



탐지

Security Fundamentals

© 2020, Amazon Web Services, Inc. 또는 계열사. All rights reserved.

보안 수명 주기 - 탐지를 시작하겠습니다.

교육 내용

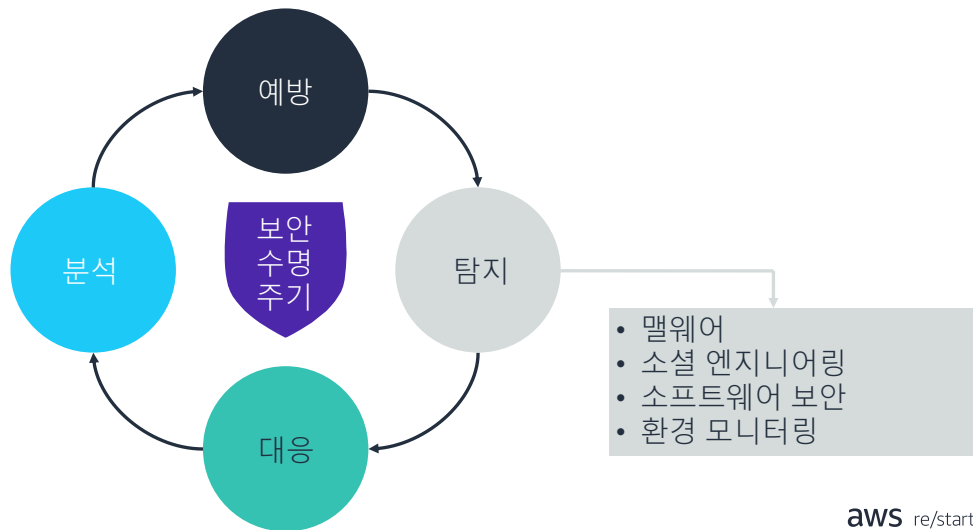
이 강의의 핵심

배울 내용은 다음과 같습니다.

- 맬웨어가 무엇인지 설명하고 악성 프로그램의 다양한 유형을 설명합니다.
- 맬웨어 위협을 예방하는 방법을 설명합니다.
- 소셜 엔지니어링의 목표를 알아보고 일반적인 소셜 엔지니어링 공격을 나열합니다.
- 소셜 엔지니어링 공격을 예방하기 위한 조치를 파악합니다.
- 애플리케이션에 영향을 미치는 보안 문제를 나열합니다.
- 불법 이용의 위험을 줄이기 위한 소프트웨어 개발 보안 원칙을 파악합니다.



보안 수명 주기: 탐지




3

aws re/start

복습하자면 보안 수명 주기는 이렇게 구성됩니다.

- 예방 - 첫 번째 방어선입니다.
- 탐지 - 예방이 실패했을 때 수행됩니다.
- 대응 - 보안 위협을 탐지했을 때 취해야 할 조치를 설명합니다.
- 분석 - 향후에 인시던트가 다시 발생하지 않도록 예방하는 새로운 조치를 구현하면서 주기가 완료됩니다.

이 강의에서는 보안 수명 주기의 **탐지** 단계를 배웁니다. 예방 단계의 일환으로 구현된 보안 조치를 통과하는 공격을 모니터링하고 탐지하는 방법을 다룹니다.



맬웨어

맬웨어란?

악성 소프트웨어(맬웨어)는 CIA의 세 가지 요소 중 하나를 방해하여 컴퓨터 시스템에 해를 끼치도록 설계된 소프트웨어입니다.

- 기밀성
- 무결성
- 가용성



맬웨어란 컴퓨터 시스템에 해를 입히는 애플리케이션으로, CIA의 세 가지 요소인 기밀성, 무결성, 가용성 중 하나 이상을 방해합니다. 맬웨어에 관해 알고, 감염을 피하는 방법과 감염된 시스템에 대응하는 방법을 아는 것이 보안 관리의 핵심입니다.

토론: 맬웨어



6

- 맬웨어의 공격을 당해본 적이 있습니까?
- 현재 컴퓨터에서 어떤 맬웨어 백신을 실행하고 있습니까?
- 현재 다른 디바이스(태블릿, 휴대폰)에서 어떤 맬웨어 백신을 실행하고 있습니까?

aws re/start

컴퓨터용 맬웨어 백신에는 McAfee, Norton, Kaspersky, AVG가 있습니다. 디바이스 맬웨어 백신에는 다음이 있습니다.

- Kaspersky Security Cloud
- Kaspersky Total Security
- McAfee LiveSafe
- McAfee Total Protection
- Bitdefender

감염 방법

신뢰할 수
없는 사이트

제거 가능한
디바이스

이메일

취약한 보호

패치되지
않은 시스템

소셜 엔지니어링

부적절한
사용자 권한

불법 복제된
소프트웨어

제거 가능한 디바이스를 사용한 감염의 예

USB 디바이스가 우편으로 도착합니다. 열어 보니 무단 사용자 또는 제3자에게 내 시스템에 대한 원격 액세스를 제공하는 백도어가 들어 있습니다.

