АНТИВИРУСНАЯ ЗАЩИТА БИЗНЕС-КЛАССА

Новый бэкдор кибергруппы TeleBots: первое доказательство связи Industroyer и NotPetya

11 октября 2018 года

Исследование нового бэкдора кибергруппы TeleBots, стоящей за эпидемией шифратора NotPetya, выявило значительное сходство кода с основным бэкдором Industroyer, что подтверждает взаимосвязь, о которой ранее только распространялись слухи.

В числе крупнейших киберинцидентов последних лет – атаки на украинские энергетические предприятия и эпидемия шифратора NotPetya. В посте мы рассмотрим взаимосвязь между этими событиями.

Первое в истории массовое отключение электроэнергии, вызванное кибератакой, произошло в декабре 2015 года, его причина – комплекс вредоносных программ BlackEnergy. Специалисты ESET отслеживали активность APT-группы, использующей BlackEnergy, как до, так и после этого события. После блэкаута 2015 года группа, по всей видимости, свернула активную работу с BlackEnergy и эволюционировала в то, что мы сейчас называем TeleBots.

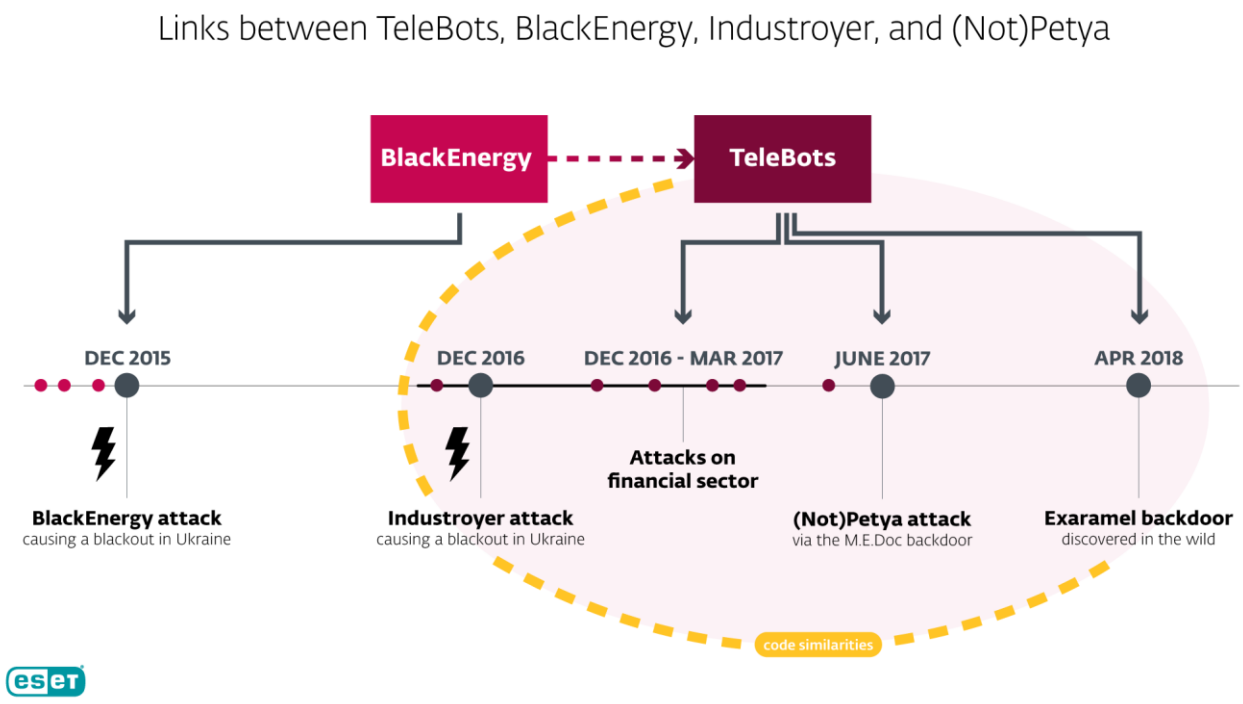
Здесь надо отметить, что, говоря об APT группе, мы имеем в виду общие технические индикаторы: сходство кода, общую сетевую (C&C) инфраструктуру, цепочки выполнения вредоносных программ и пр. Как правило, мы не принимаем непосредственного участия в расследовании и идентификации разработчиков или операторов. Термин «APT группа» не имеет четкого определения и часто используется для характеристики параметров вредоносного ПО. По этим причинам мы воздерживаемся от предположений относительно источника атак, национальной или государственной принадлежности атакующих.

