

EQST INSIGHT

오픈 소스 소프트웨어 보안 가이드



Introduction

안녕하십니까? SK인포섹EQST그룹 이재우입니다.

이번에 저희 EQST그룹에서 오픈 소스 소프트웨어(Open Source Software) 보안 가이드를 발간 하였습니다.

최근 몇 년 동안 빅데이터 기술 분야를 중심으로 오픈 소스 소프트웨어 도입이 증가하고 있습니다. 오픈 소스는 개방성을 기반으로 여러 장점을 가지고 있지만, 초기 도입부터 보안성을 고려하지 않은 경우에는 해킹 공격의 타깃이 되곤 합니다. 실제 지난해 1억건 이상의 개인정보가 유출되었던 美 대형신용평가사 에퀴팩스 해킹 사건 역시 오픈 소스인 아파치 스트러츠의 취약점을 방치해 큰 피해를 입었습니다.

이처럼 오픈 소스에 대한 사이버 위협 리스크가 지속적으로 증가하고 있음에도 불구하고, 아직 오픈 소스에 대한 보안 지식이 체계화 되어 있지 않아 이를 사용하는 기업들의 고민이 깊어지고 있습니다.

이에 EQST그룹에서는 이런 문제를 해결하기 위해 오픈 소스 22종에 대한 보안 가이드를 발간하였습니다. 본 가이드에는 각 오픈 소스마다 가지고 있는 취약점 정보를 비롯해, 이에 대한 보안 설정, 운영 가이드 등이 자세하게 설명돼 있습니다.

본 가이드를 통해 오픈 소스 보안에 대한 고민이 해결될 수 있길 바라며, EQST 그룹은 앞으로도 오픈 소스 보안의 기준을 제시하는데 앞장서도록 하겠습니다.

감사합니다.

EQST 그룹장
이재우

목 차

I. 전체목록.....	4
1. 체크리스트 항목.....	4
1.1. Node.js.....	4
1.2. Wildfly	4
1.3. NginX.....	4
1.4. Elastic Search.....	5
1.5. Docker	5
1.6. Cassandra.....	6
1.7. Hadoop.....	6
1.8. HBase	6
1.9. Hive.....	6
1.10. Impala	7
1.11. Jenkins	7
1.12. Kafka	7
1.13. Maven	7
1.14. Mesos.....	8
1.15. Oauth	8
1.16. Spark.....	10
1.17. Squid	10
1.18. Storm	11
1.19. MongoDB.....	11
1.20. MySQL.....	11
1.21. PostgreSQL.....	12
1.22. Redis.....	12
2. 위험도 및 적용 권고 시기 구분.....	14
2.1. 위험도.....	14
2.2. 적용 권고 시기.....	14
II. 세부항목 설정.....	15
1. Node.js.....	15
1.1. 설정	15
1.2. 보안 패치.....	21
2. Wildfly	23
2.1. 설정	23
2.2. 솔루션 취약점	43
2.3. 보안 패치.....	47
2.4. 접근 제어.....	48
3. NginX	55
3.1. 설정	55
3.2. 솔루션 취약점.....	65
3.3. 보안 패치.....	68

4.	Elastic Search.....	70
4.1.	X-Pack 설정.....	70
4.2.	사용자 접근 통제.....	72
4.3.	네트워크 접근 통제.....	75
4.4.	로그 설정.....	77
5.	Docker.....	78
5.1.	설정	78
5.2.	Docker 신뢰 설정.....	79
5.3.	Container 설정	80
5.4.	Log 설정	83
5.5.	추가 기능 설정.....	83
6.	Cassandra.....	85
6.1.	암호화 통신 설정.....	85
6.2.	인증 및 권한 설정	86
7.	Hadoop.....	89
7.1.	설정	89
7.2.	HDFS설정.....	91
7.3.	기타 보안 설정.....	97
8.	HBase.....	98
8.1.	계정 관리.....	98
8.2.	Kerberos 설정	99
9.	Hive.....	101
9.1.	인증 설정.....	101
9.2.	운영 매트릭스	103
10.	Impala	106
10.1	설정	106
11.	Jenkins	112
11.1.	설정	112
11.2.	계정 관리.....	113
11.3.	추가 기능 설정.....	113
12.	Kafka.....	115
12.1.	SSL 설정	115
13.	Maven	118
13.1.	설정	118
13.2.	계정 암호화 설정	120
14.	Mesos.....	122
14.1.	인증 설정	122
14.2.	접근통제	130
14.3.	SSL 설정	140
15.	Oauth	144
15.1.	일반 설정.....	144
15.2.	인증서버 설정.....	167
15.3.	클라이언트 앱 보안 설정	181
15.4.	리소스 서버 설정.....	187
16.	Spark.....	189
16.1.	접근 통제.....	189

16.2.	로그 설정	190
16.3.	암호화 설정	190
16.4.	기타	192
17.	Squid.....	193
17.1.	설정	193
17.2.	계정 관리	196
17.3.	파일 및 디렉터리 관리	197
17.4.	서비스 관리	197
17.5.	접근제어	198
18.	Storm	200
18.1.	설정	200
19.	MongoDB(NoSQL)	202
19.1.	계정 관리	202
19.2.	권한 관리	204
19.3.	DBMS 보안설정	205
19.4.	환경 파일 점검	209
19.5.	보안 패치	213
20.	MySQL.....	214
20.1.	계정 관리	214
20.2.	권한 관리	220
20.3.	DBMS 보안설정	223
20.4.	환경 파일 점검	228
20.5.	보안 패치	234
21.	PostgreSQL.....	235
21.1.	계정 관리	235
21.2.	권한 관리	238
21.3.	DBMS 보안설정	240
21.4.	환경 파일 점검	242
21.5.	보안 패치	247
21.6.	보안 감사 설정	249
22.	Redis(NoSQL)	251
22.1.	계정 관리	251
22.2.	권한 관리	253
22.3.	DBMS 보안설정	254
22.4.	환경 파일 점검	259
22.5.	보안 패치	261

I. 전체목록

1. 체크리스트 항목

진단에 사용될 체크리스트는 국내외 기술 자료를 바탕으로 작성되었다.

1.1. Node.js

설정(5개 항목), 보안패치(1개 항목)으로 총 2개 영역에서 6개 항목으로 구성되었다.

[표] 1. Node.js 보안진단 체크리스트

구분	항목코드	항목명	중요도
설정	1.1	데몬 관리	상
	1.2	로그 디렉토리/파일 권한 설정	중
	1.3	로그 포맷 설정	상
	1.4	로그 저장 주기	상
	1.5	헤더 정보 노출 방지	하
보안 패치	2.1	보안 패치 적용	상

1.2. Wildfly

설정(12개 항목), 솔루션 취약점(3개 항목), 보안패치(1개 항목), 접근 제어(4개 항목)으로 총 4개 영역에서 20개 항목으로 구성되었다.

[표] 2. WildFly 보안진단 체크리스트

구분	항목코드	항목명	중요도
설정	1.1	데몬 관리	상
	1.2	관리서버 디렉토리 권한 설정	중
	1.3	설정파일 권한 설정	상
	1.4	로그 디렉토리/파일 권한 설정	중
	1.5	로그 포맷 설정	상
	1.6	로그 저장 주기	상
	1.7	HTTP Method 제한	하
	1.8	디렉토리 검색 기능 제거	중
	1.9	데이터소스의 패스워드 암호화	중
	1.10	Session Timeout 설정	중
	1.11	헤더 정보 노출 방지	하
	1.12	에러 메시지 관리	중
솔루션 취약점	2.1	불필요한 파일 삭제	하
	2.2	SSL v3.0 POODLE 취약점	상
	2.3	Apache Commons-Collection 라이브러리 취약점	상
보안 패치	3.1	보안 패치 적용	상
접근 제어	4.1	관리자 콘솔 접근통제	상
	4.2	관리자 default 계정명 변경	하
	4.3	관리자 패스워드 암호정책	상
	4.4	패스워드 파일 권한 설정	중

1.3. NginX

설정(10개 항목), 솔루션 취약점(2개 항목), 보안패치(1개 항목)으로 총 3개 영역에서 13개 항목으로 구성되었다.

[표] 3. Nginx 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
설정	1.1		데몬 관리	상
	1.2		관리서버 디렉토리 권한 설정	중
	1.3		설정파일 권한 설정	상
	1.4		디렉토리 검색 기능 제거	중
	1.5		로그 디렉토리/파일 권한 설정	중
	1.6		로그 포맷 설정	상
	1.7		로그 저장 주기	상
	1.8		헤더 정보 노출 방지	하
	1.9		HTTP Method 제한	하
	1.10		에러 메시지 관리	중
솔루션 취약점	2.1		기본 문서명 사용 제한	하
	2.2		SSL v3.0 POODLE 취약점	상
보안 패치	3.1		보안 패치 적용	상

1.4. Elastic Search

X-Pack설정(2개 항목), 사용자접근통제(3개 항목), 네트워크 접근통제(2개 항목), 로그설정(2개 항목)으로 총 4개 영역에서 9개 항목으로 구성되었다.

[표] 4. ElasticSearch 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
X-Pack설정	1.1		X-Pack 활성화 설정	상
	1.2		기본패스워드 설정	상
사용자 접근 통제	2.1		접근 정책 설정	상
	2.2		사용자 계정 설정	상
	2.3		익명연결 비활성화	상
네트워크 접근 통제	3.1		SSL/TLS 설정	권고
	3.2		네트워크 필터링 설정	권고
로그 설정	4.1		감사 로그 설정	상
	4.2		사용자 감사 설정	중

1.5. Docker

설정(3개 항목), Docker 신뢰 설정(2개 항목), Container 설정(4개 항목), Log설정(1개 항목), 추가기능설정(1개 항목)으로 총 5개 영역에서 11개 항목으로 구성되었다.

[표] 5. Docker 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
설정	1.1		데몬관리	상
	1.2		리소스 제한 설정	상
	1.3		SUID / SGID 제거 설정	하
Docker 신뢰 설정	2.1		불필요한 파일 삭제	중
	2.2		Apache Commons-Collection 라이브러리 취약점	중
Container	3.1		네트워크 설정	참고
	3.2		Root외 권한으로 컨테이너 실행	참고

	3.3		Shell 권한 설정	중
	3.3		SSH 설정	상
Log 설정	4.1		Docker Log 설정	상
추가 기능 설정	5.1		AppArmor 연결	하

1.6. Cassandra

암호화 통신 설정(3개 항목), 인증 및 권한 설정(3개 항목)으로 총 2개 영역에서 6개 항목으로 구성되었다.

[표] 6. Cassandra 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
암호화 통신 설정	1.1.		노드 간 암호화	상
	1.2.		클라이언트 대 노드 암호화	상
	1.3		SSL 통신 JMX 설정	권고
인증 및 권한설정	2.1		인증 설정	상
	2.2		내부 권한 구성(caching)	상
	2.3		JMX 인증 및 권한 부여	중

1.7. Hadoop

설정(2개 항목), HDFS설정(5개 항목), 기타 보안 설정(2개 항목)으로 총 3개 영역에서 9개 항목으로 구성되었다.

[표] 7. Hadoop 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
설정	1.1		Hadoop 보안 사용	상
	1.2		SPNEGO/Kerberos 웹 접속 관련 보안 설정	상
HDFS설정	2.1		보안 HDFS 구성	중
	2.2		HDFS에 SSL 사용	중
	2.3		Secure Web HDFS 설정	중
	2.4		HDFS ACL 설정	상
	2.5		HDFS 권한 설정	권고
기타 보안 설정	3.1		Secure HDFS NFS Gateway 설정	중
	3.2		Variables for Secure DataNodes 설정	중

1.8. HBase

계정 관리(2개 항목), Kerberos 설정(1개 항목) 2개영역에서 총 3개 항목으로 구성되었다.

[표] 8. HBase 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
계정 관리	1.1		인증 활성화 설정	상
	1.2		계정 권한 부여 설정	상
Kerberos 설정	2.1		Kerberos 인증 설정	권고

1.9. Hive

인증 설정(2개 항목)으로 총 1개 영역에서 2개 항목으로 구성되었다.

[표] 9. Apache Hive 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
인증 설정	1.1		SQL 기반 인증	상
	1.2		저장소 기반 인증	상
매트릭스	2.1		권한관리 매트릭스	권고
	2.2		운영 매트릭스	권고

1.10. Impala

설정(4개 항목)으로 총 1개 영역에서 4개 항목으로 구성되었다.

[표] 10. Impala 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
설정	1.1		데몬관리	상
	1.2		접근통제 설정	상
	1.3		기본 포트 변경	중
	1.4		설정	권고

1.11. Jenkins

설정(2개 항목), 계정 관리(1개 항목), 추가 기능 설정(1개 항목)으로 총 4개 항목으로 구성되었다.

[표] 11. Jenkins 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
설정	1.1		기본 보안설정 활성화	상
	1.2		기본 사용자 설정	중
계정 관리	2.1		사용자 권한 설정	상
추가 기능 설정	3.1		CSRF 차단설정	중

1.12. Kafka

설정 2개 항목으로 구성되었다.

[표] 12. Kafka 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
설정	1.1		SSL 설정 (Kafka Broker)	권고
	1.2		SSL 설정 (Kafka 클라이언트)	권고

1.13. Maven

설정(3개 항목), 계정암호화 설정(2개 항목)으로 총 2개 영역에서 5개 항목으로 구성되었다.

[표] 13. Maven 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
설정	1.1		HTTP Method 제한	하
	1.2		설정파일 권한 설정	상
	1.3		세션타임아웃 설정	하
계정 암호화 설정	2.1		마스터 비밀번호 암호화 설정	상
	2.2		사용자 계정 비밀번호 암호화 설정	상

1.14. Mesos

인증 설정(5개 항목), 접근통제 1개항목, SSL설정(3개 항목)으로 총 3개 영역에서 9개 항목으로 구성되었다.

[표] 14. Mesos 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
인증설정	1.1		인증설정(Master)	상
	1.2		인증설정(Agent)	상
	1.3		다중 HTTP Authenticator 설정	중
	1.4		Executor 인증 설정	중
	1.5		Framework 인증 설정	하
접근통제	2.1		ACL 설정	중
SSL설정	3.1		환경 설정	권고
	3.2		운영 설정	권고
	3.3		WebUI 설정	권고

1.15. Oauth

일반 설정(25개 항목), 인증서버 설정(19개 항목), 클라이언트 앱 보안 설정(5개 항목), 리소스 서버 설정(3개 항목)으로 총 4개 영역에서 52개 항목으로 구성되었다.

[표] 15. OAuth 2.0 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
일반 설정	1.1		OAuth 요청의 기밀성 설정	하
	1.2		서버 인증 설정	하
	1.3		리소스 소유자(Resource Owner)에게 알린 내용 유지 설정	하
	1.4		자격증명 - 표준 시스템 보안 수단 강제 설정	하
	1.5		자격증명 - 표준 SQL 인젝션 방지 강제 설정	하
	1.6		자격증명 - 자격증명 평문 저장 방지 설정	하
	1.7		자격증명 - 자격증명 암호화 설정	하
	1.8		자격증명 - 공개키 암호화 알고리즘 사용 설정	하
	1.9		Secret - 안전한 비밀번호 정책 설정	하
	1.10		Secret - Secret에 높은 엔트로피 설정	하
	1.11		Secret - 계정 잠금 설정	하
	1.12		Secret - Tar Pit 설정	하
	1.13		Secret - CAPTCHA 설정	하
	1.14		Token (접근, 갱신, 인증코드) - 토큰 범위 제한 설정	하
	1.15		Token (접근, 갱신, 인증코드) - 토큰 만료 시간 설정	하
	1.16		Token (접근, 갱신, 인증코드) - 짧은 토큰 만료 시간 설정	하
	1.17		Token (접근, 갱신, 인증코드) - 토큰 최대 사용 횟수 또는 1회용으로 설정	하
	1.18		Token (접근, 갱신, 인증코드) - 특정 리소스서버로 토큰 바인딩 설정(Audience)	하
	1.19		Token (접근, 갱신, 인증코드) - 토큰 Audience로서	하

		엔드포인트 주소 사용 설정	
	1.20	Token (접근, 갱신, 인증코드) - Audience, 토큰에 명시적으로 정의된 범위 설정	하
	1.21	Token (접근, 갱신, 인증코드) - 클라이언트 ID에 대한 토큰 바인딩 설정	하
	1.22	Token (접근, 갱신, 인증코드) - Self-Contained 토큰에 서명 설정	하
	1.23	Token (접근, 갱신, 인증코드) - 토큰 내용 암호화 설정	하
	1.24	Token (접근, 갱신, 인증코드) - 표준 Assertion 포맷 설정	하
	1.25	Token (접근, 갱신, 인증코드) - 접근토큰 보호 설정	하
인증서버 설정	2.1	인증코드 - 토큰 악용사례 탐지시 파생된 토큰 자동 폐기 설정	하
	2.2	갱신토큰 - 갱신토큰 발행 제한 설정	하
	2.3	갱신토큰 - "client_id"로 갱신토큰 바인딩 설정	하
	2.4	갱신토큰 - 갱신토큰 순환 설정	하
	2.5	갱신토큰 - 갱신토큰 폐기 설정	하
	2.6	갱신토큰 - 장치 식별 설정	하
	2.7	갱신토큰 - X-FRAME-OPTIONS 헤더 사용 설정	하
	2.8	클라이언트 인증 및 인가 - 클라이언트로 Secret 발행에 필요한 보안정책 설정	하
	2.9	클라이언트 인증 및 인가 - Secret 없는 일반 클라이언트 위한 사용자 약관 설정	하
	2.10	클라이언트 인증 및 인가 - "redirect_uri" 조합으로 "client_id"만 발행 설정	하
	2.11	클라이언트 인증 및 인가 - 설치 전용 클라이언트 Secret 발행 설정	하
	2.12	클라이언트 인증 및 인가 - 미리 등록된 "redirect_uri" 유효성 입증 설정	하
	2.13	클라이언트 인증 및 인가 - 클라이언트 Secret 폐기 설정	하
	2.14	클라이언트 인증 및 인가 - 강력한 클라이언트 인증 사용 설정	하
	2.15	최종사용자 인증 - 자동 반복 인증 처리시 클라이언트 유효성 입증 설정	하
	2.16	최종사용자 인증 - 투명성에 기반하여 현명한 결정을 위한 정보제공 기능 설정	하
	2.17	최종사용자 인증 - 최종사용자에 의한 클라이언트 Property 유효성 입증 설정	하
	2.18	최종사용자 인증 - 인증코드를 "client_id"에 바인딩 설정	하
	2.19	최종사용자 인증 - 인증코드를 "redirect_uri"에 바인딩 설정	하
클라이언트 앱 보안	3.1	소프트웨어 패키지와 번들된 코드 또는 리소스 내	하

설정		자격증명 저장 금지 설정	
	3.2	표준 웹 서버 보호 수단 설정	하
	3.3	안전한 저장소에 Secret 저장 설정	하
	3.4	비인증된 장치 접근을 방지하기 위한 디바이스 잠금 설정	하
	3.5	User Agent 세션에 대해 “state” 파라미터로 연결 설정	하
리소스 서버 설정	4.1	HTTP 인증헤더 사용 설정	하
	4.2	인증된 요청 사용 설정	하
	4.3	서명된 요청 사용 설정	하

1.16. Spark

접근 통제(2개 항목), 로그 설정(1개 항목), 암호화 설정(2개 항목), 기타(2개 항목)으로 총 4개 영역에서 7개 항목으로 구성되었다.

[표] 16. Spark 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
접근통제	1.1		관리자 설정	상
	1.2		사용자 접근 통제 설정	상
로그 설정	2.1		로그 설정	중
암호화 설정	3.1		일반 암호화 설정	권고
	3.2		로컬 임시파일 암호화 설정	권고
기타	4.1		HTTP 보호 설정	권고
	4.2		Spark 기본 포트	권고

1.17. Squid

설정(6개 항목), 계정관리(2개 항목), 파일 및 디렉터리 관리(1개 항목)으로 총 5개 영역에서 14개 항목으로 구성되었다.

[표] 17. Squid 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
설정	1.1		세션 타임아웃 설정	하
	1.2		HTTP 헤더 허용 사이즈 설정	하
	1.3		클라이언트 주소 숨김	중
	1.4		알 수 없는 네임서버 무시	하
	1.5		클라이언트 라이프타임 설정	하
	1.6		로그 설정	상
계정 관리	2.1		사용자 권한 설정	상
	2.2		사용자 인증 설정	상
파일 및 디렉터리 관리	3.1		중요 디렉터리 권한 설정	상
서비스 관리	4.1		FTP 설정	하
	4.3		SNMP 설정	중
접근제어	5.1		http 접근 포트 설정	하
	5.2		ACL 설정	중
	5.3		관리콘솔 관리	중

1.18. Storm

설정(2개 항목) 총 2항목으로 구성되었다.

[표] 18. Storm 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
설정	1.1		SSL 구성	권고
	1.2		Create Headless Principals and keytabs	중

1.19. MongoDB

계정관리(2개 항목), 권한 관리(2개 항목), DBMS 보안설정(5개 항목), 환경 파일 점검(3개 항목), 보안패치(1개 항목)으로 총 5개 영역에서 13개 항목으로 구성되었다.

[표] 19. MongoDB 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
계정 관리	1.1	-	MongoDB null 패스워드 점검	상
	1.2	-	패스워드 복잡도 설정	중
권한 관리	2.1	-	개발 및 운영 시스템 분리 사용	하
	2.2	-	root 권한으로 서버 구동 제한	상
DBMS 보안설정	3.1	-	백업 관리	하
	3.2	-	DB 접속 IP 통제	하
	3.3	-	로그 저장 주기	상
	3.4	-	로그 레벨 설정	상
	3.5	-	DBMS 서버 보안 연결	상
환경 파일 점검	4.1	-	MongoDB 환경설정 파일 접근 제한	중
	4.2	-	.dbshell 파일 접근 제한	중
	4.3	-	Log 파일 접근 제한	하
보안 패치	5.1	-	보안 패치 적용	상

1.20. MySQL

계정 관리(5개 항목), 권한 관리(4개 항목), DBMS 보안설정(5개 항목), 환경 파일 점검(6개 항목), 보안 업데이트(1개 항목)으로 총 5개 영역에서 21개 항목으로 구성되었다.

[표] 20. MySQL 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
계정 관리	1.1	D-01 D-02	불필요한 계정 확인	하
	1.2	D-03	패스워드 복잡도 설정	중
	1.3	D-01	root null 패스워드 점검	상
	1.4	D-03 D-05	취약한 패스워드 사용 점검	상
	1.5	D-02	Anonymous 계정 확인	하
권한 관리	2.1	-	개발 및 운영 시스템 분리 사용	하
	2.2	-	root 권한으로 서버 구동 제한	상
	2.3	D-08	mysql.user 테이블 접근 제한	상
	2.4	D-08	데이터베이스 접근 권한 제한	중

DBMS 보안설정	3.1	-	백업 관리	하
	3.2	D-10	샘플 DB 제거	하
	3.3	D-07	DB 접속 IP 통제	하
	3.4	-	LOCAL INFILE 사용제한	중
	3.5	-	로그 저장 주기	상
환경 파일 점검	4.1	D-13	mysql 명령 히스토리 검사	하
	4.2	D-13	Initialization 파일 접근 권한 설정	중
	4.3	D-13	mysql.server 파일 접근 권한 설정	중
	4.4	D-13	\$datadir 디렉토리 및 데이터 파일 접근 제한 설정	중
	4.5	D-13	.mysql_history 파일 접근 제한	중
	4.6	D-13	Log 파일 접근 제한	하
보안 업데이트	5.1	D-21 D-23	보안 업데이트 적용	상

1.21. PostgreSQL

계정 관리(3개 항목), 권한 관리(3개 항목), DBMS 보안설정(3개 항목), 환경 파일 점검(6개 항목), 보안패치(1개 항목), 보안 감사 설정(2개 항목)으로 총 6개 영역에서 18개 항목으로 구성되었다.

[표] 21. PostgreSQL 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
계정 관리	1.1	-	불필요한 계정 확인	하
	1.2	-	postgres null 패스워드 점검	상
	1.3	-	취약한 패스워드 사용 점검	상
권한 관리	2.1	-	개발 및 운영 시스템 분리 사용	하
	2.2	-	root 권한으로 서버 구동 제한	상
	2.3	-	schema 접근 권한 제한	중
DBMS 보안설정	3.1	-	백업 관리	하
	3.2	-	DB 접속 IP 통제	하
	3.3	-	로그 저장 주기	상
환경 파일 점검	4.1	-	PostgreSQL 명령 히스토리 검사	하
	4.2	-	PostgreSQL 환경설정 파일 접근 제한	중
	4.3	-	DB접속 통제 설정 파일 접근 권한 설정	중
	4.4	-	\$datadir 디렉토리 및 데이터 파일 접근 제한	중
	4.5	-	.psql_history 파일 접근 제한	중
	4.6	-	Log 파일 접근 제한	하
보안 패치	5.1	-	보안 패치 적용	상
보안 감사 설정	6.1	-	Log 감사 수행 설정	하
	6.2	-	로그 기록 설정	하

1.22. Redis

계정관리(2개 항목), 권한 관리(2개 항목), DBMS 보안설정(7개 항목), 환경 파일 점검(3개 항목), 보안패치(1개 항목)으로 총 5개 영역에서 15개 항목으로 구성되었다.

[표] 22. DBMS 보안진단 체크리스트

구분	항목코드	기반시설	항목명	중요도
----	------	------	-----	-----

계정 관리	1.1	-	Redis null 패스워드 점검	상
	1.2	-	패스워드 복잡도 설정	중
권한 관리	2.1	-	개발 및 운영 시스템 분리 사용	하
	2.2	-	root 권한으로 서버 구동 제한	상
DBMS 보안설정	3.1	-	백업 관리	하
	3.2	-	DB 접속 IP 통제	하
	3.3	-	로그 저장 주기	상
	3.4	-	로그 레벨 설정	상
	3.5	-	관리자 command 보호 설정	중
	3.6	-	DBMS 서버 보안 연결	상
	3.7	-	샘플 및 테스트 파일 제거	하
환경 파일 점검	4.1	-	Redis 환경설정 파일 접근 제한	중
	4.2	-	.rediscli_history 파일 접근 제한	중
	4.3	-	Log 파일 접근 제한	하
보안 패치	5.1	-	보안 패치 적용	상



2. 위험도 및 적용 권고 시기 구분

2.1. 위험도

각 취약점으로 인해 발생 가능한 피해에 대하여 위험도 산정을 통해 상, 중, 하 3단계로 분류함.

[표] 23. 위험도 구분

위험도	내 용	비고
상	관리자 계정 및 주요정보 유출로 인한 치명적인 피해 발생	
중	노출된 정보를 통해 서비스/시스템 관련 추가 정보 유출 발생 우려	
하	타 취약점과 연계 가능한 잠재적인 위협 내재	

2.2. 적용 권고 시기

각 취약점 항목의 위험도 및 그에 대한 대응 조치 과정에서 예상되는 서비스/시스템 영향도를 고려하여 단기, 중기, 장기로 적용 권고시기를 분류함.

[표] 24. 적용 권고 시기 구분

적용 권고 시기	내 용	비고
단기	연계 서비스/시스템 영향도가 낮으며 빠른 조치가 필요한 경우	
기본 (중기)	기본적인 서비스/시스템 영향도 검토가 필요한 경우	
장기	조치로 인한 연계 서비스/시스템 영향이 예상되는 경우	

II. 세부항목 설정

1. Node.js

1.1. 설정

1.1.1. 데몬 관리

분류	설정	중요도	상
항목명	데몬 관리		
항목 설명	Node 어플리케이션이 root 권한으로 구동, 또는 개발 모드로 구동될 경우 공격자가 어플리케이션의 에러나 버그를 악용하여 전체 시스템을 작동 불능상태에 빠뜨릴 수 있으므로 Node 어플리케이션이 root 권한으로 구동되지 않도록 관리해야 함.		
설정 방법	1. root 권한으로 어플리케이션을 구동하지 않도록 관리 아래 예시와 같은 형태로 어플리케이션 구동 금지 <pre># sudo node app.js // 금지 - root 권한으로 어플리케이션 구동</pre> 2. 요청을 전달할 HTTP server/proxy 형태로 구성 Apache, nginx와 같은 HTTP 서버를 통해 요청을 전달받도록 어플리케이션 구성 3. NODE_ENV 설정을 production 모드로 설정 <pre># NODE_ENV="production" node app.js</pre> 또는 <pre># export NODE_ENV="production" # node app.js</pre>		
진단 기준	양호 - Node 어플리케이션이 전용 WAS Server 계정으로 구동중이며 production 모드로 설정된 경우 취약 - Node 어플리케이션이 root 권한으로 구동중이거나 development 모드로 설정된 경우		
진단 방법	[진단예시] <pre># ps -ef grep node</pre>		
비고	장기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목			

1.1.2. 로그 디렉토리/파일 권한 설정

분류	설정	중요도	중
항목명	로그 디렉토리/파일 권한 설정		
항목	로그 파일에는 공격자에게 유용한 정보가 들어있어 권한 관리가 필요하므로 일반 사용자에게 의한 정보		

설명	유출이 불가능 하도록 설정을 강화 해야 함.
설정 방법	* 로그 디렉토리/파일은 Node 어플리케이션과 다른 경로로 설정, 생성되도록 하여 어플리케이션 안정성을 높이고 일반 사용자의 접근 및 쓰기 권한을 제한 해야 함 1. 로그 디렉토리/파일 권한 설정 <pre># chown nodeapp:node /[구동중인 Node 어플리케이션 로그 디렉토리] # chmod 750 /[구동중인 Node 어플리케이션 로그 디렉토리] # chown nodeapp:node /[구동중인 Node 어플리케이션 로그 디렉토리]/* # chmod 640 /[구동중인 Node 어플리케이션 로그 디렉토리]/*</pre>
진단 기준	양호 - 전용 WAS Server 계정 소유이고, 디렉토리 750(drwxr-x---) / 파일 640(-rw-r-----) 권한인 경우 취약 - 전용 WAS Server 계정 소유가 아니거나, 디렉토리 752(drwxr-x--w-) / 파일 642(-rw-r---w-) 이상인 경우
진단 방법	[진단예시] <pre># ls -lad /[Node 어플리케이션 로그 디렉토리] # ls -la /[Node 어플리케이션 로그 디렉토리]/*</pre>
비고	단기 적용(적용 시 개발자 및 운영자 협의)
기반시설 기준항목	

1.1.3. 로그 포맷 설정

분류	설정	중요도	상
항목명	로그 포맷 설정		
항목 설명	로그 포맷을 설정하지 않으면, 공격 여부 파악, 공격자 사용 툴 파악, 공격자 위치 파악이 불가능하므로 반드시 로그 포맷을 설정해야 함.		
설정 방법	* Node의 기본 console.log 또는 Express 4.x 이전 버전으로 지원 가능한 로그 출력 기능 외 morgan 등의 추가 로거 모듈을 사용하는 경우 해당되며, 파일의 형태가 아닌 콘솔 출력을 의미함. 어플리케이션 코드 내에서 제어하는 형태로 사용. 1. 모든 요청에 대해 combined 포맷의 로그를 출력하도록 설정 <pre>var express = require('express') var morgan = require('morgan') var app = express() app.use(morgan('combined')) app.get('/', function (req, res) { res.send('hello, world!') })</pre> * 로그 포맷 지시자 - combined : 표준 Apache combined 로그 출력 :remote-addr - :remote-user [:date[clf]] ":method :url HTTP/:http-version" :status :res[content-length] ":referrer" ":user-agent" - common : 표준 Apache common 로그 출력 :remote-addr - :remote-user [:date[clf]] ":method :url HTTP/:http-version" :status :res[content-length] - dev : 개발을 위해 response에 따라 색상이 입혀진 축약 로그 출력 :method :url :status :response-time ms - :res[content-length] - short : 기본 설정보다 짧은 로그 출력, 응답 시간 포함 :remote-addr :remote-user :method :url HTTP/:http-version :status :res[content-length] - :response-time ms - tiny : 최소화된 로그 출력 :method :url :status :res[content-length] - :response-time ms * 사전에 morgan 모듈 설치를 전제함		
진단 기준	양호 - 로그포맷 설정 값이 combined 이거나 그에 준하는 포맷 토큰 조합으로 설정되어 있는 경우 취약 - 로그포맷 설정 값이 combined가 아니거나 그에 준하지 않는 포맷 토큰 조합으로 설정되어 있는 경우		
진단 방법	[진단예시] 어플리케이션 코드 내 log 관련 내용 확인, 또는 개발자 인터뷰		

비고	중기 적용(적용 시 개발자 및 운영자 협의)
기반시설 기준항목	



1.1.4. 로그 저장 주기

분류	설정	중요도	상
항목명	로그 저장 주기		
항목 설명	'정보통신망이용촉진및정보보호등에관한법률', '개인정보보호법', '회사사규' 등에 따라 로그 파일은 최소 6개월 이상의 기간은 보관해야하며, 담당자는 로그 기록을 정기적으로 백업·확인·감독 하여야 함.		
설정 방법	1. 접속 기록 보유 기간은 사업 환경에 따라 조정 할 수 있으나, '정보 통신망 이용 촉진 및 정보보호 등에 관한 법률', '개인정보보호법', '회사사규' 등에 따라 최소 아래 기간 이상은 보관 해야 함.		
	1) 사용자접속기록		
	사용자 로그인/로그아웃/정보변경 등	6개월이상	
	2) 개인정보취급자의개인정보처리시스템접속기록		
	정보주체 식별정보/개인정보취급자 식별정보/ 접속일시/접속지 정보/ 부여된 권한 유형에 따른 수행업무 등	2년이상	
설정 방법	3) 개인정보취급자권한변경기록		
	개인정보취급자 권한 생성/변경/삭제 등	5년이상	
	2. 담당자는 접속 기록을 월 1회 이상 정기적으로 확인·감독하여, 접속과 관련 된 오류 및 부정행위가 발생하거나 예상 되는 경우 즉각적인 보고 조치가 되도록 해야 함		
	3 접속 기록이 위·변조 되지 않도록 별도의 물리적인 저장 장치에 보관하여야 하며 정기적인 백업을 수행 해야 함		
	※ 수정이 가능해야 할 경우, 위·변조 여부를 확인 할 수 있도록 별도 보호조치 - 위·변조 여부를 확인할 수 있는 정보(HMAC값 또는 전자서명값 등)는 별도의 HDD 또는 관리대장에 보관 하는 방법으로 관리		
진단 기준	양호 - 로그 저장 주기 기준에 맞게 운영 중인 경우 취약 - 로그 저장 주기 기준에 맞게 운영 중인 경우		
진단 방법	[진단예시] 서버 운영 또는 담당자 인터뷰		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목			

1.1.5. 헤더 정보 노출 방지

분류	설정	중요도	하
항목명	헤더 정보 노출 방지		
항목 설명	HTTP 요청에 대한 응답 시에 헤더에 서버의 이름, 버전 등의 정보를 제공하는 경우, 공격자가 해당 정보를 이용해 공격에 이용할 수 있음.		
설정 방법	* Node의 헤더 관련 개별 설정 방법 외 helmet과 보안 관련 HTTP 헤더 모듈을 사용한 경우엔 해당됨 1. Node 어플리케이션 코드 내에 모듈 사용 설정 <pre>var express = require('express'); var helmet = require('helmet'); var app = express(); app.use(helmet());</pre> Apache, nginx 등 연계 구성한 Web Server 환경 설정을 통해서도 동일한 설정 적용 가능 * 사전에 helmet 모듈 설치를 전제함 ※ 서비스 및 운영상의 영향도를 확인하시어 설정하여 적용해야 합니다.		
진단 기준	양호 - 헤더 정보 노출 방지 모듈 적용, 또는 코드로 동작 중인 경우 취약 - 헤더 정보 노출 방지 모듈 미적용, 헤더 정보 노출 코드로 동작 중인 경우		
진단 방법	[진단예시] 어플리케이션 코드 내 헤더 설정 관련 내용 확인, 또는 개발자 인터뷰		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목			

1.2. 보안 패치

1.2.1. 보안 패치 적용

분류	보안 패치	중요도	상
항목명	보안 패치 적용		
항목 설명	주기적으로 보안 패치를 적용하지 않으면 exploit 공격, 제로데이 공격 등의 서버 침해가 발생할 수 있음.		
설정 방법	1. 기간 산정해서 보안 패치 적용 (정기 PM 등, 2017년 12월 기준)		
	Node.js 버전	권고 기준	Release 일자
	Node v8.x	8.9.3 이상	2017-12-08
	Node v7.x	7.9.0 이상	2017-04-11
	Node v6.x	6.10.2 (LTS) 이상	2017-02-22
	Node v5.x	5.12.0 (Stable)	2016-06-23
	Node v4.x	4.8.0 (LTS)	2017-02-22
	Node v0.12.x	0.12.18	2017-02-22
	Node v0.10.x	0.10.48	2016-10-18
	* Node.js 최신 패치(https://nodejs.org/en/download/)		
설정 방법	2. Express 모듈 추가 사용 시 (2017년 3월 기준)		
	Express 버전	권고 기준	Release 일자
	Express 4.x	4.15.2	2017-03-06
	Express 3.x	3.19.1	-
	Express 2.x	-	-
	* Express : MVC Framework 사용 목적으로 Node.js에 추가하는 모듈		
	※ Node.js 0.10.42 이전 0.10.x, 0.12.10 이전 0.12.x, 4.3.0 이전 4.x, 5.6.0 이전 5.x 버전들에서 원격 공격자의 조작된 Content-Length HTTP 헤더를 통한 HTTP request smuggling 공격을 허용할 수 있음 (CVE-2016-2086, 2016.04.07)		
	3. SSL 관련 CVE 점수와 관계없이 영향도가 높은 CVE 코드 표		
	CVE 코드	CVSS	내 용
	CVE-2014-3566	4.3	SSLv3.0 프로토콜 관련 POODLE 취약점
	CVE-2015-0204	4.3	OpenSSL FREAK(Factoring attack on RSA-EXPORT Keys) 취약점 관련
	CVE-2016-0800	5.9	DROWN(Decrypting RSA using Obsolete Weakened eNcryption)으로 명명된, SSLv2를 이용한 TLS에 대한 프로토콜 간 공격 취약점 관련
	CVE-2017-3733	5.0	OpenSSL 1.1.0 버전에서의 서비스 거부(Denial-of-Service) 공격 취약점 관련
	CVE-2017-3737	4.3	OpenSSL 1.0.2 버전부터 1.0.2n하위버전까지의 서비스 거부(Denial-of-Service) 공격 취약점 관련
※ 서비스 및 운영상의 영향도를 확인하시어 설정하여 적용해야 합니다.			

진단 기준	<u>양호</u> - Node.js 및 Express 사용 시 권고 기준 이상의 버전을 사용할 경우 <u>취약</u> - Node.js 및 Express 사용 시 권고 기준 미만의 버전을 사용할 경우
진단 방법	[진단예시] <pre># /[Node.js 설치 디렉토리]/node -v # /[Node.js 설치 디렉토리]/express -V</pre>
비고	중기 적용(적용 시 개발자 및 운영자 협의)
기반시설 기준항목	



2. Wildfly

2.1. 설정

2.1.1. 데몬관리

분류	설정	중요도	상
항목명	데몬 관리		
항목 설명	Unix 시스템의 경우, WAS 데몬이 root 권한으로 운영될 경우 Web Application의 취약점이나 Buffer Overflow시 공격자에게 root권한을 유출할 수 있으므로 WildFly 서버 데몬이 root 권한으로 운영되지 않도록 관리해야 함.		
설정 방법	1. WildFly 데몬 기동을 위한 계정을 별도로 관리 <pre>[root@localhost ~]# ps -ef grep wildfly root 1056 1 0 13:52 ? 00:00:00 /bin/sh /etc/rc.d/init.d/wildfly start root 1061 1056 0 13:52 ? 00:00:00 runuser -s /bin/bash wildfly -c ulimit -S -c 0 >/dev/null 2>&1 ; LAUNCH_JBOSS_IN_BACKGROUND=1 JBOSS_PIDFILE=/var/run/wildfly/wildfly.pid /opt/wildfly/bin/standalone.sh -c standalone.xml wildfly 1062 1061 0 13:52 ? 00:00:00 bash -c ulimit -S -c 0 >/dev/null 2>&1 ; LAUNCH_JBOSS_IN_BACKGROUND=1 JBOSS_PIDFILE=/var/run/wildfly/wildfly.pid /opt/wildfly/bin/standalone.sh -c standalone.xml wildfly 1063 1062 0 13:52 ? 00:00:00 /bin/sh /opt/wildfly/bin/standalone.sh -c standalone.xml wildfly 1106 1063 0 13:52 ? 00:00:33 java -D[Standalone] -server -Xms64m -Xmx512m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=org.jboss.byteman -Djava.awt.headless=true -Dorg.jboss.boot.log.file=/opt/wildfly/standalone/log/server.log -Dlogging.configuration=file:/opt/wildfly/standalone/configuration/logging.properties -jar /opt/wildfly/jboss-modules.jar -mp /opt/wildfly/modules org.jboss.as.standalone -Djboss.home.dir=/opt/wildfly -Djboss.server.base.dir=/opt/wildfly/standalone -c standalone.xml root 11746 2370 0 18:07 pts/0 00:00:00 grep --color=auto wildfly</pre>		
진단 기준	양호 - 구동중인 WildFly 데몬의 계정이 전용 WAS Server 계정 인 경우 취약 - 구동중인 WildFly 데몬의 계정이 root 인 경우		
진단 방법	[진단예시] <pre># ps -ef grep wildfly</pre>		
비고	장기 적용(적용 시 개발자 및 운영자 협의)		

2.1.2. 관리서버 디렉토리 권한 설정

분류	설정	중요도	중
항목명	관리서버 디렉토리 권한 설정		
항목 설명	관리서버 홈디렉토리에 일반 사용자가 접근할 수 없도록 권한 관리가 필요함.		
설정 방법	1. 관리서버 홈디렉토리 권한 변경 [Unix - Standalone Mode] - WildFly # chmod 750 [Djboss.home.dir]/standalone # chown wildfly:wildfly [Djboss.home.dir]/standalone [Unix - Domain Mode] - WildFly # chmod 750 [Djboss.home.dir]/domain # chown wildfly:wildfly [Djboss.home.dir]/domain [Window - Standalone Mode] - [Djboss.home.dir]/standalone 디렉토리의 속성 설정 1) Administrator 또는 전용 WAS Server 계정으로 소유자 변경 2) 전용 WAS Server 계정 그룹(Administrator) - 모든 권한 3) Users 그룹 - 쓰기 권한 제거 4) Everyone 그룹 - 그룹 제거 [Window - Domain Mode] - [Djboss.home.dir]/domain 디렉토리의 속성 설정 1) Administrator 또는 전용 WAS Server 계정으로 소유자 변경 2) 전용 WAS Server 계정 그룹(Administrator) - 모든 권한 3) Users 그룹 - 쓰기 권한 제거 4) Everyone 그룹 - 그룹 제거		
진단 기준	양호 - 전용 WAS Server 계정 소유이고, 750(drwxr-x---) 권한인 경우 취약 - 전용 WAS Server 계정 소유가 아니거나, 752(drwxr-x-w-) 이상의 권한인 경우		
진단 방법	[진단예시] [Unix - Standalone Mode] <pre># ls -ald [Djboss.home.dir]/standalone</pre> [Unix - Domain Mode] <pre># ls -ald [Djboss.home.dir]/domain</pre>		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

2.1.3. 설정파일 권한 설정

분류	설정	중요도	상
항목명	설정파일 권한 설정관리		
항목 설명	일반 사용자가 웹 서버의 설정 파일을 삭제, 변경할 수 있을 경우 시스템이 오작동하여 사용 불능 상태에 빠질 우려가 있음.		
설정 방법	1. 설정파일 권한 변경 [Unix - Standalone Mode] <pre># chown wildfly:wildfly [Djboss.home.dir]/standalone/configuration/*.xml # chown wildfly:wildfly [Djboss.home.dir]/standalone/configuration/*.properties # chmod 600 [Djboss.home.dir]/standalone/configuration/*.xml //또는 700 # chmod 600 [Djboss.home.dir]/standalone/configuration/*.properties //또는 700</pre> [Unix - Domain Mode] <pre># chown wildfly:wildfly [Djboss.home.dir]/domain/configuration/*.xml # chown wildfly:wildfly [Djboss.home.dir]/domain/configuration/*.properties # chmod 600 [Djboss.home.dir]/domain/configuration/*.xml //또는 700 # chmod 600 [Djboss.home.dir]/domain/configuration/*.properties //또는 700</pre> [Window - Standalone Mode] [Djboss.home.dir]/standalone/configuration/*.xml [Djboss.home.dir]/standalone/configuration/*.properties 1) Administrator 또는 전용 WAS Server 계정으로 소유자 변경 2) 전용 WAS Server 계정 그룹(Administrator) - 모든 권한 3) Users, Everyone 그룹 - 그룹 제거 [Window - Domain Mode] [Djboss.home.dir]/domain/configuration/*.xml [Djboss.home.dir]/domain/configuration/*.properties 1) Administrator 또는 전용 WAS Server 계정으로 소유자 변경 2) 전용 WAS Server 계정 그룹(Administrator) - 모든 권한 3) Users, Everyone 그룹 - 그룹 제거 * 설정 파일의 Backup은 삭제 (Backup 파일 필요시에는 설정 파일과 동일한 권한 설정)		
진단 기준	양호 - 전용 WAS Server 계정 소유이고, 600(-rw-----) 또는 700(-rwx-----) 권한인 경우 취약 - 전용 WAS Server 계정 소유가 아니거나, 600(-rw-----) 또는 700(-rwx-----) 권한 초과인 경우		
진단 방법	[진단예시] [Unix - Standalone Mode] <pre># find [Djboss.home.dir]/standalone/configuration/ -name *.xml -ls -o -name *.properties -ls</pre>		

	<pre>[Unix - Domain Mode] # find [Djboss.home.dir]/domain/configuration/ -name *.xml -ls -o -name *.properties -ls</pre>
비고	중기 적용(적용 시 개발자 및 운영자 협의)



2.1.4. 로그 디렉토리/파일 권한 설정

분류	설정	중요도	중
항목명	로그 디렉토리/파일 권한 설정		
항목 설명	로그 파일에는 공격자에게 유용한 정보가 들어있어 권한 관리가 필요하므로 일반 사용자에게 의한 정보 유출이 불가능 하도록 설정을 강화 해야 함.		
설정 방법	1. 로그 디렉토리/파일 권한 변경 [Unix - Standalone Mode] <pre># chown wildfly:wildfly [Djboss.home.dir]/standalone/log # chmod 750 [Djboss.home.dir]/standalone/log # chown wildfly:wildfly [Djboss.home.dir]/standalone/log/[로그 파일] # chmod 640 [Djboss.home.dir]/standalone/log/[로그 파일]</pre> [Unix - Domain Mode] <pre># chown wildfly:wildfly [Djboss.home.dir]/domain/log # chmod 750 [Djboss.home.dir]/domain/log # chown wildfly:wildfly [Djboss.home.dir]/domain/log/[로그 파일] # chmod 640 [Djboss.home.dir]/domain/log/[로그 파일]</pre> [Window - Standalone Mode] - 아래 로그 디렉토리 및 로그 파일의 권한 설정 [Djboss.home.dir]/standalone/log [Djboss.home.dir]/standalone/log/[로그 파일] 1) Administrator 또는 전용 WAS Server 계정으로 소유자 변경 2) 전용 WAS Server 계정 그룹(Administrator) - 모든 권한 3) Users 그룹 - 쓰기 권한 제거 4) Everyone 그룹 - 그룹 제거 [Window - Domain Mode] - 아래 로그 디렉토리 및 로그 파일의 권한 설정 [Djboss.home.dir]/domain/log [Djboss.home.dir]/domain/log/[로그 파일] 1) Administrator 또는 전용 WAS Server 계정으로 소유자 변경 2) 전용 WAS Server 계정 그룹(Administrator) - 모든 권한 3) Users 그룹 - 쓰기 권한 제거 4) Everyone 그룹 - 그룹 제거 * 설정 파일 또는 데몬 구동 시 Custom Log 설정하여 운영 시 동일한 권한으로 설정		
진단 기준	양호 - 전용 WAS Server 계정 소유이고, 디렉토리는 750(drwxr-x---) / 파일은 640(-rw-r-----) 권한인 경우 취약 - 전용 WAS Server 계정 소유가 아니거나, 디렉토리는 752(drwxr-x--w-) / 파일은 642(-rw-r---w-) 이상인 경우		
진단 방법	[진단예시] [Unix - Standalone Mode]		

	<pre># ls -ald [Djboss.home.dir]/standalone/log # ls -la [Djboss.home.dir]/standalone/log [Unix - Domain Mode] # ls -ald [Djboss.home.dir]/domain/log # ls -la [Djboss.home.dir]/domain/log</pre>
비고	단기 적용(적용 시 개발자 및 운영자 협의)



2.1.5. 로그 포맷 설정

분류	설정	중요도	상
항목명	로그 포맷 설정		
항목 설명	로그 포맷을 설정하지 않으면, 공격 여부 파악, 공격자 사용 툴 파악, 공격자 위치 파악이 불가능하므로 반드시 로그 포맷을 설정해야함.		
설정 방법	1. 해당 설정파일에 pattern 값 설정 [Unix - Standalone Mode] # vi [Djboss.home.dir]/standalone/configuration/standalone.xml <pre> formatter.PATTERN=org.jboss.logmanager.formatters.PatternFormatter formatter.PATTERN.properties=pattern formatter.PATTERN.pattern=%d{yyyy-MM-dd HHW:mmW:ss,SS} %-5p [%c] (%t) %s%e%n formatter.COLOR-PATTERN=org.jboss.logmanager.formatters.PatternFormatter formatter.COLOR-PATTERN.properties=pattern formatter.COLOR-PATTERN.pattern=%K{level} %d{HHW:mmW:ss,SS} %5-p [%c] (%t) %s%e%n </pre> [Unix - Domain Mode] # vi [Djboss.home.dir]/domain/configuration/domain.xml <pre> formatter.PATTERN=org.jboss.logmanager.formatters.PatternFormatter formatter.PATTERN.properties=pattern formatter.PATTERN.pattern=%d{yyyy-MM-dd HHW:mmW:ss,SS} %-5p [%c] (%t) %s%E%n formatter.COLOR-PATTERN=org.jboss.logmanager.formatters.PatternFormatter formatter.COLOR-PATTERN.properties=pattern formatter.COLOR-PATTERN.pattern=%K{level} %d{HHW:mmW:ss,SS} %5-p [%c] (%t) %s%E%n </pre> * 포맷 지시자 Combined (default) : %d{yyyy-MM-dd HH:mm:ss,SSS} %-5p [%c] (%t) %s%e%n * 로그 포맷 스트링 %c The category of the logging event %p The level of the log entry (info/debug/etc) %P The localized level of the log entry %d The current date/time (yyyy-MM-dd HH:mm:ss,SSS form) %r The relative time (milliseconds since the log was initialized) %z The time zone %k A log resource key (used for localization of log messages) %m The log message (including exception trace) %s The simple log message (no exception trace) %e The exception stack trace (no extended module information)		

	%E The exception stack trace (with extended module information) %t The name of the current thread %n A newline character %C The class of the code calling the log method (slow) %F The filename of the class calling the log method (slow) %l The source location of the code calling the log method (slow) %L The line number of the code calling the log method (slow) %M The method of the code calling the log method (slow) %x The Nested Diagnostic Context %X The Message Diagnostic Context %% A literal percent character (escaping)
진단 기준	양호 - 로그포맷 설정 값이 그에 준하는 포맷 스트링으로 설정되어 있는 경우 취약 - 로그포맷 설정 값이 그에 준하지 않는 포맷 스트링으로 설정되어 있는 경우
진단 방법	[진단예시] [Unix - Standalone Mode] <pre># cat [Djboss.home.dir]/standalone/configuration/standalone.xml grep pattern</pre> [Unix - Domain Mode] <pre># cat [Djboss.home.dir]/domain/configuration/domain.xml grep pattern</pre>
비고	단기 적용(적용 시 개발자 및 운영자 협의)

2.1.6. 로그 저장 주기

분류	설정	중요도	상
항목명	로그 저장 주기		
항목 설명	'정보통신망이용촉진및정보보호등에관한법률', '개인정보보호법', '회사사규' 등에 따라 로그 파일은 최소 6개월 이상의 기간은 보관해야하며, 담당자는 로그 기록을 정기적으로 백업·확인·감독 하여야 함.		
설정 방법	1. 접속 기록 보유 기간은 사업 환경에 따라 조정 할 수 있으나, '정보 통신망 이용 촉진 및 정보보호 등에 관한 법률', '개인정보보호법', '회사사규' 등에 따라 최소 아래 기간 이상은 보관 해야 함.		
	4) 사용자접속기록		
	사용자 로그인/로그아웃/정보변경 등	6개월이상	
	5) 개인정보취급자의개인정보처리시스템접속기록		
	정보주체 식별정보/개인정보취급자 식별정보/ 접속일시/접속지 정보/ 부여된 권한 유형에 따른 수행업무 등	2년이상	
설정 방법	6) 개인정보취급자권한변경기록		
	개인정보취급자 권한 생성/변경/삭제 등	5년이상	
	2. 담당자는 접속 기록을 월 1회 이상 정기적으로 확인·감독하여, 접속과 관련 된 오류 및 부정행위가 발생하거나 예상 되는 경우 즉각적인 보고 조치가 되도록 해야 함		
설정 방법	3 접속 기록이 위·변조 되지 않도록 별도의 물리적인 저장 장치에 보관하여야 하며 정기적인 백업을 수행 해야 함		
	※ 수정이 가능해야 할 경우, 위·변조 여부를 확인 할 수 있도록 별도 보호조치		
	- 위·변조 여부를 확인할 수 있는 정보(HMAC값 또는 전자서명값 등)는 별도의 HDD 또는 관리대장에 보관 하는 방법으로 관리		
진단 기준	양호 - 로그 저장 주기 기준에 맞게 운영 중이면 취약 - 로그 저장 주기 기준에 맞게 운영 중이 아니면		
진단 방법	[진단예시] 서버 운영 또는 담당자에게 문의		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

2.1.7. HTTP Method 제한

분류	설정	중요도	하
항목명	HTTP Method 제한		
항목 설명	OPTIONS, GET, POST 이외의 다른 HTTP Method 를 지원하는 경우, 악의적인 공격자가 임의의 파일을 삭제하거나 업로드하여 서버의 정상 운영에 지장을 줄 수 있음.		

설정 방법	1. Standalone.xml, Domain.xml 파일에서 Method 설정을 아래와 같이 제한 <pre>[Unix - Standalone Mode] # vi [Djboss.home.dir]/standalone/configuration/standalone.xml <subsystem xmlns="urn:jboss:domain:undertow:X.X"> ... <server name="default-server"> ... <http-listener name="default" socket-binding="http" disallowed-methods="TRACE, PUT, DELETE" redirect-socket="https" enable-http2="true"/> <https-listener name="default" socket-binding="https" disallowed-methods="TRACE, PUT, DELETE" security-realm="ApplicationRealm" enable-http2="true"/> ... </server> </subsystem></pre> <pre>[Unix - Domain Mode] # vi [Djboss.home.dir]/domain/configuration/domain.xml <subsystem xmlns="urn:jboss:domain:undertow:X.X"> ... <server name="default-server"> ... <http-listener name="default" socket-binding="http" disallowed-methods="TRACE, PUT, DELETE" redirect-socket="https" enable-http2="true"/> <https-listener name="default" socket-binding="https" disallowed-methods="TRACE, PUT, DELETE" security-realm="ApplicationRealm" enable-http2="true"/> ... </server> </subsystem></pre> 2. 설정파일에 설정이 없는 경우 기본적으로 PUT, DELETE, TRACE Method 를 제한 ※ 서비스 및 운영상의 영향도를 확인하시어 설정하여 적용해야 합니다.
진단 기준	양호 - 불필요한 HTTP-Method 제한 설정이 되어있는 경우 취약 - 불필요한 HTTP-Method 제한 설정이 되어있지 않은 경우
진단 방법	[진단예시] <pre>[Unix - Standalone Mode] # cat [Djboss.home.dir]/standalone/configuration/standalone.xml grep disallowed- methods</pre> <pre>[Unix - Domain Mode] # cat [Djboss.home.dir]/domain/configuration/domain.xml grep disallowed-methods</pre>
비고	단기 적용(적용 시 개발자 및 운영자 협의)

2.1.8. 디렉토리 검색 기능 제거

분류	설정	중요도	중
항목명	디렉토리 검색 기능 제거		
항목 설명	디렉토리 검색 기능이 활성화 되어 있으면 해당 디렉토리에 존재하는 모든 파일 리스트를 보여주어, WAS 서버 구조 노출 및 주요 설정파일의 내용이 유출될 가능성이 있음.		
설정 방법	1 해당 설정파일에 디렉토리 검색 기능 제거 설정 <pre>[Unix - Standalone Mode] # vi [Djboss.home.dir]/standalone/configuration/standalone.xml <subsystem xmlns="urn:jboss:domain:undertow:X.X"> ... <handlers> <file name="welcome-content" path="\${jboss.home.dir}/welcome-content" directory-listing="false" /> </handlers> </subsystem></pre> <pre>[Unix - Domain Mode] # vi [Djboss.home.dir]/domain/configuration/domain.xml <subsystem xmlns="urn:jboss:domain:undertow:X.X"> ... <handlers> <file name="welcome-content" path="\${jboss.home.dir}/welcome-content" directory-listing="false" /> </handlers> </subsystem></pre> 2. Jboss 디렉토리 검색 기능은 기본적으로 false로 설정 적용되어 있음. ※ 서비스 및 운영상의 영향도를 확인하시어 설정하여 적용해야 합니다.		
진단 기준	양호 - “directory-listings” param-name의 값이 false인 경우 취약 - “directory-listings” param-name의 값이 false가 아닌 경우		
진단 방법	[진단예시] <pre>[Unix - Standalone Mode] # cat [Djboss.home.dir]/standalone/configuration/standalone.xml grep directory-listing</pre> <pre>[Unix - Domain Mode] # cat [Djboss.home.dir]/domain/configuration/domain.xml grep directory-listing</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

2.1.9. 데이터소스의 패스워드 암호화

분류	설정	중요도	중
항목명	데이터소스의 패스워드 암호화		
항목 설명	Application의 DB 사용의 효율성을 제공하기 위해 DB 접속정보를 저장하여 해당 설정파일이 유출 될 경우 DB 접속 정보까지 유출될 수 있음.		
설정 방법	[Unix – Standalone Mode] 1. 데이터소스 패스워드 암호화 <pre># java -cp [Djboss.home.dir]/modules/system/layers/base/org/picketbox/main/picketbox-x.x.x.Final.jar org.picketbox.datasource.security.SecureIdentityLoginModule [Datasource-password]</pre> 암호화처리 예시) => Encoded password: [Datasource-enc-password] 2. 설정 파일 내 보안 도메인 추가 [Djboss.home.dir]/standalone/configuration/standalone.xml 설정 파일에 암호화 한 데이터소스 패스워드가 작성되어 있는 보안 도메인 설정 추가 <pre># vi [Djboss.home.dir]/standalone/configuration/standalone.xml ... 중략... <subsystem xmlns="urn:jboss:domain:security:2.0"> <security-domains> <security-domain name="EncryptPassword" cache-type="default"> <authentication> <login-module name="encrypt-ds-domain" code="org.picketbox.datasource.security.SecureIdentityLoginModule" flag="required"> <module-option name="username" value="[Datasource-username]" /> <module-option name="password" value="[Datasource-enc-password]" /> </login-module> </authentication> </security-domain> </security-domains> </subsystem> ... 중략...</pre> 3. 위에서 설정한 해당 보안 도메인을 개별 데이터소스 설정 내 추가 <pre># vi [Djboss.home.dir]/standalone/configuration/standalone.xml ... 중략... <subsystem xmlns="urn:jboss:domain:databases:5.0"> <databases> <datasource jndi-name="java:jboss/databases/ExampleDS" pool-name="ExmampleDS" enabled="true" use-java-context="true"> <connection-url></pre>		

```

    jdbc: h2: mem: test; DB_CLOSE_DELAY=-1; DB_CLOSE_ON_EXIT=FALSE
  </connection-url>
  <driver>h2</driver>
  <security>
    <security-domain>EncryptPassword</security-domain>
  </security>
</datasource>
... 중략...
</datasources>
</subsystem>

```

[Unix - Domain Mode]

1. 데이터소스 패스워드 암호화

```
# java -cp [Djboss.home.dir]/modules/system/layers/base/org/picketbox/main/picketbox-
x.x.x.Final.jar org.picketbox.datasource.security.SecureIdentityLoginModule [Datasource-
password]
```

암호화처리 예시) => Encoded password: [Datasource-enc-password]

2. 설정 파일 내 보안 도메인 추가

[Djboss.home.dir]/domain/configuration/domain.xml 설정 파일에 암호화 한 데이터 소스
패스워드가 작성되어 있는 보안 도메인 설정 추가

```

# vi [Djboss.home.dir]/domain/configuration/domain.xml
... 중략...

<subsystem xmlns="urn:jboss:domain:security:2.0">
  <security-domains>
    <security-domain name="EncryptPassword" cache-type="default">
      <authentication>
        <login-module name="encrypt-ds-domain"
code="org.picketbox.datasource.security.SecureIdentityLoginModule" flag="required">
          <module-option name="username" value="[Datasource-username]" />
          <module-option name="password" value="[Datasource-enc-password]" />
        </login-module>
      </authentication>
    </security-domain>
  </security-domains>
</subsystem>

... 중략...

