508.1 - Intro Speech

Sunday, March 25, 2018 7:59 PM

- DFIR Last line of defense
- DFIR be hunting for threats or responding to found threats
    - DFIR is the choice between success and failure when "low hanging disruptors"(AV, IDS, Firewall, etc) fail to prevent threat actor activity.
- No assigned seating, enforced seat swapping - Move around daily
- "15 Day Free Rental" program at Best Buy, Walmart, etc… Buy and Return.
- Good studying habits trumps last minute reliance on expertise.

IR Goals and Assumptions

Tuesday, April 3, 2018 9:59 AM

- Regardless of security posture a dedicated adversary WILL find a way into your organization
    - Adversary will get in, focus should be on rapid detection and prevention of re-infection/breach

- Management assumes that the initial detection is the beachhead
    - Assumption of dwell time is that it is 0, it should not be.
    - Management/Business objective of IR is typically to get back to BAU

- IR objective is to get in-front of the attacker and prevent goal achievement.
    - IR goal should be to learn about the adversary
    - Learn the attacker TTPs
    - Dwell time reduction good marker for IR improvement.

- Good security is hard and contradictory to convenience
    - Security in large scale organizations is like "Sick Child" analogy

- Security Incidents typically boil down to 1 bad security habit.
    - IE like multiple local admin on same user.
    - Failure to recognize basic security hygiene. Do you wash your hands?

- Once you've become the target of an APT, you'll never fully remediate.
    - "They" will continue to come back
    - IR is about playing keep away and making it difficult for attackers to move/gain foot hold.

Who are the APT groups

Tuesday, April 3, 2018 10:05 AM

- Current APT groups number in the 1000's
- The proliferation of APTs has skyrockets
    - Golden age of cybersec (again)
    - Easy to use tools, access to large amounts of data.
- Every nationstate has an APT group
- "Solar Sunrise"
    - First widely acknowledged nation state type cybersecurity attack.
- "Moonlight Maze"
    - Russian group attacking US military bases and research facilities.
    - JTF-CND Joint Task Force Computer Network Defense tasked to fight them (eventually became US Cyber Command)
    - 2015 - code found on server showed that "Moonlight Maze" group is Russian APT group called "Turla"
- Syrian Electronic Army
    - Hacktivist group

SRL Intrusion Scenario

Tuesday, April 3, 2018 10:15 AM

- Mirrors a highly defensible network
- Focus on single system analysis initially then move to applying at scale.
- APT haven't really changed their behavior /TTP
    - Soft gooey middle of the network
        - Gain foot hold
        - Escalate privs
        - Lateral movement
        - ???
        - Profit

- Built with 6~8 months of data for the SRL exercise.
    - Real world experience for a very defensible network

- Class with show case how to build up skillsets and promote to enterprise/at scale.

Incident Response Today

Tuesday, April 3, 2018 11:08 AM

- Average number of days compromise to detection 180 days/half year
- Most follow the 6-step incident response process
    - PICRRL
        - Identification - Detect AND scoping of all systems compromised.
        - "Containment" fallacy - can some things be truly contained?
        - Containment by Eradication is not a viable strategy

    - **Primary goal** is to learn about the adversary
        - If you tip your hand to the adversary, they typically double down and go harder to ensure survivability.
            - Emotions and Management may vie for immediate eradication.
            - Immediate eradication tends to accelerate attacker objectives
            - Minor problems get ballooned to major problems
    - **Secondary goal** is to remove the adversary
    - No Scoping - No idea how deep intrusion might be
    - No Containment = Intrusion "whack a mole"

- Once breached by an APT, you'll never go back to "BAU". There will always be an APT attempting to access your network.

Containment (AKA Heavy Monitoring)

Tuesday, April 3, 2018 11:23 AM

- APT containment is similar to law enforcement "containing" a terrorist cell
    - Monitoring of cell to obtain vital information
    - What are the Techniques, Tactics, and Procedures
    - What is the motivation of the APT
    - Gather IoCs and campaign identification.

- AVOID TIPPING YOUR HAND to the attackers, it's a chess game not a dueling match.

- Intelligence Development helps to reinforce and focus ID/Scope
    - Stay in Scoping -> Containment - Scoping loop until strike zone for found systems is hit
    - Popcorn popping method (when comprised system are just right)

Hunting vs Reactive Response

Tuesday, April 3, 2018 11:34 AM

- To accomplish hunting
    - Really appropriate for mature teams, hunting does not happen based just wanting to do it.
- Threat hunting is a special operations side of SOC, not everyone.
- Threat hunting will be much more successful with a narrow-focus on the "crown jewels" or what adversary will pop-up at.

- Must Haves;
    - Must have a SOC
        - There is no official term or criteria that defines a "Security Operations Center"
        - 24x7, multiple appliances, mature policies, etc

    - Must have inside the SOC a "Cyber Threat Intelligence" capability" (CtI)
        - TTP recognition
        - Signature incorporation for adversaries and campaigns acting against your org

    - Hunt Team as a role/function.
        - Not SIEM, not automated alerting.
        - A human using TTP/CTI generated by the adversaries attacking you

- Hunting is not an automated activity
    - The "Spell Check" example, however often does spell check work 100%.

Intel driven IR

Tuesday, April 3, 2018 11:47 AM

- TTPs will vary and should be cataloged/logs
- Lateral Movement is a choke point
    - Only 6 different methods to real do lateral movement based on built in tools(Windows)

- Forensics Analysis vs Threat Hunting
    - Don’t know what I’m looking for: forensic analysis

- If you're sleeping on a cot, you're in IR. If you get to go home at night you're doing threat hunting.
    - Immediate/Quick Response
        - 4-6 hours
        - Enterprise Response
        - Mem Forensics
        - Timeline Analysis
        - Initial Assessment
        - Threat Indicators
    - Deep Dive Analysis
        - 1-2 days answers
        - Anti-forensics detection
        - Malware and adversary ID
        - Threat Capabilities ad purpose

- A compromise has happened and our job is to find it.
    - Move from a "preventable compromise" mindset to "Inevitable but detectable and actionable"
    - "The bad guys are already here, our job is to find them"
    - "Our goal is to get in front of the adversary and prevent them from achieving goals"

Remediation Events

Tuesday, April 3, 2018 12:04 PM

- Remediation is HARD
  - Threats are good at avoiding detection and ensuring survivability
  - Threats react to Blue team tactics and counter measures
    - Blue team reaction accelerates red team actions on objectives or re-compromise.
  - Threats will return
    - Reduce your dwell time

- Remediation Event
  - Remediation Goals
    - Deny access
    - Restrict reactions
    - Remove presence
    - Degrade survivability
  - Event plan
    - Posturing
    - Execute
    - Implement controls

- Critical Remediation Steps

Cyber Threat Intelligence

Tuesday, April 3, 2018 12:07 PM

- CTI position needs to be considered as directly drives the IR cycle
    - Information about adversities

- CTI all about statistical probably that "evil exists"
    - Looking for the probability that something exists or doesn't exist on a system.

- Large problem with security tools/appliances are tuned to near 100% true detection

- Risk = Vulnerability + Impact + Threat (Intent + Opportunity + Capability)

- True intelligence based security posture is a much better way to approach IR.

Kill Chain and ATT&CK

Tuesday, April 3, 2018 12:14 PM

Malware-ology

Tuesday, April 3, 2018 1:45 PM

- "Traditional hunting fails when you put a mustache on a deer" - The simplest technique that works is usually what the threat actors will leverage.
- Why don't we see this? Normal signature based technologies only want true positive.

- Malware TTPs
    - The Malware Paradox
        - Malware can hide, but it must **run.**
        - Evidence of execution trails and living off of the land.

    - Malwre-ish things
        - Command line usage
        - NET usage
        - PSEXEC, Powershell
- 3 Detection Situations
    - Malware Active
        - C2 Ips, etc
        - Easiest to Identify

    - Malware exists but not active
        - More difficult ID
        - Persistent via scheduled tasks, cron, etc
        - Must ensure survivability

    - No Malware but compromised
        - Compromised via living off the land techniques

- Hiding in plain sight
    - Svchost top contender for hiding in plain sight
        - Known good is in the %Systemroot%\System32
        - Google Typographical and Homomorphic abuse of svchost.dll

- Common Evasion Techniques
    - Service hijacking/replacement
    - Py2exe or perl2exe
        - Export to HEX,
        - Python to run HEX
        - Python2exe encode/packing
        - Simple but effective

    - Signed code allow for easy kernel mode access.
        - Windows Hardware Quality Labs (WHQL) required in Server 2016
        - Only 4% of malware is signed
            - Increased risk to malware effectiveness (due to CRL revocation if burned)
            - Increase cost/effort to enact.
        - Higher percentage of signed code in nation-state APTs
        - Focus on non-well known authorities (Microsoft, Apple, Google)
        - Signed programs in "system locations" not from well-known entity are suspicious.
- Frequency Analysis

    - Encryption doesn't compress easily, PE/files that don't have high compression values are typically encrypted.
        - Densityscout (SIFT)
    - Pescan (SIFT)
    - Sigcheck (Windows)

- Malware Persistence
    - Malware wants to hide, but must survive a reboot.
        - AutoStart Locations
        - Service Creation
        - Service Failure Recovery
        - Scheduled Tasks
        - DLL Hijacking
        - WMI Event Consumers
        - Adv - Local GPO, MS-Office Addin, BIOS flashing.

