DICTATOR
FANCLUB TEAM

2차 합동 프로젝트
기획서(초안)

2025.12.

Blue, Purple, Red and Gray

목차

1. 프로젝트 개요

1.1. 프로젝트명

기관 인프라 구축 및 모의해킹을 통한 취약점 진단

1.2. 프로젝트 배경 및 목적

- 본 프로젝트는 보안에 대한 명확한 개념 이해를 바탕으로 네트워크 및 서버 단의 보안 취약 요소를 식별·제거하고, 안전한 시스템 운영 환경을 구축하는 것을 목적으로 한다.
- 이를 위해 모의해킹을 통해 구성된 서버의 취약점을 분석하고, 분석 결과를 보안 정책에 반영하여 통합 보안 시스템을 구축한다.
- 지속적인 점검과 보완 활동을 통해 보안 취약성으로 인한 위험도를 실현 가능한 수준으로 저감하고, 안정적인 보안 운영 체계를 확립하고자 한다.

1.3. 프로젝트 목표 (Goals):

분야	내용	
네트워크	구축	- 고가용성을 위한 회선 이중화 및 Gateway 이중화를 적용한 네트워크 구축 - 대규모 네트워크에서 빠른 경로 탐색, 라우팅 업데이트 최소화를 위해 EIGRP 적용 - 소규모 네트워크에서는 RIP 또는 static 라우팅 적용 - 특정 트래픽에 대한 정책 기반 라우팅 및 접근 제어
	관제	- 네트워크 상에서의 비정상 행위 탐지를 위한 MRTG와 같은 네트워크 장비 모니터링 시스템 구축 - Wireshark를 활용한 패킷 캡처 및 분석으로 비정상 트래픽 탐지
	보안	- 프로토콜별 인증 구성과 Access-List 및 Distribute-List를 활용하여 인가되지 않은 침입을 차단하는 안전한 네트워크 구축 - 외부와 통신하지 않는 사설망 설정을 위해 Port-based NAT 적용
	대응	- Python 기반의 패킷 분석 프로그램 개발 - 패킷 분석 결과를 바탕으로 Access-list 추가 - Port-security를 활용하여 MAC 주소 접근 제어 목록 추가
서버	구축	- 단일 장애점 제거를 위한 서비스별 분산 구축으로 시스템 안정성 및 운영 효율성 극대화 - 독립접 백업 서버 운용을 통한 서비스 가용성 및 무결성 보장

		- nginx reverse proxy + apache backend 구성을 통해 트래픽 부하 분산 및 최적화된 콘텐츠 전달속도 구현
	관제	- Cacti 기반의 웹 GUI 환경을 구축하여 서버 성능 지표(CPU, MEM, Disk)의 실시간 가시성 확보 및 위험 감지 - MRTG를 활용한 네트워크 대역폭 및 장비 상태의 시각화로 트래픽 이상 징후 조기 포착 - 서버 및 서비스별 로그 취합·분석을 통해 비정상적인 접근 시도 및 침해 사고 흔적의 실시간 탐지 및 기록
	보안	- SSL 키인증을 통한 데이터 전송 구간 암호화 및 통신 보안 강화 - Web, DB, Mail 등 각 서비스 별 보안 설정을 통해 안전한 서버 구축
	대응	- iptables 및 ufw 를 활용하여 허용된 신뢰 IP 외 모든 접근을 원천 차단 - NFS에 저장된 최신 백업본을 활용한 서비스 가용성 즉시 회복 - 저장된 로그를 분석하여 공격 경로 파악 및 재발 방지
관제	인프라 점검	- 주요정보통신기반시설 기술적 취약점 분석·평가 상세 가이드의 '상' 항목을 기반으로 Unix 서버, 윈도우즈 서버, 보안 장비, 네트워크장비, PC, DBMS, 웹에 대한 취약점 점검 수행 - 기밀성, 무결성, 가용성에 영향을 미치는 위협요인 파악, 사용자 및 공격자 관점에서의 위협요소 도출 및 대응 방안 제시
	침투테스트 대비	- 외부망에서 접근 가능한 서비스에 대해 공격자 관점의 접근 및 공격 테스트, 정보 유출 및 내부서버망과 내부 사용자망에 대한 침투 가능성 확인
모의해킹	침투테스트	- 내/외부망에서 악의적인 목적을 가진 사용자로 가장하여 모의해킹 수행, 내부 및 외부 사용자에 의한 정보 유출 가능성 확인 - 시라니오 기반 침투 테스트 수행을 통해 보안 위협의 사전 발견 및 대응방안 제시

1.4. 기대효과 (Expected Results)

분야	내용
네트워크	- 네트워크 분산 구축을 통한 데이터 전송 안정성과 가용성 보장
서버	- 서버 및 서비스 분산 구축을 통한 데이터 전송 안정성과 가용성 보장 - 주기적ㅊ인 로그 분석 및 백업을 통한 데이터 손실 위협에 대한 대응 및 복구 보장 - 통합 모니터링 및 로그분석 선제적 위협 탐지 및 즉각 대응 - 화이트리스트기반 접근 제어와 SSL 암호화를 통해 내부 데이터 유출 최소화 - nginx reverse proxy + apache backend를 통한 부하 분산으로 서버 자원의 병목 현상 을 해결하고, 사용자에게 빠르고 쾌적한 웹 응답 속도 제공
관제	- 통합 모니터링 및 로그분석 서버 구축을 통한 통합 관제 수행
모의해킹	

