TCS HackQuest Season X - Complete 10-Day Preparation Guide

Introduction

Welcome to the comprehensive preparation guide for **TCS HackQuest Season X**. This guide is specifically designed to help you prepare for the competition within a tight **10-day timeline** using your **macOS** environment. With proper dedication and following this structured approach, you can build the essential skills needed to compete effectively in all three categories: Beginner, Intermediate, and Expert level challenges [1] [2] [3].

Understanding TCS HackQuest

Contest Structure

TCS HackQuest is based on the **Capture the Flag (CTF)** format where participants solve cybersecurity challenges to capture flags. The competition consists of three rounds [1] [2]:

Round 1 (Online - 6 hours)

- Challenge-based test with three difficulty levels
- Beginner: Fundamental cybersecurity concepts, basic tools
- Intermediate: Advanced tools, AI/ML concepts, web app security
- Expert: Complex challenges requiring deep understanding
- Mandatory detailed report submission with screenshots

Round 2 (Online Proctored)

- Penetration testing on hosted challenges
- Topics: System exploitation, web apps, mobile security, digital forensics, threat hunting, incident response, source code analysis, SOC scenarios
- Camera must remain on throughout

Round 3 (In-Person at TCS Campus)

- Present approach to solving challenges
- Answer jury panel questions
- Top 10 performers compete for prizes

Evaluation Criteria

Your performance is judged on [1] [2]:

1. Number of flags captured
2. Quality of submitted report
3. Approach and methodology used
4. Demonstration of cybersecurity knowledge

Prizes and Opportunities

- Total prizes worth up to ₹5 Lakhs
- Potential job offers (Ninja, Digital, or Prime)
- Certificates of merit for top performers
- Opportunity to work with TCS Cybersecurity Centre of Excellence

10-Day Master Preparation Plan

Day 1: Foundation & Environment Setup (8-10 hours)

Morning Session (4 hours)

Set up your macOS security environment using Homebrew package manager [4] [5]:

```
# Install Homebrew
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/instal

# Install essential tools
brew install nmap wireshark burpsuite python3 git
brew install binwalk exiftool radare2
brew install --cask utm docker visual-studio-code
```

Install UTM virtual machine manager and download Kali Linux ARM64 image for Apple Silicon Macs [4].

Afternoon Session (4-6 hours)

Study CTF fundamentals and understand the TCS HackQuest structure [2] [6] [7]:

- Learn about different CTF challenge types
- Understand flag formats and submission process
- Review OWASP Top 10 vulnerabilities [8]
- Complete 5-10 beginner challenges on PicoCTF platform [6] [7]

Day 2: Web Security Basics (8-10 hours)

Core Topics

SQL Injection [9]

- Understanding database query manipulation
- UNION-based, Boolean-based, Time-based techniques
- Practice on DVWA (Damn Vulnerable Web App)
- Use SQLmap for automated exploitation

Cross-Site Scripting (XSS) [9]

- Stored, Reflected, and DOM-based XSS
- Payload crafting and filter bypassing
- Practice on PortSwigger Academy labs

Command Injection

- OS command execution vulnerabilities
- Bypass filters and WAF evasion techniques

Practical Setup

```
# Install web security tools
brew install sqlmap gobuster
pip install requests beautifulsoup4

# Run vulnerable web apps via Docker
docker run -d -p 80:80 vulnerables/web-dvwa
docker run -d -p 3000:3000 bkimminich/juice-shop
```

Complete 5 PortSwigger Academy labs focusing on SQL injection and XSS [9] .

Day 3: Cryptography & Network Security (8-10 hours)

Cryptography Fundamentals [10] [11]

Classical Ciphers

- Caesar cipher (shift-based substitution)
- Vigenère cipher (polyalphabetic)
- Substitution and transposition ciphers

Modern Cryptography

- AES (Advanced Encryption Standard) - symmetric encryption [12] [13]
- RSA (Rivest-Shamir-Adleman) - asymmetric encryption [11]
- Hash functions: MD5, SHA-1, SHA-256

Network Protocols[14] [15]

- TCP/IP fundamentals
- HTTP/HTTPS differences
- Common port numbers and services
- DNS and DHCP protocols

Practical Tools

```
# Hash cracking
brew install hashcat john-jumbo

# Network analysis
brew install wireshark tcpdump nmap
```

Practice cryptography challenges on CryptoHack platform and complete network analysis exercises using Wireshark[16].

