😄

实验的知识点

1. 跨站请求伪造CSRF

- 简介：**跨站请求伪造**（Cross-site request forgery），也被称为 **one-click attack** 或者 **session riding**，是一种挟制用户在当前已登录的Web应用程序上执行非本意的操作的攻击方法。跟跨网站脚本（XSS）相比，**XSS** 利用的是用户对指定网站的信任，CSRF 利用的是网站对用户网页浏览器的信任
- 攻击原理：攻击者通过一些技术手段欺骗用户的浏览器去访问一个自己曾经认证过的网站并运行一些操作（如发邮件，发消息，甚至财产操作如转账和购买商品）。由于浏览器曾经认证过，所以被访问的网站会认为是真正的用户操作而去运行。这利用了web中用户身份验证的一个漏洞：**简单的身份验证只能保证请求发自某个用户的浏览器，却不能保证请求本身是用户自愿发出的**。
- 防御措施：
    - **加校验token**：由于CSRF的本质在于攻击者欺骗用户去访问自己设置的地址，所以如果要求在访问敏感数据请求时，要求用户浏览器提供不保存在cookie中，并且攻击者无法伪造的数据作为校验，那么攻击者就无法再运行CSRF攻击。这种数据通常是窗体中的一个数据项。服务器将其生成并附加在窗体中，其内容是一个伪随机数。当客户端通过窗体提交请求时，这个伪随机数也一并提交上去以供校验。正常的访问时，客户端浏览器能够正确得到并传回这个伪随机数，而通过CSRF传来的欺骗性攻击中，攻击者无从事先得知这个伪随机数的值，服务端就会因为校验token的值为空或者错误，拒绝这个可疑请求。
    - **检查Referer字段**：HTTP头中有一个Referer字段，这个字段用以标明请求来源于哪个地址。在处理敏感数据请求时，通常来说，Referer字段应和请求的地址位于同一域名下。以上文银行操作为例，Referer字段地址通常应该是转账按钮所在的网页地址，应该也位于www.examplebank.com之下。而如果是CSRF攻击传来的请求，Referer字段会是包含恶意网址的地址，不会位于www.examplebank.com之下，这时候服务器就能识别出恶意的访问。

2.Cookie劫持（也可参考C7的ip欺骗部分）

- 原理：Cookie劫持就是攻击者盗取用户的Cookie，然后利用这些Cookie模拟用户的身份访问Web应用程序。Cookie是Web应用程序中用于验证和跟踪用户的一种机制，它通常存储在用户的浏览器中。当攻击者获得了用户的Cookie，就可以利用这些Cookie来访问Web应用程序，并获得与该用户相关的敏感信息，例如用户名、密码、银行账号等。攻击者可以通过各种方式窃取Cookie，例如网络嗅探、恶意软件。
- 防御：**加密Cookie**。为了增加Cookie的安全性，应用程序可以对Cookie进行加密。加密后的Cookie只有在服务器端才能被解密，从而减少了攻击者窃取Cookie的可能性。**在Cookie中加入时间戳**。为了防止攻击者使用过期的Cookie来访问应用程序，应用程序可以在Cookie中加入时间戳，并在服务器端验证时间戳是否合法。**在Cookie中加入IP地址。**为了防止攻击者通过欺骗DNS服务器来劫持Cookie，应用程序可以在Cookie中加入IP地址，并在服务器端验证IP地址是否合法。**采用HTTPS协议。**HTTPS协议可以为Web应用程序提供数据加密和身份验证，从而防止Cookie被窃取。

3.sql注入（WEB安全威胁及防范技术ppt19-36）

- 原理：前端的数据传入到后台处理时，没有做严格的判断，导致其传入的“数据”拼接到SQL语句中后，被当作SQL语句的一部分执行， 从而导致数据库受损（被脱裤、被删除、甚至整个服务器权限沦陷）。
- 类型
  - 数字型注入
  - 字符型注入
  - 搜索注入
  - 盲注
  - XX注入
  - 增删改注入
- 步骤：
  - 检测注入点
  - 判断是否存在 SQL 注入可能

    - 数据库爆破
    - 数据库表爆破
    - 字段爆破
    - 用户名、密码爆破
- 防御：对传进SQL语句里面的变量进行过滤，不允许危险字符传入。

4.蠕虫：（参考计算机病毒的C8）

蠕虫病毒（Worm ）是一种在计算机系统中传播并自我复制的恶意软件。与其他病毒类型不同，蠕虫病毒无需依靠宿主文件或程序来传播，而是通过网络或系统漏洞自动传播到其他计算机。
蠕虫病毒通常利用操作系统或应用程序的安全漏洞，通过网络传输或利用共享文件、电子邮件等方式进行传播。一旦蠕虫病毒感染了一台计算机，它会利用系统漏洞自我复制，并尝试传播到其他连接的计算机，形成一个传播链。

