

# Week 26\_ 구글해킹 예시

---

Refactoring Study

2022.4.30 Kwon Moonjeong

# 익명성을 위한 캐시 사용

---

- 캐시 링크를 클릭하면 웹 브라우저는 구글 데이터베이스에 저장된 페이지뿐만 아니라 실제 서버에 저장된 그래픽이나 **HTML**이 아닌 콘텐츠까지 가져옴
- 캐시 **URL**의 맨 뒤에 **&strip=1** 을 붙이면 캐시 페이지의 **HTML** 부분만을 볼 수 있음.  
-> 이렇게 캐시 페이지에 접근하면 실제 웹 서버에 연결하지 않기 때문에 익명성을 지킬 수 있음

# 디렉터리 목록 활용하기

- 디렉터리 목록에서 특정 디렉터를 찾으려면 **index.of** 요청문에 디렉터리 이름을 추가한다.
  - ex) **intitle:index.of inurl:backup**  
URL에 **backup**이라는 단어를 포함하고 있는 디렉터리 목록 찾기

Google search results for the query `intitle:index.of inurl:backup`. The search shows about 47,900 results in 0.29 seconds. Two results are displayed:

1. <http://elearning.olade.org/backup/>  
Index of /backup  
Name · Last modified · Size · Description. [PARENTDIR], Parent Directory, -, [], backup.class.php, 2017-12-21 20:42, 5.8K.

2. <https://lms.ugr.ac.id/backup/>  
Index of /backup - LMS UGR

Name	Last modified	Size
Parent Directory		-
backup.class.php	2021-09-03 07:51	6.4K
backup.php	2021-09-03 07:51	9.6K

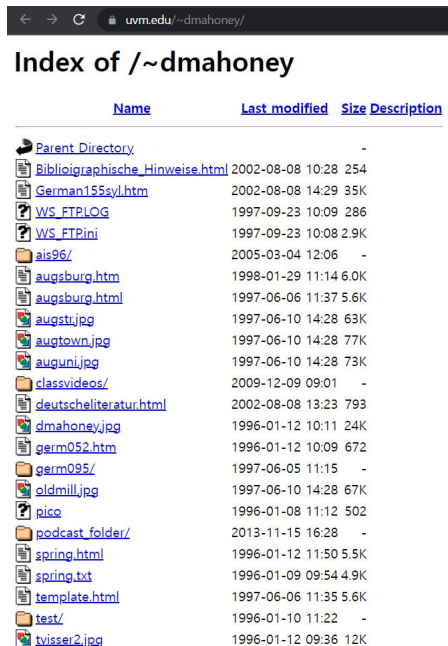
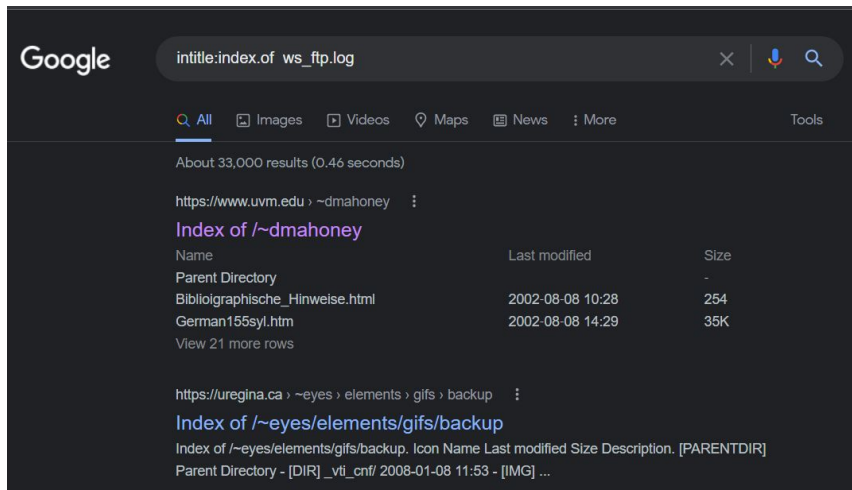
View 17 more rows

