

Forum:[Hack Forums](#)

Thread: [Dynamic Function Resolving GetApi](#)

SIM-3437924 - May 1, 2020, 7:00 am

I have recently been learning about dynamically resolving function to help malware evade AV. Because of this, I was looking through the source of Carberp's "GetApi" files. I understood most of it until I found the function "GetApiAddr".

Because it is called "GetApiAddr" i'm guessing that it's getting the api's address, but can somebody explain how this code is able to do that.

Code #define RVATOVA(base, offset) ((DWORD)base + (DWORD)offset)

DWORD CalcHash(char* Str)

```
{
    if (!Str)
    {
        return -1;
    }

    DWORD dwHash = 0;
    char* CopyStr = Str;

    while (*CopyStr)
    {
        dwHash = ((dwHash << 7) & (DWORD)(-1)) | (dwHash >> (32 - 7));
        dwHash = dwHash ^ (*CopyStr);
        CopyStr++;
    }
    return dwHash;
}
```

LPVOID GetApiAddr(HMODULE hModule, DWORD dwProcNameHash)

```
{
    PIMAGE_OPTIONAL_HEADER poh = (PIMAGE_OPTIONAL_HEADER)((char*)hModule +
((PIMAGE_DOS_HEADER)hModule)->e_lfanew + sizeof(DWORD) +
sizeof(IMAGE_FILE_HEADER));
    PIMAGE_EXPORT_DIRECTORY ped = (IMAGE_EXPORT_DIRECTORY*)RVATOVA(hModule,
poh->DataDirectory[IMAGE_DIRECTORY_ENTRY_EXPORT].VirtualAddress);
```

int nOrdinal;

```
if (HIWORD((DWORD)dwProcNameHash) == 0)
{
    nOrdinal = (LOWORD((DWORD)dwProcNameHash)) - ped->Base;
}
else
{
    DWORD* pdwNamePtr = (DWORD*)RVATOVA(hModule, ped->AddressOfNames);
```

```
WORD* pwOrdinalPtr = (WORD*)RVATOVA(hModule, ped->AddressOfNameOrdinals);
```

```
unsigned int i = 0;
```

```
for (i = 0; i < ped->NumberOfNames; i++, pdwNamePtr++, pwOrdinalPtr++)  
{  
    if (CalcHash((char*)RVATOVA(hModule, *pdwNamePtr)) == dwProcNameHash)  
    {  
        nOrdinal = *pwOrdinalPtr;  
        break;  
    }  
}
```

```
if (i == ped->NumberOfNames)  
{  
    return 0;  
}  
}
```

```
PDWORD pAddrTable = (PDWORD)RVATOVA(hModule, ped->AddressOfFunctions);  
DWORD dwRVA = pAddrTable[nOrdinal];  
DWORD ret = (DWORD)RVATOVA(hModule, dwRVA);
```

```
return (LPVOID)ret;
```

```
}(example of second parameter that would be passed in if calling CreateFileA: "0x8F8F114")
```

also can somebody explain what the hashes are about? how/why they are used?

Sources:

[GetApi.cpp](#)

[GetApi.h](#)

saltycracker - May 1, 2020, 7:10 am

Are you looking for what a hash is and how to use one?

The function CalcHash looks like its just generating a hash value from the character pointer. The hash value is used as a index value into a data structure(usually a array with added features). Why a hash and hash value? Its fast.

dracx - May 1, 2020, 7:59 am

basically runs thru the IMAGE_EXPORT_DIRECTORY structure of the module and hash every function name and see if the hash (of that function name) matches the hash that was passed as the argument. Another reason why hash: t get's rid of those ascii strings in your binary.