مرکز تخصصی
آپا دانشگاه کردستان

بررسی ویژگی ADS در سیستم فایل NTFS و مدیریت آن‌ها در ویندوز

مسلم حقیقیان

شماره سند: A96001

۱۳۹۶/۰۷/۰۱

www.cert.uok.ac.ir
apa@uok.ac.ir
087-33662932

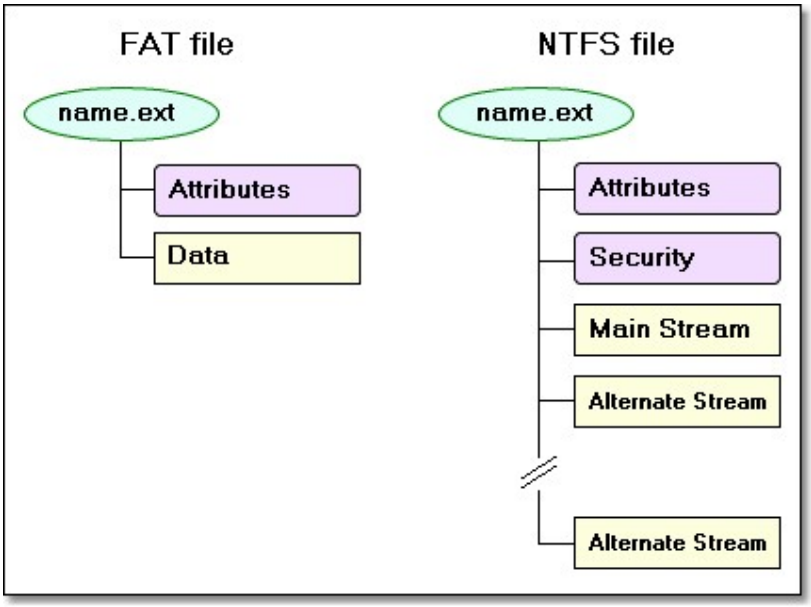
۱- چکیده

در ویندوزهای NT3.1، سیستم فایل NTFS به سیستم‌عامل مایکروسافت به‌عنوان یک سیستم فایل امن روی کارآمد. ویژگی ADS یا Alternate Data Stream در سیستم فایل NTFS به متا دیتا این امکان را می‌دهد تا بتوان یک متن و یا فایل و ... را در پشت یک فایل قرار دهیم بدون اینکه محتویات فایل یا حتی حجم فایل تغییر کند اما بدافزار نویسان سو استفاده‌های فراوانی را از این امکان کردند تا بتوانند آنتی‌ویروس‌ها را دور بزنند و بدافزار خود را در پشت یک فایل مخفی و وارد سیستم‌عامل کنند.در این تحقیق سعی شده به معرفی روش‌های سو استفاده از این امکان توسط بدافزارها بپردازیم و همچنین روش کد نویسی در آن را توسط زبان برنامه‌نویسی c++ و روش‌های نوشتن و حذف استریم در داخل ADS یک فایل را معرفی کنیم.

کلمات کلیدی: ADS, Alternate Data Stream,NTFS,Powershell,cmd,get-content,set-content,set-content, get-item ,set-item,type,more

۲- ویژگی ADS در سیستم فایل NTFS

از مهم‌ترین ویژگی در سیستم فایل NTFS امکان اضافه کردن Stream به یک فایل است. یعنی می‌توان یک یا چند فایل را در یک فایل پنهان کرد که این امکان در FAT وجود ندارد.



شما می‌توانید یک فایل که نمی‌خواهید کسی آن را ببیند و یا اینکه یک پسورد را در داخل این فایل‌ها ذخیره کنید. مسئله جالب اینجاست که ما هر فایلی را با هر حجمی با استفاده از این متد مخفی کنیم حجم فایل اصلی در windows explorer تغییر نمی‌کند (در اصل windows explorer آن را نمی‌تواند نشان دهد وگرنه درواقع تغییر می‌کند). در این گزارش سعی داریم که روش مخفی سازی را شرح دهیم و سپس به طریقه مقابله و تشخیص فایل‌ها و کلمات مخفی‌شده بپردازیم.