맬웨어의 유형

- 바이러스
- 웜
- 봇
- 백도어
- 루트킷
- 스파이웨어
- 애드웨어 및 스케어웨어
- 랜섬웨어

맬웨어의 유형:

- **바이러스** - 바이러스는 시스템 애플리케이션에 붙어 정상적인 프로그램이 실행될 때마다 실행됩니다.
- **웜** - 바이러스와 다릅니다. 웜에는 실행 파일이 없고 애플리케이션의 취약성에 의존하여 배포됩니다. 웜을 사용하면 작성자가 감염된 컴퓨터를 원격으로 제어할 수 있습니다. 빠르게 확산되기 때문에 격리하기가 어렵습니다. 예: **Morris, MyDoom, Sobig, Stuxnet**
- **봇** - 컴퓨터를 제어하거나 취약한 시스템에 분산 서비스 거부(DDoS) 공격을 시작하는 데 사용됩니다. 예: **Poison Ivy**
- **백도어** - '트로이 목마'로도 알려진 백도어는 피해자의 시스템에서 정보를 훔치는 비밀 서버인 경우가 많습니다. 백도어를 사용하여 시스템에 침입할 수 있습니다. 시스템과 네트워크를 스캔하여 트래픽 패턴을 찾으면 백도어에 관해 알 수 있습니다. 예: **Sub7, GirlFriend, wack-a-mole, Zeus**
- **루트킷** - 루트킷은 정체를 노출할 수 있는 시스템 파일을 대체함으로써 정체를 숨깁니다. 정보를 검색하는 데 사용됩니다. 운영 체제의 일부가 될 수 있기 때문에 찾아서 제거하기가 어렵습니다. 대부분의 경우 제거하려면 시스템을 포맷해야 합니다. 예: **Hacker Defender**

- **스파이웨어** - 스파이어웨어는 개인 정보를 위험에 빠뜨리고 일반적으로 흥미로워 보이는 무료 애플리케이션에 숨어들어옵니다. 사람들이 금융과 기타 개인 활동을 온라인에서 점차 더 많이 수행하기 때문에 이런 활동이 탐지되고 노출될 수 있으며 정보를 도난당할 수 있습니다. 예: Real-time spy
- **애드웨어 및 스케어웨어** - 애드웨어는 광고 콘텐츠를 배포하고 방문한 웹사이트와 같은 사용자 활동을 모니터링합니다. 스파이웨어와 비슷하지만 광고와 클릭되는 콘텐츠에 집중합니다. 애드웨어는 셰어웨어 애플리케이션에 숨어들어오는 경우가 많습니다. 예: 스파이웨어 툴바, Conduit Search
- **랜섬웨어** - 랜섬웨어는 사용자가 대가를 지불할 때까지 시스템을 잠그거나 데이터를 사용하지 못하게 합니다.

바이러스

바이러스는 감염시키는 시스템에서 다른 애플리케이션을 장악하는 애플리케이션입니다.

바이러스의 유형:

- 직접 작용 바이러스
- 다형성 바이러스
- 논리 폭탄
- 메모리 상주 바이러스

바이러스의 유형

- **직접 작용 바이러스** - 코드가 실행될 때마다 파일 또는 프로그램을 감염시키기 위해 즉시 공격하는 바이러스입니다.
- **다형성 바이러스** - 자체 암호화되어 탐지를 피하는 바이러스로, 스스로 조금 변형된 사본을 복제합니다.
- **논리 폭탄** - 의도적으로 소프트웨어 시스템에 코드를 배치하여 특정 요구 사항이 충족되면 악의적인 함수를 실행하는 바이러스입니다.
- **메모리 상주 바이러스** - 컴퓨터의 메모리에 설치되어 숨어 있는 바이러스입니다. 실행된 후에 감염시킬 또 다른 파일 또는 프로그램을 찾습니다.

기타 바이러스의 유형:

- **클러스터 바이러스** - 다른 소프트웨어 프로그램의 구현에 연결되는 바이러스입니다.
- **캐비티 바이러스** - 파일 내의 빈 공간에 연결하여 감염시키는 파일 속에 설치하려고 시도하는 바이러스입니다.

바이러스의 예: ILOVEYOU, Klez, Chernobyl, Anna Kournikova, Flame, Michelangelo

Stuxnet은 '성공적인' 타겟 맬웨어의 예입니다. 이란의 핵 프로그램에 피해를 주기 위해 미국과 이스라엘 정부의 협력으로 만들어졌을 가능성이 있습니다.

자세한 정보는 Broadcom의 'W32.Stuxnet Dossier' 기술 문서 (<https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/security-response-w32-stuxnet-dossier-11-en.pdf>)를 참조하십시오.

대응 조치

- 사용자 인식 프로그램을 구현합니다.
- 사용자 권한을 제어합니다.
- 바이러스 백신 또는 스파이웨어 백신을 업데이트합니다.
- 주기적으로 시스템을 스캔합니다.
- 방화벽을 설치 및 구성합니다.
- 네트워크 활동을 모니터링합니다.



바이러스
백신



스�파이웨어
백신



방화벽

맬웨어 탐지와 예방에는 사용자 인식 및 계층화된 보호 조치가 중요합니다. 하나의 대응 조치를 모든 상황에 적용할 수 없습니다.

대응 조치(계속)

- 파일 무결성을 확인합니다.
- 시스템을 강화합니다.
- 침입 테스트를 수행합니다.
- 기준을 구현합니다.
- 물리적 보안을 구현합니다.
- 정책을 수립합니다.
- 수신 통신을 스캔합니다.



침입 탐지 시스템(IDS)

시스템을 맬웨어로부터 보호하기 위해 기술적, 물리적, 관리적 대응 조치를 사용할 수 있습니다.

