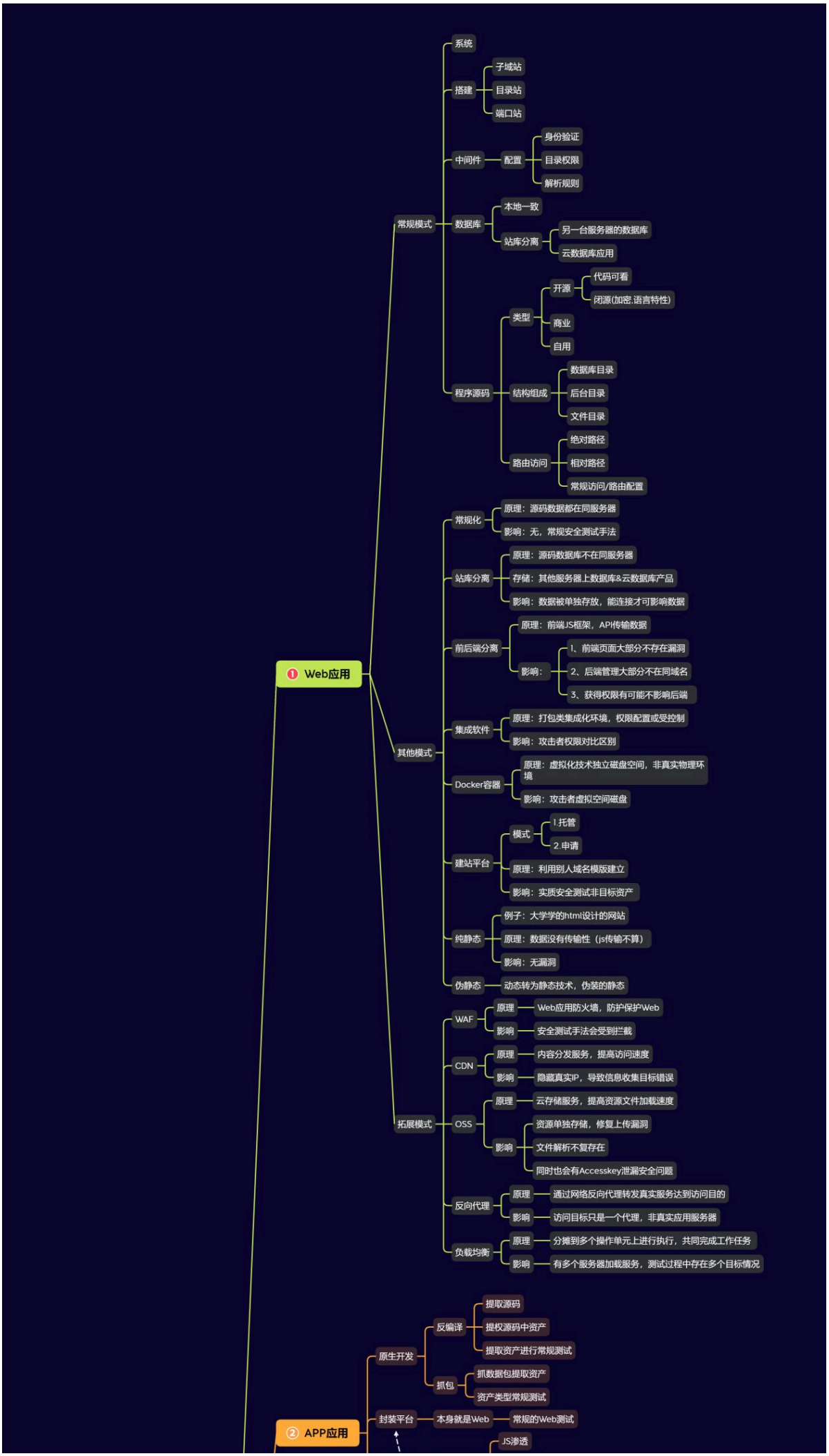
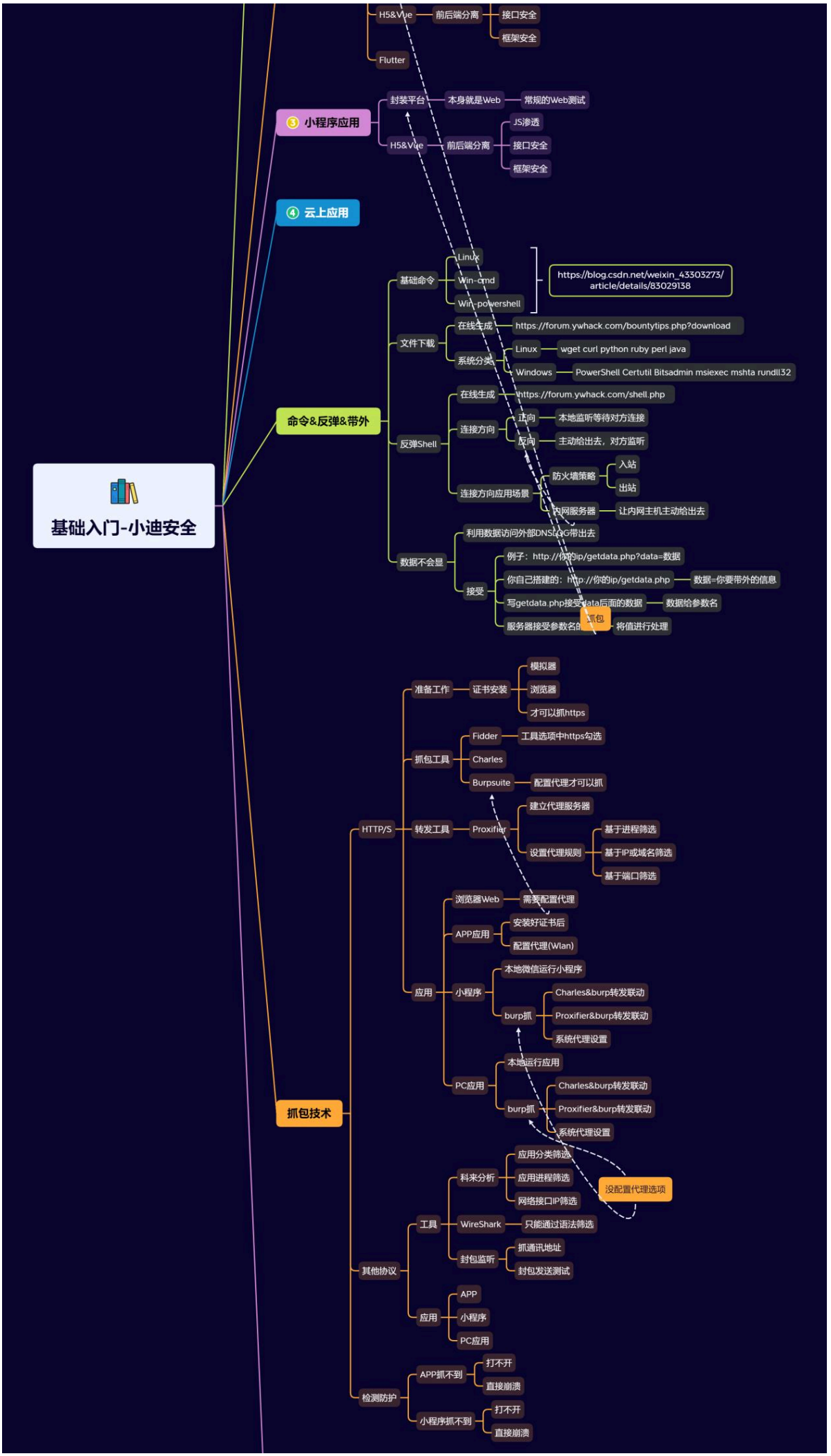
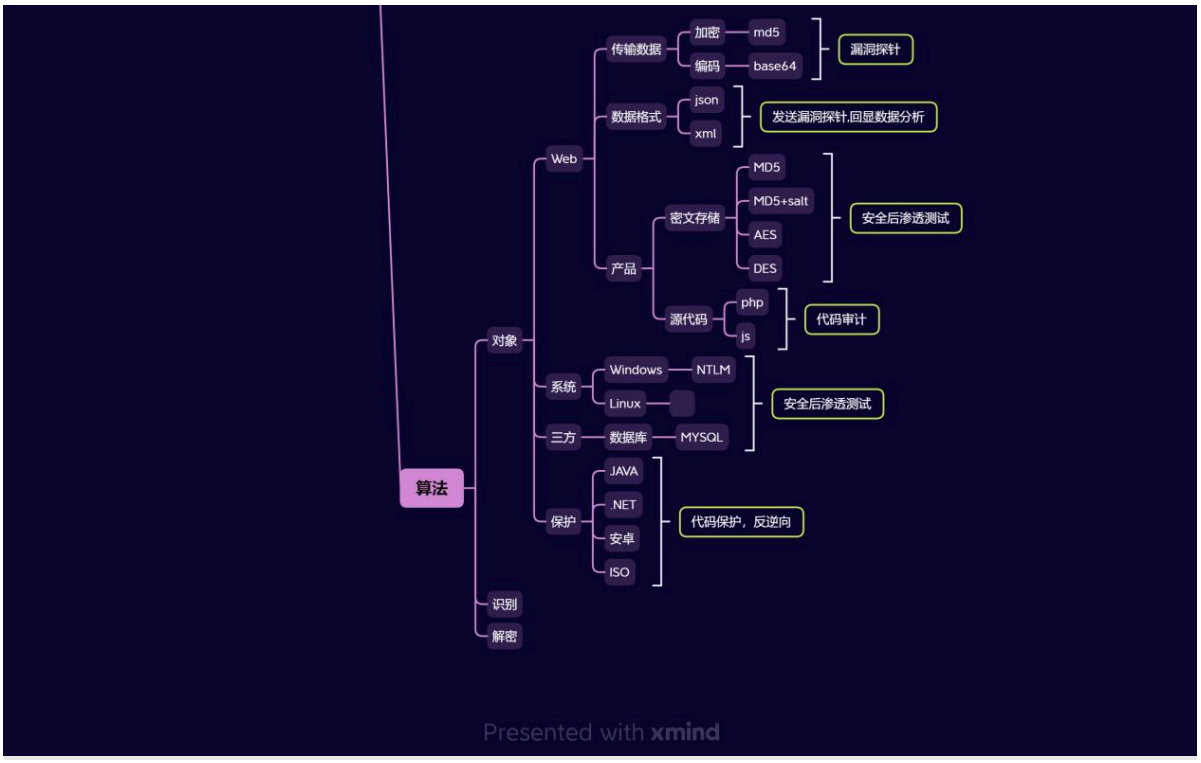
# Day08　基础入门-算法分析&传输加密&数据格式&密文存储&代码混淆&逆向保护


