OWASP

Open Web Application Security Project

OWASP TOP 10 이란?

2004년 부터 현재까지 3~4년에 한번씩

주로 웹에 관한 취약점 중에서 빈도가 많이 발생하고,

보안상 위협을 크게 줄 수 있는 것들을 10가지 선정하여

Top 10 으로 발표하고있는 보안프로젝트

OWASP TOP 10

A1 - Injection	A6 – Security Misconfigurations
A2 – Broken Authentication	A7 – Cross Site Scripting (XSS)
A3 – Sensitive Data Exposure	A8 – Insecure Deserialization
A4 – XML External Entities (XXE)	A9 – Using Components with Known Vulnerabilities
A5 – Broken Access Control	A10 – Insufficient Logging and Monitoring

OWASP TOP 10

A1 - Injection	A6 – Security Misconfigurations
A2 – Broken Authentication	A7 – Cross Site Scripting (XSS)
A3 – Sensitive Data Exposure	A8 – Insecure Deserialization
A4 – XML External Entities (XXE)	A9 – Using Components with Known Vulnerabilities
A5 – Broken Access Control	A10 – Insufficient Logging and Monitoring

Broken Authentication

- “admin/admin”과 같은 약한 암호, 잘 알려진 암호를 허용
- 다중 인증이 없음
- 세션 ID가 URL에 노출
- 세션 ID를 무효화 시키지 않음
- ...

Broken Authentication

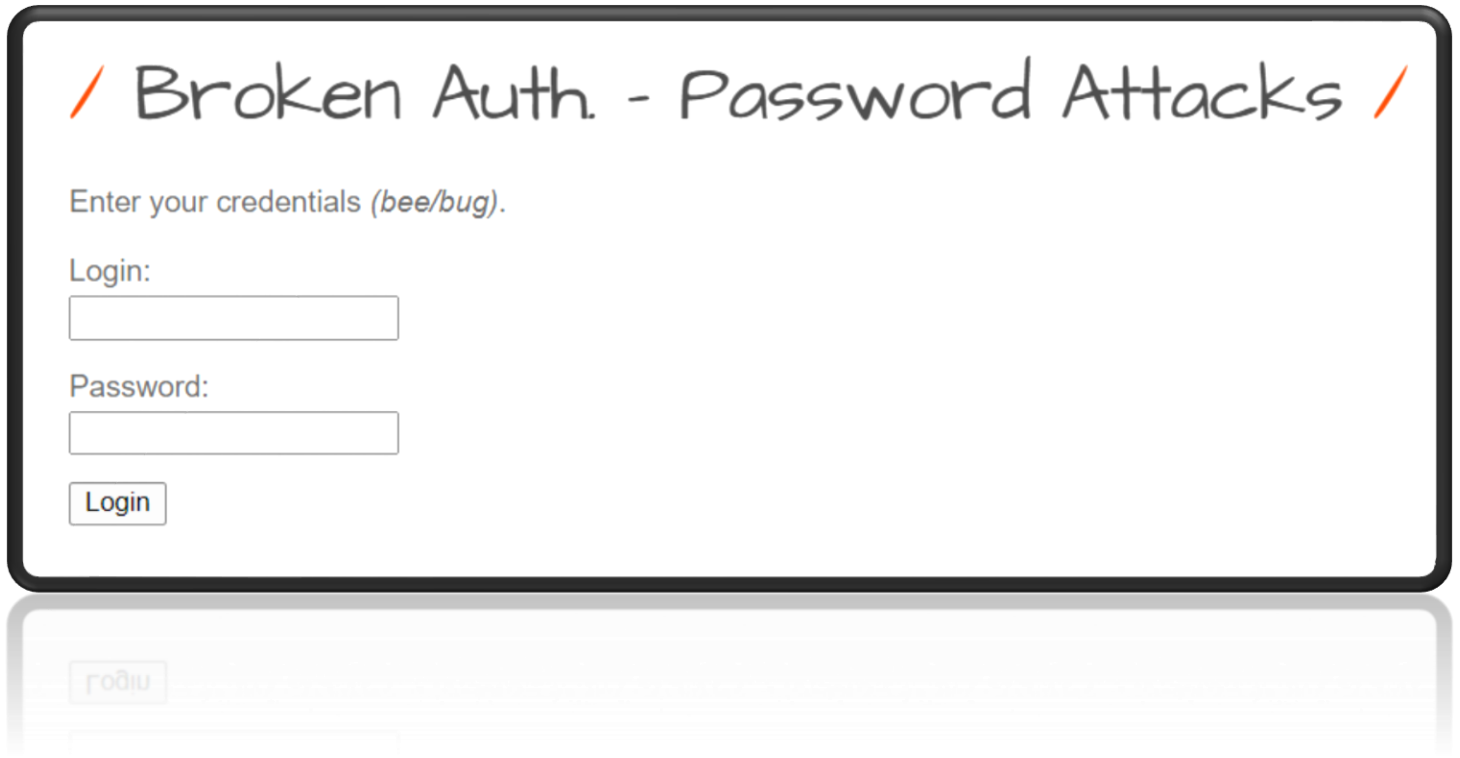
시나리오

#1. 잘 알려진 암호 목록을 이용하거나,
무차별 대입 공격(브루트 포스)을 통해 계정을 알아낼 수 있다.

#2. 세션에 대한 적절한 만료 시간을 정해 놓지 않는다면,
'로그아웃'을 선택하지 않고 단순히 브라우저 탭을 닫고 나갈 경우,
다음 사용자가 이용할 때 여전히 인증되어 있어 이전사용자의 권한을 가질 수 있다.

