

Creating a Field Manual

BRM

Creating a Field Manual



Theory
~8min

Practice
~18min

whoami

- Application Security Engineer (6 months)
 - Mostly white-box pentesting
- Software Engineering Bachelor
 - One year professional experience
- CPTS certified



Why Note-Taking Is Important

- Most do not take good notes
 - Not organized
 - Not comprehensive
 - Not straightforward
- As a result
 - Feel lost or stuck at CTFs
 - Frustration, Impostor Syndrome
 - Time wasted
- The solution: **Field Manuals!**

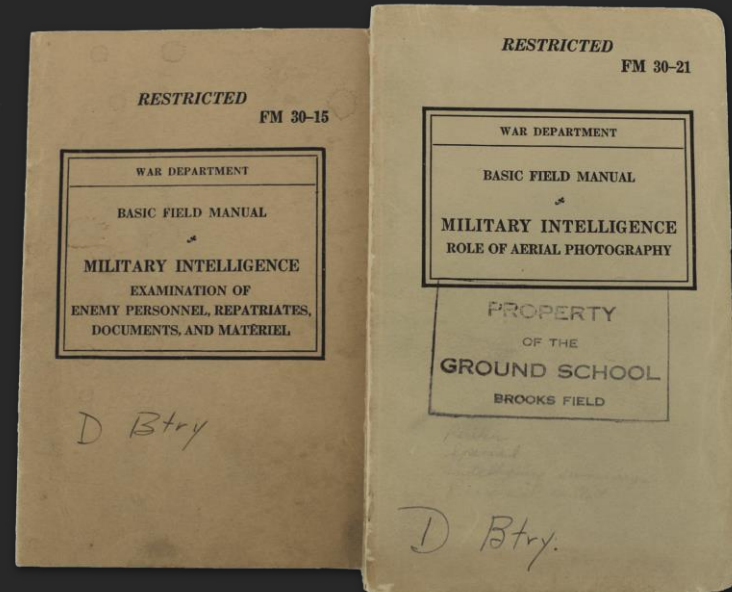
Military Problems

- Military operations are **time-sensitive**
- Tactical decision-making is **complex** and requires a lot of **thinking**
- Thinking takes a lot of **time**



Military Field Manual

- Document that can be referenced by operators
- Many challenges are repeatable and solvable with standard procedures
- No need to reinvent the wheel
- Saves times



BRM

Hacker Problems

- Hacking engagements are **time-sensitive**
- Hacking is **complex** and requires a lot of **thinking**
- Thinking takes a lot of **time**

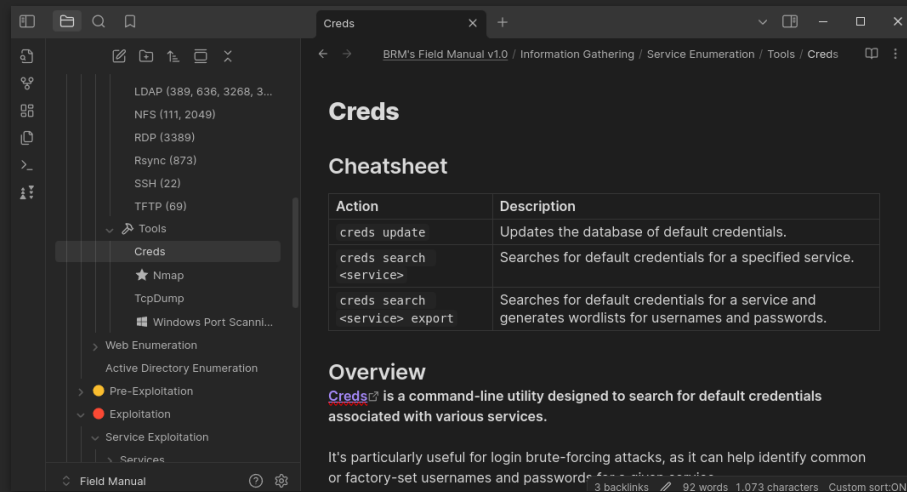


Hacking Field Manual

- Digital document
- Front-load as much of the thinking as possible before the engagement
- External memory

- **Requirements**

1. Organized and easy to navigate
2. List every known technique
3. Map technique to scenarios

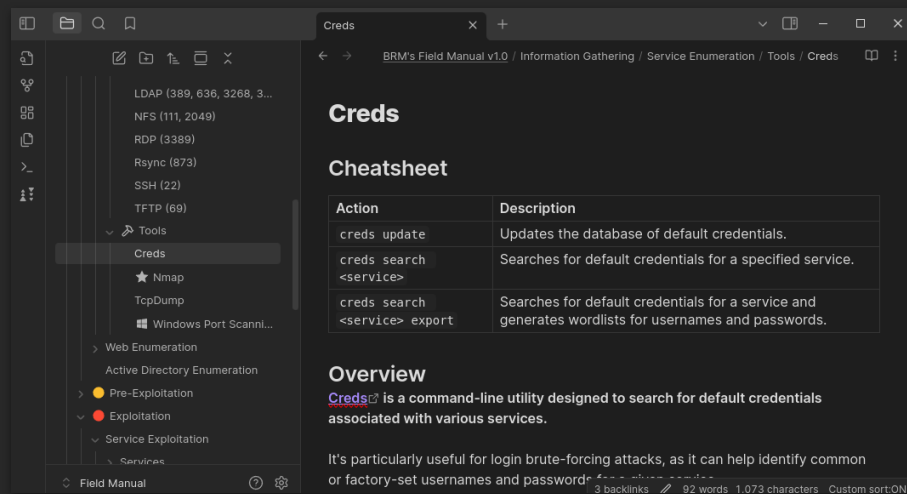


Hacking Field Manual

- Digital document
- Front-load as much of the thinking as possible before the engagement
- External memory

