企业账号泄露风险检测云主机

使用手册（V1.0）

浙江浩安信息技术有限公司

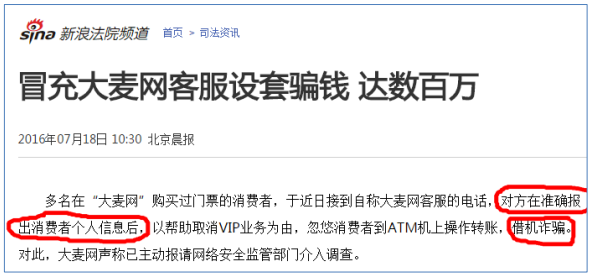
2018.04

目　录

1. 应用背景

数据泄漏问题已经被公认为互联网+时代最大的安全问题之一，各类型的用户信息在互联网的各种隐蔽渠道被进行交易传播。在泄露数据中最隐私、最有价值、也最危险的数据是账号密码信息，但据不完全统计，目前已经泄漏的用户账号数据已经多达几十亿的规模。

用户账号具有很强的通用性，用户为了记忆和使用方便，多个网站采用同样或相似的用户名和密码。一旦账号密码在某个网站因防范不严而发生泄露，就非常容易被不法分子利用这些已泄露的账号对其他站点进行**“撞库攻击”和“账号劫持”**，继而引发隐私泄露、商业诈骗、黑客攻击等一系列危害，最终对企业造成巨大经济损失。例如，采用在A网站泄露的账号密码去登陆B网站，一旦登录成功，即可假冒真实用户进行黑客攻击或获取真实用户信息后进行诈骗。



电商诈骗

网银诈骗

企业虽投入大量资金进行安全管理工作，但不得不承担其他企业防范不严造成账号信息泄露的恶果，企业由此将蒙受不白之冤，并被迫承担经济和声誉的巨大损失。

泄露账号是悬在企业头上的达摩克利斯之剑，随时可能爆发巨大危机。由于企业无法有效判断输入正确账号和密码的用户是否是真实用户本人，进一步加强账号登录审核的业务影响和管理成本较大，因此必须首先检测评估本企业员工或用户账号数据在互联网的整体泄露情况，随后对泄露账号采取针对性的应对措施。但是企业在实际开展泄露账号检测过程中将面临以下难题：

- 收集互联网上其他企业的泄露账号数据进行分析可能存在潜在法律或商业风险；
- 资源和渠道有限，无法大规模、大范围地对泄露数据进行全面收集；
- 海量泄露数据的收集、清理、维护和更新工作繁琐且需要时间积累，资源消耗极大。

2. 总体介绍

账号泄露风险检测云主机（以下简称为“**检测主机**”）是针对企业用户的账号泄露管理痛点，推出的具有自主知识产权的、**业界第一款专为企业用户进行账号泄露风险检测的商用产品**，应用了多项信息安全专利技术。

企业账号泄露风险检测云主机具有以下几个突出特点：

- **私有网络检测，过程自主可控**

运行于云端环境，但完全可设置安全规则，只连通客户内网环境，无需与外部网络进行连接，企业自有账号信息无需出内网；

- **数据全面真实，持续维护更新**

预置互联网真实泄露的九十亿条以上账号风险数据，数据全面真实，并能够持续更新补充。

数据来源于真实互联网环境，准确可靠且持续更新，对企业具有巨大的实际参考价值；

- **专用硬件处理，高效安全可靠**

硬件安全芯片作为数据安全保障，确保账号数据无法被记录和入侵，账号分析比对过程在硬件芯片中进行，全过程安全无忧；

- **功能使用简便，集成接口丰富**

提供管理界面、检测软件和本地 API 集成接口，支持 Java/PHP/Python/Javascript 等多种语言，以通用数据格式反馈账号风险分析结果，并提供多个常用检测代码模板供用户直接使用。

3. 功能使用介绍

3.1 企业账号泄露风险检测

企业账号泄露风险检测采用脚本的形式进行配置和工作，为用户提供了充分的灵活性。

检测工具所在的企业账号泄露风险检测主机中的文件夹位置如下：

C:\checkPortal\vericlouds_setup\scripts

检测工具将输入的用户名和密码（明文密码或hash密码）与检测主机内置的海量账号风险情报数据进行比对分析，从而直接得出泄露风险结论。

检测工具使用Python/Java 程序脚本对企业当前的员工账号或外部用户账号进行批量泄露检测和弱密码检测。检测前应导出员工账号或外部用户账号用户名和密码（一般为Hash密码格式）的检测内容清单。

检测脚本具体执行方式如下：

C:\checkPortal\vericlouds_setup\scripts> C:\Python27\python.exe .\accounts_scanner.py

在脚本中指定读取了待检测账号清单文件，（脚本中默认指明了清单文件名称为test_accounts_md5.csv，可根据需要更改），文件类型不限，但对文件内容的格式有具体要求。

