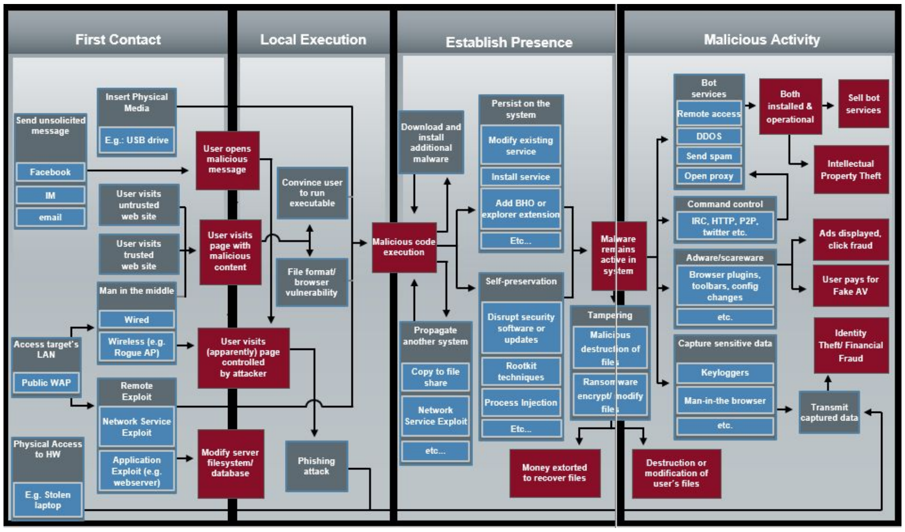
Writeup 3
CS 373
James Tyler Ball

Malware Defenses Lesson 1 (Craig Schmugar):

Happy99 is the first “big” worm to hit the public. This sparked interest in cybersecurity.

The lectures will examine how malware operates, what kind of defenses can be established, and how anti-malware is written.


First Contact
Local Execution
Establish Presence
Malicious Activity
Send unsolicited message
Facebook
IM
email
Insert Physical Media
E.g.: USB drive
User visits untrusted web site
User visits trusted web site
Man in the middle
Wired
Wireless (e.g. Rogue AP)
Access target's LAN
Public WAP
Remote Exploit
Network Service Exploit
Application Exploit (e.g. webserver)
Physical Access to HW
E.g. Stolen laptop
User opens malicious message
User visits page with malicious content
User visits (apparently) page controlled by attacker
Modify server filesystem/ database
Convince user to run executable
File format/ browser vulnerability
Phishing attack
Download and install additional malware
Malicious code execution
Propagate another system
Copy to file share
Network Service Exploit
etc...
Persist on the system
Modify existing service
Install service
Add BHO or explorer extension
Etc...
Self-preservation
Disrupt security software or updates
Rootkit techniques
Process Injection
Etc...
Malware remains active in system
Tampering
Malicious destruction of files
Ransomware encrypt/ modify files
Money extorted to recover files
Destruction or modification of user's files
Bot services
Remote access
DDOS
Send spam
Open proxy
Command control
IRC, HTTP, P2P, twitter etc.
Adware/scareware
Browser plugins, toolbars, config changes
etc.
Capture sensitive data
Keyloggers
Man-in-the browser
etc.
Both installed & operational
Sell bot services
Intellectual Property Theft
Ads displayed, click fraud
User pays for Fake AV
Identity Theft/ Financial Fraud
Transmit captured data

The big takeaway from this graph is the four large steps Contact, Local Execution, Establish Presence, and Malicious Activity

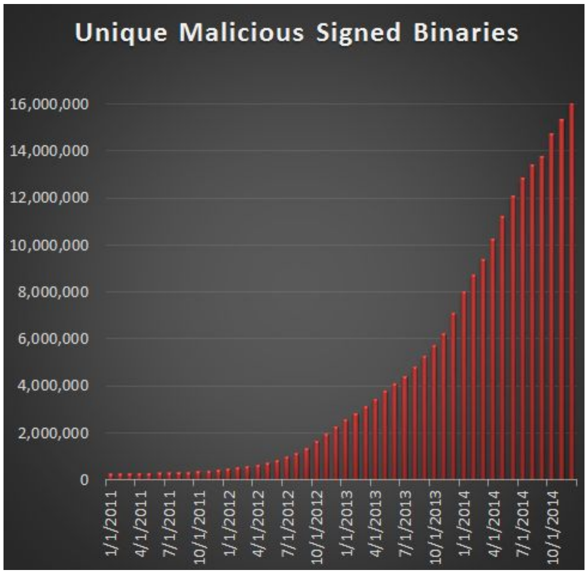
There are different ways that malware can be introduced to a system:

- ad networks
- emails
- direct attacks
- formerly instant messaging
- poisoned search results
- Watering Hole
- Physical Access

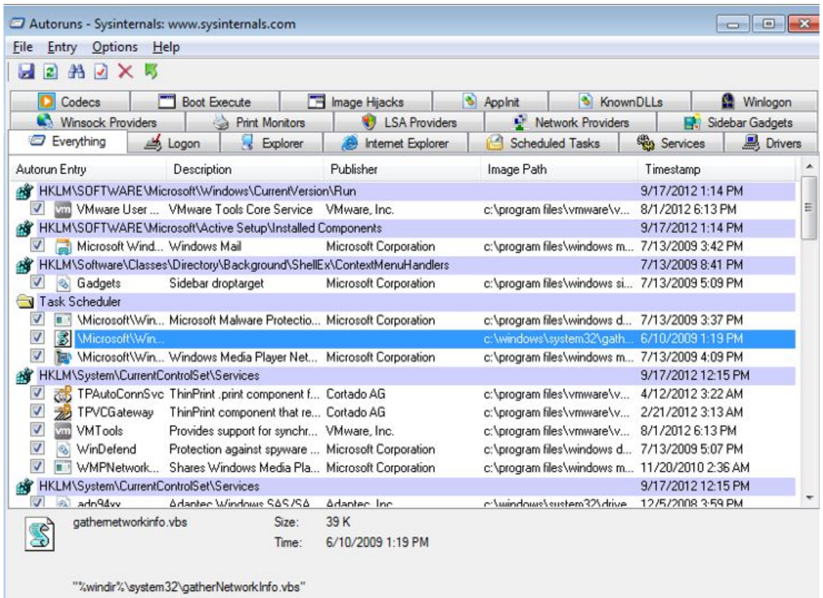
Next the code has to activate to impact the potential victim. This can be done in a number of ways:

- Autorun
- Social Engineering or fooling a user into running it
- Burying aspects into a EULA
- User clicks a malicious ad
- Exploitation feature

Third step is to establish a presence. This often involves avoiding detection by trying to appear legitimate:

- using filenames similar to the OS filenames
- changing timestamps to look like they’ve been there longer than they have
- being signed (which is becoming more common over time)
    - 

- bootkits and rootkits
    - modifying records to execute malicious code
- Note that the software also tries to remain in the system through reboots
    - All of the ways displayed below are methods that software can use to link to the startup of a Windows OS

  - 

Autorun Entry	Description	Publisher	Image Path	Timestamp
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				9/17/2012 1:14 PM
☑ VMware User ...	VMware Tools Core Service	VMware, Inc.	c:\program files\vmware\v...	8/1/2012 6:13 PM
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components				9/17/2012 1:14 PM
☑ Microsoft Wind...	Windows Mail	Microsoft Corporation	c:\program files\windows m...	7/13/2009 3:42 PM
HKLM\Software\Classes\Directory\Background\ShellEx\ContextMenuHandlers				7/13/2009 8:41 PM
☑ Gadgets	Sidebar droptarget	Microsoft Corporation	c:\program files\windows si...	7/13/2009 5:09 PM
Task Scheduler				
☑ \Microsoft\Win...	Microsoft Malware Protectio...	Microsoft Corporation	c:\program files\windows d...	7/13/2009 3:37 PM
☑ \Microsoft\Win...			c:\windows\system32\gath...	6/10/2009 1:19 PM
☑ \Microsoft\Win...	Windows Media Player Net...	Microsoft Corporation	c:\program files\windows m...	7/13/2009 4:09 PM
HKLM\System\CurrentControlSet\Services				9/17/2012 12:15 PM
☑ TPAutoConnSvc	ThinPrint .print component f...	Cortado AG	c:\program files\vmware\v...	4/12/2012 3:22 AM
☑ TPVCGateway	ThinPrint component that re...	Cortado AG	c:\program files\vmware\v...	2/21/2012 3:13 AM
☑ VMTools	Provides support for synchr...	VMware, Inc.	c:\program files\vmware\v...	8/1/2012 6:13 PM
☑ WinDefend	Protection against spyware ...	Microsoft Corporation	c:\program files\windows d...	7/13/2009 5:07 PM
☑ WMPNetwork...	Shares Windows Media Pla...	Microsoft Corporation	c:\program files\windows m...	11/20/2010 2:36 AM
HKLM\System\CurrentControlSet\Services				9/17/2012 12:15 PM
☑ adp94xx	Adaptec Windows SAS/SA...	Adaptec, Inc.	c:\windows\system32\drive...	12/5/2008 3:59 PM

gathernetworkinfo.vbs Size: 39 K
Time: 6/10/2009 1:19 PM

"%windir%\system32\gatherNetworkInfo.vbs"

