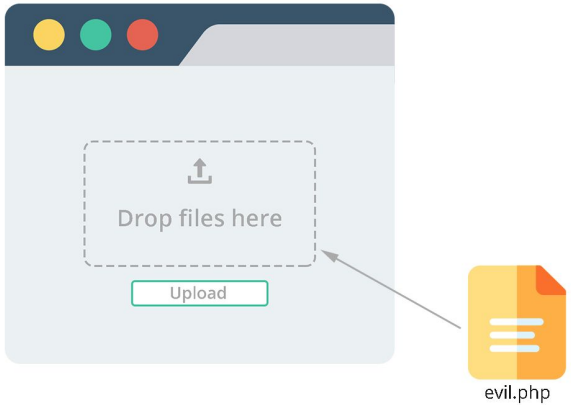
File upload 취약점을 활용한 Ransomware 공격 및 대응

Team GoodNight
21800213 김휘진
22000216 나보림
22100579 이진주

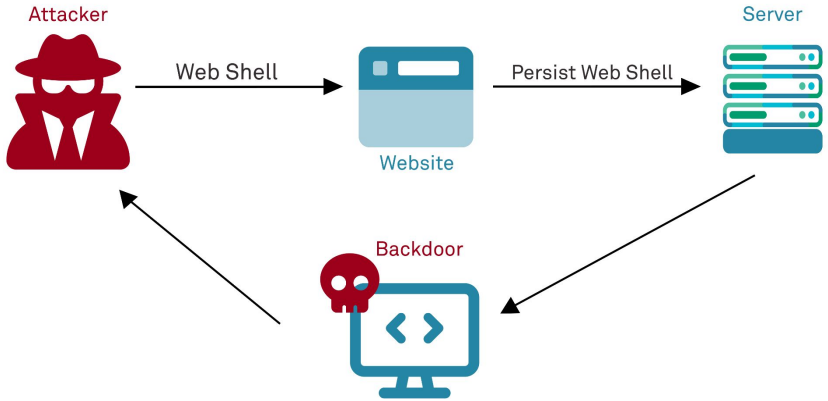
목차

File upload vulnerability

: 웹 서버가 사용자가 업로드하려는 파일에 대해 충분히 검증하지 않고 업로드를 허용하는 경우 발생하는 취약점



파일 업로드가 가능한 모든 서버에 잠재적 위협



공격유형 중 하나인 웹쉘을 활용한 RCE 공격

File upload vulnerability: 사례

中 해킹 공격 '웹호스팅 서비스' 집중… 무방비 사이트 널려

발행일 : 2023-01-29 15:00 지면 : 2023-01-30 1면

외부에 노출된 홈페이지의 SQL 인젝션·파일 업로드 취약점을 노렸다. SQL 인젝션은 데이터베이스 관리 언어 SQL을 활용, 웹사이트 취약점을 찾은 후 데이터베이스를 조작하는 공격 방식이다. 거점 확보에 사용한 웹셸은 디렉터리 조작, 파일 다운로드·업로드가 가능한 종합 웹셸 등인 것으로 확인됐다. 웹셸은 웹서버 장악을 목적으로 업로드 취약점을 노리는 악성코드다.

https://www.etnews.com/20230127000133

한국 사이트 100여곳 홈페이지 변조… 왜?

발행일 : 2018-08-27 14:31 지면 : 2018-08-28 2면

지난해 중국 해커 조직은 고고도미사일방어체계(THAAD·사드) 설치 보복으로 한국 기업 홈페이지를 변조했다. 홈페이지 변조 공격은 통상 WebDAV나 파일 업로드 취약점을 이용해 이뤄진다. KISA가 발표한 웹서버 보안 강화 안내서에 따르면 WebDAV는 윈도 환경에서 IIS 설치 시 기본 설치된 원격관리 서비스다. 잘못된 보안 설정으로 홈페이지 디렉토리에 쓰기 권한이 있으면 공격에 노출된다. 공격자가 WebDAV를 사용해 임의 파일을 업로드하는 방식으로 웹 콘텐츠를 변조한다.