```

3. 위에서 설정한 해당 보안 도메인을 개별 데이터소스 설정 내 추가

```

# vi [Djboss.home.dir]/domain/configuration/domain.xml

... 중략...

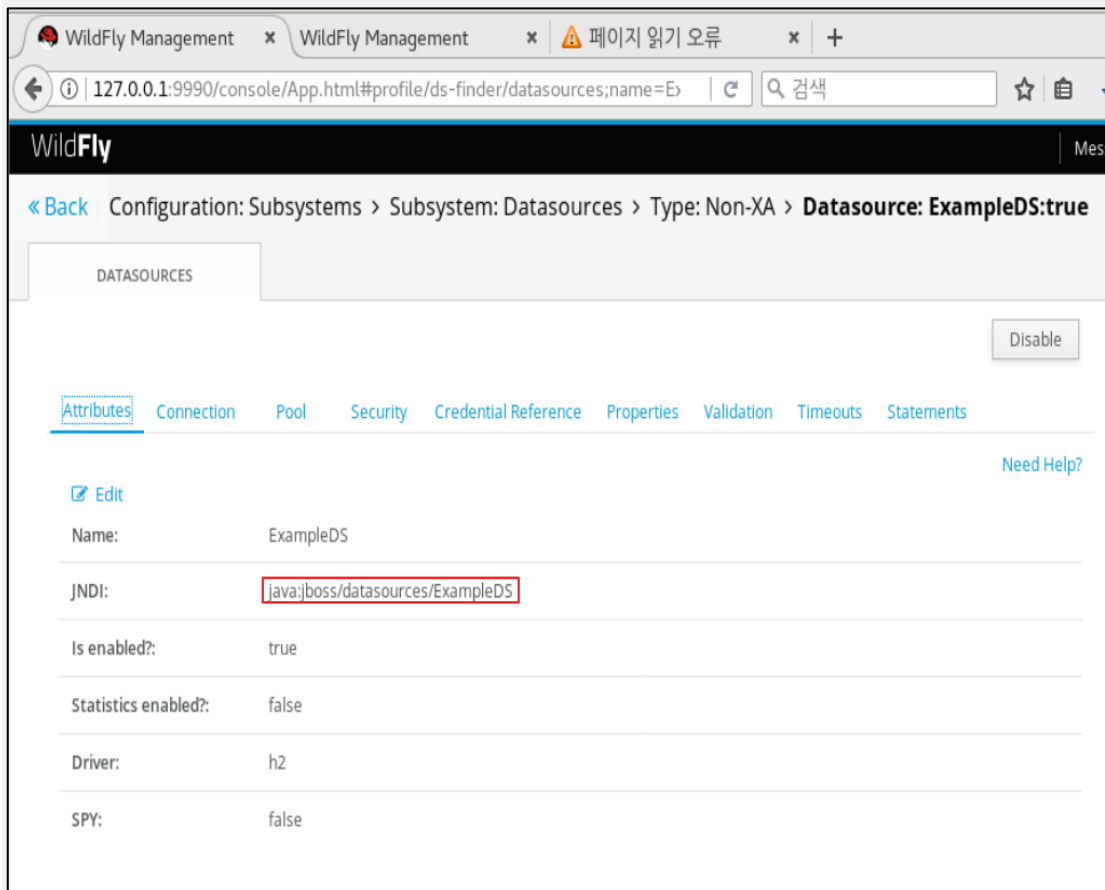
<subsystem xmlns="urn:jboss:domain:datasources:5.0">
  <datasources>

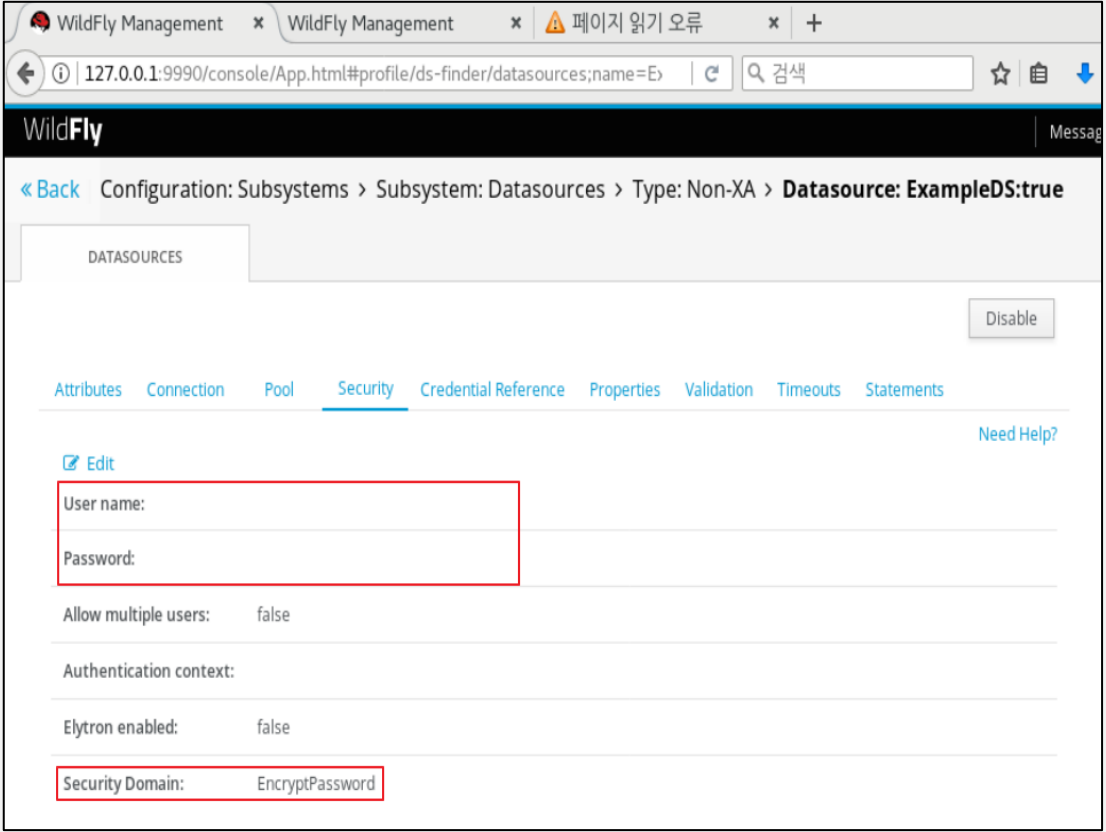
```

```

<datasource jndi-name="java:jboss/datasources/ExampleDS" pool-
name="ExmampleDS" enabled="true" use-java-context="true">
  <connection-url>
    jdbc: h2: mem: test; DB_CLOSE_DELAY=-1; DB_CLOSE_ON_EXIT=FALSE
  </connection-url>
  <driver>h2</driver>
  <security>
    <security-domain>EncryptPassword</security-domain>
  </security>
</datasource>
... 중략...
</datasources>
</subsystem>

```



	 WildFly Management console screenshot showing the configuration for a Datasource. The 'Security' tab is selected, and the 'Security Domain' is set to 'EncryptPassword'. The 'User name' and 'Password' fields are highlighted with a red box.
진단 기준	양호 - 별도의 보안 도메인을 생성 후 패스워드를 암호화 하여 사용할 경우 취약 - 별도의 보안 도메인을 생성 후 패스워드를 암호화 하여 사용하지 않는 경우
진단 방법	[진단예시] [Unix - Standalone Mode] <pre># cat [Djboss.home.dir]/standalone/configuration/standalone.xml grep password</pre> [Unix - Domain Mode] <pre># cat [Djboss.home.dir]/domain/configuration/domain.xml grep password</pre>
비고	단기 적용(적용 시 개발자 및 운영자 협의)

2.1.10. Session Timeout 설정

분류	설정	중요도	중
항목명	Session Timeout 설정		
항목 설명	Session Timeout 이 설정되어 있지 않을 경우, 악의적인 공격자에 의해 세션을 가로채어 허용되지 않는 페이지에 비정상적 공격 및 접근이 가능함.		
설정 방법	1. 해당 설정파일의 Session Timeout 값 60분 이내로 변경 [Unix - Standalone Mode] # vi [Djboss.home.dir]/standalone/configuration/standalone.xml <pre> <subsystem xmlns="urn:jboss:domain:undertow:X.X"> ... <servlet-container name="default" default-session-timeout="60"> </servlet-container> </subsystem> </pre> [Unix - Domain Mode] # vi [Djboss.home.dir]/domain/configuration/standalone.xml <pre> <subsystem xmlns="urn:jboss:domain:undertow:X.X"> ... <servlet-container name="default" default-session-timeout="60"> </servlet-container> </subsystem> </pre> 2. 응용프로그램에 대한 개별적인 설정도 가능하나 서버의 설정 값이 우선시 되며, 장시간 미사용 세션에 대하여 자동 로그오프 하는 기능을 설정하여야 함.		
진단 기준	양호 - timeout 값이 60 이하로 설정되어 있는 경우 취약 - timeout 값이 60 초과로 설정되어 있는 경우		
진단 방법	[진단예시] [Unix - Standalone Mode] # cat [Djboss.home.dir]/standalone/configuration/standalone.xml grep servlet-container [Unix - Domain Mode] # cat [Djboss.home.dir]/domain/configuration/domain.xml grep servlet-container		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

2.1.11. 헤더 정보 노출 방지

분류	설정	중요도	하
항목명	헤더 정보 노출 방지		
항목 설명	HTTP 요청에 대한 응답 시에 헤더에 서버의 이름, 버전 등의 정보를 제공하는 경우, 공격자가 해당 정보를 이용해 공격에 이용할 수 있음.		

1. 설정 파일 내에 SERVER 헤더 값 설정

[Unix - Standalone Mode]

```
# vi [Djboss.home.dir]/standalone/configuration/standalone.xml
```

```
<subsystem xmlns="urn:jboss:domain:undertow:X.X">
    ...
    <filters>
        <response-header name="server-header" header-name="Server" header-
value="Server" />
        <response-header name="x-powered-by-header" header-name="X-Powered-By"
header-value="Server" />
    </filters>
</subsystem>
```

[Unix - Domain Mode]

```
# vi [Djboss.home.dir]/domain/configuration/domain.xml
```

```
<subsystem xmlns="urn:jboss:domain:undertow:X.X">
    ...
    <filters>
        <response-header name="server-header" header-name="Server" header-
value="Server" />
        <response-header name="x-powered-by-header" header-name="X-Powered-By"
header-value="Server" />
    </filters>
</subsystem>
```

2. 설정 파일 내에 X-Powered-By 헤더 값 설정

[Unix - Standalone Mode]

```
# vi [Djboss.home.dir]/standalone/configuration/standalone.xml
```

```
<subsystem xmlns="urn:jboss:domain:undertow:X.X">
    ...
    <filters>
        <response-header name="server-header" header-name="Server" header-
value="Server" />
        <response-header name="x-powered-by-header" header-name="X-Powered-By"
header-value="Server" />
    </filters>
</subsystem>
```

[Unix - Domain Mode]

```
# vi [Djboss.home.dir]/domain/configuration/domain.xml
```

```
<subsystem xmlns="urn:jboss:domain:undertow:X.X">
    ...
    <filters>
```

	<pre> <response-header name="server-header" header-name="Server" header-value=" Server" /> <response-header name="x-powered-by-header" header-name="X-Powered-By" header-value="Server" /> </filters> </subsystem> </pre> ※ 서비스 및 운영상의 영향도를 확인하시어 설정하여 적용해야 합니다.
진단 기준	양호 - Filter 지시어 내 server-header, X-Powered-By 설정 값이 기본 값이 아닐 경우 취약 - Filter 지시어 내 server-header, X-Powered-By 설정 값이 기본 값일 경우
진단 방법	[진단예시] [Unix - Standalone Mode] <pre># cat [Djboss.home.dir]/standalone/configuration/standalone.xml grep response-header</pre> [Unix - Domain Mode] <pre># cat [Djboss.home.dir]/domain/configuration/domain.xml grep response-header</pre>
비고	단기 적용(적용 시 개발자 및 운영자 협의)

2.1.12. 에러 메시지 관리

분류	설정	중요도	중
항목명	에러 메시지 관리		
항목 설명	공격자가 대상 시스템의 정보를 획득하기 위해 고의적으로 다양한 에러를 유발하여 돌아오는 에러 메시지를 통해 웹 프로그램의 구조 및 환경 설정을 추정할 수 있음.		
설정 방법	1. 에러핸들링 설정 값 적용(필수항목 : 400, 401, 403, 404, 500) [Unix – Standalone Mode] <pre># vi [Djboss.home.dir]/standalone/configuration/standalone.xml</pre> <pre><subsystem xmlns="urn:jboss:domain:undertow:X.X"> ... <filter-ref name="400-handler" /> <filter-ref name="401-handler" /> <filter-ref name="403-handler" /> <filter-ref name="404-handler" /> <filter-ref name="500-handler" /> ... <filters> <error-page name="400-handler" code="400" path="error.html" /> <error-page name="401-handler" code="401" path="error.html" /> <error-page name="403-handler" code="403" path="error.html" /> <error-page name="404-handler" code="404" path="error.html" /> <error-page name="500-handler" code="500" path="error.html" /> </filters> </subsystem></pre> ※ 에러발생 시 <filter-ref 속성 내 predicate="true" 값 추가 ? [Unix – Domain Mode] <pre># vi [Djboss.home.dir]/domain/configuration/domain.xml</pre> <pre><subsystem xmlns="urn:jboss:domain:undertow:X.X"> ... <filter-ref name="400-handler" /> <filter-ref name="401-handler" /> <filter-ref name="403-handler" /> <filter-ref name="404-handler" /> <filter-ref name="500-handler" /> ... <filters> <error-page name="400-handler" code="400" path="error.html" /> <error-page name="401-handler" code="401" path="error.html" /> <error-page name="403-handler" code="403" path="error.html" /> <error-page name="404-handler" code="404" path="error.html" /> <error-page name="500-handler" code="500" path="error.html" /> </filters></pre>		

	<pre> </filters> </subsystem> </pre> 2. 에러 페이지 생성 <pre> # vi /에러페이지경로/에러페이지명.html ... 중략... <p>시스템 오류</p> <div class="p" style="margin-bottom:2em"> 이용에 불편을 드려 죄송합니다.
 잠시 후 다시 이용해 주시길 바랍니다. </div> ... 중략... </pre>
진단 기준	양호 - 에러 페이지 설정 및 에러 페이지가 존재할 경우 취약 - 에러 페이지 설정 및 에러 페이지가 없을 경우
진단 방법	[진단예시] <pre> [Unix - Standalone Mode] # cat [Djboss.home.dir]/standalone/configuration/standalone.xml grep error-page [Unix - Domain Mode] # cat [Djboss.home.dir]/domain/configuration/domain.xml grep error-page </pre>
비고	단기 적용(적용 시 개발자 및 운영자 협의)

2.2. 솔루션 취약점

2.2.1. 불필요한 파일 삭제

분류	솔루션 취약점	중요도	하
항목명	불필요한 파일 삭제		
항목 설명	서버에 대한 상세 정보를 제공하고 있고, 예제 프로그램 취약점 공격 예방을 위해서는 삭제하는 것이 바람직함.		
설정 방법	1. 해당 Examples 디렉토리 삭제 <pre># rm -rf [Djboss.home.dir]/docs/examples/</pre>		
진단 기준	양호 - Examples 디렉토리가 존재하지 않는 경우 취약 - Examples 디렉토리가 존재하는 경우		
진단 방법	[진단예시] <pre># find [Djboss.home.dir]/docs/ -name examples</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

2.2.2. SSL v3.0 POODLE 취약점

분류	솔루션 취약점	중요도	상
항목명	SSL v3.0 POODLE 취약점		
항목 설명	TLS(Transport Layer Security)은 인터넷에서 정보를 암호화해서 송수신하는 프로토콜로 넷스케이프 커뮤니케이션스사가 개발한 SSL(Secure Socket Layer)에서 표준화된 기술로, 국제 인터넷 표준화 기구에서 표준으로 인정받은 프로토콜. 표준에 명시된 정식 명칭은 TLS 이지만 아직도 SSL이라는 용어가 많이 사용되고 있음. SSL 3.0버전은 1996년 발표된 통신규약으로 현재도 많은 웹서버와 브라우저에서 과거 개발된 시스템과 어플리케이션과의 호환성 문제로 SSL 통신이 가능하게 설정되어 있으나 최근 발표된 POODLE 취약점(CVE-2014-3566)을 비롯해 지속적으로 SSL 프로토콜의 취약점이 발견되어 시스템과 어플리케이션 등에서 SSL통신 설정을 제거하고 국제 표준인 TLS 통신만 사용 가능하도록 설정해야 함. POODLE 취약점(CVE-2014-3566)은 공격자가 클라이언트에게 서버가 TLS를 지원하지 않는다고 속여 클라이언트로 하여금 SSL v.3.0을 사용하게 강제한 후 이 과정에서 중간 공격자가 보호된 HTTP 쿠키의 암호를 풀 수 있게 되는 취약점으로 시스템과 어플리케이션에서 암호화 통신 프로토콜 설정 중 SSL통신을 제거함으로써 예방 가능함		
설정 방법	1. SSL통신 설정을 제거하고 TLS설정만 가능하도록 설정 [Unix - Standalone Mode] <pre># vi [Djboss.home.dir]/standalone/configuration/standalone.xml</pre> <pre><subsystem xmlns="urn:jboss:domain:undertow:X.X"> ... <filters> <https-listener name="httpsServer" security-realm="ApplicationRealm" socket-binding="https" enabled-protocols="TLSv1.2"/> [Unix - Domain Mode] <pre># vi [Djboss.home.dir]/domain/configuration/domain.xml</pre> <pre><subsystem xmlns="urn:jboss:domain:undertow:X.X"> ... <filters> <https-listener name="httpsServer" security-realm="ApplicationRealm" socket-binding="https" enabled-protocols="TLSv1.2"/> 2. 호환성 이슈로 인해 SSLv3 비활성화가 불가하고, OpenSSL 버전 업그레이드 또한 불가할 경우 OpenSSL 설정파일에서 CBC Cipher Suite 제거 - SSLv3에서 사용할 수 있는 아래의 Cipher Suite를 사용하지 않도록 설정하여야 함. </pre></pre>		

	IDEA-CBC-SHA, EXP-DES-CBC-SHA, DES-CBC-SHA, DES-CBC3-SHA, EXP-DH-DSS-DES-CBC-SHA, DH-DSS-DES-CBC-SHA, DH-DSS-DES-CBC3-SHA, EXP-DH-RSA-DES-CBC-SHA, DH-RSA-DES-CBC-SHA, DH-RSA-DES-CBC3-SHA, EXP-DHE-DSS-DES-CBC-SHA, DHE-DSS-CBC-SHA, DHE-DSS-DES-CBC3-SHA, EXP-DHE-RSA-DES-CBC-SHA, DHE-RSA-DES-CBC-SHA, DHE-RSA-DES-CBC3-SHA, EXP-ADH-DES-CBC-SHA, ADH-DES-CBC-SHA, ADH-DES-CBC3-SHA, EXP-RC2-CBC-MD5, IDEA-CBC-SHA, EXP-DES-CBC-SHA, DES-CBC-SHA, DES-CBC3-SHA, EXP-DHE-DSS-DES-CBC-SHA, DHE-DSS-CBC-SHA, DHE-DSS-DES-CBC3-SHA, EXP-DHE-RSA-DES-CBC-SHA, DHE-RSA-DES-CBC-SHA, DHE-RSA-DES-CBC3-SHA, ADH-DES-CBC-SHA, ADH-DES-CBC3-SHA, AES128-SHA, AES256-SHA, DH-DSS-AES128-SHA, DH-DSS-AES256-SHA, DH-RSA-AES128-SHA, DH-RSA-AES256-SHA, DHE-DSS-AES128-SHA, DHE-DSS-AES256-SHA, DHE-RSA-AES128-SHA, DHE-RSA-AES256-SHA, ADH-AES128-SHA, ADH-AES256-SHA ※ TLS1.0 프로토콜 중 하기 내용과 같이 보안상 취약한 알고리즘은 제거되어야 함 TLS_DH_anon_WITH_AES_256_CBC_SHA TLS_DH_anon_WITH_3DES_EDE_CBC_SHA TLS_DH_anon_WITH_AES_128_CBC_SHA TLS_DH_anon_WITH_RC4_128_MD5 TLS_DH_anon_WITH_DES_CBC_SHA - 조치방안 : https://www.owasp.org/index.php/Securing_tomcat#Encryption
진단 기준	양호 - 암호화 통신 프로토콜에서 TLS가 설정되어 있는 경우 취약 - 암호화 통신 프로토콜에서 TLS가 설정되어 있지 않는 경우
진단 방법	[진단예시] [동적테스트] // 진단 사이트를 이용한 진단 : poodlebleed.com 테스트할 도메인 정보를 입력 후 취약점 존재여부 확인 [정적테스트] [Unix - Standalone Mode] # cat [Djboss.home.dir]/standalone/configuration/standalone.xml grep https-listener [Unix - Domain Mode] # cat [Djboss.home.dir]/domain/configuration/domain.xml grep https-listener
비고	장기 적용(적용 시 개발자 및 운영자 협의)

2.2.3. Apache Commons-Collection 라이브러리 취약점

분류	솔루션 취약점	중요도	상
항목명	Apache Commons-Collection 라이브러리 취약점		
항목 설명	자바 관련 공통 컴포넌트 개발을 위한 Apache commons-collection 라이브러리에서 원격코드실행 취약점이 발견. 공격자가 취약한 대상 서비스에 악의적인 데이터를 삽입하여 전송할 경우 시스템 명령어 실행, 악성코드 다운로드 및 실행 등 가능		
설정 방법	1. Apache commons-collection 4.1 최신버전으로 업데이트 적용 http://commons.apache.org/proper/commons-collections/download_collections.cgi 2. 버전 업그레이드가 불가피할 경우 원격에서 명령 실행 가능성이 있는 Serialization support 제거 - 안전하지 않은 클래스 리스트 CloneTransformer, ForClosure, InstantiateFactory, InstantiateTransformer, InvokerTransformer, PrototypeCloneFactory, PrototypeSerializationFactory, WhileClosure		
진단 기준	양호 - 안전한 버전의 라이브러리를 사용하거나, 안전하지 않은 클래스를 사용하지 않을 경우 취약 - 취약한 버전의 라이브러리를 사용하고 있으며, 안전하지 않은 클래스를 사용할 경우		
진단 방법	[진단예시] - 어플리케이션 소스 루트 경로에서 library 폴더 내 commons-collections-*.*.jar 파일 확인 - grep -RI [안전하지 않은 클래스 리스트] /[소스 루트 경로]/*		
비고	장기 적용(적용 시 개발자 및 운영자 협의)		

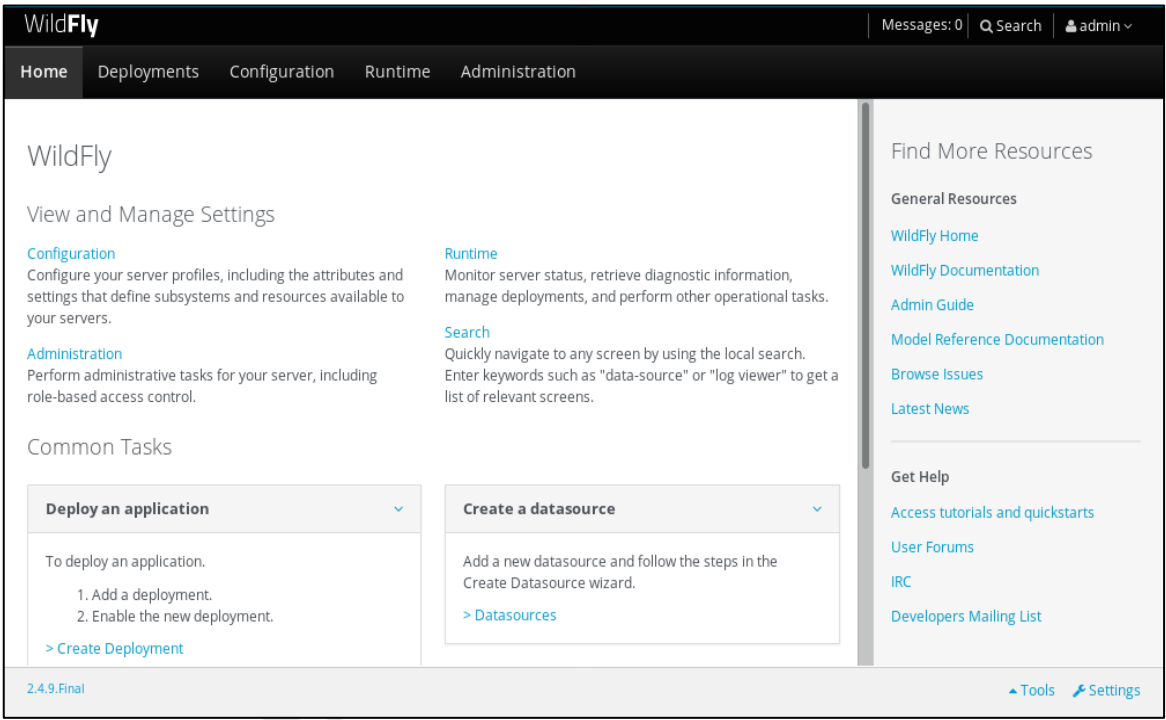
2.3. 보안 패치

2.3.1. 보안 패치 적용

분류	보안 패치	중요도	상																	
항목명	보안 패치 적용																			
항목 설명	주기적으로 보안 패치를 적용하지 않으면 exploit 공격, 제로데이 공격 등의 서버 침해가 발생할 수 있음.																			
설정 방법	1. 기간 산정해서 보안 패치 적용 (정기 PM등)																			
	<table><tr><th>WildFly 버전</th><th>권고 기준</th></tr><tr><td>WildFly 12.x.x</td><td>12.0.0 Final 이상</td></tr><tr><td>WildFly 11.x.x</td><td>11.0.0 Final 이상</td></tr><tr><td rowspan="2">WildFly 10.x.x</td><td>10.1.0 Final 이상</td></tr><tr><td>10.0.0 Final 이상</td></tr><tr><td>WildFly 9.x.x</td><td>9.0.2 Final 이상</td></tr></table>			WildFly 버전	권고 기준	WildFly 12.x.x	12.0.0 Final 이상	WildFly 11.x.x	11.0.0 Final 이상	WildFly 10.x.x	10.1.0 Final 이상	10.0.0 Final 이상	WildFly 9.x.x	9.0.2 Final 이상						
	WildFly 버전	권고 기준																		
	WildFly 12.x.x	12.0.0 Final 이상																		
	WildFly 11.x.x	11.0.0 Final 이상																		
	WildFly 10.x.x	10.1.0 Final 이상																		
		10.0.0 Final 이상																		
	WildFly 9.x.x	9.0.2 Final 이상																		
	※ 기준 버전은 위와 같으나 신규 CVSS 이슈가 발생할 경우 추가 취약 보고될 수 있음																			
	* WildFly 최신 패치 - http://wildfly.org/downloads/																			
2. SSL 관련 CVE 점수와 관계없이 영향도가 높은 CVE 코드 표																				
<table><tr><th>CVE 코드</th><th>CVSS</th><th>내 용</th></tr><tr><td>CVE-2014-3566</td><td>4.3</td><td>SSLv3.0 프로토콜 관련 POODLE 취약점</td></tr><tr><td>CVE-2015-0204</td><td>4.3</td><td>OpenSSL FREAK(Factoring attack on RSA-EXPORT Keys) 취약점 관련</td></tr><tr><td>CVE-2016-0800</td><td>5.9</td><td>DROWN(Decrypting RSA using Obsolete Weakened eNcryption)으로 명명된, SSLv2를 이용한 TLS에 대한 프로토콜 간 공격 취약점 관련</td></tr><tr><td>CVE-2017-3733</td><td>5.0</td><td>OpenSSL 1.1.0 버전에서의 서비스 거부(Denial-of-Service) 공격 취약점 관련</td></tr><tr><td>CVE-2017-3737</td><td>4.3</td><td>OpenSSL 1.0.2 버전부터 1.0.2n하위버전까지의 서비스 거부(Denial-of-Service) 공격 취약점 관련</td></tr></table>			CVE 코드	CVSS	내 용	CVE-2014-3566	4.3	SSLv3.0 프로토콜 관련 POODLE 취약점	CVE-2015-0204	4.3	OpenSSL FREAK(Factoring attack on RSA-EXPORT Keys) 취약점 관련	CVE-2016-0800	5.9	DROWN(Decrypting RSA using Obsolete Weakened eNcryption)으로 명명된, SSLv2를 이용한 TLS에 대한 프로토콜 간 공격 취약점 관련	CVE-2017-3733	5.0	OpenSSL 1.1.0 버전에서의 서비스 거부(Denial-of-Service) 공격 취약점 관련	CVE-2017-3737	4.3	OpenSSL 1.0.2 버전부터 1.0.2n하위버전까지의 서비스 거부(Denial-of-Service) 공격 취약점 관련
CVE 코드	CVSS	내 용																		
CVE-2014-3566	4.3	SSLv3.0 프로토콜 관련 POODLE 취약점																		
CVE-2015-0204	4.3	OpenSSL FREAK(Factoring attack on RSA-EXPORT Keys) 취약점 관련																		
CVE-2016-0800	5.9	DROWN(Decrypting RSA using Obsolete Weakened eNcryption)으로 명명된, SSLv2를 이용한 TLS에 대한 프로토콜 간 공격 취약점 관련																		
CVE-2017-3733	5.0	OpenSSL 1.1.0 버전에서의 서비스 거부(Denial-of-Service) 공격 취약점 관련																		
CVE-2017-3737	4.3	OpenSSL 1.0.2 버전부터 1.0.2n하위버전까지의 서비스 거부(Denial-of-Service) 공격 취약점 관련																		
※ 서비스 및 운영상의 영향도를 확인하시어 설정하여 적용해야 합니다.																				
진단 기준	양호 - WildFly 권고 기준 이상의 버전을 사용할 경우 취약 - WildFly 권고 기준 미만의 버전을 사용할 경우																			
진단 방법	[진단예시] #cd / [Djboss.home.dir/bin/ ./standalone.sh --version																			
비고	중기 적용(적용 시 개발자 및 운영자 협의)																			

2.4. 접근 제어

2.4.1. 관리자 콘솔 접근통제

분류	접근 제어	중요도	상
항목명	관리자 콘솔 접근통제		
항목 설명	관리자 콘솔의 경우, 외부로부터 침해되는 경우 Web에 관련된 모든 권한을 누출할 수 있으므로 관리에 주의해야 함.		
설정 방법	1. 필요한 경우에 한하여 관리자 콘솔을 사용하고 불필요한 경우 프로세스 종료  2. 관리자 콘솔 Default 포트 9990(http)은 공격자가 유추 할 수 있으므로, 유추 할 수 없는 포트로 포트 번호를 지정하여 사용 (권장 포트 범위 : 1024 ~ 65534) [Unix - Standalone Mode] <pre># vi [Djboss.home.dir]/standalone/configuration/standalone.xml <socket-binding-group name="standard-sockets" default-interface="public" port-offset="\${jboss.socket.binding.port-offset:0}"> <socket-binding name="management-http" interface="management" port="\${jboss.management.http.port:9999}" /> </socket-binding-group></pre> [Unix - Domain Mode] <pre># vi [Djboss.home.dir]/domain/configuration/domain.xml <socket-binding-group name="standard-sockets" default-interface="public" port-offset="\${jboss.socket.binding.port-offset:0}"> <socket-binding name="management-http" interface="management" port="\${jboss.management.http.port:9999}" /> </socket-binding-group></pre> 3. 관리자 콘솔 사용 시 IP 접근제어		

	[Unix - Standalone Mode] <pre># vi [Djboss.home.dir]/standalone/configuration/standalone.xml</pre> <pre><subsystem xmlns="urn:jboss:domain:weld:X.X"> ... <interface name="management"> <inet-address value="허용할 IP" /> </interface> </subsystem></pre> ※ Standalone Mode의 경우 관리자 콘솔은 기본적으로 사용하도록 설정되어 있으며 127.0.0.1만 접근 가능하도록 설정되어 있음 [Unix - Domain Mode] <pre># vi [Djboss.home.dir]/domain/configuration/domain.xml</pre> <pre><subsystem xmlns="urn:jboss:domain: weld:X.X"> ... <interface name="management"> <inet-address value="허용할 IP" /> </interface> </subsystem></pre> ※ Domain Mode의 경우 관리자 콘솔은 기본적으로 사용하지 않도록 설정되어 있음
진단 기준	양호 - 관리자 콘솔을 사용하지 않거나, 콘솔 설정이 주석 해제되어 있는 경우 취약 - 관리자 콘솔을 사용하고 있으며, 콘솔 설정이 주석되어 있는 경우
진단 방법	[진단예시] [Unix - Standalone Mode] <pre># cat [Djboss.home.dir]/standalone/configuration/standalone.xml grep management</pre> [Unix - Domain Mode] <pre># cat [Djboss.home.dir]/domain/configuration/domain.xml grep management</pre>
비고	단기 적용(적용 시 개발자 및 운영자 협의)

2.4.2. 관리자 default 계정명 변경

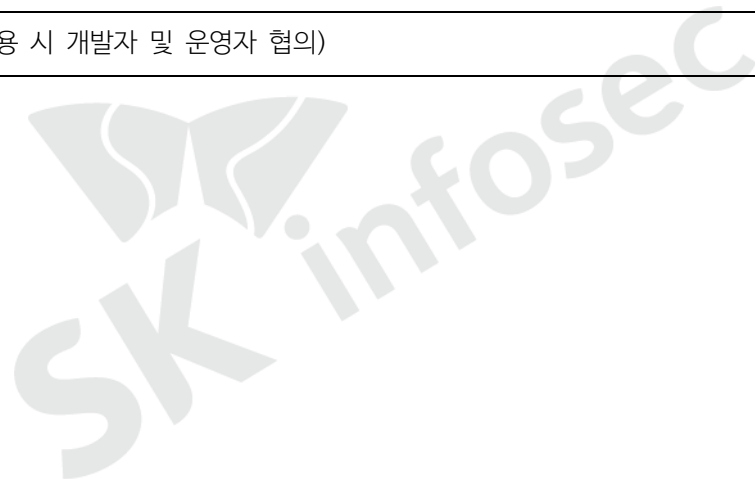
분류	접근 제어	중요도	하
항목명	관리자 default 계정명 변경		
항목 설명	Default 계정을 그대로 사용하는 경우, [brute-force] 공격의 위험에 노출되는 취약점이 존재 하므로 타 유추 불가능한 계정 명으로 변경 권고함.		
설정 방법	1. 기본 관리자 계정 사용중지 [Unix - Standalone Mode] <pre># vi [Djboss.home.dir]/standalone/configuration/mgmt-user.properties</pre>		

	<pre>#admin=2a0923285184943425d15741adw723a7a</pre> ※ 기본 관리자 계정인 admin 주석(#) 처리 <pre>[Unix - Domain Mode] # vi [Djboss.home.dir]/domain/configuration/mgmt-user.properties #admin=2a0923285184943425d15741adw723a7a</pre> ※ 기본 관리자 계정인 admin 주석(#) 처리 2. 관리자 계정 생성 <pre>[Unix - Standalone, Domain Mode] # cd [Djboss.home.dir]/bin/ # ./add-user.sh - 관리자(Management User) 생성 시 'a' 입력 - ID 입력 - PW 입력</pre> ※ /bin/add-user.sh 를 통해 관리자 생성 시 mgmt-users.properties에 저장 됨
진단 기준	양호 - Default 계정으로 설정되어 있지 않는 경우 취약 - Default 계정으로 설정되어 있는 경우
진단 방법	[진단예시] <pre>[Unix - Standalone Mode] # cat [Djboss.home.dir]/standalone/configuration/mgmt-user.properties grep "admin="</pre> <pre>[Unix - Domain Mode] # cat [Djboss.home.dir]/domain/configuration/ mgmt-user.properties grep "admin="</pre>
비고	단기 적용(적용 시 개발자 및 운영자 협의)

2.4.3. 관리자 패스워드 암호정책

분류	접근 제어	중요도	상										
항목명	관리자 패스워드 암호정책												
항목 설명	관리자 계정의 패스워드를 취약하게 설정하여 사용하는 경우, 비인가 사용자가 패스워드 유추로 공격을 시도하여, 관리자 권한을 획득 할 수 있음.												
설정 방법	■ 기 준 가. 패스워드 설정 시 아래 요구조건이 반영될 수 있도록 설계 <table><tr><th>구분</th><th>공통 기준</th></tr><tr><td>패스워드 길이/복잡성</td><td>9자리 이상/3종류 이상</td></tr><tr><td>변경 주기</td><td>3개월/1개월(중요시스템)</td></tr><tr><td>재사용 금지</td><td>직전 1개 패스워드</td></tr><tr><td>잠금</td><td>10회 실패 시</td></tr></table> 1) 패스워드는 아래의 4가지 문자 종류 중 3종류 이상을 조합하여 최소 9자리 이상 의 길이로 구성 (1) 영문 대문자 (26개) (2) 영문 소문자 (26개) (3) 숫자 (10개) (4) 특수문자 (32개) 2) 패스워드는 비인가자에 의한 추측이 어렵도록 다음의 사항을 반영하여 설계 (1) Null 패스워드 사용 금지 (2) 문자 또는 숫자만으로 구성 금지 (3) 사용자 ID와 동일한 패스워드 금지 (4) 연속적인 문자/숫자(예. 1111, 1234, abcd) 사용 금지 (5) 주기성 패스워드 재사용 금지 (6) 전화번호, 생일같이 추측하기 쉬운 개인정보를 패스워드로 사용 금지 3) 초기 패스워드는 사용자에게 부여 후 최초 접속 시 즉시 변경되도록 설계 4) 10회 이상의 연속적인 패스워드 입력 실패 시 해당 사용자 ID는 사용권한이 일시 중지되도록 설계 5) 패스워드는 최대 3개월, 업무 중요도에 따라 1개월 주기로 변경 6) 패스워드 변경 시 직전 1개와 동일한 패스워드 사용금지 7) 패스워드는 마스킹 처리등을 통해 화면상에서 읽을 수 없는 형태로 표시 8) 패키지 등을 도입한 경우 패키지의 기본 기능에서 위의 항목이 제공되지 않는 경우 보안운영자에게 해당 내용 문의 및 보안성 검토 후 요구조건 반영 나. 패스워드 관리기능 구현 패스워드 분실로 인한 신규 패스워드 발급 절차 및 클리핑 레벨 적용 - 패스워드 분실시 패스워드를 초기화한 후 안전한 전송수단을 통해 패스워드 제공 - 패스워드 초기화 처리후 로그인시 패스워드 변경을 유도 - 사용자 식별/인증 실패시 계정잠금 및 접속차단 기능 적용 - 계정잠금 해제를 위한 절차 적용 다. 패스워드 변경기능 구현			구분	공통 기준	패스워드 길이/복잡성	9자리 이상/3종류 이상	변경 주기	3개월/1개월(중요시스템)	재사용 금지	직전 1개 패스워드	잠금	10회 실패 시
구분	공통 기준												
패스워드 길이/복잡성	9자리 이상/3종류 이상												
변경 주기	3개월/1개월(중요시스템)												
재사용 금지	직전 1개 패스워드												
잠금	10회 실패 시												

	관리자에 의한 변경과 사용자가 스스로 패스워드를 변경할 수 있는 기능 제공 사용자로부터 패스워드 변경 요청이 있을 경우, 사용자 신원 확인이 완료된 후 패스워드 변경될 수 있도록 설정
진단 기준	양호 - 보안기준에 충족하는 패스워드로 설정되어 있는 경우 취약 - 보안기준에 충족하지 않는 패스워드로 설정되어 있는 경우
진단 방법	[진단예시] 패스워드 암호정책에 맞게 설정하였는지 담당자 확인 필요
비고	단기 적용(적용 시 개발자 및 운영자 협의)



2.4.4. 패스워드 파일 권한 설정

분류	접근 제어	중요도	중
항목명	패스워드 파일 권한 설정		
항목 설명	패스워드 파일내에는 계정과 패스워드가 평문으로 저장되어 있어 일반계정이 읽을 경우, 관리 콘솔, JMX 콘솔 계정 정보가 쉽게 노출됨.		
설정 방법	1. 패스워드 파일, Role 파일 권한 변경 [Unix – Standalone Mode] <pre># chmod 600 [Djboss.home.dir]/standalone/configuration/application-roles.properties # chmod 600 [Djboss.home.dir]/standalone/configuration/application-users.properties # chmod 600 [Djboss.home.dir]/standalone/configuration/mgmt-groups.properties # chmod 600 [Djboss.home.dir]/standalone/configuration/mgmt-users.properties</pre> <pre># chown wildfly:wildfly [Djboss.home.dir]/standalone/configuration/application-roles.properties # chown wildfly:wildfly [Djboss.home.dir]/standalone/configuration/application-users.properties # chown wildfly:wildfly [Djboss.home.dir]/standalone/configuration/mgmt-groups.properties # chown wildfly:wildfly [Djboss.home.dir]/standalone/configuration/mgmt-users.properties</pre> [Unix – Domain Mode] <pre># chmod 600 [Djboss.home.dir]/domain/configuration/application-roles.properties # chmod 600 [Djboss.home.dir]/domain/configuration/application-users.properties # chmod 600 [Djboss.home.dir]/domain/configuration/mgmt-groups.properties # chmod 600 [Djboss.home.dir]/domain/configuration/mgmt-users.properties</pre> <pre># chown wildfly:wildfly [Djboss.home.dir]/domain/configuration/application-roles.properties # chown wildfly:wildfly [Djboss.home.dir]/domain/configuration/application-users.properties # chown wildfly:wildfly [Djboss.home.dir]/domain/configuration/mgmt-groups.properties # chown wildfly:wildfly [Djboss.home.dir]/domain/configuration/mgmt-users.properties</pre> [Window – Standalone Mode] - 아래 파일의 권한 설정 [Djboss.home.dir]/standalone/configuration/application-roles.properties [Djboss.home.dir]/standalone/configuration/application-users.properties [Djboss.home.dir]/standalone/configuration/mgmt-groups.properties [Djboss.home.dir]/standalone/configuration/mgmt-users.properties 1) Administrator 또는 전용 WAS Server 계정으로 소유자 변경 2) 전용 WAS Server 계정 그룹(Administrator) - 모든 권한 3) Users 그룹, Everyone 그룹 - 그룹 제거		

	[Window - Domain Mode] - 아래 파일의 권한 설정 [Djboss.home.dir]/domain/configuration/application-roles.properties [Djboss.home.dir]/domain/configuration/application-users.properties [Djboss.home.dir]/domain/configuration/mgmt-groups.properties [Djboss.home.dir]/domain/configuration/mgmt-users.properties 1) Administrator 또는 전용 WAS Server 계정으로 소유자 변경 2) 전용 WAS Server 계정 그룹(Administrator) - 모든 권한 3) Users 그룹, Everyone 그룹 - 그룹 제거
진단 기준	양호 - 전용 WAS Server 계정 소유이고, 700(-rwx-----) 또는 600(-rw-----) 권한인 경우 취약 - 전용 WAS Server 계정 소유가 아니거나, 700(-rwx-----) 또는 600(-rw-----) 권한이 아닌 경우
진단 방법	[진단예시] [Unix - Standalone Mode] <pre># ls -al [Djboss.home.dir]/standalone/configuration/application-roles.properties # ls -al [Djboss.home.dir]/standalone/configuration/application-users.properties # ls -al [Djboss.home.dir]/standalone/configuration/mgmt-groups.properties # ls -al [Djboss.home.dir]/standalone/configuration/mgmt-users.properties</pre> [Unix - Domain Mode] <pre># ls -al [Djboss.home.dir]/domain/configuration/application-roles.properties # ls -al [Djboss.home.dir]/domain/configuration/application-users.properties # ls -al [Djboss.home.dir]/domain/configuration/mgmt-groups.properties # ls -al [Djboss.home.dir]/domain/configuration/mgmt-users.properties</pre>
비고	단기 적용(적용 시 개발자 및 운영자 협의)

3. NginX

3.1. 설정

3.1.1. 데몬 관리

분류	설정	중요도	상
항목명	데몬 관리		
항목 설명	웹 서버 데몬이 [root] 권한으로 구동될 경우, Web Application 취약점이나 Buffer Overflow 취약점 등을 이용하여 공격자가 [root] 권한을 도용할 수 있음.		
설정 방법	1. 데몬 User / Group 설정 <pre># vi /[nginx 설치 디렉토리]/conf/nginx.conf</pre> User daemon daemon; * “User [user] [group];” 설정 시 group 생략 가능(user 값과 동일하게 적용)		
진단 기준	양호 - 전용 Web Server 계정으로 설정 및 데몬이 구동 중인 경우 취약 - root 권한의 계정으로 설정 및 데몬이 구동 중인 경우		
진단 방법	[진단예시] <pre># cat /[nginx 설치 디렉토리]/conf/nginx.conf grep "User"</pre> <pre># ps -ef grep nginx</pre>		
비고	장기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

3.1.2. 관리서버 디렉토리 권한 설정

분류	설정	중요도	중
항목명	관리서버 디렉토리 권한 설정		
항목 설명	일반 사용자가 관리서버 디렉토리에 접근할 경우 홈페이지 변조, 설정 변경 등으로 인한 장애가 발생할 수 있으므로 일반 사용자의 접근 권한을 제한해야 함.		
설정 방법	1. 관리서버 디렉토리 권한 설정 [Unix] <pre># chown daemon:daemon /[nginx 설치 디렉토리]/ # chmod 750 /[nginx 설치 디렉토리]</pre> [Window] 1) Administrator 또는 전용 Web Server 계정으로 소유자 변경 2) 전용 Web Server 계정 그룹(Administrator) - 모든 권한 3) Users 그룹 - 쓰기 권한 제거 4) Everyone 그룹 - 그룹 제거 ※ 파일 업로드 폴더 또는 게시판(DBMS 미연동시)만 쓰기 권한 부여		
진단 기준	양호 - 전용 Web Server 계정 소유이고, 750(drwxr-x---) 권한인 경우 취약 - 전용 Web Server 계정 소유가 아니거나, 750(drwxr-x---) 권한 초과인 경우		
진단 방법	[진단예시] <pre># ls -ald /[nginx 설치 디렉토리]/</pre>		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

3.1.3. 설정파일 권한 설정

분류	설정	중요도	상
항목명	설정파일 권한 설정		
항목 설명	일반 사용자가 웹 서버의 설정 파일을 삭제, 변경할 수 있을 경우 시스템이 오작동하여 사용 불능 상태에 빠질 우려가 있음.		
설정 방법	1. 설정 파일 권한 설정 [Unix] <pre># chown daemon:daemon / [Nginx 설치 디렉토리] /conf/*.conf</pre> <pre># chmod 600 / [Nginx 설치 디렉토리] /conf/*.conf // 또는 700</pre> [Windows] 1) Administrator 또는 전용 WAS 계정으로 소유자 변경 2) 전용 WAS 계정 그룹(Administrator) - 모든 권한 3) Users 그룹 - 그룹 제거 4) Everyone 그룹 - 그룹 제거		
진단 기준	양호 - 전용 Web Server 계정 소유이고, 600(-rw-----) 또는 700(-rwx-----) 권한인 경우 취약 - 전용 Web Server 계정 소유가 아니거나, 600(-rw-----) 또는 700(-rwx-----) 권한 초과인 경우		
진단 방법	[진단예시] <pre># find / [Nginx 설치 디렉토리] /conf/*.conf - name "*.conf" - exec ls -al {} \;</pre>		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

3.1.4. 디렉토리 검색 기능 제거

분류	설정	중요도	중
항목명	디렉토리 검색 기능 제거		
항목 설명	디렉토리 검색 기능이 활성화 되어 있으면 해당 디렉토리에 존재하는 모든 파일이 리스팅 되어, Web 서버 구조 노출 및 주요 설정 파일의 내용이 유출될 가능성이 있음.		
설정 방법	1. /[nginx 설치 디렉토리]/conf/nginx.conf 파일에서 autoindex off; 옵션으로 설정하거나 autoindex 옵션 삭제 (default : autoindex off;) <pre>#vi /[nginx 설치 디렉토리]/conf/nginx.conf</pre> ... 중략... <pre>autoindex off;</pre> ... 중략... ※ 서비스 및 운영상의 영향도를 확인하시어 설정하여 적용해야 합니다.		
진단 기준	양호 - autoindex 옵션이 없거나 off로 설정되어 있을 경우 취약 - autoindex 옵션이 on으로 설정되어 있을 경우		
진단 방법	[진단예시] <pre># cat /[nginx 설치 디렉토리]/conf/nginx.conf grep "autoindex"</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

3.1.5. 로그 디렉토리/파일 권한 설정

분류	설정	중요도	중
항목명	로그 디렉토리/파일 권한 설정		
항목 설명	로그 파일에는 공격자에게 유용한 정보가 들어있을 수 있으므로 일반 사용자에게 의한 정보 유출이 불가능하도록 권한 설정해야 함.		
설정 방법	1. default 로그 디렉토리 및 파일 권한 설정 <pre># chown daemon: daemon / [nginx 설치 디렉토리] /logs/ # chmod 750 / [nginx 설치 디렉토리] /logs/</pre> 2. nginx.conf 파일에서 로그 디렉토리 위치 확인 <pre># cat / [nginx 설치 디렉토리] /conf/nginx.conf grep "error_log\ access_log" error_log logs/error.log; access_log logs/access.log combined;</pre> 3. 로그 디렉토리 및 파일 권한 설정 <pre># chown -R daemon: daemon / [nginx 설치 디렉토리] /logs/ # chmod 750 / [nginx 설치 디렉토리] /logs/ # chmod 640 / [nginx 설치 디렉토리] /logs/ [로그파일]</pre>		
진단 기준	양호 - 전용 Web Server 계정 소유이고, 디렉토리는 750(drwxr-x---), 파일은 640(-rw-r-----) 권한인 경우 취약 - 전용 Web Server 계정 소유가 아니거나, 디렉토리는 750(drwxr-x---), 파일은 640(-rw-r-----) 초과인 경우		
진단 방법	[진단예시] <pre># ls -ald / [nginx 설치 디렉토리] /logs/ # ls -al / [nginx 설치 디렉토리] /logs/ [로그파일]</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

3.1.6. 로그 포맷 설정

분류	설정	중요도	상
항목명	로그 포맷 설정		
항목 설명	웹 서버의 로그 포맷을 Combined로 설정하지 않으면, 공격 여부 파악, 공격자 사용 툴 파악, 공격자 위치 파악이 불가능하므로 반드시 Combined 포맷 또는 그에 준하는 포맷 스트링으로 설정해야 함.		
설정 방법	1. /[nginx 설치 디렉토리]/conf/nginx.conf 파일에서 access_log의 로그 포맷을 Combined로 설정 <pre>#vi /[nginx 설치 디렉토리]/conf/nginx.conf</pre> ... 중략... <pre># log_format combined '\$remote_addr - \$remote_user [\$time_local] ' '"\$request" \$status \$body_bytes_sent ' '"\$http_referer" "\$http_user_agent";</pre> <pre>access_log logs/access.log combined;</pre> * 로그 포맷 스트링 \$remote_addr : 원격의 IP 주소 \$remote_user [\$time_local] : 시간 \$request : 요청값 \$status : 상태코드 \$body_bytes_sent : 전송 된 body 용량 \$http_referer : 현재 nginx 서버에 접속하기 전에 머물렀던 URL을 기록 \$http_user_agent : 접속자의 웹 브라우저(OS 포함) 종류를 기록		
진단 기준	양호 - 로그포맷 설정 값이 combined 이거나 그에 준하는 포맷 스트링으로 설정되어 있는 경우 취약 - 로그포맷 설정 값이 combined가 아니거나 그에 준하지 않는 포맷 스트링으로 설정되어 있는 경우		
진단 방법	[진단예시] <pre># cat /[nginx 설치 디렉토리]/conf/nginx.conf grep "access_log"</pre>		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

3.1.7. 로그 저장 주기

분류	설정	중요도	상
항목명	로그 저장 주기		
항목 설명	'정보통신망이용촉진및정보보호등에관한법률', '개인정보보호법', '회사사규' 등에 따라 접속 기록은 정해진 최소 보유 기간 동안 보관 해야 하며, 담당자는 접속 기록을 정기적으로 백업·확인·감독하여야 함.		
설정 방법	1. 접속 기록 보유 기간은 사업 환경에 따라 조정 할 수 있으나, '정보통신망 이용 촉진 및 정보보호 등에 관한 법률', '개인정보보호법', '회사사규' 등에 따라 최소 아래 기간 이상은 보관 해야 함.		
	7) 사용자접속기록		
	사용자 로그인/로그아웃/정보변경 등	6개월이상	
	8) 개인정보취급자의개인정보처리시스템접속기록		
	정보주체 식별정보/개인정보취급자 식별정보/ 접속일시/접속지 정보/ 부여된 권한 유형에 따른 수행업무 등	2년이상	
설정 방법	9) 개인정보취급자권한변경기록		
	개인정보취급자 권한 생성/변경/삭제 등	5년이상	
	2. 담당자는 접속 기록을 월 1회 이상 정기적으로 확인·감독하여, 접속과 관련 된 오류 및 부정행위가 발생하거나 예상 되는 경우 즉각적인 보고 조치가 되도록 해야 함		
	3. 접속 기록이 위·변조 되지 않도록 별도의 물리적인 저장 장치에 보관하여야 하며 정기적인 백업을 수행 해야 함		
	※ 수정이 가능해야 할 경우, 위·변조 여부를 확인 할 수 있도록 별도 보호조치 - 위·변조 여부를 확인할 수 있는 정보(HMAC값 또는 전자서명값 등)는 별도의 HDD 또는 관리대장에 보관 하는 방법으로 관리		
진단 기준	양호 - 로그 저장 주기 기준에 맞게 운영 중일 경우 취약 - 로그 저장 주기 기준에 맞게 운영 중이 아닐 경우		
진단 방법	[진단예시] 서버 운영 또는 담당자에게 문의		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

3.1.8. 헤더 정보 노출 방지

분류	설정	중요도	하
항목명	헤더 정보 노출 방지		
항목 설명	공격자가 대상 시스템의 정보를 획득하기 위해 고의적으로 웹 서버 헤더 정보를 유출을 유도할 수 있음.		
설정 방법	1. server_tokens 설정(default : server_tokens on;) <pre>#vi /[nginx 설치 디렉토리]/conf/nginx.conf</pre> ... 중략... <pre>server_tokens off;</pre> ... 중략... * server_tokens 은 http, server, location 절에 모두 설정 가능 ※ 서비스 및 운영상의 영향도를 확인하시어 설정하여 적용해야 합니다.		
진단 기준	양호 - server_tokens 설정 값이 off인 경우 취약 - server_tokens 설정 값이 off가 아닌 경우		
진단 방법	[진단예시] <pre># cat /[nginx 설치 디렉토리]/conf/nginx.conf grep "server_tokens"</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

3.1.9. HTTP Method 제한

분류	설정	중요도	하
항목명	HTTP Method 제한		
항목 설명	GET, POST, HEAD, OPTIONS 이외의 다른 HTTP Method 를 지원하는 경우, 악의적인 공격자가 임의의 파일을 삭제하거나 업로드하여 서버의 정상 운영에 지장을 줄 수 있음.		
설정 방법	1. nginx 웹 서버는 Default 값으로 GET, POST, HEAD, OPTIONS Method만을 허용 2. Dav 모듈 사용 제한 - nginx.conf 파일에서 Dav 모듈을 사용하면 PUT, DELETE와 같은 불필요한 Method 의 사용이 가능해지므로 사용을 제한 (Default : dav_methods off) * 취약한 예제 (활성화 예) <pre># cat /[nginx 설치 디렉토리]/conf/nginx.conf location / { root /data/www; dav_methods PUT DELETE MKCOL COPY MOVE; }</pre> ※ 서비스 및 운영상의 영향도를 확인하시어 설정하여 적용해야 합니다.		
진단 기준	양호 - Dav 모듈 사용 시 GET, POST, HEAD, OPTIONS만 사용할 경우 취약 - Dav 모듈 사용 시 GET, POST, HEAD, OPTIONS 이외의 Method를 사용할 경우		
진단 방법	[진단예시] <pre># cat /[nginx 설치 디렉토리]/conf/nginx.conf grep "dav_methods"</pre>		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

3.1.10. 에러 메시지 관리

분류	설정	중요도	중
항목명	에러 메시지 관리		
항목 설명	공격자가 대상 시스템의 정보를 획득하기 위해 고의적으로 다양한 에러를 유발하여 돌아오는 에러 메시지를 통해 웹 프로그램의 구조 및 환경 설정을 추정할 수 있음.		
설정 방법	1. 에러핸들링 설정 값 적용(필수항목 : 400, 401, 403, 404, 500) <pre># vi /[nginx 설치 디렉토리]/conf/nginx.conf ... 중략... location / { root html; index index.html index.htm; } error_page 400 401 403 404 500 /error.html; ... 중략...</pre> 2. 에러페이지 생성 <pre># vi /[nginx 설치 디렉토리]/error.html ... 중략... <p>시스템 오류</p> <div class="p" style="margin-bottom:2em"> 이용에 불편을 드려 죄송합니다.
 잠시 후 다시 이용해 주시길 바랍니다. </div> ... 중략...</pre>		
진단 기준	양호 - 필수 에러코드가 핸들링 설정이 있고, 에러페이지가 있을 경우 취약 - 필수 에러코드가 핸들링 설정이 없거나, 에러페이지가 없을 경우		
진단 방법	[진단예시] <pre># cat /[nginx 설치 디렉토리]/conf/nginx.conf grep "error_page"</pre>		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

3.2. 솔루션 취약점

3.2.1. 기본 문서명 사용 제한

분류	솔루션 취약점	중요도	하
항목명	기본 문서명 사용 제한		
항목 설명	웹 사이트의 기본문서명을 디폴트 값으로 사용하는 경우 악의적인 공격자가 서버의 종류, 버전에 대한 정보를 쉽게 유추하여, 취약점에 대해 공격할 가능성이 높아짐.		
설정 방법	1. nginx.conf 파일에서 기본 문서명 설정 값 변경 <pre># vi /[nginx 설치 디렉토리]/conf/nginx.conf location / { root html; index example_main.html; }</pre>		
진단 기준	양호 - 기본 문서명이 "index.html", "index.htm"이 아닐 경우 취약 - 기본 문서명이 "index.html", "index.htm"일 경우		
진단 방법	[진단예시] <pre># cat /[nginx 설치 디렉토리]/conf/nginx.conf grep "index"</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

3.2.2. SSL v3.0 POODLE 취약점

분류	솔루션 취약점	중요도	상
항목명	SSL v3.0 POODLE 취약점		
항목 설명	TLS(Transport Layer Security)은 인터넷에서 정보를 암호화해서 송수신하는 프로토콜로 넷스케이프 커뮤니케이션스사가 개발한 SSL(Secure Socket Layer)에서 표준화된 기술로, 국제 인터넷 표준화 기구에서 표준으로 인정받은 프로토콜. 표준에 명시된 정식 명칭은 TLS 이지만 아직도 SSL이라는 용어가 많이 사용되고 있음. SSL 3.0버전은 1996년 발표된 통신규약으로 현재도 많은 웹서버와 브라우저에서 과거 개발된 시스템과 어플리케이션과의 호환성 문제로 SSL 통신이 가능하게 설정되어 있으나 최근 발표된 POODLE 취약점(CVE-2014-3566)을 비롯해 지속적으로 SSL 프로토콜의 취약점이 발견되어 시스템과 어플리케이션 등에서 SSL통신 설정을 제거하고 국제 표준인 TLS 통신만 사용 가능하도록 설정해야 함. POODLE 취약점(CVE-2014-3566)은 공격자가 클라이언트에게 서버가 TLS를 지원하지 않는다고 속여 클라이언트로 하여금 SSL v.3.0을 사용하게 강제한 후 이 과정에서 중간 공격자가 보호된 HTTP 쿠키의 암호를 풀 수 있게 되는 취약점으로 시스템과 어플리케이션에서 암호화 통신 프로토콜 설정 중 SSL통신을 제거함으로써 예방 가능 함.		
설정 방법	3. nginx.conf 파일에서 SSL 설정 파일 주석 해제 # vi /[nginx 설치 디렉토리]/conf/nginx.conf <pre> ... 중략... server { listen 443 ssl; server_name www.example.com; ssl_certificate www.example.com.crt; ssl_certificate_key www.example.com.key; ssl_protocols TLSv1 TLSv1.1 TLSv1.2; ssl_ciphers HIGH:!aNULL:!MD5; } ... 중략...</pre> 4. 호환성 이슈로 인해 SSLv3 비활성화가 불가하고, OpenSSL 버전 업그레이드 또한 불가할 경우 OpenSSL 설정파일에서 CBC Cipher Suite 제거 - SSLv3에서 사용할 수 있는 아래의 Cipher Suite를 사용하지 않도록 설정하여야 함. IDEA-CBC-SHA, EXP-DES-CBC-SHA, DES-CBC-SHA, DES-CBC3-SHA, EXP-DH-DSS-DES-CBC-SHA, DH-DSS-DES-CBC-SHA, DH-DSS-DES-CBC3-SHA, EXP-DH-RSA-DES-CBC-SHA, DH-RSA-DES-CBC-SHA, DH-RSA-DES-CBC3-SHA, EXP-DHE-DSS-DES-CBC-SHA, DHE-DSS-CBC-SHA, DHE-DSS-DES-CBC3-SHA, EXP-DHE-RSA-DES-CBC-SHA, DHE-RSA-DES-CBC-SHA, DHE-RSA-DES-CBC3-SHA, EXP-ADH-DES-CBC-SHA, ADH-DES-CBC-SHA, ADH-DES-CBC3-SHA, EXP-RC2-CBC-MD5, IDEA-CBC-SHA,		

