

# 加密算法&cdn&信息收集 (6-10)

## P6. 加密算法

### 6.1 常见加密编码等算法解析

- MD5, SHA, ASC, 进制, 时间戳, URL, BASE64, Unescape, AES, DES 等

### 6.2 常见加密形式算法解析

- 直接加密, 带 salt, 带密码, 带偏移, 带位数, 带模式, 带干扰, 自定义组合等

### 6.3 常见解密方式

- 枚举, 自定义逆向算法, 可逆向

### 6.4 了解常规加密算法的特性

- 长度位数, 字符规律, 代码分析, 搜索获取等

### 6.5 工具

链接: <https://pan.baidu.com/s/1A5Jjc-spYBKIKiPNbaetbQ> 提取码: 8eh7

<http://tool.chacuo.net/cryptaes>



#### 自定义加密算法

```
<?php
function encrypt($data, $key)
{
    $key = md5('ISCC');
    #print $key;
```

```

$x = 0;
$len = strlen($data);
$klen = strlen ($key);
#print $len;
for($i=0; $i < $len; $i++){
    if($x == $klen)
        $x = 0 ;
    $char .= $key[$x];
    $x += 1;
    #print $key[$x];
}
#print $char[0].$char[1].$char[2];
for ($i=0;$i<$len; $i++){
    $str .= chr ((ord ($data[$i]) + ord ($char[$i]))%128);
}
return base64_encode ($str);
}
echo encrypt('helloworld');
?>

```

## 自定义解密算法

```

<?php
function decrypt($str){
    $mkey = md5('ISCC');
    $klen = strlen($mkey);
    $tmp = $str;
    $tmp = base64_decode($tmp); //解密base64
    $md_len = strlen($tmp); //获取输入加密字符长度
    $x = 0;
    $char = ""; //临时数组

    for ($i=0; $i<$md_len; $i++){
        if ($x == $klen){ //当加密字符串长度超出key
            $x = 0; //的长度时查重头开始和获取
        }
        $char .= $mkey[$x]; // .= 累积函数
        $x += 1;
    }

    $md_data = array(); //获取加密字符中的ASCII数据
    for ($i=0; $i<$md_len; $i++){
        array_push($md_data,ord($tmp[$i]));
    }

    $md_data_source = array();
    $data1 = "";
    $data2 = "";
    foreach ($md_data as $key => $value){ //最终还原
        $i = $key;
        if($i >= strlen($mkey)){
            $i = $i - strlen($mkey);
        }
        $dd = $value;
        $od = ord($mkey[$i]);
        array_push($md_data_source,$dd);
        $data1 .=chr(($dd+128)-$od); //原数据加key的Ascii大于128
    }
}

```

```

        $data2 .=chr($dd-$od);
    }

    print "data1 =>".$data1."<br>\n";
    print "data2 =>".$data2."<br>\n";
}

$str = 'HxclIiEqiUY';
decrypt($str);
?>

```

//原数据加key的Ascii小于128

带盐加密: md5(md5(\$pass),\$salt))

## AES加密

```

require_once '../config.php';
//解密过程
function decode($data) {
    $td = mcrypt_module_open(MCRYPT_RIJNDAEL_128, '', MCRYPT_MODE_CBC, '');
    mcrypt_generic_init($td, 'YdhaqPQnexoaDuW3', '2018201920202021');
    $data = mdecrypt_generic($td, base64_decode(base64_decode($data)));
    mcrypt_generic_deinit($td);
    mcrypt_module_close($td);
    if(substr(trim($data), -6) != '_mozhe') {
        echo '<script>>window.location.href="/index.php";</script>';
    } else {
        return substr(trim($data), 0, strlen(trim($data))-6);
    }
}

```

加密位数: 128  
jiamimoshi  
盐: 'YdhaqPQnexoaDuW3'  
偏移量: '2018201920202021'

浏览器地址: tool.chacuo.net/cryptaes

### 加密解密工具

对称性加密解密

- DES加密解密
- 3DES加密解密
- AES加密解密
- chacha20加密解密
- RC2加密解密
- RC4加密解密
- RC5加密解密
- RC6加密解密
- Blowfish加密解密
- Twofish加密解密
- Serpent加密解密
- Gost加密解密
- Rijndael加密解密
- Cast加密解密
- Xtea加密解密

