NETWORK SECURITY

한양대학교 소프트웨어융합대학 소프트웨어학부
이연준 교수

주요 사항

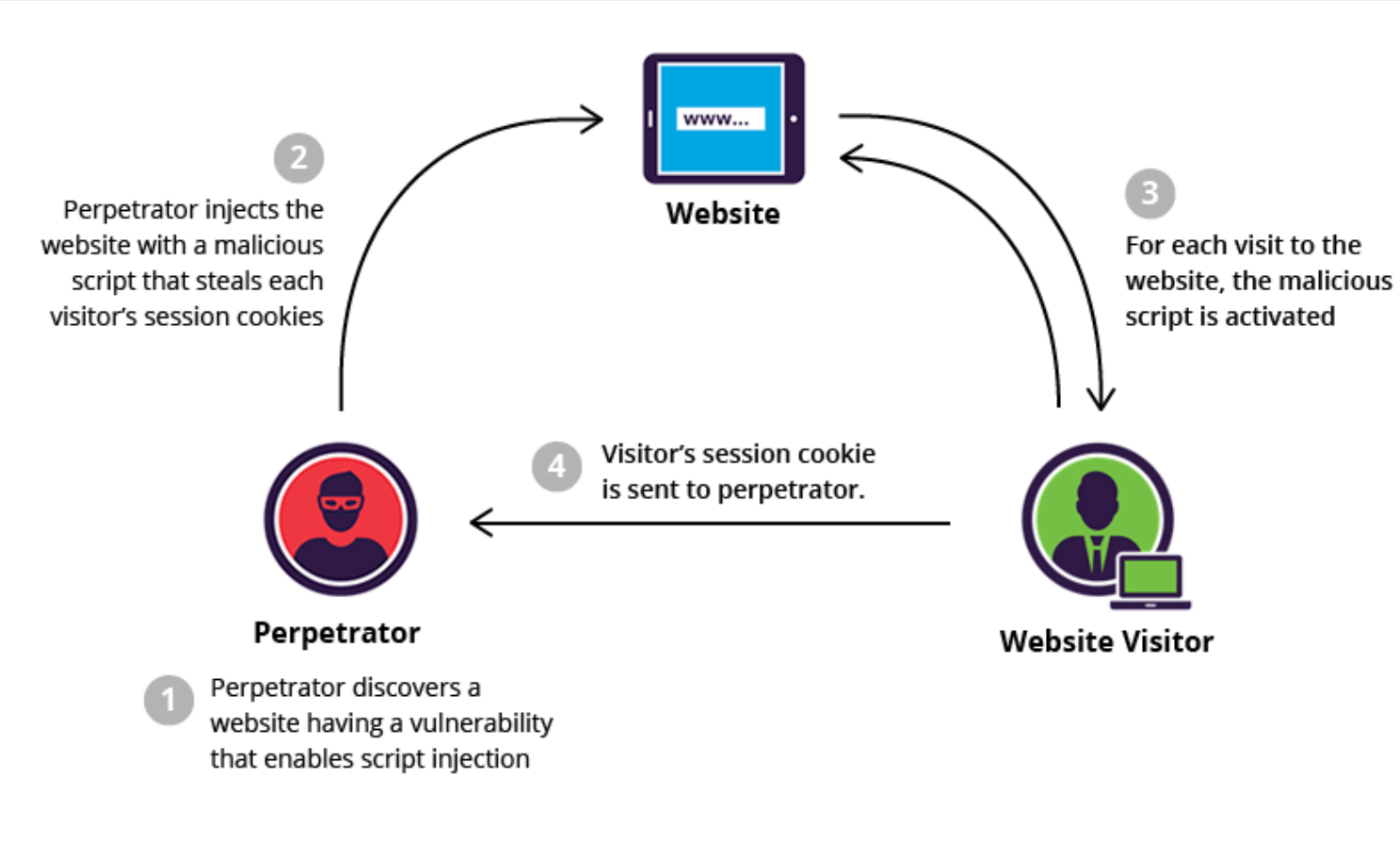
- **Cross-Site Request Forgery (CSRF)** 공격 기법에 대한 이해
- **Lab Preparation**
  - 실습 환경 구성 **(Elgg)**
- **Lab Task**
  - **HTTP Header Live Tool**
  - 다양한 **CSRF Attack** 실습
  - **CSRF Attack** 대응법
- **Lab Question**
- **Evaluation**