① Web应用
常规模式
系统
搭建
子域站
目录站
端口站
中间件
配置
身份验证
目录权限
解析规则
数据库
本地一致
站库分离
另一台服务器的数据库
云数据库应用
程序源码
类型
开源
代码可看
闭源(加密,语言特性)
商业
自用
结构组成
数据库目录
后台目录
文件目录
路由访问
绝对路径
相对路径
常规访问/路由配置
其他模式
常规化
原理：源码数据都在同服务器
影响：无，常规安全测试手法
站库分离
原理：源码数据库不在同服务器
存储：其他服务器上数据库&云数据库产品
影响：数据被单独存放，能连接才可影响数据
前后端分离
原理：前端JS框架，API传输数据
影响：
1、前端页面大部分不存在漏洞
2、后端管理大部分不在同域名
3、获得权限有可能不影响后端
集成软件
原理：打包类集成化环境，权限配置或受控制
影响：攻击者权限对比区别
Docker容器
原理：虚拟化技术独立磁盘空间，非真实物理环境
影响：攻击者虚拟空间磁盘
建站平台
模式
1.托管
2.申请
原理：利用别人域名模版建立
影响：实质安全测试非目标资产
纯静态
例子：大学学的html设计的网站
原理：数据没有传输性（js传输不算）
影响：无漏洞
伪静态
动态转为静态技术，伪装的静态
拓展模式
WAF
原理
Web应用防火墙，防护保护Web
影响
安全测试手法会受到拦截
CDN
原理
内容分发服务，提高访问速度
影响
隐藏真实IP，导致信息收集目标错误
OSS
原理
云存储服务，提高资源文件加载速度
影响
资源单独存储，修复上传漏洞
文件解析不复存在
同时也会有Accesskey泄漏安全问题
反向代理
原理
通过网络反向代理转发真实服务达到访问目的
影响
访问目标只是一个代理，非真实应用服务器
负载均衡
原理
分摊到多个操作单元上进行执行，共同完成工作任务
影响
有多个服务器加载服务，测试过程中存在多个目标情况
② APP应用
原生开发
反编译
提取源码
提权源码中资产
提取资产进行常规测试
抓包
抓数据包提取资产
资产类型常规测试
封装平台
本身就是Web
常规的Web测试
JS渗透