https://www.etnews.com/20180827000226

매우 고전적이고 전형적인 기법으로 평가되나 국내 다양한 피해 사례가 발견되고 있음

File upload vulnerability: 사례

워드프레스 플러그인 취약점으로 100,000개 이상의 사이트 해킹

입력 : 2015-01-03 14:14

웹셸을 업로드할 수 있는 파일 업로드 취약점 발견

[보안뉴스 민세아] 워드프레스의 플러그인 형태로 설치되는 슬라이더 레볼루션(RevSlider)에서 웹셸을 업로드할 수 있는 파일 업로드 취약점이 발견됐으며, 이로 인해 100,000개의 사이트가 해킹에 영향을 받은 것으로 드러났다.

https://www.boannews.com/media/view.asp?idx=44941

EBS 해킹, 홈페이지 게시판 취약점 이용한 'Webshell' 추정

입력 : 2012-05-18 15:17

하지만 전반적인 보안관리 소홀과 홈페이지 관리가 허술했던 부분에 대해서는 책임이 크다는 지적이다. 이와 관련 익명의 한 보안전문가는 "게시판과 같은 글을 쓰는 곳에 첨부된 악성파일이라면 웹서버 악성코드인 'Webshell(웹쉘)'이다. 이는 파일 업로드 취약점으로, 게시판 등에 웹쉘을 올리고 웹쉘에서 SQL로 조회를 통해 DB에서 정보를 탈취해 간 것으로 보인다"고 설명했다.

https://www.boannews.com/media/view.asp?idx=31349&kind=0

File upload attack

file upload vulnerability를 통해 실행될 수 있는 공격

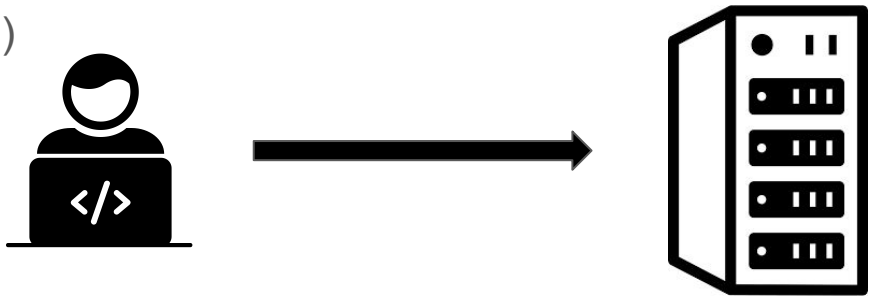
- RCE(Remote code execution) → 쉘을 획득하여 원격 명령어로 서버를 장악
- 랜섬웨어 공격 → 랜섬웨어 파일을 업로드하여 설치를 유도, 서버 파일 암호화
- 디도스 공격 → 한계 용량에 달하도록 파일을 업로드하여 서비스 장애 유발
- Deface 공격 → 이미지를 덮어씌워 웹사이트 등의 표면 페이지 이미지를 변조
- miner program 업로드 → 서버의 CPU, 용량을 사용해서 코인채굴 등 리소스 점유

File upload attack: RCE

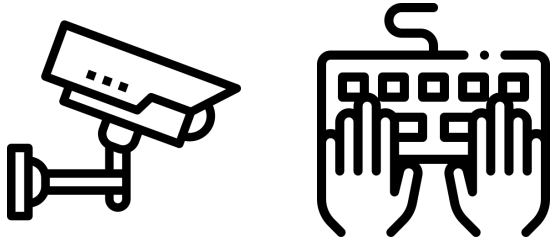
: PHP, Java, Python 등의 server side script를 업로드하여 서버에 원격 명령을 내리는 공격. 임의의 파일을 읽고 쓰거나 민감한 데이터에 접근, 기타 공격으로의 연결 등이 가능.

