|  |  |  |  |
| --- | --- | --- | --- |
| 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 |  |  |  |
| Identifying Suspicious Network Activity | beaconing | 1 | 43 |
| Examining Services | Get-Service Get-CimInstance | 1 | 44 |
| Registry Interrogation | Get-ChildItem Get-ItemProperty | 1 | 45 |
| Unusual Accounts | Get-LocalUser Get-LocalGroup Get-LocalGroupMember | 1 | 46 |
| Unusual Scheduled Tasks | Get-ScheduledTask Get-ScheduledTaskInfo Export-ScheduledTask | 1 | 47 |
| Unusual Log Entries | Get-WinEvent | 1 | 48 |
| Differential Analysis | Compare-Object | 1 | 50 |
| PowerShell Cheat Sheet |  | 1 | 51 |
| Additional Supporting Tools |  | 1 | 53 |
| Visual summary: Live Examination |  | 1 | 54 |
| Network Investigations |  | 1 | 59 |
| Analyzing Packet Captures |  | 1 | 60 |
| Tcpdump |  | 1 | 61 |
| Useful tcpdump Options |  | 1 | 62 |
| Berkeley Packet Filters (BPF) |  | 1 | 63 |
| BPF Examples |  | 1 | 64 |
| Web Proxies |  | 1 | 65 |
| Access Logs | Cat access log | 1 | 66 |
| Visual Summary: Network Investigations | WinPmem: capture RAM in Windows | 1 | 67 |
| Memory Investigations |  | 1 | 70 |
| Volatility Overview | windows.pslist.PsList | 1 | 71 |
| Listing Processes | windows.pstree.PsTree | 1 | 72 |
| Parent and Child Processes | windows.netscan.NetScan | 1 | 73 |
| Scanning for Network Connection | windows.cmdline.CmdLine | 1 | 74 |
| Process Command Line | WinPmem: capture RAM in Windows | 1 | 75 |
| Applying Memory Investigation |  | 1 | 76 |
| Visual Summary: Memory Investigations |  | 1 | 77 |
| Memory Investigations |  | 1 | 81 |
| Malware Investigations |  | 1 | 82 |
| Online Analysis Sites | VirusTotal and Hybrid Analysis | 1 | 83 |
| Practicing Good Hygiene |  | 1 | 84 |
| Basic Attributes | Get-FileHash sha256sum file | 1 | 85 |
| Monitoring the Environment |  | 1 | 86 |
| Snapshot vs Continuous Recording |  | 1 | 87 |
| Regshot | Regshot | 1 | 88 |
| Regshot Output |  | 1 | 89 |
| Process Monitor | Process Monitor: Microsoft tool to monitor operating environment | 1 | 90 |
| Summarizing Process Monitor Output |  | 1 | 92 |
| Analyzing Code | IDA Pro (includes decompiler Hex-Rays) Ghidra | 1 | 93 |
| Visual Summary: Malware Investigations |  | 1 | 81 |
| Cloud Investigations |  | 1 | 96 |
| Security Responsibility Demarcation (IaaS, PaaS, SaaS) |  | 1 | 97 |
| Preparation: Cloud IR Account Access |  | 1 | 98 |
| Preparation: Configuring Logging |  | 1 | 99 |
| Detection: Cloud Analysis Tools | AWS GuardDuty Azure Sentinel GCP CloudArmor | 1 | 101 |
| Containment: Isolate, Protect, Snapshot, Label |  | 1 | 102 |
| Containment: Data Collection with Storage | aws ec2 describe-volumes aws ec2 attach-volume | 1 | 103 |
| Analysis: Cloud Logging Data, Dashboards | [s3logparse.py](http://s3logparse.py) | 1 | 104 |
| Response: Access Key Revocation | **Reset password twice for Azure AD** | 1 | 105 |
| Recovery and Remediation |  | 1 | 106 |
| Cloud IR - Additional Considerations |  | 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 |
