

# LatentBot – modularny i silnie zaciemniony bot

Archived: 2026-04-05 13:34:40 UTC

LatentBot jest widoczny w sieci od 2013 roku, a na początku października rozpoczął swoją drugą młodość – dodany został jako payload do bardzo aktywnego [exploit-kita Rig](#) i jest serwowany zamiennie z takim złośliwym oprogramowaniem jak: Cerber, CryptFile2, Gootkit czy Vawtrak.

Głównym wektorem infekcji dla użytkowników były dokumenty pakietu Microsoft Office zawierające exploity na podatności [CVE-2010-3333](#), [CVE-2012-0158](#), [CVE-2013-3906](#) oraz [CVE-2014-1761](#). Pliki zbudowane zostały za pomocą pakietu [Microsoft Word Intruder / MWISTAT](#).

Kampanie z tym złośliwkiem prowadzone były m.in. w USA, Kanadzie, Brazylii, Singapurze, Korei Południowej, Polsce oraz Zjednoczonych Emiratach Arabskich. Głównym adresatem byli przedstawiciele sektora finansowego oraz ubezpieczeniowego.

LatentBot na tle konkurencyjnego malware znacznie się wyróżnia:

- Dynamiczne deszyfrowanie stringów („autorski” algorytm) oraz nazw funkcji WinAPI i usuwanie ich po użyciu
- Możliwość nadpisania (uszkodzenia) MBR
- Modułowość (zaciemnione moduły przechowywane w rejestrze Windows)
- Wykorzystanie malware Pony 2.0 celem wykradania informacji oraz portfeli kryptowalut takich jak BitCoin, LiteCoin, Namecoin i wiele innych.

W poście zostanie opisana wersja rozpowszechniana obecnie za pomocą exploit-kita Rig (z pominięciem pierwszych stadiów procesu infekcji – więcej o nich można przeczytać [tutaj](#)).

## Analiza techniczna

Malware na początku swojego działania sprawdza wersję systemu operacyjnego ofiary oraz nazwę procesu rodzica. Działanie programu zostanie zakończone jeżeli maszyna działa pod kontrolą Windows Vista lub Windows Server 2008 lub proces rodzica ma nazwę różną od **explorer.exe** lub **svchost.exe**.

```
loc_42AC2D:
    lea    eax, [ebp+var_28]
    call   GetProcessName
    mov    eax, [ebp+var_28]
    lea    edx, [ebp+var_24]
    call   UpperCase
    mov    eax, [ebp+var_24]
    push  eax
    lea    edx, [ebp+var_30]
    mov    eax, offset suchost_exe ; "49UL870G9BAc2P"
    call   DecryptWinApiString

loc_42AC51:
    mov    eax, [ebp+var_30]
    lea    edx, [ebp+var_2C]
    call   UpperCase
    mov    edx, [ebp+var_2C]
    pop    eax
    call   StrCmp
    jz     short loc_42ACA1

    lea    eax, [ebp+var_38]
    call   GetProcessName
    mov    eax, [ebp+var_38]
    lea    edx, [ebp+var_34]
    call   UpperCase
    mov    eax, [ebp+var_34]
    push  eax
    lea    edx, [ebp+var_40]
    mov    eax, offset explorer_exe ; "uBGZFt2UKm+0bBe0"
    call   DecryptWinApiString
    mov    eax, [ebp+var_40]
    lea    edx, [ebp+var_3C]
    call   UpperCase
    mov    edx, [ebp+var_3C]
    pop    eax
    call   StrCmp
    jnz    short loc_42ACA6
```