- More recent methods involve side loading (replacing legitimate libraries), proxy configurations to modify the network to provide an IP which contains malware
- Buffer overflow exploits in otherwise legitimate programs can modify the software

Finally the malware executes

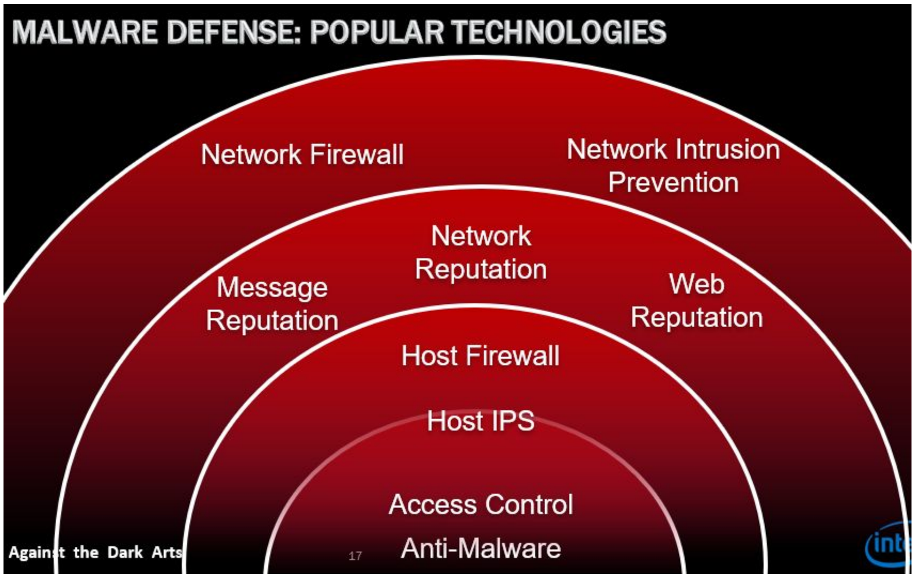
  - It may try to gather information on the victim to seek valuable documents, passwords
    - It can do this via keyloggers of screen captures or a number of other methods

Anti-malware defenders need to use a variety of methods to protect their systems

- educate their users
- browsers can blacklist sites
- new systems are often sold with updated software
- script blockers
- stricter criteria for OS or browsers
- firewall

  **** The lesson to learn here is that it's hard to build a single method that blocks all possible “points of entry” without blocking too much

  - the result is layered security:

- 

MALWARE DEFENSE: POPULAR TECHNOLOGIES

Network Firewall

Network Intrusion Prevention

Network Reputation

Message Reputation

Web Reputation

Host Firewall

Host IPS

Access Control

Anti-Malware

Against the Dark Arts

17

- other developed nations are generally better at using two-factor authorization

Regarding the image above, the outer layers are more common and from what I can see also more distant from the end-user’s experience (network firewalls are not often modified by the majority of users in a company).

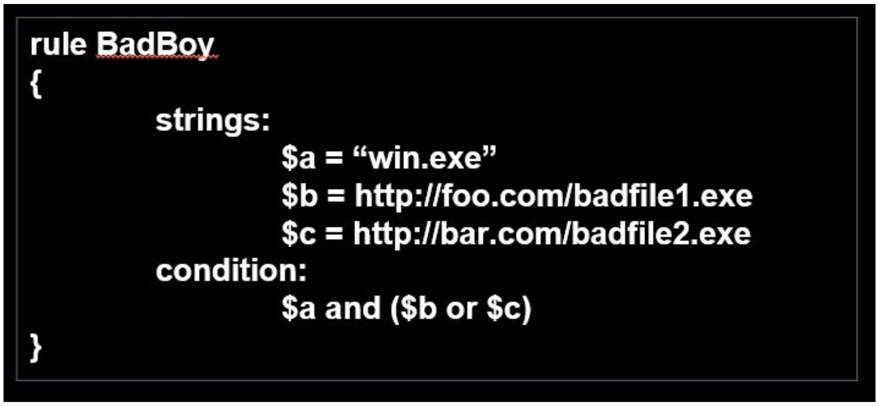
Backups are an effective method of combating ransomware attacks. You can just ignore the encrypted files and rely on your backups. The key is convincing clients to actively take defensive measures before they encounter a problem, which is often hard to do.

From an investigative and defensive standpoint, this multi-layered defense also means that each layer or component of the defensive network can provide different pieces of information to better assess a threat.

Malware scanners have (not surprisingly) changed over time. They used to be on demand from a floppy disk (scan my computer, go!). Scanners are also more specialized now (some only look at rootkits, some only scan scripts).

Again Professor Schmugar talks about the dangers on desensitizing users to alerts (warning about every cookie or script).

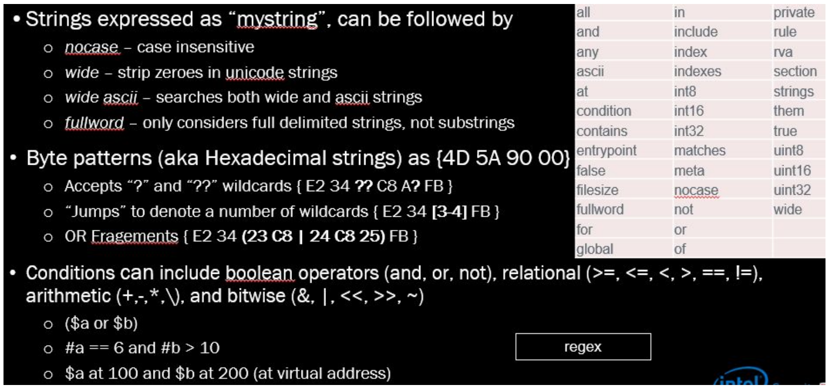
Now we hit YARA - “the pattern matching swiss army knife for malware researchers”



```
rule BadBoy
{
        strings:
                $a = “win.exe”
                $b = http://foo.com/badfile1.exe
                $c = http://bar.com/badfile2.exe
        condition:
                $a and ($b or $c)
}
```

Users wanted to write their own scripts for scanning, but anti-malware software was often closed, so YARA was created.

- yara is meant to be simple
- quick process from writing to scanning
- has some basic rules:

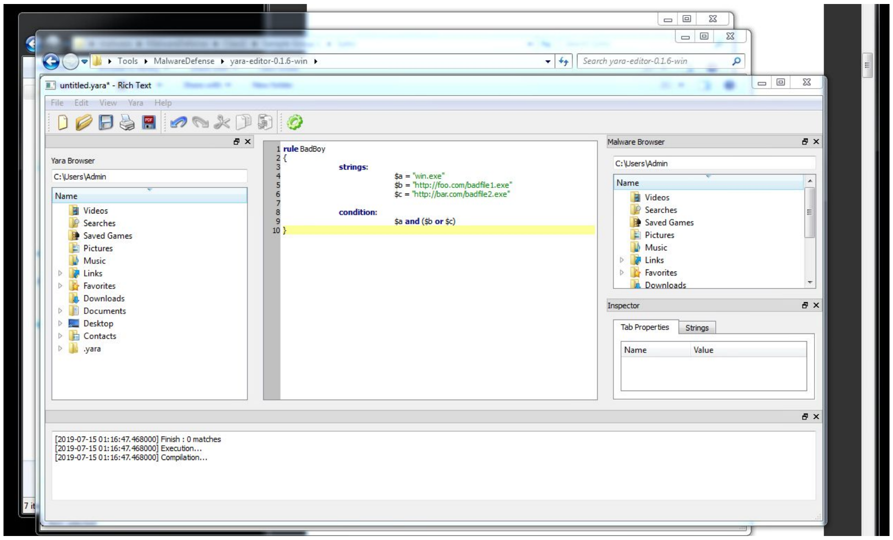


- Strings expressed as “mystring”, can be followed by
  - *nocase* – case insensitive
  - *wide* – strip zeroes in unicode strings
  - *wide ascii* – searches both wide and ascii strings
  - *fullword* – only considers full delimited strings, not substrings
- Byte patterns (aka Hexadecimal strings) as {4D 5A 90 00}
  - Accepts “?” and “??” wildcards { E2 34 **??** C8 A**?** FB }
  - “Jumps” to denote a number of wildcards { E2 34 **[3-4]** FB }
  - OR Fragements { E2 34 **(23 C8 | 24 C8 25)** FB }

all	in	private
and	include	rule
any	index	rva
ascii	indexes	section
at	int8	strings
condition	int16	them
contains	int32	true
entrypoint	matches	uint8
false	meta	uint16
filesize	nocase	uint32
fullword	not	wide
for	or	
global	of	

