1. Защита от XSS.

XSS - это когда злоумышленник пытается через формы на сайте (обратная связь, оформление заказа и т.п.) добавить свой javascript-код, который затем выполнится в браузере админа/менеджера сайта или других пользователей.
Когда мы запускаем функцию htmlspecialchars(), она берёт нашу строку и заменяет некоторые символы в ней (кавычки, угловые скобки и т.д.) на мнемоники, чтобы браузер гарантированно вывел нашу строку на экран как строку, не пытаясь выполнять её как код. Поэтому необходимо в (index.php) использовать функцию htmlspecialchars с аргументами
$_POST[‘то что передаем’], ENT_QUOTES,’UTF-8’):

```
$name=htmlspecialchars($_POST['name'],ENT_QUOTES,'UTF8');
$email=htmlspecialchars ($_POST['email'], ENT_QUOTES, 'UTF-8');
$year=htmlspecialchars ($_POST['year'], ENT_QUOTES, 'UTF-8');
$pol=htmlspecialchars ($_POST['pol'], ENT_QUOTES, 'UTF-8');
$limb=htmlspecialchars ($_POST['limb'], ENT_QUOTES, 'UTF-8');
$super=htmlspecialchars ($_POST['super'], ENT_QUOTES, 'UTF-8');
```

Так же используем эту функцию когда берем данные из бд:

```
$values['name']=htmlspecialchars ($inf[0]['name'],ENT_QUOTES, 'UTF-8');
$values['email']=htmlspecialchars ($inf[0]['email'],ENT_QUOTES, 'UTF-8');
$values['year']=htmlspecialchars ($inf[0]['year'],ENT_QUOTES, 'UTF-8');
$values['pol']=htmlspecialchars($inf[0]['sex'],ENT_QUOTES, 'UTF-8');
$values['limb']=htmlspecialchars ($inf[0]['limbs'],ENT_QUOTES, 'UTF-8');
$values['bio']=($inf[0]['bio'],ENT_QUOTES, 'UTF-8');
```

2. Защита от SQL Injection

Уязвимость заключается в том, что у злоумышленника имеется возможность изменить запрос к базе данных (это может привести к потере файлов, нарушению работы бд, к раскрытию конфиденциальных данных). Чтобы снизить риски взлома, необходимо использовать PDO и плейстхолдеры. Ведь сначала производится подключение к базе, потом подготавливается запрос, затем отдельно указываются переменные, и в конце выполняется запрос. Данные отправляются в виде переменных и сервер, увидев кавычку или любой другой символ не относящийся к переменной, поймет, что это не часть запроса.

```
$upd1=$db- >prepare("insert into power set p_name=:power,p_id=:id");
$upd1->bindParam(':id',$id);
```

Один из примеров.

3. Защита от CSRF

Уязвимость заключается в том, что браузер не может понять, было ли выполнено действие непосредственно пользователем, или же он сделал это неумышленно, например при посещении вредоносного сайта, его ресурсом был отправлен запрос на наш сайт. Необходимо создать токен, случайное число (случайный набор байт), его сервер передает клиенту, а клиент возвращает серверу. Сама защита заключается в проверке токена, который сгенерировал сервер и токена который прислал пользователь.

4. Защита от Include Upload

Доступ загрузки файлов отсутствует, необходимость защиты отсутствует. К включаемым файлам доступа не имеется (include('form.php');)).