Cross-Site Request Forgery (CSRF)

- 사이트 간 요청 위조
- 즉, 사용자가 자신의 의지와는 관계 없이 공격자가 의도한 행위 **(Modify, Delete, Insert)**를 특정 **Website**에 요청
- **XSS**와 **CSRF**의 차이점

	XSS	**CSRF**
공격 방식	악성 **Script**가 **Client**에서 동작	인증 권한 기반**,** 공격자가 의도한 행위를 수행
공격 시점	클라이언트에서 인증 정보 **(Cookie)** 전송	인증된 사용자가 서버를 공격 **or** 변조
Script 사용 여부	**O**	**X**
공격 준비	**XSS** 취약점 분석 후 즉시 공격이 가능	서버 **Request** 및 **Response**의 분석 필요
예시	**<iframe>** **<script>document.location=…** **</script>** **</iframe>**	**(**사용자 로그인 후**)** **<img src="admin password change">**

Cross-Site Request Forgery (CSRF)

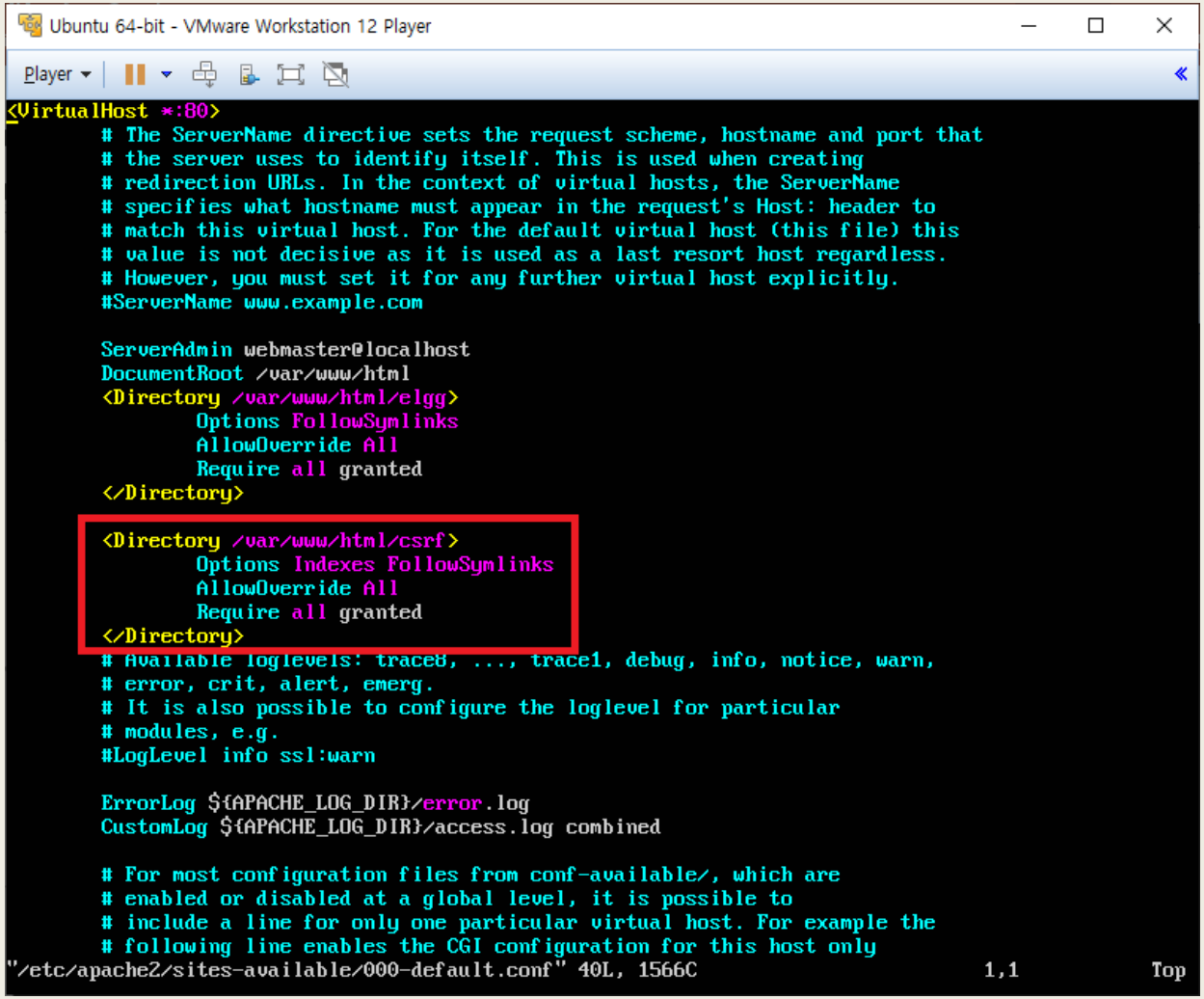


LAB PREPARATION

실습 환경 구성 준비

- **sudo vi /etc/apache2/site-available/000-default.conf**
- **000-default.conf** 내용을 다음과 같이 수정