508.2 - Intro Speech

Sunday, March 25, 2018 8:00 PM

- Public Service Announcement from Rob:
    - Memory Forensics is a volatile area, constantly changing with acquisitions methods and OS changes (Windows 10)
    - Real world is not going to mirror the canned/prepared scenario in class.
        - Race to keep up with changing space.

Why Memory

Wednesday, April 4, 2018 9:25 AM

- Everything running across the processor is going to be ram.
    - In RAM everything is in the clear, no code obfuscation survives to the memory space.
    - "The cloaked Star Trek ship" analogy

- Rob Pro Tip:
    - Volitility /Rekall executable dump executables, then run simple AV/hash scan.

- Rob Pro Tip
    - String search memory for your own passwords

- Many memory based security issues are more complex than advertised (Meltdown/Spectre) however finding an unknown value is the crux of these type of vulernabilities. Just because the RAM/Memory is exposed doesn't mean finding the sensitive data is easy.

- Advantages
    - Best place to ID activity software(malware) activity

- Disadvantages
    - Limited "horizon view", live memory will only hold a certain timeframe.

Memory Acquisition

Wednesday, April 4, 2018 9:40 AM

- Server 2016 with WHQL enforcement could be problematic for memory acquisition programs (signed drivers required)

- Try all tools on home machine (Server 2016)

- Virtual can be treated as bare metal machines, aside from certain situations where VM has a raw memory file on disk.

- VMware memory acquisition
    - Pause VM and copy .vmem file.

- Windows 10 Notes
    - Rapid updates schedule, causes support lag in memory tools
    - Rekall tends to be better supported (if connected to internet to get latest updates)
        - But doesn't have the same number of plugins as volitility (pslist, etc)
    - Volitility has much better plugin support

Hibernation and Sleep

Wednesday, April 4, 2018 9:48 AM

- www.comae.io
  - Dumpit and hibr2bin

- Hibernation files are backed up into Volume Shadow copy.

- Sleep mode versus Hibernation (Laptop)
  - Connected standby/Sleep = low power but things still happening in the background due to new Win8/Win10 sleep mode processes.

  - Hibernation tends to occur around 10% when sleeping.

- Hibernation files (Desktop)
  - Every time shutdown/restart, win10 creates a hibernation file
    - Possibly for app restart at same place prior or fast startup (theory)

- Windows 8/10 Notes
  - Hiberation files format changed in 8+
  - New Shim/Prefetch cache formats

What is Memory Forensics

Wednesday, April 4, 2018 10:01 AM

- Study of captured data from memory of a target system.
- E-process structure
    - Forward/Backward link list
        - Often exploited by root kits to hide processes
    - Forward link (f-link)
    - Backward link (b-link)

- Hiding process
    - Pslist or psscan
    - Or full memory scan

Volatility

Wednesday, April 4, 2018 10:25 AM

- Protip
  - To speed up volalitity use profile variables
    - Export VOLATILITY_*thing*

- Protip use -g KDBG to speed up imageinfo
  - export VOLATILITY_KDBG=*hex value*

- Look in DROPBOX link for KDBG method
  - Latest version needs to be in the following format:
    - export VOLATILITY_KDBG=$[*hex value*]

- PSTOTAL (dot output for visual aides)
  - Hot Reboot
    - Doesn't always clear/overwrite RAM, Volatility can pull back processes from the previous boot.
    - Unallocated spaces may show up in output.

- Malprocfind
  - Designed to find common evil/suspect procesess (on the blue poster)

Code Injection

Wednesday, April 4, 2018 3:43 PM

- Hiding in plain sight
  - Legitimate process with a backdoor
- Avoid protective technologies

- Stability methods (process migration)

- Code injection comes from a thing
  - Dormant Malware (service initiated)
  - Rootkits

- Code injection is very common with modern malware
  - Built in feature of windows

- Process hollowing
  - Hollow out dead process

- **Injection Detection**:
  - Step 1: Memory marked Page_ExecuteReadWrite
  - Step 2: Memory section not backed with a file on disk
  - Step 3: Memory section contains code (MZ or shell code)

    MZ header = PE Code

    Shell Code/Assembly = Peanut Butter and Jelly example

    Shell building blocks on top of each other that are linked together

    PUSH EDR
    MOV EDR, ESI
    ADD ESI, 0x087687
    MOV ESI [EBP+0x08]
    LEA EAX, [ESI+0x0878493]
    POP EAX
    ….

Rootkits

Wednesday, April 4, 2018 4:46 PM

- Memory Analysis is one of the few tools that will show case Rootkit artifacts.

- Rootkit code is highly temporal and subject to failure with OS updates. Rootkits must be thoroughly tested against a particular OS to ensure

- Rootkits primarily coming in the form of a driver
    - Because allows for access to the kernel

- Service System Descriptor Table (SSDT) hooking
    - If a processes is exited is will show up in psscan

- API Hooking
    - Modify
        - Look/Grep for
            - Camera
            - Clipboard
            - Font
            - Keyboard
            - Microphone

- /cases/example-memory-images/apt.img
    - Step 1
    - Step 2
    - Step 3 - Win XP Machine (use connscan and sockscan)
    - Step 4
    - Step 5
    - Step 6

APT Walkthrough

Sunday, March 25, 2018 8:00 PM

Tips;

1) Screen Cap software
2) Rules of network connections
    a. Any process NOT a browser communicating over 80/443/8080
    b. Any process that's a browser but communication over NOT 80/443/8080

Step 1 - malproc
Step 2 - Pstree
Step 3 - Conscan (network)
Step 4 - ssdt
Step 5 - malfind / code detection / strings
Step 6 - getsides of suspect processes ^
Step 7 - dllhost /dlllist process
Step 8 - svcscan -v > svcscan.txt

Object dumping

Thursday, April 5, 2018 9:57 AM

- Dlldump and moddump similar
  - -r (regex) --dump-dir="directory"
    - dlldump -r --dump-dir="directory"
    - Moddump -r --dump-dir="directory"
  - (Or just dump it all and check for AV hits)
    - dlldump --dump-dir="directory"
    - moddump --dump-dir="directory"
- Procdump and memdump
  - procdump -n(regex) --dump-dir="directory"
  - procdump --dump-dir="directory"
- Just dump all of it then
  - Check for file type (file *) look for MZ/PE32
  - Check for AV hits

String Dumping

Thursday, April 5, 2018 10:12 AM

- String Searching with memdump
  - Vol.py -f memory.img memdump -n csrss --dump-dir=.
  - Vol.py -f memory.img memdump -n conhost --dump-dir=.

- strings -t d -e l *.dmp >> conhost.uni
- grep -l "command prompt" conhost.uni

- Conhost contains all command lines in memory
  - Command line
  - Powershell

- cmdscan and consoles (lower percentage of success)
  - Cmdscan = command entered
  - Consoles = command entered and Output

Extracting Files

Thursday, April 5, 2018 10:15 AM

- Filescan and dumpfiles for extracting files.
- If it was open and mapped to memory is no longer protected.

Registry Dumping

Thursday, April 5, 2018 10:21 AM

- Timestamps in registry keys are typically much more reliable (forensically) for accurate timestamping activity.
- Anti-forensics is a camouflage activity, not usually intended to achieve invisibility.
- printkey -K
- autoruns

Time Stamps - Now What?

Thursday, April 5, 2018 11:22 AM

From Exercise 2.4

vol.py -f win7-32-nromanoff-memory-raw.001 --profile=Win7SP1x86 printkey -K "CONTROLSET001\SERVICES\NETMAN\DOMAIN"
Volatility Foundation Volatility Framework 2.6
Legend: (S) = Stable (V) = Volatile

Registry: \REGISTRY\MACHINE\SYSTEM
Key name: domain (S)
Last updated: 2012-04-03 23:42:04 UTC+0000 - **MALWARE INSTALL TIME?**

Subkeys:

Values:
REG_SZ home : (S) http://12.190.135.235/ads/
REG_DWORD pause : (S) 64

root@siftworkstation:/cases/win7-32-nromanoff-memory# vol.py -f win7-32-nromanoff-memory-raw.001 --profile=Win7SP1x86 mimikatz
Volatility Foundation Volatility Framework 2.6
WARNING : volatility.debug : [Credential:decrypt_epwd] unicode decode error
Module User Domain Password
-------- ---------- ---------- ----------
wdigest nromanoff SHIELDBASE blackwidow
wdigest tdungan SHIELDBASE !dumdum!
wdigest vibranium SHIELDBASE hailhydra
wdigest rsydow SHIELDBASE ThisIsAnActualPassphrase
wdigest WKS-WIN732BITA$ SHIELDBASE a3e98e75130c13c5e16...093766a2f6d18a283e

Intrusion Forensics Agenda

Thursday, April 5, 2018 11:27 AM

- Karate Kid montage
- The flash forward then backwards method

Advanced Evidence of Execution

Thursday, April 5, 2018 11:40 AM

- Prefetch Files
  - Review: Windows Prefetch (Common on Workstations NOT often on Servers
    - Thing
  - pf by TZWorks (Commercial - Linux)
    - Pf -v "file name"
  - PECmd by Eric Zimmerman (FREE - Windows) https://github.com/EricZImmerman/PECmd
    - PECMD.exe -f "file name"
  - Volatility
    - Vol.py -f memory.img prefetchparser
  - When a prefetch file is created it never overwrites the original prefetch file on disk due to inuse restrictions of prefetch.
- Wiping commands (Anti-forensics techniques)
  - Leave prefetch traces
  - Wiping the wiper typically is the point of no return

Prefetch Files

Thursday, April 5, 2018 11:40 AM

Review: Windows Prefetch (Common on Workstations NOT often on Servers

- Available XP through WIN10
- Used for frequent/last used application re-launch acceleration.
- Also used for performance improvement in running applications

- Locations:
  - C:\Windows\Prefetch
  - Up to Win7 file limit of 128 files
  - Win8+ 1024 file limit.

