GooAnn
gooann.com 谷安天下

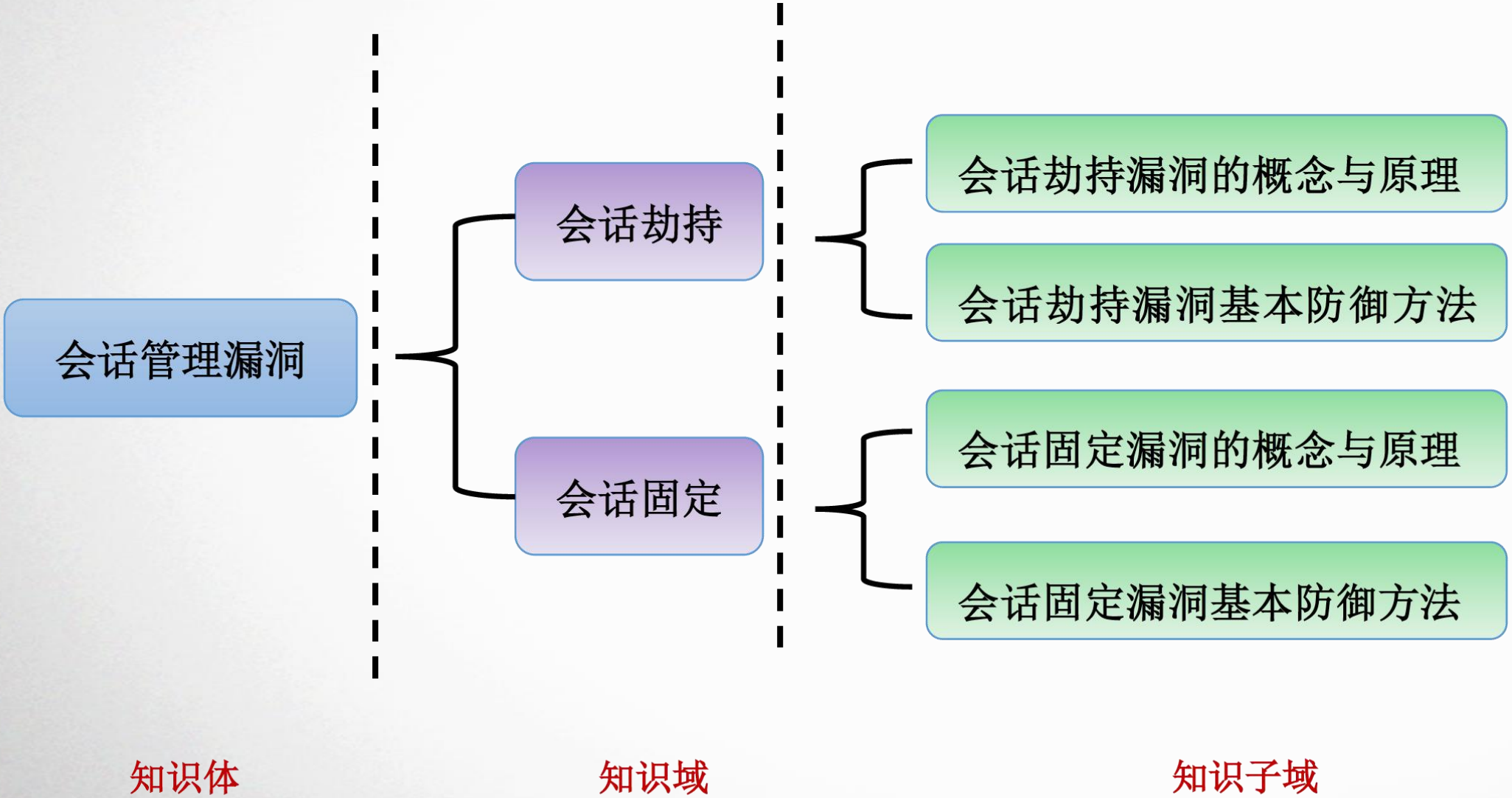
培训教育
Training Services

CISP-PTE

Web 安全基础(7) – 会话管理漏洞

主讲：

会话管理漏洞



会话劫持

会话劫持

- 通过本知识域，我们会：
  - 会话劫持漏的概念与原理
    - 了解什么是会话劫持漏洞
    - 了解会话劫持漏洞的危害
  - 会话劫持漏洞基本防御方法
    - 了解Session机制
    - 了解HttpOnly的设置方法
    - 掌握会话劫持漏洞防御方法