非对称性加密解密

- rsa公钥加密解密
- rsa私钥加密解密
- rsa密钥对
- rsa私钥密码清除
- rsa私钥密码修改
- PKCS#1转

### 在线AES加密解密、AES在线加密解密、AES encryption and decryption

AES,高级加密标准 (英语: Advanced Encryption Standard, 缩写: AES), 在密码学中又称Rijndael加密法, 是美国联邦政府采用的一种区块加密标准。这个标准用来替代原先的DES, 已经被多方分析且广为全世界所使用。严格地说, AES和Rijndael加密法并不完全一样 (虽然在实际应用中二者可以互换), 因为Rijndael加密法可以支持更大范围的区块和密钥长度: AES的区块长度固定为128比特, 密钥长度则可以是128, 192或256比特; 而Rijndael使用的密钥和区块长度可以是32位的整数倍, 以128位为下限, 256比特为上限。包括AES-ECB,AES-CBC,AES-CTR,AES-OFB,AES-CFB

华纳云 香港服务器减400 香港云18/月 全球服务器支持虚拟币 免费试用可天付

AES加密模式: ECB 填充: zeropadding 数据块: 128位 密码: 请输入密码! 偏移量: iv偏移量, ecb模式 输出: base64 字符集: gb2312编码 (简体)

待加密、解密的文本: 欢迎来到chacuo.net

↑ 将您电脑文件直接拖入试试 ^\_^

AES加密 AES解密

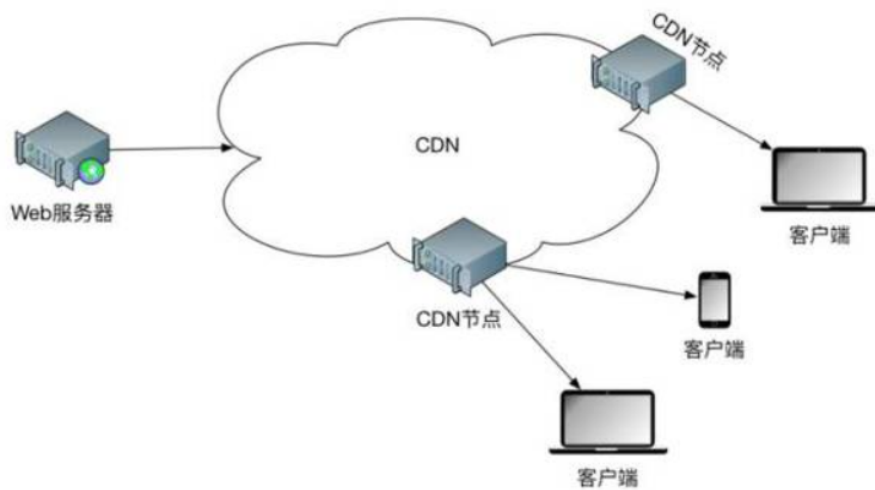
AES加密、解密转换结果(base64了)

## 资源

<https://www.mozhe.cn>  
<https://www.cmd5.com>  
<http://tool.chacuo.net/cryptaes>  
<https://ctf.bugku.com/challenges>

## P7. DNS 绕过

CDN的全称是Content Delivery Network,即内容分发网络。CDN是构建在现有网络基础之上的智能虚拟网络, 依靠部署在各地的边缘服务器, 通过中心平台的负载均衡、内容分发、调度等功能模块, 使用户就近获取所需内容, 降低网络拥塞, 提高用户访问响应速度和命中率。但在安全测试过程中, 若目标存在CDN服务, 将会影响到后续的安全测试过程。



## 7.1 判断是否有CDN

### 看响应时间

工具: <http://tool.chinaz.com/speedtest>



ping ip或域名, 看是否会出现变化  
无

```
C:\Users\Master>ping xiaodi8.com
```

```
正在 Ping xiaodi8.com [47.75.212.155] 具有 32 字节的数据:  
来自 47.75.212.155 的回复: 字节=32 时间=60ms TTL=108  
来自 47.75.212.155 的回复: 字节=32 时间=59ms TTL=108  
来自 47.75.212.155 的回复: 字节=32 时间=58ms TTL=108  
来自 47.75.212.155 的回复: 字节=32 时间=55ms TTL=108
```

```
47.75.212.155 的 Ping 统计信息:
```

```
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),  
往返行程的估计时间(以毫秒为单位):  
最短 = 55ms, 最长 = 60ms, 平均 = 58ms
```