- **Requirements**

1. Organized and easy to navigate
2. List every known technique
3. Map technique to scenarios

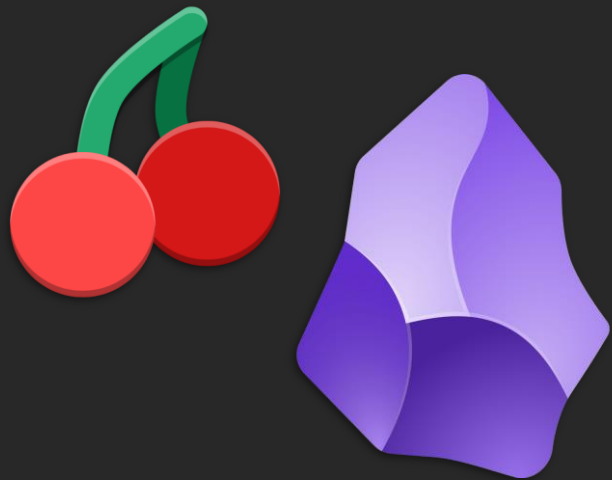


Creating a **Field Manual**

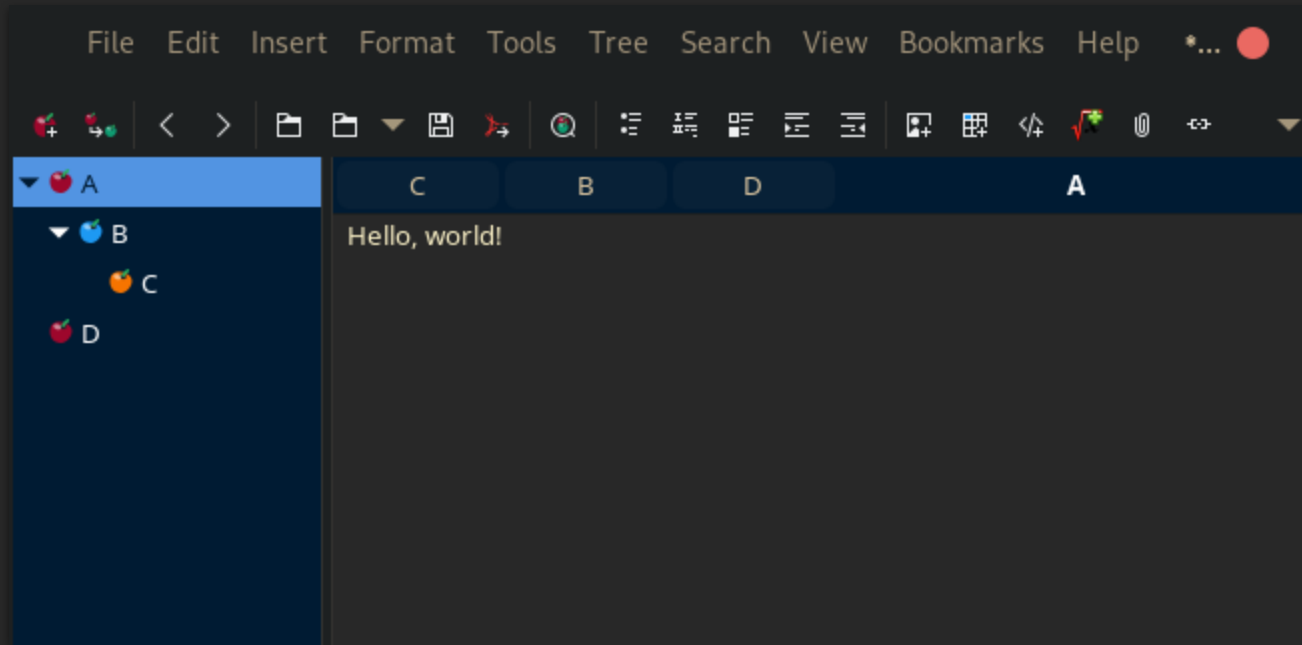
Practical Portion

Step 1: Choose Note-Taking Software

- Requirements
 - Hierarchical structure
 - Image Embedding
 - Tags
- Recommendations
 - CherryTree (FOSS)
 - Obsidian

