威胁猎人
THREAT HUNTER

以情报构筑数字化安全基石

# 2023年Q1数据资产泄露分析报告

出品方：威胁猎人

THREAT HUNTER

以客户为中心 | 正直 | 求真

# 目录

# Contents

# 前言

2023 年 Q1 发生近 1000 起数据泄露事件，涉及 1204 家企业、38 个行业，黑产的数据交易主要集中在更加隐蔽和便利的匿名社交平台。值得一提的是通过短信通道泄露的情况，虽然占比不高但影响极大，仅一起事件就涉及 1000+家企业。

近年来，国家数据安全和个人信息保护相关法律法规相继出台和逐渐细化，2022 年国家级攻防演练中更是新增了对于数据泄漏的攻防点，说明数据安全保护逐步从监管法规落实到具体的攻防实战中来。

今年三月《国务院政府工作报告》再次强调数据安全的重要性，对企业而言，发生数据泄露不仅会受到监管和法律的惩罚，还有可能遭受财产和名誉损失。

Q1 金融企业的用户遭遇钓鱼仿冒类电信诈骗案件上升，其中一个很重要的原因是诈骗分子基于泄漏的用户信息数据，通过钓鱼仿冒网站来实施精准诈骗，因此，及时监测数据泄露风险，做好数据安全建设，是企业发展路上的重要课题。

威胁猎人发布了《2023 年 Q1 数据资产泄露分析报告》，报告内含详细的态势分析、值得关注案例和攻防建议，希望借此报告为企业数据安全建设、防数据泄露提供启发和建议。

# 01

# 数据泄漏风险概况

# 一、2023 年 Q1 数据泄漏风险概况

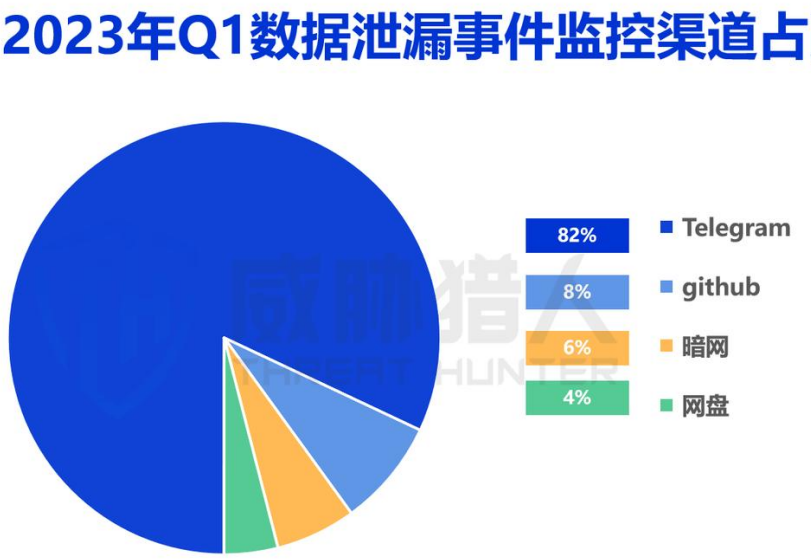
## 1、捕获近 1000 起数据泄露事件，涉及 1204 家企业

2023 年 Q1，威胁猎人情报平台监测和验证到的有效数据泄露事件高达 987 起，疫情结束后黑产更加活跃，相较 2022 年 Q1，本季度的数据泄露事件数上升了 42%，涉及企业多达 1204 家。一月份是春节档期，绝大多数的黑产在休假，因此事件数相对较少，到了二三月，黑产逐渐“上岗”，风险事件数量也在逐月增加。


威胁猎人对事件发布者进一步研究发现，其中有两个黑产在 Q1 发布了 244 个数据泄露事件，从过往发布信息来看，主要针对物流行业，推测应该与全国多家快递公司的快递员合作。

## 2、匿名社交软件 Telegram 是主要数据交易平台，多为二手转卖数据

从威胁猎人监控渠道上来看，2023 年 Q1 的数据泄露渠道主要集中在 Telegram、Github、暗网、网盘 4 个渠道。其中，匿名社交软件 Telegram 因信息传输的私密性和便利性，成为数据交易和传播非法信息的理想平台，占比高达 82%，经威胁猎人溯源，拥有一手数据的人为保护自身安全，会找代理来推广和交易数据，因此传播数据大多为二手转卖数据。