有

```
C:\Users\Master>ping www.bilibili.com
```

```
正在 Ping a.w.bilicdn1.com [123.159.205.1] 具有 32 字节的数据:  
来自 123.159.205.1 的回复: 字节=32 时间=38ms TTL=51  
来自 123.159.205.1 的回复: 字节=32 时间=92ms TTL=51  
来自 123.159.205.1 的回复: 字节=32 时间=33ms TTL=51  
来自 123.159.205.1 的回复: 字节=32 时间=32ms TTL=51
```

```
123.159.205.1 的 Ping 统计信息:
```

```
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),  
往返行程的估计时间(以毫秒为单位):  
最短 = 32ms, 最长 = 92ms, 平均 = 48ms
```

nslookup 域名, 看是否会有很多节点

```
C:\Users\Master>nslookup bilibili.com
```

```
服务器: localhost
```

```
Address: 192.168.43.1
```

```
非权威应答:
```

```
名称: bilibili.com
```

```
Addresses: 119.3.238.64
```

```
120.92.174.135
```

```
120.92.78.97
```

```
110.43.34.66
```

```
119.3.70.188
```

```
139.159.241.37
```

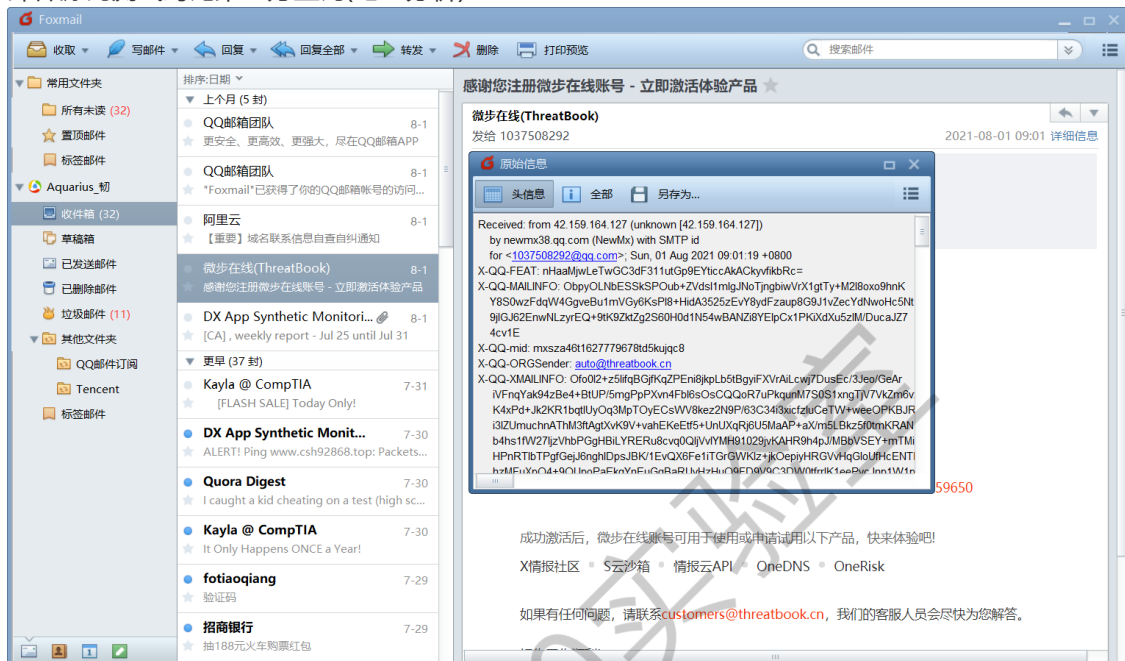
## 7.2 CDN对测试有何影响&如何绕过

1. 子域名查询: 有的网站主域名会做CDN, 但是子域名可能不会做
2. 邮件服务查询: 我们访问别人, 可能通过CND, 但别人访问我们通常不会走CDN
3. 国外地址请求: 国外没有cdn节点的话, 可能直接走原ip
4. 遗留文件, 扫描全网
5. 黑暗引擎搜索特定文件
6. dns历史记录, 以量打量: CDN节点是有流量上限的, 用光之后就会直通原机, 这也是一种流量攻击