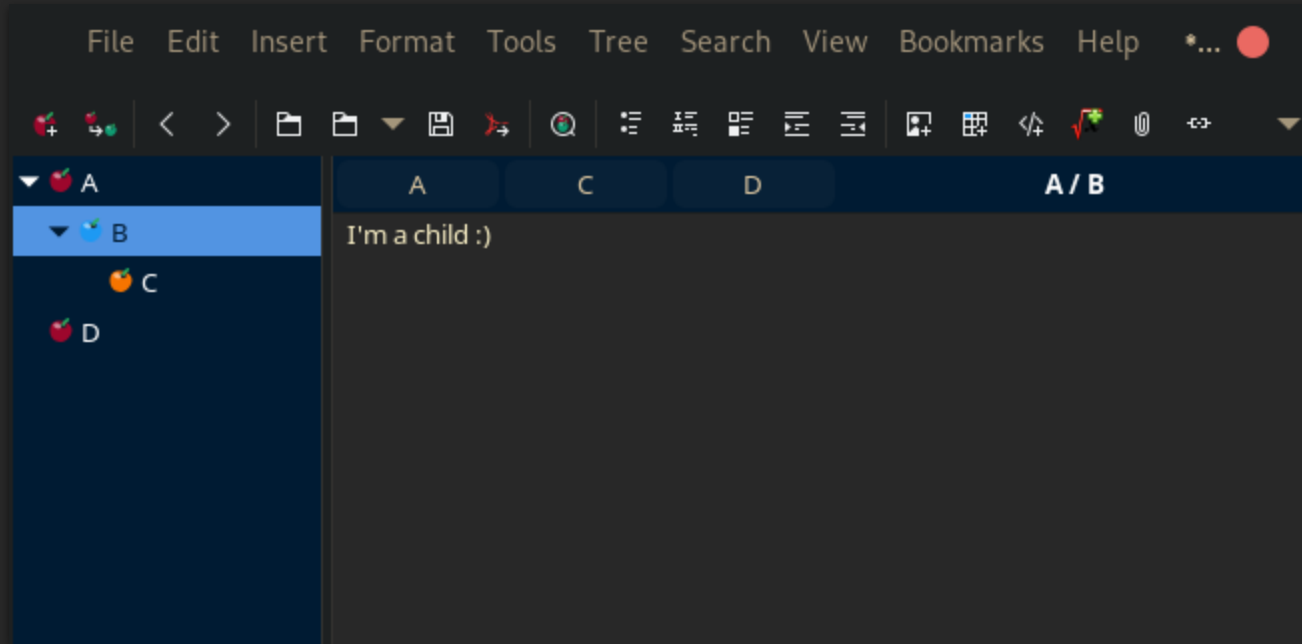


Step 0x01: Choose Note-Taking Software



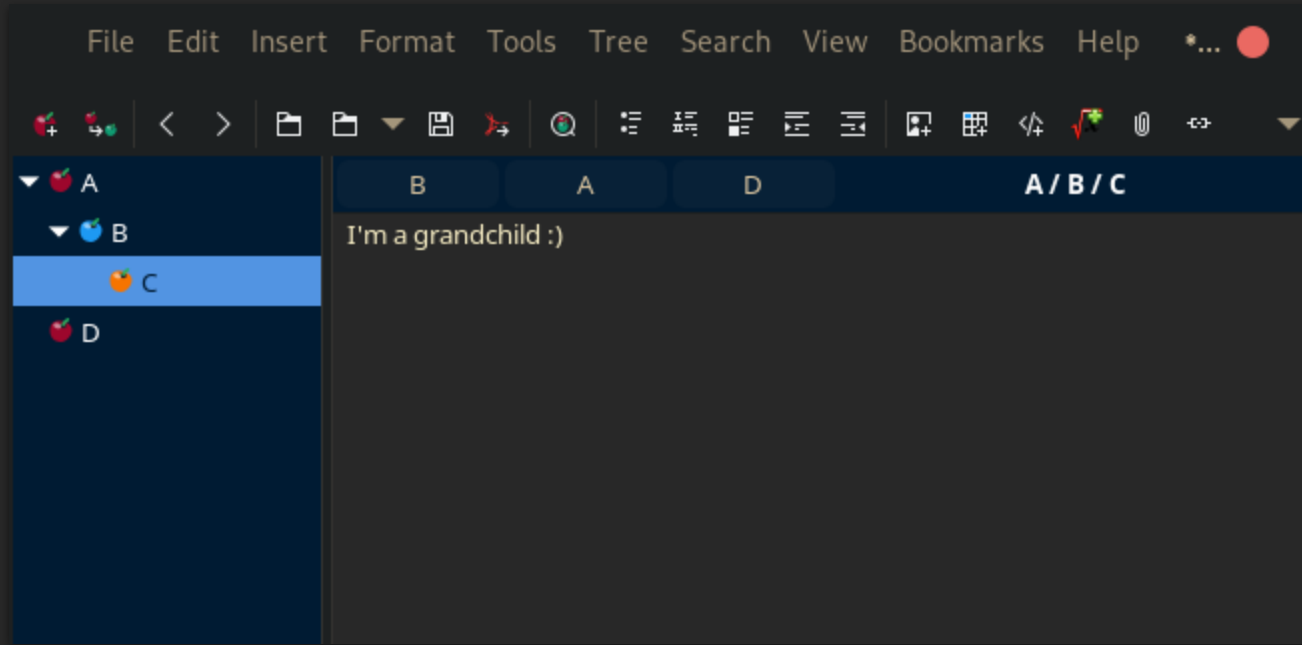
BRM

Step 0x01: Choose Note-Taking Software



BRM

Step 0x01: Choose Note-Taking Software



BRM

Step 0x02: Establish a Structure

- Requirement #1: Organized and easy to navigate
- Use hierarchy to organize
- Similar notes grouped together



Step 0x02: Establish a Structure

- ▶ ● Information Gathering
- ▶ ● Pre-Exploitation
- ▶ ● Exploitation
- ▶ ● Post-Exploitation
- ▶ ○ Lateral Movement

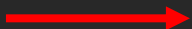
Step 2: Establish a Structure

QUESTION: Where is FTP enumeration?