会话劫持漏洞概念

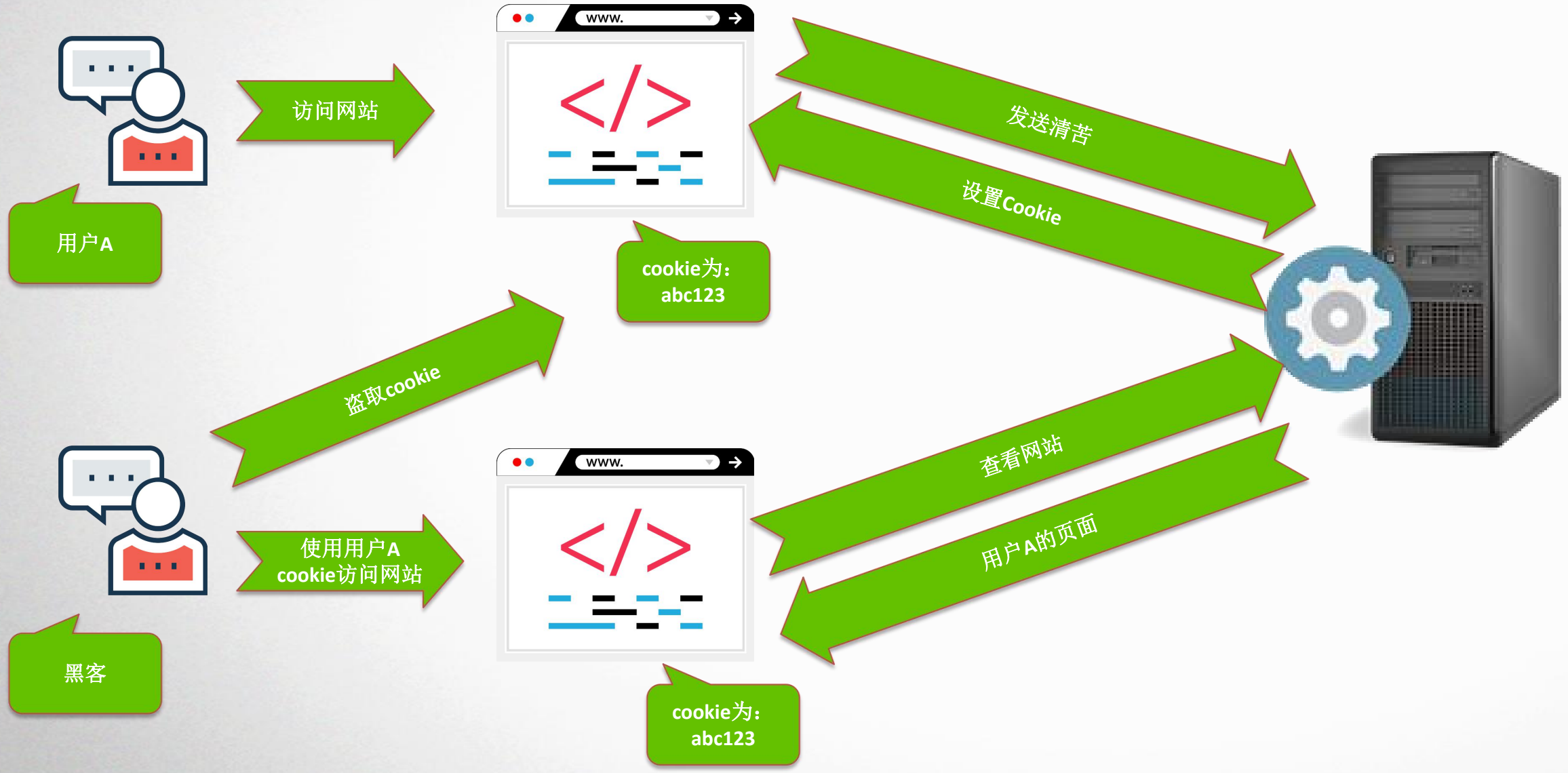
- 会话劫持（Session hijacking），这是一种通过获取用户Session ID后，使用该Session ID登录目标账号的攻击方法，此时攻击者实际上是使用了目标账户的有效Session。会话劫持的第一步是取得一个合法的会话标识来伪装成合法用户。

