

# دوره‌ی آموزش ساختار فایل‌های PE

Site: [onhexgroup.ir](http://onhexgroup.ir)

Youtube: [onhexgroup](https://www.youtube.com/onhexgroup)

Telegram: [onhex\\_ir](https://t.me/onhex_ir)

X: [onhexgroup](https://x.onhexgroup)

Github: [onhexgroup](https://github.com/onhexgroup)

ارائه شده توسط  
**onhexgroup**

Youtube: onhexgroup

## Optional Header

- ImageBase
- SizeOfStackReserve
- SizeOfStackCommit
- SizeOfHeapReserve
- SizeOfHeapCommit

▪ در داخل `winnt.h` با عنوان `IMAGE_OPTIONAL_HEADER32` و `IMAGE_OPTIONAL_HEADER64` تعریف شدن.

## NT Headers

▪ مهمترین هدر داره:

- نسخه ۳۲ بیتی ۳۱ عضو (`BaseOfData`) و نسخه ۶۴ بیتی ۳۰ عضو دارد.
- فیلدهای زیر در ۳۲ بیتی سایز `DWORD` اما در ۶۴ بیتی `ULONGLONG`:

Site: onhexgroup.ir

ساختار  
Optional  
HEADER

: مشخص کننده معماری فایل اجرایی هستش. ۳ تا مقدار

داره:

PE32 : 10B ■

PE32+ : 20B ■

ROM Image : 107 ■

: نسخه اصلی لینکر

: نسخه فرعی یا جزئی لینکر

x: onhexgroup

ساختار  
Optional  
HEADER

: سایز بخش کد **SizeOfCode** ■

: سایز بخش داده هایی که **SizeOfInitializedData** ■

مقداردهی اولیه شدن. (.data - .rdata)

: سایز بخش داده هایی که **SizeOfUninitializedData** ■

مقداردهی اولیه نشدن. (.bss)

Telegram: onhex\_ir

ساختار  
Optional  
HEADER

■ آدرس نقطه ورود به برنامه: **AddressOfEntryPoint** ■

■ EP یا

■ آدرس مجازی نسبی (RVA)

■ عموماً در:

■ برنامه ها، اولین دستور اجرایی

■ در درایورها تابع **DriverInit** یا **DriverEntry**

■ در DLL تابع **DllMain** اما الزامی نیست، اگه نباشه

■ مقدارش صفره

Github: onhexgroup

ساختار  
Optional  
HEADER

آدرس (RVA) شروع بخش کد: **BaseOfCode** ■  
آدرس (RVA) شروع بخش داده-  
 فقط در ۳۲ بیتی

Github: onhexgroup

ساختار  
Optional  
HEADER

■ آدرس ترجیحی اولین بایت در مموری **ImageBase** ■

■ باید مضربی از **64k** (0x10000) ■

■ بدلیل ASLR (Address Space Layout Randomization) نادیده گرفته میشه.

■ **relocation section (.reloc)** ■

■ مقدار پیش فرض برای DLLها 0x10000000 برای Windows CE و بقیه 0x10000 برابر 0x400000

Site: onhexgroup.ir

ساختار  
Optional  
HEADER

مشخص کننده سایز تراز هر **SectionAlignment** ■

سکشن در مموری. (آدرس شروع مضربی از این مقدارست)

$4\text{kb}=4096\text{b}=0x1000$  ■ مقدار پیش فرض یک صفحه =

باید بزرگتر یا مساوی **FileAlignment** (سکتور/صفحه) ■

مشخص کننده سایز تراز هر سکشن در **FileAlignment** ■

فایل. (آدرس شروع مضربی از این مقدارست)

مقدار پیش فرض معمولاً  $0x200=512$  ■

Youtube: onhexgroup

ساختار  
Optional  
HEADER

: MajorOperatingSystemVersion ■

: MinorOperatingSystemVersion ■

نسخه‌ی حداقلی اصلی و فرعی سیستم عامل رو نشون  
میده.

لیست کامل ■

Youtube: onhexgroup

ساختار  
Optional  
HEADER

**:MajorImageVersion** ■

**:MinorImageVersion** ■

■ نسخه اصلی و فرعی فایل رو نشون میده.

Youtube: onhexgroup

ساختار  
Optional  
HEADER

## **:MajorSubsystemVersion** ■

## **:MinorSubsystemVersion** ■

- نسخه‌ی حداقلی اصلی و فرعی زیرسیستم رو مشخص میکنن.
- منظور از Subsystem: محیط اجرای باینری. مثلا کامندلاین، گرافیکی و ...

Github: onhexgroup

ساختار  
Optional  
HEADER

■ **Win32VersionValue**: یک مقدار رزرو شده که همیشه برابر صفر.

■ **SizeOfImage**: مشخص کننده سایز باینری در مموری به مضربی از **SectionAlignment** گرد میشے.