	EXP-DES-CBC-SHA, DES-CBC-SHA, DES-CBC3-SHA, EXP-DHE-DSS-DES-CBC-SHA, DHE-DSS-CBC-SHA, DHE-DSS-DES-CBC3-SHA, EXP-DHE-RSA-DES-CBC-SHA, DHE-RSA-DES-CBC-SHA, DHE-RSA-DES-CBC3-SHA, ADH-DES-CBC-SHA, ADH-DES-CBC3-SHA, AES128-SHA, AES256-SHA, DH-DSS-AES128-SHA, DH-DSS-AES256-SHA, DH-RSA-AES128-SHA, DH-RSA-AES256-SHA, DHE-DSS-AES128-SHA, DHE-DSS-AES256-SHA, DHE-RSA-AES128-SHA, DHE-RSA-AES256-SHA, ADH-AES128-SHA, ADH-AES256-SHA ※ TLS1.0 프로토콜 중 하기 내용과 같이 보안상 취약한 알고리즘은 제거되어야 함 TLS_DH_anon_WITH_AES_256_CBC_SHA TLS_DH_anon_WITH_3DES_EDE_CBC_SHA TLS_DH_anon_WITH_AES_128_CBC_SHA TLS_DH_anon_WITH_RC4_128_MD5 TLS_DH_anon_WITH_DES_CBC_SHA - 조치방안 : https://www.owasp.org/index.php/Securing_tomcat#Encryption
진단 기준	양호 - 암호화 통신 프로토콜에서 TLS가 설정되어 있는 경우 취약 - 암호화 통신 프로토콜에서 TLS가 설정되어 있지 않는 경우
진단 방법	[진단예시] [동적테스트] // 진단 사이트를 이용한 진단 : poodlebleed.com 테스트할 도메인 정보를 입력 후 취약점 존재여부 확인 [정적테스트] # cat /[nginx 설치 디렉토리]/conf/nginx.conf grep "ssl_protocols"
비고	장기 적용(적용 시 개발자 및 운영자 협의)
기반시설 기준항목	-

3.3. 보안 패치

3.3.1. 보안 패치 적용

분류	보안 패치	중요도	상																																							
항목명	보안 패치 적용																																									
항목 설명	주기적으로 보안 패치를 적용하지 않으면 exploit 공격, 제로데이 공격 등의 서버 침해가 발생할 수 있음.																																									
설정 방법	1. 기간 산정 후 보안 패치 적용 (정기 PM 등)																																									
	<table><tr><th>Nginx 버전</th><th>권고 기준</th><th>Release 일자</th></tr><tr><td>Nginx 1.11</td><td>1.11.10 이상</td><td>2017-02-14</td></tr><tr><td>Nginx 1.10</td><td>1.10.3 이상</td><td>2017-01-31</td></tr><tr><td>Nginx 1.9</td><td>1.9.10 이상</td><td>2016-01-09</td></tr><tr><td>Nginx 1.8</td><td>1.8.1 이상</td><td>2016-01-26</td></tr><tr><td>Nginx 1.6</td><td>1.6.3 이상</td><td>2015-04-07</td></tr><tr><td>Nginx 1.4</td><td>1.4.7 이상</td><td>2014-05-18</td></tr><tr><td>Nginx 1.2</td><td>1.2.9 이상</td><td>2013-05-13</td></tr><tr><td>Nginx 1.0</td><td>1.0.15 이상</td><td>2012-04-12</td></tr><tr><td>Nginx 0.8</td><td>0.8.55 이상</td><td>2011-07-19</td></tr><tr><td>Nginx 0.7</td><td>0.7.69 이상</td><td>2011-07-19</td></tr><tr><td>Nginx 0.6</td><td>0.6.39 이상</td><td>2009-10-14</td></tr><tr><td>Nginx 0.5</td><td>0.5.38 이상</td><td>2009-10-14</td></tr></table>			Nginx 버전	권고 기준	Release 일자	Nginx 1.11	1.11.10 이상	2017-02-14	Nginx 1.10	1.10.3 이상	2017-01-31	Nginx 1.9	1.9.10 이상	2016-01-09	Nginx 1.8	1.8.1 이상	2016-01-26	Nginx 1.6	1.6.3 이상	2015-04-07	Nginx 1.4	1.4.7 이상	2014-05-18	Nginx 1.2	1.2.9 이상	2013-05-13	Nginx 1.0	1.0.15 이상	2012-04-12	Nginx 0.8	0.8.55 이상	2011-07-19	Nginx 0.7	0.7.69 이상	2011-07-19	Nginx 0.6	0.6.39 이상	2009-10-14	Nginx 0.5	0.5.38 이상	2009-10-14
	Nginx 버전	권고 기준	Release 일자																																							
	Nginx 1.11	1.11.10 이상	2017-02-14																																							
	Nginx 1.10	1.10.3 이상	2017-01-31																																							
	Nginx 1.9	1.9.10 이상	2016-01-09																																							
	Nginx 1.8	1.8.1 이상	2016-01-26																																							
	Nginx 1.6	1.6.3 이상	2015-04-07																																							
	Nginx 1.4	1.4.7 이상	2014-05-18																																							
	Nginx 1.2	1.2.9 이상	2013-05-13																																							
Nginx 1.0	1.0.15 이상	2012-04-12																																								
Nginx 0.8	0.8.55 이상	2011-07-19																																								
Nginx 0.7	0.7.69 이상	2011-07-19																																								
Nginx 0.6	0.6.39 이상	2009-10-14																																								
Nginx 0.5	0.5.38 이상	2009-10-14																																								
* Nginx 최신 패치(http://nginx.org/en/download.html)																																										
※ 기준 버전은 위와 같으나 신규 CVSS 이슈가 발생할 경우 추가 취약 보고될 수 있음																																										
2. 아래 사이트를 참고하여, 원격 exploit 취약점, 제로 데이 취약점 등 크리티컬한 취약점 발견 시 즉시 패치 적용																																										
* Nginx Security Update(http://nginx.org/en/security_advisories.html)																																										
3. SSL 관련 CVE 점수와 관계없이 영향도가 높은 CVE 코드표																																										
<table><tr><th>CVE 코드</th><th>CVSS</th><th>내용</th></tr><tr><td>CVE-2014-3566</td><td>4.3</td><td>SSLv3.0 프로토콜 관련 POODLE 취약점</td></tr><tr><td>CVE-2015-0204</td><td>4.3</td><td>OpenSSL FREAK(Factoring attack on RSA-EXPORT Keys) 취약점 관련</td></tr><tr><td>CVE-2016-0800</td><td>5.9</td><td>DROWN(Decrypting RSA using Obsolete Weakenede Ncryption)으로 명명된 SSLv2를 이용한 TLS에 대한 프로토콜간 공격 취약점 관련</td></tr><tr><td>CVE-2017-3733</td><td>5.0</td><td>OpenSSL 1.1.0 버전에서의 서비스 거부(Denial-of-Service) 공격 취약점 관련</td></tr><tr><td>CVE-2017-3737</td><td>4.3</td><td>OpenSSL 1.0.2 버전부터 1.0.2n 하위버전까지의 서비스 거부(Denial-of-Service) 공격 취약점 관련</td></tr></table>			CVE 코드	CVSS	내용	CVE-2014-3566	4.3	SSLv3.0 프로토콜 관련 POODLE 취약점	CVE-2015-0204	4.3	OpenSSL FREAK(Factoring attack on RSA-EXPORT Keys) 취약점 관련	CVE-2016-0800	5.9	DROWN(Decrypting RSA using Obsolete Weakenede Ncryption)으로 명명된 SSLv2를 이용한 TLS에 대한 프로토콜간 공격 취약점 관련	CVE-2017-3733	5.0	OpenSSL 1.1.0 버전에서의 서비스 거부(Denial-of-Service) 공격 취약점 관련	CVE-2017-3737	4.3	OpenSSL 1.0.2 버전부터 1.0.2n 하위버전까지의 서비스 거부(Denial-of-Service) 공격 취약점 관련																						
CVE 코드	CVSS	내용																																								
CVE-2014-3566	4.3	SSLv3.0 프로토콜 관련 POODLE 취약점																																								
CVE-2015-0204	4.3	OpenSSL FREAK(Factoring attack on RSA-EXPORT Keys) 취약점 관련																																								
CVE-2016-0800	5.9	DROWN(Decrypting RSA using Obsolete Weakenede Ncryption)으로 명명된 SSLv2를 이용한 TLS에 대한 프로토콜간 공격 취약점 관련																																								
CVE-2017-3733	5.0	OpenSSL 1.1.0 버전에서의 서비스 거부(Denial-of-Service) 공격 취약점 관련																																								
CVE-2017-3737	4.3	OpenSSL 1.0.2 버전부터 1.0.2n 하위버전까지의 서비스 거부(Denial-of-Service) 공격 취약점 관련																																								
※ 서비스 및 운영상의 영향도를 확인하시어 설정하여 적용해야 합니다.																																										

진단 기준	양호 - Nginx 권고기준 이상 버전을 적용 중인 경우 취약 - Nginx 권고기준 이상 버전을 적용 중이지 않은 경우
진단 방법	[진단예시] # nginx -v
비고	중기 적용(적용 시 개발자 및 운영자 협의)
기반시설 기준항목	-



4. Elastic Search

4.1. X-Pack 설정

4.1.1. X-Pack 활성화 설정

분류	X-Pack 설정	중요도	상
항목명	X-Pack 활성화 설정		
항목 설명	모니터링이나 기본 보안설정을 하기 위하여 X-Pack을 활성화 하여 관리가 필요함		
설정 방법	1. 사전에 X-Pack, Kibana, logstash 설치 2. Elasticsearch 및 Kibana를 시작 3. X-Pack 활성화 elasticsearch.yml 파일에서 xpack.security.enabled		
진단 방법	[진단기준] - X-Pack이 활성화 된 경우 양호 - X-Pack이 비활성화 된 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

4.1.2. 기본 패스워드 변경

분류	X-Pack 설정	중요도	상
항목명	X-Pack 기본패스워드 변경		
항목 설명	X-Pack에 제공된 기본 패스워드를 사용하는 경우 비 인가자의 접근으로 인한 시스템 접근이 이루어 질 수 있으므로 변경이 필요함		
설정 방법	1. X-Pack 패스워드 설정 bin / x-pack / setup-passwords interactive 명령어 입력 후 패스워드 입력 2. Elastic 패스워드 변경 curl -XPUT -u elastic 'localhost:9200/_xpack/security/user/elastic/_password' -H "Content-Type: application/json" -d '{ "password": "elasticpassword" }' 3. kibana 패스워드 변경 curl -XPUT -u elastic 'localhost:9200/_xpack/security/user/kibana/_password' -H "Content-Type: application/json" -d '{ "password": "kibanapassword" }' 4. Logstash 패스워드 변경 curl -XPUT -u elastic 'localhost:9200/_xpack/security/user/logstash_system/_password' -H "Content-Type: application/json" -d '{ "password": "logstashpassword" }' 5. 기본 패스워드 비활성화 elasticsearch.yml에 아래와 같이 설정 xpack.security.authc.accept_default_password false		
진단 방법	[진단기준] - 기본패스워드가 비활성화 되어 있고 패스워드를 변경한 경우 양호 - 기본패스워드 사용이 활성화 되어 있거나 패스워드 변경을 실시하지 않은 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

4.2. 사용자 접근 통제

4.2.1. 접근 정책 설정

분류	접근 통제	중요도	상
항목명	접근 정책 설정		
항목 설명	Elastic Search에 접근하여 수행할 수 있는 접근 정책을 설정하여야 함. 접근정책을 정의하지 않은 경우 사용자가 불필요한 정보에 접근할 우려가 있음		
설정 방법	<pre>curl -XPOST -u elastic 'localhost:9200/_xpack/security/role/권한설정 이름' -H "Content-Type: application/json" -d '{ "indices": [{ "names": ["events*"], // 접근 영역 "privileges": ["all"] // 접근 권한 }, { "names": [".kibana*"], // 접근 영역 "privileges": ["manage", "read", "index"] //접근권한 설정 }] }'</pre>		
진단 방법	[진단기준] - 권한별 정책을 설정하여 운영하고 있는 경우 양호 - 권한별 정책 설정이 없이 모든 권한 정책만 있는 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

4.2.2. 사용자 계정 설정

분류	접근 통제	중요도	상
항목명	사용자 계정 설정		
항목 설명	시스템에 접근 할 수 있는 사용자를 지정하고 각 사용자별 정책을 차등으로 관리하여야 함.		
설정 방법	<pre>curl -XPOST -u elastic 'localhost:9200/_xpack/security/user/사용자ID' -H "Content-Type: application/json" -d '{ "password": "사용자패스워드", "full_name": "사용자 이름", "email": "사용자이메일주소", "roles": ["권한설정 이름"] }'</pre>		
진단 방법	[진단기준] - 사용자별 계정 설정을 하고 있는 경우 양호 - 사용자별 계정 설정을 하지 않고 단일 계정을 사용하고 있는 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

4.2.3. 익명연결 비활성화

분류	접근 통제	중요도	상
항목명	익명연결 비 활성화		
항목 설명	익명연결은 비인가자가 시스템에 접근 할 수 있는 우려가 있어서 반드시 비활성화 처리가 필요함		
설정 방법	1. elasticsearch.yml 파일에 아래의 설정을 하는 경우는 익명연결을 허용함 <pre> xpack.security.authc: anonymous: username: anonymous_user roles: role1, role2 authz_exception: true </pre> 2. 위의 설정 제거		
진단 방법	[진단기준] - 익명연결 설정이 없는 경우 양호 - 익명연결 설정이 활성화 되어 있는 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

4.3. 네트워크 접근 통제

4.3.1. SSL/TLS 설정

분류	네트워크 접근 통제	중요도	권고
항목명	SSL/TLS 설정		
항목 설명	Elastic Search의 각 Node간 통신에 암호화 통신을 적용하여야 함. 암호화 통신을 적용하지 않는 경우 Sniffing 등의 위험에 노출 될 수 있음		
설정 방법	각 Node의 elasticsearch.yml 파일의 설정을 통하여 SSL 통신 설정을 할 수 있음 1. SSL통신 활성화 xpack.security.http.ssl.enabled 2. SSL프로토콜 설정 (TLS 1.2 이상 권고) xpack.security.http.ssl.supported_protocols 프로토콜 3. Key 설정 xpack.security.transport.ssl.enabled: true xpack.security.transport.ssl.verification_mode: certificate xpack.security.transport.ssl.keystore.path: 인증서 저장소 위치 xpack.security.transport.ssl.truststore.path: 인증서 저장소 위치 4. Elastic 재 시작		
진단 방법	권고 사항		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

4.3.2. 네트워크 필터링 설정 설정

분류	네트워크 접근 통제	중요도	권고
항목명	네트워크 필터링 설정		
항목 설명	비인가 네트워크로부터의 접근을 차단이 필요한 경우 설정하여야 함		
설정 방법	elasticsearch.yml 파일에 설정 1. IP 필터링 설정 접근허용 : xpack.security.transport.filter.allow: "192.168.0.1" 접근차단 : xpack.security.transport.filter.deny: "192.168.0.0/24" 2. HTTP필터링 설정 xpack.security.transport.filter.allow: localhost xpack.security.transport.filter.deny: '*.google.com' xpack.security.http.filter.allow: 172.16.0.0/16 xpack.security.http.filter.deny: _all		
진단 방법	권고 사항		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

4.4. 로그 설정

4.4.1. 감사 로그 설정

분류	로그 설정	중요도	상
항목명	감사 로그 설정		
항목 설명	로그를 설정하지 않으면, 공격 여부 파악, 공격자 사용 툴 파악, 공격자 위치 파악이 불가능하므로 반드시 로그를 설정해야 함.		
설정 방법	1. 기본적으로 비활성화 2. elasticsearch.yml에 아래와 같이 설정 xpack.security.audit.outputs : logfile		
진단 방법	[진단기준] - 감사로그 설정이 활성화 되어 있는 경우 양호 - 감사로그 설정이 활성화 되어 있지 않은 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

4.4.2. 사용자 감사 설정

분류	로그 설정	중요도	중
항목명	사용자 감사 설정		
항목 설명	사용자의 행위에 대하여 감사를 실시하여 이상징후 발생 시 데이터로 활용하여야 함		
설정 방법	elasticsearch.yml 파일에 아래의 설정 추가 xpack.security.audit.enabled : true		
진단 방법	[진단기준] - 감사로그 설정이 활성화 되어 있는 경우 양호 - 감사로그 설정이 활성화 되어 있지 않은 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

5. Docker

5.1. 설정

5.1.1. 데몬관리

분류	설정	중요도	상
항목명	데몬 관리		
항목 설명	Unix 시스템의 경우, Docker 데몬이 root 권한으로 운영될 경우 Application의 취약점이나 Buffer Overflow시 공격자에게 root권한을 유출할 수 있으므로 Docker 데몬이 root 권한으로 운영되지 않도록 관리해야 함.		
설정 방법	1. Docker 데몬 구동을 위한 계정을 별도로 지정하여 관리 docker run -u <username> -it <container_name> /bin/bash		
진단 방법	[진단기준] - 구동중인 Docker 데몬의 계정이 전용 계정 인 경우 양호 - 구동중인 Docker 데몬의 계정이 root 인 경우 취약		
비고	장기 적용(적용 시 개발자 및 운영자 협의)		

5.1.2. 리소스 제한 설정

분류	설정	중요도	상
항목명	리소스 제한 설정		
항목 설명	시스템 자원 고갈을 통한 DoS (Denial of Service) 공격을 방지하기 위해 특정 명령 줄 인수를 사용하여 많은 리소스 제한을 적용		
설정 방법	CPU 제한 docker run -it --rm --cpuset=0,1 -c 2 ... 메모리 사용량 제한 docker run -it --rm -m 128m ... 스토리지 사용량 제한 docker -d --storage-opt dm.basesize=5G		
진단 방법	[진단기준] - 리소스 제한 설정을 사용하는 경우 양호 - 리소스 제한 설정을 하지 않는 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

5.1.3. SUID / SGID 제거 설정

분류	설정	중요도	하
항목명	SUID / SGID 제거 설정		
항목 설명	SUID 및 GUID 바이너리는 프로세스의 파일 소유자 나 그룹의 컨텍스트에서 실행될 때 임의 코드 실행 (예 : 버퍼 오버플로우)에 이르는 공격에 취약한 경우 위험 할 수 있음		
설정 방법	불필요한 사항 제거 방법 docker run -it --rm --cap-drop SETUID --cap-drop SETGID ...		
진단 방법	[진단기준] - 불필요한 SUID, SGID가 설정되어 있지 않을 경우 양호 - 불필요한 SUID, SGID가 설정되어 있을 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

5.2. Docker 신뢰 설정

5.2.1. 신뢰할 수 있는 인증서 설정

분류	Docker 신뢰 설정	중요도	중
항목명	신뢰할 수 있는 인증서 설정		
항목 설명	신뢰할 수 있는 인증서 설정을 통하여 안전한 통신 보장		
설정 방법	1. 사전 인증서 발행 작업 수행 2. docker 데몬 설정 인증서 설정 \$ dockerd --tlsverify --tlscacert=ca.pem --tlscert=server-cert.pem --tlskey=server-key.pem ₩ -H=0.0.0.0:2376 3. Client 설정 \$ docker --tlsverify --tlscacert=ca.pem --tlscert=cert.pem --tlskey=key.pem ₩ -H=\$HOST:2376 version (반드시 2376 포트에서 실행)		
진단 방법	[진단기준] - 인증서 설정이 되어 있는 경우 양호 - 인증서 설정이 없는 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

5.2.1. 신뢰할 수 있는 콘텐츠 설정

분류	설정	중요도	중
항목명	신뢰할 수 있는 콘텐츠 설정		
항목 설명	신뢰할 수 있는 콘텐츠 설정이 되어있지 않은 경우 비인가자의 악성 컨테이너가 배포될 우려가 있으므로 신뢰할 수 있는 컨테이너 설정을 하여야 함		
설정 방법	bash shell에서 아래와 같이 입력 export DOCKER_CONTENT_TRUST = 1		
진단 방법	[진단기준] - 신뢰할 수 있는 콘텐츠 설정이 되어 있는 경우 양호 - 신뢰할 수 있는 콘텐츠 설정이 되어 있지 않은 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

5.3. Container 설정

5.3.1. 네트워크 설정

분류	Container 설정	중요도	참고
항목명	네트워크 설정		
항목 설명	Container의 네트워크 모드는 사용자의 현황에 따라 설정이 필요함		
설정 방법	Bridge Mode 기본 네트워크 방식으로 각 컨테이너마다 고유한 network namespace를 생성하여 docker0 bridge에 인터페이스가 binding 되는 구조 Host 방식 컨테이너가 독립적인 네트워크 영역을 갖지 않고 host와 네트워크를 함께 공유하는 구조 Docker run --net=host 컨테이너명 컨테이너종류 Container 방식 기존에 존재하는 다른 컨테이너와 네트워크 환경을 공유하며 IP와 MAC주소도 동일 Docker run --name 신규컨테이너명 --net=container:참조컨테이너_ID -d 컨테이너종류 None 네트워크 인터페이스가 없음 Docker run --name 컨테이너명 --net=none -d 컨테이너종류		
비고			

5.3.2. Root외 권한으로 컨테이너 실행

분류	Container 설정	중요도	참고
항목명	Root 외 권한으로 컨테이너 실행		
항목 설명	컨테이너를 root외 사용자로 지정하여 각 컨테이너별 권한설정을 하여야 합니다.		
설정 방법	설정방법 일반적인 사용자 설정 Docker run -ti --rm -u User_No 작업디렉토리 이미지명 Nobody 활용 Docker run -ti --rm -u 65534 작업디렉토리 이미지명		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

5.3.3. Shell 권한 설정

분류	Container 설정	중요도	중
항목명	Shell 권한 설정		
항목 설명	컨테이너 내 셸을 구동하는데 있어 호스트의 셸권한을 부여하지 않아도 컨테이너 내 셸을 사용할 수 있습니다. 컨테이너 관리용 계정을 별도로 생성하여 관리가 필요합니다.		
설정 방법	컨테이너 전용 계정 기본 셸 제거 /etc/passwd 파일내 docker 계정에 bin/false 부여 계정명:x:1001:1001://home/계정명:bin/false docker 셸 실행방법 docker exec -user 계정명 -i -t 컨테이너주소 /bin/bash		
진단 방법	[진단기준] - 컨테이너 계정에 shell이 부여되어 있지 않은 경우 양호 - 컨테이너 계정에 shell이 부여되어 있는 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

5.3.4. SSH 설정

분류	Container 설정	중요도	상
항목명	SSH 설정		
항목 설명	컨테이너에 SSH 접속을 허용하는 경우 SSH버전취약점과 SSH를 관리하기 위한 프로세스매니저가 필요하며 비인가자가 컨테이너를 제거할 위험이 존재합니다. 또한 SSH접속을 하기 위하여는 root 권한을 사용하여야 합니다.		
설정 방법	1. 컨테이너 내 패키지에서 sshd 제거 컨테이너 접속 후 <code>apt-get remove openssh-server</code> 2. SSH 접속 비허용하기 /etc/ssh/sshd_config 파일에서 <code>PermitRootLogin No</code> Root 접속설정을 비허용한 경우 SSH접속이 불가능합니다.		
진단 방법	[진단기준] - SSH데몬이 설치되어 있지 않거나 root 로그인을 비활성화 한 경우 양호 - SSH데몬이 설치되어 있고 root 로그인을 허용하는 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

5.4. Log 설정

5.4.1. Docker Log 설정

분류	Log 설정	중요도	상
항목명	Docker Log 설정		
항목 설명	로그를 설정하지 않으면, 공격 여부 파악, 공격자 사용 툴 파악, 공격자 위치 파악이 불가능하므로 반드시 로그를 설정해야 함.		
설정 방법	로깅 설정 docker run --log-driver="옵션" --name 컨테이너명 -d <docker_image>		
	※단 syslog 로 설정하면 docker logs 커맨드는 disable되고 모든 로그는 Syslog로 redirect된다.		
	Driver 명	설명	
	none	컨테이너에 사용할 수있는 docker logs 가 없으며 docker logs 가 출력을 반환하지 않습니다.	
	json-file	로그는 JSON 으로 포맷됩니다. Docker 의 기본 로깅 드라이버입니다.	
	syslog	syslog 기능에 로깅 메시지를 씁니다. syslog 디먼이 호스트 시스템에서 실행 중이어야합니다.	
	journald	journald 로그 메시지를 기록합니다. journald 디먼은 호스트 시스템에서 실행 중이어야합니다.	
	gelf	Graylog 또는 Logstash 와 같은 Graylog Extended Log Format (GELF) 끝점에 로그 메시지를 씁니다.	
	fluentd	로그 메시지를 fluentd 기록합니다 (정방향 입력). fluentd 디먼은 호스트 시스템에서 실행 중이어야합니다.	
	awslogs	로그 메시지를 Amazon CloudWatch 로그에 기록합니다.	
	splunk	HTTP Event Collector 를 사용하여 로그 메시지를 splunk 기록합니다.	
	etwlogs	로그 메시지를 Windows 용 이벤트 추적 (ETW) 이벤트로 씁니다. Windows 플랫폼에서만 사용할 수 있습니다.	
	gcplogs	로그 메시지를 Google Cloud Platform (GCP) 로깅에 기록합니다.	
	logentries	로그 메시지를 Rapid7 Logentries 에 기록합니다.	
진단 방법	[진단기준] - 로그 설정이 되어 있는 경우 양호 - log-driver설정이 None인 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

5.5. 추가 기능 설정

5.5.1. AppArmor 연결

분류	추가 기능 설정	중요도	하
항목명	AppArmor 연결		
항목	AppArmor를 이용하여 컨테이너의 보안옵션을 편리하게 관리할 수 있음		

설명	
설정 방법	1. AppArmor에서 프로파일 로드 \$ apparmor_parser -r -W /path/to/사용자 정의 2. 사용자 정의 정책 실행 \$ docker run --rm -it --security-opt apparmor = 사용자 정의 AppArmor 정책 설정 --security-opt="label:user:USER" : 컨테이너의 레이블 사용자를 설정 --security-opt="label:role:ROLE" : 컨테이너의 레이블 역할 설정 --security-opt="label:type:TYPE" : 컨테이너의 레이블 유형을 설정 --security-opt="label:level:LEVEL" : 컨테이너의 레이블 레벨을 설정 --security-opt="apparmor:PROFILE" : 컨테이너에 적용 할 apparmor 프로필을 설정
비고	중기 적용(적용 시 개발자 및 운영자 협의)



6. Cassandra

6.1. 암호화 통신 설정

6.1.1. 노드 간 암호화

분류	설정	중요도	상
항목명	노드 간 암호화		
항목 설명	노드 간 암호화는 SSL (Secure Sockets Layer)을 사용하는 비밀 통신을 포함하여 클러스터의 노드간에 전송되는 데이터를 보호		
설정 방법	cassandra.yaml - server encryption option 노드간 암호화 : internode encryption Default: none -> rack,dc or all		
진단 방법	[진단기준] - 암호화 설정이 되어 있는 경우 양호 - 암호화 설정이 되어 있지 않을 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

6.1.2. 클라이언트 대 노드 암호화

분류	설정	중요도	상
항목명	클라이언트 대 노드 암호화		
항목 설명	클라이언트 대 노드 암호화는 SSL (Secure Sockets Layer)을 사용하여 클라이언트 시스템에서 데이터베이스 클러스터로 이동하는 데이터를 보호 및 보안 채널을 설정		
설정 방법	cassandra.yaml - client encryption options Default : : enabled, optional enabled -> true optional -> false		
진단 방법	[진단기준] - 암호화 설정이 되어 있는 경우 양호 - 암호화 설정이 되어 있지 않을 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

6.1.3. SSL 통신 JMX 설정

분류	설정	중요도	권고
항목명	SSL 통신 JMX 설정		
항목 설명	SSL과 함께 nodetool을 사용하려면 JMX 설정이 필요		

설정 방법	cassandra-env.sh (or cassandra-env.ps1 on Windows) 속성 값들 수정
	com.sun.management.jmxremote.ssl SSL을 사용시 true 변경
	com.sun.management.jmxremote.ssl.need.client.auth 클라이언트 인증서의 유효성을 검사하려면 true로 설정
	com.sun.management.jmxremote.registry.ssl 클라이언트가 JMX connector stub을 얻는 RMI 레지스트리 용 SSL 소켓을 활성화
	javax.net.ssl.keyStore 서버의 비공개 키 및 공개 증명서를 포함한 키 스토어의 로컬 파일 시스템상의 패스를 설정
	javax.net.ssl.keyStorePassword 키 스토어 파일의 패스워드를 설정
	javax.net.ssl.trustStore 클라이언트 인증서의 유효성 검사가 필요한 경우 이 속성을 사용하여 신뢰할 수 있는 클라이언트의 공용 인증서를 포함하는 트러스트 저장소의 경로를 지정
	javax.net.ssl.trustStorePassword truststore 파일의 암호를 설정
진단 방법	[진단기준] - 암호화 설정이 되어 있는 경우 양호 - 암호화 설정이 되어 있지 않을 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

6.2. 인증 및 권한 설정

6.2.1. 인증 설정

분류	설정	중요도	상
항목명	인증 설정		
항목 설명	내부 인증을 사용하는 개체 사용 권한 관리와 마찬가지로 내부 인증은 cassandra 제어 로그인 계정 및 암호를 기반으로 구성함		
설정 방법	1. cassandra.yaml - authenticator Default : AllowALLAuthenticator -> PasswordAuthenticator 2. Cassandra 노드 다시 시작 3. cqlsh -u cassandra -p cassandra (cqlsh 기본 슈퍼 유저 아이디와 패스워드로 시작 4. 키 공간을 사용할 수 있게 하려면 System_auth 키공간의 데이터 센터당 노드 3-5(권장)으로 늘림		

	<pre>cqlsh> ALTER KEYSPACE "system_auth" WITH REPLICATION = {'class' : 'NetworkTopologyStrategy', 'dc1' : 3, 'dc2' : 2};</pre> 5. 변경사항 확인 \$ nodetool repair system_auth 6. 카산드라 재 시작 7. \$ cqlsh -u cassandra -p cassandra 8. 기본 계정 말고 다른 슈퍼유저로 변환 cqlsh> CREATE ROLE <new_super_user> WITH PASSWORD = '<some_secure_password>' AND SUPERUSER = true AND LOGIN = true; 9. 새 슈퍼유저 계정으로 로그인 \$ cqlsh -u <new_super_user> -p <some_secure_password> 10. 예측 어려운 긴 패스워드로 변경 cqlsh> ALTER ROLE cassandra WITH PASSWORD='SomeNonsenseThatNoOneWillThinkOf' AND SUPERUSER=false;
진단 방법	[진단기준] - 인증이 설정되어 있을 경우 양호 - 인증이 설정되어 있지 않을 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

6.2.2. 내부 권한 구성(caching)

분류	설정	중요도	상
항목명	내부 권한 구성(caching)		
항목 설명	CassandraAuthorizer를 사용하기 위해 cassandra.yaml에서 권한 부여 옵션을 변경하는 것으로 구성		
설정 방법	1.cassandra.yaml - authorizer Default : AllowAllAuthorizer -> CassandraAuthorizer 2. permissions_validity_in_ms: 2000 (권한 유효 기간 설정 단위 밀리초) 3. permissions_update_interval_in_ms: 2000(역할 캐시의 새로 고침 간격)		
진단 방법	[진단기준] - 내부 권한 구성 설정이 되어 있을 경우 양호 - 내부 권한 구성 설정이 되어 있지 않을 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

6.2.3. JMX 인증 및 권한 부여

분류	설정	중요도	중
항목명	JMX 인증 및 권한 부여		
항목 설명	원격 JMX 연결을 활성화하려면 cassandra-env.sh의 LOCAL_JMX 설정을 변경하고 인증 및 SSL을 활성화 JMX 인증을 활성화 한 후 nodetool 및 DataStax OpsCenter와 같은 JMX를 사용하는 도구가 인증을 사용하도록 구성되었는지 확인		
설정 방법	XX:+DisableExplicitGC" 여기 부분에 다음 행을 추가 JVM_OPTS="\$JVM_OPTS -Dcom.sun.management.jmxremote.authenticate=true" JVM_OPTS="\$JVM_OPTS Dcom.sun.management.jmxremote.password.file=/etc/cassandra/jmxremote.password" JVM_OPTS="\$JVM_OPTS Dcom.sun.management.jmxremote.access.file=/etc/cassandra/jmxremote.access" 2. 패스워드 파일 작성 후 자격 환경을 지정 (/etc/cassandra/jmxremote.password) <pre>cassandra cassandra <new_superuser> <new_superuser_password> <some_other_user> <some_other_user_password> controlRole someOtherHardToRememberPassword</pre> 3. 권한 부여 <pre>\$ chown cassandra : cassandra /etc/cassandra/jmxremote.password \$ chmod 400 /etc/cassandra/jmxremote.password</pre> 4. 액세스 파일을 작성 (/etc/cassandra/jmxremote.access <pre>cassandra readwrite <new_superuser> readwrite <some_other_user> readonly controlRole readwrite ₩ create javax.management.monitor.,javax.management.timer. ₩ unregister</pre> 5. 액세스 파일 사용권한 설정 <pre>\$ chown cassandra : cassandra /etc/cassandra/jmxremote.access \$ chmod 400 /etc/cassandra/jmxremote.access</pre> 6. cassandra 재 시작해서 변경 사항 적용		
진단 방법	[진단기준] - 인증 및 권한 부여가 설정되어 있을 경우 양호 - 인증 및 권한 부여가 설정되어 있지 않을 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

7. Hadoop

7.1. 설정

7.1.1. Hadoop 보안 사용

분류	설정	중요도	상
항목명	Hadoop 보안 사용		
항목 설명	Hadoop에서 사용하는 보안설정을 사용할 수 있는 항목이며 커버로스 인증 또는 기본인증(Simple)을 구분하며 서비스 수준의 권한 부여를 활성화		
설정 방법	core-site.xml 파일에 추가 <pre><property> <name>hadoop.security.authentication</name> <value>kerberos</value> <!-- A value of "simple" would disable security. --> </property> <property> <name>hadoop.security.authorization</name> <value>true</value> </property></pre>		
진단 방법	[진단기준] - 보안 설정이 되어 있는 경우 양호 - 보안 설정이 되어 있지 않을 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

7.1.2. SPNEGO/Kerberos 웹 접속 관련 보안설정

분류	설정	중요도	중
항목명	SPNEGO/Kerberos 웹 접속 관련 보안설정		
항목 설명	SPNEGO(Simple and Protected GSS-API Negotiation Mechanism)를 WebSphere® Application Server에 대한 웹 인증 서비스로 사용하여 WebSphere Application Server에서 보호된 자원에 대한 HTTP 요청을 안전하게 협상하고 인증		
설정 방법	<pre> <name>hadoop.http.authentication.type</name> <value>kerberos</value> </property> <property> <name>hadoop.http.authentication.token.validity</name> <value>36000</value> </property> <property> <name>hadoop.http.authentication.signature.secret.file</name> <value>/home/hadoop/etc/hadoop/hadoop-http-auth-signature-secret</value> </property> <property> <name>hadoop.http.authentication.cookie.domain</name> <value>bloodguy.com</value> </property> <property> <name>hadoop.http.authentication.simple.anonymous.allowed</name> <value>>false</value> </property> <property> <name>hadoop.http.authentication.kerberos.principal</name> <value>HTTP/_HOST@BLOODGUY.COM</value> </property> <property> <name>hadoop.http.authentication.kerberos.keytab</name> <value>/home/hadoop/etc/hadoop/hdfs.keytab</value> </property> </pre>		
진단 방법	[진단기준] - 설정이 되어 있는 경우 양호 - 설정이 되어 있지 않을 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

7.2. HDSF설정

7.2.1. 보안 HDFS 구성

분류	설정	중요도	중
항목명	보안 HDFS 구성		
항목 설명	Hadoop에서 보안 HDFS 구성을 위한 hdfs-site.xml 설정파일 수정		
설정 방법	<pre> <property> <name>dfs.namenode.kerberos.internal.spnego.principal</name> <value>HTTP/_HOST@YOUR-REALM.COM</value> </property> <!-- Secondary NameNode security config --> <property> <name>dfs.secondary.namenode.keytab.file</name> <value>/etc/hadoop/conf/hdfs.keytab</value> <!-- path to the HDFS keytab --> </property> <property> <name>dfs.secondary.namenode.kerberos.principal</name> <value>hdfs/_HOST@YOUR-REALM.COM</value> </property> <property> <name>dfs.secondary.namenode.kerberos.internal.spnego.principal</name> <value>HTTP/_HOST@YOUR-REALM.COM</value> </property> <!-- DataNode security config --> <property> <name>dfs.datanode.data.dir.perm</name> <value>700</value> </property> <property> <name>dfs.datanode.address</name> <value>0.0.0.0:1004</value> </property> <property> <name>dfs.datanode.http.address</name> <value>0.0.0.0:1006</value> </property> <property> <name>dfs.datanode.keytab.file</name> <value>/etc/hadoop/conf/hdfs.keytab</value> <!-- path to the HDFS keytab --> </property> <property> </pre>		

	<pre> <name>dfs.datanode.kerberos.principal</name> <value>hdfs/_HOST@YOUR-REALM.COM</value> </property> <!-- Web Authentication config --> <property> <name>dfs.web.authentication.kerberos.principal</name> <value>HTTP/_HOST@YOUR_REALM</value> </property> </pre>
진단 방법	[진단기준] - 설정이 되어 있는 경우 양호 - 설정이 되어 있지 않을 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

7.2.2. HDFS에 SSL 사용

분류	설정	중요도	중
항목명	HDFS에 SSL 사용		
항목 설명	Hadoop에서 보안을 위한 SSL 설정 추가		
설정 방법	<pre> hdfs-site.xml 에 다음 속성 추가 <property> <name>dfs.block.access.token.enable</name> <value>true</value> </property> <property> <name>dfs.data.transfer.protection</name> <value>privacy</value> </property> <property> <name>dfs.http.policy</name> <value>HTTPS_ONLY</value> </property> </pre>		
진단 방법	[진단기준] - 설정이 되어 있는 경우 양호 - 설정이 되어 있지 않을 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

7.2.3. Secure Web HDFS 설정

분류	설정	중요도	중
항목명	Secure Web HDFS 설정		
항목 설명	http 프로토콜을 사용하여 안전하게 HDFS에 접근할 수 있는 설정		
설정 방법	hdfs-site.xml 에 다음 속성 추가 <pre> <property> <name>dfs.webhdfs.enabled</name> <value>true</value> </property> <property> <name>dfs.web.authentication.kerberos.principal</name> <value>HTTP/_HOST@YOUR-REALM.COM</value> </property> <property> <name>dfs.web.authentication.kerberos.keytab</name> <value>/etc/hadoop/conf/HTTP.keytab</value> <!-- path to the HTTP keytab --> </property> </pre>		
진단 방법	[진단기준] - 설정이 되어 있는 경우 양호 - 설정이 되어 있지 않을 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

7.2.4. HDFS ACL 설정

분류	설정	중요도	중
항목명	HDFS ACL 설정		
항목 설명	ACL을 활성화하여 불필요한 접근을 차단하여야 합니다.		
설정 방법	1. ACL 활성화 hdfs-site.xml 에 다음 속성 추가 <pre> <property> <name>dfs.namenode.acls.enabled</name> <value>true</value> </property> </pre> 2. Shell을 통한 ACL 설정 hdfs dfs -setfacl -<권한> <사용자 or 그룹> <파일 경로>		

	권한설정 파일 r: 파일 읽기 w: 쓰기 및 수정 디렉토리 r: 리스트 확인 w: 생성 또는 삭제 x: 디렉토리 접근 ACL 설정 예 : <code>hdfs dfs -setfacl -x user:alice /user/hdfs/file</code> 3. JAVA API를 통한 ACL 설정 ACL 수정 : <code>public void modifyAclEntries(Path path, List<AclEntry> aclSpec) throws IOException;</code> ACL 내 개체 제거 : <code>public void removeAclEntries(Path path, List<AclEntry> aclSpec) throws IOException;</code> 기본 ACL 제거 : <code>public void removeDefaultAcl(Path path) throws IOException;</code> ACL 제거 : <code>public void removeAcl(Path path) throws IOException;</code> ACL 추가 : <code>public void setAcl(Path path, List<AclEntry> aclSpec) throws IOException;</code> ACL 현황 보기 : <code>public AclStatus getAclStatus(Path path) throws IOException;</code>
진단 방법	[진단기준] - ACL을 활성화하고 ACL을 올바르게 설정한 경우 양호 - ACL을 비활성화하거나 취약한 ACL이 존재하는 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

7.2.5. HDFS 권한 설정

분류	설정	중요도	권고																																																																																																																																										
항목명	HDFS 접근 권한 설정																																																																																																																																												
항목 설명	HDFS권한 설정을 통하여 불필요한 접근을 차단하여야 합니다.																																																																																																																																												
설정 방법	• Ownership: 요청자가 경로 및 파일의 소유자인지 점검여부 • Parent: 상위 경로 접근 권한 • Ancestor: 요청경로의 마지막 기존 구성요소 • Final: 요청경로의 최종 구성 요소 • Sub-tree: 하위 경로 접근 권한																																																																																																																																												
	<table><tr><th>Operation</th><th>Ownership</th><th>Parent</th><th>Ancestor</th><th>Final</th><th>Sub-tree</th></tr><tr><td>append</td><td>NO</td><td>N/A</td><td>N/A</td><td>WRITE</td><td>N/A</td></tr><tr><td>concat</td><td>NO</td><td>WRITE (sources)</td><td>N/A</td><td>READ (sources), WRITE (destination)</td><td>N/A</td></tr><tr><td>create</td><td>NO</td><td>N/A</td><td>WRITE</td><td>WRITE</td><td>N/A</td></tr><tr><td>createSnapshot</td><td>YES</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr><tr><td>delete</td><td>NO</td><td>WRITE</td><td>N/A</td><td>N/A</td><td>READ, WRITE, EXECUTE</td></tr><tr><td>deleteSnapshot</td><td>YES</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr><tr><td>getAclStatus</td><td>NO</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr><tr><td>getBlockLocations</td><td>NO</td><td>N/A</td><td>N/A</td><td>READ</td><td>N/A</td></tr><tr><td>getContentSummary</td><td>NO</td><td>N/A</td><td>N/A</td><td>N/A</td><td>READ, EXECUTE</td></tr><tr><td>getFileInfo</td><td>NO</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr><tr><td>getFileLinkInfo</td><td>NO</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr><tr><td>getLinkTarget</td><td>NO</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr><tr><td>getListing</td><td>NO</td><td>N/A</td><td>N/A</td><td>READ, EXECUTE</td><td>N/A</td></tr><tr><td>getSnapshotDiffReport</td><td>NO</td><td>N/A</td><td>N/A</td><td>READ</td><td>READ</td></tr><tr><td>getStoragePolicy</td><td>NO</td><td>N/A</td><td>N/A</td><td>READ</td><td>N/A</td></tr><tr><td>getXAttrs</td><td>NO</td><td>N/A</td><td>N/A</td><td>READ</td><td>N/A</td></tr><tr><td>listXAttrs</td><td>NO</td><td>EXECUTE</td><td>N/A</td><td>N/A</td><td>N/A</td></tr><tr><td>mkdirs</td><td>NO</td><td>N/A</td><td>WRITE</td><td>N/A</td><td>N/A</td></tr><tr><td>modifyAclEntries</td><td>YES</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr><tr><td>removeAcl</td><td>YES</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr><tr><td>removeAdEntries</td><td>YES</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr><tr><td>removeDefaultAcl</td><td>YES</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr></table>			Operation	Ownership	Parent	Ancestor	Final	Sub-tree	append	NO	N/A	N/A	WRITE	N/A	concat	NO	WRITE (sources)	N/A	READ (sources), WRITE (destination)	N/A	create	NO	N/A	WRITE	WRITE	N/A	createSnapshot	YES	N/A	N/A	N/A	N/A	delete	NO	WRITE	N/A	N/A	READ, WRITE, EXECUTE	deleteSnapshot	YES	N/A	N/A	N/A	N/A	getAclStatus	NO	N/A	N/A	N/A	N/A	getBlockLocations	NO	N/A	N/A	READ	N/A	getContentSummary	NO	N/A	N/A	N/A	READ, EXECUTE	getFileInfo	NO	N/A	N/A	N/A	N/A	getFileLinkInfo	NO	N/A	N/A	N/A	N/A	getLinkTarget	NO	N/A	N/A	N/A	N/A	getListing	NO	N/A	N/A	READ, EXECUTE	N/A	getSnapshotDiffReport	NO	N/A	N/A	READ	READ	getStoragePolicy	NO	N/A	N/A	READ	N/A	getXAttrs	NO	N/A	N/A	READ	N/A	listXAttrs	NO	EXECUTE	N/A	N/A	N/A	mkdirs	NO	N/A	WRITE	N/A	N/A	modifyAclEntries	YES	N/A	N/A	N/A	N/A	removeAcl	YES	N/A	N/A	N/A	N/A	removeAdEntries	YES	N/A	N/A	N/A	N/A	removeDefaultAcl	YES	N/A	N/A	N/A	N/A
	Operation	Ownership	Parent	Ancestor	Final	Sub-tree																																																																																																																																							
	append	NO	N/A	N/A	WRITE	N/A																																																																																																																																							
	concat	NO	WRITE (sources)	N/A	READ (sources), WRITE (destination)	N/A																																																																																																																																							
	create	NO	N/A	WRITE	WRITE	N/A																																																																																																																																							
	createSnapshot	YES	N/A	N/A	N/A	N/A																																																																																																																																							
	delete	NO	WRITE	N/A	N/A	READ, WRITE, EXECUTE																																																																																																																																							
	deleteSnapshot	YES	N/A	N/A	N/A	N/A																																																																																																																																							
	getAclStatus	NO	N/A	N/A	N/A	N/A																																																																																																																																							
	getBlockLocations	NO	N/A	N/A	READ	N/A																																																																																																																																							
	getContentSummary	NO	N/A	N/A	N/A	READ, EXECUTE																																																																																																																																							
	getFileInfo	NO	N/A	N/A	N/A	N/A																																																																																																																																							
	getFileLinkInfo	NO	N/A	N/A	N/A	N/A																																																																																																																																							
	getLinkTarget	NO	N/A	N/A	N/A	N/A																																																																																																																																							
	getListing	NO	N/A	N/A	READ, EXECUTE	N/A																																																																																																																																							
	getSnapshotDiffReport	NO	N/A	N/A	READ	READ																																																																																																																																							
	getStoragePolicy	NO	N/A	N/A	READ	N/A																																																																																																																																							
	getXAttrs	NO	N/A	N/A	READ	N/A																																																																																																																																							
	listXAttrs	NO	EXECUTE	N/A	N/A	N/A																																																																																																																																							
	mkdirs	NO	N/A	WRITE	N/A	N/A																																																																																																																																							
	modifyAclEntries	YES	N/A	N/A	N/A	N/A																																																																																																																																							
	removeAcl	YES	N/A	N/A	N/A	N/A																																																																																																																																							
	removeAdEntries	YES	N/A	N/A	N/A	N/A																																																																																																																																							
	removeDefaultAcl	YES	N/A	N/A	N/A	N/A																																																																																																																																							

	removeXAttr	NO [2]	N/A	N/A	WRITE	N/A
	rename	NO [2]	WRITE (source)	WRITE (destination)	N/A	N/A
	renameSnapshot	YES	N/A	N/A	N/A	N/A
	setAcl	YES	N/A	N/A	N/A	N/A
	setOwner	YES [3]	N/A	N/A	N/A	N/A
	setPermission	YES	N/A	N/A	N/A	N/A
	setReplication	NO	N/A	N/A	WRITE	N/A
	setStoragePolicy	NO	N/A	N/A	WRITE	N/A
	setTimes	NO	N/A	N/A	WRITE	N/A
	setXAttr	NO [2]	N/A	N/A	WRITE	N/A
	truncate	NO	N/A	N/A	WRITE	N/A
비고	단기 적용(적용 시 개발자 및 운영자 협의)					



7.3. 기타 보안 설정

7.3.1. Secure HDFS NFS Gateway 설정

분류	설정	중요도	중
항목명	Secure HDFS NFS Gateway 설정		
항목 설명	NFS 게이트웨이는 프록시 사용자를 사용하여 NFS 마운트에 액세스하는 모든 사용자를 프록시 비보안 모드에서는 게이트웨이를 실행하는 사용자가 프록시 사용자이고 보안 모드에서는 Kerberos keytab의 사용자가 프록시 사용자		
설정 방법	hdfs-site.xml 에 다음 속성 추가 <pre> <property> <name>dfs.nfs.keytab.file</name> <value>/etc/hadoop/conf/hdfs.keytab</value> <!-- path to the HDFS or NFS gateway keytab --> </property> <property> <name>dfs.nfs.kerberos.principal</name> <value>hdfs/_HOST@YOUR-REALM.COM</value> </property> </pre>		
진단 방법	[진단기준] - 보안 설정되어 있을 경우 양호 - 보안 설정이 되어 있지 않을 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

7.3.2. Variables for Secure DataNodes 설정

분류	설정	중요도	중
항목명	Variables for Secure DataNodes 설정		
항목 설명	DataNode는 공격자가 DataNode 호스트에 대한 루트 권한을 얻을 수 없다는 가정에 기반 한 dfs.datanode.address 및 dfs.datanode.http.address로 지정된 권한 포트를 사용하여 자체 인증을 위한 설정		
설정 방법	/etc/default/hadoop-hdfs-datanode. 에 추가 <pre> export HADOOP_SECURE_DN_USER=hdfs export HADOOP_SECURE_DN_PID_DIR=/var/lib/hadoop-hdfs export HADOOP_SECURE_DN_LOG_DIR=/var/log/hadoop-hdfs export JSVC_HOME=/usr/lib/bigtop-utils/ </pre>		
진단 방법	[진단기준] - 인증 설정이 되어 있을 경우 양호 - 인증 설정이 되어 있지 않을 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

8. HBase

8.1. 계정 관리

8.1.1. 인증 활성화

분류	계정 관리	중요도	상
항목명	인증 활성화		
항목 설명	권한이 없으면 모든 계정이 disable table/drop table/major compact등을 포함하는 HBase 클러스터의 모든 작업을 수행 할 수 있음.		
설정 방법	1. hbase-site.xml 파일을 아래와 같이 수정 <pre> <property> <name>hbase.security.authorization</name> <value>true</value> </property> <property> <name>hbase.security.exec.permission.checks</name> <value>true</value> </property> <property> <name>hbase.coprocessor.master.classes</name> <value>org.apache.hadoop.hbase.security.access.AccessController</value> </property> <property> <name>hbase.coprocessor.region.classes</name> <value>org.apache.hadoop.hbase.security.token.TokenProvider,org.apache.hadoop.hbase.security.access.AccessController</value> </property> </pre>		
진단 방법	[진단기준] - 인증 설정이 적용되어 있는 경우 양호 - 인증 설정이 적용되어 있지 않는 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

8.1.2. 사용자 권한 설정

분류	계정 관리	중요도	상
항목명	사용자 권한 설정		
항목 설명	사용자에게 과도한 권한 부여 및 비인가 사용자가 HBase 권한을 획득할 경우 불필요한 정보 노출, 데이터 변조, 삭제 등의 위험이 있으므로 HBase 권한을 제한해야 함.		