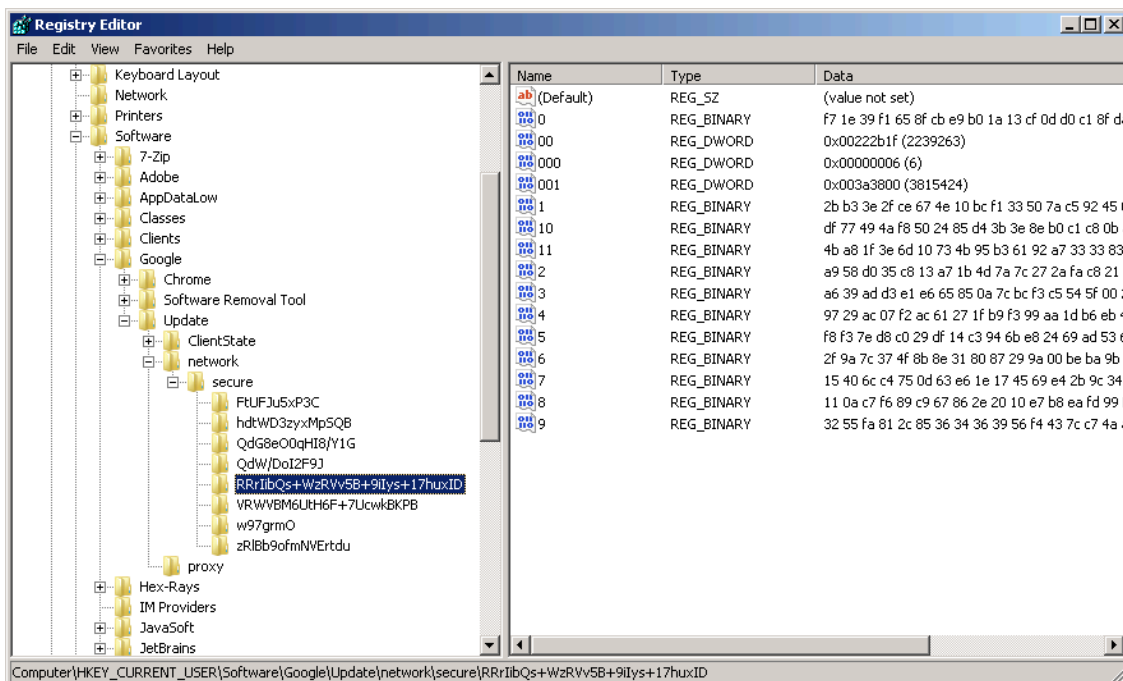
Po infekcji LatentBot sprawdza działanie serwera C&C wykonując proste zapytanie HTTP. W następstwie odpowiedzi, do serwera zarządzającego wysyłane są podstawowe informacje o zainfekowanej maszynie oraz generowany jest unikalny identyfikator ofiary. Bardzo ciekawą rzeczą i rzadko spotykaną jest sprawdzanie statusu baterii maszyny – pozwala określić czy bot zainstalowany jest na laptopie.

```

xchg ecx, [ebp+SystemPowerStatus.BatteryFullLifeTime]
push ebx
push esi
push edi
mov ebx, ecx
mov edi, eax
xor eax, eax
push ebp
push offset loc_1410C663
push dword ptr Fs:[eax]
mov fs:[eax], esp
lea eax, [ebp+SystemPowerStatus]
push eax ; lpSystemPowerStatus
call GetSystemPowerStatus
lea edx, [ebp+idgen_string]
mov eax, offset idgen_enc.Text
call DecryptWinApiString
mov eax, [ebp+idgen_string]
push eax
mov eax, ebx
call CheckPluginStatus
push eax
lea eax, [ebp+var_24]
call GetUserLocale0sVersionAndRandomValueAndParseString
mov edx, [ebp+var_24]
lea eax, [ebp+var_20] ; pvarg
call @Variants@@VarFromLStr$qqrr8TUarDatax17System@AnsiString ; Variants::_linkproc__ VarFromLStr(TUarData &,System::AnsiString)
lea eax, [ebp+var_20]
pop edx
call sub_1407B3B0
lea edx, [ebp+var_38]
mov eax, offset true_enc.Text
call DecryptWinApiString
mov edx, [ebp+var_38]
lea eax, [ebp+var_34] ; pvarg
call @Variants@@VarFromLStr$qqrr8TUarDatax17System@AnsiString ; Variants::_linkproc__ VarFromLStr(TUarData &,System::AnsiString)
lea eax, [ebp+var_34]
push eax
lea edx, [ebp+var_3C]
mov eax, offset 15Host_enc.Text
call DecryptWinApiString
mov eax, [ebp+var_3C]
push eax
mov eax, ebx
call CheckPluginStatus
    
```