2. 프로젝트 일정 및 역할 (Schedule &Roles)

2.1. 전체 일정 (Timeline)

WBS	세부 업무 (Task)	12월			1월												
		29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13
		월	화	수	목	금	토	일	월	화	수	목	금	토	일	월	화
1. 계획																	
1.1	목표 설정																
1.2	프로젝트 요구사항 분석																
1.3	프로젝트 기획안 작성																
2. 설계																	
2.1	네트워크 설계																
2.2	서버 설계																
2.3	모의해킹 시나리오 설계																
2.4	보안관제/탐지 정책 설계																
3. 구축 및 구현 테스트																	
3.1	네트워크 구축																
3.2	서버 구축																
3.3	인프라 안정성 테스트																
3.4	취약점 진단 및 스캐닝																
3.5	모의해킹																
3.6	공격 탐지 및 로그 분석																
4. 환류																	
4.1	보안대책 수립																
4.2	시정조치 수행																
4.3	시정조치 후 재검증																
4.4	보고서 작성																

2.2. 역할 및 책임 (Roles &Responsibilities)

NO	성명	역할 및 책임
1	이명재	총괄 및 품질 보증활동
2	장혜원	네트워크 총괄
3	정성현	네트워크 환경 구축, 자동화 코드
4	이하은	네트워크 세부 구성 및 구축, 자동화 코드
5	이현지	네트워크 세부구성 및 구축, 자동화 코드 총괄
6	김혜민	서버 총괄
7	김은성	smtp(mail)서버, dns서버, cms(jl, wp)서버
8	유기원	dmbs서버 , 진척관리
9	유희영	snmp서버, bak서버
10	박희진	whd서버(pydio, owncloud, apache), web서버(nginx), client gui서버
11	장지웅	모의해킹 총괄
12		
13		
14		
15		
16		
17	소윤,준희,지형,한얼,혜림,영우	모의해킹
18	김윤영	보안관제 외
19	신우,원재,지원,은비	코드 테스트 외

6. 보안관제

6.1 보안 정책 이행 흐름

- 2026년 주요정보통신기반시설 기술적 점검 가이드를 준수하며 시나리오별 보안 정책을 다음과 같이 수립 및 이행합니다.

1) 정책 수립 및 역할 배분 단계 (Hand-over)

- **기본 정책 수립:** 2026년 주정통 가이드를 기반으로 일반 보안 정책 항목 선정
- **팀별 항목 배분: 서버, 네트워크, 레드팀**에 각각 해당되는 **점검 항목**을 할당
- **결과물 도출**
  - ㅇ **코드 구현:** 주정통에서 각 팀별로 **선별한 항목** 중 **코드 자동화**를 통해 자동 점검이 가능한 항목은 각 팀에서 **주피터 스크립트(Code) 형태**로 구현하여 퍼플 관제팀에 전달
  - ㅇ **가이드라인:** 코드 구현이 시간 상 어렵거나 불가능한 항목은 주정통 PDF파일을 통해 **수동 점검 및 조치** 결과 도출하여 결과물 형태로(PDF 혹은 PNG 캡처 파일) 퍼플팀에 전달

2) 공격 시나리오 분석 및 방어 전략 설계 (Strategy)

- **공격 시뮬레이션:** 레드팀 시나리오 기반의 실전 침투 테스트를 통한 공격 경로 식별
- **방어 정책 도출:** 공격 기법 분석을 통해 시나리오별 맞춤형 탐지/차단 전략 수립
- **정책 문서 제작:** 도출된 방어 정책을 기반으로 서버·네트워크팀에 '보안 정책 문서' 전달 및 침투 테스트 결과를 기반으로 레드팀에 '침투 테스트 보고서' 전달

3) 시나리오 기반 보안 정책 구현 및 적용 (Implementation)

- 3가지 주요 공격 시나리오를 각 팀에 맞춤형 보안 정책 보고서를 전달하여 실제 인프라에 적용합니다.

- **서버팀(System)**
  - ㅇ **로그 분석:** 시나리오별 공격 로그의 발생 경로 및 확인 방법 명시
  - ㅇ **차단 정책:** 네트워크 랩 구성을 바탕으로 서버 단위의 iptables 차단 규칙 적용

- **네트워크팀(Network)**
  - ㅇ **패킷 식별:** 시나리오별 정상/비정상 패킷 식별 기준(Signature) 제공
    -> 네트워크 팀에서 실제 패킷 확인 후 캡처 혹은 덤프로 퍼플팀에 전달
  - ㅇ **ACL 정책:** 네트워크 토폴로지에 따라 최적의 라우터 위치에 ACL(Access Control List) 차단 정책 수립 및 적용

4) 방어 유효성 검증 (Retest)

- **방어 가동 확인:** 시나리오 기반 공격을 수행하여, 새로 적용한 룰이 실제 공격 패킷을 누락 없이 탐지하는지 확인
- **차단 실효성 검토:** 탐지된 공격이 서버(iptables)와 네트워크(ACL) 지점에서 설정한 대로 완벽히 차단되는지 로그를 통해 최종 검증
- **방어 기준 확정:** 이번 시나리오를 통해 수립된 보안 정책이 실전에서 작동함을 확인하고, 이를 새로운 방어 기준으로 확정