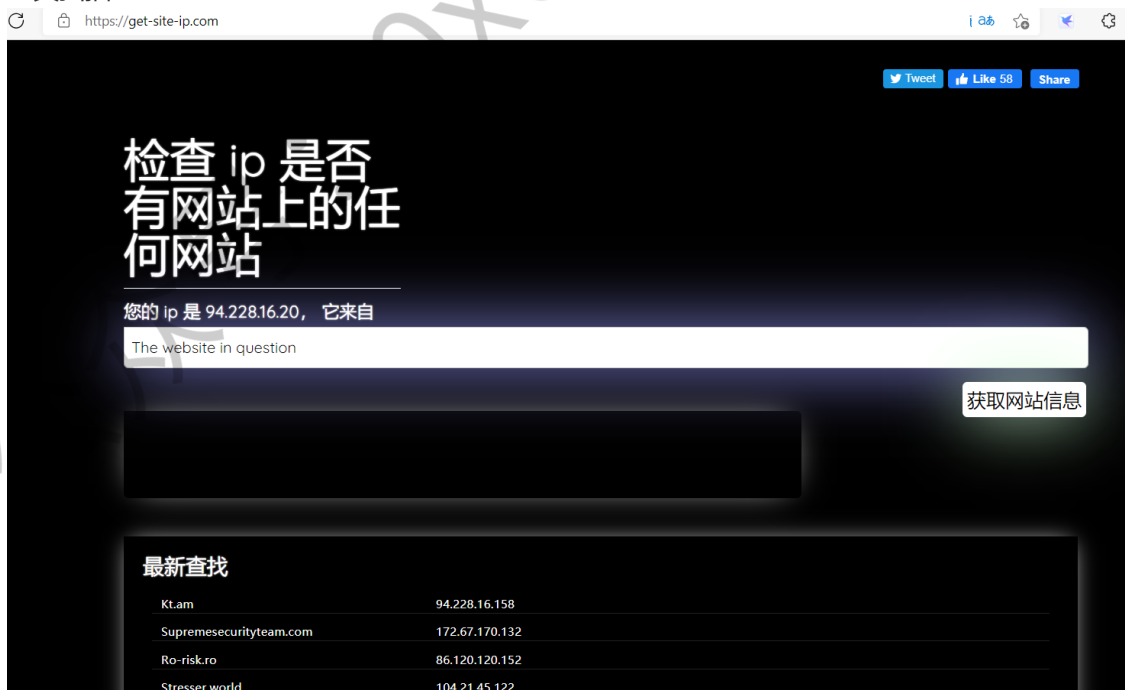
## 7.3 测试

### 子域名上面的小技巧

- 二级域名和三级域名查到的结果可能不一样
- 主域名和子域名查询到的可能不一样
- DNS历史记录=第三方接口(接口查询)
- 采集/国外请求(同类型访问)
- 邮件源码测试对比第三方查询(地区分析)



