CHANDIGARH UNIVERSITY, GHARUAN

Department of Computer Science and Engineering

INTERNSHIP REPORT

Submitted in Partial Fulfilment of the Requirements for the
Degree of
Bachelor of Engineering (B.E.) in Computer Science and Engineering

Internship Program (20 Sep-25 Dec):

Cybersecurity Internship Program – Shentinelix Sphere Pvt Ltd

Task 22
H4cked, C2C Carnage, CCT 2019

Internship Report Submitted By:
Vikash Upadhyay

Table of Contents

Executive Summary

The practical labs “H4cked”, “Carnage”, and “CCT2019” on TryHackMe provided focused, hands-on experience in network forensics and incident response using packet capture (PCAP) analysis and related tooling (Wireshark, tshark, grep, etc.). Across the three rooms I traced attacker behavior from initial compromise (phishing/malicious document) through command-and-control activity, privilege escalation, and persistence. Each room emphasized different forensic skills — H4cked concentrated on reconstructing an attack timeline and recovering artifacts from a single compromise; Carnage emphasized analyzing malicious network traffic from an enterprise intrusion; CCT2019 presented a longer, multi-step, file-recovery challenge that tested extraction and binary execution techniques in-depth.
These exercises reinforced the importance of methodical investigation: identifying indicators of compromise (IoCs) in PCAPs, extracting transferred files, reconstructing URLs and commands, and correlating network evidence with likely host actions. They also highlighted operational safety: avoid contacting live malicious domains or executing recovered binaries on production systems. Completing the rooms increased my confidence in using network-forensic workflows to answer concrete questions about who did what, when, and how.

Overall, the combined experience demonstrates readiness to contribute to Blue Team investigations — from triage and evidence collection through analysis and remediation recommendations — and provides a solid foundation for deeper work in endpoint analysis, threat hunting, and building detection rules.

Introduction

Network packet captures are often the richest single source of evidence during an intrusion investigation because they can show remote interactions, data exfiltration, and staged payload transfers even when endpoint logs are limited or wiped. The three TryHackMe PCAP rooms used curated traffic captures to simulate realistic incident scenarios and to teach practical analysis strategies in a safe, repeatable environment. The goal of this report is to summarize duties performed, accomplishments, skills obtained, challenges faced, and overall conclusions for each exercise.

My approach in each room followed a repeatable workflow: (1) initial triage — open the PCAP in Wireshark and get a high-level view (protocol distribution, top talkers, and timestamps); (2) indicator discovery — search for suspicious strings, credentials, and transferred artifacts (HTTP POSTs, FTP sessions, SMB, DNS queries); (3) extraction — reconstruct files transferred over the wire and analyze their metadata; (4) timeline assembly — correlate network events to build an attacker timeline; (5) remediation suggestions and detection rule ideas. This method ensured consistent coverage across H4cked, Carnage, and CCT2019.

The following sections detail the company context (TryHackMe), the duties I performed within the rooms, specific accomplishments, skills learned, challenges I overcame, and a concise conclusion with recommended next steps for continuing learning or operationalizing findings.

Overview of the Company

TryHackMe is an online hands-on cybersecurity training platform that delivers interactive "rooms" covering offensive and defensive topics. It provides learners with guided tasks, preconfigured virtual machines or AttackBoxes, and scoring to validate practical skill acquisition. The platform is particularly valued by students and early-career security practitioners because it lowers the barrier to hands-on practice through clear learning objectives and incremental tasks.

In these PCAP-focused rooms, TryHackMe supplies curated evidence sets and realistic scenarios (compromised workstation, malicious Word macro, multi-stage C2) so learners can practice forensic methods without risk to production systems. Community-contributed challenges like the CCT2019 set demonstrate how TryHackMe aggregates real-world-sourced PCAPs (e.g., Brad Duncan captures) and turns them into educational problems that exercise deep technical skills.

As a training provider, TryHackMe balances guided walkthroughs with open-ended challenges so trainees learn both "how" and "why" — how to run the tools and why certain indicators are meaningful. This structure is ideal for building a repeatable investigative mindset that can be transferred to workplace incident response.

Description of Duties

For each room I completed the guided tasks end-to-end and created an evidence-backed timeline for the simulated incident. In H4cked, duties included downloading and inspecting the PCAP in Wireshark, identifying the attacked service (FTP), extracting credentials and transferred artifacts (shell.php), following the reverse-shell traffic to determine commands run by the attacker, and documenting the attacker's escalation steps (python pty spawn, sudo su, rootkit download).

In Carnage, I deployed the provided virtual machine where applicable, loaded the PCAP, and focused on protocol analysis and identifying outbound C2 patterns. Work included isolating suspicious HTTP/HTTPS requests, decoding embedded payloads, reconstructing staged downloads, and enumerating beaconing intervals and domain patterns that would inform detection rules. Tasks also required safe handling of URLs and not interacting with live malicious infrastructure.