5) 지속적 방어 체계 자산화 (Update)

- **실전 탐지 룰 반영:** 검증이 완료된 시나리오별 룰을 관제 시스템의 상시 정책으로 전환하여 실시간 대응력을 확보
- **보안 자산화:** 분석 과정에서 도출된 로그 경로, 패킷 식별 기준, 차단 코드를 데이터베이스화하여 향후 유사 사고 대응 및 정기 점검의 기초 자료로 활용

6.2 기술스택

구분	기능	주요 기술 요소	상세 설명
관제	통합 보안 관제	Security Onion	IDS, 로그 관리, 트래픽 분석 기능이 통합된 보안 관제 플랫폼
	침입 탐지	Snort	네트워크 패킷 분석 및 공격 패턴 매칭을 통한 실시간 침입 탐지
	이벤트 분석	Sguil	탐지된 보안 이벤트의 가시성 확보 및 정밀 분석용 인터페이스
	패킷 분석	Wireshark	네트워크 트래픽 상세 분석을 통한 공격 증거 및 통신 경로 검증
자동화	대응 자동화	Rule 정책	공격 시나리오별 탐지를 위한 룰 규칙 자동화 코드 구현

6.3 기술 활용 계획

	사용도구	내용
1	Security Onion	- Security Onion을 활용한 실시간 통합 관제 체계 수립 및 시각화 - 트래픽 미러링 프로브 및 중앙 로그 수집 서버의 고가용성(HA) 구성 방안
2	Snort	Snort 기반의 지능형 침입 탐지 룰셋(Rule-set) 설계 및 튜닝
	Sguil	Sguil 연동을 통한 실시간 위협 가시화 및 심층 분석 프로세스 수립
3	Wireshark	- Deep Packet Inspection(DPI) 기반의 위협 트래픽 심층 분석 방안 -공격 단계별 네트워크 증적 추출 및 시각화
4	python	- 공격 시나리오별 맞춤형 탐지 규칙(Rule-set) 자동화 구현 - 탐지-대응 시퀀스 최적화를 통한 오탐(False Positive) 최소화

6.4 운용상세

6.4.1 Snort 탐지 정책 설계 및 자동화 전략

1) 탐지 정책 체계 수립

- 의도 : 룰 번호만으로 공격 유형(정찰, DoS, 웹 등)을 즉시 식별하여 관제 요원의 직관적인 대응을 지원함
- SID 구조 : 2 + XX + YY + ZZ (고유 식별자 + 대분류 + 중분류 + 순번)
- 체계화 : 임의의 번호가 아닌, 내부 표준 아키텍처를 기반으로 룰을 계층화하여 관리 효율성 극대화

2) 공격 유형별 탐지 및 분류 체계

대분류 (XX)	공격 유형	중분류(YY)	탐지 목표 및 내용	매칭되는 실제 공격 (룰)
1	정보 수집 (염탐/정찰)	01 스텔스/ 특수 스캔	Nmap 등 도구의 특수 옵션을 사용하여 방화벽을 우회하거나 몰래 스캔하는 행위 식별	Null Scan, Xmas Scan
		02 일반 / 스크립트 스캔	핑(Ping)을 무시하거나 스크립트 엔진(NSE)을 사용하여 서버 정보를 수집하는 행위 탐지	-Pn 스캔, NSE 스크립트 스캔, DNS Zone Transfer
2	서비스 마비 (DoS 공격)	01 ICMP 과부하	패킷 크기를 비정상적으로 키우거나(Large Packet) 대량의 핑을 쏘아 가용성을 저해하는 행위 식별	ICMP Flood, Large Packet(-s)
		02 트래픽 폭탄	TCP/UDP 연결을 무한대로 요청하여 시스템 자원을 고갈시키는 공격 탐지	(TCP·UDP Flood/향후 확장 예정)
3	웹 해킹 (사이트 공격)	01 입력값 변조	로그인 창 등에 악성 쿼리나 스크립트를 주입하여 정보를 탈취하거나 인증을 우회하는 행위 식별	SQL Injection, XSS(Cross Site Scripting)
		02 명령어 주입	웹 서비스를 통해 서버 컴퓨터의 시스템 명령어를 몰래 실행하려는 시도 탐지	Command Injection, Reverse Shell Connection
		03 경로 탐색	허용되지 않은 상위 폴더(../)로 이동하여 시스템 주요 파일을 열람하거나 탈취하려는 행위 식별	Directory Traversal (../)
4	패스워드 공격 (무차별 대입)	01 SSH 접속 시도	관리자 포트(22번) 등 특정 서비스 포트에 패스워드를 반복적으로 대입하여 탈취하려는 행위 탐지	Hydra Brute Force Attack

3) Snort 룰 자동화 프로세스 (자료 7.1.4 참고)

- 관리자로부터 공격 유형 및 침입 지표 (IP, Port, 패턴 등)를 입력받아 Snort의 local.rules 파일을 실시간으로 업데이트하는 자동화 코드를 설계함

