hw 3

小组成员：

- PB20111704 张宇昂
- PB20111647 鲍润晖
- PB20000103 王炳勋
- PB20111651 何泽昊
- PB19071508 唐思渝

注：部分回答末尾的括号内容为该回答的依据，来自于课程PPT等。

Q1

案例：攻击者建立服务器，在与https://xyz.com相同IP地址的不同端口号上监听，设监听端口为x。假设浏览器用户打开https://xyz.com进行账户登录时，该站点会向https://xyz.com.login发送信息（例如，对后者进行AJAX请求，前者会发送用户凭证）。攻击者能够在用户浏览器上写入脚本，向https://xyz.com:x/login发送信息，以捕获用户信息。

由于修改后的同源策略只基于域名，而该攻击从同域名不同端口上调取资源，所以能够成功。但完整的同源策略能够防止该攻击。

Q2

a）

- pseudocode

```
<script>
    var displayData = function (leaked) {
      document.write(JSON.stringify(leaked));
    };
</script>
<script src="//bank.com/userdata.js"></script>
```

将JavaScript 对象转换为字符串并输出
输出

```
{
"name": "John Doe",
"AccountNumber": 12345,
"Balance": 45
}
```

b）

```
<script src="//bank.com/userdata.js"> </script> (*)
```

可以改为

```
<script type="text/javascript">
    displayData(
{
"name": "John Doe",
"AccountNumber": 12345,
"Balance": 45
}
)
</script>
```

并将userdata.js的对应代码删去
由于evil.com无法直接对bank.com进行访问，从而无法重写已存在的函数，因此避免了XSSI攻击

Q3

a

DoH可以防止DNS被中间人攻击：DoH可以加密客户端和服务器之间的数据，可以防止用户的DNS请求被窃听，从而避免攻击者DNS劫持
相较于DoT使用专用的853端口，DoH流量和普通HTTPS流量一样走443端口，所以DoH还可以防止流量监视

b

DoH，DoT只能加密DNS查询，所以以下攻击还是有效的：

- DNS放大攻击（理由见下题）
- DNS欺骗、DNS重新绑定（理由见下下题）
- DNS拼写仿冒（攻击者注册一个和目标域名很相似的域名）

同时DoH，DoT还会遭受降级攻击：
在浏览器中，DoT或DoH通常有三种配置模式：关闭、机会模式和强制模式。机会模式尝试为DNS使用安全传输，但如果前者不可用，则退回到未加密的DNS。这种模式容易受到降级攻击。

c

DNS放大攻击时，攻击者利用僵尸网络中的大量的被控主机，伪装成被攻击主机(IP地址)，在特定时间点连续向多个允许递归查询的DNS服务器发送大量DNS服务请求，迫使其提供应答服务，经DNS服务器放大后的大量应答数据发送到被攻击主机，形成攻击流量

但是DoH不能避免伪装IP，也因此不能避免DNS放大攻击。但是考虑到DoH包比DNS更复杂，所以此时开展DNS放大攻击的代价就更高了

d

DNS重绑定攻击时，用户在浏览器输入域名，浏览器通过DNS服务器将它解析为IP，但是当这个解析的TTL时间过去以后，浏览器需要再次通过DNS服务器解析这个域名，但是此时DNS服务器会回复一个新的IP，所以称为重绑定。此时浏览器的同源策略不会发现问题，因为前后的域名没有变化。

此时DoH，DoT也不能防止这些问题，因为是DNS服务器有害，而和传输的协议是否加密没有关系。

Q4

a

1. 攻击者让已登录被攻击网站的用户访问一个**恶意**链接/表单
2. 攻击者通过提供的**恶意**链接/表单获取用户在被攻击网站上的cookie
3. 攻击者利用cookie伪造身份进行攻击

b

由于**Same Origin Policy**的存在，攻击者无法访问token的值，所以无法构造出包含有效令牌的恶意请求

c

我认为这种策略是可以防止攻击的，因为攻击者无法预测发送请求时网站的token值，所以无法有效的构造攻击请求

d

我认为这种策略是不能防止攻击的，因为攻击者可以监听或者截取其他请求的csrf_token来获取有效token，由于token是固定的字符串，这时攻击者就可以通过截获的固定cookie构造出有效的攻击请求了

e

它限制了浏览器只能发送来自于同一个源域名下的请求，可以防止其他网站或域名访问受攻击网站的 cookie 和令牌，能够有效避免来自其他网站的访问以及CSRF攻击的恶意请求