Następnie złośliwe oprogramowanie pobiera moduły, udające pliki ZIP, i zapisuje je (oraz swoje binaria) w postaci zaszyfrowanej w rejestrze systemowym:

**HKCU\Software\Google\Update\network\secure\[zaszyfrowana\_nazwa\_modułu].**



Probka z naszego laboratorium zawiera następujące moduły:

- hdtWD3zyxMpSQB – Bot\_Engine
- QdW/DoI2F9J – Security
- RRrIibQs+WzRVv5B+9iIys+17huxID – Remote\_desktop\_service

- VRWVBM6UtH6F+7UcwkBKPB – **Vnc\_hide\_desktop**
- zRIBb9ofmNVErtdu – **Pony\_Stealer**
- FtUFJu5xP3C – **formgrabber**
- QdG8eO0qHI8/Y1G – **send\_report**
- w97grmO – **Socks**

W stosunku do starszych wersji LatentBot, analizowana wersja posiada nowe moduły o nazwach „**formgrabber**” oraz „**Socks**”.

## **Algorytm szyfrowania danych**

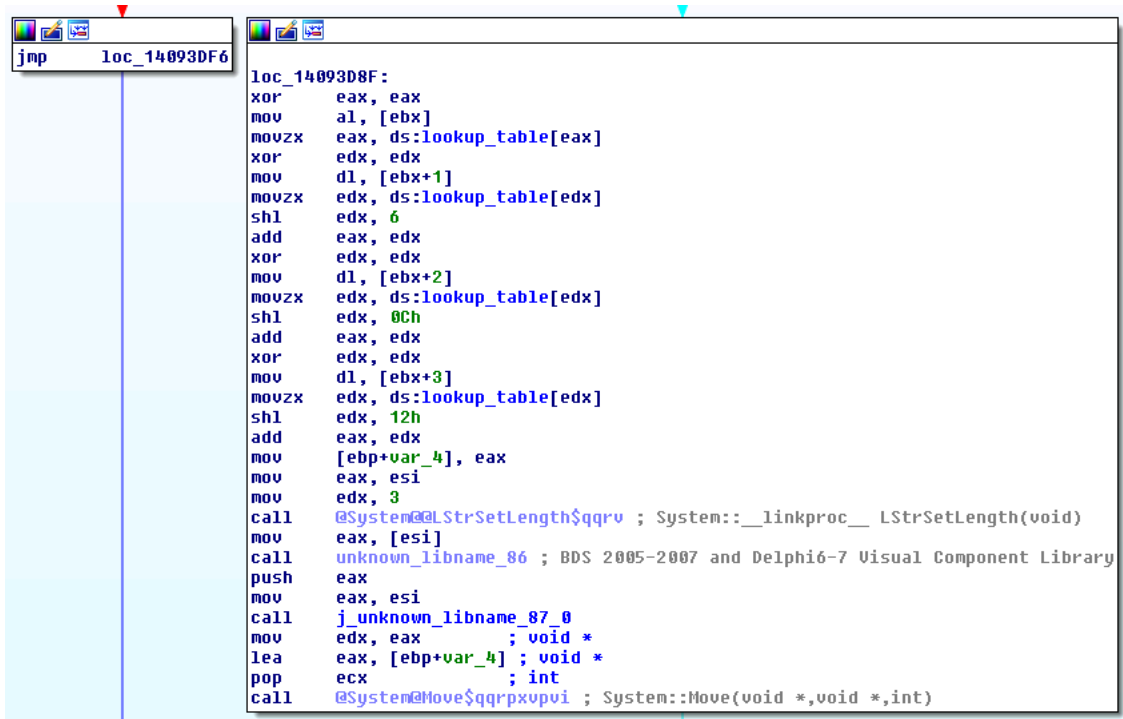
Szkodnik implementuje własny sposób szyfrowania danych:

1. Pierwszym krokiem jest kodowanie danych za pomocą tablicy bajtowej zaszytej w kodzie. Funkcja pobiera dane w czterobajtowych „porcjach” i w zależności od umiejscowienia bajtu oraz operacji (kodowanie lub dekodowanie) przesuwa odpowiednio w prawo \ lewo o wartości:

- \* 1. bajt – brak przesunięcia
- \* 2. bajt – 0x6h
- \* 3. bajt – 0xCh
- \* 4. bajt – 0x12h

2. Następnie każdy bajt jest poddany operacji XOR-owania wraz z modyfikatorem zależnym od rodzaju zasobu:

- \* 0xBB8h – nazwy funkcji Windows API
- \* 0x2328h – nazwy modułów w rejestrze OS
- \* 0x264Dh – dane wysyłane do serwera C&C
- \* 0x1918h – dane pobierane z serwera C&C



```
loc_14093D8F:
xor     eax, eax
mov     al, [ebx]
movzx  eax, ds:lookup_table[eax]
xor     edx, edx
mov     dl, [ebx+1]
movzx  edx, ds:lookup_table[edx]
shl     edx, 6
add     eax, edx
xor     edx, edx
mov     dl, [ebx+2]
movzx  edx, ds:lookup_table[edx]
shl     edx, 0Ch
add     eax, edx
xor     edx, edx
mov     dl, [ebx+3]
movzx  edx, ds:lookup_table[edx]
shl     edx, 12h
add     eax, edx
mov     [ebp+var_4], eax
mov     eax, esi
mov     edx, 3
call   @System@@LStrSetLength$qqrv ; System::_linkproc__LStrSetLength(void)
mov     eax, [esi]
call   unknown_libname_86 ; BDS 2005-2007 and Delphi6-7 Visual Component Library
push   eax
mov     eax, esi
call   j_unknown_libname_87_0
mov     edx, eax ; void *
lea    eax, [ebp+var_4] ; void *
pop     ecx ; int
call   @System@Move$qqrxpupi ; System::Move(void *,void *,int)
```

## Moduły

### Bot\_Engine

„**Bot\_Engine**” jest modulem bazowym – odpowiada za wstępny proces weryfikacji środowiska, pobranie pozostałych modułów oraz komunikację z C&C. Po uruchomieniu złośliwego pliku wysyłane są cztery żądania ICMP Echo (popularny „ping”) do serwera zarządzającego.

Po instalacji i walidacji środowiska wysyłane są informacje o nowym bocie (parametry żądania HTTP GET):

- o **idgen** – unikalny identyfikator ofiary
- o **isAv** – czy został wykryty antywirus przez moduł Security oraz jego identyfikator liczbowy
- o **isWinVer** – wersja OS
- o **isX64** – bitowość systemu operacyjnego
- o **isVer** – wyróżnik wersji LatentBota
- o **isPcNetName** – nazwa komputera ofiary
- o **isPcUserName** – nazwa użytkownika
- o **isCountry** – język systemu operacyjnego ofiary
- o **isJava** – czy Java jest zainstalowana w OS
- o **isbk** – czy ofiara ma zainstalowany bootkit (opcja wyłączona)
- o **isKeyLog** – status keyloggera
- o **isaccessadmin** – flaga czy malware jest uruchomiony z podwyższonymi uprawnieniami

- **isNote** – wykrycie stanu baterii komputera (sprawdzenie czy komputer jest notebookiem)
- **isTracertspeed** – czas odpowiedzi C2 do hosta
- **isUptime** – uptime OS
- **isAntiSB** – niewykorzystywany (zawsze wartość „0”)
- **isBitc** – czy znaleziono portfele kryptowalut na komputerze ofiary

