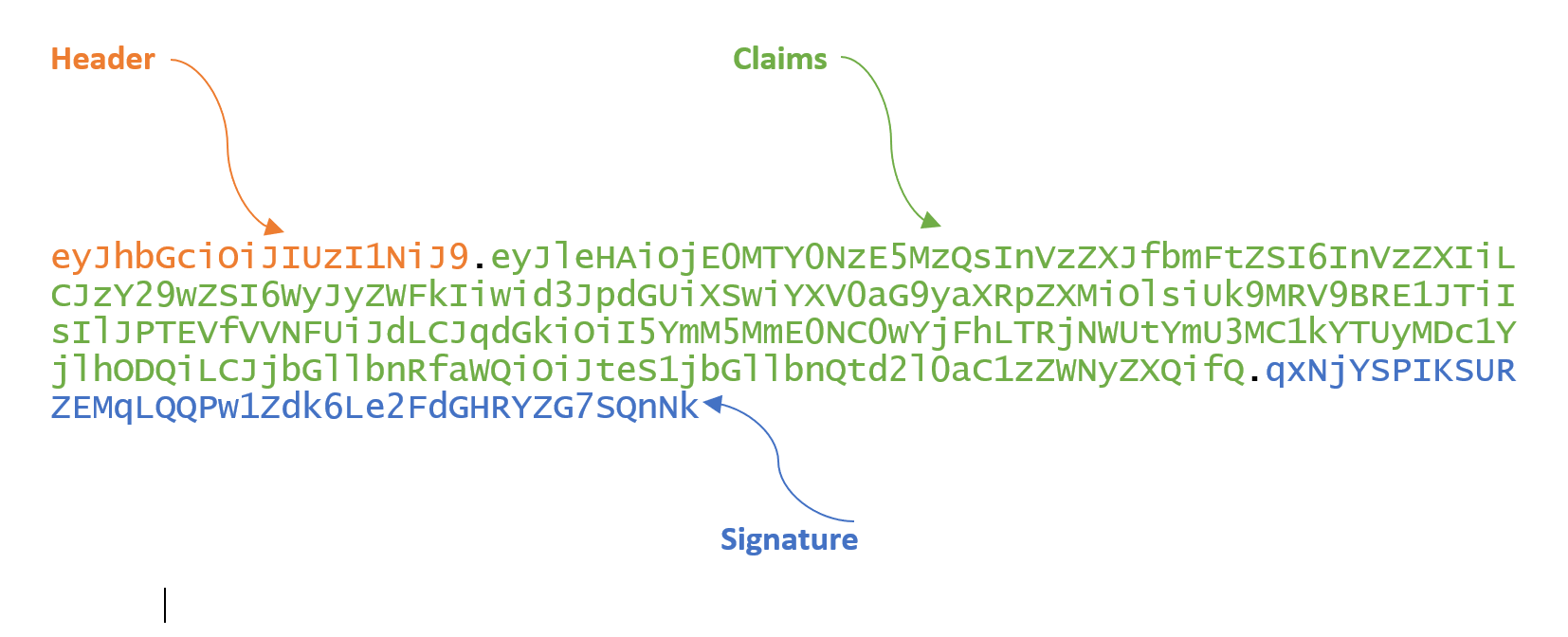
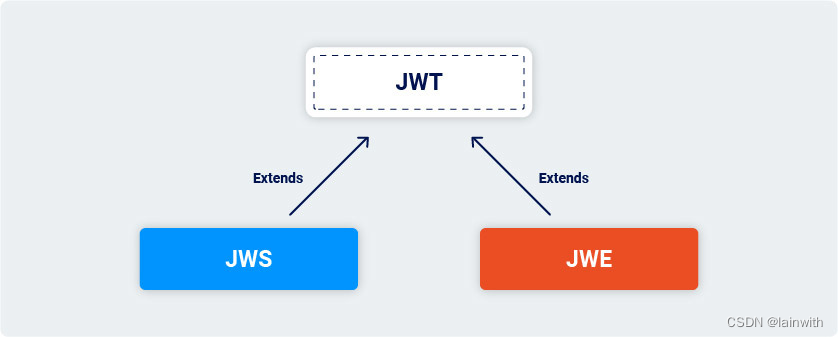
Web攻防-身份验证篇&JWT令牌&空密钥&未签名&密钥爆破&JWK&JWU&KID&算法替换