- Conditions can include boolean operators (and, or, not), relational (>=, <=, <, >, ==, !=), arithmetic (+,-,*,\), and bitwise (&, |, <<, >>, ~)
  - ($a or $b)
  - #a == 6 and #b > 10
  - $a at 100 and $b at 200 (at virtual address)

The next slide gave some loose instructions for using Yara. After booting up the cmd in the VM I followed the path, I was able to launch the yara editor. From there I ran the sample program:



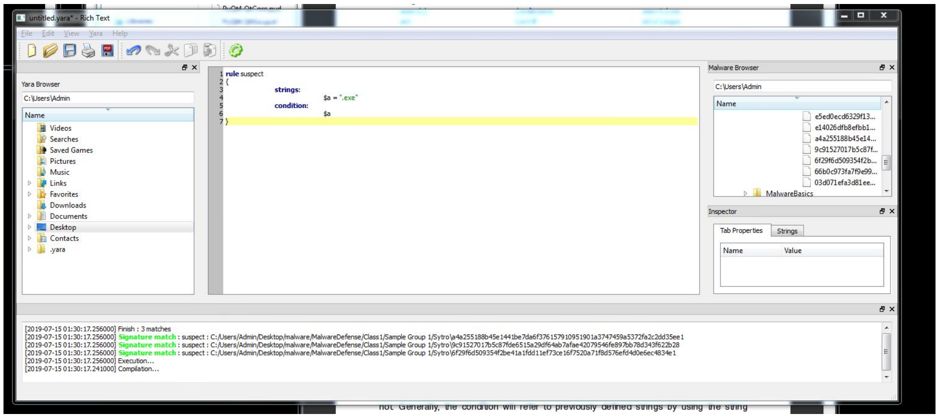
```
rule BadBoy
{
            strings:
                        $a = "win.exe"
                        $b = "http://foo.com/badfile1.exe"
                        $c = "http://bar.com/badfile2.exe"

            condition:
                        $a and ($b or $c)
}
```

```
[2019-07-15 01:16:47.468000] Finish : 0 matches
[2019-07-15 01:16:47.468000] Execution...
[2019-07-15 01:16:47.468000] Compilation...
```

Obviously this was run in the wrong place, but at least I had an idea of how the yara-editor worked.

Next I looked at Sample Group 1 and did the simplest search I could think of:



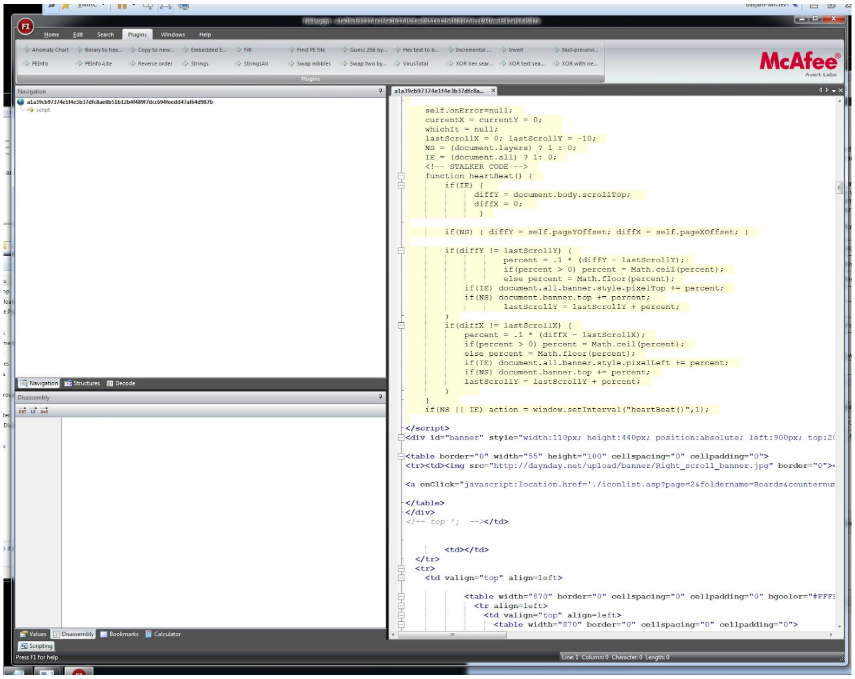
```
rule suspect
{
            strings:
                        $a = ".exe"
            condition:
                        $a
}
```

```
[2019-07-15 01:30:17.256000] Finish : 3 matches
[2019-07-15 01:30:17.256000] Signature match : suspect : C:/Users/Admin/Desktop/malware/MalwareDefense/Class1/Sample Group 1/Sytro\a4a255188b45e1441be7da6f37615791095190 1a3747459a5372fa2c2dd35ee1
[2019-07-15 01:30:17.256000] Signature match : suspect : C:/Users/Admin/Desktop/malware/MalwareDefense/Class1/Sample Group 1/Sytro\9c91527017b5c87fde6515a29df64ab7afae42079546fe897bb78d343f622b28
[2019-07-15 01:30:17.256000] Signature match : suspect : C:/Users/Admin/Desktop/malware/MalwareDefense/Class1/Sample Group 1/Sytro\6f29f6d509354f2be41a1fdd11ef73ce16f7520a71f8d576efd4d0e6ec4834e1
[2019-07-15 01:30:17.256000] Execution...
[2019-07-15 01:30:17.241000] Compilation...
```

Searching for the string “.exe returned three of the files in the target location. Right away these are the files I am most suspicious of.

Yara can be used to scan for files that do a combination of things (create executables and scans email, for example). Yara can also be used in combination with string extraction

(FileInsight) to find good targets to look for:



```
    self.onError=null;
    currentX = currentY = 0;
    whichIt = null;
    lastScrollX = 0; lastScrollY = -10;
    NS = (document.layers) ? 1 : 0;
    IE = (document.all) ? 1: 0;
    <!-- STALKER CODE -->
    function heartBeat() {
        if(IE) {
            diffY = document.body.scrollTop;
            diffX = 0;
             }

        if(NS) { diffY = self.pageYOffset; diffX = self.pageXOffset; }

        if(diffY != lastScrollY) {
                percent = .1 * (diffY - lastScrollY);
                if(percent > 0) percent = Math.ceil(percent);
                else percent = Math.floor(percent);
            if(IE) document.all.banner.style.pixelTop += percent;
            if(NS) document.banner.top += percent;
                lastScrollY = lastScrollY + percent;
        }
        if(diffX != lastScrollX) {
            percent = .1 * (diffX - lastScrollX);
            if(percent > 0) percent = Math.ceil(percent);
            else percent = Math.floor(percent);
            if(IE) document.all.banner.style.pixelLeft += percent;
            if(NS) document.banner.top += percent;
            lastScrollY = lastScrollY + percent;
        }
    }
    if(NS || IE) action = window.setInterval("heartBeat()",1);

</script>
<div id="banner" style="width:110px; height:440px; position:absolute; left:900px; top:2(

<table border="0" width="55" height="100" cellspacing="0" cellpadding="0">
<tr><td><img src="http://dayday.net/upload/banner/Right_scroll_banner.jpg" border="0"><

<a onClick="javascript:location.href='./iconlist.asp?page=2&foldername=Boards&counternun

</table>
</div>
<!-- top ³;  --></td>


        <td></td>
  </tr>
  <tr>
    <td valign="top" align=left>

        <table width="870" border="0" cellspacing="0" cellpadding="0" bgcolor="#FFFI
          <tr align=left>
            <td valign="top" align=left>
              <table width="870" border="0" cellspacing="0" cellpadding="0">
```

The debrief session that followed discuss that while the goal is to automate the process of defining malware, this is not always a quick process. A tool like yara can be used to do quick rough analysis of files/folders to reveal select suspicious criteria.

Again the professor stressed the challenges of an industry that faces ~500k pieces of new malware everyday.

