The data fields were also found to be similar to other web-based malware analysis environments. The data from manual and automated reports will be used for future analysis.

While collecting data, the team identified the need for guides that detail how to install Cuckoo from the ground up (Appendix C), how to image the system's hard drive, how to safely obtain and transport malware, and how to forensically wipe external media.

Analysis

Part of this project included safely and securely obtaining and transporting malware samples. This proved to be a serious concern because introducing malware of any kind to a production network is not acceptable. While the malware analysis system is advertised as secure, we have not proved this to be unanimously true. Due to these concerns, all malware samples were injected via the Virus Total web portal. No samples were introduced via external media or handled on any computer outside of the malware analysis environment as a security precaution.

The malware samples were injected into Cuckoo for processing. The reports generated by Cuckoo were the source of our analysis results. Per Appendix B, the types of data identified included, but were not limited to: file hash value, common name(s), MalScore, VirusTotal score, imported libraries, modified registry keys, and dropped files.

Results

At the time of publication, a working Cuckoo malware analysis environment was available. Replication of the construction process has not been conducted due to resource constraints and the priority to use the system for the furtherance of this project. It was determined that the Cuckoo system resets the VM after each malware test. The data containing the results of analysis are retained on the analysis system until it is removed by a full system wipe. While the steps taken to create the analysis system were recorded, a full wipe is not required to test multiple malware samples. The virtualization system is designed to reset after each sample analysis.

Also at the time of publication, the malware analysis system contained no known malware executables and was connected to the production network for administrative functions only. Permission to connect the system to the production network was received by LCDI Leadership after serious consideration and discussion. In an ideal situation, the Cuckoo system would be kept permanently offline. Leaving the system offline satisfies additional concerns regarding secure malware transportation that were addressed by using Virus Total and directly injecting the malware samples to Cuckoo.

Conclusion

The data identified in the Cuckoo analysis report is similar to reports provided by other web-based environments found at Malwr, Comodo Instant Malware Analysis, and Anubis. With the future in mind, our system and the data we obtained can be used to continue developing a malware capability within the LCDI. We have successfully identified and built a functional malware analysis environment at a minimal cost. We have identified common descriptors of known malware samples obtained from Virus Total (that had been identified by Virus Total as malicious) and have conceptually started our own database of malware data based on Cuckoo results.

Further Work

Future work should be done in verifying the security and practice of introducing malicious software to the analysis system. Additional future consideration should be given to work related to VM hardening and distributed malware analysis. The project could be appended by work related to the data obtained by automated malware analysis using our environment. Cuckoo is capable of advanced analysis. Our current system uses a basic Windows 7 64-bit VM with minimal applications installed within it. Future iterations of the analysis environment could include additional VM's of other OS's and/or virtual networks. With the conclusion of this project, LCDI employees can now use a local malware analysis system to identify common aspects of malware samples in a controlled environment.

References

"Attachments in Yahoo Mail." Helpcentral | SLN3390 -. Web. 10 Dec. 2015.

"CCleaner." RSS. Web. 10 Dec. 2015.

"Can I Export My Yahoo Mail Messages?" Helpcentral | SLN5033 -. Web. 10 Dec. 2015.

"Cuckoo Sandbox Book." Cuckoo Sandbox Book — Cuckoo Sandbox V2.0-dev Book. Web. 10 Dec. 2015.

"Darik's Boot And Nuke | Hard Drive Disk Wipe and Data Clearing." Darik's Boot And Nuke | Hard Drive Disk Wipe and Data Clearing. Web. 10 Dec. 2015.

"Guide to Malware Incident Prevention and Handling." Recommendations of the National Institute of Standards and Technology. Web. 10 Dec. 2015.

"How to Start Virtual Box Machines Automatically When Booting?" 12.04. Web. 10 Dec. 2015.

"Improving Reliability of Sandbox Results." Web. 10 Dec. 2015.

"Lenny Zeltser." Lenny Zeltser Content. Web. 10 Dec. 2015.

"Malware Analysis - Malware Forensics | FireEye." FireEye. Web. 10 Dec. 2015.

"Studying Spamming Botnets Using Botlab." Department of Computer Science & Engineering. University of Washington. Web. 10 Dec. 2015.

"VirusTotal - Free Online Virus, Malware and URL Scanner." VirusTotal - Free Online Virus, Malware and URL Scanner. Web. 10 Dec. 2015.

Appendix A: Malware Machine Recommended Hardware Components:

Table 2: Physical Hardware

Item	Identifier	Size/Specification
Host OS	Ubuntu	Ver 14.04 64-bit
CPU	Intel i7	8 cores
Memory	DDR-3	32GB
Hard Drive	SATA	1x 500GB (For OS), 1 + TB (For Storage)
Peripherals	CD Drive, USB ports, keyboard, mouse	Any
Network Connectivity	Default	Optional: none
Motherboard	ASRock Z77 Extreme4	Firmware: P2.80
CPU	i 7 – 3770k	8 cores @ 3.50GHz
Memory	Corsair Vengeance CMZ32GX3M4A1600C9	32GB (4x8GB)
Graphics	NVIDIA	GTX 650 TI
Hard Drive	SATA	1x 500GB, 3x 1TB (RAID 0)
Hard Drive Specifications	Western Digital Black	1001, 1002, 1002, 5003