- ▶ ● Information Gathering
- ▶ ● Pre-Exploitation
- ▶ ● Exploitation
- ▶ ● Post-Exploitation
- ▶ ○ Lateral Movement

Step 0x02: Establish a Structure

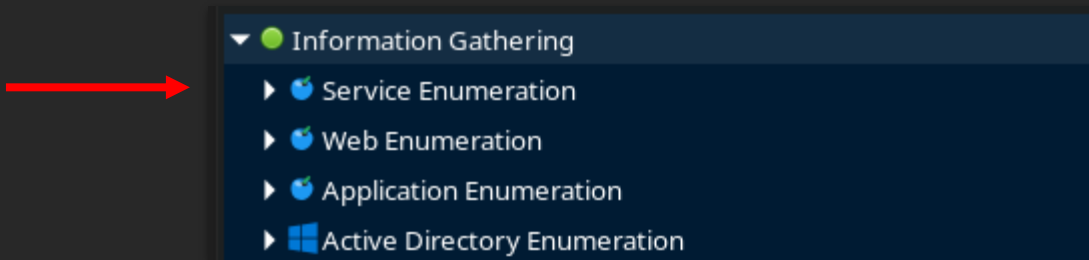
QUESTION: Where is FTP enumeration?



- ▶ ● Information Gathering
- ▶ ● Pre-Exploitation
- ▶ ● Exploitation
- ▶ ● Post-Exploitation
- ▶ ○ Lateral Movement

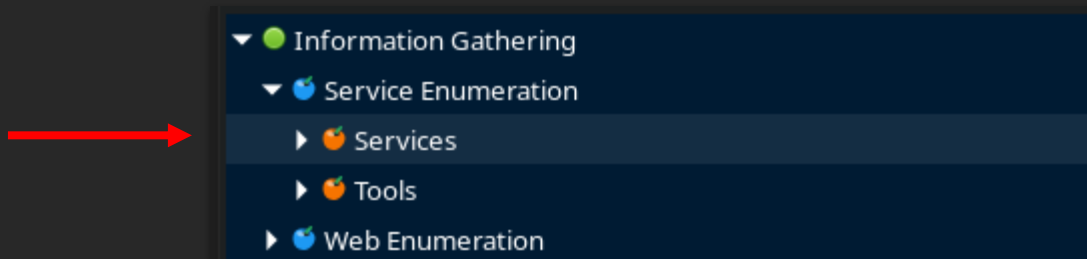
Step 0x02: Establish a Structure

QUESTION: Where is FTP enumeration?

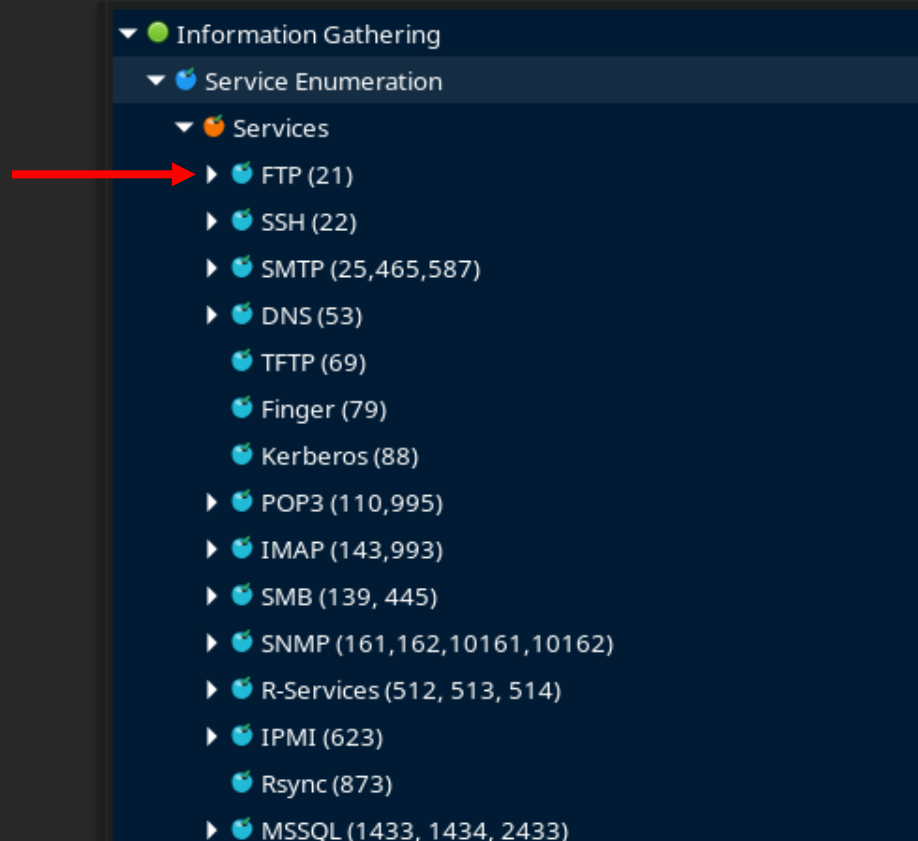


Step 0x02: Establish a Structure

QUESTION: Where is FTP enumeration?



Step 0x02: Establish a Structure



Step 0x02: Establish a Structure

Where is my note about **MSFVenom**?

▼ 🟡 Pre-Exploitation

▼ 🍏 Tools

🍎 MSFVenom

Step 0x02: Establish a Structure

Where is my note about **File Disclosure via XXE**?

- ▼ 🍒 Exploitation
 - ▼ 🍏 Web Exploitation
 - ▼ 🍊 XXE
 - 🍏 File Disclosure

Step 0x02 Establish a Structure

Where is my note about **Linux Privesc** via readable `/etc/shadow`?