6.5 공격별 로그 분석

6.5.1 공격 도구별 흔적 분석 방안

- 공격 도구(Kali, Hydra, SQLmap 등)에서 생성되는 자체 행위 로그 추출 및 분석 방안 수립

6.5.2 시나리오별 핵심 로그 정의

- OS: 시나리오 1의 SSH Brute Force 대응을 위한 auth.log 집중 수집
- Web/DB: 시나리오 2, 3의 웹 기반 공격(XSS, CSRF, LFI) 분석을 위한 Access/Error log 및 DB Audit log 확보

[표 6.5.2-1] 시나리오 1 로그 분석 계획 - 내부망 침투 및 시스템 파괴

공격 단계	공격 기법	시스템 로그	탐지 키워드	분석 방법
초기 침투	ARP 스푸핑	(로그 없음)	(탐지 불가)	서버 로그에는 남지 않으며, 네트워크 장비의 ARP 테이블 변동 확인 필요
	FTP 패킷 스니핑	(로그 없음)	(탐지 불가)	단순 수동적 스니핑은 로그가 남지 않음. 평문 전송 중단 정책 필요
	WhatWeb	/var/log/apache2/access.log	WhatWeb, X-Thurgood	User-Agent 내 스캐너 문자열 및 특정 헤더 패턴 식별
정보 수집	Nmap 스캔	/var/log/apache2/access.log	Nmap, 404 Error	짧은 시간 내 대량의 존재하지 않는 페이지(404) 요청 로그 패턴 분석
권한 탈취	SSH Brute Force	/var/log/auth.log	Failed password, root	동일 IP에서 초당 수회 발생하는 인증 실패 로그(Failed password) 개수 확인
망 확장	DNS Zone Transfer	/var/log/syslog	AXFR, denied	외부 IP로부터의 영역 전송(AXFR) 요청 시도 및 거부 기록 모니터링
관리자 획득	SQL Injection	/var/log/apache2/access.log	UNION, SELECT, ' OR	HTTP GET/POST 요청 인자에 DB 예약어 및 특수문자 포함 여부 전수 조사
시스템 파괴	Cmd Injection	/var/log/apache2/access.log	;, |, cat, passwd	파라미터 내 시스템 명령어(OS Command) 실행을 위한 메타문자 포함 여부 확인
	Reverse Shell	/var/log/syslog, 커널 로그	sh -i, bash -i, nc -e, connect	외부로의 비정상 연결(Outbound) 및 쉘 실행 프로세스 추적
최종 파괴	DoS 공격	/var/log/syslog	Out of memory, Timeout	시스템 자원 고갈 메시지 발생 시점과 트래픽 급증 시점의 연관성 분석

[표 6.5.2-2] 시나리오 2 로그 분석 계획 - 사회공학 기법을 활용한 고객 정보 탈취

공격 단계	공격 기법	시스템 로그	탐지 키워드	분석 방법
타겟 탐색	dnsenum	/var/log/syslog	AXFR, Transfer	외부 IP로부터의 비정상적인 영역 전송(Zone Transfer) 요청 및 성공 여부 분석
	dig	/var/log/syslog	query, named	특정 도메인의 반복 조회 및 버전 정보 수집 패턴 감지
	Nmap	/var/log/apache2 /access.log	Nmap, ₩x16₩x03, 404 Error	비정상 User-Agent 및 대량의 페이지 오류(404) 발생 분석
세션 탈취	Reflected XSS	/var/log/apache2 /access.log	<script>, cookie	URL 파라미터 내 자바스크립트 공격 구문 포함 여부 추출
계정 침투	Mail Brute Force	/var/log/mail.log	Login failed, AUTH	메일 서버 로그에서 동일 IP의 단시간 로그인 실패 횟수 확인
정보 탈취	Blind Injection	웹 서버 /var/log/apache2 /access.log	union, select	HTTP 요청 인자(URL) 내 SQL 예약어 포함 여부 및 응답 크기 변화 조사

[표 6.5.2-3] 시나리오 3 로그 분석 계획 - 타인 계정 활용 및 비자금 조성

공격 단계	공격 기법	시스템 로그	탐지 키워드	분석 방법
미끼 투척	악성 쉘 실행	(로그 없음)	(탐지 불가)	클라이언트 PC 내부에서 실행된 로컬 행위이므로 서버 로그에는 기록되지 않음
권한 오용	CSRF 공격	/var/log/apache2 /access.log	password_change	사용자의 실제 페이지 이동 경로와 다른 직접적인 중요 기능 호출 로그 확인
서버 침투	LFI 공격	/var/log/apache2 /access.log	../, /etc/passwd	상대 경로 탐색(../../)을 통해 시스템 중요 파일에 접근하려는 시도 분석
	File Upload	/var/www/html/uploads/	.php, .sh, .py	로그 분석보다는 업로드 디렉토리 내 실행 가능 확장자 파일의 존재 여부 점검
서버 장악	Reverse Shell	/var/log/syslog	nc, bash -i, sh -i	서버 내부에서 외부 IP로 네트워크 연결을 시도하는 프로세스(Socket) 기록 확인
금전 이득	Salami Attack	DB Query Log	UPDATE, 0.01	(탐지 매우 어려움) DB 쿼리 로그에서 소액 단위의 반복적인 변동 패턴 정밀 분석