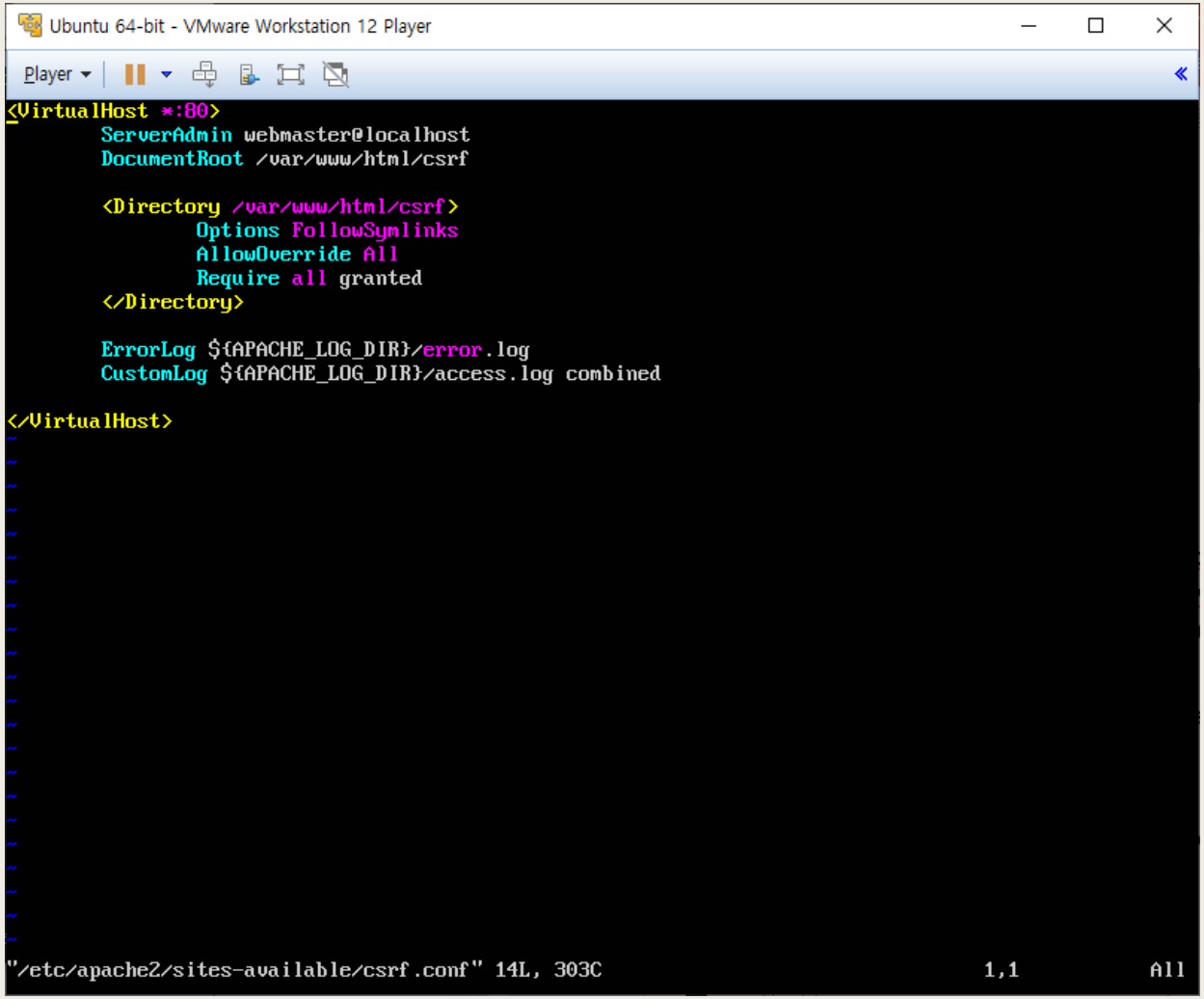
실습 환경 구성 준비


Ubuntu 64-bit - VMware Workstation 12 Player
Player
<VirtualHost *:80>
The ServerName directive sets the request scheme, hostname and port that
the server uses to identify itself. This is used when creating
redirection URLs. In the context of virtual hosts, the ServerName
specifies what hostname must appear in the request's Host: header to
match this virtual host. For the default virtual host (this file) this
value is not decisive as it is used as a last resort host regardless.
However, you must set it for any further virtual host explicitly.
#ServerName www.example.com
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
<Directory /var/www/html/elgg>
Options FollowSymlinks
AllowOverride All
Require all granted
</Directory>
<Directory /var/www/html/csrf>
Options Indexes FollowSymlinks
AllowOverride All
Require all granted
</Directory>
Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
error, crit, alert, emerg.
It is also possible to configure the loglevel for particular
modules, e.g.
#LogLevel info ssl:warn
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
For most configuration files from conf-available/, which are
enabled or disabled at a global level, it is possible to
include a line for only one particular virtual host. For example the
following line enables the CGI configuration for this host only
"/etc/apache2/sites-available/000-default.conf" 40L, 1566C
1,1
Top

실습 환경 구성 준비

- **Create CSRF Directory**
  - **sudo mkdir /var/www/html/csrf**
  - **sudo chown –R www-data:www-data /var/www/html/csrf**
  - **sudo chmod –R 755 /var/www/html/csrf**
- **Create CSRF Config File**
  - **sudo vi /etc/apache2/sites-available/csrf.conf**
- **Enable Elgg Config File & Rewrite Module**
  - **sudo a2ensite csrf.conf**
  - **sudo systemctl restart apache2.service**