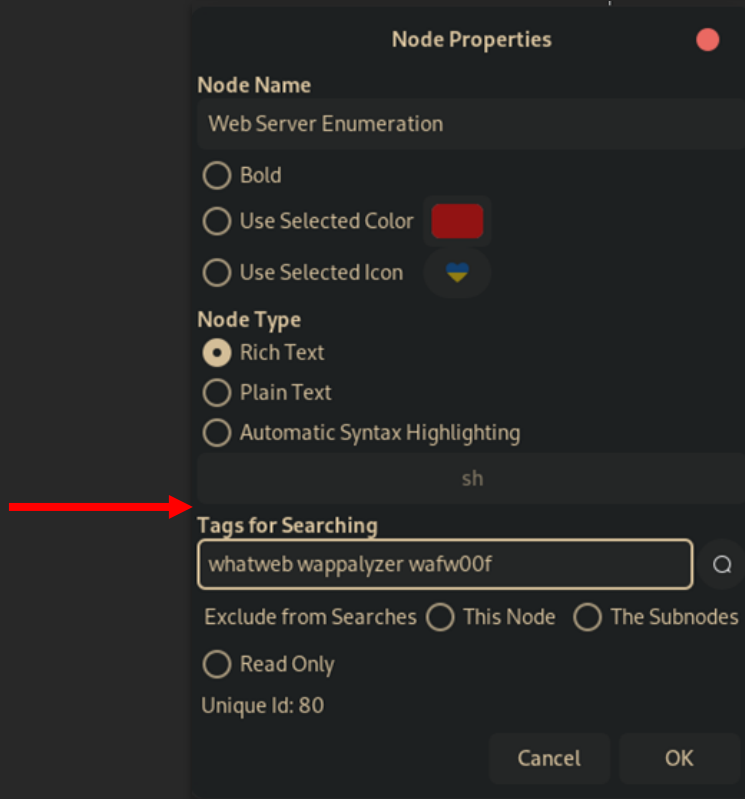
- ▼ ● Post-Exploitation
 - ▼ 🐧 Linux Post Exploitation
 - ▼ 🍊 Privilege Escalation
 - ▼ 🍏 Passwd, Shadow & Opasswd
 - 🍊 Readable `/etc/shadow`

Step 0x02: Establish a Structure

- ▼  Information Gathering
 - ▼  Service Enumeration
 - ▼  Services
 - ▼  FTP (21)
 -  Dangerous Settings
 - ▼  SSH (22)
 -  Authentication
 -  Dangerous Settings
 - ▼  SMTP (25,465,587)
 -  Common Commands
 - ▼  DNS (53)
 -  Record Types
 -  Dangerous Settings
 -  TFTP (69)
 -  Kerberos (88)
 - ▼  POP3 (110,995)

BRM

Step 0x02: Establish a Structure



The image shows a 'Node Properties' dialog box with a red arrow pointing to the 'Tags for Searching' input field. The dialog box has a title bar with a red close button. It contains several sections: 'Node Name' with a text field containing 'Web Server Enumeration'; three radio buttons for 'Bold', 'Use Selected Color' (with a red color swatch), and 'Use Selected Icon' (with a Ukrainian flag icon); 'Node Type' with three radio buttons: 'Rich Text' (selected), 'Plain Text', and 'Automatic Syntax Highlighting'; a text field containing 'sh'; 'Tags for Searching' with a text field containing 'whatweb wappalyzer wafw00f' and a search icon; 'Exclude from Searches' with three radio buttons: 'This Node' (selected), 'The Subnodes', and 'Read Only'; and 'Unique Id: 80'. At the bottom are 'Cancel' and 'OK' buttons.

Node Properties

Node Name
Web Server Enumeration

☐ Bold
☐ Use Selected Color 
☐ Use Selected Icon 

Node Type
☒ Rich Text
☐ Plain Text
☐ Automatic Syntax Highlighting

sh

Tags for Searching
whatweb wappalyzer wafw00f 

Exclude from Searches ☒ This Node ☐ The Subnodes
☐ Read Only

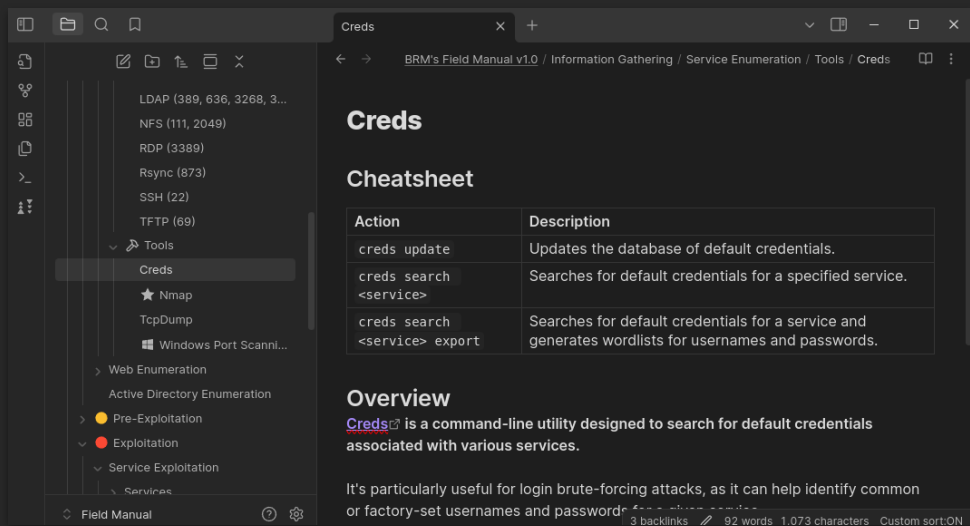
Unique Id: 80

Cancel OK

BRM

Step 0x03: Document Every Technique

- Create notes for topics and document known techniques
- Make modifications as you learn more
- The content of your notes should also have structure



The screenshot shows a web browser window with a dark theme. The address bar displays the URL: [BRM's Field Manual v1.0 / Information Gathering / Service Enumeration / Tools / Creds](#). The page title is "Creds". Below the title is a section labeled "Cheatsheet" containing a table with two columns: "Action" and "Description".