会话劫持攻击步骤

- 目标用户需要先登录站点
- 登录成功后，该用户会得到站点提供的一个会话标识SessionID
- 攻击者通过某种攻击手段捕获Session ID
- 攻击者通过捕获到的Session ID访问站点即可获得目标用户合法会话

#Session ID一般都设置在cookie

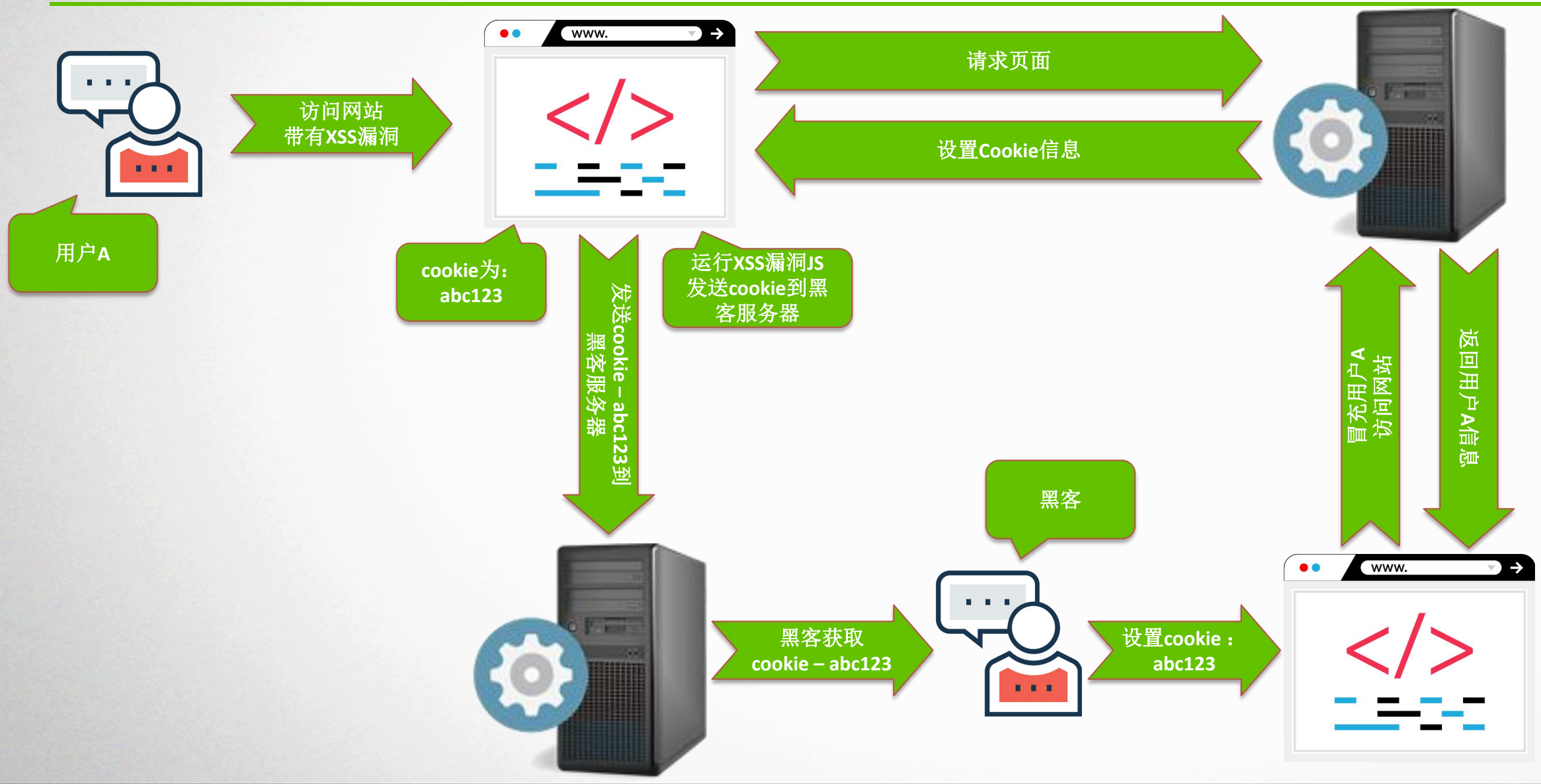
会话劫持漏洞概念图



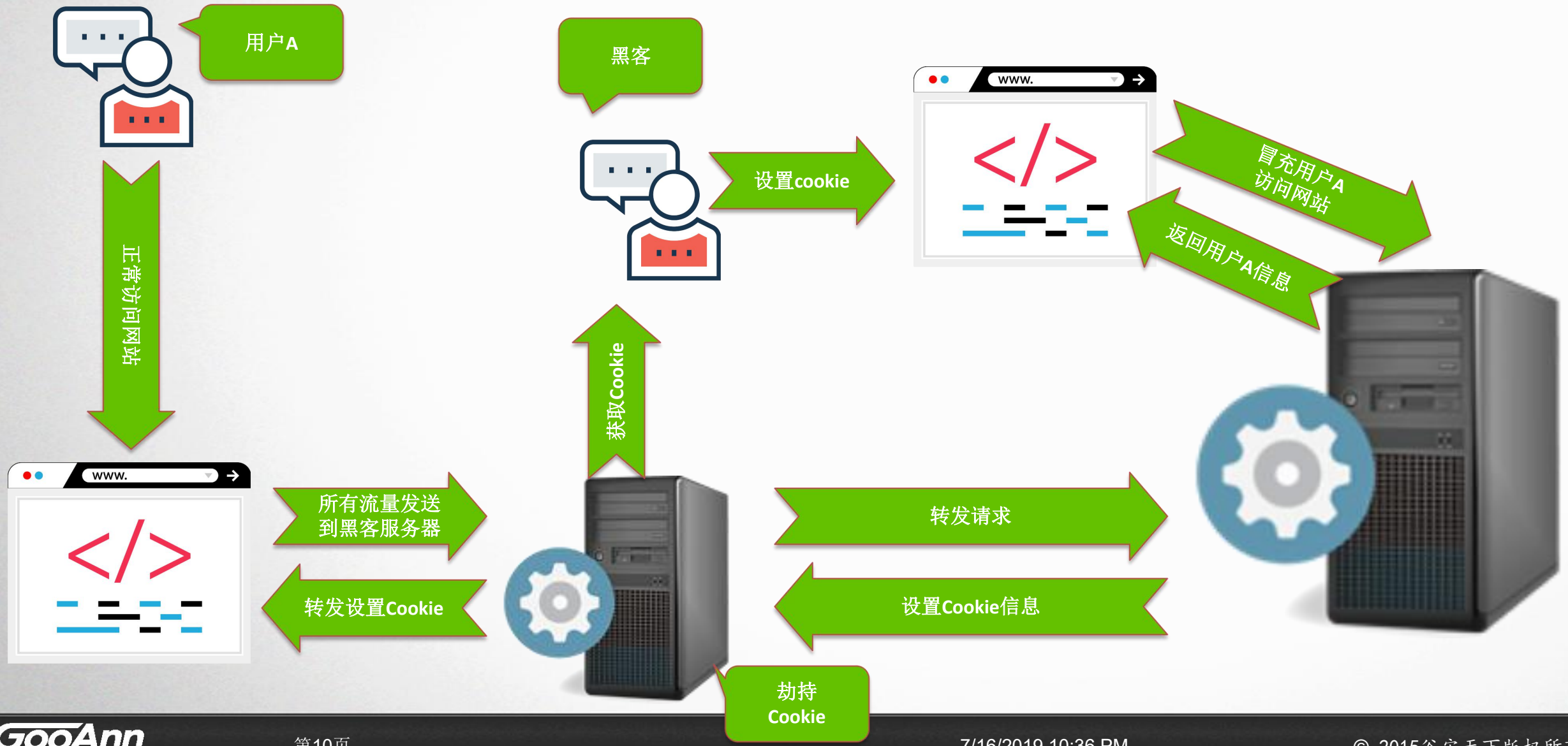
如何获取Cookie

- 了解cookie接口
  - 找到Session ID位置
- 进行破解
  - 暴力破解：尝试各种Session ID，直到破解为止
  - 预测：如果Session ID使用非随机的方式产生，那么就有可能计算出来
  - 窃取：XSS攻击、使用网络嗅探（中间人攻击）等方法获得

劫持cookie - XSS劫持



劫持cookie - 中间人攻击



会话被劫持，会有什么危害？

- 冒充其他人做事情：
  - 被冒充的人的权限越大，可以做的事情越多
  - 更改用户信息
  - 进行转账
  - 购买物品
- 会话被劫持后的一些操作，对网站的影响
  - 因为投诉等原因，会出现信誉下降