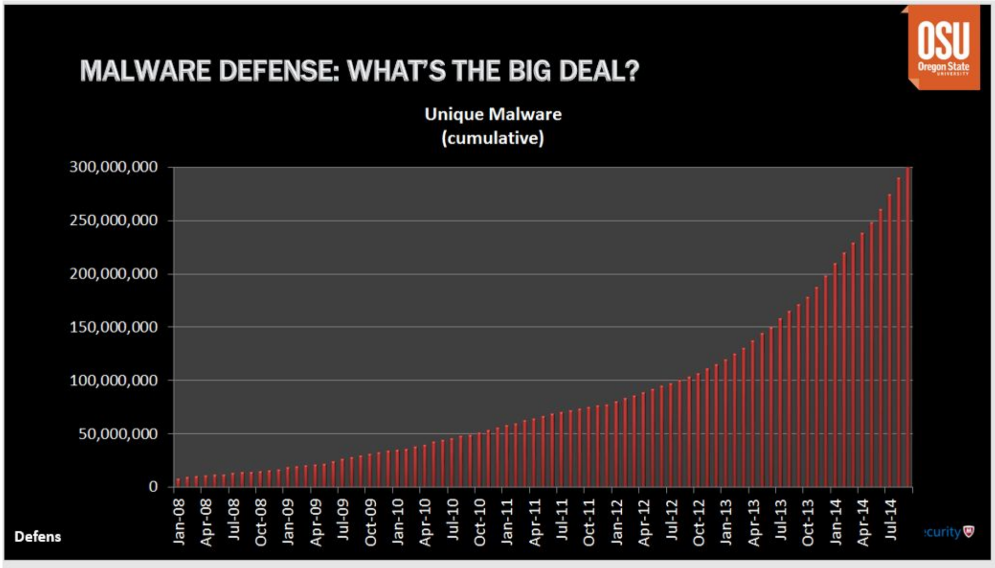
Malware Defenses Lesson 2 (Craig Schmugar):

I hadn't thought about this strategy, but Professor Schmugar talked about how once you identify a signature attached to known malware, you can check that signature against potentially millions of unknown files to identify potential threats.

Often malware investigators are handed vast numbers of files that seem to do similar things or have questionable credentials. The question was brought up what is the point of signatures if they can't be trusted? Sometimes credentials are revoked. Ultimately the "signers" are not police. Although keys are meant to establish trust, they are losing their authenticity.

Consensus within the industry is that while signatures are not dead or going away, they definitely require additional measures.

Malware continues to grow in popularity:



MALWARE DEFENSE: WHAT'S THE BIG DEAL?

Why is it growing so fast? Defense is working. New malware is required to overcome rapidly expanding defenses.

In the last set of lectures, it was mentioned that one big goal of malware analysis is to automate as much as possible. This has a number of advantages:

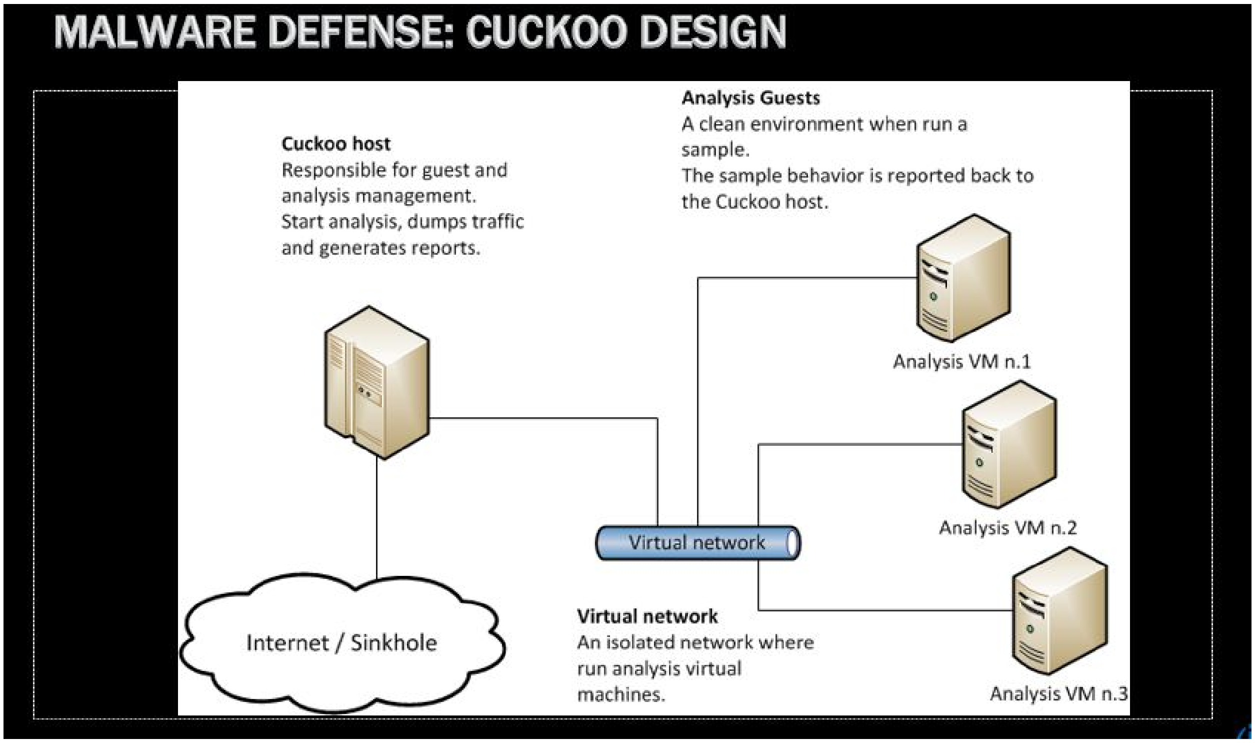
- Scale
- Speed (go through more files)
- computers are really good at comparing files
- precision

- computers can work 24/7

There are also disadvantages:

- computers aren't always good at dealing with new problems
- computers are prone to attacks and evasive tactics - malware can change the payload when it detects it's on a VM
- out of context - testing done by a machine is not the real world

CUCKOO - biggest automated analysis that's open source



MALWARE DEFENSE: CUCKOO DESIGN

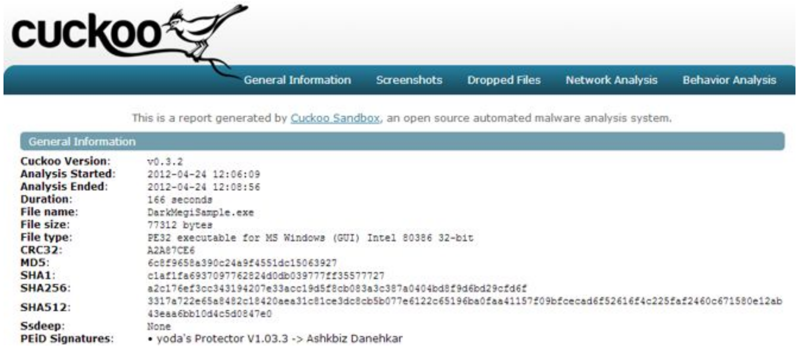
Cuckoo host
Responsible for guest and analysis management.
Start analysis, dumps traffic and generates reports.

Analysis Guests
A clean environment when run a sample.
The sample behavior is reported back to the Cuckoo host.

Virtual network
An isolated network where run analysis virtual machines.

multiple vms can run analysis and give reports

Cuckoo looks at signatures, does a static analysis of strings, takes memory dumps, does network analysis, takes screenshots, and gives a summary:



This is a report generated by Cuckoo Sandbox, an open source automated malware analysis system.

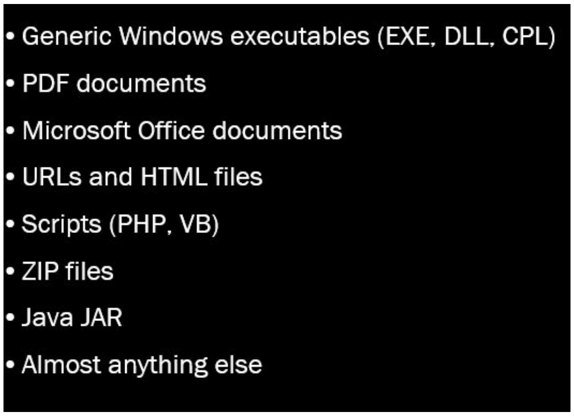
General Information

Cuckoo Version:	v0.3.2
Analysis Started:	2012-04-24 12:06:09
Analysis Ended:	2012-04-24 12:08:56
Duration:	166 seconds
File name:	DarkMegiSample.exe
File size:	77312 bytes
File type:	PE32 executable for MS Windows (GUI) Intel 80386 32-bit
CRC32:	A2A87CE6
MD5:	6c8f9658a390c24a9f4551dc15063927
SHA1:	c1af1fa693709776 2824d0db039777ff35577727
SHA256:	a2c176ef3cc343194207e33acc19d5f8cb083a3c387a0404bd8f9d6bd29cfd6f
SHA512:	3317a722e65a8482c18420aea31c81ce3dc8cb5b077e6122c65196ba0faa41157f09bfcecad6f52616f4c225faf2460c671580e12ab 43eaa6bb10d4c5d0847e0
Ssdeep:	None
PEiD Signatures:	• yoda's Protector V1.03.3 -> Ashkbiz Danehkar

Again we touched on the risks of bringing more tools to the VM also increases the chances that malware will detect it's on a virtual machine.

