ECHO

CYBER THREAT INTELLIGENCE

2024

TURLA

ANALIZ RAPORU

@echocti @echocti echocti.com

İçindekiler

Yönetici Özeti

Bu rapor, 2004 yılından bu yana faaliyet gösteren ve Rusya devleti destekli olduğuna inanılan Turla siber saldırı grubunun detaylı bir analizini sunmaktadır. Turla, başlangıçta batı ülkelerini hedef alsa da zamanla faaliyet alanını genişleterek birçok bölgeyi de kapsamına almıştır.

Bu rapor, Turla siber saldırı grubunun çeşitli kampanyalarını ve bu kampanyaların hedeflerini incelemektedir. Grubun genellikle halka açık ve özel sektör kuruluşlarına yönelik kimlik avı saldırıları, zararlı yazılımların dağıtımı operasyonları gibi çeşitli saldırı stratejileri kullandığı tespit edilmiştir.

Özellikle vurgulanması gereken bir nokta, Turla grubuna ait olduğu düşünülen bir arka kapı yazılımıdır. TinyTurla-NG olarak Cisco tarafından adlandırılan bu backdoor yazılımı, ilerleyen günlerde kurum ve kuruluşlar için tehdit oluşturması kuvvet ile muhtemeldir.

Sonuç olarak, Turla grubunun sürekli evrim geçiren saldırı stratejileri, kurumsal ve bireysel kullanıcılar için ciddi bir tehdit oluşturmaktadır. Bu raporun amacı, Turla grubunun faaliyetleri ve hedefleri hakkında bir anlayış sağlamak ve ilgili taraflara bu tür siber saldırılara karşı korunma ve önleyici tedbirler alma konusunda yol göstermektir.

Turla Grup Profili

Turla, Rusya Federal Güvenlik Servisi (FSB) ile ilişkilendirilen bir siber casusluk tehdit grubudur. Bu grup, 2004 yılından beri en az 50 ülkede faaliyet göstererek hükümet, elçilikler, ordu, eğitim, araştırma ve ilaç şirketleri gibi çeşitli sektörlere saldırılar düzenlemiştir. Turla, watering hole ve spearphishing kampanyaları yürütmesi ve Uroburos gibi şirket içi araçlardan ve kötü amaçlı yazılımlardan faydalanmasıyla bilinmektedir.

Turla'nın saldırılarında, NATO ve AB enerji diyaloğu gibi terimlerin yer aldığı tespit edilmiştir. Saldırıların kim tarafından gerçekleştirildiğini net bir şekilde tespit etmek zor olsa da, bazı bilgisayar korsanlarının Rusça isimler ve dil kullandığı görülmüştür. Turla, farklı kötü amaçlı yazılımlar kullanarak ABD, Avrupa Birliği, Ukrayna ve Asya'daki kuruluşları hedef almaktadır.

Son zamanlarda Turla'nın yeni bir siber casusluk kampanyası tespit edilmiştir. Bu kampanyada Turla, TinyTurla-NG ve TurlaPower-NG kötü amaçlı yazılım ailelerini kullanarak sivil toplum kuruluşlarına saldırılar düzenlemiştir. Operatörler, güvenlik açığı bulunan WordPress web sitelerini kullanarak komuta ve kontrol uç noktalarını ihlal etmiş ve PowerShell ve komut satırını kullanarak kötü amaçlı yazılımı dağıtmıştır. (Cisco, TurlaNG)

Turla'nın saldırıları hala devam etmekte olup hedefleri gözetlemek ve veri çalmak amacıyla faaliyet göstermektedir. Bu son kampanya, Rusya'nın stratejik ve siyasi hedeflerine destek vermek amacıyla Turla'nın hedeflerini genişlettiği bir işaret olarak değerlendirilmektedir. Bu durum, Turla'nın siber casusluk faaliyetlerine devam ettiğini ve uluslararası alanda güvenlik tehdidi oluşturmaya devam ettiğini göstermektedir.

