2021.06.11로 희망합니다!!

시큐어코딩
학생 활동 보고서

강의명: 시큐어코딩_1

교수: 우사무엘

이름: 이건욱

학번: 32163006

제출일: 2021.06.08

목 차

1. 파일 업로드 공격 실제 사례

해킹 당한 웹 서버의 91%에서 웹쉘의 흔적 (KISA)

기업/기관	공격 건수	유출 경로
T쇼핑몰 외 225개 사이트	1,700만건	웹쉘을 통한 악성코드 삽입으로 홈페이지 해킹
OO 군인회	1만 3,900건	홈페이지 해킹 (SQL Injection)
B 치킨	51만건	홈페이지 해킹
C 교육	350만건	웹서버 해킹 추정
회계법인 외 대기업	927건	웹서버 해킹 추정 (랜섬웨어)
온라인 커뮤니티 'B'	195만건	홈페이지 해킹 (SQL Injection)
D 사	3만건	홈페이지 해킹
국내 기업 10개 웹사이트	10개	디페이스 공격 (화면 변조)

보안뉴스 | 전체기사 | SECURITY | IT | SAFETY | Security World

Home > 전체기사

웹서버 해킹 주범 '웹셸' 제거하기 대작전

좋아요 100개 | 입력 : 2015-06-08 15:40

#웹셸(WebShell) #웹셸 탐지 #웹보안 #유엠브이기술

해킹당한 웹서버 중 90% 이상 웹서버에서 웹셸 발견
웹 보안의 기본은 웹셸 탐지 및 방어...웹셸 방어 솔루션 살펴보니

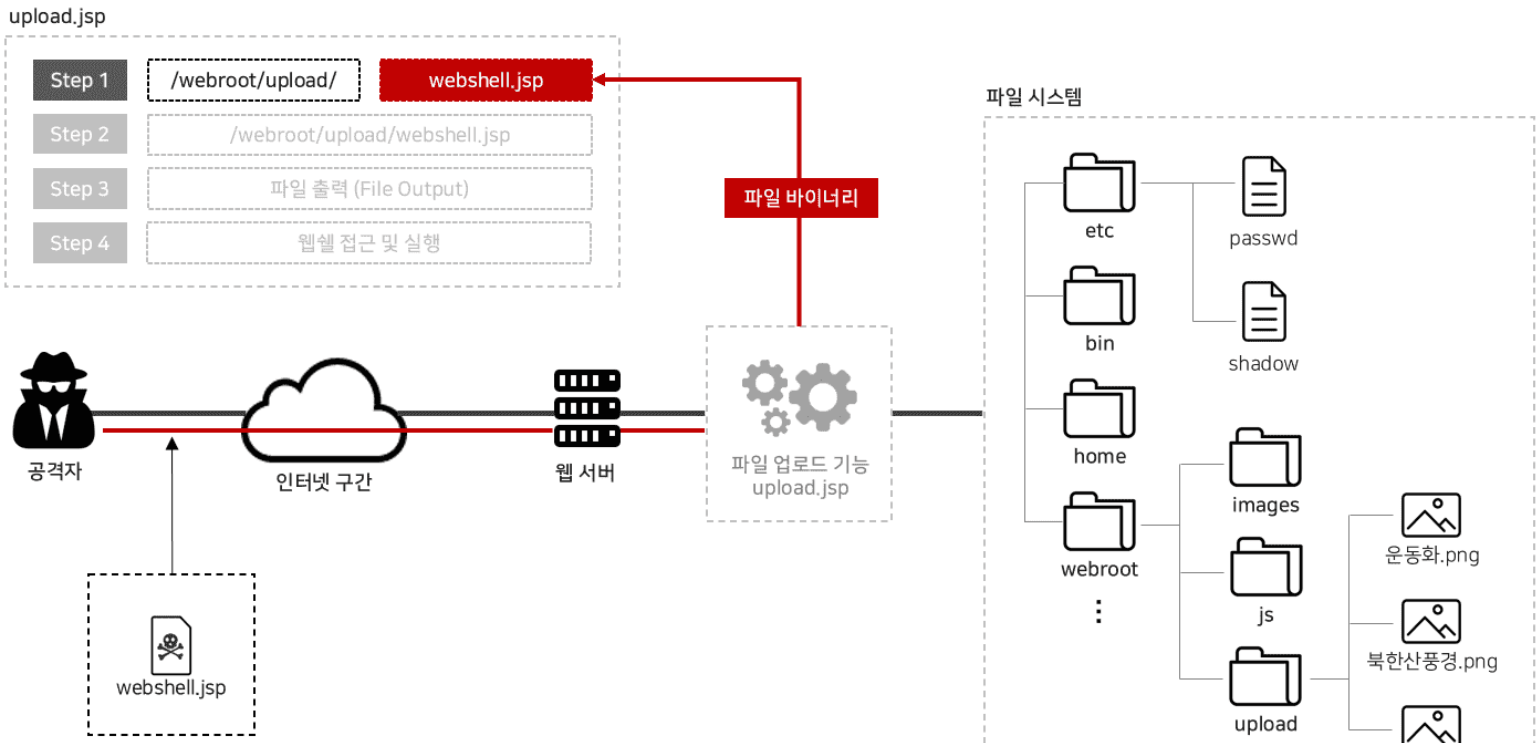
[보안뉴스 김태형] 수많은 웹 보안위협 중에서도 웹셸(WebShell)은 가장 기초적인 해킹 툴이다. 이는 공격자가 원격으로 웹서버를 제어할 수 있는 프로그램으로, 최근 이와 같은 웹셸 공격으로 인해 웹 서버에 저장된 개인정보가 유출되고, 웹사이트 변조·악성코드 유포지로 악용되는 사례가 증가하고 있다.