  - 客户认为网站本身不可信

Cookie机制

- 在动态网页语言中，某个用户（浏览器）访问（登陆）后，可以一直记录状态。这种状态浏览器使用Cookie来保存。
  - 服务器通过在HTTP的响应头中加上一行特殊的指示以提示浏览器按照指示生成相应的cookie。然而纯粹的客户端脚本如JavaScript或者VBScript也可以生成cookie。
  - 浏览器检查所有存储的cookie，如果某个cookie所声明的作用范围大于等于将要请求的资源所在的位置，则把该cookie附在请求资源的HTTP请求头上发送给服务器。
  - cookie的内容主要包括：名字，值，过期时间，路径和域。
  - 如果不设置过期时间，则表示这个cookie的生命期为浏览器会话期间

理解session机制

- session机制是一种服务器端的机制，服务器使用一种类似于散列表的结构（也可能就是使用散列表）来保存信息
- 当程序需要为某个客户端的请求创建一个session的时候，服务器首先检查这个客户端的请求里是否已包含了一个session标识 -称为session id，如果已包含一个session id则说明以前已经为此客户端创建过session，服务器就按照sessionid把这个session检索出来使用（如果检索不到，可能会新建一个），如果客户端请求不包含sessionid，则为此客户端创建一个session并且生成一个与此session相关联的session id，sessionid的值应该是一个既不会重复，又不容易被找到规律以仿造的字符串，这个session id将被在本次响应中返回给客户端保存。

理解session机制（续）

- 保存这个sessionid的方式可以采用cookie，这样在交互过程中浏览器可以自动的按照规则把这个标识发挥给服务器。
- 由于cookie可以被人为的禁止，必须有其他机制以便在cookie被禁止时仍然能够把sessionid传递回服务器。经常被使用的一种技术叫做URL重写，就是把sessionid直接附加在URL路径的后面，附加方式也有两种，一种是作为URL路径的附加信息，表现形式为：
  - http://...../xxx;jsessionid=ByOK3vjFD75aPnrF7C2HmdnV6QZcEbzWoWiBYEnLerjQ99zWpBng!-145788764
  - http://...../xxx?jsessionid=ByOK3vjFD75aPnrF7C2HmdnV6QZcEbzWoWiBYEnLerjQ99zWpBng!-145788764