…

Broken Authentication – Password Attacks



Broken Authentication – Password Attacks

#

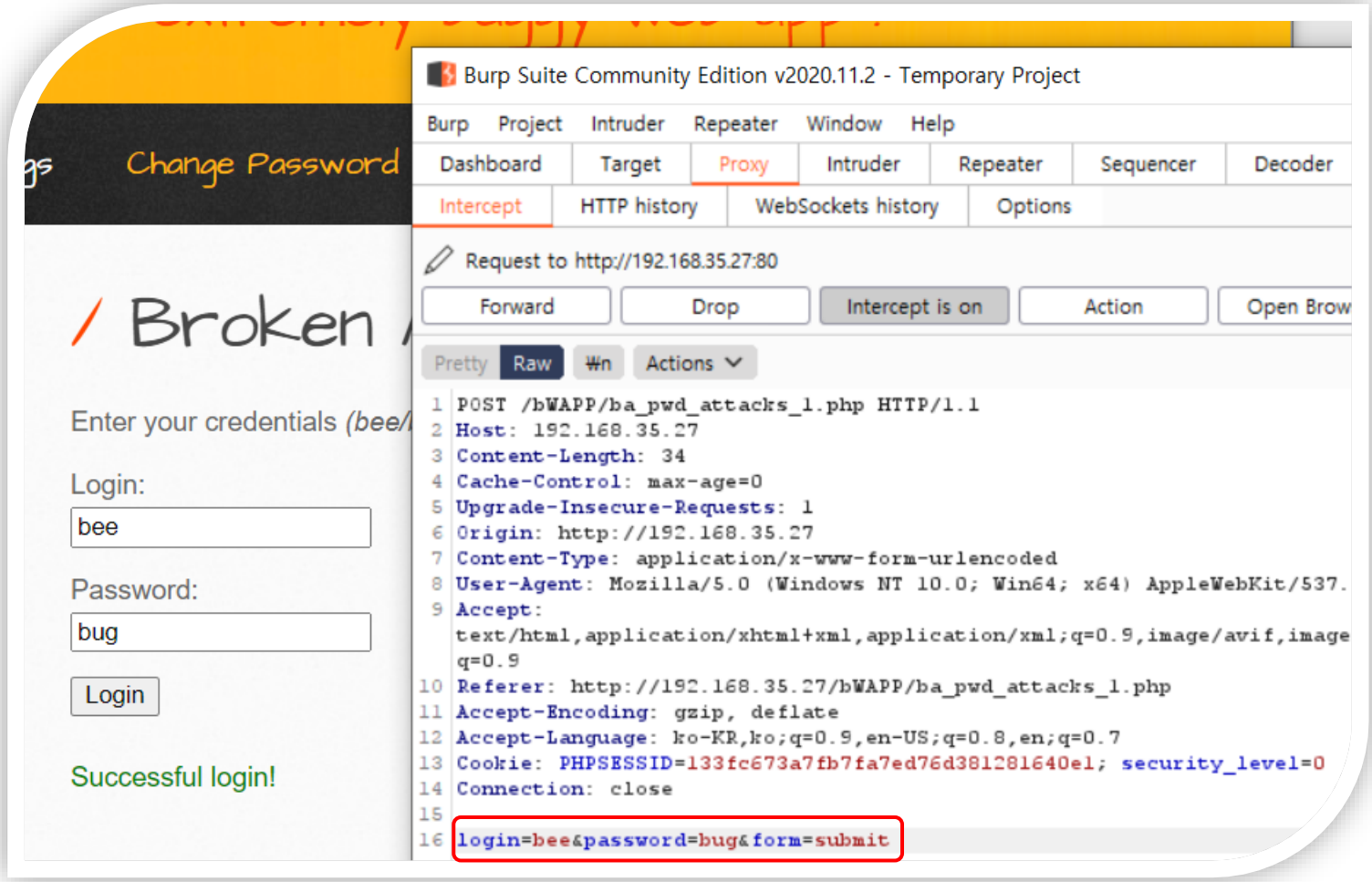
친구와 pc방에 같이 있는 상황

친구의 롤 아이디는 알고 있으며,

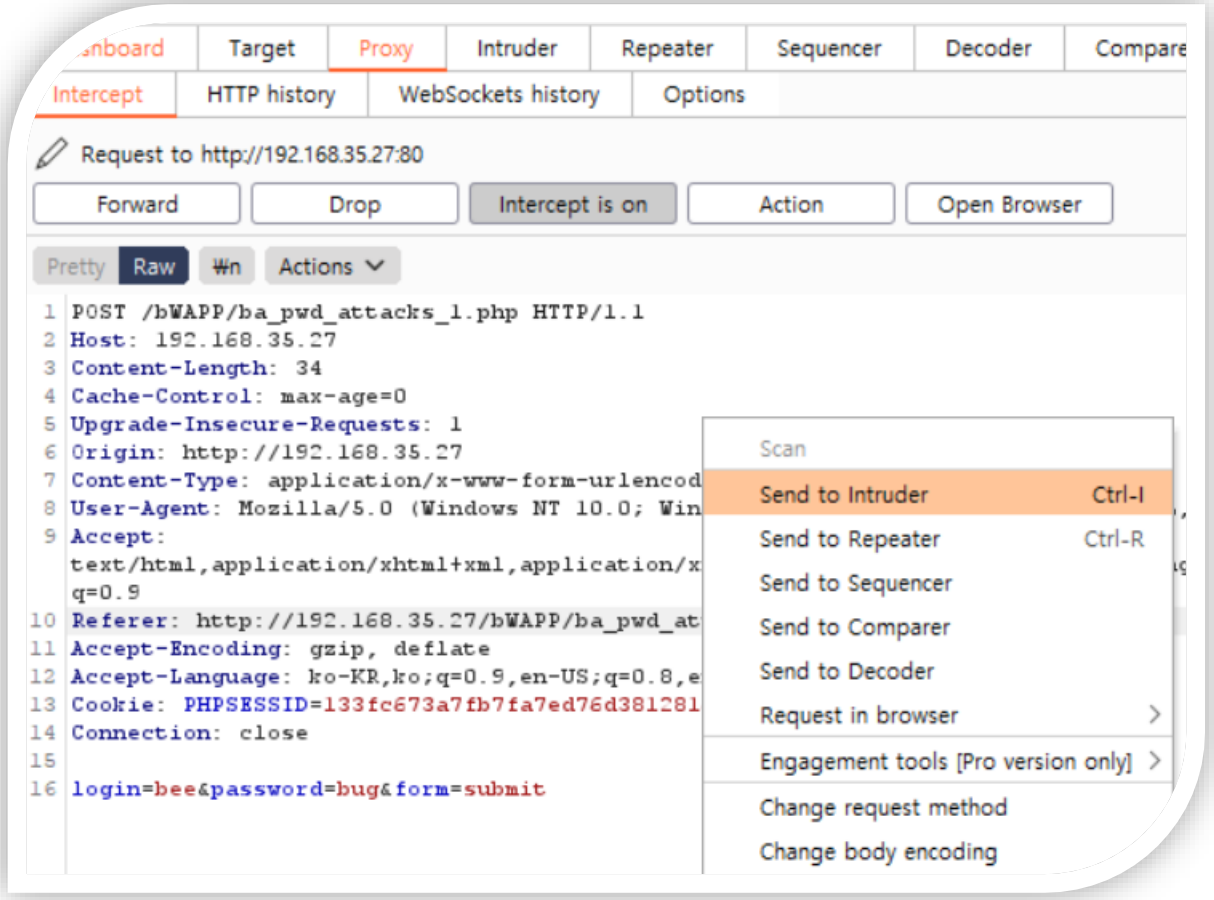