基础入门-小迪安全
H5&Vue
前后端分离
接口安全
框架安全
Flutter
③ 小程序应用
封装平台
本身就是Web
常规的Web测试
H5&Vue
前后端分离
JS渗透
接口安全
框架安全
④ 云上应用
命令&反弹&带外
基础命令
Linux
Win-cmd
Win-powershell
https://blog.csdn.net/weixin_43303273/article/details/83029138
文件下载
在线生成
https://forum.ywhack.com/bountytips.php?download
系统分类
Linux
wget curl python ruby perl java
Windows
PowerShell Certutil Bitsadmin msiexec mshta rundll32
反弹Shell
在线生成
https://forum.ywhack.com/shell.php
连接方向
正向
本地监听等待对方连接
反向
主动给出去，对方监听
连接方向应用场景
防火墙策略
入站
出站
内网服务器
让内网主机主动给出去
数据不会显
利用数据访问外部DNSLOG带出去
接受
例子：http://你的ip/getdata.php?data=数据
你自己搭建的：http://你的ip/getdata.php
数据=你要带外的信息
写getdata.php接受data后面的数据
数据给参数名
服务器接受参数名的
抓包
将值进行处理
抓包技术
HTTP/S
准备工作
证书安装
模拟器
浏览器
才可以抓https
抓包工具
Fiddler
工具选项中https勾选
Charles
Burpsuite
配置代理才可以抓
转发工具
Proxifier
建立代理服务器
设置代理规则
基于进程筛选
基于IP或域名筛选
基于端口筛选
应用
浏览器Web
需要配置代理
APP应用
安装好证书后
配置代理(Wlan)
小程序
本地微信运行小程序
burp抓
Charles&burp转发联动
Proxifier&burp转发联动
系统代理设置
PC应用
本地运行应用
burp抓
Charles&burp转发联动
Proxifier&burp转发联动
系统代理设置
没配置代理选项
其他协议
工具
科来分析
应用分类筛选
应用进程筛选
网络接口IP筛选
WireShark
只能通过语法筛选
封包监听
抓通讯地址
封包发送测试
应用
APP
小程序
PC应用
检测防护
APP抓不到
打不开
直接崩溃
小程序抓不到
打不开
直接崩溃