Index of /backup

Name	Last modified	Size
<a href="#">Parent Directory</a>		-
<a href="#">backup.class.php</a>	2021-09-03 07:51	6.4K
<a href="#">backup.php</a>	2021-09-03 07:51	9.6K
<a href="#">backupfilesedit.php</a>	2021-09-03 07:51	3.0K
<a href="#">backupfilesedit_form...&gt;</a>	2021-09-03 07:51	2.1K
<a href="#">cc/</a>	2022-02-16 21:08	-
<a href="#">controller/</a>	2022-02-16 21:08	-
<a href="#">converter/</a>	2022-02-16 21:08	-
<a href="#">copy.php</a>	2021-09-03 07:51	3.5K
<a href="#">copyprogress.php</a>	2021-09-03 07:51	2.2K
<a href="#">externallib.php</a>	2021-09-03 07:51	14K

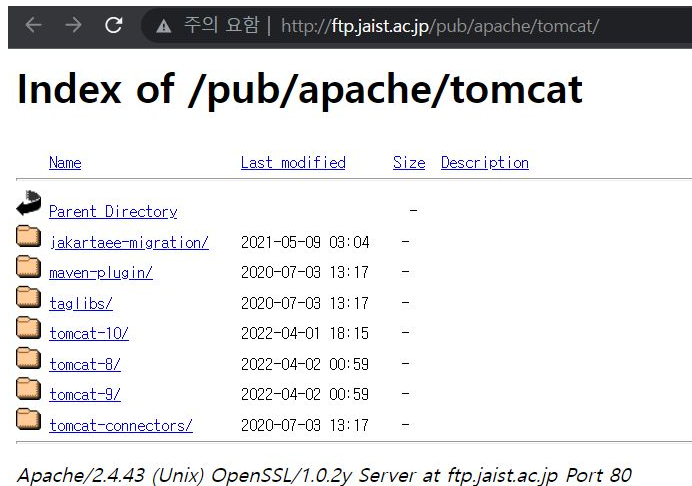
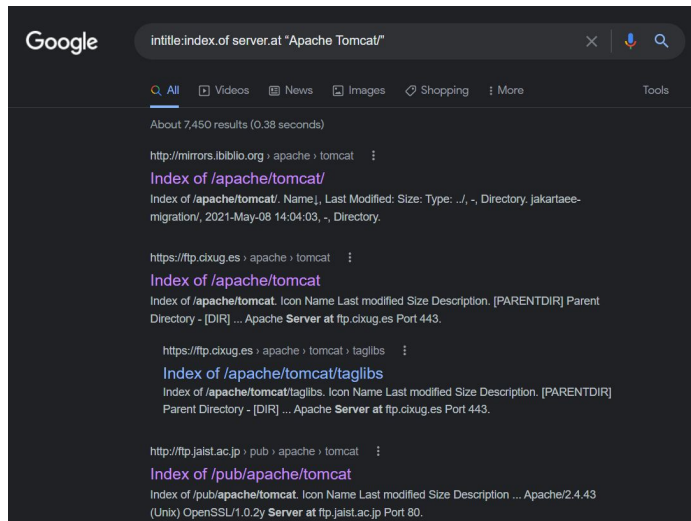
# 디렉터리 목록 활용하기

- 디렉터리 목록에서 특정 파일을 찾으려면 `index.of` 요청문 뒤에 파일 이름을 붙인다.
  - ex) `intitle:index.of ws_ftp.log`



# 디렉터리 목록으로 서버 버전 확인하기

- 일부 서버(구체적으로 Apache와 Apache 변종)는 디렉터리 목록의 밑 부분에 서버 태그를 포함시킴. 이 서버 태그는 index.of 요청문에 `server at` 구를 추가시킨 검색문으로 찾을 수 있음
  - ex) `intitle:index.of server.at "Apache Tomcat"` 으로 다양한 버전의 Apache Tomcat 서버를 운영하는 웹 서버를 찾을 수 있음



# 디렉터리 목록 탐색

- 디렉터리 목록을 활용하여 다른 디렉터리나 하위 디렉터를 찾을 수 있음
  - **parent directory** 클릭시 현재 디렉터리의 상위 디렉터리로 이동
  - 그 상위 디렉터리가 다른 디렉터를 포함하고 있다면, 링크를 타고 다른 디렉터리로 이동 가능
  - 상위 디렉터리로 이동했을 때 디렉터리 목록이 나오지 않는다면, 디렉터리 이름을 추측해 상위 디렉터리 URL의 가장 끝부분에 이름을 덧붙여야 함

← → ↻ 주의 요일 | http://ftp.jaist.ac.jp/pub/apache

## Apache Software Foundation Distribution Directory

The directories linked below contain current software releases from the Apache Software Foundation projects. Older non-recommended releases can be found on our [archive site](#).