Action	Description
<code>creds update</code>	Updates the database of default credentials.
<code>creds search <service></code>	Searches for default credentials for a specified service.
<code>creds search <service> export</code>	Searches for default credentials for a service and generates wordlists for usernames and passwords.

Below the table is an "Overview" section. It states: "Creds is a command-line utility designed to search for default credentials associated with various services." It further explains: "It's particularly useful for login brute-forcing attacks, as it can help identify common or factory-set usernames and passwords."

At the bottom of the page, there is a footer with statistics: "3 backlinks 92 words 1,073 characters Custom sort: ON".

Step 0x03: D MySQL (3306)

Cheatsheet

Action	Description
<code>sudo nmap -sV -sC -p 3306 <target></code>	Nmap scan on MySQL service, will display hostname and version.
<code>mysql -u <user> -p -h <target></code>	Connects to the MySQL server.

Related Articles

- [SQL Commands](#): Queries to interact with MySQL server.

Overview

MySQL is an open-source relational database management system (RDBMS), widely used for web applications.

MariaDB, a popular fork of MySQL, is also open-source and compatible with MySQL.

MySQL stores sensitive data such as passwords, typically in an encrypted form, although plaintext storage is possible.

It uses SQL commands to interact with relational databases, allowing users to query, update, and manage data.

The default port for MySQL is TCP 3306.

Step 0x03: D MySQL (3306)

Cheatsheet



Action	Description
<code>sudo nmap -sV -sC -p 3306 <target></code>	Nmap scan on MySQL service, will display hostname and version.
<code>mysql -u <user> -p -h <target></code>	Connects to the MySQL server.

Related Articles

- [SQL Commands](#): Queries to interact with MySQL server.

Overview

MySQL is an open-source relational database management system (RDBMS), widely used for web applications.

MariaDB, a popular fork of MySQL, is also open-source and compatible with MySQL.

MySQL stores sensitive data such as passwords, typically in an encrypted form, although plaintext storage is possible.

It uses SQL commands to interact with relational databases, allowing users to query, update, and manage data.

The default port for MySQL is TCP 3306.

Step 0x03: D MySQL (3306)

Cheatsheet



Action	Description
<code>sudo nmap -sV -sC -p 3306 <target></code>	Nmap scan on MySQL service, will display hostname and version.
<code>mysql -u <user> -p -h <target></code>	Connects to the MySQL server.

Related Articles

- [SQL Commands](#): Queries to interact with MySQL server.

Overview

MySQL is an open-source relational database management system (RDBMS), widely used for web applications.

MariaDB, a popular fork of MySQL, is also open-source and compatible with MySQL.

MySQL stores sensitive data such as passwords, typically in an encrypted form, although plaintext storage is possible.

It uses SQL commands to interact with relational databases, allowing users to query, update, and manage data.

The default port for MySQL is TCP 3306.

Step 0x03: D MySQL (3306)

Cheatsheet

Action	Description
<code>sudo nmap -sV -sC -p 3306 <target></code>	Nmap scan on MySQL service, will display hostname and version.
<code>mysql -u <user> -p -h <target></code>	Connects to the MySQL server.

Related Articles

- [SQL Commands](#): Queries to interact with MySQL server.

Overview

MySQL is an open-source relational database management system (RDBMS), widely used for web applications.

MariaDB, a popular fork of MySQL, is also open-source and compatible with MySQL.

MySQL stores sensitive data such as passwords, typically in an encrypted form, although plaintext storage is possible.

It uses SQL commands to interact with relational databases, allowing users to query, update, and manage data.

The default port for MySQL is TCP 3306.

Step 0x03: D MySQL (3306)

Cheatsheet

Action	Description
<code>sudo nmap -sV -sC -p 3306 <target></code>	Nmap scan on MySQL service, will display hostname and version.
<code>mysql -u <user> -p -h <target></code>	Connects to the MySQL server.

Related Articles

- [SQL Commands](#): Queries to interact with MySQL server.

Overview

MySQL is an open-source relational database management system (RDBMS), widely used for web applications.

MariaDB, a popular fork of MySQL, is also open-source and compatible with MySQL.

MySQL stores sensitive data such as passwords, typically in an encrypted form, although plaintext storage is possible.

It uses SQL commands to interact with relational databases, allowing users to query, update, and manage data.

The default port for MySQL is TCP 3306.

Step 0x03: Docum

Command	Description
<code>ssh ... -L <LPORT>:127.0.0.1:<RPORT></code>	(Attack) If you have SSH access, it's often easiest to use this method. Once set up, you can access the remote service locally.
<code>chisel server -v --socks5 --reverse -p <chisel-server-port></code>	(Attack) Starts a reverse Chisel server. The output will include a fingerprint, which is required for setting up the client.
<code>chisel client --fingerprint <fingerprint> <attack-ip>:<chisel-server-port> R:<proxy-port>:<target-ip>:<target-port></code>	(Proxy) Establishes a Chisel client that connects back to the attacker's server. The Chisel binary must be transferred to the proxy host. By default, this opens port 1080 on the server unless specified otherwise.