설정 방법	권한부여 hbase> grant <user> <permissions> [@<namespace> [<table>[<column family>[<column qualifier>]]]] 권한 회수 hbase> revoke <user> <permissions> [@<namespace> [<table> [<column family> [<column qualifier>]]] # revokes permissions hbase> user_permission <table>
진단 방법	[진단기준] - 권한이 적절히 부여된 경우 양호 - 부적절한 권한이 부여된 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

8.2. Kerberos 설정

8.2.1. Kerberos 인증 설정

분류	Kerberos 설정	중요도	권고
항목명	Kerberos 인증 설정		
항목 설명	HBase에서 Kerberos 사용을 위한 설정		
설정 방법	1. Hbase용 Keytab 설정 Linux root 사용자로 kadmin 인터페이스를 이용하여 생성 a. HBase용 서비스 주체 설정 \$ kadmin kadmin : addprinc -randkey hbase / domain.name@YOUR-REALM.COM b. HBase 서버용 Keytab 생성 \$ kadmin kadmin : xst -k hbase.keytab hbase / domain.name c. keytab 파일을 HBase의 모든 서버호스트의 /etc/hbase/conf에 복사하고 파일 퍼미션을 400으로 설정 2. 클러스터내 모든 서버에 Kerberos 정보 등록 모든 hbase 서버의 hbase-site.xml 파일을 아래와 같이 등록 <pre> <property> <name>hbase.regionserver.kerberos.principal</name> <value>hbase/_HOST@YOUR-REALM.COM</value> </property> <property> </pre>		

	<pre> <name>hbase.regionserver.keytab.file</name> <value>/etc/hbase/conf/ hbase.keytab</value> </property> <property> <name>hbase.master.kerberos.principal</name> <value>hbase/_HOST@YOUR-REALM.COM</value> </property> <property> <name>hbase.master.keytab.file</name> <value>/etc/hbase/conf/ hbase.keytab</value> </property> </pre> 3. 서버 보안 설정 모든 hbase 서버의 hbase-site.xml 파일을 아래와 같이 등록 <pre> <property> <name>hbase.security.authentication</name> <value>kerberos</value> </property> <property> <name>hbase.security.authorization</name> <value>>true</value> </property> <property> <name>hbase.coprocessor.region.classes</name> <value>org.apache.hadoop.hbase.security.token.TokenProvider</value> </property> </pre> 4. 클라이언트 설정 모든 클라이언트의 hbase-site.xml 파일을 아래와 같이 등록 <pre> <property> <name>hbase.security.authentication</name> <value>kerberos</value> </property> </pre> 5. 클라이언트 암호화 설정(추가설정) 암호화 통신이 필요한 경우 클라이언트의 hbase-site.xml 파일을 아래와 같이 등록 <pre> <property> <name>hbase.rpc.protection</name> <value>privacy</value> </property> </pre> ※ domain.name : HBase 서버가 실행중인 호스트(소문자 표기) ※ YOUR-REALM : Kerberos 영역의 이름(대문자 표기)
비고	중기 적용(적용 시 개발자 및 운영자 협의)

9. Hive

9.1. 인증 설정

91.1. SQL 기반 인증 설정

분류	인증 설정	중요도	상
항목명	SQL 기반 인증 설정		
항목 설명	Hive 사용을 위한 SQL 인증으로 적절한 사용자와 권한 관리가 필요함		
설정 방법	1. hive-site.xml파일에 구성정보 등록 hive.server2.enable.doAs를 false 설정 2. 사용자 추가 hive.users.in.admin.role 3. admin 권한에 ADMIN Role 부여(반드시 필요한 사용자만) GRANT admin TO USER hiveadmin; 4. hive-site.xml에 아래와 같이 설정 후 재시작 <pre><property> <name>hive.security.authorization.manager</name> <value>org.apache.hadoop.hive.ql.security.authorization.plugin.sql</value> </property> <property> <name>hive.security.authorization.enabled</name> <value>true</value> </property> <property> <name>hive.security.authenticator.manager</name> <value>org.apache.hadoop.hive.ql.security.SessionStateUserAuthenticator</value> </property> <property> <name>hive.metastore.uris</name> <value>""</value> </property></pre>		
진단 방법	[진단기준] - SQL인증을 사용하면서 관리자 권한에 필요한 사용자만 존재하는 경우 양호 - SQL인증을 사용하면서 관리자 권한에 필요한 불필요한 사용자가 존재하는 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

9.1.2. 저장소 기반 인증 설정

분류	인증 설정	중요도	상
항목명	저장소 기반 인증 설정		
항목 설명	Hive인증을 저장소 기반 데이터를 통하여 인증을 수행하는 경우 올바른 데이터를 참조하여야 하며 설정이 올바르지 않은 경우 비인가자에 의한 접근이 이루어질 우려가 존재함		
설정 방법	hive-site.xml 에 아래와 같이 설정 <pre> <property> <name>hive.security.authorization.enabled</name> <value>false</value> </property> <property> <name>hive.security.authorization.manager</name> <value>org.apache.hadoop.hive.ql.security.authorization.StorageBasedAuthorizationProvider </value> </property> <property> <name>hive.server2.enable.doAs</name> <value>true</value> </property> <property> <name>hive.metastore.pre.event.listeners</name> <name>org.apache.hadoop.hive.ql.security.authorization.AuthorizationPreEventListener</na me> </property> <property> <name>hive.security.metastore.authorization.manager</name> <value>org.apache.hadoop.hive.ql.security.authorization.StorageBasedAuthorizationProvider </value> </property> </pre>		
진단 방법	[진단기준] - 각영역이 신뢰할 수 있는 저장소 기반으로 설정하고 있는 경우 양호 - 각 영역이 신뢰하지 못하는 저장소를 기반으로 설정하고 있는 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

9.2. 운영 매트릭스

9.2.1. 권한 관리 매트릭스

분류	운영 매트릭스					중요도		권고	
항목명	권한 관리 매트릭스								
설정 방법	Hive Operation	SELECT	INSERT	DELETE	Update	Ownership	Admin	URI privilege (POSIX + ownership)	
	GRANT						Y		
	REVOKE						Y		
	SHOW GRANT						Y		
	SHOW ROLE GRANT						Y		
	CREATE ROLE						Y		
	SET ROLE						Y		
	DROP ROLE						Y		
	CREATE TABLE					Y (of database)			
	DROP TABLE					Y			
	DESCRIBE TABLE	Y							
	SHOW PARTITIONS	Y							
	ALTER TABLE LOCATION					Y		Y (for new location)	
	ALTER PARTITION LOCATION					Y		Y (for new partition location)	
	ALTER TABLE ADD PARTITION		Y					Y (for partition location)	
	ALTER TABLE DROP PARTITION				Y				
	all other ALTER TABLE commands					Y			
	TRUNCATE TABLE					Y			
	CREATE VIEW	Y + G							
	ALTER VIEW PROPERTIES					Y			
	ALTER VIEW RENAME					Y			
	DROP VIEW PROPERTIES					Y			
	DROP VIEW					Y			
	ANALYZE TABLE	Y	Y						
	SHOW COLUMNS	Y							
	SHOW TABLE STATUS	Y							
	SHOW TABLE PROPERTIES	Y							
	CREATE TABLE AS SELECT	Y (of input)					Y	Y (of database)	
	UPDATE TABLE	Y							
	CREATE INDEX							Y (of table)	
	DROP INDEX							Y	

	ALTER INDEX REBUILD						Y	
	ALTER INDEX PROPERTIES						Y	
	QUERY (INSERT, SELECT queries)	Y (input)	Y (output)	Y (output)				
	LOAD		Y (output)	Y (output)				Y (input location)
	SHOW CREATE TABLE	Y + G						
	CREATE FUNCTION						Y	
	DROP FUNCTION						Y	
	CREATE MACRO						Y	
	DROP MACRO						Y	
	MSCK (metastore check)						Y	
	ALTER DATABASE						Y	
	CREATE DATABASE							Y (for custom location)
	EXPLAIN	Y						
	DROP Database				Y			
비고	장기 적용(적용 시 개발자 및 운영자 협의)							

9.2.2. 운영 매트릭스

분류	매트릭스				중요도	권고
항목명	운영 매트릭스					
설정 방법	Operation	Database READ Access	Database WRITE Access	Table READ Access	Table WRITE Access	
	LOAD				X	
	EXPORT			X		
	IMPORT				X	
	CREATE TABLE		X			
	CREATE TABLE AS SELECT		X	X (source table)		
	DROP TABLE		X			
	SELECT			X		
	ALTER TABLE				X	
	SHOW TABLES	X				
비고	장기 적용(적용 시 개발자 및 운영자 협의)					

10. Impala

10.1 설정

1.1. 데몬 관리

분류	설정	중요도	상
항목명	데몬 관리		
항목 설명	Impala 데몬(impalad)을 root 계정으로 실행할 경우, HDFS 내 존재하는 데이터 파일에 대한 읽기/쓰기가 가능함 또한, Impala의 통제 범위에 벗어난 다른 사용자 계정 로그인 및 다른 시스템 서비스에 접근이 가능함		
설정 방법	1. Impala 사용자 계정 요구사항을 준수해야 함 - Impala는 impala라고 명명된 사용자 및 그룹을 생성 및 사용하므로, impala 계정 및 그룹의 권한을 변경해서는 안 됨 - 운영체제가 impala 계정 및 그룹의 기능에 장애가 되지 않도록 보장해야 함 - 예를 들면, 만약 화이트리스트에 없는 사용자 계정을 삭제할 스크립트를 가지고 있으면, 화이트리스트의 승인된 계정 목록에 impala 관련 계정을 추가해야 함 - DROP TABLE 연산을 하는 동안 올바른 파일 삭제를 위해, Impala는 HDFS 휴지통으로 파일을 이동시킬 수 있어야 함 - 휴지통이 생성될 수 있도록, impala 전용 계정으로 쓰기 가능한 HDFS 디렉토리(/user/impala) 생성이 필요할 수도 있음 - 그렇지 않으면, 데이터 파일은 DROP TABLE 문 실행 이후 데이터가 남아있을 수 있음 - Impala는 root로서 실행해서는 안 됨 - 최고의 Impala 성능은 Direct Read를 이용하면 성취 가능하지만, root 계정은 Direct Read를 사용토록 허용해서는 안됨 - 따라서, root로서 Impala를 실행하는 것은 성능에 부정적인 영향이 끼침 - 기본적으로, 어떤 사용자도 Impala에 연결할 수 있으며, 연관된 모든 DB와 테이블도 접근할 수 있음 - Impala 서버에 연결할 리눅스 OS 사용자 및 그 사용자와 관련된 그룹에 기반한 인증 및 인가를 활성화할 수 있음 - 이러한 보안 기능은 근본적인 파일 권한 요구사항을 변경하지 않으며, impala 계정은 여전히 데이터 파일에 접근할 수 있도록 여전히 필요로 함		
진단 방법	[진단기준] - 로컬 및 원격 모두 root 계정 로그인 비활성화 및 전용 계정(impala)으로 impalad 실행이 가능한 경우 양호 - 로컬 또는 원격으로 root 계정 로그인이 가능하거나 일반 사용자 계정으로 sudo 명령어를 이용한 impalad 실행이 가능한 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

1.2. 접근통제 설정

분류	설정	중요도	상
항목명	접근통제 설정		
항목 설명	만약 금융 관련 계좌번호 및 리눅스 파일시스템 내에 저장된 민감한 정보 등을 WHERE 절 내에 민감한 값을 포함하는 쿼리를 실행한 경우, 로그 파일에 기록이 됨 따라서, Impala와 관련된 설정, 데이터 및 로그 파일에 대한 접근통제를 설정하지 않은 경우 로그파일로부터의 정보유출이 가능함		
설정 방법	1. Impala와 관련된 모든 읽기/쓰기 연산은 impala 전용 계정의 파일시스템 권한 하에 수행되어야 함 2. Impala 전용 계정 사용자는 쿼리를 통해 모든 데이터 및 데이터 파일을 읽기 할 수 있도록 해야하고, INSERT 및 LOAD DATA 문으로 모든 디렉토리 및 데이터 파일 내 쿼리 및 쓰기 할 수 있도록 해야만 함 3. 최소한, impala 전용 계정은 Impala와 Hive 사이에 공유된 파일 및 디렉토리에 접근할 수 있도록 hive 그룹 내에 존재해야 함 4. Impala 로그 파일 위치 및 이름 [로그 파일 권한은 chmod로 640(rw-r-----)으로 설정] - o 기본적으로, /var/log/impala 디렉토리 하에 있는 로그파일이 존재함 로그 파일 위치 변경을 위해서는 /etc/default/impala 파일 내 변경이 필요함 - o impalad 프로세스를 위한 중요한 파일은 impalad.INFO, impalad.WARNING, impalad.ERROR 등이 존재함 또한, 비록 드문 조건으로 존재하지만, impalad.FATAL 파일도 볼 수 있을 수 있음 - o statestored 프로세스를 위한 중요한 파일은 statestored.INFO, statestored.WARNING, statestored.ERROR 등이 존재함 또한, 비록 드문 조건으로 존재하지만, statestored.FATAL 파일도 볼 수 있을 수 있음 - o catalogd 프로세스를 위한 중요한 파일은 catalogd.INFO, catalogd.WARNING, catalogd.ERROR 등이 존재함 또한, 비록 드문 조건으로 존재하지만, catalogd.FATAL 파일도 볼 수 있을 수 있음 - o 프로세스를 위한 환경설정을 볼 수 있는 .INFO 파일 검토 가능함 - o suboptimal 설정과 같은 것을 포함하는 모든 종류의 문제 정보 및 또한 심각한 런타임 오류를 볼 수 있는 .WARNING 파일 검토 가능함 - o 만약 프로세스 충돌 또는 쿼리 완료 실패시 대부분 심각한 오류만 볼 수 있는 .ERROR 파일 및/또는 .FATAL 파일 검토 가능함 이 메시지는 또한 .WARNING 파일 내에도 존재함		

	- o 연관된 데몬이 매번 재시작 될 때 새로운 로그 파일 집합이 생성됨 이 로그 파일은 타임스탬프를 포함한 긴 이름을 가지고 있음 .INFO, .WARNING 및 .ERROR 파일이 최신 적용 가능한 로그 파일에 대한 심볼릭 링크로 물리적으로 표현됨 - o impala-server 서비스를 위한 init 스크립트는 또한 .INFO, .WARNING 및 .ERROR 파일과 대응하는 모든 동일한 정보를 지닌 강화된 /var/logs/impalad/impala-server.log 로그 파일을 생성함 - o impala-state-store 서비스를 위한 init 스크립트는 또한 .INFO, .WARNING 및 .ERROR 파일과 대응하는 모든 동일한 정보를 지닌 강화된 /var/logs/impalad/impala-state-store.log 로그 파일을 생성함 5. Impala는 glog_v 로깅 시스템을 이용하여 정보를 저장함 C++ 파일 이름에 참조한 일부 메시지를 볼 수 있게 됨 - o GLOG_V 환경변수는 로깅될 메시지 종류를 지정함 - o impalad 데몬을 위한 -logbuflevel 시작 플래그는 얼마나 자주 로그 정보를 디스크에 기록할 지를 지정함 기본 값은 0이며, 이는 Impala가 Warning 또는 Error와 같은 중요한 메시지를 출력할 때 디스크로 즉시 로그로 기록하는 것을 의미하지만, Informational과 같은 덜 중요한 메시지는 즉시 디스크에 기록되기 보다는 메모리에 버퍼로 유지함 - o Cloudera Manager는 -logbuflevel 시작 옵션을 설정하는 Impala 환경설정을 가짐 ※ GLOG : Google Logging 모듈의 C++ 구현 또는 라이브러리
진단 방법	[진단기준] - 올바르게 파일 보호하도록 설정한 경우 양호 - 올바르게 파일 보호하도록 설정하지 않은 경우 취약
비고	중기 적용(적용 시 개발자 및 운영자 협의)

1.3. 기본 포트 변경

분류	설정	중요도	중
항목명	기본 포트 변경		
항목 설명	각 Impala 데몬(impalad, statestored 및 catalogd)은 진단내용 및 상태 정보를 표시하는 내장된 웹 서버를 포함함 만약 각 Impala 데몬이 사용하는 기본 포트 변경 및 패스워드를 이용한 접근통제를 설정하지 않으면 비인가가 접근을 통한 진단내용 및 상태 정보 노출이 가능함 ※ 기본 포트 정보 - o impalad 웹 UI [기본 포트: 25000] - 구성 설정, 실행 및 완료한 쿼리, 쿼리와 관련된 성능 및 리소스 사용 정보 - Details 링크에서 작업 계획의 그래픽 표현을 포함한 각 쿼리 뷰 및 impala-shell로부터 EXPLAIN, SUMMARY 및 PROFILE 출력 표시		

	- 특정 노드에 대해 Trace 할 수 있는 쿼리 문제를 진단하기 위해 주로 사용됨 o statestored 웹 UI [기본 포트: 25010] - 메모리 사용, 구성 설정 및 이 데몬에 의해 계속 진행 중인 헬스체크를 포함함 - 어떤 클러스터 내 이 데몬에 대한 단일 인스턴스 뿐만 있기 때문에, Impala Statestore로서 서비스하는 특정 호스트에 대해서만 웹 UI를 볼 수 있음 o catalogd 웹 UI [기본 포트: 25020] - DB, 테이블 및 그 외 Impala에 의해 관리되는 객체에 대한 정보를 포함함 - catalogd의 리소스 사용 및 구성 설정 정보를 포함함 - catalog 정보는 근본적인 Thrift 데이터 구조로서 나타냄 - 어떤 클러스터 내 이 데몬에 대한 단일 인스턴스 뿐만 있기 때문에, Impala Catalog Server로서 서비스하는 특정 호스트에 대해서만 웹 UI를 볼 수 있음 ※ 참고 웹 UI는 주로 문제 진단 및 트러블슈팅을 위해 사용하므로, 특히 모든 클러스터에 대해 Impala 헬스체크를 위해서는, Cloudera Manager Interface 사용 권고함					
설정 방법	1. Impala 웹 UI를 사용하는 경우, 패스워드로 보호되도록 설정해야 함 - o Impala Service로 이동해야 함 - o Configuration 탭을 클릭해야 함 - o Configuration 페이지 내에 Search 박스를 이용하여 “password”를 검색해야 함 이로써 Impala Catalog 서버, 데몬 및 StateStore에 대한 패스워드 관련 Property를 출력해야 함 (Username 및 Password Property) 만약 Impala 데몬 인스턴스를 위해 구성된 다중 Role 그룹이 존재한다면, 검색 결과는 그와 관련된 모든 결과를 출력해야 함 - o 이 필드 내의 username과 password를 출력해야 함 - o 변경을 적용하기 위해 Save Changes를 클릭해야 함 - o Impala 서비스를 재시작 해야 함 - o 정상 적용시, Impala Catalog 서버, 데몬 및 StateStore를 위한 웹 UI를 접근할 때, 접근에 대한 권한을 부여받기 전에 로그인을 요청받게 됨 2. 기본 포트가 아닌 다른 포트로 변경해야 함 - o 로그 파일 위치 변경을 위해서는 /etc/default/impala 파일 내 변경이 필요함 <table border="1"><tr><td>/etc/default/impala</td></tr><tr><td>IMPALA_STATE_STORE_HOST=127.0.0.1</td></tr><tr><td>IMPALA_STATE_STORE_PORT=24000</td></tr></table> 3. 웹 UI 접근을 OFF해야 함 <table border="1"><tr><td>[1] Impala Daemon</td></tr><tr><td> - o Impala Service 페이지로 이동해야 함 - o Configuration 탭을 클릭해야 함 - o Scope > Impala Daemon을 선택해야 함 - o Category > Ports and Addresses를 선택해야 함 - o Enable Impala Daemon Web Server 체크를 없애야 함 - o 변경을 적용하기 위해 Save Changes를 클릭해야 함 </td></tr></table>	/etc/default/impala	IMPALA_STATE_STORE_HOST=127.0.0.1	IMPALA_STATE_STORE_PORT=24000	[1] Impala Daemon	- o Impala Service 페이지로 이동해야 함 - o Configuration 탭을 클릭해야 함 - o Scope > Impala Daemon을 선택해야 함 - o Category > Ports and Addresses를 선택해야 함 - o Enable Impala Daemon Web Server 체크를 없애야 함 - o 변경을 적용하기 위해 Save Changes를 클릭해야 함
/etc/default/impala						
IMPALA_STATE_STORE_HOST=127.0.0.1						
IMPALA_STATE_STORE_PORT=24000						
[1] Impala Daemon						
- o Impala Service 페이지로 이동해야 함 - o Configuration 탭을 클릭해야 함 - o Scope > Impala Daemon을 선택해야 함 - o Category > Ports and Addresses를 선택해야 함 - o Enable Impala Daemon Web Server 체크를 없애야 함 - o 변경을 적용하기 위해 Save Changes를 클릭해야 함						

	o Impala 서비스를 재시작 해야 함
	[2] Impala StateStore o Impala Service 페이지로 이동해야 함 o Configuration 탭을 클릭해야 함 o Scope > Impala StateStore를 선택해야 함 o Category > All을 선택해야 함 o Enable StateStore Web Server 체크를 없애야 함 o 변경을 적용하기 위해 Save Changes를 클릭해야 함 o Impala 서비스를 재시작 해야 함
	[3] Impala Catalog Server o Impala Service 페이지로 이동해야 함 o Configuration 탭을 클릭해야 함 o Scope > Impala Catalog Server를 선택해야 함 o Category > All을 선택해야 함 o Enable Catalog Server Web Server 체크를 없애야 함 o 변경을 적용하기 위해 Save Changes를 클릭해야 함 o Impala 서비스를 재시작 해야 함
진단 방법	[진단기준] - Impala 웹 UI에 대한 패스워드 설정 및 다른 포트로 변경 또는 웹 UI를 OFF한 경우 양호 - Impala 웹 UI에 대한 패스워드 미설정 또는 기본 포트 사용 중인 경우 취약
비고	장기 적용(적용 시 개발자 및 운영자 협의)

1.4. SSL/TLS 설정

분류	설정	중요도	권고
항목명	SSL/TLS 설정		
항목 설명	SSL/TLS 설정을 하지 않은 경우, 사용자 또는 Cloudera Manager가 Impala Catalog 서버, 데몬 및 StateStoreweb 서버에 대한 인증서 평문 통신으로 인한 계정정보 탈취가 가능함		
설정 방법	1. 클라이언트 애플리케이션이 Impala에 연결할 때 SSL을 활성화하기 위해서는 다음의 플래그를 impalad 시작 옵션에 추가해야 함		
	플래그명	설명	
	--ssl_server_certificate	로컬 파일시스템 상에서 서버 인증서(공개키)에 대한 전체 경로	
	--ssl_private_key	로컬 파일시스템 상에서 서버의 개인키(비밀키)에 대한 전체 경로	
진단 방법	[진단기준] - SSL/TLS를 설정한 경우 양호 - SSL/TLS를 설정하지 않은 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

11. Jenkins

11.1. 설정

11.1.1. 기본 보안설정 활성화

분류	설정	중요도	상
항목명	기본 보안설정 활성화		
항목 설명	Jenkins 기본 보안설정 활성화하여 정보누출, 침해사고 대응 등 보안 사고에 대비하여야 함.		
설정 방법	1. \$JENKINS_HOME/config.xml내 아래 설정 확인 <useSecurity>true</useSecurity>		
진단 방법	[진단기준] - 기본 보안설정이 활성화된 경우 양호 - 기본 보안설정이 비활성화된 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

11.1.2. 기본 사용자 설정

분류	설정	중요도	중
항목명	기본 사용자 설정		
항목 설명	Jenkins는 설치 직후 Jenkins 사용자라면 누구나 실행 할 수 있도록 허용되기 때문에 Jenkins내 자체 독립 사용자 데이터베이스를 통해 관리자가 사용자 활동 내역을 확인 할 수 있도록 관리해야 함.		
설정 방법	사용자 데이터베이스 설정 1. Jenkins가 구동중인상태에서 http://_server_:8080 또는 http://_server_/jenkins:8080을 입력하여 Jenkins 대시보드로 이동 2. Manage Jenkins를 선택, Configure Global Security로 이동 3. Enable Security를 클릭 4. Jenkins' own user database를 선택 5. 사용자 추가 및 정책 설정		
진단 방법	[진단기준] - 사용자 데이터베이스 설정이 되어 있는 경우 양호 - 사용자 데이터베이스 설정이 되어 있지 않는 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

11.2. 계정 관리

11.2.1. 사용자 권한 설정

분류	계정 관리	중요도	상
항목명	사용자 권한 설정		
항목 설명	Jenkins는 설치 직후 Jenkins 사용자라면 누구나 실행 할 수 있도록 허용되기 때문에 Jenkins 권한 부여 전략을 통해 사용자 및 그룹에 특정 사용 권한을 부여해야 함.		
설정 방법	Matrix-based 1. Matrix 기반 보안을 인증으로 선택 2. 익명의 사용자에게는 Overall Read 권한 부여 3. 매트릭스 아래의 텍스트 상자에 사용자 이름(또는 새 Jenkins사용자로 등록할 이름)을 입력하고 추가 4. 전체 행에서 사용자 이름을 확인하여 알맞은 권한 부여 5. 페이지 하단의 저장버튼 클릭 6. 서비스 다시 시작		
진단 방법	[진단기준] - 권한이 적절히 부여된 경우 양호 - 부적절한 권한이 부여된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

11.3. 추가 기능 설정

11.3.1. CSRF 차단 설정

분류	추가 기능 설정	중요도	중
항목명	CSRF 차단 설정		
항목 설명	CSRF는 최종 사용자가 Jenkins에서 원하지 않는 동작을 수행하도록 하는 공격으로, Jenkins 1.x 및 2.z로 업그레이드 시 기본적으로 해제되어 있어 추가 설정이 필요함.		
설정 방법	CSRF 차단 방법 "Manage Jenkins" > "Configure Global Security" and select "Prevent Cross Site Request Forgery exploits." 에 아래내용 추가 <pre>import hudson.security.csrf.DefaultCrumbIssuer import jenkins.model.Jenkins def instance = Jenkins.instance instance.setCrumbIssuer(new DefaultCrumbIssuer(true)) instance.save()</pre>		
진단	[진단기준]		

방법	- CSRF 차단 설정이 되어 있을 경우 <u>양호</u> - CSRF 차단 설정이 되어 있지 않을 경우 <u>취약</u>
비고	중기 적용(적용 시 개발자 및 운영자 협의)



12. Kafka

12.1. SSL 설정

12.1.1. SSL 설정 (Kafka Broker)

분류	SSL 설정	중요도	권고
항목명	SSL 설정 (Kafka Broker)		
항목 설명	Apache Kafka는 클라이언트가 SSL을 통해 연결하는 것을 허용하지만, 기본적으로 SSL 기능이 비활성화가 되어 있음 만약 SSL 비활성화한 경우에는 Broker와 클라이언트 간 평문 노출로 인한 정보 유출 가능성이 존재함		
설정 방법	1. SSL용 키 생성 및 사인 수행		
	keytool -keystore server.keystore.jks -alias localhost -validity {validity} -genkey -keyalg RSA		
	※ <u>keystore</u> 인증서를 저장하는 keystore 파일. keystore 파일은 인증서의 개인키(비밀키)를 포함하므로, 안전한 보관이 필요함		
	※ <u>validity</u> 인증서 유효기간을 지정함		
	※ 기본 값으로 호스트명 검증을 수행하지 않으므로, 만약 호스트 검증을 수행할 경우 다음과 같은 Property를 설정하면 됨		
	ssl.endpoint.identification.algorithm=HTTPS		
	※ 호스트명 검증시, 다음의 2개의 필드 중 1개를 검증함		
	1. Common Name (CN)		
	2. Subject Alternative Name (SAN)		
	※ SAN 필드를 추가하려면 -ext SAN=DNS:{FQDN}과 같이 사용함		
keytool -keystore server.keystore.jks -alias localhost -validity {validity} -genkey -keyalg RSA -ext SAN=DNS:{FQDN}			
2. Kafka Broker를 통하여 SSL 활성화			
[1] SSL inter-broker communication 비활성화 상태에서			
ssl.keystore.location=/var/private/ssl/server.keystore.jks			
ssl.keystore.password=test1234			
ssl.key.password=test1234			
ssl.truststore.location=/var/private/ssl/server.truststore.jks			
ssl.truststore.password=test1234			
[2] SSL inter-broker communication 활성화 상태에서			
security.inter.broker.protocol=SSL			
3. 기타 옵션			
옵션명		설명	
.ssl.client.auth=none		1. “required” 클라이언트 인증이 필요함	

		2. “requested” 클라이언트 인증 요청을 받으나, 인증서 없어도 클라이언트는 여전히 연결 가능함 ※ “requested” 사용은 보안의 잘못된 인식 제공하며, 잘못 구성된 클라이언트가 여전히 연결 성공을 하도록 제공하게 됨
	ssl.cipher.suites (Optional)	암호화 알고리즘 집합은 유명한 인증, 암호화, MAC 및 TLS 또는 SSL을 이용하여 네트워크 연결을 위한 보안설정을 협상하기 위해 사용되는 키 교환 알고리즘 집합체임 (기본 값은 빈 목록임)
	ssl.enabled.protocols=TLSv1.2,TLSv1.1,TLSv1	클라이언트로부터 승인할 SSL 프로토콜의 목록이며, TLS 사용을 권장함
	ssl.truststore.type=JKS	Broker에서 클라이언트 인증이 필요하지 않은 경우 SSL 인증서 유형을 JKS 사용함
	ssl.keystore.type=JKS	Broker에서 클라이언트 인증이 필요한 경우 SSL 인증서 유형을 JKS 사용함
	6.ssl.secure.random.implementation=SHA1PRNG	안전한 난수 생성 알고리즘 선택함 (단, 미 수출규정에 의해 256비트 AES가 필요할 경우, 오라클 홈페이지에서 JCE Unlimited Strength Jurisdiction Policy Files을 다운로드 및 설치를 해야 함)
진단 방법	[진단기준] - Kafka Broker에 SSL 설정 적용한 경우 양호 - Kafka Broker에 SSL 설정 적용하지 않은 경우 취약	
비고		

12.1.2. SSL 설정 (Kafka 클라이언트)

분류	SSL 설정	중요도	권고
항목명	SSL 설정 (Kafka 클라이언트)		
항목 설명	Apache Kafka는 클라이언트가 SSL을 통해 연결하는 것을 허용하지만, 기본적으로 SSL 기능이 비활성화가 되어 있음 만약 SSL 비활성화한 경우에는 Broker와 클라이언트 간 평문 노출로 인한 정보 유출 가능성이 존재함 ※ <u>참고</u> SSL은 새로운 Kafka Producer와 Consumer만 지원하며, 오래된 API는 지원하지 않음 SSL을 위한 설정은 Producer와 Consumer 둘 다 동일하게 적용해야 함		
설정 방법	1. Kafka 클라이언트를 통하여 SSL 활성화		

	[1] Broker에서 클라이언트 인증이 필요하지 않은 경우	
	security.protocol=SSL ssl.truststore.location=/var/private/ssl/client.truststore.jks ssl.truststore.password=test1234	
	[2] Broker에서 클라이언트 인증이 필요한 경우	
	ssl.keystore.location=/var/private/ssl/client.keystore.jks ssl.keystore.password=test1234 ssl.key.password=test1234	
	2. 기타 옵션	
	옵션명	설명
	ssl.provider (Optional)	SSL 연결을 위해 사용하는 보안 제공자명
	ssl.cipher.suites (Optional)	TLS 또는 SSL 네트워크 프로토콜을 이용하기 위해 네트워크 연결을 위한 보안설정을 협상하기 위해 사용되는 인증, 암호화, MAC 및 키 교환 알고리즘의 명명된 조합
	ssl.enabled.protocols=TLSv1.2,TLSv1.1,TLSv1	Broker 측에 구성된 최소한 1개의 프로토콜을 열거해야 함
	ssl.truststore.type=JKS	Broker에서 클라이언트 인증이 필요하지 않은 경우 SSL 인증서 유형을 JKS 사용함
	ssl.keystore.type=JKS	Broker에서 클라이언트 인증이 필요한 경우 SSL 인증서 유형을 JKS 사용함
	3. 적용 예제	
	[참고] console-producer와 console-consumer를 이용한 예제	
	kafka-console-producer.sh --broker-list localhost:9093 --topic test --producer.config client-ssl.properties	
	kafka-console-consumer.sh --bootstrap-server localhost:9093 --topic test --consumer.config client-ssl.properties	
진단 방법	[진단기준] - Kafka 클라이언트에 SSL 설정 적용한 경우 양호 - Kafka 클라이언트에 SSL 설정 적용하지 않은 경우 취약	
비고		

13. Maven

13.1. 설정

13.1.1. HTTP Method 제한

분류	설정	중요도	하
항목명	HTTP Method 제한		
항목 설명	GET, POST, HEAD, OPTIONS 이외의 다른 HTTP Method 를 지원하는 경우, 악의적인 공격자가 임의의 파일을 삭제하거나 업로드하여 서버의 정상 운영에 지장을 줄 수 있음.		
설정 방법	1. Settings.xml 파일에 필요한 Method만 허용하도록 설정 PUT, GET, HEAD만 허용하도록 설정 <pre><settings> [...] <servers> <server> <id>the-server</id> <configuration> <httpConfiguration> <put> [설정.] </put> </httpConfiguration> </configuration> </server> </servers> </settings></pre>		
진단 방법	[진단기준] - PUT, GET, HEAD만 허용하도록 설정한 경우 양호 - PUT, GET, HEAD외 다른 HttpMethod가 활성화 된 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

13.1.2. 설정파일 권한 설정

분류	설정	중요도	상
항목명	설정파일 권한 설정		
항목 설명	일반 사용자가 설정 파일을 삭제, 변경할 수 있을 경우 시스템이 오작동하여 사용 불능 상태에 빠질 우려가 있음.		
설정 방법	아래파일의 권한을 700 이하로 설정 <pre> \${user.home}/.m2/settings-security.xml \${user.home}/.m2/settings.xml \${user.home}/.m2/repository/ </pre>		
진단 방법	[진단기준] - 전용 계정 소유이고, 600(-rw-----) 또는 700(-rwx-----) 권한인 경우 양호 - 전용 계정 소유가 아니거나, 600(-rw-----) 또는 700(-rwx-----) 권한 초과인 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

13.1.3. 세션 타임 아웃 설정

분류	설정	중요도	하
항목명	세션 타임 아웃 설정		
항목 설명	세션 타임아웃을 너무 길게 설정하여 공격자가 세션을 사용 할 수 있어 세션 타임을 설정하여야 함		
설정 방법	settings.xml 파일에 아래와 같이 수정 <pre> <settings> <servers> <server> <id>my-server</id> <configuration> <timeout>시간설정</timeout> </configuration> </server> </servers> </settings> </pre>		
진단 방법	[진단기준] - timeout 설정을 30분 미만으로 설정한 경우 양호 - timeout 설정을 30분 이상으로 설정한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

13.2. 계정 암호화 설정

13.2.1. 마스터 패스워드 암호화 설정

분류	계정 암호화 설정	중요도	상
항목명	마스터 패스워드 암호화 설정		
항목 설명	Maven 솔루션 전체를 관리 할 수 있는 마스터 계정의 패스워드를 암호화 하지 않는 경우 비인가자가 패스워드를 획득하여 마스터 계정으로 로그인 할 수 있는 우려가 있음		
설정 방법	- 명령어를 통하여 마스터 패스워드 생성 mvn --encrypt-master-password <password> - 암호화 패스워드 생성 확인 {암호화 패스워드} 3. \${user.home}/.m2/settings-security.xml에 생성한 패스워드 입력 <settingsSecurity> <master>{암호화패스워드}</master> </settingsSecurity>		
진단 방법	[진단기준] - 마스터패스워드가 암호화 되어 있는 경우 양호 - 마스터패스워드가 평문으로 설정되어 있는 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

13.2.2. 사용자 계정 비밀번호 암호화 설정

분류	계정 암호화 설정	중요도	상
항목명	사용자 계정 비밀번호 암호화 설정		
항목 설명	비밀번호를 암호화 하지 않는 경우 비인가자가 비밀번호를 획득하여 사용자 계정으로 로그인 할 수 있는 우려가 있음		
설정 방법	1. 명령어를 통하여 암호화 된 사용자 비밀번호 생성 <code>mvn --encrypt-password <password></code> 2. 생성한 암호화 비밀번호 확인 <code>{암호화 비밀번호}</code> 3. Settings.xml에 비밀번호 입력 <code><settings></code> <code>...</code> <code><servers></code> <code>...</code> <code><server></code> <code><id>my.server</id></code> <code><username>foo</username></code> <code><password>{암호화 비밀번호}</password></code> <code></server></code> <code>...</code> <code></servers></code> <code>...</code> <code></settings></code> 4. 플러그인 배포 수행 <code>mvn deploy:deploy-file -Durl=https://maven.corp.com/repo ₩</code> <code>-DrepositoryId=my.server ₩</code> <code>-Dfile=your-artifact-1.0.jar ₩</code>		
진단 방법	[진단기준] - 비밀번호가 암호화 되어 있는 경우 양호 - 비밀번호가 평문으로 설정되어 있는 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

14. Mesos

14.1. 인증 설정

14.1.1. 인증 설정 (Master)

분류	인증 설정	중요도	상								
항목명	인증 설정 (Master)										
항목 설명	인증은 Mesos 클러스터와 상호작용하기 위해 신뢰하는 엔티티만 허용해야 함 인증은 3가지 방식으로 사용할 수 있음 - o Framework가 Master에 등록하기 위해 인증되도록 요구 - o Agent가 Master에 등록하기 위해 인증되도록 요구 - o Operator가 많은 HTTP 엔드포인트를 사용하기 위해 인증되도록 요구 인증은 기본적으로 비활성화 되어있음 인증 활성화시, Operator는 Mesos에 기본 인증을 사용하거나 사용자 정의 인증 모듈을 사용하도록 구성할 수 있음 기본 Mesos 인증 모듈은 Cyrus SASL 라이브러리를 사용함 SASL은 두 개의 엔드포인트가 다양한 메소드를 이용하여 서로 인증하는 것을 허용하는 유연한 Framework임 기본적으로, Mesos는 CRAM-MD5 인증을 사용함 ※ <u>SASL(Simple Authentication and Security Layer)</u> 인증과 대체할 수 있는 메커니즘을 통해 연결 지향 프로토콜 내 보안 서비스를 제공하는 프레임워크 ※ <u>CRAM(Challenge-Response Authentication Mechanism)</u> Challenge-Response 기반 인증 메커니즘 (예: 패스워드 인증 등)										
설정 방법	1. 인증은 Mesos Master 프로세스 시작시 명령어 라인 플래그를 지정하여 구성 가능함										
	<table><tr><th>옵션명</th><th>설명</th></tr><tr><td>--[no-]authenticate</td><td>만약 참일 경우, 인증된 프레임워크만 등록을 허용함. 만약 거짓(기본 값)일 경우, 비인증된 프레임워크도 등록을 허용함</td></tr><tr><td>--[no-]authenticate_http_readonly</td><td>만약 참일 경우, 인증은 인증을 지원하는 읽기 전용 HTTP 엔드포인트에 대해 HTTP 요청 생성을 필요로 함. 만약 거짓(기본 값)일 경우, 이런 엔드포인트는 인증 없이 사용할 수 있음. 읽기 전용 엔드포인트는 클러스터의 상태를 변경하기 위해 사용할 수 없음</td></tr><tr><td>--[no-]authenticate_http_readwrite</td><td>만약 참일 경우, 인증은 인증을 지원하는 읽기/쓰기 HTTP 엔드포인트에 대해 HTTP 요청 생성을 필요로 함. 만약 거짓(기본 값)일 경우, 이런 엔드포인트는 인증 없이 사용할 수 있음. 읽기/쓰기 엔드포인트는 클러스터의 상태를 변경하기 위해 사용할 수 있음</td></tr></table>			옵션명	설명	--[no-]authenticate	만약 참일 경우, 인증된 프레임워크만 등록을 허용함. 만약 거짓(기본 값)일 경우, 비인증된 프레임워크도 등록을 허용함	--[no-]authenticate_http_readonly	만약 참일 경우, 인증은 인증을 지원하는 읽기 전용 HTTP 엔드포인트에 대해 HTTP 요청 생성을 필요로 함. 만약 거짓(기본 값)일 경우, 이런 엔드포인트는 인증 없이 사용할 수 있음. 읽기 전용 엔드포인트는 클러스터의 상태를 변경하기 위해 사용할 수 없음	--[no-]authenticate_http_readwrite	만약 참일 경우, 인증은 인증을 지원하는 읽기/쓰기 HTTP 엔드포인트에 대해 HTTP 요청 생성을 필요로 함. 만약 거짓(기본 값)일 경우, 이런 엔드포인트는 인증 없이 사용할 수 있음. 읽기/쓰기 엔드포인트는 클러스터의 상태를 변경하기 위해 사용할 수 있음
	옵션명	설명									
	--[no-]authenticate	만약 참일 경우, 인증된 프레임워크만 등록을 허용함. 만약 거짓(기본 값)일 경우, 비인증된 프레임워크도 등록을 허용함									
--[no-]authenticate_http_readonly	만약 참일 경우, 인증은 인증을 지원하는 읽기 전용 HTTP 엔드포인트에 대해 HTTP 요청 생성을 필요로 함. 만약 거짓(기본 값)일 경우, 이런 엔드포인트는 인증 없이 사용할 수 있음. 읽기 전용 엔드포인트는 클러스터의 상태를 변경하기 위해 사용할 수 없음										
--[no-]authenticate_http_readwrite	만약 참일 경우, 인증은 인증을 지원하는 읽기/쓰기 HTTP 엔드포인트에 대해 HTTP 요청 생성을 필요로 함. 만약 거짓(기본 값)일 경우, 이런 엔드포인트는 인증 없이 사용할 수 있음. 읽기/쓰기 엔드포인트는 클러스터의 상태를 변경하기 위해 사용할 수 있음										

--[no-]authenticate_agents	만약 참이라면, 인증된 에이전트만 등록을 허용함. 만약 거짓(기본 값)일 경우, 비인증된 에이전트도 등록을 허용함
--authenticators	사용할 Authenticator 모듈을 지정함. 기본 값으로 crammd5이지만, 부가 모듈도 --modules 옵션을 이용하여 추가할 수 있음
--http_authenticators	사용할 HTTP Authenticator 모듈을 지정함. 기본 값으로 basic (기본 HTTP 인증)이지만, 부가 모듈도 --modules 옵션을 이용하여 추가할 수 있음
--credentials	승인된 자격증명(Credential)의 목록을 포함하는 텍스트 파일에 대한 경로. 이 플래그는 사용하는 Authenticator에 따라 옵션으로 사용함

2. CRAM-MD5 적용 예제 (Master, Agent 공통)

[1] 다음의 내용으로 Master의 자격증명(Credential)을 생성함

```
{
  "credentials": [
    {
      "principal": "principal1",
      "secret": "secret1"
    },
    {
      "principal": "principal2",
      "secret": "secret2"
    }
  ]
}
```

[2] 자격증명 파일을 이용하여 master를 시작함

(파일은 /home/user/credentials로 가정함)

```
./bin/mesos-master.sh --ip=127.0.0.1 --work_dir=/var/lib/mesos --authenticate --
authenticate_agents --credentials=/home/user/credentials
```

[3] 아래의 경로 내에 단일 자격증명으로 또 다른 파일을 생성함

```
/home/user/agent_credential
```

[4] Agent를 시작함

```
./bin/mesos-agent.sh --master=127.0.0.1:5050 --credential=/home/user/
agent_credential
```

[5] 새로운 Agent는 지금 Master와 함께 성공적으로 인증되어야 함

	[6] 다음과 같이 Mesos와 함께 제공된 test Framework 중 하나를 사용하여 Framework 인증을 테스트할 수 있음 MESOS_AUTHENTICATE=true DEFAULT_SECRET=secret2 ./src/test-framework --master=127.0.0.1:5050 DEFAULT_PRINCIPAL=principal2
진단 방법	[진단기준] - Master에 인증 설정 적용한 경우 양호 - Master에 인증 설정 적용하지 않은 경우 취약
비고	

14.1.2. 인증 설정 (Agent)

분류	인증 설정	중요도	상					
항목명	인증 설정 (Agent)							
항목 설명	인증은 Mesos 클러스터와 상호작용하기 위해 신뢰하는 엔티티만 허용해야 함 인증은 3가지 방식으로 사용할 수 있음 - o Framework가 Master에 등록하기 위해 인증되도록 요구 - o Agent가 Master에 등록하기 위해 인증되도록 요구 - o Operator가 많은 HTTP 엔드포인트를 사용하기 위해 인증되도록 요구 인증은 기본적으로 비활성화 되어있음 인증 활성화시, Operator는 Mesos에 기본 인증을 사용하거나 사용자 정의 인증 모듈을 사용하도록 구성할 수 있음 기본 Mesos 인증 모듈은 Cyrus SASL 라이브러리를 사용함 SASL은 두 개의 엔드포인트가 다양한 메소드를 이용하여 서로 인증하는 것을 허용하는 유연한 Framework임 기본적으로, Mesos는 CRAM-MD5 인증을 사용함 ※ <u>SASL(Simple Authentication and Security Layer)</u> 인증과 대체할 수 있는 메커니즘을 통해 연결 지향 프로토콜 내 보안 서비스를 제공하는 프레임워크 ※ <u>CRAM(Challenge-Response Authentication Mechanism)</u> Challenge-Response 기반 인증 메커니즘 (예: 패스워드 인증 등)							
	1. 인증은 Mesos Agent 프로세스 시작시 명령어 라인 플래그를 지정하여 구성 가능함 <table><tr><th>옵션명</th><th>설명</th></tr><tr><td>--authenticatee</td><td>Master의 --authenticators 옵션으로 사용할 모듈을 지정하는 옵션에 대한 유사체 기본 값으로 crammd5를 사용함</td></tr><tr><td>--credential</td><td>유일한 자격증명(Credential)만 허용되는 것을 제외 하고는 Master의 --credentials 옵션과 같음</td></tr></table>			옵션명	설명	--authenticatee	Master의 --authenticators 옵션으로 사용할 모듈을 지정하는 옵션에 대한 유사체 기본 값으로 crammd5를 사용함	--credential
옵션명	설명							
--authenticatee	Master의 --authenticators 옵션으로 사용할 모듈을 지정하는 옵션에 대한 유사체 기본 값으로 crammd5를 사용함							
--credential	유일한 자격증명(Credential)만 허용되는 것을 제외 하고는 Master의 --credentials 옵션과 같음							
설정 방법								

	이 자격증명은 Master에 대한 Agent를 식별하기 위해 사용됨
--[no-]authenticate_http_readonly	만약 참이라면, 인증은 인증을 지원하는 읽기 전용 HTTP 엔드포인트에 대한 HTTP 요청 생성이 필요함. 만약 거짓(기본 값)이라면, 이 엔드포인트는 인증 없이 사용할 수 있음. 읽기 전용 엔드포인트는 에이전트의 상태를 변경하기 위해 사용될 수 없음
--[no-]authenticate_http_readwrite	만약 참이라면, 인증은 인증을 지원하는 읽기/쓰기 HTTP 엔드포인트에 대한 HTTP 요청 생성이 필요함 만약 거짓(기본 값)이라면, 이 엔드포인트는 인증 없이 사용할 수 있음. 읽기/쓰기 엔드포인트는 에이전트의 상태를 변경하기 위해 사용될 수 있음 하위 호환성의 이유로, V1 executor API는 이 플래그에 의해 영향받지 않음에 주의해야 함
--[no-]]authenticate_http_executors	만약 참이라면, 인증은 V1 executor API에 대한 HTTP 요청 생성이 필요함. 만약 거짓(기본 값)이라면, 해당 API는 인증 없이 사용할 수 있음 만약 이 플래그가 참이고, 사용자 정의 HTTP Authenticator를 지정하지 않으면, 기본 값으로 지정된 JWT Authenticator가 executor 인증을 처리하기 위해 로드함
--http_authenticators	사용할 HTTP Authenticator 모듈을 지정함. 기본 값으로 basic (기본 HTTP 인증)이지만, 부가 모듈도 --modules 옵션을 이용하여 추가할 수 있음
--http_credentials	승인된 자격증명(Credential)의 목록을 포함하는 텍스트 파일에 대한 경로 (JSON 포맷). 이 플래그는 사용하는 Authenticator에 따라 옵션으로 사용함

2. CRAM-MD5 적용 예제 (Master, Agent 공통)

[1] 다음의 내용으로 Master의 자격증명(Credential)을 생성함

```
{
  "credentials": [
    {
      "principal": "principal1",
      "secret": "secret1"
    },
    {
      "principal": "principal2",
      "secret": "secret2"
    }
  ]
}
```

[2] 자격증명 파일을 이용하여 master를 시작함

	(파일은 /home/user/credentials로 가정함) ./bin/mesos-master.sh --ip=127.0.0.1 --work_dir=/var/lib/mesos --authenticate --authenticate_agents --credentials=/home/user/credentials [3] 아래의 경로 내에 단일 자격증명으로 또 다른 파일을 생성함 /home/user/agent_credential [4] Agent를 시작함 ./bin/mesos-agent.sh --master=127.0.0.1:5050 --credential=/home/user/agent_credential [5] 새로운 Agent는 지금 Master와 함께 성공적으로 인증되어야 함 [6] 다음과 같이 Mesos와 함께 제공된 test Framework 중 하나를 사용하여 Framework 인증을 테스트할 수 있음 MESOS_AUTHENTICATE=true DEFAULT_PRINCIPAL=principal2 DEFAULT_SECRET=secret2 ./src/test-framework --master=127.0.0.1:5050
진단 방법	[진단기준] - Agent에 인증 설정 적용한 경우 양호 - Agent에 인증 설정 적용하지 않은 경우 취약
비고	

14.1.3. 다중 HTTP Authenticator 설정

분류	인증 설정	중요도	중				
항목명	다중 HTTP Authenticator 설정						
항목 설명	다중 HTTP Authenticator 미설정시, 단일 Authenticator에 대한 인증 공격 성공으로 Mesos 서비스 이용이 가능함 ※ 참고 Executor 인증을 수용하려면, 다중 Authenticator 모듈을 설정하여 로딩을 해야 함						
설정 방법	1. Mesos Master와 Agent에 --http_authenticators 플래그를 이용하여 Authenticator를 쉼표(,)로 분리하여 지정 후 로드할 수 있음 로딩할 때 사용하는 - http_authenticators 플래그는 인증된 엔드포인트에 대한 요청을 처리할 때 사용하기 위한 HTTP Authenticator 구현임 기본 값으로 basic을 사용하며, --modules를 이용하여 HTTP Authenticator 모듈을 로딩 하거나 대체 가능함						
	<table><tr><td>옵션명</td><td>설명</td></tr><tr><td>--http_authenticators</td><td>각 Authenticator는 직렬로 호출되어야 하며, 최초 성공 결과가 반환됨 만약 모든 인증 시도를 실패할 경우, 실패한 결과는</td></tr></table>			옵션명	설명	--http_authenticators	각 Authenticator는 직렬로 호출되어야 하며, 최초 성공 결과가 반환됨 만약 모든 인증 시도를 실패할 경우, 실패한 결과는
옵션명	설명						
--http_authenticators	각 Authenticator는 직렬로 호출되어야 하며, 최초 성공 결과가 반환됨 만약 모든 인증 시도를 실패할 경우, 실패한 결과는						

	다음과 같이 조합하게 됨 1. “Unauthorized”된 결과가 있을 경우 비인증된 결과는 통합되어 반환됨 2. “Unauthorized”된 결과가 없을 경우 다중 Forbidden 결과가 표시됨 Forbidden 결과는 조합 후 반환됨 3. “Forbidden”된 결과가 없을 경우 실패된 기능이 존재할 수도 있음 에러메시지는 미래에 실패하면 조합 후 반환됨
--	--

2. 만약 사용자 정의 Authenticator 모듈에다가 기본 Basic HTTP Authenticator를 지정하기
바란다면, Authenticator 목록에 이름 basic을 추가해야 함

사용자 정의 Authenticator 모듈에다가 기본 JWT HTTP Authenticator를 지정하기 위해서는
Authenticator 목록에 이름 jwt를 추가해야 함

3. 적용 결과 (Authenticator 2개 사용 : Basic Authenticator, Bearer Authenticator)

[1] 인증결과: 모두 “Unauthorized”인 경우
First result: Status code: 401 'WWW-Authenticate' header: 'Basic realm="mesos" Response body: 'Incorrect credentials'
Second result: Status code: 401 'WWW-Authenticate' header: 'Bearer realm="mesos" Response body: 'Invalid token'
Returned result: Status code: 401 'WWW-Authenticate' header: 'Basic realm="mesos", Bearer realm="mesos" Response body: '"Basic" authenticator returned: ' 'Incorrect credentials ' '"Bearer" authenticator returned: ' 'Invalid token'

[2] 인증결과: 하나는 “Unauthorized”, 나머지는 “Forbidden”인 경우
First result: Status code: 401 'WWW-Authenticate' header: 'Basic realm="mesos" Response body: 'Incorrect credentials'
Second result: Status code: 403 Response body: 'Not authorized'
Returned result: Status code: 401 'WWW-Authenticate' header: 'Basic realm="mesos" Response body: 'Incorrect credentials'

[3] 인증결과: 모두 “Forbidden”인 경우
First result: Status code: 403 Response body: 'Basic: not authorized'
Second result: Status code: 403 Response body: 'Bearer: not authorized'
Returned result: Status code: 403 Response body: '"Basic" authenticator returned: ' 'Basic: not authorized ' '"Bearer" authenticator returned: ' 'Bearer: not authorized'

진단 방법	[진단기준] - Master, Agent에 다중 HTTP Authenticator 설정 적용한 경우 양호 - Master, Agent에 다중 HTTP Authenticator 설정 적용하지 않은 경우 취약
비고	

14.1.4. Executor 인증 설정

분류	인증 설정	중요도	하			
항목명	Executor 인증 설정 (적용 대상: Mesos 1.0 이상)					
항목 설명	Executor는 /api/v1/executor Agent 엔드포인트를 통해 Mesos와 상호작용을 함 Executor는 가능하면 길게 Subscription 연결 개방을 유지하고, 이 Subscription 연결 개방이 증가하여 응답을 처리하는 것을 기대함 만약 Executor에 대한 토큰 미사용시, Agent 엔드포인트를 인증 없이 사용 가능함 ※ <u>SUBSCRIBE</u> Executor와 Agent 사이의 통신 프로세스에서의 첫 단계를 의미하며, 이는 또한 /executor 이벤트 스트림에 대한 Subscription으로 고려할 수 있음 ※ <u>/api/v1/executor Agent</u> Call/Event 메시지를 통해 에이전트와 상호작용하기 위해 Executor에 의해 사용되는 엔드포인트					
설정 방법	1. 만약 HTTP Executor 인증이 Agent에 활성화된 경우, HTTP Executor로부터의 모든 요청은 반드시 인증되어야 함					
	<table><tr><th>옵션명</th><th>설명</th></tr><tr><td>-- jwt_secret_key</td><td>Agent는 각각의 Executor가 실행되기 전에 기본 JWT를 생성하게 됨. 이 토큰은 MESOS_EXECUTOR_AUTHENTICATOR_TOKEN 환경변수를 이용하여 Executor의 환경으로 전달함 Agent와 인증하기 위해, Executor는 모든 요청에 대한 Authorization 헤더 내 이 토큰을 배치해야 함 예) Authorization: Bearer MESOS_EXECUTOR_AUTHENTICATION_TOKEN</td></tr></table>	옵션명	설명	-- jwt_secret_key	Agent는 각각의 Executor가 실행되기 전에 기본 JWT를 생성하게 됨. 이 토큰은 MESOS_EXECUTOR_AUTHENTICATOR_TOKEN 환경변수를 이용하여 Executor의 환경으로 전달함 Agent와 인증하기 위해, Executor는 모든 요청에 대한 Authorization 헤더 내 이 토큰을 배치해야 함 예) Authorization: Bearer MESOS_EXECUTOR_AUTHENTICATION_TOKEN	
옵션명	설명					
-- jwt_secret_key	Agent는 각각의 Executor가 실행되기 전에 기본 JWT를 생성하게 됨. 이 토큰은 MESOS_EXECUTOR_AUTHENTICATOR_TOKEN 환경변수를 이용하여 Executor의 환경으로 전달함 Agent와 인증하기 위해, Executor는 모든 요청에 대한 Authorization 헤더 내 이 토큰을 배치해야 함 예) Authorization: Bearer MESOS_EXECUTOR_AUTHENTICATION_TOKEN					
설정 방법	2. Executor 인증이 필요한 기존 클러스터를 업그레이드하기 위해, 다음의 절차를 따라야 함 - 모든 Agent를 업그레이드 해야하며, --jwt_secret_key 플래그를 통해 암호화키와 함께 각 Agent에 제공해야 함. 이 키는 HMAC-SHA256 프로시저를 이용하여 Executor 인증 토큰을 서명하기 위해 사용하게 됨 - Executor 인증이 성공적으로 사용할 수 있게 되기 직전에, 모든 HTTP Executor는 Executor가 기존 클러스터의 환경 내 인증 토큰을 가져야만 하며, 인증을 지원해야 함 이 것을 성공하기 위해, 업그레이드 전에 이미 실행 중인 Executor는 재시작해야 함 이는 한 번에 모든 것을 완료할 수 있으며, 그렇게 하지 않으면 클러스터는 Executor가 점진적으로 수행되는 반면, 중간(intermediate) 상태로 남을 수 있음 - 실행 중인 모든 기본/HTTP 명령 Executor를 한 번에 업그레이드된 Agent에 의해 실행 되었을 때, 사용자 정의 HTTP Executor는 업그레이드가 되었으며, 이 Agent 프로세스는 --authenticate_http_executors 플래그를 설정하여 재시작할 수 있음. 이는 필요로 하는 HTTP					

	Executor 인증을 사용할 것이며, 모든 Executor가 지금 인증 토큰과 인증을 지원하기 때문에, Agent에 대한 요청이 성공적으로 인증하게 됨 3. 실행 중인 모든 기본/HTTP 명령 Executor를 한 번에 업그레이드된 Agent에 의해 실행되었을 때, 사용자 정의 HTTP Executor는 업그레이드가 되었으며, 이 Agent 프로세스는 --authenticate_http_executors 플래그를 설정하여 재시작할 수 있음 이는 필요로 하는 HTTP Executor 인증을 사용할 것이며, 모든 Executor가 지금 인증 토큰과 인증을 지원하기 때문에, Agent에 대한 요청이 성공적으로 인증하게 됨
진단 방법	[진단기준] - Executor 인증 설정 적용한 경우 양호 - Executor 인증 설정 적용하지 않은 경우 취약
비고	

14.1.5. Framework 인증 설정

분류	인증 설정	중요도	하
항목명	Framework 인증 설정		
항목 설명	Mesos Frameworks는 작업을 관리하며, 고가용성을 위한 Mesos Framework에 대해 다양한 실패 시나리오 존재로 올바르게 작업을 관리하는 것을 계속 해야만 함 만약 Framework 인증 미설정시, 인증되지 않은 프레임워크가 작업 등록 및 제출 가능함		
설정 방법	1. 만약 Framework 인증이 활성화할 경우, 각 Framework는 Mesos Master와 함께 등록할 때 인증 자격증명(Credential)을 제공하도록 구성되어야만 함 ※ 참고 Framework 인증 구성방법은 Framework 별로 다르므로, Framework 벤더사의 기술지원을 받아야 함		
진단 방법	[진단기준] - Framework 인증 설정 적용한 경우 양호 - Framework 인증 설정 적용하지 않은 경우 취약		
비고			

14.2. 접근통제

14.2.1. ACL 설정

분류	인가 설정	중요도	하
항목명	ACL (Access Control List) 설정		
항목 설명	Mesos에서, 인증 하위시스템은 Operator가 확실한 Principal이 수행하도록 허용된 Action을 구성하도록 허용함. 예를 들면, Operator는 Principal foo가 Role bar에 Subscribe 등록만 할 수 있으며, 다른 Principal은 어떠한 Role도 Subscribe할 Framework를 등록할 수 없도록 보장해 줄 수 있음 만약 ACL을 구현하지 않을 경우, 비인증으로 어떤 Role이나 Framework에 대해 Subscribe 등록 또는 그 외 접근통제가 설정 안 되있는 모든 Principal을 통해 Action 구성이 가능함		
설정 방법	1. Permissive가 참이면, 명시적으로 허용된 Principal 제외한 모든 Object에 대한 Action을 수행하는 ANY Principal을 거부함 2. Permissive가 거짓이면, ACL 검사 없이 동작해야하는 모든 Action을 위한 종료가 될 필요가 있음 (ANY Object에 대한 Action을 수행하는 ANY Principal 허용은 제외 필요) 3. 적용 사례 [1] 다음의 ACL의 예를 고려함. 유일한 Principal인 foo는 analytics Role을 Subscribe할 Framework 등록할 수 있음. 모든 Principal은 나머지 Role(Permissive가 기본 행위 이므로 Principal foo를 포함함)에 대해 Subscribe 중인 Framework를 등록 가능함 <pre> { "register_frameworks": [{ "principals": { "values": ["foo"] }, "roles": { "values": ["analytics"] } }, { "principals": { "type": "NONE" }, "roles": { "values": ["analytics"] } }] } </pre> [2] Principal foo는 analytics 및 ads Role에 대해 Subscribe할 Framework를 등록할 수		

있으며, 별도의 Role은 없음. 나머지 Principal(또는 Principal 없는 Framework)은 어떠한 Role이든 Subscribe할 Framework를 등록할 수 있음

```
{
  "register_frameworks": [
    {
      "principals": {
        "values": ["foo"]
      },
      "roles": {
        "values": ["analytics", "ads"]
      }
    },
    {
      "principals": {
        "values": ["foo"]
      },
      "roles": {
        "type": "NONE"
      }
    }
  ]
}
```

[3] Principal foo는 analytics Role에 대해 Subscribe할 Framework를 등록할 수 있으며, 다른 Role은 없음. 나머지 Principal은 *를 포함하여 어떠한 Role에 대해서도 Subscribe할 Framework를 등록할 수 없음

```
{
  "permissive": false,
  "register_frameworks": [
    {
      "principals": {
        "values": ["foo"]
      },
      "roles": {
        "values": ["analytics"]
      }
    }
  ]
}
```

[4] 다음의 예에서, permissive가 false로 설정되면, Principal은 운영체제의 사용자 guest 또는 bar로서 작업을 수행 할 수 있지만, 나머지 사용자로서 작업을 수행 불가함

```
{
  "permissive": false,
  "run_tasks": [
```

```

    {
      "principals": { "type": "ANY" },
      "users": { "values": ["guest", "bar"] }
    }
  ]
}

```

[5] 다음 예에서, permissive가 false로 설정되면, Principal foo와 bar는 Agent 운영체제 사용자인 alice로서 작업을 수행할 수 있지만, 나머지 사용자는 할 수 없음

```

{
  "permissive": false,
  "run_tasks": [
    {
      "principals": { "values": ["foo", "bar"] },
      "users": { "values": ["alice"] }
    }
  ]
}

```

[6] Principal foo는 Agent 운영체제 사용자인 guest로서 작업을 수행할 수 있으며, 그 외 나머지 사용자로서 작업을 수행할 수 없음. 나머지 Principal(또는 Principal이 없는 Framework)은 어떠한 사용자로도 작업을 수행할 수 있음

```

{
  "run_tasks": [
    {
      "principals": { "values": ["foo"] },
      "users": { "values": ["guest"] }
    },
    {
      "principals": { "values": ["foo"] },
      "users": { "type": "NONE" }
    }
  ]
}

```

[7] Agent 운영체제 사용자인 root로서 작업을 수행할 수 있는 Principal이 없음. 어떠한 Principal(또는 Principal이 없는 Framework)이든 root를 제외한 나머지 사용자로서 작업을 수행할 수 있음

```

{
  "run_tasks": [
    {
      "principals": { "type": "NONE" },
      "users": { "values": ["root"] }
    }
  ]
}

```

```
}
```

[8] 정의된 Rule의 순서는 중요함. 다음의 예에서, ACL은 Framework를 종료하는 누군가로부터 효과적으로 금지하며, 심지어 그 의도는 명백히 Framework를 종료하는 admin만 허용함

```
{
  "teardown_frameworks": [
    {
      "principals": {"type": "admin"},
      "framework_principals": {"type": "ANY"}
    },
    {
      "principals": {"type": "NONE"},
      "framework_principals": {"type": "ANY"}
    }
  ]
}
```

[9] Principal ops는 /teardown HTTP 엔드포인트를 이용하여 어떤 Framework든 종료할 수 있음. 나머지 Principal은 어떠한 프레임워크든 종료할 수 없음

```
{
  "permissive": false,
  "teardown_frameworks": [
    {
      "principals": {
        "values": ["ops"]
      },
      "framework_principals": {
        "type": "ANY"
      }
    }
  ]
}
```

[10] Principal foo는 어떠한 Role이든 리소스를 보존할 수 있으나, 나머지 Principal은 리소스를 보존할 수 없음

```
{
  "permissive": false,
  "reserve_resources": [
    {
      "principals": {
        "values": ["foo"]
      },
      "roles": {
        "type": "ANY"
      }
    }
  ]
}
```

```

    }
  }
]
}

```

[11] Principal foo는 리소스를 보존할 수 없으나, 나머지 Principal(또는 Principal이 없는 Framework)은 어떠한 Role이든 리소스를 보존할 수 있음

```

{
  "reserve_resources": [
    {
      "principals": {
        "values": ["foo"]
      },
      "roles": {
        "type": "NONE"
      }
    }
  ]
}

```

[12] Principal foo는 Role prod와 dev만 리소스를 보존할 수 있으나, 나머지 Principal (또는 Principal이 없는 Framework)은 어떠한 Role도 리소스를 보존할 수 없음

```

{
  "permissive": false,
  "reserve_resources": [
    {
      "principals": {
        "values": ["foo"]
      },
      "roles": {
        "values": ["prod", "dev"]
      }
    }
  ]
}

```

[13] Principal foo는 자신 및 Principal bar에 의해 리소스 보존을 해제할 수 있으나, Principal bar는 자기자신의 리소스만 리소스 보존을 해제할 수 있음. 나머지 Principal은 리소스 보존을 해제할 수 없음

```

{
  "permissive": false,
  "unreserve_resources": [
    {
      "principals": {
        "values": ["foo"]
      }
    }
  ]
}

```

```

    },
    "reserver_principals": {
      "values": ["foo", "bar"]
    }
  },
  {
    "principals": {
      "values": ["bar"]
    },
    "reserver_principals": {
      "values": ["bar"]
    }
  }
]
}

```

[14] Principal foo는 어떠한 Role이든 영구 볼륨을 생성할 수 있으나, 나머지 Principal은 영구 볼륨을 생성할 수 없음

```

{
  "permissive": false,
  "create_volumes": [
    {
      "principals": {
        "values": ["foo"]
      },
      "roles": {
        "type": "ANY"
      }
    }
  ]
}

```

[15] Principal foo는 어떠한 Role이든 영구 볼륨을 생성할 수 없으나, 나머지 Principal은 어떠한 Role이든 영구 볼륨을 생성할 수 있음

```
{
  "create_volumes": [
    {
      "principals": {
        "values": ["foo"]
      },
      "roles": {
        "type": "NONE"
      }
    }
  ]
}
```

[16] Principal foo는 prod 및 dev Role만 영구 볼륨을 생성할 수 있으나, 나머지 Principal은 어떠한 Role이든 영구 볼륨을 생성할 수 없음

```
{
  "permissive": false,
  "create_volumes": [
    {
      "principals": {
        "values": ["foo"]
      },
      "roles": {
        "values": ["prod", "dev"]
      }
    }
  ]
}
```

[17] Principal foo는 자신과 Principal bar에 의해 생성된 볼륨을 삭제할 수 있으나, Principal bar는 자기자신의 볼륨만 삭제할 수 있음. 나머지 Principal은 볼륨 삭제를 할 수 없음

```
{
  "permissive": false,
  "destroy_volumes": [
    {
      "principals": {
        "values": ["foo"]
      },
      "creator_principals": {
        "values": ["foo", "bar"]
      }
    },
    {
      "principals": {
        "values": ["bar"]
      }
    }
  ]
}
```

```

        },
        "creator_principals": {
            "values": ["bar"]
        }
    }
]
}

```

[18] Principal ops는 어떠한 Role이든 Quota 상태를 쿼리할 수 있으나, Principal foo는 foo-role에 대한 Quota 상태만 쿼리할 수 있음. 나머지 Principal은 Quota 상태를 쿼리할 수 없음

```

{
  "permissive": false,
  "get_quotas": [
    {
      "principals": {
        "values": ["ops"]
      },
      "roles": {
        "type": "ANY"
      }
    },
    {
      "principals": {
        "values": ["foo"]
      },
      "roles": {
        "values": ["foo-role"]
      }
    }
  ]
}

```

[19] Principal ops는 어떠한 Role이든 Quota 정보를 갱신(설정 또는 삭제)할 수 있지만, Principal foo는 foo-role에 대한 Quota만 갱신할 수 있음. 나머지 Principal은 Quota를 갱신할 수 없음

```

{
  "permissive": false,
  "update_quotas": [
    {
      "principals": {
        "values": ["ops"]
      },
      "roles": {
        "type": "ANY"
      }
    }
  ]
}

```

```

    }
    },
    {
      "principals": {
        "values": ["foo"]
      },
      "roles": {
        "values": ["foo-role"]
      }
    }
  ]
}