To find the right download for a particular project, you should start at the project's own webpage or on our [project resource listing](#) rather than browsing the links below.

Please do not download from [apache.org!](#) If you are currently at [apache.org](#) and would like to browse, please visit [a nearby mirror site](#) instead.

### Projects

Name	Last modified	Size	Description
Parent Directory	-	-	-
<a href="#">accounts/</a>	2022-02-14 09:36	-	-
<a href="#">activeio/</a>	2022-04-29 16:57	-	-
<a href="#">aircavata/</a>	2020-07-07 00:16	-	-
<a href="#">airflow/</a>	2022-04-04 16:07	-	-
<a href="#">allura/</a>	2021-05-18 04:57	-	-
<a href="#">ankaci/</a>	2022-02-01 06:30	-	-
<a href="#">ant/</a>	2021-10-19 15:42	-	-
<a href="#">apc2/</a>	2022-09-04 07:48	-	-
<a href="#">asoc/</a>	2020-07-08 13:26	-	-
<a href="#">asisix/</a>	2022-04-15 22:43	-	-
<a href="#">asr/</a>	2022-04-30 01:25	-	-
<a href="#">archive/</a>	2022-04-30 01:25	-	-
<a href="#">aries/</a>	2022-04-14 11:19	-	-
<a href="#">arroz/</a>	2022-04-19 22:02	-	-
<a href="#">asterixdb/</a>	2022-09-18 12:06	-	-

← → ↻ 주의 요일 | http://ftp.jaist.ac.jp/pub/

## Index of /pub

Name	Last modified	Size	Description
Parent Directory	-	-	-
<a href="#">CPAN/</a>	2022-04-30 07:06	-	-
<a href="#">CTAN/</a>	2020-03-31 22:01	-	-
<a href="#">DragonFly/</a>	2014-04-30 11:48	-	-
<a href="#">FreeBSD-PC98/</a>	2013-03-21 14:19	-	-
<a href="#">FreeBSD-jp/</a>	2013-03-21 14:38	-	-
<a href="#">FreeBSD/</a>	2022-04-30 00:28	-	-
<a href="#">GNU/</a>	2022-01-22 12:09	-	-
<a href="#">Linux/</a>	2022-02-19 09:27	-	-
<a href="#">NetBSD/</a>	2021-09-08 00:10	-	-
<a href="#">OpenBSD/</a>	2022-04-30 03:35	-	-
<a href="#">RFC/</a>	2022-04-29 18:30	-	-
<a href="#">apache/</a>	2022-04-30 01:25	-	-
<a href="#">click-nikkei/</a>	2013-11-14 09:34	-	-
<a href="#">cygwin/</a>	2021-09-14 03:38	-	-
<a href="#">eclipse/</a>	2022-04-27 22:39	-	-
<a href="#">emergency/</a>	2011-03-17 14:42	-	-

← → ↻ 주의 요일 | http://ftp.jaist.ac.jp

Welcome to JAIST Public Mirror Service

FTP.JAIST.AC.JP

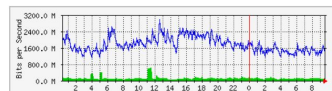
👍 144명이 좋아합니다. 친구들 같이 즐겨보세요.



Our server consists of DELL PowerEdge R7415 Server with AMD EPYC 7513P, 256 GiB memory and 64 TB storage with 3.6 TB SSD cache, and 2x 40 GbE interface. It is certified as an official mirror server by many projects. If you have any problem, please contact [ftp-admin@ml.jaist.ac.jp](mailto:ftp-admin@ml.jaist.ac.jp).

There are blogs in [English](#) and [Japanese](#) with tips for this server.

The server handles 457 HTTP, 4 FTP, and 6 rsync connections with the use of 1506.3 Mbps bandwidth and 4.9% CPU time on 30 Apr. 2022 at 9:35 in JST.