## 3、数据泄露遍布各行各业，涉及物流、金融、电商行业等

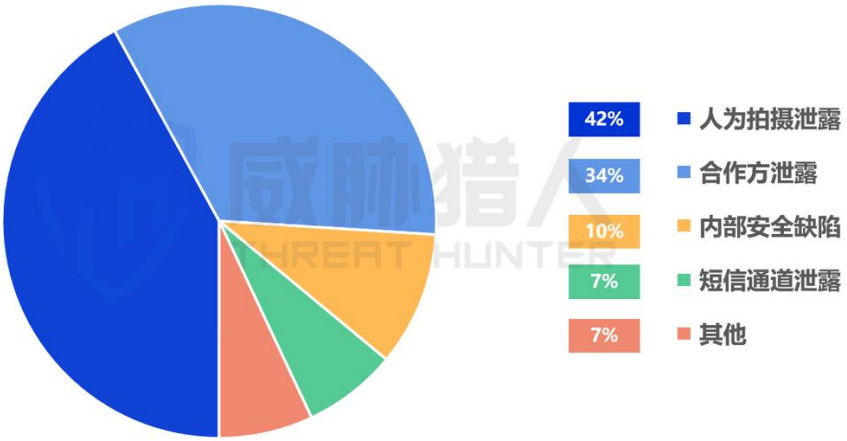
从行业分布来看，2023 年 Q1 的数据泄露事件遍布各行各业，涉及 38 个行业，包含物流、金融、电商、航空、招聘、教育、旅游等行业。

## 数据泄露遍布各行各业


## 4、人为拍摄与合作方泄露是主要泄露原因，涉及销售、快递等环节

从泄露原因来看，本季度人为拍摄信息导致数据泄露的占比最多，高达 42%，进一步研究发现，主要集中在物流行业，涉及销售、仓储、快递等环节的面单信息泄露；合作方泄露位居第二，占比 34%，攻击者往往会攻击供应链上的中小企业，这类企业的安全成本投入低甚至没有，易被攻破，与之合作企业的数据安全很难得到保障。

## 2023年Q1数据泄漏主要原因占比




值得关注的是内部安全缺陷泄露和短信泄露，虽然占比不高，分别为 10%和 7%，但影响范围大，尤其是短信泄露。威胁猎人在 2023 年 3 月 15 号，捕获了一起短信泄露事件：涉及 1000+家企业，其中一家企业被泄漏的短信数量高达 1 亿+条，被黑产在 Telegram 上售卖。

**安全建议：**

1）采购正规短信通道；

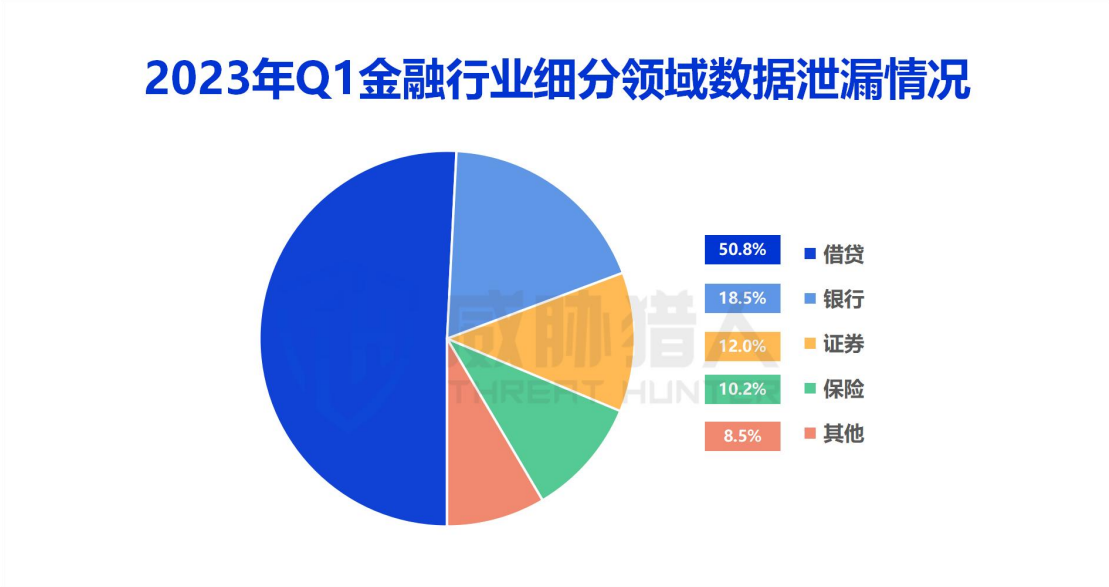
2）给不同通道的短信带上差异化标识，发生数据泄漏事件时，可快速定位问题短信通道。