۳- روش مخفی سازی

در نظر بگیرید که یک فایل بانام wininfo.txt وجود دارد و می‌خواهیم یک پسورد را در آن با استفاده از این روش ذخیره کنیم. در قدم اول فایل wininfo.txt را می‌سازیم و مقدار p4ssw0rd را به‌صورت مخفی در آن قرار می‌دهیم.

```
echo p4ssw0rd > wininfo.txt:hidden
```

حالا اگر فایل wininfo.txt را بازکنید می‌بینید که چیزی در داخل آن وجود ندارد و حتی اگر حجم آن را نیز ببینید همان 0 byte است.

```
C:\test>dir wininfo.txt
 ۰۸:۳۱  ۲۰۱۴/۲۴/۰۴PM                 0 wininfo.txt
```

فرمان زیر را به کار ببرید می‌توانید مقدار مخفی را ببینید. به شکل زیر:

```
C:\test>more < test.txt:hidden
Hidden text
```

اگر بخواهیم به‌طور دقیق‌تر بررسی کنیم شکل کلی فرمان به‌صورت زیر است:

```
filename:stream name:stream
```

تنها نوع Stream که می‌توان با command prompt به آن دسترسی داشته باشیم $DATA است.

```
C:\test>echo This is the file > wininfo.txt
C:\test>echo This is the stream > wininfo.txt:stream
```

لیستی از انواع Stream ها را می‌توان در این آدرس مشاهده کرد.

http://msdn.microsoft.com/en-us/library/aa362667%28v=VS.85%29.aspx

با استفاده از کدهای WMI می‌توانید با آن‌ها نیز کارکنید. همچنین جهت کار با ADS ها در NTFS می‌توان از زبان C++ کمک گرفت.

http://support.microsoft.com/kb/105763

۴- حالت‌های دیگر کار با ADS

در نظر بگیرید که می‌خواهیم یک فایل با نام hidden.txt که حاوی اطلاعات محرمانه است را در یک فایل متنی بانام wininfo.txt مخفی کنیم. ابتدا فایل‌های hidden.txt و wininfo.txt را می‌سازیم.

```
echo p4ssw0rd > hidden.txt
echo nothing > wininfo.txt
```

فایل hidden را در فایل wininfo مخفی می‌سازیم و آن را اجرا می‌کنیم.

```
echo Hidden text > wininfo.txt:hidden.txt
```

حالا اگر فایل wininfo را بازکنید همان مقدار nothing را در آن مشاهده می‌کنید. اما بافرمان زیر با استفاده از برنامه notepad می‌توانیم به فایل hidden.txt دسترسی پیدا کنیم.

```
notepad wininfo.txt:hidden.txt
```

۵- مخفی کردن یک عکس پشت فایل و اجرای آن

در نظر بگیرید که می‌خواهیم یک عکس بانام secret.jpg را در فایل wininfo.txt مخفی کنیم.

```
type secret.jpg > wininfo.txt:secret.jpg
```

برای خواندن آن از برنامه mspaint استفاده می‌کنیم.

```
mspaint wininfo.txt:secret.jpg
```

۶- مخفی کردن یک کد VBS در پشت یک فایل و اجرای آن

این کار می‌تواند بسیار خطرناک باشد چون می‌توان یک بدافزار که به زبان VBS و یا JS نوشته‌شده است را در پشت یک فایل ذخیره و آن را اجرا کرد.

```
type malware.vbs > wininfo.txt:malware.vbs
Wscript  wininfo.txt:malware.vbs
```

۷- مخفی کردن یک فایل EXE در پشت یک فایل دیگر و اجرای آن

قسمت اصلی بحث در این است که بتوان یک فایل exe را در پشت یک فایل دیگر مخفی و سپس آن را اجرا کرد (در نظر بگیرید که این فایل exe می‌تواند یک malware باشد).

```
type malware.exe > wininfo.txt: malware.exe
powershell  .\wininfo.txt:malware.exe
```

۸- اضافه کردن یک مقدار دیگر در Stream های فایل

در نظر بگیرید که می‌توان چندین مقدار را در Stream های یک فایل اضافه کرد. به‌عنوان‌مثال ما می‌خواهیم یک فایل دیگر که بانام malware2.exe هست را بانام KST در قسمت Stream ها اضافه کنیم.

```
type malware2.exe > wininfo.txt:KST
```