Day 4: Binary Exploitation & Reverse Engineering (8-10 hours)

Linux Command Line Mastery

Essential commands for CTF challenges:

```
# File analysis
strings binary_file
file binary_file
binwalk firmware.bin
exiftool image.jpg

# Permission and privilege escalation
sudo -l
find / -perm -4000 2&gt;/dev/null
getcap -r / 2&gt;/dev/null
```

Reverse Engineering Tools[17] [18]

Ghidra - NSA's open-source disassembler

- Import binary files
- Analyze assembly code
- Identify functions and vulnerabilities

Radare2 - Command-line reverse engineering framework

- Dynamic analysis capabilities
- Debugger integration
- Scripting support

Binary Exploitation Basics

- Buffer overflow vulnerabilities
- Return-oriented programming (ROP)
- Stack and heap exploitation

Practice Platforms

- OverTheWire: Bandit series (Linux basics)
- OverTheWire: Narnia series (binary exploitation)

Day 5: Digital Forensics & Steganography (8-10 hours)

File Carving and Recovery [19]

Tools and techniques:

```
# Steganography detection
steghide extract -sf image.jpg
binwalk -e firmware.bin
foremost -i disk_image.dd

# Metadata extraction
exiftool document.pdf
strings suspicious_file | grep -i flag
```

Steganography [19]

- Image steganography (LSB, metadata)
- Audio steganography
- Text steganography
- Detection tools: Stegsolve, zsteg, steghide

Memory Forensics

- Volatility framework for memory analysis
- Process inspection and memory dumps
- Extracting passwords and encryption keys

Log Analysis

- Understanding system logs
- Event correlation
- Timeline reconstruction

Day 6: Threat Hunting & Incident Response (8-10 hours)

Security Operations Center (SOC) Fundamentals[20] [21]

SIEM Basics (Security Information and Event Management)

- Log aggregation and correlation
- Alert generation and prioritization
- Understanding security events

Common SIEM Platforms[21]

- Splunk Enterprise Security
- ELK Stack (Elasticsearch, Logstash, Kibana)
- Microsoft Sentinel

Threat Hunting Concepts[20]

- Proactive vs reactive security
- Indicators of Compromise (IoCs)
- Threat intelligence integration
- Attack pattern recognition

Incident Response Process[20]

1. Preparation
2. Identification
3. Containment
4. Eradication
5. Recovery
6. Lessons Learned

AI/ML Security Vulnerabilities[22]

Study adversarial attacks on machine learning systems:

- Data poisoning attacks
- Model extraction
- Adversarial examples
- Evasion techniques

Day 7: Full CTF Simulation Day 1 (10-12 hours)

Objectives

- Simulate real competition environment
- Practice time management
- Develop report writing skills

Activities

1. Join active CTF on CTFtime.org or practice on HackTheBox[7]
2. Solve 10-15 challenges across different categories
3. Time-box each challenge to 30-45 minutes[23]
4. Document every step with screenshots
5. Write detailed reports using the provided template

Focus Areas

- Speed and accuracy balance
- Effective tool usage
- Note-taking discipline
- Flag screenshot documentation

Day 8: Full CTF Simulation Day 2 (10-12 hours)

Advanced Challenges

Focus on intermediate and expert level problems:

- Advanced web exploitation (XXE, SSRF, deserialization)
- Privilege escalation techniques[24]
- Complex cryptography challenges
- Multi-stage exploitation scenarios

Report Writing Practice

Perfect your reporting template:

- Clear step-by-step documentation
- Professional formatting
- Screenshot quality and timestamps
- Technical accuracy

Day 9: Weak Areas & Speed Drills (10-12 hours)

Morning Session: Review

- Review all notes and writeups
- Identify weak areas from Days 1-8
- Create personalized cheat sheets

Afternoon Session: Speed Drills

- Set 1-hour timer
- Solve maximum challenges possible
- Focus on pattern recognition
- Build muscle memory for common techniques

Evening Session: Mock CTF

- Full 6-hour simulated competition
- Follow actual contest rules
- Practice report writing under time pressure
- Verify all tools are functioning

Cheat Sheet Creation

Create quick reference guides for:

- Common payloads (SQL injection, XSS)
- Useful command one-liners
- Network port numbers
- Hash identification
- File signature magic numbers

Day 10: Final Review & Rest (4-6 hours)

Morning (2-3 hours)