```

[20] Principal ops는 GET 메소드를 이용하여 모든 HTTP 엔드포인트에 도달할 수 있으나, Principal foo는 /logging/toggle 및 /monitor/statistics에서의 HTTP GET 사용만 할 수 있음. 나머지 Principal은 어떠한 엔드포인트든 GET을 이용할 수 없음

```

{
  "permissive": false,
  "get_endpoints": [
    {
      "principals": {
        "values": ["ops"]
      },
      "paths": {
        "type": "ANY"
      }
    },
    {
      "principals": {
        "values": ["foo"]
      },
      "paths": {
        "values": ["/logging/toggle", "/monitor/statistics"]
      }
    }
  ]
}

```

4. Authorizable Action

Action명	설명
register_frameworks	Framework의 (재)등록
run_tasks	Framework에 의해 Task/Executor 실행
teardown_frameworks	Framework 종료
reserve_resources	리소스 보존
unreserve_resources	리소스 보존 해제

	create_volumes	Volumes 생성
	destroy_volumes	Volumes 삭제
	resize_volume	영구 볼륨의 증가 또는 감소
	get_quotas	Quota 상태 쿼리
	update_quotas	Quotas 변경
	view_roles	Roles 및 Weights 쿼리
	get_endpoints	엔드포인트에서 HTTP "GET" 수행
	update_weights	weights 갱신
	view_frameworks	HTTP 엔드포인트 필터링
	view_executors	HTTP 엔드포인트 필터링
	view_tasks	HTTP 엔드포인트 필터링
	access_sandboxes	Task 샌드박스 접근
	access_mesos_logs	Mesos 로그 접근
	register_agents	Agent의 (재)등록
	get_maintenance_schedules	Mesos에 의해 사용된 장비의 유지보수 스케줄 조회
	update_maintenance_schedules	Mesos에 의해 사용된 장비의 유지보수 스케줄 변경
	start_maintenances	장비 상에서 유지보수를 시작. 장비 생성을 생성할 것이며, 해당 장비 에이전트는 사용할 수 없게됨
	stop_maintenances	장비 상에서 유지보수 종료
	get_maintenance_statuses	만약 장비가 유지보수 중인지 아닌지를 조회
	5. Authorizable HTTP 엔드포인트	
	get_endpoints action은 다음과 같은 범위를 지님	
	[1] /files/debug	
	[2] /logging/toggle	
	[3] /metrics/snapshot	
	[4] /slave(id)/containers	
	[5] /slave(id)/monitor/statistics	
진단 방법	[진단기준] - ACL 설정 적용한 경우 양호 - ACL 설정 적용하지 않은 경우 취약	
비고		

14.3. SSL 설정

14.3.1. 환경 설정

분류	SSL 설정	중요도	하
항목명	환경 설정		
항목 설명	기본적으로, Mesos 클러스터를 통해 흐르는 모든 메시지는 평문으로 존재함 공격 성공시, 클러스터에 접근한 누군가가 임의 작업을 가로채고 잠재적으로 통제 가능함		
설정 방법	1. 현재 SSL을 지원하는 libprocess socket interface의 유일한 구현이 존재함 이 구현은 libevent를 사용함. 구체적으로, openssl을 랩핑하는 libevent-openssl 라이브러리 상에 의존함 2. 소스로부터 Mesos 0.23.0 빌드하기 전에, 필요로 하는 의존성 패키지를 설치해야 하며, 다음과 같이 SSL을 활성화하기 위한 configure 명령어 라인을 변경할 수 있음 <pre>../configure --enable-libevent --enable-ssl</pre> ※ 의존성 패키지 LIBPROCESS_SSL_ECDH_CURVE=(auto list of curves separated by ':') [default=auto] 선호하는 순서로, ECDHE 기반 암호화 알고리즘 사용해야 하는 타원곡선의 목록을 나타냄 사용할 수 있는 알고리즘은 사용하는 OpenSSL 버전에 따라 다름 기본 값 auto는 OpenSSL이 자동으로 타원곡선을 선택하기 위해 허용함 OpenSSL 버전 1.0.2 이전 버전은 유일한 타원곡선만 허용하므로, 그 경우에 기본 값은 prime256v1임 ※ [의존성 패키지1] libevent libevent로부터 OpenSSL 지원할 필요가 있음. 제한할 libevent 버전은 2.0.22-stable임. 새로운 libevent 버전이 출시될 경우, OpenSSL과의 호환성 여부를 검토해야 함 ※ [의존성 패키지2] OpenSSL 커뮤니티에 의해 유지되는 다양한 Branch의 OpenSSL이 존재함. 방심없는 보안이 필요하기 때문에, 최신 버전의 OpenSSL의 릴리즈 노트 내용을 검토하여 기업 내부의 적용 유무를 결정해야 함 Mesos는 특정 OpenSSL 버전에 대한 깊은 의존성이 없으므로, 기업이 보안에 관한 의사결정의 여지가 있음 event2 및 openssl 헤더가 Mesos 빌드를 위해 사용할 수 있도록 보장해주야 함		
진단 방법	[진단기준] - SSL 설정 적용한 경우 양호 - SSL 설정 적용하지 않은 경우 취약		
비고			

14.3.2. 운영 설정

분류	SSL 설정	중요도	하
항목명	운영 설정		
항목 설명	최초 Mesos 설치를 성공적으로 완료하면, 환경변수를 이용하여 SSL 설정을 해야 함 만약 안전하게 SSL 설정을 하지 않을 경우, 취약한 SSL 버전, 암호화 알고리즘 사용으로 2차 공격 발생 가능성이 존재함		
설정 방법	1. SSL 운영 설정시 아래의 환경변수를 적용해야 함		
	환경변수명	설명	
	LIBPROCESS_SSL_ENABLED=(false 0,true 1) [default=false 0]	SSL 활성화 또는 비활성화. SSL 활성화시, LIBPROCESS_SSL_CERT_FILE 및 LIBPROCESS_SSL_KEY_FILE이 반드시 필요함	
	LIBPROCESS_SSL_SUPPORT_DOWNGRADE=(false 0,true 1) [default=false 0]	비 SSL 연결 수립 여부를 통제함. 만약 비 SSL 연결 수립을 활성화할 경우, HTTP 평문 통신이 가능함	
	LIBPROCESS_SSL_KEY_FILE=(path to key)	OpenSSL에 의해 사용된 개인키(비밀키)의 위치	
	LIBPROCESS_SSL_CERT_FILE=(path to certificate)	제공될 인증서의 위치	
	LIBPROCESS_SSL_VERIFY_CERT=(false 0,true 1) [default=false 0]	제공될 때 인증서의 검증 유무를 통제함. 만약 LIBPROCESS_SSL_REQUIRE_CERT가 참일 경우, LIBPROCESS_SSL_VERIFY_CERT는 오버라이딩되며 모든 인증서는 검증/필수 적용	
	LIBPROCESS_SSL_REQUIRE_CERT=(false 0,true 1) [default=false 0]	해당 인증서는 클라이언트에 연결하면서 제공해야 함을 강제함 이는 (엔드포인트 관여하는 모든 도구를 포함하여) 모든 연결이 연결 수립을 위해 유효한 인증서를 제공해야 함을 의미함	
	LIBPROCESS_SSL_VERIFY_DEPTH=(N) [default=4]	인증서 검증을 위해 사용되는 최대 깊이(Depth)를 나타냄	
	LIBPROCESS_SSL_VERIFY_IPADD=(false 0,true 1) [default=false 0]	인증서의 Subject Alternative Name(SAN) 확장모듈에서의 IP 주소 검증을 활성화함. 만약 참일 경우, 접속자 인증서 검증시 IP 주소가 추가로 사용됨 IP 주소와 마찬가지로 호스트명도 사용 가능할 경우, IP와 호스트명의 일치여부를 검증함	
	LIBPROCESS_SSL_CA_DIR=(path to CA directory)	해당 디렉토리는 인증서 Authority/Authorities를 찾기 위해 사용됨. 인증서 인가 제한을 원할 경우, LIBPROCESS_SSL_CA_DIR 또는 LIBPROCESS_SSL_CA_FILE을 지정할 수 있음	
	LIBPROCESS_SSL_CA_FILE=(path to CA file)	해당 파일은 인증서 Authority를 찾기 위해 사용됨 인증서 인가 제한을 원할 경우, LIBPROCESS_SSL_CA_DIR 또는 LIBPROCESS_SSL_CA_FILE 지정 가능함	
	LIBPROCESS_SSL_CIPHERS=(accepted ciphers separated by ':')	콜론(:)으로 분리된 암호화 알고리즘 목록을 나타냄 OpenSSL을 위해 승인된 암호화 알고리즘을 제한 또는	

	[default=AES128-SHA:AES256-SHA:RC4-SHA:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA:DHE-RSA-AES256-SHA:DHE-DSS-AES256-SHA]	개방하기 원할 경우에 사용함
	LIBPROCESS_SSL_ENABLE_SSL_V3=(false 0,true 1) [default=false 0] LIBPROCESS_SSL_ENABLE_TLS_V1_0=(false 0,true 1) [default=false 0] LIBPROCESS_SSL_ENABLE_TLS_V1_1=(false 0,true 1) [default=false 0] LIBPROCESS_SSL_ENABLE_TLS_V1_2=(false 0,true 1) [default=true 1]	좌측 스위치는 특정 프로토콜을 활성화/비활성화함 기본적으로 TLS V1.2만 활성화되며, SSL v2는 항상 비활성화함 SSL v2 활성화하는 스위치는 없음
	--http_authenticators	사용할 HTTP Authenticator 모듈을 지정함. 기본값으로 basic (기본 HTTP 인증)이지만, 부가 모듈도 --modules 옵션을 이용하여 추가할 수 있음
	--credentials	승인된 자격증명(Credential)의 목록을 포함하는 텍스트 파일에 대한 경로. 이 플래그는 사용하는 Authenticator에 따라 옵션으로 사용함
진단 방법	[진단기준] - 안전한 SSL 운영을 위한 환경변수 설정 적용한 경우 양호 - 안전한 SSL 운영을 위한 환경변수 설정 적용하지 않은 경우 취약	
비고		

14.3.3. WebUI 설정

분류	SSL 설정	중요도	하
항목명	WebUI 설정		
항목 설명	서드파티 Framework를 포함한 Mesos WebUI가 HTTPS가 적용되었지 않으면, 평문 통신으로 인한 중요정보 탈취 가능 및 위·변조 가능성이 존재함		
설정 방법	1. 기본적으로 Mesos WebUI는 상대 링크를 사용함 이 링크 중 일부는 Master와 Agent에 의해 제공될 엔드포인트 사이에 변화됨 WebUI는 대상 엔드포인트가 SSL 활성화된 바이너리 파일의 제공 유무에 따라 http 또는 https로 변경하기 위한 충분한 정보를 가지지 않음 이는 클러스터가 SSL과 비 SSL 사이의 상태 변화로 WebUI의 특정 링크 깨짐을 유발할 수 있음 이런 엔드포인트를 관여하는 어떠한 도구로든 올바른 프로토콜을 이용토록 엔드포인트에 관여하여 접근할 수 있도록 해주거나, LIBPROCESS_SSL_SUPPORT_DOWNGRADE 옵션을 참으로 설정해야 함 ※ <u>인증서</u> 대부분의 브라우저는 다른 인증서를 사용하면서 제공되는 페이지 사이에 변화를 방지하는 보호장치가 내장됨 이러한 이유로, 다중 호스트명을 아우르는 공통 인증서를 이용하여 Master, Agent 엔드포인트 모두 제공하도록 선택해야할 수도 있음 만약 이를 준수하지 않을 경우, Agent 샌드박스과 같은 특정 링크가 안전하지 않은 인증서 사이의 변화의 차이로 인해 웹 브라우저 처리시 링크가 깨질 수 있음 ※ <u>주의</u> WebUI를 포함한 Framework는 별도로 HTTPS 지원을 추가할 필요가 있음		
진단 방법	[진단기준] - WebUI에 SSL 설정 적용한 경우 양호 - WebUI에 SSL 설정 적용하지 않은 경우 취약		
비고			

15. OAuth

15.1. 일반 설정

15.1.1. OAuth 요청의 기밀성 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하																
항목명	OAuth 요청의 기밀성 설정																		
항목 설명	이 방법은 클라이언트로부터 인증서버 또는 리소스 서버까지 전송한 모든 요청에 적용 가능함. OAuth가 요청의 무결성을 검증하는 메커니즘을 제공하지만, 요청의 기밀성 보장을 제공하지 않음. 만약 더 나은 예방책을 가지고 있지 않으면, 스니퍼는 콘텐츠를 요청하기 위한 전체 권한을 가질 것이며, 가로채기를 시작할 수 있고, 콘텐츠에 대한 요청을 이용하여 리플레이 공격이 가능할 수 있음 ※ OAuth 요청의 기밀성 미설정시 아래의 위험 발생이 가능함 ○ 토큰의 엔드포인트 또는 리소스 서버의 엔드포인트에서 획득한 접근토큰의 리플레이 공격 ○ 토큰 엔드포인트에서 획득한 갱신 토큰의 리플레이 공격 ○ 토큰의 엔드포인트에서 획득한 인증코드의 리플레이 공격 (리다이렉트?) ○ 사용자 패스워드 및 클라이언트 Secret의 리플레이 공격																		
설정 방법	1. TLS 전송계층 메커니즘을 사용하도록 설정 (최소: TLS 1.1 이상, 권장 TLS 1.2 이상) (1) Java (Oracle) : 운영체제와 상관없이 최신 버전과 호환됨 <table><tr><th>플랫폼 또는 라이브러리</th><th>호환성 참고</th></tr><tr><td>Java 8(1.8) 이상</td><td>기본적으로 TLS 1.1 이상과 호환함</td></tr><tr><td>Java 7(1.7)</td><td>HttpsURLConnection에 대하여 https.protocols Java 시스템 속성을 사용하여 TLS 1.1 및 TLS 1.2를 활성화함 비 HttpsURLConnection 연결에서 TLS 1.1 및 TLS 1.2를 활성화하려면 응용 프로그램 소스 코드 내부에 생성된 SSLSocket 및 SSLEngine 인스턴스에서 활성화된 프로토콜을 설정함</td></tr><tr><td>Java 6(1.6) 업데이트 111 이상</td><td>HttpsURLConnection에 대하여 https.protocols Java 시스템 속성을 사용하여 TLS 1.1을 활성화함 비 HttpsURLConnection 연결에서 TLS 1.1을 활성화하려면 응용 프로그램 소스 코드 내부에 생성된 SSLSocket 및 SSLEngine 인스턴스에서 활성화된 프로토콜을 설정함 이 Java 6 업데이트 및 신규 업데이트는 공개적으로 사용할 수 없으며, Oracle과 Java 6에 대한 지원 계약을 맺어야 함</td></tr><tr><td>Java 6(1.6) 이하(공개 버전)</td><td>TLS 1.1 이상의 암호화를 지원하지 않음</td></tr></table> (2) Java (IBM) : 운영체제와 상관없이 최신 버전과 호환됨 <table><tr><th>플랫폼 또는 라이브러리</th><th>호환성 참고</th></tr><tr><td>Java 8</td><td>기본적으로 TLS 1.1 이상과 호환함 애플리케이션이나 라이브러리가 SSLContext.getInstance("TLS") 사용하는 경우 com.ibm.jsse2.overrideDefaultTLS=true를 설정해야 할 수도 있음</td></tr><tr><td>Java 7 이상, Java 6.0.1</td><td>IBM's documentation에서 권장한 대로 HttpsURLConnection</td></tr></table>			플랫폼 또는 라이브러리	호환성 참고	Java 8(1.8) 이상	기본적으로 TLS 1.1 이상과 호환함	Java 7(1.7)	HttpsURLConnection에 대하여 https.protocols Java 시스템 속성을 사용하여 TLS 1.1 및 TLS 1.2를 활성화함 비 HttpsURLConnection 연결에서 TLS 1.1 및 TLS 1.2를 활성화하려면 응용 프로그램 소스 코드 내부에 생성된 SSLSocket 및 SSLEngine 인스턴스에서 활성화된 프로토콜을 설정함	Java 6(1.6) 업데이트 111 이상	HttpsURLConnection에 대하여 https.protocols Java 시스템 속성을 사용하여 TLS 1.1을 활성화함 비 HttpsURLConnection 연결에서 TLS 1.1을 활성화하려면 응용 프로그램 소스 코드 내부에 생성된 SSLSocket 및 SSLEngine 인스턴스에서 활성화된 프로토콜을 설정함 이 Java 6 업데이트 및 신규 업데이트는 공개적으로 사용할 수 없으며, Oracle과 Java 6에 대한 지원 계약을 맺어야 함	Java 6(1.6) 이하(공개 버전)	TLS 1.1 이상의 암호화를 지원하지 않음	플랫폼 또는 라이브러리	호환성 참고	Java 8	기본적으로 TLS 1.1 이상과 호환함 애플리케이션이나 라이브러리가 SSLContext.getInstance("TLS") 사용하는 경우 com.ibm.jsse2.overrideDefaultTLS=true를 설정해야 할 수도 있음	Java 7 이상, Java 6.0.1	IBM's documentation에서 권장한 대로 HttpsURLConnection
플랫폼 또는 라이브러리	호환성 참고																		
Java 8(1.8) 이상	기본적으로 TLS 1.1 이상과 호환함																		
Java 7(1.7)	HttpsURLConnection에 대하여 https.protocols Java 시스템 속성을 사용하여 TLS 1.1 및 TLS 1.2를 활성화함 비 HttpsURLConnection 연결에서 TLS 1.1 및 TLS 1.2를 활성화하려면 응용 프로그램 소스 코드 내부에 생성된 SSLSocket 및 SSLEngine 인스턴스에서 활성화된 프로토콜을 설정함																		
Java 6(1.6) 업데이트 111 이상	HttpsURLConnection에 대하여 https.protocols Java 시스템 속성을 사용하여 TLS 1.1을 활성화함 비 HttpsURLConnection 연결에서 TLS 1.1을 활성화하려면 응용 프로그램 소스 코드 내부에 생성된 SSLSocket 및 SSLEngine 인스턴스에서 활성화된 프로토콜을 설정함 이 Java 6 업데이트 및 신규 업데이트는 공개적으로 사용할 수 없으며, Oracle과 Java 6에 대한 지원 계약을 맺어야 함																		
Java 6(1.6) 이하(공개 버전)	TLS 1.1 이상의 암호화를 지원하지 않음																		
플랫폼 또는 라이브러리	호환성 참고																		
Java 8	기본적으로 TLS 1.1 이상과 호환함 애플리케이션이나 라이브러리가 SSLContext.getInstance("TLS") 사용하는 경우 com.ibm.jsse2.overrideDefaultTLS=true를 설정해야 할 수도 있음																		
Java 7 이상, Java 6.0.1	IBM's documentation에서 권장한 대로 HttpsURLConnection																		

서비스 새로 고침 1(J9 VM2.6 이상), Java 6 서비스 새로 고침 10 이상	https.protocols Java 시스템 속성을 사용하여 TLS 1.2를 사용하고 SSL소켓, SSL엔진 연결용 com.ibm.jsse2.overrideDefaultProtocol Java 시스템 속성을 사용하면 com.ibm.jsse2.overrideDefaultTLS=true를 설정해야 할 수도 있음
--	--

(3) .NET : TLS 1.1 또는 TLS 1.2를 지원하는 운영체제에서 실행시 최신 버전과 호환함

플랫폼 또는 라이브러리	호환성 참고
.NET 4.6 이상	기본적으로 TLS 1.1 이상과 호환함
.NET 4.5 ~ 4.5.2	.NET 4.5, 4.5.1 및 4.5.2는 기본적으로 TLS 1.1 및 TLS 1.2를 활성화하지 않음 아래에 명시된 두 가지 방법을 사용하여 TLS 1.1 및 TLS 1.2를 활성화할 수 있음 옵션 1: .NET 응용 프로그램의 경우 System.Net.ServicePointManager.SecurityProtocol에서 SecurityProtocolType.Tls12 및 SecurityProtocolType.Tls11을 활성화하도록 설정하면 소프트웨어 코드에서 TLS 1.1 및 TLS 1.2를 직접 활성화함 다음의 C# 코드는 예시임 <pre>System.Net.ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12 SecurityProtocolType.Tls11 SecurityProtocolType.Tls;</pre> 옵션 2: 다음의 두 가지 레지스트리 키에서 SchUseStrongCrypto DWORD 값을 1로 설정하여 소스 코드를 수정하지 않고 기본 값으로 TLS 1.2를 활성화할 수 있음 (존재하지 않을 경우 생성됨, "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NETFramework\v4.0.30319" 및 "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\NETFramework\v4.0.30319") 해당 레지스트리 키의 버전 번호는 4.0.30319이지만 .NET 4.5, 4.5.1 및 4.5.2 프레임워크 역시 이러한 값을 사용함 그러나 레지스트리 키는 시스템에 설치된 모든 .NET 4.0, 4.5, 4.5.1 및 4.5.2 응용 프로그램에서 기본값으로 TLS 1.2를 활성화함 따라서 프로덕션 서버에 배포하기 전에 변경 사항을 테스트하는 것이 좋음
.NET 4.0	.NET 4.0은 기본값으로 TLS 1.2를 활성화하지 않음 기본값으로 TLS 1.2를 사용하려면 .NET Framework 4.5 또는 최신 버전을 설치하고 다음 두 레지스트리 키에서 SchUseStrongCrypto DWORD 값을 1로 설정하여 기본값으로 TLS 1.2를 활성화할 수 있음 레지스트리 키 미존재시 생성함:"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NETFramework\v4.0.30319" 및

	"HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\ .NETFramework\v4.0.30319" 그러나 레지스트리 키는 시스템에 설치된 모든 .NET 4.0, 4.5, 4.5.1, 4.5.2 응용 프로그램에서 기본값으로 TLS 1.2를 활성화할 수 있음 프로덕션 서버에 배포하기 전에 변경 사항을 테스트하는 것이 좋음
.NET 3.5 이하	TLS 1.1 이상의 암호화를 지원하지 않음

(4) Python : TLS 1.1 또는 TLS 1.2를 지원하는 운영체제에서 실행시 최신 버전과 호환함

플랫폼 또는 라이브러리	호환성 참고
Python 2.7.9 이상	기본적으로 TLS 1.1 이상과 호환함
Python 2.7.8 이하	TLS 1.1 이상의 암호화를 지원하지 않음

(5) Ruby : OpenSSL 1.0.1 이상 버전과 연결되면 최신 버전과 호환됨

플랫폼 또는 라이브러리	호환성 참고
Ruby 2.0.0	OpenSSL 1.0.1 이상 버전과 함께 사용할 경우 TLS 1.2는 기본값으로 활성화함 :TLsv1_2(기본) 또는 :TLsv1_1 기호와 함께 SSLContext's ssl_version을 사용하면 TLS 1.0 이하 버전이 비활성화되도록 할 수 있음
Ruby 1.9.3 이하	:TLsv1_2 기호는 1.9.3 이하 버전에서 존재하지 않지만 Ruby를 패치하여 기호를 추가하고 Ruby를 OpenSSL 1.0.1 이상 버전과 컴파일할 수 있음

(6) Microsoft WinINet : 최신 버전과 호환됨

플랫폼 또는 라이브러리	호환성 참고
(서버) Windows Server 2012 R2 이상 (일반) Windows 8.1 이상	기본적으로 TLS 1.1 이상과 호환함
(서버) Windows Server 2008 R2 ~ 2012 (일반) Windows 7 및 8	Internet Explorer 11이 설치된 경우 기본값으로 호환함 Internet Explorer 8, 9 또는 10이 설치된 경우 사용자 또는 관리자가 호환성을 확인하고 TLS 1.1, TLS 1.2를 활성화해야 함
(서버) Windows Server 2008 이하 (일반) Windows Vista 이하	TLS 1.1 이상의 암호화를 지원하지 않음

(7) Microsoft Secure Channel : 최신 버전과 호환됨

플랫폼 또는 라이브러리	호환성 참고
(서버) Windows Server 2012	기본적으로 TLS 1.1 이상과 호환함

R2 이상 (일반) Windows 8.1 이상	
(서버) Windows Server 2012 (일반) Windows 8	TLS 1.1 및 TLS 1.2는 기본값으로 비활성화되어 있지만 애플리케이션에서 활성화할 경우 사용할 수 있음 TLS 1.1 및 TLS 1.2는 레지스트리 내부에서 기본값으로 활성화할 수 있음
(서버) Windows Server 2008 R2 (일반) Windows 7	TLS 1.1 및 TLS 1.2는 기본값으로 비활성화되어 있지만 Internet Explorer 11이 설치된 경우 클라이언트 모드에서 기본값으로 호환됨 Internet Explorer 11이 설치되지 않은 경우 레지스트리 내부에서 TLS 1.1 및 TLS 1.2를 기본값으로 활성화할 수 있음
(서버) Windows Server 2008 이하 (일반) Windows Vista 이하	TLS 1.1 이상의 암호화를 지원하지 않음

(8) Microsoft WinHTTP 및 Webio : 최신 버전과 호환됨

플랫폼 또는 라이브러리	호환성 참고
(서버) Windows Server 2012 R2 이상 (일반) Windows 8.1 이상	기본적으로 TLS 1.1 및 TLS 1.2와 호환함
(서버) Windows Server 2008 R2 ~ 2012 (일반) Windows 7 SP1	KB3140245가 적용됨에 따라 Webio는 기본적으로 호환되며, WinHTTP는 레지스트리 설정을 통해 TLS 1.1 및 TLS 1.2를 활성화하도록 구성할 수 있음
(서버) Windows Server 2008 이하 (일반) Windows Vista 이하	TLS 1.1 이상의 암호화를 지원하지 않음

(9) OpenSSL : 운영체제와 상관없이 최신 버전과 호환함

플랫폼 또는 라이브러리	호환성 참고
OpenSSL 1.0.1 이상	기본적으로 TLS 1.1 이상과 호환함
OpenSSL 1.0.0 이하	TLS 1.1 이상의 암호화를 지원하지 않음

(10) Mozilla NSS : 운영체제와 상관없이 최신 버전과 호환함

플랫폼 또는 라이브러리	호환성 참고
3.15.1 이상	기본적으로 TLS 1.1 이상과 호환함
3.14 ~ 3.15	TLS 1.1과 호환하지만 TLS 1.2와는 호환하지 않음

	<table border="1" data-bbox="280 201 1414 244"> <tr> <td data-bbox="280 201 608 244">3.13.6 이하</td><td data-bbox="608 201 1414 244">TLS 1.1 이상의 암호화를 지원하지 않음</td></tr> </table> ※ <u>TLS 적용 Best Practice 사례</u> - LinkedIn : TLS 1.1부터 지원함 (2017.10.10부, 보안 상의 이유로 TLS 1.0 지원 중단함) - Salesforce : TLS 1.1부터 지원함 (2018.04.23부, 보안 상의 이유로 TLS 1.0 지원 중단함) ※ <u>금융권 관련 참고사항</u> PCI DSS 3.1에 따르면, 2018년 6월 30일 이후에는 최소 TLS 1.1 이상을 적용해야 함 [예외] 공개되지 않은 위험을 포함하여 SSL/TLS 1.0을 위해 알려진 모든 취약점에 대해 민감한 사항이 아님을 검증받을 수 있는 POI(Point Of Interaction) 터미널 내의 SSL/TLS 1.0 사용은 예외적으로 DSS v3.1에서 존재하는 언어와 일치하는 2018년 6월 이후에도 사용할 수 있음 [출처] Bulletin on Migrating from SSL and Early TLS (PCI Security Standards Council, 2015) https://www.pcisecuritystandards.org/pdfs/Migrating_from_SSL_and_Early_TLS_-_v12.pdf 2. 가능하다면 IPsec VPN(RFC 4301) 등과 같은 가상사설망(VPN)을 사용하도록 설정	3.13.6 이하	TLS 1.1 이상의 암호화를 지원하지 않음
3.13.6 이하	TLS 1.1 이상의 암호화를 지원하지 않음		
진단 방법	[진단기준] - TLS 1.0 이상 프로토콜 사용하고 HTTPS를 이용한 암호화 통신 적용한 경우 양호 - SSL 3.0 이하 프로토콜 사용하거나 HTTPS를 이용한 암호화 통신 적용하지 않은 경우 취약		
비고			

15.1.2. 서버 인증 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	서버 인증 설정		
항목 설명	만약 서버가 바인딩 유효성을 입증하는 데 실패한 경우, 통신은 Man-In-The-Middle(MITM) 공격으로 여겨질 수 있음 ※ OAuth 요청의 기밀성 미설정시 아래의 위협 발생이 가능함 ○ 토큰의 엔드포인트 또는 리소스 서버의 엔드포인트에서 획득한 접근토큰의 리플레이 공격 ○ 토큰 엔드포인트에서 획득한 갱신 토큰의 리플레이 공격 ○ 토큰의 엔드포인트에서 획득한 인증코드의 리플레이 공격 (리다이렉트?) ○ 사용자 패스워드 및 클라이언트 Secret의 리플레이 공격		
설정 방법	1. 클라이언트는 서버와 서버의 도메인명과의 바인딩 유효성을 검증해야 함 2. 인증 목적으로 클라이언트를 신뢰하는 인증 기관에 따라 다르므로, 클라이언트는 주의 깊게 신뢰할 수 있는 CA를 선택하여 변조로부터 신뢰할 수 있는 CA 인증서를 위한 저장소를 보호해야 함		
진단 방법	[진단기준] - 공인 인증기관(CA)이 발급하지 않은 인증서로 정상적인 OAuth 인증이 불가능한 경우 양호 - 공인 인증기관(CA)이 발급하지 않은 인증서로 정상적인 OAuth 인증이 가능한 경우 취약 1) 실제 서비스 중인 도메인 URL과 CA 인증서의 도메인 URL과 불일치한 인증서 사용 2) 실제 서비스 중인 인증서의 발급기관명과 불일치한 인증서 사용		
비고			

15.1.3. 리소스 소유자(Resource Owner)에게 알린 내용 유지 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	리소스 소유자(Resource Owner)에게 알린 내용 유지 설정		
항목 설명	Resource Owner에 대한 투명성은 OAuth 프로토콜의 핵심 요소임. 사용자는 항상 인증 프로세스의 통제가 되어야 하며, 현명한 결정을 위한 필요한 정보를 얻어야 함. 게다가, 사용자 개입은 더 나은 보안 대책수단임. 사용자는 인증서버보다는 더 좋은 특정 공격 유형을 인식할 수 있음. 정보는 인증 프로세스동안 제공하거나 교환할 수 있으며, 인증 프로세스 이후에는, 항상 사용자가 다음과 같은 기법을 이용하여 알린 내용을 받기를 원함		
설정 방법	1. 침해사고 발생시 공격 유형을 인식하기 위해 아래의 절차를 구현 및 컴플라이언스를 준수하여 정해진 기간에 보관을 해야 함 ○ 사용자 약관 서식 ○ 알림 메시지 (예, 이메일, SMS 등) 알림은 피싱 벡터가 될 수 있음. 피싱 메시지처럼 보이는 메시지가 알림 메시지에서부터 전파될 수 없도록 해야 함 ○ 액티비티/이벤트 로그 ○ 사용자 self-care 애플리케이션 또는 포털 ※ 사용자 self-care 애플리케이션 또는 포털 연산이 모듈화되어 쉽게 순서를 재배치할 수 있거나 인증 흐름에 적합하게 사용자 정의할 수 있음을 의미하는 내장된 인증 서비스를 이용한 연산이 내장된 애플리케이션 또는 포털		
진단 방법	[진단기준] - 사용자에게 의해 작성 및 생성된 사용자 약관 서식, 알림 메시지, 액티비티/이벤트 로그, 사용자 self-care 애플리케이션 또는 포털에 관련된 내용이 보관되어 있는 경우 양호 - 사용자에게 의해 작성 및 생성된 사용자 약관 서식, 알림 메시지, 액티비티/이벤트 로그, 사용자 self-care 애플리케이션 또는 포털에 관련된 내용이 보관되지 않은 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.1.4. 자격증명 - 표준 시스템 보안 수단 강제 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	자격증명 - 표준 시스템 보안 수단 강제 설정		
항목 설명	표준 시스템 보안 수단 강제 설정을 하지 않을 경우 공격자에 의해 민감한 환경설정 파일 및 DB 탈취가 가능함		
설정 방법	1. 안전하게 서버 소프트웨어를 설치해야 함 ○ 전용 호스트 또는 만약 가상화 기술 활용이 가능한 전용 게스트 OS 상에 서버 SW를 설치해야 함 ○ 서버 소프트웨어 내 알려진 취약점 수정을 위해 패치 또는 업그레이드를 적용해야 함 ○ 만약 적용할 수 있으면, 전용 물리 디스크 또는 (OS와 서버 애플리케이션을 분리한) 논리 파티션을 생성해야 함 ○ 서버 애플리케이션에 의해 설치되었지만 불필요한 모든 서비스에 대해 삭제 또는 비활성화를 해야 함 (예: Gopher, FTP, HTTP, 원격관리 등) ○ 서버 설치시 생성된 모든 불필요한 기본 사용자 계정을 삭제 또는 비활성화를 해야 함 ○ 서버로부터 모든 벤더사의 문서를 삭제해야 함 ○ 서버로부터 샘플 콘텐츠, 스크립트 및 실행파일 코드를 포함한 모든 예제 또는 테스트 파일을 삭제해야 함 ○ 모든 불필요한 컴파일러를 삭제해야 함 ○ 서버에 대해 적절한 보안 템플릿 또는 보안 강화 스크립트를 적용해야 함 ○ 외부망 서버에 대해, 만약 가능하면, 서비스 배너가 OS 종류 및 버전을 노출하지 않도록 설정을 다시 해야 함 ○ 배너를 지원하는 모든 서비스에 대해 경고 배너를 설정해야 함 ○ 만약 가능하면, 필요한 TCP 및 UDP 포트만 클라이언트 연결에 대한 Listen을 위한 각 네트워크 서비스에 대해 설정을 해야 함 2. 접근통제를 설정해야 함 ○ 서버 애플리케이션의 계산할 리소스의 부분집합으로 접근 제한을 해야 함 ○ 더 상세한 수준의 접근통제가 필요한 서버에 의해 강제하는 부가적인 접근통제를 통해 사용자 접근을 제한해야 함 ○ 접근통제를 해야 하는 일반적인 파일은 다음과 같음 1) 애플리케이션 소프트웨어 및 환경설정 파일 2) 보안 메커니즘과 직접 관련된 파일 2-1) 인증에 사용하는 패스워드 해시 파일 및 기타 파일 2-2) 접근을 통제하기 위해 사용하는 인증 정보를 포함하는 파일 2-3) 기밀성, 무결성 및 부인방지 서비스에 사용하는 암호 키 데이터 3) 서버 로그 및 시스템 감사 파일 4) 시스템 소프트웨어 및 환경설정 파일 5) 서버 콘텐츠 파일 ○ 서비스 프로세스에 의해 접근 가능한 파일을 제한하기 위해 서버 호스트 OS 접근통제를 사용해야 함 1) 서비스 프로세스는 엄격하게 제한된 권한 집합을 가진 사용자로서 실행하도록 구성해야 함 (예: root, administrator 또는 동일한 권한을 수행하지 않아야 함) 2) 만약 필요하다면 서비스 프로세스는 서버 콘텐츠 파일 및 디렉토리에 대한 기록만 할 수 있도록 해야 함		

- 3) (만약 가능하다면) 서버 소프트웨어에 의해 생성된 임시파일은 규격에 맞고 적절하게 보호되는 하위 디렉토리로 제한해야 하며, (만약 가능하다면) 임시파일에 대한 접근은 파일을 생성했던 서버 프로세스로 제한되어야 함

3. 서버 리소스 제약조건을 적용해야 함

- OS 및 서버 소프트웨어와 다른 보조기억장치 또는 논리 파티션에서 서버 콘텐츠를 설치해야 함
- 만약 서버가 업로드를 허용한다면, 업로드 전용 보조기억장치 공간의 양을 제한하여 업로드 파일이 위치해야 하며, 이와 동일하게, 업로드는 보조기억장치 용량 제한을 초과할 수 없도록 더 강력한 보증을 제공한 분리 파티션에 업로드 파일이 위치해야 함
- 만약 업로드가 서버에 허용되면, 업로드 파일은 일부 자동 또는 수동 검토 과정이 업로드 파일을 가려내는 데 사용한 후, 서버에 의해 읽을 수 없도록 보장해야 함. 이 방법은 서버가 악성코드 또는 불법 소프트웨어, 공격 도구, 포르노 등의 트래픽을 전파하기 위해 사용되는 것을 방지함. 또한 많은 대용량 파일을 업로드하여 개입한 DoS 공격의 잠재적인 영향을 제한할 수 있도록 각 업로드된 파일의 용량을 제한하는 것이 가능함
- 로그 파일이 적절한 크기로 저장되도록 보장해야 함. 동일하게, 로그 파일은 분리된 파티션에 저장해야 함. 만약 공격이 로그 파일의 크기가 수용할 수 있는 제한을 넘어서면서 증가하도록 할 경우, 물리 파티션은 서버가 그러한 상황을 적절하게 처리하기 위해 충분한 리소스를 가지도록 보장하는 것을 도와줌
- 서버가 허용하는 서버 프로세스 및/또는 네트워크 연결 최대 개수를 설정해야 함

4. 안전한 알고리즘을 이용한 인증 및 암호화 기술을 적용해야 함

구분	공공기관	민간부문(법인·단체·개인)
대칭키 암호 알고리즘	SEED, LEA, HIGHT, ARIA-128/192/256	SEED ARIA-128/192/256 AES-128/192/256 Blowfish Camelia-128/192/256 MISTY1 KASUMI 등
공개키 암호 알고리즘 (메시지 암호·복호화)	RSAES-OAEP	RSA RSAES-OAEP RSAES-PKCS1 등
일방향 암호 알고리즘	SHA-224/256/384/512	SHA-224/256/384/512 Whirlpool 등

	※ 출처 : 개인정보의 암호화 조치 안내서 (행정자치부, KISA, 2017.01)
진단 방법	[진단기준] - 안전하게 서버 소프트웨어를 설치, 접근통제 설정, 서버 리소스 제약조건 적용 및 안전한 알고리즘을 이용한 인증 및 암호화 기술 적용을 한 경우 양호 - 취약하게 서버 소프트웨어를 설치, 접근통제 설정 미적용, 서버 리소스 제약조건 미적용 및 안전한 알고리즘을 이용한 인증 또는 암호화 기술 미적용인 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

15.1.5. 자격증명 - 표준 SQL 인젝션 방지 강제 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	자격증명 - 표준 SQL 인젝션 방지 강제 설정		
항목 설명	만약 클라이언트 식별자 또는 다른 인증 구성요소가 SQL DB에 대해 쿼리 또는 비교가 된다면, 만약 수신된 파라미터가 DB로 제출 전에 유효성을 입증하지 않으면 발생할 인젝션 공격이 가능하게 될 수도 있음		
설정 방법	1. 서버 코드는 가능한 '표면' 공격을 감소하기 위해 가능한 최소한의 DB 권한을 이용하도록 보장해야 함 2. (이진/문자) 입력 값 조합을 이용한 동적 SQL 공격을 회피해야 함. 만약 가능할 경우, 정적 SQL을 사용해야 함 3. 동적 SQL을 이용할 때, 인자 바인딩을 이용하여 쿼리를 파라미터화해야 하며, 인자 바인딩은 SQL 인젝션의 가능성을 제거해야 함 4. 입력 값 필터링을 해야함. 예를 들면, 만약 식별자가 알려진 포맷을 가진다면, 사용자 입력 값이 식별자 문법 규칙과 일치하도록 보장해야 함		
진단 방법	[진단기준] - SQL 인젝션 공격이 실패한 경우 양호 - SQL 인젝션 공격이 성공한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.1.6. 자격증명 - 자격증명 평문 저장 방지 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하												
항목명	자격증명 - 자격증명 평문 저장 방지 설정														
항목 설명	만약 인증서버가 평문으로 자격증명을 저장할 경우, 자격증명의 유출 및 위·변조가 가능함														
설정 방법	1. 인증서버는 평문으로 자격증명을 저장해서는 안됨. 일반적인 접근은 대신 해시로 저장하거나 자격증명을 암호화를 함. (사용자 패스워드이기 때문에) 만약 자격증명이 실용적인 엔트로피 수준이 누락된 경우, 부가적인 Salt는 오프라인 기반 Dictionary 공격을 더욱 어렵게 만들기 위해 저장소를 견고하게 만들게 됨														
	※ 주의 일부 인증 프로토콜은 평문으로 Secret을 접근해야 하는 인증서버가 필요함. 이러한 프로토콜은 만약 서버에서만 해시로 접근해야 한다면, 구현될 수 없음. 이런 경우 자격 증명에 안전한 암호화 알고리즘으로 암호화되어야 함														
	<table><tr><th>구분</th><th>공공기관</th><th>민간부문(법인·단체·개인)</th></tr><tr><td>대칭키 암호 알고리즘</td><td>SEED, LEA, HIGHT, ARIA-128/192/256</td><td>SEED ARIA-128/192/256 AES-128/192/256 Blowfish Camelia-128/192/256 MISTY1 KASUMI 등</td></tr><tr><td>공개키 암호 알고리즘 (메시지 암호·복호화)</td><td>RSAES-OAEP</td><td>RSA RSAES-OAEP RSAES-PKCS1 등</td></tr><tr><td>일방향 암호 알고리즘</td><td>SHA-224/256/384/512</td><td>SHA-224/256/384/512 Whirlpool 등</td></tr></table>			구분	공공기관	민간부문(법인·단체·개인)	대칭키 암호 알고리즘	SEED, LEA, HIGHT, ARIA-128/192/256	SEED ARIA-128/192/256 AES-128/192/256 Blowfish Camelia-128/192/256 MISTY1 KASUMI 등	공개키 암호 알고리즘 (메시지 암호·복호화)	RSAES-OAEP	RSA RSAES-OAEP RSAES-PKCS1 등	일방향 암호 알고리즘	SHA-224/256/384/512	SHA-224/256/384/512 Whirlpool 등
	구분	공공기관	민간부문(법인·단체·개인)												
대칭키 암호 알고리즘	SEED, LEA, HIGHT, ARIA-128/192/256	SEED ARIA-128/192/256 AES-128/192/256 Blowfish Camelia-128/192/256 MISTY1 KASUMI 등													
공개키 암호 알고리즘 (메시지 암호·복호화)	RSAES-OAEP	RSA RSAES-OAEP RSAES-PKCS1 등													
일방향 암호 알고리즘	SHA-224/256/384/512	SHA-224/256/384/512 Whirlpool 등													
※ 출처 : 개인정보의 암호화 조치 안내서 (행정자치부, KISA, 2017.01)															
진단 방법	[진단기준] - 인증서버에서 자격증명을 암호화된 데이터로 저장한 경우 양호 - 인증서버에서 자격증명을 평문으로 저장한 경우 취약														
비고	단기 적용(적용 시 개발자 및 운영자 협의)														

15.1.7. 자격증명 - 자격증명 암호화 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	자격증명 - 자격증명 평문 저장 방지 설정		

항목 설명	만약 클라이언트 저장소의 keystore 또는 DB에 자격증명이 평문으로 저장되어 있거나, 자격증명 정보가 포함된 소스코드의 컴파일로 바이너리 파일에 자격증명이 평문으로 저장되어 있으면, 자격증명 정보 유출이 가능함												
설정 방법	1. 클라이언트 애플리케이션에서, 취약하게 영구저장된 클라이언트 자격증명은 공격자가 얻기 위한 쉬운 대상임 Keystore 또는 DB와 같은 암호화된 영구저장 메커니즘을 이용하여 클라이언트 자격증명을 저장해야 함 ※ 주의 클라이언트 코드에 직접 클라이언트 자격증명을 컴파일하면, 클라이언트 애플리케이션이 스캐닝에 취약하며 관리하기 어려우므로, 확인 직후 클라이언트의 자격증명 변경해야 함 <table><tr><th>구분</th><th>공공기관</th><th>민간부문(법인·단체·개인)</th></tr><tr><td>대칭키 암호 알고리즘</td><td>SEED, LEA, HIGHT, ARIA-128/192/256</td><td>SEED ARIA-128/192/256 AES-128/192/256 Blowfish Camelia-128/192/256 MISTY1 KASUMI 등</td></tr><tr><td>공개키 암호 알고리즘 (메시지 암호·복호화)</td><td>RSAES-OAEP</td><td>RSA RSAES-OAEP RSAES-PKCS1 등</td></tr><tr><td>일방향 암호 알고리즘</td><td>SHA-224/256/384/512</td><td>SHA-224/256/384/512 Whirlpool 등</td></tr></table> ※ 출처 : 개인정보의 암호화 조치 안내서 (행정자치부, KISA, 2017.01)	구분	공공기관	민간부문(법인·단체·개인)	대칭키 암호 알고리즘	SEED, LEA, HIGHT, ARIA-128/192/256	SEED ARIA-128/192/256 AES-128/192/256 Blowfish Camelia-128/192/256 MISTY1 KASUMI 등	공개키 암호 알고리즘 (메시지 암호·복호화)	RSAES-OAEP	RSA RSAES-OAEP RSAES-PKCS1 등	일방향 암호 알고리즘	SHA-224/256/384/512	SHA-224/256/384/512 Whirlpool 등
구분	공공기관	민간부문(법인·단체·개인)											
대칭키 암호 알고리즘	SEED, LEA, HIGHT, ARIA-128/192/256	SEED ARIA-128/192/256 AES-128/192/256 Blowfish Camelia-128/192/256 MISTY1 KASUMI 등											
공개키 암호 알고리즘 (메시지 암호·복호화)	RSAES-OAEP	RSA RSAES-OAEP RSAES-PKCS1 등											
일방향 암호 알고리즘	SHA-224/256/384/512	SHA-224/256/384/512 Whirlpool 등											
진단 방법	[진단기준] - 클라이언트에서 자격증명을 암호화된 데이터로 저장한 경우 양호 - 클라이언트에서 자격증명을 평문으로 저장한 경우 취약												
비고	단기 적용(적용 시 개발자 및 운영자 협의)												

15.1.8. 자격증명 - 공개키 암호화 알고리즘 사용 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하												
항목명	자격증명 - 공개키 암호화 알고리즘 사용 설정														
항목 설명	인증서버에 저장된 자격증명의 키가 공개키(비대칭키)가 아닌 비밀키(대칭키)를 이용할 경우, 인증서버 해킹 발생시 OAuth 2.0 인증정보 탈취 가능함														
설정 방법	1. 공개키 암호화 알고리즘의 사용은 인증서버의 자격증명을 관리하기 위한 의무로부터 자유롭게 하므로, 인증서버는 공개키만 보관하면 됨														
	<table><tr><th>구분</th><th>공공기관</th><th>민간부문(법인·단체·개인)</th></tr><tr><td>대칭키 암호 알고리즘</td><td>SEED, LEA, HIGHT, ARIA-128/192/256</td><td>SEED ARIA-128/192/256 AES-128/192/256 Blowfish Camelia-128/192/256 MISTY1 KASUMI 등</td></tr><tr><td>공개키 암호 알고리즘 (메시지 암호·복호화)</td><td>RSAPES-OAEP</td><td>RSA RSAPES-OAEP RSAPES-PKCS1 등</td></tr><tr><td>일방향 암호 알고리즘</td><td>SHA-224/256/384/512</td><td>SHA-224/256/384/512 Whirlpool 등</td></tr></table>			구분	공공기관	민간부문(법인·단체·개인)	대칭키 암호 알고리즘	SEED, LEA, HIGHT, ARIA-128/192/256	SEED ARIA-128/192/256 AES-128/192/256 Blowfish Camelia-128/192/256 MISTY1 KASUMI 등	공개키 암호 알고리즘 (메시지 암호·복호화)	RSAPES-OAEP	RSA RSAPES-OAEP RSAPES-PKCS1 등	일방향 암호 알고리즘	SHA-224/256/384/512	SHA-224/256/384/512 Whirlpool 등
	구분	공공기관	민간부문(법인·단체·개인)												
	대칭키 암호 알고리즘	SEED, LEA, HIGHT, ARIA-128/192/256	SEED ARIA-128/192/256 AES-128/192/256 Blowfish Camelia-128/192/256 MISTY1 KASUMI 등												
공개키 암호 알고리즘 (메시지 암호·복호화)	RSAPES-OAEP	RSA RSAPES-OAEP RSAPES-PKCS1 등													
일방향 암호 알고리즘	SHA-224/256/384/512	SHA-224/256/384/512 Whirlpool 등													
※ 출처 : 개인정보의 암호화 조치 안내서 (행정자치부, KISA, 2017.01)															
진단 방법	[진단기준] - 인증서버에 공개키(비대칭키)로 저장된 자격증명이 존재할 경우 양호 - 인증서버에 비밀키(대칭키)로 저장된 자격증명이 존재할 경우 취약														
비고	단기 적용(적용 시 개발자 및 운영자 협의)														

15.1.9. Secret - 안전한 비밀번호 정책 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하											
항목명	Secret - 안전한 비밀번호 정책 설정													
항목 설명	인증서버의 자격증명에 사용된 비밀번호가 취약한 비밀번호 사용이 가능한 경우, 취약한 비밀번호를 사용한 자격증명의 탈취 가능성이 존재함													
설정 방법	1. 인증서버는 온라인 비밀번호 공격을 방해하기 위해 사용자 비밀번호의 엔트로피를 증가시키도록 복잡한 사용자 비밀번호 정책을 강제하도록 결정해야 할 수 있음. 너무 높은 복잡도는 사용자가 비밀번호를 재사용하거나 종이에 기록할 가능성이 있으며, 또는 그 반대로 안전하지 않은 방법으로 비밀번호를 저장할 수 있음에 주의해야 함													
	<table><tr><th></th><th>안전한 비밀번호 조건</th><th>취약한 비밀번호 조건</th></tr><tr><td>문자구성 및 길이 조건</td><td> • 3가지 종류 이상의 문자구성으로 8자리 이상의 길이로 구성된 비밀번호 • 2가지 종류 이상의 문자구성으로 10자리 이상의 길이로 구성된 비밀번호 ※ 문자 종류는 알파벳 대문자와 소문자, 특수기호, 숫자의 4가지임 </td><td> • 2가지 종류 이하의 문자구성으로 8자리 이하의 길이로 구성된 비밀번호 • 문자구성과 관계없이 7자리 이하 길이로 구성된 비밀번호 ※ 문자 종류는 알파벳 대문자와 소문자, 특수기호, 숫자의 4가지임 </td></tr><tr><td rowspan="2">특정 정보 이용 및 패턴 조건</td><td> • 한글, 영어 등의 사전적 단어를 포함하지 않은 비밀번호 • 널리 알려진 단어를 포함하지 않거나 예측이 어렵도록 가공한 비밀번호 ※ 널리 알려진 단어는 컴퓨터 용어, 기업 등의 특정명칭을 가공하지 않고 명칭 그대로 사용하는 경우 ※ 속어, 방언, 은어 등을 포함하는 경우 • 사용자 ID와 연관성이 있는 단어구성을 포함하지 않은 비밀번호 • 제3자가 쉽게 알 수 있는 개인정보를 포함하지 않은 비밀번호 ※ 개인정보는 가족, 생일, 주소, 휴대전화번호 등을 포함 </td><td> • 한글, 영어 등을 포함한 사전적인 단어로 구성된 비밀번호 ※ 스펠링을 거꾸로 구성한 비밀번호도 포함 • 널리 알려진 단어로 구성된 비밀번호 ※ 컴퓨터 용어, 사이트, 기업 등의 특정 명칭으로 구성된 비밀번호도 포함 • 사용자 ID를 이용한 비밀번호 ※ 사용자 ID 혹은 사용자 ID를 거꾸로 구성한 비밀번호도 포함 • 제3자가 쉽게 알 수 있는 개인정보를 바탕으로 구성된 비밀번호 ※ 가족, 생일, 주소, 휴대전화번호 등을 포함하는 비밀번호 </td></tr><tr><td></td><td> • 패턴이 존재하는 비밀번호 ※ 동일한 문자의 반복 : ex) aaabbbb, 123123 ※ 키보드 상에서 연속한 위치에 존재하는 문자들의 집합 : ex) qwerty, asdfgh ※ 숫자가 제일 앞이나 제일 뒤에 오는 구성의 비밀번호 : ex) security1, may12 • 숫자와 영문자를 비슷한 문자로 치환한 형태를 포함한 구성의 비밀번호 </td></tr></table>				안전한 비밀번호 조건	취약한 비밀번호 조건	문자구성 및 길이 조건	• 3가지 종류 이상의 문자구성으로 8자리 이상의 길이로 구성된 비밀번호 • 2가지 종류 이상의 문자구성으로 10자리 이상의 길이로 구성된 비밀번호 ※ 문자 종류는 알파벳 대문자와 소문자, 특수기호, 숫자의 4가지임	• 2가지 종류 이하의 문자구성으로 8자리 이하의 길이로 구성된 비밀번호 • 문자구성과 관계없이 7자리 이하 길이로 구성된 비밀번호 ※ 문자 종류는 알파벳 대문자와 소문자, 특수기호, 숫자의 4가지임	특정 정보 이용 및 패턴 조건	• 한글, 영어 등의 사전적 단어를 포함하지 않은 비밀번호 • 널리 알려진 단어를 포함하지 않거나 예측이 어렵도록 가공한 비밀번호 ※ 널리 알려진 단어는 컴퓨터 용어, 기업 등의 특정명칭을 가공하지 않고 명칭 그대로 사용하는 경우 ※ 속어, 방언, 은어 등을 포함하는 경우 • 사용자 ID와 연관성이 있는 단어구성을 포함하지 않은 비밀번호 • 제3자가 쉽게 알 수 있는 개인정보를 포함하지 않은 비밀번호 ※ 개인정보는 가족, 생일, 주소, 휴대전화번호 등을 포함	• 한글, 영어 등을 포함한 사전적인 단어로 구성된 비밀번호 ※ 스펠링을 거꾸로 구성한 비밀번호도 포함 • 널리 알려진 단어로 구성된 비밀번호 ※ 컴퓨터 용어, 사이트, 기업 등의 특정 명칭으로 구성된 비밀번호도 포함 • 사용자 ID를 이용한 비밀번호 ※ 사용자 ID 혹은 사용자 ID를 거꾸로 구성한 비밀번호도 포함 • 제3자가 쉽게 알 수 있는 개인정보를 바탕으로 구성된 비밀번호 ※ 가족, 생일, 주소, 휴대전화번호 등을 포함하는 비밀번호		• 패턴이 존재하는 비밀번호 ※ 동일한 문자의 반복 : ex) aaabbbb, 123123 ※ 키보드 상에서 연속한 위치에 존재하는 문자들의 집합 : ex) qwerty, asdfgh ※ 숫자가 제일 앞이나 제일 뒤에 오는 구성의 비밀번호 : ex) security1, may12 • 숫자와 영문자를 비슷한 문자로 치환한 형태를 포함한 구성의 비밀번호
		안전한 비밀번호 조건	취약한 비밀번호 조건											
	문자구성 및 길이 조건	• 3가지 종류 이상의 문자구성으로 8자리 이상의 길이로 구성된 비밀번호 • 2가지 종류 이상의 문자구성으로 10자리 이상의 길이로 구성된 비밀번호 ※ 문자 종류는 알파벳 대문자와 소문자, 특수기호, 숫자의 4가지임	• 2가지 종류 이하의 문자구성으로 8자리 이하의 길이로 구성된 비밀번호 • 문자구성과 관계없이 7자리 이하 길이로 구성된 비밀번호 ※ 문자 종류는 알파벳 대문자와 소문자, 특수기호, 숫자의 4가지임											
특정 정보 이용 및 패턴 조건	• 한글, 영어 등의 사전적 단어를 포함하지 않은 비밀번호 • 널리 알려진 단어를 포함하지 않거나 예측이 어렵도록 가공한 비밀번호 ※ 널리 알려진 단어는 컴퓨터 용어, 기업 등의 특정명칭을 가공하지 않고 명칭 그대로 사용하는 경우 ※ 속어, 방언, 은어 등을 포함하는 경우 • 사용자 ID와 연관성이 있는 단어구성을 포함하지 않은 비밀번호 • 제3자가 쉽게 알 수 있는 개인정보를 포함하지 않은 비밀번호 ※ 개인정보는 가족, 생일, 주소, 휴대전화번호 등을 포함	• 한글, 영어 등을 포함한 사전적인 단어로 구성된 비밀번호 ※ 스펠링을 거꾸로 구성한 비밀번호도 포함 • 널리 알려진 단어로 구성된 비밀번호 ※ 컴퓨터 용어, 사이트, 기업 등의 특정 명칭으로 구성된 비밀번호도 포함 • 사용자 ID를 이용한 비밀번호 ※ 사용자 ID 혹은 사용자 ID를 거꾸로 구성한 비밀번호도 포함 • 제3자가 쉽게 알 수 있는 개인정보를 바탕으로 구성된 비밀번호 ※ 가족, 생일, 주소, 휴대전화번호 등을 포함하는 비밀번호												
		• 패턴이 존재하는 비밀번호 ※ 동일한 문자의 반복 : ex) aaabbbb, 123123 ※ 키보드 상에서 연속한 위치에 존재하는 문자들의 집합 : ex) qwerty, asdfgh ※ 숫자가 제일 앞이나 제일 뒤에 오는 구성의 비밀번호 : ex) security1, may12 • 숫자와 영문자를 비슷한 문자로 치환한 형태를 포함한 구성의 비밀번호												

	특정 정보 이용 및 패턴 조건	※ 영문자 "O"를 숫자 "0"으로, 영문자 "I"을 숫자 "1"로 치환 등의 패스워드 • 특정 인물의 이름을 포함한 패스워드 ※ 사용자 또는 사용자 이외의 특정 인물, 유명인, 연예인 등의 이름을 포함하는 패스워드 • 한글의 발음을 영문으로, 영문단어의 발음을 한글로 변형한 형태의 패스워드 ※ 한글의 "사랑"을 영어 "Sa Rang"으로 표기, 영문자 "LOVE"의 발음을 한글 "러브"로 표기
	기타 조건	• 해당 시스템에서 사용자가 이전에 사용하지 않고 이전 패스워드와 연관성이 있는 단어구성을 포함하지 않은 패스워드 • 시스템에서 예제로 제시되는 패스워드 • 시스템에서 초기 설정된 패스워드 • 해당 시스템에서 사용자가 이전에 사용했던 패스워드
※ 출처 : 암호정책 수립 기준 안내서 (방송통신위원회, KISA, 2010.01)		
진단 방법	[진단기준] - 자격증명에 대해 안전한 패스워드 정책을 설정한 경우 양호 - 자격증명에 대해 취약한 패스워드 정책을 설정한 경우 취약	
비고	단기 적용(적용 시 개발자 및 운영자 협의)	

