防盗链方案设计

1. **UML序列图**



上述流程中的4个成员，可能分别对应于：

* 1. 移动客户端，zeus，jobsfe，和MediaServer；
  2. flv客户端，网站服务器，jobsfe，和MediaServer；

对应移动客户端和zeus之间：

* 客户端发送http://jsonfe.funshion.com/v3/media/get\_serial\_play报文，参考文档<http://redmine.funshion.com/redmine/projects/zeus/wiki/%E5%AA%92%E4%BD%93%E6%92%AD%E6%94%BE%E4%B8%8B%E8%BD%BD%E4%BB%BB%E5%8A%A1>

增加url参数 user=360|others

* zeus回应的json报文如下：

{

"retCode": "200",

"retMsg": "ok",

" encrypt ":"list",

" token":${key1\_token}${encrypt\_token},

" list ":${key2\_data}${encrypt\_data},

}

zeus内置16组加密/解密秘钥，随机选择其中一组为token或者data加密；加密完成后，将对应的解密秘钥组装到token或者data数据包内。(组装示例，请见本文附件。)

对应flash客户端和网站服务器之间：

* 客户端发送<http://api.funshion.com/ajax/get_webplayinfo/106943/1/mp4>报文，
* 网站服务器回应的json报文如下：

{

"category": "movie",

"mediaid": "106943",

" encrypt ":"playinfos ",

" token":${key1\_token}${encrypt\_token},

" playinfos ":${key2\_data}${encrypt\_data},

}

网站服务器内置16组加密/解密秘钥，随机选择其中一组为token或者data加密；加密完成后，将对应的解密秘钥组装到token或者data数据包内。(组装示例，请见本文附件。)

1. **方案设计的三个关键点**
   1. 对客户端的检查

检查客户端是否是合法的客户端。

* 1. token的生成/检查

通过检查token中的时戳，判断是否是报文重放攻击。

通过检查token中的user，检查是否为第三方，是否需要取缔/限速/特殊调度。

* 1. url的加密/解密

通过url加密，隐藏真正的媒体资源播放地址。

1. **对客户端的检查**

客户端如果可以解密，那么继续后面的流程，如果不能解密，那么服务终止。

1. **token的生成/检查**
   1. token的生成

token= "user=360&client\_ip=192.168.160.203&timestamp=1383621431"

* 1. token的检查

按照生成规则，解析得到user, client\_ip, timestamp, 如果timestamp在6小时内，那么继续服务，否则，拒绝服务。

* 1. token加密/传输

token容易被猜解出含义，token也同样加密一次，加密算法与url加密算法相同，也采用Hill加密。

为了传输，加密之后，采用Base64编码。

接收，跟解密，是逆过程，不再详述。

1. **url的加密/解密**

采用Hill2加密/解密算法

1. **Hill2加密/解密算法**

关于Hill算法的介绍，请参考下面两篇文章：

<http://hi.baidu.com/199836199836z/item/6f6261a996f67e9115107331>

<http://baike.baidu.com/link?url=umH7KIZKa2wrF1r5lv6Zmff8onnawqQ2OIx6BeSmDGG5pi1snjmW_gYpSZDAOWqJ>

* 1. Hill2加密

加密算法采用Hill密码算法，矩阵选择2阶矩阵，模选择256。

加密字符串，例如：funshion1，

每两个字符的ASCII值（例如fu），组成行列式（102， 117），与二阶矩阵，例如

{ 129, 77 },

{ 6, 99 }

做乘法运算，得到（13860,19437），做模256运算，得到（36,237）。

* 1. Hill2解密

Hill解密算法，与Hill加密算法，计算过程完全一致，唯一不同的是，解密的二阶矩阵是加密矩阵的逆阵，对应上面就是：

{ 151, 167 },

{ 146, 189 }

* 1. decrypt\_key=Base16(逆阵)

对应上面的逆阵，传输的数据应该是"97A792BD"，固定长度为8BYTE。

1. **示例**
   1. **token**

假设，token="user=360&client\_ip=192.168.160.203&timestamp=1383621431"

* 1. **key1 for token**

加密token的矩阵，假设为：

{ 129, 77 },

{ 6, 99 }

解密token的矩阵，应该为：

{ 151, 167 },

{ 146, 189 }

十六进制打印，得到：

97A792BD

* 1. **token，加密后，得到（十六进制打印如下）：**

395287CD310A1A0AD0C986D17F792AB79F7FADE0531924672AB224671A0A667C767F322ED940C9E2AEE13B5BADE075CDF1A3C043125B31

* 1. **加密后的token，做base64编码，得到：**

OVKHzTEKGgrQyYbRf3kqt59/reBTGSRnKrIkZxoKZnx2fzIu2UDJ4q7hO1ut4HXN8aPAQxJbMQ==

* 1. **组装传输：**

"token":"97A792BDOVKHzTEKGgrQyYbRf3kqt59/reBTGSRnKrIkZxoKZnx2fzIu2UDJ4q7hO1ut4HXN8aPAQxJbMQ=="

* 1. **data**

假设为："hello, funshion"

* 1. **key2 for data**

加密data的矩阵，假设为：

{ 231, 247 },

{ 66, 21 }

解密data的矩阵，应该为：

{ 145, 245 },

{ 166, 59 }

十六进制打印，得到：

91F5A63B

* 1. **data，加密后，得到：**

E2A14C1081B52C3EEFE9957D5D6A6E

* 1. **加密后的data，做base64编码，得到：**

4qFMEIG1LD7v6ZV9XWpu

* 1. **组装传输：**

"data":"91F5A63B 4qFMEIG1LD7v6ZV9XWpu"