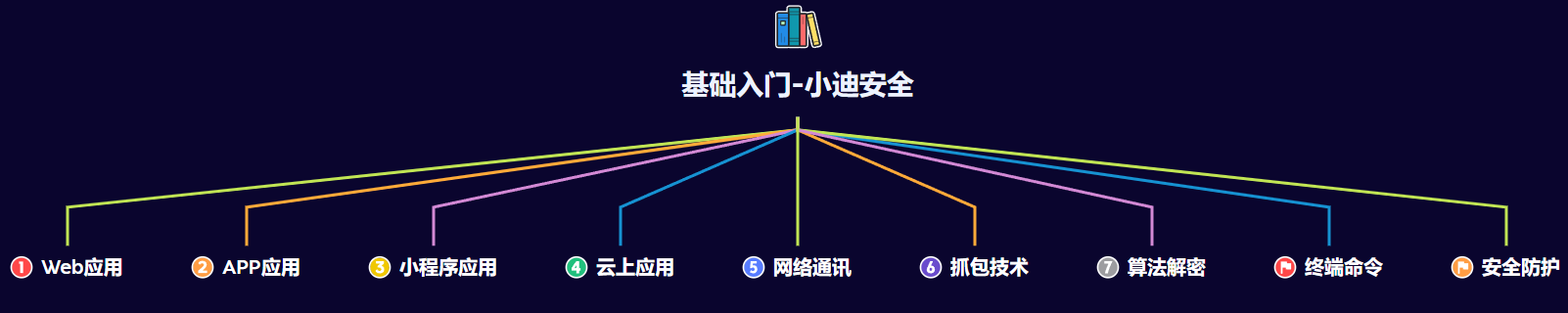
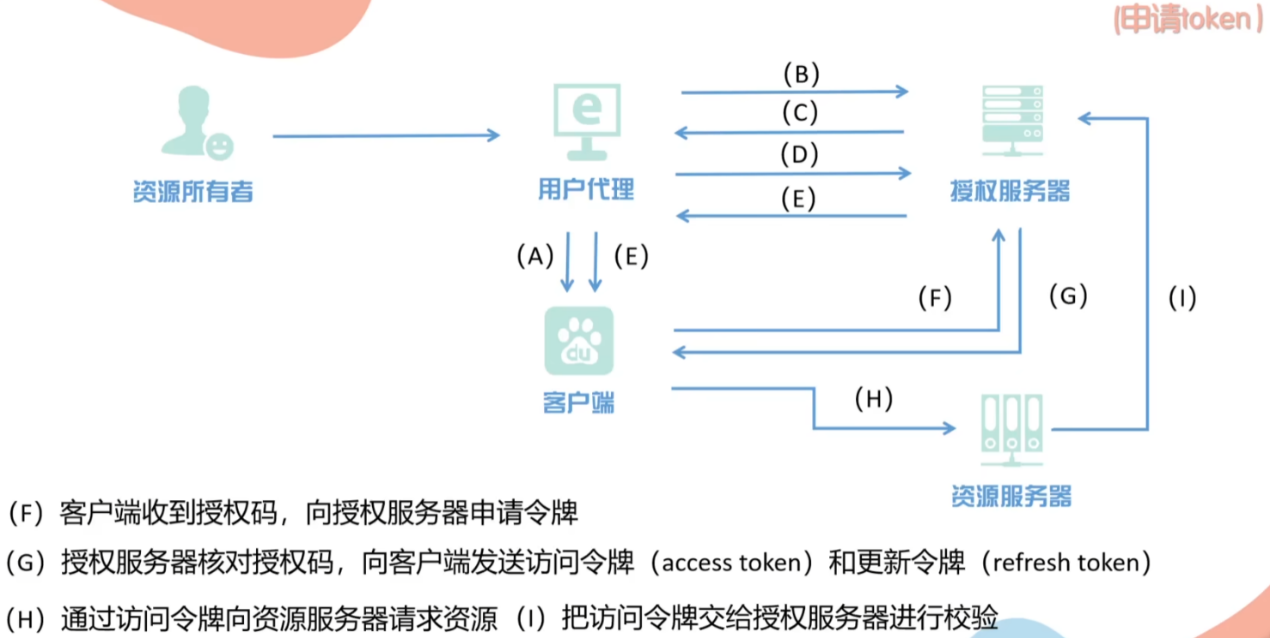
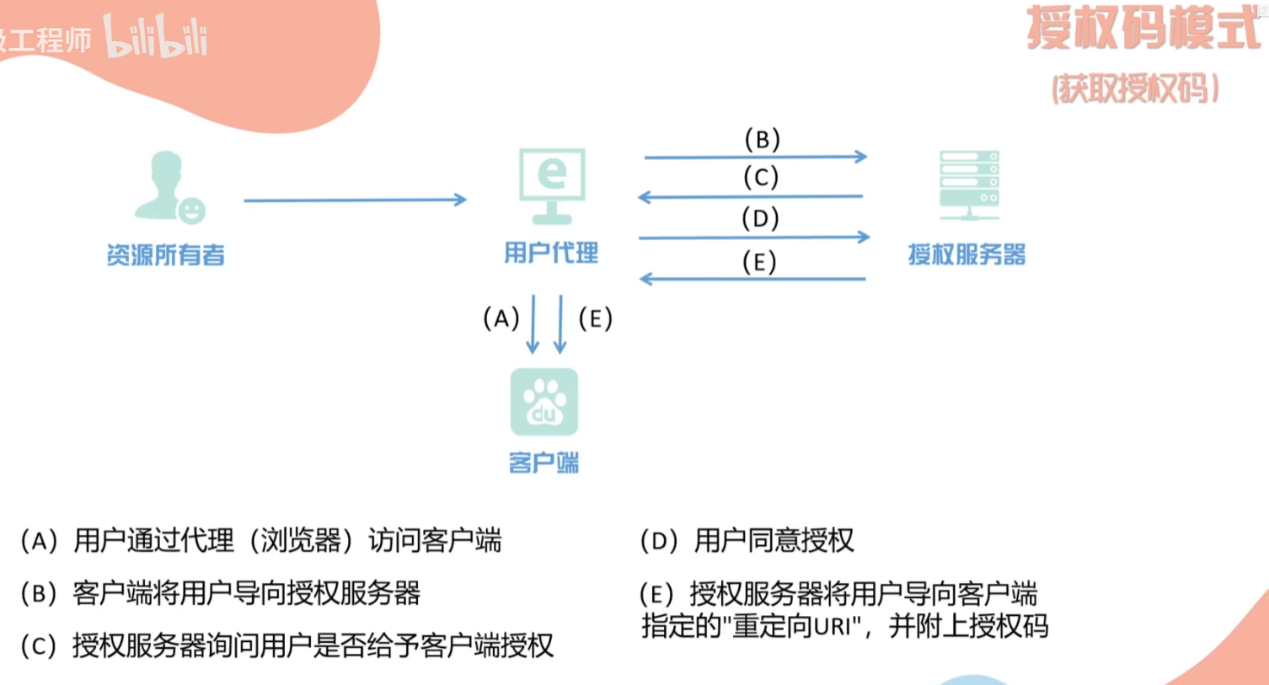
基础入门-网站协议&身份鉴权&OAuth2安全&Token令牌&JWT值&Authirization标头