비밀번호를 입력할 때 슬쩍 봤더니

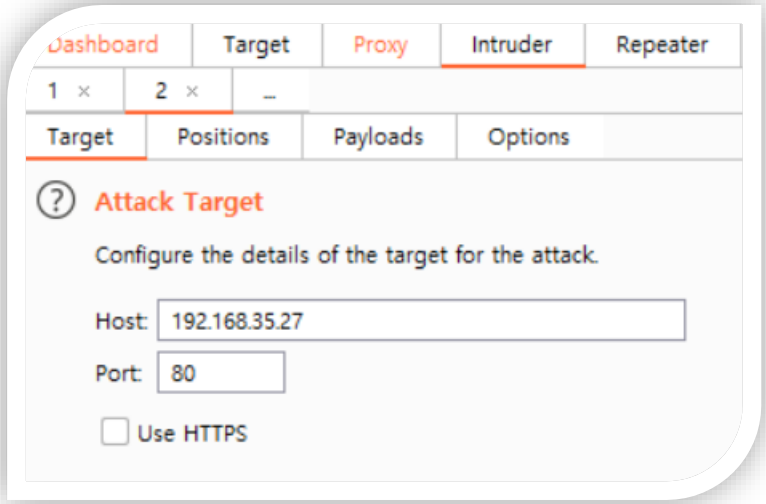
3글자이며, 영문으로 구성!!

Broken Authentication – Password Attacks



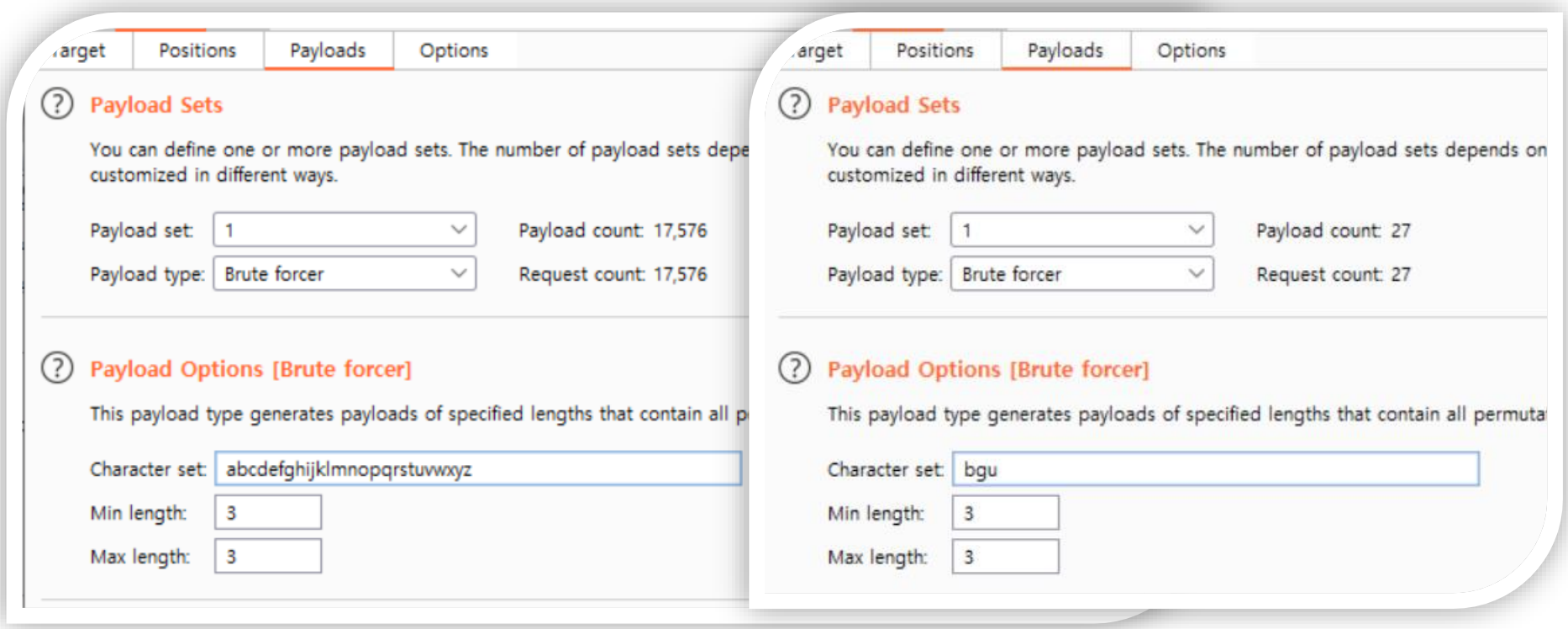
Broken Authentication – Password Attacks