5

a)

script-src 'self'指令：作用是只允许加载相同的域的资源。因此浏览器只能运行相同的域的脚本，从而防止**XSS **（跨站脚本）攻击。

b)

frame-ancestors 'none'指令：作用是禁止浏览器从frame加载任何页面，无论frame是来自同一站点还是其他站点。由于**Clickjacking Attack**（点击劫持攻击）需要iframe加载虚假的页面，该指令能有效防止这类攻击。

（From PPT 5-153）

c)

sandbox 'allow-scripts'指令：HTML的sandbox属性允许对页面的权限进行限制。该指令作用是解除对页面运行脚本权限的限制（同时保留其他限制）。它允许嵌入的页面运行脚本，但不允许创建弹出窗口。

题目中提到：“This causes the page to be treated as being from a special origin that always fails the same-origin policy, among other restrictions.”该限制来源于sandbox属性，除非添加'allow-same-origin'属性来解除。Cookie的同

源策略定义为**(domain, path)**，只允许在与www.xyz.com同一域名和路径下进行Cookie信息的共享，因此会导致该页面读取www.xyz.com 的cookie失败。

（From PPT 5-79）

如果网站允许本站用户在文本中输入代码运行脚本，同时防止来自非同源网站的恶意脚本攻击，可以使用这个CSP头。

Q6

a

- 从主机A向主机P发送一个ICMP ping请求，P会响应该请求，记数据包中的标识字段为T1。
- 等待一分钟。
- 再次从主机A向主机P发送一个ICMP ping请求。收到响应后，记数据包中的标识字段为T2。
- 如果T2 - T1 > 1 ,那么可以得出P是在特定的一分钟窗口内向任何人（A 除外）发送了数据包。

b

- 从主机A向主机P发送一个SYN数据包，将源IP伪造为V的IP，目标端口为n。
- P主机收到数据包后，会认为是主机V试图与其建立连接，而不是A。因此，如果主机V在监听端口n，那么它将收到来自主机P的SYN/ACK响应。
- 由于主机V并没有试图与P建立连接，在收到它不期望的SYN/ACK数据包后，它会向源IP（即P）发送RTS数据包。
- 从主机A向主机P发送一个ICMP ping请求，并记录响应中数据报的IP标识字段值。将此值称为T1
- 等待一段时间后，再次从主机A向主机P发送一个ICMP ping请求，并记录响应数据报中的IP标识字段值。将此值称为T2
- 如果T2 - T1 大于1，那么在这期间，主机P收到了来自主机V的RST数据报，即V 正在侦听端口n

Q7

a

- 攻击者填满具有256个连接请求的表，需要发送256个数据包
- 以30秒的间隔发送5次未确认的连接请求，需要30 * 5 = 150秒
- 每秒发送 256/150 = 1.7个数据包
- TCP SYN 数据包大小为40字节，每秒发送40 * 1.7 = 68 字节的数据
- 故需要消耗68byte/s * 8 = 544 bps 的带宽

b

- 0.5 Mbps 的链路：具有0.5 * 10^6 / 8 = 62500 byte/s的带宽，需要每秒发送 62500/500 = 125 个 DNS 响应数据包，需要发送 125 * 60 = 7500 byte/s的 DNS 请求数据包,相当于消耗 0.06 Mbps 的带宽。。
- 2 Mbps 的链路：具有2 * 10^6 / 8 = 250000 byte/s的带宽，需要每秒发送 250000/500 = 500 个 DNS 响应数据包，需要发送 500 * 60 = 30000 byte/s的 DNS 请求数据包,相当于消耗 0.24 Mbps 的带宽。
- 10 Mbps 的链路：具有10 * 10^6 / 8 = 1250000 byte/s的带宽。需要每秒发送 1250000/500 = 2500 个 DNS 响应数据包，需要发送 2500 * 60 = 150000 byte/s的 DNS 请求数据包,相当于消耗 1.2 Mbps 的带宽。