15.1.10. Secret - Secret에 높은 엔트로피 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Secret - Secret에 높은 엔트로피 설정		
항목 설명	Secret 또는 토큰 핸들 생성시 사람에 의한 사용이 의도되지 않을 때, 인증 서버는 추측(Guessing) 공격의 위험에 노출될 가능성이 존재함		
설정 방법	1. Secret 또는 토큰 핸들 생성시 사람에 의한 사용이 의도되지 않을 때, 인증 서버는 추측(Guessing) 공격의 위험을 방지하기 위해 실용적인 엔트로피 수준을 포함해야 함 토큰 값은 최소 128비트 길이를 지녀야 하며, 인증서버에 의해 생성된 암호학적으로 강력한 Random 또는 Pseudo-Random 수열로부터 생성되어야 함 (현재의 Best Practice는 RFC 4086을 참고)		
진단 방법	[진단기준] - Secret 또는 토큰 핸들의 길이가 최소 128비트(16글자) 이상이고, 생성되는 Secret 또는 토큰 핸들이 예측 불가능한 경우 양호 - Secret 또는 토큰 핸들의 길이가 최소 128비트(16글자) 미만이거나 생성되는 Secret 또는 토큰 핸들이 예측이 가능한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.1.11. Secret - 계정 잠금 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Secret - 계정 잠금 설정		
항목 설명	공격자에 의한 특정 횟수 이상 인증 연속 실패 후 계정 잠금 미구현시, Dictionary 공격 또는 Brute-force 공격에 의한 자격증명 탈취가 가능함		
설정 방법	1. 패스워드에 대한 온라인 공격은 특정 실패 횟수(최대 5회) 시도 후 각각의 계정이 잠기게 하여 대비할 수 있음 ※ 주의 이 방법은 합법적인 서비스 사용자를 Lockdown할 목적에 악용 가능함		
진단 방법	[진단기준] - 연속 5회 이상 로그인 실패 이후 계정잠금 확인 가능한 경우 양호 - 연속 5회 이상 로그인 실패 이후에도 정상 로그인 확인 가능한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.1.12. Secret - Tar Pit 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Secret - Tar Pit 설정		
항목 설명	Tar Pit 미설정시, 다중 IP를 이용하여 Dictionary 공격 또는 Brute-force 공격을 이용하여 자격증명 탈취가 가능함 (단, 로그인 실패 특정 횟수 이후 계정잠금 구현시는 해당 없음)		
설정 방법	1. 인증 서버는 각각의 계정을 일시잠금하여 사용자명/패스워드에 의한 인증 실패 횟수에 반응할 수 있으며, 특정 기간 동안의 응답을 지연할 수 있음 이 시간 동안 실패 시도 횟수가 증가할 수 있음. 응답 지연의 목적은 특정 사용자명에 대한 공격자의 공격 시도 속도를 늦추게 하기 위함 ※ 주의1: 이 방법은 인증서버에 대해 매우 복잡하고 Stateful한 설계가 필요함 ※ 주의2: 송신자가 Established된 연결을 고려하여 실제 데이터를 송신 시도하기 때문에, Tar Pit 연결은 수신자를 향해 막대한 트래픽 양을 생성할 수 있으므로 서비스 영향도를 고려하여 적용해야 하며, 가장 좋은 방법은 네트워크 이상행위 식별후 보안관제사가 빠른 시간 내 네트워크 트래픽을 완전히 Drop 해야 함		
진단 방법	[진단기준] - Tar Pit이 적용되거나, 보안관제사에 의한 네트워크 트래픽 차단 조치가 가능하거나 또는 로그인 실패 특정 횟수(5회 이상) 이후 계정이 잠기는 경우 양호 - Tar Pit이 미적용되거나, 특정 횟수 인증 실패 이후에도 인증이 가능한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.1.13. Secret - CAPTCHA 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Secret - CAPTCHA 설정		
항목 설명	CAPTCHA가 적용되지 않을 경우, Dictionary 공격 또는 Brute-force 공격에 의해 자격증명 탈취가 가능함 ※ Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) 현대 컴퓨팅에서 사용하는 “Challenge-Response” 인증 유형 중 하나이며, 컴퓨터가 사람에게 텍스트 기반 그림, 사진, 음성 등을 이용한 질문을 요청하여 답변의 정답 유무를 검토하여 사람의 유무를 판별하는 기술임		
설정 방법	1. 이 아이디어는 사람과의 상호작용이 필요하여 수 많은 패스워드를 자동으로 검사하는 프로그램을 방지함 ※ 주의: 이 방법은 사용자 경험에 부정적인 영향을 가짐 ※ 참고: 최근 컴퓨팅 기술 발달로 머신러닝을 이용하여 텍스트 기반 CAPTCHA 인증우회 가능성이 존재하므로, 구글의 reCAPTCHA 사용을 권고함		
진단 방법	[진단기준] - CAPTCHA를 적용한 경우 양호 - CAPTCHA를 적용하지 않은 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.1.14. Token (접근, 갱신, 인증코드) - 토큰 범위 제한 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Token (접근, 갱신, 인증코드) - 토큰 범위 제한 설정		
항목 설명	만약 토큰 범위 제한 설정을 적용하지 않을 경우, 공격자는 서비스 제공자가 의도하지 않은 서비스 사용이 가능함 ※ OAuth 토큰 범위 제한 미설정시 아래의 위협 발생이 가능함 ○ 토큰 유출 ○ 악성 소프트웨어에 대한 토큰 발행 ○ 리소스 소유자 자격증명과 관련된 영향이 있는 토큰의 의도되지 않은 발행		
설정 방법	1. 인증서버는 토큰과 연관된 범위를 줄이거나 제한하도록 결정해야 할 수 있음 대비책은 다음과 같음 ○ 클라이언트에 특화된 정책 (예: 일반 클라이언트에 약간 영향이 있는 토큰만 발행함) ○ 서비스에 특화된 정책 (예: 매우 민감한 서비스) ○ 리소스 소유자에 특화된 정책, 또는 ○ 그러한 정책 및 선호도와와의 조합		

	2. 인증서버는 권한부여에 따라 다른 범위를 허용할 수 있음 예를 들면, 최종사용자(인증코드)와의 직접적인 상호작용을 통한 최종사용자 인증은 '사용자명'/'패스워드' 권한부여 유형을 통한 직접 인증보다는 더 신뢰한다고 생각할 수 있음
진단 방법	[진단기준] - (접근, 갱신, 인증코드) 토큰 범위를 제한 설정한 경우 양호 - (접근, 갱신, 인증코드) 토큰 범위를 제한 설정하지 않은 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

15.1.15. Token (접근, 갱신, 인증코드) - 토큰 만료 시간 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Token (접근, 갱신, 인증코드) - 토큰 만료 시간 설정		
항목 설명	만약 토큰 만료 시간 설정을 적용하지 않을 경우, 인증코드, 갱신토큰, 접근토큰 등의 유출로 인한 지불 트랜잭션, 연락처 읽기 접근 등에 사용하는 API와 관련된 토큰 유출이 가능함		
설정 방법	1. 토큰은 일반적으로 실용적인 기간 이후 만료해야 함. 이는 다른 보안 수단(서명 등)을 보완하고 강하게 하며, 모든 유형의 토큰 유출의 영향이 감소함. 토큰 유출과 연관된 위험에 따라, 토큰은 몇 분 후 만료해야 할 수 있거나(예: 지불 트랜잭션 등) 또는 몇 시간 동안 유효해야 할 수 있음 (예: 연락처 읽기 접근 등) 2. 만료시간은 다음과 같은 몇 가지 요소에 의해 결정됨 ○ 토큰 유출과 관련된 위험 ○ 근본적인 접근 권한부여 기간 ○ 접근 권한부여의 변경이 효력을 가져야 하는 기간 및 ○ 유효한 토큰을 추측하거나 생성을 위해 공격자가 필요로 하는 시간		
진단 방법	[진단기준] - (접근, 갱신, 인증코드) 토큰 만료 시간 설정한 경우 양호 - (접근, 갱신, 인증코드) 토큰 만료 시간 설정하지 않은 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.1.16. Token (접근, 갱신, 인증코드) - 짧은 토큰 만료 시간 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Token (접근, 갱신, 인증코드) - 짧은 토큰 만료 시간 설정		
항목 설명	만약 짧은 토큰 만료 시간 설정 미적용시, 인증코드, 갱신토큰, 접근토큰 등의 유출로 인한 자격증명 탈취가 가능하며, 시간동기화 미적용시 토큰 시간 계산이 잘못될 수 있음 ※ 토큰을 위한 짧은 만료 시간 설정은 다음의 위협에 보호하는 수단임 ○ 리플레이 공격		

	○ 토큰 유출 (짧은 만료 시간은 영향을 감소하게 함) ○ 온라인 추측 공격 (짧은 만료 시간은 성공 가능성을 감소하게 함)
설정 방법	1. 짧은 토큰 시간은 인증서버와 리소스 서버와의 매우 정밀한 시간 동기화가 필요함 더 짧은 시간은 더 많은 토큰 갱신(접근토큰) 또는 반복된 최종사용자 인증 프로세스가 필요할 수 있음(인증코드 및 갱신 토큰)
진단 방법	[진단기준] - (접근, 갱신, 인증코드) 짧은 토큰 만료 시간(10분) 및 시간 동기화 설정한 경우 양호 - (접근, 갱신, 인증코드) 짧은 토큰 만료 시간(10분) 또는 시간 동기화 미설정된 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

15.1.17. Token (접근, 갱신, 인증코드) - 토큰 최대 사용 횟수 또는 1회용으로 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Token (접근, 갱신, 인증코드) - 토큰 최대 사용 횟수 또는 1회용으로 설정		
항목 설명	만약 인증서버는 요청횟수 제한 또는 1회용으로 설정하지 않거나, 특정 토큰으로 수행될 수 있는 연산을 제한하지 않은 경우, 토큰을 악용한 공격 발생이 가능함 ※ 토큰을 위한 최대 사용 횟수 설정은 다음의 위협에 보호하는 수단임 ○ 토큰 리플레이 공격 ○ 계성(추측 공격)		
설정 방법	1. 만약 인증서버가 인증코드를 Redeem하기 위해 1회 이상 시도하는 것을 관찰했다면, 인증서버는 인증코드에 기반하여 권한부여된 모든 접근토큰에 대해 권한회수하기를 원할 수 있으며, 게다가 현재의 요청을 거부하기를 원할 수 있음 2. 인증코드로 접근토큰은 또한 제한된 연산 수를 가질 수 있음 이는 클라이언트가 재인증하도록 강제하며, 신규 접근토큰을 획득하기 위한 갱신 토큰을 사용하도록 강제하거나 또는 클라이언트가 사용자 개입에 의한 접근토큰을 재인증하도록 강제함		
진단 방법	[진단기준] - (접근, 갱신, 인증코드) 토큰 최대 사용 횟수(또는 1회용으로 설정) 설정 및 토큰 사용 횟수 제한 도달시, 갱신토큰 사용 강제(또는 접근토큰 재인증)토록 설정한 경우 양호 - (접근, 갱신, 인증코드) 토큰 최대 사용 횟수(또는 1회용으로 설정) 미설정 또는 토큰 사용 횟수 제한 도달시, 갱신토큰 사용 강제(또는 접근토큰 재인증)토록 미설정된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.1.18. Token (접근, 갱신, 인증코드) - 특정 리소스서버로 토큰 바인딩 설정(Audience)

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Token (접근, 갱신, 인증코드) - 특정 리소스서버로 토큰 바인딩 설정(Audience)		

항목 설명	만약 특정 리소스서버에 대해 SAML Assertions의 Audience 엘리먼트를 이용한 토큰 바인딩을 설정하지 않은 경우, 토큰을 악용한 공격 발생이 가능함 ※ 특정 리소스서버로 토큰 바인딩 설정(Audience)은 다음의 위협에 보호하는 수단임 ○ 리플레이 공격 시도 ○ 가짜 리소스 서버 또는 클라이언트에 의한 토큰 악용 ○ 가짜 리소스 서버에 대한 유효한 토큰의 유출
설정 방법	1. 멀티서비스 환경에서의 인증서버는 다른 내용을 지닌 토큰을 다른 리소스 서버로 발행하는 것을 고려할 수 있으며, 토큰을 수신할 대상 서버는 토큰 내에 명시적으로 나타내야 함을 고려해야 할 수 있음 SAML Assertion는 이를 위해 Audience 엘리먼트를 사용함(OASIS.saml-core-2.0-os) ○ 토큰이 단일 리소스 서버에서만 적용할 수 있기 때문에, 성공적인 리플레이 공격 시도의 영향을 감소함 ○ 토큰은 특정 서버에서만 사용할 수 있기 때문에, 가짜 리소스 서버 또는 클라이언트에 의한 토큰 악용을 방지함. 다른 서버에 의해 거부됨 ○ 가짜 리소스 서버에 대한 유효한 토큰의 유출의 영향이 감소함 ※ Security Assertion Markup Language(SAML) 보안 보장 마크업 언어. 인터넷상의 비즈니스 보안 정보 교환을 위한 확장성 링크 언어(XML) 기반의 보안 표준 언어로, 국제 인터넷 민간 표준 기구인 OASIS에서 제정함 다른 시스템 간의 보안 서비스 상호 운용이 가능하도록 XML로 된 정보를 기술하며, 각 시스템에 구축된 인증/인가 서비스의 변경 없이 사용자 보안 정보를 안전하게 공유할 수 있는 언어와 방법을 정의하고, 이를 통해 통합적인 비즈니스 환경을 구축할 수 있도록 함 그리고 기업 내부 또는 기업 간의 SSO(Single Sign-On) 기능을 제공하고 기업 보안 인프라 구조에 종속되지 않는 장점이 있음 ※ <Audience> 엘리먼트 의도된 Audience를 식별하기 위한 URI를 참조하며, Audience 멤버십에 대한 약관과 조건을 설명하는 문서를 식별할 수 있음 시스템 엔티티를 설명하는 SAML 이름 식별자로부터 고유 식별자 URI를 포함할 수 있음
진단 방법	[진단기준] - (접근, 갱신, 인증코드) 특정 리소스 서버로 토큰 바인딩 설정(Audience)한 경우 양호 - (접근, 갱신, 인증코드) 특정 리소스 서버로 토큰 바인딩 미설정(Audience)한 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

15.1.19. Token (접근, 갱신, 인증코드) - 토큰 Audience로서 엔드포인트 주소 사용 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Token (접근, 갱신, 인증코드) - 토큰 Audience로서 엔드포인트 주소 사용 설정		
항목 설명	만약 토큰 Audience로서 엔드포인트 주소 사용을 설정하지 않을 경우, 가짜 리소스 서버로부터의 요청을 탐지할 수 없으므로 자격증명 토큰 유출이 가능함		
설정 방법	1. (인증서버) 엔드포인트 URL이 토큰을 얻기 위해 사용되어왔던 리소스 서버를 가리킬 목적으로 사용할 수 있음 토큰이 리소스 서버의 (인증서버) 엔드포인트 URL 주소를 포함할 것이기 때문에,		

	이 방법은 가짜 리소스 서버로부터 요청을 탐지하는 것을 허용함
진단 방법	[진단기준] - (접근, 갱신, 인증코드) 토큰 Audience로서 엔드포인트 주소 사용 설정한 경우 양호 - (접근, 갱신, 인증코드) 토큰 Audience로서 엔드포인트 주소 사용 미설정된 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

15.1.20. Token (접근, 갱신, 인증코드) - Audience, 토큰에 명시적으로 정의된 범위 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Token (접근, 갱신, 인증코드) - Audience, 토큰에 명시적으로 정의된 범위 설정		
항목 설명	만약 Audience 및 토큰에 대해 명시적으로 정의된 범위를 설정하지 않은 경우, 리소스 서버 또는 클라이언트가 의도한 목적과는 다른 목적으로 토큰 사용이 가능함		
설정 방법	1. 상용에는 모든 범위가 특정 리소스 서버와 연관되어 있게 명시적으로 정의된 범위를 지닌 토큰만 이용하는 것을 고려해야 할 수 있음		
진단 방법	[진단기준] - (접근, 갱신, 인증코드) Audience, 토큰에 명시적으로 정의된 범위를 설정한 경우 양호 - (접근, 갱신, 인증코드) Audience, 토큰에 명시적으로 정의된 범위를 미설정된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.1.21. Token (접근, 갱신, 인증코드) - 클라이언트 ID에 대한 토큰 바인딩 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Token (접근, 갱신, 인증코드) - 클라이언트 ID에 대한 토큰 바인딩 설정		
항목 설명	만약 인증서버가 특정 클라이언트 식별자에 대한 토큰 바인딩 설정을 하지 않은 경우, 토큰을 악용한 공격 발생이 가능함 ※ 클라이언트 ID에 대한 토큰 바인딩 설정은 다음의 위협에 보호하는 수단임 ○ 토큰 유출 탐지 불가 ○ 토큰 악용 방지 불가		
설정 방법	1. 인증서버는 특정 클라이언트 식별자에 대해 토큰을 바인딩해야 할 수 있음 이 식별자는 토큰과 관련된 모든 요청에 대해 유효성을 입증해야 함		
진단 방법	[진단기준] - (접근, 갱신, 인증코드) 클라이언트 ID에 대한 토큰 바인딩을 설정한 경우 양호 - (접근, 갱신, 인증코드) 클라이언트 ID에 대한 토큰 바인딩을 미설정된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.1.22. Token (접근, 갱신, 인증코드) - Self-Contained 토큰에 서명 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Token (접근, 갱신, 인증코드) - Self-Contained 토큰에 서명 설정		
항목 설명	만약 Self-Contained 토큰에 서명 설정하지 않은 경우, 공격자에 의해 변조/생성된 가짜 토큰 탐지가 불가능함 ※ <u>Self-Contained 토큰</u> 접근토큰이 검증될 수 있는 수단으로 인증정보가 포함된 토큰이며, 토큰 내의 서명을 이용하여 인증 정보를 인코딩하여 사용함		
설정 방법	1. Self-Contained 토큰은 가짜 토큰을 변조하거나 생성하기 위한 어떠한 시도를 탐지하기 위해 서명되어야 함 (예: 해시 기반 MAC 또는 디지털 서명)		
진단 방법	[진단기준] - (접근, 갱신, 인증코드) Self-Contained 토큰에 서명을 설정한 경우 양호 - (접근, 갱신, 인증코드) Self-Contained 토큰에 서명을 미설정한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.1.23. Token (접근, 갱신, 인증코드) - 토큰 내용 암호화 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Token (접근, 갱신, 인증코드) - 토큰 내용 암호화 설정		
항목 설명	만약 Self-Contained 토큰 내용 암호화 설정하지 않은 경우, 토큰 내용 평문 노출 또는 시스템 내부 데이터 노출이 가능함 ※ <u>Self-Contained 토큰</u> 접근토큰이 검증될 수 있는 수단으로 인증정보가 포함된 토큰이며, 토큰 내의 서명을 이용하여 인증 정보를 인코딩하여 사용함		
설정 방법	1. Self-Contained 토큰은 기밀성 이유 또는 시스템 내부 데이터를 보호 목적으로 암호화할 수 있음 토큰 포맷에 따라, 키(예 : 대칭키)는 서버 노드 사이에 분산되어야 할 수 있음 분산 방법은 토큰에 의해 정의되어야 하며, 암호화가 사용되어야 함		
진단 방법	[진단기준] - (접근, 갱신, 인증코드) Self-Contained 토큰 내용 암호화 설정한 경우 양호 - (접근, 갱신, 인증코드) Self-Contained 토큰 내용 암호화 미설정한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.1.24. Token (접근, 갱신, 인증코드) - 표준 Assertion 포맷 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Token (접근, 갱신, 인증코드) - 표준 Assertion 포맷 설정		
항목 설명	만약 표준 Assertion 포맷 설정을 하지 않은 경우, 안전한 OAuth 이용이 어려울 수 있음 ※ <u>표준 Assertion 포맷 설정은 다음의 위협에 보호하는 수단임</u> - o Issuer 누락 : 인증서버에 의해 인식될 Assertion이 발행된 엔티티 식별 불가 - o Subject 누락 : 요청 받는 접근 토큰에 대해 인증된 접속자 식별 불가 - o Audience 누락 : 의도된 Audience로서 인증서버 식별 불가 - o Expires 누락 : Assertion이 사용할 수 있는 시간대 제한 불가 - o Issued 누락 : Assertion이 발행된 UTC 시간 식별 불가		
설정 방법	1. Assertion 기반 토큰 설계를 구현하기 위해 의도하는 서비스 제공자를 위해, 표준 Assertion 포맷(예: SAML, OASIS.saml-core-2.0-os) 또는 JSON Web Token(JWT, OAuth-JWT)을 채택하는 것을 매우 추천함 2. 인증 서버는 유효하지 않은 서명 또는 MAC(Message Authentication Code)이 포함된 Assertion을 거부해야만 함 알고리즘은 서명 또는 MAC의 유효성을 입증하는 데 사용함		
진단 방법	[진단기준] - (접근, 갱신, 인증코드) 표준 Assertion 포맷 설정한 경우 양호 - (접근, 갱신, 인증코드) 표준 Assertion 포맷 미설정된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.1.25. Token (접근, 갱신, 인증코드) - 접근토큰 보호 설정

분류	일반 설정 (적용 대상 : 모든 OAuth 컴포넌트)	중요도	하
항목명	Token (접근, 갱신, 인증코드) - 접근토큰 보호 설정		
항목 설명	만약 접근토큰 보호 설정을 하지 않은 경우, 토큰을 악용한 공격 발생이 가능함 ※ <u>표준 Assertion 포맷 설정은 다음의 위협에 보호하는 수단임</u> - o 악성 애플리케이션이 메모리 내에 존재하는 접근토큰 접근 가능 - o 평문 통신으로 토큰 탈취 가능 - o 서드파티 취약점으로 서드파티에 공유된 토큰의 유출 가능		
설정 방법	1. 일시적인 메모리 내에 접근토큰을 유지해야 함 (클라이언트 애플리케이션만 접근할 수 있어야 함) 2. 보안전송(TLS)을 이용하여 안전하게 토큰을 전달해야 함 3. 클라이언트 애플리케이션이 서드파티로 토큰을 공유하지 않도록 보장해야 함		
진단 방법	[진단기준] - (접근) 접근토큰 보호 설정한 경우 양호 - (접근) 접근토큰 보호 미설정된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.2. 인증서버 설정

15.2.1. 인증코드 - 토큰 악용사례 탐지시 파생된 토큰 자동 폐기 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	인증코드 - 악용사례 탐지시 파생된 토큰 자동 폐기 설정		
항목 설명	만약 악용사례 탐지시 파생된 토큰 자동 폐기를 설정하지 않은 경우, 악용사례에 사용된 토큰을 이용한 공격 발생이 가능함		
설정 방법	1. 만약 인증서버가 인증 권한부여(예: 인증코드 등)를 Redeem하기 위해 여러 번 시도를 관찰할 수 있으면, 인증서버는 인증 권한부여에 기반하여 권한부여된 모든 토큰에 대한 폐기를 원할 수 있음		
진단 방법	[진단기준] - 토큰 악용사례 탐지시 파생된 토큰 자동 폐기 설정한 경우 양호 - 토큰 악용사례 탐지시 파생된 토큰 자동 폐기 미설정한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.2.2. 갱신토큰 - 갱신토큰 발행 제한 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	갱신토큰 - 갱신토큰 발행 제한 설정		
항목 설명	갱신토큰이 장기간의 자격증명(Credential)이기 때문에, 갱신토큰이 탈취될 경우, 자격증명 탈취가 가능함		
설정 방법	1. 인증서버는 적절한 정책에 기반하여 갱신 토큰을 발행하지 않도록 결정할 수 있음 예를 들면, 만약 인증서버가 클라이언트에서 토큰 등을 안전하게 저장하는 것을 신뢰하지 않으면, 클라이언트에 갱신토큰을 발행하는 것을 거부할 수도 있음		
진단 방법	[진단기준] - 동일 갱신토큰으로 다중 동시 로그인 등 성공시 갱신토큰 발행 제한 설정한 경우 양호 - 동일 갱신토큰으로 다중 동시 로그인 등 성공시 갱신토큰 발행 제한 미설정한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.2.3. 갱신토큰 - "client_id"로 갱신토큰 바인딩 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	갱신토큰 - "client_id"로 갱신토큰 바인딩 설정		
항목 설명	만약 "client_id"로 갱신토큰 바인딩 설정이 되어있지 않은 경우, 인증서버는 클라이언트 식별 불가한 상태에서 갱신토큰 탈취 및 유출이 가능함		
설정 방법	1. 인증서버는 발행 받을 클라이언트의 식별자에 대해 모든 갱신토큰과 일치해야 함 2. 인증서버는 모든 요청에 대해 접근토큰을 갱신하기 위해 동일한 "client_id"가 존재함을 검사해야 함 3. 만약 가능하면(예: 기밀성 있는 클라이언트), 인증서버는 각 클라이언트를 인증해야 함 이는 갱신토큰 탈취 또는 유출에 대한 대비책임 ※ 주의 이 바인딩은 비인증 변조로부터 보호되어야 함		
진단 방법	[진단기준] - "client_id"로 갱신토큰 바인딩 설정한 경우 양호 - "client_id"로 갱신토큰 바인딩 미설정된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.2.4. 갱신토큰 - 갱신토큰 순환 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	갱신토큰 - 갱신토큰 순환 설정		
항목 설명	만약 갱신토큰 순환 설정을 적용하지 않은 경우, 다른 앱/장치로부터 병렬로 동일한 갱신토큰 사용이 가능함		
설정 방법	1. 인증 토큰 순환은 다른 앱/장치로부터 병렬로 동일한 갱신토큰을 사용하는 것을 시도하는 것을 자동으로 탐지 및 방지하기 위해 의도함 인증 토큰 순환은 만약 토큰이 클라이언트로부터 탈취되고, 공격자 및 합법적인 클라이언트 모두에 의해 나중에 사용되면 발생해야 함 기본 아이디어는 오래된 갱신토큰을 이용하여 접근토큰을 획득하기 위한 시도를 탐지하기 위해 모든 갱신 요청이 지닌 갱신토큰 값을 변경해야 함 인증서버는 공격자 또는 합법적인 클라이언트가 접근을 시도하는지 아닌지를 결정할 수 없기 때문에, 만일 유효한 갱신토큰을 접근 시도한 경우와 유효한 갱신토큰과 관련된 접근 인증인 경우 모두 폐기해야 함 2. 토큰 응답으로 이 방법을 지원하는 OAuth 사양은 인증서버가 "refresh_token" 권한부여 유형을 지닌 요청에 대해 훨씬 새로운 갱신토큰을 반환하는 것을 허용함 ※ 주의		

	현재 유효한 갱신토큰의 사용이 보장되어야만 하기 때문에, 이 방법은 클러스터화된 환경에서 문제를 일으킬 수 있음. 그러한 환경에서는, 다른 방법이 더 적절할 수 있음
진단 방법	[진단기준] - 갱신토큰 순환 설정한 경우 양호 - 갱신토큰 순환 미설정된 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

15.2.5. 갱신토큰 - 갱신토큰 폐기 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	갱신토큰 - 갱신토큰 폐기 설정		
항목 설명	만약 갱신토큰 폐기 설정을 적용하지 않은 경우, 토큰을 악용한 공격 발생이 가능함 ※ <u>갱신토큰 폐기 설정은 다음의 위협에 보호하는 수단임</u> ○ 장비 탈취 ○ 리소스 소유자의 계정 도용 또는 ○ 수상한 상태로 위협에 노출된 클라이언트 애플리케이션		
설정 방법	1. 인증서버는 유효하지 않은 갱신 토큰을 명시적으로 요청하기 위한 클라이언트 또는 최종사용자를 거부해야 함 토큰을 폐기하는 메커니즘은 OAuth-REVOCATION에 정의됨		
진단 방법	[진단기준] - 갱신토큰 폐기 설정한 경우 양호 - 갱신토큰 폐기 미설정된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.2.6. 갱신토큰 - 장치 식별 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	갱신토큰 - 장치 식별 설정		
항목 설명	만약 갱신토큰 내 장치 식별 설정을 적용하지 않은 경우, 특정 장치로부터의 토큰 탈취 탐지가 불가함		
설정 방법	1. 인증서버는 장치 식별자에 대한 인증 자격증명(Credential) 바인딩이 필요함 예) 장치 식별자의 예: International Mobile Station Equipment Identity(IMEI) 2. 또한 운영체제에 특화된 식별자도 있음. 인증서버는 특정 장치로부터 토큰 탈취를 탐지하기 위해 사용자 자격증명을 인증할 때 식별자를 포함할 수 있음 ※ 주의 어떠한 구현도 장비 식별자를 이용하면 잠재적인 프라이버시 영향을 고려해야 함		
진단 방법	[진단기준] - 갱신토큰 내 장치 식별 설정한 경우 양호 - 갱신토큰 내 장치 식별 미설정된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.2.7. 갱신토큰 - X-FRAME-OPTIONS 헤더 사용 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	갱신토큰 - X-FRAME-OPTIONS 헤더 사용 설정		
항목 설명	만약 X-FRAME-OPTIONS 헤더 사용 설정을 하지 않은 경우, 모든 URL에 대해 <iframe> 태그 내 악성 페이지 렌더링 후 클릭재킹 공격 발생이 가능함 ※ X-FRAME-OPTIONS 웹 브라우저에서 <frame>, <iframe> 또는 <object> 태그 내 페이지를 렌더링하는 것을 허용 유무를 나타내기 위해 사용되는 HTTP 응답 헤더		
설정 방법	1. 최신 브라우저에서, iFrame의 회피는 X-FRAME-OPTIONS 헤더를 이용과 함께 서버 상에서 강제할 수 있음 2. 이 헤더는 "DENY"와 "SAMEORIGIN" 2개의 값을 가질 수 있음 "DENY"는 어떤 프레임이든 차단할 것이며, "SAMEORIGIN"은 Origin과 다른 사이트로 인한 어떤 프레임이든 차단할 것임 "ALLOW-FROM"은 iFrame이 출처로부터 유래할 수 있는 신뢰할 Origin 목록을 지정함		
진단 방법	[진단기준] - HTTP 응답 헤더에 X-FRAME-OPTIONS 설정한 경우 양호 - HTTP 응답 헤더에 X-FRAME-OPTIONS 미설정된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.2.8. 클라이언트 인증 및 인가 - 클라이언트로 Secret 발행에 필요한 보안정책 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	클라이언트 인증 및 인가 - 클라이언트로 Secret 발행에 필요한 보안정책 설정		
항목 설명	Secret을 보호할 수 있는 수단이 없는 일반 클라이언트에 대해 Secret 발행시, 클라이언트 상에서 Secret의 유·노출 및 위·변조로 인한 클라이언트 식별자 신뢰가 불가함 ※ 클라이언트로 Secret 발행에 필요한 보안정책 설정은 다음의 위협에 보호하는 수단임 - o Native 애플리케이션의 전체 설치로 공유된 단일 클라이언트 ID 및 Secret을 생성하기 위한 이득이 제한됨 - 그러한 시나리오는 이 Secret이 반드시 각각의 배포 채널(예: 최종사용자 장치 상에서 모든 응용프로그램 설치를 위한 애플리케이션 마켓 등)을 통해 개발자로부터 전송 받아야만 함 - 애플리케이션 소스코드 내 또는 연관된 리소스 번들에 기록된 Secret은 역공학 (Reverse Engineering)으로부터 보호되지 않음 - o Secret은 일을 중단하면서 모든 설치를 즉시 제거할 수 없기 때문에, 그러한 Secret은 폐기할 수 없음. 게다가, 인증서버는 클라이언트의 식별자를 진짜 신뢰할 수 없기 때문에, 최종사용자에게 클라이언트의 신뢰성을 나타내는 것이 위험하게 됨		
설정 방법	1. 인증서버는 Secret을 보호할 수 없는 클라이언트("일반" 클라이언트)에 대해 Secret을 발행해서는 안됨 이는 강력한 인증 방식으로 클라이언트를 처리해야 하는 서버의 개연성이 감소함		
진단 방법	[진단기준] - 클라이언트 보안정책 설정이 존재하거나 Secret 발행이 불가능한 경우 양호 - 클라이언트 보안정책 설정이 존재하지 않고 Secret 발행이 가능한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.2.9. 클라이언트 인증 및 인가 - Secret 없는 일반 클라이언트 위한 사용자 약관 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	클라이언트 인증 및 인가 - Secret 없는 일반 클라이언트 위한 사용자 약관 설정		
항목 설명	클라이언트 ID만 사용하는 일반 클라이언트가 인증서버로 사용자 약관 설정 미적용으로 최종사용자 승인 없이 자동인증이 가능한 경우, 클라이언트 ID 탈취시 자동인증으로 인한 OAuth 부정 사용이 가능함 ※ Secret 없는 일반 클라이언트 위한 사용자 약관 설정은 다음의 위협에 보호하는 수단임 o 일반 클라이언트 애플리케이션의 도용		
설정 방법	1. 인증서버는 일반 클라이언트를 위한 자동인증을 허용해서는 안됨 인증서버는 개별 클라이언트 ID를 발행할 수 있지만, 모든 인증은 최종사용자에 의해 승인 받도록 요구해야 함		
진단 방법	[진단기준] - 클라이언트 ID만 이용한 OAuth 서비스 이용시 최종사용자 승인 없이 자동인증 불가능한 경우 양호 - 클라이언트 ID만 이용한 OAuth 서비스 이용시 최종사용자 승인 없이 자동인증 가능한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.2.10. 클라이언트 인증 및 인가 - “redirect_uri” 조합으로 “client_id”만 발행 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	클라이언트 인증 및 인가 - “redirect_uri” 조합으로 “client_id”만 발행 설정		
항목 설명	만약 “redirect_uri” 조합으로 “client_id”만 발행하도록 설정하지 않은 경우, Secret 없는 클라이언트에 대한 공격이 가능함 ※ “redirect_uri” 조합으로 “client_id”만 발행하는 설정은 다음의 위협에 보호하는 수단임 - o Cross-Site Scripting(XSS) 공격 - o 일반 클라이언트 애플리케이션의 도용		
설정 방법	1. 인증서버는 "client_id"를 발행할 수 있으며, 확실하게 미리 구성된 "redirect_uri"에 대해 "client_id"와 바인딩해야 함 2. 또 다른 리다이렉트 URI를 지닌 어떤 인증 요청이라도 자동으로 거부됨 그 대신에, 인증서버는 "client_id"에 대한 동적 리다이렉트 URI를 수용하면 안 되고, 대신 잘 알려지면서 미리 구성된 리다이렉트 URI로만 항상 리다이렉트하도록 해야 함		
진단 방법	[진단기준] - “redirect_uri” 조합으로 “client_id”만 발행하도록 설정한 경우 양호 - “redirect_uri” 조합으로 “client_id”만 발행하도록 미설정인 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.2.11. 클라이언트 인증 및 인가 - 설치 전용 클라이언트 Secret 발행 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	클라이언트 인증 및 인가 - 설치 전용 클라이언트 Secret 발행 설정		
항목 설명	만약 설치 전용 클라이언트 Secret을 발행하지 않은 경우, 유출된 Secret을 이용하여 다른 용도에 활용이 가능하며, 중요 서비스와 관련시 토큰 폐기로 인한 서비스 영향이 존재함		
설정 방법	1. 인증서버는 구분된 클라이언트 식별자를 발행해야 할 수 있으며, 특정 클라이언트(예: 소프트웨어 패키지)의 다른 설치에 대해 상응하는 Secret을 발행해야 할 수 있음 그러한 접근법의 결과는 "일반" 클라이언트가 "기밀" 클라이언트로 변하게 됨 2. 웹 애플리케이션에서, 이 방법은 소프트웨어 패키지가 설치된 웹 사이트 별로 하나의 "client_id"와 "client_secret"을 생성하는 것을 의미할 수 있음 그래서, 특정 사이트의 제공자는 웹 사이트의 구축동안 인증서버로부터 클라이언트 ID와 Secret을 요청할 수 있음 이는 해당 웹 사이트의 일부 Property(예: 리다이렉트 URI, 웹 사이트 URL, 그리고 그 밖에 유용한 Property)의 유효성을 입증하는 것을 허용함 웹 사이트 제공자는 사이트 상의 클라이언트 Secret에 대한 보안을 보장해야만 함 3. Native 애플리케이션에서, 현재 사정으로는 어떤 장치에서라도 특정 애플리케이션의 모든 카피본 때문에 더욱 복잡해지게 됨. 이 시나리오에서 설치 전용 Secret은 "client_id"와 "client_secret" 모두를 얻기 위해 요청하게 됨 - o 1. 애플리케이션 마켓으로부터 다운로드 프로세스 동안 또는 - o 2. 장치 상에서 설치하는 동안 첫 번째 접근법은 애플리케이션의 정품에서 특정 신뢰 수준의 달성을 허용하는 것이나, 두 번째 접근법은 클라이언트의 Property의 유효성 검증이 아닌 설치에 대한 인증만 허용함 그러나, 이 방법은 몇 가지 리플레이 공격을 방지하기 위한 최소한의 도움이 됨 게다가, 설치 전용 "client_id"와 Secret은 한 번에 특정 설치의 모든 갱신토큰의 선택적인 폐기를 허용함		
진단 방법	[진단기준] - 설치 전용 Secret 발행하도록 설정한 경우 양호 - 설치 전용 Secret 발행하도록 미설정된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.2.12. 클라이언트 인증 및 인가 - 미리 등록된 "redirect_uri" 유효성 입증 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	클라이언트 인증 및 인가 - 미리 등록된 "redirect_uri" 유효성 입증 설정		
항목 설명	미리 등록된 "redirect_uri"의 유효성 입증 설정을 하지 않은 경우, 리다이렉션 공격으로 공격자에게 인증코드 유출이 가능함		

	※ 미리 등록된 "redirect_uri" 유효성 인증 설정은 다음의 위협에 보호하는 수단임 ○ 인증코드는 가짜 웹 사이트를 통해 유출함 : 인증서버가 최종사용자 인증 엔드포인트에 대해 최초 리다이렉트 후 공격 시도를 탐지하는 것을 허용함 (RFC 6819, Section 4.4.1.7 참고) ○ 클라이언트 리다이렉션 엔드포인트를 통한 Open Redirector 공격 (RFC 6819, Section 4.1.5 참고) ○ 인증서버 리다이렉션 엔드포인트를 통한 Open Redirector 피싱 공격 (RFC 6819, Section 4.1.5 참고) ※ 참고 이 방법의 근본적인 가정은 공격자가 인증코드를 접근하기 위해 또 다른 리다이렉트 URI를 사용할 필요가 있음 상용에서는 공격대상 장치에 대해 Spoofing 공격을 이용한 공격자가 이 대비책을 회피하는 가능성을 고려할 필요가 있음
설정 방법	1. 인증서버는 모든 클라이언트가 "redirect_uri"를 등록하고, "redirect_uri"는 RFC 6749 (The OAuth 2.0 Authorization Framework)에 정의된 것처럼 전체 URI가 되어야 함을 요구해야 함 2. 이미 결정된 핵심 사양에 따라, 최종사용자 인증 엔드포인트에 대한 각각 "client_id"로 받는 모든 실제 리다이렉트 URI는 등록된 리다이렉트 URI와 일치해야만 함. 일치하지 않은 URI에, 인증서버는 공격자가 전송한 인바운드 GET 요청으로 가정하여 해당 요청을 거부해야 함 ※ 주의1 인증서버는 User Agent가 인증요청과 같은 리다이렉트 URI로 복귀하지 않도록 해야 함 ※ 주의2 미리 등록 중인 클라이언트는 (수동 프로세스를 사용하는) 일부 상용에서 변경 불가할 수 있으나, (아직 명세되지 않았지만) 동적 클라이언트 등록이 필요할 수 있음 동적 클라이언트 등록 누락으로, 미리 등록된 "redirect_uri"는 개발/환경구성 시간에 특정 상용에 대해 바인딩된 클라이언트에서만 동작함 동적 리소스 서버 발견을 요청 받은 다음부터는, 미리 등록된 "redirect_uri"가 더 이상 실현 가능하지 않을 수 있음
진단 방법	[진단기준] - 미리 등록된 "redirect_uri" 유효성 인증 설정한 경우 양호 - 미리 등록된 "redirect_uri" 유효성 인증 미설정된 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

15.2.13. 클라이언트 인증 및 인가 - 클라이언트 Secret 폐기 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	클라이언트 인증 및 인가 - 클라이언트 Secret 폐기 설정		
항목 설명	만약 클라이언트 Secret 폐기 설정을 하지 않은 경우, 유출된 클라이언트 Secret에 대한 악용이 가능함 ※ 클라이언트 Secret 폐기 설정은 다음의 위협에 보호하는 수단임 o 개인 클라이언트의 공개된 클라이언트 Secret의 악용		
설정 방법	1. 인증서버는 공개된 Secret의 악용을 방지하기 위해 클라이언트의 Secret을 폐기해야 할 수 있음 ※ 주의 이 방법은 각 클라이언트에 발행된 어떠한 인증코드 또는 갱신토큰이 유효하지 않음을 즉시 입증해야 함 이 방법은 특정 Native 또는 웹 애플리케이션의 다수 상용에 걸쳐 사용되는 클라이언트 식별자 및 Secret에 의도하지 않은 영향이 있을 수 있음		
진단 방법	[진단기준] - 클라이언트 Secret 폐기 설정한 경우 양호 - 클라이언트 Secret 폐기 미설정된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

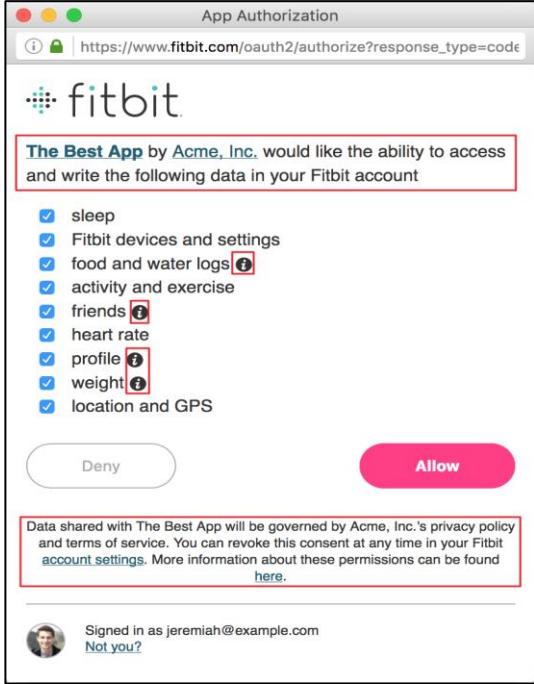
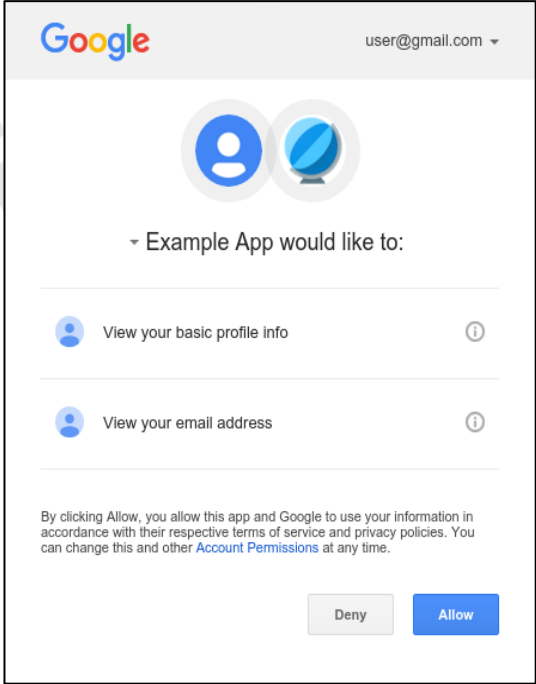
15.2.14. 클라이언트 인증 및 인가 - 강력한 클라이언트 인증 사용 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	클라이언트 인증 및 인가 - 강력한 클라이언트 인증 사용 설정		
항목 설명	만약 client_assertion 및 client_token을 이용하여 강력한 클라이언트 인증 사용 설정을 하지 않은 경우, 유출된 클라이언트 Secret에 대한 악용이 가능함 ※ <u>client_assertion</u> 클라이언트 인증서를 사용하여 서명된 JWT 토큰이 포함된 파라미터 ※ <u>client_token</u> 리소스 서버가 클라이언트로 반드시 전달해야만 하는 정보가 포함된 파라미터 ※ <u>client_secret</u> 클라이언트 자격증명이 포함된 파라미터		
설정 방법	1. 클라이언트 Assertion(OAuth-ASSERTION)과 같은 인증의 대체 형태를 이용하여, "client_secret"의 배포 필요성이 없어짐 이 방법은 안전한 개인 Keystore 사용 또는 클라이언트의 인증 프로세스에서 클라이언트 Assertion 발행자에 의해 지정된 다른 추가 인증 시스템을 요구할 수 있음		
진단 방법	[진단기준] - client_assertion 및 client_token을 이용한 강력한 클라이언트 인증 설정한 경우 양호 - client_assertion 및 client_token을 이용한 강력한 클라이언트 인증 미설정된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.2.15. 최종사용자 인증 - 자동 반복 인증 처리시 클라이언트 유효성 입증 설정

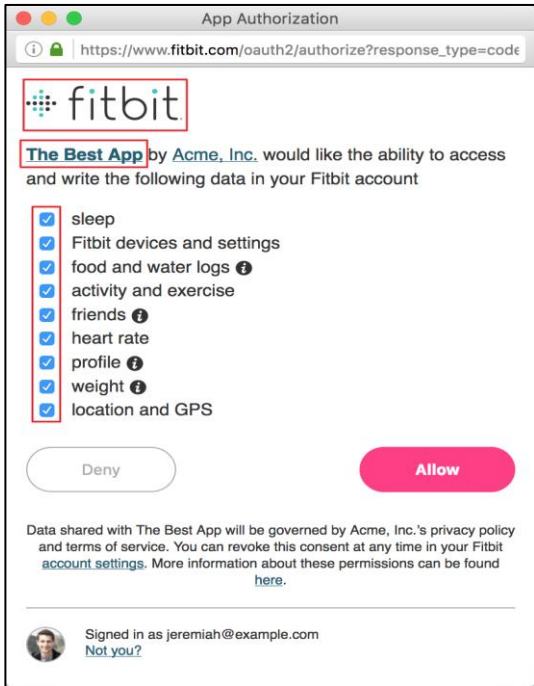
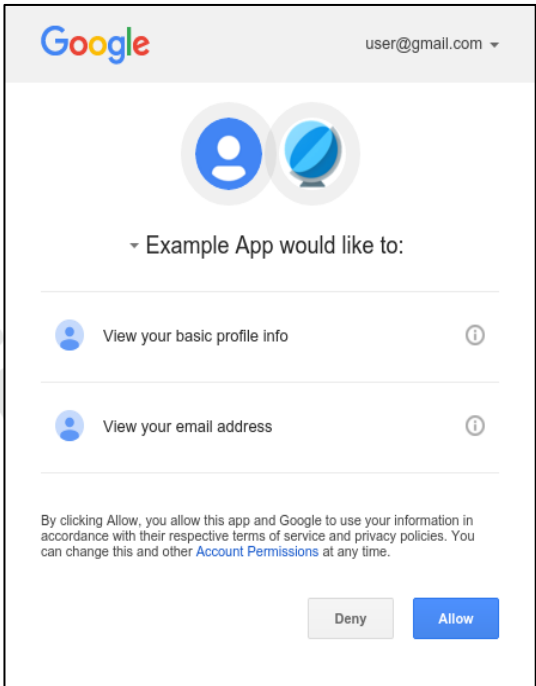
분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	최종사용자 인증 - 자동 반복 인증 처리시 클라이언트 유효성 입증 설정		
항목 설명	만약 인증서버에 자동 반복 인증 처리시 클라이언트 유효성 입증 설정을 하지 않은 경우, 비인증 클라이언트에 의한 자동 반복 인증 요청이 가능함		
설정 방법	1. 인증서버는 클라이언트 Secret 또는 일부 다른 인증 메커니즘(서명된 인증 Assertion 인증서 또는 미리 등록된 리다이렉트 URI의 유효성 입증)을 통해 인증되지 않는 클라이언트가 자동으로 인증을 반복하여 처리해서는 안 됨 ※ <u>서명된 인증 Assertion 인증서</u> : RFC 6819, Section 5.2.3.7. (문서 2.14.) 참고 ※ <u>미리 등록된 리다이렉트 URI의 유효성 입증</u> : RFC 6819, Section 5.2.3.5. (문서 2.12.) 참고		
진단 방법	[진단기준] - 자동 반복 인증 처리시 클라이언트 유효성 입증 설정한 경우 양호 - 자동 반복 인증 처리시 클라이언트 유효성 입증 미설정된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.2.16. 최종사용자 인증 - 투명성에 기반하여 현명한 결정을 위한 정보제공 기능 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	최종사용자 인증 - 투명성에 기반하여 현명한 결정을 위한 정보제공 기능 설정		
항목 설명	만약 투명성에 기반하여 현명한 결정 기능 설정을 하지 않은 경우, 최종사용자는 OAuth 인증 프로세스에 필요한 대상 URL 또는 앱, 이용 목적, 이용할 데이터, 보안 정책 등에 대한 정보를 제공을 받지 못하여 기업 신뢰도에 악영향이 미칠 수 있음		
설정 방법	1. 인증서버는 인증 프로세스에서 무엇이 발생하는지, 그 결과가 무엇인지를 최종사용자에 명확하게 설명을 해야 함 예를 들면, 사용자는 누군가 무엇을 하기 위해 누군가의 클라이언트에 권한을 부여하려 하기 위해 접근하려는 이유가 무엇인지를 이해해야 함 2. 또한 인증서버가 어떤 클라이언트 Property(웹 사이트 URL, 보안 정책)를 확실하게 증명할 수 있는지 없는지를 사용자에게 명확하게 해야 함 ※ 투명성에 기반하여 현명한 결정을 위한 정보제공 기능 설정 (설정 : 좌측, 미설정 : 우측)  		
진단 방법	[진단기준] - 투명성에 기반하여 현명한 결정을 위한 정보제공 기능 설정한 경우 양호 - 투명성에 기반하여 현명한 결정을 위한 정보제공 기능 미설정한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.2.17. 최종사용자 인증 - 최종사용자에 의한 클라이언트 Property 유효성 입증 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
----	------------------------------------	-----	---

항목명	최종사용자 인증 - 최종사용자에 의한 클라이언트 Property 유효성 입증 설정
항목 설명	만약 최종사용자에 의한 클라이언트 Property 유효성 입증 설정을 하지 않은 경우, 최종 사용자는 OAuth 클라이언트 애플리케이션 및 Property의 수집 목적을 불충분하게 제공받으며, 제공을 원치않는 Property를 제외하지 못하여 기업 신뢰도에 악영향이 미칠 수 있음
설정 방법	1. 인증 프로세스에서 사용자는 일반적으로 인증을 위해 클라이언트의 요청을 승인하기 위해 요청받은 이는 인증서버에 잘 알려진 클라이언트 이름이 최종사용자가 이용 중인 웹 사이트 또는 애플리케이션의 이름과 알맞은지 아닌지의 여부를 클라이언트 Property에 대한 유효성 입증에서 개입할 수 있기 때문에 최종사용자 스스로 중요한 보안 메커니즘임 이 방법은 특히 인증서버가 클라이언트 인증을 위해 사용할 수 없는 상황에 도움이 됨 ※ 최종사용자에 의한 클라이언트 Property 유효성 입증 설정 (설정 : 좌측, 미설정 : 우측)  
진단 방법	[진단기준] - 최종사용자에 의한 클라이언트 Property 유효성 입증 설정한 경우 양호 - 최종사용자에 의한 클라이언트 Property 유효성 입증 미설정한 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

15.2.18. 최종사용자 인증 - 인증코드를 “client_id”에 바인딩 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	최종사용자 인증 - 인증코드를 “client_id”에 바인딩 설정		
항목 설명	만약 인증코드를 “client_id”에 바인딩 설정을 하지 않으면, 인증코드 탈취 또는 인증코드로 리플레이 공격이 가능함 ※ 인증코드를 “client_id”에 바인딩 설정은 다음의 위협에 보호하는 수단임		

	○ 공격자는 또 다른 "client_id"에 토큰 내의 인증코드를 변경하는 데 사용할 수 없기 때문에, 다른 클라이언트 자격증명(Credential)과 함께 인증코드로 리플레이 공격 ○ 인증코드에 대한 온라인 추측(게싱) 공격
설정 방법	1. 인증서버는 모든 인증코드를 최종사용자 인증 프로세스로 초기화된 각 클라이언트에 대한 ID에 대해 바인딩을 해야 함 ※ 주의 이 바인딩은 비인증 상태의 변경으로부터 보호되어야 함 (예: 메모리 보호 및/또는 안전한 DB 사용)
진단 방법	[진단기준] - 인증코드를 "client_id"에 바인딩 설정한 경우 양호 - 인증코드를 "client_id"에 바인딩 미설정된 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

15.2.19. 최종사용자 인증 - 인증코드를 "redirect_uri"에 바인딩 설정

분류	인증서버 설정 (적용 대상 : OAuth 인증서버 엔드포인트)	중요도	하
항목명	최종사용자 인증 - 인증코드를 "redirect_uri"에 바인딩 설정		
항목 설명	만약 인증코드를 "redirect_uri"에 바인딩 설정을 하지 않으면, 가짜 웹 사이트를 통한 인증코드 유출이 가능함 ※ 인증코드를 "redirect_uri"에 바인딩 설정은 다음의 위협에 보호하는 수단임 ○ 공격자가 토큰 내에 인증토큰을 변경하는 데 또 다른 리다이렉트 URI를 사용할 수 없기 때문에, 가짜 웹 사이트를 통한 인증코드 유출 공격 시도		
설정 방법	1. 인증서버는 모든 인증코드를 최종사용자 인증 프로세스 내 클라이언트의 리다이렉트 대상으로 사용된 실제 리다이렉트 URI에 대해 바인딩을 할 수 있어야 함 2. 이 바인딩은 클라이언트가 접근토큰을 위한 각각 인증코드를 교환하기 위해 시도할 때 유효성이 입증되어야 함		
진단 방법	[진단기준] - 인증코드를 "redirect_uri"에 바인딩 설정한 경우 양호 - 인증코드를 "redirect_uri"에 바인딩 미설정된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.3. 클라이언트 앱 보안 설정

15.3.1. 소프트웨어 패키지와 번들된 코드 또는 리소스 내 자격증명 저장 금지 설정

분류	클라이언트 앱 보안 설정 (적용 대상 : 클라이언트 앱)	중요도	상
항목명	소프트웨어 패키지와 번들된 코드 또는 리소스 내 자격증명 저장 금지 설정		
항목 설명	소프트웨어 패키지와 번들된 코드 또는 리소스 내 자격증명을 저장시, 클라이언트 상에서 Secret의 유·노출 및 위·변조로 인한 클라이언트 식별자 신뢰가 불가함 ※ 클라이언트로 Secret 발행에 필요한 보안정책 설정은 다음의 위협에 보호하는 수단임 - 클라이언트 소프트웨어 카피본 수 때문에, 애플리케이션의 모든 설치에 의해 공유된 단일 클라이언트 ID 및 Secret을 생성하면 이익이 제한됨 "일반" 클라이언트로 간주될, 클라이언트에 의한 애플리케이션은 클라이언트 Secret을 유지할 수 있음을 추정할 수 없음 - 애플리케이션 소스코드 내 또는 연관된 리소스 번들에 기록된 Secret은 역공학(Reverse Engineering)으로부터 보호되지 않음 - Secret은 일을 중단하면서 모든 설치를 즉시 제거할 수 없기 때문에, 그러한 Secret은 폐기할 수 없음. 게다가, 인증서버는 클라이언트의 식별자를 진짜 신뢰할 수 없기 때문에, 최종사용자에게 클라이언트의 신뢰성을 나타내는 것이 위험하게 됨		
설정 방법	1. 소프트웨어 패키지와 번들된 코드 또는 리소스 내 자격증명(Credential) 저장하면 안 됨		
진단 방법	[진단기준] - S/W 패키지와 번들된 코드 또는 리소스 내 자격증명 저장 금지 설정한 경우 양호 - S/W 패키지와 번들된 코드 또는 리소스 내 자격증명 저장 금지 미설정한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.3.2. 표준 웹 서버 보호 수단 설정

분류	클라이언트 앱 보안 설정 (적용 대상 : 클라이언트 앱)	중요도	상
항목명	표준 웹 서버 보호 수단 설정 (대상: 환경설정 파일 및 DB)		
항목 설명	표준 웹 서버 보호 수단 강제 설정을 하지 않을 경우 공격자에 의해 민감한 환경설정 파일 및 DB 탈취가 가능함		
설정 방법	1. 안전하게 서버 소프트웨어를 설치해야 함 ○ 전용 호스트 또는 만약 가상화 기술 활용이 가능한 전용 게스트 OS 상에 서버 SW를 설치해야 함 ○ 서버 소프트웨어 내 알려진 취약점 수정을 위해 패치 또는 업그레이드를 적용해야 함 ○ 만약 적용할 수 있으면, 전용 물리 디스크 또는 (OS와 서버 애플리케이션을 분리한) 논리 파티션을 생성해야 함 ○ 서버 애플리케이션에 의해 설치되었지만 불필요한 모든 서비스에 대해 삭제 또는 비활성화를 해야 함 (예: Gopher, FTP, HTTP, 원격관리 등) ○ 서버 설치시 생성된 모든 불필요한 기본 사용자 계정을 삭제 또는 비활성화를 해야 함 ○ 서버로부터 모든 벤더사의 문서를 삭제해야 함 ○ 서버로부터 샘플 콘텐츠, 스크립트 및 실행파일 코드를 포함한 모든 예제 또는 테스트 파일을 삭제해야 함 ○ 모든 불필요한 컴파일러를 삭제해야 함 ○ 서버에 대해 적절한 보안 템플릿 또는 보안 강화 스크립트를 적용해야 함 ○ 외부망 서버에 대해, 만약 가능하면, 서비스 배너가 OS 종류 및 버전을 노출하지 않도록 설정을 다시 해야 함 ○ 배너를 지원하는 모든 서비스에 대해 경고 배너를 설정해야 함 ○ 만약 가능하면, 필요한 TCP 및 UDP 포트만 클라이언트 연결에 대한 Listen을 위한 각 네트워크 서비스에 대해 설정을 해야 함 2. 접근통제를 설정해야 함 ○ 서버 애플리케이션의 계산할 리소스의 부분집합으로 접근 제한을 해야 함 ○ 더 상세한 수준의 접근통제가 필요한 서버에 의해 강제하는 부가적인 접근통제를 통해 사용자 접근을 제한해야 함 ○ 접근통제를 해야 하는 일반적인 파일은 다음과 같음 6) 애플리케이션 소프트웨어 및 환경설정 파일 7) 보안 메커니즘과 직접 관련된 파일 2-1) 인증에 사용하는 패스워드 해시 파일 및 기타 파일 2-2) 접근을 통제하기 위해 사용하는 인증 정보를 포함하는 파일 2-3) 기밀성, 무결성 및 부인방지 서비스에 사용하는 암호 키 데이터 8) 서버 로그 및 시스템 감사 파일 9) 시스템 소프트웨어 및 환경설정 파일 10) 서버 콘텐츠 파일 ○ 서비스 프로세스에 의해 접근 가능한 파일을 제한하기 위해 서버 호스트 OS 접근통제를 사용해야 함 4) 서비스 프로세스는 엄격하게 제한된 권한 집합을 가진 사용자로서 실행하도록 구성해야 함 (예: root, administrator 또는 동일한 권한을 수행하지 않아야 함) 5) 만약 필요하다면 서비스 프로세스는 서버 콘텐츠 파일 및 디렉토리에 대한 기록만 할 수 있도록 해야 함		

- 6) (만약 가능하다면) 서버 소프트웨어에 의해 생성된 임시파일은 규격에 맞고 적절하게 보호되는 하위 디렉토리로 제한해야 하며, (만약 가능하다면) 임시파일에 대한 접근은 파일을 생성했던 서버 프로세스로 제한되어야 함

3. 서버 리소스 제약조건을 적용해야 함

- o OS 및 서버 소프트웨어와 다른 보조기억장치 또는 논리 파티션에서 서버 콘텐츠를 설치해야 함
- o 만약 서버가 업로드를 허용한다면, 업로드 전용 보조기억장치 공간의 양을 제한하여 업로드 파일이 위치해야 하며, 이와 동일하게, 업로드는 보조기억장치 용량 제한을 초과할 수 없도록 더 강력한 보증을 제공한 분리 파티션에 업로드 파일이 위치해야 함
- o 만약 업로드가 서버에 허용되면, 업로드 파일은 일부 자동 또는 수동 검토 과정이 업로드 파일을 가려내는 데 사용한 후, 서버에 의해 읽을 수 없도록 보장해야 함. 이 방법은 서버가 악성코드 또는 불법 소프트웨어, 공격 도구, 포르노 등의 트래픽을 전파하기 위해 사용되는 것을 방지함. 또한 많은 대용량 파일을 업로드하여 개입한 DoS 공격의 잠재적인 영향을 제한할 수 있도록 각 업로드된 파일의 용량을 제한하는 것이 가능함
- o 로그 파일이 적절한 크기로 저장되도록 보장해야 함. 동일하게, 로그 파일은 분리된 파티션에 저장해야 함. 만약 공격이 로그 파일의 크기가 수용할 수 있는 제한을 넘어서면서 증가하도록 할 경우, 물리 파티션은 서버가 그러한 상황을 적절하게 처리하기 위해 충분한 리소스를 가지도록 보장하는 것을 도와줌
- o 서버가 허용하는 서버 프로세스 및/또는 네트워크 연결 최대 개수를 설정해야 함

4. 안전한 알고리즘을 이용한 인증 및 암호화 기술을 적용해야 함

	구분		
	구분	공공기관	민간부문(법인·단체·개인)
	대칭키 암호 알고리즘	SEED, LEA, HIGHT, ARIA-128/192/256	SEED ARIA-128/192/256 AES-128/192/256 Blowfish Camelia-128/192/256 MISTY1 KASUMI 등
	공개키 암호 알고리즘 (메시지 암호·복호화)	RSAPES-OAEP	RSA RSAPES-OAEP RSAPES-PKCS1 등
	일방향 암호 알고리즘	SHA-224/256/384/512	SHA-224/256/384/512 Whirlpool 등
	※ 출처 : 개인정보의 암호화 조치 안내서 (행정자치부, KISA, 2017.01)		
진단 방법	[진단기준] - 안전하게 서버 소프트웨어를 설치, 접근통제 설정, 서버 리소스 제약조건 적용 및 안전한 알고리즘을 이용한 인증 및 암호화 기술 적용을 한 경우 양호 - 취약하게 서버 소프트웨어를 설치, 접근통제 설정 미적용, 서버 리소스 제약조건 미적용 및 안전한 알고리즘을 이용한 인증 또는 암호화 기술 미적용인 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.3.3. 안전한 저장소에 Secret 저장 설정

