



,

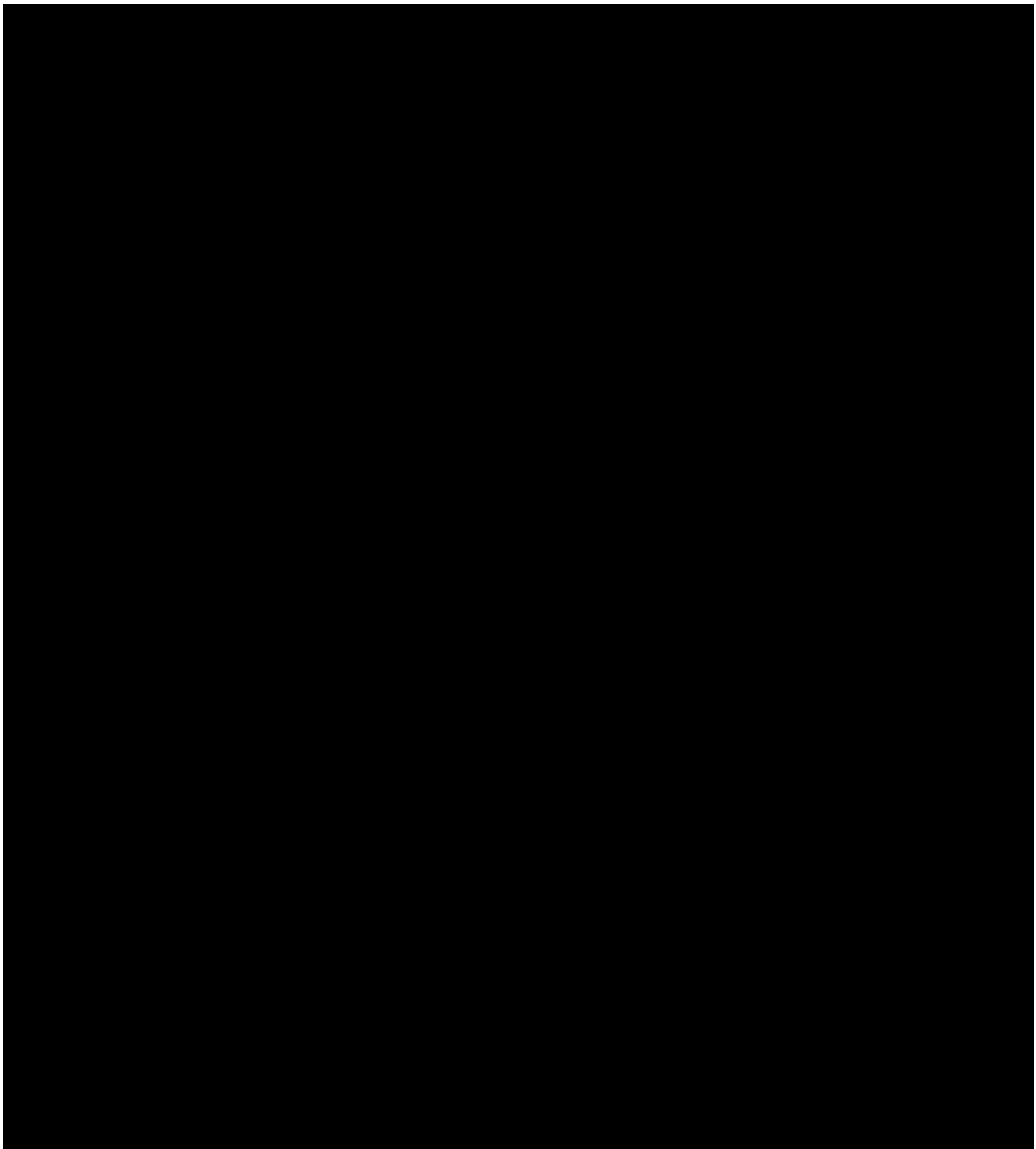
A

A A

A

[illegible]





```
1 array 226545696 forall % proc = D80D020
```

015C2BD0	00030000	ASCII "Actx "
015C2BD4	00000000	
015C2BD8	015D8748	
015C2BDC	015D5500	
2BE8	00000000	
2BE8	00000000	
2BE8	015D1E50	ASCII "array"
2BEC	00000000	