- File Name = "Program Executed"-"HASH".pf
  - Hash is calculated based on running location and commands

- Default ON, requires reboot to turn OFF

- Time Stamps
  - Date Modified = Last Execution
    - Win 8 / Win 10 has last 8 executions embedded
  - Date Created = First Executed

Application Compatibility

Thursday, April 5, 2018 12:05 PM

- Also called the "Shim Cache"
- Exists from XP to Current
    - XP = SYSYTEM\CurrentControlSet\Control\SessionManager\AppCompatibility \AppCompatCache
    - WIN7+ SYSYTEM\CurrentControlSet\Control\SessionManager\AppCompatCache \AppCompatCache

- Ensures compatibility between levels of OS.

- "Shimmed" mean certain OS specific properties may need to be applied to.

- **ONLY WRITTEN to disk/SAM/Registry hive at reboot**

- What it does track
    - Name, Path, Last Mod of Executable file

- What it doesn't track
    - Doesn't record time shimmed or time executed.

- Applications will be reshimmed if:
    - Content is updated/renamed
    - Files moved
        - File moved/renames/updated will have the same time stamp (Windows has a 64bit timestamp, effectively a 64bit hash)
    - Time Stomped
        - If the files Modified Time and ShimCache Mod Time, time stomping has occurred.

- Shim Cache is written in **TEMPORAL ORDER**
    - Top entries are *newest*
    - Bottom entries are *oldest*

- Application Compat/Shim Cache parsing
    - Vol.py shimcachemem (Volatility) - From Shim Cache registry hive in memory
    - ShimCacheParser.py (SIFT) - From the Shim Cache Hive File

- Application Compat - RecentFileCache
    - Only on Win7+/Server2008+ systems
    - Only exists for the past 24hrs
    - Rfc.pl RecentFileCache.bcf

- Application Compat - Amcache.hve
    - Features
        - Contains Drive GUIDs and MFT keys for applications run
        - First time (Last write time of key)
        - SHA-1
        - Executable path\name
    - Parsing with

- "Zimmermans" Registry Explorer
- Amcacheparser.exe
- Amcacheparser

Volume Shadow Copy (VSS)

Thursday, April 5, 2018 2:08 PM

- Happens every day on servers, weekly on workstations

- Ask

- 2008+ Server OS heavily uses VSS for it's backup and restore level functions.

- Workstations should have the VSS turned-on for extending horizon view, troubleshooting, etc.
    - Not same as WinXP anymore
    - A myth that VSS needs to be off.

- Good for having to extend event log "horizon"

- vhadowninfo and vshadownmount

- SIFT Shadow Analysis
    - /mnt/vss
        - For usage with vshadowmount to show all shadow libraries
        - A simple loop to mount all shadows - "for i in vss*: do mountwin $i /mnt/shadow_mount$i; done"

    - /mnt/shadow_mount
        - Use to mount individual shadow copies.

Lateral Movement Tactics

Thursday, April 5, 2018 3:42 PM

- Limited ways to gain access to credentials on local host (post initial compromise)
    - Looking for non-normal lateral movement is the best way to identify hidden evil
        - Workstation to Workstation traffic

- Gain Authority
    - **Get legit creds and account creation**
        - Detection - Event Logs
        - UAC - Not entirely effective/easily bypassed
        - NTLM Hashes -
            - Interactive Session hands on keyboard, RDP, PsExec (with user creds)
            - 445 NTLM / Local User
            - Win8+ No longer allows admin level accounts to authenticate via NTLM
                - Event 4762 (Priv Account Logon)
            - Defending
                - Prevent Admin Compromise
                - Win10 (Credential Guard)
                - Stop RDP with highly priv user
                - **Domain Protected User Group**
                    - Prevent sending Hashes to interactive sessions
        - Token
            - Must have *SeImpersonate* privilege on account to steal.
            - Domain Protected Users Group spawn protected processes which do not have tokens
            - Defending
                - Prevent Admin Compromise
                - Win10 (Credential Guard)
                - Stop RDP with highly priv user
                - **Domain Protected User Group**
                    - Users in this group do not create delegate tokens

        - Cached Credentials
            - Are salted with domain information and must be cracked to clear password
            - 25 Exist on servers
            - SYSTEM and SAM <- Local NT Hashes
            - SYSTEM and <-- Cached Credentials
            - Defending
                - Prevent Admin Compromise
                - Enforce password length
                - Limit number of cached logon accounts
                - **Domain Protected User Group**
                    - Users in this group do not cache credentials

        - LSASecrets
            - Defending
                - Prevent Admin Compromise
                - Do not employ serve or scheduled tasks requiring priv accounts
                - Limit number of cached logon accounts
                - Domain Protected User Group

        - ◇ Users in this group do not cache credentials

    - Tickets
        - □ Not entirely closed off by Domain Protected User Group
        - □ Kerberos based
        - □ Typically cached in memory and valid for 10hrs
        - □ Kerberoasting
        - □ Pass the Ticket
        - □ Defending
            - ◆ Credential Guard / Remote Credential Guard
            - ◆ Long passwords
            - ◆ Audit service accounts for unusual activity
            - ◆ Change KRBTGT password regularly (yearly)
- **RDP**
    - Jumplists?
    - RDPClip allows for moving files around

- **Windows Admin Shares**
    - Who is doing that
        - □ 4624 and 4672
    - Explorer registry key
        - □ Mount points
- **PsExec**
    - Accept EULA registry entry good for time stamping

- Win Remote Management Tools
- PowerShell Remoting / WMIC
- Vulnerability Exploits / Application Dev Software

- Internal harvesting is not easy, attackers must do substantial work to enumerate the individuals/assets with the data/information that is part of their actions on objectives.

Event Log Fundamentals

Friday, April 6, 2018 9:17 AM

- NT/Win200/XP/2003
  - .evt file type
  - %systemroom%\System32\config
- Win7+ and Win2008+
  - evtx file type
  - %systemroom%\System32\winevt\logs
  - Oalerts event log
    - Records office pop-up boxes

- More adversaries using local event log manipulation
- Open up the default event log sizes to as big you can make them (10% of drive size)
  - GPO all workstations to enforced large security event log size to 5~10Gb
  - HDD space is cheap theses

- Definition of fail sauce is to say "We should've been doing that before now".

- Who watches to watchers
  - Monitor the high priv / high access users.
- Account Logon = Authentication/Authorization
  - Boarding a Plane
    - TSA = Authentication Event
    - Ticket to Board = Logon Event
- Logon Event = Logon/Logoff locally

Analysis Scenarios

Friday, April 6, 2018 9:40 AM

- 4624 + 4672 = Administrator/Privileged user logon
    - Logon Type # + Logon with Special Permissions
    - Be wary of SeImpersonatePrivilege in permissions (allows token stealing)

- **Account Creation**
    - 4720

- Logon Type 2 and Type 10 - Need to be carefully inspected due to having potential to carry Hash/Token and Cached Credentials

- Logon Type 3 - Typically equals network share or RPC calls

- **Identifying Logon Sessions**
    - 4624 (Logon) and 4647 (Logoff) will have a matching logon ID

- **Network Logons**
    - Network logons (Type 3) may have a higher/or no threshold for account lockouts in an enterprise environment.

- **Builtin Service Accounts**
    - Normal and natual to see
        - SYSTEM
        - LOCAL SERVICE
        - NETWORK SERIVE
- **Tracking Account Creation -** Event ID 4720
    - Account Creation.