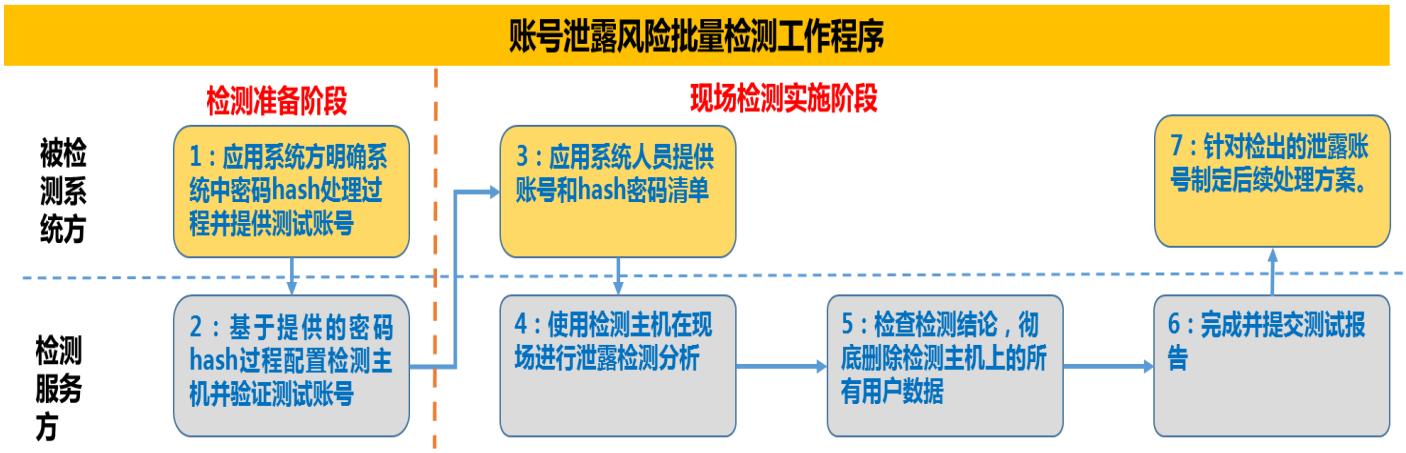
检测账号清单的格式要求如下：

Test1@xxx.com----e10adc3949ba59abbe56e057f20f883e

Test2@xxx.com----14e1b600b1fd579f47433b88e8d85291

每一行为一个用户名和对应的hash密码，中间用四个短横线“----”分开

批量检测的工作程序如下图所示：



步骤1：明确员工账号/外部用户账号的密码hash处理方式，如是采用md5($pass)或者md5(md5($pass))，ssha512($pass)等等，如果采用了固定盐值，需首先获取相关盐值。

步骤2：在批量检测脚本中按照密码的hash处理方式进行配置，在“***source_config***”字段中选择相应的hash方式，建议在正式大批量检测时，选取少量数据进行测试，确保配置正确；

步骤3：准备好包含用户名和密码清单内容的检测内容清单，按照格式要求，用户名和密码中间用“----”隔开；

步骤4：调整批量检测脚本，以python版本的批量检测脚本accounts_scanner.py为例：

可以根据需要调整脚本的参数配置，主要在source_config行调整参数：

source_config =

{'location':'test_accounts_md5.csv','source':'test_accounts','parse_function':parse_account,'checks':['compromise','common'],'source_type':'md5','index_begin':0,'count':100,'is_local':True,'ignoreEmailDomain':False,'showCompromise':True,'repeat':1,'noCache':True}

字段解释：

location：检测清单文件的名称和文件位置，建议此清单文件与批量检测脚本放置于同一文件夹；

source：无需调整

parse_function：选择检测清单文件的解析函数，可以选择本批量脚本内已有的解析函数，也可新建其他解析函数；

checks：检测泄露密码（compromis）或是弱密码（common），可以保留一个，也可以都保留；

source_type: 检测清单文件中密码采用的hash算法名称，如MD5，SSHA512等；

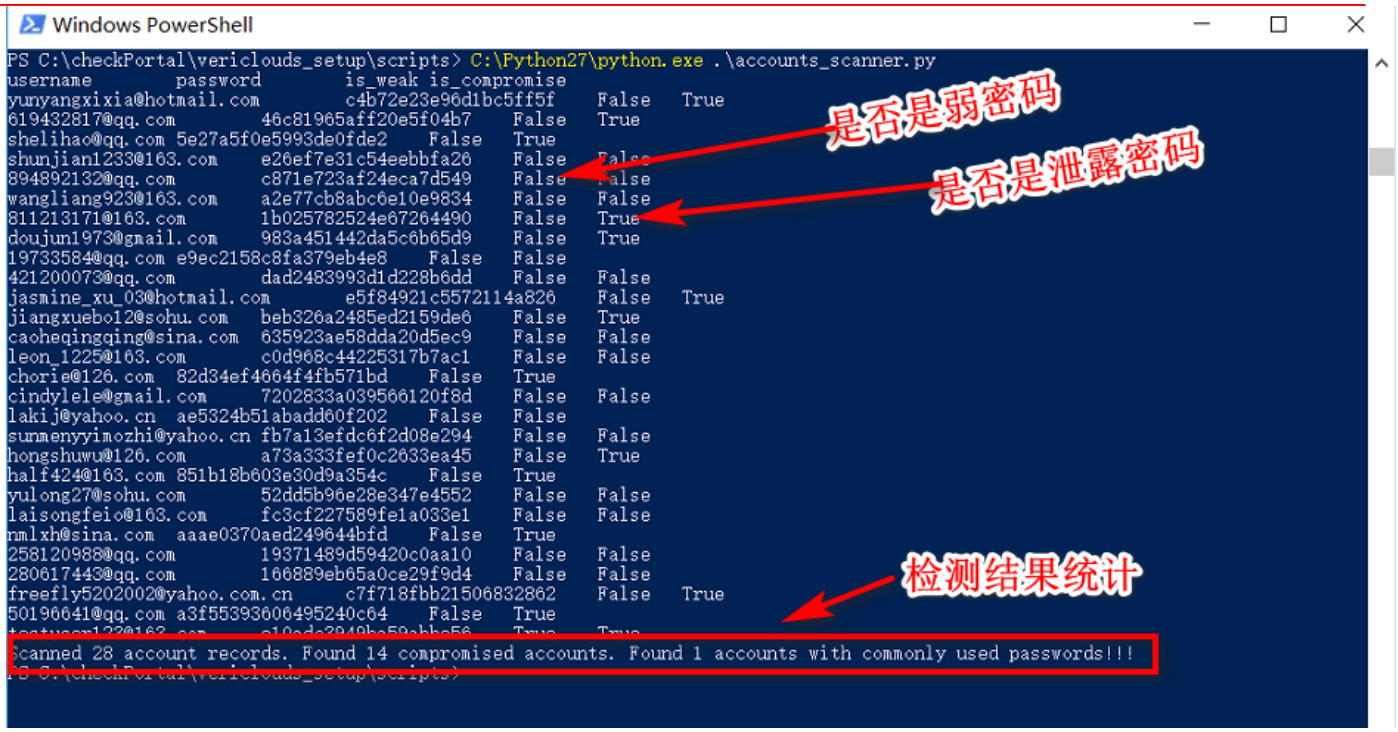
index_begin: 检测清单文件中开始检测的行数，默认是0，即从第一行开始检测；