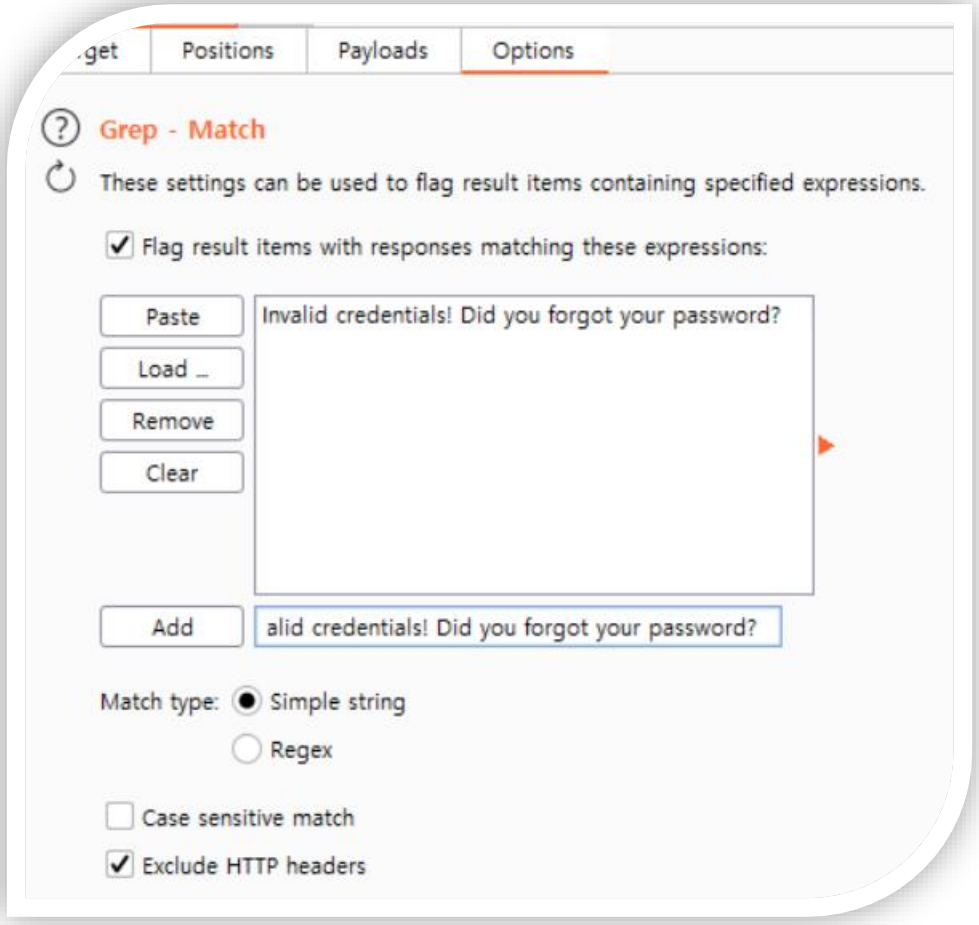
Broken Authentication – Password Attacks