- Light review of all cheat sheets
- Solve 2-3 easy challenges for confidence
- Verify all tools are working
- Prepare final report templates

Afternoon: Mental Preparation

- Organize your workspace
- Prepare necessary items (charger, water, snacks)
- Review contest rules one final time

- REST - Get adequate sleep before contest day

Pre-Contest Checklist

- [ ] All tools installed and tested
- [ ] Report templates ready
- [ ] Screenshots folder organized
- [ ] Internet connection stable
- [ ] Camera working (for Round 2)
- [ ] TCS reference ID (CT/DT number) noted
- [ ] Contest URL bookmarked
- [ ] Backup power source available

Essential Tools and Resources

macOS-Specific Tools

Native Security Tools

```
# Network utilities
nmap -sV -sC target.com
wireshark # GUI packet analyzer
tcpdump -i en0 # Command-line packet capture

# Web testing
burpsuite # Proxy and web vulnerability scanner
curl -X POST http://target.com/api
wget --spider -r http://target.com

# Cryptography
openssl enc -aes-256-cbc -in file.txt -out file.enc
hashcat -m 0 hash.txt wordlist.txt

# Reverse engineering
radare2 binary_file
ghidra # GUI disassembler
```

Practice Platforms [6] [7] [23]

Beginner-Friendly

- PicoCTF - Educational CTF with tutorials
- OverTheWire - Progressive challenge series
- TryHackMe - Guided learning paths

Intermediate

- HackTheBox - Realistic penetration testing labs

- Root Me - Diverse challenge categories
- VulnHub - Downloadable vulnerable VMs

Advanced

- CTFtime.org - Live competitions
- RingZer0 Team - Advanced challenges
- Pwnable.kr - Binary exploitation focus

Online Resources

Learning Platforms

- PortSwigger Academy - Free web security training
- CryptoHack - Interactive cryptography challenges
- Exploit Education - Binary exploitation tutorials

Reference Sites

- OWASP Top 10 - Common vulnerabilities[8]
- GTFOBins - Unix privilege escalation
- CyberChef - Data encoding/decoding
- RevShells - Reverse shell generator

Essential Wordlists

```
# Download SecLists
git clone https://github.com/danielmiessler/SecLists.git

# Common wordlists location
~/CTF/resources/SecLists/Passwords/
~/CTF/resources/SecLists/Discovery/Web-Content/
~/CTF/resources/SecLists/Fuzzing/
```

Challenge-Specific Strategies

Web Exploitation

Reconnaissance

1. View page source
2. Check robots.txt and sitemap.xml
3. Inspect HTTP headers
4. Directory enumeration with gobuster
5. Parameter fuzzing

Common Vulnerabilities[9] [8]

- SQL Injection
- Cross-Site Scripting (XSS)
- CSRF (Cross-Site Request Forgery)
- SSRF (Server-Side Request Forgery)
- File upload vulnerabilities
- Directory traversal

Burp Suite Workflow

1. Configure browser proxy
2. Intercept and modify requests
3. Use Repeater for testing payloads
4. Intruder for automated attacks
5. Decoder for encoding/decoding

Cryptography Challenges

Approach Strategy[10] [25]

1. **Identify the cipher type**
    - Look for patterns in ciphertext
    - Check character distribution
    - Consider encryption algorithm hints
2. **Common techniques**
    - Frequency analysis for classical ciphers
    - Known plaintext attacks
    - Brute force for small keyspaces
    - Mathematical cryptanalysis for RSA
3. **Useful tools**
    - CyberChef for general decoding
    - RsaCtfTool for RSA attacks
    - John the Ripper for hash cracking
    - Hashcat for GPU-accelerated cracking

Binary Exploitation & Reverse Engineering

Analysis Workflow[17] [18]

1. **Static analysis**

   ```
   file binary
   strings binary | grep flag
   objdump -d binary
   radare2 -A binary
   ```

2. **Dynamic analysis**

   ```
   ltrace ./binary
   strace ./binary
   gdb ./binary
   ```

3. **Ghidra analysis**
   - Decompile to C code
   - Identify main function
   - Look for suspicious functions
   - Track variable flow

Common Vulnerabilities

- Buffer overflow
- Format string bugs
- Use-after-free
- Integer overflow
- Race conditions

