REPORT

1번 과제: 용어 이해, Log4j 취약점 원리와 대응

과목명 : SW보안개론 1분반
학 번 : 32200327
이 름 : 김경민
제출일 : 2022 - 4월 4일

DKU 단국대학교 DANKOOK UNIVERSITY

문제 1)

1. Botnet(용어 및 동작 방식)
 : Robot(로봇)과 Network를 합성한 신조어로 동일한 악성코드에 감염되어 중앙 통제 장치로부터 지시를 받아 조종 되어지는 디바이스들로 구성된 네트워크를 말한다.

감염된 디바이스들은 외부 중앙 통제 장치인 서버로부터 원격으로 명령을 받아 움직이는 command&control(C&C) 방식으로 동작한다. (이렇게 외부 명령을 받아 동작하는 프로그램을 백도어 프로그램, 트로이목마 바이러스라 함) 따라서 외부 서버의 통제를 받아 대규모의 디바이스들이 일사불란하게 명령을 수행하며 동작하는 경우를 봇넷이라 부를 수 있다. 이러한 동작을 수행하기 위해서는 서버와 봇넷 모두 인터넷에 연결되어 있어야 한다.
봇넷을 활용하여 시행하는 사이버 공격을 Botnet Attack 이라 하고, 대표적인 공격은 DDos Attack이다. DDos 공격은 공격 당하는 서버가 감당 불가 수준의 요청을 보내는 (강의자료 활용) 이러한 공격을 통해 공격자는 스팸 메일 발송이나 암호화폐 채굴 등의 악성 행위를 한다. 또한 Worm 바이러스를 가진 봇넷의 경우 자가증식도 가능하다.
C&C 서버 운영을 통해 동작하는 봇넷은 제 3자에게 발각되지 않도록 특정 protocol을 사용하는데 이 protocol 방식에는 3가지 방식이 있다.

IRC 기반 봇넷
다른 방식에 비해 발각되지 않을 확률은 떨어지지만 가장 보편적으로 사용한다. 실시간 채팅 방식을 통해 비밀 채널을 생성하여 사용한다. DDos공격에서 많이 사용된다.

P2P 기반 봇넷
C&C 방식으로 동작하는 봇넷의 경우 control을 하는 control tower 서버가 한 곳인 경우 해당 서버가 차단되면 봇넷 전체가 무력화 된다는 약점이 있다. 이를 보완하기 위해 봇넷 스스로가 명령어를 내리는 서버가 될 수 있는 P2P 방식도 사용한다. 새로 감염된 디바이스가 특정 서버에 접속하여 연결할 IP 주소를 가져오는 방식으로 사용된다.

HHTP 기반 봇넷
정상적인 HTTP Web Traffic 속에 섞여져 오고 가기 때문에 발각되는 것을 방지하는 기법이다. 도메인 생성 알고리즘을 사용하여 고정된 IP가 아닌 동적으로 생성한 도메인 주소를 사용하여 서버가 노출되는 것을 방지한다.

2. Cryptojacking
 : 암호화폐(Cryptojacking) 과 납치(Hijacking)을 합성하여 만든 합성 용어로 악성코드를 통해서 타인의 컴퓨터에서 암호 화폐를 몰래 채굴하는 것을 말한다.
암호화폐 채굴은 컴퓨터의 CPU나 GPU 연산능력을 이용하여 블로체인의 블록 속에 든 암호를 풀고 그 보상으로 일정 수량의 코인을 얻어가는 과정이다. 따라서 성능이 좋은 컴퓨터나 전용 채굴기를 사용할수록 유리한데 해커들은 이때 자신의 컴퓨터 자원을 사용하는 것이 아닌 다수의 컴퓨터를 해킹하여 채굴 프로그램을 설치하고 CPU 자원을 이용하여 암호 화

폐를 채굴해낸 다음 채굴한 암호 화폐는 자신의 암호화폐 지갑으로 전송하는 것이다. 악성 코드에 감염되면 하드웨어 자원의 성능과 속도가 저하되거나 손상될 수 있다.
해커는 Cryptojacking을 위해 malware나 Drive by Cryptojacking(드라이브 바이 크립토재킹) 방식을 사용한다. malware에 감염되면 사용자의 컴퓨터가 해커에게 인계 되어 조종 당하는 것이고, Drive by Cryptojacking(드라이브 바이 크립토재킹) 방식의 경우, 웹페이지에 악성 스크립트를 추가해놓고 사용자가 해당 웹브라우저에 머무는 동안에만 채굴하는 방식이다. 이 경우 암호화폐 채굴을 차단하는 브라우저 확장 플러그인을 설치하는 방법 등을 사용하여 Cryptojacking을 차단할 수도 있다.

