Two-page Summary

Result

I successfully completed four classic security challenge games on the OverTheWire platform: Natas (Level 0-20), Narnia (Level 0-4), Krypton (Level 0-7), and Leviathan (Level 0-7), totaling 42 security challenge levels conquered. Key achievements include:

1. Diversified Vulnerability Exploitation
    - Implemented HTTP parameter injection (Level 4), cookie forgery (Level 5), and file inclusion vulnerability exploitation (Level 7) in Natas.
    - Completed buffer overflow attacks (Level 0/2/4), environment variable injection (Level 1), and symbolic link permission bypass (Level 3) in Narnia.
    - Appendix Evidence: Narnia Level 2 Buffer Overflow GDB Analysis Screenshot (see Writeups.md)
2. Cryptography Practice Breakthroughs
    - Cracking Krypton's ROT13 (Level 1), the Virginia Cipher (Level 4-5), and Stream Cipher CPA Attacks (Level 6)
    - Appendix Evidence: Python Decryption Script for Krypton Level 6 (see Writeups.md)
3. Most Proud Achievement The stack overflow exploitation (exploit code) for Narnia Level 2 is my proudest achievement:
    - Precisely calculating the 128-byte offset using GDB debugging
    - Using shellcode containing setreuid() to achieve privilege escalation
    - Successfully obtaining a high-privilege shell and capturing the password (result screenshot)

What I did

Time allocation and strategy

1. Early stage (70% of time): Focus on binary vulnerabilities (Narnia series), spending 2 hours per day debugging buffer overflows.
2. Middle stage (20% of time): Rapidly break through web application vulnerabilities (Natas) and cryptography (Krypton).
3. Late stage (10% of time): Clean up Leviathan's file system permission challenges.

Challenge

Setback Management

1. Narnia discovered insufficient permissions after successfully running the shell. After attempting to simulate the program step by step, the problem (euid and ruid) was identified and resolved.
2. When time is limited: prioritize attacks on weak cryptographic systems (such as Krypton) to build confidence.

How I was challenged

1. Binary Vulnerability Depth
    - Challenge: Understanding the EBP/RIP register overwriting mechanism (Narnia Level 4)
    - Breakthrough: Visualizing the stack layout using the GDB command x/300wx $esp
2. Understanding the Essence of Cryptography
    - Previous Misconception: Believing that stream ciphers (such as Krypton Level 6) are absolutely secure
    - Aha Moment: Mastering known-plaintext attacks (CPA) can break weak random streams

3. Self-Reflection on Abilities
    - Strengths: Web vulnerability exploitation (100% completion of the Natas series) and automated script development (Python brute-force scripts)
    - Areas for Improvement: Understanding assembly language (over-reliance on patterned attacks during Narnia debugging)

Future Optimization Directions

1. Prioritize the development of automated tools (e.g., blind injection scripts for Natas Level 15 to save time)
2. Strengthen foundational knowledge: Plan to study Linux binary analysis and deepen understanding of ELF structure
3. Vulnerability defense perspective: Attempt to add vulnerability remediation solutions after an attack (e.g., Stack Canary implantation)
4. Deeply study assembly language knowledge, as this is essential for reverse engineering analysis