۹- ایجاد ADS با استفاده از powershell ویندوز

powershell دارای فرمان‌ها قدرتمند و بهتری نسبت به CMD ویندوز است. از فرامین زیر جهت ساختن و دیدن محتویات فایل متنی استفاده می‌کنیم.

```
$file = "wininfo.txt"
Set-Content -Path $file -Value 'Test'
Get-Content -Path $file
```

و از فرمان زیر جهت اضافه کردن مقادیر به Stream های فایل استفاده می‌شود.

```
Add-Content -Path $file -Value 'P4ssw0rd' -Stream 'secret'
```

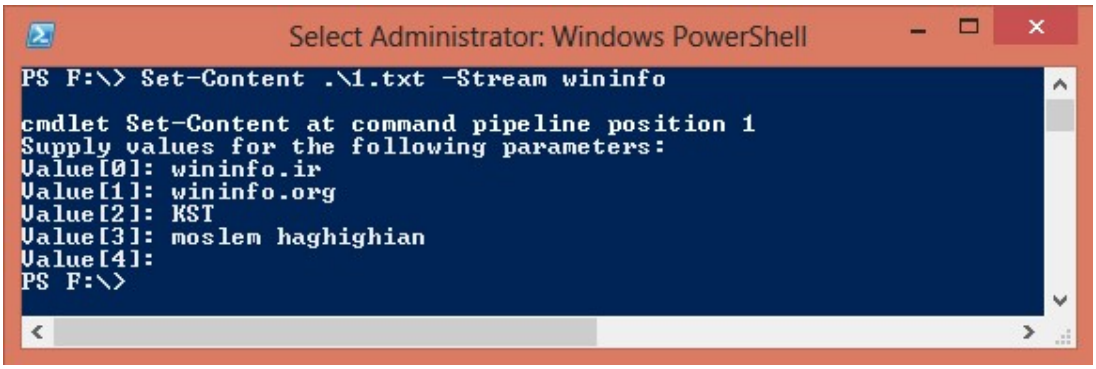
همچنین جهت دیدن محتویات فایل در حالت عادی از فرمان زیر استفاده می‌شود.

```
Get-Content -Path $file
```

و جهت دیدن متن مخفی و دسترسی به Stream فایل از فرمان زیر استفاده می‌شود.

```
Get-Content -Path $file -Stream 'secret'
```

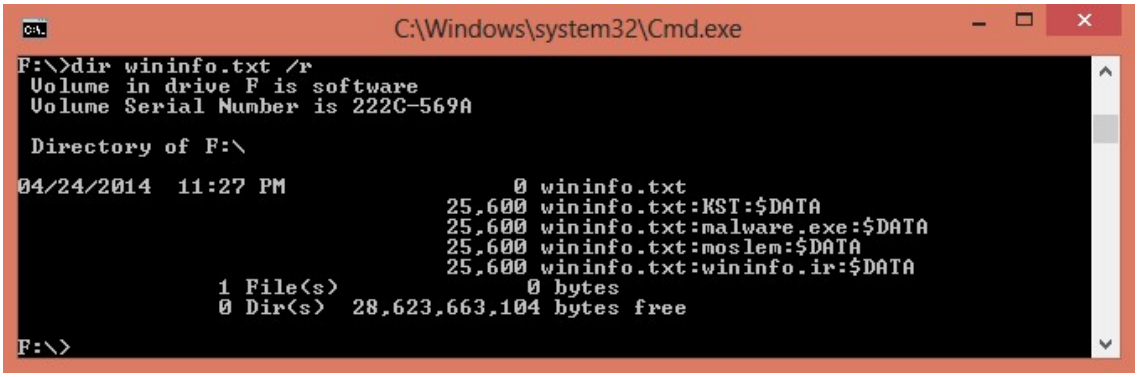
نحوه دیگر اعمال این دستور:



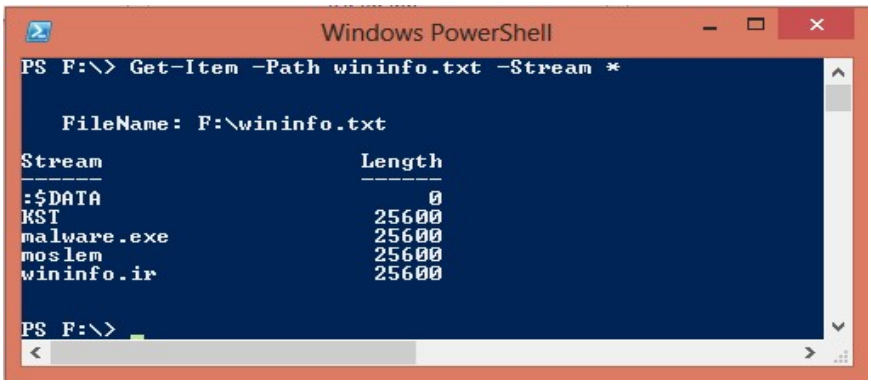
۱۰- طریقه‌ی شناسایی فایل‌ها باقابلیت ADS

درصورتی‌که یک فایل که دارای ADS باشد از اینترنت دانلود نمایید مرورگر پیغام می‌دهد که فایل دارای ADS (Zone.Identifier) است. مثلاً هنگام دانلود فایل setup.exe که دارای ADS باشد مرورگر IE پیغام زیر را نشان می‌دهد. علاوه بر این انواع آنتی‌ویروس‌ها به سیستم تشخیص ADS مجهز هستند.

جهت تشخیص و شناسایی این فایل‌ها روش‌های مختلفی وجود دارد. روش اول که ساده‌ترین روش است استفاده از برنامه CMD ویندوز و فرمان DIR هست. درصورتی‌که فرمان DIR با استفاده از سویچ R/ استفاده شود می‌تواند مقادیر Stream موجود در یک فایل را با آن مشاهده کرد. با این فرمان لیست تمامی Stream ها را می‌توانید مشاهده کرد که در اینجا بانام‌های Wininfo.ir و Moslem و Malware.exe و KST است.



روش دیگر و قوی‌تر استفاده از powershell ویندوز است که بافرمان زیر آن‌ها را بررسی می‌کند.



می‌توان به‌جای ستاره فقط نام stream موردنظر را بنویسید.

با استفاده از فرمان Find در CMD امکان لیست کردن تمام فایل‌های دارای ADS وجود دارد.

```
C:\Windows\system32\cmd.exe
F:\>dir /r | find "DATA"
                                    26 1.jpg:Zone.Identifier:$DATA
                                    26 1527093_665228840238432_1131853141_n (1).
jpg:Zone.Identifier:$DATA
                                    26 1982295_1433384020242223_991864484_n.jpg:
Zone.Identifier:$DATA
                                    14 test.exe:my_ads.txt:$DATA
                                    14 test_copy.exe:my_ads.txt:$DATA
                                    14 wininfo.exe:my_ads.txt:$DATA
                                     0 wininfo.exe:n2.e:$DATA
                                     0 wininfo.exe:n2.exe:$DATA
                                    16 wininfo.exe:yourads.txt:$DATA

F:\>_
```

همچنین با استفاده از کد زیر در powershell می‌توان لیستی از تمام فایل‌هایی که در $DATA stream هستند را در داخل یک فولدر مشاهده کرد.