Dodatkowo moduł zajmuje się obsługą poniższych poleceń od C&C:

- **restart** – restart maszyny
- **shutdown** – wyłączenie OS
- **logoff** – wylogowanie użytkownika
- **stop\_engine\_and\_plugins** – zatrzymanie modułu głównego oraz pluginów
- **plugin\_stop\_all** – zatrzymanie wszystkich pluginów
- **plugin\_stop** – zatrzymanie działania wybranego pluginu
- **plugin\_stop\_and\_uninstall** – zatrzymanie działania wybranego pluginu oraz jego odinstalowanie
- **plugin\_stop\_auto** – automatyczne zatrzymanie pluginu
- **plugin\_start** – uruchomienie modułu
- **plugin\_start\_auto** – automatyczne uruchomienie pluginu
- **plugin\_restart** – ponowne uruchomienie pluginu
- **clear\_cookies** – wyczyszczenie ciasteczek w Chrome / Firefox / IE oraz Operze
- **uninstall\_all** – usunięcie Bot\_Engine oraz reszty pluginów

## Security

„Security” jest modułem wyszukującym oprogramowanie antywirusowe i narzędzia analityczne na komputerze ofiary. Sprawdzane jest istnienie poniższych folderów na dysku:

Documents and Settings\All Users\Application Data\Agnitum  
Documents and Settings\All Users\Application Data\avg10  
Documents and Settings\All Users\Application Data\avg8  
Documents and Settings\All Users\Application Data\avg9  
Documents and Settings\All Users\Application Data\Avira  
Documents and Settings\All Users\Application Data\Doctor Web  
Documents and Settings\All Users\Application Data\ESET  
Documents and Settings\All Users\Application Data\f-secure  
Documents and Settings\All Users\Application Data\G DATA  
Documents and Settings\All Users\Application Data\Kaspersky Lab\  
Documents and Settings\All Users\Application Data\McAfee  
Documents and Settings\All Users\Application Data\Microsoft\Microsoft Antimalware  
Documents and Settings\All Users\Application Data\PC Tools  
Documents and Settings\All Users\Application Data\Symantec  
Documents and Settings\All Users\Application Data\Trend Micro

Documents and Settings\All Users\AVAST Software  
Documents and Settings\NetworkService\Local Settings\Application Data\F-Secure  
Program Files\Agnitum  
Program Files\Alwil Software  
Program Files\AVAST Software  
Program Files\AVG  
Program Files\Avira  
Program Files\BitDefender9  
Program Files\Common Files\Doctor Web  
Program Files\Common Files\G DATA  
Program Files\Common Files\PC Tools  
Program Files\DrWeb  
Program Files\ESET  
Program Files\F-Secure Internet Security  
Program Files\FRISK Software  
Program Files\Kaspersky Lab  
Program Files\McAfee  
Program Files\Microsoft Security Essentials  
Program Files\Norton AntiVirus  
Program Files\Panda Security  
Program Files\PC Tools Internet Security  
Program Files\Symantec  
Program Files\Trend Micro  
Program Files\Vba32