# 디렉터리 목록 탐색 - 증분 치환 활용

- 증분 치환은 어떤 숫자를 그보다 크거나 작은 숫자로 치환하는 기법
  - 디렉터리나 파일 이름에 숫자를 사용하는 사이트를 탐색할 때 활용
  - 나머지 부분은 그대로 둔 채, 숫자를 1 증가시키거나 감소 시킴

← → ↻ 주의 요함 | <http://ftp.jaist.ac.jp/pub/apache/tomcat/tomcat-9/>

## Index of /pub/apache/tomcat/tomcat-9

Name	Last modified	Size	Description
 <a href="#">Parent Directory</a>		-	
 <a href="#">v9.0.62/</a>	2022-04-01 18:25	-	

Apache/2.4.43 (Unix) OpenSSL/1.0.2y Server at ftp.jaist.ac.jp Port 80

← → ↻ 주의 요함 | <http://ftp.jaist.ac.jp/pub/apache/tomcat/tomcat-10/>

## Index of /pub/apache/tomcat/tomcat-10

Name	Last modified	Size	Description
 <a href="#">Parent Directory</a>		-	
 <a href="#">v10.0.20/</a>	2022-04-01 17:33	-	
 <a href="#">v10.1.0-M14/</a>	2022-04-01 17:29	-	

Apache/2.4.43 (Unix) OpenSSL/1.0.2y Server at ftp.jaist.ac.jp Port 80

# 디렉터리 목록 탐색 - 확장자 탐색

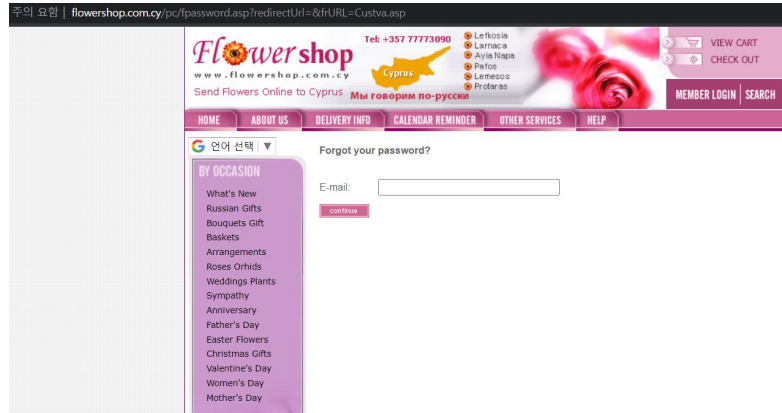
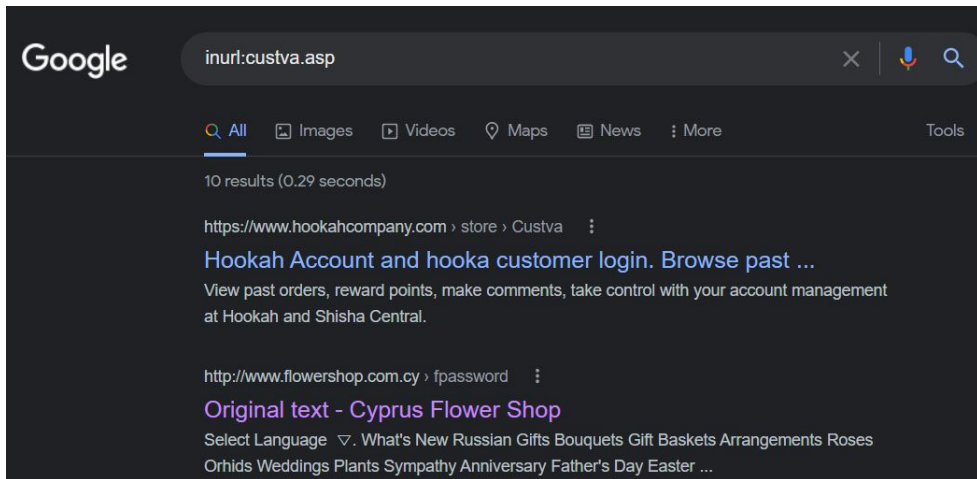
---