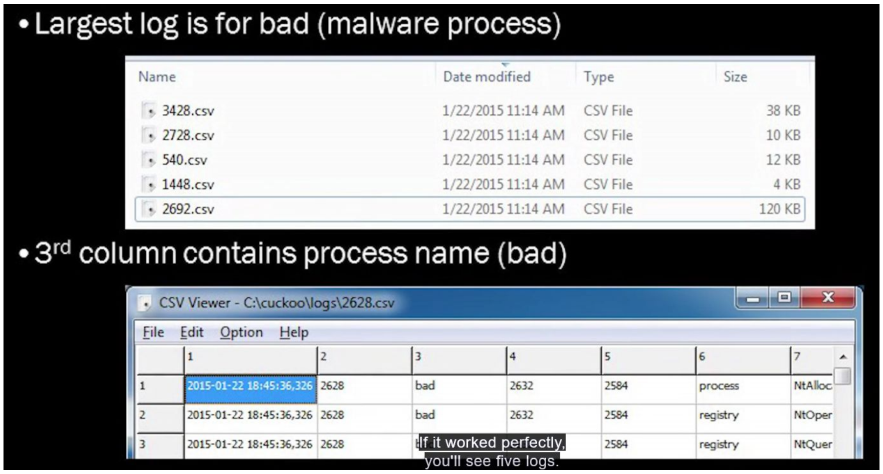
So why is malware so scared of a VM? There aren't enough people using VMs for valuable operations that an attacker would view full execution on a VM a higher priority than evasion.

Cuckoo's behavior analysis is valuable. It can tell us when files were created, the PIDs, files being copied, locations of these occurrences, all with timestamps. It also handles a number of different formats:



- Generic Windows executables (EXE, DLL, CPL)
- PDF documents
- Microsoft Office documents
- URLs and HTML files
- Scripts (PHP, VB)
- ZIP files
- Java JAR
- Almost anything else

The next process the class went through was working with Cuckoo in the VM. The process was as straightforward as "run this, then that, wait for this window" and there seemed to be a large number of students (including myself) who encountered crashes or errors. Had it worked, there would have been 5 logs:



- Largest log is for bad (malware process)

Name	Date modified	Type	Size
3428.csv	1/22/2015 11:14 AM	CSV File	38 KB
2728.csv	1/22/2015 11:14 AM	CSV File	10 KB
540.csv	1/22/2015 11:14 AM	CSV File	12 KB
1448.csv	1/22/2015 11:14 AM	CSV File	4 KB
2692.csv	1/22/2015 11:14 AM	CSV File	120 KB

- 3rd column contains process name (bad)

CSV Viewer - C:\cuckoo\logs\2628.csv

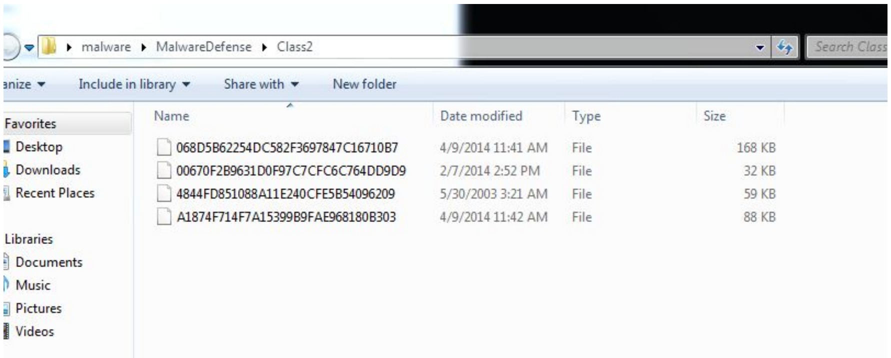
	1	2	3	4	5	6	7
1	2015-01-22 18:45:36,326	2628	bad	2632	2584	process	NtAlloc
2	2015-01-22 18:45:36,326	2628	bad	2632	2584	registry	NtOper
3	2015-01-22 18:45:36,326	2628	[illegible]	[illegible]	2584	registry	NtQuer

If it worked perfectly,
you'll see five logs.

- note that the default sorting in cuckoo is to list the events as they occurred chronologically
- “cmd/c” is an instruction to run the cmd prompt with the following command

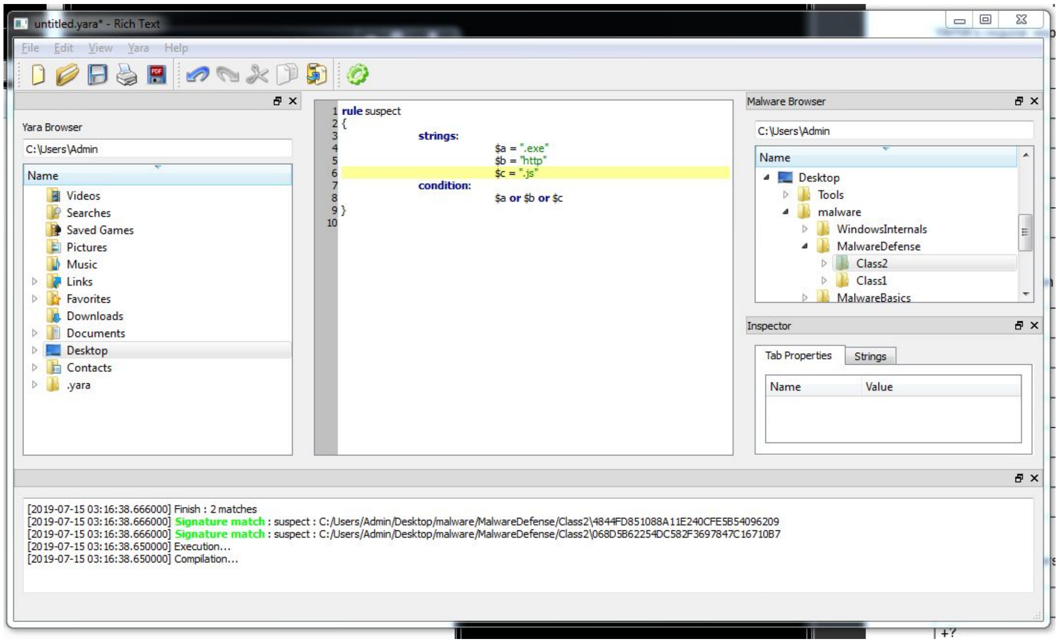
CLASS2 SAMPLES

In the desktop/malware/malwaredefense/class2 folder there are the following samples:



Name	Date modified	Type	Size
068D5B62254DC582F3697847C16710B7	4/9/2014 11:41 AM	File	168 KB
00670F2B9631D0F97C7CFC6C764DD9D9	2/7/2014 2:52 PM	File	32 KB
4844FD851088A11E240CFE5B54096209	5/30/2003 3:21 AM	File	59 KB
A1874F714F7A15399B9FAE968180B303	4/9/2014 11:42 AM	File	88 KB

To try and figure out which are malicious, the first thing I'm going to try is search the files for known strings that we've identified as subject for suspicion:

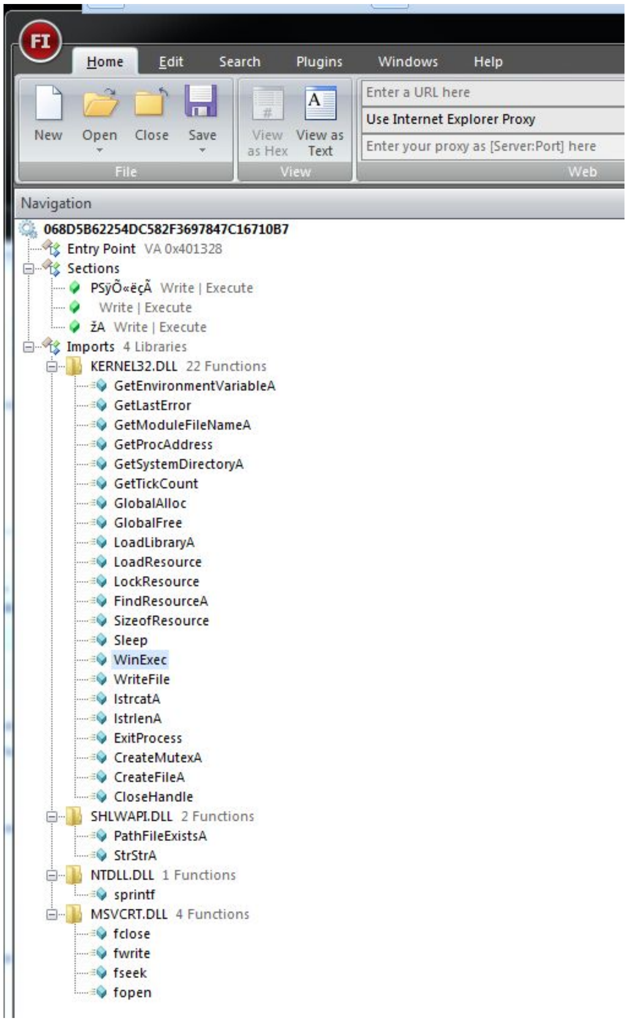


```
rule suspect
{
		strings:
				$a = ".exe"
				$b = "http"
				$c = ".js"
		condition:
				$a or $b or $c
}
```

