SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

A Guide to GIAC Certified Forensic Analyst

Authored by: Ishrag Hamid

FOR508

NON POTESTIS CELARE

ADVANCED INCIDENT RESPONSE AND THREAT HUNTING

About the Author

Ishrag Hamid, a cybersecurity specialist with a Master's degree in Cybersecurity, is passionate about incident response and threat hunting. With a strong foundation in SOC operations and a comprehensive set of certifications from renowned organizations such as SANS, CompTIA, eLearnSecurity, and Fortinet, she strives to empower others in their cybersecurity journey.

She authored this book during her preparation for the GIAC Certified Forensic Analyst (GCFA) certification. This comprehensive guide aims to equip readers with the knowledge and strategies necessary to successfully pass the exam on their first attempt.

Key features of this book include:

- In-depth coverage of core GCFA certification aspects: exam objectives, suitable candidates, prerequisites, training costs, exam format, test centers, access expiration, retaking procedures, and extensions.
- Detailed exploration of the five primary GCFA topics: Advanced Incident Response and Threat Hunting, Intrusion Analysis, Memory Forensics in IR & TH Timeline Analysis, and Advanced Adversary and Anti-Forensics Detection.
- Clear and concise summaries, comparison and engaging mind maps to facilitate understanding.
- Practical tips and strategies for effective exam preparation, including study techniques and recommendations for practice exams.

This book serves as a valuable resource for aspiring GCFA professionals, providing them with the knowledge they need to pass the cert.

Contents

Introduction

SANS Institute, established in 1989, is a leading organization specializing in information security training and certifications. It provides hands-on training, research, and resources for cybersecurity, IT, and related professionals.

SANS offers a wide range of cybersecurity training programs, including:

- Cyber Defense & Blue Team Operations
- Offensive Operations
- Digital Forensics and Incident Response
- Cloud Security
- ICS/SCADA Security
- Cyber Security Leadership

To further guide professionals in their cybersecurity career development, SANS has created a Cyber Security Skills Roadmap

One of SANS's prominent courses is FOR508: Advanced Incident Response, Threat Hunting, and Digital Forensics. This course equips participants with the essential skills to detect, investigate, and respond to advanced cyber threats. It emphasizes rapid incident response, threat hunting techniques, and in-depth forensic analysis to uncover hidden threats within networks. Participants learn to analyze file systems, memory, and timelines using tools like Volatility and Plaso, focusing on identifying adversary techniques. Designed for security analysts, incident responders, and forensic experts, the course includes hands-on labs to apply knowledge to real-world scenarios and prepares attendees for the GIAC Certified Forensic Analyst (GCFA) certification.

Certification Information

Exam Certification Objectives & Outcome Statements

- Analyzing Volatile Malicious Event Artifacts
  The candidate will demonstrate an understanding of abnormal activity within the structure of Windows memory and be able to identify artifacts such as malicious processes, suspicious drivers and malware techniques such as code injection and rootkits.
- Analyzing Volatile Windows Event Artifacts
  The candidate will demonstrate an understanding of normal activity within the structure of Windows memory and be able to identify artifacts such as network connections, memory resident command line artifacts and processes, handles and threads.
- Enterprise Environment Incident Response

The candidate will demonstrate an understanding of the steps of the incident response process, attack progression, and adversary fundamentals and how to rapidly assess and analyze systems in an enterprise environment scaling tools to meet the demands of large investigations.

- File System Timeline Artifact Analysis
  The candidate will demonstrate an understanding of the Windows filesystem time structure and how these artifacts are modified by system and user activity.
- Identification of Malicious System and User Activity
  The candidate will demonstrate an understanding of the techniques required to identify and document indicators of compromise on a system, detect malware and attacker tools, attribute activity to events and accounts, and identify and compensate for anti-forensic actions using memory and disk resident artifacts.
- Identification of Normal System and User Activity
  The candidate will demonstrate an understanding of the techniques required to identify, document, and differentiate normal and abnormal system and user activity using memory and disk resident artifacts.
- Introduction to File System Timeline Forensics
  The candidate will demonstrate an understanding of the methodology required to collect and process timeline data from a Windows system.
- Introduction to Memory Forensics
  The candidate will demonstrate an understanding of how and when to collect volatile data from a system and how to document and preserve the integrity of volatile evidence.
- NTFS Artifact Analysis
  The candidate will demonstrate an understanding of core structures of the Windows filesystems, and the ability to identify, recover, and analyze evidence from any file system layer, including the data storage layer, metadata layer, and filename layer.
- Windows Artifact Analysis
  The candidate will demonstrate an understanding of Windows system artifacts and how to collect and analyze data such as system back up and restore data and evidence of application execution. For more information, refer to the following resources:
  - SANS Advanced Incident Response and Threat Hunting Training
  - GIAC Certified Forensic Analyst (GCFA)

Suitable Candidates for GCFA Certification

- Incident Response Team Members
- Threat Hunters
- SOC Analysts
- Experienced Digital Forensic Analysts
- Information Security Professionals

- Federal Agents and Law Enforcement Professionals
- Red Team Members, Penetration Testers, and Exploit Developers

Prerequisites

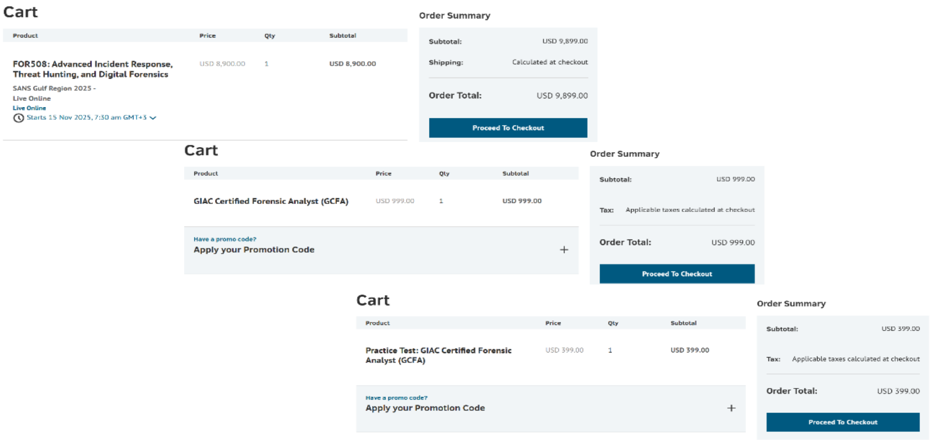
The FOR508 course provides advanced training in incident response and threat hunting, focusing on advanced persistent threats (APTs) and organized crime groups, excluding foundational topics like incident response policies and basic digital forensics; a background in FOR500: Windows Forensics is recommended, and for laptop requirements, please visit the official website SANS Advanced Incident Response and Threat Hunting Training.

Training cost

The SANS 508 course, Advanced Incident Response, Threat Hunting, and Digital Forensics, varies in cost depending on the chosen training format: OnDemand, Live Online, or In-Person. Below are the details:

- **OnDemand**: This format provides access to pre-recorded lectures and course materials, allowing participants to study at their own pace, making it ideal for those with busy schedules.
  Cost: $8,780 USD + GCFA Certification Voucher ($999).
- **Live Online**: This format involves live, interactive online sessions with instructors in real-time. Participants can engage in discussions, ask questions, and participate in virtual labs, providing a classroom-like experience from anywhere.
  Cost (Middle East): $8,900 USD + GCFA Certification Voucher ($999) Includes 2 practice tests + OnDemand Bundle ($999).
- **In-Person**: This option involves face-to-face training at a physical location, such as a training center or conference venue. Participants benefit from direct interaction with instructors and peers and hands-on access to labs and resources.
  Cost (Middle East): $8,900 USD + GCFA Certification Voucher ($999) Includes 2 practice tests + OnDemand Bundle ($999).

Additionally, you can purchase the GCFA Certification **Voucher separately** without the training, which costs **$999** (without practice tests). To add one practice test, it will cost **$399**. Practice Tests offer a simulated experience of the real exam, helping you familiarize yourself with the test interface and question styles. They serve as a tool to assess whether your preparation is sufficient. However, note that the question bank is limited, so you may encounter repeated questions if you purchase multiple practice tests. Importantly, practice exams do not include actual exam questions. GIAC recommends supplementing these practice tests with additional study methods for better preparation.



Cart

Product	Price	Qty	Subtotal
FOR508: Advanced Incident Response, Threat Hunting, and Digital Forensics SANS Gulf Region 2025 - Live Online Live Online Starts 15 Nov 2025, 7:30 am GMT+3	USD 8,900.00	1	USD 8,900.00

Order Summary

Subtotal: USD 9,899.00

Shipping: Calculated at checkout

Order Total: USD 9,899.00

Proceed To Checkout

Cart

Product	Price	Qty	Subtotal
GIAC Certified Forensic Analyst (GCFA)	USD 999.00	1	USD 999.00

Have a promo code?

Apply your Promotion Code +

Order Summary

Subtotal: USD 999.00

Tax: Applicable taxes calculated at checkout

Order Total: USD 999.00

Proceed To Checkout

Cart

Product	Price	Qty	Subtotal
Practice Test: GIAC Certified Forensic Analyst (GCFA)	USD 399.00	1	USD 399.00

Have a promo code?

Apply your Promotion Code +

Order Summary

Subtotal: USD 399.00

Tax: Applicable taxes calculated at checkout

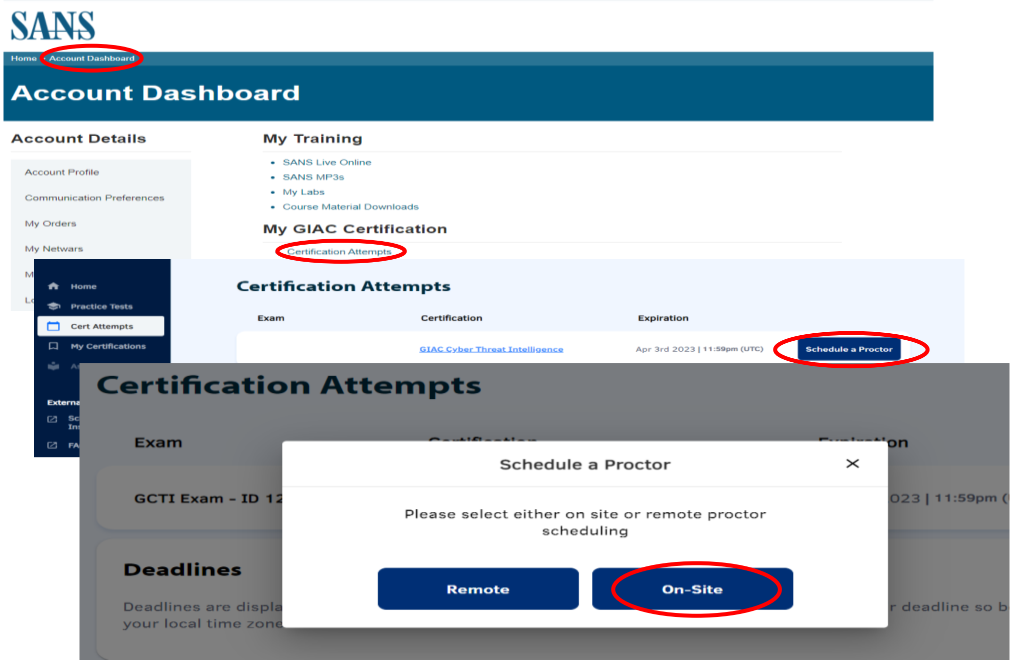
Order Total: USD 399.00

Proceed To Checkout

Test center

GIAC certification exams are web-based and require proctoring. Candidates have the option of choosing between remote proctoring through **ProctorU or onsite** proctoring at Pearson VUE testing centers.

When purchasing a certification exam without training, GIAC will validate your information within seven business days, after which the exam will be added to your account. You can then schedule your exam at any time within four months. Failure to schedule within this timeframe will result in the exam expiring.



Account Dashboard

Account Details

- Account Profile
- Communication Preferences
- My Orders
- My Netwars

My Training

- SANS Live Online
- SANS MP3s
- My Labs
- Course Material Downloads

My GIAC Certification

Certification Attempts

Certification Attempts

Exam	Certification	Expiration	
	GIAC Cyber Threat Intelligence	Apr 3rd 2023 | 11:59pm (UTC)	Schedule a Proctor

Certification Attempts

Exam

GCTI Exam - ID 12

Deadlines

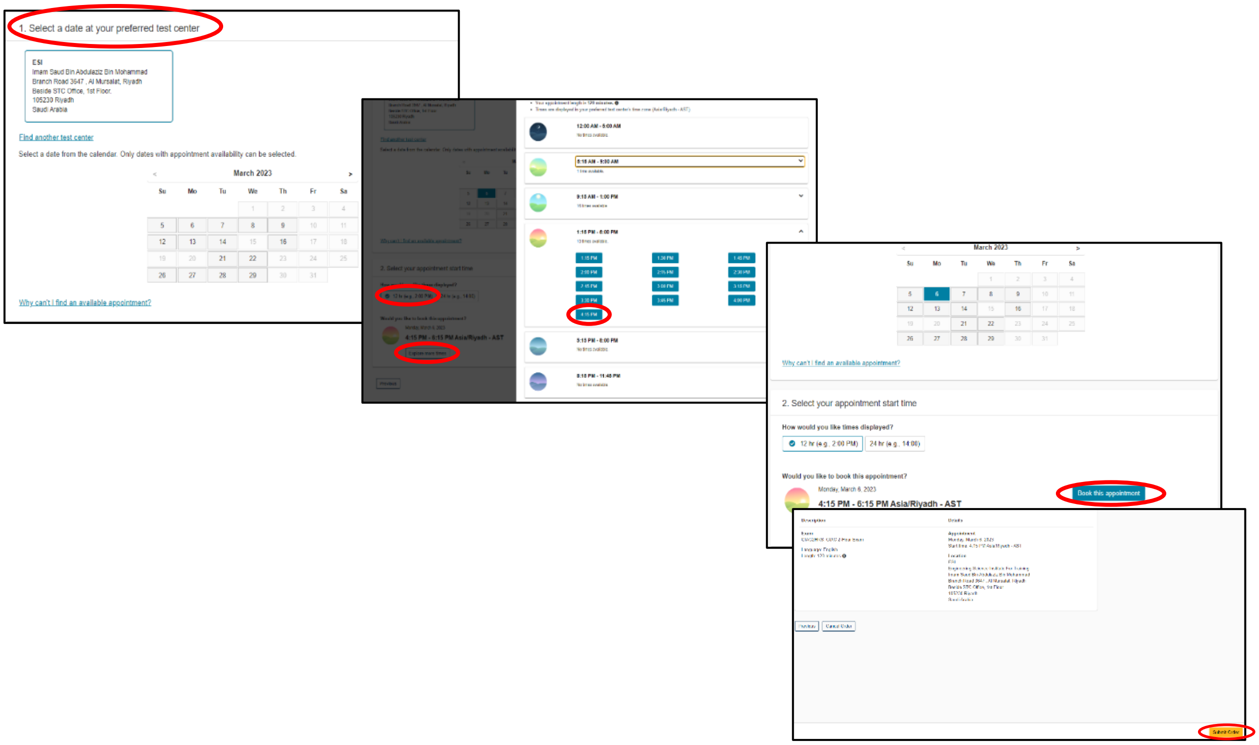
Deadlines are displa... your local time zone

Schedule a Proctor

Please select either on site or remote proctor scheduling

Remote

On-Site



1. Select a date at your preferred test center

ESI
Imam Saud Bin Abdulaziz Bin Mohammad
Branch Road 3647 , Al Mursalat, Riyadh
Beside STC Office, 1st Floor.
105230 Riyadh
Saudi Arabia

Find another test center

Select a date from the calendar. Only dates with appointment availability can be selected.

< March 2023 >

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Why can't I find an available appointment?

- Your appointment length is 120 minutes.
- Times are displayed in your preferred test center's time zone (Asia/Riyadh - AST).

12:00 AM - 5:00 AM
No times available.

5:15 AM - 9:00 AM
1 time available.

9:15 AM - 1:00 PM
16 times available.

1:15 PM - 5:00 PM
13 times available.

5:15 PM - 8:00 PM
No times available.

8:15 PM - 11:45 PM
No times available.

2. Select your appointment start time

How would you like times displayed?

● 12 hr (e.g., 2:00 PM) ○ 24 hr (e.g., 14:00)

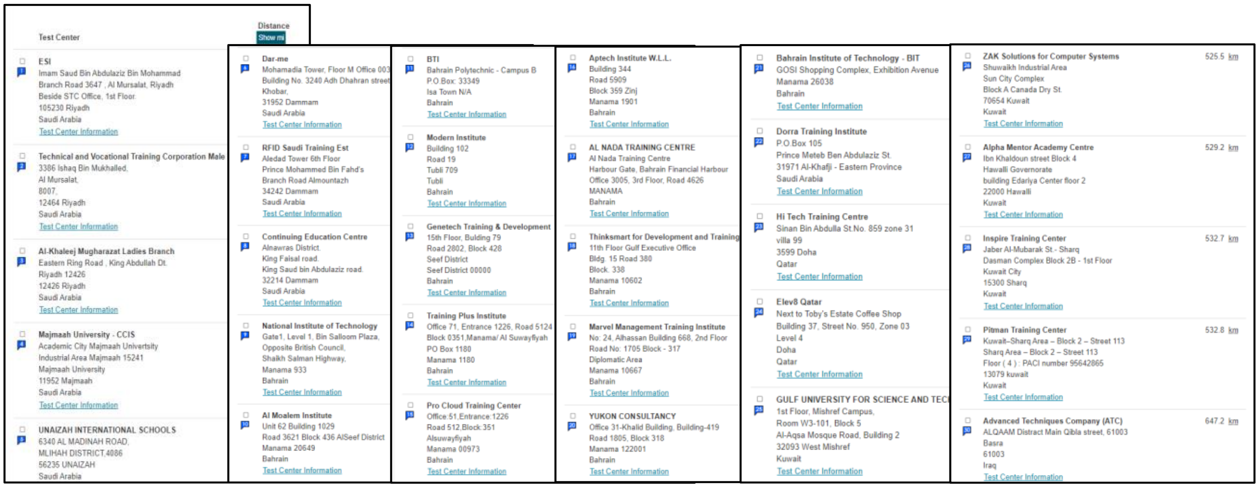
Would you like to book this appointment?

Monday, March 6, 2023
4:15 PM - 6:15 PM Asia/Riyadh - AST

Description
Exam
Language: English
Length: 120 minutes

Details
Appointment
Monday, March 6, 2023
Start time: 4:15 PM Asia/Riyadh - AST
Location
ESI
Engineering Science Institute For Training
Imam Saud Bin Abdulaziz Bin Mohammad
Branch Road 3647 , Al Mursalat, Riyadh
Beside STC Office, 1st Floor
105230 Riyadh
Saudi Arabia

Below is a list of some Pearson VUE centers.



Test Center — Distance Show mi

1 ESI
Imam Saud Bin Abdulaziz Bin Mohammad
Branch Road 3647 , Al Mursalat, Riyadh
Beside STC Office, 1st Floor.
105230 Riyadh
Saudi Arabia
Test Center Information

2 Technical and Vocational Training Corporation Male
3386 Ishaq Bin Mukhalled,
Al Mursalat,
8007,
12464 Riyadh
Saudi Arabia
Test Center Information

3 Al-Khaleej Mugharazat Ladies Branch
Eastern Ring Road , King Abdullah Dt.
Riyadh 12426
12426 Riyadh
Saudi Arabia
Test Center Information

4 Majmaah University - CCIS
Academic City Majmaah Univertsity
Industrial Area Majmaah 15241
Majmaah University
11952 Majmaah
Saudi Arabia
Test Center Information

5 UNAIZAH INTERNATIONAL SCHOOLS
6340 AL MADINAH ROAD,
MLIHAH DISTRICT,4086
56235 UNAIZAH
Saudi Arabia

6 Dar-me
Mohamadia Tower, Floor M Office 003
Building No. 3240 Adh Dhahran street
Khobar,
31952 Dammam
Saudi Arabia
Test Center Information

7 RFID Saudi Training Est
Aledad Tower 6th Floor
Prince Mohammed Bin Fahd's
Branch Road Almountazh
34242 Dammam
Saudi Arabia
Test Center Information

8 Continuing Education Centre
Alnawras District.
King Faisal road.
King Saud bin Abdulaziz road.
32214 Dammam
Saudi Arabia
Test Center Information

9 National Institute of Technology
Gate1, Level 1, Bin Salloom Plaza,
Opposite British Council,
Shaikh Salman Highway,
Manama 933
Bahrain
Test Center Information

10 Al Moalem Institute
Unit 62 Building 1029
Road 3621 Block 436 AlSeef District
Manama 20649
Bahrain
Test Center Information

11 BTI
Bahrain Polytechnic - Campus B
P.O.Box: 33349
Isa Town N/A
Bahrain
Test Center Information

12 Modern Institute
Building 102
Road 19
Tubli 709
Tubli
Bahrain
Test Center Information

13 Genetech Training & Development
15th Floor, Bulding 79
Road 2802, Block 428
Seef District
Seef District 00000
Bahrain
Test Center Information

14 Training Plus Institute
Office 71, Entrance 1226, Road 5124
Block 0351,Manama/ Al Suwayfiyah
PO Box 1180
Manama 1180
Bahrain
Test Center Information

15 Pro Cloud Training Center
Office:51,Entrance:1226
Road 512,Block:351
Alsuwayfiyah
Manama 00973
Bahrain
Test Center Information

16 Aptech Institute W.L.L.
Building 344
Road 5909
Block 359 Zinj
Manama 1901
Bahrain
Test Center Information

17 AL NADA TRAINING CENTRE
Al Nada Training Centre
Harbour Gate, Bahrain Financial Harbour
Office 3005, 3rd Floor, Road 4626
MANAMA
Bahrain
Test Center Information

18 Thinksmart for Development and Training
11th Floor Gulf Executive Office
Bldg. 15 Road 380
Block. 338
Manama 10602
Bahrain
Test Center Information

19 Marvel Management Training Institute
No: 24, Alhassan Building 668, 2nd Floor
Road No: 1705 Block - 317
Diplomatic Area
Manama 10667
Bahrain
Test Center Information

20 YUKON CONSULTANCY
Office 31-Khalid Building, Building-419
Road 1805, Block 318
Manama 122001
Bahrain
Test Center Information

21 Bahrain Institute of Technology - BIT
GOSI Shopping Complex, Exhibition Avenue
Manama 26038
Bahrain
Test Center Information

22 Dorra Training Institute
P.O.Box 105
Prince Meteb Ben Abdulaziz St.
31971 Al-Khafji - Eastern Province
Saudi Arabia
Test Center Information

23 Hi Tech Training Centre
Sinan Bin Abdulla St.No. 859 zone 31
villa 99
3599 Doha
Qatar
Test Center Information

24 Elev8 Qatar
Next to Toby's Estate Coffee Shop
Building 37, Street No. 950, Zone 03
Level 4
Doha
Qatar
Test Center Information

25 GULF UNIVERSITY FOR SCIENCE AND TECI
1st Floor, Mishref Campus,
Room W3-101, Block 5
Al-Aqsa Mosque Road, Building 2
32093 West Mishref
Kuwait
Test Center Information

26 ZAK Solutions for Computer Systems — 525.5 km
Shuwaikh Industrial Area
Sun City Complex
Block A Canada Dry St.
70654 Kuwait
Kuwait
Test Center Information

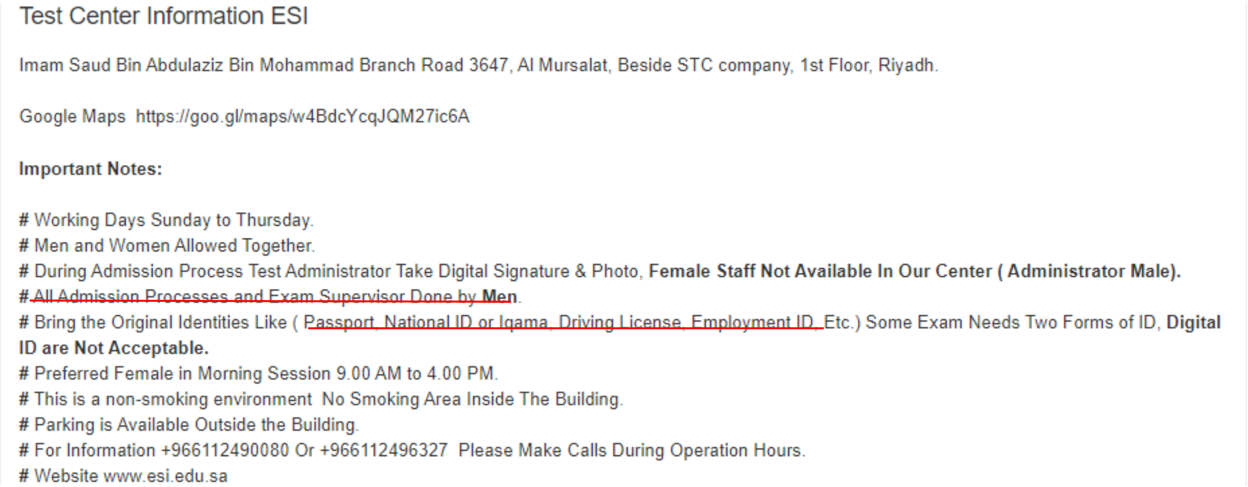
27 Alpha Mentor Academy Centre — 529.2 km
Ibn Khaldoun street Block 4
Hawalli Governorate
building Edariya Center floor 2
22000 Hawalli
Kuwait
Test Center Information

28 Inspire Training Center — 532.7 km
Jaber Al-Mubarak St.- Sharq
Dasman Complex Block 2B - 1st Floor
Kuwait City
15300 Sharq
Kuwait
Test Center Information

29 Pitman Training Center — 532.8 km
Kuwait–Sharq Area – Block 2 – Street 113
Sharq Area – Block 2 – Street 113
Floor (4) : PACI number 95642865
13079 kuwait
Kuwait
Test Center Information

30 Advanced Techniques Company (ATC) — 647.2 km
ALQAAM Distract Main Qibla street, 61003
Basra
61003
Iraq
Test Center Information

After selecting your preferred test center, you will receive important information about the location, available dates, test center policies, and any specific requirements or instructions for your exam day. Make sure to review this information carefully to ensure a smooth testing experience.



Test Center Information ESI

Imam Saud Bin Abdulaziz Bin Mohammad Branch Road 3647, Al Mursalat, Beside STC company, 1st Floor, Riyadh.

Google Maps https://goo.gl/maps/w4BdcYcqJQM27ic6A

Important Notes:

Working Days Sunday to Thursday.
Men and Women Allowed Together.
During Admission Process Test Administrator Take Digital Signature & Photo, **Female Staff Not Available In Our Center (Administrator Male).**
~~All Admission Processes and Exam Supervisor Done by **Men**~~.
Bring the Original Identities Like (Passport, National ID or Iqama, Driving License, Employment ID, Etc.) Some Exam Needs Two Forms of ID, **Digital ID are Not Acceptable.**
Preferred Female in Morning Session 9.00 AM to 4.00 PM.
This is a non-smoking environment No Smoking Area Inside The Building.
Parking is Available Outside the Building.
For Information +966112490080 Or +966112496327 Please Make Calls During Operation Hours.
Website www.esi.edu.sa

For further details, refer to the following resources:

- GIAC Proctored Exam Reference Guide which provides comprehensive information on exam policies, procedures, and requirements
- Exam Scheduling Guide which offers step-by-step instructions for selecting a test center, scheduling your exam, and understanding the registration process

Expiration of Access

Access to the training and exam voucher will expire after **four months**, starting from the purchase date.

Retaking a Failed Exam

You can purchase a retake within 30 days of your exam deadline through the GIAC Certification Portal. After this period, a new certification attempt must be purchased. Retakes are available only after a failed attempt, with a mandatory 30-day wait period. A waiver may reduce this to 14 days in emergencies. After three failed attempts, a one-year wait is required unless a waiver is approved.