Forensics & Steganography

File Analysis Checklist[19]

```
# Basic file information
file suspicious_file
exiftool suspicious_file

# Strings search
strings -n 8 suspicious_file | grep -i flag
strings suspicious_file | grep -E "^[A-Za-z0-9+/=]{20,}$"

# Binwalk for embedded files
binwalk -e firmware.bin
foremost -i disk_image.dd

# Steganography
steghide extract -sf image.jpg
```

```
stegseek image.jpg wordlist.txt
zsteg image.png --all
```

Memory Forensics

```
# Volatility 3
vol3 -f memory.dump windows.info
vol3 -f memory.dump windows.pslist
vol3 -f memory.dump windows.cmdline
vol3 -f memory.dump windows.filescan | grep flag
```

Report Writing Best Practices

Key Components[26] [^144]

1. Metadata Section

- Challenge name and category
- Difficulty level and points
- Date and time taken
- Your credentials

2. Executive Summary

- 2-3 sentences describing the challenge
- High-level approach used
- Key vulnerability exploited

3. Detailed Methodology

- Step-by-step reproduction
- Commands executed with outputs
- Tools used with versions
- Screenshots with timestamps

4. Flag Capture

- Screenshot showing flag
- Timestamp visible
- Challenge context visible
- Clear and high-quality image

5. Technical Analysis

- Root cause of vulnerability
- Why the exploit worked

- Security implications

6. Remediation

- How to fix the vulnerability
- Best practices recommendation
- Secure code examples

Screenshot Requirements

CRITICAL: Every screenshot must include [2]

- The captured flag clearly visible
- Timestamp (system clock or screenshot metadata)
- Challenge name or URL visible
- High resolution (minimum 1920x1080)

Screenshot Tools

- macOS: Cmd+Shift+4 for selection capture
- Annotate with Preview or Skitch
- Name format: `challenge_category_YYYYMMDD_HHMMSS.png`

Report Formatting

Document Structure

- Use consistent heading hierarchy
- Include table of contents for long reports
- Number all pages
- Use code blocks for commands
- Highlight key findings

File Submission

- Save as PDF format
- File naming: `HackQuest_YourName_ChallengeName.pdf`
- Check file size limits
- Verify PDF readability before submission

Time Management During Competition

6-Hour Strategy

First Hour (0:00 - 1:00)

- Read ALL challenge descriptions
- Categorize by difficulty
- Identify "easy wins"
- Plan attack order

Hours 2-4 (1:00 - 5:00)

- Solve beginner challenges first (quick points)
- Document each solution immediately
- Take screenshots with timestamps
- Move to intermediate challenges

Hour 5 (5:00 - 6:00)

- Focus on challenges close to solution
- Abandon stuck challenges
- Verify all flags are captured
- Start report compilation

Final 30 Minutes (5:30 - 6:00)

- Finalize all reports
- Verify screenshot quality
- Upload reports before deadline
- Keep backup copies

Challenge Selection Strategy[6] [23]

Priority Order

1. Beginner challenges you're confident about
2. Intermediate challenges in your strong areas
3. Expert challenges with partial solutions
4. Completely new challenge types (learning opportunity)

Time Boxing[23]

- Set 30-minute limit per beginner challenge
- Set 45-minute limit per intermediate challenge
- Set 60-minute limit per expert challenge
- If stuck, move on and return later

Common Pitfalls to Avoid

Technical Mistakes

1. **Not reading challenge description carefully**
    - Missing important hints
    - Misunderstanding requirements
2. **Rabbit holes**[23]
    - Spending too long on one approach
    - Not recognizing dead ends early
3. **Tool dependency**
    - Over-relying on automated tools
    - Not understanding manual exploitation
4. **Poor documentation**
    - Missing screenshots
    - Incomplete step documentation
    - No timestamp evidence

Report Writing Errors[2]

1. **Plagiarism**
    - Copying other writeups
    - Results in immediate disqualification
2. **Missing mandatory sections**
    - No screenshots
    - No step-by-step approach
    - Incomplete technical details
3. **Late submission**
    - Uploading after deadline
    - Not verifying upload success
4. **File format issues**
    - Exceeding size limits
    - Corrupted PDF files
    - Unreadable screenshots

Mental Preparation and Well-being

Before the Contest

Physical Preparation

- Get 7-8 hours of sleep
- Eat nutritious breakfast
- Stay hydrated throughout
- Avoid excessive caffeine

Mental Preparation

- Review cheat sheets calmly
- Trust your preparation
- Accept that you won't solve everything
- Focus on doing your best