- **Tracking RDP - Event ID 4778(Connect)/4779(Disconnect)**
    - Tracking for RDP sessions
        - 4778 Event
            - Additional Data
                - Client Name comes from the original device/attacker hostname, not the last hop.
                - Client Address comes from the original device/attacker IP, not the last hop
        - 4624 Event.
            - Look at "Source Network Address" for original attacker IP.
- 4776 (NTLM)
    - Successful / Failed
    - NTLM forces workstation to workstation
    - Network Logon (Type 3) with NTLM
    - Cached Domain Creds (11,12,13)

- **Pass the Hash Attack**
    - 4672 + 4776 + 4624

- **4771 Logon Errors Codes**
    - 40 Errors code, use reference sheets

- **Network Share Access**
  - Target will show a 4624 /5140 (Network)
    - Domain Controller will show 4768/4769 for authentications
  - Target will show a 4624/5140/4776 (Local)
  - Shares mounts via run-as (4848 on local, 5150 on target

- **Scheduled Tasks**
  - 3 log locations for scheduled tasks
    - Security Event Log
    - Task Scheduler Event Log
    - Schedule XML files (C:\Windows\System32\Tasks folder)

- **Service Events**
  - Event ID's - 7030~7050(System log), 4697 (Security log)

- **Application Installs**
  - Compromised GPO would require MSI files, which create install records. (Event 11707)
  - Event 1015 - Application crash logs

- **Process Tracking**
  - With caution due to logging rate
  - Detailed command line auditing
  - Powershell script block logging
    - PSv5 for automatic logging
    - Do it now, cause only should be used by admins to servers
    - Powershell/Operational log
      - 4104 Script content
      - 4105 Script Start
      - 4106 Script Stop

- **Event Log clearing**
  - Event ID's 1102 and 102, others
  - Can be cleared complete by adversary tools. With no lasting evidence by mimikatz.

LAB 3.2

Friday, April 6, 2018 9:17 AM

- Date Range:
  - 3/29/2012 5:02:59 AM to 4/7/2012 17:29:58 PM
- Local Account (Success)
  - SRL-Helpdesk
- Date Range of Authentications (Success)
  - 4/3/2012 18:39:34 PM to 4/4/2012 20:02:55 PM
- Source Workstations (Success)
  - WKS-WIN732BITA
  - WIN-9119IJK2JVP
  - CONTROLLER
  - 4y6jJ74CKqiNXccI
- Local Account (Failures)
  - SRL-Helpdesk
  - tdungan
  - 10.3.58.5\Administrator
  - Administrator
  - John Strand2
  - ?
  - vibranium
  - vibranium@sheild
  - VIBRANIUM

- Account Responsible for failed logons
  - John Strand 2

- Additional Logon failure from same source
  - SRL-Helpdesk
- Source workstation Investigation
  - SRL-Helpdesk
  - 10.3.58.7
  - Type 3 (Network)
- Date/Time ^
  - 4/3/2012 18:39:34
- RDP Events
  - Vibranium
  - 4/4/2012
  - LaNMaSteR's Mac
- Type 10 Logons
  - Nromanoff
  - Rsydow
  - Vibranium
  - SRL-Helpdesk
- IP Address of RDP
  - 127.0.0.1
  - 10.3.16.5
  - 10.3.58.4
  - 10.3.58.7

- Priority for IR
    - 10.3.58.4
        - SRL-Help
        - Rsydow
    - 10.3.58.7
        - Vibranium
- 5140 Failures
    - 26
- Ip Address
    - 10.3.58.7
- Account Associated
    - SRL-Helpdesk
- Shares
    - C$
    - ADMIN$
- Failure Reason
    - SRL-Helpdesk not a domain admin or in local admin group
- PSEXESVC (Secutiry)
    - 8 Events
    - Local and incorrect domian used
    - PSEXEC -I switch (interactive) or -u -p
- PSEXESVC (System)
    - 7 Events
    - 4/3/2012 and 4/4/2012
    - Files
        - %SYSTEMROOT%\TopLZAGU.exe
        - %SYSTEMROOT%\oSCMpGpk.exe
        - c:\windows\system32\dllhost\svchost.exe
        - C:\Program Files\Jetico\BCWipe\BCWipeSvc.exe
        - C:\Windows\system32\drivers\BCSWAP.sys
        - C:\Windows\system32\drivers\fsh.sys
    - Auto-Starts
        - BCWipe service
        - Fsh
        - winsvchost
    - User Accounts
        S-1-5-21-2036804247-3058324640-2116585241-1114
        S-1-5-21-2036804247-3058324640-2116585241-1673

Timeline Analysis

Friday, April 6, 2018 9:17 AM

- Timeline analysis, previously law enforcement would only have certain ways to capture data.
  - Wire taps and Network Monitoring made allowable by:
    - US Code Title 3 - Wire Taps
    - US Code Title 18 Chapter 2510 - Consensual Wiretaps
      - https://www.law.cornell.edu/uscode/text/18/part-I/chapter-119
        - Can't do random monitoring
        - "Safe Work Freak Flag"
    - Computer Trespasser exception in Patriot Act

- Timeline born out of need to track attacker movements based on law restrictions.
  - Netflow data + local file system metadata after attacker leaves the system
  - Artifacts left behind by attackers are considered "stored communications" on a system.
  - "MACDaddy" was one of the original timeline analysis

- Timeline Benefits
  - Examine around incident
  - Detect C2
  - Makes anti-forensics very difficult
  - Adversaries leave footprints everywhere.

- Do you understand the artifacts to help you define the timeline.

- Only a small sliver of forensics artifacts are covered in this time line.

- Timeline analysis is not simply artifact recovery

- Biggest mistake is to compile a timeline then start read page 1,2,3,4….X
  - Use a PIVOT POINT. The point you are aware that something occurred or when an known artifact was created
  - Context is crucial to understanding timeline analysis

- Timeline data is good for importation into Database (ELK, bigdata, etc).

- In timeline analysis, just because you don't see it, doesn't mean it didn't happen.
- Time line evidence is extremely sensitive to system changes.
- Time line paradox
  - When an action happens that isn't appropriately/accurately time stamps instead of what logically happened
  - File Created in Folder A, Folder B created, File Moved to Folder B. Timestamp will show that Folder B was created with File 1 artifacts before the Timeline says it was .

- Time Line Tools
  - File System Focused (fls)
    - Extremely fast
    - Triage based, only a 60~70% solution

  - Super Timeline (log2timline)

- Obtain Everything - Kitchen Sink
- Windows, Linux, Mac

Timestamps

Friday, April 6, 2018 3:34 PM

- M Data Content Change Time
  - Time the data content of a file was last modified
- A Data Access Time
  - Approximate time when the file data was accessed
- C Metadata Change Time
  - …
- B Metadata Creation Time
  - …

- FileSystem Time Zones
  - FAT is a local time storage filesystem
  - NTFS / exFAT is UTC

- File Rules
  - Made/Born
    - All time stamps will be equal for initial object.
  - Copied
    - When Modification Time (M) less than (<) a Creation Time(C) = The file was copied
  - Moved
    - When Access Time is greater (>) than Modified AND Creation Time = The file was moved

- Time Rule Exceptions
  - Office documents, you're not opening up the original
  - Original copy is the ~….tmp file
  - Uses MFT shifting

- Use File Timestamps as **Timeline Analysis Pivot Points**

LAB 3.3

Friday, April 6, 2018 4:47 PM

1. 2012-04-03 23:35:07
2. 380-144-1
3. 60919-128-9
4. "home"="http://12.190.135.235/ads/"
   "pause"=dword:00000040
   [Handle (Key)] MACHINE\SYSTEM\CONTROLSET001\SERVICES\NETMAN\DOMAIN svchost.exe PID: 6404/PPID: 2100/POffset: 0x7f69d880

5 2012-04-03 23:54:56
2012-04-04 00:05:15
2012-04-04 02:22:10

6
Timestamp
2012-04-03 21:19:53
File Name
[SHIMCACHE] \??\C:\Windows\TopLZAGU.exe
File Name
[SHIMCACHE] \??\C:\Windows\PSEXESVC.EXE
File Name
[SHIMCACHE] \??\C:\Windows\oSCMpGpk.exe

7

1. Modified - 2012-04-03 22:53:39, Created 2012-04-03 22:59:43.
2. File was copied
3. G
4. G

Super Timelining

Saturday, April 7, 2018 1:18 PM

- Just because you didn't see it, doesn't mean it didn't happen
  - However you must interpret the facts as they are presented.
  - You must make intelligent guess and be ok with it, based on probability.

- Pivot Points
  - Follow the bouncing ball

- Modification Times(M) can be ahead of Creation Time (B) during download or network transfer situations

- SHIMCACHE Entries are when the entry was written to the registry, not execution or modification time of the executable.

- USER ASSIST entries are when the user runs a GUI based program.