침입 탐지 시스템(IDS)



모니터링



알림

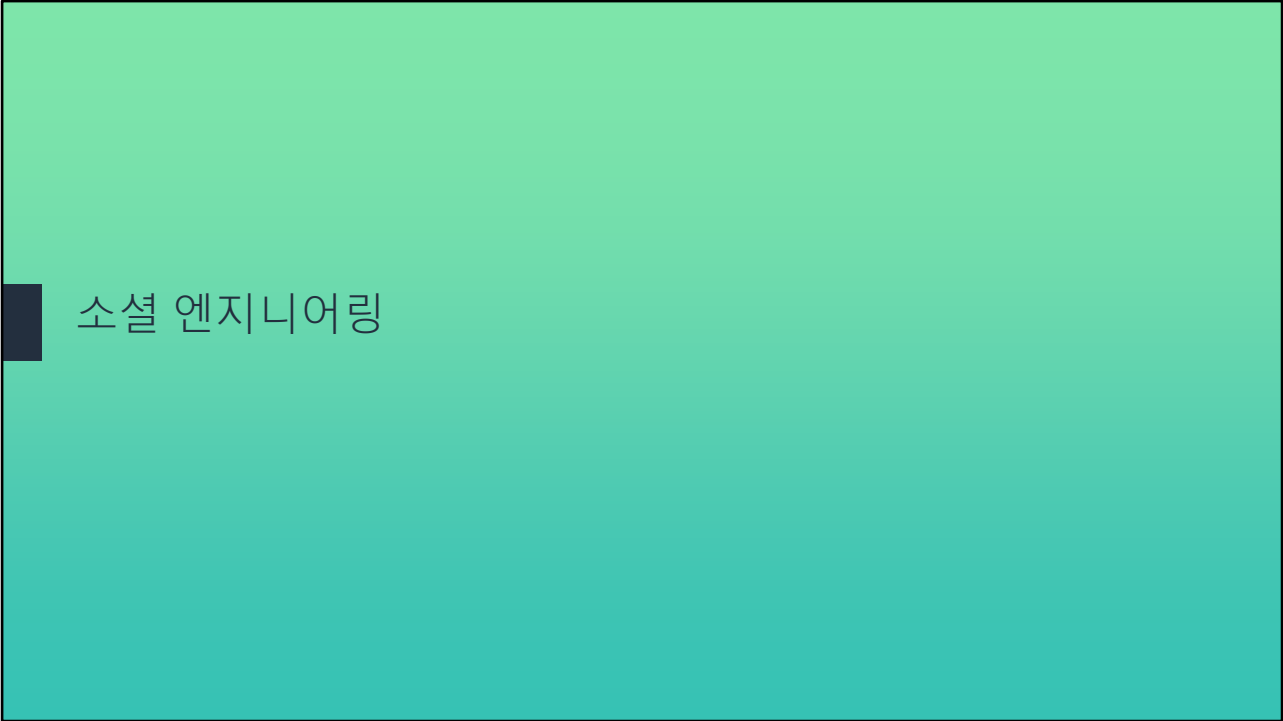


보고

모니터링 - 네트워크 또는 시스템에서 이상 활동을 모니터링하는 소프트웨어 애플리케이션 또는 디바이스

알림 - 악의적인 활동이 발견되면 관리자에게 경고를 보냄

보고 - 소스 주소, 피해자 주소, 공격 유형 등의 정보를 보고함



소셜 엔지니어링

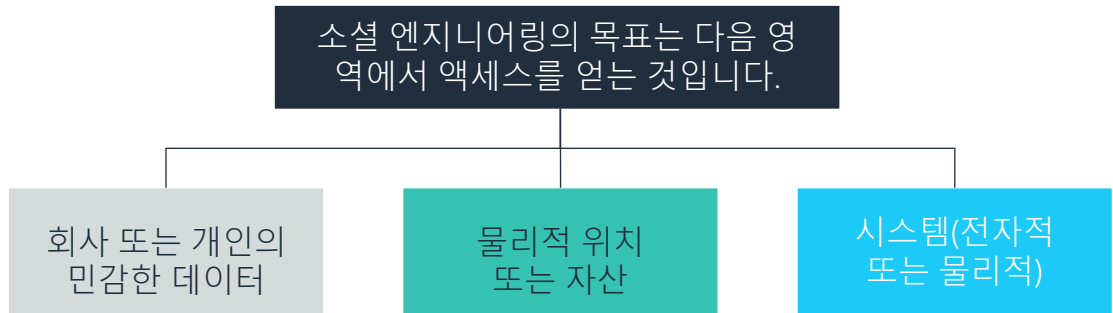
소셜 엔지니어링이란?

소셜 엔지니어링은 개인을 속여 관리적, 기술적, 물리적 제어 조치를 우회하려는 시도를 의미합니다.

IT와 컴퓨터 맥락에서 소셜 엔지니어링은 개인을 속이거나 조종해서 관리적, 기술적, 물리적 제어 조치를 우회하려는 인간의 시도 또는 상호 작용을 의미합니다. 소셜 엔지니어링의 목표는 사람을 설득하여 권한이 있는 데이터, 시스템 또는 시설에 대한 액세스를 포기하게 하거나 민감한 정보를 노출하는 것입니다.

기술이 아닌 사람에 중점을 두는 공격 및 불법 이용 방법입니다.

소셜 엔지니어링의 목표



소셜 엔지니어링이 가능한 이유

소셜 엔지니어링은 다음과 같은 상황을 이용합니다.

- 나쁜 습관 또는 게으른 습관
- 재정적 또는 기타 불이익의 부재
- 보안 유출을 추적할 역량의 부재
- 직원의 주인의식 부재
- 기존 정책 또는 기타 제어 조치의 미시행
- 교육 또는 도구 불충분

소셜 엔지니어링은 다음과 같은 상황을 이용합니다.