최근 이러한 웹셸에 의한 해킹 사고가 증가하고 있다. 지난해말 개인정보가 유출된 배달 앱 '배달통'의 해킹원인도 웹셸에 의한 것으로 밝혀졌다. 지난해 전 세계 개인정보 유출사고의 65%는 해킹에 의한 것으로 나타났다. 그리고 지난해 우리나라에서 발생한 3.20 사이버테러와 6.25 사이버테러도 해킹에 의한 침해사고였는데, 이들은 모두 웹셸 공격에서부터 시작됐다.

과거에도 이와 같은 웹셸 공격은 지속되어 왔다. 지난 2008년 옥션의 개인정보유출 사고와 2011년 현대캐피탈 정보유출, 2012년 EBS 정보유출 등도 웹셸에 의한 해킹 사건이었다.

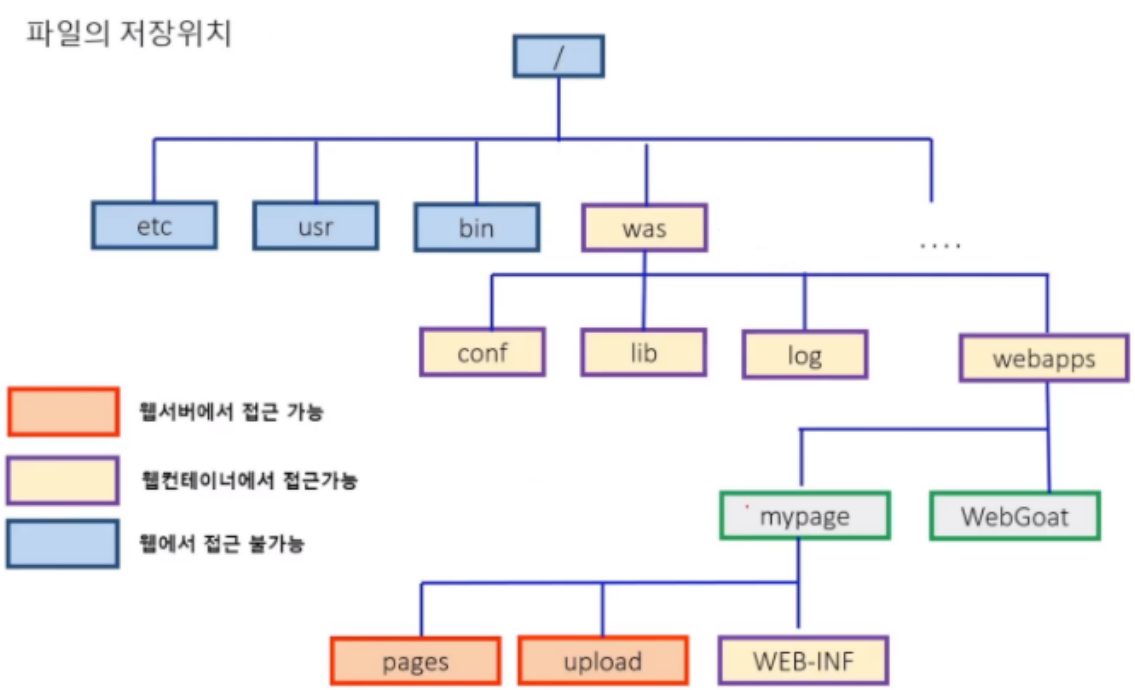
KISA 인터넷침해대응센터에 따르면, 해킹당한 웹서버 중 웹셸이 발견된 웹서버는 90% 이상이다. 지난해 2월 26일 의사협회 8만명, 치과의사협회 5만 6천명, 한의사협회 2만명 등 총 15만 6천명의 개인정보가 유출됐다. 이 3개 협회 홈페이지는 공격자가 악성코드를 웹사이트에 심어서 관리자 권한을 획득한 후, 웹셸을 이용해 개인정보를 탈취한 것으로 조사됐다. 지난해 3월 7일 113만명의 개인정보가 유출된 티켓몬스터의 경우에도 공격자가 홈페이지 게시판 등에 웹셸을 심어 해킹했다.