3. SQL injection
: 악의적인 사용자가 웹사이트의 보안상 허점을 이용해 특정 SQL문을 주입하여 데이터베이스가 비정상적으로 동작하도록 조작하는 행위이다.

사용자 정보 조회 시, 조회하고자 하는 사용자의 조회명을 사용하여 url을 변경하고 웹서버가 해당 url을 인정할 수 있도록 해주는 특정 쿼리문을 추가해주면 자신 외의 다른 개인의 민감한 데이터에 무단 액세스가 발생할 수 있다,
SQL injection 공격에는 논리적 에러를 이용한 공격, UNION 명령어를 이용한 공격, 참거짓 기반 블라인드 공격, 시간 기반 공격 등이 있다.

논리적 에러를 이용한 공격
위의 예시처럼 다른 사람의 조회명을 사용할 때 비밀번호처럼 추가로 접근 인증을 해야 하는 경우가 있다. SQL 문에 password=‘ ’ or ‘1’=’1’ 같은 구문을 추가해주면 or 연산을 사용하기 때문에 password가 거짓이어도 전체 결과는 참이기 때문에 조회가 가능하게 된다.

UNION 명령어를 이용한 공격
중복 값을 제외하고 여러개의 SQL문을 합치는 UNION 명령어를 사용하면 ID와 password 목록을 전부 조회할 수 있게 된다.

참거짓 기반 블라인드 공격
Blind SQL문은 쿼리가 참일 때와 거짓일 때 서버의 반응 만으로 데이터를 얻어낼 수 있다. 따라서 쿼리를 한글자씩 끊어 아스키코드로 변환 시킨 다음 임의의 숫자와 비교해 참거짓을 비교하는 과정을 반복하여 올바른 조합을 얻어내 원하는 정보에 접근할 수 있는 쿼리를 알아내는 것이다.

시간 기반 공격
쿼리의 응답시간의 차이로 참거짓을 판별하여 얻고자 하는 정보의 쿼리 조합을 알아내는 방식이다.

4. Cross-Site Scripting (XSS)
: 웹 애플리케이션에서 일어나는 공격으로 관리자가 아닌 권한이 없는 사용자가 웹사이트의 취약점을 이용해 해당 사이트에 악성 스크립트를 삽입하는 injection 공격의 일종이다.

일반 사용자는 해당 스크립트가 신뢰할만한 웹사이트에서 존재하는 스크립트라고 생각하여 악성 스크립트인지 알아차리기가 어렵다. 사용자가 해당 악성 스크립트가 있는 게시글을 열람할 경우 본인의 쿠키가 해커들에게 전송되고 세션ID가 포함된 쿠키를 탈취한 해커는 해당 사용자의 계정으로 로그인할 수 있게 되는 것이다. 데이터를 입력할 때 혹은 입력한 데이터를 그대로 출력하여 사용자에게 보여줄 때 유효성 검사와 같은 검증을 하는 과정이 없는 웹사이트라면 이러한 허점을 노려 해당 공격이 일어날 수 있다. 따라서 입력 및 출력에 대한 검증이나 악의적인 스크립트가 실행되지 않도록 브라우저 확장 앱을 사용하여 XSS 공격에 대비해야 한다. 웹 방화벽을 사용하여 각종 injection 공격을 방어하는 방법도 있다.

[참고]
https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=aepkoreanet&logNo=221532280623
https://www.koreascience.or.kr/article/CFKO201028451823815.pdf
https://www.fortinet.com/resources/cyberglossary/cryptojacking
https://noirstar.tistory.com/264
https://m.blog.naver.com/lstarrlodyl/221837243294
https://portswigger.net/web-security/sql-injection
https://velog.io/@yanghl98/Database-SQL-Injection
https://owasp.org/www-community/attacks/xss/
https://noirstar.tistory.com/266

문제 2)
1. Attack surface란?
: 공격자가 시스템이나 네트워크 환경에 진입하는 지점, 시스템이나 환경에 영향을 미치거나 데이터를 뽑아 가게 되는 지점 혹은 해킹에 취약한 조직이나 영역의 총 집합을 의미한다. 보안공격을 수행하기 위해 악용될 수 있는 모든 취약점도 Attack surface에 포함된다. 공격자가 접근 권한만 가지고 있다면 언제든 잠재적인 공격 지점 즉, Attack surface를 찾을 수 있다.
Attack surface에는 디지털 Attack surface, 물리 Attack surface가 있다. 디지털의 경우, 애플리케이션, 코드, 포트, 웹사이트와 서버 등이 있다. default password 나 빈약한 코딩에서 취약점이 발견될 수 있다. 물리적으로는 PC, 노트북, USB 등이 있다.