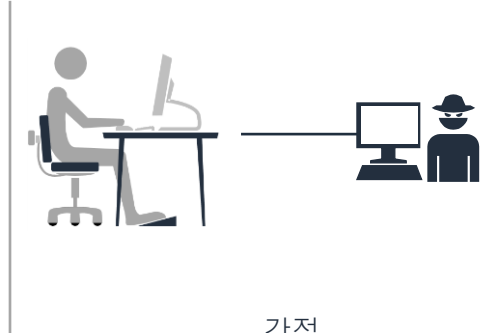
- 정상적인 보안 절차를 지키지 않는 나쁜 습관 또는 게으른 습관
- 정보를 포기하는 데 재정적 또는 기타 불이익이 없음
- 보안 유출을 추적하여 특정 직원을 밝혀낼 역량이 없음
- 직원의 주인의식 부재(조사할 때쯤에 나는 여기 없을 거야)
- 기존 정책 또는 기타 제어 조치의 미시행
- 소셜 엔지니어링에 대응하기 위한 교육 또는 도구 불충분

소셜 엔지니어링 공격

- 소셜 공격은 시스템과 보호되는 정보에 액세스하기 위한 첫 단계로 사용되는 경우가 많습니다.
- 소셜 공격은 직접적이거나 간접적일 수 있습니다.



직접



간접

소셜 공격은 무단 사용자가 물리적 또는 기술적 제어 조치의 첫 번째 수준을 통과하도록 합니다.

소셜 엔지니어링 공격의 유형:

- **직접** - 무단 사용자가 현장에 있으며 공격에 직접 가담합니다. 예:
 - 음식 또는 꽃 배달
 - 채용 면접
 - 다른 사람을 따라 들어옴
 - 쓰레기통 뒤지기
 - 영업 회의
 - 책상에 놓인 디바이스를 들고 감
- **간접** - 무단 사용자가 타겟이 있는 현장에 없거나 타겟이 정보를 포기하려고 결정할 때 무단 사용자가 공격에 적극적으로 가담하지 않는 경우입니다. 예:
 - 스팸 또는 피싱
 - 기술 지원
 - 핑계(공격자가 액세스를 얻기 위해 만들어 낸 상황을 이용하거나 다른 사람인 척함)
 - 제품 데모 또는 샘플
 - 채용 사이트에 도움을 구하는 공고를 게시함

피싱

피싱 공격은 비용이 적게 들고 제한된 대응으로 성공할 수 있습니다.



피싱에 관한 자세한 내용은 [PhishTank 웹 사이트](#)를 참조하십시오. PhishTank는 인터넷상의 피싱과 관련한 데이터 및 정보의 협력 정보 처리 기관입니다. 개발자와 연구자를 위해 애플리케이션에 데이터 피싱 백신을 통합할 수 있도록 무료로 오픈 API를 제공합니다.

이 사이트의 기능:

- 피싱 사이트로 의심되는 사이트가 있는 경우 URL을 입력하여 PhishTank 데이터베이스를 검색하면 피싱 사이트가 맞는지 확인할 수 있습니다. URL이 데이터베이스에 없으면 직접 추가할 수 있습니다.
- 사이트의 FAQ 페이지에서는 사이트에 관한 일반적인 정보와 함께 기능을 사용하는 방법에 관한 정보도 제공합니다.

소셜 엔지니어링 방지

교육은 소셜 엔지니어링 공격에 가장 효과적인 방어 수단입니다. 그러나 교육으로는 방지할 수 없는 경우가 있습니다.



토론: 소셜 엔지니어링



20

회사에서의 상황:

- 내가 있는 구역에서 직원 배지가 없는 모르는 사람을 발견한다면 신원을 물어보시겠습니까?
- 직원 배지는 없지만 내가 아는 사람이라면 어떻게 하시겠습니까?
- 조직에서 높은 위치에 있는 사람이라면 어떻게 하시겠습니까?

aws re/start

여기에 설명된 상황을 생각해 보십시오. 소셜 엔지니어링 상황과 관련이 있을 수 있습니다.

사이버 인식: 정책 및 절차

- 직원에게 정책을 교육하고 정책을 시행합니다.
- 정책을 항상 우선시합니다.
- 경영진을 관여시킵니다.
- 규정 미준수 시 조치를 시행합니다.

규정 준수 시행

- 화면 보호기
- 회사 배지 분실
- 회사 배지 착용
- 앞사람을 따라 들어오는 행위



규정 미준수에 불이익을 적용하면 정책을 무시하지 않도록 할 수 있습니다.