count： 检测清单文件从index_begin行开始检测的行数

gnoreEmailDomain：检测清单文件中的用户名为邮箱是，是否忽略邮箱域名，即自动选取邮箱@符号前的内容作为用户名；

noCache：对于同一用户名重复检测时，是否读取上一次检测的缓存数据而不真正进行检测比对分析；

步骤5：运行批量检测脚本正式检测，查阅检测结论，并可将检测结果进行后续分析。


Windows PowerShell
PS C:\checkPortal\vericlouds_setup\scripts> C:\Python27\python.exe .\accounts_scanner.py
username password is_weak is_compromise
yunyangxixia@hotmail.com c4b72e23e96d1bc5ff5f False True
619432817@qq.com 46c81965aff20e5f04b7 False True
shelihao@qq.com 5e27a5f0e5993de0fde2 False True
shunjian1233@163.com e26ef7e31c54eebbfa26 False False
8948921320@qq.com c871e723af24eca7d549 False False
wangliang923@163.com a2e77cb8abc6e10e9834 False False
811213171@163.com 1b025782524e67264490 False True
doujun1973@gmail.com 983a451442da5c6b65d9 False True
19733584@qq.com e9ec2158c8fa379eb4e8 False False
421200073@qq.com dad2483993d1d228b6dd False False
jasmine_xu_03@hotmail.com e5f84921c5572114a826 False True
jiangxuebo12@sohu.com beb326a2485ed2159de6 False True
caoheqingqing@sina.com 635923ae58dda20d5ec9 False False
leon_1225@163.com c0d968c44225317b7ac1 False False
chorie@126.com 82d34ef4664f4fb571bd False True
cindylele@gmail.com 7202833a039566120f8d False False
lakij@yahoo.cn ae5324b51abadd60f202 False False
sunmenyyimozhi@yahoo.cn fb7a13efdc6f2d08e294 False False
hongshuwu@126.com a73a333fef0c2633ea45 False True
half424@163.com 851b18b603e30d9a354c False True
yulong27@sohu.com 52dd5b96e28e347e4552 False False
laisongfeio@163.com fc3cf227589fe1a033e1 False False
nmlxh@sina.com aaae0370aed249644bfd False True
258120988@qq.com 19371489d59420c0aa10 False False
280617443@qq.com 166889eb65a0ce29f9d4 False False
freefly5202002@yahoo.com.cn c7f718fbb21506832862 False True
50196041@qq.com a3f55393606495240c64 False True
Scanned 28 account records. Found 14 compromised accounts. Found 1 accounts with commonly used passwords!!!
是否是弱密码
是否是泄露密码
检测结果统计

3.2 账号泄露检测API接口

检测云主机提供安全并易于使用的API接口，Http://<检测主机IP>/vericlouds/API/index.php

此接口使客户通过将自有账号信息与数十亿计的账号泄露情报数据进行比对从而或者自有账号是否发生泄露。此接口可与外部系统进行集成，例如与用户登录管理系统进行集成。用户在登录/注册/修改密码时，直接调用泄露检测API接口对输入密码进行实时的检测分析，发现泄露后提醒用户修改密码；检测接口还适用于其他管理系统的集成使用场景。

检测云主机在交付给客户使用前，将配置生产唯一的“api_secret”字符串，此字符串将作为此客户独有的密钥用于接口反馈信息的解密，应被妥善管理和保存。

3.2.1 API使用的基本场景

使用场景1：向API发送用户名，检查是否存在泄露密码

为了检查一个用户名是否在账号泄露情况数据集中，可以将用户名 userid 发出一个 REST API 请求到 API，并设置参数为“search_leaked_password_with_userid”。如果检测成功，API 将返回一个加密后的泄露密码的特征信息（包括泄露密码的首尾字母和长度信息），此泄露密码特征信息采用 AES 256 CBC 加密。

使用客户独有的“api_secret”可对接口反馈的泄露密码特征信息进行解密，得到密码首字母、尾字母和长度信息。

如果客户有此用户名的明文密码信息，可将明文密码与解密后的泄露密码特征进行比较，从而判断此明文密码是否已经发生泄露；如果客户没有此用户名的明文密码，则可根据 API 接口反馈的泄露密码数量宏观判断此用户名的泄露风险大小，但无法得出准确的泄露结论。