- Log2timeline is SLOW
  - Be aware of this during IR needs. Moderate size drives can take up to an hour without shadow copies collected.
  - Using "Targeted Timeline" analysis is much faster, but more manual.
    - Will automatically append to existing dump files if executing successive commands:

      - **Use Filter File to build an expedited super timeline**
        - Log2timeline.py -f /cases/filter_windows.txt /source/disk_image.img /cases/plaso.dump
        - Log2timeline.py --parsers="mactime" /source/disk_image.img /cases/plaso.dump

      - **Use FLS and Volatility to dump file system/memory artifacts**
        - Lfs -r -m C: (or appropriate drive letter substitution) /source/disk_image.img /case/disk.bodyfile
        - Vol.py -f /source/memory_image.img timeliner --output-body --output-file= /case/memory.bodyfile
        - Cat /cases/memory.bodyfile >> /cases/disk.bodyfile

      - **Append FLS file with log2timeline --parsers command**
        - Log2timeline.py --parsers="mactime" /cases/plaso.dump /cases/disk.bodyfile

      - **Psort plaso.dump file with appropriate time bracketing (if needed)**
        - psort.py -o L2tcsv plaso.dump "date > '2012-04-03 00:00:00' AND date < '2012-04-07 00:00:00'" > plaso_sorted.csv

      - **Reduce noise with whitelist grepping commands**
        - grep -a -i -f -v /cases/whitelist.txt plaso_sorted.csv > /cases/supertimeline.csv
          - MUST have -a or will not import into Timeline Explorer

LAB 4.3

Saturday, April 7, 2018 10:54 AM

2. Vibanium first logon
    a. Perform a search
        i. 2012-04-03 21:02:49 UTC
    b. Event Type
        i. 4625 - Account Logon
    c. Success
        i. No
    d. 10.3.58.7
    e. Type 3 - Network Logon
3. When was successful logon?
    a. 2012-04-03 21:03:05 UTC
    b. Elevated User logon
    c. 10.3.58.7
4. File Paths
    a. C:\Windows
    b. C:\Windows\Temp
5. What Happened after TOPLZAGU.exe
    a. MqlXmtLRaYQDMsvljY service installed and started
6. PSEXEC
    a. 2012-04-03 21:11:07
7. DLLHOT.exe
    a. MEI118482/kernel32.dll
    b. MEI118482/python25.dll
    c. MEI118482/avbypass.exe.manifest
    d. Likely and av bypass solution for rootkit?
8. DLLHOT.exe existence?
    a. Moved or Deleted
9. Spinlock Times
    a. B 2012-04-03 22:59:57
    b. M 2012-04-03 22:53:39
    c. B > M = Copied
10. Spinlock Execution
    a. 2012-04-04 18:54:51
11. When and how svchost.exe

LAB 4.4

Saturday, April 7, 2018 12:54 PM

1- IEXPLORE.EXE-1B894AFB.pf
   a. 2012-03-22 11:11:28
   b. 2012-04-03 18:33:41
   c. 2012-04-04 16:11:11

2- # of DEFRAG executions
   a. 5

3- Examine NTUSER.DAT information
   a. Vibranium exist prior to 4/2/2012
      i. No
   b. Why
      i. born date of NTUSER.DAT = 2012-04-03 21:21:19
   c. First interactive logon?
      i. 4624 Type 2 @ 2012-04-03 21:19:53
4- Does]dllhost\svchost.exe exist
   a. Yes
   b. Likely created on 2012-04-03 22:40:25

Enterprise IR Hunting

Saturday, April 7, 2018 1:47 PM

- Scalable hunting should be done on devices you know what you're looking for.
- Future of IR … Powershell (WMI, .NET, and COM)
    - V4 or v5 preferred
    - Powershell directly to the searchindex to check if a file exists (cool?)

- WinRM
    - Enter-PSSession
    - Invoke-Command

- Good for security
    - No delegation via Kerberos
    - Type 3 logon
    - Credentials not passed to remote system.

- Kansa
    - Welcome back to 504
    - Target List and Count recommended due to how Kansa queries
        - In mass response could DDOS network/self
    - Modules.conf
        - Must uncomment individual scripts

LAB 5.1 (Optional)

Saturday, April 7, 2018 2:19 PM

- Needs VPN Setup is using FOR508 Experimental LAB

Anti-Forensics Detection

Saturday, April 7, 2018 2:21 PM

- Advanced Attackers till be super sneaky, using TTPs near seen before, and living off of the land.
- Fileless malware != artifact-less
- Many anti-forensics tools require 3rd party tools to effect, which leave foot-prints

Data Storage Organization

Saturday, April 7, 2018 2:21 PM

- Cluster, made of sectors.
  - Allocated - currently storing data
  - Unallocated - 0's or storing previously deleted data.

- Contiguous disk space
  - Windows goes out of it's way to right files contiguously
  - Space between related files on disk has a good possibility of containing unknown but usable files
  - Slack space can be very useful in identifying deleted folder structures.

- MetaData layer
  - Similar to a "card catalog"
  - Inode = MFT entry
    - Inode are created contiguously (numerically)
    - Files are stored in folder names alphabetically by design.
  - Both allocated and unallocated space (could be recovered)

  - Creation Time, True Creation Time, and Inode numbering analysis to find file system outliers

  - As sorting of file creation(B) by Inode / MFT file value can showcase interesting data and additional adversary activity.

- File Name Layer
  - File System Metadata
  - ExFat lobbied for creation by Porn industry.
  - NTFS first on OS2 operating system.
  - ReFS parsed by very few forensic tools
  - NTFS
    - 255 files name length, 32,767 characters for path
    - Max File len - 16Tb - 64k
    - Max Volume - 256Tb
      - 2TB MBR restrictions, GTP to get full key
    - 2^32 -1 (almost 4.3 billion)
    - Journaling File System
      - Allows for reversing of the file system in case of system crash/error for recoverability.
        - Change tracking file - about a week
        - Journaling file - about 24hrs

  - Forensics tools will show the NTFS system files
    - $BITMAP - tracks allocating
    - $LOGFILE - Journaling
      - Operation codes, some arcane but lots of variance in what can be recorded

    - $MFT -
      - Database-like and very structured
      - MFT records allocated as files/folders created

        - "MFT-Zone" reserved for MFT growth (filesystem DMZ almost), prevents MFT fragmentation
        - Long and short file names are automatically created behind the scenes most files
        - No win32 API to modify the $FILE_NAME timestamps

    - Timestamp Anomalies
        - $Standard_Information time is before $File_Name times
        - All 0 nano second times (EXTREMELY IMPROBABLE)
        - Use "istat" to compare $standard_informatoin and $file_name times
            - Also analyzeFT.py
        - Always be on the look out for those "FN" times

    - $DATA
        - If the data of a file is small enough (500~600bytes), it will be assigned direct to the MFT file.
        - Attributes and Cluster content values allow for ADS in NTFS
        - ZoneIdentifier in ADS allows for identification of "Warning this document came from the internet".
            - ZoneID=3 "From the internet"

    - $I30
        - NTFS Index allocation files
        - Directory Listing file
            - Entries are written in alphabetical order
            - Files deleted are removed from enteries, but slack space is left at the end of the file.
            - Files that are later in the alphabet have a higher chance of remaining in $I30 index files slack space.
            - **Windows Index Slack Parser** (wisp - image /cases/image.raw (won't take E01 -bodyfile > /Cases/wisp.bodyfile.) = Timeline file for $I30 slack space

- $USNJRNL -
    - More detailed information embedded about journaling actions
        - Most NTFT journaling log file tools are very manual, low
        - (Windows) NTFS $Logfile Parser
        - (SIFT) Main.py
        - (SIFT) jp (from TZ works)
        - (Windows) Ntfs_linker.exe

    - Advanced NTFS Jounal Parser (ANJP) - **Current best of breed**
        - Extract $MFT, $Logfile, $J (USN)files
        - Takes 30mins~1hr to process

Information / Setup

Sunday, March 25, 2018 8:00 PM

Challenge Things of Note

Thursday, April 5, 2018 10:07 AM

- String Searching with memdump
  - Vol.py -f memory.img memdump -n csrss --dump-dir=.
  - Vol.py -f memory.img memdump -n conhost --dump-dir=.

- strings -t d -e l *.dmp >> conhost.uni
- grep -I "command prompt" conhost.uni

- Conhost contains all command lines in memory
  - Command line
  - Powershell

- cmdscan and consoles (lower percentage of success)
  - Cmdscan = command entered
  - Consoles = command entered and Output

GCFA

Sunday, March 25, 2018 8:01 PM

- MP3's on portal within a week of finish
- 3 hours
- 115 questions
- 40 ~ 80 additional hours of studying
- 3~4weeks later practice exam

xxd command?