Broken Authentication – Password Attacks

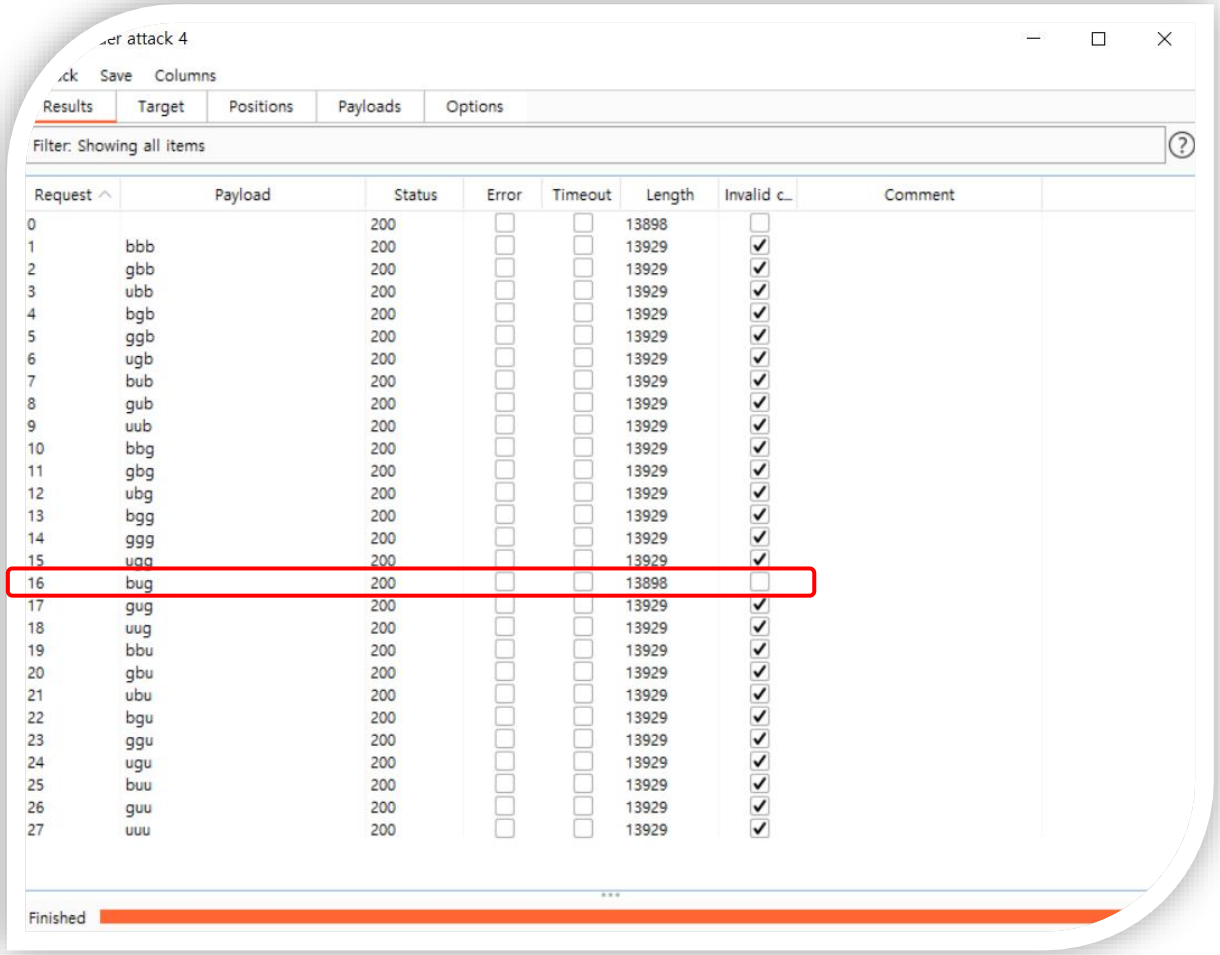


Broken Authentication – CAPTCHA Bypassing

Broken Authentication – Password Attacks



Broken Authentication – Password Attacks



How to prevent
Password Attacks

- 약한 비밀번호 검사

- 다중 인증 구현

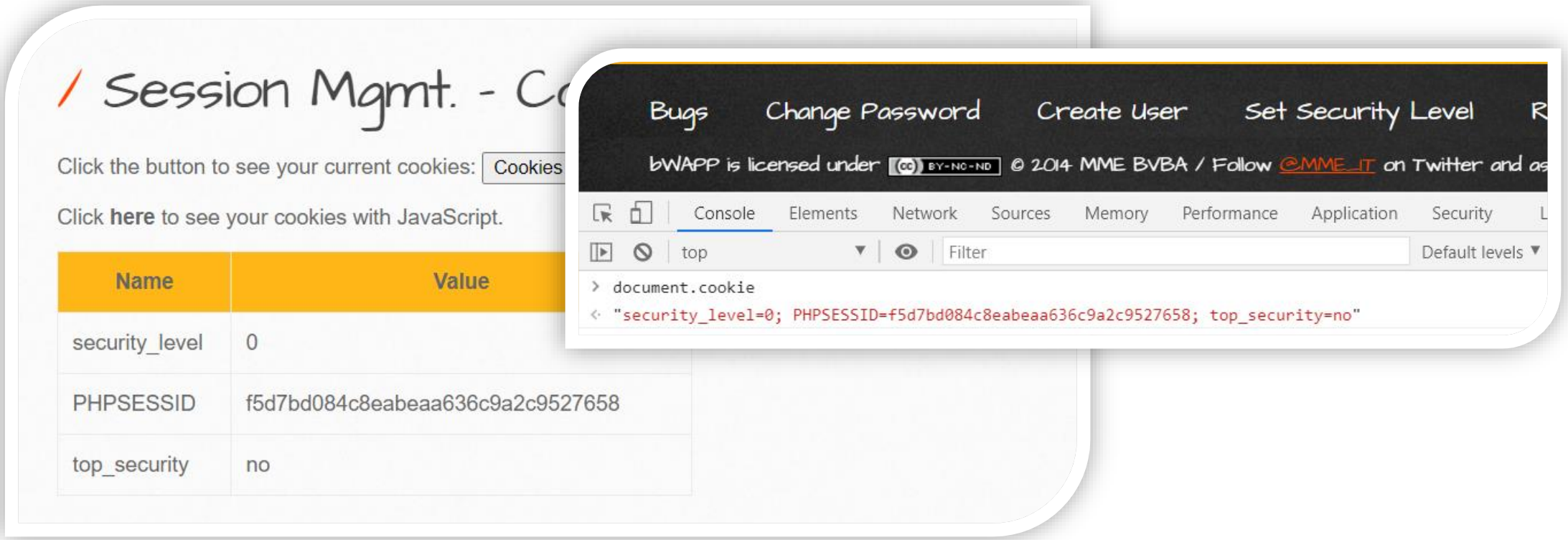