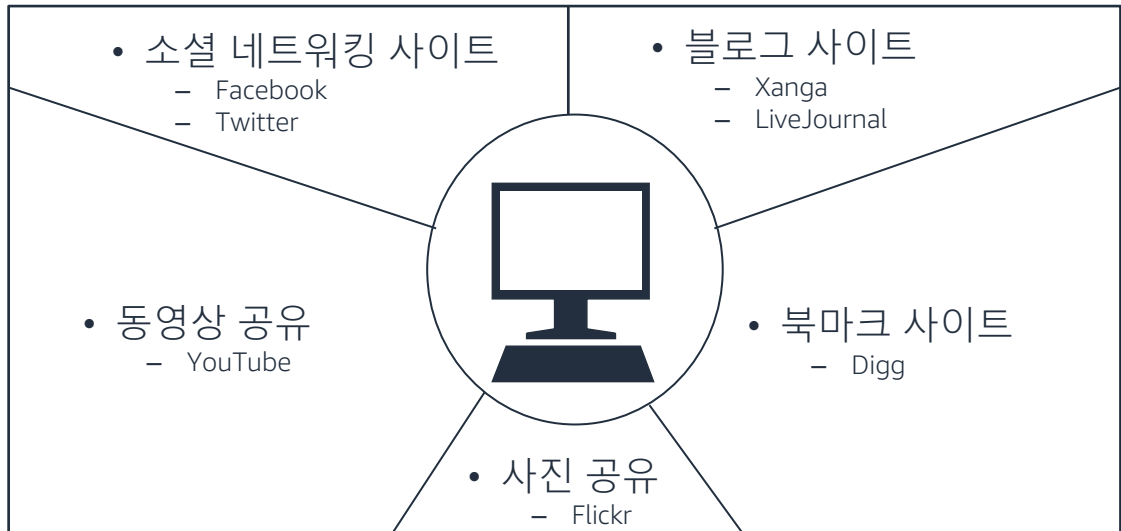
소셜 미디어

소셜 미디어는 소셜 엔지니어링이 성행하는 또 다른 영역입니다.

구독하거나 사용하는 소셜 미디어 사이트가 몇 개인지 생각해 보십시오.

- 얼마나 활발히 사용하십니까?
- 더 중요한 질문을 하겠습니다. 직원이 소셜 미디어에 업무 관련 정보를 게시하지 못하도록 하는 정책이 조직에 있습니까?
- 그렇다면 누가 모니터링합니까?

소셜 미디어 사이트의 유형



소셜 미디어의 취약성

어떤 소셜 미디어가 소셜 공격에 취약한가요?

모두 취약합니다.

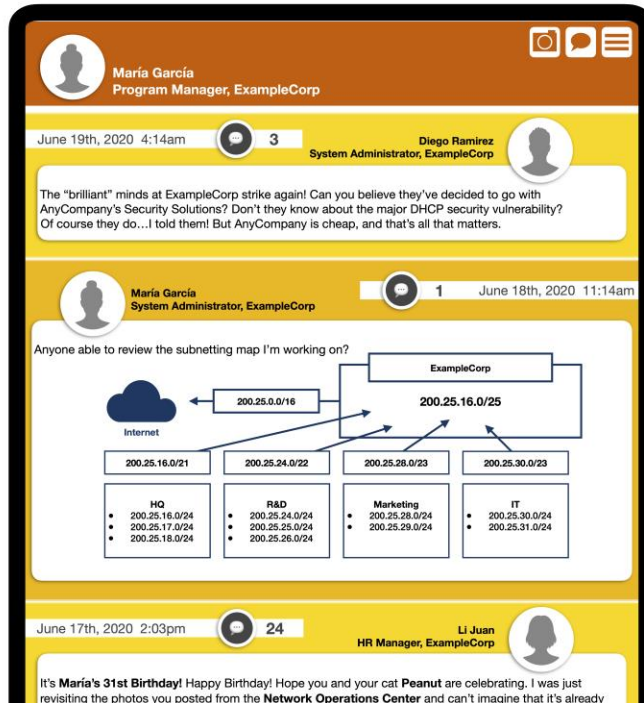
대부분의 공격은 소셜 공격으로부터 시작합니다.

모든 소셜 미디어가 소셜 엔지니어링 공격에 취약합니다.

무단 사용자의 태도

- 무단 사용자
 - 들어와서 해를 입혀도 되는지 묻지 않음
 - 제어 조치를 위반하는 것을 신경 쓰지 않음
 - 사용자의 업무가 늘어나는 것을 신경 쓰지 않음
 - 고객의 개인 식별 정보(PII)를 보호하는 것을 신경 쓰지 않음

예:



26

aws re/start

소셜 미디어 콘텐츠를 보호하는 방법

이 소셜 미디어 앱에서 부적절하거나 정보가 노출될 가능성이 있는 부분을 찾아 보시겠습니까?

취약한 부분은 다음과 같습니다.

- 첫 번째 게시물에서 Diego Ramirez는 ExampleCorp에서 사용하는 **보안 회사**를 공개합니다. 그 회사의 알려진 보안 취약성에 대해 이야기합니다.
- 두 번째 게시물에서는 **IP 주소를 비롯해** 자신이 다니는 회사 네트워크의 구조에 관한 기밀 정보를 공유합니다.
- 휴대폰 아래쪽에 있는 세 번째 게시물에서는 게시물을 올린 **6월 17일이 Maria의 생일**이라고 언급합니다. 또한 Maria에게 **Peanut**이라는 **고양이**가 있다고 말합니다. 이러한 개인 정보는 암호를 추측하려는 시도에 사용될 가능성이 있습니다. 6월 17일이 Maria의 31번째 생일이라는 것을 알기 때문에 Maria가 1989년에 태어났다는 것을 짐작할 수 있습니다. 사람들은 반려동물의 이름, 생일 및 그와 유사한 개인 정보를 조합하여 암호를 만드는 경우가 많습니다. Maria의 암호가 'Peanut89'일 수도 있을까요?
- 세 번째 게시물에서는 Maria가 ExampleCorp의 네트워크 운영 센터(NOC)의 사진을 게시했다고도 언급합니다. 이 사진에는 비공개로 보호해야 할 민감한 정보가 포함되었을 수도 있습니다.

토론: 소셜 미디어



27

- 사람들이 소셜 미디어에 민감한 정보를 게시하는 이유는 무엇입니까?
- 사람들이 소셜 미디어에 곤란하게 만드는 이미지를 게시하는 이유는 무엇입니까?

aws re/start

이 질문에 답이 될 수 있는 것은 다음과 같습니다.