- 이름은 같지만 확장자가 다른 파일을 찾을 수 있음 (ex. 백업 파일)
  - 디렉터리 목록, 특히 캐시에 저장된 목록으로 사이트에 백업 파일이 존재하는지, 사이트의 다른 부분에 어떤 종류의 파일 확장자가 쓰이는지 알 수 있음
  - URL에 포함된 확장자를 다른 확장자로 치환하기 (ex. html -> bak)



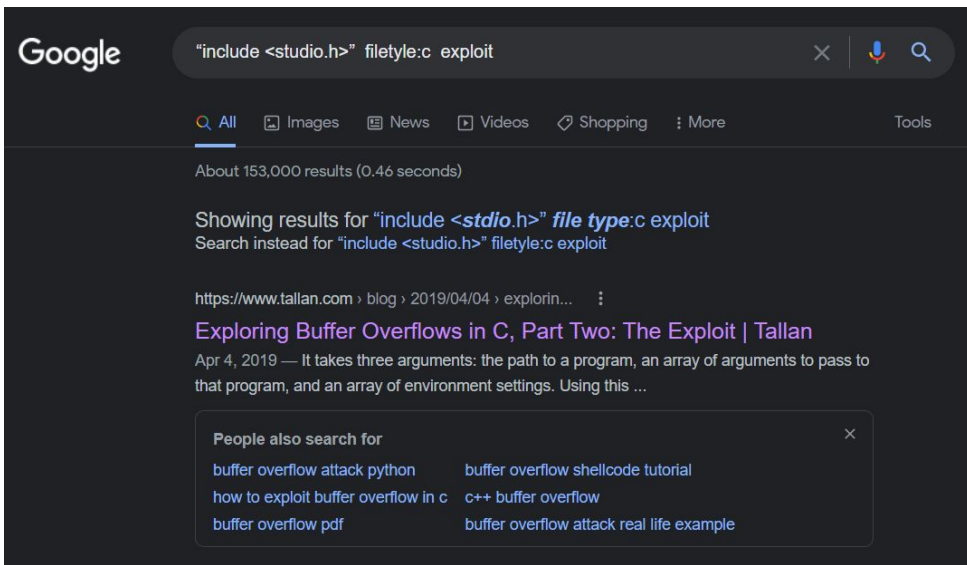
# 취약한 공격 대상 찾기

- “Powered by”와 같은 특징적인 문자열로 취약한 공격 대상을 찾을 수 있음
  - ex) inurl:custva.asp
  - EarlyImpact Productcart v1.5는 여러 취약점이 있음



# 구글로 공격 코드 찾기

- 구글로 **exploit**이나 **vulnerability** 등을 검색하면 공개 공격코드 사이트를 찾을 수 있음
- 공격코드 검색범위를 좁히기 위해 **filetype** 연산자, 소스코드에 흔하게 포함된 문자열을 검색하면 특정 프로그래밍 언어로만 작성된 공격 코드를 찾을 수 있음
  - ex) "include <stdio.h>" filetype:c exploit



Google search results for the query "include <stdio.h> filetype:c exploit". The search returned approximately 153,000 results in 0.46 seconds. The top result is from Tallan's blog, titled "Exploring Buffer Overflows in C, Part Two: The Exploit | Tallan", dated April 4, 2019. The snippet describes a program that takes three arguments: the path to a program, an array of arguments to pass to that program, and an array of environment settings. A "People also search for" section is visible at the bottom, listing related search terms like "buffer overflow attack python", "buffer overflow shellcode tutorial", "how to exploit buffer overflow in c", "c++ buffer overflow", "buffer overflow pdf", and "buffer overflow attack real life example".