Extensions

Certification attempts last 4 months (120 days). You can buy a 45-day extension up to 30 days after the deadline. Extensions cancel exam appointments if more than 24 hours away. A **$150** fee applies for cancellations within 24 hours. A maximum of 10 extensions is allowed per attempt.

Complimentary Extensions

GIAC grants free extensions only for bereavement, medical emergencies, military deployment, or government duty. For more details, please refer to the official GIAC Retakes and Extensions Policy.

Contact Information for GIAC Support

If you have any questions, feel free to email **info@giac.org**. To ensure quicker assistance, include your curriculum User ID or portal SD number, along with the certification you are attempting, so they can easily reference your account.

Exam Format

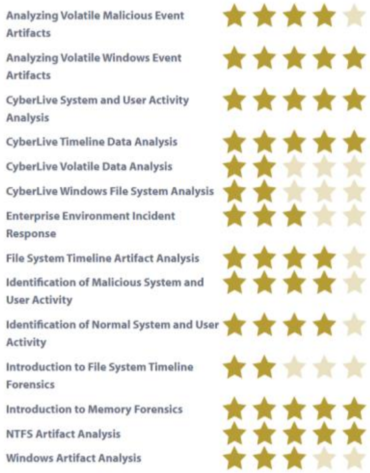
The GCFA certification exam is an **open-book** exam consisting of **82 multiple-choice questions** to be completed within **3 hours**. The minimum passing score required is 71%.

Note that scoring 85% or higher invites you to the Instructor Development Program, offering potential recruitment opportunities. Scoring 90% or higher earns "Honors" and an invitation to the GIAC Advisory Board, providing access to an exclusive email list and expert Q&A resources.

Most of the exam questions are not straightforward or direct; rather, they require a deep understanding of the content and the ability to apply key concepts effectively. It is recommended to spend no more than **3 minutes per question** to avoid time management issues. Exceeding this limit may result in time constraints as you progress. Additionally, SANS allows you to **skip up to 15 questions** and return to them later before final submission. You are allowed a total of **15 minutes of break time** during the GCFA exam. This break time can be used all at once or split across two sessions, depending on your preference. The GCFA exam incorporates CyberLive to meet the demand for discipline-specific certifications and practical skills validation in cybersecurity. CyberLive provides a hands-on, real-world testing environment where professionals demonstrate their expertise through the use of actual programs, code, and virtual machines. This approach evaluates candidates with practical, task-based questions that mimic specialized job roles, ensuring their knowledge and skills are directly applicable to real-world cybersecurity challenges. A GCFA Exam will contain **8 CyberLive questions.**

Tips for Preparing for the GCFA Exam

- Take your time to study the material thoroughly, use **sticky tabs** to organize each section of the book, and **highlight key points** to use as quick references during the exam.
- Read the course books carefully multiple times to ensure a solid understanding
    - **First reading**: Understand the content of the entire course.
    - **Second Reading**: Familiarize yourself with the structure of each section and create a mind map to visualize key concepts. I have shared my mind map at the beginning of each section in this book.
    - **Third reading**: Create an index organized by page numbers to help you locate information quickly. You can refer to the "Prepare Index" section for a step-by-step guide on creating your own index. Additionally, I have uploaded a sample index in the last section of the book for reference.
    - **Final reading**: Review to ensure you fully understand the material, can efficiently navigate the index, and that your index covers the most useful and frequently referenced topics.
- Since this is an advanced-level course, you will encounter new and unfamiliar concepts, particularly if you have not completed the GCFE. To enhance your understanding:
    - Search online using keywords on Google or YouTube. Many YouTube channels provide excellent explanations; I recommend **13Cubed** and **@SANSForensics**.
    - Use AI tools like ChatGPT, Copilot, or Gemini to clarify concepts. You can write down the main topic, such as "MFT Attributes" and **generate multiple-choice questions** for practice or upload pictures of your notes or book content. These tools are invaluable for deepening your understanding of key concepts and familiarizing yourself with exam-style questions, as the GCFA exam emphasizes conceptual understanding rather than simple fact recall.
- Do not skip the **course labs**, as they simulate real-world environments. Summarize important commands and notes from each lab, print them, and bring them to the exam. The final eight exam questions will be practical exercises involving virtual machines, as the GCFA is a cyberlive certification. Key labs include Volatility, MFTEcmd.exe, PEcmd and other execution tools of shim cash and amcash, Autorunsc, Timeline Explorer, EventID Explorer and Journals Labs.
- To further practice, explore **TryHackMe** rooms related to Volatility and Windows Forensics, which cover many important GCFA topics.
- Familiarize yourself with your index and practice using it to locate information quickly.
- Take the **practice test** seriously to familiarize yourself with the format and main topics covered in the real exam. Ensure that you complete them after preparing your index and you can take a screenshot of your exam to review it later. You will receive the assessment below after completing the practice test. It will evaluate your skills and highlight the domains that require improvement



Analyzing Volatile Malicious Event Artifacts	★★★★☆
Analyzing Volatile Windows Event Artifacts	★★★★★
CyberLive System and User Activity Analysis	★★★★★
CyberLive Timeline Data Analysis	★★★★★
CyberLive Volatile Data Analysis	★★☆☆☆
CyberLive Windows File System Analysis	★★☆☆☆
Enterprise Environment Incident Response	★★★☆☆
File System Timeline Artifact Analysis	★★★★☆
Identification of Malicious System and User Activity	★★★★☆
Identification of Normal System and User Activity	★★★★☆
Introduction to File System Timeline Forensics	★★☆☆☆
Introduction to Memory Forensics	★★★★★
NTFS Artifact Analysis	★★★★★
Windows Artifact Analysis	★★★☆☆

- There are wonderful posters and **cheat sheets** created by SANS that provide a summarized collection of the most valuable information, making them excellent resources for quick reference and review. For more details, explore the available resources at SANS Posters and Cheat Sheets
- Before your exam, ensure you get a good night's sleep to stay focused and alert. On the day of the exam, eat a nutritious meal to maintain your energy and concentration levels.
- During the exam, bring:
  - Your Books and index.
  - Two forms of ID (Identification card and International Travel Passport or Driver’s license)
  - A printed copy of your lab notes.
  - Hard copies of key references, such as MFT structures, Event IDs, and SANS posters (e.g., Evil Hunt, Network Forensics). These materials will help speed up your responses.
- Be thorough when answering exam questions—read each question and all answer choices carefully before selecting.

Preparing Index

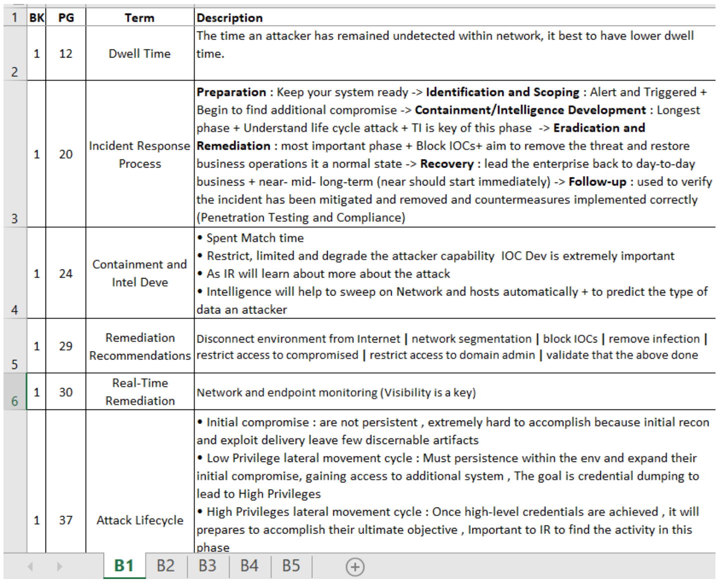
A well-organized index is essential for enhancing accessibility during your exam. Below are three effective approaches to indexing. For additional insights, I highly recommend reading the following articles:

- GIAC Testing Tips and Tricks
- Tips for GIAC Certifications

GCFA Index Samples

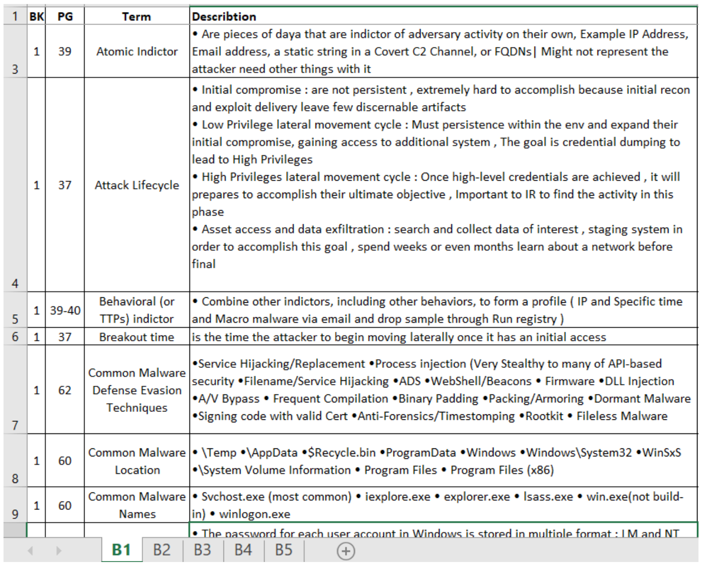
1. Excel-Based Approach (Preferred)

This method involves creating an index in Excel with five sheets—one for each book. Each sheet contains four labeled columns: Book Number (BK), Page Number (PG), Term (Keyword), and Description. Once completed, sort each sheet alphabetically by the Term column to enhance searchability then print each sheet separately. This approach ensures that the index is well-organized and easily accessible for quick reference.

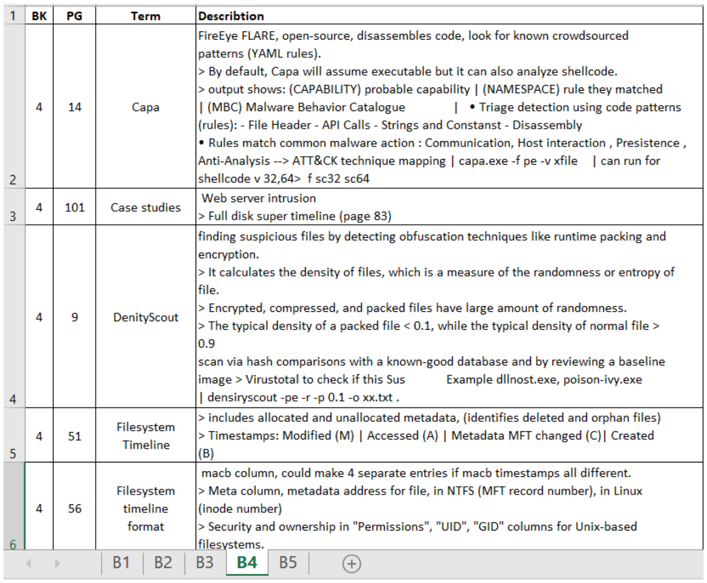


1	BK	PG	Term	Description
2	1	12	Dwell Time	The time an attacker has remained undetected within network, it best to have lower dwell time.
3	1	20	Incident Response Process	**Preparation** : Keep your system ready -> **Identification and Scoping** : Alert and Triggered + Begin to find additional compromise -> **Containment/Intelligence Development** : Longest phase + Understand life cycle attack + TI is key of this phase -> **Eradication and Remediation** : most important phase + Block IOCs+ aim to remove the threat and restore business operations it a normal state -> **Recovery** : lead the enterprise back to day-to-day business + near- mid- long-term (near should start immediately) -> **Follow-up** : used to verify the incident has been mitigated and removed and countermeasures implemented correctly (Penetration Testing and Compliance)
4	1	24	Containment and Intel Deve	• Spent Match time • Restrict, limited and degrade the attacker capability IOC Dev is extremely important • As IR will learn about more about the attack • Intelligence will help to sweep on Network and hosts automatically + to predict the type of data an attacker
5	1	29	Remediation Recommendations	Disconnect environment from Internet | network segmentation | block IOCs | remove infection | restrict access to compromised | restrict access to domain admin | validate that the above done
6	1	30	Real-Time Remediation	Network and endpoint monitoring (Visibility is a key)
	1	37	Attack Lifecycle	• Initial compromise : are not persistent , extremely hard to accomplish because initial recon and exploit delivery leave few discernable artifacts • Low Privilege lateral movement cycle : Must persistence within the env and expand their initial compromise, gaining access to additional system , The goal is credential dumping to lead to High Privileges • High Privileges lateral movement cycle : Once high-level credentials are achieved , it will prepares to accomplish their ultimate objective , Important to IR to find the activity in this phase

After ordering books 1 and 4 alphabetically by the Term column, they are now ready to print.



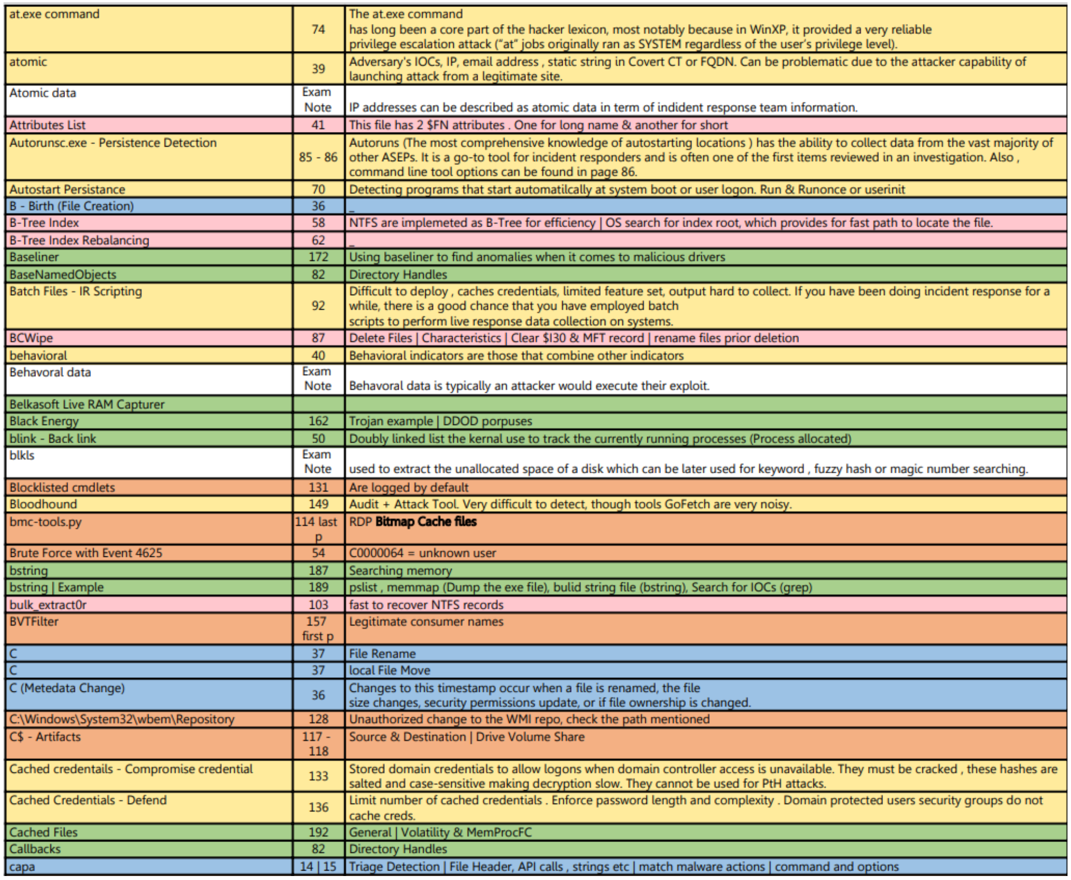
1	BK	PG	Term	Describtion
3	1	39	Atomic Indictor	• Are pieces of daya that are indictor of adversary activity on their own, Example IP Address, Email address, a static string in a Covert C2 Channel, or FQDNs| Might not represent the attacker need other things with it
4	1	37	Attack Lifecycle	• Initial compromise : are not persistent , extremely hard to accomplish because initial recon and exploit delivery leave few discernable artifacts • Low Privilege lateral movement cycle : Must persistence within the env and expand their initial compromise, gaining access to additional system , The goal is credential dumping to lead to High Privileges • High Privileges lateral movement cycle : Once high-level credentials are achieved , it will prepares to accomplish their ultimate objective , Important to IR to find the activity in this phase • Asset access and data exfiltration : search and collect data of interest , staging system in order to accomplish this goal , spend weeks or even months learn about a network before final
5	1	39-40	Behavioral (or TTPs) indictor	• Combine other indictors, including other behaviors, to form a profile (IP and Specific time and Macro malware via email and drop sample through Run registry)
6	1	37	Breakout time	is the time the attacker to begin moving laterally once it has an initial access
7	1	62	Common Malware Defense Evasion Techniques	•Service Hijacking/Replacement •Process injection (Very Stealthy to many of API-based security •Filename/Service Hijacking •ADS •WebShell/Beacons • Firmware •DLL Injection •A/V Bypass • Frequent Compilation •Binary Padding •Packing/Armoring •Dormant Malware •Signing code with valid Cert •Anti-Forensics/Timestomping •Rootkit • Fileless Malware
8	1	60	Common Malware Location	• \Temp •\AppData •$Recycle.bin •ProgramData •Windows •Windows\System32 •WinSxS •\System Volume Information • Program Files • Program Files (x86)
9	1	60	Common Malware Names	• Svchost.exe (most common) • iexplore.exe • explorer.exe • lsass.exe • win.exe(not build-in) • winlogon.exe
				• The password for each user account in Windows is stored in multiple format : LM and NT



1	BK	PG	Term	Describtion
2	4	14	Capa	FireEye FLARE, open-source, disassembles code, look for known crowdsourced patterns (YAML rules). > By default, Capa will assume executable but it can also analyze shellcode. > output shows: (CAPABILITY) probable capability | (NAMESPACE) rule they matched | (MBC) Malware Behavior Catalogue | • Triage detection using code patterns (rules): - File Header - API Calls - Strings and Constanst - Disassembly • Rules match common malware action : Communication, Host interaction , Presistence , Anti-Analysis --> ATT&CK technique mapping | capa.exe -f pe -v xfile | can run for shellcode v 32,64> f sc32 sc64
3	4	101	Case studies	Web server intrusion > Full disk super timeline (page 83)
4	4	9	DenityScout	finding suspicious files by detecting obfuscation techniques like runtime packing and encryption. > It calculates the density of files, which is a measure of the randomness or entropy of file. > Encrypted, compressed, and packed files have large amount of randomness. > The typical density of a packed file < 0.1, while the typical density of normal file > 0.9 scan via hash comparisons with a known-good database and by reviewing a baseline image > Virustotal to check if this Sus Example dllnost.exe, poison-ivy.exe | densiryscout -pe -r -p 0.1 -o xx.txt .
5	4	51	Filesystem Timeline	> includes allocated and unallocated metadata, (identifies deleted and orphan files) > Timestamps: Modified (M) | Accessed (A) | Metadata MFT changed (C)| Created (B)
6	4	56	Filesystem timeline format	macb column, could make 4 separate entries if macb timestamps all different. > Meta column, metadata address for file, in NTFS (MFT record number), in Linux (inode number) > Security and ownership in "Permissions", "UID", "GID" columns for Unix-based filesystems.

2. Single-Document Approach

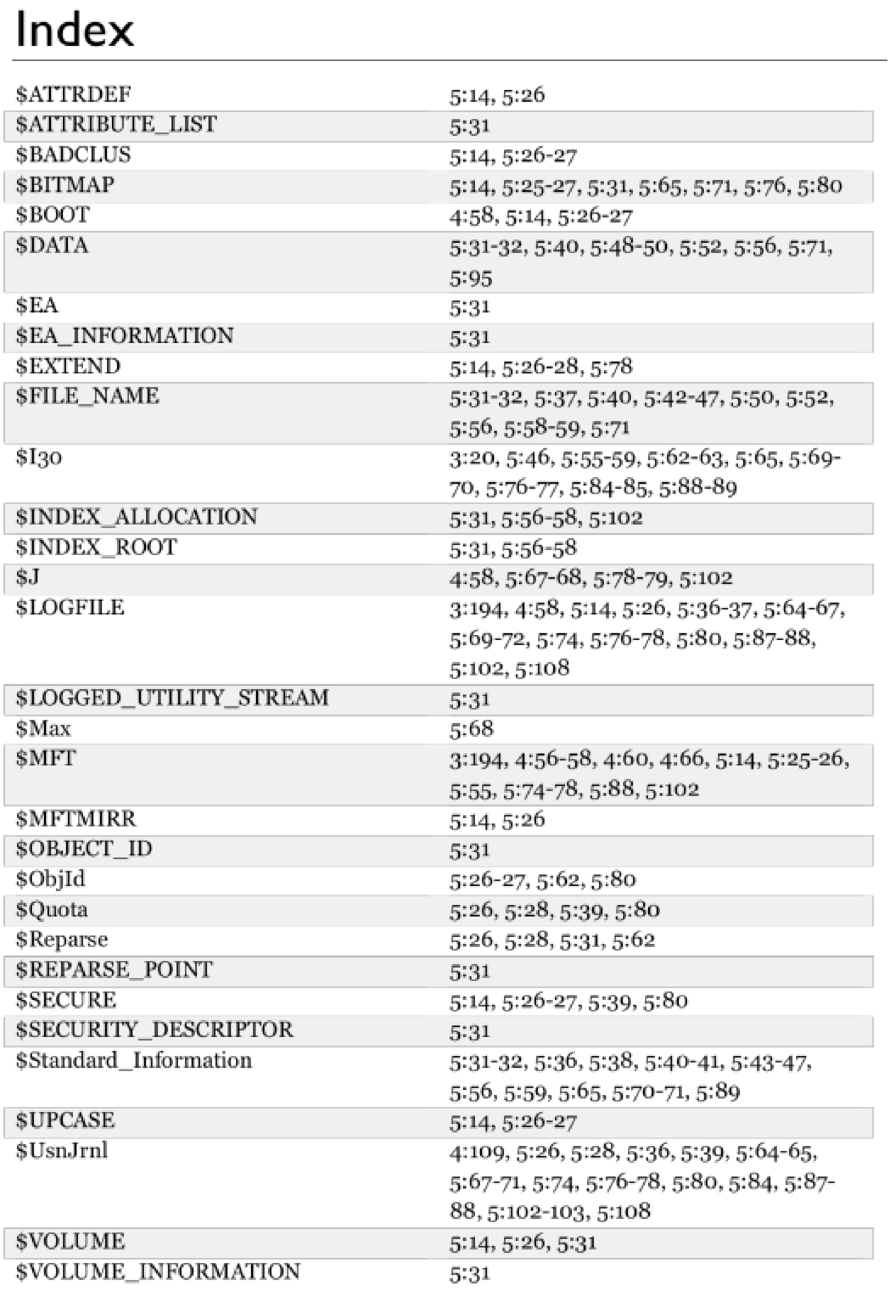
As shown in the image, this method involves creating a single document with three columns: Name, Page Number, and Description. To improve visual organization, each book is assigned a different color. This approach provides a consolidated index, making it easier to scan through information while maintaining clear differentiation between books.



at.exe command	74	The at.exe command has long been a core part of the hacker lexicon, most notably because in WinXP, it provided a very reliable privilege escalation attack ("at" jobs originally ran as SYSTEM regardless of the user's privilege level).
atomic	39	Adversary's IOCs, IP, email address , static string in Covert CT or FQDN. Can be problematic due to the attacker capability of launching attack from a legitimate site.
Atomic data	Exam Note	IP addresses can be described as atomic data in term of indident response team information.
Attributes List	41	This file has 2 $FN attributes . One for long name & another for short
Autorunsc.exe - Persistence Detection	85 - 86	Autoruns (The most comprehensive knowledge of autostarting locations) has the ability to collect data from the vast majority of other ASEPs. It is a go-to tool for incident responders and is often one of the first items reviewed in an investigation. Also , command line tool options can be found in page 86.
Autostart Persistance	70	Detecting programs that start automatilcally at system boot or user logon. Run & Runonce or userinit
B - Birth (File Creation)	36	_
B-Tree Index	58	NTFS are implemeted as B-Tree for efficiency | OS search for index root, which provides for fast path to locate the file.
B-Tree Index Rebalancing	62	_
Baseliner	172	Using baseliner to find anomalies when it comes to malicious drivers
BaseNamedObjects	82	Directory Handles
Batch Files - IR Scripting	92	Difficult to deploy , caches credentials, limited feature set, output hard to collect. If you have been doing incident response for a while, there is a good chance that you have employed batch scripts to perform live response data collection on systems.
BCWipe	87	Delete Files | Characteristics | Clear $I30 & MFT record | rename files prior deletion
behavioral	40	Behavioral indicators are those that combine other indicators
Behavoral data	Exam Note	Behavoral data is typically an attacker would execute their exploit.
Belkasoft Live RAM Capturer		
Black Energy	162	Trojan example | DDOD porpuses
blink - Back link	50	Doubly linked list the kernal use to track the currently running processes (Process allocated)
blkls	Exam Note	used to extract the unallocated space of a disk which can be later used for keyword , fuzzy hash or magic number searching.
Blocklisted cmdlets	131	Are logged by default
Bloodhound	149	Audit + Attack Tool. Very difficult to detect, though tools GoFetch are very noisy.
bmc-tools.py	114 last p	RDP **Bitmap Cache files**
Brute Force with Event 4625	54	C0000064 = unknown user
bstring	187	Searching memory
bstring | Example	189	pslist , memmap (Dump the exe file), bulid string file (bstring), Search for IOCs (grep)
bulk_extract0r	103	fast to recover NTFS records
BVTFilter	157 first p	Legitimate consumer names
C	37	File Rename
C	37	local File Move
C (Metedata Change)	36	Changes to this timestamp occur when a file is renamed, the file size changes, security permissions update, or if file ownership is changed.
C:\Windows\System32\wbem\Repository	128	Unauthorized change to the WMI repo, check the path mentioned
C$ - Artifacts	117 - 118	Source & Destination | Drive Volume Share
Cached credentails - Compromise credential	133	Stored domain credentials to allow logons when domain controller access is unavailable. They must be cracked , these hashes are salted and case-sensitive making decryption slow. They cannot be used for PtH attacks.
Cached Credentials - Defend	136	Limit number of cached credentials . Enforce password length and complexity . Domain protected users security groups do not cache creds.
Cached Files	192	General | Volatility & MemProcFC
Callbacks	82	Directory Handles
capa	14 | 15	Triage Detection | File Header, API calls , strings etc | match malware actions | command and options

3. SANS-Style Index (Book 5 Format)

This method follows the SANS INDEX format, typically found in the last section of Book 5. Terms are arranged alphabetically and tagged with the corresponding book and page number. If a term appears in multiple books and on different pages, all references are listed and separated by commas. This method is particularly effective for quick lookups, especially during an exam.