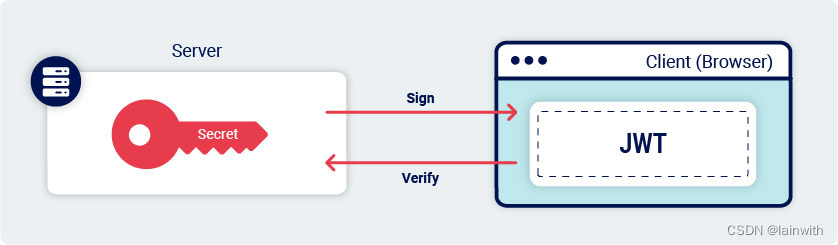


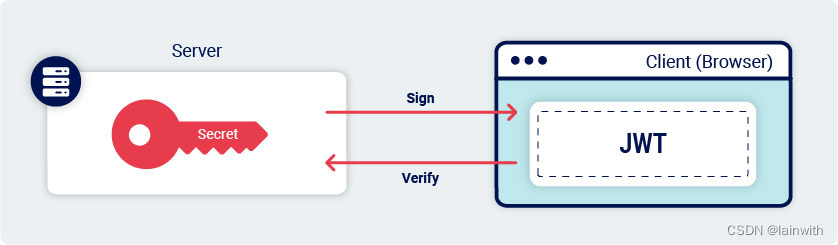
#知识点：  
1、Web攻防-SQL注入-产生原理&应用因素  
2、Web攻防-SQL注入-各类数据库类型利用  
----------------------------------------------  
1、Web攻防-SQL注入-参数类型\*参数格式  
2、Web攻防-SQL注入-XML&JSON&BASE64等  
3、Web攻防-SQL注入-数字字符搜索等符号绕过  
----------------------------------------------  
1、Web攻防-SQL注入-操作方法&增删改查  
2、Web攻防-SQL注入-HTTP头&UA&Cookie  
3、Web攻防-SQL注入-HTTP头&XFF&Referer  
----------------------------------------------  
1、Web攻防-SQL注入-操作方法&增删改查  
2、Web攻防-SQL注入-布尔&延时&报错&盲注  
----------------------------------------------  
1、Web攻防-SQL注入-高权限用户差异  
2、Web攻防-SQL注入-跨库&文件读写带外  
----------------------------------------------  
1、Web攻防-SQL注入-堆叠&二次注入  
2、Web攻防-SQL注入-SQLMAP进阶使用  
----------------------------------------------  
1、Web攻防-XSS跨站-原理&分类&利用点  
2、Web攻防-XSS跨站-反射&存储&DOM型  
----------------------------------------------  
1、Web攻防-XSS跨站-文件类型-html&pdf&swf&svg  
2、Web攻防-XSS跨站-功能逻辑-postMessage&localStorage  
----------------------------------------------  
1、Web攻防-XSS跨站-浏览器&转换-UXSS&MXSS  
2、Web攻防-XSS跨站-框架和库-VUE&React&Electron&JQuery  
----------------------------------------------  
1、Web攻防-XSS跨站-手工代码&框架工具&在线平台  
2、Web攻防-XSS跨站-Cookie盗取&数据提交&网络钓鱼  
----------------------------------------------  
1、Web攻防-XSS跨站-安全防护&CSP&Httponly&WAF等  
2、Web攻防-XSS跨站-工具项目&XSStrike&Chypass\_pro  
----------------------------------------------  
1、Web攻防-CSRF-原理&检测&利用&防御  
2、Web攻防-CSRF-防御-Referer策略隐患  
3、Web攻防-CSRF-防御-Token校验策略隐患  
----------------------------------------------  
1、WEB攻防-SSRF利用绕过-伪协议&IP及域名  
2、WEB攻防-SSRF挖掘思路-功能逻辑&SRC复盘  
----------------------------------------------  