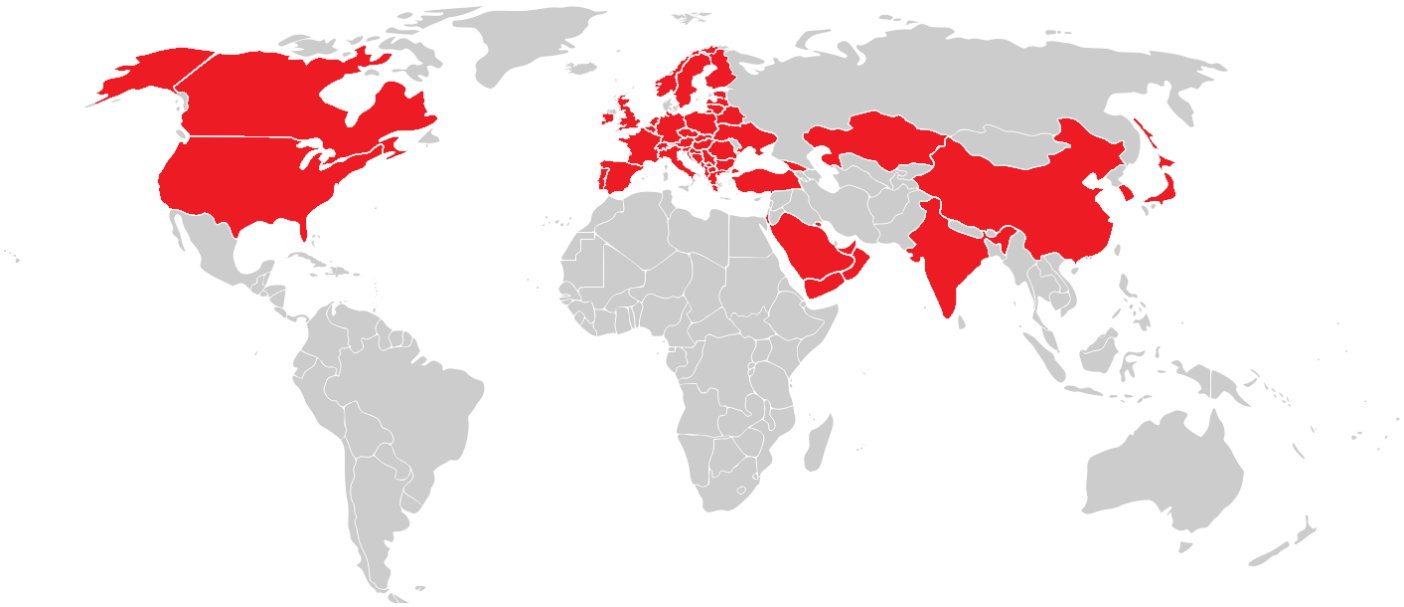

Figure 1 Targeted Countries

Teknik Analiz

Turla-NG Backdoor Analizi

SHA256	d6ac21a409f35a80ba9ccfe58ae1ae32883e44ecc724e4ae8289e7465ab2cf40
MD5	0f2e9f501ca9780eff309b7022c9b01a
File Type	PE64 - DLL

Table 1 File Informations

```
void __fastcall ServiceMain(__int64 a1, LPCWSTR *a2)
{
  SERVICE_STATUS_HANDLE v2; // rax
  char (**v3)(); // rax
  unsigned int ThrdAddr[4]; // [rsp+30h] [rbp-28h] BYREF
  _Thrd_t v5; // [rsp+40h] [rbp-18h] BYREF

  v2 = RegisterServiceCtrlHandlerW(*a2, HandlerProc);
  qword_1800460D0 = v2;
  if ( v2 )
  {
    ServiceStatus.dwCurrentState = 4;
    if ( SetServiceStatus(v2, &ServiceStatus) )
    {
      v3 = operator new(8ui64);
      if ( v3 )
        *v3 = Malware_Main;
      *ThrdAddr = beginthreadex(0i64, 0, sub_18000DC00, v3, 0, &ThrdAddr[2]);
      if ( !*ThrdAddr )
      {
        ThrdAddr[2] = 0;
        std::_Throw_Cpp_error(6);
      }
      if ( !ThrdAddr[2] )
        std::_Throw_Cpp_error(1);
      if ( ThrdAddr[2] == GetCurrentThreadId() )
        std::_Throw_Cpp_error(5);
      v5 = *ThrdAddr;
      if ( Thrd_join(&v5, 0i64) )
        std::_Throw_Cpp_error(2);
      *ThrdAddr = 0i64;
      Sleep(0x3E8u);
      if ( ThrdAddr[2] )
```

Figure 2 Service Main Function

Yapılan incelemelerde bir servis oluşturma işlemi tespit edilmiştir.

```
1  char Malware_Main()
2  {
3    char result; // al
4    char FileMappingStruct[48]; // [rsp+20h] [rbp-3E8h] BYREF
5    configuration cfg; // [rsp+50h] [rbp-3B8h] BYREF
6
7    get_predefined_campaign_id(FileMappingStruct);
8    result = Setup_Create_File_Mapping(FileMappingStruct);
9    if ( result )
10   {
11     ConfigInitializingEventCreation(&cfg);
12     Checking_OS_Starting_Threads(&cfg);
13     return CleanUp(&cfg);
14   }
15   return result;
16 }
```