#知识点：  
1、基础入门-Web应用-域名上的技术要点  
2、基础入门-Web应用-源码上的技术要点  
3、基础入门-Web应用-数据上的技术要点  
4、基础入门-Web应用-解析上的技术要点  
-----------------------------------------------------  
1、基础入门-Web应用-搭建架构上的技术要点  
2、基础入门-Web应用-源码类别上的技术要点  
-----------------------------------------------------  
1、基础入门-Web应用-防护产品-WAF保护  
2、基础入门-Web应用-加速服务-CDN节点  
3、基础入门-Web应用-文件托管-OSS存储  
4、基础入门-Web应用-通讯服务-反向代理  
5、基础入门-Web应用-运维安全-负载均衡  
-----------------------------------------------------  
1、基础入门-Web应用-蜜罐系统  
2、基础入门-Web应用-堡垒机运维  
3、基础入门-Web应用-内外API接口  
4、基础入门-Web应用-第三方拓展架构  
-----------------------------------------------------  
1、基础入门-APP应用-开发架构安全问题  
2、基础入门-小程序应用-开发架构安全问题  
-----------------------------------------------------  
1、抓包技术-Web应用-http/s-Burp&Yakit  
2、抓包技术-APP应用-http/s-Burp&Yakit  
3、抓包技术-PC端应用-http/s-Burp&Yakit  
4、抓包技术-WX小程序-http/s-Burp&Yakit  
5、抓包技术-软件联动-http/s-Proxifier  
6、抓包技术-通用方案-http/s-ReqableApi  
7、抓包技术-其他工具-http/s-Fiddler&Charles  
-----------------------------------------------------  
1、抓包技术-HTTP/S双层代理-扶墙环境  
2、抓包技术-HTTP/S上游下游-项目联动  
3、抓包技术-全局协议-WireShark&科来  
-----------------------------------------------------  
1、数据不回显原因和解决-带外延迟反弹写文件  
2、数据不出网原因和解决-出入站策略正反向连接  
-----------------------------------------------------  
1、反弹Shell-项目&命令&语言等  
2、系统渗透命令-网络&文件&操作等  
-----------------------------------------------------  
1、传输格式&数据-类型&编码&算法  
2、密码存储&混淆-不可逆&非对称性  
-----------------------------------------------------  
0、算法类型-单向散列&对称性&非对称性  
1、算法识别加解密-MD5&AES&DES&RSA  
2、解密条件寻找-逻辑特征&源码中&JS分析  
-----------------------------------------------------  
1、请求头&返回包-方法&头修改&状态码等  
2、数据包分析-红队攻击工具&蓝队流量研判  
3、数据包构造-Reqable自定义添加修改请求  
-----------------------------------------------------  
1、网站协议-http/https安全差异（抓包）  
2、身份鉴权-HTTP头&OAuth2&JWT&Token  
  