1、WEB攻防-SSRF插件结合-SRC复盘与实战  
----------------------------------------------  
1、WEB攻防-文件上传-前端&黑白名单&MIME&文件头等  
2、WEB攻防-文件上传-执行权限&解码还原&云存储&分站等  
----------------------------------------------  
1、WEB攻防-文件上传-思维导图形成  
2、WEB攻防-文件上传-JS提取&特定漏洞&第三方编辑器  
----------------------------------------------  
1、WEB攻防-文件包含-LFI&RFI利用思路  
2、WEB攻防-文件上传-伪协议玩法&无文件  
3、WEB攻防-文件上传-黑白盒案例&SRC复盘  
----------------------------------------------  
1、WEB攻防-文件下载&读取&删除-功能点&URL  
2、WEB攻防-目录遍历&泄露\*穿越-功能点&URL  
----------------------------------------------  
1、WEB攻防-XML&XXE-注入原理&四大分类  
2、WEB攻防-XML&XXE-文件读取&SSRF&实体引用  
3、WEB攻防-XML&XXE-无回显&升级拓展&挖掘思路  
----------------------------------------------  
1、WEB攻防-XML&XXE-黑盒功能点挖掘  
2、WEB攻防-XML&XXE-白盒函数点挖掘  
----------------------------------------------  
1、WEB攻防-RCE-代码执行&命令执行  
2、WEB攻防-RCE-黑盒功能点&数据包  
3、WEB攻防-RCE-白盒函数方法&组件  
----------------------------------------------  
1、WEB攻防-SSTI-利用分类&功能点  
2、WEB攻防-SSTI-利用项目&挖掘思路  
----------------------------------------------  
1、WEB攻防-PHP反序列化-魔术方法&触发规则  
2、WEB攻防-PHP反序列化-POP链构造&黑白盒  
----------------------------------------------  
1、WEB攻防-PHP反序列化-原生类&生成及利用条件  
2、WEB攻防-PHP反序列化-Exception触发XSS  
3、WEB攻防-PHP反序列化-SoapClient触发SSRF  
4、WEB攻防-PHP反序列化-SimpleXMLElement触发XXE  
----------------------------------------------  
1、WEB攻防-PHP反序列化-CVE&wakeup绕过  
2、WEB攻防-PHP反序列化-版本属性解析差异  
3、WEB攻防-PHP反序列化-字符增多减少逃逸  
----------------------------------------------  
1、WEB攻防-业务逻辑-越权&访问控制&未授权  
2、WEB攻防-业务逻辑-检测项目&挖掘思路报告  
----------------------------------------------  
1、WEB攻防-JWT令牌-组成&识别&检测&安全  
2、WEB攻防-JWT令牌-空算法&未签名&爆破密钥  
3、WEB攻防-JWT令牌-JWK&JWU&KID&加密替换  
   