- 돈
- 악명
- 명성
- 자부심

프로필 관리

- 프로필 정보가 보호된다고 추측하지 마십시오.
- 많은 이벤트가 특히 프로필 정보를 타겟팅합니다.
- 얼마나 많은 정보를 포함해야 합니까?
 - 정보가 필요합니까?
 - 정보를 수정해야 합니까?
- 소셜 미디어 공급자에게 사용자는 고객이 아니라 제품입니다.

소셜 미디어 프로필에 저장하는 정보의 유형에 관해 생각해 보는 것도 중요합니다.

소프트웨어 보안

개인을 타겟팅하는 보안 위협이 있는 것처럼 우리가 개발하는 소프트웨어에 관한 보안 우려도 있습니다. 소프트웨어 개발에서 어떤 예방 조치를 취해야 하는지 인식하면 독점 코드에서부터 민감한 인프라 정보, 심지어 고객 데이터까지 이런 위협의 액세스를 예방할 수 있습니다. 여기에서는 소프트웨어 보안을 위한 일반적인 방법을 논의합니다.

소프트웨어 엔지니어

소프트웨어 엔지니어링은 다음을 위한 원칙입니다.

- 소프트웨어 설계
- 소프트웨어 개발
- 소프트웨어 구현
- 소프트웨어 유지 관리



구조화된 접근법을 사용하면 솔루션의 전체 수명 주기에 보안이 구현되도록 할 수 있습니다.

소프트웨어 엔지니어링은 소프트웨어 솔루션의 설계, 개발, 구현, 유지 관리를 돕기 위해 방법론적으로 적용되는 원칙을 의미합니다.

소프트웨어 엔지니어링의 핵심적인 측면 중 하나는 애플리케이션 개발에 사용되는 전략입니다. 프로그램 개발에는 다양한 기법을 사용할 수 있지만 전체 솔루션 수명 주기에서 보안을 유지하려면 구조적인 접근법을 취해야 합니다.

소프트웨어 개발 수명 주기

- SDLC는 소프트웨어 개발에 사용되는 단계입니다. SDLC에는 다음 단계가 포함됩니다.



31

소프트웨어 개발 수명 주기의 아주 초기 단계에서 보안을 고민하십시오.

보안 가이드라인



32

aws re/start

애플리케이션 개발과 관련한 위험을 최소화하려면 이러한 보안 원칙을 따라야 합니다.

- **변경 관리:**
 - 원하는 변경 사항과 관련하여 지정된 작업을 완료하기 위해 준비, 지원, 후속 작업을 개인, 팀, 조직에 전달하는 프로세스입니다. 작업을 완료하는 과정 전체에서 공통의 프로세스와 필요한 투명성이 구현 되도록 보장합니다.
- **업무 분리:**
 - 소프트웨어의 개발과 전달에 참여하는 한 사람 또는 한 팀이 가진 권한의 양을 제한합니다.
- **피어 리뷰:**
 - 소프트웨어 검수의 유형으로, 작성자와 한 명 이상의 동료는 작업물의 기술적인 콘텐츠와 품질을 평가하기 위해 제품을 검사하는 것입니다.
- **프로덕션 및 개발 팀:**
 - 각 팀이 서로에게 빌드하기로 한 제품의 품질과 그에 따른 진행 사항에 대해 피드백을 제공하도록 합니다. 개발되는 소프트웨어의 성능이 요구를 충족하도록 하는 데 두 팀의 관계가 중요합니다. 필요한 보안 파라미터 내에 이 과정을 포함해야 합니다.
- **품질 보장:**
 - 최종 클라이언트에게 특정 수준의 품질을 보장하며 소프트웨어 개

발 팀이 프로세스의 초기에 문제를 파악하도록 돕습니다.

- **프로그래머의 배경 확인:**
 - 참여하는 팀원이 검증되고 개인의 동기를 강요하거나 숨길 수 있는 잠재적인 영향을 받지 않도록 합니다.
- **코드 에스스로:**
 - 코드의 유지 관리를 우선시하여 무시되지 않도록 합니다. 코드가 방치되어 보안 위협이 되지 않도록 예방합니다.

소프트웨어 취약성

취약성은 솔루션 아키텍처의 다양한 계층, 즉 프론트엔드, 비즈니스 로직, 백엔드 등에 존재합니다.

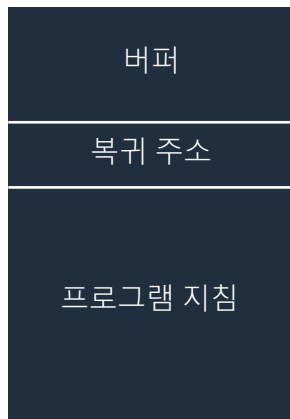
- 버퍼 오버플로
- 데이터베이스 주입 공격
- 크로스 사이트 스크립팅(XSS)
- 디렉터리 순회
- 보안 구성 오류
- 권한 문제
- 세션 하이재킹

다음 주제에서는 이런 취약성을 더 자세히 다룹니다.

버퍼 오버플로

- 애플리케이션이 경계 확인으로 보호되지 않았을 때 생성됩니다.
- 버퍼 오버플로로 인해 나타날 수 있는 두 가지 결과:
 - 애플리케이션 실행이 중단됩니다.
 - 애플리케이션의 메모리 상태가 변경됩니다.

정상적인 운영
프로그램 메모리 스택



버퍼 오버플로 공격
프로그램 메모리 스택



무단 사용자는 버퍼 오버플로 취약성을 이용하여 실행되는 프로그램에 악성 코드를 삽입할 수 있습니다. 이런 방법으로 무단 사용자는 보호받는 정보에 액세스하거나 프로그램을 제어할 수 있게 됩니다.