분류	클라이언트 앱 보안 설정 (적용 대상 : 클라이언트 앱)	중요도	상
항목명	안전한 저장소에 Secret 저장 설정		
항목 설명	안전한 저장소에 Secret을 저장 설정하지 않은 경우, 타 사용자 또는 타 애플리케이션에 의한 비인증 접근으로 Secret 유출 및 악용이 가능함		
설정 방법	1. 장치 또는 서버 상 안전하게 모든 종류(토큰, 클라이언트 Secret)의 Secret을 저장하기 위한 다른 방법이 존재함 대부분의 멀티유저 운영체제는 시스템 사용자와 다른 개인 저장소로 분리함 게다가, 대부분의 최신 스마트폰 운영체제는 심지어 파일시스템에 대해 분리된 영역 내 애플리케이션 전용 데이터 저장소를 지원하며, 다른 애플리케이션에 의한 접근으로부터 데이터를 보호함 2. 부가적으로, 애플리케이션은 PIN 또는 패스워드와 같은 사용자가 입력한 Secret을 이용하여 기밀 데이터를 구현할 수 있음 3. 또 다른 옵션은 갱신토큰 저장소를 신뢰할 수 있는 백엔드 서버로 변경하면 됨 이 옵션은 교대작업으로 클라이언트와 백엔드 서버 사이의 탄력적인 인증 메커니즘을 요구함 ※ 주의 일반적으로 애플리케이션의 로컬 메모리 내 기밀 데이터를 유지한다는 것을 의미하므로 애플리케이션은 기밀 데이터가 안전한 저장소로부터 읽어들인 후에도 기밀성을 유지해야 함을 보장해야 함		
진단 방법	[진단기준] - 안전한 저장소에 Secret 저장 설정 적용한 경우 양호 - 안전한 저장소에 Secret 저장 설정 미적용한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.3.4. 비인증된 장치 접근을 방지하기 위한 디바이스 잠금 설정

분류	클라이언트 앱 보안 설정 (적용 대상 : 클라이언트 앱)	중요도	상
항목명	비인증된 장치 접근을 방지하기 위한 디바이스 잠금 설정		
항목 설명	비인증된 장치 접근을 방지하기 위한 디바이스 잠금 설정하지 않은 경우, 공격자가 단말기 습득 후 탈옥/루팅을 시도한 후 내부 저장소에 저장된 인증정보 및 개인정보, 금융정보 등 민감한 정보 유출 및 악용이 가능함		
설정 방법	1. 일반적인 최신 스마트폰에서, 장치가 도난당하거나 분실했을 때 부가적인 보호를 제공하기 위해 활용할 수 있는 많은 '디바이스 잠금'이 존재함 이 '디바이스 잠금'은 PIN, 패스워드, 그리고 기타 '안면인식'과 같은 생체인증 기능을 포함함 ※ 참고 OAuth 2.0을 지원하는 MDM을 이용하여 API 기반으로 디바이스 잠금 설정이 가능함 (기능 지원 여부, 지원 OS 확인 및 정책 설정 방법은 벤더사에 확인 권고) ※ 주의 장치가 제공하는 보안의 수준은 동일하지 않음		
진단 방법	[진단기준] - 비인증 장치 접근을 방지하기 위한 디바이스 잠금 설정 적용한 경우 양호 - 비인증 장치 접근을 방지하기 위한 디바이스 잠금 설정 미적용한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.3.5. User Agent 세션에 대해 “state” 파라미터로 연결 설정

분류	클라이언트 앱 보안 설정 (적용 대상 : 클라이언트 앱)	중요도	상
항목명	User Agent 세션에 대해 “state” 파라미터로 연결 설정		
항목 설명	"state" 파라미터는 클라이언트 요청을 연결하고, URI 리다이렉트와 같은 공격을 포함한 CSRF 공격을 방지하기 위해 사용됨 공격자는 결과적으로 클라이언트가 공격 대상(예: 공격 대상의 은행 계좌 정보를 공격자가 통제하는 보호된 리소스 저장)보다는 공격자의 보호된 리소스와 관련된 접근토큰을 이용하는 것을 야기할 수 있는 공격 대상의 인증코드 또는 접근토큰에 인젝션이 가능함		
설정 방법	1. 클라이언트는 인증요청을 생성 중일 때 User Agent의 인증된 상태(예: User Agent 인증 위해 사용된 세션 쿠키의 해시 등)에 대한 요청과 바인딩할 값을 인증서버에 전송하기 위해 'state' 요청 파라미터를 활용해야 함 일단 인증코드를 최종사용자로부터 획득을 하였다면, 인증서버는 User-Agent를 'state' 파라미터 내 포함된 필요로 하는 바인딩된 값을 클라이언트로 다시 보내기 위해 리다이렉트를 함 2. 바인딩된 값은 클라이언트가 User Agent의 인증된 상태에 대한 바인딩된 값과 일치함으로써 인증요청의 유효성을 검증하는 것을 가능하게 함		
진단	[진단기준]		

방법	- User Agent 세션에 대해 “state” 파라미터로 연결 설정 적용한 경우 양호 - User Agent 세션에 대해 “state” 파라미터로 연결 설정 미적용한 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

15.4. 리소스 서버 설정

15.4.1. HTTP 인증헤더 사용 설정

분류	리소스 서버 설정 (적용 대상 : 리소스 서버)	중요도	상
항목명	HTTP 인증헤더 사용 설정		
항목 설명	만약 HTTP 인증헤더 사용 설정 또는 취약한 HTTP 인증헤더를 사용하는 경우, 접근토큰 유출 또는 의도하지 않은 인증 값이 가능함		
설정 방법	1. 인증헤더는 HTTP 프록시 및 서버에 의해 특별히 인식 및 취급함 따라서, 리소스 서버에 대한 접근토큰을 전송하기 위한 헤더(특히 인증헤더)의 사용은 유출 또는 일반적으로 인증 요청의 의도하지 않은 저장 가능성이 감소함		
진단 방법	[진단기준] - Bearer 인증헤더(JWT 등) 등 안전한 인증헤더 사용 설정한 경우 양호 - Basic, Digest Access 인증헤더 사용 설정 또는 HTTP 인증헤더 미사용 설정한 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

15.4.2. 인증된 요청 사용 설정

분류	리소스 서버 설정 (적용 대상 : 리소스 서버)	중요도	상
항목명	인증된 요청 사용 설정		
항목 설명	만약 인증된 요청 사용 설정을 안 한 경우, 가짜 리소스 서버에 의한 토큰 악용이 가능함		
설정 방법	1. 인증서버는 확실한 클라이언트 식별자에게 토큰을 바인딩할 수 있으며, 리소스 서버가 리소스 접근과 연관된 유효성을 입증하는 것을 가능하게 할 수 있음 이 방법은 리소스 서버가 특정 토큰의 합법적인 소유자로서 요청 발생자를 인증하는 것이 필요하게 됨 이 대비책을 구현하기 위한 몇 가지 옵션이 있음 ○ 인증 서버는 (내부 또는 Self-Contained된 토큰의 Payload 내) 토큰으로 클라이언트 식별자를 연상할 수 있음 그러면 클라이언트는 클라이언트의 신원인증을 위해 리소스 서버의 엔드포인트에서 클라이언트 인증서 기반 HTTP 인증을 사용하며, 리소스 서버는 토큰에 의해 참조된 이름으로 해당 이름의 유효성을 입증하게 됨 ○ 위의 옵션과 동일하지만, 클라이언트는 리소스 서버에 대한 요청을 서명하기 위해		

	개인키를 사용함 (공개키는 토큰 내 포함되거나 요청에 따라 전달됨) o 그 대신, 인증서버는 클라이언트가 클라이언트의 토큰 사용을 인증하기 위한 Holder-of-Key 증거로 사용해야 할 토큰에 바인딩된 키를 발행해야 할 수 있음 리소스 서버는 인증서버로부터 직접 Secret을 획득하며, 그렇지 않을 경우 Secret은 토큰의 암호화된 섹션 내에 포함됨 그러한 방식으로, 리소스 서버는 클라이언트를 "알지" 못 하지만, 인증서버가 해당 클라이언트에 대한 토큰을 발행했는지 아닌지에 대한 유효성을 입증할 수 있음 ※ Holder-of-Key 상호인증을 이용하여 무결성과 기밀성 보호를 지니며 클라이언트 공개키와 인증정보를 전송하기 위해 서명된 SAML Assertion을 지닌 메시지를 보호하는 메커니즘
진단 방법	[진단기준] - 인증된 요청 사용 설정한 경우 양호 - 인증된 요청 사용 미설정된 경우 취약
비고	단기 적용(적용 시 개발자 및 운영자 협의)

15.4.3. 서명된 요청 사용 설정

분류	리소스 서버 설정 (적용 대상 : 리소스 서버)	중요도	상
항목명	서명된 요청 사용 설정		
항목 설명	만약 서명된 요청 사용 설정을 안 한 경우, 메시지 변경, 리플레이 공격 시도가 가능함		
설정 방법	1. 리소스 서버는 전송 수준 보안 수단을 변경하거나 그러한 수단을 보완하여 서명된 요청만 수용하도록 결정해야 할 수 있음 2. 모든 서명된 요청은 고유하게 식별 가능해야 하고, 리소스 서버에 의해 2번 처리되면 안 됨		
진단 방법	[진단기준] - 서명된 요청 사용 설정한 경우 양호 - 서명된 요청 사용 미설정된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

16. Spark

16.1. 접근 통제

16.1.1. 관리자 설정

분류	접근 통제	중요도	상
항목명	관리자 설정		
항목 설명	시스템에 접근할 수 있는 사용자나 그룹은 최소화하여 관리하여야 함		
설정 방법	설정파일에 아래의 노드에 필요한 관리자만 설정 spark.admin.acls - 관리자 계정 설정 spark.admin.acls.groups - 관리자 그룹 설정		
진단 방법	[진단기준] - 관리자 권한이 필요한 최소한의 사용자에게만 부여한 경우 양호 - 관리자 권한이 불필요하게 부여하고 있는 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

16.1.2. 사용자 접근 통제 설정

분류	접근 통제	중요도	상
항목명	사용자 접근 통제 설정		
항목 설명	시스템에 접근이 가능한 사용자는 별도로 지정하여 관리하여야 하며 비인가자의 접근을 차단하여야 한다.		
설정 방법	Spark 설정파일에 아래의 내용을 참조하여 작성 spark.acls.enable - 사용자 접근 ACL 활성화 spark.modify.acls - 접근 가능 사용자 계정 설정 spark.modify.acls.groups - 접근 가능 사용자 그룹 설정		
진단 방법	[진단기준] - 접근이 필요한 최소한의 사용자에게만 부여한 경우 양호 - 접근이 필요한 사용자를 별도로 지정하고 있지 않으면 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

16.2. 로그 설정

16.2.1. 로그 설정

분류	로그 설정	중요도	중
항목명	로그 설정		
항목 설명	로그를 설정하지 않으면, 공격 여부 파악, 공격자 사용 툴 파악, 공격자 위치 파악이 불가능하므로 반드시 로그를 설정해야 함.		
설정 방법	로그는 기본적으로 비활성화 상태이며 설정파일에 아래와 같이 설정하여야 활성화 됨 spark.eventLog.dir - 로그파일 디렉토리 디렉토리 권한을 drwxrwxrwx 설정		
진단 방법	[진단기준] - 로그 디렉토리 설정이 존재하는 경우 양호 - 로그가 비활성화 되어 있는 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

16.3. 암호화 설정

16.3.1. 일반 암호화 설정

분류	암호화 설정	중요도	권고
항목명	통신 암호화 설정		
항목 설명	암호화 설정을 수행하지 않는 경우 스니핑등의 취약점에 노출될 우려가 있음		
설정 방법	Spark 설정파일에 아래의 사항을 참조하여 설정 spark.network.crypto.enabled - 암호화 설정 활성화(기본 False) spark.network.crypto.keyLength - 암호키 길이 설정(기본 128) spark.network.crypto.keyFactoryAlgorithm - 암호화키 팩토리 알고리즘(기본 PBKDF2WithHmacSHA1) spark.network.crypto.config.* - commons.crypto 라이브러리 설정(기본 None) spark.network.crypto.saslFallback - 인증실패시 SALS로 폴백할지 여부 설정(기본 True) spark.authenticate.enableSaslEncryption - SALS 암호화 인증 설정 (기본 True) spark.network.sasl.serverAlwaysEncrypt - 비암호화 통신 비활성화 설정 (기본 False)		
진단 방법	권고 사항		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

16.3.2. 로컬 임시 파일 암호화 설정

분류	암호화 설정	중요도	권고
항목명	로컬 임시 파일 암호화 설정		
항목 설명	암호화 되어 있지 않은 파일은 비인가자가 접근하여 정보를 취득할 수 있는 우려가 있으므로 암호화 조치를 취해야 함		
설정 방법	spark.io.encryption.enabled - 로컬디스크 I/O 암호화 설정 (기본 False) spark.io.encryption.keySizeBits - I/O 키 암호화 길이 (기본 128) spark.io.encryption.keygen.algorithm - I/O 키 암호화 알고리즘 (기본 HmacSHA1)		
진단 방법	권고 사항		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		



16.4. 기타

16.4.1. HTTP 보호 설정

분류	기타	중요도	권고
항목명	HTTP 보호 설정		
설정 방법	1. XSS(크로스 사이트 스크립팅 보호 설정) spark.ui.xXssProtection 2. X-Content-Type-Options헤더가 nosniff로 설정(기본 True 설정) spark.ui.xContentTypeOptions.enabled 3. Timeout 설정(기본 False) spark.ui.strictTransportSecurity		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

16.4.2. Spark 기본 포트

분류	로그 설정	중요도	-
항목명	Spark 기본 포트		
설정 방법	독립형		
	Default Port	Purpose	설정 값
	8080	Web UI	spark.master.ui.port /SPARK_MASTER_WEBUI_PORT
	8081	Web UI	spark.worker.ui.port /SPARK_WORKER_WEBUI_PORT
	7077	Submit job to cluster / Join cluster	SPARK_MASTER_PORT
	(random)	Schedule executors	SPARK_WORKER_PORT
	All 클러스터 매니저		
	Default Port	Purpose	설정 값
	4040	Web UI	spark.ui.port
	18080	Web UI	spark.history.ui.port
	(random)	Connect to application / Notify executor state changes	spark.driver.port
	(random)	Block Manager port	spark.blockManager.port

17. Squid

17.1. 설정

17.1.1. 세션 타임아웃 설정

분류	설정	중요도	하
항목명	세션 타임아웃 설정		
항목 설명	세션 타임아웃을 너무 길게 설정하여 공격자가 세션을 사용 할 수 있어 세션 타임을 설정하여야 함		
설정 방법	/etc/squid/squid.conf 파일에서 세션 타임아웃 설정 pconn_timeout 시간(분) minute		
진단 방법	[진단기준] - 세션 타임아웃을 30분 미만으로 설정한 경우 양호 - 세션 타임아웃을 30분 미만으로 설정하지 않는 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

17.1.2. HTTP 헤더 허용 사이즈 설정

분류	설정	중요도	하
항목명	HTTP 헤더 허용 사이즈 설정		
항목 설명	응답 헤더 크기를 제한하면 persistent connections 와 같은 특정 버그가 발생하고 버퍼 오버플로우 또는 서비스 거부 공격이 발생할 수 있음.		
설정 방법	/etc/squid/squid.conf 파일에서 HTTP 헤더 최대 사이즈 옵션 설정 request_header_max_size 64KB		
진단 방법	[진단기준] - 헤더 최대 사이즈가 충분할 경우 양호 - 헤더 최대 사이즈가 충분하지 않을 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

17.1.3. 클라이언트 주소 숨김

분류	설정	중요도	중
항목명	클라이언트 주소 숨김		
항목 설명	HTTP 요청에 대한 응답 시에 헤더에 클라이언트의 이름, 버전 등의 정보를 제공하는 경우, 공격자가 해당 정보를 이용해 공격에 이용할 수 있음.		
설정 방법	/etc/squid/squid.conf 파일에서 클라이언트 주소 노출 설정 off forwarded_for off		
진단 방법	[진단기준] - 클라이언트 주소 노출 설정이 off로 되어있는 경우 양호 - 클라이언트 주소 노출 설정이 on로 되어있는 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

17.1.4. 알 수 없는 네임서버 무시

분류	설정	중요도	하
항목명	알 수 없는 네임서버 무시		
항목 설명	기본적으로 Squid는 보낸 사람과 동일한 IP 주소에서 DNS 응답을 받는지 확인함. 알 수 없는 nameserver로부터 DNS 캐시를 보호해야 함.		
설정 방법	/etc/squid/squid.conf 파일에서 알 수 없는 네임서버 무시 설정 ignore_unknown_nameservers on		
진단 방법	[진단기준] - ignore_unknown_nameservers 이 on으로 되어있는 경우 양호 - ignore_unknown_nameservers 이 off으로 되어있는 경우 취약		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

17.1.5. 클라이언트 라이프타임 설정

분류	설정	중요도	하
항목명	클라이언트 라이프타임 설정		
항목 설명	클라이언트가 허용되는 최대 시간을 설정하여 올바르게 종료하지 않고 사라지는 원격 클라이언트의 CLOSE_WAIT 상태에서 많은 소켓을 연결하지 못하도록 Cache를 보호함.		
설정 방법	/etc/squid/squid.conf 파일에서 client_lifetime 설정(1일이내) client_lifetime 1 day		
진단 방법	[진단기준] - client_lifetime 설정이 1일 이내인 경우 양호 - client_lifetime 설정이 1일 초과인 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

17.1.6. 로그 설정

분류	설정	중요도	상
항목명	로그 설정		
항목 설명	SQUID 로그 포맷 및 로그보관기간 설정		
설정 방법	/etc/squid/squid.conf 파일내 설정 1. Logformat 설정 logformat combined combined내 포함되어 있는 항목 %>a %[ui %[un [%t] "%rm %ru HTTP/%rv" %>Hs %<st "%{Referer}>h" "%{User-Agent}>h" %Ss:%Sh 2. 로그 디렉토리 권한 설정 /etc/squid/squid.conf 파일에서 로그 디렉토리 확인 access log : cache_access_log cache log : cache_log store log : cache_store_log		
진단 방법	[진단기준] - logformat이 combined, 주석처리가 되어 있지 않고 로그 디렉토리 및 파일의 권한이 400 이하인 경우 양호 - logformat이 combined가 아니고 주석처리가 되어 있거나 로그 디렉토리 및 파일의 권한이 400 이상인 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

17.2. 계정 관리

17.2.1. 사용자 권한 설정

분류	계정 관리	중요도	상
항목명	사용자 권한 설정		
항목 설명	사용자 별로 권한을 구분하여, 주어진 권한 이외의 행동을 통제하는 설정이 필요함.		
설정 방법	/etc/squid/squid.conf 파일에서 아래 항목을 nobody 로 설정 • cache_effective_user squid_user • cache_effective_group squid_group		
진단 방법	[진단기준] - 사용자 권한이 적절하게 부여된 경우 양호 - 사용자 권한이 부적절하게 부여된 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

17.2.2. 사용자 인증 설정

분류	계정 관리	중요도	상
항목명	사용자 인증 설정		
항목 설명	사용자 인증을 통해 중요 파일 및 데이터 접근은 허가된 사용자만 가능하도록 제한함.		
설정 방법	/etc/squid/squid.conf 파일에서 아래 항목으로 수정 <pre>acl trusted_users proxy_auth REQUIRED http_access allow mynetwork trusted_users auth_param digest /usr/lib/squid3/nsc_auth /etc/squid3/i_just_put_here_the_passwd auth_param digest children 5 auth_param digest realm Squid proxy-caching web server</pre>		
진단 방법	[진단기준] - 사용자 인증 설정이 되어있는 경우 양호 - 사용자 인증 설정이 되어있지 않는 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

17.3. 파일 및 디렉터리 관리

17.3.1. 중요 디렉터리 권한 설정

분류	파일 및 디렉터리 관리	중요도	상
항목명	중요 디렉터리 권한 설정		
항목 설명	중요 디렉터리 권한 설정을 통해 비인가자의 중요 파일 접근을 제한함.		
설정 방법	아래 디렉터리 권한을 770으로 설정 /var/log/squid /etc/squid/		
진단 방법	[진단기준] - 중요 디렉터리 권한 설정이 770으로 되어있는 경우 양호 - 중요 디렉터리 권한 설정이 770으로 되어있지 않는 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

17.4. 서비스 관리

17.4.1. FTP 설정

분류	서비스 관리	중요도	하
항목명	FTP 설정		
항목 설명	FTP 서비스는 아이디, 비밀번호가 암호화 되지 않은 채로 전송되어 스니핑이 가능함		
설정 방법	/etc/squid/squid.conf 파일에서 FTP 설정 파일 수정 ftp_user my_email@adress.com ftp_passive on ftp_sanitycheck on		
진단 방법	[진단기준] - 적절한 FTP 설정이 되어있는 경우 양호 - 부적절한 FTP 설정이 되어있는 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

17.4.2. SNMP 설정

분류	서비스 관리	중요도	중
항목명	SNMP 설정		
항목 설명	SNMP 서비스로 인하여 시스템의 주요정보 유출 및 정보의 불법수정이 발생할 수 있음		
설정 방법	/etc/squid/squid.conf 파일에서 SNMP 서비스가 불필요한 경우 비활성화 snmp_port 0 snmp_access deny all		
진단 방법	[진단기준] - SNMP 서비스를 사용하고 있지 않는 경우 양호 - SNMP 서비스를 사용하고 있는 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

17.5. 접근제어

17.5.1. http 접근 포트 설정

분류	접근제어	중요도	하
항목명	http 접근 포트 설정		
항목 설명	Default 포트인 3128은 공격자가 유추 할 수 있으므로, 유추 할 수 없는 포트로 포트 번호를 지정하여 사용		
설정 방법	/etc/squid/squid.conf 파일에서 아래 항목을 임의의 포트로 변경 • http_port : port_num		
진단 방법	[진단기준] - default port를 사용하고 있지 않는 경우 양호 - default port를 사용하고 있는 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

17.5.2. ACL 설정

분류	접근제어	중요도	중
항목명	ACL 설정		
항목 설명	ACL 설정을 통해 접속 대상 호스트를 지정하여 접근이 가능한 IP를 제한하여 비인가자의 접근을 차단하여야 함.		
설정 방법	/etc/squid/squid.conf 파일에서 ACL 설정 - 신뢰 IP설정		

	<pre>acl mynetwork src 192.168.1.0/24 http_access allow mynetwork</pre> - 신뢰포트 설정 <pre>acl trusted_ports port 21 80 443 http_access deny !trusted_ports</pre> - 허용 외 차단 설정 <pre>http_access deny all</pre>
진단 방법	[진단기준] - 적절한 ACL 설정이 되어 있는 경우 양호 - 적절한 ACL 설정이 되어 있지 않는 경우 취약
비고	중기 적용(적용 시 개발자 및 운영자 협의)

17.5.3. 관리콘솔 관리

분류	접근제어	중요도	중
항목명	관리콘솔 관리		
항목 설명	관리콘솔 접근 제한 설정을 통해 비인가자의 서비스 중지 및 네트워크 레이아웃 등 중요 정보 접근을 차단하여야 함.		
설정 방법	<pre>/etc/squid/squid.conf 파일에서 관리콘솔 설정 수정 acl manager proto cache_object acl localhost src 127.0.0.1/255.255.255.255 acl webserver src 192.168.1.X/255.255.255.255 # webserver IP http_access allow manager localhost http_access allow manager webserver http_access deny manager</pre>		
진단 방법	[진단기준] - 관리콘솔 접근제어가 적절하게 되어 있는 경우 양호 - 관리콘솔 접근제어가 적절하게 되어 있지 않는 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

18. Storm

18.1. 설정

18.1.1. SSL 구성

분류	설정	중요도	권고
항목명	SSL 구성		
항목 설명	Apache Stormd에서 SSL사용을 위한 설정		
설정 방법	UI : (storm.yaml에서 설정) 1. ui.https.port 2. ui.https.keystore.type (예 : "jks") 3. ui.https.keystore.path (예 : "/etc/ssl/storm_keystore.jks") 4. ui.https.keystore.password (키 저장소 비밀번호) 5. ui.https.key.password (개인 키 암호) (선택적 구성) 6. ui.https.truststore.path (예 : "/etc/ssl/storm_truststore.jks") 7. ui.https.truststore.password (트러스트 스토어 암호) 8. ui.https.truststore.type (예 "jks") (사용자가 양방향 인증을 설정하려는 경우) 9. ui.https.want.client.auth (true 서버가 클라이언트 인증서 인증을 요청했지만 인증이 제공되지 않은 경우 연결을 유지하는 경우) 10. ui.https.need.client.auth (true 서버로 설정하면 클라이언트가 인증을 제공해야 함) DRPC : 1. drpc.https.port 2. drpc.https.keystore.type (예 : "jks") 3. drpc.https.keystore.path (예 : "/etc/ssl/storm_keystore.jks") 4. drpc.https.keystore.password (키 저장소 비밀번호) 5. drpc.https.key.password (개인 키 암호) (선택적 구성) 6. drpc.https.truststore.path (예 "/etc/ssl/storm_truststore.jks") 7. drpc.https.truststore.password (트러스트 스토어 암호) 8. drpc.https.truststore.type (예 "jks") (사용자가 양방향 인증을 설정하려는 경우) 9. drpc.https.want.client.auth (true 서버가 클라이언트 인증서 인증을 요청했지만 인증이 제공되지 않은 경우 연결을 유지하는 경우) 10. drpc.https.need.client.auth (true 서버로 설정하면 클라이언트가 인증을 제공해야 함)		
진단 방법	[진단기준] - 암호화 설정이 되어 있는 경우 양호 - 암호화 설정이 되어 있지 않을 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

18.1.2. Create Headless Principals and keytabs

분류	설정	중요도	중
항목명	Create Headless Principals and keytabs		
항목 설명	각 주키퍼 서버, Nimbus 및 DRPC 서버에는 규칙에 따라 실행될 호스트의 FQDN이 포함 된 서비스 원칙이 필요, 수퍼바이저와 UI에는 실행 주체가 필요하지만 발신 연결이기 때문에 서비스 주체 일 필요는 없음 다음은 kerberos principals and keytabs를 설정하는 방법		
설정 방법	<pre># Zookeeper (Will need one of these for each box in teh Zk ensamble) sudo kadmin.local -q 'addprinc zookeeper/zk1.example.com@STORM.EXAMPLE.COM' sudo kadmin.local -q "ktadd -k /tmp/zk.keytab zookeeper/zk1.example.com@STORM.EXAMPLE.COM" # Nimbus and DRPC sudo kadmin.local -q 'addprinc storm/storm.example.com@STORM.EXAMPLE.COM' sudo kadmin.local -q "ktadd -k /tmp/storm.keytab storm/storm.example.com@STORM.EXAMPLE.COM" # All UI logviewer and Supervisors sudo kadmin.local -q 'addprinc storm@STORM.EXAMPLE.COM' sudo kadmin.local -q "ktadd -k /tmp/storm.keytab storm@STORM.EXAMPLE.COM"</pre>		
진단 방법	[진단기준] - 설정이 되어 있는 경우 양호 - 설정이 되어 있지 않을 경우 취약		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

19. MongoDB(NoSQL)

19.1. 계정 관리

19.1.1. MongoDB null 패스워드 점검

분류	계정 관리	중요도	상
항목명	MongoDB null 패스워드 점검		
항목 설명	NoSQL은 기본적으로 계정 없이 사용이 가능하여 누구나 DB에 접근 가능하므로 허가된 사용자만 접근 할 수 있도록 서비스 사용 시 계정 및 패스워드를 사용해야 함		
설정 방법	1. 패스워드가 취약하게 설정된 경우 패스워드를 다음 기준을 준수하여 변경 [패스워드 설정 기준] - 특수문자, 숫자, 문자 혼합 설정 - 영 대/소문자, 숫자, 특수문자에 대해 3종류 조합 10자 이상 설정 - 추측 가능한 패스워드 설정 금지 - 연속되는 숫자, 문자 사용 금지 [MongoDB 계정 패스워드 설정] 1) 관리자 계정 추가 <pre>use admin db.createUser({ user: "adminID", pwd: "adminPassword", roles: [{ role: "userAdminAnyDatabase", db: "admin" }] })</pre> 2) 인증모드 구동 <pre>\$) mongod --auth</pre>		
진단 기준	양호 - MongoDB 접근 후 인증 없이 admin 모드로 변경 불가능한 경우 취약 - MongoDB 접근 후 인증 없이 admin 모드로 변경 가능한 경우		
진단 방법	[진단예시] <pre># /[설치dir]/bin/mongo > use admin // 취약 - admin 모드 변경 시 인증 없음 # /[설치dir]/bin/mongo > use admin > db.auth("adminID", "adminPassword") // 양호</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

19.1.2. 비밀번호 복잡도 설정

분류	계정 관리	중요도	중
항목명	비밀번호 복잡도 설정		
항목 설명	비밀번호 복잡도가 설정되지 않은 경우 Brute force 공격을 통하여 비밀번호를 쉽게 획득할 위험이 존재함		
설정 방법	1. 관리자 비밀번호가 아래 기준에 맞게 설정되었는지 확인 [비밀번호 설정 기준] - 영문 대/소문자, 숫자, 특수문자 혼합 설정 - 2종류 조합으로 10자리 이상, 3종류 조합으로 8자 이상 설정 - 추측 가능한 비밀번호 설정 금지 - 연속되는 문자, 숫자 사용 금지		
진단 기준	양호 - 설정된 비밀번호가 설정 기준을 준수하고 있을 경우 취약 - 설정된 비밀번호가 설정 기준을 준수하지 않은 경우		
진단 방법	[진단예시] 담당자 인터뷰		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

19.2. 권한 관리

19.2.1. 개발 및 운영 시스템 분리 사용

분류	권한 관리	중요도	하
항목명	개발 및 운영 시스템 분리 사용		
항목 설명	개발용과 운영용이 같이 사용되는 경우 개발 시 취약하게 설정된 것으로 인하여 외부에 취약점이 노출될 위험이 있으므로 개발 시스템과 운영 시스템은 물리적으로 분리해야 함		
설정 방법	1. 개발 시스템과 운영 시스템의 분리는 회사 정책으로 정의 2. 개발 시스템과 운영 시스템은 하드웨어적으로 분리되어야 하며 원칙적으로 개발자와 운영자는 분리 3. 개발 시스템과 운영 시스템에는 Link가 설정되어 있지 않아야 하며, 운영 시스템의 데이터는 개발 시스템으로 이전 시 중요 데이터 삭제 등과 같은 통제 절차를 거쳐야 하며 이는 검증 필요 4. 개발자의 운영 시스템 접근은 제한		
진단 기준	양호 - 개발 시스템과 운영시스템이 분리되어 사용하는 경우 취약 - 개발 시스템과 운영시스템이 분리되어 있지 않은 상태에서 사용하는 경우		
진단 방법	[진단예시] 담당자 인터뷰		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

19.2.2. root 권한으로 서버 구동 제한

분류	권한 관리	중요도	상
항목명	root 권한으로 서버 구동 제한		
항목 설명	유닉스 root 사용자 권한으로 MongoDB 서비스를 구동하지 말고 전용 daemon 계정으로 구동해야 함		
설정 방법	1. MongoDB 관리 용도의 일반 계정을 생성하여 처리하는 것이 보다 안전함 MongoDB 서비스가 root로 구동되어 있는 경우 MongoDB 전용 계정 또는 일반 계정으로 서비스를 재시작 해야 함		
진단 기준	양호 - MongoDB 프로세스가 root 계정 소유로 구동되고 있지 않을 경우 취약 - MongoDB 프로세스가 root 계정 소유로 구동되고 있을 경우		
진단 방법	[진단예시] # ps -aux grep mongo		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

19.3. DBMS 보안설정

19.3.1. 백업 관리

분류	DBMS 보안설정	중요도	하
항목명	백업 관리		
항목 설명	MongoDB는 메모리 기반의 DB이기 때문에 전원 유실 시 데이터가 사라지므로 주기적인 백업이 수행되어야 하며 특히 DBMS의 유지보수 및 Upgrade 작업에는 전체 Dump를 실시하여 장애 및 외부 침입 등에 대한 변조가 발생할 경우를 대비해야 함		
설정 방법	1. 백업 정책을 바탕으로 주기적인 백업 절차를 수립 2. Dump 복사본을 회사 외부의 안전한 위치에 보관 3. DBMS의 유지보수 및 Upgrade 작업 시에는 전체 Dump 절차를 수립 4. 개인정보처리시스템의 경우 접속 기록이 위.변조 되지 않도록 별도의 물리적인 저장 장치에 보관하여야 하며 정기적 백업을 수행		
진단 기준	양호 - 주기적으로 백업되고 있는 경우 취약 - 백업을 하지 않는 경우		
진단 방법	[진단 예시] 담당자 인터뷰		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

19.3.2. DB 접속 IP 통제

분류	DBMS 보안 설정	중요도	하
항목명	DB 접속 IP 통제		
항목 설명	임의의 사용자에게 의한 원격 접속을 차단하기 위해 IP 접근 제한을 설정 함		
설정 방법	1. mongodb.conf 파일 내 bindIp 설정을 통해 설정 가능 ... bindIp 192.168.1.100, 192.168.0.101 // mongodb 서버에 접근 가능한 IP를 지정 ...		
진단 기준	양호 - bindIp 설정에 IP가 지정되어 있는 경우 취약 - bindIp 설정에 IP가 지정되어 있지 않거나 설정값이 0.0.0.0인 경우		
진단 방법	[진단 예시] # cat /[설치dir]/conf/mongodb.conf grep -C3 bindIp		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

13.3.3. 로그 저장 주기

분류	DBMS 보안 설정		중요도	상
항목명	로그 저장 주기			
항목 설명	'정보통신망이용촉진및정보보호등에관한법률', '개인정보보호법', '회사사규' 등에 따라 접속 기록은 정해진 최소 보유 기간 동안 보관해야하며, 담당자는 접속 기록을 정기적으로 백업 · 확인 · 감독하여야 함			
설정 방법	1. 접속 기록 보유 기간은 사업 환경에 따라 조정 할 수 있으나, '정보통신망이용촉진 및 정보보호등에관한법률', '개인정보보호법', '회사사규' 등에 따라 최소 아래 기간 이상은 보관 해야함			
	1) 사용자접속기록			
	사용자로그인/로그아웃/정보변경 등		6개월이상	
	2) 개인정보취급자의개인정보처리시스템접속기록			
	정보주체 식별정보/개인정보취급자 식별정보/ 접속일시/접속지 정보/ 부여된 권한 유형에 따른 수행업무 등		2년이상	
설정 방법	3) 개인정보취급자권한변경기록			
	개인정보취급자권한생성/변경/삭제 등		5년이상	
	2. 담당자는 접속 기록을 월 1회 이상 정기적으로 확인·감독하여, 접속과 관련 된 오류 및 부정행위가 발생하거나 예상 되는 경우 즉각적인 보고 조치가 되도록 해야함			
	3. 접속 기록이 위·변조 되지 않도록 별도의 물리적인 저장 장치에 보관하여야 하며 정기적인 백업을 수행 해야 함			
	※ 수정이 가능해야 할 경우, 위·변조 여부를 확인 할 수 있도록 별도 보호조치 - 위·변조 여부를 확인할 수 있는 정보(HMAC값 또는 전자서명값 등)는 별도의 HDD 또는 관리대장에 보관 하는 방법으로 관리			
설정 방법	4. mongodb.conf 설정 파일에서 LOG 파일 및 경로 지정			
	※ mongoDB의 환경 파일은 사용자가 생성 하여 파일명이 다를 수 있음			
진단 기준	양호 - 접속 기록 보유 및 백업 등이 안전하게 관리되고 있는 경우 취약 - 접속 기록 보유 및 백업 등이 안전하게 관리되고 있지 않은 경우			
진단 방법	[진단 예시] # cat /[설치dir]/conf/mongodb.conf grep logpath 로그 관리 및 저장 주기에 대한 담당자 인터뷰			
비고	중기 적용(적용 시 개발자 및 운영자 협의)			
기반시설 기준항목	-			

19.3.4. 로그 레벨 설정

분류	DBMS 보안 설정	중요도	상
항목명	로그 레벨 설정		
항목 설명	로그 레벨을 낮게 설정 할 경우 공격 정보에 대한 파악이 어려울 수 있으므로 파악이 가능한 최소한의 레벨을 설정해야 함		
설정 방법	1. <code>mongodb.conf</code> 설정 파일 내 <code>quiet</code> 값에 <code>false</code> 설정 환경설정 파일 내 log 파일 위치 확인 및 해당 경로의 log 설정 확인 [설정 예시] <code>// quiet=true</code> 일 경우 critical 이벤트만 기록 <code># cat /[설치dir]/conf/mongodb.conf grep quiet</code> ※ MongoDB의 환경 파일은 사용자가 생성하여 파일명이 다를 수 있음		
진단 기준	양호 - quiet 설정 값이 true로 설정된 경우 취약 - quiet 설정 값이 false로 설정된 경우		
진단 방법	[진단 예시] <code># cat /[설치dir]/conf/mongodb.conf grep quiet</code>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

19.3.4. DBMS 서버 보안 연결

분류	DBMS 보안 설정	중요도	상
항목명	DBMS 서버 보안 연결		
항목 설명	평문 또는 취약한 버전의 SSL 접속을 통한 DBMS 서버 connect를 허용할 경우 스니핑을 통해 로그인 계정 등 주요 정보가 노출 될 위험이 있음		
설정 방법	1. 실행중인 DBMS 프로세스의 실행 상태 connect 시 SSL 접속 설정 확인 MongoDB command 를 통해 주요 정보를 전송하므로 SSL 연결을 통해 서버와 통신하도록 하여 주요 정보 노출 방지		
진단 기준	양호 - sslMode 및 적절한 sslProtocol을 사용 중인 경우 취약 - sslMode 및 적절한 sslProtocol을 사용 중이지 않은 경우		
진단 방법	[진단 예시] 1) 프로세스 실행 상태 확인 <pre># ps -ef grep mongo mongod --sslMode requireSSL --sslDisabledProtocols TLS1_0,TLS1_1 --sslPEMKeyFile /etc/ssl/mongodb.pem --sslCAFile /etc/ssl/ca.pem</pre> 또는 2) 설정 파일 내용 확인 <pre># cat /[설치dir]/conf/mongodb.conf grep -C5 requireSSL net: ssl: mode: requireSSL PEMKeyFile: /etc/ssl/mongodb.pem CAFile: /etc/ssl/ca.pem disabledProtocols: TLS1_0,TLS1_1</pre>		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

19.4. 환경 파일 점검

19.4.1. MongoDB 환경설정 파일 접근 제한

분류	환경 파일 점검	중요도	중
항목명	MongoDB 환경설정 파일 접근 제한		
항목	MongoDB 환경설정 파일(mongodb.conf) 소유자 외 타 사용자가 읽기, 쓰기 가능한 경우 서버의		

설명	주요 정보가 노출될 수 있음
설정 방법	1. 환경설정 파일(mongodb.conf)의 접근 권한을 다음과 같이 설정 환경설정 파일에 대한 보호를 위하여 접근 권한 설정을 디렉터리는 700, 파일은 600으로 설정 [접근 권한 설정] <pre># chmod 600 /[설치dir]/conf/mongodb.conf</pre> ※ MongoDB의 환경 파일은 사용자가 생성하여 파일명이 다를 수 있음
진단 기준	양호 - 환경설정 파일 접근권한은 디렉터리 700, 파일 600으로 설정되어 있는 경우 취약 - 환경설정 파일 접근권한이 디렉터리 700, 파일 600 초과로 설정되어 있는 경우
진단 방법	[진단 예시] <pre># ls -ald /[설치dir]/conf</pre> <pre># ls -al /[설치dir]/conf/mogodb.conf</pre>
비고	단기 적용(적용 시 개발자 및 운영자 협의)
기반시설 기준항목	-

19.4.2. .dbshell 파일 접근 제한

분류	환경 파일 점검	중요도	중
항목명	.dbshell 파일 접근 제한		
항목 설명	.dbshell 파일에는 명령 실행에 대한 주요 정보가 평문으로 노출될 수 있어 해당 파일을 허가 받지 않은 사용자가 읽는 경우 주요 정보를 획득할 수 있으므로 파일의 접근을 제한해야 함 (CVE-2016-6494)		
설정 방법	1. .dbshell 파일의 접근 권한을 600 또는 640 으로 설정 (타 사용자 모든 권한 제거) <pre># ls -al ~/.dbshell</pre> <pre># chmod 600 .dbshell</pre>		
진단 기준	양호 - 접근권한이 600 또는 640 이하로 설정되어 있는 경우 취약 - 접근권한이 600 또는 640 초과로 설정되어 있는 경우		
진단 방법	[진단 예시] <pre># ls -al ~/.dbshell</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

19.4.3. Log 파일 접근 제한

분류	환경 파일 점검	중요도	하
항목명	Log 파일 접근 제한		
항목 설명	Log 파일에는 중요 내용이 포함되어 있음. 이를 활용하여 침해 사고 시 분석 자료로 사용하고 있음. Log 파일이 비인가 자에게 읽히거나 쓰여지는 경우 로그 파일의 정보 노출 및 변조가 발생할 수 있으므로 접근을 제한할 필요가 있음.		
설정 방법	1. Log 디렉터리 및 파일의 접근 권한을 다음과 같이 설정 ■ Unix 계열 Log 디렉터리는 750, 파일은 640 이하로 접근 권한 설정 <pre># chmod 750 [로그 디렉터리]</pre> <pre># chmod 640 [로그 파일]</pre> [Log 파일 위치 및 찾는 방법] <pre># cat /[설치dir]/conf/mongodb.conf grep logpath</pre>		
진단 기준	양호 - 접근권한이 디렉터리 750, 파일 640 이하로 설정되어 있는 경우 취약 - 접근권한이 디렉터리 750, 파일 640 초과로 설정되어 있는 경우		
진단 방법	[진단 예시] <pre># ls -ald [로그 디렉터리]</pre> <pre># ls -al [로그 디렉터리]/[로그 파일]</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

19.5. 보안 패치

19.5.1. 보안 패치 적용

분류	보안 패치	중요도	상										
항목명	보안 패치 적용												
항목 설명	버그로 인한 침해 사고 발생 가능하므로 주기적으로 최신 패치를 적용하여 취약점을 제거 해야 함												
설정 방법	1. 데이터베이스에 대한 최신의 버전을 확인 후 업그레이드 및 패치 수행 2. 원격 Exploit 취약점, 제로데이 취약점은 즉시 패치												
	〈Latest Release 2017.03.10〉												
	<table><tr><td>Version</td><td>Last Version</td></tr><tr><td>2.4.x</td><td>2.4.14-4.el6</td></tr><tr><td>2.6.x</td><td>2.6.12-3.el7 (지원종료)</td></tr><tr><td>3.4.x</td><td>3.4.2</td></tr><tr><td>3.5.x</td><td>3.5.1</td></tr></table>			Version	Last Version	2.4.x	2.4.14-4.el6	2.6.x	2.6.12-3.el7 (지원종료)	3.4.x	3.4.2	3.5.x	3.5.1
	Version	Last Version											
	2.4.x	2.4.14-4.el6											
2.6.x	2.6.12-3.el7 (지원종료)												
3.4.x	3.4.2												
3.5.x	3.5.1												
※ 신규 보안가이드라인 기준으로, 2.x 버전 관련 내용은 CVE 취약점만을 다룸													
- MongoDB 3.2 이하 버전 (사용자 인증 활성화한 경우 영향 없음) MongoDB 관리용 포트(27107)의 인증 설정이 부재하여 공격자가 접근할 가능성 존재 이에, MongoDB 소프트웨어 최신버전 업데이트를 수행하거나 기존버전으로 사용 시 mongod.conf 설정파일의 auth = true을 추가하여 사용자 인증을 활성화 - 참고 자료 : https://www.mongodb.com/download-center#community ※ 서비스 및 운영상의 영향도를 확인하시어 설정하여 적용해야 합니다.													
진단 기준	양호 - 최신의 버전 적용한 경우 취약 - 최신의 버전 적용하지 않은 경우												
진단 방법	[진단 예시] # /[설치dir]/bin/mongod --version												
비고	중기 적용(적용 시 개발자 및 운영자 협의)												
기반시설 기준항목	-												

20. MySQL

20.1. 계정 관리

20.1.1. 불필요한 계정 확인

분류	계정 관리	중요도	하
항목명	불필요한 계정 확인		
항목 설명	데이터베이스의 계정 중 실질적으로 업무에 사용하지 않은 불필요한 계정들이 있는 경우 악의적인 공격자가 쉽게 데이터베이스에 접속하여 데이터를 열람, 삭제, 수정 등을 할 위험이 존재함		
설정 방법	1. 현재 DBMS에서 사용하는 계정의 용도를 파악하여, 불필요한 계정을 삭제, 그리고 user, password 가 null 인 항목이 존재하는 경우도 해당 필드를 삭제 [계정 삭제] mysql> Delete from user where user='삭제할 계정';		
진단 기준	양호 - 불필요한 계정이 없는 경우 취약 - 불필요한 계정이 존재할 경우		
진단 방법	[진단예시] mysql> select host, user from mysql.user where user!='root';		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	D-01, D-02		

20.1.2. 패스워드 복잡도 설정

분류	계정 관리	중요도	중								
항목명	패스워드 복잡도 설정										
항목 설명	패스워드 복잡도가 설정되지 않은 경우 Brute force 공격을 통하여 패스워드를 쉽게 획득할 위험이 존재함.										
설정 방법	1. 영문 대/소문자, 숫자, 특수문자를 혼합하여 3종류 이상을 조합하여 최소 9자리 이상 설정 (단, v5.6 이상에서 validate_password Plugin 사용시 설정) [설정 옵션 및 설정값] - Validate_password_length : 9자리 이상 (전체 길이) - Validate_password_mixed_case_count : 1자리 이상 (소문자, 대문자 최소 개수) - Validate_password_number_count : 1자리 이상 (숫자) - Validate_password_policy : MEDIUM (패스워드 정책) - Validate_password_special_char_count : 0 또는 1자리 이상 (특수문자) <table><tr><th>Polic</th><th>Tests Performed</th></tr><tr><td>0 or LOW</td><td>Length</td></tr><tr><td>1 or MEDIUM</td><td>Length; numeric, lowercase/uppercase, and special characters</td></tr><tr><td>2 or STRONG</td><td>Length; numeric, lowercase/uppercase, and special characters; dictionary file</td></tr></table>			Polic	Tests Performed	0 or LOW	Length	1 or MEDIUM	Length; numeric, lowercase/uppercase, and special characters	2 or STRONG	Length; numeric, lowercase/uppercase, and special characters; dictionary file
Polic	Tests Performed										
0 or LOW	Length										
1 or MEDIUM	Length; numeric, lowercase/uppercase, and special characters										
2 or STRONG	Length; numeric, lowercase/uppercase, and special characters; dictionary file										
진단 기준	양호 - 패스워드 복잡도를 만족하도록 설정되어 있는 경우 취약 - 패스워드 복잡도를 만족하도록 설정되어 있지 않은 경우										
진단 방법	[진단예시] mysql> select @@version mysql> show global variables like '%vali%'										
비고	단기 적용(적용 시 개발자 및 운영자 협의)										
기반시설 기준항목	D-03										

20.1.3. root null 패스워드 점검

분류	계정 관리	중요도	상
항목명	root null 패스워드 점검		
항목 설명	root 계정의 패스워드가 default 설정 값이 null을 사용할 경우, 시스템에 접근한 임의의 모든 사용자가 root 권한으로 접속하여 mysql의 모든 작업을 할 수 있어 mysql DB에 저장된 모든 정보가 유출 되는 등 침해사고를 일으킬 위험이 있음.		
설정 방법	1. 패스워드가 취약하게 설정된 경우 패스워드를 다음 기준을 준수하여 변경 [패스워드 설정 기준] - 특수문자, 숫자, 문자 혼합 설정 - 영 대/소문자, 숫자, 특수문자에 대해 3종류 조합 9자 이상 설정 - 추측 가능한 패스워드 설정 금지 - 연속되는 숫자, 문자 사용 금지 [root 계정 패스워드 설정] <pre>mysql> use mysql mysql> update user set password=password('new password') where user='root'; mysql> flush privileges; 또는 mysql> set password for root=password('new password');</pre>		
진단 기준	양호 - 패스워드가 안전하게 설정되어 있는 경우 취약 - 패스워드가 null 이거나 취약하게 설정되어 있는 경우		
진단 방법	[진단예시] <pre>mysql> select host, user, password from mysql.user where user='root';</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	D-01		

20.1.4. 취약한 패스워드 사용 점검

분류	계정 관리	중요도	상										
항목명	취약한 패스워드 사용 점검												
항목 설명	취약한 패스워드 사용시 무차별 대입 공격 등에 의해 쉽게 패스워드가 노출되어 계정 도용 등의 위험이 존재하므로 패스워드 요구조건을 반영하여 사용자가 안전한 패스워드를 생성 및 관리할 수 있도록 설정규칙을 제공해야 함.												
설정 방법	[패스워드 설정규칙 적용] 패스워드 설정규칙에 맞추어 패스워드를 설정할 수 있도록 시스템 차원에서 기능 제공												
	<table><tr><th>구분</th><th>공통 기준</th></tr><tr><td>패스워드 길이/복잡성</td><td>9자리 이상/3종류 이상</td></tr><tr><td>변경 주기</td><td>3개월/1개월(중요시스템)</td></tr><tr><td>재사용 금지</td><td>직전 1개 패스워드</td></tr><tr><td>잠금</td><td>10회 실패 시</td></tr></table>			구분	공통 기준	패스워드 길이/복잡성	9자리 이상/3종류 이상	변경 주기	3개월/1개월(중요시스템)	재사용 금지	직전 1개 패스워드	잠금	10회 실패 시
	구분	공통 기준											
	패스워드 길이/복잡성	9자리 이상/3종류 이상											
	변경 주기	3개월/1개월(중요시스템)											
	재사용 금지	직전 1개 패스워드											
	잠금	10회 실패 시											
	1) 패스워드는 아래의 4가지 문자 종류 중 2종류 이상을 조합하여 최소 10자리 이상 또는 3종류 이상을 조합하여 최소 8자리 이상의 길이로 구성												
	(1) 영문 대문자 (26개)												
	(2) 영문 소문자 (26개)												
(3) 숫자 (10개)													
(4) 특수문자 (32개)													
2) 패스워드는 비인가자에 의한 추측이 어렵도록 다음의 사항을 반영하여 설계													
(1) Null 패스워드 사용 금지													
(2) 문자 또는 숫자만으로 구성 금지													
(3) 사용자 ID와 동일한 패스워드 금지													
(4) 연속적인 문자/숫자(예. 1111, 1234, abcd) 사용 금지													
(5) 주기성 패스워드 재사용 금지													
(6) 전화번호, 생일같이 추측하기 쉬운 개인정보를 패스워드로 사용 금지													
3) 초기 패스워드는 사용자에게 부여 후 최초 접속 시 즉시 변경되도록 설계													
4) 10회 이상의 연속적인 패스워드 입력 실패 시 해당 사용자 ID는 사용권한이 일시 중지되도록 설계													
5) 패스워드는 최대 3개월, 업무 중요도에 따라 1개월 주기로 변경													
6) 패스워드 변경 시 직전 1개와 동일한 패스워드 사용금지													
7) 패스워드는 마스킹 처리 등을 통해 화면상에서 읽을 수 없는 형태로 표시													
8) 패키지 등을 도입한 경우 패키지의 기본 기능에서 위의 항목이 제공되지 않는 경우 보안운영자에게 해당 내용 문의 및 보안성 검토 후 요구조건 반영													
[패스워드 관리 적용] 패스워드 신규 적용 및 초기화 시 설정규칙에 맞추어 관리하고, 저장 시에는 일방향 암호 알고리즘을 통한 암호화 처리(One-Way Encryption) 패스워드 분실로 인한 신규 패스워드 발급 절차 및 클리핑 레벨 적용 - 패스워드 분실 시 패스워드를 초기화한 후 안전한 전송수단을 통해 패스워드 제공 - 패스워드 초기화 처리 후 로그인시 패스워드 변경을 유도 - 사용자 식별/인증 실패 시 계정 잠금 및 접속차단 기능 적용													

	- 계정 잠금 해제를 위한 절차 적용 [패스워드 변경기능 구현] 관리자에 의한 변경과 사용자가 스스로 패스워드를 변경할 수 있는 기능 제공 사용자로부터 패스워드 변경 요청이 있을 경우, 사용자 신원 확인이 완료된 후 패스워드 변경될 수 있도록 설정 [패스워드 설정] mysql> use mysql mysql> update user set password=password('new password') where user='user name'; mysql> flush privileges; 또는 mysql> set password for 'user name'@'%'=password('new password') mysql> flush privileges;
진단 기준	양호 - 패스워드가 안전하게 설정되어 있는 경우 취약 - 패스워드가 USER명과 같거나 취약하게 설정되어 있는 경우
진단 방법	[진단예시] mysql> select host, user, user as password from mysql.user where password=password(user); mysql> select host, user, password from mysql.user where password='' and user!='root';
비고	단기 적용(적용 시 개발자 및 운영자 협의)
기반시설 기준항목	D-03, D-05

20.1.5. Anonymous 계정 확인

분류	계정 관리	중요도	하
항목명	Anonymous 계정 확인		
항목 설명	Anonymous 계정이 존재하고 패스워드가 설정되어 있지 않은 경우 DB에 임의 접속이 가능함. 주어진 권한이 제한적이지만 위험의 소지가 있으므로 제거해야 함.		
설정 방법	1. anonymous 계정 삭제 <pre>mysql> drop user '';</pre> 2. anonymous 패스워드 설정 <pre>mysql> set password for ''@'localhost' = password('new password');</pre>		
진단 기준	양호 - 패스워드가 안전하게 설정되어 있는 경우 취약 - 패스워드가 null 이거나 취약하게 설정되어 있는 경우		
진단 방법	[진단예시] <pre>mysql> select host, user, password from mysql.user where user=' ';</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	D-02		

20.2. 권한 관리

20.2.1. 개발 및 운영 시스템 분리 사용

분류	권한 관리	중요도	하
항목명	개발 및 운영 시스템 분리 사용		
항목 설명	개발용과 운영용이 같이 사용되는 경우 개발 시 취약하게 설정된 것으로 인하여 외부에 취약점이 노출될 위험이 있으므로 개발 시스템과 운영 시스템은 물리적으로 분리해야 함.		
설정 방법	1. 개발 시스템과 운영 시스템의 분리는 회사 정책으로 정의 2. 개발 시스템과 운영 시스템은 하드웨어적으로 분리되어야 하며 원칙적으로 개발자와 운영자는 분리 3. 개발 시스템과 운영 시스템에는 Link가 설정되어 있지 않아야 하며, 운영 시스템의 데이터는 개발 시스템으로 이전 시 중요 데이터 삭제 등과 같은 통제 절차를 거쳐야 하며 이는 검증 필요 4. 개발자의 운영 시스템 접근은 제한		
진단 기준	양호 - 개발 시스템과 운영시스템이 분리되어 사용하는 경우 취약 - 개발 시스템과 운영시스템이 분리되어 있지 않은 상태에서 사용하는 경우		
진단 방법	[진단예시] 담당자 인터뷰		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

20.2.2. root 권한으로 서버 구동 제한

분류	권한 관리	중요도	상
항목명	root 권한으로 서버 구동 제한		
항목 설명	유닉스 root 사용자 권한으로는 절대 mysql 서버를 구동하지 말아야 함. File 권한을 가진 사용자라면 root 의 권한으로 /root/.bashrc와 같은 파일을 생성할 수 있음.		
설정 방법	1. mysql 관리 용도의 일반 계정을 생성하여 처리하는 것이 보다 안전함 my.cnf 옵션 파일의 [mysqld]그룹에 사용자 이름을 지정하는 user 옵션을 추가함 user=〈시스템 일반 사용자 명〉		
진단 기준	양호 - 없거나 일반 사용자로 정의되어 있는 경우 취약 - MySQL Worker프로세스가 root계정으로 구동되고 있는 경우 ※ MySQL Worker프로세스가 root계정으로 구동되는지 My.cnf파일 내용 중 user=user_name 확인		
진단 방법	[진단예시] # ps -ef grep mysql grep root		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

20.2.3. mysql.user 테이블 접근 제한

분류	권한 관리	중요도	상
항목명	mysql.user 테이블 접근 제한		
항목 설명	일반 사용자의 mysql.user 테이블 접근이 허용될 경우, 일반 사용자가 DB에 등록되어 있는 사용자 계정 및 패스워드를 알 수 있게 됨.		
설정 방법	1. 일반 사용자로부터 mysql.user 테이블 모든 접근 권한 제거 <pre>mysql> revoke all on mysql.user from '[user name]'@[hosts]'; mysql> flush privileges</pre> 2. 일반 사용자로부터 mysql.user 테이블 접근 권한 제거 <pre>mysql> revoke [권한] on mysql.user from [user name]; mysql> flush privileges</pre>		
진단 기준	양호 - 존재하지 않거나 타당성 있는 사용자에게 권한이 부여된 경우 취약 - 불필요한 사용자에게 권한이 부여된 경우 ※ root를 제외한 일반계정에 select권한이 부여되어 있는지 확인		
진단 방법	[진단예시] <pre>mysql> select host, user, select_priv from mysql.tables_priv where (db='mysql' and table_name='user') and user <> 'root'; mysql> select host, user, select_priv from mysql.user where select_priv='Y' and user<>'root'; mysql> select host, user, db, select_priv from mysql.db where (db='mysql' and select_priv='Y') and user<>'root';</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	D-08		

20.2.4. 데이터베이스 접근 권한 제한

분류	권한 관리	중요도	중
항목명	데이터베이스 접근 권한 제한		
항목 설명	비 인가자가 시스템 테이블에 접근 할 경우 주요 정보 획득이 가능하므로 테이블 소유자나 DBA 권한을 가진 사용자 외에는 가능한 접근을 제한해야 함.		
설정 방법	1. mysql.user 테이블에 적용된 권한은 모든 데이터베이스에 적용이 되므로 host, user, password 를 제외한 나머지 권한은 허용하지 않음('N')으로 설정 [사용자 등록] mysql> insert into mysql.user (host, name, password) values('%', 'user name', password ('password')) ◀ 디폴트로 모든 권한 'N' 설정 [권한 변경] mysql> update mysql.user set <권한>='N' where user='user name' 2. 각 사용자는 접근하고자 하는 DB를 mysql.db에 등록 후 접근 권한을 부여하여 사용 [DB등록 시 권한 부여] mysql> insert into mysql.db values('%', 'DB name', 'username', 'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y') mysql> flush privileges [DB 권한 업데이트] mysql> update mysql.db set <권한>='Y' where db=<DB name> and user='user name' mysql> flush privileges		
진단 기준	양호 - 존재하지 않거나 타당성 있는 사용자에게 권한이 부여된 경우 취약 - 불필요한 사용자에게 권한이 부여된 경우 ※ root를 제외한 일반계정에 select_priv, insert_priv 권한이 부여되어있는지 확인		
진단 방법	[진단예시] mysql> select * from mysql.user		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	D-08		

20.3. DBMS 보안설정

20.3.1. 백업 관리

분류	DBMS 보안설정	중요도	하
항목명	백업 관리		

항목 설명	주기적인 백업이 수행되어야 하며 특히 DBMS의 유지보수 및 Upgrade 작업에는 전체 full 백업을 실시하여 장애 및 외부 침입 등에 대한 변수가 발생할 경우를 대비해야 함.
설정 방법	1. 백업 정책을 바탕으로 주기적인 백업 절차를 수립 2. 백업 복사본을 회사 외부의 안전한 위치에 보관 3. DBMS의 유지보수 및 Upgrade 작업 시에는 전체 full 백업 절차를 수립 4. 개인정보처리시스템의 경우 접속 기록이 위.변조 되지 않도록 별도의 물리적인 저장 장치에 보관하여야 하며 정기적 백업을 수행
진단 기준	양호 - 주기적으로 백업되고 있는 경우 취약 - 백업을 하지 않는 경우
진단 방법	[진단 예시] 담당자 인터뷰
비고	단기 적용(적용 시 개발자 및 운영자 협의)
기반시설 기준항목	-



20.3.2. 샘플 DB 제거

분류	DBMS 보안 설정	중요도	하
항목명	샘플 DB 제거		
항목 설명	비 인가자가 DB 설치 시 기본으로 생성되는 샘플 DB를 통하여 서버에 접근하여 주요 정보를 획득 및 삭제할 수 있는 위험이 있음.		
설정 방법	1. 다폴트로 설치되는 샘플 DB(test DB)에 패스워드 없이 접근 가능하므로 샘플 DB 삭제 mysql> delete from DB where DB='test'		
진단 기준	양호 - 샘플 DB(test DB)가 없는 경우 취약 - 샘플 DB(test DB)가 있는 경우		
진단 방법	[진단 예시] mysql> show databases;		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	D-10		