During the Contest

Stay Focused

- Take short breaks every 90 minutes
- Stretch and move around
- Maintain positive mindset
- Don't panic if stuck

Manage Stress

- Deep breathing if anxious
- Skip difficult challenges temporarily
- Focus on progress, not perfection
- Remember it's a learning experience

Advanced Tips from CTF Veterans

Reconnaissance Techniques

Web Challenges

```
# Quick recon
curl -I http://target.com
nikto -h http://target.com
gobuster dir -u http://target.com -w wordlist.txt

# Source code review
```

```
wget -r -np -k http://target.com
grep -r "flag" downloaded_source/
```

Network Challenges

```
# Port scanning
nmap -sV -sC -p- target.com
nmap --script vuln target.com

# Service enumeration
nc target.com 1234
telnet target.com 80
```

Automation Scripts

Python Template for CTF

```
import requests
import re
from pwn import *

# Configuration
TARGET_URL = "http://challenge.com"
SESSION = requests.Session()

# Exploitation
def exploit():
    payload = "malicious_input"
    response = SESSION.post(TARGET_URL, data={"input": payload})
    flag = re.search(r'TCS\{[^}]+\}', response.text)
    if flag:
        print(f"Flag found: {flag.group()}")
        return flag.group()
    return None

if __name__ == "__main__":
    exploit()
```

Privilege Escalation Quick Wins[24]

Linux

```
# SUID binaries
find / -perm -4000 2&gt;/dev/null

# Sudo permissions
sudo -l

# Capabilities
getcap -r / 2&gt;/dev/null
```

```
# Writable /etc/passwd
ls -la /etc/passwd

# Kernel exploits
uname -a
searchsploit kernel 5.4.0
```

Windows

```
# Unquoted service paths
wmic service get name,pathname,startmode | findstr /i "auto" | findstr /i /v "C:\Windows\

# Always Install Elevated
reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
```

Resources Compilation

Documentation to Review

Must-Read Before Contest

- OWASP Top 10 Web Application Security Risks
- MITRE ATT&CK Framework basics
- Common Weakness Enumeration (CWE) Top 25
- NIST Cybersecurity Framework overview

Video Tutorials

Recommended Channels

- John Hammond (CTF walkthroughs)
- IppSec (HackTheBox writeups)
- LiveOverflow (Binary exploitation)
- Computerphile (Theory concepts)

Books (Quick Reference)

Essential Reading

- "The Web Application Hacker's Handbook" (selected chapters)
- "Penetration Testing: A Hands-On Introduction to Hacking" (Weidman)
- "Practical Malware Analysis" (first 3 chapters)

Final Checklist

Day Before Contest

- [ ] All tools installed and tested
- [ ] Virtual machines configured
- [ ] Report template prepared
- [ ] Screenshot folder created
- [ ] Internet connection verified
- [ ] Workspace organized
- [ ] Snacks and water ready
- [ ] Phone on silent mode
- [ ] Backup plan for technical issues

Contest Day Morning

- [ ] Good night's sleep (7-8 hours)
- [ ] Nutritious breakfast
- [ ] All devices charged
- [ ] Contest URL accessible
- [ ] Login credentials ready
- [ ] Camera working (for Round 2)
- [ ] Arrive early (for in-person rounds)

During Contest

- [ ] Read all challenges first
- [ ] Prioritize by confidence level
- [ ] Take screenshots immediately
- [ ] Document each step
- [ ] Manage time with timer
- [ ] Upload reports before deadline
- [ ] Verify successful submission

Conclusion

Success in TCS HackQuest requires a combination of **technical skills**, **time management**, **documentation discipline**, and **mental preparation**. This 10-day intensive preparation plan provides you with a structured pathway to develop all necessary competencies.

Remember these key principles:

1. **Fundamentals matter** - Master the basics before attempting advanced challenges
2. **Practice deliberately** - Focus on understanding, not just solving
3. **Document everything** - Your report quality matters as much as flag capture
4. **Manage time wisely** - Don't get stuck on single challenges
5. **Stay calm** - Anxiety reduces performance; confidence improves it
6. **Learn continuously** - Every challenge teaches something valuable

The competition is designed to test not just your hacking skills, but also your ability to think creatively, work under pressure, and communicate technical concepts clearly. Even if you don't win, the skills you develop and the experience you gain will be invaluable for your cybersecurity career.