- 로그인 실패에 대한 로그를 남기고,

무차별 공격 등이 탐지됐을 때, 관리자에게 알림

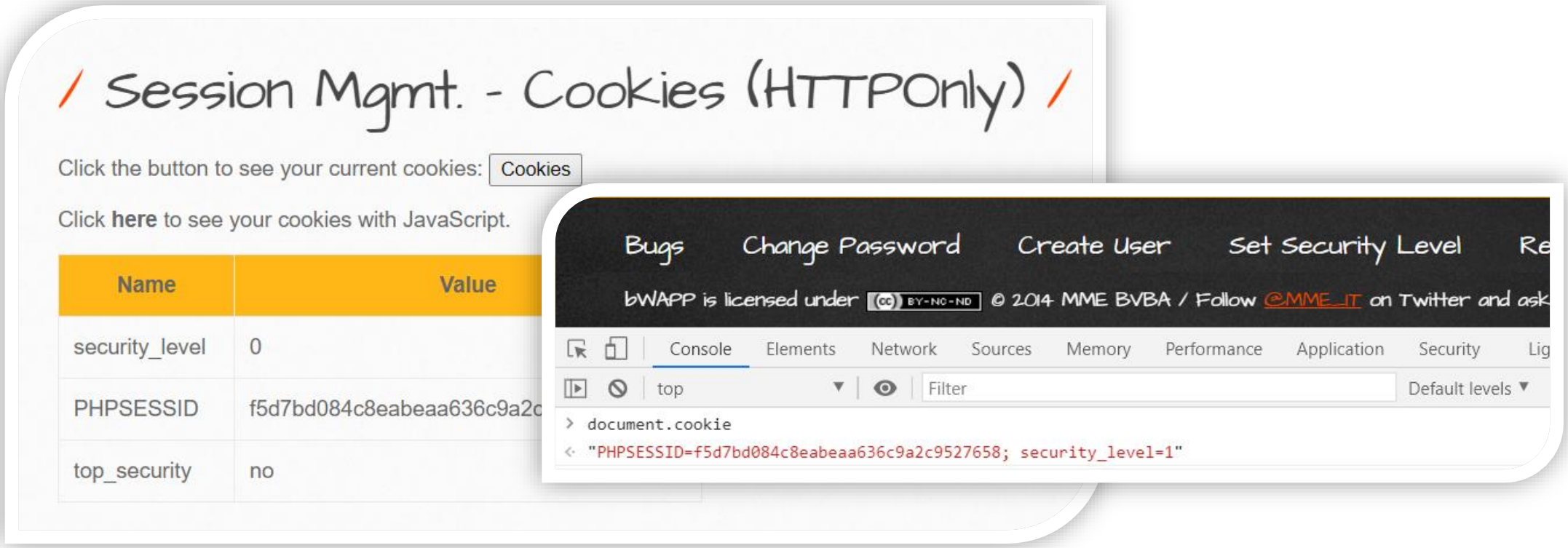
- Salt 이용

- ...

Session management – Cookies (HTTPOnly)



Session management – Cookies (HTTPOnly)



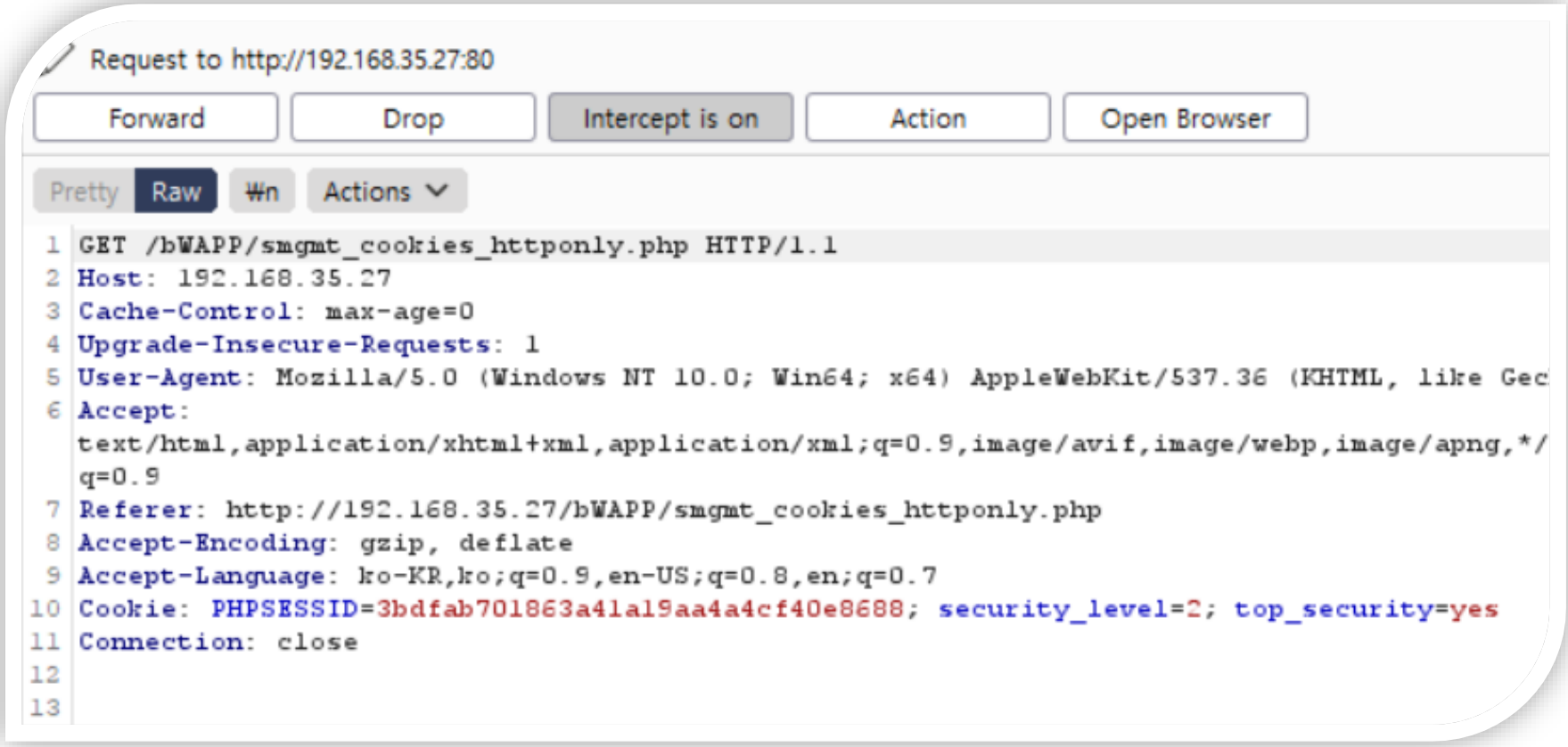
Session management – Cookies (HTTPOnly)