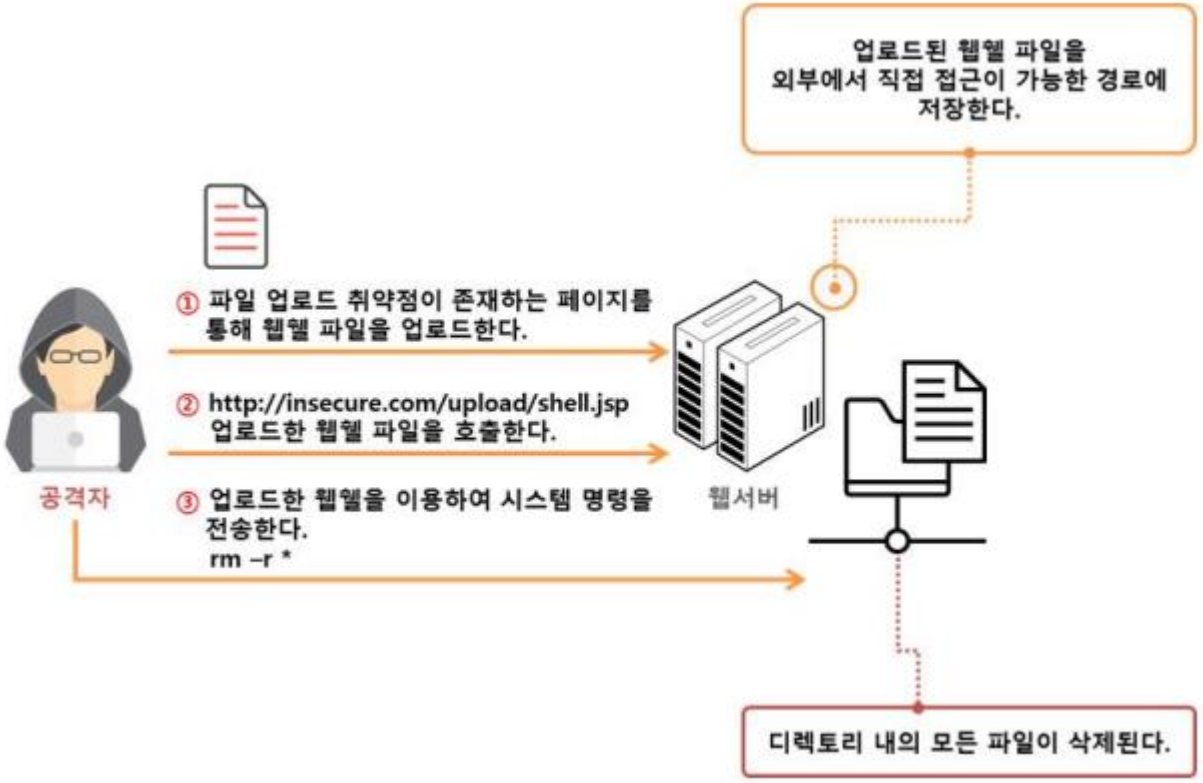
2. 파일 업로드의 공격의 유형



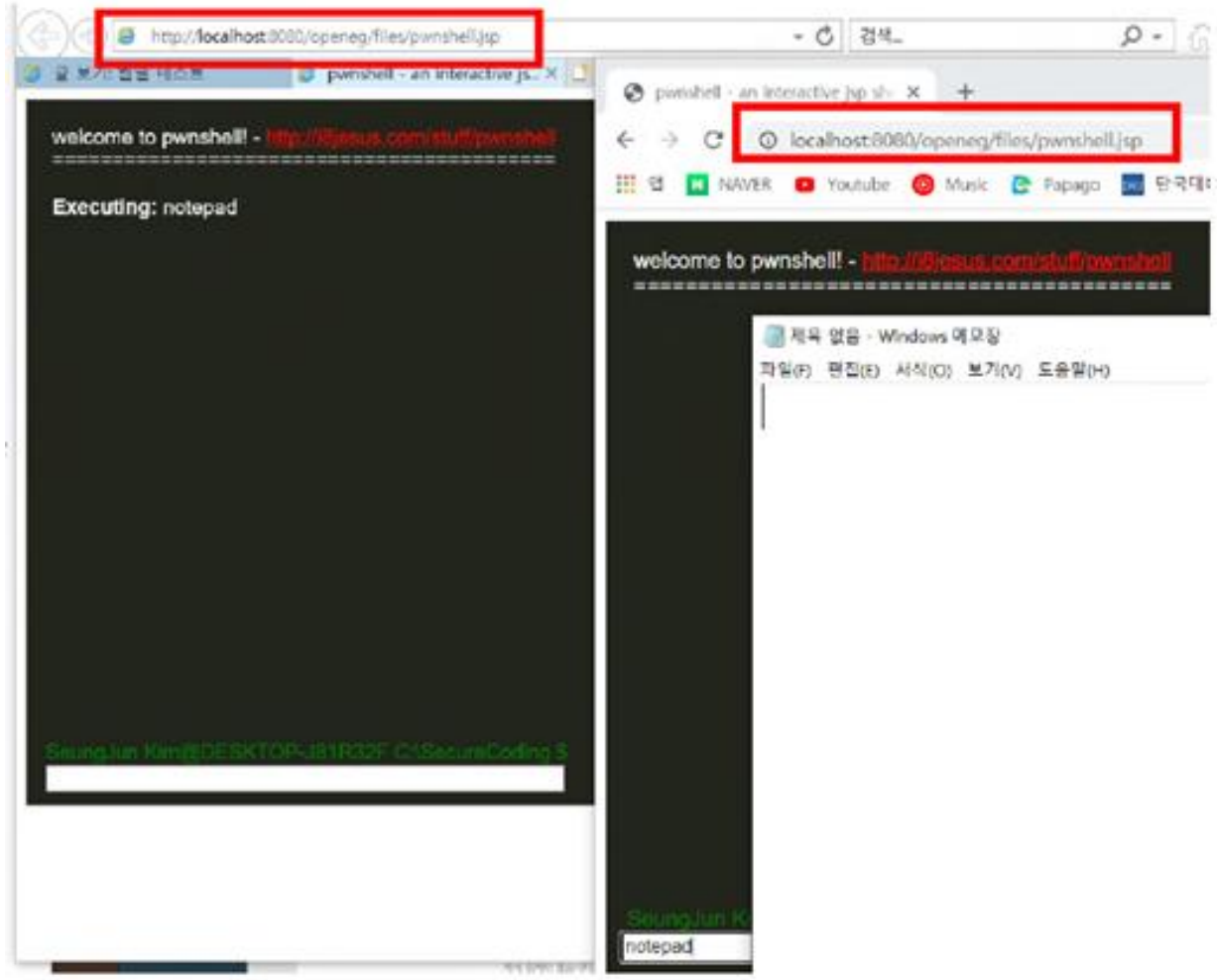
언어	확장자
asp, aspx	asp, aspx, htm, html, asa
php	phtml, php, php3, php4, php5, inc, htm, html
jsp, java	jsp, jspx, jsw, jsv, jspf, htm, html
perl	pl, pm, cgi, lib, htm, html
coldfusion	cfm, cfml, cfc, dbm, htm, html