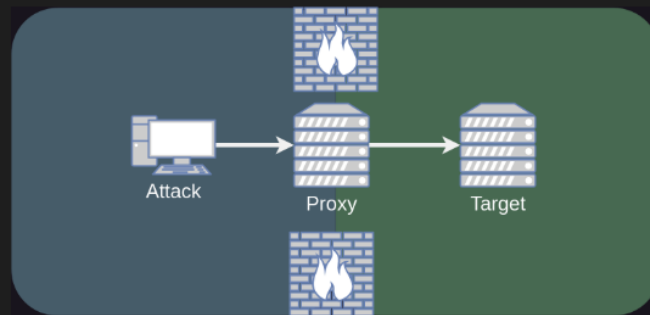
Note

After establishing the connection, you can access <target-ip>:<target-port> via <attack-ip>:<proxy-port>. No need for proxychains.

Overview

Local port forwarding securely tunnels a port from the local machine to a remote server, allowing access to remote services as if they were running locally.

With this method, a specific port on the local machine is forwarded to a corresponding port on the remote server. This is particularly useful for accessing individual services (e.g., SSH, databases) without exposing or interacting with other ports or hosts on the network like with [Dynamic Port Forwarding](#).



Network setup. A port of the target will be forwarded to a port of the attacker's.

Step 0x03: Docu

Related Articles

- [Spidering SMB Shares](#): Using NetExec to look for files in shares.
- [Interacting from Windows](#): Using Powershell and CMD to interact with SMB shares.
- [RpcClient](#): Using the RpcClient CLI tool for enumeration.

Overview

Server Message Block (SMB) is a network protocol that enables file and printer sharing, as well as communication between devices on a local network.

Although SMB is most commonly used in Microsoft Windows environments, it is also supported by Linux and macOS, making it cross-platform.

The main purpose of SMB is to facilitate the sharing of files, printers, and other resources across a network. It allows users to access and interact with shared data and devices seamlessly.

An SMB server can share arbitrary parts of its local file system as shares, and access control is enforced using Access Control Lists (ACLs).

SMB makes use of NetBIOS, although they are distinct protocols. NetBIOS is a session layer service for local network communication. While modern SMB functions without it, NetBIOS over TCP (NBT) is often enabled for backward compatibility.

The default ports for SMB are 139 (NetBIOS) and 445 (Direct SMB).

Authentication

SMB authentication occurs during the connection establishment, where the client and server negotiate the protocol dialect. The client proves its identity through challenge-response mechanisms, such as NTLM or Kerberos, during the session setup. If the authentication is successful, the user is granted access to the shared resources based on their credentials.

SMB can be configured to allow null sessions, meaning no authentication is required for certain access. However, this can expose sensitive data, including credentials, and may be vulnerable to exploits like EternalBlue, especially on outdated versions of SMB.

Step 0x03: Docu

File Upload Vulnerabilities

Related Articles

- [Web Shells](#): If files can be uploaded to exposed directories, a web shell may enable RCE.
- [Local File Inclusion \(LFI\)](#): An LFI vulnerability can enable the execution of uploaded files.
- [XML External Entity \(XXE\)](#): Another vector to consider if arbitrary file upload is not possible.

Overview

File Upload Vulnerabilities are security flaws that arise when a web application improperly handles user-uploaded files, allowing attackers to upload malicious files that can compromise the system.

The primary goal of these attacks is to either inject a malicious file or overwrite an existing one to achieve Remote Code Execution (RCE).

Methodology

1. Upload a normal file

Before anything else, try uploading a harmless file of the type the application expects.

Look to understand what the application does with the file. Does it store it somewhere and if so, where? Does it process the file in some other way?

2. Webshell upload

Assuming the file is uploaded somewhere and you know where, try uploading a [webshell](#) of the appropriate language (e.g. [.php](#), [.asp](#)).

We are going for the low hanging fruit here. If it works, good. But it's likely that there are [mitigations](#) in place to prevent this. The next steps aim to bypass these [mitigations](#).

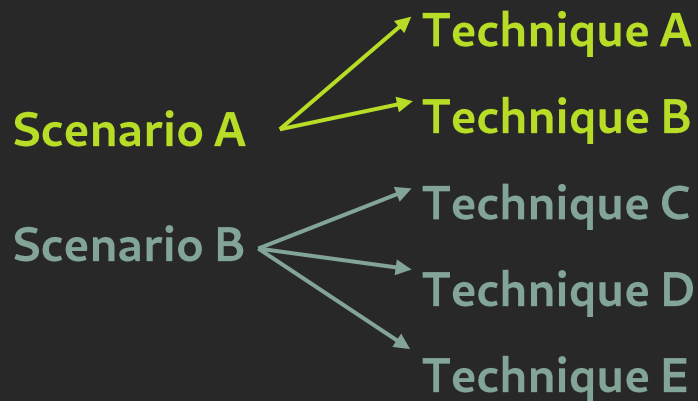
3. Client Side Bypass

It's possible that there are [client side verifications](#) (i.e. JavaScript) preventing the file