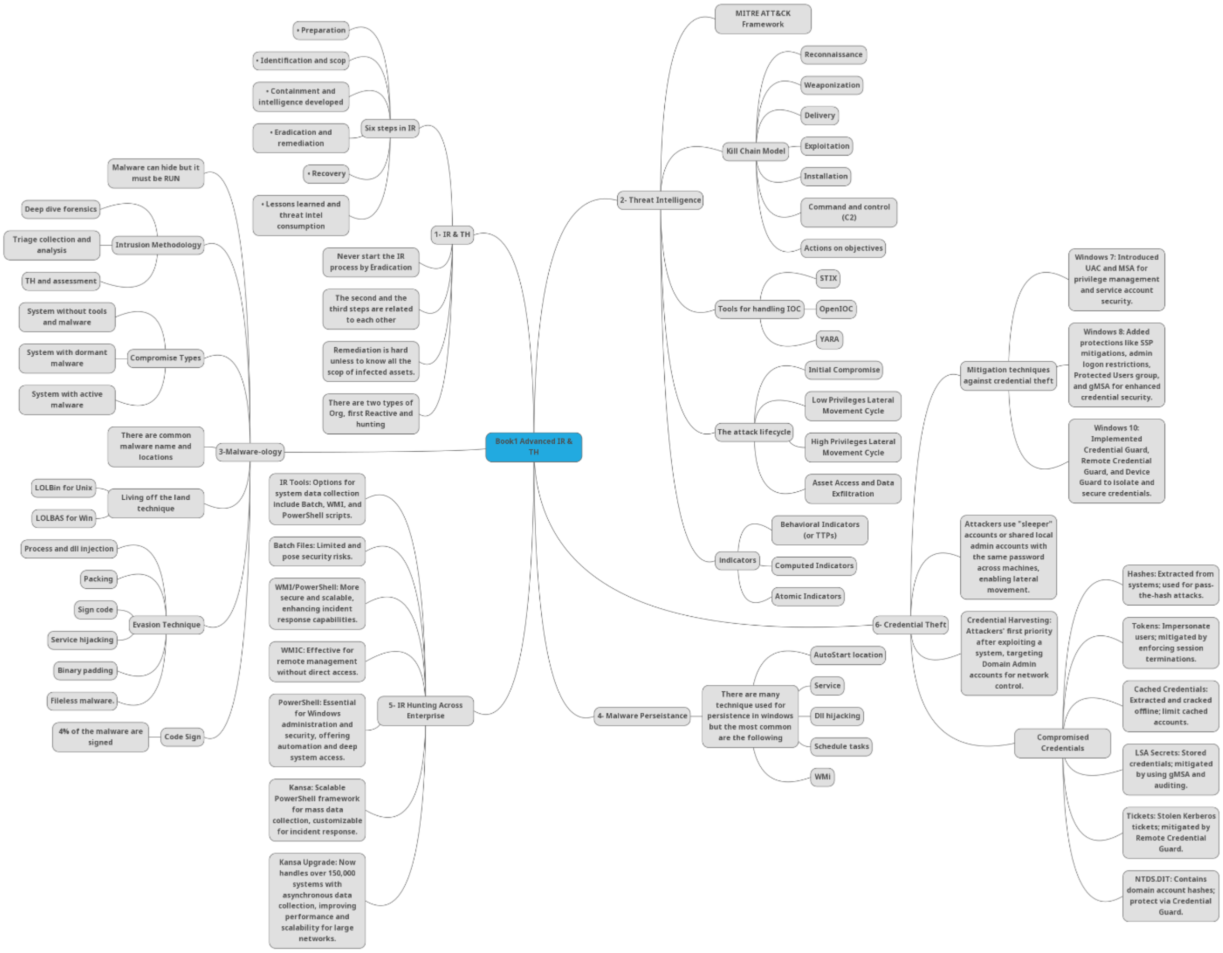


Index

$ATTRDEF	5:14, 5:26
$ATTRIBUTE_LIST	5:31
$BADCLUS	5:14, 5:26-27
$BITMAP	5:14, 5:25-27, 5:31, 5:65, 5:71, 5:76, 5:80
$BOOT	4:58, 5:14, 5:26-27
$DATA	5:31-32, 5:40, 5:48-50, 5:52, 5:56, 5:71, 5:95
$EA	5:31
$EA_INFORMATION	5:31
$EXTEND	5:14, 5:26-28, 5:78
$FILE_NAME	5:31-32, 5:37, 5:40, 5:42-47, 5:50, 5:52, 5:56, 5:58-59, 5:71
$I30	3:20, 5:46, 5:55-59, 5:62-63, 5:65, 5:69-70, 5:76-77, 5:84-85, 5:88-89
$INDEX_ALLOCATION	5:31, 5:56-58, 5:102
$INDEX_ROOT	5:31, 5:56-58
$J	4:58, 5:67-68, 5:78-79, 5:102
$LOGFILE	3:194, 4:58, 5:14, 5:26, 5:36-37, 5:64-67, 5:69-72, 5:74, 5:76-78, 5:80, 5:87-88, 5:102, 5:108
$LOGGED_UTILITY_STREAM	5:31
$Max	5:68
$MFT	3:194, 4:56-58, 4:60, 4:66, 5:14, 5:25-26, 5:55, 5:74-78, 5:88, 5:102
$MFTMIRR	5:14, 5:26
$OBJECT_ID	5:31
$ObjId	5:26-27, 5:62, 5:80
$Quota	5:26, 5:28, 5:39, 5:80
$Reparse	5:26, 5:28, 5:31, 5:62
$REPARSE_POINT	5:31
$SECURE	5:14, 5:26-27, 5:39, 5:80
$SECURITY_DESCRIPTOR	5:31
$Standard_Information	5:31-32, 5:36, 5:38, 5:40-41, 5:43-47, 5:56, 5:59, 5:65, 5:70-71, 5:89
$UPCASE	5:14, 5:26-27
$UsnJrnl	4:109, 5:26, 5:28, 5:36, 5:39, 5:64-65, 5:67-71, 5:74, 5:76-78, 5:80, 5:84, 5:87-88, 5:102-103, 5:108
$VOLUME	5:14, 5:26, 5:31
$VOLUME_INFORMATION	5:31

Advanced Incident Response and Threat Hunting

This section was designed to help organizations increase their capability to detect and respond to intrusion events. This is an achievable goal and begins by teaching the tools and techniques necessary to find evil in your network. This course is designed to make you and your organization an integral part of the solution. To keep pace, incident responders and threat hunters must be armed with the latest tools, analysis techniques, and enterprise methodologies to identify, track, and contain advanced adversaries with the ultimate goal of rapid remediation of incidents and damage mitigation. Further, incident response and threat hunting analysts must be able to scale their efforts across potentially thousands of systems in the enterprise. We start by examining the six-step incident response methodology as it applies to incident response for advanced threat groups. The importance of developing cyber threat intelligence to impact the adversaries' "kill chain" is discussed and forensic live response techniques and tactics are demonstrated that can be applied both to single systems and across the entire enterprise. Refer to



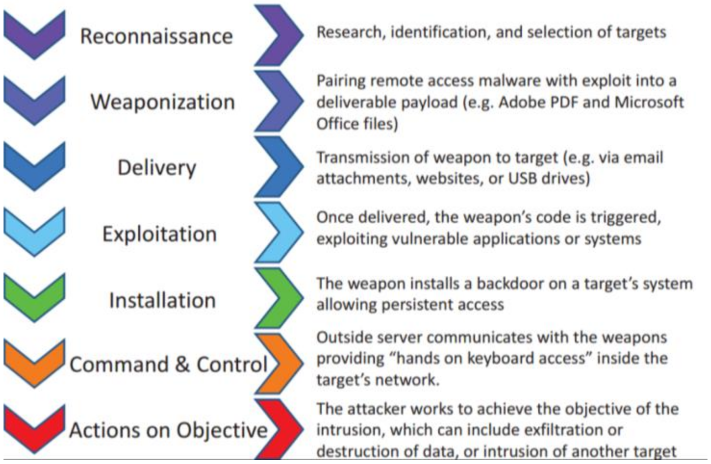
- Book1 Advanced IR & TH
  - 1- IR & TH
    - Six steps in IR
      - • Preparation
      - • Identification and scop
      - • Containment and intelligence developed
      - • Eradication and remediation
      - • Recovery
      - • Lessons learned and threat intel consumption
    - Never start the IR process by Eradication
    - The second and the third steps are related to each other
    - Remediation is hard unless to know all the scop of infected assets.
    - There are two types of Org, first Reactive and hunting
  - 2- Threat Intelligence
    - MITRE ATT&CK Framework
    - Kill Chain Model
      - Reconnaissance
      - Weaponization
      - Delivery
      - Exploitation
      - Installation
      - Command and control (C2)
      - Actions on objectives
    - Tools for handling IOC
      - STIX
      - OpenIOC
      - YARA
    - The attack lifecycle
      - Initial Compromise
      - Low Privileges Lateral Movement Cycle
      - High Privileges Lateral Movement Cycle
      - Asset Access and Data Exfiltration
    - Indicators
      - Behavioral Indicators (or TTPs)
      - Computed Indicators
      - Atomic Indicators
  - 3-Malware-ology
    - Malware can hide but it must be RUN
    - Intrusion Methodology
      - Deep dive forensics
      - Triage collection and analysis
      - TH and assessment
    - Compromise Types
      - System without tools and malware
      - System with dormant malware
      - System with active malware
    - There are common malware name and locations
    - Living off the land technique
      - LOLBin for Unix
      - LOLBAS for Win
    - Evasion Technique
      - Process and dll injection
      - Packing
      - Sign code
      - Service hijacking
      - Binary padding
      - Fileless malware.
    - Code Sign
      - 4% of the malware are signed
  - 4- Malware Perseistance
    - There are many technique used for persistence in windows but the most common are the following
      - AutoStart location
      - Service
      - Dll hijacking
      - Schedule tasks
      - WMi
  - 5- IR Hunting Across Enterprise
    - IR Tools: Options for system data collection include Batch, WMI, and PowerShell scripts.
    - Batch Files: Limited and pose security risks.
    - WMI/PowerShell: More secure and scalable, enhancing incident response capabilities.
    - WMIC: Effective for remote management without direct access.
    - PowerShell: Essential for Windows administration and security, offering automation and deep system access.
    - Kansa: Scalable PowerShell framework for mass data collection, customizable for incident response.
    - Kansa Upgrade: Now handles over 150,000 systems with asynchronous data collection, improving performance and scalability for large networks.
  - 6- Credential Theft
    - Mitigation techniques against credential theft
      - Windows 7: Introduced UAC and MSA for privilege management and service account security.
      - Windows 8: Added protections like SSP mitigations, admin logon restrictions, Protected Users group, and gMSA for enhanced credential security.
      - Windows 10: Implemented Credential Guard, Remote Credential Guard, and Device Guard to isolate and secure credentials.
    - Attackers use "sleeper" accounts or shared local admin accounts with the same password across machines, enabling lateral movement.
    - Credential Harvesting: Attackers' first priority after exploiting a system, targeting Domain Admin accounts for network control.
    - Compromised Credentials
      - Hashes: Extracted from systems; used for pass-the-hash attacks.
      - Tokens: Impersonate users; mitigated by enforcing session terminations.
      - Cached Credentials: Extracted and cracked offline; limit cached accounts.
      - LSA Secrets: Stored credentials; mitigated by using gMSA and auditing.
      - Tickets: Stolen Kerberos tickets; mitigated by Remote Credential Guard.
      - NTDS.DIT: Contains domain account hashes; protect via Credential Guard.

1.1 Incident Response & Threat Hunting

- Six Steps in Incident Response (IR):
    - Preparation: Establish a robust methodology and ensure all necessary tools, resources, and plans are in place to handle incidents effectively.
    - Identification and Scope: Critical steps where the incident is identified, and affected systems are verified. Ensure accurate detection and confirmation of compromised systems.
    - Containment and Intelligence Development: Stop the attack to prevent further spread. This step can be the most time-consuming in the IR cycle. Gather intelligence on indicators of compromise (IOCs), tactics, techniques, and procedures (TTPs). Understand how the attacker gained initial access, persistence mechanisms, command and control (C2), and lateral movement.
    - Eradication and Remediation: Remove all malicious artifacts by blocking IOCs. Modify settings and reset passwords to prevent further access.
    - Recovery: Ensure business operations resume with improved security measures. Implement new goals and changes to reduce the risk of recurrence.
    - Lessons Learned and Threat Intelligence Utilization: Review the incident to update the IR playbook and avoid similar mistakes in the future.
- Key Considerations in Incident Response:
    - Avoid starting the IR process with eradication or shutting down systems prematurely, as this can hinder effective threat elimination and resolution.
    - Dynamic Relationship Between Steps 2 and 3: Identification and containment are interconnected. Every newly discovered infected device during containment requires updating the identification step to document all affected assets accurately.
    - Challenges in Remediation: Successful remediation requires a complete understanding of the infection's scope. The process involves assessing the posture, executing remediation actions, and implementing new security controls.
    - Proactive vs. Reactive Organizations:
        - Reactive Organizations: Respond after receiving alerts.
        - Proactive Organizations (Threat Hunting): Actively search for suspicious activity before alerts are triggered.
- The Role of Human-Led Threat Hunting:
    - Human analysts play a crucial role in detecting sophisticated threats that automated tools may overlook. Effective threat detection and response require well-structured teams, a sustainable operational tempo, and a mindset focused on continuous improvement. Cyber threat intelligence and a proactive attitude are vital for maintaining robust security defenses.

1.2 Threat Intelligence

- The attack lifecycle, focusing on how threat intelligence maps attacker techniques, tactics, and procedures (TTPs) throughout different stages of an attack with main four key phases:
    - Initial Compromise: The adversary gains initial access, often fragile and easily disreputable by a response team.
    - Low Privileges Lateral Movement Cycle: The attacker maintains persistence, escalates privileges, and moves laterally within the network, often establishing backdoors.
    - High Privileges Lateral Movement Cycle: With high-level credentials, the attacker further entrenches, preparing for major objectives like data exfiltration. This phase is crucial for defenders to detect and disrupt before significant damage occurs.
    - Asset Access and Data Exfiltration: Adversaries search for and collect valuable data, leaving detectable footprints. Exfiltrating this data is challenging due to network monitoring, making this phase critical for defenders to intercept before attackers can successfully remove or destroy sensitive information.
- The Kill Chain is a framework in threat intelligence that categorizes the sequence of actions in cyberattacks, helping in organizing detection indicators
    - Reconnaissance: Research, identification, and selection of targets.
    - Weaponization: Determining the best method of exploitation.
    - Delivery: Sending the exploit capability to the remote system.
    - Exploitation: Gaining initial access through the vulnerability provided by the exploit.
    - Installation: Installing software or creating persistent access on the target system.
    - Command and Control: Establishing remote control over the target system.
    - Actions on Objective: Achieving the attacker's ultimate goals, such as data exfiltration, disruption, or destruction.



Reconnaissance
Research, identification, and selection of targets

Weaponization
Pairing remote access malware with exploit into a deliverable payload (e.g. Adobe PDF and Microsoft Office files)

Delivery
Transmission of weapon to target (e.g. via email attachments, websites, or USB drives)

Exploitation
Once delivered, the weapon's code is triggered, exploiting vulnerable applications or systems

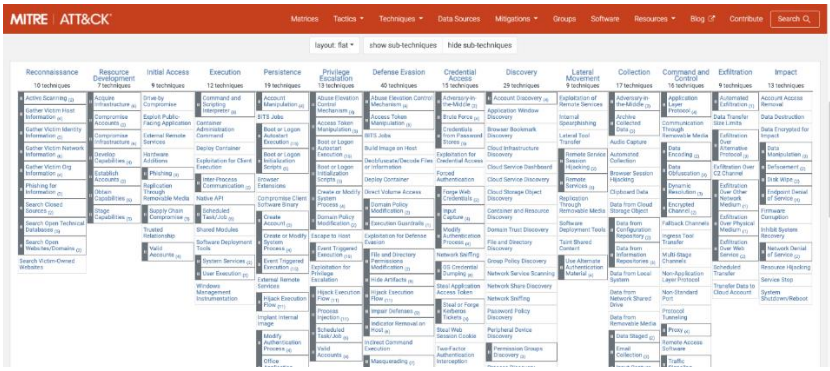
Installation
The weapon installs a backdoor on a target's system allowing persistent access

Command & Control
Outside server communicates with the weapons providing "hands on keyboard access" inside the target's network.

Actions on Objective
The attacker works to achieve the objective of the intrusion, which can include exfiltration or destruction of data, or intrusion of another target

- The Cyber Kill Chain through security intelligence by identifying and understanding the indicators that can reveal adversary activity, categorizes indicators into three types:
    - Atomic Indicators: Basic pieces of data, such as IP addresses or domain names, that may indicate adversary activity. These indicators can be problematic because they may not always reliably signal malicious intent.
    - Computed Indicators: More complex indicators derived from the analysis of multiple data points, often providing a stronger signal of malicious activity.
    - Behavioral Indicators (or TTPs): Patterns of behavior that are characteristic of specific adversaries or types of attacks, which can help in identifying and responding to threats more effectively.
- Effective use of security intelligence and consistent attack attributes in the Cyber Kill Chain enables defenders to detect, analyze, and disrupt advanced persistent threats at various stages of an attack.
- MITRE ATT&CK Framework: the challenges of threat hunting without effective threat intelligence and emphasizes the importance of actionable intelligence in defending against advanced attacks. It introduces the MITRE ATT&CK framework, which provides a comprehensive model for understanding and describing adversary behavior, particularly focusing on post-compromise tactics. The framework is highly valued for its detailed, platform-agnostic coverage of adversarial tactics and techniques, making it a critical tool for organizations aiming to improve their defensive strategies. The ATT&CK framework helps defenders prioritize network defenses by providing insight into adversaries' methods and supports continuous improvement in threat detection and response efforts.
- The MITRE ATT&CK framework categorizes 14 tactics used in the later stages of a cyber attack, providing a detailed guide for understanding and responding to adversarial actions. Each tactic, like "Persistence," includes various techniques that attackers use, along with technical details, detection methods, and possible defenses. This framework helps security teams focus on common threats and identify security gaps, aiming for efficient and rapid detection of intrusion For more information, visit:

    - MITRE ATT&CK Framework
    - MITRE ATT&CK Navigator



Reconnaissance 10 techniques	Resource Development 7 techniques	Initial Access 9 techniques	Execution 12 techniques	Persistence 19 techniques	Privilege Escalation 13 techniques	Defense Evasion 40 techniques	Credential Access 15 techniques	Discovery 29 techniques	Lateral Movement 9 techniques	Collection 17 techniques	Command and Control 16 techniques	Exfiltration 9 techniques	Impact 13 techniques
Active Scanning (3)	Acquire Infrastructure (6)	Drive-by Compromise	Command and Scripting Interpreter (8)	Account Manipulation (4)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)	Adversary-in-the-Middle (2)	Account Discovery (4)	Exploitation of Remote Services	Adversary-in-the-Middle (3)	Application Layer Protocol (4)	Automated Exfiltration (1)	Account Access Removal
Gather Victim Host Information (4)	Compromise Accounts (2)	Exploit Public-Facing Application	Container Administration Command	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Brute Force (4)	Application Window Discovery	Internal Spearphishing	Archive Collected Data (3)	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information (3)	Compromise Infrastructure (6)	External Remote Services	Deploy Container	Boot or Logon Autostart Execution (14)	Boot or Logon Autostart Execution (14)	BITS Jobs	Credentials from Password Stores (5)	Browser Bookmark Discovery	Lateral Tool Transfer	Audio Capture	Data Encoding (2)	Exfiltration Over Alternative Protocol (3)	Data Encrypted for Impact
Gather Victim Network Information (6)	Develop Capabilities (4)	Hardware Additions	Exploitation for Client Execution	Boot or Logon Initialization Scripts (5)	Boot or Logon Initialization Scripts (5)	Build Image on Host	Exploitation for Credential Access	Cloud Infrastructure Discovery	Remote Service Session Hijacking (2)	Automated Collection	Data Obfuscation (3)	Exfiltration Over C2 Channel	Data Manipulation (3)
Gather Victim Org Information (4)	Establish Accounts (2)	Phishing (3)	Inter-Process Communication (2)	Browser Extensions	Create or Modify System Process (4)	Deobfuscate/Decode Files or Information	Forced Authentication	Cloud Service Dashboard	Remote Services (6)	Browser Session Hijacking	Dynamic Resolution (3)	Exfiltration Over Other Network Medium (1)	Defacement (2)
Phishing for Information (3)	Obtain Capabilities (6)	Replication Through Removable Media	Native API	Compromise Client Software Binary	Domain Policy Modification (2)	Deploy Container	Forge Web Credentials (2)	Cloud Service Discovery	Replication Through Removable Media	Clipboard Data	Encrypted Channel (2)	Exfiltration Over Physical Medium (1)	Disk Wipe (2)
Search Closed Sources (2)	Stage Capabilities (5)	Supply Chain Compromise (3)	Scheduled Task/Job (6)	Create Account (3)	Escape to Host	Direct Volume Access	Input Capture (4)	Cloud Storage Object Discovery	Software Deployment Tools	Data from Cloud Storage Object	Fallback Channels	Exfiltration Over Web Service (2)	Endpoint Denial of Service (4)
Search Open Technical Databases (5)		Trusted Relationship	Shared Modules	Create or Modify System Process (4)	Event Triggered Execution (15)	Domain Policy Modification (2)	Modify Authentication Process (4)	Container and Resource Discovery	Taint Shared Content	Data from Configuration Repository (2)	Ingress Tool Transfer	Scheduled Transfer	Firmware Corruption
Search Open Websites/Domains (2)		Valid Accounts (4)	Software Deployment Tools	Event Triggered Execution (15)	Exploitation for Privilege Escalation	Execution Guardrails (1)	Network Sniffing	Domain Trust Discovery	Use Alternate Authentication Material (4)	Data from Information Repositories (3)	Multi-Stage Channels	Transfer Data to Cloud Account	Inhibit System Recovery
Search Victim-Owned Websites			System Services (2)	External Remote Services	Hijack Execution Flow (11)	Exploitation for Defense Evasion	OS Credential Dumping (8)	File and Directory Discovery		Data from Local System	Non-Application Layer Protocol		Network Denial of Service (2)
			User Execution (3)	Hijack Execution Flow (11)	Process Injection (11)	File and Directory Permissions Modification (2)	Steal Application Access Token	Group Policy Discovery		Data from Network Shared Drive	Non-Standard Port		Resource Hijacking
			Windows Management Instrumentation	Implant Internal Image	Scheduled Task/Job (6)	Hide Artifacts (9)	Steal or Forge Kerberos Tickets (4)	Network Service Scanning		Data from Removable Media	Protocol Tunneling		Service Stop
				Modify Authentication Process (4)	Valid Accounts (4)	Hijack Execution Flow (11)	Steal Web Session Cookie	Network Share Discovery		Data Staged (2)	Proxy (4)		System Shutdown/Reboot
				Office		Impair Defenses (9)	Two-Factor Authentication Interception	Network Sniffing		Email Collection (3)	Remote Access Software		
						Indicator Removal on Host (6)		Password Policy Discovery		Input Capture	Traffic Signaling		
						Indirect Command Execution		Peripheral Device Discovery					
						Masquerading (7)		Permission Groups Discovery (3)					
								Process Discovery					

- Indicators of Compromise (IOCs) are essential tools in cybersecurity, using a formal language to describe and detect attacker tactics through clear, standardized terms that facilitate information sharing. They help identify compromised systems by matching specific host-based or network-based characteristics to known threats, allowing security teams to efficiently narrow down and respond to potential threats. The effectiveness of IOCs varies, and they work best when customized to the specific environment they protect, driving the incident response process with precise and relevant threat indicators.
- Three key tools for handling Indicators of Compromise (IOCs):
  - STIX (Structured Threat Information eXpression): A community-driven effort to create a standardized, expressive, and flexible language for representing structured cyber threat information. It aims to be both machine-readable and human-readable, supporting broad participation and collaboration.
  - YARA: A tool designed primarily to help malware researchers identify and classify malware samples by defining rules based on textual or binary patterns. Each rule consists of strings and Boolean expressions to determine the logic.
  - OpenIOC: Originally developed by MANDIANT for collecting and sharing security threat information. It has been standardized and some tools are available for its use, but its popularity has declined as newer standards like STIX have emerged.
- CRITS and MISP as popular platforms for managing IOCs, both of which support the above formats.

1.3 Malware-ology

- Malware can hide but it must be RUN.
- Intrusion Methodology:
  - Threat Hunting (TH) and Assessment: Could be automated and applied to the whole environment using a SIEM system.
  - Triage Collection and Analysis: Begins either with alerts or hypotheses to collect data and analyze it.
  - Deep Dive Forensics: Answers all questions across the attack lifecycle.
- Compromise Types:
  - System with Active Malware: Easy to detect since it generates a large number of artifacts.
  - System with Dormant Malware: Hard to detect; typically runs once for credential dumping or as a scheduled task.
  - System without Tools and Malware: The hardest to detect; attackers use built-in utilities (living-off-the-land techniques).
- Context in Malware Investigation

Common malware names and locations should be reviewed carefully. Example: If svchost.exe runs from the \temp folder or \$Recycle.Bin, it raises suspicion and should be investigated.

- Living Off the Land (LOTL) Technique

Advanced attackers exploit legitimate binaries (LOLBins) such as rundll32.exe to execute commands in memory, reducing detection chances. There are two detection projects:

  - LOLBAS for Windows
  - LOLBin for Unix
- Malware Evasion Techniques
  - Process Injection: Injects code into legitimate processes to bypass security tools.
  - Packing: Compresses executables to complicate analysis.
  - Code Signing: Uses valid digital certificates to appear trustworthy, though signed code can still be malicious if the certificate is revoked. Code signing verifies the authenticity and integrity of software using digital certificates. Initially, physical media was used for software distribution. Online distribution increased the need for signature verification. Though rare, only 4% of malware is signed, but it is highly dangerous. Signed malware is increasing, so both signed and unsigned code must be examined. Prioritize inspecting unsigned code.
- Threat hunting involves adapting continuously to detect sophisticated evasion techniques and uncovering subtle malicious artifacts.

1.4 Malware Persistence

There are numerous techniques—up to 50—that attackers use to maintain persistence in Windows environments. Among these, some of the most common methods are:

- AutoStart Locations

AutoStart locations provide an excellent starting point for investigating malicious activity by examining programs set to run automatically through the registry or file system.

  - Microsoft provides AutoStart Extension Points (ASEPs), which are legitimate mechanisms allowing tasks to execute automatically. Unfortunately, attackers frequently exploit these points to gain persistence.
  - Several forensic tools assist with investigating these locations:
    - Registry Explorer – for in-depth registry analysis.
    - RegRipper – a tool to parse and report registry data quickly.
    - Autoruns – a utility from Sysinternals for viewing programs configured to run at startup.
    - Kansa – a modular framework useful for PowerShell-based data collection during investigations.

  - The following registry keys are frequently used for persistence by attackers:
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
  - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
  - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
- Service-Based Persistence

Attackers often manipulate services to maintain persistence in a compromised Windows system. Common tactics include:

- New Service Creation
  Attackers may create a new service or modify an existing one to ensure it starts automatically during system boot, providing continuous access to the system.
- Service Replacement
  An existing service may be modified to load a different binary. In this case, a legitimate service executable is replaced with a malicious one, allowing attackers to run unauthorized code with elevated privileges.
- Service Failure Recovery
  Windows services can be configured to trigger recovery actions upon failure. Attackers exploit this feature by setting the recovery option to launch malicious code whenever the service crashes.
- Tools for Investigating and Analyzing Services
  Several tools can assist with detecting and responding to malicious service manipulation:
  - Sysinternals Autoruns – Identifies services configured to start automatically during system boot.
  - Command-Line Tools
    These tools are essential for analyzing Windows service-related persistence mechanisms and responding effectively to potential security incidents
    - sc (Service Control) command for managing services.
    - queryex and qc for examining live system service configurations.
    - PowerShell Script: Get-SvcFail.ps1 – A script that helps identify unusual service failure recovery settings, which may indicate malicious activity.
- Scheduled Tasks-Based Persistence

Attackers frequently leverage scheduled tasks to achieve persistence and sometimes even privilege escalation in Windows systems. Common tactics include:

  - at.exe (Windows XP and Windows 7)
    The at.exe command-line tool was commonly used in earlier Windows versions (Windows XP and Windows 7) for scheduling tasks. Attackers exploited at.exe to

execute commands with elevated privileges because it did not require administrative rights for task creation in some configurations.

  - schtasks.exe (Modern Replacement for at.exe)
    schtasks.exe is an updated command used in newer Windows versions for task scheduling. It is often utilized for persistence, as it allows the creation of tasks that run automatically at specified intervals or events. schtasks.exe can be executed remotely, making it a potential vector for malware propagation and script execution across a network.
  - Tool for Investigating Scheduled Tasks
    Sysinternals Autoruns – A powerful utility that helps identify scheduled tasks configured to start automatically. This tool is invaluable in detecting unauthorized or suspicious scheduled tasks that may indicate persistence mechanisms.