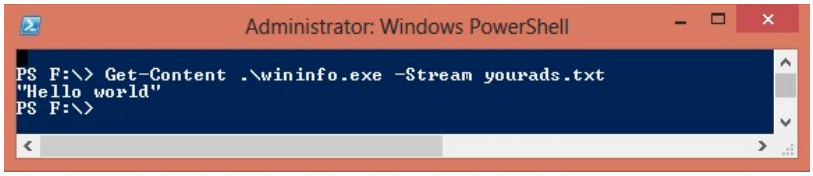
```
Get-Item * -stream *
```

روش دیگر استفاده از فرمان زیر است.

```
Administrator: Windows PowerShell
PS F:\> gci -recurse | % { gi $_.FullName -stream * } | where stream -ne ':$Data'

   FileName: F:\1.jpg
Stream                    Length
------                    ------
Zone.Identifier               26

   FileName: F:\1527093_665228840238432_1131853141_n (1).jpg
Stream                    Length
------                    ------
Zone.Identifier               26

   FileName: F:\1982295_1433384020242223_991864484_n.jpg
Stream                    Length
------                    ------
Zone.Identifier               26

   FileName: F:\test.exe
Stream                    Length
------                    ------
my_ads.txt                    14

   FileName: F:\test_copy.exe
Stream                    Length
------                    ------
my_ads.txt                    14

   FileName: F:\wininfo.exe
Stream                    Length
------                    ------
my_ads.txt                    14
n2.e                           0
n2.exe                         0
yourads.txt                   16
```

همچنین جهت دیدن محتویات stream موردنظر با استفاده از فرامین powershell می‌توان از فرمان زیر استفاده کرد.

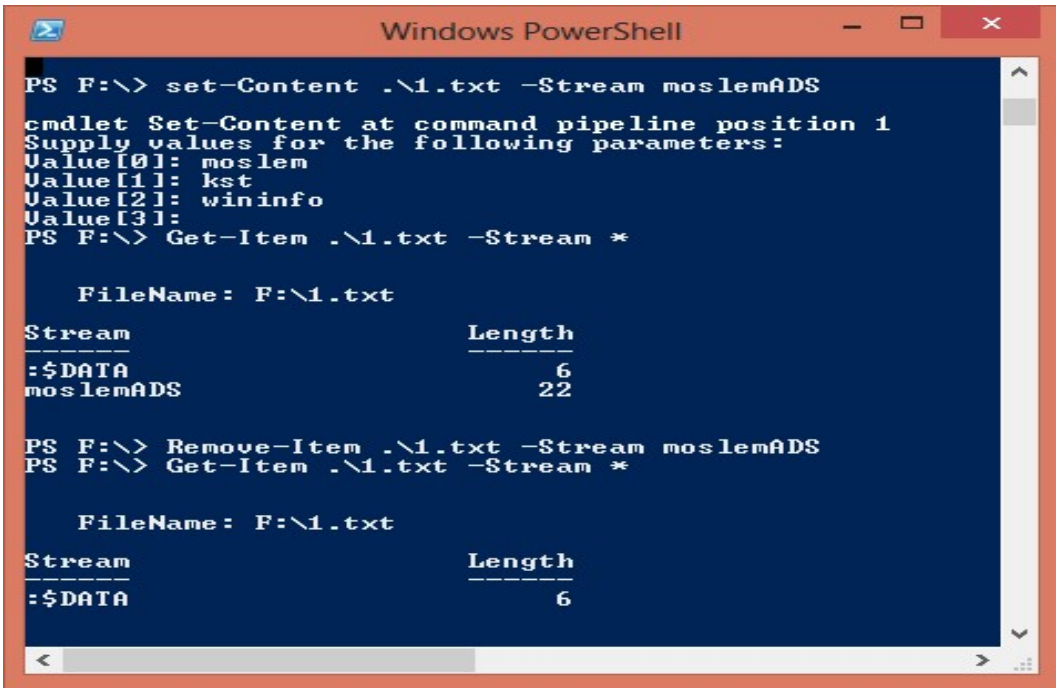
```
Get-Content .\wininfo.exe -Stream yourads.txt
```



۱۱- روش حذف Stream های فایل

جهت حذف کردن Stream های یک فایل با استفاده از فرامین powershell می‌توان به‌صورت زیر عمل کرد:

```
Remove-Item .\1.txt -Stream moslemADS
```