# 02

# 金融借贷行业
# 数据泄露风险洞察

## 二、金融借贷行业数据泄露风险洞察

威胁猎人 2022 年度《数据资产泄露报告》显示：借贷行业的数据泄露事件数占据金融行业总数据泄露事件数的 38%，位居第一。到 2023 年 Q1，这一数值已经飙升到了 51%，威胁猎人情报平台共监测到相关事件 91 起，远多于其他金融细分行业。




威胁猎人研究员进一步分析原因，发现金融借贷行业通过第三方软件服务（如 SDK）和短信通道泄露的占比最多。通过这两个原因泄露的数据基本只有手机号，难以获取姓名等详细信息，但有心黑产可通过社工库、历史泄露信息等渠道，查询到姓名、地址、身份证等具体用户信息，加上被泄露数据的实时性较高，在黑产领域有巨大市场。


**针对金融借贷行业泄露的数据，黑产多用于诈骗：**

以低利息、下款快等诱惑，引导用户前往其他平台借贷，以“个人信息输入错误为由冻结贷款，需缴纳保证金才能解冻”的方式进行诈骗；或伪装成平台客服，以“账户违规、注销等借口”要求用户缴纳费用进行诈骗。黑产获取的用户个人信息数据越多，让用户确信是客服的可能性越高，实施诈骗的成功率也越高。

近期，威胁猎人情报平台在 Telegram 上捕获一起黑产出售某网贷平台用户信息的安全事件，每天 1-2 千条，涉及字段包含姓名、手机号、下款时间和额度等，极有可能被黑产用于诈骗。

| 姓名 | 电话 | 年龄 | 贷款时间 | 地址 | 身份证 | 贷款金额 |
|---|---|---|---|---|---|---|
| | | 44 | 2023/2/19 19:13 | 陕西省榆林市神木县 | | 601 |
| | | 20 | 2023/2/19 19:14 | 河北省廊坊市霸州市 | | 582 |
| | | 34 | 2023/2/19 19:15 | 河南省焦作市解放区 | | 640 |
| | | 50 | 2023/2/19 19:16 | 陕西省西安市新城区 | | 575 |
| | | 25 | 2023/2/19 19:16 | 福建省福州市鼓楼区 | | 551 |
| | | 29 | 2023/2/19 19:21 | 浙江省杭州市余杭区 | | 572 |

经该网贷平台内部排查，定位到数据泄漏原因是 Spring Boot Actuator 未授权漏洞，API 泄漏了数据库的连接信息，数据库还支持公网连接，黑客直接连接数据库即可窃取数据。

企业需要监控应用程序状态，Spring Boot 内置监控功能 Actuator，当 Spring Boot Actuator 配置不当时，攻击者可通过访问默认的内置 API，轻易获得应用程序的敏感信息：

- 在 Actuator 1.5.x 以下版本，所有 API 都默认无需授权直接访问，存在巨大的安全隐患；
- 在 1.5.x 版本以上，默认只能访问到/health、/info 这两个通常不会泄露敏感信息的 API。但如果配置不当，将/env、/heapdump 等 API 配置为无需授权即可访问，也可能带来安全隐患。