6.6 네트워크 패킷 분석 계획

6.6.1 비정상 패킷 분석

- Snort & Sguil: 발생된 경보(Alert)와 실제 패킷 데이터 간의 상관관계 분석
- Wireshark: 시나리오 1(ARP Spoofing) 및 시나리오 3(Reverse Shell)의 L7 페이로드 내 공격 구문 심층 식별

6.6.2 정상/비정상 패킷 식별 계획

[표 6.6.2-1] 시나리오 1 패킷 식별 계획 - 내부망 침투 및 시스템 파괴

공격 단계	공격 기법	필터 (Wireshark)	정상 패킷 패턴	비정상(공격) 패킷 패턴	정상/비정상 식별 기준
초기 침투	ARP 스푸핑	arp	IP-MAC 주소 1:1 매칭	ARP Reply 대량 발생	동일 IP에 MAC 주소 지속 변조
	FTP 패킷 스니핑	ftp	암호화된 전송 (식별불가)	ID/PW 평문 노출	Payload 내 계정 정보 노출 여부
	WhatWeb	http.user_agent	브라우저 기반 요청	UA 내 'WhatWeb' 포함	헤더 내 스캐너 명시
정보 수집	Nmap 스캔	tcp.flags.syn == 1	특정 포트 단일 접속	수천 개의 포트로 SYN 전송	단시간 내 다수 포트 접속 시도
권한 탈취	SSH Brute Force	tcp.port == 22	간헐적 원격 접속	초당 수십 회 TCP 연결/해제	동일 IP의 인증 시도 횟수 폭증
망 확장	DNS Zone Transfer	dns.flags.opcode == 5	특정 도메인 단일 질의	Standard query AXFR 요청	응답 내 모든 레코드 포함 여부
관리자 획득	SQL Injection	http.request.uri	일반적인 파라미터 요청	SELECT, cat, ; 등 포함	요청 데이터 내 DB 예약어 포함
시스템 파괴	Cmd Injection	http.request.uri	정상 URI 경로 호출	; , cat , passwd 포함	메타문자 기반 OS 명령어 시도
	Reverse Shell	tcp.flags.syn == 1	인바운드 서비스 접속	비정상 아웃바운드 접속	외부 연결 및 쉘 명령어
최종 파괴	DoS 공격	icmp / tcp.flags.syn	안정적 트래픽 유지	SYN/ICMP 과다 유입	PPS가 정상 임계치를 초과

[표 6.6.2-2] 시나리오 2 패킷 식별 계획 - 사회공학 기법을 활용한 고객 정보 탈취

공격 단계	공격 기법	필터 (Wireshark)	정상 패킷 패턴	비정상(공격) 패킷 패턴	정상/비정상 식별 기준
타겟 탐색	dnsenum	dns.flags.opcode == 5	특정 도메인에 대한 단일 질의	Standard query AXFR 요청 패킷 발생	응답 패킷 내 모든 존(Zone) 레코드 포함 여부
	dig	dns.flags.response == 0	일반적인 도메인 이름 해석 요청	특정 도메인 반복 쿼리 및 버전 정보 요청	동일 도메인 반복 질의 빈도 및 DNS 버전 정보 수집 패턴
	Nmap	tcp.flags.syn == 1	서비스 제공 포트로의 단일 접속	수천 개의 포트로 SYN 패킷 대량 전송	단시간 내 다수 포트(Multi-port) 접속 시도
세션 탈취	Reflected XSS	http.request.uri	일반 검색 파라미터	<script>, cookie	URI 내 인코딩된 스크립트 포함
계정 침투	Mail Brute Force	http.request.method=="POST"	1회성 로그인 요청	로그인 POST 집중	대량의 401 에러 후 200 발생
정보 탈취	Blind Injection	http.request.uri	일반적인 웹 페이지 파라미터 요청	union, select 등 예약어 포함	특정 IP의 요청 빈도 폭증 및 응답 크기 변화