使用场景2：向API发送用户名和hash密码，检查密码是否已经泄露

如果用户有hash格式的密码，需首先明确密码的hash过程和方法，如SHA1、MD5或其他方式，检测API需要提前进行相关配置。

检测API完成Hash处理方式配置后，实际进行检测使用时，用户将用户名和对应的hash密码前6位发送给API接口， API将首先根据用户名进行检索，找到匹配用户名后，再将账号泄露情报数据库中，此用户名下的泄露密码按照配置好的Hash方法进行处理，然后将前6为hash字符与输入的hash密码前6位字符进行比对，比对成功则说明输入的用户名和密码已经发生泄露，API接口将反馈泄露检测结论。

3.2.2 API参数说明

API请求参数:

所有的 API 请求参数使用"urlencoding"方式进行编码，并已 POST 方法发送到 API 接口。

参数名	参数说明
mode	**search_leaked_password_with_userid** 和 **search_leaked_password_with_hash_segment**. 前者允许基于用户名发起API请求，后者允许基于密码hash发起API请求。前者模式下, **userid**是必选参数.后者模式下, **hash_segment** 是必选参数
api_key	客户独有的固定参数，产品交付前提供给用户。
api_secret	客户独有的固定参数，产品交付前提供给用户。用于解密API反馈的加密数据，例如泄露密码的特征信息。此参数内容非常重要，不应分享给第三方。
userid	用户名. 目前API仅支持email地址形式的用户名.此参数仅在 **mode** 设置为 **search_leaked_password_with_userid 时使用**.
userid_type	**userid的类型**. API支持**default** and **hash两种形式的userid**. 如果设置成**hash**，则 **userid** 应为 SHA256 计算之后的user ID 哈希值，而不是明文形式的user ID.
hash_segment	密码哈希值的前6位字符。仅当**mode**选择 **search_leaked_password_with_hash_segment时需设置此参数值**.
Context	**请求API反馈的其他内容，仅在mode=search_leaked_password_with_userid** 模式下可设置，包括 **source_type（泄露源类型）**, **source_count（泄露源数量）**, 等，使用逗号隔开

API响应参数:

API 反馈的数据采用 JSON 格式，如下:

```
{
  "result": "succeeded",
  "passwords_encrypted": [
```

```
    ":",
    ":",
    ...
  ],
  "quota": "1000",
  "quota_used": "130",
  "time_lapse": "0.23"
}
```

参数名称	参数描述
result	请求的状态，**succeeded** 表示API call成功处理， **failed标识API call处理失败，且 reason** 字段将给出原因描述.
passwords_encrypted	找到的泄露密码列表，每个密码均采用AES 256 CBC mode加密.
<ecncrypt_passwod>	API接口基于 **userid所找到的泄露密码，且密码使用AES 256加密** . **api_secret** 参数可用来对找到的泄露密码进行解密。处于保护用户隐私的目的，找到的泄露密码进行了脱敏处理，只显示首尾字母和长度信息。例如找到的泄露密码是 "123456",解密后将只能看到 "1****6" .
<iv>	初始向量值，使用初始向量对 ecncrypt_passwod进行加密
source_types	**passwords_encrypted** 列表中每一个泄露密码的泄露源类型. 此字段仅在输入参数**context** 包含了 **source_type**时生效
source_counts	**passwords_encrypted** 列表中每一个泄露密码的泄露源类型. 此字段仅在输入参数**context** 包含了 **source_count**时生效 .
password_hashes	仅当**mode=search_leaked_password_with_hash_segment**时生效，根据输入**hash_segment**匹配到的泄露密码列表（hash格式）此参数不为空时表示检测到了与输入的6位hash值字符串匹配的泄露密码,说明此用户名和密码已经发生泄露。
quota	客户能够检测的账号数量.
quota_used	客户已经检测的账号数量
time_lapse	处理此API请求所用时间长度

API使用的代码示例：

JavaScript代码案例：

```
var url = "Http://<检测主机 IP>/vericlouds/API/index.php"
var api_key = 'XXXXXXXX'
var api_secret = 'xxxxxxxxxxxxxxxx'
var userid = 'this_is_test@gmail.com'
var password = '123456'
```

```
function mark_password(password) {
    var masked_password = '';
    if (password.length > 2) {
        masked_password = password.charAt(0)+Array(password.length-
1).join("*")+password.charAt(password.length-1);
    } else {
        masked_password = password;
    }
    return masked_password
}

$.post(url,
{mode:'search_leaked_password_with_userid',api_key:api_key,api_secret:api_secret,userid:u
serid}, function(data){
        var obj = jQuery.parseJSON( data );
        if (obj.result === 'succeeded') {
            var index = 1;
            var match = false;
            for (var password_encrypted in obj.passwords_encrypted) {
                password_encrypted =
obj.passwords_encrypted[password_encrypted].split(':');
                var enc = aesjs.utils.hex.toBytes(password_encrypted[0]);
                var iv = aesjs.utils.hex.toBytes(password_encrypted[1]);
                var aesCbc = new
aesjs.ModeOfOperation.cbc( aesjs.utils.hex.toBytes(api_secret), iv);
                var decrypted = aesjs.utils.utf8.fromBytes(aesCbc.decrypt(enc));
                decrypted = decrypted.slice(0, decrypted.length -
decrypted.charCodeAt(decrypted.length-1))
                if (mark_password(decrypted) == mark_password(password))
                    match = true;

                if (match) {
                    alert('true')
                    break;
                }
                index++;
            }
            if (!match) {
                alert('false')
            }
        } else {
            alert("Query failed! Reason: "+obj.reason);
```

```
        }
    });
```