#章节点：(待补充)  
Web架构，App架构，小程序架构，前后端分离，容器技术，云产品服务，  
数据加解密算法，数据包抓取，数据包解析，正反向网络通讯，内外网通讯，  
防火墙出入站，Windows&Linux渗透命令，WAF产品，负载均衡，加壳保护等  
  
#具体点：(待补充)  
架构：WEB,APP,小程序,前后端,容器化等  
服务：OSS存储,CDN加速,云数据库,负载均衡等  
网络：不回显,反向代理,防火墙出入站,内外网,正反向连接等  
算法：MD5,Base64,AES,DES,Salt,自定义,代码加密算法等  
命令：Windows&Linux,文件下载,网络查看,反弹权限,用户等  
防护：WAF防护,蜜罐系统,CDN加速,权限设置,加壳加密反调试等

演示案例：

* 网站协议-http&https-安全测试差异性
* 身份鉴权-Authorization&Token&JWT&OAuth

数据包：  
客户端请求Request  
 请求方法  
 请求路径  
 请求头  
服务端返回Response  
 状态码  
  
-方法  
1、常规请求-Get  
2、用户登录-Post  
•get：向特定资源发出请求（请求指定页面信息，并返回实体主体）；  
•post：向指定资源提交数据进行处理请求（提交表单、上传文件），又可能导致新的资源的建立或原有资源的修改；  
•head：与服务器索与get请求一致的相应，响应体不会返回，获取包含在小消息头中的原信息（与get请求类  
似，返回的响应中没有具体内容，用于获取报头）；  
•put：向指定资源位置上上传其最新内容（从客户端向服务器传送的数据取代指定文档的内容），与post的区别是put为幂等，post为非幂等；  
•trace：回显服务器收到的请求，用于测试和诊断。trace是http8种请求方式之中最安全的l  
•delete：请求服务器删除request-URL所标示的资源\*（请求服务器删除页面）  
•option：返回服务器针对特定资源所支持的HTML请求方法 或web服务器发送\*测试服务器功能（允许客户 端查看服务器性能）；  
•connect ： HTTP/1.1协议中能够将连接改为管道方式的代理服务器  
  
-请求头  
见上图  
  
-返回头  
见上图  
  
-Response状态码  
1、数据是否正常  
2、文件是否存在  
3、地址自动跳转  
4、服务提供错误  
注：容错处理识别  
•-1xx:指示信息—表示请求已接收，继续处理。  
•-2xx:成功—表示请求已经被成功接收、理解、接受。  
•-3xx:重定向—要完成请求必须进行更进一步的操作。  
•-4xx:客户端错误—请求有语法错误或请求无法实现。  
•-5xx:服务器端错误—服务器未能实现合法的请求。  
•200 OK：客户端请求成功  
•301 redirect：页面永久性移走，服务器进行重定向跳转；  
•302 redirect：页面暂时性移走，服务器进行重定向跳转，具有被劫持的安全风险；  
•400 BadRequest：由于客户端请求有语法错误，不能被服务器所理解；  
•401 Unauthonzed：请求未经授权。  
•403 Forbidden：服务器收到请求，但是拒绝提供服务。  
•404 NotFound：请求的资源不存在，例如，输入了错误的URL；  
•500 InternalServerError：服务器发生不可预期的错误，无法完成客户端的请求；  
•503 ServiceUnavailable：服务器当前不能够处理客户端的请求  
  