- DLL Hijacking-Based Persistence

DLL hijacking is a common technique attacker use to gain persistence by exploiting the way Windows applications load dynamic-link libraries (DLLs). Various forms of DLL hijacking include:

  - DLL Search Order Hijacking
    Windows uses a default DLL search order when loading libraries, which is influenced by the SafeDllSearchMode setting (enabled by default). An attacker places a malicious DLL in a location earlier in the search order, causing the system to load it instead of the legitimate DLL. The search process follows this sequence:
    - DLLs already loaded in memory
    - Side-by-Side Components
    - KnownDLLs list
    - Directory from which application is loaded
    - C:\Windows\System32
    - C:\Windows\system
    - C:\Windows
    - Current Directory
    - System %PATH%
  - Phantom DLL Hijacking
    Attackers create a malicious DLL using the name of an old, obsolete DLL no longer used by modern systems (e.g., fxsst.dll). When an application attempts to load the non-existent DLL, the malicious version is executed instead.
  - DLL Side-Loading
    The Side-by-Side (SxS) loading mechanism allows Windows to manage multiple versions of DLLs. An attacker uses a legitimate executable to load a malicious SxS DLL, enabling persistence or code execution under the guise of a trusted application.
  - Relative Path DLL Hijacking

    In this technique, attackers move a legitimate program to a different directory and create a custom DLL in the new location. When the application is executed, it loads the attacker's malicious DLL instead of the legitimate one.

    - To detect and respond to potential DLL hijacking attempts: Monitor unusual DLL loading paths using tools like Autoruns and Process Monitor. Analyze binary dependencies to identify potential side-loading vectors with tools like Dependency Walker. Review system logs for unexpected DLL load operations.

- WMI Event Consumer-Based Persistence

Attackers can leverage Windows Management Instrumentation (WMI) to achieve persistence by creating event filters, event consumers, and bindings between them. This method gained significant attention due to its use in advanced attacks like Stuxnet.

  - Key Components
    - Event Filter: Defines the event that triggers an action.
    - Event Consumer: Specifies the action (e.g., running a script or executable) to be performed when the event occurs.
    - Filter-to-Consumer Binding: Connects the event filter with the event consumer in the WMI repository.
  - MOF used to registry new class into WMI which stand for managed object format and it compiled automatically.
    - Get-WMIObject -Namespace root\Subscription -Class __EventFilter
    - Get-WMIObject -Namespace root\default -Class __EventFilter
    - Get-WMIObject -Namespace root\Subscription -Class __EventCONSUMER
    - Get-WMIObject -Namespace root\default -Class __EventCONSUMER
    - Get-WMIObject -Namespace root\Subscription -Class __filtertoCONSUMERBINDING
    - Get-WMIObject -Namespace root\default -Class __filtertoCONSUMERBINDING

1.5 IR: Hunting Across Enterprise

- Live Response and Data Collection Options

For performing live response across multiple systems, three scripting methods are commonly available: Batch, WMI, and PowerShell.

  - Batch Files:

  Batch scripts have traditionally been used for live response data collection, but they pose several limitations:
    - Deployment complexity
    - Limited functionality

        - Risk of caching credentials on remote systems, which can compromise security
    - Windows Management Instrumentation (WMI):

      WMI provides a scalable, secure alternative for detailed system reporting without caching credentials. It supports output in multiple formats, including XML, HTML, and CSV.

    - PowerShell:
      PowerShell enhances WMI by integrating scripting capabilities, post-processing options, and the ability to run arbitrary binaries. Together, WMI and PowerShell offer an efficient solution for modern incident response in Windows environments.

- WMIC (Windows Management Instrumentation Command Line Interface)

WMIC is a robust tool for remote system management and incident response. It allows gathering extensive system data, including:

        - Auto-start processes
        - Running processes
        - Network configurations
    - Key Advantages:
        - Uses native authentication, minimizing credential exposure
        - Ideal for identifying suspicious executables running from non-standard paths
    - PoSh-R2 Project:
      This project automates WMI data collection using PowerShell, making the process faster and more efficient.
- PowerShell for Incident Response

PowerShell is integral to Windows administration (from Windows 7 to Windows 10 and Windows Server 2008+). It integrates WMI, .NET, and COM, providing powerful features for remote analysis and local logging.

    - Features:
        - Consistent verb-noun naming conventions (e.g., Get-Process, Get-Service)
        - Output as objects that can be piped to other cmdlets for post-processing
        - Seamless access to filesystem and registry
        - Support for running commands interactively and in the background
- Kansa Framework for Large-Scale Incident Response

Kansa, developed by Dave Hull, is a scalable PowerShell framework for incident response, enabling data collection from thousands of systems using PowerShell Remoting.

    - Key Features:

        - Modular design allows execution of custom tasks beyond standard PowerShell cmdlets
        - Parallel data collection with centralized logging of errors
        - Saves output into organized directories
        - Integration with tools like Logparser.exe for efficient log analysis
    - Usage Scenarios:
        - Incident response
        - Threat hunting
        - Building baselines for enterprise environments
- Kansa Analysis Scripts

  Kansa's analysis scripts use stacking techniques to highlight uncommon occurrences, which can be indicative of malicious activity.

    - Examples:
        - Get-LogparserStack.ps1: Detects anomalies through stacking
        - Meta Scripts: Analyze deviations in file sizes to find unusual patterns
    - With the -analysis flag, scripts in Analysis.conf are automatically executed. These scripts are highly customizable, allowing responders to adapt to specific environments and efficiently detect rare and suspicious behavior
- Major Upgrade with DistributedKansa.ps1

A significant upgrade led by John Ketchum and the USAA threat hunting team transformed Kansa into a highly scalable framework for enterprise-level security investigations, expanding its capabilities to handle large-scale data collection and analysis. Key enhancements include:

- Scalable data collection from over 150,000 systems
- Fire & Forget modules for asynchronous data transmission to ELK
- Performance optimizations with kill switches, VDI/CPU throttling, and alert suppression

1.6 Credential Theft

Credential theft remains a critical tactic for attackers, enabling lateral movement and privilege escalation within networks. Below are key aspects of credential theft, associated vulnerabilities, and mitigation strategies.

- Credential Harvesting

  Attackers target credentials for high-privilege accounts like Domain Admins after gaining initial access. Shared local admin accounts with identical passwords across multiple systems are common vulnerabilities, simplifying lateral movement.

    - Detection Strategies:

    - Monitor event logs for privileged account usage, new account creation, and abnormal logins (e.g., after-hours or inter-workstation logins).
    - Use alerts for failed login attempts and unusual access patterns.
  - Mitigations:
    - Avoid shared local admin uses the same password across multiple devices.
    - Disable or limit the use of built-in administrator accounts (RID 500).
    - Restrict remote access and use tools like ELK for log correlation.
- Windows Security Enhancements to Mitigate Credential Theft
  Microsoft has evolved its defense mechanisms against credential theft across Windows versions:
  - Windows 7: Introduced User Account Control (UAC) and Managed Service Accounts (MSA).
  - Windows 8: Added restrictions on local admin remote logons and SSP plaintext password mitigations.
  - Windows 10: Improved credential protection with Credential Guard and Remote Credential Guard.

  1. Password Hash Extraction
     Attackers extract password hashes (LM and NT hashes) from the LSASS process using tools like Mimikatz and fgdump. These hashes allow pass-the-hash attacks, bypassing the need for cleartext passwords.
     - Mitigations:
       - Use Credential Guard and upgrade to Windows 10 for enhanced security.
       - Disable outdated protocols like WDigest and prevent high-privilege account use in RDP sessions.
       - Enforce proper session terminations and restrict administrative actions.

  2. Token-Based Attacks
     Tokens store authentication context for Single Sign-On (SSO). Attackers exploit delegate tokens to impersonate users or escalate privileges. Tools like Mimikatz and Metasploit target systems with incomplete session logoffs.
     - Mitigations:
       - Restrict SeImpersonate privileges.
       - Use Restricted Admin Mode and Remote Credential Guard.
       - Mark sensitive accounts as "Sensitive and Cannot be Delegated" in Active Directory.
  3. Cached Credentials
     Windows caches credentials to allow logons when domain controllers are unavailable, storing hashes that can be extracted with tools like cachedump or PWDumpX.
     - Mitigations:
       - Limit cached credentials using the cachedlogonscount registry setting.

    - Enforce strong passwords to resist brute-force attacks.
    - Use Domain Protected Users groups to prevent credential caching for privileged accounts

4. LSA Secrets
   Stored in the registry, LSA Secrets contain sensitive data for service accounts, VPNs, and auto-logons. Tools like Mimikatz and Cain decrypt and expose these secrets.
   - Mitigations:
     - Reduce the use of privileged accounts for services.
     - Implement Group Managed Service Accounts (gMSA) to automate password management.
     - Regularly audit service accounts and privileged credentials.
5. Kerberos Ticket Theft
   Kerberos issues tickets for authenticated users, which attackers steal from memory for Pass-the-Ticket attacks. Techniques like Golden Ticket and Kerberoasting allow attackers to forge tickets or escalate privileges.
   The table below summarizes common Kerberos attack types, their descriptions, and recommended mitigation strategies

Attack Type	Description	Mitigation
Pass the Ticket	An attacker captures a Kerberos ticket from the memory of a compromised system and reuses or transfers it to other systems in the network, allowing unauthorized access without the need for credentials.	Credential Guard; Remote Credential Guard
Overpass the Hash	An attacker uses the NT (New Technology) hash of a user's password to request service tickets without needing the plain-text password. This method bypasses traditional authentication by substituting the hash for the password.	Credential Guard; Protected Users Group; disable RC4 authentication
Kerberoasting	The attacker requests a service ticket for a privileged service account (such as an administrator) and extracts the NTLM hash from the ticket. The hash is then cracked offline, potentially revealing the account's password.	Long and complex service account passwords; Managed Service Accounts
Golden Ticket	The attacker creates a forged Kerberos TGT (Ticket Granting Ticket) for any account, including high-privileged accounts like domain admins. This ticket has no expiration and persists even after password resets, allowing indefinite access to the network.	Protect domain admin accounts; change KRBTGT password regularly
Silver Ticket	The attacker forges a service ticket (TGS) for a specific service (e.g., a file server or web app) and gains unauthorized access to that service. This method bypasses the domain controller by going directly to the service.	Regular computer account password updates
Skeleton Key	The attacker installs a backdoor on the domain controller by patching LSASS, allowing them to set a universal password (skeleton key) that grants access to any account in the domain, while legitimate credentials continue to function as normal.	Protect domain admin accounts; smart card usage for privileged accounts
DC Sync	The attacker mimics a domain controller and uses the replication process to extract password hashes and other sensitive data from the Active Directory. This method enables attackers to directly obtain authentication data from the domain controller.	Protect domain admin; audit/limit accounts with replication rights

6. NTDS.DIT File Exploitation
   The NTDS.DIT file contains all domain account hashes and is a prime target. Attackers use tools like ntdsutil or Volume Shadow Copy to access it.

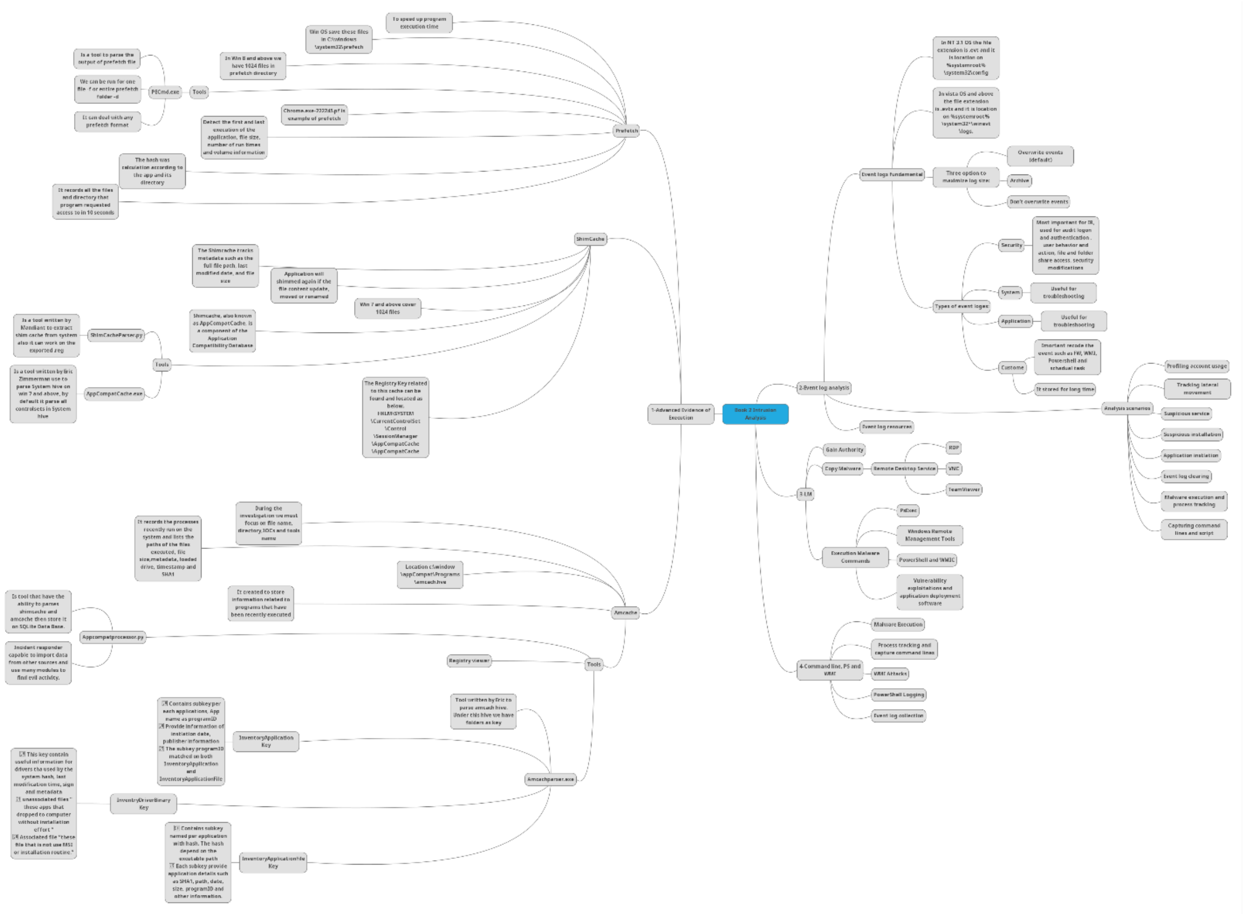
- Mitigations:
    - Protect domain controllers.
    - Restrict access to Volume Shadow Copy
    - Monitor suspicious processes.
- The table below outlines various types of credentials that can be targeted during a security breach, including where they are stored, the tools typically used to extract them, and additional notes about each type. This information is critical for understanding potential attack vectors and how attackers exploit system vulnerabilities to gain unauthorized access

	Description	Location	Tools Used to Extract	Notes
Hashes	Password hashes used for authentication in Windows, typically stored in NTLM and LM formats.	Local Accounts: Stored in the SAM database located at: C:\Windows\System32\config\SAM Domain Accounts: Stored in NTDS.DIT file on Domain Controllers located at: C:\Windows\NTDS\NTDS.DIT	Mimikatz, Pwdump, FGDump, Metasploit, Cain & Abel	The SAM database is protected and can only be accessed by the system. NTDS.DIT stores domain accounts and password hashes. Tools like Mimikatz can extract hashes directly from memory or files.
Tokens	Security tokens that represent user identity and permissions for accessing system resources.	Stored in memory within the LSASS process.	Mimikatz, Rubeus, Incognito	Tokens are created at login and remain in memory for the duration of the user session. Mimikatz and Rubeus can extract or manipulate tokens for privilege escalation.
Cached Credentials	Locally cached credentials that allow login without direct access to a domain controller.	Stored in the Registry at: HKEY_LOCAL_MACHINE\Security\Cache	Mimikatz, Cachedump, Metasploit	Cached credentials are encrypted by the system. Mimikatz and Cachedump can extract cached credentials for offline attacks.
LSA Secrets	Sensitive information stored by the Local Security Authority (LSA), such as service account passwords.	Stored in the Registry at: HKEY_LOCAL_MACHINE\Security\Policy\Secrets	Mimikatz, Gsecdump, Metasploit	LSA secrets include service account passwords, VPN credentials, and other sensitive data. Mimikatz can extract these secrets from the registry.
Tickets	Kerberos tickets used for authenticating users in domain environments.	Stored in memory within the LSASS process. Also cached in: C:\Users\<username>\AppData\Local\Microsoft\Credentials	Mimikatz, Rubeus, Kerberoast, Impacket	Mimikatz and Rubeus are commonly used to extract Kerberos tickets for Pass-the-Ticket attacks. Kerberoast can extract service tickets for offline cracking.
NTDS.DIT	The Active Directory database containing all domain user accounts and related information.	Stored on Domain Controllers at: C:\Windows\NTDS\NTDS.DIT	NTDSUtil, Metasploit, Impacket, DSInternals	NTDS.DIT stores the entire Active Directory database, including user accounts and password hashes. Tools like Impacket and DSInternals can be used to dump or extract information from it.

- BloodHound is a powerful tool used to map and analyze relationships within Active Directory environments, helping attackers and defenders visualize connections between users, computers, groups, and other objects. It identifies paths for privilege escalation, such as how a regular user could potentially reach Domain Admin status by exploiting trust relationships or misconfigurations. BloodHound acts as both an audit and attack tool, significantly reducing the effort required to find these vulnerabilities. While it's difficult to detect due to its reliance on standard Active Directory queries, related tools like GoFetch can be noisier and easier to spot in well-monitored networks.

Intrusion Analysis

Cyber defenders have a wide variety of tools and artifacts available to identify, hunt, and track adversary activity in a network. Each attacker action leaves a corresponding artifact, and understanding what is left behind as footprints can be crucial to both red and blue team members. Attacks follow a predictable pattern, and we focus our detective efforts on immutable portions of that pattern. As an example, at some point an attacker will need to run code to accomplish their objectives. We can identify this activity via application execution artifacts. The attacker will also need one or more accounts to run code. Consequently, account auditing is a powerful means of identifying malicious. An attacker also needs a means to move throughout the network, so we look for artifacts left by the relatively small number of ways there are to accomplish internal lateral movement. In this section, we cover common attacker tradecraft and discuss the various data sources and forensic tools you can use to identify malicious activity in the enterprise. Refer to



Book 2 Intrusion Analysis

1-Advanced Evidence of Execution

Prefetch

To speed up program execution time

Win OS save these files in C:\windows \system32\prefetch

In Win 8 and above we have 1024 files in prefetch directory

Tools

PECmd.exe

Is a tool to parse the output of prefetch file

We can be run for one file -f or entire prefetch folder -d

It can deal with any prefetch format

Chrome.exe-222245.pf is example of prefetch

Detect the first and last execution of the application, file size, number of run times and volume information

The hash was calculation according to the app and its directory

It records all the files and directory that program requested access to in 10 seconds

ShimCache

The Shimcache tracks metadata such as the full file path, last modified date, and file size

Application will shimmed again if the file content update, moved or renamed

Win 7 and above cover 1024 files

Shimcache, also known as AppCompatCache, is a component of the Application Compatibility Database

Tools

ShimCacheParser.py

Is a tool written by Mandiant to extract shim cache from system also it can work on the exported .reg

AppCompatCache.exe

Is a tool written by Eric Zimmerman use to parse System hive on win 7 and above, by default it parse all controlsets in System hive

The Registry Key related to this cache can be found and located as below. HKLM\SYSTEM \CurrentControlSet \Control \SessionManager \AppCompatCache \AppCompatCache

Amcache

During the investigation we must focus on file name, directory, IOCs and tools name

It records the processes recently run on the system and lists the paths of the files executed, file size, metadata, loaded drive, timestamp and SHA1

Location c:\window \appCompat\Programs \amcach.hve

It created to store information related to programs that have been recently executed

Tools

Appcompatprocessor.py

Is tool that have the ability to parses shimcache and amcache then store it on SQLite Data Base.

Incident responder capable to import data from other sources and use many modules to find evil activity.

Registry viewer

Amcachparser.exe

Tool written by Eric to parse amcach hive. Under this hive we have folders as key

InventoryApplication Key

Contains subkey per each applications, App name as programID
Provide information of instlation date, publisher information
The subkey programID matched on both InventoryApplication and InventoryApplicationFile

InventoryDriverBinary Key

This key contain useful information for drivers tha used by the system hash, last modification time, sign and metadata
unassociated files " these apps that dropped to computer without installation effort "
Associated file "these file that is not use MSI or installation routine."

InventoryApplicationFile Key

Contains subkey named per application with hash. The hash depend on the excutable path
Each subkey provide application details such as SHA1, path, date, size, programID and other information.

2-Event log analysis

Event logs fundemental

In NT 3.1 OS the file extension is .evt and it is location on %systemroot% \system32\config

In vista OS and above the file extension is .evtx and it is location on %systemroot% \system32*\winevt \logs.

Three option to maximize log size:

Overwrite events (default)

Archive

Don't overwrite events

Types of event loges:

Security

Most important for IR, used for audit logon and authentication , user behavior and action, file and folder share access, security modifications

System

Useful for troubleshooting

Application

Useful for troubleshooting

Custome

Imortant recode the event such as FW, WMI, Powershell and schadual task

It stored for long time

Analysis scenarios

Profiling account usage

Tracking lateral movement

Suspicious service

Suspicious installation

Application instlation

Event log clearing

Malware execution and process tracking

Capturing command lines and script

Event log resources

3-LM

Gain Authority

Copy Malware

Remote Desktop Service

RDP

VNC

TeamViewer

Execution Malware Commands

PsExec

Windows Remote Management Tools

PowerShell and WMIC

Vulnerability exploitations and application deployment software

4-Command line, PS and WMI

Malware Execution

Process tracking and capture command lines

WMI Attacks

PowerShell Logging

Event log collection

2.1 Advanced Evidence of Execution

Understanding which programs were running prior to an investigation is crucial, as it provides several clues for identifying evidence of execution.

2.1.1 Prefetch

- Windows operating systems store Prefetch files in C:\Windows\System32\Prefetch to enhance performance by preloading code into memory before it is required.
- A typical Prefetch file name example is Chrome.exe-222245.pf. The hash value in the Prefetch filename is computed based on the application's directory. If a Prefetch file such as Chrome.exe-333345.pf is found, it indicates that Chrome was executed from a different path, potentially signaling malicious activity.
- However, Windows-hosted applications like svchost.exe, dllhost.exe, rundll32.exe, and backgroundtaskhost.exe calculate their hash values using both the path and the command-line arguments.
- Prefetch is enabled by default on Windows workstations but can be disabled through the registry at: Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParameters.
- Windows 8 and later systems can store up to 1,024 Prefetch files.
- Tools like PECmd.exe allow parsing of Prefetch files. It can analyze a single file using the -f option or an entire folder using the -d option. The tool provides insights such as the first and last execution times, file size, number of runs, volume information, and directories or files accessed by the application. It supports compressed Prefetch formats used in Windows 10 (.pf files).
- Microsoft implemented Prefetch to improve program load times by preloading dependencies. For example, when program.exe runs, it may require program.dll, other system DLLs, and a configuration file. Normally, these dependencies are loaded sequentially as needed. Prefetch records files accessed during the first 10 seconds of execution and preloads them for future runs, reducing wait times.
- The creation date of a Prefetch file typically represents the first execution of the program, while the modified date reflects the most recent execution. Prefetch files also store the number of times a program has been executed and a list of accessed files. However, if the Prefetch directory exceeds the maximum number of files, older entries are deleted, potentially causing a new Prefetch file to misrepresent the true first execution time.
- It is essential to corroborate Prefetch data with other evidence sources. For instance, if an application was first executed on 2/2/2018, but the Prefetch directory reached its capacity of 1,024 files, older Prefetch files would be deleted. If the program was later run

on 2/2/2023, the newly created Prefetch file would show 2023 as the first execution date, not 2018

2.1.2 ShimCache

ShimCache, also known as the Application Compatibility Cache, records execution-related information about applications. This cache supports the "Compatibility Mode" feature in Windows, which helps troubleshoot compatibility issues by storing basic metadata about executables and their dependencies. It is important to note that ShimCache data for the current session resides in memory and is only written to disk upon shutdown or reboot.

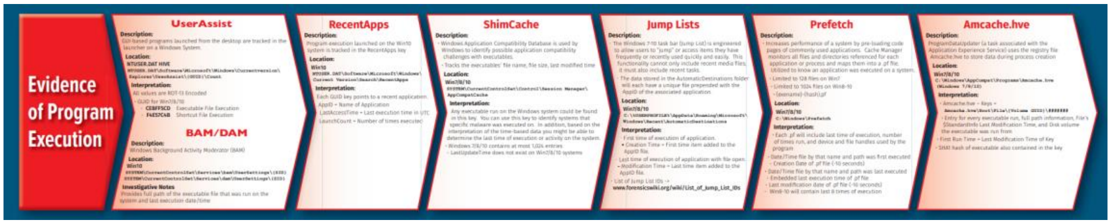
- Most applications, including older programs, games, or utilities designed for earlier versions of Windows, remain functional in Windows 10 and 11 through the Application Compatibility feature.
- ShimCache (AppCompatCache) is a component of the Application Compatibility Database developed by Microsoft. It allows the operating system to quickly determine whether compatibility shims are needed for a module.
- Eric Zimmerman's AppCompatCache tool parses the System Hive in Windows 7 and later. By default, it processes all control sets within the hive.
- ShimCacheParser.py, created by Mandiant, extracts ShimCache data from the System Hive or an exported .reg file.
- A Shim is a small library that facilitates compatibility by allowing applications to use older APIs in newer environments or vice versa. This mechanism provides backward and forward compatibility on various software platforms.
- The registry key associated with ShimCache is located at: HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\AppCompatCache\AppCompatCache
- Metadata Tracked by ShimCache
  - Full file path
  - Last modified date
  - File size
- Events in ShimCache.hve are organized in chronological order, with the most recent event listed first.
- ShimCache data is a valuable resource for recreating event timelines to identify and investigate malicious activities.
- If the content of a file is updated, moved, or renamed, the application may be shimmed again, creating a new entry in the cache.

2.1.3 Amcache

AmCache.hve is a replacement for the RecentFilesCache used in older versions of Windows. It stores detailed information about programs that have been recently executed on the system, making it a valuable artifact in forensic investigations. This cache provides insights into processes, file paths, file sizes, metadata, loaded drives, timestamps, and SHA1 hashes.

- Unlike Prefetch, entries in AmCache do not necessarily confirm program execution. Prefetch files can be used to validate execution.
- Location: C:\Windows\AppCompat\Programs\Amcache.hve
- Important Registry Keys in AmCache
    - InventoryApplicationFile
      Contains subkeys for each application, identified by a hash derived from the executable path.
        - Subkeys provide detailed information, including:
            - SHA1 hash
            - Path
            - Date and size
            - ProgramID
    - InventoryApplication
      Contains subkeys for each application, where the subkey name corresponds to the ProgramID.
        - Provides:
            - Installation date
            - Publisher information
            - The ProgramID links InventoryApplication and InventoryApplicationFile entries.
    - InventoryDriverBinary
      Tracks drivers loaded by the system, often targeted by attackers.
        - Driver files typically have a .sys extension and are located in %SystemRoot%\Drivers.
        - Contains:
            - Driver hash
            - Last modification time
            - Digital signature and metadata
        - This key helps identify unassociated files—files that were dropped onto the system without formal installation.
- Tools for Parsing AmCache
    - AmcacheParser.exe (by Eric Zimmerman): Parses AmCache hive files, organizing data under keys such as InventoryApplicationFile, InventoryApplication, and InventoryDriverBinary.
    - AppCompatProcessor.py: Parses both ShimCache and AmCache, storing the output in an SQLite database. It allows importing data from tools like FireEye and Redline and provides modules for identifying malicious activity.