웹쉘(Web shell) : 웹(Web)과 쉘(shell)의 합성어. 올바른 HTTP request를 통해 원격 웹 서버에서 임의의 명령을 실행할 수 있도록 하는 악성 스크립트

1. 이미지 파일 등을 먼저 올려 파일이 업로드되는 경로 확인
2. 웹쉘 파일 업로드 (필요시 우회 기법 적용)
3. 확인한 경로를 통해 웹쉘에 접속
4. 웹쉘을 통해 서버 원격 제어



Keylogger



: 사용자가 입력을 수행하는 동안 키 입력을 추적하고 기록하는 malware또는 hardware. 해커가 피해자의 키 입력을 열람하기 위해 사용하는 가장 일반적인 방법.

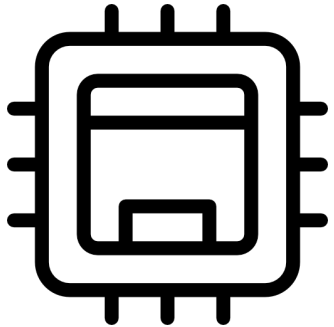
software keylogger: 컴퓨터에 설치하는 애플리케이션 형식으로 구성. 한번 설치되면 사용중인 OS의 키 입력을 모니터링하여 추적하고 기록할 수 있다. 기록한 정보를 해커에게 자동으로 전송시키는 등 스파이웨어의 한 종류로 발전시킬 수 있다.

hardware keylogger: 대상 컴퓨터에 물리적으로 연결될 수 있는 장치 형식으로 구성. 한번 구성된 후에는 software keylogger와 유사하게 동작한다.

Keylogger

keylogger는 사용자가 키보드에서 키를 누르는 시점과 키 입력의 정보가 모니터에 display되는 시점 사이에서 동작한다. 이를 위해 사용되는 방법은 다음과 같이 다양하다.

1. 키보드와 화면을 볼 수 있는 비디오 카메라 사용
2. 키보드-컴퓨터 간 상호작용을 촉진하는 드라이버를 입력 기록 드라이버로 교체
3. 키보드 스택 내부의 필터 드라이버 혹은 커널 기능을 사용
4. 키보드 자체 내부, 혹은 배선 내부에 하드웨어 버그를 삽입
5. 동적 연결 라이브러리(DDL)의 기능을 가로채기



Keylogger: How to attack

- Spear Phishing

  피싱 이메일, 링크 등을 전송하거나 유포하여 피해자의 클릭을 유도한다. 친척이나 친구, 거래처 등이 보낸 것 처럼 위장하거나 안전한 것으로 보이는 이메일, 문서 등이 열릴 때 자동으로 다운로드 되도록 처리한다.

- Trojan Horse

  다른 안전하거나 필수적인 프로그램의 번들에 키로거를 끼워넣어 함께 다운로드되고 실행되도록 유도한다.

- Drive-by Download

  악성 웹사이트 등에 숨겨두어 해당 사이트를 방문하면 코드가 설치되도록 한다. 설치된 코드는 감지되지 않은 채 백그라운드에서 실행되며 동작한다.

Ransomware

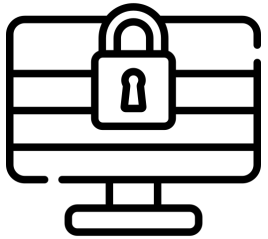
: 사용자나 조직이 컴퓨터의 파일에 액세스하는 것을 거부하도록 설계된 악성 코드.

주로 공격자가 파일을 암호화하고 해독 키에 대한 몸값을 요구함.

각기 고유한 특성을 가진 다양한 변종이 존재함.

오늘날 사회에서 기업/공공 서비스를 마비시키는 등 큰 피해를 입히고 있음.

Ransomware: How to attack



랜섬웨어 공격 단계

1. 공격 대상지에 랜섬웨어 프로그램을 다운로드하고 실행시키거나 실행을 유도한다.
2. 랜섬웨어 프로그램이 시스템에 엑세스하여 공격자가 시스템 파일을 암호화한다.

   일부 변종은 해독 키 없이 복구하기 더 어렵게 만들기 위해 파일의 백업 및 복사본을 삭제한다.

3. 피해자에게 돈을 요구하고 돈이 지불되면 복호화를 위한 키를 제공한다.