Figure 3 Malware Main

Detaylıca incelendiğinde, ana zararlı fonksiyon tespit edilmiştir. *get_predefined_campaign_id* fonksiyonu, önceden tanımlanan bir belirteci yüklemektedir. *ConfigInitializingEventCreation* ise gerekli başlangıç işlemlerinin gerçekleştiği ve bir event HANDLE yapısının oluşturulduğu fonksiyondur.

```
1  int __fastcall Checking_OS_Starting_Threads(configuration *a1)
2  {
3    thread_struct *thread_1; // rax
4    thread_struct *thread_2; // rax
5    int result; // eax
6    unsigned int ThrdAddr[4]; // [rsp+30h] [rbp-168h] BYREF
7    _Thrd_t threads_handles; // [rsp+40h] [rbp-158h] BYREF
8    unsigned int v8; // [rsp+58h] [rbp-140h] BYREF
9    struct _OSVERSIONINFOW VersionInformation; // [rsp+60h] [rbp-138h] BYREF
10
11   GetPowershellVersion(a1);
12   VersionInformation.dwOSVersionInfoSize = 276;
13   if ( GetVersionExW(&VersionInformation)
14     && (VersionInformation.dwMajorVersion == 5
15       || VersionInformation.dwMajorVersion == 6 && VersionInformation.dwMinorVersion < 2) )
16   {
17     a1->is_OS_Version_Win7_or_older = 1;
18   }
```

Figure 4 OS Checking

İşletim sistemi ve powershell versiyonlarının çekildiği tespit edilmiştir.

```
19  thread_1 = operator new(0x10ui64);
20  if ( thread_1 )
21  {
22    thread_1->args = a1;
23    thread_1->thread_function = thread_1_function;
24  }
25  else
26  {
27    thread_1 = 0i64;
28  }
29  threads_handles._Hnd = thread_1;
30  *ThrdAddr = beginthreadex(0i64, 0, StartAddress, thread_1, 0, &ThrdAddr[2]);
31  if ( !*ThrdAddr )
32  {
33    ThrdAddr[2] = 0;
34    std::_Throw_Cpp_error(6);
35  }
36  thread_2 = operator new(0x10ui64);
37  if ( thread_2 )
38  {
39    thread_2->args = a1;
40    thread_2->thread_function = thread_2_function;
41  }
42  else
43  {
44    thread_2 = 0i64;
45  }
46  threads_handles._Hnd = thread_2;
47  if ( !beginthreadex(0i64, 0, StartAddress, thread_2, 0, &v8) )
48  {
49    v8 = 0;
50    std::_Throw_Cpp_error(6);
51  }
```

Figure 5 Starting Worker Threads

Hemen ardından birbiriyle senkronize çalışan iki adet thread oluşturulduğu tespit edilmiştir.

```
31  Block[0] = 1414745936;
32  v4 = http_connection_open(a1, Block, a1 + 144) == 0;
33  if ( v19 >= 0x10 )
34  {
35    v5 = Block[0];
36    if ( v19 + 1 >= 0x1000 )
37    {
38      v5 = *(Block[0] - 1);
39      if ( (Block[0] - v5 - 8) > 0x1F )
40        invalid_parameter_noinfo_noreturn();
41    }
42    j_j_free(v5);
43  }
44  if ( !v4 )
45  {
46    if ( *(a1 + 192) )
47    {
48      v6 = sub_180002650(Block, a1 + 176);
49      if ( http_send(a1, v6) )
50      {
51        if ( http_data_read(a1, Src) )
52        {
53          Block[0] = 0i64;
54          v19 = 15i64;
55          v7 = 4i64;
56          if ( Size < 4 )
57            v7 = Size;
58          v8 = Src;
59          v3 = v22;
60          if ( v22 >= 0x10 )
61            v8 = Src[0];
62          v18 = v7;
```