5.时序攻击

- 原理：时序攻击是一种侧信道攻击,攻击者试图通过分析加密算法的时间执行来推导出密码。每一个逻辑运算在计算机需要时间来执行,根据输入不同,精确测量执行时间,根据执行时间反推出密码。
- 防御：
    - 频繁更新密钥
    - 常数时间操作：在设计和实现算法时，尤其是涉及到敏感信息处理的算法时，应尽可能保证所有操作都在常数时间内完成

6.内容安全策略CSP（属于防御）

- 简介：为了缓解很大一部分潜在的跨站脚本问题，浏览器的扩展程序系统引入了内容安全策略（CSP）的一般概念。这将引入一些相当严格的策略，会使扩展程序在默认情况下更加安全，开发者可以创建并强制应用一些规则，管理网站允许加载的内容。
- 原理：定义了哪些来源的内容可以被加载和执行，用于保护 Web应用程序免受各种攻击，包括跨站脚本（XSS）、点击劫持和数据注入。csp 策略可以通过 HTTP 响应

头或 <meta> 标签中的 Content-Security-Policy 字段来定义和传输给浏览器，浏览器在加载页面时会根据该策略来执行相应的安全限制，我们可以通过使用'nonce' 关键字来限制只允许特定 nonce 值的脚本执行。

7.ARP欺骗

- 简介：攻击者通过伪造或篡改 ARP 协议的信息，将目标计算机发送的数据包转发到攻击者的计算机，从而实现窃取数据或进行中间人攻击等恶意行为。
- 原理：攻击者通常会向目标计算机发送伪造的 ARP 广播请求，以欺骗目标计算机将攻击者的MAC 地址误认为是与目标计算机相连的网关或其他计算机的 MAC 地址，从而将目标计算机发送的数据包转发到攻击者的计算机
  上。

8.DNS欺骗

- 简介：也称为 DNS 缓存投毒（DNS Cache Poisoning），是一种网络攻击方式，攻击者通过伪造或篡改DNS服务器返回的DNS解析结果，将受害者重定向到恶意网站或者收集用户的敏感信息。
- 原理：DNS 欺骗攻击利用 DNS 解析过程中的漏洞，向受害者发送伪造的 DNS 解析响应，将目标域名解析到攻击者控制的恶意 IP 地址，从而实现重定向、劫持和信息窃取等恶意行为。
- 防御：检查本机的HOSTS文件，以免被攻击者加了恶意站点进去；其次要确认自己使用的DNS服务器是ISP提供的，因为当前ISP服务器的安全工作还是做得比较好的，一般水平的攻击者无法成功进入；如果是依靠网关设备自带的DNS解析来连接Internet的，就要拜托管理员定期检查网关设备是否遭受入侵。

9.中间人攻击MITM

- 简介：攻击者在网络通信过程中窃取、篡改或伪造通信内容，使得通信双方无法察觉到攻击的存在。
- 原理：攻击者常考虑的方式是ARP欺骗或DNS欺骗等，将会话双方的通讯流暗中改变，而这种改变对于会话双方来说是完全透明的。以常见的DNS欺骗为例，目标将其DNS请求发送到攻击者这里，然后攻击者伪造DNS响应，将正确的IP地址替换为其他IP，之后你就登陆了这个攻击者指定的IP，而攻击者早就在这个IP中安排好了一个伪造的网站如某银行网站，从而骗取用户输入他们想得到的信息，如银行账号及密码等。

- 防御：可以将一些机密信息进行加密后再传输

10.会话劫持

- 简介：结合了嗅探以及欺骗技术在内的攻击手段
- 原理：在一次正常的通信过程中，攻击者作为第三方参与到其中，或者是在数据里加入其他信息，甚至将双方的通信模式暗中改变，即从直接联系变成有攻击者参与的联系。简单地说，就是攻击者把自己插入到受害者和目标机器之间，并设法让受害者和目标机器之间的数据通道变为受害者和目标机器之间存在一个看起来像“中转站”的代理机器（攻击者的机器）的数据通道，从而干涉两台机器之间的数据传输，例如监听敏感数据、替换数据等。由于攻击者已经介入其中，他能轻易知道双方传输的数据内容，还能根据自己的意愿去左右它。
- 防御：会话劫持都要结合嗅探以及欺骗技术在内的攻击手段，必须依靠ARP和MAC做基础，所以网管应该使用交换式网络（通过交换机传输）代替共享式网络（通过集线器传输），这可以降低被窃听的机率，当然这样并不能根除会话劫持，还必须使用静态ARP、捆绑MAC+IP等方法来限制欺骗，以及采用认证方式的连接等。