Memory Analysis

Wednesday, April 4, 2018 5:58 PM

Step1

Step 1 - Identify Rogue Processes

Wednesday, April 4, 2018 6:01 PM

Name, path, parent, command line, start time and SIDs

Command	Output	Suspect Artifacts
vol.py -f APT.img imageinfo	INFO : volatility.debug : Determining profile based on KDBG search... Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86) AS Layer1 : IA32PagedMemoryPae (Kernel AS) AS Layer2 : FileAddressSpace (/cases/example-memory-images/APT.img) PAE type : PAE DTB : 0x319000L KDBG : 0x80545b60L Number of Processors : 1 Image Type (Service Pack) : 3 KPCR for CPU 0 : 0xffdff000L KUSER_SHARED_DATA : 0xffdf0000L Image date and time : 2009-05-05 19:28:57 UTC+0000 Image local date and time : 2009-05-05 15:28:57 -0400	- Volatility Profile = WinXPSP3x86 - Image Type (Service Pack) : 3
vol.py -f APT.img --profile=WinXPSP3x86 pslist	0x823c8830 System 4 0 55 254 ------ 0 0x8230aad8 smss.exe 564 4 3 19 ------ 0 2009-04-16 16:10:01 UTC+0000 0x822ca2c0 csrss.exe 636 564 10 356 0 0 2009-04-16 16:10:06 UTC+0000 0x81f63020 winlogon.exe 660 564 16 502 0 0 2009-04-16 16:10:06 UTC+0000 0x81f22020 services.exe 704 660 15 254 0 0 2009-04-16 16:10:06 UTC+0000 0x82164da0 lsass.exe 716 660 21 342 0 0 2009-04-16 16:10:06 UTC+0000 0x822cb458 vmacthlp.exe 872 704 1 25 0 0 2009-04-16 16:10:07 UTC+0000 0x81e54da0 svchost.exe 884 704 17 208 0 0 2009-04-16 16:10:07 UTC+0000 0x81da4590 svchost.exe 968 704 10 241 0 0 2009-04-16 16:10:07 UTC+0000 0x81f739b0 svchost.exe 1088 704 70 1445 0 0 2009-04-16 16:10:07 UTC+0000 0x8232c020 svchost.exe 1140 704 5 60 0 0 2009-04-16 16:10:08 UTC+0000 0x81e91da0 svchost.exe 1212 704 14 208 0 0 2009-04-16 16:10:09 UTC+0000 0x8219b630 spoolsv.exe 1512 704 10 129 0 0 2009-04-16 16:10:10 UTC+0000 0x81da71a8 explorer.exe 1672 1624 15 586 0 0 2009-04-16 16:10:10 UTC+0000 0x81f1c7e8 VMwareTray.exe 1984 1672 1 37 0 0 2009-04-16 16:10:11 UTC+0000 0x81dc1a78 VMwareUser.exe 2004 1672 8 228 0 0 2009-04-16 16:10:11 UTC+0000 0x81f1a650 ctfmon.exe 2020 1672 1 71 0 0 2009-04-16 16:10:11 UTC+0000 0x81dc2570 VMwareService.e 1032 704 3 175 0 0 2009-04-16 16:10:16 UTC+0000 0x81d33628 alg.exe 464 704 6 105 0 0 2009-04-16 16:10:21 UTC+0000 0x81f96220 wscntfy.exe 1260 1088 1 39 0 0 2009-04-16 16:10:22 UTC+0000 0x8231eda0 msiexec.exe 1464 704 6 294 0 0 2009-04-16 16:11:02 UTC+0000 0x81e4d648 cmd.exe 840 1672 1 33 0 0 2009-05-05 15:56:24 UTC+0000 0x81dbdda0 iexplore.exe 796 884 8 152 0 0 2009-05-05 19:28:28 UTC+0000 0x82161558 MIRAgent.exe 456 840 1 77 0 0 2009-05-05 19:28:40 UTC+0000	**Suspect Chain with CMD.exe involved and no PPID (All PPID 1672)** 0x81da71a8 explorer.exe 1672 1624 15 586 0 0 2009-04-16 16:10:10 UTC+0000 0x81f1c7e8 VMwareTray.exe 1984 1672 1 37 0 0 2009-04-16 16:10:11 UTC+0000 0x81dc1a78 VMwareUser.exe 2004 1672 8 228 0 0 2009-04-16 16:10:11 UTC+0000 0x81f1a650 ctfmon.exe 2020 1672 1 71 0 0 2009-04-16 16:10:11 UTC+0000 0x81e4d648 cmd.exe 840 1672 1 33 0 0 2009-05-05 15:56:24 UTC+0000 **Suspect Parent Process (PID 884)** 0x81e54da0 svchost.exe 884 704 17 208 0 0 2009-04-16 16:10:07 UTC+0000 0x81dbdda0 iexplore.exe 796 884 8 152 0 0 2009-05-05 19:28:28 UTC+0000 **Suspect Naming or Function as Service(PID 1032 & 1464)** 0x81dc2570 VMwareService.e 1032 704 3 175 0 0 2009-04-16 16:10:16 UTC+0000 0x8231eda0 msiexec.exe 1464 704 6 294 0 0 2009-04-16 16:11:02 UTC+0000
vol.py -f APT.img --profile=WinXPSP3x86 pstree		Process 1672 appears to have been instanciated by VMWare to run the MIRAgent.exe (Mandiant Active Response) - Looks to be IR related. 0x81da71a8:explorer.exe 1672 1624 15 586 2009-04-16 16:10:10 UTC+0000 . 0x81f1c7e8:VMwareTray.exe 1984 1672 1 37 2009-04-16 16:10:11 UTC+0000 . 0x81e4d648:cmd.exe 840 1672 1 33 2009-05-05 15:56:24 UTC+0000 .. 0x82161558:MIRAgent.exe 456 840 1 77 2009-05-05 19:28:40 UTC+0000 **svchost.exe spawning iexpore.exe as process.** .... 0x81e54da0:svchost.exe 884 704 17 208 2009-04-16 16:10:07 UTC+0000 ..... 0x81dbdda0:iexplore.exe 796 884 8 152 2009-05-05 19:28:28 UTC+0000
vol.py -f APT.img --profile=WinXPSP3x86 psinfo -p 1464	Volatility Foundation Volatility Framework 2.6 Process Information: Process: msiexec.exe PID: 1464 Parent Process: services.exe PPID: 704 Creation Time: 2009-04-16 16:11:02 UTC+0000 Process Base Name(PEB): msiexec.exe Command Line(PEB): C:\WINDOWS\system32\msiexec.exe /V VAD and PEB Comparison: Base Address(VAD): 0x1000000 Process Path(VAD): \WINDOWS\system32\msiexec.exe Vad Protection: PAGE_EXECUTE_WRITECOPY Vad Tag: Vad Base Address(PEB): 0x1000000 Process Path(PEB): C:\WINDOWS\system32\msiexec.exe Memory Protection: PAGE_EXECUTE_WRITECOPY Memory Tag: Vad Similar Processes: C:\WINDOWS\system32\msiexec.exe msiexec.exe(1464) Parent:services.exe(704) Start:2009-04-16 16:11:02 UTC+0000 Suspicious Memory Regions: 0x12f0000(No PE/Possibly Code) Protection: PAGE_EXECUTE_READWRITE Tag: VadS 0x1350000(No PE/Possibly Code) Protection: PAGE_EXECUTE_READWRITE Tag: VadS	**MSIEXEC running as a service is abnormal.** **PID1464** Appears to have code injection artifacts based on output, remember for later.
vol.py -f APT.img --profile=WinXPSP3x86 malprocfind	Offset ProcessName PID PPID Name Path Priority Cmdline User Sess Time CMD PHollow SPath ---------- ---------- ----- ----- ----- ----- ---------- ------- ----- ----- ----- ----- ------- ----- 0x81f739b0 svchost.exe 1088 True True True True True True True True True True True True 0x823c8830 system 4 True True True True True True None True True True 0x81da71a8 explorer.exe 1672 True True True True True True True True True True True True 0x81da4590 svchost.exe 968 True True True True True True True True True True True True 0x81dbdda0 iexplore.exe 796 False True True True True False True True True True True True 0x8230aad8 smss.exe 564 True True True True True True None True True True True 0x81e54da0 svchost.exe 884 True True True True True True True True True True True True 0x81e91da0 svchost.exe 1212 True True True True True True True True True True True True 0x81f22020 services.exe 704 True True True True True True True True True True True True 0x81e4d648 cmd.exe 840 True True True True True True True True True True True True 0x82164da0 lsass.exe 716 True True True True True True True True True True False True 0x8219b630 spoolsv.exe 1512 True True True True True True True True True True True True 0x8232c020 svchost.exe 1140 True True True True True True True True True True True True 0x81f63020 winlogon.exe 660 True True True True True True True True True True True True 0x822ca2c0 csrss.exe 636 True True True True True True True True True True True True Unusual process counts: ---------- Processes without running parent process: ---------- PID 1672 Offset: 0x81da71a8 Name: explorer.exe PID 4 Offset: 0x823c8830 Name: System	System and Explorer.exe**(PID 1672)** running without a parent process?**Very unusual**.