Figure 6 C&C Communication

İlk iş parçacığı incelendiğinde, C2 sunucusu iletişimi tespit edilmiştir. C2 sunucusundan gelen veri *changeshell* ise bu durumda ulaştırılan zararlı komutların çalıştırılacağı CLI uygulaması değiştirilmektedir. Bu durum, arka kapının bulunduğu bilgisayarın kısıtlamalarını büyük ölçüde aşmayı sağlamaktadır.

```
__int64 __fastcall thread_2_function(configuration *a1)
{
  __int64 result; // rax

  WaitForSingleObject(*&a1->gap0[576], 0xFFFFFFFF);
  while ( 1 )
  {
    result = a1->gap0[0];
    if ( !result )
      break;
    if ( Get_Config_info_0x20(&a1->gap0[8]) )
      Parsing_C2_Command(a1);
    if ( Get_Config_info_0x48(&a1->gap0[8]) )
    {
      if ( a1->is_powershell_version_greater_or_equal_5 )
        execute_command_with_powershell(a1);
      else
        execute_command_with_cmd(a1);
    }
    if ( !Get_Config_info_0x20(&a1->gap0[8]) && !Get_Config_info_0x48(&a1->gap0[8]) )
      WaitForSingleObject(*&a1->gap0[576], 0xFFFFFFFF);
  }
  return result;
}
```

Figure 7 Command Execution

C2 sunucusundan gelen komutlar ikinci iş parçacığı tarafından sağlanmaktadır. Eğer, powershell sürümü 5'ten büyük veya eşit ise komutlar **powershell.exe** ile değil ise **cmd.exe** ile çalıştırılmaktadır.

Yürütülen powershell komutlarının sonrasında incelenememesi için aşağıdaki komut çalıştırılmaktadır:

Set-PSReadLineOption -HistorySaveStyle SaveNothing

C2 sunucusu tarafından gelen komutlar başlıca şunlardır:

- **timeout**: C2 sunucusu ile iletişime geçilme sıklığını belirtmek için kullanılan komut.
- **changeshell**: Komutlar eğer powershell.exe üzerinden yürütülüyorsa, cmd.exe üzerinden yürütülmesini ya da tam tersini sağlamak için kullanılan komut.
- **changepoint**: Yürütülen komutların sonuçları C2 sunucusuna gönderilirken kullanılan komut.
- **get**: C2 sunucusundan lokal bilgisayara dosya aktarımında kullanılan komut.

```
v10 = *a2 + 64;
NumberOfBytesWritten[0] = 0;
if ( *(v10 + 24) >= 0x10ui64 )
  v10 = *v10;
FileA = CreateFileA(v10, 0x40000000u, 0, 0i64, 2u, 0x80u, 0i64);
v12 = FileA;
if ( FileA == '\xFF' )
{
```

Figure 8 C&C Command: File downloading to local machine

```
v13 = lpBuffer;
if ( v19 >= 0x10 )
  v13 = lpBuffer[0];
if ( !WriteFile(FileA, v13, nNumberOfBytesToWrite[0], NumberOfBytesWritten, 0i64) )
{
  CloseHandle(v12);
  goto LABEL_25;
}
CloseHandle(v12);
```

Figure 9 C&C Command: File downloading to local machine

- **post**: Yerel bilgisayardan C2 sunucusuna dosya aktarımında kullanılan komut.

```
19  unsigned __int64 v23; // [rsp+60h] [rbp-38h]
20
21  NumberOfBytesRead = 0;
22  if ( !sub_180007960() )
23    return 0;
24  if ( *(a2 + 3) >= 0x10ui64 )
25    a2 = *a2;
26  FileA = CreateFileA(a2, 0x80000000, 1u, 0i64, 4u, 0x80u, 0i64);
27  v7 = FileA;
28  if ( FileA == -1i64 )
29    return 0;
30  FileSize = GetFileSize(FileA, 0i64);
31  v9 = FileSize;
32  if ( FileSize == -1 )
33  {
34    CloseHandle(v7);
35    return 0;
36  }
37  ProcessHeap = GetProcessHeap();
38  v12 = HeapAlloc(ProcessHeap, 0, v9);
39  if ( ReadFile(v7, v12, v9, &NumberOfBytesRead, 0i64) )
40  {
41    v14 = NumberOfBytesRead;
42    v21[0] = 0i64;
43    v23 = 15i64;
44    if ( NumberOfBytesRead > 0xFui64 )
45    {
46      v15 = NumberOfBytesRead | 0xFi64;
47      if ( v15 < 0x16 )
48        v15 = 22i64;
49      if ( v15 + 1 < 0x1000 )
50      {
```