  - Registry Viewer: A common tool for examining registry contents.
- When analyzing AmCache, focus on:
  - Suspicious file names (e.g., single-letter or uncommon names)
  - Execution directories (e.g., %TEMP% or $Recycle.Bin)
  - Indicators of Compromise (IOCs)
  - Known tools used by attackers, such as scrcons.exe, certutil.exe, wmic.exe, and nmap.exe.



Evidence of Program Execution

UserAssist

Description:
GUI-based programs launched from the desktop are tracked in the launcher on a Windows System.
Location:
NTUSER.DAT HIVE
NTUSER.DAT\Software\Microsoft\Windows\Currentversion\Explorer\UserAssist\{GUID}\Count
Interpretation:
All values are ROT-13 Encoded
- GUID for Win7/8/10
  - CEBFF5CD Executable File Execution
  - F4E57C4B Shortcut File Execution

BAM/DAM

Description:
Windows Background Activity Moderator (BAM)
Location:
Win10
SYSTEM\CurrentControlSet\Services\bam\UserSettings\{SID}
SYSTEM\CurrentControlSet\Services\dam\UserSettings\{SID}
Investigative Notes
Provides full path of the executable file that was run on the system and last execution date/time

RecentApps

Description:
Program execution launched on the Win10 system is tracked in the RecentApps key
Location:
Win10
NTUSER.DAT\Software\Microsoft\Windows\Current Version\Search\RecentApps
Interpretation:
Each GUID key points to a recent application.
AppID = Name of Application
LastAccessTime = Last execution time in UTC
LaunchCount = Number of times executed

ShimCache

Description:
- Windows Application Compatibility Database is used by Windows to identify possible application compatibility challenges with executables.
- Tracks the executables' file name, file size, last modified time
Location:
Win7/8/10
SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache
Interpretation:
Any executable run on the Windows system could be found in this key. You can use this key to identify systems that specific malware was executed on. In addition, based on the interpretation of the time-based data you might be able to determine the last time of execution or activity on the system.
- Windows 7/8/10 contains at most 1,024 entries
- LastUpdateTime does not exist on Win7/8/10 systems

Jump Lists

Description:
- The Windows 7-10 task bar (Jump List) is engineered to allow users to "jump" or access items they have frequently or recently used quickly and easily. This functionality cannot only include recent media files; it must also include recent tasks.
- The data stored in the AutomaticDestinations folder will each have a unique file prepended with the AppID of the associated application.
Location:
Win7/8/10
C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations
Interpretation:
- First time of execution of application.
  - Creation Time = First time item added to the AppID file.
- Last time of execution of application with file open.
  - Modification Time = Last time item added to the AppID file.
- List of Jump List IDs -> www.forensicswiki.org/wiki/List_of_Jump_List_IDs

Prefetch

Description:
- Increases performance of a system by pre-loading code pages of commonly used applications. Cache Manager monitors all files and directories referenced for each application or process and maps them into a .pf file. Utilized to know an application was executed on a system.
- Limited to 128 files on Win7
- Limited to 1024 files on Win8-10
- (exename)-(hash).pf
Location:
Win7/8/10
C:\Windows\Prefetch
Interpretation:
- Each .pf will include last time of execution, number of times run, and device and file handles used by the program
- Date/Time file by that name and path was first executed
  - Creation Date of .pf file (-10 seconds)
- Date/Time file by that name and path was last executed
  - Embedded last execution time of .pf file
  - Last modification date of .pf file (-10 seconds)
  - Win8-10 will contain last 8 times of execution

Amcache.hve

Description:
ProgramDataUpdater (a task associated with the Application Experience Service) uses the registry file Amcache.hve to store data during process creation
Location:
Win7/8/10
C:\Windows\AppCompat\Programs\Amcache.hve (Windows 7/8/10)
Interpretation:
- Amcache.hve – Keys = Amcache.hve\Root\File\{Volume GUID}\#######
- Entry for every executable run, full path information, File's $StandardInfo Last Modification Time, and Disk volume the executable was run from
- First Run Time = Last Modification Time of Key
- SHA1 hash of executable also contained in the key

2.2 Event Log Analysis For Incident Responder And Hunters

2.2.1 Event Logs Fundamentals

Event logging was first introduced with Windows NT 3.1 in 1993.

- Earlier versions: Used the .evt file extension with logs named secevent, appevent, and sysevent, stored at: %systemroot%\system32\config
- Significant changes in Windows Vista and Later for file extension to .evtx. Logs renamed to Security, Application, System, and custom logs and stored in %systemroot%\system32\winevt\logs. Logs can be sent to a remote collector for centralized analysis and monitoring.
- To manage storage effectively, Windows offers three options:
  - Overwrite Events as Needed (default).
  - Archive Logs When Full.
  - Do Not Overwrite Events (risk of missed logs when full).
- Types of Event Logs
  - Security Logs
    - Significance for Incident Response
    - Audits logon attempts, user behavior, authentication events, and access to files, folders, and security policies.
    - Updated Exclusively by LSASS Process (Local Security Authority Subsystem Service).
    - Security Event Categories:
      - Account Logon Events
      - Logon Events
      - Account Management

    - Directory Service Access
    - Object Access
    - Policy Changes
    - Privilege Use
    - Process Tracking
    - System Events
  - System Logs
    Contains information useful for troubleshooting operating system-level issues.
  - Application Logs
    Provides event data related to specific applications for application-level troubleshooting.
  - Custom Logs
    Logs events for specific services and tools like Firewall, WMI, PowerShell, and Scheduled Tasks. Often retained for extended periods for detailed auditing and analysis.

2.2.2 Analysis scenarios

2.2.2.1 Profiling Account Usage

Account usage profiling is crucial for identifying potential compromises. The following steps and Event IDs are useful in tracking and analyzing account activity.

1. Detecting Compromised Accounts

- Login Attempts
  Use the Event Viewer to investigate login attempts and identify compromised accounts.
- Logon Type Codes
  Logon events (Event ID 4624) reveal critical information about account access types:

Logon Type	Description
2	Logon via console
3	Network logon
4	Batch logon
5	Windows service logon
7	Unlocking the screen using credentials
8	Network logon with cleartext credentials
9	Alternate credentials (RunAs)
10	Remote interactive logon (RDP)
11	Cached credentials
12	Cached remote interactive logon
13	Cached unlock

2. Identifying Logon Sessions

   - Compare 4624 and 4647 events. If the Security ID and Logon ID match, this indicates the same session, even if the username changed.
   - You can determine how long a user was logged in by analyzing the time between these events.
3. Detecting Brute Force Attacks
   - Event ID 4625 records failed login attempts, often signaling brute force activity.
4. Excluding Noisy Built-in Accounts
   - Common system accounts such as SYSTEM, LOCAL SERVICE, NETWORK SERVICE, hostname$, DWM, and WMFD create noise in logs. Exclude these from your search to reduce false positives.
5. Tracking Privileged Account Activity
   - Event ID 4672 tracks administrator-level actions.
6. Monitoring Account Creation
   - Event ID 4720 indicates when new accounts are created.
7. Tracking RDP Usage
   - Key events for Remote Desktop Protocol (RDP) activity:
     - 4778: Reconnect
     - 4779: Disconnect
     - 4624: Connect (logon type 3, 7, or 10)
     - Use 4778 to gather details about the attacker's machine.
     - Source system events include 4648, 1024, 1102.
     - Destination system events include 4624 (type 10), 4778, 4779, and others like 131, 98, 1149, 21, 22, 25, 41.
8. Domain Logon Events
   - NTLM authentication: Event ID 4776
   - Kerberos authentication:
     - 4768: Ticket Granting Ticket (TGT) request
     - 4769: Ticket Granting Service (TGS) request
     - 4771: Failed TGT request
9. Privilege Abuse and Pass-the-Hash Attacks
   - Monitor 4776 and 4624 for abuse of local accounts and Pass-the-Hash techniques.
10. Reconnaissance Activity
    - 4798: User enumeration
    - 4799: Group enumeration
    - Commands like cmd, PowerShell, and WMI are commonly used for reconnaissance.
    - Enumeration of All Computer Groups: Use 4799 for comprehensive group enumeration logs.

- Tools for Event Log Analysis

  - **Event Log Explorer** provides advanced features for managing and analyzing Windows event logs, surpassing the capabilities of the built-in Event Viewer
  - EvtxECmd is a command-line tool for parsing and analyzing Windows event logs with powerful options for forensic investigations.

2.2.2.2 Tracking Lateral Movement

- Network Share Events
  - Event ID (EID) 5140: Detects access to network shares.
  - EID 5145: Provides more detailed information on network share usage.
  - EIDs 5142–5144: Used for tracking specific changes in shared resources.
- Cobalt Strike and Lateral Movement
  Cobalt Strike often leverages network shares to move laterally within a network. The following sequence is typical:
  - EID 4624 (Type 3) logs a user's network logon.
  - Followed by EID 5140 for the computer account (e.g., admin$ or IPC$ shares with 127.0.0.1).
  - A subsequent EID 5140 logs the same activity for the user, including the user's IP and share folder (IPC$).
- Lateral Movement with Runas
  - Runas with Different Credentials:
    - The source machine logs EID 4648, showing the source account, IP, destination account, and hostname.
    - The destination machine logs the source IP, destination account, hostname, and the share folder.
  - Cobalt Strike make_token or Pass-the-Hash (PTH):
    This technique uses explicit credentials for lateral movement, typically leveraging PowerShell. Detect this activity by checking:
    - EID 4624 (Type 9): Indicates a new token generation.
    - EID 4648: Displays the remote workstation accessed.
- Scheduled Task Execution for Lateral Movement
  - Security Log :
    - 4698: Task created
    - 4702: Task updated
    - 4699: Task deleted
    - 4700 & 4701: Task completion
  - Task Scheduler Log:
    - 106: Task registered
    - 140: Task updated
    - 141: Task deleted
    - 200 & 201: Task execution

  - Scheduled tasks can be executed locally or remotely.
  - Remote schedule task activity logs include EID 4624 (Type 3).
  - Scheduled tasks are used for both lateral movement and persistence.
  - Scheduled Task Versions
    - Version 1.0 (Windows XP and 2003):
      - Provides limited information.
      - Stored at: %systemRoot%\tasks with .job extension (binary format).
      - Use tools like jobparser.py or jobparser.pl for parsing.
    - Version 1.2 (Windows Vista and Server 2008 onward):
      - Provides detailed information, including:
      - Task registration time
      - User who registered the task
      - User executing the task
      - Path and trigger schedule
      - Stored in:
      - %systemRoot%\system32\tasks or %systemRoot%\sysWOW64\tasks

2.2.2.3 Suspicious Service

- Key Event IDs for Service Monitoring
  - Event ID 7034: A service terminated unexpectedly (crashed).
  - Event ID 7035: A service start or stop control was sent.
  - Event ID 7036: A service started or stopped.
  - Event ID 7040: A service type was changed.
  - Event ID 7045: A new service was installed (system log).
  - Event ID 4697: A new service was installed on the system (security log).
- Metasploit often uses PsExec for lateral movement, creating a service for remote code execution. PsExec is a tool from the Sysinternals Suite used for remote command execution. Typical behavior when PsExec is used:
  - The service name may be registered as PSEXESVC or a random name.
  - Commands are executed using %COMSPEC%, which points to cmd.exe.
  - Each time PsExec runs, it downloads its service to the remote machine.
- Detection Tips:
  - Look for frequent or unexpected service installations (EIDs 7045, 4697).
  - Monitor unusual or random service names associated with remote commands.
  - Investigate repeated PsExec activity, especially when associated with suspicious network logons (EID 4624 Type 3).

2.2.2.4 Suspicious Installation

2.2.2.5 Application Installation

2.2.2.6 Event Log Clearing

- If the Security logs are cleared, Event ID 1102 will likely be recorded in the Security logs themselves. This event typically indicates that the Security log has been cleared.
- If any logs other than the Security log are cleared, Event ID 104 will usually appear in the System log. This event signifies that the associated log has been cleared.
- Log deletion typically requires elevated privileges, such as:
    - Local Administrator: Provides full control over the local system.
    - Domain Administrator: Possesses administrative rights across the entire domain.
    - Local System: A powerful built-in account with extensive system privileges.
- Logs can be deleted through various methods:
    - Graphical User Interface (GUI): Using tools like Event Viewer.
    - Command-Line Interface (CLI): Employing PowerShell cmdlets, the eventvwr command (for Event Viewer), or the wevtutil command-line utility.
- Several tools can be used to interfere with logon events, including:
    - Mimikatz: A powerful penetration testing tool capable of extracting credentials and manipulating system memory.
    - DanderSpritz: A tool designed to evade detection and manipulate system processes.
    - Invoke-Phantom: A PowerShell script used for red teaming and penetration testing, often involving credential theft and privilege escalation.

2.2.2.7 Malware Execution And Process Tracking

2.2.2.8 Capturing Command Lines And Script

2.2.3 Event Log Resources

- Event log collection
    - Live system collection
        - Event Viewer: You can directly export event logs from the Event Viewer in various formats like .evt, .evtx, .csv, .xml, and .txt.
        - PsLogList: This Sysinternals tool offers command-line capabilities for collecting event logs, including dumping logs to text or .csv files, reading native .evt/.evtx formats, pre-filtering output, and even collecting logs from remote systems.
        - Triage Collection Tools: Solutions like F-Response, KAPE, and Velociraptor can be used for triage collection, allowing for quick and efficient gathering of event logs from live systems.
        - PowerShell: PowerShell provides native access to event logs, enabling scripting for automated and scalable collection.
    - Log forwarding

        - Windows Event Forwarding (WEF): A built-in Windows feature that allows you to forward event logs from one computer to another, centralizing log collection for easier analysis.
        - Winlogbeat: An open-source log shipper from the Elastic stack that can collect event logs from various sources, including Windows systems, and forward them to a central location like Elasticsearch.
    - System monitor Sysmon
      Sysmon is a high-fidelity and low-overhead system monitoring tool specifically designed for Digital Forensics and Incident Response (DFIR).
      Functionality: It collects detailed information about various system activities, including process creation, network connections, file system changes, and more.
    - Event log analysis resources
        - EventID.net
        - Ultimatewindowssecurity.com
        - Microsoft

2.3 Lateral Movement Adversary Tactics

2.3.1 Copying Malware via Remote Desktop Service

Attackers often use RDP (Remote Desktop Protocol), VNC, and TeamViewer to access and copy malware to remote machines.

2.3.2 Copying Malware via Windows Administrative Shares

Leverages the default administrative shares (e.g., \<hostname>\admin$) on Windows systems to access sensitive files and potentially deploy malware.

2.3.3 Executing Malware with PsExec

PsExec is a lightweight tool from Sysinternals for executing commands on remote systems. It is commonly used by both administrators and adversaries.

- On the source machine, the artifact is psexec.exe.
- On the destination machine, psexesvc is created temporarily.
- psexec \\<ip-address> -u <username> -p <password> cmd

2.3.4 Executing Malware via Command Line

Leverages various command-line tools for remote execution:

- sc: Creates and starts remote services.
- at: Schedules remote tasks.
- reg add: Interacts with remote registry keys.
- winrs: Executes any remote command.

2.3.5 Execution Malware via WMI

WMI can execute commands and scripts remotely, making it a common technique for lateral movement.

2.3.6 Execution Malware via PowerShell

PowerShell is a powerful tool often exploited for executing scripts and commands on local and remote systems.

2.3.7 Application Deployment Software

Attackers may leverage legitimate deployment tools to propagate malware across systems.

2.3.8 Exploiting Vulnerabilities

Exploiting known or zero-day vulnerabilities is another tactic for lateral movement. This involves gaining unauthorized access to systems by taking advantage of unpatched software or configuration weaknesses.

2.4 Command Line, PowerShell And WMI Analysis

When identifying and investigating malware activity:

- Reviewing System and Application logs, particularly Critical, Warning, and Error events, for identifying potential malware execution, as Security logs may be less comprehensive due to audit policy limitations.
- Check Antivirus and Windows Defender Logs for any warnings or detections of suspicious activity.
- Check Windows Error Reporting (.WER) Files located in C:\ProgramData\Microsoft\Windows\WER, provide valuable information such as hashes, execution paths, timestamps, and involved DLLs, aiding in malware identification and investigation.
- Monitor Process Tracking
  Review Security event IDs 4688 and 4689 for the full commands used to launch processes.
- Investigate WMI Activity

  Check 4688 (Security log), Sysmon, and Endpoint Detection and Response (EDR) solutions to detect WMI Activity. WMI is often exploited by attackers for reconnaissance, privilege escalation and lateral movement.

    - Reconnaissance
      Example commands: process get, qfe get, startup get, useraccount list full, group list full, netuse list full
    - Privilege Escalation

    Techniques include using wmic.exe or PowerShell-based WMI commands. Monitoring command-line audit logs and PowerShell logging helps detect these actions. Tools are available to identify system misconfigurations that allow privilege escalation.

  - Lateral Movement
    Example commands: process call create \C:\win\system32\rundll32.dll \\\c:\windows\\malware.bat\\\
- Identify Persistence Mechanisms
  Monitor event IDs 5858/5857/5861 and WMI-activity operational logs for indicators of persistence

- Analyze PowerShell Activity

    - Track PowerShell commands event logs IDs (103, 4104, 4105, 4106)
    - PowerShell stores its command history in a text file even after a reboot. By default, the history file is saved in the following location: C:\Users\<username>\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt.
    - Using the Set-PSReadLineOption cmdlet to configure the history save behavior.
    - There various PowerShell syntax options to achieve stealthy execution. These techniques aim to make malicious PowerShell activity more difficult to detect by hiding the activity from traditional monitoring methods. Common Stealth Techniques:
      - -Hidden: Suppresses the PowerShell window from appearing on the screen.
      - -nop: Disables the loading of the PowerShell profile, which can contain malicious scripts or configurations.
      - -noni: Prevents the display of the interactive prompt, making it harder to identify PowerShell activity.
      - -exec bypass: Bypasses the execution policy, allowing the script to run even if the execution policy is set to block unsigned scripts.
      - Downloading Scripts from Remote Locations: Using commands like System.Net.WebClient.DownloadString() or Invoke-WebRequest to download and execute scripts from remote servers.

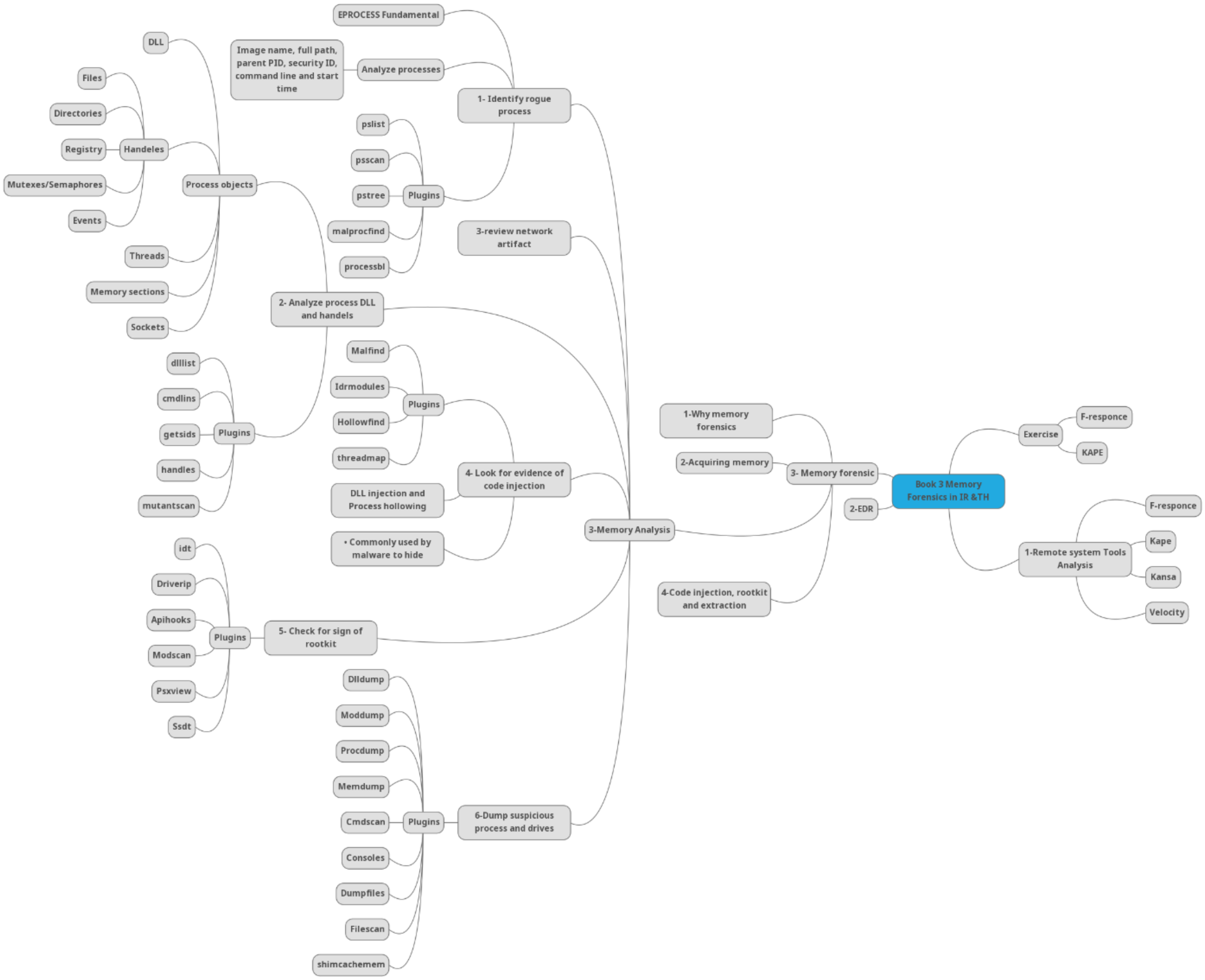
- The following table lists common Windows Event IDs and their descriptions

Event ID	Description
104	Audit log cleared (System Log)
200	Scheduled task executed (Task Scheduler)
201	Scheduled task completed (Task Scheduler)
1000	Application hangs Windows Error Reporting (WER) (Application Logs)
1002	Application Crashes Windows Error Reporting (WER) (Application Logs)
1102	The audit log was cleared
4624	An account was successfully logged on, Check Logon Type
4625	Failed Logon
4634	An account was logged off
4647	User initiated logoff (Session time)
4648	A logon was attempted using explicit credentials (Runas) / password is different than the current session user password
4672	Account logon with superuser rights (Administrator) with 4624
4688	A new process has been created
4689	A process has exited
4697	A service was installed in the system
4698	A scheduled task was created (Task Scheduler 106)
4699	A scheduled task was deleted (Task Scheduler 141)
4700	A scheduled task was enabled
4701	A scheduled task was disabled
4702	A scheduled task was updated (Task Scheduler 140)
4720	A user account was created , followed by 4732
4722	Account was enabled
4724	An attempt was made to reset an accounts password
4725	A user account was disabled
4726	A user account was deleted
4728	a member was added to security enabled global group
4732	a member was added to security enabled local group
4735	A security-enabled local group was changed
4738	A user account was changed
4756	A member was added to a security-enabled universal group
4768	A Kerberos authentication ticket (TGT) was requested
4769	successful authentication for (access to resource such as a file share)
4771	Kerberos pre-authentication failed
4772	A Kerberos authentication ticket request failed
4776	The domain controller attempted to validate the credentials for an account (NTML)
4778	A session was reconnected to a Window Station
4779	A session was disconnected from a Window Station
4798	A user's local group membership was enumerated.
4799	A security-enabled local group membership was enumerated

4801	The workstation was unlocked
5140	A network share object was accessed (no reference in file accessed)
5142	A network share object was added.
5143	A network share object was modified
5144	A network share object was deleted.
5145	A network share object was checked to see whether client can be granted desired access (accessed file individualy)
5861	New event Consumer was created (WMI-Activity/Operational Log)
7034	service crashed (System Log)
7035	service sent start/stop control (System Log)
7036	service started/stopped (System Log)
7040	(start type changed) (Boot, on request, Disabled) (System Log)
7045	(new service installed on system) (System Log)

Memory Forensics in IR &TH

Memory forensics has come a long way in just a few years. It is now a critical component of many advanced tool suites (notably EDR) and the mainstay of successful incident response and threat hunting teams. Memory forensics can be extraordinarily effective at finding evidence of worms, rootkits, PowerShell attacks, ransomware precursors, and advanced malware used by targeted attackers. In fact, some fileless attacks may be nearly impossible to unravel without memory analysis. Memory analysis was traditionally the domain of Windows internals experts and reverse engineers, but new tools, techniques, and detection heuristics have greatly leveled the playing field making it accessible today to all investigators, incident responders, and threat hunters. Further, understanding attack patterns in memory is a core analyst skill applicable across a wide range of endpoint detection and response (EDR) products, making those tools even more effective. This extremely popular section will cover many of the most powerful memory analysis capabilities available and give analysts a solid foundation of advanced memory forensic skills to super-charge investigations, regardless of the toolset employed. Refer to



3.1 Enterprise and Remote System Analysis

Many Incident Response (IR) tools are available for either accessing systems remotely or analyzing collected data.

- Access Agents:
  - Purpose: Facilitate remote data acquisition from target machines.
  - Lightweight and minimally invasive to preserve system integrity.
  - Capture both volatile (e.g., memory, running processes) and non-volatile (e.g., disk files) data for forensic investigations.
  - Remote data collection agents from tools like F-Response, KAPE, or Velociraptor.
- Analysis Agents:
  - Purpose: Examine and interpret the data acquired by access agents.
  - Equipped with advanced forensic capabilities to identify indicators of compromise (IOCs), analyze malware behavior, and provide insight into security incidents.
  - Often integrated into forensic workstations or part of comprehensive suites like EnCase, FTK
- Complementary Roles
  - Access Agents and Analysis Agents together form the foundation of effective digital forensic and incident response processes:
  - Access Agents prioritize data acquisition while minimizing system impact.
  - Analysis Agents specialize in deep investigation and evidence interpretation.

	Kansa	KAPE	F-Response	Velociraptor
Type	Script-based framework	Forensic data collection and analysis tool	Remote forensic and incident response tool	Advanced digital forensics and incident response platform
Primary Use	Data collection and analysis	Data collection, parsing, and analysis	Remote data access and acquisition	Data collection, hunting, and investigation
Deployment	PowerShell scripts	Standalone executable	Agent-based, requiring installation on target systems	Agent-based, supporting multiple platforms
Data Acquisition	Collects system information and logs	Acquires files and artifacts using predefined modules	Provides live access to disks, memory, and more	Acquires a wide range of data types, including files, registry, and memory
Analysis Capabilities	Basic analysis through PowerShell scripts	Advanced parsing and analysis with modules	Primarily focused on data acquisition, analysis depends on integrated tools	Built-in analysis capabilities with queries and scripts
Ease of Use	Requires PowerShell knowledge	User-friendly with GUI and command-line options	Intuitive for experienced users, with a focus on forensic professionals	Flexible with a learning curve for advanced queries

	Kansa	KAPE	F-Response	Velociraptor
Flexibility	Modular scripts allow for customization	Highly customizable with modules and targets	Focused on acquisition, integrates with other forensic tools for analysis	Highly customizable with VQL (Velociraptor Query Language)
Forensic Focus	General system data collection	Comprehensive artifact collection specific to forensic analysis	Emphasizes live forensic data acquisition and remote access	Broad focus including live response, forensics, and threat hunting
Integration	Can be integrated into larger workflows	Integrates with other forensic tools for further analysis	Often used in conjunction with analysis tools	Can integrate with third-party tools and services
OS	Windows	Windows	Windows	Cross-platform (Windows, macOS, Linux)
Community and Support	Open-source with community support	Maintained by Kroll, with community modules available	Commercial product with professional support	Open-source with active community and support

3.2 EDR Overview

Endpoint Detection and Response (EDR) solutions are critical components of modern cybersecurity strategies, providing real-time monitoring, threat detection, and incident response capabilities across endpoint devices. These platforms continuously collect and analyze endpoint activity data to detect advanced threats that evade traditional security measures.