Google

"include <stdio.h>" filetype:c exploit

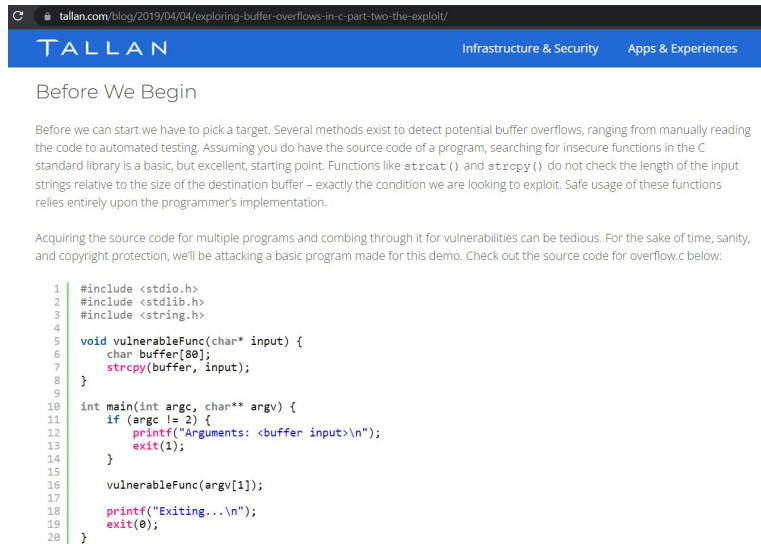
About 153,000 results (0.46 seconds)

Showing results for "include <stdio.h>" **file type**:c exploit  
Search instead for "include <stdio.h>" filetype:c exploit

https://www.tallan.com › blog › 2019/04/04 › explorin...  
**Exploring Buffer Overflows in C, Part Two: The Exploit | Tallan**  
Apr 4, 2019 — It takes three arguments: the path to a program, an array of arguments to pass to that program, and an array of environment settings. Using this ...

People also search for

- buffer overflow attack python
- buffer overflow shellcode tutorial
- how to exploit buffer overflow in c
- c++ buffer overflow
- buffer overflow pdf
- buffer overflow attack real life example



A screenshot of a web browser displaying a blog post from Tallan's website. The page title is "Exploring Buffer Overflows in C, Part Two: The Exploit". The content begins with a section titled "Before We Begin", which discusses the importance of selecting a target and mentions methods for detecting buffer overflows. It notes that functions like `strcpy()` and `strcat()` do not check the length of the input strings relative to the destination buffer. Below this, there is a section titled "Acquiring the source code for multiple programs and combing through it for vulnerabilities can be tedious. For the sake of time, sanity, and copyright protection, we'll be attacking a basic program made for this demo. Check out the source code for overflow.c below:". This is followed by a C code snippet that defines a vulnerable function `vulnerableFunc` which uses `strcpy` to copy input data into a fixed-size buffer, and a `main` function that calls this vulnerable function with command-line arguments.

tallan.com/blog/2019/04/04/exploring-buffer-overflows-in-c-part-two-the-exploit/

TALLAN Infrastructure & Security Apps & Experiences

## Before We Begin

Before we can start we have to pick a target. Several methods exist to detect potential buffer overflows, ranging from manually reading the code to automated testing. Assuming you do have the source code of a program, searching for insecure functions in the C standard library is a basic, but excellent, starting point. Functions like `strcpy()` and `strcat()` do not check the length of the input strings relative to the size of the destination buffer - exactly the condition we are looking to exploit. Safe usage of these functions relies entirely upon the programmer's implementation.