0xD80D020被错误的当成了proc处理函数类型对象的指针

0D80D020	00 00 00 00	30 D0 80 00	00 00 00 00	02 00 00 00	...
0D80D030	10 D0 80 00	02 00 00 00	3C D0 80 00	00 05 00 00	...
0D80D040	00 00 00 00	00 00 00 00	5C D0 80 00	00 00 03 00	...
0D80D050	00 00 00 00	F8 87 5D 01	A8 55 5D 01	3C D0 80 00	...
0D80D060	6C D0 80 00	00 00 00 00	F0 FF FF 7F	50 D0 80 00	...
0D80D070	00 00 00 00	F1 FF FF 7F	00 00 00 00	00 00 00 00	...

```

.text:6DC66D2A handleFun      proc near                                ; CODE XREF: sub_6DC4C297+551p
.text:6DC66D2A                                         ; sub_6DC4C317+881p ...
.text:6DC66D2A
.text:6DC66D2A var_4          = dword ptr -4
.text:6DC66D2A arg_0          = dword ptr 8
.text:6DC66D2A
.text:6DC66D2A push        ebp
.text:6DC66D2B mov         ebp, esp
.text:6DC66D2D push        ecx
.text:6DC66D2E push        ebx
.text:6DC66D32
.text:6DC66D35
.text:6DC66D36
.text:6DC66D37
.text:6DC66D37
.text:6DC66D39 mov         e
.text:6DC66D3A mov         [
.text:6DC66D3B test        e
.text:6DC66D3C jz          s
.text:6DC66D3D mov         e
.text:6DC66D3E mov         e
.text:6DC66D3F
.text:6DC66D40
.text:6DC66D41
.text:6DC66D42
.text:6DC66D43
.text:6DC66D44
.text:6DC66D45
.text:6DC66D46
.text:6DC66D47
.text:6DC66D48
.text:6DC66D49
.text:6DC66D4A
.text:6DC66D4B
.text:6DC66D4C
.text:6DC66D4D
.text:6DC66D4E
.text:6DC66D4F
.text:6DC66D50
.text:6DC66D51
.text:6DC66D52
.text:6DC66D53
.text:6DC66D54
.text:6DC66D55
.text:6DC66D56 loc_6DC66D56:      ; CODE XREF: handleFun+9D1j
.text:6DC66D57 push        offset dword_6DC66D58
.text:6DC66D58 lea         eax, [ebp+arg_0]
.text:6DC66D59 push        eax
.text:6DC66D5A call        _CxxThrowException

```


A

A

026A5F40	026A5F44	
026A5F44	6B5AB522	EPSIMP32.6B5AB522
026A5F48	6B5E9E30	EPSIMP32.6B5E9E30
026A5F4C	00000000	
026A5F50	00000000	
026A5F54	6B5E9E2F	EPSIMP32.6B5E9E2F
026A5F58	76ED5F18	ntdll.ZwProtectVirtualMemory
026A5F5C	026A6140	
026A5F60	FFFFFFFF	
026A5F64	026A6040	
026A5F68	026A6044	
026A5F6C	00000000	



6B5D1218	E8 46B0DFF	call EPSIMP32.6B5AC263	
6B5D121D	C745 D8 170000	mov dword ptr ss:[ebp-0x28],0x17	
6B5D1224	EB C9	jmp XEPSIMP32.6B5D11EF	
6B5D1226	8B4D F8	mov ecx,dword ptr ss:[ebp-0x8]	
6B5D1229	8B01	mov eax,dword ptr ds:[ecx]	
6B5D122B	FF50 10	call dword ptr ds:[eax],0x101	EPSIMP32.6B5E9E2F
6B5D122E	3BC7	cmp eax,0x7	
6B5D1230	7E 0A	jg 6B5D123A	
6B5D1232	8388: FF	cmp eax,0xFFFFFFFF	
6B5D1235	8B4D F8	mov ecx,dword ptr ss:[ebp-0x8]	
ds:[026A5F54]=6B5E9E2F (EPSIMP32.6B5E9E2F)			
地址	数值	注释	
026A5F40	026A5F44		
026A5F44	6B5AB522	EPSIMP32.6B5AB522	
026A5F48	6B5E9E30	EPSIMP32.6B5E9E30	

shellcode


```

026B682C 8D9B 00000000 mov ebx,edx
026B6834 BA 4D5A0000 lea ebx,dword ptr ds:[ebx]
026B6839 66:3913 cmp word ptr ds:[ebx],dx
026B683E 8B43 3C mov eax,dword ptr ds:[ebx+8]
026B6841 3D 80180000 jnb short 026B6851
026B6846 73 09 jnb short 026B6851
026B6848 813C18 50450000 jnb short 026B6854
026B684F 74 03 jnb short 026B6854
026B6851 43 jnb short 026B6834
026B6852 1E 19 jnb short 026B6834
026B6854 80CF FF jnb short 026B6834
026B6857 8B02 jnb short 026B6834
dx=5A4D
ds:[026B6D17]=5A4D