Poniżej fragment głównej funkcji w module:

```
push ebx
mov ebx, eax
xor eax, eax
push ebp
push offset loc_14098A40
push dword ptr fs:[eax]
mov fs:[eax], esp
mov eax, ebx
call @System@@LStrClr$qqrv ; System::__linkproc__ LStrClr(void *)
lea edx, [ebp+System::AnsiString]
mov eax, 3
call GetWindowsDirectory
lea eax, [ebp+System::AnsiString]
push eax
lea edx, [ebp+var_8]
mov eax, offset Program_Files_Alwil_Software_enc.Text
call DecryptWinApiString
mov edx, [ebp+var_8]
pop eax
call @System@@LStrCat$qqrv ; System::__linkproc__ LStrCat(void)
mov eax, [ebp+System::AnsiString] ; System::AnsiString
call @Sysutils@DirectoryExists$qqrx17System@AnsiString ; Sysutils::DirectoryExists(System::AnsiString)
test al, al
jnz short loc_140980A8
```

```
lea edx, [ebp+var_C]
mov eax, 3
call GetWindowsDirectory
lea eax, [ebp+var_C]
push eax
lea edx, [ebp+var_10]
mov eax, offset Documents_and_Settings_All_Users_AVAST_Software_enc.Text
call DecryptWinApiString
mov edx, [ebp+var_10]
pop eax
call @System@@LStrCat$qqrv ; System::__linkproc__ LStrCat(void)
mov eax, [ebp+var_C] ; System::AnsiString
call @Sysutils@DirectoryExists$qqrx17System@AnsiString ; Sysutils::DirectoryExists(System::AnsiString)
test al, al
jnz short loc_140980A8
```

Złoźnik potrafi wykryć następujące narzędzia do potencjalnej analizy malware:

- a2HiJackFree
- Ad-Aware
- Advanced Spyware Remover Pro
- Arovax Shield
- CounterSpy
- EffeTech HTTP Sniffer
- gmer
- HijackThis
- HTTPAnalyzerFullV5
- Kerish Doctor
- Malwarebytes' Anti-Malware
- MalwareSecure
- OSAM Autorun Manager
- OSSS Proactive / Firewall
- PC Tools Spyware Doctor
- Process Hacker
- Protector Plus 2013
- RemoveIT Pro 2014 Ultra
- SecureAnywhere
- Security Stronghold Active Shield

Spybot

Spyware Terminator 2012

SUPERAntiSpyware

SysInspector

Trend Micro AntiSpyware

True Sword 5

viewtcp

VIRUSfighter

WinPatrol

### **Vnc\_hide\_desktop**

Najbardziej rozbudowaną częścią tego malware'u jest moduł VNC o funkcjonalności znacząco wykraczającej poza zdalne połączenie do komputera ofiary.

Umożliwia atakującemu:

- Usunięcie malware'u z systemu operacyjnego oraz uszkodzenie MBR
- Włączenia serwera VNC na komputerze ofiary
- Logowanie naciśnień klawiszy (keylogger)
- Wyszukiwanie portfeli BitCoin, Electrum oraz MultiBit
- Zablokowanie użytkownikowi dostępu do myszy
- Wysyłanie żądań ICMP
- Wylogowanie użytkownika lub wyłączenie OS

Na początku działania modułu jest sprawdzane działanie innych serwerów VNC uruchomionych na zainfekowanym systemie:

- **tvnserver.exe** – TightVNC Software
- **winvnc.exe** – UltraVNC Software
- **vncserver.exe** – RealVNC Software
- **vncservice.exe** – RealVNC Software