| Why We're Covering What We're Covering |  | 2 | 6 |
| Always Get Permission |  | 2 | 7 |
| General Trends: Attack for Fun and Profit |  | 2 | 8 |
| General Trends: Consumer Device Attacks |  | 2 | 9 |
| General Trends: The Golden Age |  | 2 | 10 |
| **MITRE ATT&CK Framework** |  | 2 |  |
| MITRE ATT&CK Framework |  | 2 |  |
| How We Integrate SEC504 with ATT&CK |  | 2 | 12 |
| MITRE ATT&CK References |  | 2 | 14 |
| **Open-Source Intelligence** | whois | 2 | 15 |
| Reconnaissance | crt.sh | 2 |  |
| Open-Source Intelligence | haveibeenpwned whatsmyname  SpiderFoot | 2 | 17 |
| Certificate Transparency: The New WHOIS |  | 2 | 18 |
| OSINT Data Collection |  | 2 | 20 |
| Not OSINT Data |  | 2 | 23 |
| Visual Summary: Open-Source Intelligence | nslookup dig | 2 | 27 |
| **DNS Interrogation** | nslookup server 81.4.108.41 settype=AXFR ls -d zonetransfer.me tcp/port 53 | 2 | 28 |
| DNS Interrogation | dig @81.4.108.41 AXFR zonetransfer.me | 2 |  |
| DNS Zone Transfer in Windows | sudo nmap --script dns-brute | 2 | 31 |
| DNS Zone Transfer in UNIX | whois | 2 | 32 |
| DNS Automated Interrogation | crt.sh | 2 | 33 |
| DNS Reconnaissance Defenses |  | 2 | 34 |
| Visual Summary: DNS Interrogation |  | 2 | 35 |
| **Website Reconnaisance** |  | 2 |  |
| Website Reconnaisance |  | 2 | 39 |
| Exiftool | Exiftool | 2 | 40 |
| Website Crawl and Wordlist Generation: CeWL | CeWL | 2 | 41 |
| Reconnaisance with Search Engines | Google Dorks  Google Hacking Database | 2 | 42 |
| Other Website Information |  | 2 | 43 |
| Web-Based Recon/Attack Tools | www.shodan.io www.network-tools.com  viewdns.info www.securityspace.com | 2 | 44 |
| Website Reconnaisance Defenses | robots.txt | 2 | 46 |
| Visual Summary: Website Reconnaisance | Exiftool | 2 | 47 |
| **Network and Host Scanning with Nmap** |  | 2 |  |
| Network Mapping with Nmap |  | 2 | 49 |
| Sweeping for Network Mapping | ICMP Echo Request  TCP SYN to port 43  TCP ACK to port 80  ICMP Timestamp Request | 2 | 50 |
| Nmap Host Discovery | sudo nmap -sn 0.0.0.0 | 2 | 51 |
| Closer Look at the IP Header |  | 2 | 52 |
| How Traditional Traceroute Works | sudo nmap -sn --traceroute | 2 | 53 |
| Port Scanning |  | 2 | 56 |
| TCP and UDP Ports |  | 2 | 57 |
| TCP Three-Way Handshake |  | 2 | 58 |
| Nmap: Scan Types |  | 2 | 59 |
| Nmap NSE Scripts |  | 2 | 61 |
| Visual Summary: Nmap |  | 2 | 63 |
| **Cloud Spotlight: Cloud Scanning** |  | 2 |  |
| Cloud Spotlight: Cloud Scanning |  | 2 | 66 |
| JQ and JSON Data |  | 2 | 67 |
| Cloud Scanning Process |  | 2 | 68 |
| Exhaustive IP Address Enumeration |  | 2 | 69 |
| Scanning Large Ranges: Masscan | Masscan awk | 2 | 70 |
| Masscan Scan of AWS us-east-1 |  | 2 | 71 |
| Attributing Hosts |  | 2 | 72 |
| TLS-Scan | TLS-Scan | 2 | 73 |
| Interpreting TLS-Scan Results |  | 2 | 74 |
| Other Scanners: EyeWitness |  | 2 | 75 |
| Cloud Scanning Defense |  | 2 | 77 |
| Visual Summary: Cloud Scanning |  | 2 | 79 |
| **SMB Security** |  | 2 |  |
| SMB Security | TCP port 445 | 2 | 82 |
| SMB Security Features |  | 2 | 83 |