**安全建议：**

1）尽量使用最新版本的应用和系统；

2）数据库等敏感应用和系统，不暴露到公网，或者限制 IP 白名单。

# 03

# 数据泄漏案例

# 三、2023 年 Q1 值得关注的数据泄露案例

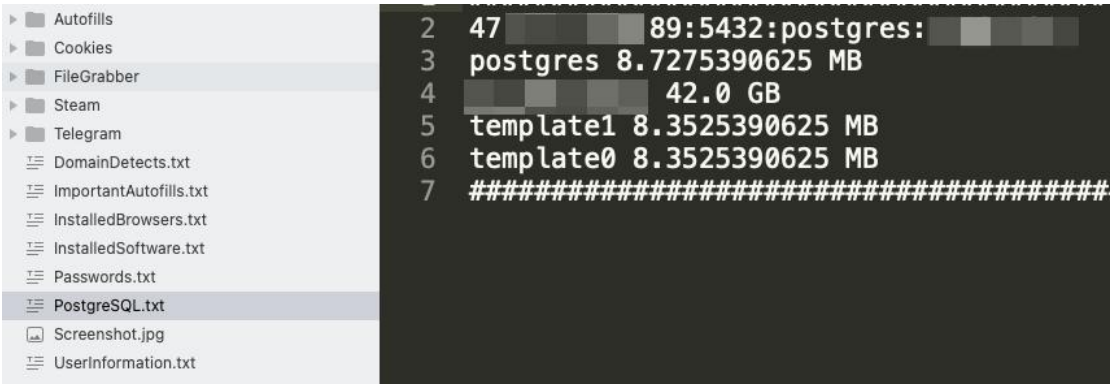
## 1、员工信息泄露案例

### ● 大数据平台员工电脑遭 Stealer log 病毒木马攻击，造成数据泄露

2023 年 3 月 1 日，威胁猎人在暗网发现有黑产出售某大数据平台的数据，包含 50W+条 Json 格式涉及姓名、手机号、部门等字段的数据。威胁猎人情报研究员根据过往黑产发布的信息分析发现，绝大部分数据都是从数据库获取，涉及国家、行业广阔，大概率是通过 MongoDB、MySQL 等数据库弱口令或未授权等方式获取。

经威胁猎人情报专家分析和溯源，发现该数据泄露事件由 Stealer log 导致。Stealer log 是指：记录病毒木马从计算机中窃取的敏感信息的日志文件。文件中包含各类软件/浏览器保存/企业后台系统/FTP/数据库等的账号密码、Cookie 等隐私数据，会导致企业机密、客户资料泄漏等安全事件。

起因是员工个人电脑中过病毒木马，连接过公司的 PostgreSQL 数据库被 Stealer log 记录了下来，因该数据库可被公网访问，黑产直接连接数据库即可窃取。以下为黑产展示其窃取到的 PostgreSQL 数据库连接信息，同时黑产表示还破解了数据库中其他 9 个用户的哈希，仅修改泄漏的账号密码已无法解决该问题。



该黑产从 19 年开始行动，员工个人电脑在 21 年已被病毒木马攻击，直到黑产在 23 年售卖相关数据被监测到才暴露问题，最终导致了此次数据泄漏事件发生。

**安全建议：**

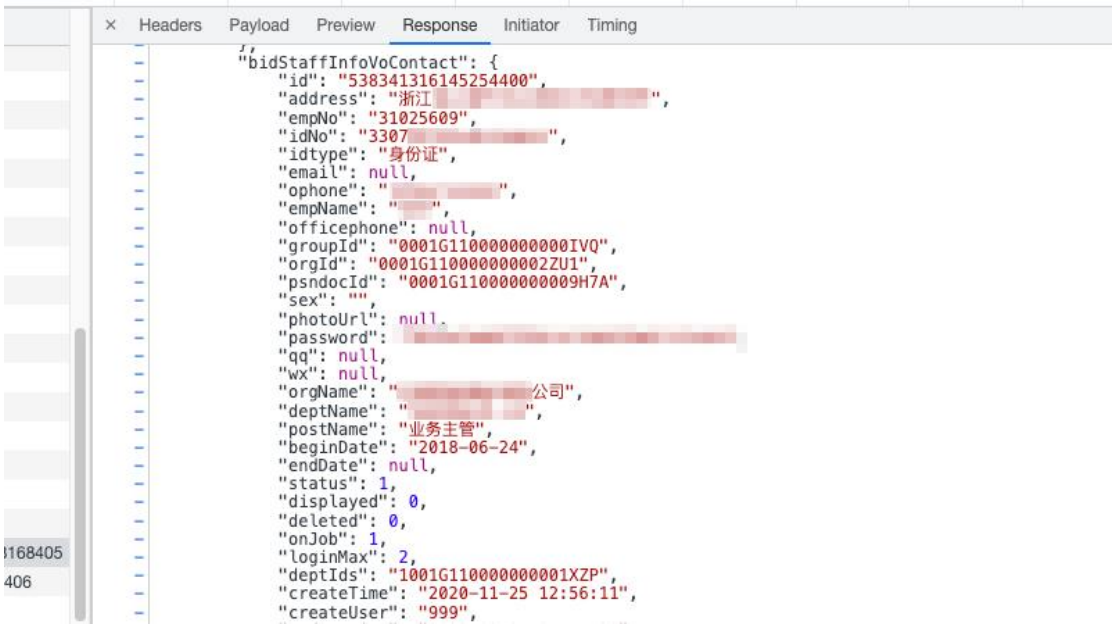
1）更换数据库连接地址，设置 IP 白名单，限制可访问 IP，或将数据库放到内网中，不暴露到公网；