Best of luck with TCS HackQuest Season X! May your preparation be thorough, your execution be flawless, and your flags be many!

This guide is based on official TCS HackQuest documentation and best practices from the cybersecurity community. All information is current as of November 2025.

References

[1] TCS HackQuest Season 10 Official Guidelines
[2] TCS HackQuest Preparation Resources
[3] InfoSec Write-ups: TCS HackQuest Season 9 Experience
[6] CTF Hacking Competitions: Preparation Guide
[4] Kali Linux Installation on Mac M1/M2/M3 via UTM
[7] CTF Platforms and Practice Resources
[23] Reflare CTF Starter's Guide
[5] Kali Linux Penetration Testing Tools
[9] Web Application Security: SQL Injection and XSS
[17] Reverse Engineering with Ghidra
[16] Wireshark Penetration Testing Tutorial
[10] Cryptography Basics for CTF
[11] RSA Encryption for Cybersecurity Novices
[8] OWASP Mobile Top 10 2024
[19] Digital Forensics and Steganography
[25] CTF Cryptography Challenges Guide
[18] Malware Analysis with AI and Radare2
[20] Incident Response Process: SOC Guide
[22] Adversarial Attacks on AI Systems
[21] SOC Tools Selection Guide 2025
[26] CTF Write-up Best Practices
[24] Windows Privilege Escalation Techniques
[^144] Incident Response Report Template
[27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [58] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95] [96] [97] [98] [99] [100] [101] [102] [103] [104] [105] [106] [107] [108] [109] [110] [111] [112] [113] [114] [115] [116] [117] [118] [119] [120] [121] [122] [123] [124] [125] [126] [127] [128] [129] [130] [131] [132] [133] [134] [135] [136] [137] [138] [139] [140] [141] [142]

⁂

1. https://offcampusjobs4u.com/tcs-hackquest-2025/
2. https://onlinestudy4u.in/crack-tcs-hackquest-season-9/
3. https://infosecwriteups.com/tcs-hackquest-season-9-53ccf2017575
4. https://www.youtube.com/watch?v=bcaF1OSivYI
5. https://ijits-bg.com/sites/default/files/archive/2024(vol.16)/No2/contents/2024-N2-10.pdf
6. https://www.exam-labs.com/blog/ctf-hacking-competitions-how-to-get-ready-and-succeed
7. https://snyk.io/articles/ctf/platforms-practice/
8. https://owasp.org/www-project-mobile-top-10/2023-risks/
9. https://www.nature.com/articles/s41598-023-48845-4
10. https://digibug.ugr.es/bitstream/10481/74740/1/11-24.pdf
11. http://arxiv.org/pdf/2308.02785.pdf
12. https://ijsrst.com/IJSRST52310245
13. https://www.nature.com/articles/s41598-025-01315-5
14. https://rsglobal.pl/index.php/ijite/article/view/2836
15. https://www.simplilearn.com/tutorials/cyber-security-tutorial/what-is-tcp-ip-model
16. https://www.youtube.com/watch?v=v8ThuHceRfM
17. https://vickieli.dev/binary exploitation/intro-to-reverse-engineering/
18. https://arxiv.org/html/2504.07574
19. https://www.sifs.in/blog-details/steganography-in-digital-forensics
20. https://www.netwitness.com/blog/incident-response-process/
21. https://www.dropzone.ai/resource-guide/soc-tools-buyers-guide-2025
22. https://www.sentinelone.com/cybersecurity-101/cybersecurity/adversarial-attacks/
23. https://reflare.com/research/the-starters-guide-to-strengthening-skills-with-ctf
24. https://www.hackthebox.com/blog/writing-incident-response-report-template
25. https://www.scribd.com/document/784482802/Getting-Started-With-Ctf-Challenges-a-Comprehensive-Guide-for-Beginners
26. https://www.hackthebox.com/blog/intro-to-mobile-pentesting
27. https://academic.oup.com/ndt/article/doi/10.1093/ndt/gfaf116.1729/8296506
28. https://www.tcshackquest.com/about
29. https://arxiv.org/abs/2210.07465
30. http://ieeexplore.ieee.org/document/8170097/
31. https://ieeexplore.ieee.org/document/11133632/
32. https://ijmserh.com/admin/pdf/2025/7/74_AI-Generated Code Detection and Quality Security Assessment Framework.pdf
33. https://dl.acm.org/doi/10.1145/3475716.3475781
34. https://ieeexplore.ieee.org/document/10766004/
35. https://link.springer.com/10.1007/s13198-024-02262-6