В любом случае, мы обнаружили связь между атаками BlackEnergy (нацеленными на украинские энергокомпании, а также другие отрасли и высокоранговые объекты) и кампаниями группы TeleBots (преимущественно против украинского финансового сектора).

В июне 2017 года компании по всему миру пострадали от вайпера Diskcoder.C (более известного как Petya/NotPetya) — массовый характер заражения, по всей видимости, был неким побочным эффектом. Изучая инцидент, мы выяснили, что «нулевым пациентом» этой эпидемии стали компании, зараженные бэкдором группы TeleBots, в результате компрометации популярного в украинских компаниях бухгалтерского ПО M.E.Doc.

Вопрос в том, какое отношение к этой истории имеет Industroyer – сложный вредоносный комплекс, ставший причиной блэкаута в Киеве в декабре 2016 года. Сразу после публикации отчета ESET некоторые ИБ-компании и СМИ предположили, что Industroyer тоже разработала группа BlackEnergy/TeleBots (иногда также называемая Sandworm). Тем не менее, доказательств до настоящего времени представлено не было.

В апреле 2018 года мы зафиксировали новую активность группы TeleBots – попытку развернуть новый бэкдор **Win32/Exaramel**. Наш анализ показал, что этот бэкдор является усовершенствованной версией основного бэкдора Industroyer – что и стало первым доказательством.



Анализ бэкдора Win32/Exaramel

Бэкдор Win32/Exaramel изначально устанавливается с помощью дроппера. Согласно метаданным дроппера, бэкдор скомпилирован в Microsoft Visual Studio непосредственно перед развертыванием на конкретном скомпрометированном компьютере.

```
Count of sections                6 | Machine                      Intel386
Symbol table  00000000[00000000]   | UTC     Fri Apr 13 08:36:04 2018
Size of optional header      00E0  | Magic optional header        010B
Linker version              14.00  | OS version                   5.01
Image version                0.00  | Subsystem version            5.01
Entry point              00004344  | Size of code             0000DA00
Size of init data        0001AA00  | Size of uninit data      00000000
Size of image            0002D000  | Size of header           00000400
Base of code             00001000  | Base of data             0000F000
Image base               00400000  | Subsystem                     GUI
Section alignment        00001000  | File alignment           00000200
Stack          00100000/00001000   | Heap           00100000/00001000
Checksum                 00000000  | Number of dirs                  16
```

Рисунок 1. Временная метка PE дроппера бэкдора Win32/Exaramel

После выполнения дроппер размещает бинарный файл Win32/Exaramel в системном каталоге Windows и запускает службу Windows с именем `wsmproav` с описанием «Windows Check AV». Название файла и описание службы Windows жестко закодировано в дроппере.

