|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | KITRI 모의해킹 28기 | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | 작성: 염찬호 | | |  | |
|  | OWASP Top 10(2020) | | | | | | |  |
|  | | -Insecure Deserialization- | | | |  | | |
|  | |  | | | |  | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |

|  |
| --- |
| 1. 개요(공격개념/원리/취약점/공격 시 예상피해 등) |
|  |
| 1. Serialization 이란 ? |
| * 직렬화, 개체를 저장 및 전송 가능한 스트림 형태로 변환하는 것이다. * 어떤 개체를 직렬화 할 때 자동으로 각 속성을 할당된 값과 함께 보존 된다.(Private필드 포함) * 직렬화를 방지하고 싶은 필드가 있다면, 클래스 선언에서 해당 필드에 ‘transient’명시한다. |
| 1. Deserialization 이란? |
| * 역직렬화로써 직렬화 된 객체를 다시 객체 형태로 변환하는 것이다. |
| 1. Insecure Deserialization |
| * 보통의 역직렬화를 이용한 공격은 역직렬화가 완료되기도 전에 이미 완료가 된다. * 취약점 발생 이유 * 웹사이트 소유자들이 역직렬화 된 데이터를 검사하는 것으로 안전하다고 생각 하기 때문이다. * 개발자들이 역직렬화 된 객체가 바이너리로 변환될 경우, 해독이 어려울 것이라고 생각 하기 때문이다. * 웹 사이트 각각의 고유 종속성으로 인해 악의적 데이터의 입력 및 호출 예측이 어렵기 때문이다. |
| 1. JAVA에서의 Deserialize |
| * 객체들의 데이터를 연속적인 데이터로 변형하여 Stream을 통해 데이터를 읽도록 해준다. 주로 객체들을 통째로 파일로 저장하거나 다른 곳으로 전송할 때 사용된다. * 반대로 역직렬화(Deserialize)는 직렬화 된 데이터를 역으로 직렬화 하여 다시 객체의 형태로 만든다. 저장된 파일을 읽거나 전송된 스트림 데이터를 읽어 원래 객체의 형태로 복원한다. * ObjectInputStream과 ObjectOutputStream을 활용하여 직렬화 한다. * 코드가 어떻게 직렬화가 되는지 간단하게 확인해 본다. |
| 1. WebLogic 에서의 Deserialization 공격 메커니즘 |
| * 여기에서의 핵심은 공격자가 Weblogic 서버에서 오픈해 놓은 서비스와 제작된 Java객체를 이용해 Socket연결을 맺고, 공격자가 조작한 패킷을 서버 측에 보내어 역 직렬화 원격 명령을 실행하는 취약점이라는 것이다. * 기본 포트 (7001)을 사용하는 서버를 대상으로 공격이 이루어지지만 다른 포트를 사용 중인 경우에도 공격 된다. |
| 1. 공격 코드 분석 |
| 1. 먼저 T3 서비스 연결 및 Payload전송(RMI Connection 포트 Open)에 사용될 44553.py의 메인 함수를 확인하면 아래의 그림과 같다      * 공격 코드는 Python언어로 개발되었으며 공격 코드를 실행할 때 입력되는 매개 변수 6개(Weblogic서버 IP, Weblogic 서버 기본 포트(ysoserials)경로, 공격자 IP, RMI Connection포트, 사용 라이브러리)를 배열로 지어하여 메인 함수에서 exploit 함수에 사용할 변수로 생성한다. 그 변수를 이용해 exploit함수가 실행되는데 다음으로 해당 함수 과정을 확인해보도록 한다.  1. exploit함수의 코드를 호가인해보면 아래의 그림과 같다.      * exploit 함수에서 실행 코드의 진행 과정을 확인해 보면 a~c으로 구성된다.  1. T3서비스와의 통신을 위한 Socket을 생성한다. 2. 서버 정보를 따로 저장한 변수를 사용해 Socket을 이용하여 T3서비스와 통신 및 Socket연결을 맺는다. 3. Socket 연결이 맺어진 후에 ysoserial의 JRMPClient 라이브러리를 사용하여 RMI Connection 포트(1099)를 오픈하는 Payload를 생성한 뒤 Weblogic Server에 Socket을 통해 Payload를 전송한다. 4. Exploit 함수에서 호출되는 함수들을 보면 아래의 그림들과 같다.   T3\_handshake 함수는 서버정보를 통해 socket을 이용하여 T3서비스와 통신을 위한 함수이다.   1. Build\_t3\_request\_object 함수는(아래 그림) T3서비스와 통신 후에 socket 연결을 맺는 함수이다.      1. Send\_payload\_bojdata 함수 (아래 그림)은 Weblogic Server에 Socket을 통해 Payload를 전송하는 함수이다.      1. 공격 코드 수행 프로세스는  * 44553.py명령어 실행 * -> main()함수의 exploit() 실행 * -> exploit()함수에서 Socket 생성 * -> t3\_handshake()함수 실행, T3서비스와 통신 * -> build\_t3\_request\_objecct()함수 실행, T3와 Socket연결 * -> generate\_payload()함수 실행, RMI Connection포트 오픈 Payload 생성 * -> Send\_payload\_objdata() 함수 실행, Socket을 통해 Payload 전송 * -> RMI Connection포트를 통해 역 직렬화로 원격 명령 실행 가능 |
|  |
|  |
| 2. 공격과정 (필요시 가정이나 예상 시나리오 포함) |
| 실습 환경  공격자 PC(kali linux)192.168.0.161/24  공격 대상 PC(window 10)192.168.0.165/24 |
| 1. Weblogic Deserialization RCE |
| * Weblogic이란 Oracle사의 J2EE(java 2 Enterprise Edition)기반의 웹서버로 WAS(Web Application Server)의 사용 제품 중 하나이다. * 실습으로 알아볼 취약점은 CVE-2018-2628이다. Weblogic T3프로토콜을 이용하여 RMI(Remote Method Invocation) 연결을 맺고 공격자가 미리 정해 놓은 페이로드를 전송하여 원격 명령을 실행하는 취약점이다. * JAVA의 역직렬화 취약점을 통해 공격을 가능하게 하는 ysoserial도구와 Exploit-db를 이용하여 취약점이 존재하는 Weblogic서버에 공격을 시도한다. 위약한 Weblogic버전은 10.3.6.0, 12.1.3.0, 12.2.1.2, 12.2.1.3 버전이다. 해당 취약점 분석에서는 Weblogic 10.3.6.0 번전을 윈도우에 구성하여 진행할 것이다. * CVE란 : 공객적으로 알려진 보안 취약점을 가르키는 고유표기 (MITRE에서 관리) |
|  |
| 1. 공격 시나리오 |
| 1. 44553.py 및 ysoseria을 이용하여 T3서비스와 Socket연결 + RMI Connection 포트(1099) 오픈 2. Ysoserial을 이용하여 서버 측에 조작된 패킷을 보내어 서버 측에서 원하는 원격 명령 (리버스 텔넷 사용) 실행 3. 원격 명령을 실행한 후 시스템 정보 획득 |
| 1. 공격 전 준비 |
| 1. 공격 대상 PC에 Weblogic 10.3.6.0을 설치하기 위해   <https://www.oracle.com/middleware/technologies/weblogic-server-installers-downloads.html>  페이지로 가서 10.3.6-win32.exe다운로드를 한다. (오라클의 계정이 필요 하다)   1. 아래 이미지를 따라서 설치한다.                                                1. 공격자 PC에 역직렬화 취약점 검증 툴 인 ysoserial을 설치한다.  * <https://jitpack.io/com/github/frohoff/ysoserial/master-SNAPSHOT/ysoserial-master-SNAPSHOT.jar> * 위의 주소에서 다운 받는다.  1. 공격자 PC에 Exploit-db의 44553.py 파일을 다운 받는다.  * [Oracle Weblogic Server 10.3.6.0 / 12.1.3.0 / 12.2.1.2 / 12.2.1.3 - Deserialization Remote Command Execution - Multiple remote Exploit (exploit-db.com)](https://www.exploit-db.com/exploits/44553) |
| 1. 공격 수행 |
| 1. 공격을 위해 칼리 리눅스에서 총 3개의 터미널을 준비한다. 2. 공격자 PC에서 터미널을 실행하고 임의의 포트를 지정하여 다음과 같이 nc를 포트 대기 상태로 만든다.      1. 다운로드 받은 ysoserial을 이용하여 nc연결을 위한 페이로드와 함께 RMI Connection포트를 대기 상태로 만든다. -cp 옵션은 사용하려는 Java클래스와 라이브러리를 찾을 수 있도록 경로를 정의한다.      1. Ysoserial.jar을 압축 프로그램으로 확인해 보면, 우의 그림에서 사용 하는 java class는 ysoserial 폴더 아래 exploit 경로에 JRMPListener.class가 존재하는 것을 확인 할 수 있다.      1. Exploit-db에서 다운로드 받은 python을 이용하여 exploit공격을 실행한다. 앞에는 피해자, 즉 weblogic서버의 ip와 port를 입력하고 뒤에는 공격자의 ip와 앞서 진행했던 RMI connection을 위한 port를 입력하고 JRMPListener와 상호 작용하는 payload인 JRMPClient를 입력한다.      1. Exploit을 진행하면 payload를 weblogic서버에 전달하게 된다.      1. 아래 그림과 같이 RMI Connection 대기중인 1099 port에 서버에서 연결 신호를 보내 피해자와 연결되는 것을 확인 할 수 있다. |
|  |