20.3.3. DB 접속 IP 통제

분류	DBMS 보안 설정	중요도	하
항목명	DB 접속 IP 통제		
항목 설명	임의의 사용자에게 의한 원격 접속을 차단하기 위해 IP 접근 제한을 설정 함.		
설정 방법	1. mysql.user 테이블과 mysql.db 테이블을 조회하여 host가 “%”인 필드 삭제하고 접속 IP를 지정하여 등록 mysql> delete from user where host='%'; mysql> delete from db where host='%';		
진단 기준	양호 - 특정 host를 지정하여 등록되어 있을 경우 취약 - 모든 host를 허용하는 경우 ※ 특정 DB 접속에 대해 “%”로 등록하여 모든 host를 허용하는지 확인		
진단 방법	[진단 예시] mysql> select host, user from user; mysql> select host, db, user from db;		
비고	단기 적용(적용 시 운영자 협의)		
기반시설 기준항목	D-07		

20.3.4. LOCAL INFILE 사용제한

분류	DBMS 보안 설정	중요도	중
항목명	LOCAL INFILE 사용 제한		
항목 설명	“LOCAL INFILE”이 활성화되는 경우 내부 설정 및 기타 파일에 대한 정보가 비 인가자에 의해 읽혀질 수 있음. 특히, SQL Injection 취약점이 있는 경우 더욱 위험함.		
설정 방법	1. User 테이블에서 file_priv 권한이 root 외 다른 사용자에게 부여된 경우 권한 제거 [권한 변경] mysql> update user set file_priv='N' where user='user name'; mysql> flush privileges;		
진단 기준	양호 - LOCAL INFILE의 값이 0으로 설정되어 있는 경우 취약 - LOCAL INFILE의 값이 0으로 설정되어 있지 않은 경우		
진단 방법	[진단 예시] // local-infile 사용제한 설정 확인(my.cnf 파일의 경로는 시스템마다 다를 수 있음) # cat /etc/my.cnf grep -i local-infile grep -v “^#”;		
비고	중기 적용(적용 시 운영자 협의)		
기반시설 기준항목	-		

20.3.5. 로그 저장 주기

분류	DBMS 보안 설정	중요도	상						
항목명	로그 저장 주기								
항목 설명	'정보통신망이용촉진및정보보호등에관한법률', '개인정보보호법', '회사사규' 등에 따라 접속 기록은 정해진 최소 보유 기간 동안 보관해야하며, 담당자는 접속 기록을 정기적으로 백업·확인·감독하여야 함.								
설정 방법	1. 접속 기록 보유 기간은 사업 환경에 따라 조정 할 수 있으나, '정보통신망이용촉진 및 정보보호등에관한법률', '개인정보보호법', '회사사규' 등에 따라 최소 아래 기간 이상은 보관 해야함 1) 사용자접속기록 <table><tr><td>사용자로그인/로그아웃/정보변경 등</td><td>6개월이상</td></tr></table> 2) 개인정보취급자의개인정보처리시스템접속기록 <table><tr><td>정보주체 식별정보/개인정보취급자 식별정보/ 접속일시/접속지 정보/ 부여된 권한 유형에 따른 수행업무 등</td><td>2년이상</td></tr></table> 3) 개인정보취급자권한변경기록 <table><tr><td>개인정보취급자권한생성/변경/삭제 등</td><td>5년이상</td></tr></table> 2. 담당자는 접속 기록을 월 1회 이상 정기적으로 확인·감독하여, 접속과 관련 된 오류 및 부정행위가 발생하거나 예상 되는 경우 즉각적인 보고 조치가 되도록 해야함 3. 접속 기록이 위·변조 되지 않도록 별도의 물리적인 저장 장치에 보관하여야 하며 정기적인 백업을 수행 해야 함 ※ 수정이 가능해야 할 경우, 위·변조 여부를 확인 할 수 있도록 별도 보호조치 - 위·변조 여부를 확인할 수 있는 정보(HMAC값 또는 전자서명값 등)는 별도의 HDD 또는 관리대장에 보관 하는 방법으로 관리 4. MySQL DB의 Error Log, MySQL Log 저장해야 함 - MY.CNF, MY.INI 파일에서 LOG 파일 위치 확인 예) Error Log : /\$datadir/hostname.err MySQL Log : /\$datadir/ib_logfile			사용자로그인/로그아웃/정보변경 등	6개월이상	정보주체 식별정보/개인정보취급자 식별정보/ 접속일시/접속지 정보/ 부여된 권한 유형에 따른 수행업무 등	2년이상	개인정보취급자권한생성/변경/삭제 등	5년이상
사용자로그인/로그아웃/정보변경 등	6개월이상								
정보주체 식별정보/개인정보취급자 식별정보/ 접속일시/접속지 정보/ 부여된 권한 유형에 따른 수행업무 등	2년이상								
개인정보취급자권한생성/변경/삭제 등	5년이상								
진단 기준	양호 - 접속 기록 보유 및 백업 등이 안전하게 관리되고 있는 경우 취약 - 접속 기록 보유 및 백업 등이 안전하게 관리되고 있지 않은 경우								
진단 방법	[진단 예시] 담당자 인터뷰								
비고	중기 적용(적용 시 개발자 및 운영자 협의)								
기반시설 기준항목	-								

20.4. 환경 파일 점검

20.4.1. mysql 명령 히스토리 검사

분류	환경 파일 점검	중요도	하
항목명	mysql 명령 히스토리 검사		
항목 설명	로컬 시스템에서 mysql을 사용하여 DBMS에 접근할 때 명령어 실행에서 계정 및 패스워드를 함께 붙여서 사용하는 경우 쉘 히스토리(.history/.sh_history) 파일과 process 상에 사용된 계정/패스워드의 기록이 남으므로 접근 권한 설정이 필요함.		
설정 방법	1. mysql 명령 사용시 계정/패스워드를 입력하지 말고 <code>mysql -u <계정명> -p</code> 명령만 실행 후 패스워드를 개별 입력해야 함 2. 쉘 히스토리(.history/.sh_history) 파일에 대한 보호를 위하여 접근권한 설정을 600 이하로 설정 [접근 권한 설정] # <code>chmod 600 <쉘 히스토리></code>		
진단 기준	양호 - 접근권한이 600 이하로 설정되어 있는 경우 취약 - 접근권한이 600 초과로 설정되어 있는 경우		
진단 방법	[진단 예시] <pre>// 쉘 히스토리 파일 현황 확인 # ls -al ~ grep history</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	D-13		

20.4.2. Initialization 파일 접근 권한 설정

분류	환경 파일 점검	중요도	중
항목명	Initialization 파일 접근 권한 설정		
항목 설명	mysql의 중요 파일 중에 하나인 initialization 파일(my.cnf, my.ini)의 변경으로 인한 시스템 장애 발생 가능함.		
설정 방법	1. 초기화 파일(my.cnf, my.ini)의 접근 권한을 다음과 같이 설정 ■ Unix 계열 초기화 파일(my.cnf, my.ini)에 대한 보호를 위하여 접근 권한 설정을 640 이하로 설정 my.cnf 파일 디폴트 위치: /etc/my.cnf, 〈각 홈디렉토리〉/my.cnf [접근 권한 설정] # chmod 640 ./my.cnf [my.cnf 파일 우선순위 확인] # [mysql 설치 디렉토리/bin/mysql - verbose - help grep - A 1 'Default options' ■ Windows 계열 초기화 파일의 접근 권한은 Administrators, SYSTEM, Owner에게 모든 권한 또는 이하로 설정하고 기타 다른 그룹은 제거		
진단 기준	[Unix 확인방법] 양호 - 접근권한이 640 이하로 설정되어 있는 경우 취약 - 접근권한이 640 초과로 설정되어 있는 경우 [Windows 확인방법] 양호 - Administrators, SYSTEM, Owner에게 모든 권한 또는 이하인 경우 취약 - 기타 다른 그룹에 권한이 부여되어있을 경우		
진단 방법	[진단 예시] // 초기화 파일의 접근권한 설정 현황 # ls -al /etc/my.cnf		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	D-13		

20.4.3. mysql.server 파일 접근 권한 설정

분류	환경 파일 점검	중요도	중
항목명	mysql.server 파일 접근 권한 설정		
항목 설명	mysql DB의 자동 실행스크립트인 mysql.server, mysqld.exe, mysqladmin.exe을 이용하여 데몬 을 stop, start, restart 등을 할 수 있음. 또한 mysql_safe 파일도 mysql을 재실행할 수 있으므로 일반		

	사용자가 접근하는 것을 차단해야 함.
설정 방법	1. DB 자동 실행 스크립트 파일의 접근 권한을 다음과 같이 설정 ■ Unix 계열 mysql.server 접근권한을 750 이하로 설정 mysqld_safe 접근권한을 750 이하로 설정 [파일 권한 설정] <pre># chmod 750 ./mysql.server # chmod 750 ./mysqld_safe</pre> 〈mysql.server 위치 확인〉 - [mysql 설치 디렉토리]/share/mysql.server - 서버 구동 시 자동으로 구동된다면 /etc/init.d/mysqld의 링크를 확인하여 위치 확인 〈mysqld_safe 위치 확인〉 - [mysql 설치 디렉토리]/bin/mysqld_safe ■ Windows 계열 mysqld.exe, mysqld-nt.exe, mysqladmin.exe 접근 권한 부여 : Administrators, SYSTEM, Owner 만 모든 권한 부여하고 기타 다른 그룹은 제거
진단 기준	[Unix 확인방법] 양호 - 접근권한이 750 이하로 설정되어 있는 경우 취약 - 접근권한이 750 초과로 설정되어 있는 경우 [Windows 확인방법] 양호 - Administrators, SYSTEM, Owner에게만 모든 권한 부여한 경우 취약 - 기타 다른 그룹에 권한이 부여되어있을 경우
진단 방법	[진단 예시] <pre>// mysql.server 파일의 접근권한 설정 확인 # ls -al [mysql 설치 디렉토리]/share/mysql.server</pre> <pre>// mysqld_safe 파일의 접근권한 설정 확인 # ls -al [mysql 설치 디렉토리]/bin/mysqld_safe</pre>
비고	단기 적용(적용 시 개발자 및 운영자 협의)
기반시설 기준항목	D-13

20.4.4. \$datadir 디렉토리 및 데이터 파일 접근 제한 설정

분류	환경 파일 점검	중요도	중
항목명	\$datadir 디렉토리 및 데이터 파일 접근 제한 설정		
항목 설명	mysql의 데이터 파일에 대한 복사, 삭제 및 변경으로 인한 정보 유출 및 시스템 장애 발생이 가능함.		
설정 방법	1. 데이터 디렉토리, 데이터 파일의 접근 권한을 다음과 같이 설정 ■ Unix 계열 \$datadir(mysql의 데이터 파일이 저장된 디렉토리)의 권한을 750이하로 설정하고, 데이터 파일을 640 이하로 설정 [datadir 권한] <pre># chmod 750 \$datadir</pre> [데이터 파일 권한 설정] <pre># chmod 640 file_name</pre> ■ Windows 계열 - 데이터 파일의 권한 : Administrators, SYSTEM, Owner 만 모든 권한 부여하고 기타 다른 그룹은 제거 - 데이터 디렉토리 권한 : Administrators, SYSTEM, CREATOR OWNER만 모든 권한 부여하고 기타 다른 그룹은 제거		
진단 기준	[Unix 확인방법] 양호 - 데이터 디렉토리의 접근권한이 750 이하로, 파일의 접근권한이 640 이하로 설정되어 있는 경우 취약 - 데이터 디렉토리의 접근권한이 750 초과로, 파일의 접근권한이 640 초과로 설정되어 있는 경우 [Windows 확인방법] 양호 - Administrators, SYSTEM, Owner에게만 모든 권한 부여한 경우 취약 - 기타 다른 그룹에 권한이 부여되어있을 경우		
진단 방법	[진단 예시] <pre>// 데이터 디렉토리의 위치 확인 # ps -ef grep mysql grep datadir</pre> // 데이터 디렉토리 및 파일의 접근권한 설정 현황 <pre># ls -al [데이터 디렉토리]</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	D-13		

20.4.5. .mysql_history 파일 접근 제한

분류	환경 파일 점검	중요도	중
항목명	.mysql_history 파일 접근 제한		
항목 설명	.mysql_history 파일에는 DB에서 사용한 쿼리문이 평문으로 노출되고 있음. 해당 파일을 허가받지 않는 사용자가 읽는 경우 중요 정보를 획득할 수 있으므로 파일의 접근을 제한해야 함		
설정 방법	1. .mysql_history 파일의 접근 권한을 600 이하로 설정 # chmod 600 .mysql_history		
진단 기준	양호 - 접근권한이 600 이하로 설정되어 있는 경우 취약 - 접근권한이 600 초과로 설정되어 있는 경우		
진단 방법	[진단 예시] # ls -al ~/.mysql_history		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	D-13		



20.4.6. Log 파일 접근 제한

분류	환경 파일 점검	중요도	하
항목명	Log 파일 접근 제한		
항목 설명	Log 파일에는 중요 내용이 포함되어 있음. 이를 활용하여 침해 사고 시 분석 자료로 사용하고 있음. Log 파일이 비인가 자에게 읽히거나 쓰여지는 경우 로그 파일의 정보 노출 및 변조가 발생할 수 있으므로 접근을 제한할 필요가 있음.		
설정 방법	1. Log 파일의 접근 권한을 다음과 같이 설정 ■ Unix 계열 Log 파일 접근 권한을 640 이하로 설정 \$chmod 640 [log file] ■ Windows 계열 Log 파일 접근 권한이 Administrators, SYSTEM, Owner 그룹에만 모든 권한이 부여되도록 하고 기타 다른 그룹은 제거 [Log 파일 위치 및 찾는 방법] - MY.CNF, MY.INI 파일에서 LOG 파일 위치 확인 - Error log 디폴트 위치: data directory 에 저장(파일명: "hostname.err") - MySQL Log 디폴트 위치: data directory에 저장(파일명:*logfileXY →ib_logfile0)		
진단 기준	[Unix 확인방법] 양호 - 접근권한이 640 이하로 설정되어 있는 경우 취약 - 접근권한이 640 초과로 설정되어 있는 경우 [Windows 확인방법] 양호 - Administrators, SYSTEM, Owner에게만 모든 권한 부여한 경우 취약 - 기타 다른 그룹에 권한이 부여되어있을 경우		
진단 방법	[진단 예시] // 로그파일의 위치 확인 # cat /etc/my.cnf grep -i log-bin grep -v "^#" // my.cnf 파일에서 로그파일의 접근권한 설정 현황(로그파일 위치 확인 가능) # ls -al [로그파일 경로] egrep '(*.err ib_logfile* mysql-bin.*)' // my.cnf 파일에서 로그파일의 접근권한 설정 현황(로그파일 위치 확인 불가능) # ls -al [mysql 설치 디렉토리]/var egrep '(*.err ib_logfile* mysql-bin.*)'		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	D-13		

20.5. 보안 패치

20.5.1. 보안 패치 적용

분류	보안 패치		중요도	상
항목명	보안 패치 적용			
항목 설명	버그로 인한 침해 사고 가능하므로 주기적으로 최신 패치를 적용하여 취약점을 제거함.			
설정 방법	1. 데이터베이스에 대한 최신의 버전을 확인 후 업그레이드 및 패치 수행 2. 원격 Exploit 취약점, 제로데이 취약점은 즉시 패치 <최신 버전-2016/12/12> MySQL Community Server : 5.7.17			
	패치대상	중요도	최소 패치 기준	취약점 명
	5.1.64 이하 5.5.23 이하	High (windows)	5.1.64 이상, 5.5.26 이상	CVE-2012-3163 : Remote MySQL Security Vulnerability (인증 필요)
	5.1.64 이하 5.5.26 이하	High	5.1.64 이상, 5.5.26 이상	CVE-2012-3163 : Remote MySQL Security Vulnerability (인증 불필요)
	5.5.28 이하	High (windows)	5.5.28 이상	CVE-2012-5612 : Heap Overflow Vulnerability
	5.1.66 이하 5.5.28 이하	High (windows)	5.1.66 이상, 5.5.28 이상	CVE-2012-5611 : 'acl_get()' Buffer Overflow Vulnerability
	※ 기준 버전은 위와 같으나 신규 CVSS 이슈가 발생할 경우 추가 취약 보고될 수 있음 - 참고 자료 : 버그 패치된 릴리즈 사이트 http://downloads.mysql.com/archives.php 버그 현황 사이트 http://bugs.mysql.com/bugstats.php ※ 서비스 및 운영상의 영향도를 확인하시어 설정하여 적용해야 합니다.			
진단 기준	양호 - 최신의 서비스 팩 적용한 경우 취약 - 최신의 서비스 팩 적용하지 않을 경우			
진단 방법	[진단 예시] <pre>mysql> select @@version;</pre>			
비고	장기 적용(적용 시 개발자 및 운영자 협의)			
기반시설 기준항목	D-21, D-23			

21. PostgreSQL

21.1. 계정 관리

21.1.1. 불필요한 계정 확인

분류	계정 관리	중요도	하
항목명	불필요한 계정 확인		
항목 설명	데이터베이스의 계정 중 실질적으로 업무에 사용하지 않은 불필요한 계정들이 있는 경우 악의적인 공격자가 쉽게 데이터베이스에 접속하여 데이터를 열람, 삭제, 수정 등을 할 위험이 존재함		
설정 방법	1. 기본 계정 외 계정의 용도를 파악 후 불필요한 계정은 삭제, 그리고 user, password 가 null 인 항목이 존재하는 경우도 해당 필드를 삭제 [계정 삭제] <pre>postgres=# drop user '삭제할 계정';</pre>		
진단 기준	양호 - 불필요한 계정이 없는 경우 취약 - 불필요한 계정이 존재할 경우		
진단 방법	[진단예시] <pre>Wdu+</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

21.1.2. postgres null 패스워드 점검

분류	계정 관리	중요도	상
항목명	postgres null 패스워드 점검		
항목 설명	DBA 계정의 패스워드가 default 설정 값이 null을 사용할 경우, 시스템에 접근한 임의의 모든 사용자가 관리자 권한으로 접속하여 postgresql 의 모든 작업을 할 수 있어 postgresql DB에 저장된 모든 정보가 유출 되는 등 침해사고를 일으킬 위험이 있음.		
설정 방법	1. 패스워드가 취약하게 설정된 경우 패스워드를 다음 기준을 준수하여 변경 [패스워드 설정 기준] - 특수문자, 숫자, 문자 혼합 설정 - 영 대/소문자, 숫자, 특수문자에 대해 3종류 조합 10자 이상 설정 - 추측 가능한 패스워드 설정 금지 - 연속되는 숫자, 문자 사용 금지 [postgres 계정 패스워드 설정] <pre>postgres=# alter user postgres with password '패스워드';</pre>		

진단 기준	양호 - 패스워드가 안전하게 설정되어 있는 경우 취약 - 패스워드가 null 이거나 취약하게 설정되어 있는 경우
진단 방법	[진단예시] 담당자 인터뷰
비고	단기 적용(적용 시 개발자 및 운영자 협의)
기반시설 기준항목	-

21.1.3. 취약한 패스워드 사용 점검

분류	계정 관리	중요도	상										
항목명	취약한 패스워드 사용 점검												
항목 설명	취약한 패스워드 사용시 무차별 대입 공격 등에 의해 쉽게 패스워드가 노출되어 계정 도용 등의 위험이 존재하므로 패스워드 요구조건을 반영하여 사용자가 안전한 패스워드를 생성 및 관리할 수 있도록 설정규칙을 제공해야 함.												
설정 방법	[패스워드 설정규칙 적용] 패스워드 설정규칙에 맞추어 패스워드를 설정할 수 있도록 시스템 차원에서 기능 제공												
	<table><tr><th>구분</th><th>공통 기준</th></tr><tr><td>패스워드 길이/복잡성</td><td>10자리 이상/3종류 이상</td></tr><tr><td>변경 주기</td><td>3개월/1개월(중요시스템)</td></tr><tr><td>재사용 금지</td><td>직전 1개 패스워드</td></tr><tr><td>잠금</td><td>10회 실패 시</td></tr></table>			구분	공통 기준	패스워드 길이/복잡성	10자리 이상/3종류 이상	변경 주기	3개월/1개월(중요시스템)	재사용 금지	직전 1개 패스워드	잠금	10회 실패 시
	구분	공통 기준											
	패스워드 길이/복잡성	10자리 이상/3종류 이상											
	변경 주기	3개월/1개월(중요시스템)											
	재사용 금지	직전 1개 패스워드											
	잠금	10회 실패 시											
	1) 패스워드는 아래의 4가지 문자 종류 중 2종류 이상을 조합하여 최소 10자리 이상 또는 3종류 이상을 조합하여 최소 8자리 이상의 길이로 구성												
	(1) 영문 대문자 (26개)												
	(2) 영문 소문자 (26개)												
(3) 숫자 (10개)													
(4) 특수문자 (32개)													
2) 패스워드는 비인가자에 의한 추측이 어렵도록 다음의 사항을 반영하여 설계													
(1) Null 패스워드 사용 금지													
(2) 문자 또는 숫자만으로 구성 금지													
(3) 사용자 ID와 동일한 패스워드 금지													
(4) 연속적인 문자/숫자(예. 1111, 1234, abcd) 사용 금지													
(5) 주기성 패스워드 재사용 금지													
(6) 전화번호, 생일같이 추측하기 쉬운 개인정보를 패스워드로 사용 금지													
3) 초기 패스워드는 사용자에게 부여 후 최초 접속 시 즉시 변경되도록 설계													
(4) 10회 이상의 연속적인 패스워드 입력 실패 시 해당 사용자 ID는 사용권한이 일시 중지되도록 설계													
(5) 패스워드는 최대 3개월, 업무 중요도에 따라 1개월 주기로 변경													
(6) 패스워드 변경 시 직전 1개와 동일한 패스워드 사용금지													
(7) 패스워드는 마스킹 처리 등을 통해 화면상에서 읽을 수 없는 형태로 표시													

	8) 패키지 등을 도입한 경우 패키지의 기본 기능에서 위의 항목이 제공되지 않는 경우 보안운영자에게 해당 내용 문의 및 보안성 검토 후 요구조건 반영 [패스워드 관리 적용] 패스워드 신규 적용 및 초기화 시 설정규칙에 맞추어 관리하고, 저장 시에는 일방향 암호 알고리즘을 통한 암호화 처리(One-Way Encryption) 패스워드 분실로 인한 신규 패스워드 발급 절차 및 클리핑 레벨 적용 - 패스워드 분실 시 패스워드를 초기화한 후 안전한 전송수단을 통해 패스워드 제공 - 패스워드 초기화 처리 후 로그인시 패스워드 변경을 유도 - 사용자 식별/인증 실패 시 계정 잠금 및 접속차단 기능 적용 - 계정 잠금 해제를 위한 절차 적용 [패스워드 변경기능 구현] 관리자에 의한 변경과 사용자가 스스로 패스워드를 변경할 수 있는 기능 제공 사용자로부터 패스워드 변경 요청이 있을 경우, 사용자 신원 확인이 완료된 후 패스워드 변경될 수 있도록 설정 [패스워드 설정] postgres=# alter user 계정명 with password '패스워드';
진단 기준	양호 - 패스워드가 안전하게 설정되어 있는 경우 취약 - 패스워드가 USER명과 같거나 취약하게 설정되어 있는 경우
진단 방법	[진단예시] 담당자 인터뷰
비고	단기 적용(적용 시 개발자 및 운영자 협의)
기반시설 기준항목	-

21.2. 권한 관리

21.2.1. 개발 및 운영 시스템 분리 사용

분류	권한 관리	중요도	하
항목명	개발 및 운영 시스템 분리 사용		
항목 설명	개발용과 운영용이 같이 사용되는 경우 개발 시 취약하게 설정된 것으로 인하여 외부에 취약점이 노출될 위험이 있으므로 개발 시스템과 운영 시스템은 물리적으로 분리해야 함.		
설정 방법	1. 개발 시스템과 운영 시스템의 분리는 회사 정책으로 정의 2. 개발 시스템과 운영 시스템은 하드웨어적으로 분리되어야 하며 원칙적으로 개발자와 운영자는 분리 3. 개발 시스템과 운영 시스템에는 Link가 설정되어 있지 않아야 하며, 운영 시스템의 데이터는 개발 시스템으로 이전 시 중요 데이터 삭제 등과 같은 통제 절차를 거쳐야 하며 이는 검증 필요 4. 개발자의 운영 시스템 접근은 제한		
진단 기준	양호 - 개발 시스템과 운영시스템이 분리되어 사용하는 경우 취약 - 개발 시스템과 운영시스템이 분리되어 있지 않은 상태에서 사용하는 경우		
진단 방법	[진단예시] 담당자 인터뷰		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

21.2.2. root 권한으로 서버 구동 제한

분류	권한 관리	중요도	상
항목명	root 권한으로 서버 구동 제한		
항목 설명	유닉스 root 사용자 권한으로는 절대 postgresql 서버를 구동하지 말아야 함. File 권한을 가진 사용자라면 root 의 권한으로 /root/.bashrc와 같은 파일을 생성할 수 있음.		
설정 방법	1. postgresql 관리 용도의 일반 계정을 생성하여 처리하는 것이 보다 안전함 postgresql 서버가 root로 구동되어 있는 경우 postgres 계정 또는 일반계정으로 서비스 재시작 해야 함.		
진단 기준	양호 - 없거나 일반 사용자로 정의 되어있는 경우_ 취약 - MySQL Worker프로세스가 root계정으로 구동되고 있는 경우 ※ MySQL Worker프로세스가 root계정으로 구동되는지 My.cnf파일 내용 중 user=user_name 확인		
진단 방법	[진단예시] # ps -ef grep postgres		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

21.2.3. schema 접근 권한 제한

분류	권한 관리	중요도	중
항목명	schema 접근 권한 제한		
항목 설명	모든 사용자는 해당 사용자가 연결할 수 있는 모든 데이터베이스의 Public Schema에 개체생성이 가능하므로 권한을 제한해야 함		
설정 방법	1. schema명에 해당되는 table들에 대한 권한을 user에게 제한 postgres=# revoke [all,select,insert,update....] on all tables in schema schema명 from user명;		
진단 기준	양호 - Public Schema에 대해 불필요한 사용자에게 권한이 부여되지 않은 경우 취약 - Public Schema에 대해 불필요한 사용자에게 권한이 부여된 경우		
진단 방법	[진단예시] Wdn		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

21.3. DBMS 보안설정

21.3.1. 백업 관리

분류	DBMS 보안설정	중요도	하
항목명	백업 관리		
항목 설명	주기적인 백업이 수행되어야 하며 특히 DBMS의 유지보수 및 Upgrade 작업에는 전체 full 백업을 실시하여 장애 및 외부 침입 등에 대한 변수가 발생할 경우를 대비해야 함.		
설정 방법	1. 백업 정책을 바탕으로 주기적인 백업 절차를 수립 2. 백업 복사본을 회사 외부의 안전한 위치에 보관 3. DBMS의 유지보수 및 Upgrade 작업 시에는 전체 full 백업 절차를 수립 4. 개인정보처리시스템의 경우 접속 기록이 위.변조 되지 않도록 별도의 물리적인 저장 장치에 보관하여야 하며 정기적 백업을 수행		
진단 기준	양호 - 주기적으로 백업되고 있는 경우 취약 - 백업을 하지 않는 경우		
진단 방법	[진단 예시] 담당자 인터뷰		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

21.3.2. DB 접속 IP 통제

분류	DBMS 보안 설정	중요도	하
항목명	DB 접속 IP 통제		
항목 설명	임의의 사용자에게 의한 원격 접속을 차단하기 위해 IP 접근 제한을 설정 함.		
설정 방법	1. Data 디렉터리 안에 있는 pg_hba.conf 파일설정을 통해서 설정가능 TYPE DATABASE USER CIDR-ADDRESS METHOD host all (사용자) (접속허용IP) md5 USER 에 접근허용 사용자명과 CIDR-ADDRESS 에 접속을 허용할 IP 를 설정 함		
진단 기준	양호 - IP 차단이 설정되어 있는 경우 취약 - IP 차단이 설정되어 있지 않는 경우		
진단 방법	[진단 예시] # cat /postgres/data/pg_hba.conf/postgres/data/ pg_hba.conf		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		
기반시설	-		

기준항목	
------	--

21.3.3. 로그 저장 주기

분류	DBMS 보안 설정	중요도	상						
항목명	로그 저장 주기								
항목 설명	'정보통신망이용촉진및정보보호등에관한법률', '개인정보보호법', '회사사규' 등에 따라 접속 기록은 정해진 최소 보유 기간 동안 보관해야하며, 담당자는 접속 기록을 정기적으로 백업·확인·감독하여야 함.								
설정 방법	1. 접속 기록 보유 기간은 사업 환경에 따라 조정 할 수 있으나, '정보통신망이용촉진 및 정보보호등에관한법률', '개인정보보호법', '회사사규' 등에 따라 최소 아래 기간 이상은 보관 해야함 1) 사용자접속기록 <table><tr><td>사용자로그인/로그아웃/정보변경 등</td><td>6개월이상</td></tr></table> 2) 개인정보취급자의개인정보처리시스템접속기록 <table><tr><td>정보주체 식별정보/개인정보취급자 식별정보/ 접속일시/접속지 정보/ 부여된 권한 유형에 따른 수행업무 등</td><td>2년이상</td></tr></table> 3) 개인정보취급자권한변경기록 <table><tr><td>개인정보취급자권한변경/생성/변경/삭제 등</td><td>5년이상</td></tr></table> 2. 담당자는 접속 기록을 월 1회 이상 정기적으로 확인·감독하여, 접속과 관련 된 오류 및 부정행위가 발생하거나 예상 되는 경우 즉각적인 보고 조치가 되도록 해야함 3. 접속 기록이 위·변조 되지 않도록 별도의 물리적인 저장 장치에 보관하여야 하며 정기적인 백업을 수행 해야 함 ※ 수정이 가능해야 할 경우, 위·변조 여부를 확인 할 수 있도록 별도 보호조치 - 위·변조 여부를 확인할 수 있는 정보(HMAC값 또는 전자서명값 등)는 별도의 HDD 또는 관리대장에 보관 하는 방법으로 관리 4. MySQL DB의 Error Log, MySQL Log 저장해야 함 - MY.CNF, MY.INI 파일에서 LOG 파일 위치 확인 예) Error Log : /\$datadir/hostname.err MySQL Log : /\$datadir/ib_logfile			사용자로그인/로그아웃/정보변경 등	6개월이상	정보주체 식별정보/개인정보취급자 식별정보/ 접속일시/접속지 정보/ 부여된 권한 유형에 따른 수행업무 등	2년이상	개인정보취급자권한변경/생성/변경/삭제 등	5년이상
사용자로그인/로그아웃/정보변경 등	6개월이상								
정보주체 식별정보/개인정보취급자 식별정보/ 접속일시/접속지 정보/ 부여된 권한 유형에 따른 수행업무 등	2년이상								
개인정보취급자권한변경/생성/변경/삭제 등	5년이상								
진단 기준	양호 - 접속 기록 보유 및 백업 등이 안전하게 관리되고 있는 경우 취약 - 접속 기록 보유 및 백업 등이 안전하게 관리되고 있지 않은 경우								
진단 방법	[진단 예시] 담당자 인터뷰								
비고	중기 적용(적용 시 개발자 및 운영자 협의)								
기반시설 기준항목	-								

21.4. 환경 파일 점검

21.4.1. PostgreSQL 명령 히스토리 검사

분류	환경 파일 점검	중요도	하
항목명	PostgreSQL 명령 히스토리 검사		
항목 설명	히스토리 파일(.history)은 로그인하는 모든 사용자들의 명령어를 저장하는 스크립트이므로 보안 상의 관리가 요구 됨.		
설정 방법	1. 쉘 히스토리(.history /.sh_history) 파일에 대한 보호를 위하여 접근권한 설정을 600 이하로 설정 [접근 권한 설정] # chmod 600 <셸 히스토리>		
진단 기준	양호 - 접근권한이 600 이하로 설정되어 있는 경우 취약 - 접근권한이 600 초과로 설정되어 있는 경우		
진단 방법	[진단 예시] // 쉘 히스토리 파일 현황 확인 # ls -al ~ grep history		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

21.4.2. PostgreSQL 환경설정 파일 접근 제한

분류	환경 파일 점검	중요도	중
항목명	PostgreSQL 환경설정 파일 접근 제한		
항목 설명	PostgreSQL의 중요 파일 중에 하나인 환경설정 파일(postgresql.conf)의 변경으로 인한 시스템 장애 발생 가능 함.		
설정 방법	1. 환경설정 파일(postgresql.conf)의 접근 권한을 다음과 같이 설정 ■ Unix 계열 환경설정 파일에 대한 보호를 위하여 접근 권한 설정을 600 또는 640 으로 설정 postgresql.conf 파일 디폴트 위치: <\$datadir> [접근 권한 설정] # chmod 600 <\$datadir>/postgresql.conf# chmod 600 <\$datadir>/postgresql.conf ■ Windows 계열 환경설정 파일의 접근 권한은 Administrators, SYSTEM, Owner에게 모든 권한 또는 이하로 설정하고 기타 다른 그룹은 제거		
진단 기준	[Unix 확인방법] 양호 - 접근권한이 600 또는 640 이하로 설정되어 있는 경우 취약 - 접근권한이 600 또는 640 초과로 설정되어 있는 경우 [Windows 확인방법] 양호 - Administrators, SYSTEM, Owner에게 모든 권한 또는 이하인 경우 취약 - 기타 다른 그룹에 권한이 부여되어있을 경우		
진단 방법	[진단 예시] # ls -al /<\$datadir>/conf/postgresql.conf		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

21.4.3. DB접속 통제 설정 파일 접근 권한 설정

분류	환경 파일 점검	중요도	중
항목명	DB접속 통제 설정 파일 접근 권한 설정		
항목 설명	postgresql DB의 접속 통제 환경설정 파일인 pg_hba.conf, pg_ident.conf 에는 외부에서 접속이 가능하게 하거나 패스워드 없이 로그인을 가능하게 하는 설정이 포함되어 있기 때문에 타사용자가 접근하지 못하도록 접근제한을 해야 함.		
설정 방법	1. DB 접속 통제 설정파일의 접근 권한을 다음과 같이 설정 ■ Unix 계열 pg_hba.conf와 pg_ident.conf 파일의 퍼미션을 600 또는 640 이하로 설정 [파일 권한 설정] # chmod 600 ./pg_hba.conf # chmod 600 ./pg_ident.conf <pg_hba.conf, pg_ident.conf 위치> - postgresql Data 디렉터리 : /postgres/data/pg_hba.conf/postgres/data/pg_ident.conf ■ Windows 계열 pg_hba.conf, pg_ident.conf 접근 권한 부여 : Administrators, SYSTEM, Owner 만 모든 권한 부여하고 기타 다른 그룹은 제거		
진단 기준	[Unix 확인방법] 양호 - 접근권한이 600 또는 640 이하로 설정되어 있는 경우 취약 - 접근권한이 600 또는 640 초과로 설정되어 있는 경우 [Windows 확인방법] 양호 - Administrators, SYSTEM, Owner에게만 모든 권한 부여한 경우 취약 - 기타 다른 그룹에 권한이 부여되어있을 경우		
진단 방법	[진단 예시] # ls -al /postgres/data/pg_hba.conf/postgres/data/ pg_hba.conf # ls -al /postgres/data/pg_hba.conf/postgres/data/ pg_ident.conf		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

21.4.4. \$datadir 디렉토리 및 데이터 파일 접근 제한

분류	환경 파일 점검	중요도	중
항목명	\$datadir 디렉토리 및 데이터 파일 접근 제한		

항목 설명	postgresql 의 데이터 파일에 대한 복사, 삭제 및 변경으로 인한 정보 유출 및 시스템 장애 발생이 가능 함.
설정 방법	1. 데이터 디렉토리, 데이터 파일의 접근 권한을 다음과 같이 설정 ■ Unix 계열 \$datadir(PostgreSQL의 데이터 파일이 저장된 디렉터리)의 권한을 750이하로 설정하고, 데이터 파일을 600 또는 640 이하로 설정 [\$datadir 권한] # chmod 750 <데이터 파일이 저장된 디렉터리 경로> [데이터 파일 권한 설정] # chmod 640 file_name ■ Windows 계열 - 데이터 파일의 권한 : Administrators, SYSTEM, Owner 만 모든 권한 부여하고 기타 다른 그룹은 제거 - 데이터 디렉토리 권한 : Administrators, SYSTEM,CREATOR OWNER만 모든 권한 부여하고 기타 다른 그룹은 제거
진단 기준	[Unix 확인방법] 양호 - 데이터 디렉토리의 접근권한이 750 이하로, 파일의 접근권한이 600 또는 640 이하로 설정되어 있는 경우 취약 - 데이터 디렉토리의 접근권한이 750 초과로, 파일의 접근권한이 600 또는 640 초과로 설정되어 있는 경우 [Windows 확인방법] 양호 - Administrators, SYSTEM, Owner에게만 모든 권한 부여한 경우 취약 - 기타 다른 그룹에 권한이 부여되어있을 경우
진단 방법	[진단 예시] # ls -ald /postgres/data/pg_hba.conf/postgres/data/ # ls -al /postgres/data/pg_hba.conf/postgres/data/
비고	단기 적용(적용 시 개발자 및 운영자 협의)
기반시설 기준항목	-

21.4.5. .psql_history 파일 접근 제한

분류	환경 파일 점검	중요도	중
항목명	.psql_history 파일 접근 제한		
항목 설명	.psql_history 파일에는 DB 에서 사용한 쿼리문이 평문으로 노출되고 있음. 해당 파일을 허가 받지 않는 사용자가 읽는 경우 중요 정보를 획득할 수 있으므로 파일의 접근을 제한해야 함.		
설정 방법	1. .psql_history 파일의 접근 권한을 600 또는 640 으로 설정 <code>\$chmod 600 .psql_history</code> 2. .psql_history 파일의 내용 제거 <code>\$>cat /dev/null > ~/.psql_history</code>		
진단 기준	양호 - 접근권한이 600 또는 640 이하로 설정되어 있는 경우 취약 - 접근권한이 600 또는 640 초과로 설정되어 있는 경우		
진단 방법	[진단 예시] <code># find / -name ".psql_history" ls -al</code>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

21.4.6. Log 파일 접근 제한

분류	환경 파일 점검	중요도	하
항목명	Log 파일 접근 제한		
항목 설명	Log 파일에는 중요 내용이 포함되어 있음. 이를 활용하여 침해 사고 시 분석 자료로 사용하고 있음. Log 파일이 비인가 자에게 읽히거나 쓰여지는 경우 로그 파일의 정보 노출 및 변조가 발생할 수 있으므로 접근을 제한할 필요가 있음.		
설정 방법	1. Log 파일의 접근 권한을 다음과 같이 설정 ■ Unix 계열 Log 파일 접근 권한을 600 또는 640 이하로 설정 \$chmod 640 [log file] ■ Windows 계열 Log 파일 접근 권한이 Administrators, SYSTEM, Owner 그룹에만 모든 권한이 부여되도록 하고 기타 다른 그룹은 제거 [Log 파일 위치 및 찾는 방법] postgres=# show log_directory; log_directory ----- pg_log (1 row) <\$data 디렉터리>/pg_log/		
진단 기준	[Unix 확인방법] 양호 - 접근권한이 600 또는 640 이하로 설정되어 있는 경우 취약 - 접근권한이 600 또는 640 초과로 설정되어 있는 경우 [Windows 확인방법] 양호 - Administrators, SYSTEM, Owner에게만 모든 권한 부여한 경우 취약 - 기타 다른 그룹에 권한이 부여되어있을 경우		
진단 방법	[진단 예시] postgres=# show log_directory; # ls -al /postgres/data/pg_hba.conf/postgres/data/		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

21.5. 보안 패치

21.5.1. 보안 패치 적용

분류	보안 패치	중요도	상
항목명	보안 패치 적용		

항목 설명	버그로 인한 침해 사고 가능하므로 주기적으로 최신 패치를 적용하여 취약점을 제거함.		
설정 방법	1. 데이터베이스에 대한 최신의 버전을 확인 후 업그레이드 및 패치 수행 2. 원격 Exploit 취약점, 제로데이 취약점은 즉시 패치		
	〈Latest Release 2017.02.09〉		
	Version	Last Version	Release 일자
	9.6	9.6.2	2017-02-09
	9.5	9.5.6	2017-02-09
	9.4	9.4.11	2017-02-09
	9.3	9.3.16	2017-02-09
	9.2	9.2.20	2017-02-09
	9.1	9.1.24	2016-10-27
	- 참고 자료 : http://www.postgresql.org/docs/		
	※ 서비스 및 운영상의 영향도를 확인하시어 설정하여 적용해야 합니다.		
진단 기준	양호 - 최신의 서비스 팩 적용한 경우 취약 - 최신의 서비스 팩 적용하지 않을 경우		
진단 방법	[진단 예시] postgres=# SELECT VERSION();		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

21.6. 보안 감사 설정

21.6.1. Log 감사 수행 설정

분류	보안 감사 설정	중요도	하
항목명	Log 감사 수행 설정		
항목 설명	중요 정보에 대한 DBA로 접속하는 사용자에게 대해 감사 기능을 설정함으로써 침해 발생 분석 및 침해 시도 여부를 확인할 수 있음.		
설정 방법	1. Log 감사 설정 여부 확인방법 (쿼리문) <pre>postgres=# show logging_collector; logging_collector ----- on (1 row)</pre> 2. postgresql.conf 파일 내 logging_collector 을 on 으로설정 <pre>logging_collector = on</pre>		
진단 기준	양호 - 감사기능 설정이 활성화된 경우 취약 - 감사기능 설정이 비활성화된 경우		
진단 방법	[진단예시] <pre># cat /postgres/data/pg_hba.conf/postgres/data/postgresql.conf grep "logging_collector"</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

21.6.2. 로그 기록 설정

분류	보안 감사 설정	중요도	하
항목명	로그 기록 설정		
항목 설명	사용자의 쿼리문 명령어에 대한 추적이 가능하도록 하여 침해 사고 및 장애 발생 시 감사 자료를 통해 정확한 분석을 할 수 있음.		
설정 방법	1. #TB_HOME/config/\$TB_SID.tip 파일 내 'Audit_trail'의 값을 [DB TRUE OS DB_EXTENDED] 중 하나 설정 <pre>postgres=# show log_statement; log_statement ----- all (1 row)</pre> 2. 기록 설정 방법 (시스템 재 시작 필요) <pre>postgresql.conf 파일에 log_statement 를 all 로 설정 log_statement_stats = all</pre>		
진단 기준	양호 - 로그기록 설정 활성화 되어있는 경우 취약 - 로그기록 설정 비활성화 되어있는 경우		
진단 방법	[진단예시] <pre># cat /postgres/data/pg_hba.conf/postgres/data/postgresql.conf grep "log_statement"</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

22. Redis(NoSQL)

22.1. 계정 관리

22.1.1. Redis null 패스워드 점검

분류	계정 관리	중요도	상
항목명	redis null 패스워드 점검		
항목 설명	NoSQL은 기본적으로 계정 없이 사용이 가능하여 누구나 DB에 접근 가능하므로 허가된 사용자만 접근 할 수 있도록 서비스 사용 시 계정 및 패스워드를 사용해야 함		
설정 방법	1. 패스워드가 취약하게 설정된 경우 패스워드를 다음 기준을 준수하여 변경 [패스워드 설정 기준] - 특수문자, 숫자, 문자 혼합 설정 - 영 대/소문자, 숫자, 특수문자에 대해 3종류 조합 10자 이상 설정 - 추측 가능한 패스워드 설정 금지 - 연속되는 숫자, 문자 사용 금지 [Redis 계정 패스워드 설정] 1) redis.conf 설정 파일 내 requirepass 주석 해제 후 패스워드 설정 (수정 전) # requirepass foobared // 주석 해제 후 패스워드 설정 (수정 후) requirepass 패스워드 2) /etc/init.d 로 이동하여 redis 서버 재기동 <pre>\$)./redis stop \$)./redis start</pre> 3) redis 설치 폴더 내 redis-cli 실행 후 명령어 입력을 통해 동작하지 않는 것이 확인되면 양호 <pre>[root@redishost src]# ./redis-cli redis 127.0.0.1:6379> ping // 접속상태를 확인하는 명령 (error) ERR operation not permitted redis 127.0.0.1:6379> redis 127.0.0.1:6379> auth 패스워드 OK redis 127.0.0.1:6379> ping PONG redis 127.0.0.1:6379></pre>		
진단 기준	양호 - redis 접근 후 패스워드를 입력해야만 명령어 사용 가능한 경우 취약 - redis 접근 후 패스워드를 없이 명령어 사용 가능한 경우		
진단 방법	[진단예시] <pre># /[설치dir]/src/redis-cli redis > ping</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		

기반시설 기준항목	-
--------------	---

22.1.2. 패스워드 복잡도 설정

분류	계정 관리	중요도	중
항목명	패스워드 복잡도 설정		
항목 설명	패스워드 복잡도가 설정되지 않은 경우 Brute force 공격을 통하여 패스워드를 쉽게 획득할 위험이 존재함		
설정 방법	1. redis.conf 파일 내 requirepass에 아래 기준을 만족하도록 패스워드 설정 [패스워드 설정 기준] - 영문 대/소문자, 숫자, 특수문자 혼합 설정 - 2종류 조합으로 10자리 이상, 3종류 조합으로 8자 이상 설정 - 추측 가능한 패스워드 설정 금지 - 연속되는 문자, 숫자 사용 금지		
진단 기준	양호 - 설정된 패스워드가 설정 기준을 준수하고 있을 경우 취약 - 설정된 패스워드가 설정 기준을 준수하지 않은 경우		
진단 방법	[진단예시] <pre># cat /[설치dir]/redis.conf grep - C3 requirepass ... requirepass 패스워드 // 설정된 패스워드의 복잡도 확인 ...</pre>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

22.2. 권한 관리

22.2.1. 개발 및 운영 시스템 분리 사용

분류	권한 관리	중요도	하
항목명	개발 및 운영 시스템 분리 사용		
항목 설명	개발용과 운영용이 같이 사용되는 경우 개발 시 취약하게 설정된 것으로 인하여 외부에 취약점이 노출될 위험이 있으므로 개발 시스템과 운영 시스템은 물리적으로 분리해야 함		
설정 방법	1. 개발 시스템과 운영 시스템의 분리는 회사 정책으로 정의 2. 개발 시스템과 운영 시스템은 하드웨어적으로 분리되어야 하며 원칙적으로 개발자와 운영자는 분리 3. 개발 시스템과 운영 시스템에는 Link가 설정되어 있지 않아야 하며, 운영 시스템의 데이터는 개발 시스템으로 이전 시 중요 데이터 삭제 등과 같은 통제 절차를 거쳐야 하며 이는 검증 필요 4. 개발자의 운영 시스템 접근은 제한		
진단 기준	양호 - 개발 시스템과 운영시스템이 분리되어 사용하는 경우 취약 - 개발 시스템과 운영시스템이 분리되어 있지 않은 상태에서 사용하는 경우		
진단 방법	[진단예시] 담당자 인터뷰		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

22.2.2. root 권한으로 서버 구동 제한

분류	권한 관리	중요도	상
항목명	root 권한으로 서버 구동 제한		
항목 설명	유닉스 root 사용자 권한으로 redis 서비스를 구동하지 말고 전용 daemon 계정으로 구동해야 함		
설정 방법	1. redis 관리 용도의 일반 계정을 생성하여 처리하는 것이 보다 안전함 redis 서비스가 root로 구동되어 있는 경우 redis 전용 계정 또는 일반 계정으로 서비스를 재시작 해야 함		
진단 기준	양호 - redis 프로세스가 root 계정 소유로 구동되고 있지 않을 경우 취약 - redis 프로세스가 root 계정 소유로 구동되고 있을 경우		
진단 방법	[진단예시] # ps -ef grep redis		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

22.3. DBMS 보안설정

22.3.1. 백업 관리

분류	DBMS 보안설정	중요도	하
항목명	백업 관리		
항목 설명	redis는 메모리 기반의 DB이기 때문에 전원 유실 시 데이터가 사라지므로 주기적인 백업이 수행되어야 하며 특히 DBMS의 유지보수 및 Upgrade 작업에는 전체 Dump를 실시하여 장애 및 외부 침입 등에 대한 변수가 발생할 경우를 대비해야 함		
설정 방법	1. 백업 정책을 바탕으로 주기적인 백업 절차를 수립 2. Dump 복사본을 회사 외부의 안전한 위치에 보관 3. DBMS의 유지보수 및 Upgrade 작업 시에는 전체 Dump 절차를 수립 4. 개인정보처리시스템의 경우 접속 기록이 위.변조 되지 않도록 별도의 물리적인 저장 장치에 보관하여야 하며 정기적 백업을 수행		
진단 기준	양호 - 주기적으로 백업되고 있는 경우 취약 - 백업을 하지 않는 경우		
진단 방법	[진단 예시] 담당자 인터뷰		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

22.3.2. DB 접속 IP 통제

분류	DBMS 보안 설정	중요도	하
항목명	DB 접속 IP 통제		
항목 설명	임의의 사용자에게 의한 원격 접속을 차단하기 위해 IP 접근 제한을 설정 함		
설정 방법	1. redis.conf 파일 내 bind 설정을 통해 설정 가능 ... bind 192.168.1.100 // redis 서버에 접근 가능한 IP를 지정 ...		
진단 기준	양호 - bind IP가 지정되어 있는 경우 취약 - bind IP가 지정되어 있지 않은 경우		
진단 방법	[진단 예시] # cat /[설치dir]/redis.conf grep -C3 bind		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		

기반시설 기준항목	-
--------------	---

22.3.3. 로그 저장 주기

분류	DBMS 보안 설정	중요도	상
항목명	로그 저장 주기		
항목 설명	'정보통신망이용촉진및정보보호등에관한법률', '개인정보보호법', '회사사규' 등에 따라 접속 기록은 정해진 최소 보유 기간 동안 보관해야하며, 담당자는 접속 기록을 정기적으로 백업 · 확인 · 감독하여야 함		
설정 방법	1. 접속 기록 보유 기간은 사업 환경에 따라 조정 할 수 있으나, '정보통신망이용촉진 및 정보보호등에관한법률', '개인정보보호법', '회사사규' 등에 따라 최소 아래 기간 이상은 보관 해야 함		
	1) 사용자접속기록		
	사용자로그인/로그아웃/정보변경 등	6개월이상	
	2) 개인정보취급자의개인정보처리시스템접속기록		
	정보주체 식별정보/개인정보취급자 식별정보/ 접속일시/접속지 정보/ 부여된 권한 유형에 따른 수행업무 등	2년이상	
3) 개인정보취급자권한변경기록	개인정보취급자권한생성/변경/삭제 등	5년이상	
	2. 담당자는 접속 기록을 월 1회 이상 정기적으로 확인·감독하여, 접속과 관련 된 오류 및 부정행위가 발생하거나 예상 되는 경우 즉각적인 보고 조치가 되도록 해야함		
3. 접속 기록이 위·변조 되지 않도록 별도의 물리적인 저장 장치에 보관하여야 하며 정기적인 백업을 수행 해야 함			
※ 수정이 가능해야 할 경우, 위·변조 여부를 확인 할 수 있도록 별도 보호조치			
– 위·변조 여부를 확인할 수 있는 정보(HMAC값 또는 전자서명값 등)는 별도의 HDD 또는 관리대장에 보관 하는 방법으로 관리			
4. redis.conf 설정 파일에서 LOG 파일 및 경로 지정			
진단 기준	양호 - 접속 기록 보유 및 백업 등이 안전하게 관리되고 있는 경우 취약 - 접속 기록 보유 및 백업 등이 안전하게 관리되고 있지 않은 경우		
진단 방법	[진단 예시] # cat /[설치dir]/redis.conf grep - C3 logfile 로그 관리 및 저장 주기에 대한 담당자 인터뷰		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

22.3.4. 로그 레벨 설정

분류	DBMS 보안 설정	중요도	상
항목명	로그 레벨 설정		
항목 설명	로그 레벨을 낮게 설정 할 경우 공격 정보에 대한 파악이 어려울 수 있으므로 파악이 가능한 최소한의 레벨을 설정해야 함		
설정 방법	1. redis.conf 설정 파일 내 loglevel 값에 notice 이상의 값을 설정 [loglevel 단계] - warnig < notice < verbose < debug (가장 많은 정보를 기록)		
진단 기준	양호 - loglevel 값이 notice 이상으로 설정된 경우 취약 - loglevel 값이 notice 미만으로 설정된 경우		
진단 방법	[진단 예시] # cat /[설치dir]/redis.conf grep -C3 loglevel		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

22.3.5. 관리자 command 보호 설정

분류	DBMS 보안 설정	중요도	중
항목명	관리자 command 보호 설정		
항목 설명	보호가 필요한 관리자 command를 변경 또는 비활성 설정하여 주요 명령에 대한 brute force 공격을 방지해야 함		
설정 방법	1. redis.conf 설정 파일 내 rename-command 주석 해제 후, 변경 또는 비활성화 할 command에 대한 설정 추가 [rename-command 설정] // 명령어 변경 예시 rename-command set CONFIG b840fc02d524045429941cc15f59e41cb7be6c52 rename-command set AUTH cc15f59e41cb7be6c52b840fc02d524045429941 // 명령어 변경 후 redis-cli 테스트 예시 127.0.0.1:6379> CONFIG (error) ERR unknown command 'CONFIG' // 명령어 비활성화 예시 (설정파일 직접 수정을 통한 변경만 가능하게 됨) rename-command CONFIG "" // CONFIG 명령어를 disable 함		

	두 대 이상의 redis DBMS를 master-slave 구성한 경우 모두 동일하게 설정 적용해야 함 ※ 적용시 보안성이 향상되나 담당자 판단에 따른 적용 결정 필요
진단 기준	양호 - rename-command 설정하여 사용중인 경우 취약 - rename-command 설정을 사용하지 않은 경우
진단 방법	[진단 예시] # cat /[설치dir]/redis.conf grep rename-command
비고	단기 적용(적용 시 개발자 및 운영자 협의)
기반시설 기준항목	-

22.3.6. DBMS 서버 보안 연결

분류	DBMS 보안 설정	중요도	상
항목명	DBMS 서버 보안 연결		
항목 설명	평문 또는 취약한 버전의 SSL 접속을 통한 DBMS 서버 connect를 허용할 경우 스니핑을 통해 로그인 계정 등 주요 정보가 노출 될 위험이 있음		
설정 방법	1. 실행중인 DBMS 프로세스의 실행 상태 connect 시 SSL 접속 설정 확인 redis command 를 통해 주요 정보를 전송하므로 spiped와 같은 SSL proxy를 사용, 서버와 통신하도록 하여 주요 정보 노출 방지 (AUTH command 사용 시 ID/PW를 서버로 전송)		
진단 기준	양호 - SSL proxy를 이용하여 통신하는 경우 취약 - SSL proxy를 이용하여 통신하지 않는 경우		
진단 방법	[진단 예시] # netstat -plnt [결과 예시] Active Internet connections (only servers) Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name tcp 0 0 public_IP:6379 0.0.0.0:* LISTEN 4292/spiped tcp 0 0 127.0.0.1:6379 0.0.0.0:* LISTEN 2679/redis-server 1		
비고	중기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

22.3.7. 샘플 및 테스트 파일 제거

분류	DBMS 보안 설정	중요도	하
항목명	샘플 및 테스트 파일 제거		
항목 설명	DB 설치 시 기본으로 생성되는 샘플 및 테스트 파일에 비 인가자가 접근하여 주요 정보를 획득 및 삭제할 수 있는 위험이 있음.		
설정 방법	1. 디폴트로 설치되는 sample 및 tests 디렉터리 삭제		
진단 기준	양호 - tests 디렉터리 및 테스트 파일이 없는 경우 취약 - tests 디렉터리 및 테스트 파일이 있는 경우		
진단 방법	[진단 예시] <code># /[설치dir]/bin/tests</code> <code># find /[설치dir] -name "*test*"</code>		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

22.4. 환경 파일 점검

22.4.1. Redis 환경설정 파일 접근 제한

분류	환경 파일 점검	중요도	중
항목명	Redis 환경설정 파일 접근 제한		
항목 설명	Redis 환경설정 파일(redis.conf)에 관리자 패스워드가 평문으로 저장되어 있어 소유자 외 타 사용자가 읽기, 쓰기 가능한 경우 서버의 주요 정보가 노출될 수 있음		
설정 방법	1. 환경설정 파일(redis.conf)의 접근 권한을 다음과 같이 설정 ■ Unix 계열 환경설정 파일에 대한 보호를 위하여 접근 권한 설정을 디렉터리는 700, 파일은 600으로 설정 [접근 권한 설정] <pre># chmod 600 /[설치dir]/redis.conf</pre>		
진단 기준	양호 - 환경설정 파일 접근권한은 디렉터리 700, 파일 600으로 설정되어 있는 경우 취약 - 환경설정 파일 접근권한이 디렉터리 700, 파일 600 초과로 설정되어 있는 경우		
진단 방법	[진단 예시] <pre># find /[설치dir] - name redis.conf</pre> 해당 파일 내 include하여 사용중인 추가 설정 파일/폴더에 대해서도 확인 필요		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

22.4.2. .rediscli_history 파일 접근 제한

분류	환경 파일 점검	중요도	중
항목명	.rediscli_history 파일 접근 제한		
항목 설명	.rediscli_history 파일에는 명령 실행에 대한 주요 정보가 평문으로 노출될 수 있어 해당 파일을 허가 받지 않은 사용자가 읽는 경우 주요 정보를 획득할 수 있으므로 파일의 접근을 제한해야 함 (CVE-2013-7458)		
설정 방법	1. .rediscli_history 파일의 접근 권한을 600 또는 640 으로 설정 (타 사용자 모든 권한 제거) <pre># chmod 600 .rediscli_history</pre> 2. .rediscli_history 파일의 내용 제거 <pre># cat /dev/null > ~/.rediscli_history</pre>		
진단 기준	양호 - 접근권한이 600 또는 640 이하로 설정되어 있는 경우 취약 - 접근권한이 600 또는 640 초과로 설정되어 있는 경우		
진단	[진단 예시]		

방법	# ls -al ~/.rediscli_history
비고	단기 적용(적용 시 개발자 및 운영자 협의)
기반시설 기준항목	-

22.4.3. Log 파일 접근 제한

분류	환경 파일 점검	중요도	하
항목명	Log 파일 접근 제한		
항목 설명	Log 파일에는 중요 내용이 포함되어 있음. 이를 활용하여 침해 사고 시 분석 자료로 사용하고 있음. Log 파일이 비인가 자에게 읽히거나 쓰여지는 경우 로그 파일의 정보 노출 및 변조가 발생할 수 있으므로 접근을 제한할 필요가 있음.		
설정 방법	1. Log 디렉터리 및 파일의 접근 권한을 다음과 같이 설정 ■ Unix 계열 Log 디렉터리는 750, 파일은 640 이하로 접근 권한 설정 # chmod 750 [로그 디렉터리] # chmod 640 [로그 파일] [Log 파일 위치 및 찾는 방법] # cat /[설치dir]/redis.conf grep logfile		
진단 기준	양호 - 접근권한이 디렉터리 750, 파일 640 이하로 설정되어 있는 경우 취약 - 접근권한이 디렉터리 750, 파일 640 초과로 설정되어 있는 경우		
진단 방법	[진단 예시] # ls -ald [redis 로그 디렉터리] # ls -al [redis 로그 디렉터리]/[로그 파일]		
비고	단기 적용(적용 시 개발자 및 운영자 협의)		
기반시설 기준항목	-		

22.5. 보안 패치

22.5.1. 보안 패치 적용

분류	보안 패치	중요도	상								
항목명	보안 패치 적용										
항목 설명	버그로 인한 침해 사고 발생 가능하므로 주기적으로 최신 패치를 적용하여 취약점을 제거 해야 함										
설정 방법	1. 데이터베이스에 대한 최신의 버전을 확인 후 업그레이드 및 패치 수행 2. 원격 Exploit 취약점, 제로데이 취약점은 즉시 패치 〈Latest Release 2017.03.10〉 <table><tr><td>Version</td><td>Last Version</td></tr><tr><td>2.x</td><td>2.8</td></tr><tr><td>3.0.x</td><td>3.0.7</td></tr><tr><td>3..2.x</td><td>3.2.8</td></tr></table> ※ 신규 보안가이드라인 기준으로, 2.x 버전 관련 내용은 CVE 취약점만을 다룸 - 참고 자료 : https://redis.io/download ※ 서비스 및 운영상의 영향도를 확인하시어 설정하여 적용해야 합니다.			Version	Last Version	2.x	2.8	3.0.x	3.0.7	3..2.x	3.2.8
Version	Last Version										
2.x	2.8										
3.0.x	3.0.7										
3..2.x	3.2.8										
진단 기준	양호 - 최신의 버전 적용한 경우 취약 - 최신의 버전 적용하지 않은 경우										
진단 방법	[진단 예시] # /[설치dir]/src/redis-server -v										
비고	중기 적용(적용 시 개발자 및 운영자 협의)										
기반시설 기준항목	-										

EQST INSIGHT

오픈 소스 소프트웨어 보안 가이드

2018. 06



경기도 성남시 분당구 판교동 255번길 46 4층
www.skinfosec.com

발행인 : EQST Group

© 2018, SK infosec All rights reserved.

본 저작물은 SK인포섹의 EQST Group에서 작성한 콘텐츠로 어떤 부분도 SK인포섹의 서면 동의 없이 사용될 수 없습니다.