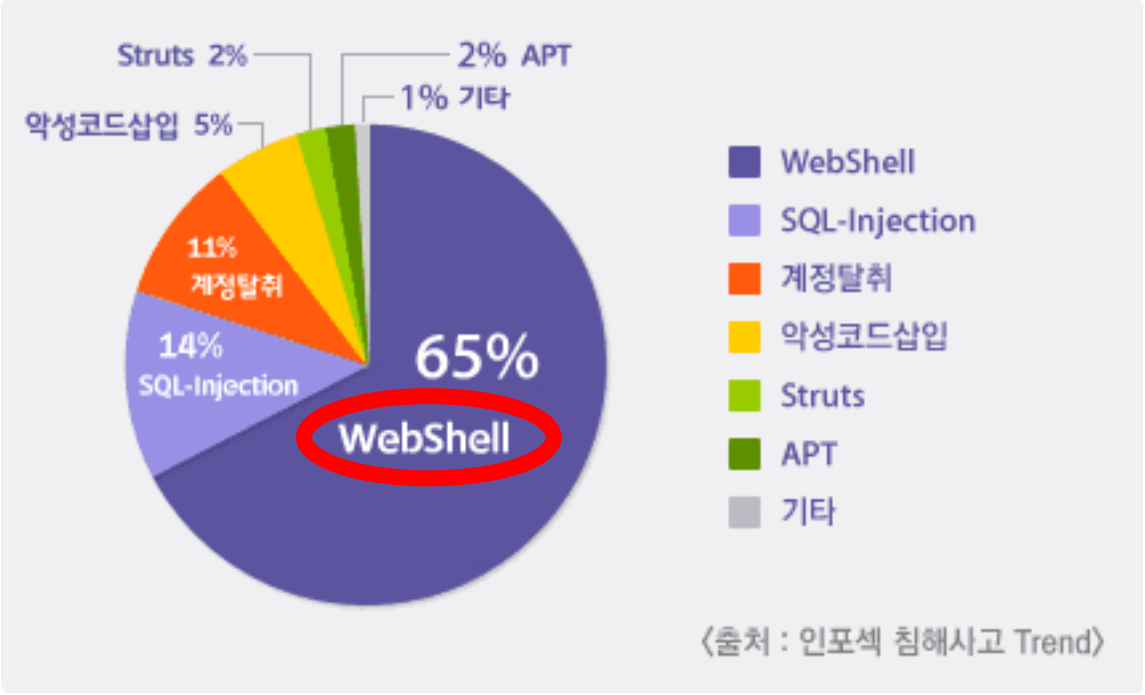
2. 파일 업로드의 공격의 유형_직접 접근이 가능한 경우



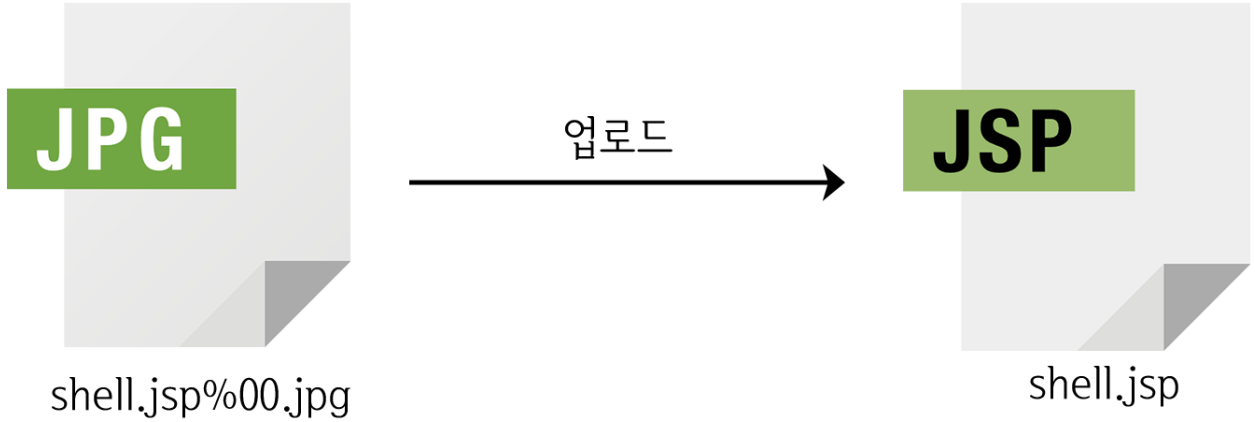


2. 파일 업로드의 공격의 유형_웹쉘 공격





2. 파일 업로드의 공격의 유형_NULL을 이용한 파일 업로드



[종단 문자 우회]

언어	우회패턴	처리패턴
php	test.php%00.jpeg	test.php
asp	test.asp%00.jpeg	test.asp
jsp	test.jsp%00.jpeg	test.jsp

3. 보안 정보 공유 체계_CVE

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
1999	894	177	112	172			2	7		25	16	103			2
2000	1020	257	208	206		2	4	20		48	19	139			
2001	1677	403	403	297		7	34	123		83	36	220		2	2
2002	2156	498	553	435	2	41	200	103		127	76	199	2	14	1
2003	1527	381	477	372	2	50	129	60	1	62	69	144		16	5
2004	2451	580	614	408	3	148	291	111	12	145	96	134	5	38	5
2005	4935	838	1627	657	21	604	786	202	15	289	261	221	11	100	14
2006	6610	893	2719	664	91	967	1302	322	8	267	272	184	18	849	30
2007	6520	1101	2601	955	95	706	883	338	14	267	326	242	69	700	45
2008	5632	894	2310	699	128	1101	807	362	7	288	268	188	83	170	76
2009	5736	1035	2185	698	188	963	851	323	9	337	302	223	115	138	738
2010	4653	1102	1714	676	342	520	605	276	8	234	284	238	86	73	1501
2011	4155	1221	1334	735	351	294	470	108	7	197	411	206	58	17	557
2012	5297	1425	1459	833	423	243	759	122	13	344	392	250	166	14	623
2013	5191	1455	1186	856	366	156	650	110	7	352	512	274	123	1	206
2014	7939	1599	1572	841	420	304	1103	204	12	457	2106	239	264	2	403
2015	6504	1793	1830	1084	749	221	784	151	12	577	753	366	248	5	129
2016	6454	2029	1496	1313	717	94	498	99	15	444	870	602	86	7	1
2017	14714	3155	3004	2495	745	508	1518	279	11	629	1659	459	327	18	6
2018	16557	1853	3041	2121	400	517	2048	545	11	708	1239	247	461	31	4
2019	17344	1342	3201	1270	488	549	2390	465	10	710	983	202	535	57	13
2020	18325	1351	3248	1618	409	460	2178	401	14	966	1345	310	402	37	62
2021	7986	800	1663	680	143	233	935	190	1	339	404	123	167	16	
Total	154277	26182	38557	20085	6083	8688	19227	4921	187	7895	12699	5513	3226	2305	4423
% Of All		17.0	25.0	13.0	3.9	5.6	12.5	3.2	0.1	5.1	8.2	3.6	2.1	1.5	

Security Vulnerabilities (File Inclusion)