#章节点：(待补充)  
OWTOP10，SQL注入，文件安全（上传,下载,读取,删除,包含等），目录遍历，  
CSRF，SSRF，CRLF，SSTI注入，XML&XXE安全，RCE执行（代码或命令等）  
反序列化，业务逻辑（验证码，接口枚举，支付购买，机制验证，越权，并发等）  
未授权访问，失效访问控制，弱口令安全，第三方组件安全，CORS，JSONP等

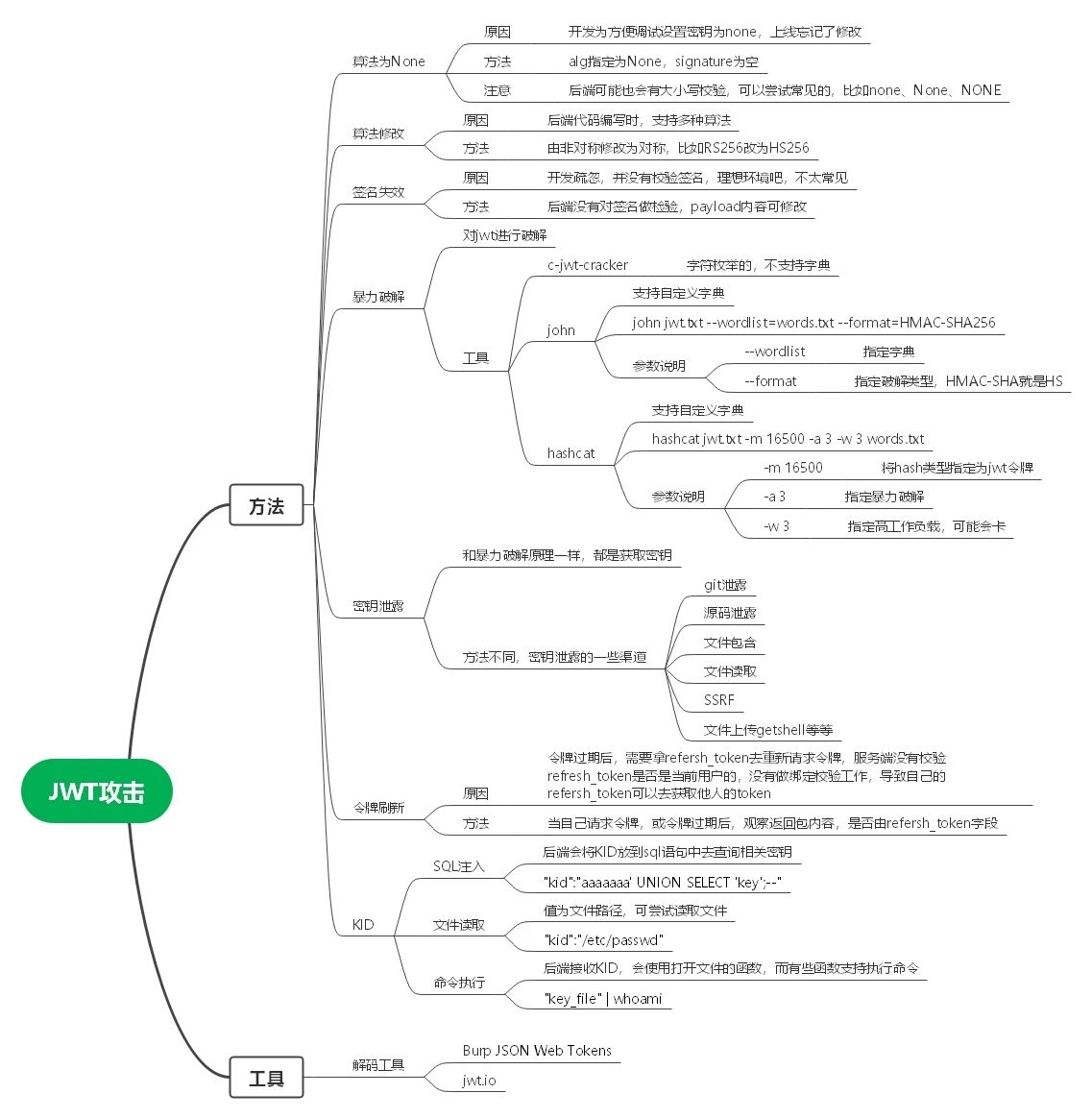












演示案例：

* WEB攻防-JWT令牌-组成&识别&检测&安全
* WEB攻防-JWT令牌-空算法&未签名&爆破密钥
* WEB攻防-JWT令牌-JWK&JWU&KID&加密替换

#JWT  
JSON Web Token(JWT)。它遵循JSON格式，将用户信息加密到token里，服务器不保存任何用户信息，只保存密钥信息，通过使用特定加密算法验证token，通过token验证用户身份。基于token的身份验证可以替代传统的cookie+session身份验证方法。这使得JWT成为高度分布式网站的热门选择，在这些网站中，用户需要与多个后端服务器无缝交互。  
   
-JWT识别  
1、标头（Header）  
Header是JWT的第一个部分，是一个JSON对象，主要声明了JWT的签名算法，如"HS256”、"RS256"等，以及其他可选参数，如"kid"、"jku"、"x5u"等  
alg字段通常用于表示加密采用的算法。如"HS256"、"RS256"等  
typ字段通常用于表示类型  
还有一些其他可选参数，如"kid"、"jku"、"jwk"等  
   
2、有效载荷（Payload）  
Payload是JWT的第二个部分，这是一个JSON对象，主要承载了各种声明并传递明文数据，用于存储用户的信息，如id、用户名、角色、令牌生成时间和其他自定义声明。  
iss：该字段表示jwt的签发者。  
sub：该jwt面向的用户。  
aud：jwt的接收方。  
exp：jwt的过期时间,通常来说是一个时间戳。  
iat：jwt的签发时间,常来说是一个时间戳。  
jti：此jwt的唯一标识。通常用于解决请求中的重放攻击。该字段在大多数地方没有被提及或使用。因为使用此字段就意味着必须要在服务器维护一张jti表， 当客户端携带jwt访问的时候需要在jti表中查找这个唯一标识是否被使用过。使用这种方式防止重放攻击似乎让jwt有点怪怪的感觉, 毕竟jwt所宣称的优点就是无状态访问  
   