- 黑暗引擎(shodan搜指定hash文件)
- 扫全网 fuckcdn, w8 fuckcdn, zmap等
- 工具扫描



- 认为判定, 根据网站的域名备案推测
- 本地清下dns, 然后hosts里写上得到的ip和域名, 如果是cdn可能会出现刷新异常, 如果打开很快大概率是原机

## 资源

<https://www.shodan.io>

<https://x.threatbook.cn>

<http://ping.chinaz.com>

<https://www.get-site-ip.com/>

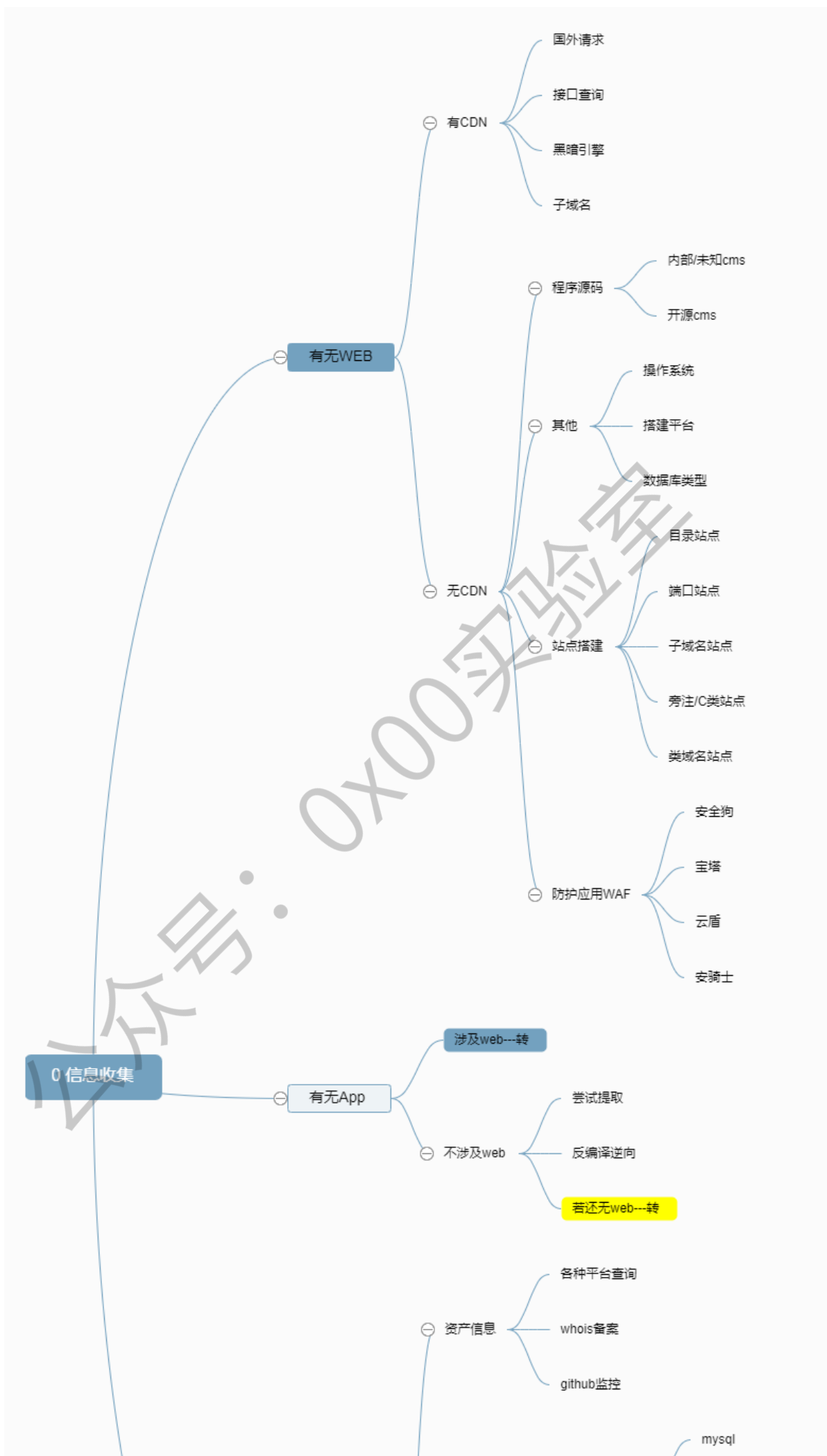
<https://asm.ca.com/en/ping.php>

<https://github.com/boy-hack/w8fuckcdn>

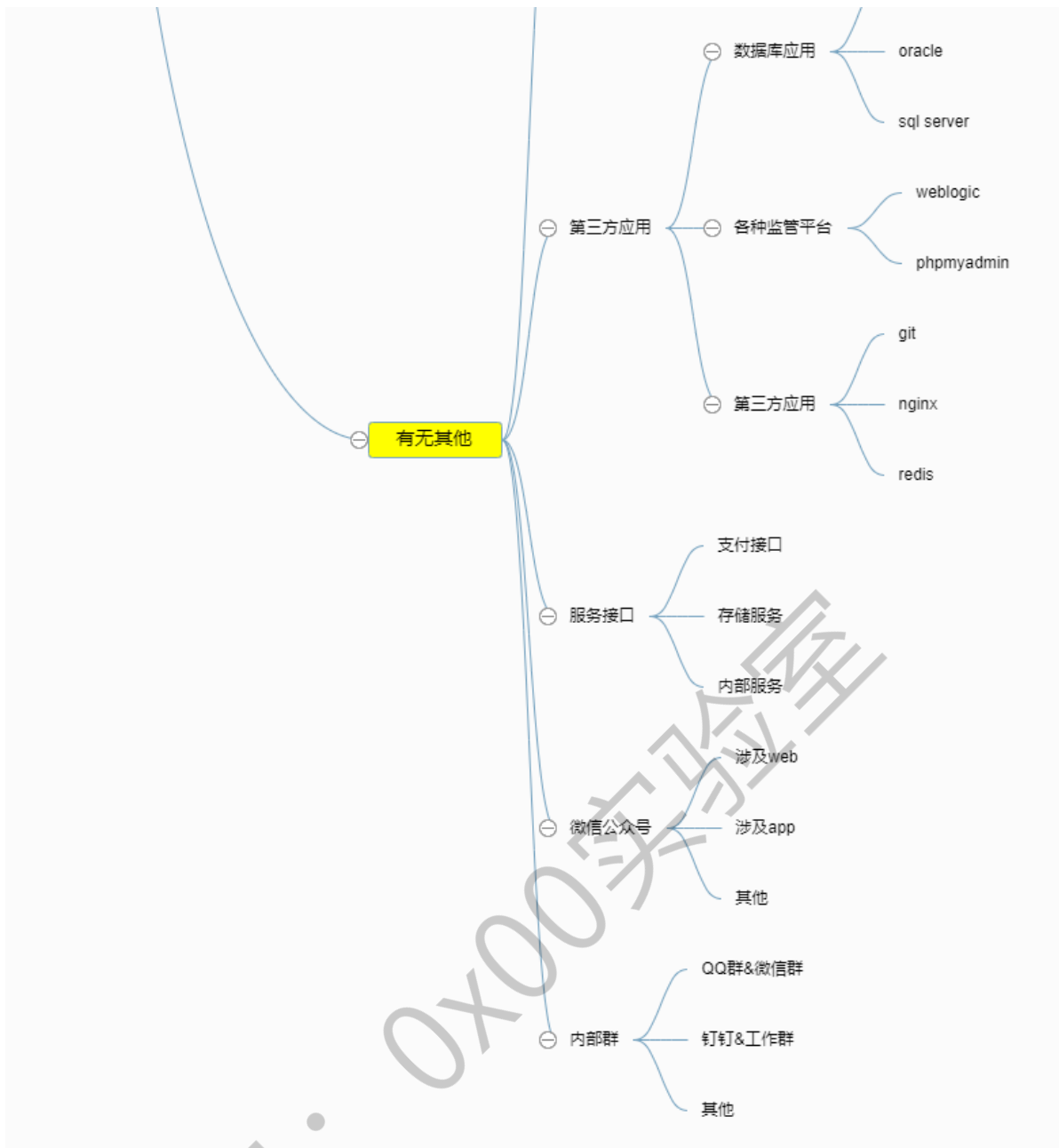
## P8. 架构、搭建、waf

---

公众号: 0x00实验室







源码获取技术  
架构信息获取

## 8.1 站点搭建分析

- 搭建习惯-目录型站点
- 搭建习惯-端口类站点
- 搭建习惯-子域名站点
- 搭建习惯-类似域名站点
- 搭建习惯-旁注, c段站点
- 搭建习惯-搭建软件特征站点

## 8.2 WAF防护分析

- 什么是WAF应用?
  - Web应用防护系统（也称为：网站应用级入侵防御系统。英文：Web Application Firewall，简称：WAF）。
- 如何快速识别WAF?
- 识别WAF对于安全测试的意义?

## 资源

<https://www.shodan.io/>

<https://www.webscan.cc/>