BRM

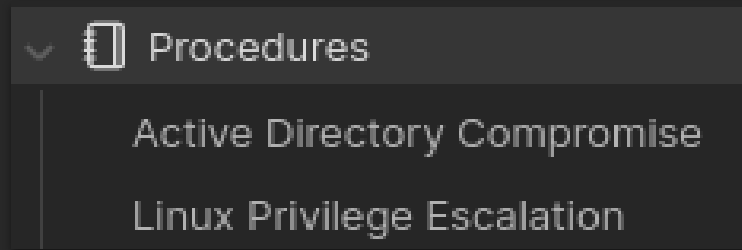
Step 0x04: Create Procedures

- Listing techniques alone is not enough
- Must map scenarios to techniques
- Establish standard procedures



Step 0x04: Create Procedures

- Scenarios are commonly faced situations with a clear procedure
- Examples:
 - Web Application Enumeration
 - Linux / Windows Privilege Escalation
 - Active Directory Foothold
- Procedures are just checklists
- Should be on separate section



Step 0x04: Crea

Linux Privilege Escalation

Global

- ☐ Identify the Linux distribution and Kernel version
- ☐ Check for credentials in web application configuration files
- ☐ Check interesting directories (e.g. `/opt` , `/var/mail` , etc.)
- ☐ Check capabilities
- ☐ Check if `sudo` version is vulnerable ([CVE-2023-22809](#))
- ☐ Internal Nmap scan
- ☐ Check PwnKit
- ☐ Check LogRotate
- ☐ Monitor processes. Look for anything interesting.
- ☐ Look for writable Docker socket files.
- ☐ Look for Tmux sessions that can be hijacked
- ☐ Check for NFS shares with `no_root_squash` enabled
- ☐ Check kernel exploits (e.g. DirtyCow, DirtyPipe)
- ☐ Listen to traffic using TcpDump. Any cleartext credential?

Per User

- ☐ Check which groups user belongs to
- ☐ Check `sudo` rights
- ☐ Check for environment variables
- ☐ Look for ssh keys on home directory
- ☐ Check for hidden files in home directory
- ☐ Check history files on home directory
- ☐ Enumerate SUID / GUID binaries
- ☐ Check cronjobs
- ☐ Try to read other user's home directory (`.ssh/id_rsa` , `.bash_history` , etc.)
- ☐ Try using user's password for other users
- ☐ Run `linpeas.sh`

Active Directory Compromise

1. Live Host Enumeration

- ☐ Conduct a ping sweep on the IP range
- ☐ Use NetExec on the IP range (better information)
- ☐ Use Responder to catch IP addresses

Note

Be sure to properly understand the role of each host. Do your service enumeration.

2. User Enumeration

With foothold

- ☐ Get user list via SMB

Without foothold

- ☐ Attempt to get user list via SMB Null Authentication
- ☐ Attempt to get user list via LDAP Anonymous Bind
- ☐ Attempt to get user list via RPCClient
- ☐ Attempt to get user list via RID brute-forcing
- ☐ Attempt to get user list via Kerbruting

Caution

Some of these techniques are not guaranteed to discover all users. At least try the SMB, LDAP, RPCClient and RID methods.

3. Get Foothold

- ☐ Find Kerberoastable users from the user list
- ☐ Find ASREProastable users from the user list
- ☐ Use Responder to catch credential hashes
- ☐ Try SMB Null Authentication to pillage SMB shares looking for credentials
- ☐ Get SYSTEM / root on Domain connected host to get a Computer account
- ☐ As a last resource, try password spraying with the user list

Danger

Password spraying can lock accounts due to repeated failed attempts and should be used cautiously.

4. Attacks

- ☐ Use SharpHound to collect data to feed BloodHound
- ☐ Check compromised hosts on BloodHound for outbound attack paths
- ☐ Use NetExec to check for command execution via SMB, WinRM, and RDP for each compromised user.
- ☐ Kerberoast
- ☐ ASREProast
- ☐ Look for credentials in GPOs (gpp_password , autologin)
- ☐ For each compromised user, pillage readable SMB shares for sensitive information
- ☐ For each compromised user, conduct SMB Hash Theft attacks on writable SMB shares
- ☐ Look for passwords in user's description fields
- ☐ Check the DC's SYSVOL SMB share for scripts containing credentials
- ☐ [NoPac](#)
- ☐ [PrintNightmare](#)
- ☐ [PetitPotam](#)
- ☐ Try compromised local administrator hashes on other hosts
- ☐ Try Responder on different hosts
- ☐ Look for users with the PASSWD_NOTREQD field
- ☐ Password spray using previously found passwords

Step 0x04: Create Procedures

- What if the solution is not in your manual?
 - You figure out what the solution is
 - Then you add it to your manual :)
- A field manual is iterative
- Failure is an opportunity for growth

Step 0x04: Create Procedures

- What if the solution is not in your manual?
 - You figure out what the solution is
 - Then you add it to your manual :)
- A field manual is iterative
- Failure is an opportunity for growth

The background of the slide is a dark gray circuit board pattern. It features a grid of thin, light gray lines representing traces, with numerous small black dots representing solder points or vias. Scattered across the board are several black rectangular components, each labeled with a white hexadecimal code (e.g., 00023, 00024, 00025, 00026, 00027, 00028, 00029, 0002A, 0002B, 0002C, 0002D, 0002E, 0002F, 00030, 00031, 00032, 00033, 00034, 00035, 00036, 00037, 00038, 00039, 0003A, 0003B, 0003C, 0003D, 0003E, 0003F).

Thank you!

BrunoRochaMoura.com

BRM