2）用户哈希已被破解，尽快修改密码，防止被密码喷洒攻击；

3）要求内部员工定期修改密码。

## ● 招标平台 API 返回过多员工敏感信息，遭黑产攻击

威胁猎人情报系统监测到，某招标平台的 API 接口正在遭受黑产攻击，原因是该接口返回过多的敏感信息。黑产可直接攻击 API 获取到该公司员工明文的姓名、身份证、手机号、住址、密码等信息，泄漏的密码虽然经过 md5 加密，但仍可以还原出明文密码。



同时该公司的 OA 系统暴露在公网，黑产甚至可以通过接口泄露的账号密码，直接登录 OA 系统，窃取企业内部信息或是实施其他恶意行为。

**安全建议：**

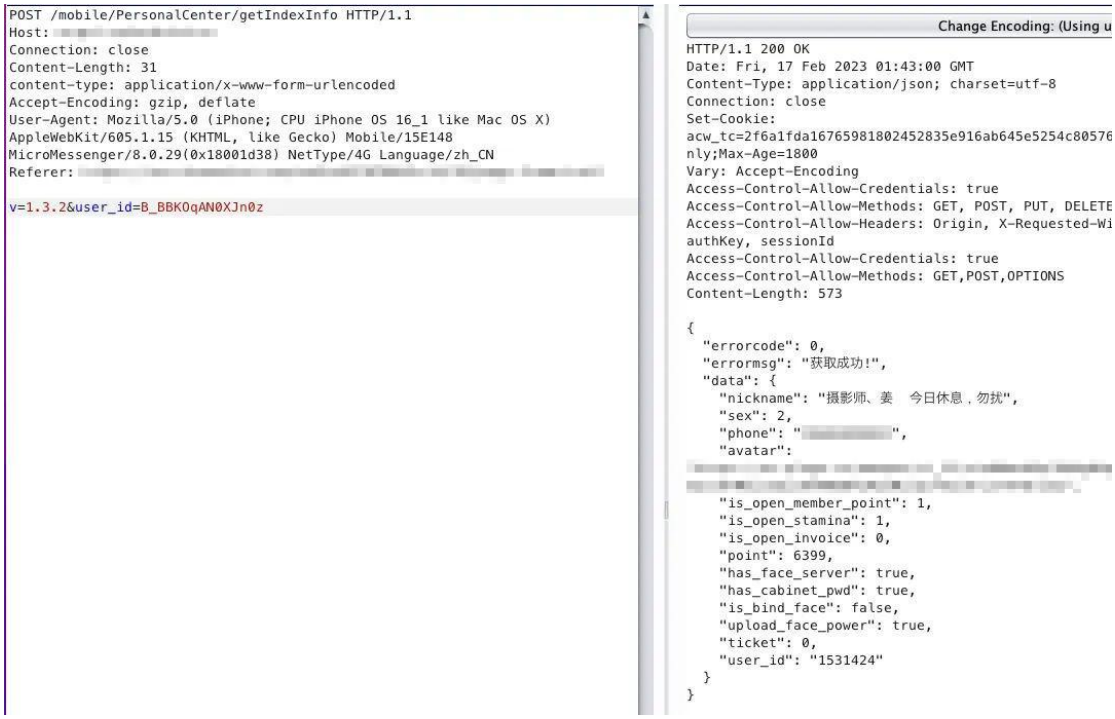
1）确认业务逻辑中是否会用到 API 接口所返回的字段，删除返回多余的字段；

2）告知员工修改密码，排查内部系统是否已被入侵。

## 2、用户信息泄露案例

### ● 健身房存在 API 越权漏洞，会员手机号被爬取

威胁猎人研究员观察到，某健身平台的 API 存在越权漏洞，上传 user_id 就会返回对应用户的手机号，user_id 看起来是不可遍历、预测的，难以被利用。




威胁猎人研究员观察到，该健身房的另一个 API 接口返回了 100 条 user_id，如下图所示：

| 密文user_id | 明文user_id |
| --- | --- |
| B_BBKOsA9Icl30z | 1550971 |
| B_BBKOsAdkYj30z | 1552035 |
| B_BBKOsBNoakn0z | 1555158 |
| B_BBKOsBd8dl30z | 1556681 |
| B_BBKOsBtEdln0z | 1557880 |

通过分析发现，密文 user_id 前面都是 B_BBKOs，明文 userid 前面都是 155，可以说明 user_id 是根据规则来生成的，黑产可自行生成密文 user_id，实现越权获取任意用户的手机号。

**安全建议：**

1）API 接口需要加上鉴权，会员才可以进入系统进行操作，减少暴露面；

2）查询敏感数据时，若查询参数带有 id 类型的字段，使用不可遍历、预测的字符串；

3）校验查询敏感数据的对象是否为当前用户，不属于则不予查询。

## ● 保险代理供应商存在漏洞，合作甲方数据遭泄露

近期，威胁猎人在分析蜜罐流量时发现，多家保险代理公司均存在 API 漏洞，极有可能泄露与其合作甲方的用户数据。

**以保险代理公司 A 为例：**

为给甲方推广保险业务，保险代理公司 A 推出“完善个人信息，领取保险”等活动。页面虽然展示的是脱敏后的个人用户信息，但威胁猎人分析发现，该保险代理公司只是在前端展

示时做了脱敏，API 接口返回的其实是明文数据。并且，该漏洞目前已经被黑产利用，预计泄漏的用户数据高达 3000w+。


```
HTTP/1.1 200
Server: nginx
Date: Wed, 01 Mar 2023 01:33:01 GMT
Content-Type: application/json
Connection: close