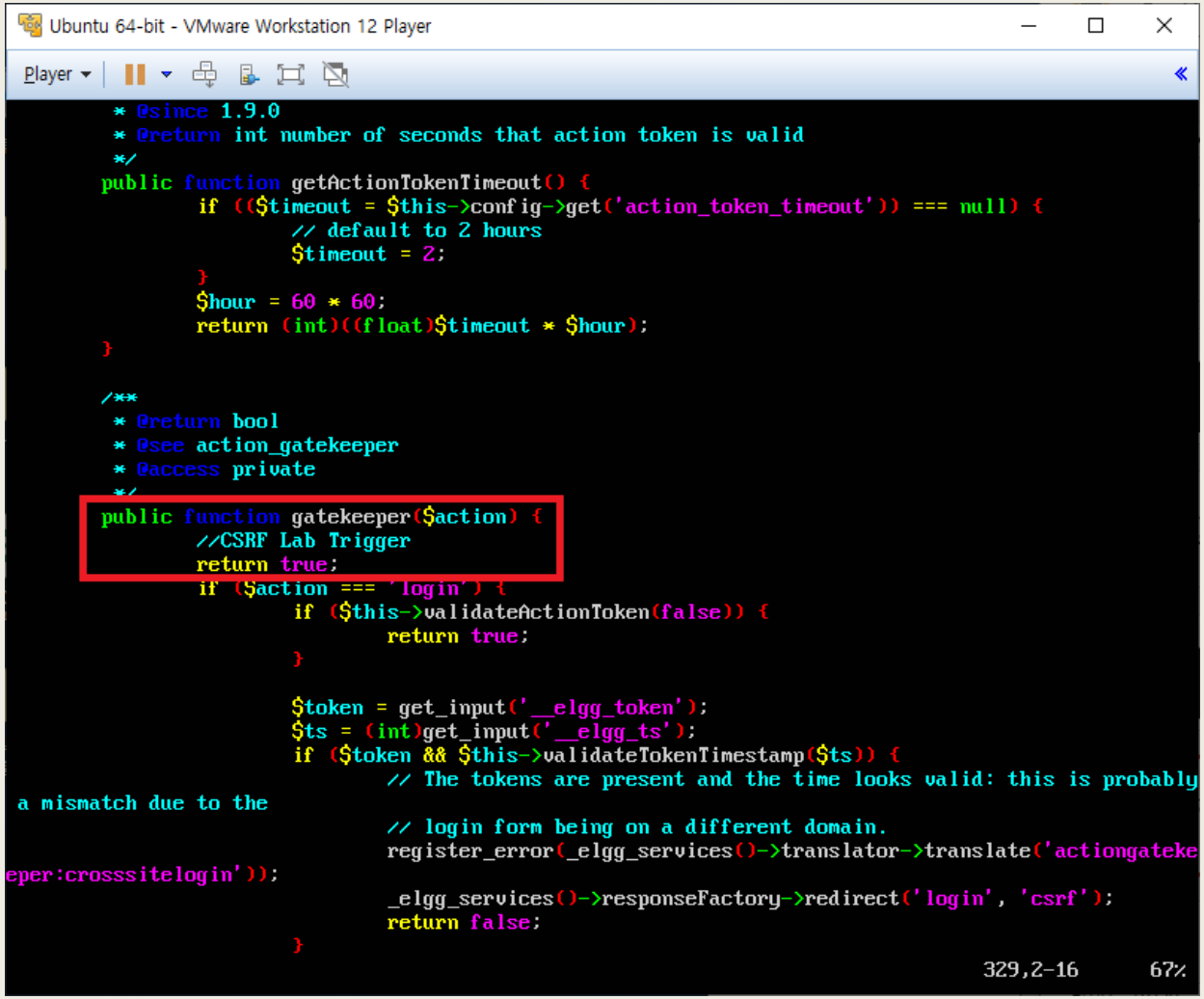
실습 환경 구성 준비


Ubuntu 64-bit - VMware Workstation 12 Player
Player
<VirtualHost *:80>
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/csrf
<Directory /var/www/html/csrf>
Options FollowSymlinks
AllowOverride All
Require all granted
</Directory>
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
"/etc/apache2/sites-available/csrf.conf" 14L, 303C
1,1
All

실습 환경 구성 준비

- **Modify CSRF Config**
  - **cd /var/www/html/elgg/vendor/elgg/elgg/engine/classes/Elgg/**
  - **sudo vi ActionsService.php**
  - **gatekeeper()** 함수를 찾아서 다음과 같이 수정할 것
  - **Vi editor**에서 특정 **Keyword** 찾는 방법
    - **:/(찾을 키워드)**