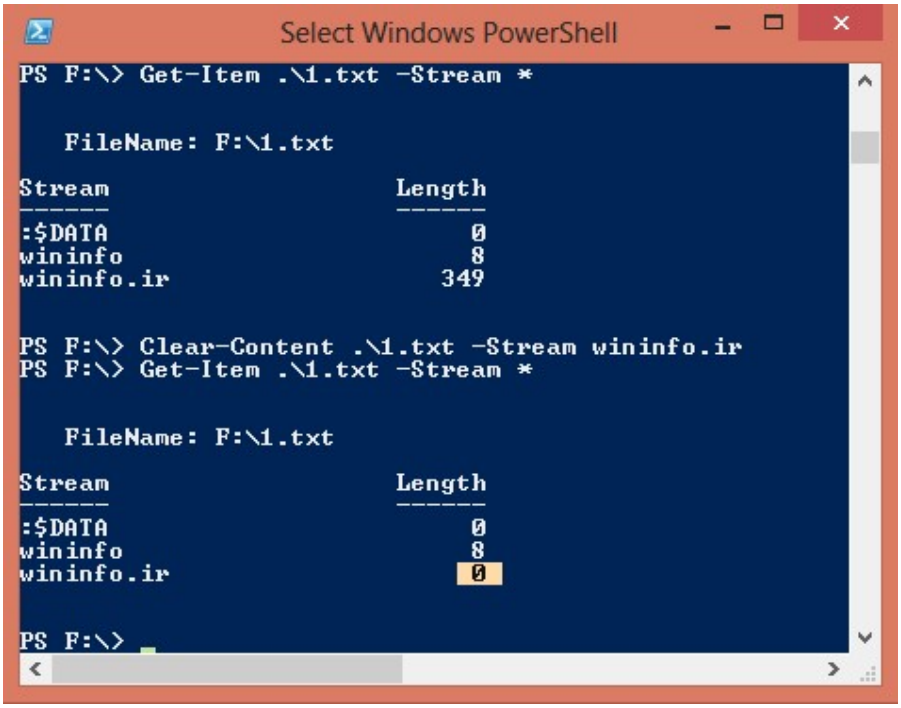


همان‌طور که در شکل بالا می‌بینید Stream بانام moslemADS ساخته‌شده است و مقدار moslem mKST,wininfo به آن داده‌شده است. سپس لیست تمامی Stream های موجود در فایل را بررسی و Stream ساخته‌شده را بافرمان Remove-item حذف کردیم. جهت حذف تمامی مقادیر Steam می‌توان در powershell از فرمان زیر استفاده کرد:

۱۲- پاک کردن محتویات داخل Stream موجود در فایل

بسیاری از مواقع ما نمی‌خواهیم نام Stream از بین برود بلکه می‌خواهیم محتویات آن را از بین ببریم. در این صورت از فرمان زیر استفاده می‌شود.

```
Clear-Content .\1.txt -Stream wininfo.ir
```



در این شکل مقدار wininfo.ir بعد از اجرای دستور ۰ شده است اما نام آن پاک نشده است.

۱۳- حذف کلیه‌ی Stream ها در CMD ویندوز

به کمک فرامین CMD هم امکان حذف Stream ها وجود دارد کافی است فرمان زیر را بنویسید.

```
Type filename > filename
```

```
C:\Windows\system32\cmd.exe
F:\>dir 1.txt /r
 Volume in drive F is software
 Volume Serial Number is 222C-569A

 Directory of F:\

04/25/2014  11:45 PM                 0 1.txt
                                    83 1.txt:wininfo:$DATA
               1 File(s)             0 bytes
               0 Dir(s)  28,623,667,200 bytes free

F:\>type 1.txt > 1.txt

F:\>dir 1.txt /r
 Volume in drive F is software
 Volume Serial Number is 222C-569A

 Directory of F:\

04/25/2014  11:46 PM                 0 1.txt
               1 File(s)             0 bytes
               0 Dir(s)  28,623,667,200 bytes free

F:\>
```

همان‌طور که در بالا می‌بینید ۱.txt:wininfo:$DATA را نشان داده است. اما در اولین گزارش‌گیری بعد از اجرای دستور type 1.txt > 1.txt دیگر Stream ها کامل پاک‌شده‌اند.

۱۴- بررسی stream های Zone.Identifier

هر فایلی که به هر طریقی به یک کامپیوتر وارد می‌شود به‌صورت خودکار یک Stream در قسمت $DATA به فایل اضافه می‌شود که از ۰ تا ۵ شماره‌گذاری می‌شود.

```
Select Select Administrator: Windows PowerShell
PS F:\> Get-Content '.\profile.jpg' -Stream Zone.Identifier
[ZoneTransfer]
ZoneId=3
PS F:\>
```

هرکدام از این شماره‌ها نشان می‌دهند که فایل چگونه وارد سیستم شده است.