Table 3: Windows VM

Item	Identifier	Size/Specification
Host OS	Windows	7 64-bit
CPU	Virtualized	1 core
Memory	Virtualized	4GB
Hard Drive	Virtualized	25GB
Required Software	Python	2.7
Optional Software	Adobe Reader, Google Chrome, Firefox, LibreOffice	Any
Network Connectivity	default	Through VirtualBox

Table 4: Host Settings

Item	Identifier	Size/Specification
Host OS	Ubuntu	Ver 14.04 64-bit
Kernel	Linux	3.19.0-30-generic
Software	see Appendix C	see Appendix C
Peripherals	CD Drive, USB ports, keyboard, mouse	Any
Network Connectivity	default	Built in

Appendix B: Malware Analysis Report

Examiner Name		Date/Time of Analysis
File Hash (MD5)		File Type
MalScore	VT Score	PEiD Signature / Magic Number
Imported Libraries		Entropy
Compile Time	Section Names	
Original File Name	Original Description	

Overview of Screenshots: ____________________

Overview of Modified Registry Keys: ____________________

Overview of Dropped Files: ____________________

Overview of Strings output: ____________________

Overview of Antivirus Identifications:

Appendix C: Commands Run on Malware Machine

Software & Dependencies Installation

```
sudo apt-get install gparted mdadm git libtool autoconf byacc htop openssh-server vim python-dev libfuzzy-dev flask wireshark traceroute python-sqlalchemy python-bson python-dpkt python-jinja2 python-magic python-pymongo python-gridfs python-libvirt python-bottle python-pefile python-chardet swig libssl-dev clamav-daemon python-geoip geoip-database mono-utils wkhtmltopdf xvfb xfonts-100dpi
sudo pip install jinja2 pymongo bottle pefile django chardet pygal m2crypto clamd django-ratelimit pycrypto weasyprint rarfile jsbeautifier
```

Clone needed git repositories

```
git clone https://github.com/brad-accuvant/cuckoo-modified.git
git clone https://github.com/gdabah/distorm.git
git clone https://github.com/kbandla/pydeep.git
git clone https://github.com/volatilityfoundation/volatility.git
```

Install Cuckoo Requirements & Dependencies

```
sudo apt-get install python python-pip
cd cuckoo-modified/
sudo pip install -r requirements.txt
python /utils/community.py --force --rewrite --all
```

Install Pydeep

```
sudo pip install pydeep
```

Install Yara (Download first)

```
cd yara-3.4.0/
libtoolize --force
aclocal
autoheader
automake --force-missing --add-missing
autoconf
./configure
make
sudo make install
```

Install Virtualbox

```
sudo sh -c "echo 'deb http://download.virtualbox.org/virtualbox/debian '$(lsb_release -cs)' contrib non-free' >
/etc/apt/sources.list.d/virtualbox.list" && wget -q
http://download.virtualbox.org/virtualbox/debian/oracle_vbox.asc -O- | sudo apt-key add - && sudo apt-get
update && sudo apt-get install virtualbox-5.0
```

Install TCPDump

```
sudo apt-get install tcpdump
sudo apt-get install libcap2-bin
sudo setcap cap_net_raw,cap_net_admin=eip /usr/sbin/tcpdump
getcap /usr/sbin/tcpdump
        Should return "/usr/sbin/tcpdump = cap_net_admin,cap_net_raw+eip"
```

Install Volatility

```
cd volatility/
sudo python setup.py build
sudo python setup.py install
```

Make a cuckoo user

```
sudo adduser cuckoo
        password: bawk
        confirm: bawk
        press Enter until complete (go with default options)
sudo usermod -a -G vboxusers cuckoo
```

Install distorm3

```
cd distorm/
python setup.py build
sudo python setup.py install
```

Set up Host-Only Networking

In VirtualBox

- Create Host-Only Network under File->Preferences->Network
- Keep Default settings

In Guest

- Configure network settings
  - Static IP -
  - DNS - any DNS server
  - Default Gateway -
- run ‘netsh winsock reset’ in cmd
- restart and take snapshot

In Terminal on Host

```
sudo iptables -A FORWARD -o eth0 -i vboxnet0 -s 192.168.56.0/24 -m conntrack --ctstate NEW -j ACCEPT
```

sudo iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
sudo iptables -A POSTROUTING -t nat -j MASQUERADE

Alias generation

- Open .bash_aliases in text editor of choice
- Add the following lines:
    - alias cuckoo-start='python ~/cuckoo-modified/cuckoo.py'
    - alias cuckoo-submit='python ~/cuckoo-modified/utils/submit.py'
    - alias cuckoo-runapi='./cuckoo-modified/utils/api.py'
    - alias cuckoo-runserver='cd ~/cuckoo-modified/web && python manage.py runserver 0.0.0.0:8080 ; cd'
    - alias cuckoo-process='./cuckoo-modified/utils/process.py'
- Save the file and restart any open terminals

Having processes run on boot

(#created link to VM in /home/user/.config/autostart/ folder#)

- Create a file in the /home/user/.config/autostart/ directory
    - 'sudo nano cuckooStart.bash
- Add the following code to the file to allow processes to run at boot:
- ##http://askubuntu.com/questions/404665/how-to-start-virtual-box-machines-automatically-when-booting## (reference for getting VMs to run at boot)
- #!/bin/bash
- cd ~/cuckoo-modified/web && python manage.py runserver 0.0.0.0:8080 &
- cd &
- sleep 2
- firefox -new-tab -url XXX.XXX.XXX.XXX:8080 &
- sleep 2
- python ~/cuckoo-modified/cuckoo.py