|  |
| --- |
|  |
| 3. 공격 결과 |
|  |
| 1. 공격 결과 |
| 1. 8888포트로 대기중인 첫번째 터미널을 확인해보면, 피해자와 리버스 연결이 성공된 것을 확인할 수 있다. 윈도우 커맨드 systeminfo를 입력해 보면 공격 피해자PC의 정보를 받아올 수 있다 |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| 4. 보안진단 및 대책 |
|  |
| 1. 최신 버전으로 업데이트 |
| 1. 취약한 Weblogic Server버전(10.3.6.0, 12.1.3.0, 12.2.1.2, 12.2.1.3)을 사용하지 않고 최신 버전으로 업데이트하여 사용 할 것을 권고 한다. |
|  |
|  |
|  |

|  |
| --- |
|  |
| 5. 보안대책 적용시 공격결과 (최종 매뉴얼에 포함) |
|  |
| 1. 최신 패치로 인한 공격 불가 |
| 1. 역직렬화 취약점에 문제가 있던 10.3.6.0 환경에서의 readSubject 메소드 구문이 WSFilteringObjectInputStream 메소드를 이용하여 검증이 되게 패치가 되어서 해당 메소드는 역 직렬화 할 객체가 subject의 서브 클래스인지 점검하고 아니라면 에러 발생 후 역직렬화 과정을 종료시킨다. |
|  |
|  |
|  |