- Key Capabilities of EDR Solutions
  - Continuous Data Collection and Analysis:
    EDR tools gather comprehensive data on processes, file activities, network connections, and user behaviors to detect suspicious or malicious activity.
  - Threat Detection and Alerts:
    EDR solutions leverage behavioral analysis, machine learning, and signature-based techniques to identify potential threats. When a threat is detected, security teams receive detailed alerts.
  - Incident Response Features:
    Endpoint Isolation: Prevents further spread of malware by disconnecting compromised systems from the network.
  - Forensic Data Collection: Provides detailed logs and telemetry for root cause analysis and investigation.
    Automated Response Actions: Enables swift remediation through actions like terminating malicious processes or deleting harmful files.
- EDR vs. Enterprise Forensics Tools
  Although EDR platforms offer valuable insights and data useful for forensic investigations, they are not designed to replace enterprise forensics tools. The primary function of EDR is

real-time threat monitoring and response, whereas enterprise forensic tools are tailored for in-depth post-incident analysis and detailed data recovery.
In summary, EDR solutions enhance endpoint visibility and enable proactive defenses against sophisticated attacks, contributing to an organization's overall security posture while complementing, but not substituting, dedicated forensic investigation tools.

3.3 Memory Forensics

3.3.1 Why memory forensics

Memory forensics is a vital investigative technique in modern cybersecurity, providing unique capabilities for malware detection and analysis. Unlike traditional disk-based forensics, memory forensics allows analysts to examine a system's live memory, where advanced malware often resides to evade detection. Many sophisticated threats manipulate memory structures or operate solely in-memory without leaving artifacts on the file system, making this technique essential for uncovering otherwise hidden malicious activity.

- Key Advantages of Memory Forensics
  - Detection of Stealthy Malware:
    Memory forensics can reveal malware variants that utilize fileless techniques, rootkits, or in-memory-only payloads. These threats often bypass traditional antivirus and endpoint security solutions by avoiding disk-based signatures.
  - Insights into Runtime System State:
    Analyzing memory captures provides a snapshot of active processes, threads, loaded drivers, and kernel objects. This can highlight unauthorized or malicious activities, such as injected code, hidden processes, or suspicious network connections.
  - Uncovering Indicators of Compromise (IOCs):
    Memory artifacts, including malicious DLLs, hooks, or strings related to command-and-control (C2) communications, help analysts piece together the tactics, techniques, and procedures (TTPs) used by adversaries.
  - Identifying Persistence Mechanisms:
    Some persistence techniques, like hooking or direct manipulation of process memory, can be directly observed only through memory analysis.
  - Tracking Network Activity:
    Examining sockets and connections to detect unauthorized data exfiltration or remote access tools (RATs).
  - Reverse Engineering Malware
    Extracting in-memory code for further analysis without relying on disk files.

3.3.2 Acquiring Memory

- In live system scenarios, specialized tools are used to capture volatile memory without powering off the system. These tools allow investigators to acquire memory images while preserving the system's state, providing a rich source of runtime data for forensic analysis. Common tools include:
    - F-Response
    - DumpIt
    - WinPMEM
    - FTK Imager
    - Redline
- Even after a system shutdown, valuable memory artifacts remain in files created during normal operations or crash events. These files, while not standard memory dumps, offer vital information for forensic investigations when analyzed alongside captured memory from live systems. These files include:
    - Memory Dump Files
        - Location: C:\Windows\Memory.dmp
        - Contains: System crash dumps for diagnostic purposes. Multiple minidump files may be available, named with timestamps to distinguish separate incidents.
    - Page File (Pagefile.sys) and Swap File (Swapfile.sys)
        - Location: C:\pagefile.sys and C:\swapfile.sys
        - Purpose: The page file provides virtual memory management by temporarily storing data from RAM. It allows the operating system to swap out less active data, freeing up physical memory.
    - Hibernation File (Hiberfil.sys)
        - Location: C:\hiberfil.sys
        - Purpose: Stores the entire RAM content when the system enters hibernation mode, enabling a rapid resume of the previous session.

3.3.2.1 Hibernation File

A hibernation file (typically stored as hiberfil.sys on Windows systems) saves the current state of the system to the hard drive before the computer enters hibernation. This process enables faster restarts and restores the system to its exact previous state upon waking. Unlike standard text documents, hiberfil.sys is not directly human-readable. It contains a raw, compressed, and encoded snapshot of the system memory, making it valuable for forensic analysis but requiring specialized tools to interpret.

- Several tools are used in digital forensics to handle hibernation files:
    - Volatility Framework: A widely used memory forensics tool capable of extracting and analyzing data from hibernation files.

- Arsenal Image Mounter: A tool for mounting and examining hibernation files and other forensic images.
- Comae Toolkit: Designed for advanced memory forensics, including hibernation file analysis.
- Passware: Known for its password recovery capabilities, Passware also supports hibernation file analysis in certain cases.

3.3.2.2 VM Memory Acquisition

For successful analysis, It is importance of understanding the specific virtualization platform and its memory storage format in acquiring and analyzing memory from virtual machines.

- The recommended technique is to suspend the VM, which forces a copy of the memory to be created on the host system.
- Memory Image Formats:
  - VMware (Fusion/Workstation/Server/Player):
    - .vmem: Raw memory image
    - .vmss and .vmsn: Contain memory image
  - Microsoft Hyper-V:
    - .bin: Memory image
    - .vsv: Save state
  - Parallels:
    - .mem: Raw memory image
  - VirtualBox:
    - .sav: Partial memory image
- Memory Analysis Tools: Raw memory images can be directly analyzed using memory analysis tools like Volatility.
- VirtualBox Considerations:
  - VirtualBox's .sav files only contain memory actively in use, not the entire assigned memory.
  - Volatility has limited support for analyzing VirtualBox images.
  - Debugvm feature can be used to force a full memory dump.
- VMware ESX and Hyper-V:
  - Memory might be stored in a more complex format that needs conversion.
  - Volatility has address spaces for analyzing some VMware ESX and Hyper-V memory files.

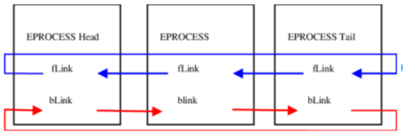
3.3.3 Introduction To Memory Analysis

- Effective memory forensics begins with two fundamental steps:
  - Memory Acquisition:
    - This involves capturing both physical and virtual memory, including files like:

        - Page File (pagefile.sys): Stores less frequently accessed memory pages.
        - Swap File (swapfile.sys): Used in modern Windows versions for additional memory management.
        - Hibernation File (hiberfil.sys): Stores a compressed snapshot of memory when the system hibernates.
    - Contextual Information:
      Memory forensics tools require knowledge of the target system's operating system version, architecture, and service pack to properly interpret data. Without these details, analysis may yield incomplete or inaccurate results.

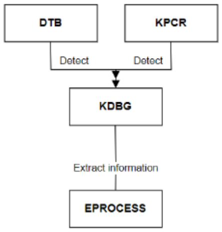
- Key Structures in Windows Memory Forensics

1. Kernel Debugger Data Block (KDBG)
   The KDBG structure is critical for analyzing memory dumps in Windows systems. It provides essential information for navigating the memory image, including pointers to key kernel structures.
    - The KDBG structure can be located through the Kernel Processor Control Region (KPCR) and Directory Table Base (DTB).
    - Identifies the Windows version and build for accurate profile application.
    - Provides access to crucial kernel structures like the Process List and Loaded Module List.
    - Essential for Volatility to initialize memory analysis.
2. Virtual Address Descriptors (VADs)
   VADs track memory allocations within a process. They are managed by the Windows memory manager and recorded in a VAD tree when a process uses functions like VirtualAlloc. Analyzing VAD structures helps investigators:
    - Identify memory regions allocated by a process.
    - Detect unauthorized memory injections or suspicious allocations.
3. EPROCESS Structure
   The EPROCESS (Executive Process) structure represents each process in the Windows operating system. It provides detailed process metadata, enabling forensic tools to:
    - Enumerate running and hidden processes.
    - Extract memory associated with a specific process.
    - Analyze process hierarchies and security contexts.



- Volatility relies on locating the KDBG structure to correctly interpret:
    - Process List: A linked list of _EPROCESS structures.
    - Loaded Module List: A list of _KLDR_DATA_TABLE_ENTRY structures representing kernel modules.
    - VADs: Provides insights into allocated memory regions.

By extracting and examining these structures, forensic investigators gain deep visibility into system activity, detect hidden threats, and identify indicators of compromise (IOCs).



- Volatility is a powerful, open-source memory forensics framework widely used in incident response and malware analysis. Designed to analyze volatile memory (RAM) dumps, it supports multiple operating systems, including Windows, Linux, and macOS. Built on Python, Volatility provides forensic investigators with the ability to extract and analyze artifacts from memory snapshots, offering insights into system activity, hidden threats, and indicators of compromise (IOCs).
- Steps to effectively analyze memory dump using Volatility 2:
  - Open a terminal window with administrative privileges.
  - Run the Volatility Framework by typing volatility or vol.py at the command prompt.
  - Identify the memory dump file you want to analyze using the -f option followed by the path to the file.
    ```
    # volatility -f memory.dump
    ```
  - Gather basic information about the memory dump, including the operating system and kernel version, using the imageinfo plugin or kdbgscan plugin. This information is crucial for selecting the appropriate profile for further analysis. Check the table below for the differences between these plugins
    ```
    # volatility -f memory.dump imageinfo
    # volatility -f memory.dump kdbgscan
    ```
  - Based on the information gathered from imageinfo and kdbgscan, select the most appropriate profile for your analysis. The Volatility Framework provides a variety of profiles for different operating systems and versions. You can specify the profile using the --profile option followed by the profile name.
    ```
    # volatility -f memory.dump --profile=xx
    ```
  - Once you have selected a profile, you can start using Volatility's plugins to analyze the memory dump. Each plugin has its own set of options and functionalities. You can explore the available plugins and their usage by using the -h or --help option after the plugin name.
    ```
    # volatility -f memory.dump --profile=xx pslist -h
    ```

	kdbgscan	imageinfo
Purpose	Locates and parses the Kernel Debugger Data Block (KDBG) to identify system parameters.	Provides a summary of the memory image, suggesting the most appropriate profile for analysis.
Usage	Used to find detailed system information, including OS version, loaded kernel modules, and active processes.	Used to get an overview of the memory dump, including the type of dump, and the capture time frame.
Output	Outputs the address of the KDBG structure, OS version, system compilation timestamp, and more.	Suggests profiles for analysis, dump type, creation date and time, and other general information.
Specificity	More specific, focusing on the KDBG structure for in-depth system information.	Broader, providing a general summary and profile suggestion for the memory image.
Analysis Stage	Typically used in the initial stages for detailed system analysis and profile identification.	Often used at the beginning to understand the basic context of the memory dump and suggest profiles.

- The imagecopy plugin in Volatility can be used to convert non-standard memory dumps, such as hibernation and crash dumps, into a more easily analyzable format. While generally effective, it may produce suboptimal results in some cases

  ```
  # volatility -f source_memory_dump.img --profile=xx imagecopy -O output_memory_dump.raw
  ```

- Memory forensics, detecting intrusions requires a systematic approach. Here's a breakdown of essential steps:

3.3.3.1 Identified The Rogue Processes

Rogue processes are those that are running on a system without authorization or are behaving suspiciously. Identifying these processes is a crucial step in intrusion investigations. There are many plugins in Volatility that can be used to identify rogue processes, including:

- Pslist: This plugin lists the processes that were running on the system when the memory snapshot was captured. It provides a detailed overview of active processes, including process IDs, parent process IDs, thread counts, and other relevant data, giving insight into system activity.
- Psscan: Unlike Pslist, which relies on in-memory process structures to enumerate processes, Psscan scans the memory dump to locate processes based on their signatures. This allows it to identify hidden or unlinked processes, such as those used by rootkits or malware attempting to evade detection. It also provides process creation and exit times, offering valuable forensic details.
- Pstree: This plugin presents a tree-like structure that visualizes the parent-child relationships between processes. It helps investigators understand the hierarchy

  of process creation, which can be instrumental in spotting suspicious or malicious activity.
  - Mailprocfind: While not a standard Volatility plugin, this custom or specialized plugin appears to be designed for locating and analyzing email-related processes. It may be particularly useful in cases involving email-based attacks or malware.
  - Processbl: Similar to Mailprocfind, this is not a core Volatility plugin. Based on its name, it likely serves as a "Process Blacklist" tool, identifying known malicious processes by comparing their attributes (e.g., names or IDs) against a predefined blacklist. This can expedite the identification of suspicious or harmful processes in a memory dump.
  - Baseline: This plugin aids in automating incident response by comparing memory objects from a suspect image to those from a baseline (a known clean image). It consists of three components—processbl, servicebl, and driverbl. It allows for filtering results by displaying items either absent (-U) or present (-K) in the baseline image, simplifying the detection of anomalies."

3.3.3.2 Analyze Process DLLs and Handles

Windows processes are complex entities comprising more than just executable files. Each process in Windows includes several components, such as:

  - DLLs (Dynamic Link Libraries): Libraries that provide additional functionality to processes.
  - Handles: References to system resources like files, directories, registry keys, mutexes/semaphores, and events.
  - Threads: Execution units within the process.
  - Memory Sections: Allocated areas of memory.
  - Sockets: Network communication endpoints.

- Analyzing these process objects enables investigators to uncover suspicious behaviors that may not be evident from process names or parent-child relationships alone.
- Key aspects to focus on during analysis include:
  - DLLs: These can reveal the capabilities of a process and indicate malicious injections or hidden functionalities.
  - Handles: These show which resources a process is accessing, potentially identifying unauthorized or suspicious activity.
  - Mutexes: Often used by malware to prevent multiple instances or signal its presence.
  - Threads, Events, and Memory Sections: Investigating these components can expose malware masquerading as legitimate processes or tampering with system communication mechanisms.
- Several Volatility plugins are specifically designed for investigating process objects:
  - dlllist: Lists the DLLs loaded by each process, which helps identify malicious injections or hidden functionalities.

    - cmdline: Displays the command-line arguments used to launch a process, providing context for its purpose or actions.
    - getsids: Prints the Security Identifiers (SIDs) associated with each process, revealing permissions and access levels.
    - handles: Lists open handles for a process, showing the resources it is accessing. This can help detect unauthorized access or suspicious resource usage.
    - mutantscan: Scans memory for mutant objects (KMUTANT), focusing on mutex handles. This is particularly useful for identifying malware attempting to manage synchronization or avoid detection.

3.3.3.3 Review Network Artifacts

Identifying suspicious network connections is a critical aspect of memory analysis. This process focuses on detecting anomalies in ports, connections, and processes associated with network activity.

- Suspicious Ports
    - Ports used for abnormal or unauthorized communication.
    - Ports known to be associated with backdoors.
    - Unexpected listening ports that deviate from normal configurations.
- Suspicious Connections
    - External connections to IPs with known bad reputations.
    - Connections using TCP/UDP protocols with unusual creation times.
- Examples of suspicious connections include:
    - Processes communicating over common web ports (80, 443, 8080) that are not web browsers.
    - Connections to unexpected internal or external IP addresses.
    - Web requests bypassing domain names and directly targeting IP addresses.
    - Remote Desktop Protocol (RDP) traffic (port 3389) originating from unusual IPs.
    - Unexpected workstation-to-workstation traffic within the network.
- Suspicious Processes
    - Processes that exhibit network capabilities, such as open sockets, when they typically should not have such capabilities.
- Plugins for Network Artifact Analysis in Volatility
- Volatility offers several plugins to assist in identifying and analyzing network-related anomalies within memory dumps:
    - connections: Lists the network connections that were active at the time the memory snapshot was taken.
    - connscan: Scans for network connection objects in memory. This plugin can identify connection artifacts that might not be visible using the standard connections plugin.
    - sockets: Displays socket connections retrieved from the memory image.

    - sockscan: Scans for socket objects in memory, potentially uncovering hidden or unlinked sockets caused by rootkits.
    - netscan: A comprehensive plugin that scans for all network-related artifacts, including connections, sockets, and other network objects.

3.3.3.4 Check For Signs Of Code Injection

- Malware frequently uses code injection techniques to remain hidden, which can often be detected through memory analysis. Three primary code injection methods include DLL Injection, Process Hollowing, and Reflective DLL Injection:
    - DLL Injection
        - Involves injecting a Dynamic Link Library (DLL) into the memory space of another process.
        - Once injected, the malicious DLL executes within the context of the target process.
        - The process typically involves:
            - OpenProcess(): Attaches to the target process.
            - VirtualAllocEx(): Allocates memory in the target process.
            - WriteProcessMemory(): Writes the malicious DLL path into the allocated memory.
            - CreateRemoteThread(): Creates a thread in the target process to execute the DLL.
            - LoadLibrary(): Loads the malicious DLL into memory.
    - Process Hollowing
        - A stealthier method where an attacker creates a legitimate process in a suspended state, replaces its memory image with malicious code, and then resumes execution.
        - The process often looks legitimate (e.g., notepad.exe), but it executes the attacker's malicious payload.
        - Steps include:
            - Starting a legitimate process in a suspended state.
            - Removing the legitimate code from memory.
            - Writing malicious code in its place.
            - Resuming the process, now executing the malicious code.
    - Reflective DLL Injection
        - A sophisticated technique allowing a DLL to load and execute in the memory space of a target process without using standard Windows API calls.
        - The injected DLL contains a custom loader that prepares it for execution independently, bypassing the operating system's DLL loader.
        - This self-contained approach avoids detection by standard memory and API monitoring techniques.

- Volatility, a popular memory forensic framework, provides specific plugins to detect code injection and other memory anomalies:
    - ldrmodules: Compares three module lists maintained by the Windows Process Environment Block (PEB):
            - InLoadOrderModuleList: Tracks modules by load order.
            - InMemoryOrderModuleList: Tracks modules by memory address order.
            - InInitializationOrderModuleList: Tracks modules by initialization order.
        - Discrepancies between these lists may indicate hidden or unlinked modules, a common evasion technique used by malware.
        - Legitimate DLLs will appear in all three lists, while suspicious modules may only appear in some.
    - malfind: Scans the virtual address space of processes to identify memory regions with suspicious characteristics. Specifically detects pages with write and execute permissions, which are uncommon in legitimate software due to security practices like Data Execution Prevention (DEP).
    - hollowfind: Designed to detect process hollowing by identifying mismatches between the process image and its actual behavior.
    - Threadmap: Maps and analyzes threads within processes to detect anomalies, such as threads executing in unexpected memory regions or linked to injected code.D

3.3.3.5 Check For Signs Of Rootkit

Rootkits are malicious software tools designed to hide the presence of malware and alter system behavior to maintain unauthorized control. They commonly use various techniques to hook into critical system structures, including:

- System Service Descriptor Table (SSDT)
    - The SSDT is a critical structure in Windows that stores the addresses of system service routines, which are used by the kernel for system calls.
    - Rootkits hook into this table to intercept and manipulate system calls, enabling them to hide or alter their actions.
    - The ssdt Volatility plugin scans the memory dump for hooked functions in the SSDT, displaying the original and current addresses of each function. This helps to identify malicious modifications, as any function whose address doesn’t point to the expected module could be suspect.
    - It’s common to use a filter like egrep -v '(ntos|win32k)' to eliminate false positives and focus on the relevant hooks.
- Interrupt Descriptor Table (IDT)

The IDT is another table used by the operating system to handle interrupts. By hooking into the IDT, malware can redirect interrupts to malicious code, allowing it to execute in response to system events.

  - Import Address Table (IAT) and Inline API Hooks
    The IAT is a table used by programs to link to external functions, and rootkits can hook this table to intercept API calls. Inline API hooks involve directly modifying the code of a program to alter its behavior, often by replacing function addresses with addresses to malicious code.
  - I/O Request Packets (IRP)
    IRPs are used by Windows to communicate between device drivers and the kernel. Rootkits can hook IRPs to control or monitor interactions with hardware and drivers.

- Volatility provides several plugins that assist in detecting rootkits by identifying modifications to critical system structures or behavior. Some of these plugins include:
  - Ssdt: This plugin scans the memory dump for hooks in the System Service Descriptor Table. It compares the original addresses of system calls with their current addresses and identifies any modifications. It provides useful outputs that list services by name, index, and address in the SSDT, along with the module providing the service. Discrepancies may point to suspicious activities. To filter false positives, use commands like egrep -v '(ntos|win32k)' to exclude legitimate processes.
  - Psxview: This plugin helps identify hidden or malicious processes by comparing process lists from different sources within the operating system. Malware may hide its presence by removing itself from standard process listings, but it might still appear in other process lists. psxview uses various tools like pslist, psscan, thrdproc, pspcid, csrss, session, and deskthrd to gather a comprehensive view of running processes. The -R argument can be used to eliminate false positives from known exceptions (e.g., processes like smss and csrss).
  - modscan and modules : These plugins scan memory for loaded kernel modules.
  - modscan identifies kernel modules based on specific signatures, providing the names, paths, and sizes of modules. modules offers a broad overview of the loaded modules.
  - By using driverbl, you can compare the modules from a clean system with those from a suspicious system, identifying hidden or injected modules.
  - Apihooks : This plugin scans memory for API hooks, including both user-mode and kernel-mode hooks. It checks common hooking locations, such as the SSDT, Inline hooks, IAT, and EAT (Export Address Table) hooks. API hooks are commonly used by malware to intercept or alter the behavior of legitimate functions.
  - Driverip : This plugin scans for and identifies malicious drivers that might have been loaded into the system. These drivers are often used to manipulate or monitor system activity in a rootkit attack.

  - Idt : The Interrupt Descriptor Table (IDT) plugin scans for any hooks or modifications to the IDT, which malware can exploit to intercept hardware interrupts.

3.3.3.6 Dump Suspicious Process And Drivers

3.3.4 Code injection, Rootkits and extraction

3.4 code injection

3.5 Hooking and rootkit detection

- Rootkit hooking
  - System service descriptor table (SSDT)
  - Interrupt descriptor table (IDT)
  - Import address table (IAT) and inline API
  - I/O request packets (IRP)
- Volatility rootkit detection plugins:
  - Ssdt
    - is used to display the Service Descriptor Table hooks within a system. The System Service Descriptor Table is a critical data structure in Windows operating systems that stores the addresses of the system service routines. These routines are functions used by Windows kernel and system calls made by Windows programs. Malware and rootkits often hook into the SSDT to intercept system calls for hiding themselves, spying, or altering the behavior of the operating system.
    -it scans the memory dump for the SSDT and lists the functions that are hooked, along with their original and current addresses. This can help identify potentially malicious modifications to the system call table
    The output will list each service in the SSDT, including the service name, index, address in the SSDT, and the module that provides this service. If a service's address does not point to the expected module.
    -mostly we used egrep -v '(ntos|win32k)' to eliminate the false positive.

  - Psxview
    - is designed to identify potentially hidden or malicious processes by comparing process listings obtained through different mechanisms within the Windows operating system
    - malware might attempt to hide its presence by removing itself from certain process lists that are commonly queried by system administrators and security tools. However, it might not be able to remove itself from all

    the different listings due to the way the operating system manages these lists.
    - The psxview plugin uses multiple internal tools like pslist, psscan, thrdproc, pspcid, csrss, session, and deskthrd
    - It is recommended to use -R argument to eliminate the false positive since we have many exeption such as the process that start on boot cycle smss,csrss will nor shown in csrss column, also any process start before smss will not appear on session and deskthrd column.

  - Modscan & modules
    - it searches for loaded kernel modules by scanning the memory for specific signatures or patterns that match those of kernel modules.
    - Provide me the name,path and size of drive
    - We can use then driverbl command to check the defference between clean drive and suspicious one
    - it's common to start with a broad analysis using commands like modules to get an overview, and then delve deeper with commands like modscan to uncover more details and potentially hidden elements within the memory image.
  - Apihooks
    - scan a memory image for API hooks. This plugin scans for both user-mode and kernel-mode hooks, looking at common locations where hooks might be installed, such as the System Service Descriptor Table (SSDT), Inline hooks, IAT (Import Address Table), and EAT (Export Address Table) hooks.
  - Driverip
  - Idt

5. Extracting Processes, Drives, and Objects

The process of extracting suspicious objects from memory dumps is a crucial step in forensic investigations. Volatility, a powerful memory analysis tool, provides various plugins to facilitate the extraction of processes, kernel modules, and objects for further analysis.

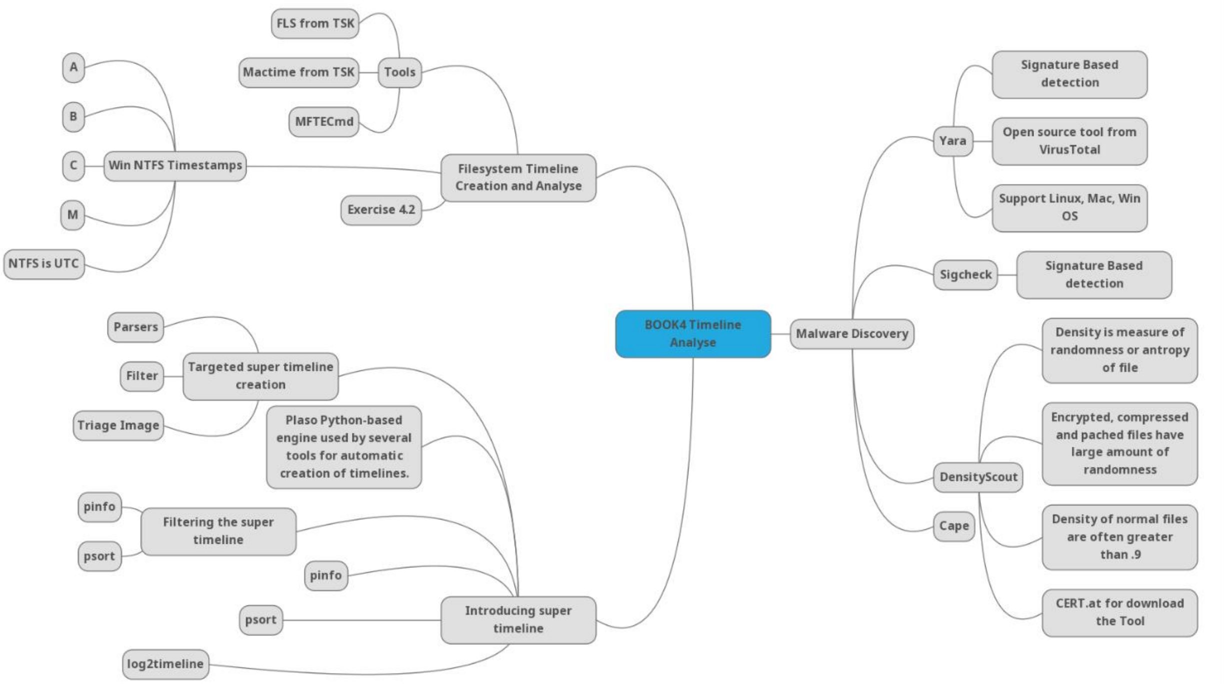
- Volatility extraction plugins :
  - dlllist: This plugin lists all loaded DLLs (Dynamic Link Libraries) for a process or system memory dump.
  - dlldump: Similar to dlllist, but also allows for the dumping of DLLs from memory. It's useful for extracting DLLs to analyze potentially malicious modules loaded by processes. Parameters for dlldump:

    - -p or --pid: Target a specific Process ID (PID) to dump DLLs from that process.
    - -D or --dump-dir: Specify the output directory to store the extracted DLL files.
    - -b or --base: Define the base address of a DLL to target a particular DLL for extraction.
    - --regex: Use a regular expression to filter and dump DLLs that match a specific pattern.
  - Moddump : The moddump plugin extracts loaded kernel modules (drivers) from memory dumps. This is particularly useful when investigating rootkits or suspicious kernel activity. Parameters for moddump:
    - -D or --dump-dir: Define the directory to store the extracted modules.
    - -b or --base: (Optional) Specify the base address of a kernel module for targeted dumping.
    - -r or --regex: (Optional) Use regular expressions to filter and extract modules matching specific name patterns.
  - Procdump : The procdump plugin is designed to extract process images from memory dumps. It's useful for analyzing suspicious or malicious processes by dumping their memory contents. Parameters for procdump:
    - -p or --pid: Target a specific process by its PID.
    - -D or --dump-dir: Define the output directory for storing the dumped process images.
    - -o: (Optional) Specify the offset for dumping hidden or unlinked processes.
    - -n or --regex: Use regular expressions to filter and dump processes matching a pattern.
  - Memdump : The memdump plugin extracts the entire memory space of a specific process from the memory dump. This includes not just the executable code but also loaded libraries, stack, and heap data, providing a comprehensive view of the process's memory. Parameters for memdump:
    - -p or --pid: Target a specific process by its PID.
    - -D or --dump-dir: Define the output directory for the dumped memory files.
    - After extracting memory data, tools like strings and grep can be used to analyze the contents and extract useful information.
  - Cmdscan: The cmdscan plugin scans for command history from CMD.exe instances in memory dumps. It displays the commands that were executed but not the output of these commands.