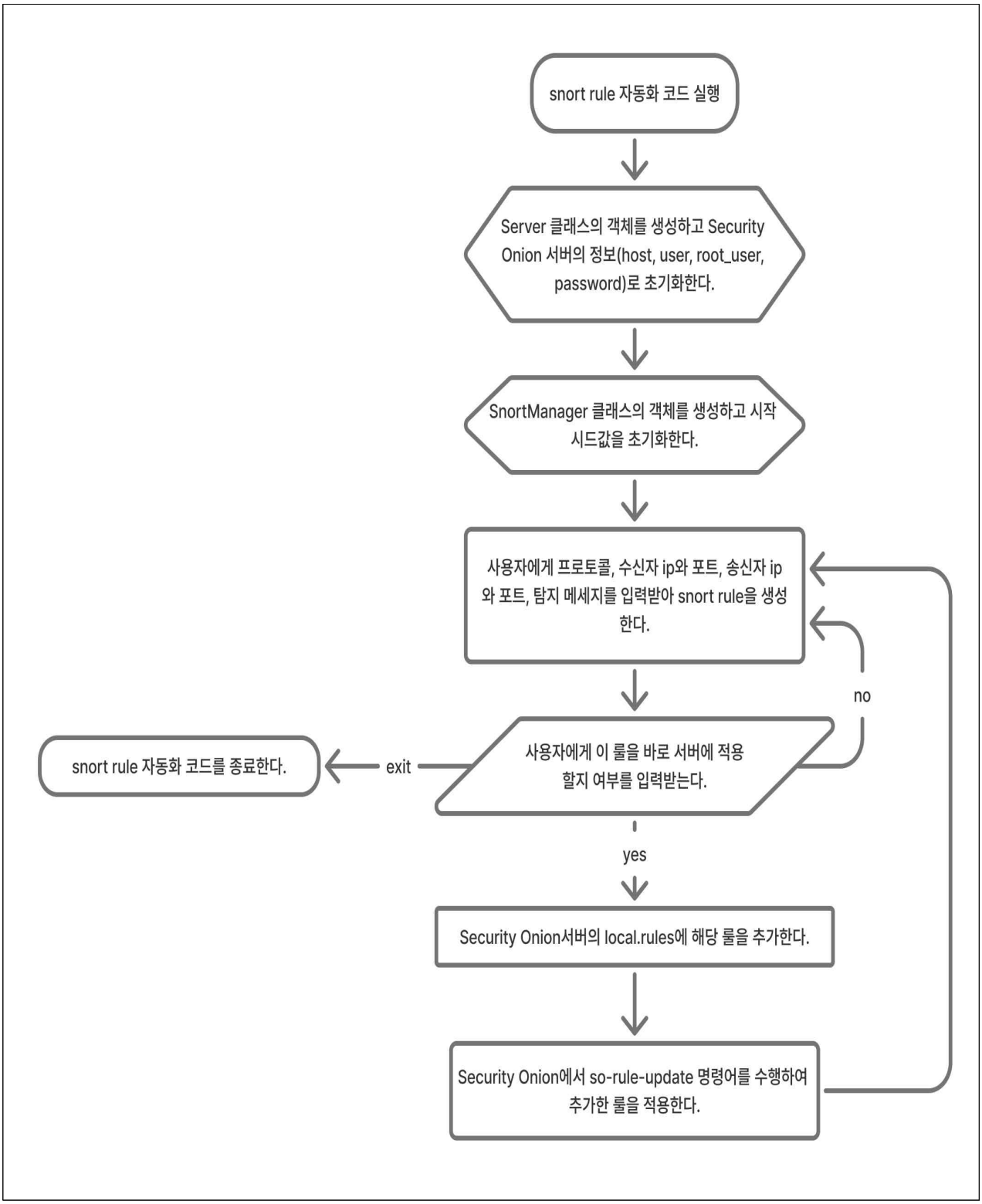
[표 6.6.2-3] 시나리오 3 패킷 분석 계획 - 타인 계정 활용 및 비자금 조성

공격 단계	공격 기법	필터 (Wireshark)	정상 패킷 패턴	비정상(공격) 패킷 패턴	정상/비정상 식별 기준
미끼 투척	악성 쉘 실행	tcp (외부 IP)	목적지 직접 통신	해커 IP(Proxy) 경유	게이트웨이가 비인가 IP로 변경
권한 오용	CSRF 공격	http.referer	내부 도메인발 호출	외부 / 비어 있는 Referer	외부 주소발 중요 기능 호출
서버 침투	LFI 공격	http.request.uri	정상 파일 경로 호출	../ , /etc/passwd	상위 디렉토리 접근 패턴 존재
	File Upload	http.request.method=="POST"	이미지 데이터 전송	파일 내 <?php 포함	이미지 시그니처 내 스크립트 포함
서버 장악	Reverse Shell	tcp.flags.syn==1	외부 접속 (Inbound)	서버발 아웃바운드	서버가 비인가 IP로 먼저 연결
금전 이득	Salami Attack	mysql / tcp.port==3306	정상 트랜잭션 쿼리	소액 UPDATE 반복	미세 금액 조작 쿼리 임계치 초과