vol.py -f APT.img --profile=WinXPSP3x86 pstotal --output=dot --output-file=APT.dot	Image not show	Process relationship inspection appears odd, numerous processes are indicated as running but absent from pslist vs psscan. Multiple "system" processes exist. One running, one hidden from view. **Possible rootkit?**
vol.py -f APT.img --profile=WinXPSP3x86 psxview	Offset(P) Name PID pslist psscan thrdproc pspcid csrss session deskthrd ExitTime ---------- -------------------- ------ ------ ------ -------- ------ ------ ------- -------- -------- 0x02163020 winlogon.exe 660 True True True True True True True 0x02122020 services.exe 704 True True True True True True True 0x0211a650 ctfmon.exe 2020 True True True True True True True 0x01fa71a8 explorer.exe 1672 True True True True True True True 0x0252c020 svchost.exe 1140 True True True True True True True 0x0204d648 cmd.exe 840 True True True True True True True 0x01fc1a78 VMwareUser.exe 2004 True True True True True True True 0x02054da0 svchost.exe 884 True True True True True True True 0x02196220 wscntfy.exe 1260 True True True True True True True 0x021739b0 svchost.exe 1088 True True True True True True True 0x01fa4590 svchost.exe 968 True True True True True True True 0x02361558 MIRAgent.exe 456 True True True True True True True 0x02364da0 lsass.exe 716 True True True True True True False 0x0211c7e8 VMwareTray.exe 1984 True True True True True True True 0x02091da0 svchost.exe 1212 True True True True True True True 0x01fbdda0 iexplore.exe 796 True True True True True True True 0x024cb458 vmacthlp.exe 872 True True True True True True True 0x0239b630 spoolsv.exe 1512 True True True True True True True 0x0251eda0 msiexec.exe 1464 True True True True True True True 0x01f33628 alg.exe 464 True True True True True True True 0x01fc2570 VMwareService.e 1032 True True True True True True True 0x0250aad8 smss.exe 564 True True True True False False False 0x025c8830 System 4 True True True True False False False 0x024ca2c0 csrss.exe 636 True True True True False True True 0x03178220 wscntfy.exe 1260 False True False False False False False 0x0c605020 svchost.exe 1140 False True False False False False False 0x0ad69da0 iexplore.exe 796 False True False False False False False 0x0edd0628 alg.exe 464 False True False False False False False 0x032b3da0 svchost.exe 884 False True False False False False False 0x0eed3628 alg.exe 464 False True False False False False False 0x10b54628 alg.exe 464 False True False False False False False 0x15934830 System 4 False True False False False False False 0x1b217da0 iexplore.exe 796 False True False False False False False 0x04097020 svchost.exe 1140 False True False False False False False 0x035c1590 svchost.exe 968 False True False False False False False 0x07b1ada0 iexplore.exe 796 False True False False False False False 0x0edd59b0 svchost.exe 1088 False True False False False False False 0x12f3dda0 svchost.exe 884 False True False False False False False	Process relationship inspection appears odd, numerous processes are indicated as running but absent from pslist vs psscan. Multiple "system" processes exist. One running, one hidden from view. **Possible rootkit?**

Step 2 - Analyze DLLs and Handles

Wednesday, April 4, 2018 6:01 PM

Name, path, parent, command line, start time and SIDs

Command	Output	Suspect Artifacts
vol.py -f APT.img --profile=WinXPSP3x86 dlllist -p 1672	C:\WINDOWS\AppPatch\AcGenral.DLL C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll C:\WINDOWS\WinSxS\x86_Microsoft.VC80.CRT_1fc8b3b9a1e18e3b_8.0.50727.3053_x-ww_b80fa8ca\MSVCR80.dll C:\WINDOWS\system32\irykmmww.dll	4 DLL's here seem suspect
vol.py -f APT.img --profile=WinXPSP3x86 getsids -p 1672	Volatility Foundation Volatility Framework 2.6 explorer.exe (1672): S-1-5-21-583907252-1123561945-1606980848-1003 (demo) explorer.exe (1672): S-1-5-21-583907252-1123561945-1606980848-513 (Domain Users) explorer.exe (1672): S-1-1-0 (Everyone) explorer.exe (1672): S-1-5-32-544 (Administrators) explorer.exe (1672): S-1-5-32-545 (Users) explorer.exe (1672): S-1-5-4 (Interactive) explorer.exe (1672): S-1-5-11 (Authenticated Users) explorer.exe (1672): S-1-5-5-0-47384 (Logon Session) explorer.exe (1672): S-1-2-0 (Local (Users with the ability to log in locally))	Explorer.exe appears to be owned by "demo" user
vol.py -f APT.img --profile=WinXPSP3x86 handles -p 1672 -t mutant	Offset(V) Pid Handle Access Type Details ---------- ------ ---------- ---------- ---------------- ------- 0x822f2108 1672 0x20 0x1f0001 Mutant SHIMLIB_LOG_MUTEX 0x81e76ca0 1672 0x80 0x1f0001 Mutant 0x822b7950 1672 0x88 0x1f0001 Mutant 0x81e55b00 1672 0xd8 0x1f0001 Mutant ExplorerIsShellMutex 0x821607f0 1672 0xdc 0x120001 Mutant ShimCacheMutex 0x81f20710 1672 0x16c 0x1f0001 Mutant 0x81f9bb20 1672 0x25c 0x1f0001 Mutant CTF.LBES.MutexDefaultS-1-5-21-583907252-1123561945-1606980848-1003 0x821650f0 1672 0x260 0x1f0001 Mutant CTF.Compart.MutexDefaultS-1-5-21-583907252-1123561945-1606980848-1003 0x81f1a988 1672 0x264 0x1f0001 Mutant CTF.Asm.MutexDefaultS-1-5-21-583907252-1123561945-1606980848-1003 0x81f1a560 1672 0x268 0x1f0001 Mutant CTF.Layouts.MutexDefaultS-1-5-21-583907252-1123561945-1606980848-1003 0x81dbed50 1672 0x26c 0x1f0001 Mutant CTF.TMD.MutexDefaultS-1-5-21-583907252-1123561945-1606980848-1003 0x81dbe858 1672 0x278 0x1f0001 Mutant 0x82165ec8 1672 0x2cc 0x1f0001 Mutant 0x81e5ccb0 1672 0x2e8 0x1f0001 Mutant 0x81eaaa68 1672 0x2f0 0x1f0001 Mutant 0x81dc0020 1672 0x31c 0x1f0001 Mutant 0x82164668 1672 0x34c 0x1f0001 Mutant _SHuassist.mtx 0x81dbfea8 1672 0x3e0 0x1f0001 Mutant ZonesCounterMutex 0x81e615e8 1672 0x408 0x1f0001 Mutant CTF.TimListCache.FMPDefaultS-1-5-21-583907252-1123561945-1606980848-1003MUTEX.DefaultS-1-5-21-583907252-1123561945-1606980848-1003 0x822de0e0 1672 0x40c 0x1f0001 Mutant MSCTF.Shared.MUTEX.EKG 0x81dc0ea8 1672 0x410 0x1f0001 Mutant ZoneAttributeCacheCounterMutex 0x81dc0ea8 1672 0x414 0x1f0001 Mutant ZoneAttributeCacheCounterMutex 0x81e5e180 1672 0x418 0x1f0001 Mutant ZonesCacheCounterMutex 0x81f1b180 1672 0x41c 0x1f0001 Mutant ZonesLockedCacheCounterMutex 0x81daa580 1672 0x420 0x1f0001 Mutant pork_bun 0x822de0e0 1672 0x448 0x1f0001 Mutant MSCTF.Shared.MUTEX.EKG 0x822cef48 1672 0x450 0x1f0001 Mutant MSCTF.Shared.MUTEX.AEH 0x81e499a8 1672 0x454 0x1f0001 Mutant MSCTF.Shared.MUTEX.MIG 0x822c5dd0 1672 0x464 0x1f0001 Mutant HGFSMUTEX 0x81e499a8 1672 0x474 0x1f0001 Mutant MSCTF.Shared.MUTEX.MIG 0x81e4b7b8 1672 0x4ac 0x1f0001 Mutant 0x81e4b738 1672 0x4b4 0x1f0001 Mutant 0x81f2e4c8 1672 0x574 0x1f0001 Mutant MSCTF.Shared.MUTEX.IEF 0x82195450 1672 0x610 0x100000 Mutant c:!documents and settings!demo!local settings!temporary internet files!content.ie5! 0x81dca030 1672 0x62c 0x100000 Mutant c:!documents and settings!demo!cookies! 0x821954a0 1672 0x630 0x100000 Mutant _!MSFTHISTORY!_ 0x81f0e448 1672 0x644 0x100000 Mutant c:!documents and settings!demo!local settings!history!history.ie5! 0x81df0a80 1672 0x650 0x100000 Mutant WininetStartupMutex 0x81e86698 1672 0x660 0x1f0001 Mutant WininetConnectionMutex 0x821645c0 1672 0x664 0x100000 Mutant WininetProxyRegistryMutex 0x822e0a58 1672 0x7e0 0x1f0001 Mutant MSCTF.Shared.MUTEX.AP 0x81e4eba8 1672 0x92c 0x1f0001 Mutant MSCTF.Shared.MUTEX.EKE 0x822de0e0 1672 0x930 0x1f0001 Mutant MSCTF.Shared.MUTEX.EKG 0x81e4e838 1672 0x94c 0x1f0001 Mutant MSCTF.Shared.MUTEX.MME 0x81e4eba8 1672 0x950 0x1f0001 Mutant MSCTF.Shared.MUTEX.EKE 0x81e4e9f0 1672 0x960 0x1f0001 Mutant MSCTF.Shared.MUTEX.IPG 0x81db9a18 1672 0x96c 0x1f0001 Mutant MSCTF.Shared.MUTEX.EIH	These mutex/mutants look odd 0x82164668 1672 0x34c 0x1f0001 Mutant _SHuassist.mtx Multiple google hits for Trojan 0x81daa580 1672 0x420 0x1f0001 Mutant pork_bun Not much google
ol.py -f APT.img --profile=WinXPSP3x86 handles -p 796 -t file	Volatility Foundation Volatility Framework 2.6 Offset(V) Pid Handle Access Type Details ---------- ------ ---------- ---------- ---------------- ------- 0x821ace98 796 0xc 0x100020 File \Device\HarddiskVolume1\WINDOWS\system32 0x82189028 796 0x578 0x12019f File \Device\NamedPipe\ROUTER 0x81db29b0 796 0x5a8 0x1200a0 File \Device\Ip 0x81db2a48 796 0x5ac 0x100003 File \Device\Ip 0x81f0f7f0 796 0x5b0 0x1200a0 File \Device\Ip 0x81f0f888 796 0x5b4 0x120116 File \Device\Tcp 0x822f7890 796 0x5b8 0x1200a0 File \Device\Tcp 0x822ea0d0 796 0x5bc 0x12019f File \Device\NamedPipe\ROUTER 0x81df14a8 796 0x5c4 0x21f01ff File \Device\Afd\AsyncConnectHlp 0x81f07b88 796 0x694 0x12019f File \Device\HarddiskVolume1\WINDOWS\system32\config\systemprofile\Local Settings\History\History.IE5\index.dat 0x81f03b70 796 0x6ec 0x100020 File \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83 0x82149c88 796 0x718 0x12019f File \Device\HarddiskVolume1\WINDOWS\system32\config\systemprofile\Cookies\index.dat 0x81f19978 796 0x724 0x12019f File \Device\HarddiskVolume1\WINDOWS\system32\config\systemprofile\Local Settings\Temporary Internet Files\Content.IE5\index.dat 0x822e9578 796 0x758 0x12019f File \Device\irykmmww 0x81d298c8 796 0x788 0x100020 File \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83 0x81f224c0 796 0x7a4 0x12019f File \Device\WMIDataDevice 0x81dcc028 796 0x7b0 0x12019f File \Device\WMIDataDevice 0x81dd0338 796 0x7c0 0x100020 File \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83 0x81ddb8a0 796 0x7c4 0x100020 File \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83 0x81f59f90 796 0x7c8 0x100001 File \Device\KsecDD	irykmmww.sys found earlier -> C:\WINDOWS\system32\irykmmww.dll