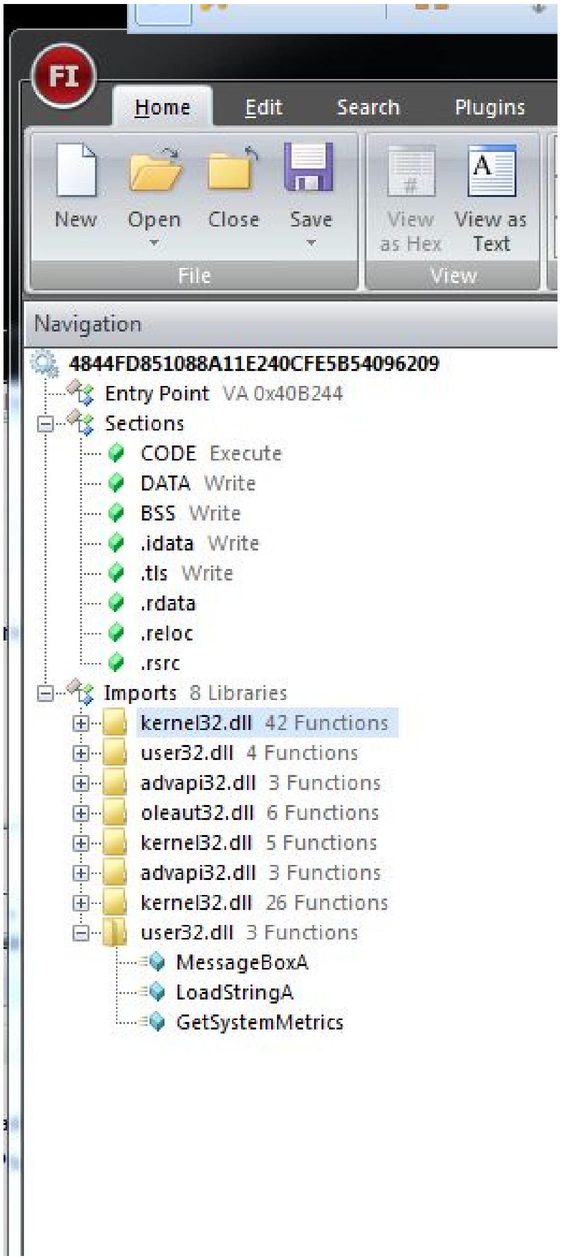
```
[2019-07-15 03:16:38.666000] Finish : 2 matches
[2019-07-15 03:16:38.666000] Signature match : suspect : C:/Users/Admin/Desktop/malware/MalwareDefense/Class2\4844FD851088A11E240CFE5B54096209
[2019-07-15 03:16:38.666000] Signature match : suspect : C:/Users/Admin/Desktop/malware/MalwareDefense/Class2\068D5B62254DC582F3697847C16710B7
[2019-07-15 03:16:38.650000] Execution...
[2019-07-15 03:16:38.650000] Compilation...
```

Two results are returned, so let's use another method to look at these two files:

In the two suspect files, we see a large series of libraries and functions:

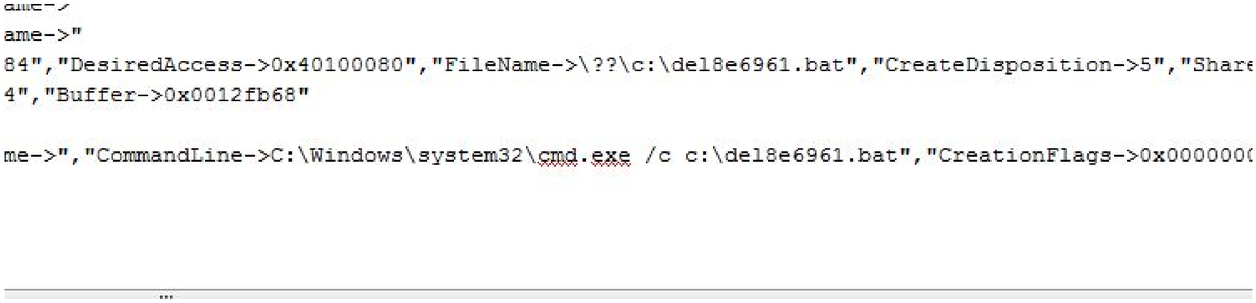


```
068D5B62254DC582F3697847C16710B7
  Entry Point  VA 0x401328
  Sections
    PSÿÕ«ëçÃ  Write | Execute
      Write | Execute
    žA  Write | Execute
  Imports  4 Libraries
    KERNEL32.DLL  22 Functions
      GetEnvironmentVariableA
      GetLastError
      GetModuleFileNameA
      GetProcAddress
      GetSystemDirectoryA
      GetTickCount
      GlobalAlloc
      GlobalFree
      LoadLibraryA
      LoadResource
      LockResource
      FindResourceA
      SizeofResource
      Sleep
      WinExec
      WriteFile
      lstrcatA
      lstrlenA
      ExitProcess
      CreateMutexA
      CreateFileA
      CloseHandle
    SHLWAPI.DLL  2 Functions
      PathFileExistsA
      StrStrA
    NTDLL.DLL  1 Functions
      sprintf
    MSVCRT.DLL  4 Functions
      fclose
      fwrite
      fseek
      fopen
```



```
4844FD851088A11E240CFE5B54096209
  Entry Point  VA 0x40B244
  Sections
    CODE  Execute
    DATA  Write
    BSS  Write
    .idata  Write
    .tls  Write
    .rdata
    .reloc
    .rsrc
  Imports  8 Libraries
    kernel32.dll  42 Functions
    user32.dll  4 Functions
    advapi32.dll  3 Functions
    oleaut32.dll  6 Functions
    kernel32.dll  5 Functions
    advapi32.dll  3 Functions
    kernel32.dll  26 Functions
    user32.dll  3 Functions
      MessageBoxA
      LoadStringA
      GetSystemMetrics
```

Continuing with our newest tool, Cuckoo shows us the following line near the end of the log:



```
ame->
ame->"
84","DesiredAccess->0x40100080","FileName->\??\c:\del8e6961.bat","CreateDisposition->5","Share
4","Buffer->0x0012fb68"

me->","CommandLine->C:\Windows\system32\cmd.exe /c c:\del8e6961.bat","CreationFlags->0x0000000
```

Here we see the file trying to execute a command from the command line

FINDINGS

068D5:

- write and execute Sections
- attempt to execute on the cmd line (cuckoo log)

00670:

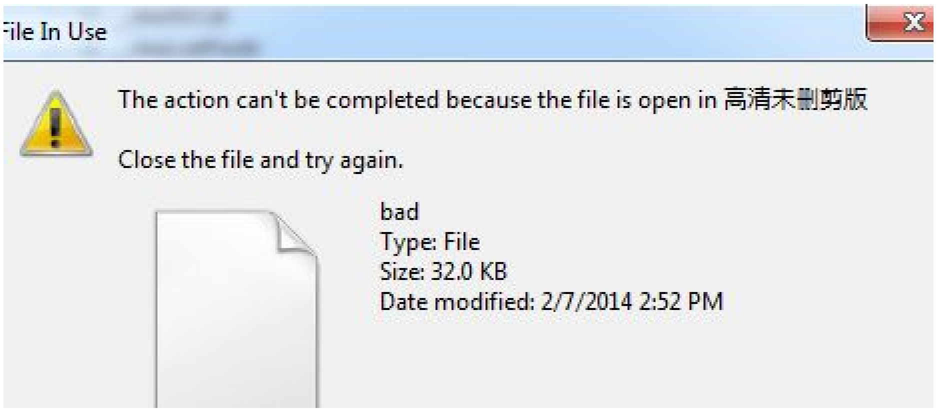
- execute and write sections
- only one library
- many vba functions

4844F:

- execute and write sections
- function “filetimetolocalfiletime”
- function “AdjustTokenPrivliges”
- functions for Registry Query, Open, and Close
- Weirdly I don’t see too much I recognize as malicious in the cuckoo logs

A1874:

- write and execute sections
- function for registry close key
- I can’t say with my limited knowledge that I see anything too alarming in the Cuckoo logs


File In Use

The action can't be completed because the file is open in 高清未删剪版

Close the file and try again.

bad
Type: File
Size: 32.0 KB
Date modified: 2/7/2014 2:52 PM

Chinese characters… I couldn’t get this to pop up again and I’m not sure what caused it

After looking over the functions and the logs, I remain in line with my initial findings: I suspect the files

- 4844FD851088
- 068D5B62254D

of being malicious files.