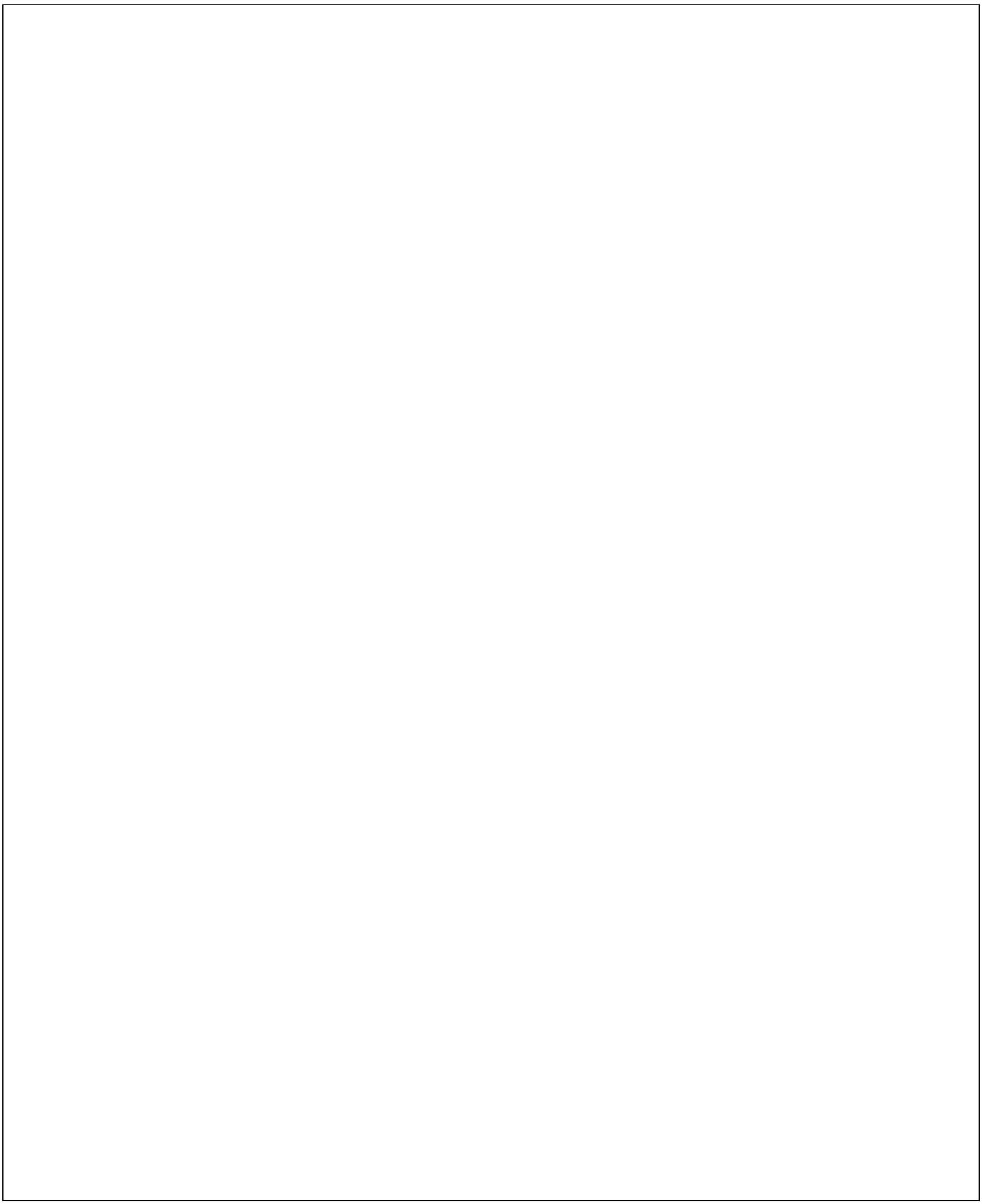
7. 자동화 코드 관리

7.1 코드 흐름도

7.1.4 관제 Snort Rule 자동화 코드 흐름도



7.1.5 주요정보통신시설 기반 코드 흐름도



7.2 함수정의서

7.2.4 관제 및 대응부분 함수정의서

- 입력받은 snort rule 옵션을 바탕으로 snort rule을 자동으로 작성하고, snort rule을 구분하기 위한 값인 sid값을 자동으로 배분한다. 룰을 작성한 이후 문법을 검증한다.

목적	간편한 snort rule 작성 및 모든 Security Onion 내 snort rule 동기화
특징	사용자 입력을 통한 snort rule 생성, IDS서버에 snort rule 추가, IDS 서버 내부 명령어를 통한 snort rule 문법 검증
사용 환경	Security Onion

NO	클래스	함수명	주요 기능 (기획서 연계)	입력 예시	출력/결과
1	Server	init()	인자값으로 클래스의 변수를 초기화한다.	서버 ip, ssh id, ssh pw	N/A(해당없음)
2		ssh()	ssh 접속하여 인자로 받은 명령 실행	실행할 명령어	실행한 명령어의 결과 반환
3	SnortManager	init()	인자값으로 시작 SID 번호를 설정한다.	시작 SID 번호	N/A(해당없음)
4		create_rule()	사용자로부터 입력받아 snort rule 문자열을 생성한다.	프로토콜, src ip, src port, dst ip, dst port, 탐지 메세지 등	생성한 snort rule 문자열 반환
5		deploy_to_so()	create_rule()을 통해 만든 문자열을 local.rules 파일 내에 추가하고 문법을 검증한다.	N/A(해당없음)	문법 적용 결과(적용 성공 or 실패) 출력

7.2.5 주요정보통신시설 기반 함수정의서

- 요약

목적	
특징	
사용 환경	

NO	스크립트/함수 그룹	대상 서버/OS	주요 기능 (기획서 연계)	입력 예시 (IP/옵션)	출력/결과
1	init_ssh + menu	Ubuntu/Rocky9 (Core)	DHCP/Apache/VirtualHost/PHP/phpMyAdmin/FTP/MariaDB 선택/전체 설치	IP:172.16.16.122, [1-3]	서비스 활성 + index.html
2	apa_in() apa_ch_dir_in()	Rocky9 (Web EN/KR)	Apache + VirtualHost (team1.com) + DocRoot 변경	/home/team1	httpd.conf 수정 + 재시작
3	smb_install() go()	Rocky9 (Storage)	Samba 설치/공유[/share/smb]/user(smb) 생성	PW:asd123!@	smbd/nmbd active
4	install_and_configure() main()	Ubuntu (MB)	VNC(TigerVNC) + GNOME + xstartup(GNOME-session)	PW:asd123!@, :5901	vncserver@:1 active
5	nfs_install() main()	Rocky9 (Storage)	NFS utils + /mnt/nfs_share 공유 + 마운트	NFS IP:172.16.16.100	exportfs -v 확인
6	download() main()	Rocky9 (MB-Monitorix)	Monitorix + Perl deps 설치/재시작	-	systemctl status monitorix
7	install_gnome() main()	Rocky9 (MB)	XRDP + GNOME GUI + user(team1) + 그룹 추가	user:team1, PW:asd123!@	xrdp active (RDP:3389)
8	install() mod_zone()	Rocky9 (DNS Server)	BIND + team1.com zone + 동적 A 레코드 편집	dns_data.txt 입력	nslookup team1.com 성공
9	format_partition() main()	Ubuntu (ST-R5/10)	Partition/FMT/ext4 + fstab + Disk Quota(team1)	/dev/sda1, /jupyter	repquota 제한 확인
10	cc(cmd)	모든 서버	SSH 명령 실행 + 출력 파싱/오류 처리	임의 cmd	stdout/stderr 실시간 출력