Vary: Accept-Encoding
Set-Cookie: SESSION=ZDBiYzMxYWEtYjYzMC00NzQ3LWExYWQtOWNmYzc4YTRkMzM4;
Path=/api/; HttpOnly; SameSite=Lax
Content-Length: 1108

{"code":"000000","message":"SUCCESS","data":{"name":"[redacted]","phone":"[redacted]
[redacted]","orig_id":"[redacted]","fromActivationCode":false,"insure
dName":"[redacted]","insuredIdNo":"[redacted]","relation":"01","securi
ty":"Y","premium":null,"price":null,"residueCount":null,"decryUserInfo":{
"activation_code_id":33894231,"agreeZxUpgrade":"0","batchName":"失效数据2-2
0230220-3000条","channel":"xindan","code":"zxbwyl-djmf","insuredUserId":33
891940,"internal_source":"sms","launchChannel":"fengshengdt","launchSubCh
annel":"fs_gdt_djmf_c03","marketingChannel":"xindan","marketingSubChannel
":"xindan_17_000040000300006","name":"[redacted]","orig_id":"[redacted]
90","payTerm":"month","phone":"[redacted]","position":"sms_activation_co
de_page","productDataType":"","subchannel":"xindan_17_000040000300006"},"
actionCodeId":null,"code":"zxbwyl-djmf","rybUserId":null,"productName":"[redacted]
[redacted]","ifInsured":false,"insureNameShow":"[redacted]先生","idNo":"[redacted]
[redacted]","autoRenewal":"","payTerm":"month","checkVerificationCode
":0,"url":null,"overdue":null,"redirectUrl":null}}
```


该缺陷 API 通过传入 encryptStr 来获取个人敏感信息，而 encryptStr 是通过另一个缺陷 API 接口获取。黑产通过遍历 url 中的参数即可拿到不同用户的 encryptStr，再去上述的接口请求，即可获取到不同用户的个人信息。