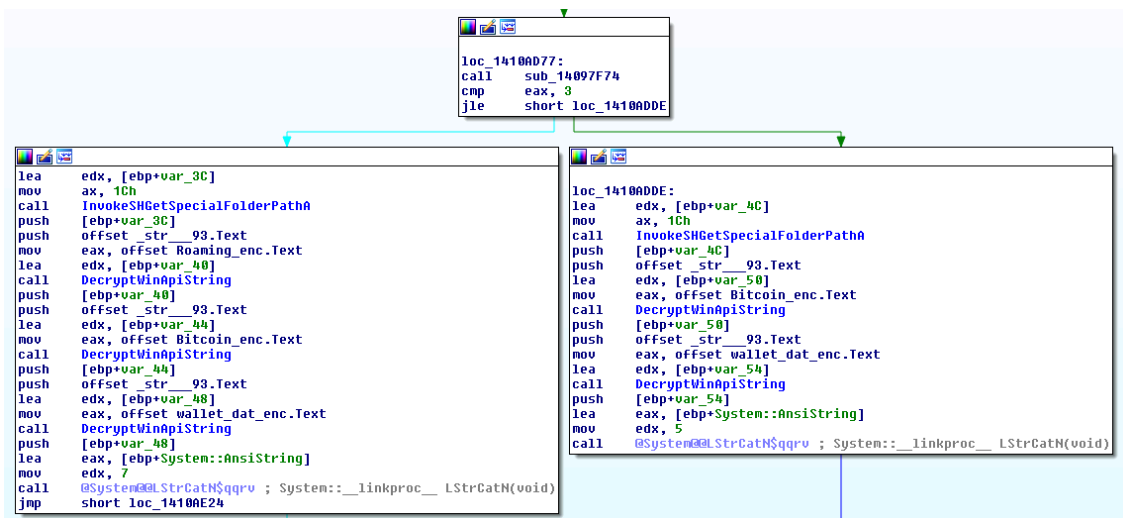
```
push    ecx
push    ebx
push    esi
push    edi
xor     eax, eax
push    ebp
push    offset loc_1410DB78 ; uExitCode
push    dword ptr fs:[eax] ; hProcess
mov     fs:[eax], esp
lea    edx, [ebp+var_24]
mov     eax, offset tunserver_exe_enc.Text
call   DecryptWinApiString
mov     eax, [ebp+var_24]
call   IsProcessPresent
lea    edx, [ebp+var_28]
mov     eax, offset winvnc_exe_enc.Text
call   DecryptWinApiString
mov     eax, [ebp+var_28]
call   IsProcessPresent
lea    edx, [ebp+var_2C]
mov     eax, offset vncserver_exe_enc.Text
call   DecryptWinApiString
mov     eax, [ebp+var_2C]
call   IsProcessPresent
lea    edx, [ebp+var_30]
mov     eax, offset vncservice_exe_enc.Text
call   DecryptWinApiString
mov     eax, [ebp+var_30]
call   IsProcessPresent
cmp    ds:dword_1411D87C, 0
jz     short loc_1410D92B

mov     eax, ds:dword_1411D87C
call   TerminateProcess_0
test   al, al
jz     short loc_1410D92B
```

Następnie oczekuje na polecenia:

- **killosanduninstall** – usunięcie szkodnika z systemu, nadpisanie MBR i restart systemu operacyjnego.
- **ClearTemp** – usunięcie plików tymczasowych wykorzystywanych przez malware
- **fm\_compress** – kompresja plików pobieranych z OS ofiary
- **fm\_compress\_getstat** – pobranie rozmiaru plików
- **fm\_test** – test funkcjonalności File Manager
- **fm\_get\_folder** – pobieranie folderu / pliku od ofiary
- **newvnc** – nowy serwer VNC (wstrzyknięcie do procesu svchost.exe)
- **EWX\_REBOOT** – restart OS ofiary
- **EWX\_LOGOFF** – wylogowanie użytkownika
- **EWX\_SHUTDOWN** – wyłączenie komputera

- **closeonesessions** – zamknięcie jednej sesji RDS
- **disablerds** – wyłączenie modułu Remote Desktop Service
- **getinstallpluginlist** – wysłanie do C&C modułów bota
- **uninstallbot** – usunięcie malware z systemu operacyjnego
- **startkey** – uruchomienie keyloggera
- **stopkey / stopkeylog** – zatrzymanie keyloggera i usunięcie zebranych danych
- **sendkey** – wysłanie danych keyloggera do C&C
- **sendfg** – wysłanie logów do C&C
- **clearkeylog** – usunięcie plików keyloggera z OS ofiary
- **findgold** – wyszukanie portfeli BitCoin, Electrum, MultiBit
- **explorer\_restart** – restart procesu explorer.exe



## Remote\_desktop\_service

Umożliwia zdalny dostęp do komputera ofiary za pomocą protokołu [RDP](#). De facto duplikuje funkcjonalność modułu VNC.