For CCT2019, duties were more investigative and multi-step: recover files transferred through fractured streams, reassemble binaries, run safe static analysis (strings, ldd, objdump) in an isolated lab when necessary, and follow the chained clues to reach the final artifact. Across all rooms I documented each investigative step, captured screenshots of key Wireshark findings, and compiled recommended indicators and detection logic.

Acomplishments

H4cked: I fully reconstructed the attack sequence — identified the FTP brute-force (Hydra usage), the successful credential (Jenny), located the uploaded backdoor (shell.php), recovered the reverse-shell source URL, and traced the post-compromise commands (whoami, pty spawn, sudo su). I also identified the GitHub project (Reptile) used to install a rootkit-like persistence mechanism. These findings gave a complete narrative from initial access to persistence.

Carnage: I identified multiple suspicious outbound connections and reconstructed a malicious macro-triggered download chain. I was able to extract transferred payloads, identify a probable C2 hostname pattern, and map beaconing behavior suitable for signature/detection creation. I produced a short list of IoCs (IP addresses, domains, file hashes where recoverable) and suggested Snort/Suricata rule ideas based on observed URI and User-Agent patterns.

CCT2019: I successfully followed the multi-stage hints to recover the first file correctly and progressed through the subsequent steps of the challenge, demonstrating reliable file reassembly from streams and cautious static analysis of recovered binaries. Completing parts of this challenge verified competence in working with complex PCAP puzzles that mirror competition-style forensic problems.

Skills Learned

Technical skills strengthened across the rooms include: advanced Wireshark usage (display filters, follow TCP streams, export objects), PCAP triage heuristics (identify protocol anomalies, uncommon ports, suspicious HTTP verbs and headers), and extraction techniques (reassembling files from HTTP/FTP/SMB, using tshark/export-object). I improved command-line fluency with tools like tshark, tcpflow, netcat, strings, and basic static binary analysis tools.

Methodological skills improved as well: assembling reliable timelines from disparate network events, correlating network evidence to likely host activity, and deriving actionable indicators for signatures and detection rules (e.g., Snort/Suricata). I also practiced safe handling of malicious artifacts by avoiding active network interactions and restricting binary analysis to sandboxed environments.

Soft skills gained include clearer incident-writeup techniques — crafting concise executive summaries, enumerating reproducible steps for triage, and producing prioritized remediation guidance for ops teams (contain, eradicate, detect, recover). These reporting skills make technical findings consumable by stakeholders.

Challenges Faced

Interpreting obfuscated or fragmented data was a recurring challenge, especially in the CCT2019 challenge where files were split across many packets and streams. Recovering the initial file intact required patience and strict attention to ordering and reassembly tools — missing a single chunk can invalidate later steps. Overcoming this required switching between Wireshark, tshark, and tcpflow and validating file integrity at each step.

Another challenge was safely analyzing downloaded or reconstructed payloads without touching live malicious domains. I adhered to best practices: no outbound connections, no execution on host systems, and using isolated VMs or offline static analysis. Determining intent from short command sequences or stripped binaries also required careful inference and conservative conclusions where evidence was incomplete.

Finally, building detection rules from observed traffic required striking a balance between specificity and generality — too-specific rules miss variants, too-broad rules create noise. I practiced writing candidate Snort/Suricata rule strings and suggested telemetry checks (DNS anomaly monitoring, unusual FTP uploads, suspicious User-Agent signatures) while noting the need for tuning in production environments.

Conclusion

The H4cked, Carnage, and CCT2019 rooms together form an excellent practical curriculum for network forensics and incident response. They reinforced a repeatable investigative workflow: triage → extraction → analysis → timeline → remediation. The scenarios improved my technical toolset, forensic reasoning, and reporting capability, making me better prepared to assist in real-world incident investigations.

Recommended next steps: practice endpoint artifact correlation (Windows Event Logs, Sysmon, Linux auditd) to pair network findings with host evidence; build and test detection rules in a controlled environment (use Suricata/Snort + Zeek for richer telemetry); and work through more multi-stage capture challenges to refine file reassembly and binary analysis skills. Finally, capture and document each step of future investigations so the findings can be converted into repeatable runbooks for your SOC.

Acknowledgments

I would like to thank the team at Shentinelix Sphere Pvt. Ltd. for providing access to lab environments, learning resources, and a supportive environment that encouraged practical learning and skill development. Special thanks to the TryHackMe platform and the authors of the Research Methodology room for providing structured, hands-on lessons that significantly enhanced my research and problem- solving skills. Finally, I extend my gratitude to my peers and family for their continuous support and encouragement throughout the internship period.