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2021-33408			File Inclusion	2021-05-27	2021-05-27	0.0	None	???	???	???	???	???	???
Local File Inclusion vulnerability in Ab Initio Control>Center before 4.0.2.6 allows remote attackers to retrieve arbitrary files. Fixed in v4.0.2.6 and v4.0.3.1.														
2	CVE-2021-32100			File Inclusion	2021-05-07	2021-05-14	4.0	None	Remote	Low	???	Partial	None	None
A remote file inclusion vulnerability exists in Artica Pandora FMS 742, exploitable by the lowest privileged user.														
3	CVE-2021-31783	345		File Inclusion	2021-04-26	2021-05-04	5.0	None	Remote	Low	Not required	Partial	None	None
show_default.php in the LocalFilesEditor extension before 11.4.0.1 for Piwigo allows Local File Inclusion because the file parameter is not validated with a proper regular-expression check.														
4	CVE-2021-30173	36		File Inclusion	2021-05-07	2021-05-18	4.0	None	Remote	Low	???	Partial	None	None
Local File Inclusion vulnerability of the omni-directional communication system allows remote authenticated attacker inject absolute path into Url parameter and access arbitrary file.														
5	CVE-2021-27236	94		Exec Code File Inclusion	2021-02-16	2021-02-22	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
An issue was discovered in Mutare Voice (EVM) 3.x before 3.3.8. getfile.asp allows Unauthenticated Local File Inclusion, which can be leveraged to achieve Remote Code Execution.														
6	CVE-2021-24242	22		Dir. Trav. File Inclusion	2021-04-22	2021-04-30	5.5	None	Remote	Low	???	Partial	Partial	None
The Tutor LMS â€" eLearning and online course solution WordPress plugin before 1.8.8 is affected by a local file inclusion vulnerability through the maliciously constructed sub_page parameter of the plugin's Tools, allowing high privilege users to include any local php file														
7	CVE-2021-23340	22		Dir. Trav. File Inclusion	2021-02-18	2021-02-25	5.5	None	Remote	Low	???	Partial	Partial	None
This affects the package pimcore/pimcore before 6.8.8. A Local FIle Inclusion vulnerability exists in the downloadCsvAction function of the CustomReportController class (bundles/AdminBundle/Controller/Reports/CustomReportController.php). An authenticated user can reach this function with a GET request at the following endpoint: /admin/reports/custom-report/download-csv?exportFile=&91;filename]. Since exportFile variable is not sanitized, an attacker can exploit a local file inclusion vulnerability.														
8	CVE-2020-35942	352		Exec Code XSS Bypass CSRF File Inclusion	2021-02-09	2021-02-12	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
A Cross-Site Request Forgery (CSRF) issue in the NextGEN Gallery plugin before 3.5.0 for WordPress allows File Upload and Local File Inclusion via settings modification, leading to Remote Code Execution and XSS. (It is possible to bypass CSRF protection by simply not including a nonce parameter.)														
9	CVE-2020-35580	522		File Inclusion	2021-05-20	2021-05-28	5.0	None	Remote	Low	Not required	Partial	None	None
A local file inclusion vulnerability in the FileServlet in all SearchBlox before 9.2.2 allows remote, unauthenticated users to read arbitrary files from the operating system via a /searchblox/servlet/FileServlet?col=url= request. Additionally, this may be used to read the contents of the SearchBlox configuration file (e.g., searchblox/WEB-INF/config.xml), which contains both the Super Admin's API key and the base64 encoded SHA1 password hashes of other SearchBlox users.														
10	CVE-2020-35566	706		File Inclusion	2021-02-16	2021-02-19	5.0	None	Remote	Low	Not required	Partial	None	None
An issue was discovered in MB CONNECT LINE mymbCONNECT24 and mbCONNECT24 through 2.6.2. An attacker can read arbitrary JSON files via Local File Inclusion.														
11	CVE-2020-29279			Exec Code File Inclusion	2020-12-02	2020-12-04	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
PHP remote file inclusion in the assign_resume_tpl method in Application/Common/Controller/BaseController.class.php in 74CMS before 6.0.48 allows remote code execution.														

3. 보안 정보 공유 체계_CWE

Common Weakness Enumeration
A Community-Developed List of Software & Hardware Weakness Types

Home > CWE List > CWE- Individual Dictionary Definition (4.4)

Home | About | CWE List | Scoring | Community | News | Guidance | Search

CWE VIEW: Weaknesses in the 2020 CWE Top 25 Most Dangerous Software Weaknesses

View ID: 1350
Type: Graph

Objective

CWE entries in this view are listed in the 2020 CWE Top 25 Most Dangerous Software Weaknesses.

Audience

Stakeholder	Description
Software Developers	By following the CWE Top 25, developers are able to significantly reduce the number of weaknesses that occur in their software.
Product Customers	Customers can use the weaknesses in this view in order to formulate independent evidence of a claim by a product vendor to have elimiated / mitigated the most d
Educators	Educators can use this view to focus curriculum and teachings on the most dangerous weaknesses.