## 1.知识点

- 1、Web常规-系统&中间件&数据库&源码等
- 2、Web其他-前后端&软件&Docker&分配站等
- 3、Web拓展-CDN&WAF&OSS&反向&负载均衡等

---

- 1、APP架构-封装&原生态&H5&flutter等
- 2、小程序架构-Web&H5&JS&VUE框架等

---

- 1、渗透命令-常规命令&文件上传下载
- 2、反弹Shell-防火墙策略&正反向连接
- 3、数据回显-查询带外&网络协议层级

---

- 1、抓包技术-HTTP/S-Web&APP&小程序&PC应用等
- 2、抓包工具-Burp&Fidder&Charles&Proxifier

---

- 1、抓包技术-全局-APP&小程序&PC应用
- 2、抓包工具-Wireshark&科来分析&封包

---

- 1、存储密码加密-应用对象
- 2、传输加密编码-发送回显
- 3、数据传输格式-统一格式
- 4、代码特性混淆-开发语言

---

## 2.演示案例

### 2.1 传输数据-编码型&加密型等

```
例：
-某视频
-某web站
-博客登录
-APP-斗地主
影响：漏洞探针
```

### 2.2 传输格式-常规&JSON&XML等

```
例：
-App-期H
-APP-斗地主
影响：发送漏洞探针,回显数据分析
```

### 2.3 密码存储-Web&系统&三方应用

```
例：
-ZZZCMS&Dz
-Win&Linux
-MSSQL&MYSQL
影响：安全后渗透测试
```

### 2.4 代码混淆-源代码加密&逆向保护

```
例：
-PHP&JS混淆加密
-EXE&JAR代码保护
影响：代码审计，逆向破解
```

---

## 资源补充

```
https://indialms.in/wfp_login.php?r_id=1
base64编码
username=YWRtaW4=
https://indialms.in/wfp_login.php?r_id=MQ==
112123

数据在传输的时候进行编码 为什么要了解？

对方服务器可能会在接受的时候进行解码在带入
如果我们还是按照原有思路不对自己的Payload进行同样编码的话 传入过去的东西就是不认识的东西 测试无效

正确：测试的话也要进行payload同样的加密或编码进行提交
安全测试漏洞时候 通常都会进行数据的修改增加提交测试
```

以数据的正确格式发送 接受才行

登录的数据包：
admin 123456

MD5加密

username=admin&password=123456
username=admin&password=e10adc3949ba59abbe56e057f20f883e

如果现在我要进行密码的破解爆破

字典文件：
帐号什么都不用更改 去替换username=值即可
密码需要进行密码算法 保证和password=值同等加密才行

https://tv.sohu.com/v/dXMvMzg1MjM2NzE5LzQyNzUyODUzOC5zaHRtbA==.html

开发：数组 列表

btnPost=%E7%99%BB%E5%BD%95&username=admin and &password=e10adc3949ba59abbe56e057f20f883e&savedate=1