Acquiring the source code for multiple programs and combing through it for vulnerabilities can be tedious. For the sake of time, sanity, and copyright protection, we'll be attacking a basic program made for this demo. Check out the source code for `overflow.c` below:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 void vulnerableFunc(char* input) {
6     char buffer[80];
7     strcpy(buffer, input);
8 }
9
10 int main(int argc, char** argv) {
11     if (argc != 2) {
12         printf("Arguments: <buffer input>\n");
13         exit(1);
14     }
15
16     vulnerableFunc(argv[1]);
17
18     printf("Exiting...\n");
19     exit(0);
20 }
```

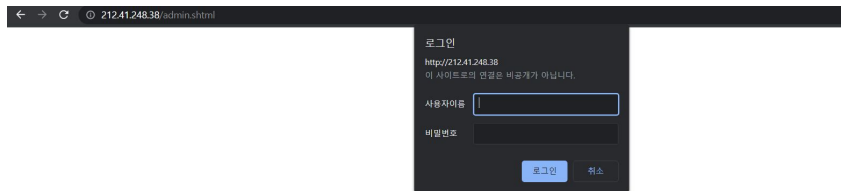
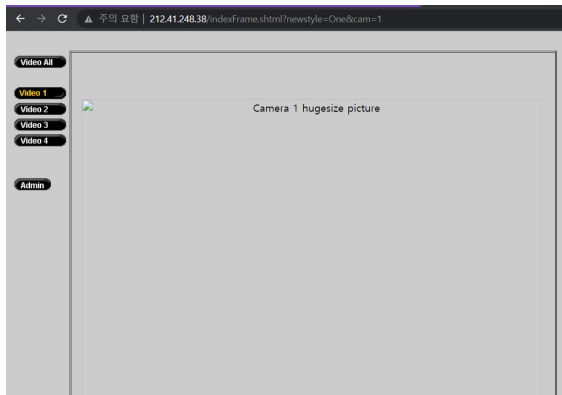
# 웹 서버 검색과 프로파일링 - 기본 애플리케이션 찾기

---

- 웹 소프트웨어는 문서와 매뉴얼 뿐만 아니라 기본 애플리케이션을 포함하고 있는 경우가 많음
- 기본 웹 페이지처럼 소프트웨어의 기능을 사용자에게 보여주는 예제 역할을 하며, 개발자가 참고할 수 있는 예제 루틴과 코드를 제공하기도 함
- 취약점, 취약한 기능을 포함하기도 함.
  - ex) Microsoft Index Server의 기본 검색 페이지는 다른 페이지에 링크되지 않은 페이지나 민감한 정보를 담고 있는 페이지를 찾음.
- 로그인 포털은 웹사이트의 정문 역할
- 기본 로그인 포털 페이지가 노출되어 있다면, 그 서버의 보안 수준이 낮으리라 짐작 가능
  - ex) Outlook Web Access 기본 포털 : 프로그램 소프트웨어 개정 버전을 노출하여 소프트웨어 버전의 취약점 검색 가능, 로그인하지 않은 사용자도 볼 수 있는 공개 접근 영역을 제공.

# 웹 서버 검색과 프로파일링 - 네트워크 하드웨어 찾기

- 네트워크 장비가 인터넷에 연결되어 있고, 그 웹페이지 링크가 구글 데이터베이스에 저장되어 있다면 구글 해킹으로 찾을 수 있음
- 네트워크 장비에 대한 정보를 노출하는 페이지는 네트워크를 분석할 때 사용할 수 있음
- 구글로 검색 가능한 다수의 장비가 기본 설정을 그대로 사용하고 있어, 사용자 이름이나 비밀번호를 몰라도 장비를 제어할 수 있음
- ex) Axis Video Server (CAM) : inurl:indexFrame.shtml Axis



# 민감한 정보를 찾기 위한 유용한 검색문

---

- username | userid | employee.ID | "your username is"
  - 사용자 이름 수집을 위해 가장 많이 쓰이는 요청문
  - 사용자 이름을 찾을 수 없어도, 추후 공격할 때 이용 가능한 정보가 있을 수 있음
- 범용 사용자 목록 파일 찾기
  - inurl:admin inurl:userlist
  - inurl:admin inurl:userlist filetype:asp
- password | passcode | "your password is"
  - **site** 연산자와 함께 사용할 시, 해당 홈페이지에서 비밀번호를 잃어버린 사람을 위한 페이지를 찾을 수 있음
  - 비밀번호 생성 정책 페이지를 찾을 수도 있음. -> 비밀번호 **brute force** 공격에 사용 가능

# 민감한 정보를 찾기 위한 유용한 검색문

---

- 범용 비밀번호 목록 찾기
  - index.of passlist
  - inurl:passlist.txt
  - "password.dat" filetype:dat
  - inurl:password.log filetype:log
- administrator | admin
  - login을 함께 검색하면 관리자 로그인 페이지를 찾을 수 있음
  - inurl 연산자를 사용하여 URL 부분에서 검색하면, 관리자 기능과 관련된 페이지를 찾을 수 있음
- inurl:temp | inurl:tmp | inurl:backup | inurl : bak
  - site 연산자와 결합해서 사용하면 서버의 임시/백업 파일 혹은 디렉토리를 찾을 수 있음
  - inurl 확장자를 사용하기 때문에 저 단어들이 확장자인 파일도 찾음 (ex. index.html.bak)