데이터베이스 주입 공격

- 데이터베이스 주입 공격은 프론트엔드 메커니즘을 통해 백엔드 시스템에 악성 데이터를 도입하는 공격입니다.
- 대응 조치:
 - 코드 검토
 - 웹 애플리케이션 방화벽
 - 입력 완전 삭제
 - 퍼즈 테스트

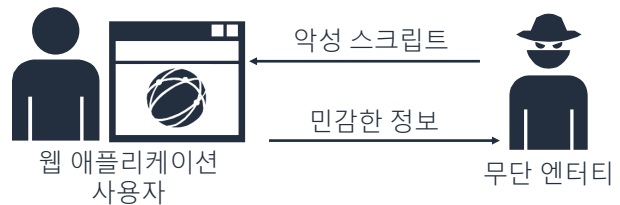


데이터베이스 주입 공격은 웹 페이지 또는 다른 프론트엔드 인터페이스를 통해 백엔드 데이터베이스 시스템을 변경하려고 합니다. 이 취약성은 입력 검증을 하지 않거나 입력 검증이 부족한 데서 유발됩니다. 결국 코드 리뷰를 하지 않았거나 개발 프로세스의 초기에 보안을 고려하지 않았다는 것을 나타냅니다.

- 잠재적인 취약성을 찾는 면밀한 **코드 리뷰**를 구현할 수 있다면 보안 조치를 취해야 할 부분을 찾는 데 도움이 됩니다.
- **웹 애플리케이션 방화벽**을 구현하면 허용된 트래픽만 데이터에 액세스하도록 보장하는 데 도움이 되며, 민감 데이터에 액세스할 수 있는 것과 액세스할 수 없는 것에 대한 제어 조치를 취할 수 있게 됩니다.
- 데이터 입력이 '완전 삭제'(sanitize)되도록 하여 알려진 표준을 충족하며 형식이 올바른 데이터만 민감한 프로세스를 통과하도록 합니다.
- 퍼즈 테스트는 형식이 잘못된 데이터, 예기치 않은 데이터 또는 임의의 데이터로 소프트웨어를 테스트하여 예상된 입력 표준을 충족하지 않는 데이터를 처리하지 않도록 소프트웨어를 인식시키는 프로세스입니다.

크로스 사이트 스크립팅

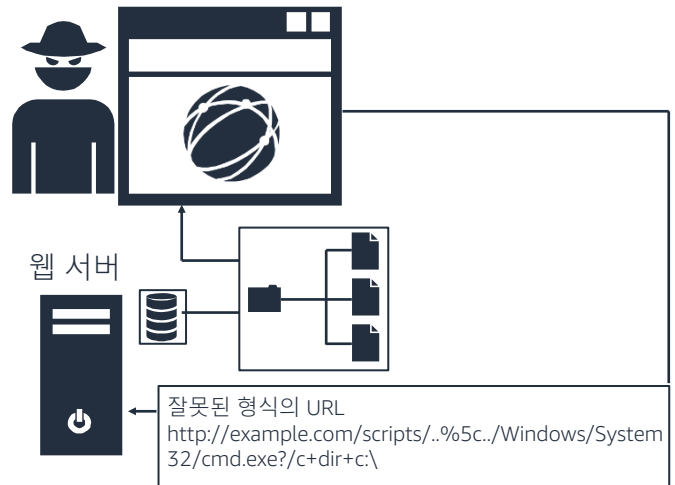
- 취약한 웹 애플리케이션은 양식을 통해 코드 주입이 가능합니다.
- 코드는 새로운 사용자가 페이지에 연결할 때마다 실행됩니다.
- 대응 조치:
 - 보안 엔진(방화벽 또는 IDS)을 통해 양식 보호
 - 스크립트 실행 제한
 - 침투 테스트와 취약성 평가 수행
 - 악성 스크립트 실행을 차단하도록 웹 브라우저 보안 구성



크로스 사이트 스크립팅(XSS)은 클라이언트 컴퓨터에서 악의적인 활동을 수행하려는 클라이언트 측 공격입니다. 무단 엔터티가 악성 스크립트를 주입하고 브라우저에서 웹 페이지를 로드할 때 실행되도록 합니다.

디렉터리 순회

- 취약성으로 인해 무단 사용자가 웹 서버에서 웹 사이트 디렉터리 외부를 탐색할 수 있습니다.
- 공격에 잘못된 형식의 URL을 이용합니다.
- 패치 서버로 문제를 완화할 수 있으며 안전한 코딩 관행을 적용합니다.

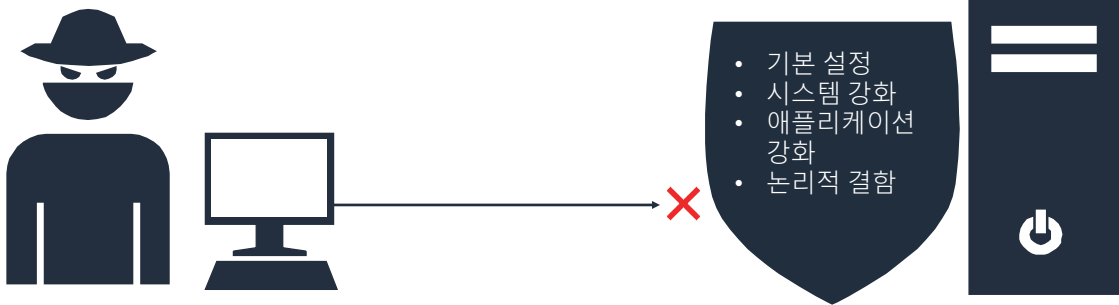


디렉터리 순회는 무단 엔터티가 사이트의 루트 폴더 외부로 이동하여 시스템의 다른 폴더를 탐색하는 것입니다.