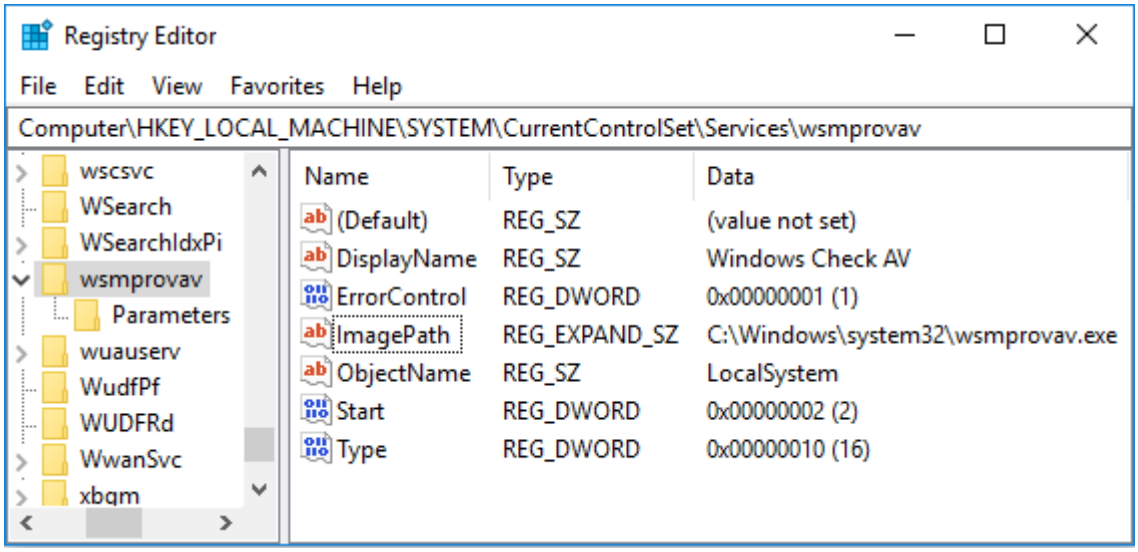


Рисунок 2. Настройки реестра службы Windows, созданные бэкдором Win32/Exaramel

Кроме того, дроппер записывает конфигурацию бэкдора в реестр Windows в формате XML.

```
<settings group="eset" name="" version="18.04.12">
    <interval>10000</interval>
    <servers>
        <server current="true">https://esetsmart.org</server>
    </servers>
    <check>https://www.yandex.ua</check>
    <proxy password="" user="">[redacted]:3128</proxy>
    <storage>c:\Intel</storage>
    </settings>
```

Рисунок 3. XML-конфигурация Win32/Exaramel

Конфигурация состоит из нескольких блоков:

— *Interval* – время в миллисекундах, используемое для функции Sleep

— *Servers* – список командных серверов (C&C)
— *Check* – веб-сайт, используемый для определения того, имеет ли хост доступ к интернету
— *Proxy* – прокси-сервер в сети хоста
— *Storage* – путь, используемый для хранения файлов, предназначенных для эксфильтрации

Как видно из первой строки конфигурации, злоумышленники группируют цели на основе используемых антивирусных продуктов. Подобный подход применяется в комплексе Industroyer – в частности, некоторые бэкдоры Industroyer были также замаскированы как служба, связанная с антивирусом (развернутая под именем `avtask.exe`) и использовали тот же подход к образованию групп.

Еще один интересный факт – бэкдор использует C&C-серверы, доменные имена которых имитируют домены, принадлежащие ESET. В дополнение к `esetsmart[.]org` из вышеупомянутой конфигурации мы обнаружили похожий домен `um10eset[.]net`, который использовался недавно открытой Linux-версией вредоносного ПО TeleBots. Важно отметить, что эти контролируемые атакующими серверы не имеют отношения к легитимной сетевой инфраструктуре ESET. В настоящее время мы не обнаружили, чтобы Exaramel использовал домены, имитирующие инфраструктуру других ИБ-компаний.

После запуска бэкдор устанавливает связь с C&C-сервером и получает команды для выполнения в системе. Ниже список доступных команд:

— Запуск процесса
— Запуск процесса от определенного пользователя Windows
— Запись данных в файл по заданному пути
— Скопировать файл в подкаталог хранилища (Загрузить файл)
— Выполнить шелл-команду
— Выполнить шелл-команду от определенного пользователя Windows
— Выполнить VBS-код, используя `MSScriptControl.ScriptControl.1`