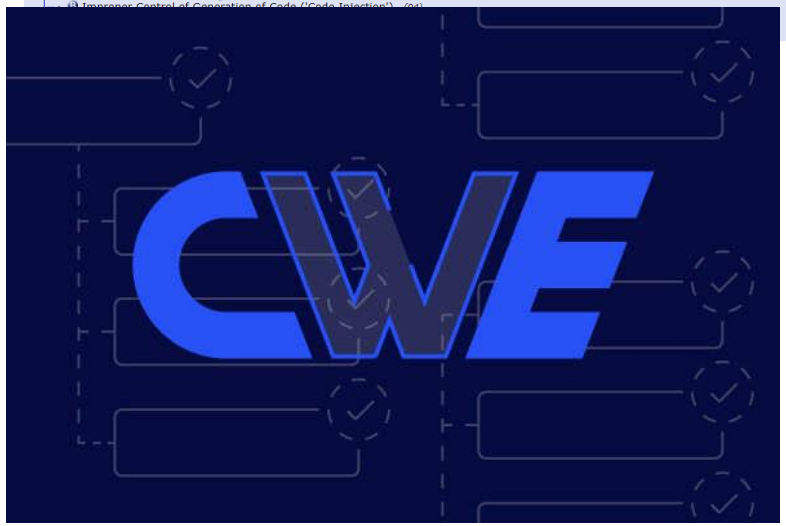
Relationships

The following graph shows the tree-like relationships between weaknesses that exist at different levels of abstraction. At the highest level, categories and pillars exist to group weaknesse special CWE entries used to group weaknesses that share a common characteristic. Pillars are weaknesses that are described in the most abstract fashion. Below these top-level entries ar still very abstract, typically independent of any specific language or technology. Base level weaknesses are used to present a more specific type of weakness. A variant is a weakness that a specific language or technology. A chain is a set of weaknesses that must be reachable consecutively in order to produce an exploitable vulnerability. While a composite is a set of weakn produce an exploitable vulnerability.

Expand All | Collapse All

1350 - Weaknesses in the 2020 CWE Top 25 Most Dangerous Software Weaknesses

- Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') - *(79)*
- Out-of-bounds Write - *(787)*
- Improper Input Validation - *(20)*
- Out-of-bounds Read - *(125)*
- Improper Restriction of Operations within the Bounds of a Memory Buffer - *(119)*
- Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') - *(89)*
- Exposure of Sensitive Information to an Unauthorized Actor - *(200)*
- Use After Free - *(416)*
- Cross-Site Request Forgery (CSRF) - *(352)*
- Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') - *(78)*
- Integer Overflow or Wraparound - *(190)*
- Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') - *(22)*
- NULL Pointer Dereference - *(476)*
- Improper Authentication - *(287)*
- Unrestricted Upload of File with Dangerous Type - *(434)*
- Incorrect Permission Assignment for Critical Resource - *(732)*
- Improper Control of Generation of Code ('Code Injection') - *(94)*



Demonstrative Examples

Example 1

The following code intends to allow a user to upload a picture to the web server. The HTML code that drives the form on the user end has an input field of type "file".

Example Language: **HTML** *(good code)*

```
<form action="upload_picture.php" method="post" enctype="multipart/form-data">

Choose a file to upload:
<input type="file" name="filename"/>
<br/>
<input type="submit" name="submit" value="Submit"/>

</form>
```

Once submitted, the form above sends the file to upload_picture.php on the web server. PHP stores the file in a temporary location until it is retrieved (or discarded) by the server side code. In this example, the file is moved to a more permanent pictures/ directory.

Example Language: **PHP** *(bad code)*

```
// Define the target location where the picture being

// uploaded is going to be saved.
$target = "pictures/" . basename($_FILES['uploadedfile']['name']);

// Move the uploaded file to the new location.
if(move_uploaded_file($_FILES['uploadedfile']['tmp_name'], $target))
{
  echo "The picture has been successfully uploaded.";
}
```

46 CVE-2019-19745 | 434 | File Inclusion | 2019-12-17 | 2019-12-18 | 6.5

Contao 4.0 through 4.8.5 allows PHP local file inclusion. A back end user with access to the form generator can uplo

The problem with the above code is that there is no check regarding type of file being uploaded. Assuming that pictures/ is available in the web document root, an attacker could upload a file with the name:

(attack code)

aded file, the

Common Weakness Enumeration
A Community-Developed List of Software & Hardware Weakness Types

CWE TOP 25 Most Dangerous Software Weaknesses

Home > CWE List > CWE- Individual Dictionary Definition (4.4) ID Lookup: Go

Home | About | CWE List | Scoring | Community | News | Guidance | Search

CWE-434: Unrestricted Upload of File with Dangerous Type

Weakness ID: 434
Abstraction: Base
Structure: Simple

Status: Draft

Presentation Filter: Complete

3. 보안 정보 공유 체계_NVD

Information Technology Laboratory

NATIONAL VULNERABILITY DATABASE

NVD

VULNERABILITIES

CVE-2021-3395 Detail

Current Description

A cross-site scripting (XSS) vulnerability in Pryaniki 6.44.3 allows remote authenticated users to upload an arbitrary file. The JavaScript code will execute when someone visits the attachment.

+View Analysis Description

Severity CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD **Base Score:** 5.4 MEDIUM **Vector:** CVSS:3.1/AV:N/AC:L/PR:L/UI:

NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS informati
CVE List from the CNA.