실습 환경 구성 준비


Ubuntu 64-bit - VMware Workstation 12 Player
Player
* @since 1.9.0
* @return int number of seconds that action token is valid
*/
public function getActionTokenTimeout() {
if (($timeout = $this->config->get('action_token_timeout')) === null) {
// default to 2 hours
$timeout = 2;
}
$hour = 60 * 60;
return (int)((float)$timeout * $hour);
}
/**
* @return bool
* @see action_gatekeeper
* @access private
*/
public function gatekeeper($action) {
//CSRF Lab Trigger
return true;
if ($action === 'login') {
if ($this->validateActionToken(false)) {
return true;
}
$token = get_input('__elgg_token');
$ts = (int)get_input('__elgg_ts');
if ($token && $this->validateTokenTimestamp($ts)) {
// The tokens are present and the time looks valid: this is probably
a mismatch due to the
// login form being on a different domain.
register_error(_elgg_services()->translator->translate('actiongateke
eper:crosssitelogin'));
_elgg_services()->responseFactory->redirect('login', 'csrf');
return false;
}
329,2-16 67%

LAB TASK

HTTP Requests

- **HTTP Request**를 파악하기 위해서는 여러 가지 방법이 있음

1. **Web browser**에서 자체적으로 제공하는 개발자 도구를 사용
2. **Google Chrome** 또는 **Firefox**의 **Web Store**에서 **HTTP Header Live**라는 **Plug-In**을 사용
3. **Wireshark**를 통한 **HTTP Request** 분석

- 어느 쪽을 사용해도 무방함

CSRF Attack using GET Request

- **Boby**를 공격자, **Alice**를 피해자로 가정한다.
- **Boby**는 **Alice**에게 친구 신청을 하였으나 **Alice** 쪽에서는 이를 거부하고 있으므로 **CSRF** 공격을 통해 강제로 친구추가를 하는 것이 목적.
- **Alice**에게 특정 게시물을 통해 **URL**을 전송하여 **CSRF** 공격을 실행
- **Alice**가 **Elgg**에 접속한 상태에서 해당 **URL**을 통해 페이지를 방문하면 **Boby**가 친구 목록에 추가되어 있어야 함.
- 이를 해결하기 위해서는 정상적인 친구 신청을 했을 때 나오는 **GET Request**를 분석해야 함.
- **GET Request**의 경우 자동으로 동작하는 **img** 태그를 사용하는 것을 권장.
- **Hint : elgg ts**와 **elgg token**값을 반드시 사용해야함

CSRF Attack using POST Request

- 공격자와 피해자는 이전과 동일하게 설정함.
- **Boby**는 **Alice**의 **Profile**에 **“Boby is my Hero”**라는 메시지를 남기는 것이 이번 **Task**의 목표임.
- 이전과 동일하게 **URL**을 클릭하여 특정 페이지로 이동하면 프로필의 내용이 변경되도록 함.
- 프로필 수정은 **/profile/edit.php**에서 이뤄지며, **GET**과 **POST Request**를 모두 허용함.
- 이번 **Task**에서는 **POST Request**를 위조하여 **CSRF**를 시도함.
- **POST**는 **GET**과 유사하나 매개변수가 들어있는 위치가 다름

CSRF Attack using POST Request (Cont'd)

- **GET Request** 때 헤더를 분석한 것과 마찬가지로 **POST**에서의 구조 파악을 마친 후, **JavaScript** 코드를 통한 **CSRF** 공격을 수행할 것.
- 예제 코드는 다음과 같음

CSRF Attack using POST Request (Cont'd)

```
<html>
<body>
<h1>This page forges an HTTP POST request.</h1>
<script type="text/javascript">

function forge_post()
{
    var fields;

    // The following are form entries need to be filled out by attackers.
    // The entries are made hidden, so the victim won't be able to see them.
    fields += "<input type='hidden' name='name' value='****'>";
    fields += "<input type='hidden' name='briefdescription' value='****'>";
    fields += "<input type='hidden' name='accesslevel[briefdescription]'
                                    value='2'>";                          ①
    fields += "<input type='hidden' name='guid' value='****'>";

    // Create a <form> element.
    var p = document.createElement("form");

    // Construct the form
    p.action = "http://www.example.com";
    p.innerHTML = fields;
    p.method = "post";

    // Append the form to the current page.
    document.body.appendChild(p);

    // Submit the form
    p.submit();
}


// Invoke forge_post() after the page is loaded.
window.onload = function() { forge_post();}
</script>
</body>
</html>
```