Python代码样例：

```
import urllib, urllib2, json
from Crypto.Cipher import AES

unpad = lambda s : s[:-ord(s[len(s)-1:])]
def AESCipherdecrypt( key, enc ):
    enc, iv = enc.split(':')
    cipher = AES.new(key.decode("hex"), AES.MODE_CBC, iv.decode("hex") )
    return unpad(cipher.decrypt( enc.decode("hex") ))

url = " Http://<检测主机 IP>/vericlouds/API/index.php "
api_key = 'XXXXXXXX'
api_secret = 'xxxxxxxxxxxxxxxx'

def is_compromised(userid, password):
      reqdata = {'mode':'search_leaked_password_with_userid', 'api_key': api_key,
'api_secret': api_secret, 'userid': userid}
      reqdata = urllib.urlencode(reqdata)
      resp = urllib2.urlopen(urllib2.Request(url, reqdata)).read()
      resp = json.loads(resp)
      if resp['result'] != 'succeeded':
            print resp['reason']
            return None
      for pass_enc in resp['passwords_encrypted']:
            plaintext = AESCipherdecrypt(api_secret, pass_enc)
            if (len(password), password[0], password[-1]) == (len(plaintext),
plaintext[0], plaintext[-1]) :
                  return True
      return False

   print is_compromised('this_is_test@gmail.com', '123456')
```

4.主机管理和维护

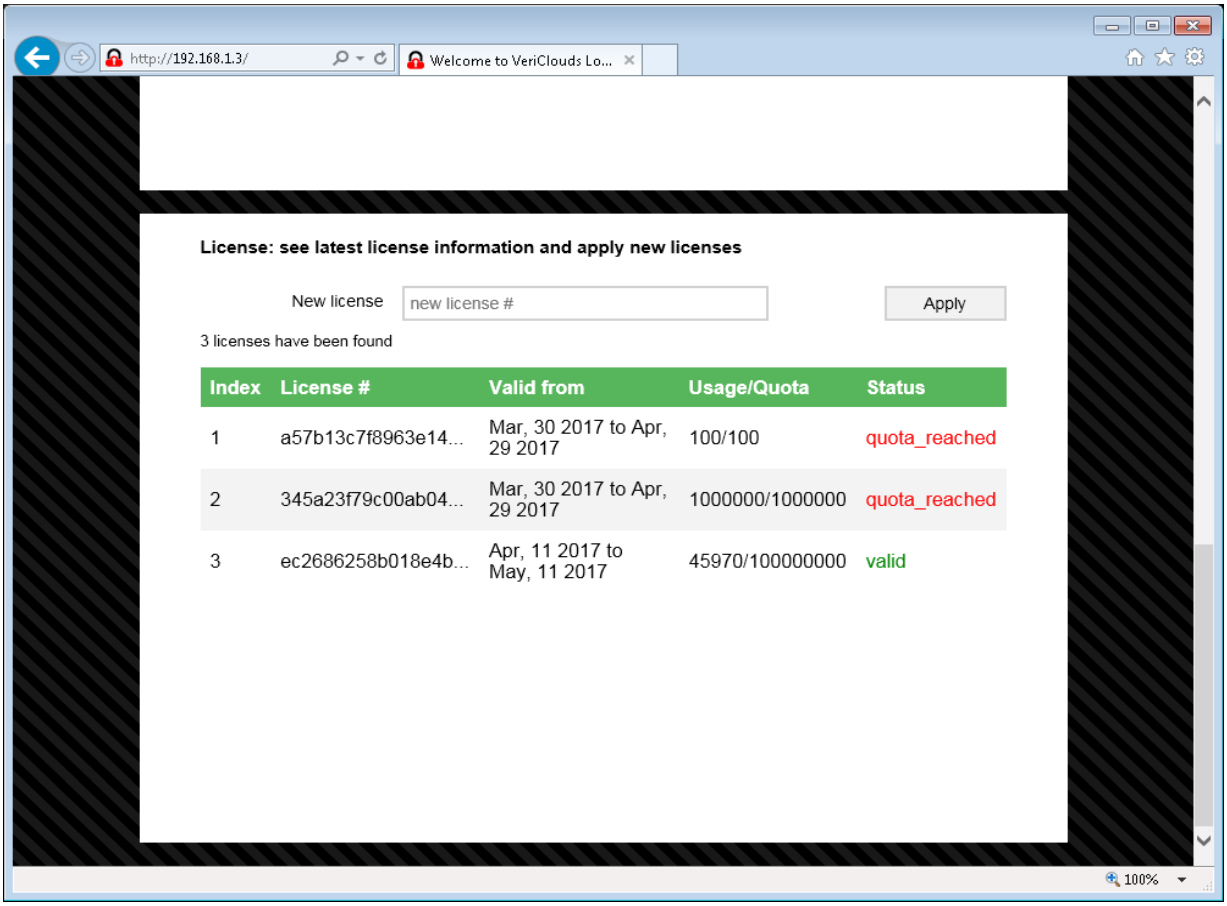
检测主机为用户提供web管理首页，用户可通过检测主机IP登录管理首页:

Http://<检测主机IP地址>/vericlouds 对检测主机进行管理使用。

4.1 许可证安装

账号泄露风险检测主机主要通过Rest API接口对外提供服务。为激活此服务功能，用户需要至少一个有效的使用许可证。检测主机的软件许可由一串字符串组成，用户可将此字符串复制后粘贴到管理首页的“new license”区域，点击“Apply”后即可看到反馈消息 “The license has been successfully applied”，标明许可证已经成功导入。

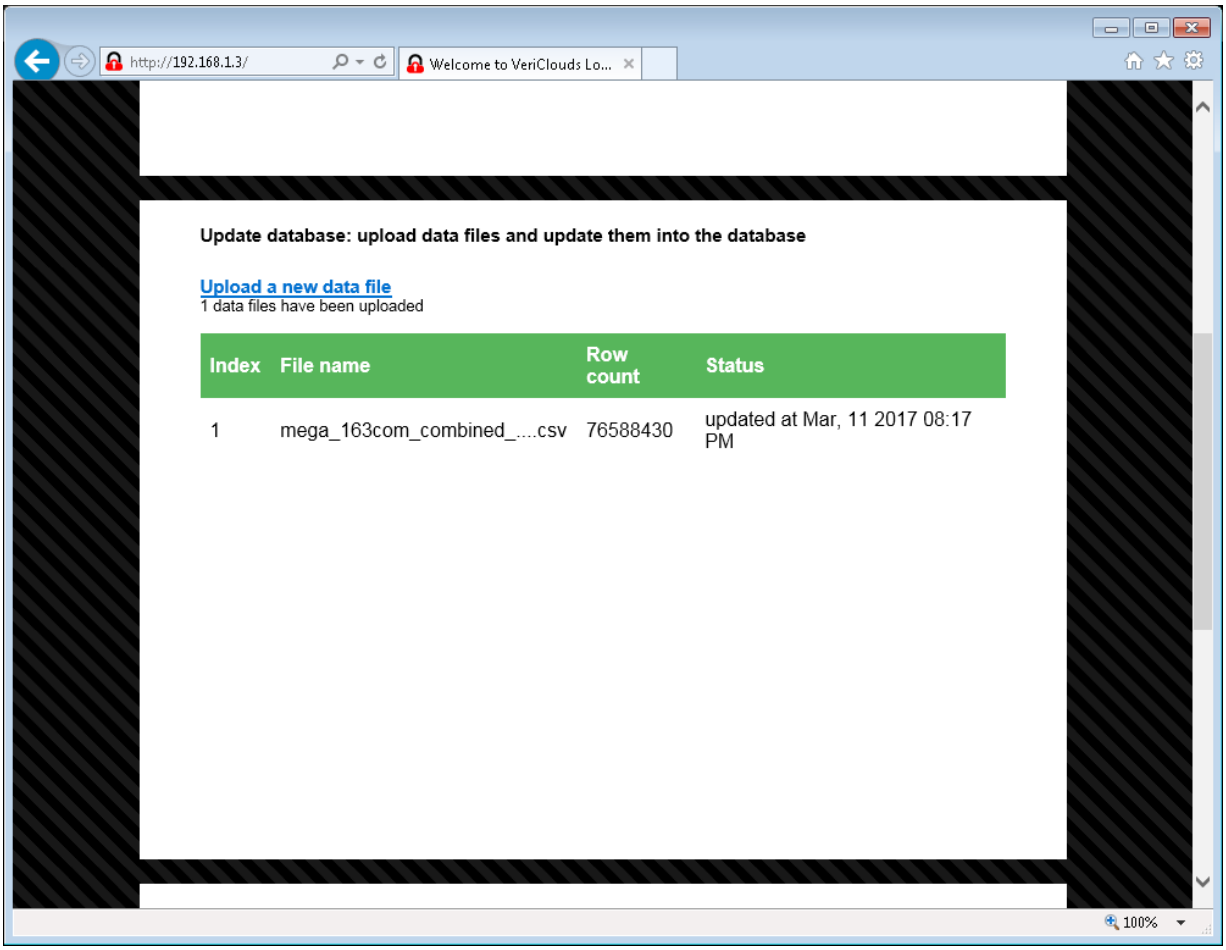
刷新管理首页后可以看到使用许可信息。检测主机的服务使用许可由使用时间和查询次数两个参数进行限制。当日期超过有效使用日期期限或者查询次数超过允许查询次数，二者之一满足的情况下，则表明许可证已到期，而检测主机将随机停止泄露检测服务。用户需要重新购买新的使用许可，在新许可导入生效后，检测主机将恢复正常服务。



管理首页的许可证区域内容

4.2 数据文件更新

为了保持检测主机中的泄露数据最新，反映最新的泄露情况，浩安信息将周期性地发送新的数据文件给客户进行检测主机数据库更新。这些更新数据文件通过“暗网（Black Net）”或其他渠道收集并进行整理。更新的数据文件采用csv文件格式，所包含的用户名和密码信息也全部进行加密，以避免二次泄露。检测主机的维护人员可在 “Data Updates”区域进行数据文件上传，上传完成后，刷新管理首页可看到已上传的文件信息，然后点击“Update to DB”按钮，上传的数据文件内容随即将导入检测主机的泄露数据库中。