Scenarios

1. 서버 관리측에 피싱 메일 등을 전송하여 keylogger가 심어져 있는 파일을 설치하고 실행하도록 유도한다.
2. file upload 취약점을 활용하여 웹쉘과 랜섬웨어 파일을 업로드한다.
3. 웹쉘로 접근하여 keylogger 로그 기록을 열람하고 관리자 계정의 pw를 탈취한다.
4. ssh를 이용, 관리자 계정으로 접속하여 upload해둔 랜섬웨어 파일을 실행시킨다.

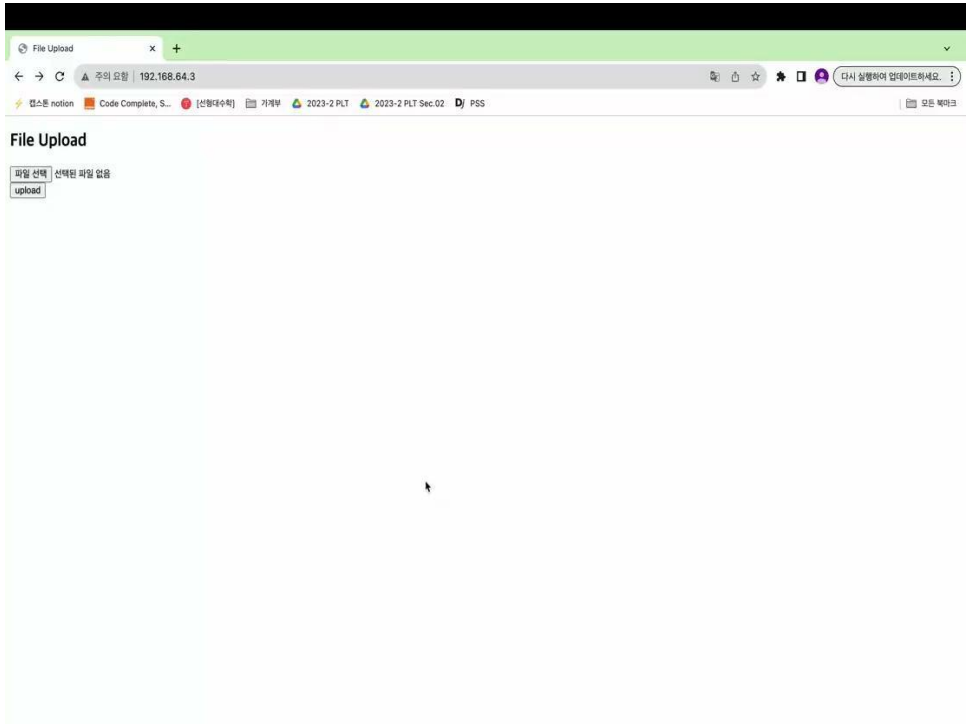
Demonstration

web server

apache 서버

파일 업로드에 대해 어떤 대비도 하지 않은 형태

Demonstration



Remediation

1. 확장자 검증 → 블랙리스트가 아닌 허용된 확장자 화이트리스트를 기준으로 확장자 확인하기

code:

```
$allowedExtensions = ['jpeg', 'gif', 'png', 'jpg'];
```

```
    $fileExtension = pathinfo($uploadedFileName,
        PATHINFO_EXTENSION);
```

```
    if(in_array($fileExtension, $allowedExtensions)){
```

Remediation

2. 파일 유효성 검증 → 업로드 파일의 실행 권한을 제한

code:

```
    chmod($targetFilePath, 0644);
```

Remediation

3. 저장되는 파일이 외부에서 식별되지 않도록 하기 → 파일명을 무작위로 지정하기

code:

```
        $randomFileName = uniqid() . '.' . $fileExtension;
        $targetFilePath = $uploadDirectory . $randomFileName;

    if (move_uploaded_file($uploadedFile, $targetFilePath)) {
```

Demonstration

Remediation

1. 다운로드된 파일에 대한 바이러스/무결성 검사 → 검사를 통과하지 못할 경우 디렉토리에서 제거.

2. 파일 업로드 전처리를 위한 프레임워크 사용하기 (TypeScript, JavaScript)

Thank you