BLOG POST

James T Ball
July 14th 2019
20:33

I have come across a file within the Class2 folder that I highly suspect of being malicious. It began with a simple search within yara-editor that looked for the simplest of strings that may arouse suspicion:

```
        rule suspect
        {
                strings:
                                $a = “.exe”
                                $b = “http”
                                $c = “.js”
                condition:
                                $a or $b or $c
        }
```

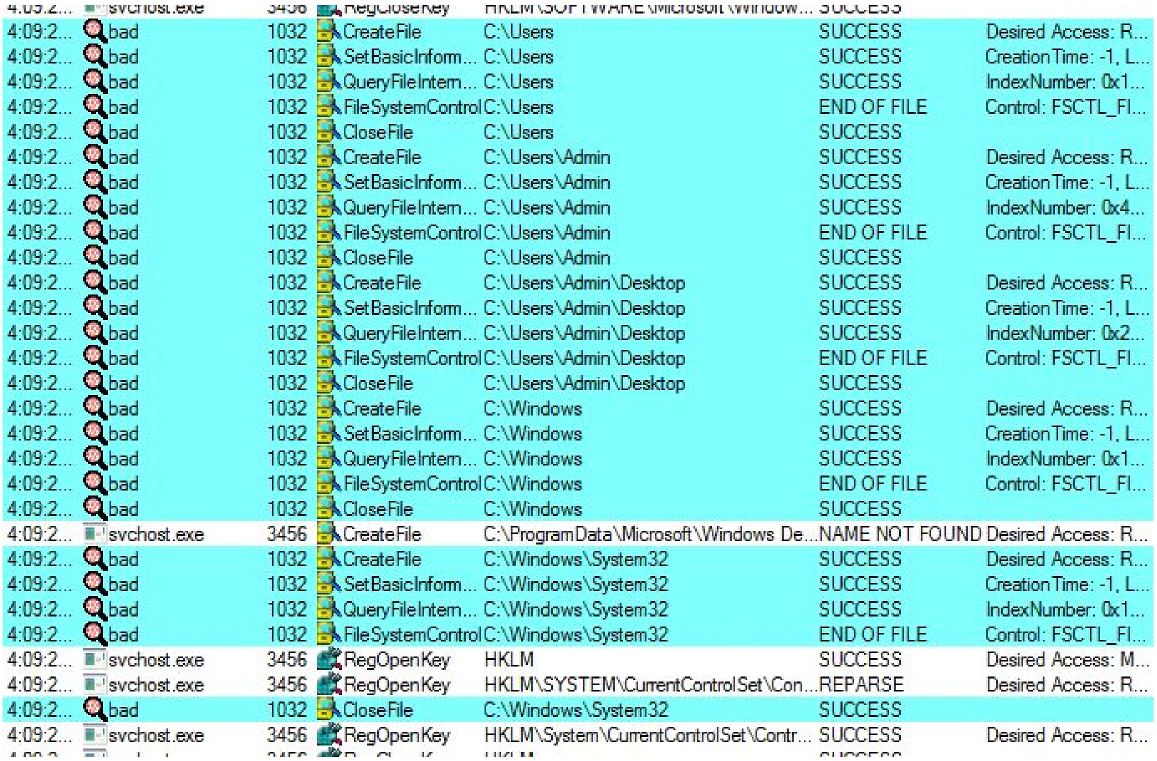
This returned two of our files one of which was the subject of this blog: 4844FD851088A11E240CFE5B54096209 which will be referred to as “4844” going forward.

To further investigate 4844, I opened the file within the FileInsight tool, and found 4844 contained a number of Sections for writing code, one for Execution, and a few other I wasn’t sure of.

Digging further into the libraries, I found odd function names like function “filetimetolocalfiletime”. Changing times is a common method in malware to avoid detection be changing records of when events occur. I also saw a function “AdjustTokenPrivliges”, and now had reason to suspect this already questionable program of trying to alter privileges.

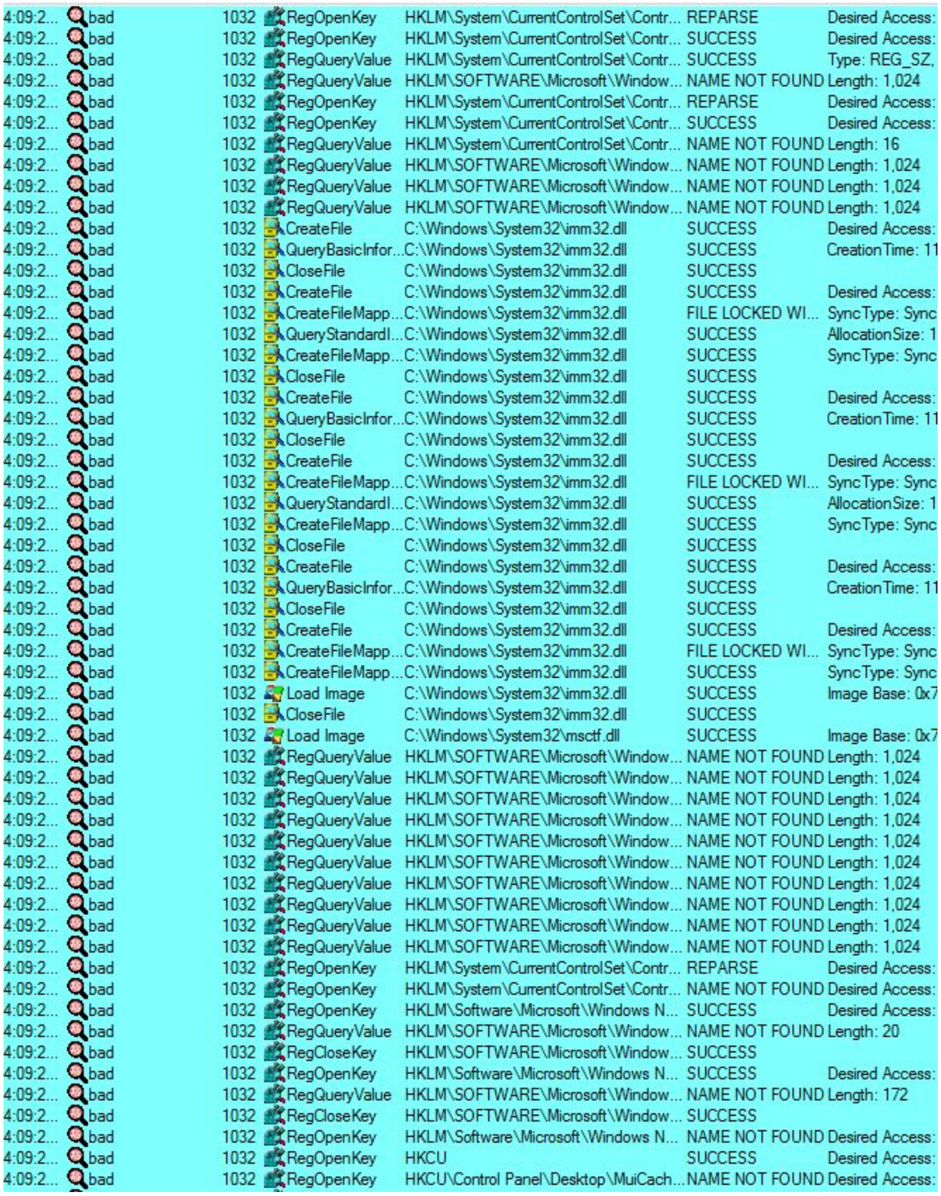
I then fired up the newest tool from week 3, Cuckoo.