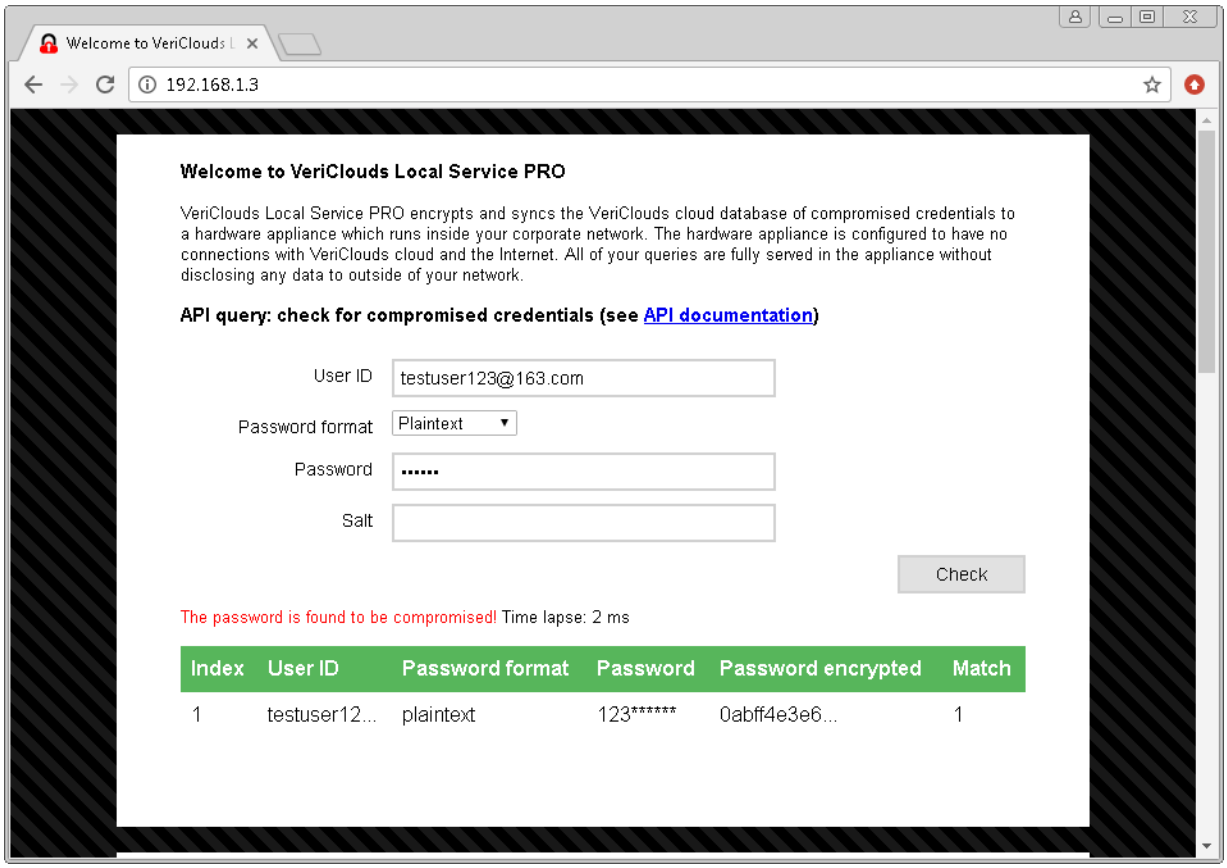
管理首页的数据更新区域内容

4.3 Rest API接口测试和说明

检测主机管理首页提供了泄露检测API的测试功能以及API使用说明和案例。

检测主机通过Rest API接口对外提供数据泄露检测服务。对每一个账号记录而言，用户可以

向检测主机发送一个Rest API请求，检测主机将决定此账号是否已经发生泄露。Rest API接口可以在管理首页的“API测试”区域进行测试，如图所示。



管理首页的接口测试区域内容

在测试时，用户需要提供一个测试账号名称(例如, 一个用户名字符串、邮箱地址或手机号码)，并选择此账号对应密码的形式(“Plaintext”，“MD5”，“SHA1”，等等.), 然后需要输入此测试账号的当前密码信息，最后点击 “Check”，一个 Rest API 请求将被加密后发送到检测主机。检测结果将直观地进行展示。如果用户进行测试的账号密码加了salt指，则测试时同样需要提供salt信息。

检测主机的Rest API接口对参数采用HTTP POST 而不是GET方式。

- 用户使用Rest API接口对自有账号密码进行检测比对时，支持明文和哈希格式的密码形式。针对密码为哈希格式时，检测主机支持目前业界被广泛使用的大部分哈希算法，包括 MD5, SHA1, SHA256,以及 BCRYPT，同时支持这些算法中加salt值的密码加密方式。
- 用户使用Rest API接口对自有账号密码进行检测比对时，支持账号形式包括字符串形式的用户名，邮箱地址、手机号码。

- 支持Rest API接口的以JSON形式进行数据响应

- Rest API接口对输入的用户账号密码数据进行泄露检测后反馈四种结果之一： **SAFE, COMPROMISED, SIMILAR, FAILED**.