Session management – Cookies (HTTPOnly)

```
.cch($_COOKIE["security_level"])

    case "0" :

        // The cookie will be available within the entire domain
        setcookie("top_security", "no", time()+3600, "/", "", false, false);
        break;

    case "1" :

        // The cookie will be available within the entire domain
        // Sets the Http Only flag
        setcookie("top_security", "maybe", time()+3600, "/", "", false, true);
        break;

    case "2" :

        // The cookie will be available within the entire domain
        // The cookie expires at end of the session
        // Sets the Http Only flag
        setcookie("top_security", "yes", time()+300, "/", "", false, true);
        break;

    default :

        // The cookie will be available within the entire domain
        setcookie("top_security", "no", time()+3600, "/", "", false, false);
        break;

}
```

setcookie (쿠키명, 쿠키값, 만료시간, 경로, 도메인, secure, httponly);

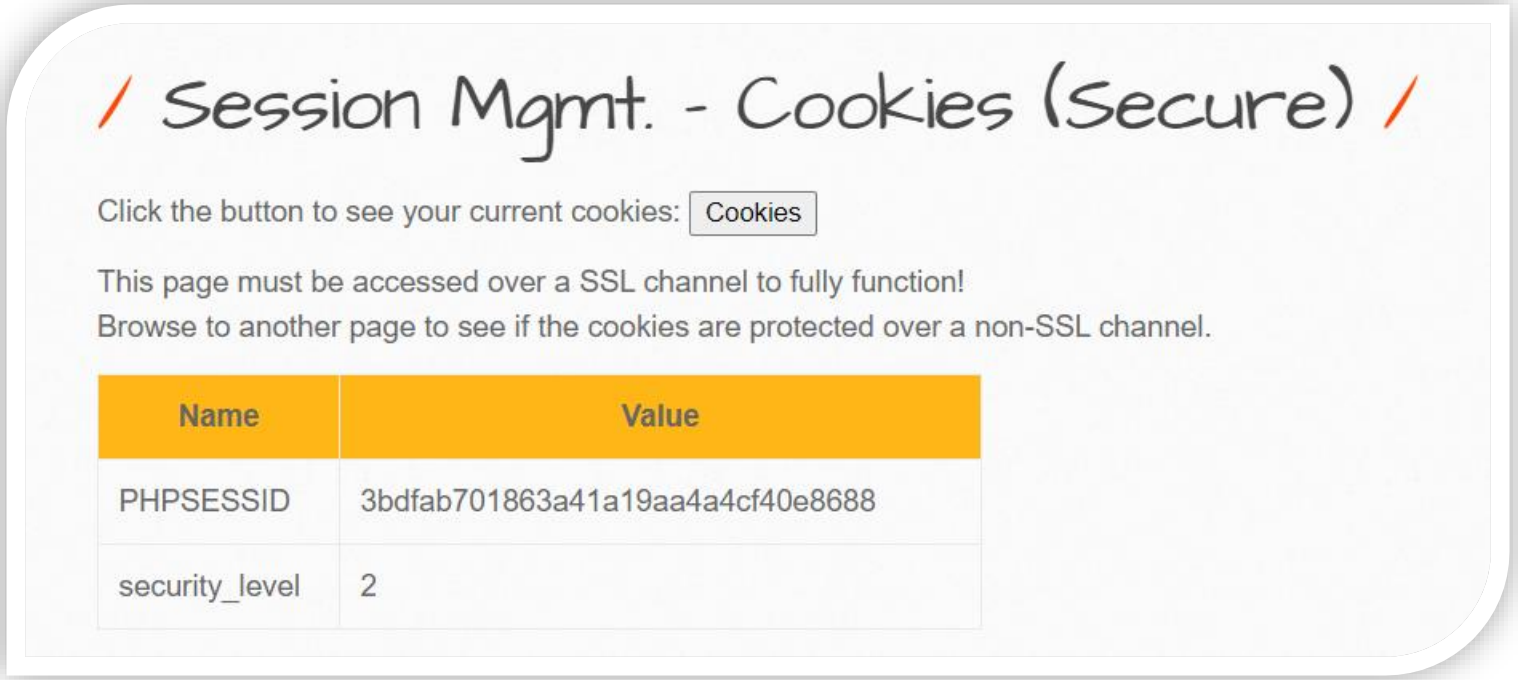
Session management – Cookies (HTTPOnly)