什么是HTTP-Only

- 服务端发送cookie的时候，可以设置HTTP-Only
  - Set-Cookie: SESSIONID=abc123; expires=Wednesday, 17-Nov-99 23:12:40 GMT; HttpOnly
- 这个参数的优点是不会被js获取
  - 尝试打开有HTTP-Only cookie设置的网站
  - 查看cookie
- 一般，session-id是用这个来判断
  - 即使有xss漏洞，也获取不了session-id相关的cookie值

什么是secure

- Cookie除了HTTP-Only，还有一个属性：secure
- 当我们设置cookie的某个值secure为True的话：
  - 此cookie只有在HTTPS协议中才会进行传输
  - HTTP协议传输时，是不传输此协议的。

如何防御会话劫持漏洞

- XSS漏洞引起的会话劫持：
  - 可以使用http-only来防止js获取cookie中的sessionid信息
- 会话劫持引起的会话劫持
  - 可以使用HTTP-SSL（https）+secure来保证sessionid不被获取

会话固定

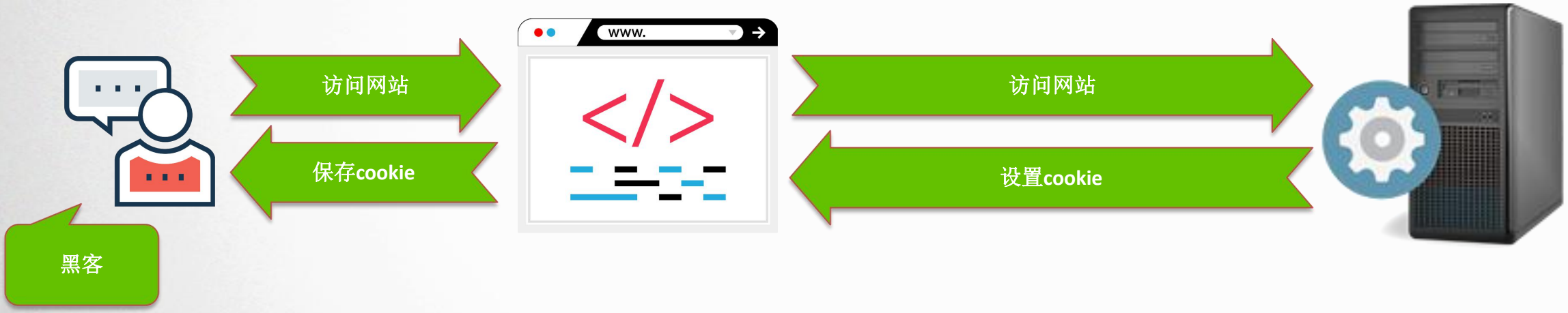
会话固定

- 通过本知识域，我们会：
  - 会话固定漏洞的概念与原理
    - 了解什么是会话固定漏洞
    - 了解会话固定漏洞的检测方法
  - 会话固定漏洞基本防御方法
    - 了解会话固定漏洞的形成的原因
    - 了解会话固定漏洞的风险
    - 掌握会话固定漏洞的防范方法