3、签名（Signature）  
Signature是对Header和Payload进行签名，具体是用什么加密方式写在Header的alg 中。同时拥有该部分的JWT被称为JWS，也就是签了名的JWT。  
对Header和Payload进行签名，具体是用什么加密方式写在Header的alg中。  
同时拥有该部分的JWT被称为JWS，也就是签了名的JWT。  
   
第一部分：对 JSON 的头部做 base64 编码处理得到  
第二部分：对 JSON 类型的 payload 做 base64 编码处理得到  
第三部分：分别对头部和载荷做base64编码，并使用.拼接起来  
使用头部声明的加密方式，对base64编码前两部分合并的结果加盐加密处理，作为JWT  
   
HS256：对称性加密（同一的密钥加密和解密） 可以采用爆破密钥攻击  
RS256：非对称性加密（公钥和私钥，私钥解密或加密，公钥加密或解密）  
   
识别检测利用项目：  
BURP插件：Hae&JSON Web Tokens&JWT Editor  
https://jwt.io/  
https://github.com/z-bool/Venom-JWT  
https://github.com/ticarpi/jwt\_tool  
https://github.com/wallarm/jwt-secrets  
https://github.com/CompassSecurity/jwt-scanner  
   
   
参考：blog.csdn.net/weixin\_44288604/article/details/128562796  
1、实验室：通过未验证的签名绕过JWT身份验证  
2、实验室：通过有缺陷的签名验证绕过JWT身份验证  
3、实验室：通过弱签名密钥绕过JWT身份验证  
4、实验室：通过jwk标头注入绕过JWT身份验证  
5、实验室：通过jku标头注入绕过JWT身份验证  
6、实验室：通过kid头路径遍历绕过JWT身份验证  
   
7、算法混淆   
-Web349（公钥私钥泄露）  
pip install PyJWT==1.7.1  
import jwt  
public = open('private.key', 'r').read()  
payload={"user":"admin"}  
print(jwt.encode(payload, key=public, algorithm='RS256'))  
   
-Web350(密钥混淆攻击RS256=>HS256)  
将RS256算法改为HS256（非对称密码算法=>对称密码算法）  
HS256算法使用密钥为所有消息进行签名和验证。  
而RS256算法则使用私钥对消息进行签名并使用公钥进行身份验证。  
var jwt = require('jsonwebtoken');  
var fs = require('fs');  
var privateKey = fs.readFileSync('./public.key');  
var token = jwt.sign({ user: 'admin' }, privateKey, { algorithm: 'HS256' });  
console.log(token)  
   
8、某个测试目标  
参考：https://mp.weixin.qq.com/s/obiU3BaFoZ7272z2vS0QgQ  
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFkbWluIiwiZXhwIjo0MDcyNjE1MzE0fQ.mwI2P1j8CIvhxBKFvcyU7TNLBeuFtiUM1mPrKanF1w4  
Content-Type: application/x-www-form-urlencoded  
   
db=sample&q=show users  
   
复盘：  
https://mp.weixin.qq.com/s/ITVFuQpA8OCIRj4wW-peAA  
https://mp.weixin.qq.com/s/xuY1oTwFcM1pyiql0U3NPQ  
https://mp.weixin.qq.com/s/AVW8DsnLiviopeJYQYKC3A  
https://mp.weixin.qq.com/s/st0xma6KoRbo1NUp9rtZhw  
https://mp.weixin.qq.com/s/9OL5jZK7S1MiEUb8Q\_F1Pw  
   
首先找到需要JWT鉴权后才能访问的页面，如个人资料页面，将请求重放测试：  
1）未授权访问：删除Token后仍然可以正常响应对应页面  
2）敏感信息泄露：通过JWt.io解密出Payload后查看其中是否包含敏感信息，如弱加密的密码等  
3）破解密钥+越权访问：通过JWT.io解密出Payload部分内容，通过空加密算法或密钥爆破等方式实现重新签发Token并修改Payload部分内容，重放请求包，观察响应包是否能够越权查看其他用户资料  
4）检查Token时效性：解密查看payload中是否有exp字段键值对（Token过期时间），等待过期时间后再次使用该Token发送请求，若正常响应则存在Token不过期  
5）通过页面回显进行探测：如修改Payload中键值对后页面报错信息是否存在注入，payload中kid字段的目录遍历问题与sql注入问题

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