```

地址	HEX 数据	ASCII
026B6D17	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00	MZ? ...!...yy..
026B6D27	B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00	?.....@.....
026B6D37	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	f..
026B6D47	00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00	f..
026B6D57	0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68	■■?..??L?Th
026B6D67	69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F	is program canno

```

026B6CD4 33D2 xor edx,edx
026B6CD6 6A 00 push 0x0
026B6CD8 FFD2 call edx
026B6CDA 8B45 FC mov eax,dword ptr ss:[ebp-0x4]
026B6CDD 83C4 04 add esp,0x4
026B6CE0 6A 00 push 0x0
026B6CE2 6A 00 push 0x0
026B6CE4 50 push eax
026B6CE5 5ED7 call edi

```

```

00412D35 56 push esi
00412D36 57 push edi
00412D37 E8 FE010000 call 00412F3A
00412D3C 83F8 01 cmp eax,0x1
00412D3F 75 0C jnz short 00412D4D
00412D41 B9 686C4300 mov ecx,0x436C68
00412D44 8B43 3C mov eax,dword ptr ds:[ebx+8]
00412D47 3D 80180000 jnb short 00412D57
00412D4A 73 09 jnb short 00412D57
00412D4C 813C18 50450000 jnb short 00412D54
00412D4F 74 03 jnb short 00412D54
00412D51 43 jnb short 00412D3C
00412D52 1E 19 jnb short 00412D3C
00412D54 80CF FF jnb short 00412D3C
00412D57 8B02 jnb short 00412D3C

```

A

```

00412DB8 74 1D je short 00412DD7
00412DBA 6A 00 push 0x0
00412DBC 6A 01 push 0x1
00412DBE 57 push edi
00412DBF FFD0 call eax
00412DC1 E8 D0EDFFFF call 00411BA3
00412DC6 83F8 03 cmp eax,0x3
00412DC8 74 1D je short 00412DD7
00412DCB E8 11F0FFFF call 004123E1

```

00412E41	6A 01	push 0x1	WINWORD.exe 0041348C
00412E43	33D2	xor edx,edx	
00412E45	B9 E8974200	mov ecx,0x4297E8	
00412E4A	E8 82FEFFFF	call 00412CD1	
00412E4F	59	pop ecx	
00412E50	8BC8	mov ecx,eax	
00412E52	E8 14FDFFFF	call 00412B6B	
00412E57	85C0	test eax,eax	
00412E59	0F84 C8000000	je 00412F27	
00412E5F	8BC8	mov ecx,eax	
00412E61	E8 5CFDFFFF	call 00412BC2	

[illegible]

```
54 do
55 f
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
1
```

A A A

```

015 v0[1] = 00;
016 lpString2 = (LPCWSTR)decrypt((int)v0); // "apisecconnect.dll"
017 *v0 = aB0A;
018 v0[1] = 10;
019 lpName = (LPCWSTR)decrypt((int)v0); // "SYSTEM\\CurrentControlSet\\Services\\
020 * = "Software\\Microsoft\\Windows\\CurrentVersion\\Run";
021 [1] = 38;
022 = (const WCHAR *) "Microsoft\\Windows\\CurrentVersion\\Run"; // "Software\\Micro
023 *C = &v0[1];
024 [4] = 50;
025 = (const WCHAR *)decrypt((int)C); // "Office test\\
026 RegOpenKeyExW(HKEY_CURRENT_USER, v0, 0, REG_SZ, publicSub
027 RegCreateKeyExW(publicSub, v0, 0, REG_SZ, publicSub, v0, 0,

```

Office test

Special

Perf

Name	Type	Data
(Default)	REG_SZ	C:\User\ \AppData\Local\Temp\apisecconnect.dll