3XX  
1、网站做了容错处理 一旦访问了错误页面将跳转到某个固定地址 3XX  
中间件规则配置  
请求路径或文件 不存在  
  
2、网站代码文件做了跳转 一旦触发就自动跳转到设置的固定地址 3XX  
文件代码配置  
请求路径或文件 存在  
  
200和404误报问题  
网站做了容错处理 一旦访问了错误页面将固定显示某个内容 显示正常  
访问错误 返回的状态码200  
  
#红队案例：  
1、UA头-系统平台  
常见举例影响：  
-比如网站有UA收集整理功能，如果将UA信息记录到数据库的话，这个过程有接受UA信息及写入数据库的操作，这个地方可能存在SQL注入  
-渗透测试习惯  
部分网站做了限制设备访问的需求，手机或电脑只能访问一个，如果我们自己要电脑测试需要更改成手机访问才可以继续  
2、Cookie-用户身份  
3、Post数据-登录爆破  
4、返回状态码-文件探针   
#蓝队案例：  
1、攻击漏洞  
2、安全工具  
  
#Reqable数据包自定义构造

#HTTP/HTTPS  
1. 加密方式  
HTTP：使用明文传输，数据在传输过程中可以被任何人截获和查看。  
HTTPS：通过SSL/TLS协议对数据进行加密，确保数据在传输过程中不被第三方截获和篡改。  
2. 身份验证  
HTTP：不需要进行身份验证，任何人都可以访问网站。  
HTTPS：通过数字证书和SSL/TLS协议验证服务器的身份，防止"中间人攻击"。  
3. 端口号  
HTTP：默认使用80端口。  
HTTPS：默认使用443端口，提供更高的安全性。  
  
#身份验证鉴权技术：  
Cookie，Session，Token，JWT，oauth2等  
参考：https://mp.weixin.qq.com/s/Z6rt\_ggCA8dNVJPgELZ44w  
应用场景：  
Cookie+Session简单，建议在内网使用；  
Token相对完善，推荐在跨域外网使用；  
JWT推荐使用，常用在SSO单点登录中；  
OAuth灵活方便，对于第三方系统登录更友好。  
  
#OAuth2技术  
授权框架，使网站和Web应用程序能够请求对另一个应用上的用户帐户进行有限访问。至关重要的是，OAuth允许用户授予此访问权限，而无需向请求应用程序公开其登录凭据。这意味着用户可以微调他们想要共享的数据，而不必将其帐户的完全控制权移交给第三方。  
  
四种验证模式：  
authorization\_code 授权码模式  
implicit code 简单模式  
password 密码模式  
client\_credentials 客户端模式  
  
授权码模式流程：  
-见上图说明及案例测试  
  
安全漏洞问题：  
1、redirect\_url 校验不严格导致code被劫持到恶意网站（fuzz各种bypass方式）  
2、client\_id与redirect\_url 不一致造成滥用劫持  
3、A应用生成的code可以用在B应用上  
4、state未设置csrf防护，导致csrf风险  
5、scope提权，将低scope权限的code用于高权限场景  
6、HTTP劫持，网络层中间人攻击  
7、点击劫持：通过点击劫持，恶意网站会在以下位置加载目标网站： 透明 iFrame（参见 [ iFrame ]）覆盖在一组虚拟的顶部 精心构造的按钮直接放置在 目标站点上的重要按钮。当用户单击可见的 按钮，他们实际上是在单击一个按钮（例如“授权” 按钮）在隐藏页面上。  
  
#Authorization头：  
参考：https://juejin.cn/post/7300812626279251987  
授权方案：  
1、Basic认证  
2、Digest认证  
3、Bearer认证  
4、JWT认证  
5、API密钥认证  
6、双因素认证  
7、其他一些认证方式  
安全漏洞问题：  
JWT攻防，Token劫持等  
  
安全影响：  
1、通过以上学习了解到http/https差异（抓包）  
2、身份鉴权测试中要区别oauth,jwt,token差异

涉及资源：[资源下载地址](https://docs.qq.com/doc/DQ3Z6RkNpaUtMcEFr)