36. https://www.semanticscholar.org/paper/4951c56bb240123c885ae27b1bfd6d720bcfd317
37. https://ieeexplore.ieee.org/document/10371505/
38. https://www.tandfonline.com/doi/full/10.1080/23311916.2017.1335470
39. https://www.coursejoiner.com/internship/tcs-hackquest-season-10-college/
40. https://arxiv.org/vc/arxiv/papers/2103/2103.08010v1.pdf
41. https://arxiv.org/abs/2105.03346
42. https://arxiv.org/pdf/2401.12443.pdf
43. https://arxiv.org/pdf/2310.00205.pdf
44. https://arxiv.org/pdf/2202.13026.pdf
45. https://arxiv.org/pdf/2210.07465.pdf
46. https://arxiv.org/ftp/arxiv/papers/2107/2107.07300.pdf
47. https://www.aptori.com/blog/the-difference-between-source-code-analysis-sca-and-sast
48. https://www.prutech.com/in/the-crucial-role-of-ethical-hackers-in-protecting-our-digital-landscapes/
49. https://www.veracode.com/security/static-code-analysis/
50. https://meridian.allenpress.com/jgb/article/19/2/i/499655/COMMUNICATION-SKILLS-AND-REPORT-WRITING-FOR
51. https://www.jmaj.jp/detail.php?id=10.31662/jmaj.2023-0045
52. https://vidyajournal.org/index.php/vidya/article/view/115
53. https://www.ssrn.com/abstract=3549317
54. https://www.mdpi.com/2304-6775/8/1/10
55. https://ejournal.upi.edu/index.php/IJOMR/article/view/60570
56. https://ieeexplore.ieee.org/document/10391222/
57. https://www.ijfmr.com/research-paper.php?id=18590
58. http://www.magonlinelibrary.com/doi/10.12968/bjha.2023.17.10.376
59. https://www.sciencedirect.com/science/article/pii/S1084804524001814
60. https://www.aasv.org/shap/issues/v29n6/v29n6p327.html
61. https://pmc.ncbi.nlm.nih.gov/articles/PMC9163716/
62. https://bjsm.bmj.com/lookup/doi/10.1136/bjsports-2021-105058
63. https://pmc.ncbi.nlm.nih.gov/articles/PMC43690/
64. https://arxiv.org/pdf/2206.08971.pdf
65. https://pmc.ncbi.nlm.nih.gov/articles/PMC232730/
66. https://www.i-jmr.org/2025/1/e51598
67. https://pmc.ncbi.nlm.nih.gov/articles/PMC11995452/
68. https://pmc.ncbi.nlm.nih.gov/articles/PMC8593010/
69. https://pequalsnp-team.github.io/cheatsheet/writing-good-writeup
70. https://www.sandipuniversity.edu.in/blog/the-role-of-ethical-hacking-in-protecting-digital-assets/
71. https://delinea.com/blog/windows-privilege-escalation
72. https://www.youtube.com/watch?v=Cd_3ZvYdToM
73. https://arxiv.org/pdf/2408.16033.pdf