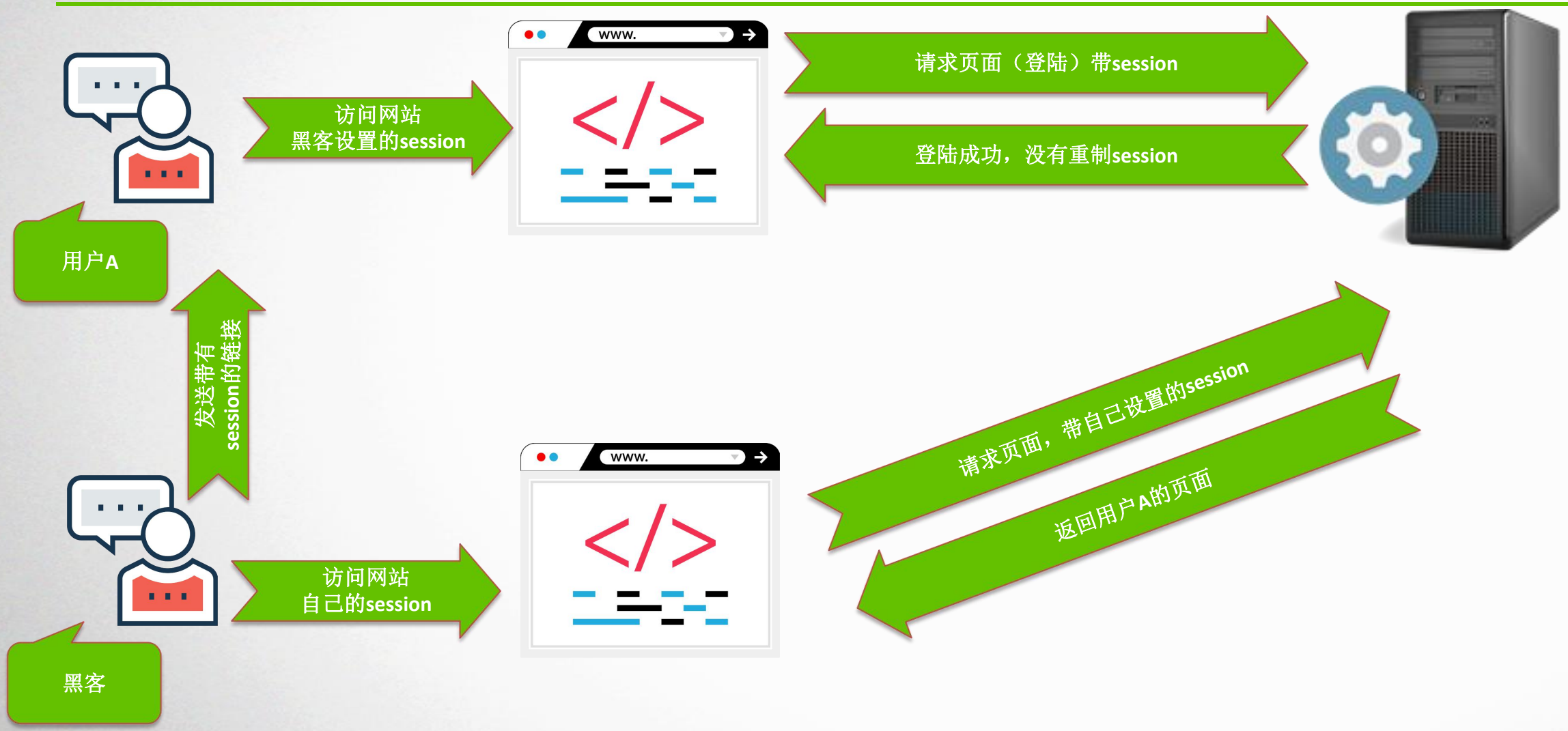
会话固定漏洞的概念

- 会话固定（Session fixation）是一种诱骗受害者使用攻击者指定的会话标识（SessionID）的攻击手段。这是攻击者获取合法会话标识的最简单的方法。会话固定也可以看成是会话劫持的一种类型，原因是会话固定的攻击的主要目的同样是获得目标用户的合法会话，不过会话固定还可以是强迫受害者使用攻击者设定的一个有效会话，以此来获得用户的敏感信息。

会话固定原理图



会话固定原理图（续）



会话固定漏洞的原理

- 访问网站时，网站会设置cookie中的session
- 当用户等候后，cookie中的session保持不变
- 只要获取登陆前的session内容，就可以知道登陆后的session

会话固定漏洞的检测方法

- 访问网站（未登录）
  - 获取cookie信息，获取sessionid
- 登录网站
  - 查看cookie信息，获取sessionid
- 查看登录前，登录后sessionid是否相同

会话固定漏洞的防范方法

- 在用户登录成功后重新创建一个session id
- 登录前的匿名会话强制失效
- session id与浏览器绑定
  - session id与所访问浏览器有变化，就立即重制
- session id与所访问的IP绑定
  - session id与所访问IP有变化，就立即重制