Код командного цикла и реализация первых шести команд в Win32/Exaramel очень напоминает бэкдор из программного комплекса Industroyer.

```
DWORD __stdcall cmd_thread(thread_param *param)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  result1 = 0x16;
  v2 = init_CMD_struct(param->xml, &CMD);
  SetEvent((HANDLE)param->event);
  if ( v2 )
    return 1;
  cmd_struct1 = CMD;
  switch ( CMD->cmd_id )
  {
    case 1:
      result = cmd_create_proccess(CMD);
      goto end;
    case 2:
      result = cmd_create_proccess_as_user(CMD);
      goto end;
    case 3:
      result = cmd_write_file(CMD);
      goto end;
    case 4:
      result = cmd_copy_file_aka_upload(CMD);
      goto end;
    case 5:
      result = cmd_execute_shell_cmd(CMD);
      goto end;
    case 6:
      result = cmd_execute_shell_cmd_as_user(CMD);
      goto end;
    case 7:
      result = cmd_eval_VBS_code(CMD);
end:
      result1 = result;
      break;
    default:
      break;
  }
  PathCombineW(&pszDest, (LPCWSTR)cmd_struct1->storage_path, L"done");
  file_write(&pszDest, 0, 0);
  mem_free((LPVOID)cmd_struct1->field_0);
  mem_free((LPVOID)cmd_struct1->cmd_content);
  mem_free((LPVOID)cmd_struct1->file_content);
  mem_free(cmd_struct1);
  return result1;
}
```

```
int __cdecl run_command(cmd_internal *CMD)
{
  int result; // eax

  result = LOBYTE(CMD->cmd_id) - 1;
  switch ( LOBYTE(CMD->cmd_id) )
  {
    case 1u:
      result = cmd_create_proccess(CMD);
      break;
    case 2u:
      result = cmd_create_proccess_as_user(CMD);
      break;
    case 3u:
      result = cmd_write_file(CMD);
      break;
    case 4u:
      result = cmd_copy_file_aka_upload(CMD);
      break;
    case 5u:
      result = cmd_execute_shell_cmd(CMD);
      break;
    case 6u:
      result = cmd_execute_shell_cmd_as_user(CMD);
      break;
    case 7u:
      ExitProcess(0);
      return result;
    case 8u:
      result = cmd_stop_service(CMD);
      break;
    case 9u:
      result = cmd_stop_service_as_user(CMD);
      break;
    case 0xAu:
      result = cmd_start_service_as_user(CMD);
      break;
    case 0xBu:
      result = cmd_service_change_path_to_binary_as_user(CMD);
      break;
    default:
      return result;
  }
  return result;
}
```

Рисунок 4. Сравнение декомпилированного кода бэкдоров Win32/Exaramel (слева) и Win32/Industroyer (справа)

Оба семейства вредоносных программ используют файл отчета для хранения выходных результатов выполненных шелл-команд и запущенных процессов. В случае с бэкдором Win32/Industroyer файл отчета хранится во временной папке с рандомным названием; в Win32/Exaramel файл отчета называется `report.txt`, а его путь к хранилищу задан в файле конфигурации бэкдора.

Чтобы перенаправить стандартный вывод (stdout) и стандартную ошибку (stderr) в файл отчета, оба бэкдора задают параметры `hStdOutput` и `hStdError` дескриптору файла отчета. Это еще одно сходство между этими семействами вредоносных программ.

```
int __cdecl cmd_execute_shell_cmd(cmd_internal *CMD)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  mem_set(&StartupInfo, 68);
  mem_set(&ProcessInformation, 16);
  SecurityAttributes.nLength = 12;
  SecurityAttributes.lpSecurityDescriptor = 0;
  SecurityAttributes.bInheritHandle = 1;
  reportfile_handle = CreateFileW(
                        CMD->reportfile_path,
                        GENERIC_WRITE|GENERIC_READ,
                        1u,
                        &SecurityAttributes,
                        2u,
                        0x80u,
                        0);
  if ( reportfile_handle == -1 )
    ExitCode = GetLastError();
  StartupInfo.dwFlags |= STARTF_USESTDHANDLES|STARTF_USESHOWWINDOW;
  StartupInfo.wShowWindow = 0;
  StartupInfo.cb = 68;
  StartupInfo.hStdError = reportfile_handle;
  StartupInfo.hStdOutput = reportfile_handle;
  ExpandEnvironmentStringsW(L"%ComSpec%", &Dst, 0x104u);
  wsprintfW(&CommandLine, L"/c %s", CMD->cmd_content);
  CreateProcessW(&Dst, &CommandLine, 0, 0, 1, 0x8000000u, 0, 0, &StartupInfo, &ProcessInformation);
  if ( ProcessInformation.hProcess )
  {
    if ( !WaitForSingleObject(ProcessInformation.hProcess, 0xFFFFFFFF) )
    {
      FlushFileBuffers(reportfile_handle);
      GetExitCodeProcess(ProcessInformation.hProcess, &ExitCode);
      CloseHandle(ProcessInformation.hThread);
      CloseHandle(ProcessInformation.hProcess);
    }
  }
  else
  {
    ExitCode = GetLastError();
  }
  CloseHandle(reportfile_handle);
  return ExitCode;
}
```

```
int __cdecl cmd_execute_shell_cmd(cmd_internal *CMD)
{
  int result; // edi
  WCHAR CommandLine; // [esp+8h] [ebp-A7Ch]
  WCHAR Dst; // [esp+828h] [ebp-25Ch]
  struct _STARTUPINFOW StartupInfo; // [esp+A30h] [ebp-54h]
  struct _PROCESS_INFORMATION ProcessInformation; // [esp+A74h] [ebp-10h]

  mem_set(&StartupInfo, 68);
  mem_set(&ProcessInformation, 16);
  StartupInfo.dwFlags |= STARTF_USESTDHANDLES|STARTF_USESHOWWINDOW;
  StartupInfo.cb = 68;
  StartupInfo.hStdError = (HANDLE)CMD->reportfile_handle;
  StartupInfo.hStdOutput = StartupInfo.hStdError;
  StartupInfo.wShowWindow = 0;
  ExpandEnvironmentStringsW(L"%ComSpec%", &Dst, 0x104u);
  wsprintfW(&CommandLine, L"/c %s", CMD->cmd_content);
  result = CreateProcessW(&Dst, &CommandLine, 0, 0, 1, 0x8000000u, 0, 0, &StartupInfo, &ProcessInformation);
  if ( ProcessInformation.hProcess && !WaitForSingleObject(ProcessInformation.hProcess, 0xFFFFFFFF) )
  {
    FlushFileBuffers((HANDLE)CMD->reportfile_handle);
    CloseHandle(ProcessInformation.hThread);
    CloseHandle(ProcessInformation.hProcess);
  }
  return result;
}
```