8. 테스트 및 검증계획

8.4 관제 및 대응 구현테스트 및 침투테스트

8.4.1 관제 및 대응 구현테스트

8.4.2 시나리오 기반 침투 테스트

[프로세스]

순서	프로세스	내용
1	**공격시나리오 설계**	- 대상 시스템별 위협 모델링 및 공격 경로 설정
2	**단계별 침투 시뮬레이션 실행**	- 공격 도구별 페이로드(Payload) 실행 및 흔적 생성 - 공격 단계별 타임라인(Timeline) 기록 및 증적 확보 계획
3	**실시간 공격-탐지 동기화 및 커뮤니케이션 계획**	- 공격 발생 시점과 IDS 알람 발생 시점을 실시간으로 대조하는 협업 프로세스(Sync-up) 정의
4	**단계별 탐지 유효성 검증 매트릭스 수립**	- 공격 기법(TTPs)별로 우리 시스템이 실제로 탐지했는지 판정하는 기준 정의

[표 6.3.4-1] 시나리오 1 탐지 및 대응 매트릭스 - 내부망 침투 및 시스템 파괴

단계	공격 기법	탐지 포인트 (IDS/로그)	대응 포인트	증적 확보 항목
정찰	ARP Spoofing / Scanning	Snort (ARP reply/Scan alert), Wireshark	스위치 Port Security 설정 적용	PCAP 패킷, Scan 로그
침투	SSH Brute Force (Hydra)	Auth log (Failed password), Snort (SSH Brute)	iptables (Recent 모듈로 반복 시도 IP 차단)	/var/log/auth.log
확장	DNS Zone Transfer (dig)	DNS 쿼리 로그, Snort (DNS Zone Transfer)	ACL (DNS 서버 쿼리 허용 IP 제한)	DNS 쿼리 로그
장악	SQL & Command Injection	Web 로그 (404/500), Snort (Web App Attack)	iptables (불필요한 outbound 포트 차단)	Web Access/Error 로그
파괴	DoS 공격	트래픽 임계치 초과 알람 (Security Onion)	ACL (공격지 IP 원천 차단)	트래픽 통계 리포트

[표 6.3.4-2] 시나리오 2 탐지 및 대응 매트릭스 - 사회공학 기법을 활용한 고객 정보 탈취

단계	공격 기법	탐지 포인트 (IDS/로그)	대응 포인트	증적 확보 항목
정찰	Version Detection (Nmap)	Snort (OS Fingerprinting 탐지)	ACL (외부망 대상 스캔 트래픽 차단)	IDS Alert 리스트
취약점	Reflected XSS	Web 로그 (특수문자 포함 URL), Snort (XSS)	Web 방화벽 정책 (Script 구문 필터링)	공격 URL 캡처
인증	Mail Brute Force	Mail 서버 로그 (로그인 시도 횟수 초과)	iptables (메일 서버 접속 제한 정책)	Mail Auth 로그
탈취	Cookie Hijacking / SQLi	Snort (SQL UNION 탐지), 외부 도메인 유출	ACL (비인가 외부 도메인 통신 차단)	PCAP, DB 쿼리 로그

6.3.4-3] 시나리오 3 탐지 및 대응 매트릭스 - 타인 계정 활용 및 비자금 조성 (Salami Attack)

단계	공격 기법	탐지 포인트 (IDS/로그)	대응 포인트	증적 확보 항목
유인	Proxy 강제 변경 (.sh)	시스템 프록시 설정 변경 로그 (Auditd)	iptables (비인가 프록시 포트 통신 차단)	.sh 스크립트 파일
우회	CSRF (비밀번호 변경)	Web 로그 (Referer 헤더 부재/변조)	Web 서버 설정 (CSRF Token 검증 강화)	HTTP Request 로그
침투	LFI / Webshell Upload	Snort (LFI/Webshell 탐지), 파일 생성 로그	iptables (Web 업로드 디렉토리 실행 권한 제한)	업로드된 Webshell 파일
장악	Reverse Shell (Netcat)	Snort (Reverse Shell Payload), 커넥션 로그	ACL (내부망 -> 외부망 비인가 포트 차단)	넷캣 세션 로그
수행	DB 조작 (Salami)	DB Audit 로그 (Update/Insert 쿼리 급증)	iptables (DB 서버 접근 IP 화이트리스트)	DB 감사(Audit) 로그