**安全建议：**

1）与合作方传输涉敏数据时，需对数据进行加密，同时提高数据导出权限，避免传输过程中暴露，并做好内部审查；

2）要求合作方依据个保法要求保护、加密、脱敏等，同时必须了解具体数据传输流程，避免数据流露到其他平台或平台防护较为薄弱；

3）及时捕捉数据交易市场实时动态，根据相关线索分析涉及公司的泄露事件，判断泄露环节。

数据的对象是否为当前用户，不属于则不予查询。

### ● 快递运单信息泄露，黑产以 0.9 元一条出售

威胁猎人监测到，Telegram 上有黑产在出售某快递的运单信息数据，字段包含运单编号、产品类型、收件人地址、手机号、姓名，派送员姓名等，一天可提供的数据 5 万+条，价格为 0.9 元一条。

| 运单编号 | 派件网点 | 派件时间 | 派件员 | 签收时间 | 签收网点 | 签收人 | 收件地址 | 收件手机 |
|---|---|---|---|---|---|---|---|---|
| [redacted] | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

在获得该快递平台授权后，威胁猎人展开调查，通过黑产透露的后台带水印的截图，定位到此次数据泄漏事件由于离职员工账号权限未及时收回导致。目前，该快递平台已收回离职员工权限并反馈警方，窃取数据的黑产已拘捕在狱。

**安全建议：**

1）及时收回离职员工账号和权限；

2）后台系统不开放到公网访问，需要 VPN 才能访问，或限制 IP 白名单；

3）限制敏感数据能够访问的权限，只对少数确实有需要的账号开放。

## 3、代码信息泄露案例

### ● 某企业代码在 Github 发生泄露，因员工导致

近期，Twitter 在 Github 上发生代码泄露的事件引发全网热议，除此之外，威胁猎人情报情报也监测到一起 Github 泄露事件，经溯源发现，事因某员工往 Github 上传的日志文件中包含了插入用户表的 SQL 语句，因此泄露了管理员账号密码，黑产拿到该账号密码后直接登录管理后台，最终造成了数据泄露、甚至系统权限失限等严重后果。

```
  File "/usr/local/lib/python3.7/site-packages/sqlalchemy/util/compat.py", line 128, in reraise
    raise value.with_traceback(tb)
  File "/usr/local/lib/python3.7/site-packages/sqlalchemy/engine/base.py", line 1244, in _execute_context
    cursor, statement, parameters, context
  File "/usr/local/lib/python3.7/site-packages/sqlalchemy/engine/default.py", line 552, in do_execute
    cursor.execute(statement, parameters)
  File "/usr/local/lib/python3.7/site-packages/pymysql/cursors.py", line 170, in execute
    result = self._query(query)
  File "/usr/local/lib/python3.7/site-packages/pymysql/cursors.py", line 328, in _query
    conn.query(q)
  File "/usr/local/lib/python3.7/site-packages/pymysql/connections.py", line 517, in query
    self._affected_rows = self._read_query_result(unbuffered=unbuffered)
  File "/usr/local/lib/python3.7/site-packages/pymysql/connections.py", line 732, in _read_query_result
    result.read()
  File "/usr/local/lib/python3.7/site-packages/pymysql/connections.py", line 1075, in read
    first_packet = self.connection._read_packet()
  File "/usr/local/lib/python3.7/site-packages/pymysql/connections.py", line 684, in _read_packet
    packet.check_error()
  File "/usr/local/lib/python3.7/site-packages/pymysql/protocol.py", line 220, in check_error
    err.raise_mysql_exception(self._data)
  File "/usr/local/lib/python3.7/site-packages/pymysql/err.py", line 109, in raise_mysql_exception
    raise errorclass(errno, errval)
sqlalchemy.exc.ProgrammingError: (pymysql.err.ProgrammingError) (1146, "Table 'demo.users' doesn't exist")
[SQL: SELECT users.is_delete AS users_is_delete, users.create_time AS users_create_time, users.update_time AS users_update_time, users.id AS users_id, users.email
AS users_email, users.username AS users_username, users.password AS users_password, users.permission AS users_permission, users.avatar AS users_avatar,
users.login_time AS users_login_time
FROM users
WHERE users.username = %(username_1)s
 LIMIT %(param_1)s]
[parameters: {'username_1': 'admin', 'param_1': 1}]
(Background on this error at: http://sqlalche.me/e/f405)
[redacted] 09:36:34,633 - users.py[line:68] - INFO: (pymysql.err.IntegrityError) (1062, "Duplicate entry '[redacted]' for key 'users.email'")
[SQL: INSERT INTO users (is_delete, create_time, update_time, email, username, password, permission, avatar, login_time) VALUES (%(is_delete)s, %(create_time)s,
%(update_time)s, %(email)s, %(username)s, %(password)s, %(permission)s, %(avatar)s, %(login_time)s)]
[parameters: {'is_delete': 0, 'create_time': datetime.datetime(2021, 2, 3, 9, 36, 34, 623655), 'update_time': datetime.datetime(2021, 2, 3, 9, 36, 34, 623701),
'email': '[redacted]', 'username': '[redacted]', 'password': '[redacted]',
'permission': '[redacted]', 'avatar': '[redacted]', 'login_time': [redacted]}]
(Background on this error at: http://sqlalche.me/e/gkpj)
```