| SMB Shares | Get-WmiObject (WMI)  net.exe (Windows an non-Windows) | 2 | 84 |
| Searching for SMB Shares: SMBeagle | SMBeagle | 2 | 85 |
| SMB Share Data Harvesting, Search |  | 2 | 86 |
| SMB Access: The Linux Way | smbclient | 2 | 87 |
| Using Samba's rpcclient from Linux for More Info | rpcclient | 2 | 88 |
| SMB Exploits |  | 2 | 89 |
| SMB Password Attacks |  | 2 | 90 |
| Identifying and Dropping SMB Sessions | Get-SmbSession Net use | 2 | 91 |
| Preparation: Defenses Against Evil SMB Sessions | Block access to TCP/445,135,137,139 UDP/445,137,138 | 2 | 93 |
| Visual Summary: SMB Attacks |  | 2 | 94 |
| Defense Spotlight: DeepBlueCLI | useful, not perfect | 2 | 98 |
| DeepBlueCLI Capabilities |  | 2 | 99 |
| DeepBlueCLI Metasploit Attack Detection |  | 2 | 101 |
| DeepBlueCLI Non-Malicious Event |  | 2 | 102 |
| DeepBlueCLI Output Formatting |  | 2 | 103 |
| Conclusion |  | 2 | 104 |
| Visual Summary: DeepBlueCLI |  | 2 | 105 |
| Password Guessing Attacks |  | 3 | 4 |
| Password Guessing Alternative: Password Spraying |  | 3 | 5 |
| THC Hydra Password Guessing | Hydra | 3 | 6 |
| Password Guess Selection |  | 3 | 7 |
| Credential Stuffing |  | 3 | 8 |
| Visual Summary: Password Attacks |  | 3 | 10 |
| Microsoft 365 Password Attacks |  | 3 | 13 |
| Microsoft 365 Authentication API |  | 3 | 14 |
| MSOLSpray | invoke-MSOLSpray | 3 | 15 |
| Azure Smart Lockout |  | 3 | 16 |
| AWS API Gateway |  | 3 | 17 |
| FireProx and MSOLSpray | invoke-MFASweep | 3 | 18 |
| MFA Bypass and Microsoft 365 |  | 3 | 19 |
| Microsoft 365 Incident Response: Identification | Account license downgrade: Get-MsoUser | 3 | 20 |
| Visual Summary: Microsoft 365 Password Attacks |  | 3 | 21 |
| Understanding Password Hashes |  | 3 | 24 |
| Windows LANMAN Hashes |  | 3 | 25 |
| NT Hashes |  | 3 | 26 |
| Password Hashes without Salt |  | 3 | 27 |
| Password Salting |  | 3 | 28 |
| Obtaining Windows Domain Controller Hashes | ntdutil secretsdump.py | 3 | 29 |
| 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 |  | 3 | 33 |
| Decoding UNIX/Linux Password Hashes |  | 3 | 34 |
| Hashing Rounds | MD5: 1000 rounds | 3 | 35 |
| Password Hashing Rounds | SHA256-512: 5000 rounds | 3 | 36 |
| Mitigating GPU-Based Password Cracking |  | 3 | 37 |
| Visual Summary: Understanding Password Hashes |  | 3 | 38 |
| Password Cracking |  | 3 | 41 |
| John the Ripper | sudo unshadow /etc/passwd /etc/shadow > combined    john combined | 3 | 42 |
| John's Cracking Modes |  | 3 | 43 |
| John's Input and Output | cracked passwords stored in john.pot | 3 | 44 |
| John Sample Session |  | 3 | 45 |
| Hashcat Hash and Cracking Support |  | 3 | 46 |
| Hashcat Attack Modes | Mass attack mode = Brute-force | 3 | 47 |
| Hashcat Attack Modes - By Example | Combinator mode (-a 1) uses 2 wordlists | 3 | 48 |
| Hashcat Mask Attack |  | 3 | 51 |
| Hashcat Rules |  | 3 | 55 |
| Preparation: Disable LANMAN Authentication |  | 3 | 56 |
| Preparation: Password Complexity Tools |  | 3 | 57 |