✓ **SAFE** 表明检测主机的账号泄露数据库中没有账号与用户输入查询的账号匹配上，用户账号暂时没有泄露。

✓ **COMPROMISED** 表明检测主机的账号泄露数据库中存在与用户输入账号一样的账号，且密码也完全一致，由此说明用户输入的账号信息已经完全泄露。

✓ **SIMILAR** 表明检测主机的账号泄露数据库中存在与用户输入账号一样的账号，且密码部分一致，由此说明用户输入的账号信息虽暂未完全泄露，但存在泄露风险。

✓ **FAILED** 表明检测不成功，检测发生错误，将返回错误代码。

当用户需要利用检测主机的Rest API接口进行集成开发的时候，可以点击管理首页上的“API documentation”查看使用API进行代码编写的说明。

5. 账号泄露风险情报数据说明

- 检测云主机预置了数十亿计的账号泄露风险情报数据用以与用户的账号数据进行比对，这些已泄露账号风险情报数据来源于的互联网现实环境，一旦比对成功则意味着用户账号数据已经发生泄露，存在巨大的未知风险；
- 泄露账号风险情报数据库中的情报数据来源于互联网和暗网等环境，利用爬虫技术和专人从特定渠道进行收集和整理，并持续进行维护和更新；
- 泄露账号风险情报数据库中预置的账号和密码采用军工级的加密算法AES-256 CBC 模式预先进行加密，确保不发生二次泄露；
- 泄露账号风险情报数据库中的密码采用不同的随机向量进行加密，即使单一密码被破解也能确保其他密码安全；
- 泄露账号风险情报数据库中的账号采用同一随机向量进行加密，便于快速比对；
- 泄露账号风险情报数据库中的数据处理操作有专用硬件加密芯片管理和保护，只允许从检测主机授权的功能模块进行连接操作。

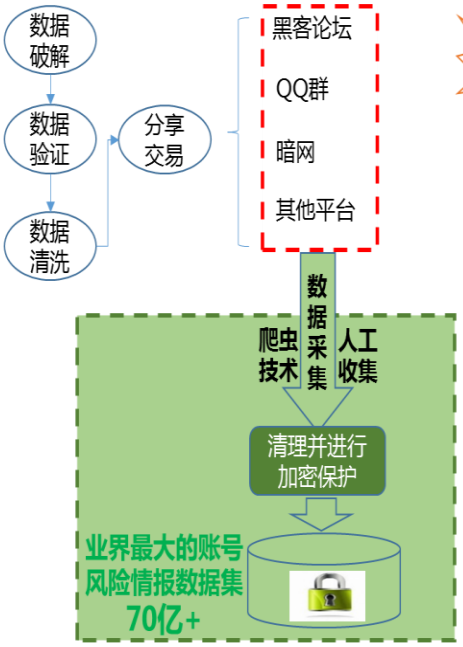
黑客入侵、内鬼窃取造成账号数据泄露频发

近年国内数据泄密事件一览：

2017年：雅虎30亿用户账号信息被泄露
http://news.163.com/17/1004/07/CVSV23MF00018AOQ.html
2016年：京东12G用户数据泄露
http://www.sohu.com/a/121228682_117869
2015年：网易邮箱过亿账号数据泄露
http://tech.qq.com/a/20151019/047851.htm

当当： 泄露约1200万用户资料：账号、密码、姓名、邮箱、住址、电话
大麦网：泄露约600万账号、密码、邮件地址
领英LinkedIn：泄露超过1亿的账号、密码数据
CSDN： 泄露640万个帐号、明文密码、电子邮件；
多玩： 泄露800万个帐号、MD5加密密码、部分明文密码，邮箱
178.COM：泄露188万个账号、MD5加密密码、明文密码、邮箱
天涯： 泄露4000万个帐号、明文密码、邮箱
人人网： 泄露500万个帐号、明文密码、电子邮件
UUU9.COM：泄露700万个帐号、MD5加密密码、全部明文密码、邮件
网易土木在线：泄露约4.3GB 137个文件：帐号、MD5加密密码
木蚂蚁： 泄露130万帐号、加密密码、
塞班论坛： 泄露约140万帐户、明文密码、邮箱
开心网：泄露约400万帐号、密码、salt、邮箱
凡客： 泄露约200万用户资料：账号、密码、姓名、邮箱、住址、电话
卓越： 泄露约200万用户资料：账号、密码、姓名、邮箱、住址、电话
。。。。。。

黑产数据链



利用泄露数据撞库破坏

在黑客利用泄露数据进行破坏攻击前，提前检测账号泄露风险，事前预防。

6. 补充说明

本文档提供了企业账号泄露风险检测云主机功能介绍和使用维护的详细说明。

检测云主机的功能定位在于发现存在泄露风险的账号，如弱密码账号、发生过历史泄露的账号、当前密码已经泄露的账号等，企业应该基于检测主机的检测结果开展进一步的处理工作，如通知用户修改密码、实施针对性的密码安全策略、用户登录时进行提醒等等。

此外，在本文档中仅介绍了泄露风险检测API接口的常用场景和功能，使企业用户能够利用检测主机中数十亿计的泄露账号情报数据，快速发现自身账号的泄露风险情况，提升信息安全管理能力。此API非常灵活且开放，其应用场景绝不限于本文档所描述的内容，用户可根据自身管理需要和业务场景，利用API接口进一步集成开发更多、更强大的安全管理功能。

7. 公司简介

浙江浩安信息技术有限公司（简称“浩安信息”）是一家科技型技术创新企业，于2018年由美国印第安纳大学网络安全博士、微软研究院资深信息安全研究员王锐博士创建，注册资金1000万元。

公司专注于账号安全、区块链安全、企业数据安全保护的产业方向，目前已经研发出业界第一款应用于企业账号泄露风险检测的主机产品，以及应用于区块链技术体系，用于保护用户私钥和交易签名的区块链安全主机产品。此外公司还提供账号安全管理服务、区块链开发服务等。

2018年初，浩安信息与浙江清华长三角研究院联合成立信息安全研究中心，共同推进信息安全技术研究和产品研发工作。

公司联系方式：

地址：浙江省嘉兴市加创路321号上海交大（嘉兴）科技园研发楼620室

电话：0573-82716672

邮箱：support@haoaninfo.com

企业微信公众号：haoaninfo