효과적인 해결 방법은 특정 URL 형식을 차단하고 시스템을 패치하는 것입니다.

보안 구성 오류

서버를 잘못 구성하면 공격 가능성이 높아집니다.

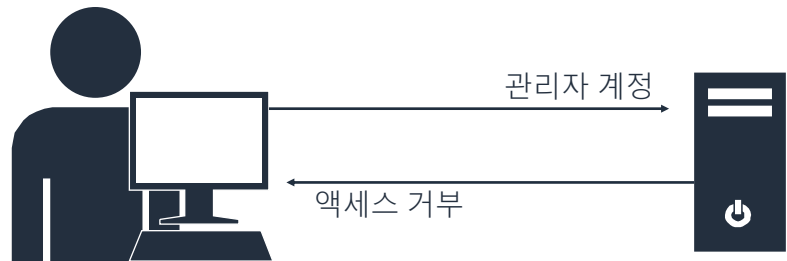


보안 제어 조치는 서버 또는 하나의 소프트웨어 구성에서 제공되는 경우가 많습니다. 그러나 제어 조치를 사용 설정하지 않거나 제대로 구성하지 않으면 서버 또는 소프트웨어가 취약해질 수 있습니다.

웹 서버 수준과 웹 애플리케이션 수준에서 보안을 최적화해야 합니다. 자세한 내용은 [Open Web Application Security Project\(OWASP\) 웹 사이트](#)를 참조하십시오.

권한 문제

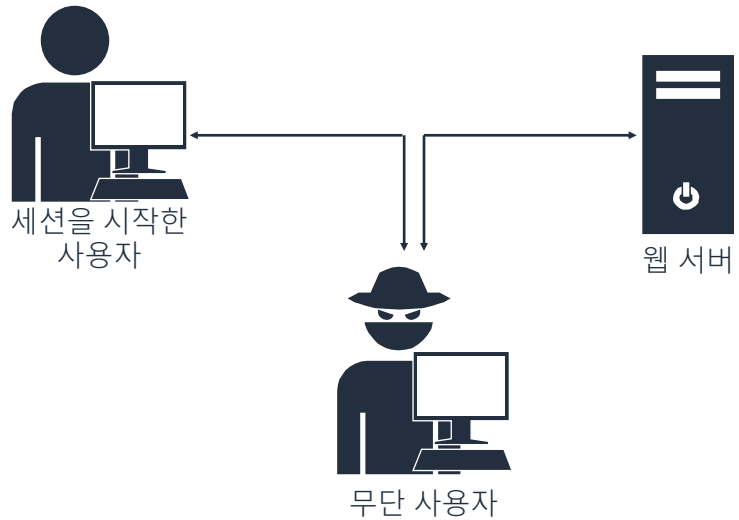
- 액세스 제어 목록(ACL)을 정의하여 애플리케이션에서 사용하는 보안 디렉터리를 정의합니다.
 - 디렉터리 브라우징을 통한 정보 노출의 위험이 있음
- 해킹에 대한 시스템의 노출을 줄이기 위해 권한을 축소합니다.



원격 시스템에 액세스할 수 있는 권한을 너무 많이 부여하면 시스템이 공격에 취약해질 수 있습니다. 권한을 정의할 때 최소 권한의 원칙을 사용하십시오.

세션 하이재킹

- 무단 엔터티가 기존 세션을 장악하도록 합니다.
- 두 가지 정보를 발견합니다.
 - 세션 ID
 - 세션 쿠키
- 대응 조치:
 - 세션 시간제한 및 재설정
 - 예측할 수 없는 세션 ID를 사용하고, 재사용하지 않음
 - 영구 쿠키를 사용하지 않음



세션 강탈 공격에서는 **중간에 있는 공격자**가 수신 및 송신 HTTP 요청을 가로채고 변경해서 세션을 장악합니다. 무단 사용자는 세션을 처음 시작한 사용자의 세션 ID와 쿠키를 사용하여 인증된 사용자를 사칭합니다.

세션 하이재킹의 위험을 완화하는 데는 권한에 짧은 시간제한을 정의하고 브라우저에서 일시적 쿠키를 사용하는 방법이 있습니다.

핵심 사항



© 2020, Amazon Web Services, Inc. 또는 계열사. All rights reserved.

41

- 바이러스 백신 또는 스파이웨어 백신 프로그램, 방화벽, 침입 탐지 시스템(IDS)을 비롯한 여러 계층의 보호 조치를 사용하여 맬웨어를 효과적으로 탐지하고 해결합니다.
- 소셜 엔지니어링을 예방하는 데 직원 교육과 정책 및 절차 수립이 중요합니다.
- 보안 문제를 초기에 해결하고 소프트웨어 개발 수명 주기 전체에서 해결합니다.

aws re/start

이 강의에서 다룬 핵심 사항은 다음과 같습니다.

- 바이러스 백신 또는 스파이웨어 백신 프로그램, 방화벽, 침입 탐지 시스템(IDS)을 비롯한 여러 계층의 보호 조치를 사용하여 맬웨어를 효과적으로 탐지하고 해결합니다.
- 소셜 엔지니어링을 예방하는 데 직원 교육과 정책 및 절차 수립이 중요합니다.
- 보안 문제를 초기에 해결하고 소프트웨어 개발 수명 주기 전체에서 해결합니다.