- Consoles: The consoles plugin is similar to cmdscan, but broader, as it captures interactions from various command-line interfaces (CLI), including PowerShell. This includes both the commands entered and the output displayed within the console.
- Dumpfiles: The dumpfiles plugin allows for the extraction of different types of files from memory dumps, including executable files (EXEs, DLLs), documents, temporary files, logs, and even hidden files injected by malware.
- Filescan:The filescan plugin scans memory dumps to locate FILE_OBJECTs, which represent open files in the Windows kernel. This tool is particularly useful for identifying hidden or unlinked files that may be used by malware or have been removed from standard file systems.
- Shimcachemem: The shimcache or AppCompatCache in Windows is a component of the Application Compatibility Database, which is used by the operating system to apply compatibility fixes to applications that were developed for previous versions of Windows. The Shim Cache logs basic information about applications that have been executed on a system, making it a valuable artifact for forensic investigators to determine which programs have been run on a computer.

Timeline Analyze

Learn advanced incident response and hunting techniques uncovered via timeline analysis directly from the authors who pioneered timeline analysis tradecraft. Temporal data is located everywhere on a computer system. Filesystem modified/access/creation/change times, log files, network data, registry data, and browser history files all contain time data that can be correlated and analyzed to rapidly solve cases. Pioneered by Rob Lee as early as 2001, timeline analysis has grown to become a critical incident response, hunting, and forensics technique. New timeline analysis frameworks provide the means to conduct simultaneous examinations on a multitude of systems across a multitude of forensic artifacts. Analysis that once took days now takes minutes. This section will step you through two primary methods of building and analyzing timelines used during advanced incident response, threat hunting, and forensic cases. Exercises will show analysts how to create timelines and how to introduce the key analysis methods necessary to help you use those timelines effectively in your cases. Refer to


BOOK4 Timeline Analyse
Filesystem Timeline Creation and Analyse
Tools
FLS from TSK
Mactime from TSK
MFTECmd
Win NTFS Timestamps
A
B
C
M
NTFS is UTC
Exercise 4.2
Introducing super timeline
Targeted super timeline creation
Parsers
Filter
Triage Image
Plaso Python-based engine used by several tools for automatic creation of timelines.
Filtering the super timeline
pinfo
psort
pinfo
psort
log2timeline
Malware Discovery
Yara
Signature Based detection
Open source tool from VirusTotal
Support Linux, Mac, Win OS
Sigcheck
Signature Based detection
DensityScout
Density is measure of randomness or antropy of file
Encrypted, compressed and pached files have large amount of randomness
Density of normal files are often greater than .9
CERT.at for download the Tool
Cape

4.1 Malware Discovery

Several tools and techniques used for identifying and analyzing anomalies that could indicate the presence of malware:

- YARA: Used to search for strings and header-based signatures, standardize IOC (Indicator of Compromise) sharing, and detect new malware.
- Sigcheck: A Microsoft Sysinternals tool, focusing on digital signature checking. Sigcheck is used to verify digital signatures and dump version information of executable files within a directory. It provides various command-line options for different outputs like CSV, scanning executable images only, showing file hashes, recursing subdirectories, and checking against VirusTotal.
- DensityScout: It is used for finding suspicious files by identifying those with uncommon levels of density, which is related to the concept of entropy—a measure of randomness within a file. High entropy is often associated with files that are packed or encrypted, which can make them stand out from ordinary files.
- capa: Analyzes files for capabilities, it is open-source project from the FireEye FLARE team. The project, capa, aims to streamline the initial phases of malware investigations by using rules that help in detecting triage using crowdsourced code patterns. The rules are designed to identify common malware actions like communication, host interaction, persistence, and anti-analysis tactics, and also include mappings to the MITRE ATT&CK framework and Malware Behaivoral Catalog MBC

In conclusion, files that have lacking digital signatures and exhibiting anomalous behavior, low-density scores and high VT detection rates, indicating they are likely not legitimate software and could pose a threat.

4.2 Timeline analysis

4.2.1 Timeline analysis overviews

- Timeline analysis is a highly effective technique in digital forensics and incident response, streamlining the investigative process by organizing and visualizing data from various system artifacts in a chronological sequence. This method allows investigators to reconstruct events that took place on a system over a specific time frame, helping to build a coherent narrative of the incident.
- Timeline analysis is not without its challenges. One of the primary concerns is that some individuals may distrust timeline data due to the possibility of anti-forensic techniques being used to manipulate or hide certain artifacts, potentially skewing the results. To effectively conduct timeline analysis, a deep understanding of the following areas is essential:
    - Filesystem Metadata:

    The filesystem contains crucial metadata (timestamps, file attributes, etc.) that can provide insight into when files were created, modified, or accessed. Understanding this metadata is vital for establishing a timeline of file activity and identifying anomalies.

  - Windows Artifact Data:
    Windows artifacts, such as event logs, prefetch files, and jump lists, hold valuable data about user activities and system events. These artifacts can reveal actions like program execution, system reboots, or user logins, contributing to the timeline.
  - Windows Registry Information:
    The Windows registry stores configuration and system information, such as installed programs, recently accessed files, and system settings. Analyzing registry hives can uncover important clues about system behavior and user actions, further aiding in timeline reconstruction.

- Process for Conducting Timeline Analysis in Digital Forensics
  - Determine Timeline Scope: Identify the key questions to answer and establish the timeframe of the incident under investigation.
  - Narrow Pivot Points: Focus on specific timeframes or artifact-related aspects that are directly relevant to the incident.
  - Select the Best Timeline Creation Method: Decide on the appropriate timeline creation approach
    - Super Timeline (log2timeline): For detailed, comprehensive analysis.
    - Filesystem-based Timeline (fls or MFTCmd): For more focused, filesystem-level analysis.
  - Filter the Timeline: Eliminate irrelevant data to focus solely on the pivot points and critical artifacts.
  - Analyze the Timeline: Examine the context and sequence of events in the timeline, utilizing tools like the Windows Forensic Analysis Poster for additional guidance and reference.
- Comparison of Timeline Creation Approaches

	Filesystem (fls or MFTCmd)	**Super Timeline (log2timeline)**
Data Captured	Filesystem metadata only	Everything (filesystem metadata, artifact timestamps, registry timestamps)
Filesystem Types	Many (HFS, UFS, EXT, FAT/NTFS, CD-ROM)	Windows, Linux, and Mac
OS Compatibility	Wider (Includes various OS)	Specifically Windows, Linux, and Mac
Use Case	Rapid analysis, greater filesystem flexibility	Detailed, comprehensive analysis

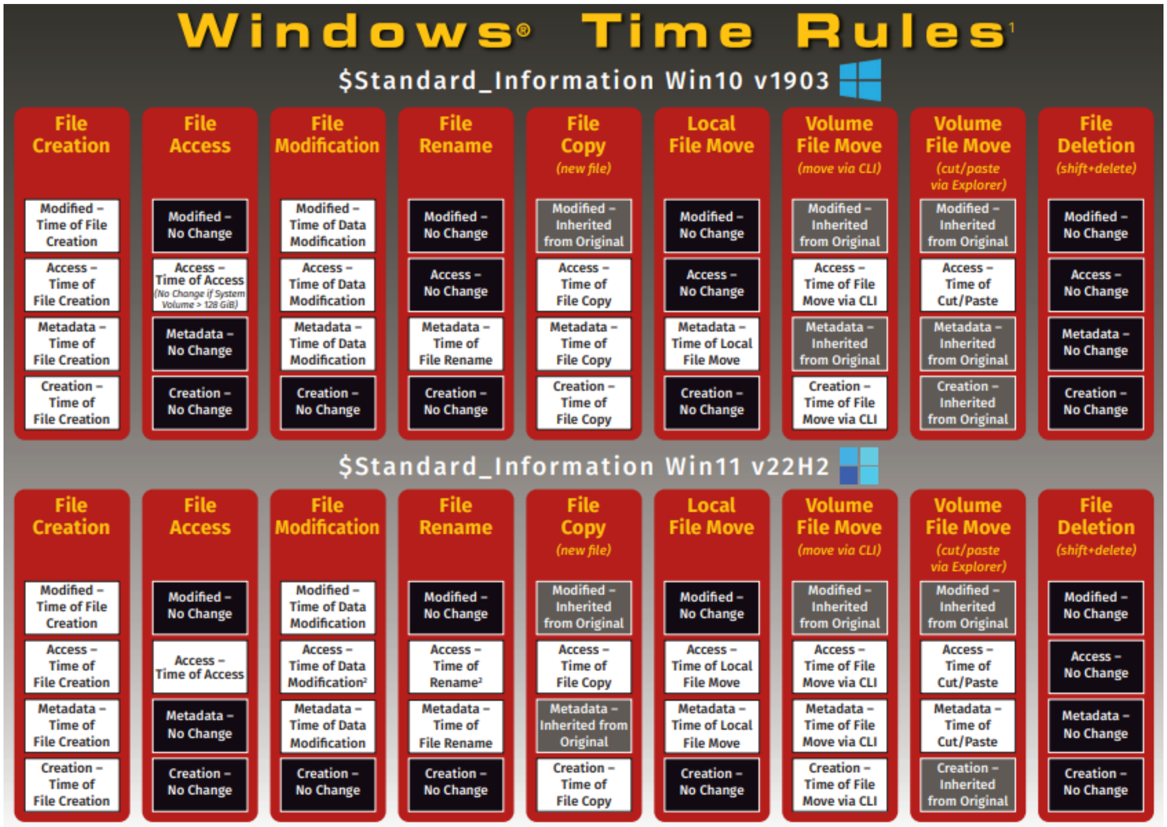
- A pivot point is a critical concept in digital forensics, marking the starting point of an investigation based on key events identified in a timeline. It allows a forensic analyst to focus their investigation on a specific event or series of events that seem to be central to the case. By examining the temporal proximity of events before and after the pivot point, analysts gain a clearer understanding of what happened on the system. This method is vital for identifying patterns, such as program executions, system interactions, or suspicious behaviors that may indicate malicious activity.

- Determining the pivot point involves focusing on several key areas:
    - Incident Time: Using alerts from Security Information and Event Management (SIEM) systems, Intrusion Detection Systems (IDS), or Antivirus (AV) alerts to determine when the suspicious activity started.
    - Network Activity: Identifying malicious URLs or DNS queries, which can point to external threats or attempts to establish communication with a command-and-control server.
    - Process Activity: Examining running processes, including potential DLL injections or other abnormal processes, which can highlight the presence of malware or malicious activity.
    - File Name/Type: Focusing on executables, scripts, compressed files, or other file types that are commonly used in malware deployment.
    - User Account Activity: Investigating suspicious actions or account behavior that may point to insider threats or unauthorized access.
    - Lateral Movement: Analyzing event logs, file operations, and network connections to detect movement within the system or across systems.
    - Anti-Forensics: Identifying tools or techniques like wiper tools, which attempt to erase traces of malicious activity.
- Understanding the temporal context is essential in artifact analysis within digital forensics. The meaning of a single artifact can change significantly when analyzed alongside other related data, much like how the meaning of a word such as "sweet" varies depending on the context of its use within a sentence. Contextual analysis helps investigators uncover the relationships between artifacts and provides a more accurate reconstruction of events.

4.2.2 Filesystem timeline creation and analysis

- Filesystem timeline analysis reconstructs activity on a system by examining metadata from files and directories, including timestamps for modifications, access, changes, and creation. NTFS, a widely used Windows filesystem, provides four key timestamps:
    - M (Modification Time): Last content modification.
    - A (Access Time): Last file access.
    - C (MFT Change Time): Metadata changes.
    - B (Creation Time): File or directory creation.
- These timestamps, stored in UTC, ensure reliability across time zones, unlike FAT filesystems. The M and B timestamps are particularly valuable for identifying file changes and origins.

- Filesystem timelines are versatile and, when combined with other evidence, help investigators identify tampering, unauthorized access, or anti-forensic activities, making them a vital tool in digital forensics.



Windows® Time Rules[1]

$Standard_Information Win10 v1903

File Creation	File Access	File Modification	File Rename	File Copy (new file)	Local File Move	Volume File Move (move via CLI)	Volume File Move (cut/paste via Explorer)	File Deletion (shift+delete)
Modified – Time of File Creation	Modified – No Change	Modified – Time of Data Modification	Modified – No Change	Modified – Inherited from Original	Modified – No Change	Modified – Inherited from Original	Modified – Inherited from Original	Modified – No Change
Access – Time of File Creation	Access – Time of Access (No Change if System Volume > 128 GiB)	Access – Time of Data Modification	Access – No Change	Access – Time of File Copy	Access – No Change	Access – Time of File Move via CLI	Access – Time of Cut/Paste	Access – No Change
Metadata – Time of File Creation	Metadata – No Change	Metadata – Time of Data Modification	Metadata – Time of File Rename	Metadata – Time of File Copy	Metadata – Time of Local File Move	Metadata – Inherited from Original	Metadata – Inherited from Original	Metadata – No Change
Creation – Time of File Creation	Creation – No Change	Creation – No Change	Creation – No Change	Creation – Time of File Copy	Creation – No Change	Creation – Time of File Move via CLI	Creation – Inherited from Original	Creation – No Change

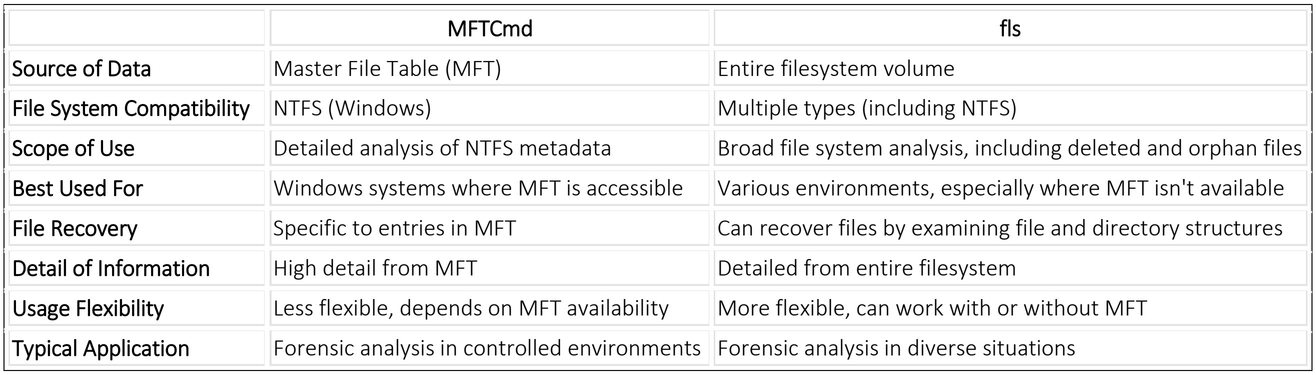
$Standard_Information Win11 v22H2

File Creation	File Access	File Modification	File Rename	File Copy (new file)	Local File Move	Volume File Move (move via CLI)	Volume File Move (cut/paste via Explorer)	File Deletion (shift+delete)
Modified – Time of File Creation	Modified – No Change	Modified – Time of Data Modification	Modified – No Change	Modified – Inherited from Original	Modified – No Change	Modified – Inherited from Original	Modified – Inherited from Original	Modified – No Change
Access – Time of File Creation	Access – Time of Access	Access – Time of Data Modification[2]	Access – Time of Rename[2]	Access – Time of File Copy	Access – Time of Local File Move	Access – Time of File Move via CLI	Access – Time of Cut/Paste	Access – No Change
Metadata – Time of File Creation	Metadata – No Change	Metadata – Time of Data Modification	Metadata – Time of File Rename	Metadata – Inherited from Original	Metadata – Time of Local File Move	Metadata – Time of File Move via CLI	Metadata – Time of Cut/Paste	Metadata – No Change
Creation – Time of File Creation	Creation – No Change	Creation – No Change	Creation – No Change	Creation – Time of File Copy	Creation – No Change	Creation – Time of File Move via CLI	Creation – Inherited from Original	Creation – No Change

- Certain factors can alter file timestamps, complicating digital forensic investigations forensic analysts must account for these exceptions and conduct thorough testing to ensure accurate conclusions:
  - Applications & Tools: Programs like Office or WinZip can modify timestamps.
  - Anti-Forensic Methods: Tools like Timestomp and privacy cleaners manipulate timestamps to obscure activity.
  - Archived Files: Formats like ZIP or RAR retain original modification times, often ignoring creation timestamps.
  - Antivirus Scans: These can impact timestamps based on the software's design.
- When files are transferred via SMB (Server Message Block) using NTFS, the original modification time is preserved, while the creation time reflects the transfer timestamp. This distinction is key for tracking lateral movements.
- Creating a Filesystem Timeline

  1. Generate a Triage Timeline Bodyfile: Use tools like MFTCmd or fls from The Sleuth Kit (TSK).

- MFTCmd: Developed by Eric Zimmerman, it extracts NTFS metadata (e.g., $MFT) and converts it into JSON, CSV, or bodyfile formats for detailed analysis.
- fls (TSK): Extracts metadata from the entire filesystem volume, including active, deleted, and orphan files. It creates a bodyfile timeline and is particularly useful when no disk image is available, offering a broader analysis compared to MFTCmd.



	MFTCmd	fls
Source of Data	Master File Table (MFT)	Entire filesystem volume
File System Compatibility	NTFS (Windows)	Multiple types (including NTFS)
Scope of Use	Detailed analysis of NTFS metadata	Broad file system analysis, including deleted and orphan files
Best Used For	Windows systems where MFT is accessible	Various environments, especially where MFT isn't available
File Recovery	Specific to entries in MFT	Can recover files by examining file and directory structures
Detail of Information	High detail from MFT	Detailed from entire filesystem
Usage Flexibility	Less flexible, depends on MFT availability	More flexible, can work with or without MFT
Typical Application	Forensic analysis in controlled environments	Forensic analysis in diverse situations

2. Processing the Bodyfile: The bodyfile format produced by MFTCmd or fls is then processed using the mactime tool from The Sleuth Kit. This tool converts the bodyfile data into a human-readable, chronological timeline that illustrates file creation, access, modification, and deletion events.

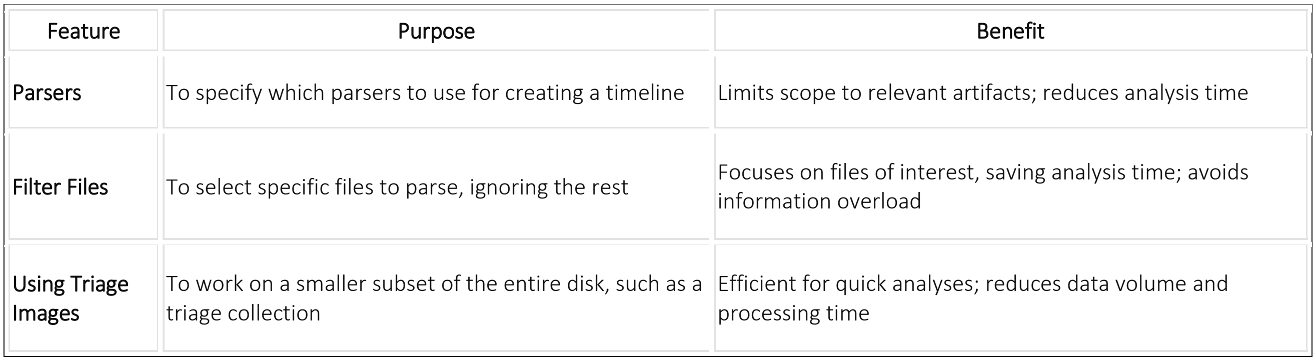
4.2.3 Introduction the super timeline

- log2timeline is a machine-oriented frontend tool for Plaso, developed by Kristinn Gudjonsson. Plaso is the standard tool for extracting and consolidating time-based events from multiple file sources across different operating systems such as Mac, Linux, and Windows. This combination allows investigators to compile events into a unified timeline, which simplifies reference and analysis of these events during forensic investigations.
- Key Components of Plaso:
  - log2timeline: A command-line tool that extracts events from various sources. It acts as the primary frontend to Plaso, transforming raw data from different files into a coherent timeline format.
  - pinfo: A tool that provides metadata about the Plaso storage file, including details about preprocessing stages and storage information.
  - psort: A post-processing tool designed to filter, sort, and process Plaso storage files. It helps make the storage files more readable and suitable for further investigation.
- Plaso operates as a backend parser, meaning it requires a frontend tool (like log2timeline.py) to function effectively. The log2timeline tool processes data from files within directories, mounted devices, forensic images, and virtual disk images. It automatically recurses through all subdirectories to gather relevant event information.

- Once the data is extracted, pinfo is used to display metadata about the Plaso storage, and psort is employed to process and filter this storage file, preparing it for further analysis and making it usable for forensic examination.
- The combination of these tools ensures that investigators can handle complex datasets from multiple sources and operating systems efficiently, facilitating the creation of a super timeline that can provide critical insights into the sequence of events during an investigation.

4.2.4 Targeted super timeline creation

- There are two different approaches to timeline analysis in digital forensics using tools like log2timeline.py:
  - Kitchen Sink Approach: This method involves running log2timeline.py against a disk image to extract all timestamps and artifacts it supports. It's comprehensive and can extract every item possible, which aligns with what some people consider a "super timeline." However, due to its broad scope, it can produce overwhelmingly large timelines.
  - Targeted Acquisition: This method is more selective, gathering data only from files relevant to the investigation, which reduces the amount of data and the time required to analyze it. log2timeline.py has features that support this approach by allowing users to target more specific data sets, typically taking between 5-30 minutes to complete. The downside is the risk of missing important artifacts, but the user can later add artifacts as needed, which is a smaller risk once you have a clearer understanding of the case.
- Parsers: log2timeline.py allows for specifying a list of parsers (PARSER_LIST) to limit the scope of the analysis to only artifacts relevant to the chosen parsers, enhancing the targeted acquisition method. Parsers can be defined in the command line using the --parsers option or through a list file. A list file is a text file with a comma-separated list where each entry can be either a parser name or another parser list.
- log2timeline.py now includes multiple features to facilitate creating these more focused, targeted timelines.



Feature	Purpose	Benefit
Parsers	To specify which parsers to use for creating a timeline	Limits scope to relevant artifacts; reduces analysis time
Filter Files	To select specific files to parse, ignoring the rest	Focuses on files of interest, saving analysis time; avoids information overload
Using Triage Images	To work on a smaller subset of the entire disk, such as a triage collection	Efficient for quick analyses; reduces data volume and processing time

- The process of creating timelines for digital forensics investigations. It suggests that instead of using traditional disk-based forensics, which requires imaging the entire hard drive, one can use triage tools like KAPE to selectively collect files critical to the investigation. Plaso/log2timeline can then be used to create a timeline from these files. This approach is deemed faster as it avoids full hard drive acquisition and can be integrated into existing processes. Triage images can be rapidly collected and processed, and the authors believe that after using this method a few times, one may find little reason to revert to full disk imaging as the standard operating procedure.

4.2.5 Filter the super timeline

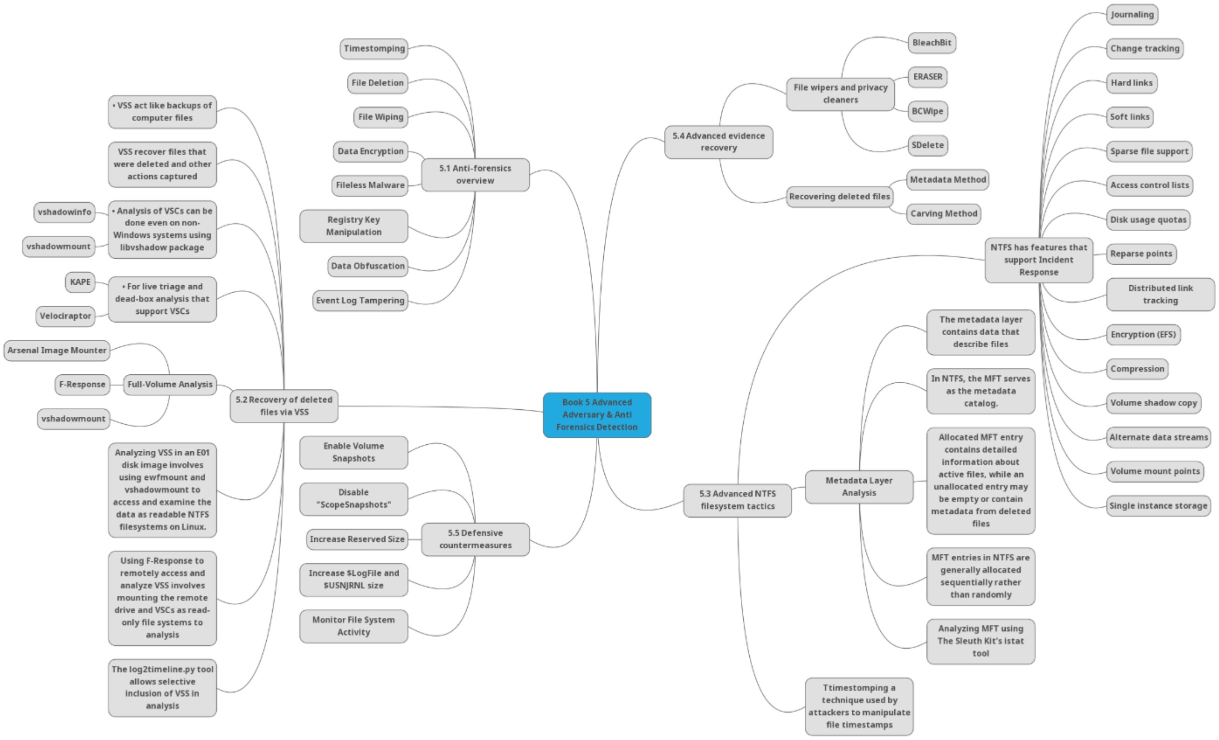
- pinfo.py: A command-line tool to inspect and validate Plaso forensic databases. It provides detailed metadata about tool execution, parsers, filters, preprocessing data, and parsed artifacts. It ensures accurate validation of the parsing process and highlights errors or irregularities.
- psort: Processes Plaso's raw SQLite data into human-readable timelines. It allows filtering, duplicate removal, timezone adjustments, and mini-timeline creation to focus on relevant artifacts.

4.2.6 super timeline analysis

The Super Timeline format is a detailed chronological record of digital events commonly used in digital forensics investigations. It contains numerous columns, including date, time, user, host, and event type, providing insights into activities like file creation, modification, deletion, and user access. To optimize screen space, it's recommended to start with only the essential columns. As analysis deepens, additional columns such as "user," "host," and "inode" can be included for further investigation. The "extra" column may contain additional information parsed by the plugin.