Рисунок 5. Сравнение декомпилированного кода бэкдоров Win32/Exaramel и Win32/Industroyer соответственно

Если операторы вредоносного ПО хотят эксфильтровать файлы с компьютера жертвы, им достаточно скопировать эти файлы в подкаталог пути хранения `data`, заданный в конфигурации. Перед установкой нового соединения с C&C-сервером бэкдор автоматически сожмет и зашифрует эти файлы, прежде чем их отправить.

Главное различие бэкдора Industroyer и нового бэкдора TeleBots в том, что последний использует формат XML для связи и конфигураций вместо кастомного бинарного формата.

Вредоносные инструменты для кражи паролей

Вместе с Exaramel группа TeleBots использует некоторые из старых инструментов, включая средство для кражи паролей (внутреннее название – CredRaptor или PAI) и незначительно модифицированный Mimikatz.

Усовершенствованный инструмент для кражи паролей CredRaptor, используемый только этой группой с 2016 года, был доработан. В отличие от предыдущих версий, он собирает сохраненные пароли не только из браузеров, но и из Outlook и ряда FTP-клиетов. Ниже список поддерживаемых приложений:

— BitKinex FTP
— BulletProof FTP Client
— Classic FTP
— CoffeeCup
— Core FTP
— Cryer WebSitePublisher
— CuteFTP
— FAR Manager
— FileZilla
— FlashFXP
— Frigate3
— FTP Commander
— FTP Explorer
— FTP Navigator
— Google Chrome
— Internet Explorer 7 – 11
— Mozilla Firefox
— Opera
— Outlook 2010, 2013, 2016
— SmartFTP
— SoftX FTP Client
— Total Commander
— TurboFTP
— Windows Vault
— WinSCP
— WS_FTP Client

Доработки позволяют атакующим собирать данные от учетных записей веб-мастеров веб-сайтов и от серверов во внутренней инфраструктуре. Получив доступ к таким серверам, можно установить дополнительные бэкдоры. Достаточно часто на этих серверах установлена ОС, отличная от Windows, поэтому атакующим приходится адаптировать бэкдоры.

В ходе мероприятий по реагированию на инцидент мы обнаружили Linux-бэкдор группы TeleBots – **Linux/Exaramel.A**.

Анализ бэкдора Linux/Exaramel

Бэкдор написан на языке программирования Go и скомпилирован как 64-битный бинарный файл ELF. Атакующие могут развернуть бэкдор в выбранном каталоге под любым именем.