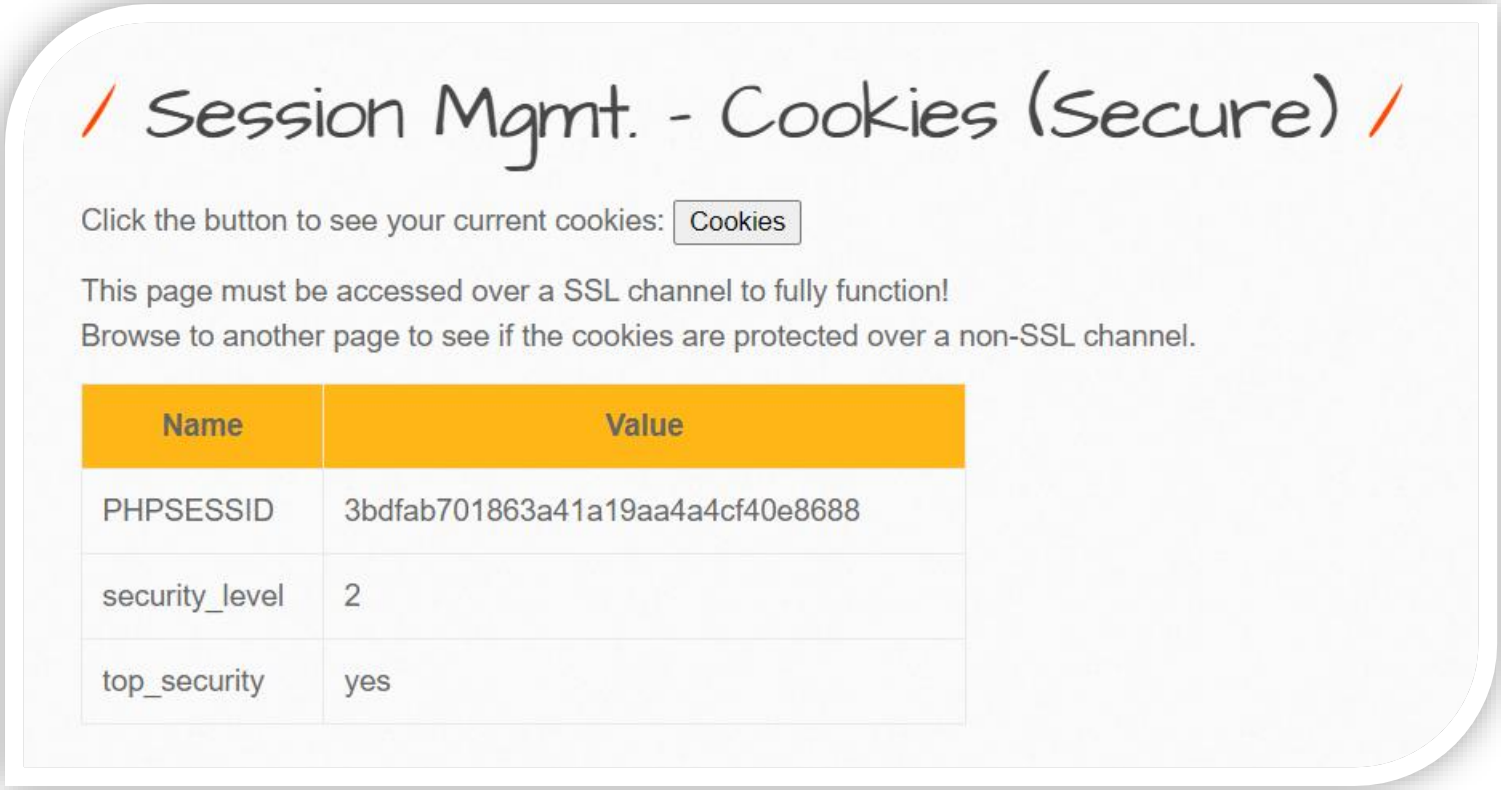
Session management – Cookies (Secure)

Session management – Cookies (Secure)

Session management – Cookies (Secure)



Session management – Cookies (Secure)



Session management – Cookies (Secure)

setcookie (쿠키명, 쿠키값, 만료시간, 경로, 도메인, secure, httponly);

```
(($_COOKIE["security_level"])

case "0" :

    $message.= "<p>Browse to another page to see if the cookies are protected over a non-SSL channel.</p>";

    // The cookie will be available within the entire domain
    // Sets the Http Only flag
    setcookie("top_security", "no", time()+3600, "/", "", false, true);
    break;

case "1" :

    $message = "<p>This page must be accessed over a SSL channel to fully function!<br />";
    $message.= "Browse to another page to see if the cookies are protected over a non-SSL channel.</p>";

    // The cookie will be available within the entire domain
    // Sets the Http Only flag and the Secure flag
    setcookie("top_security", "maybe", time()+3600, "/", "", true, true);
    break;

case "2" :

    $message = "<p>This page must be accessed over a SSL channel to fully function!<br />";
    $message.= "Browse to another page to see if the cookies are protected over a non-SSL channel.</p>";

    // The cookie will be available within the entire domain
    // The cookie expires at end of the session
    // Sets the Http Only flag and the Secure flag
    setcookie("top_security", "yes", time()+300, "/", "", true, true);
    break;

default :

    $message.= "<p>Browse to another page to see if the cookies are protected over a non-SSL channel</p>";

    // The cookie will be available within the entire domain
    // Sets the Http Only flag
    setcookie("top_security", "no", time()+3600, "/", "", false, true);
    break;;
```

OWASP TOP 10

A1 - Injection	A6 – Security Misconfigurations
A2 – Broken Authentication	A7 – Cross Site Scripting (XSS)
A3 – Sensitive Data Exposure	A8 – Insecure Deserialization
A4 – XML External Entities (XXE)	A9 – Using Components with Known Vulnerabilities
A5 – Broken Access Control	A10 – Insufficient Logging and Monitoring