0 My Computer
1 Local Intranet Zone
2 Trusted sites Zone
3 Internet Zone
4 Restricted Sites Zone

که این همان لیست موجود در internet option است.

۱۵- مدیریت ADS با C++

زبان برنامه‌نویسی c++ قوی‌ترین و بهترین زبان برنامه‌نویسی برای کار با ADS فایل‌ها است.

ایجاد stream: می‌توان با استفاده از تابع GetVolumeInformation تشخیص داد که یک درایو قابلیت ADS را دارد یا خیر.

```
char szVolName[MAX_PATH], szFSName[MAX_PATH];
DWORD dwSN, dwMaxLen, dwVolFlags;
::GetVolumeInformation("C:\\", szVolName, MAX_PATH, &dwSN,
                       &dwMaxLen, &dwVolFlags, szFSName, MAX_PATH);

if (dwVolFlags & FILE_NAMED_STREAMS) {
  // File system supports named streams
}
else {
  // Named streams are not supported
}
```

همچنین می‌توانید در قسمت شرطی برای فهمیدن NTFS بودن درایو مقایسه‌ای انجام دهید.

```
if (_stricmp(szFSName, "NTFS") == 0)   // If NTFS
```

جهت ایجاد یک استریم برای فایل خاص از تابع CreateFile استفاده کنید بدین شکل:

```
HANDLE hFile = ::CreateFile("file.dat:alt", ...
```

حذف Stream: با استفاده از تابع DeleteFile می‌توانید یک استریم را در فایل NTFS حذف نمایید بدین شکل:

```
::DeleteFile("file.dat:alt");
```

کپی کردن یک استریم: با استفاده از توابع CopyFile/CopyFileEx می‌توان Stream یک فایل را به داخل Stream فایل دیگر کپی نمایید. هنگام انجام عملیات کپی باید جریانی را که دارای نام مشخص است به جریانی دیگر بانام مشخص کپی کنیم، در غیر این صورت امکان بروز نتیجه‌ای غیرمنتظره وجود دارد.

```
HANDLE hInFile = ::CreateFile(szFromStream, GENERIC_READ, FILE_SHARE_READ, NULL,
             OPEN_EXISTING, FILE_FLAG_SEQUENTIAL_SCAN, NULL);
HANDLE hOutFile = ::CreateFile(szToStream, GENERIC_WRITE, FILE_SHARE_READ, NULL,
             CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL | FILE_FLAG_SEQUENTIAL_SCAN, NULL);

BYTE buf[64*1024];
DWORD dwBytesRead, dwBytesWritten;

do {
  ::ReadFile(hInFile, buf, sizeof(buf), &dwBytesRead, NULL);
  if (dwBytesRead) ::WriteFile(hOutFile, buf, dwBytesRead, &dwBytesWritten, NULL);
} while (dwBytesRead == sizeof(buf));
```

```
::CloseHandle(hInFile);
::CloseHandle(hOutFile);
```

حال در مرحله‌ی بعدی ما نیاز داریم که لیستی از تمامی فایل‌ها را در داخل پوشه‌ی خاص به دست آوریم و سپس با استفاده از فرامین بالا به نوشتن و یا خواندن ADS در یک فایل بپردازیم. برای این کار با تعریف فضای نام windows.h دسترسی خود را به کتاب‌خانه‌ی توابع api می‌دهیم و سپس از تابع FindFirstFile برای جستجوی لیست تمامی فایل‌ها استفاده می‌کنیم.

```
WIN32_FIND_DATA file;
HANDLE search_handle=FindFirstFile(L"C:\\*",&file);
if (search_handle) {}{
  do {
    std::wcout << file.cFileName << std::endl;
  } while(FindNextFile(search_handle,&file));
  CloseHandle(search_handle);
}
```

ساختار این تابع به شکل زیر است.

```
HANDLE WINAPI FindFirstFile(
  _In_  LPCTSTR           lpFileName,
  _Out_ LPWIN32_FIND_DATA lpFindFileData
);
```

آرگومان دوم این تابع باید یک متغیر از نوع WIN32_FIND_DATA باید و خروجی که file.cFileName است همان نام فایل‌ها است.