Note: NVD Analysts have published a CVSS score for this CVE based on publicly available information at the time of analysis. The
a score within the CVE List.

QUICK INFO

CVE Dictionary Entry:
CVE-2021-3395
NVD Published Date:
02/02/2021
NVD Last Modified:
02/04/2021
Source:
MITRE

NVD

CVE

CVE List | CNAs | WGs | Board | About | News & Blog

NVD Go to for: CVSS Scores CPE Info

Search CVE List | Downloads | Data Feeds | Update a CVE Record | Request CVE IDs

TOTAL CVE Records: 154789

HOME > ABOUT CVE > CVE AND NVD RELATIONSHIP

CVE and NVD Relationship

CVE and NVD Are Two Separate Programs

The CVE List was launched by MITRE as a community effort in 1999, and the U.S. National Vulnerability Database (NVD) was launched by the National Institute of Standards and Technology (NIST) in 2005.

- **CVE** - A list of records—each containing an identification number, a description, and at least one public reference—for publicly known cybersecurity vulnerabilities. CVE Records are used in numerous cybersecurity products and services from around the world, including NVD.
- **NVD** - A vulnerability database built upon and fully synchronized with the CVE List so that any updates to CVE appear immediately in NVD.
- **Relationship** – The CVE List *feeds* NVD, which then builds upon the information included in CVE Records to provide enhanced information for each record such as fix information, severity scores, and impact ratings. As part of its enhanced information, NVD also provides advanced searching features such as by OS; by vendor name, product name, and/or version number; and by vulnerability type, severity, related exploit range, and impact.

While separate, both CVE and NVD are sponsored by the U.S. Department of Homeland Security (DHS) Cybersecurity and Infrastructure Security Agency (CISA), and both are available to the public and free to use.

4. 파일 업로드 공격 유형 별 방어 대책

```
@RequestMapping(value="/write.do", method=RequestMethod.POST)
public String boardWriteProc(@ModelAttribute("BoardModel") BoardModel boardModel, MultipartHttpServletRequest request, HttpSes
    //파일저장 위치
    String uploadPath = session.getServletContext().getRealPath("/")
                        +"WEB-INF/files/";
    System.out.println("uploadPath: "+uploadPath);
    MultipartFile file = request.getFile("file");
    //업로드 되는 파일 사이즈 제한
    if ( file != null && ! "".equals(file.getOriginalFilename())
        && file.getSize() < 1024000 && file.getContentType().contains("image")){
        //업로드 파일명
        String fileName = file.getOriginalFilename();
        if (  fileName.toLowerCase().endsWith(".jpg")){
            //저장할 파일명을 랜덤하게 생성하여 사용한다
            String savedFileName = UUID.randomUUID().toString();
            File uploadFile = new File(uploadPath+ savedFileName);
            try {
                file.transferTo(uploadFile);  //실제 파일이 저장되는 라인
            } catch (Exception e) {
                System.out.println("upload error");
            }
            boardModel.setFileName(fileName);
            boardModel.setSavedFileName(savedFileName);
        }
    }

    String content = boardModel.getContent().replaceAll("\r\n", "<br/>");
    boardModel.setContent(content);

    service.writeArticle(boardModel);

    return "redirect:list.do";
}
```

1

```
//added code
@RequestMapping("/get_image.do")
public void getImage(HttpServletRequest request, HttpSession session,
                     HttpServletResponse response){
    int idx=TestUtil.getInt(request.getParameter("idx"));
    // 읽어본 게시물인지 확인
    if ( session.getAttribute("idx") ==  null  ||
            (Integer)session.getAttribute("idx") != idx  ) {
        return;
    }
    // 저장된 파일명을 읽어오는 작업이 필요
    BoardModel board = service.getOneArticle(idx);

    String filePath=session.getServletContext().getRealPath("/")+
                    "WEB-INF/files/"+board.getSavedFileName();

    System.out.println("filename: "+filePath);
    BufferedOutputStream out=null;
    InputStream in=null;
    try {
        response.setContentType("image/jpeg");
        response.setHeader("Content-Disposition", "inline;filename="+board.getFileName());
        File file=new File(filePath);
        in=new FileInputStream(file);
        out=new BufferedOutputStream(response.getOutputStream());
        int len;
        byte[] buf=new byte[1024];
        while ( (len=in.read(buf)) > 0) {
            out.write(buf,0,len);
        }
    }catch(Exception e){
        e.printStackTrace();
        System.out.println("파일 전송 에러");
    } finally {
        if ( out != null ) try { out.close(); }catch(Exception e){}
        if ( in != null ) try { in.close(); }catch(Exception e){}
    }
}
```

2

```
<td colspan="4" align="left">
<span class="date">첨부파일: 
<a href="get_image.do?idx=${board.idx}"
target="_blank">${board.fileName}</a></span></td>
```

발표를 마치겠습니다!

감사합니다!