Implementing a countermeasure for Elgg

- **Elgg**에는 기본적으로 **CSRF** 공격을 방어하기 위한 대책이 내장되어 있음
- **CSRF** 방어는 크게 어렵지 않으며, 일반적인 방법 중 두 가지를 소개함

1. **Secret-token approach : Web App**은 해당 페이지에 비밀 토큰을 포함할 수 있으며, 이 페이지에서 오는 모든 **Request**는 해당 토큰을 포함함. 서로 다른 사이트에서의 **Request**는 이 토큰을 얻을 수 없으므로 **Request**가 위조될 경우 쉽게 식별이 가능
2. **Referrer header approach : Web App**은 **Referrer header**를 통해 **Request**의 원본 페이지를 확인 할 수도 있음. 그러나, 개인 정보 보호의 문제로 인해 **Referrer header** 정보는 이미 **Client**단에서 필터링됨.

Implementing a countermeasure for Elgg (Cont'd)

- Elgg는 Secret-token 방식을 사용함.
- 앞에서 설명한 elgg ts와 elgg token이라는 두 parameter를 통해 구현하였음.
- POST에서는 메시지의 본문에, GET에서는 URL 문자열에서 발견 가능함.
- Elgg에서 일어나는 모든 작업에는 secret token과 timestamp가 추가됨
- Secret token과 timestamp는 사용자의 action을 요하는 페이지에 모두 포함되어 있음.

Implementing a countermeasure for Elgg (Cont'd)

- 실습 환경 구성 시 추가했던 **ActionsService.php**에서 **return true;** 부분을 주석처리.
- 주석 처리를 한 이후 이전에 시행했던 **CSRF** 공격들을 다시 수행해볼 것.
- 수행하면서 보이는 **HTTP Request** 분석 결과를 보고 이전 결과와 다른 점이 있는지 확인할 것.

LAB QUESTION

Lab Question

1. 변조된 **HTTP Request**가 제대로 작동하려면 **Alice**의 사용자 **ID(guid)**가 필요합니다. **Boby**는 공격 전에 **Alice**를 확실한 목표로 정했으며 **Alice**의 사용자 **ID**를 얻는 방법을 찾을 수 있습니다. 하지만, **Boby**는 **Alice**의 **Elgg** 비밀번호를 모르기에 **Alice**의 계정에 로그인하여 정보를 얻을 수는 없습니다. 그럼 사용자 **ID**를 얻기 위해서 어떻게 해결해야 하는지 설명하세요.

2. **Boby**는 자신이 생성한 공격용 웹 페이지를 방문하는 사람이라면 누구든지 공격을 당하도록 하고 싶습니다. 하지만 방문하는 사람을 미리 알 수 없습니다. 불특정 다수의 방문자의 **Elgg** 프로필을 수정하는 공격을 한다고 했을 때, 여전히 **CSRF** 공격이 가능합니까?

Evaluation

- **Lab Task** 진행
  - **GET**과 **POST Request**를 분석한 화면을 캡처할 것
  - **Get**과 **POST Request** 변조를 통한 **Task**를 수행하고 작성한 코드와 결과 화면을 캡처할 것
  - 초기에 추가한 **return true;** 부분을 주석처리 한 후 **CSRF** 공격을 수행한 후의 결과 화면과 **HTTP Request** 결과를 캡처할 것
- **Lab Question**
  - 주어진 문항에 대한 답과 해결 방안에 대해 간략하게 서술
- **Lab Task** 수행 결과를 위와 같이 명시한 대로 캡처하여 **MS Word** 또는 **PDF** 파일로 결과를 제출할 것.

Evaluation

- 과제 제출 기한 : 2019/11/04 23:59
- 과제 제출 시 메일 제목은 ‘본인 이름_학번’으로 제출
  - 예) 이석원_2019101059
- sevenshards00@gmail.com으로 보낼 것.

Q&A