2. Attack surface를 줄이는(최소화하는) 방법은?
 : 첫 번째는 조직 내에서 민감한 데이터와 리소스에 대한 접근을 제어해야 한다. 잠금, 액세스 카드, 생체 인식 및 다단계 인증 등의 조치를 통해 내부 및 외부의 접근 제어에 주의를 기울여야 한다. 두 번째는 복잡도 제거 세 번째는 정기적으로 디지털 자산과 데이터를 스캔하여 잠재적인 취약점을 찾도록 노력해야 한다. 네 번째는 방화벽과 같은 Tool을 사용한 네트워크 세분화를 진행하여 네트워크를 구획화하고 각 하위 네트워크에 고유한 보안 체계 및 서비스를 제공할 수 있도록 하는 것이다.

3. Code Review에 대해 설명하시오
: 코드 리뷰란 여러 개발자가 본인이 만들지 않은 코드의 내용을 점검하고 피드백을 주는 과정을 말한다. 오타, 버그 가능성, 개발 표준, 중복 방지 및 모듈의 재사용성에 대한 의견을 줄 수도 있고 좋은 코드에 대한 긍정적인 패드백 등도 모두 코드 리뷰가 될 수 있다. 따라서 코딩 스타일을 일관되게 유지하거나 예상 문제 파악을 일찍 끝내는 것에 그치지 않고 더 발전된 코드를 모두가 만들어가는 것이다.
코드 리뷰에도 다양한 방식이 있다. 같은 코드를 함께 개발하는 pair programming, 경험이 많은 동료를 찾아 자신의 코드를 설명하며 검토하는 방식인 over-the-shoulder, 코드 리뷰 Tool을 사용하는 방법 등이 있다.

4. Penetration testing에 대해 설명하시오
: Penetration testing 즉, 침투 테스트는 보안 전문가가 컴퓨터 시스템의 취약점을 찾아내는 보안 실습이다. 이 시뮬레이션을 통해 공격자가 침투할 수 있는 시스템의 취약점 및 약점을 미리 찾아내고 앞으로를 대비하여 방어할 수 있도록 보안 시스템을 구축해나갈 수 있다. Penetration testing를 진행하는 해커 역할은 오히려 보안 방법에 대한 지식이 거의 없는 사람이 진행하는 것이 시스템 개발자가 놓친 사각 지대를 더 잘 발견할 수 있다는 점에서 효과가 좋을 수 있다. 또한 테스트를 진행하는 방식도 여러 가지가 있다.

open-box 테스트
 테스트를 진행할 해커에게 미리 대상의 보안 정보의 일부를 제공한다.

close-box 테스트(블라인드 테스트)
테스트를 진행할 해커에게 대상의 이름 외의 배경 정보가 제공되지 않는 테스트이다.

이중 블라인드 테스트
해커가 테스트 공격을 진행하면 그 공격에 대응할 전문가 및 테스트 진행 여부에 대해서도 사전에 알지 못한 상태로 기습적으로 테스트를 진행하는 것이다.

외부 테스트
웹사이트나 외부 네트워크 서버와 같은 외부망을 공격하는 테스트이다. 또 해커는 대상 회사의 내부에 들어가지 않고 원격으로 공격을 수행하거나 근처 주차된 트럭 등에서 공격을

수행하는 방식으로 진행되기도 한다.

내부 테스트
해커가 내부 네트워크망을 통해 테스르를 진행하는 방식이다. 내부 조직에서 발생한 해킹 피해의 정도의 가늠하는데 있어서 유용하다,

[참고]
https://withnetworks.tistory.com/47
https://whatis.techtarget.com/definition/attack-surface
https://whatis.techtarget.com/definition/attack-surface
https://www.vmware.com/topics/glossary/content/network-segmentation.html
https://smartbear.com/learn/code-review/what-is-code-review/
https://www.cloudflare.com/ko-kr/learning/security/glossary/what-is-penetration-testing/

문제 3)

1. 심각한 Log4j 취약점을 3가지 설명하시오.

1-1. CVE-2021-44228
log4j 2.0-beta9~2.14.1이하 버전을 대상으로 원격 코드 실행이 가능한 critical 10.0 level의 취약점이다. Java 디렉터리 인터페이스가 "${jndi}" 명령의 특정 서식이 포함된 로그 메시지를 받으면 취약점이 발동하게 된다. 해커는 네트워크 상에서 파일, 개인정보 등을 찾아볼 수 있게 해주는 경량 디렉터리 액세스 프로토콜인 LDAP 등을 사용하여 원격으로 웹서버에 JNDI 라고 하는 Java API를 보내는데 정보를 요청한다. 그리고 웹서버가 얻고자 하는 정보를 찾기 위해 다시 해커에게 디렉토리를 요청하면 이때 해커는 악성코드를 다운로드 받을 수 있는 정보를 웹서버에 보내주게 된다.