{

    btnPost:"%E7%99%BB%E5%BD%95";

```
    username:"admin";
    password:"e10adc3949ba59abbe56e057f20f883e
and";
    savedate:1;
}


json xml 常规
x=123

x=123
{
    x:123
}


zzzcms admin /123456 密文利用md5加密

md5(123456)=密文

dz3.2 admin /123456

md5(md5(123456).salt)=密文

dz3.5 admin / 123456
aes des（密匙 偏移量 填充 模式等）
$2y$10$OtsSmawENczg1BLcQCEn5OdLqJC9GLiDrClwEUoo
Nnn8b609DfJc.

大部分的解密都是碰撞式解密
不是算法的逆向的还原解密

```

1.常见加密编码进制等算法解析

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

2.常见加密编码形式算法解析

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

3.常见解密解码方式（针对）

枚举，自定义逆向算法，可逆向

4.常见加密解码算法的特性

长度位数，字符规律，代码分析，搜索获取等

#本课意义：

1.了解加密编码进制在安全测试中的存在

2.掌握常见的加密解密编码解码进制互转的操作

3.了解常见的加密解密编码解密进制互转的影响

识别算法编码方法：

1、看密文位数

2、看密文的特征（数字，字母，大小写，符号等）

3、看当前密文存在的地方（Web，数据库，操作系统等应用）

#拓展补充参考资料：

-传输数据编码：

BASE64 URL HEX ASCII

BASE64值是由数字"0-9"和字母"a-f"所组成的字符串,大小写敏感,结尾通常有符号=

URL编码是由数字"0-9"和字母"a-f"所组成的字符串,大小写敏感,通常以%数字字母间隔

HEX编码是计算机中数据的一种表示方法,将数据进行十六进制转换,它由0-9,A-F,组成

ASCII编码是将128个字符进行进制数来表示,常见ASCII码表大小规则：0~9<A~Z<a~z

-传输数据加密：同密码存储加密
-传输数据格式：常规字符串 JSON XML等

-密码存储加密：
MD5 SHA1 NTLM AES DES RC4
MD5值是32或16位位由数字"0-9"和字母"a-f"所组成的字符串
SHA1这种加密的密文特征跟MD5差不多，只不过位数是40
NTLM这种加密是Windows的哈希密码，标准通讯安全协议
AES,DES,RC4这些都是非对称性加密算法，引入密钥，密文特征与Base64类似

代码混淆：
JS前端代码加密：
JS颜文字 jother JSFUCK
颜文字特征：一堆颜文字构成的js代码，在F12中可直接解密执行
jother特征：只用! + ( ) [ ] { }这八个字符就能完成对任意字符串的编码。也可在F12中解密执行
JSFUCK特征：与jother很像，只是少了{ }

后端代码混淆：
PHP .NET JAVA
PHP：乱码，头部有信息
.NET：DLL封装代码文件，加保护
JAVA：JAR&CLASS文件，，加保护
举例：加密平台 Zend ILSpy IDEA
应用场景：版权代码加密，开发特性，CTF比赛等

特定应用-数据库密文加密：
MYSQL MSSQL Oracle Redis等

数据显示编码：

```
126  UTF-8 GBK2312等
127
128  部分资源:
129  https://www.cmd5.com
130  http://tmxk.org/jother
131  http://www.jsfuck.com
132  http://www.hiencode.com
133  http://tool.chacuo.net/cryptaes
134  https://utf-8.jp/public/aaencode.html
135  https://github.com/guyoung/CaptfEncoder
136
137  1.30余种加密编码类型的密文特征分析（建议收藏）
138  https://mp.weixin.qq.com/s?
     __biz=MzAwNDcxMjI2MA==&mid=2247484455&idx=1&sn=
     e1b4324ddcf7d6123be30d9a5613e17b&chksm=9b26f60c
     ac517f1a920cf3b73b3212a645aeef78882c47957b9f3c2
     135cb7ce051c73fe77bb2&mpshare=1&scene=23&srcid=
     1111auAYWmr1NONAs9Wp2hGz&sharer_sharetime=16051
     45141579&sharer_shareid=5051b3eddbbe2cb698aedf9
     452370026#rd
139
140  2.CTF中常见密码题解密网站总结（建议收藏）
141  https://blog.csdn.net/qq_41638851/article/detai
     ls/100526839
142
143  3.CTF密码学常见加密解密总结（建议收藏）
144  https://blog.csdn.net/qq_40837276/article/detai
     ls/83080460
```