| Deploy Multi-factor Authentication |  | 3 | 58 |
| Visual Summary: Password Cracking |  | 3 | 59 |
| Cloud Spotlight: Insecure Storage |  | 3 | 63 |
| Cloud Storage Access | https://s3.amazonaws.com/BUCKETNAME  https://www.googleapis.com/storage/v1/b/BUCKETNAME https://ACCOUNTNAME.blob.core.windows.net/CONTAINERNAME | 3 | 65 |
| Storage Scanning: AWS S3 | cat words  bucket\_finder.rb words --download | 3 | 66 |
| Storage Scanning: Google Cloud Bucket | gcpbucketbrute.py -u -k falsimentis gsutil ls gs:/falsimentis-dev | 3 | 67 |
| Azure Scanning: Basic Blob Finder | 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 |  | 3 | 72 |
| Scanning Your Own Organization |  | 3 | 73 |
| DNS Logs, HTTP Proxy, Network Logs |  | 3 | 74 |
| The Need for Cloud Storage Logging |  | 3 | 75 |
| Visual Summary: Cloud Storage |  | 3 | 76 |
| Netcat |  | 3 | 79 |
| Netcat Client Mode | Standard Error (stderr) | 3 | 80 |
| Netcat Listen Mode |  | 3 | 81 |
| Some Netcat Uses |  | 3 | 82 |
| Netcat: Data Transfer |  | 3 | 83 |
| Netcat: Port Scanning |  | 3 | 84 |
| Netcat: Backdoors |  | 3 | 85 |
| Netcat: Persistent Backdoor Listeners |  | 3 | 86 |
| Netcat: Relays |  | 3 | 87 |
| Make a Netcat Relay on Linux |  | 3 | 88 |
| Netcat: Defense |  | 3 | 89 |
| Netcat: Closing |  | 3 | 90 |
| Visual Summary: Netcat |  | 3 | 91 |
| The Metasploit Arsenal |  | 4 | 6 |
| Metasploit User Interfaces |  | 4 | 7 |
| Exploits Currently Included Metasploit Framework |  | 4 | 8 |
| Payloads Included in Metasploit Framework |  | 4 | 9 |
| Metepreter Features |  | 4 | 10 |
| Attack Framework Defense: Preparation |  | 4 | 11 |
| Additional Defenses: Preparation |  | 4 | 12 |
| Visual Summary: Metasploit Framework |  | 4 | 13 |
| Drive-By Attacks |  | 4 | 16 |
| Drive-By Attack Operation |  | 4 | 17 |
| The Problem with Browsers |  | 4 | 18 |
| Watering Hole Attacks |  | 4 | 19 |
| Code-Executing Microsoft Office Files |  | 4 | 20 |
| Conventional Exploit Delivery |  | 4 | 23 |
| Fake Installers |  | 4 | 24 |
| Browser Exploitation Framework (BeEF) |  | 4 | 25 |
| Building Payloads - Metasploit MSFVenom |  | 4 | 27 |
| Drive-By Attack Defense |  | 4 | 30 |
| Visual Summary: Drive-By Attacks |  | 4 | 31 |
| Defense Spotlight: System Resource Usage Monitor Dum |  | 4 | 34 |
| SRUM Data Storage |  | 4 | 35 |
| SRUM-Dump Auto Extract |  | 4 | 37 |
| Visual Summary: System Resource Usage Monitor |  | 4 | 40 |
| Command Injection |  | 4 | 43 |
| Identifying Command Injection Vulnerabilities |  | 4 | 44 |
| More than a Web Vunerability |  | 4 | 45 |
| Command Injection Defenses |  | 4 | 48 |
| Visual Summary: Command Injection |  | 4 | 49 |
| Cross-site Scripting |  | 4 | 53 |
| Attack Opportunity |  | 4 | 54 |
| Stored Cross-site Scripting Attack |  | 4 | 55 |
| Reflected Cross-site Scripting Attack |  | 4 | 56 |
| Reflected XSS Phishing Link Example |  | 4 | 57 |
| What can an Attacker Do With XSS |  | 4 | 58 |
| Testing for XSS |  | 4 | 59 |
| Cross-site Scripting Defense |  | 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 |