Step 3 - Review Network Artifacts

Wednesday, April 4, 2018 6:01 PM

Suspicious ports, connections, and processes. Be wary of tunneling.

Command	**Output**	**Suspect Artifacts**
vol.py -f APT.img --profile=WinXPSP3x86 connscan	Volatility Foundation Volatility Framework 2.6 Offset(P) Local Address Remote Address Pid ---------- ---------- ---------- --- 0x0205ece0 192.168.157.10:1050 222.128.1.2:443 1672 0x020611f8 192.168.157.10:1053 218.85.133.23:89 796 0x032c01f8 192.168.157.10:1053 218.85.133.23:89 796 0x0337dce0 192.168.157.10:1050 222.128.1.2:443 1672 0x08a4ace0 192.168.157.10:1050 222.128.1.2:443 1672 0x18200ce0 192.168.157.10:1050 222.128.1.2:443 1672	explorer**(1672)** talking to the internet seems suspect iexplore.exe **(796)** talking to the internet over a non-standard port (89) Review IoCs in iexplorer
vol.py -f APT.img --profile=WinXPSP3x86 getsids -p 1672	Volatility Foundation Volatility Framework 2.6 explorer.exe (1672): S-1-5-21-583907252-1123561945-1606980848-1003 (demo) explorer.exe (1672): S-1-5-21-583907252-1123561945-1606980848-513 (Domain Users) explorer.exe (1672): S-1-1-0 (Everyone) explorer.exe (1672): S-1-5-32-544 (Administrators) explorer.exe (1672): S-1-5-32-545 (Users) explorer.exe (1672): S-1-5-4 (Interactive) explorer.exe (1672): S-1-5-11 (Authenticated Users) explorer.exe (1672): S-1-5-5-0-47384 (Logon Session) explorer.exe (1672): S-1-2-0 (Local (Users with the ability to log in locally))	Explorer.exe appears to be owned by "demo" user
vol.py -f APT.img --profile=WinXPSP3x86 getsids -p 796	Volatility Foundation Volatility Framework 2.6 iexplore.exe (796): S-1-5-18 (Local System) iexplore.exe (796): S-1-5-32-544 (Administrators) iexplore.exe (796): S-1-1-0 (Everyone) iexplore.exe (796): S-1-5-11 (Authenticated Users)	Iexplore.exe appears to be owned by System - suspect.

Step 4 - Code Injection Check

Wednesday, April 4, 2018 6:01 PM

Command	Output	Suspect Artifacts
vol.py -f APT.img --profile=WinXPSP3x86 malfind --dump-dir=./malfind-output > APT_malfind.txt Less | APT_malfind.txt		
strings -a process.0x81da71a8.0x1820000.dmp | more	B~k!C~ |!WE~& exploder.exe admin 222.128.1.2 pork_bun StubPath SOFTWARE\Classes\http\shell\open\command explorer.exe Software\Microsoft\Active Setup\Installed Components\ C:\WINDOWS\system32\exploder.exe C:\WINDOWS\system32\exploder.exe otA." admin zh-pork-demo SOFTWARE\Microsoft\Windows\CurrentVersion\Run porkbun	- 222.128.1.2 - pork_bun - C:\WINDOWS\system32\exploder.exe - C:\WINDOWS\system32\exploder.exe

Step 5 - Rootkit Check

Wednesday, April 4, 2018 6:01 PM

Check SSDT, IDT, IRP and inline hooks

Command	Output	Suspect Artifacts
vol.py -f APT.img --profile=WinXPSP3x86 ssdt | egrep -v '(ntoskrnl|win32k.sys)'	Volatility Foundation Volatility Framework 2.6 [x86] Gathering all referenced SSDTs from KTHREADs... Finding appropriate address space for tables... SSDT[0] at 80501b9c with 284 entries Entry 0x0042: 0xf836fe9c (NtDeviceIoControlFile) owned by irykmmww.sys Entry 0x0047: 0xf83706dc (NtEnumerateKey) owned by irykmmww.sys Entry 0x0049: 0xf837075e (NtEnumerateValueKey) owned by irykmmww.sys Entry 0x0077: 0xf837028f (NtOpenKey) owned by irykmmww.sys Entry 0x0091: 0xf8370a8c (NtQueryDirectoryFile) owned by irykmmww.sys Entry 0x00ad: 0xf836fe3e (NtQuerySystemInformation) owned by irykmmww.sys Entry 0x00b1: 0xf837091a (NtQueryValueKey) owned by irykmmww.sys	**Service System Descriptor Table (SSDT) hooking** C:\windows\system32\drivers\irykmmww.sys

Step 6 - Dump Suspect Artifacts

Wednesday, April 4, 2018 6:01 PM

Review strings, antivirus scans, and reverse-engineer

Command	**Output**	**Suspect Artifacts**
vol.py -f APT.img --profile=WinXPSP3x86 dlldump --dump-dir ./ -b 0xf836f000	A file module.1672.1fa71a8.f836f000.dll	Dumped https://www.virustotal.com/#/file/a80d0353c34c20a50a35771e3794de255e9030d8b7ab21ef6d1953afa692dd97/detection
vol.py -f APT.img --profile=WinXPSP3x86 svcscan -v	Offset: 0x38ab98 Order: 252 Start: SERVICE_DEMAND_START Process ID: - Service Name: irykmmww Display Name: irykmmww Service Type: SERVICE_KERNEL_DRIVER Service State: SERVICE_RUNNING Binary Path: \Driver\irykmmww ServiceDll: ImagePath: \??\C:\WINDOWS\system32\drivers\irykmmww.sys FailureCommand:	Persistence Auto Start Service with driver
vol.py -f APT.img --profile=WinXPSP3x86 svcscan -v	fset: 0x383b18 Order: 52 Start: SERVICE_AUTO_START Process ID: 1088 Service Name: dmserver Display Name: Logical Disk Manager Service Type: SERVICE_WIN32_SHARE_PROCESS Service State: SERVICE_RUNNING Binary Path: C:\WINDOWS\System32\svchost.exe -k netsvcs ServiceDll: %SystemRoot%\System32\irykmmww.d1l ImagePath: %SystemRoot%\System32\svchost.exe -k netsvcs FailureCommand:	Persistence Auto Start Service with driver

Memory Analysis

Wednesday, April 4, 2018 5:58 PM

Step 1 - Identify Rogue Processes

Wednesday, April 4, 2018 6:01 PM

Name, path, parent, command line, start time and SIDs

Command	Output	Suspect Artifacts

Step 2 - Analyze DLLs and Handles

Wednesday, April 4, 2018 6:01 PM

Review the DLLs and Handles

Command	Output	Suspect Artifacts

Step 3 - Review Network Artifacts

Wednesday, April 4, 2018 6:01 PM

Suspicious ports, connections, and processes. Be wary of tunneling.

Command	Output	Suspect Artifacts

Step 4 - Code Injection Check

Wednesday, April 4, 2018 6:01 PM

Injected memory sections and process hollowing

Command	Output	Suspect Artifacts

Step 5 - Rootkit Check

Wednesday, April 4, 2018 6:01 PM

Check SSDT, IDT, IRP and inline hooks

Command	Output	Suspect Artifacts

Step 6 - Dump Suspect Artifacts

Wednesday, April 4, 2018 6:01 PM

Review strings, antivirus scans, and reverse-engineer

Command	Output	Suspect Artifacts