While Cuckoo ran, I also had the old Process Monitor capturing data. The first thing 4844 did was create an alternative series of folders closely resembling windows folders:



```
4:09:2...  svchost.exe  3456  RegCloseKey       HKLM\SOFTWARE\Microsoft\Window...  SUCCESS
4:09:2...  bad          1032  CreateFile        C:\Users                           SUCCESS          Desired Access: R...
4:09:2...  bad          1032  SetBasicInform... C:\Users                           SUCCESS          CreationTime: -1, L...
4:09:2...  bad          1032  QueryFileIntern... C:\Users                          SUCCESS          IndexNumber: 0x1...
4:09:2...  bad          1032  FileSystemControl C:\Users                           END OF FILE      Control: FSCTL_FI...
4:09:2...  bad          1032  CloseFile         C:\Users                           SUCCESS
4:09:2...  bad          1032  CreateFile        C:\Users\Admin                     SUCCESS          Desired Access: R...
4:09:2...  bad          1032  SetBasicInform... C:\Users\Admin                     SUCCESS          CreationTime: -1, L...
4:09:2...  bad          1032  QueryFileIntern... C:\Users\Admin                    SUCCESS          IndexNumber: 0x4...
4:09:2...  bad          1032  FileSystemControl C:\Users\Admin                     END OF FILE      Control: FSCTL_FI...
4:09:2...  bad          1032  CloseFile         C:\Users\Admin                     SUCCESS
4:09:2...  bad          1032  CreateFile        C:\Users\Admin\Desktop             SUCCESS          Desired Access: R...
4:09:2...  bad          1032  SetBasicInform... C:\Users\Admin\Desktop             SUCCESS          CreationTime: -1, L...
4:09:2...  bad          1032  QueryFileIntern... C:\Users\Admin\Desktop            SUCCESS          IndexNumber: 0x2...
4:09:2...  bad          1032  FileSystemControl C:\Users\Admin\Desktop             END OF FILE      Control: FSCTL_FI...
4:09:2...  bad          1032  CloseFile         C:\Users\Admin\Desktop             SUCCESS
4:09:2...  bad          1032  CreateFile        C:\Windows                         SUCCESS          Desired Access: R...
4:09:2...  bad          1032  SetBasicInform... C:\Windows                         SUCCESS          CreationTime: -1, L...
4:09:2...  bad          1032  QueryFileIntern... C:\Windows                        SUCCESS          IndexNumber: 0x1...
4:09:2...  bad          1032  FileSystemControl C:\Windows                         END OF FILE      Control: FSCTL_FI...
4:09:2...  bad          1032  CloseFile         C:\Windows                         SUCCESS
4:09:2...  svchost.exe  3456  CreateFile        C:\ProgramData\Microsoft\Windows De...NAME NOT FOUND Desired Access: R...
4:09:2...  bad          1032  CreateFile        C:\Windows\System32                SUCCESS          Desired Access: R...
4:09:2...  bad          1032  SetBasicInform... C:\Windows\System32                SUCCESS          CreationTime: -1, L...
4:09:2...  bad          1032  QueryFileIntern... C:\Windows\System32               SUCCESS          IndexNumber: 0x1...
4:09:2...  bad          1032  FileSystemControl C:\Windows\System32                END OF FILE      Control: FSCTL_FI...
4:09:2...  svchost.exe  3456  RegOpenKey        HKLM                               SUCCESS          Desired Access: M...
4:09:2...  svchost.exe  3456  RegOpenKey        HKLM\SYSTEM\CurrentControlSet\Con...REPARSE        Desired Access: R...
4:09:2...  bad          1032  CloseFile         C:\Windows\System32                SUCCESS
4:09:2...  svchost.exe  3456  RegOpenKey        HKLM\System\CurrentControlSet\Contr... SUCCESS      Desired Access: R...
```

After that, 4844 began opening and querying registry values.



```
4:09:2... bad  1032 RegOpenKey        HKLM\System\CurrentControlSet\Contr... REPARSE         Desired Access:
4:09:2... bad  1032 RegOpenKey        HKLM\System\CurrentControlSet\Contr... SUCCESS         Desired Access:
4:09:2... bad  1032 RegQueryValue     HKLM\System\CurrentControlSet\Contr... SUCCESS         Type: REG_SZ,
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 1,024
4:09:2... bad  1032 RegOpenKey        HKLM\System\CurrentControlSet\Contr... REPARSE         Desired Access:
4:09:2... bad  1032 RegOpenKey        HKLM\System\CurrentControlSet\Contr... SUCCESS         Desired Access:
4:09:2... bad  1032 RegQueryValue     HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND  Length: 16
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 1,024
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 1,024
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 1,024
4:09:2... bad  1032 CreateFile        C:\Windows\System32\imm32.dll          SUCCESS         Desired Access:
4:09:2... bad  1032 QueryBasicInfor...C:\Windows\System32\imm32.dll          SUCCESS         CreationTime: 11
4:09:2... bad  1032 CloseFile         C:\Windows\System32\imm32.dll          SUCCESS
4:09:2... bad  1032 CreateFile        C:\Windows\System32\imm32.dll          SUCCESS         Desired Access:
4:09:2... bad  1032 CreateFileMapp... C:\Windows\System32\imm32.dll          FILE LOCKED WI... SyncType: Sync
4:09:2... bad  1032 QueryStandardI... C:\Windows\System32\imm32.dll          SUCCESS         AllocationSize: 1
4:09:2... bad  1032 CreateFileMapp... C:\Windows\System32\imm32.dll          SUCCESS         SyncType: Sync
4:09:2... bad  1032 CloseFile         C:\Windows\System32\imm32.dll          SUCCESS
4:09:2... bad  1032 CreateFile        C:\Windows\System32\imm32.dll          SUCCESS         Desired Access:
4:09:2... bad  1032 QueryBasicInfor...C:\Windows\System32\imm32.dll          SUCCESS         CreationTime: 11
4:09:2... bad  1032 CloseFile         C:\Windows\System32\imm32.dll          SUCCESS
4:09:2... bad  1032 CreateFile        C:\Windows\System32\imm32.dll          SUCCESS         Desired Access:
4:09:2... bad  1032 CreateFileMapp... C:\Windows\System32\imm32.dll          FILE LOCKED WI... SyncType: Sync
4:09:2... bad  1032 QueryStandardI... C:\Windows\System32\imm32.dll          SUCCESS         AllocationSize: 1
4:09:2... bad  1032 CreateFileMapp... C:\Windows\System32\imm32.dll          SUCCESS         SyncType: Sync
4:09:2... bad  1032 CloseFile         C:\Windows\System32\imm32.dll          SUCCESS
4:09:2... bad  1032 CreateFile        C:\Windows\System32\imm32.dll          SUCCESS         Desired Access:
4:09:2... bad  1032 QueryBasicInfor...C:\Windows\System32\imm32.dll          SUCCESS         CreationTime: 11
4:09:2... bad  1032 CloseFile         C:\Windows\System32\imm32.dll          SUCCESS
4:09:2... bad  1032 CreateFile        C:\Windows\System32\imm32.dll          SUCCESS         Desired Access:
4:09:2... bad  1032 CreateFileMapp... C:\Windows\System32\imm32.dll          FILE LOCKED WI... SyncType: Sync
4:09:2... bad  1032 CreateFileMapp... C:\Windows\System32\imm32.dll          SUCCESS         SyncType: Sync
4:09:2... bad  1032 Load Image        C:\Windows\System32\imm32.dll          SUCCESS         Image Base: 0x7
4:09:2... bad  1032 CloseFile         C:\Windows\System32\imm32.dll          SUCCESS
4:09:2... bad  1032 Load Image        C:\Windows\System32\msctf.dll          SUCCESS         Image Base: 0x7
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 1,024
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 1,024
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 1,024
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 1,024
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 1,024
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 1,024
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 1,024
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 1,024
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 1,024
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 1,024
4:09:2... bad  1032 RegOpenKey        HKLM\System\CurrentControlSet\Contr... REPARSE         Desired Access:
4:09:2... bad  1032 RegOpenKey        HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND  Desired Access:
4:09:2... bad  1032 RegOpenKey        HKLM\Software\Microsoft\Windows N...   SUCCESS         Desired Access:
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 20
4:09:2... bad  1032 RegCloseKey       HKLM\SOFTWARE\Microsoft\Window...      SUCCESS
4:09:2... bad  1032 RegOpenKey        HKLM\Software\Microsoft\Windows N...   SUCCESS         Desired Access:
4:09:2... bad  1032 RegQueryValue     HKLM\SOFTWARE\Microsoft\Window...      NAME NOT FOUND  Length: 172
4:09:2... bad  1032 RegCloseKey       HKLM\SOFTWARE\Microsoft\Window...      SUCCESS
4:09:2... bad  1032 RegOpenKey        HKLM\Software\Microsoft\Windows N...   NAME NOT FOUND  Desired Access:
4:09:2... bad  1032 RegOpenKey        HKCU                                   SUCCESS         Desired Access:
4:09:2... bad  1032 RegOpenKey        HKCU\Control Panel\Desktop\MuiCach...  NAME NOT FOUND  Desired Access:
```

Regrettably, I am still not sure what this file was hoping to do. I hope to revisit this file and process as I learn more methods of analyzing malware and become more comfortable with the processes I’ve been exposed to.