<https://github.com/EnableSecurity/wafw00f>

## P9. APP及其他资产

---

在安全测试中，若WEB无法取得进展或无WEB的情况下，我们需要借助APP或其他资产在进行信息收集,从而开展后续渗透

### 9.1 APE提取一键反编译提取

- 使用反编译工具，尝试获取包里的源码

### 9.2 APP抓数据包进行工具配合

- 使用burp suite设置代理，或者wireshark抓数据包，进行分析

### 9.3 各种第三方应用相关探针技术

- <https://www.shodan.io/>
- <https://fofa.so/>

### 9.4 各种服务接口信息相关探针技术

## 记录

使用burp代理模拟器时，提示证书错误，后将burp生成的证书导入后正常，导入时需要将.cer改成.der。

## P10. 资产监控拓展

---

Github监控

- 便于收集整理最新exp或poc
- 便于发现相关测试目标的资产

各种子域名查询

DNS,备案,证书

全球节点请求cdn

- 枚举爆破或解析子域名对应
- 便于发现管理员相关的注册信息

黑暗引擎相关搜索

- fofa, shodan, zoomeye

微信公众号接口获取dgh



## 资源

<https://crt.sh>

<https://dnsdb.io>

<https://sct.ftqq.com/login>

<https://tools.ipip.net/cdn.php>

<https://github.com/bit4woo/teemo>

<https://securitytrails.com/domain/www.baidu.com/history/a>