1-2. CVE-2021-45046
log4j 2.16 이하 버전을 대상으로 한 critical 9.0 level의 해당 취약점은 log4j 취약점 대응을 위해 배포된 신규 버전인 log4j 2.15에서 발견된 새로운 취약점이다. 기본 로깅이 아닌 컨텍스트 조회 또는 스레드 컨텍스트 맵 패턴이 포함된 패턴 레이아웃을 사용하는 경우 JDI 패턴을 이용하여 입력 데이터를 조작해 DOS 공격을 일으킬 수 있는 취약점이다.

1-3. CVE-2022-23305
log4j 1.x 버전 모두를 대상으로 한 critical 98. level의 취약점으로 JDBAppender가 SQL을 매개변수로 허용하는데 이때 패턴 레이아웃의 메시지 convert가 해당 입력값에 대한 검증을 진행하지 않아 발생하는 SQL injection 취약점이다. 취약점 특성상 JDBAppender를 사용하지 않으면 취약점 영향이 없다.

2. 공격자가 이 3개의 취약점들을 악용하여 어떻게 공격할 수 있는지 설명하시오.
: 해커가 웹서버에 악성코드를 다운로드 받을 수 있는 정보를 보내 서버를 통제할 수 있는 기능을 가질 경우 해커에게 민감한 정보부터 시작해 모든 정보가 유출될 수 있고 시스템까지 파괴되는 2차 피해를 가져올 수 있다. 특히 클라우드 서비스나 apple 같은 소프트웨어에서도 사용될 수 있기 때문에 많은 개인정보가 유출될 수 있는 것이다. 또한 1-3 같은 SQL injection 취약점의 경우 응용 프로그램의 입력 필드 또는 헤더에 조작된 문자열을 입력하여 의도하지 않은 SQL 쿼리를 실행시켜 정보를 빼내거나 손상시킬 수 있다.

3. 이 3가지 취약점을 완화하거나 제거하기 위한 대응기법을 설명하시오.
: 먼저 log4j의 버전을 업데이트하는 방법이 있다. 1-1 의 경우 Java 6인 경우 2.3.1 이상으로, : Java7인 경우 2.12.3 이상으로, Java8인 경우 2.17.0 이상으로 업데이트하면 된다. 하지만 신규 업데이트가 불가능한 경우 자바 실행 인자에 시스템 속성을 추가해 메시지(-Dlog4j2.formatMsgNoLookups=true) lookup을 막는 방법을 사용하거나 환경변수 설정에서 lookup과 관련된 내용을 추가(LOG4J_FORMAT_MSG_NO_LOOKUPS=true)해주는 것도 방법이 될 수 있다. 1-3 취약점도 log4j의 버전을 업데이트 하는 것이 첫 번째 방안이다. 해당 취약점은 1.x 버전에서 발생하는 취약점이므로 2.x 버전으로 업데이트 해야 하는데 업데이트가 어려운 상황에서는 압축 프로그램등을 이용하여 JDBCAppender.class를 제거해야 한다.

[참고]
https://asecurity.dev/entry/Apache-Log4j-%EC%B7%A8%EC%95%BD%EC%A0%90CVE-2021-44228-%EC%A1%B0%EC%B9%98%EC%99%80-%ED%83%90%EC%A7%80-%EB%B0%A9%EC%95%88-%EC%A0%95%EB%A6%AC
https://www.whitesourcesoftware.com/vulnerability-database/CVE-2021-45046
https://nvd.nist.gov/vuln/detail/CVE-2021-44228
https://nvd.nist.gov/vuln/detail/CVE-2021-45046
https://nvd.nist.gov/vuln/detail/CVE-2022-23305
https://hagsig.tistory.com/61
https://hagsig.tistory.com/70

Discussion)

사실 지금까지 보안에 대해서 전혀 알지 못했고 사각 지대의 보안 취약점을 찾는다는 것이 너무 어려울 것 같아서 크게 관심을 두고 있지 않았었기 때문에 이번 과제에 나온 대부분의 용어 조사 및 질문들이 나에게는 굉장히 생소하게 다가왔다. 특히 Log4j 조사를 하면서 로깅이나 JNDI 같은 기술들이 사용된다는 내용을 찾고도 이것이 어떤 기술이고 어떻게 활용되는지 확 와닿지 않아서 힘들었던 것 같다. Log4j의 취약점을 직접 공격할 수 있는 쿼리도 찾을 수 있었는데 그런 쿼리에 대한 이해가 부족해서 그럴지도 모른다는 생각이 들었다. 그래서 이번 강의에서도 대체로 조사하면 찾을 수 있는 용어에 대한 설명보다는 실제 보안 공격의 방법 및 과정을 배우고 공격에 사용되는 명령문들을 위주로 배우보고 싶다, 또 다음 과제로 해당 내용을 간단하게 실습해봐도 재미있겠다고 생각했다.