Patrząc z poziomu możliwości wtyczki i różnic pomiędzy wersjami LatentBota, można stwierdzić że rozwój tej części malware został zarzucony, a jego rolę przejęło VNC – zapewne przestępcom chodziło o mniejszą wykrywalność działań w systemie operacyjnym ofiary.

## Pony\_Stealer

Ten moduł zawiera inny malware – **Pony Stealer 2.0**. Złośliwe oprogramowanie z tej rodziny jest często wykorzystywane przez przestępców z powodu [wycieku kodu źródłowego](#), zarówno wersji 1.9, jak i 2.0.

Autor Latentbota skupił się głównie na wykradaniu portfeli kryptowaluty BitCoin, zbieraniu danych zapisanych w przeglądarkach internetowych oraz klientach poczty. **Bardzo ciekawą właściwością modułu jest to, że uruchamiany jest tylko raz, podczas infekcji.**

## form\_grabber & Socks

Dwa niewielkie moduły – wykradanie danych wpisywanych przez użytkowników w formularzach, oraz wystawienie Socks proxy na komputerze ofiary.

### Co zrobić w przypadku infekcji?

Przede wszystkim zalecamy zmianę wszystkich haseł do zasobów (poczta, serwisy społecznościowe, systemy transakcyjne) wykorzystywanych na zainfekowanej maszynie – warto skorzystać z menadżera haseł do zarządzania nimi np. [KeePassa](#).

Hasła należy zmienić z innego komputera niż zainfekowana maszyna. Na zainfekowanej maszynie trzeba przeprowadzić pełne skanowanie programem AV, a w skrajnym wypadku będziemy zmuszeni do **przeinstalowania systemu operacyjnego**.

### Skróty kryptograficzne analizowanych próbek i reguła YARA

```
e52b4d2d6c26891794d1eaa3ed81471870fd594b8d624a0826fc1e8eb9cc13fa
9b2699969896d0b301ab47e2f2f7f2051534ea526d862d75f4cda83b29408348
127285f3fb4b200dc8f47cfdcc8bceedd52e77df1bd68a7d3eeb0996d50ecef
```

```
rule latentbot : trojan
```

```
{
```

```
meta:
```

```
author="kamilf"
```

```
strings:
```

```
$encrypted_plugins_reg_loc =
```

```
"YRjMiR5pJQ2BYQGnxrtHJr/rc1ldUMq+LwntFlv2clCGXRO+WLP"
```

```
$encrypted_autostart_reg_loc =
```

```
"YRjMiR5pJQ2BS01054IZ+IU8u00RCk2L9tm+lACTf28OI7vow9xZfWqV7V0q"
```

```
$encrypted_id_reg_loc = "YRjMiR5pJQ2BeUoQ648eI9DVfta9CWKqhycjWD"
```

```
$decrypt_strings_and_call = { 89 ?? 8D [2] B8 [4] E8 [4] 8D [2] 50 8D [2] B8 [4] E8 [4] 8B [2] 58
E8 [4] 8B [2] 8B ?? E8 [4] E8 [4] EB ?? }
```

```
$encryption_decryption_core = { 88 [3] 8B [2] 0F [4] 66 [2] 66 [2] 6D CE 66 ?? BF 58 }
```

```
$lookup_table = { 3E 00 00 00 3F 34 35 36 37 38 39 3A 3B 3C 3D [8] 01 02 03 04 05 06 07 08 09
0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 }
```

```
$lookup_table_byte_shift = { C1 ?? 06 [14] C1 ?? 0C [14] C1 ?? 12 }
```

```
condition:
```

```
4 of ($encrypted_id_reg_loc, $encrypted_autostart_reg_loc, $encrypted_plugins_reg_loc,
```

```
$lookup_table_byte_shift, $lookup_table, $encryption_decryption_core) and
```

```
#decrypt_strings_and_call &gt; 20
```

```
}
```

Source: <https://www.cert.pl/news/single/latentbot-modularny-i-silnie-zaciemniony-bot/>