Advanced Adversary and Anti Forensics Detection

Attackers commonly take steps to hide their presence on compromised systems. While some anti-forensics steps can be relatively easy to detect, others are much harder to deal with. As such, it’s important that forensic professionals and incident responders are knowledgeable on various aspects of the operating system and file system which can reveal critical residual evidence. Criminal and ransomware syndicates have become particularly aggressive in their use of anti-forensic techniques. In this section, we focus on recovering files, file fragments, and file metadata of interest to the investigation. These trace artifacts can help the analyst uncover deleted logs, attacker tools, malware configuration information, exfiltrated data, and more. This often results in a deeper understanding of the attacker TTPs and provides more threat intelligence for rapid scoping of an intrusion and mitigating damage. In some cases, these deepdive techniques could be the only means for proving that an attacker was active on a system of interest and ultimately determining root cause. While very germane to intrusion cases, these techniques are applicable in nearly every forensic investigation. Refer to


Book 5 Advanced Adversary & Anti Forensics Detection
5.1 Anti-forensics overview
Timestomping
File Deletion
File Wiping
Data Encryption
Fileless Malware
Registry Key Manipulation
Data Obfuscation
Event Log Tampering
5.2 Recovery of deleted files via VSS
• VSS act like backups of computer files
VSS recover files that were deleted and other actions captured
• Analysis of VSCs can be done even on non-Windows systems using libvshadow package
vshadowinfo
vshadowmount
• For live triage and dead-box analysis that support VSCs
KAPE
Velociraptor
Full-Volume Analysis
Arsenal Image Mounter
F-Response
vshadowmount
Analyzing VSS in an E01 disk image involves using ewfmount and vshadowmount to access and examine the data as readable NTFS filesystems on Linux.
Using F-Response to remotely access and analyze VSS involves mounting the remote drive and VSCs as read-only file systems to analysis
The log2timeline.py tool allows selective inclusion of VSS in analysis
5.5 Defensive countermeasures
Enable Volume Snapshots
Disable "ScopeSnapshots"
Increase Reserved Size
Increase $LogFile and $USNJRNL size
Monitor File System Activity
5.4 Advanced evidence recovery
File wipers and privacy cleaners
BleachBit
ERASER
BCWipe
SDelete
Recovering deleted files
Metadata Method
Carving Method
5.3 Advanced NTFS filesystem tactics
NTFS has features that support Incident Response
Journaling
Change tracking
Hard links
Soft links
Sparse file support
Access control lists
Disk usage quotas
Reparse points
Distributed link tracking
Encryption (EFS)
Compression
Volume shadow copy
Alternate data streams
Volume mount points
Single instance storage
Metadata Layer Analysis
The metadata layer contains data that describe files
In NTFS, the MFT serves as the metadata catalog.
Allocated MFT entry contains detailed information about active files, while an unallocated entry may be empty or contain metadata from deleted files
MFT entries in NTFS are generally allocated sequentially rather than randomly
Analyzing MFT using The Sleuth Kit's istat tool
Ttimestomping a technique used by attackers to manipulate file timestamps

5.1 Anti-forensics overview

Anti-forensic techniques are strategies employed by attackers to hinder detection and complicate forensic investigations. Key methods include:

- Timestomping: Alters file timestamps to disguise the actual creation or modification date, blending malicious files with legitimate ones.
- File Deletion/Wiping: Erases evidence, often using third-party tools that prevent file recovery.
- Data Encryption: Encrypts stolen data (e.g., with RAR format) to prevent access without a complex password.
- Fileless Malware: Operates directly in memory or utilizes legitimate tools (e.g., PowerShell) to avoid detection by traditional methods.
- Registry Manipulation:
    - Key/Value Deletion/Wiping: Removes or wipes registry entries, although recovery may be possible due to Windows registry backups.
    - Hiding Data: Stores obfuscated or encrypted data in the registry (e.g., fileless malware scripts), often noticeable by the unusually large size.
- Event Log Tampering:
    - Deletion: Clears event logs to erase traces of malicious activity.
    - Manipulation: Modifies logs or exploits centralized log management to evade detection.

5.2 Recovery of deleted files via VSS

Volume Shadow Copies (VSCs) serve as backups of computer files at specific points in time, akin to virtual machine snapshots. They are essential tools in forensic investigations due to their ability to recover deleted files and provide historical views of data.

- Functionality:
    - VSCs allow investigators to restore files such as event logs, registry information, and deleted files.
    - Snapshots preserve file states before modifications or deletions.
- Challenges:
    - ScopeSnapshots (Windows 8 and later): Limit data stored in snapshots, impacting forensic analysis.
    - Not all data is included in VSCs due to system configurations and limitations.
- Live Triage and Dead-Box Analysis:
    - KAPE (Kroll Artifact Parser and Extractor)
    - Velociraptor
- Full-Volume Forensic Image Analysis:

- Arsenal Image Mounter
- F-Response
- vshadowmount

- Analyzing Volume Shadow Copies in E01 disk image using the libvshadow project. The process involves mounting an E01 disk image as a raw image in a Linux system using the ewfmount tool, which is not part of the libvshadow project but developed by Joachim Metz. After mounting the image, which is displayed as a file named ewf1, vshadowmount is used to map the Volume Shadow Copies contained within the E01 image. These copies are then accessible as virtual directories, vss1 and vss2, which represent different Volume Shadow Copies. Finally, mount one of these shadow copies (vss2) as a readable NTFS filesystem in Linux, making the files and system information within the shadow copy available for examination.
- Using F-Response to remotely access and analyze Volume Shadow Copies (VSCs) on a networked system. Initially, the remote system's drive is mounted as a raw image file using F-Response commands, allowing for subsequent mounting of the device's VSCs with vshadowmount. Finally, a FOR loop is employed to systematically mount and make each snapshot accessible as a logical file system in a read-only state for detailed analysis, using standard mount options compatible with Windows shadow copies. This multi-step process enables forensic analysts to scrutinize the VSCs for evidence or artifacts that could be pertinent to their investigation.
- The log2timeline.py tool can identify and selectively include VSS during analysis, giving analysts the flexibility to work with none, one, several, or all snapshots. Additionally, psort, another tool in the Plaso suite, is designed to remove duplicate entries which commonly occur in VSS analysis. It provides statistics on the number of events processed and the count of events excluded due to deduplication.

5.3 Advanced NTFS filesystem tactics

- Advanced NTFS Features Supporting Incident Response (IR):
  NTFS was designed to address FAT's limitations with features that aid forensics:
  - Journaling: Logs changes for crash recovery.
  - Change tracking: Records file and directory modifications.
  - Access control lists (ACLs): Offer granular security management.
  - Encryption (EFS) & Compression: Secure and optimize storage.
  - Sparse file support & Reparse points: Efficient disk space management.
  - Distributed link tracking: Maintains links integrity across network changes.
- Metadata Layer Analysis: Master File Table (MFT)
  - The Master File Table (MFT) in NTFS acts like a card catalog, storing critical metadata for all files and directories, including details about content,

timestamps, and permissions, which is essential for efficient data management and digital forensics.

  - The structure and data types inside an MFT entry for NTFS. The MFT entries contain essential information such as timestamps, file info, security data, and cluster lists. For non-resident files, the data is stored in clusters on the volume rather than in the MFT entry itself. These entries are stored in a hidden file called $MFT, while clusters are tracked using another hidden file called $Bitmap. $Bitmap indicates which clusters are in use and helps maintain the file system's organization. Critical file details like directory name, timestamps, permissions, and pointers to data are stored in the MFT entry, ensuring efficient data management and retrieval in NTFS.
  - Allocated MFT entry contains filled-out metadata such as name, timestamps, and permissions, with pointers to clusters holding file contents, indicating active files.
  - In contrast, an Unallocated MFT entry may have incomplete or no metadata and often relates to deleted files, with clusters that may or may not still hold the deleted file’s data, as these clusters might have been reused. This distinction helps in understanding the state and organization of files within the file system.
  - MFT entries in NTFS are generally allocated sequentially rather than randomly. This means that as new files are created, they are assigned the next available MFT entry. By analyzing contiguous metadata values, investigators can identify files likely created in quick succession, even if they are in different directories. This pattern provides valuable insights during forensic investigations, as sequential MFT allocations reveal clustering behavior, which can indicate related file creation events.
  - MFT (Master File Table) entry for a file with resident data in NTFS, showing its structured components: the MFT header with allocation status, $STANDARD_INFORMATION with timestamps, $FILE_NAME with short and normal names, and $DATA containing the actual data or pointers to data clusters. Each entry is 1024 bytes, starting with the header and followed by these attributes, which are crucial for detailing and managing files within the NTFS file system.
- Analyzing file system metadata using The Sleuth Kit's istat tool. The Sleuth Kit is an open-source set of tools designed for disk-based forensic analysis
- By using istat and other tools from The Sleuth Kit, forensic analysts can gain a deeper understanding of the core components and attributes of NTFS and other file systems, enabling thorough and effective forensic analysis.
- The $DATA attribute in NTFS manages file data by storing small files directly in the MFT record (resident data) and using pointers for larger files (non-resident data).

NTFS allows multiple data streams per file, including unnamed primary streams and named Alternate Data Streams (ADS). Forensic analysis of $DATA attributes helps uncover how data is stored, detect hidden data, and understand file activities. Resident storage is efficient for small files, while non-resident storage handles large files but adds complexity. ADS can hide additional data, making them crucial in forensic investigations.

- The icat, a tool from The Sleuth Kit, to extract data from disk images using metadata addresses (inode numbers). The icat command can recover deleted files and display slack space, outputting data based on a specified metadata entry. For NTFS file systems, icat extracts the primary $DATA stream by default but can also extract from Alternate Data Streams (ADS) using a specific syntax. This tool is essential in forensic investigations for recovering and analyzing data from NTFS file systems, including metadata and alternate streams, to gather comprehensive forensic evidence.
- NTFS, filenames are stored in the MFT entry and directory data. While file wiping software usually does not remove directory entries, slack space within directories can still contain metadata like filenames and timestamps. This slack space can be crucial for forensic investigations, providing historical data about deleted files often overlooked by some forensic tools. This technique underscores the importance of examining all potential data storage areas, including slack space, to uncover valuable evidence.
- NTFS directory attributes, focusing on the $I30 index used to store filenames and metadata. The $I30 index consists of $INDEX_ROOT, which is required and always resident in the MFT, and $INDEX_ALLOCATION, which is used for larger directories and always non-resident, stored in clusters.
- NTFS uses B-tree indexing to efficiently manage directory entries, starting searches at the $INDEX_ROOT for quick narrowing of choices. If directories grow large, additional entries are stored in $INDEX_ALLOCATION clusters. This hierarchical structure allows fast, predictable file searches, crucial for performance and forensic analysis, by organizing entries in a balanced, searchable tree structure.
- The main difference between the $INDEX_ALLOCATION header and the $I30 index entry in NTFS is their purpose and content. The $INDEX_ALLOCATION header provides general information about the entire index, including the total number of entries and the allocated size for those entries, used primarily for organizing large directories that exceed the capacity of the $INDEX_ROOT. In contrast, the $I30 index entry contains specific metadata for each file or folder, such as the MFT entry number, creation and modification times, file size, flags, and file name. This entry is used within both small and large indexes to store detailed information about individual files and directories.
- Indx2Csv and Velociraptor are forensic tools for parsing $I30 directory indexes in NTFS. Indx2Csv is a free tool that parses active and slack entries from exported $I30

files, aiding in recovering deleted or partial entries. Velociraptor analyzes live file systems and mounted images, parsing entries recursively to uncover both current and residual data. These tools are essential for forensic analysts to recover deleted files, examine slack space, and reconstruct past activities

- File system journaling in NTFS utilizes two key components, $LogFile and $UsnJrnl, to record changes to file system metadata. $LogFile captures low-level transactional data to help restore the file system to a consistent state after errors, similar to a flight data recorder. $UsnJrnl logs higher-level file and directory changes, aiding applications like antivirus and backup software in monitoring and responding to these changes, much like a cockpit voice recorder. These journals serve as continuous logs, enabling forensic analysts to trace file states and changes over time, unlike point-in-time snapshots from Volume Shadow Copies (VSS), thus providing critical insights for both system recovery and forensic investigations.
- $LogFile in NTFS provides file system resilience by storing low-level transactional logs that ensure consistency. It tracks changes to critical NTFS features, including MFT records and directory indexes, and maintains detailed data about file system events. This log is efficient, recording only necessary changes to minimize storage use and typically retains data for a few hours on active systems, extending to days on less active ones. Forensic analysts benefit from $LogFile as it provides detailed information about recent file system activities, helping to reconstruct events and understand file modifications, crucial for investigations.
- $UsnJrnl in NTFS provides a change tracking service by recording high-level summary information about file and directory modifications. It helps applications like backup and antivirus software determine which files have been changed without scanning every file, thus enhancing efficiency. Typically, $UsnJrnl retains data for a few days on active systems but can last weeks on less active systems or secondary drives, storing this information in large alternate data streams (ADS). For forensic analysts, $UsnJrnl is invaluable for reconstructing recent file system activities, as it logs concise and consistent details about changes, including file creation, modifications, and deletions.
- Parent directory filtering in journal logs targets key directories like C:\Windows\System32 and C:\$Recycle.Bin to uncover malicious activities and hidden files. Searching for specific file types such as executables and scripts helps identify recent file changes. Journal analysis provides a detailed history of file activities, unlike SMFT snapshots, revealing events like file renaming, moving, and deletion, which aids in reconstructing actions for forensic investigations and incident response.
- The LogFileParser tool simplifies analyzing the complex $LogFile, which logs detailed file system events like file creations, deletions, and modifications. The primary output, "LogFile.csv," summarizes these events, including Log Sequence Numbers, operations, and file names. Supplementary files provide additional details like

timestamps and MFT record numbers, aiding in the forensic reconstruction of file system activities, essential for incident response and threat hunting.

- The MFTECmd tool to analyze $UsnJrnl for changes in NTFS volumes. The tool parses $UsnJrnl files to output CSV data, including file names, MFT numbers, timestamps, update reasons, and attribute flags, including options for processing volume snapshots with the --vss switch, making it useful for thorough forensic investigations.
- When a file is deleted in NTFS, changes occur across three layers: Data, Metadata, and Filename. In the Data Layer, clusters are marked unallocated in $Bitmap, but the data remains until clusters are reused. In the Metadata Layer, a bit in the file's $MFT record is flipped, retaining all metadata until the record is reused, with $LogFile, $USNJrnl, and other logs still referencing the file. In the Filename Layer, the $FILE_NAME attribute and $I30 index entry in the parent directory are preserved until the $MFT record is reused. These processes ensure data resilience and provide forensic analysts with detailed insights into file system changes during deletion.

5.4Advanced Evidence Recovery

- The detecting and overcoming anti-forensics techniques used by attackers to hide their tracks, such as file wipers and privacy cleaners like BleachBit, ERASER, BCWipe, and SDelete. These tools overwrite deleted files to prevent recovery, complicating forensic analysis. As cyber intrusions rise, attackers increasingly use such methods to avoid detection, driven by improved organizational security measures. Detecting missing data, identifying signs of data destruction, and employing advanced recovery techniques are crucial for forensic analysts. Understanding the behavior of these anti-forensics tools helps uncover hidden evidence and ensures thorough investigations.
- SDelete, a file wiper from Microsoft's Sysinternals suite, which wipes files, directories, and free space but leaves numerous forensic artifacts. Key artifacts include entries in the $USNJrnl, Windows Search Index logs, $I30 slack space, and Prefetch files, which can reveal file wiping activities despite the deletion. For instance, the $USNJrnl shows successive name changes to files, while Prefetch files list files accessed by SDelete within the first 10 seconds. These artifacts highlight that even sophisticated wiping tools like SDelete leave traces that can be crucial for forensic investigations, providing significant evidence of file tampering and deletion attempts.
- BCWipe, Eraser, and Cipher, three other popular file wipers, highlighting their functions and the forensic artifacts they leave. BCWipe is a commercial tool for enterprises with extensive configuration options, renaming files randomly and leaving artifacts like $UsnJrnl, $LogFile, and MFT records. Eraser, an open-source tool recommended by US-CERT, similarly renames files and MFT records, leaving similar artifacts. Cipher, a built-in Windows utility, primarily encrypts but also overwrites free space, creating a persistent directory and temp files. Despite their effectiveness in

overwriting data, these tools leave behind artifacts such as $UsnJrnl, $LogFile, $I30 slack, and MFT records, aiding forensic

- The recovery of deleted registry data in Windows investigations, emphasizing that registry hives, like filesystems, have unallocated space where deleted keys and values can be recovered. Eric Zimmerman's Registry Explorer tool is highlighted for its ability to make this recovery process straightforward, especially after the use of privacy cleaners like BleachBit. The registry data can be reconstructed from transaction logs, which may persist for days or weeks, providing further insights even when data is seemingly deleted.
- Windows registry used by attackers to hide "fileless" malware, such as PowerShell scripts, making it difficult to detect amongst legitimate entries. Registry Explorer by Eric Zimmerman is praised for its effectiveness in identifying these hidden threats through features like searching for specific key and value names, detecting large or Base64-encoded values, and examining "value slack" space. The tool's ability to filter and prioritize results helps investigators uncover and analyze suspicious activities within the registry, mitigating the risks posed by these stealthy malware techniques.
- There are two main methods for recovering deleted files: the Metadata Method, which utilizes intact metadata to locate and extract data from disk clusters when files are deleted but not overwritten, and the Carving Method, which searches for known file signatures to recover data even when metadata has been reused or overwritten. It emphasizes that the choice of method depends on the specific circumstances, such as the type of storage device and the nature of the data loss, and lists potential targets for recovery like link files, jump lists, recycle bin contents, and web history.
    - File recovery using the Metadata Method, highlighting tools like icat and tsk_recover from The Sleuth Kit, which extract deleted files by locating unallocated metadata entries and corresponding data clusters. It emphasizes that forensic tools can efficiently find and export these entries, allowing for the recovery of deleted files, as demonstrated with the FTK Imager tool. However, it notes that successful recovery requires the metadata entries and data clusters to remain unused; otherwise, recovery may be partial or unsuccessful, underscoring the need for careful validation during the recovery process.
    - File recovery using the Carving Method, specifically highlighting PhotoRec as a powerful and free tool that supports Windows, Linux, and Mac. PhotoRec uses known file signatures to recover data from over 300 file types and leverages metadata from carved files. It supports various file systems like NTFS, FAT, exFAT, ext2/ext3/ext4, and can identify cluster size to minimize false positives. Additionally, PhotoRec can replace generic file names with real names stored in metadata, enhancing the recovery process. While primarily designed for general data recovery, PhotoRec includes features beneficial for forensic

analysis, such as running directly against raw disk images and even E01 forensic images.

- The recovery of deleted Volume Shadow Snapshots (VSS), which are valuable for retrieving previously deleted files or versions. These shadow copy files, stored in the System Volume Information folder, can be carved and recreated using tools like vss_carver.py. Although VSS files provide a snapshot in time and are limited in number, they can be critical in cases where malware or attackers have deleted snapshots. An open-source tool has made it easier to recover these files, even those deleted due to capacity limits or malicious actions.
    - The quick method for recovering Volume Shadow Snapshots.The process includes running vss_carver on the raw image, reordering the recovered snapshots, presenting them as raw disk images, and mounting the filesystems. This method can recover a small image in about 10 minutes, with recent deletions having a higher success rate. It also notes challenges like deleted catalog file references, which complicate recovery, but vss_carver remains effective if the data is still present on the disk.
- The Bulk Extractor with Record Carving (bulk_extractor-rec), a tool used for stream-based data carving. It highlights its potential to recover important NTFS records like MFT, $I30, $USNJRNL, and $LogFile, as well as event log EVTX records. Bulk Extractor is praised for its speed and efficiency in scanning and processing input data to find useful information. The tool is designed to use multi-core systems, detect and decompress files, and use specific "scanners" for different file types. The forked version, bulk_extractor-rec, adds additional record carvers for NTFS and EVTX logs, making it a valuable tool for forensic analysis.
- There are two string searching techniques in digital forensics: Direct String Search, which uses tools like grep or bstrings for precise bit-by-bit searches, and Indexed String Search, which creates a comprehensive keyword index using tools like Apache Solr or DfSearch. Direct String Search is thorough and ideal for uncompressed files but struggles with compressed files and requires decompression tools. Indexed String Search, while time-consuming to set up, allows fast and comprehensive searches across various file types post-indexing but may exclude certain characters or strings. Examples include using bstrings for Bitlocker key searches and Autopsy for indexing and keyword searches within disk images.

5.5 Defensive Countermeasures

- The importance of leveraging file system history for forensic analysis by ensuring volume snapshots are enabled and properly configured, such as disabling "ScopeSnapshots" and increasing the reserved size for snapshots to enhance data recovery capabilities. It also recommends increasing NTFS journal sizes ($LogFile and $USNJRNL) for better logging efficiency and monitoring for suspicious file system activity using tools like fsutil, vssadmin, wmic shadowcopy, and win32_shadowcopy. For example, adjusting the USN journal size can significantly extend the period over which file system changes are recorded, aiding in detailed forensic investigations. The practical steps, including creating a larger USN journal allocation, configuring VSC allocation, and setting scheduled tasks for regular snapshot creation to improve system visibility and historical data retention.

- The importance of comprehensive logging to enhance visibility and security, recommending consistent log collection, central log forwarding, and treating endpoints as critical data sensors. Advocates for deploying advanced logging configurations, including improved PowerShell and Windows audit policies and EDR technologies like Sysmon, while increasing local log storage. Emphasizing the necessity of robust logging to detect attacker activities such as malware execution and lateral movement, it highlights the use of tools like Sysmon for detailed event logging. To safeguard the logging infrastructure from tampering, strategies like log forwarding and heartbeat monitoring are suggested, ensuring better detection and response to security incidents through careful planning and resource utilization.

Cybersecurity Community Groups

- SANS GCFA
- SANS GCTI
- SANSs
- Security +
- Cysa+
- A+ Channel
- A+ Group
- eCIR

Command Reference

- The table below provides a summary of tools used in GCFA course, including their respective command arguments and a brief description of their purpose

Tool Name	Command Arguments	Description
densityscout	densityscout –pe –r -p 0.1 -o results.txt c:\Windows\System32	finding suspicious files common obfuscation techniques
EvtxECmd (Windows Lab)	evtxecmd -f E:\c\Windows\system32\winevt\logs\Security.evtx --csv g:\event-logs --csvf security.csv	facilitate event log analysis.
fls tool	fls [options] image [inode]	The fls tool allows us to interact with a forensic mage as though it were a normal filesystem
Get-LogparserStack.ps1	.\Get-LogparserStack.ps1 -FilePattern *SvcAll.csv -Delimiter "," -Direction asc -OutFile NameOfOutPutFile.csv	general script in the Kansa framework allowing the analyst to stack many different types of data
grep	grep -i search_term sorted_strings.txt	
grep	grep -i unknown zeus-apihooks.txt | wc -l	How many "unknown" found in a string
AppCompatProcessor.py	AppCompatProcessor.py ./database. db tcorr "volrest.exe"	tcorr feature takes a known filename as an input and try to give you related files in the dataset you target (Used Linux)
AppCompatProcessor.py	AppCompatProcessor.py ./database. db search	search is a feature with this tool that search in its signature collection to find anomalies and present it to you
AppCompatProcessor.py	AppcompatProcessor.py ./dataset.db reconscan AppcompatProcessor.py ./dataset.db List	Reconnaissance Activity and score for each system , Scale analysis.
AppCompatProcessor.py (Linux Lab)	AppCompatProcessor.py ./database.db load /cases/precooked/appcompat/SRL_Shi m_Amcache.zip	AppCompatProcessor.py is a feature rich toolset designed to perform scalable hunting of ShimCache and Amcache databases
autorunsc	autorunsc.exe /accepteula -a * -c -h -s '*' -nobanner	For reference only, the Kansa script executes Autorunsc.exe
Baseline	Python3 baseline.py -drv -i **.img -b baseline.img --showknown –imphash -o out.txt	Drivers (-drv) and matching them with baseline to find anomalies
bstrings	bstrings -f 0x0000000002be0000.vvmem -m 8	in windows
mactime	mactime [options] –d -b bodyfile -z timezone > timeline.csv	Create the filesystem timeline in CSV format
mactime	mactime -z UTC -y -d -b /mnt/g/timeline/baserd01-mftecmd.body 2018-08-23..2018-09-07 > /mnt/g/timeline/ baserd01-filesystem-timeline.csv	Create the filesystem timeline in CSV format
MemProcFS	MemProcFS.exe –forensic 1 –device memory.img –pagefile0 pagefile.sys	Mounting a windows memory image as a virtual file system
MemProcFS	MemProcFS.exe -forensic 1 -devicebase-rd01-memory.img	Windows
MemProcFS.exe	MemProcFS.exe -forensic 1 -devicebase-rd01-memory.img	start memory image
MFTECmd.exe	MFTECmd.exe -f “E:\C\$MFT” --csv “G:\timeline” --csvf mft.csv	Extracting data from $MFT files (NTFS system files)
MFTECmd.exe	MFTECmd.exe -f "E:\C\$MFT" --body "G:\timeline" –bodyf baserd01-mftecmd.body --blf --bdl C:	Extracting data from $MFT files (NTFS system files)
PEcmd	pecmd -d C\Windows\prefetch -q --csvf name.csv --csv C:\Path	Extract Data from all Prefetch data
Process-EventLogs.ps1	.\Process-EventLogs.ps1 -source E: \c\Windows\system32\winevt\logs\ - dest C:\"Any Path"	This is the script that batch or collect every log of interest from the Logs folder and group them in a single CSV file for you to analyze all (Amazing)
Select-String	Slelect-String "partOfText" *multipleFiles.txt	Similar to findstr & grep
strings	strings -a -t d file > strings.txt strings -a -t d -e l file >> strings.txt sort strings.txt > sorted_strings.txt	

- The table below highlights the equivalent commands in Volatility 2 and Volatility 3

Usage	Volatility 2	Volatility 3
List of Running Processes (EPROCESS)	vol2.py -f img --profile=profile pslist	vol.py -f img windows.pslist.PsList
identify profile information	vol2.py -f img --profile=profile kdbgscan	vol.py -f img windows.info.Info
List of Process in Tree	vol2.py -f img --profile=profile pstree	vol.py -f img windows.pstree.PsTree -r pretty
Scan for process blocks	vol2.py -f img --profile=profile psscan	vol.py -f img windows.psscan.PsScan -r pretty
Display loaded DLLs	vol2.py -f img --profile=profile dlllist	vol.py -f img windows.dlllist.DllList --pid # -r pretty
Display command line argumants	vol2.py -f img --profile=profile cmdline	vol.py -f img windows.cmdline.CmdLine --pid # -r pretty
Display SIDs foe each process	vol2.py -f img --profile=profile getsids	vol.py -f img windows.getsids.GetSIDs --pid # -r pretty
list handles opened by the process	vol2.py -f img --profile=profile handles -t resource	vol.py -f img windows.handles.Handles --pid # -r pretty
Vol2 dumps exe Vol3 DLLs and exe	vol2.py -f img --profile=profile procdump -p # --dump-dir="/path/to/dir"	vol.py -f img windows.dumpfiles -o "/path/to/file"
Scan unusual memory sections	vol2.py -f img --profile=profile malfind	vol.py -f img windows.malfind.Malfind
Scan Files	vol2.py -f img --profile=profile filescan	vol.py -f img windows.filescan.FileScan
Dump Files	vol2.py -f img --profile=profile dumpfiles -p # --dump-dir="/path/to/dir"	vol.py -f img windows.dumpfiles -o "/path/to/file"
Hidden, unliked DLLs 32bit	vol2.py -f img --profile=profile ldrmodules	vol.py -f img windows.ldrmodules.LdrModules
Linked List Network Connection	vol2.py -f img --profile=profile netstat	vol.py -f img windows.netstat.NetStat
Scan for network connection	vol2.py -f img --profile=profile netscan	vol.py -f img windows.netscan.NetScan
Identify process hallowing artifcats	vol2.py -f img --profile=profile hollowfind	none
Display hooked function in SSDT	vol2.py -f img --profile=profile ssdt	vol.py -f img windows.ssdt.SSDT
Cross-view analysis	vol2.py -f img --profile=profile psxview -R	none
Extract/identify Loaded kernel drivers	vol2.py -f img --profile=profile moddump	vol.py -f img windows.modules.Modules vol.py -f img windows.modscan.ModScan
Detect Inline & IAT hooks	vol2.py -f img --profile=profile apihooks	none