74. https://digitalcommons.uri.edu/cgi/viewcontent.cgi?article=1084&context=theses
75. https://www.semanticscholar.org/paper/9b44e89acd51970b8a6d96999564f896d84db367
76. https://digital.library.unt.edu/ark:/67531/metadc2356157/
77. https://ccsenet.org/journal/index.php/ies/article/view/0/52266
78. https://www.mdpi.com/1996-1944/14/14/4023
79. https://academic.oup.com/femsle/article/doi/10.1093/femsle/fnac111/6852944
80. https://cvirendovasc.springeropen.com/articles/10.1186/s42155-025-00529-y
81. https://jklst.org/index.php/home/article/view/43
82. https://bjui-journals.onlinelibrary.wiley.com/doi/10.1111/bju.70014
83. https://he02.tci-thaijo.org/index.php/ramajournal/article/view/267918
84. https://www.tandfonline.com/doi/full/10.1080/10588167.2018.1496754
85. https://digitalcommons.uri.edu/cgi/viewcontent.cgi?article=1454&context=theses
86. http://arxiv.org/pdf/2408.08926.pdf
87. https://arxiv.org/pdf/2107.12566.pdf
88. https://linkinghub.elsevier.com/retrieve/pii/S2666166724007159
89. https://arxiv.org/pdf/2308.10443.pdf
90. https://arxiv.org/pdf/1708.05844.pdf
91. https://arxiv.org/pdf/2101.01421.pdf
92. https://arxiv.org/html/2502.10931v1
93. http://arxiv.org/pdf/2101.02108.pdf
94. https://peerj.com/articles/cs-25.pdf
95. https://www.mdpi.com/2073-431X/13/3/81
96. https://arxiv.org/pdf/2502.16730.pdf
97. http://arxiv.org/pdf/2406.08242.pdf
98. http://arxiv.org/pdf/2308.06782.pdf
99. https://astesj.com/?download_id=13068&smd_process_download=1
100. https://arxiv.org/pdf/0912.3970.pdf
101. https://www.mdpi.com/2079-9292/11/19/2991/pdf?version=1663759457
102. https://arxiv.org/pdf/1901.09286.pdf
103. https://arxiv.org/pdf/2212.11449.pdf
104. http://arxiv.org/pdf/2412.01778.pdf
105. https://www.rootshellsecurity.net/7-pentesting-tools-you-should-know-about/
106. https://link.springer.com/10.1007/s11277-024-11353-3
107. https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJISP.340728
108. http://thesai.org/Publications/ViewPaper?Volume=15&Issue=12&Code=ijacsa&SerialNo=1
109. http://arxiv.org/pdf/2107.02846.pdf
110. https://ijarcs.info/index.php/Ijarcs/article/view/7181
111. https://ieeexplore.ieee.org/document/10924049/

112. https://journals.scholarpublishing.org/index.php/ASSRJ/article/view/17764
113. https://ieeexplore.ieee.org/document/10846266/
114. https://www.semanticscholar.org/paper/f189d8a1bc812bb59387b56185a2209eab0acaf2
115. http://arxiv.org/pdf/2412.13352.pdf
116. https://rajpub.com/index.php/ijct/article/download/2745/2669
117. https://www.aclweb.org/anthology/2020.acl-main.117.pdf
118. https://downloads.hindawi.com/journals/complexity/2020/8316454.pdf
119. https://arxiv.org/ftp/arxiv/papers/1307/1307.7786.pdf
120. https://www.mdpi.com/1424-8220/23/6/3287/pdf?version=1679315363
121. http://www.ijimai.org/journal/sites/default/files/IJIMAI20101_3_6.pdf
122. https://www.semanticscholar.org/paper/0e6f49d370aa18dd555ad9bcee808eacfb9d1688
123. https://www.semanticscholar.org/paper/0db437a5eeb1013b8d9e171f09788443d9782a74
124. https://arxiv.org/html/2503.23466
125. https://www.semanticscholar.org/paper/6f6ac7b6394588ac849a134a10b130f5f4c61b2c
126. https://penerbit.unimap.edu.my/index.php?option=com_quix&view=page&preview=true&id=118
127. https://www.semanticscholar.org/paper/8937a77d05e6c4e9a155017b5d65514b021e2eb1
128. https://www.semanticscholar.org/paper/0da0d19c00e5f2461d0be5f8e634e28aaa2efd5b
129. https://www.semanticscholar.org/paper/9d1becc1197222a85d6e7ded17224984ad2a6b78
130. https://www.semanticscholar.org/paper/e762e1673e623bb6426f3c8df06fc7915a3cd9fd
131. https://www.semanticscholar.org/paper/3494382a14ea7a11e530bb423974ad1d3a6f7233
132. https://wjaets.com/sites/default/files/WJAETS-2024-0098.pdf
133. https://wjaets.com/sites/default/files/WJAETS-2024-0127.pdf
134. https://arxiv.org/html/2411.09895v1
135. https://arxiv.org/pdf/1911.02984.pdf
136. https://arxiv.org/ftp/arxiv/papers/2308/2308.01713.pdf
137. https://ijcsrr.org/wp-content/uploads/2024/02/41-1702-2024.pdf
138. https://www.ijtsrd.com/papers/ijtsrd8312.pdf
139. http://arxiv.org/pdf/1912.13371.pdf
140. https://engrxiv.org/preprint/download/387/967
141. https://www.a1.digital/knowledge-hub/tcp-ip-explained/
142. https://www.semanticscholar.org/paper/be0d62a6b097a5a82995dffdc00d391c96a85815