Figure 10 C&C Command: File uploding from local machine

- **killme**: killme komutu aşağıdaki içeriğe sahip bir batch dosyası oluşturur. Arka kapı DLL dosyasının esasen bir hizmet olduğunu belirtmek ilginçtir, ancak batch dosyası **HKCU\SW\classes\CLSID** üzerindeki bir kayıt defteri anahtarını siler ve **explorer[.]exe**'yi yeniden başlatır, bu da Turla'nın geçmişte kötü amaçlı yazılımları için kalıcılık sağlamak için kullandığı bir taktik olan COM kaçırma kullanarak kalıcılık yaratma girişimini göstermektedir.(Cisco, TurlaNG)

```
@echo off
reg delete "HKEY_CURRENT_USER\Software\Classes\CLSID\{C2796011-81BA-4148-8FCA-C6643245113F}" /F
taskkill /F /IM explorer.exe
start explorer.exe
timeout 5
:d
del "%s"
if exist "%s" goto d
del /F "%s"
```

Figure 11 Cisco: BAT file contents template.

C2 iletişimi incelendiğinde bazı adresler tespit edildi. Zararlı yazılım, başlangıçta C2 sunucusuna *ClientReady* mesajı göndermektedir. C2 sunucusu hala erişilebilir ise sonrasında iletişim devam etmektedir. Eğer, ilgili domain adresine erişilemiyor ise domain listesindeki diğer domainler de aynı şekilde iletişim için kontrol edilir.

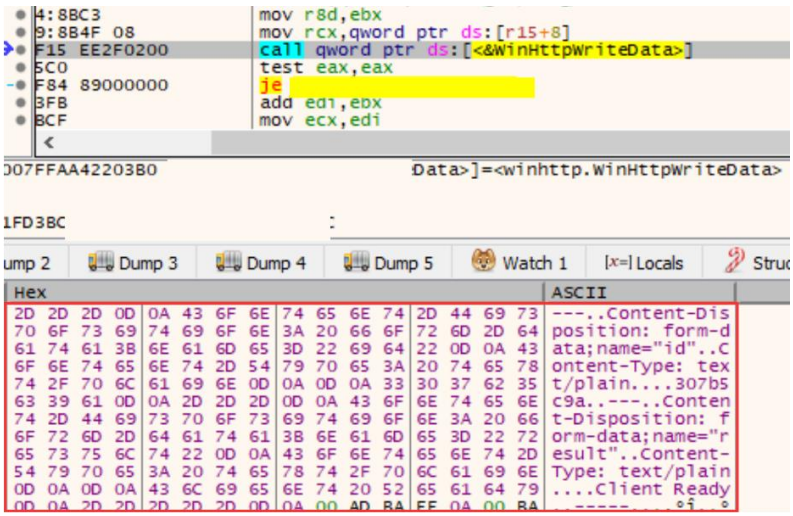

Figure 12 ClientReady message

Başlangıçta "https[:]//jeepcarlease[.]com/wp-includes/blocks/rss.old[.]php" adresine ClientReady http isteği atıldığı tespit edildi. Ayrıca bağlanmaya çalışıldığı tespit edilen diğer domain adresleri aşağıdaki gibidir:

- caduff-sa[.]ch
- hanagram[.]jp
- buy-new-car[.]com
- thefinetreats[.]com
- carleasingguru[.]com

Kurallar

SIGMA

```
title: C2 Comminication for TinyTurla-NG
description: Detects communication with the command and control server
author: Bilal Bakartepe
date: 2024/04/18
status: experimental
logsource:
  product: windows
  category: network_connection
detection:
  selectionURL:
    cs-uri|contains:
    - "https://jeepcarlease.com/wp-includes/blocks/rss.old.php"
    - "https://caduff-sa.ch/wp-includes/blocks/rss.old.php"
    - "hanagram.jp"
    - "buy-new-car.com"
    - "thefinetreats.com"
    - "carleasingguru.com"

  condition: selectionURL
falsepositives:
- Unknown
level: high
```

YARA

```
rule TinyTurlaNG {
    meta:
        date = "18.04.2024"
        author = "Bilal BAKARTEPE"
        hash = "0f2e9f501ca9780eff309b7022c9b01a"
    strings:

        $pwshll_command_1 = "Set-PSReadLineOption -HistorySaveStyle SaveNothing"
        $pwshll_command_2 = "chcp 437 > $null"

        $c2_command_1= "timeout"
        $c2_command_2= "killme"
        $c2_command_3= "changeshell"
        $c2_command_4= "changepoint"
        $c2_command_5= "get"
        $c2_command_6= "post"

    condition:
        all of them
}
```


ECHO
CYBER THREAT INTELLIGENCE