Если бэкдор выполнен атакующими со строкой ‘none’ в качестве аргумента командной строки, он

пытается использовать механизмы персистентности, чтобы автоматически запускаться после перезагрузки. Если бэкдор не выполняется под учетной записью root, он использует файл `crontab`. Однако если он запущен с правами root, он поддерживает различные системы Linux `init`. Он определяет, какая `init` система используется в настоящее время, выполняя команду:

```
strings /sbin/init | awk 'match($0, /(upstart|systemd|sysvinit)/){ print substr($0, RSTART, RLENGTH);exit; }'
```

В зависимости от результата бэкдор использует следующие жестко закодированные локации для обеспечения персистентности (далее — Init система и ее расположение):

sysvinit — /etc/init.d/syslogd
upstart — /etc/init/syslogd.conf
systemd — /etc/systemd/system/syslogd.service

В ходе запуска бэкдор пытается открыть файл конфигурации, который хранится в том же каталоге, что и бэкдор, под именем `config.json`. Если файла конфигурации не существует, создается новый файл. Конфигурация шифруется с помощью ключа **s0m3t3rr0r** и алгоритма RC4.

Рисунок 6. Расшифрованная конфигурация JSON бэкдора Linux/Exaramel

Бэкдор подключается к жестко закодированному C&C (по умолчанию 176.31.225[.]204 в образце, который мы видели) или к C&C-серверу, указанному в файле конфигурации в значении `Hosts`. Коммуникации осуществляются через HTTPS. Бэкдор поддерживает следующие команды:

App.Update — обновление до новой версии
App.Delete — самоудаление из системы
App.SetProxy — задать параметры прокси-сервера в конфигурации
App.SetServer — обновление C&C-сервера в конфигурации
App.SetTimeout — установка значения таймаута (интервалы между соединениями с C&C-сервером)
IO.WriteFile — загрузка файла с удаленного сервера
IO.ReadFile — выгрузка файла с локального диска на C&C-сервер
OS.ShellExecute — выполнить шелл-команду

Вывод

Открытие Exaramel показывает, что группа TeleBots сохраняет активность в 2018 году, и атакующие продолжают совершенствовать тактику и инструментарий.

Значительное сходство кода Win32/Exaramel и основного бэкдора Industroyer – первое публично представленное доказательство, связывающее Industroyer с группой TeleBots и, следовательно, с киберкампаниями NotPetya и BlackEnergy. Устанавливая связь между источниками кибератак, стоит принимать во внимание возможность ошибки или преднамеренного обмана, однако в этом случае мы считаем это маловероятным.

Стоит отметить, что Win32 и Linux-версии бэкдора Exaramel обнаружены в организации, которая не имеет отношение к промышленности. Специалисты ESET сообщили о находке в следственные органы Украины, благодаря чему атака была своевременно локализована и предотвращена.

ESET продолжает отслеживать активность данной кибергруппы.

Индикаторы компрометации (IoCs)

Детектирование продуктами ESET

```
Win32/Exaramel trojan
Win32/Agent.TCD trojan
Linux/Agent.EJ trojan
Linux/Exaramel.A trojan
Win32/PSW.Agent.OEP trojan
Win32/RiskWare.Mimikatz.Z application
Win64/Riskware.Mimikatz.AI application
```

Хеши SHA-1

Бэкдор Win32/Exaramel кибергруппы TeleBots

```
65BC0FF4D4F2E20507874F59127A899C26294BC7
3120C94285D3F86A953685C189BADE7CB575091D
```

Инструмент для кражи паролей

```
F4C4123849FDA08D1268D45974C42DEB2AAE3377
970E8ACC97CE5A8140EE5F6304A1E7CB56FA3FB8
DDDF96F25B12143C7292899F9D5F42BB1D27CB20
64319D93B69145398F9866DA6DF55C00ED2F593E
1CF8277EE8BF255BB097D53B338FC18EF0CD0B42
488111E3EB62AF237C68479730B62DD3F52F8614
```

Mimikatz

```
458A6917300526CC73E510389770CFF6F51D53FC
CB8912227505EF8B8ECCF870656ED7B8CA1EB475
```

Linux/Exaramel

```
F74EA45AD360C8EF8DB13F8E975A5E0D42E58732
```

C&C-серверы

```
um10eset[.]net (IP: 176.31.225.204)
esetsmart[.]org (IP: 5.133.8.46)
```