**安全建议：**

对于因员工失误将代码推送到 GitHub 上导致的信息泄漏，威胁猎人安全专家建议：

1）立即移除代码：立即将代码从 GitHub 上移除，尽早停止信息泄漏的影响；

2）分析泄漏程度：评估信息泄漏程度和影响范围，包括可能泄露的数据、已访问到数据的人数等。

3）加强安全意识教育：加强员工的安全意识教育，提高他们对代码安全的重视和保护意识；

4）定期审查：定期审查代码库，发现漏洞或问题时及时修复，防止类似问题再次发生。

## 4、敏感文件泄露案例

## ● 多家企业敏感文件被泄露，涉及机密文件、服务器等信息

Telegram 和暗网是敏感文件数据泄露的高发地段，除此之外，网盘、文库、在线文档等渠道也可能泄露企业敏感信息。近期，威胁猎人情报平台监测到，多家企业的敏感信息在网上被泄露，此处列举两个案例。

**案例一：**某银行贷款业务机密文件被泄露在某文库，不仅影响银行声誉，而且可能导致银行违反法规承担法律责任，除此之外，黑灰产也可以利用被泄露的机密文件实施违法行为，比如复印该文件假装该银行工作人员行骗。


**案例二：**某企业的内部在线文档遭泄露，泄露字段包含详细的服务器信息、APP 信息、企业在支付宝等平台的信息等，甚至暴露了具体的登录账号和密码，如果这些信息被黑产发现，黑产可直接登录账号密码，实施窃取机密文件等恶意行为。


**安全建议：**

1）告知员工保护数据的重要性，引导员工学习企业的数据安全政策和网安知识；

2）使用安全的文件共享平台，例如：加密的云存储或私有文件服务器；

3）实施访问控制措施，例如：员工使用 VPN 连接到公司网络，才能访问敏感文件；

4）定期安全审计和漏洞扫描，以确保网络和系统的安全，如发现问题立即补救；

5）建立安全响应计划，应对数据泄露和安全事件。

# 04

# 数据保护措施与建议

# 四、数据保护措施与建议

## 对外：数据泄漏风险监测

企业数据资产多样化并且价值越来越高，涉及用户信息、员工信息、敏感文件、业务代码等，数字化带来数据的泄露面也更大，及早感知可能的数据泄漏风险越发重要。威胁猎人数据泄漏监测情报，全面监测黑产的数据交易渠道、敏感文件和代码的外发渠道，助力企业及时感知、及早溯源和防御潜在数据泄漏风险，避免大规模的数据泄漏影响业务的正常发展。

## 对内：加强 API 安全建设

企业应当在“业务优先”的基础上，加强 API 安全建设，通过 API 安全管控平台，全面梳理对外开放的 API、流动的敏感数据和访问账号，实现对敏感数据异常访问风险的及时监测。威胁猎人的 API 安全管控平台对企业的 API、敏感数据和访问账号进行全面的梳理，评估 API 资产的未授权、越权访问、敏感数据暴露过多等设计缺陷，基于黑产攻击情报及时识别数据爬取、账号异常数据访问等风险，从根源杜绝数据泄露风险发生。

做到内部资产安全与外部威胁情报监测结合，企业才能高效和有效地应对数据泄露风险，在数字化建设与创新发展的道路上，走得又快又稳。

威胁猎人
THREAT HUNTER

以情报构筑数字化安全基石

关注“威胁猎人”


**深圳永安在线科技有限公司**

总部地址：深圳市南山区科华路3号讯美科技广场3号楼1609

官方服务热线：400-809-3699

官方合作邮箱：marketing@threathunter.cn

官网地址：www.threathunter.cn