■ **DOS** و **DOS Header** مجموع **SizeOfHeaders** **Section Headers** و **NT Headers** و **stub** به مضربی از **FileAlignment** گرد میشے.

■ در حالت کلی اگه اندازه فایل رو منهای اندازه کل **Section** ها کنیم، این مقدار بدست میاد یا به عبارتی میشه گفت، این مقدار مشخص کننده آفست اولین **Section** هستش .



■ **Checksum** : یک مقدار عددی که برای بررسی یکپارچگی

فایل استفاده میشه.

■ داخل **IMAGHELP.DLL** تعریف شده. ([MapFileAndCheckSumA](#))

■ برای موارد زیر محاسبه و بررسی میشه:

■ همهی درایورها

■ DLLهایی که در زمان بوت لود میشن (Ntdll و ...)

■ kernel32هایی که وارد پروسس های حیاتی ویندوز میشن (

( و ... )

Youtube: onhexgroup

ساختار  
Optional  
HEADER

: محیط اجرای باینری رو مشخص میکنه.

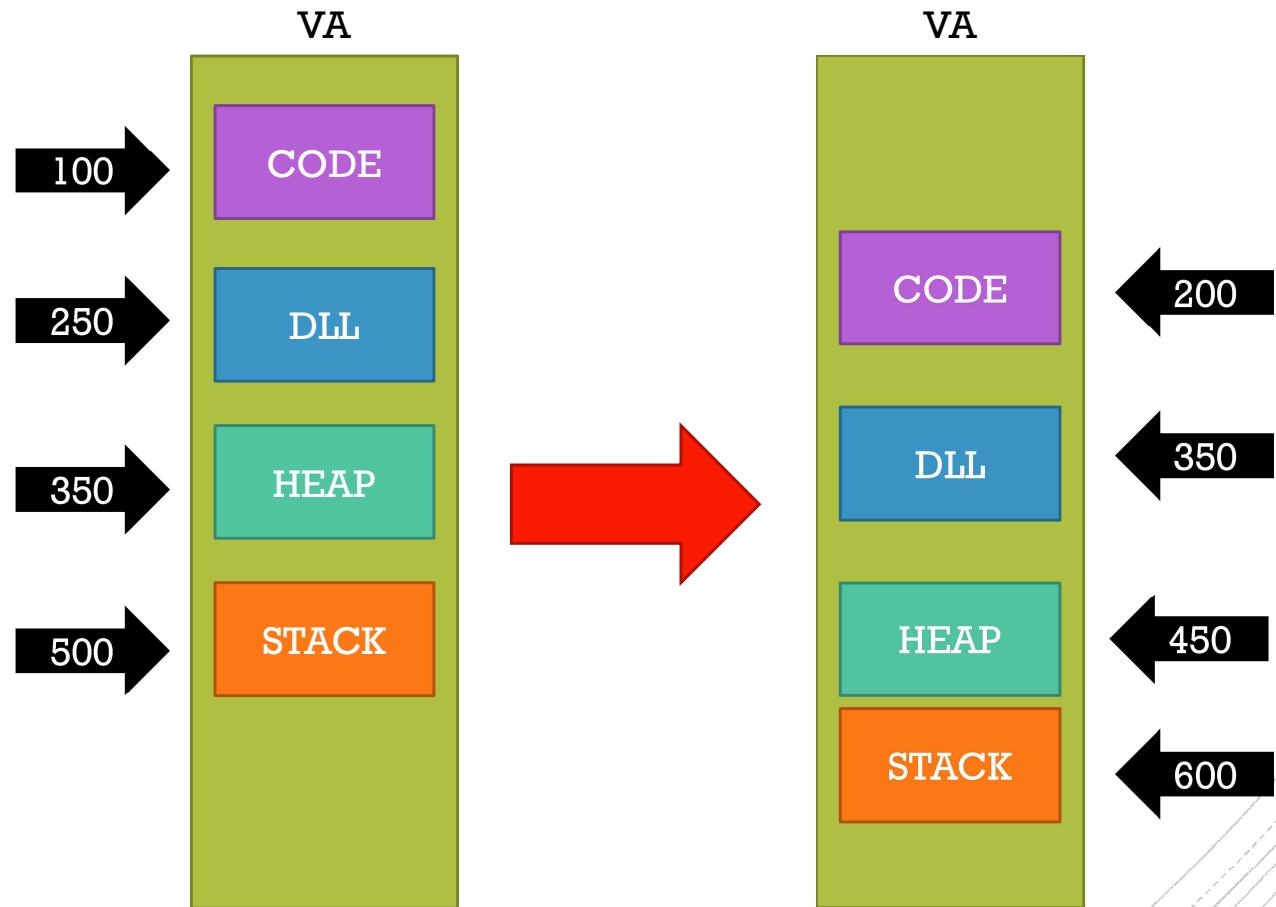
لیست

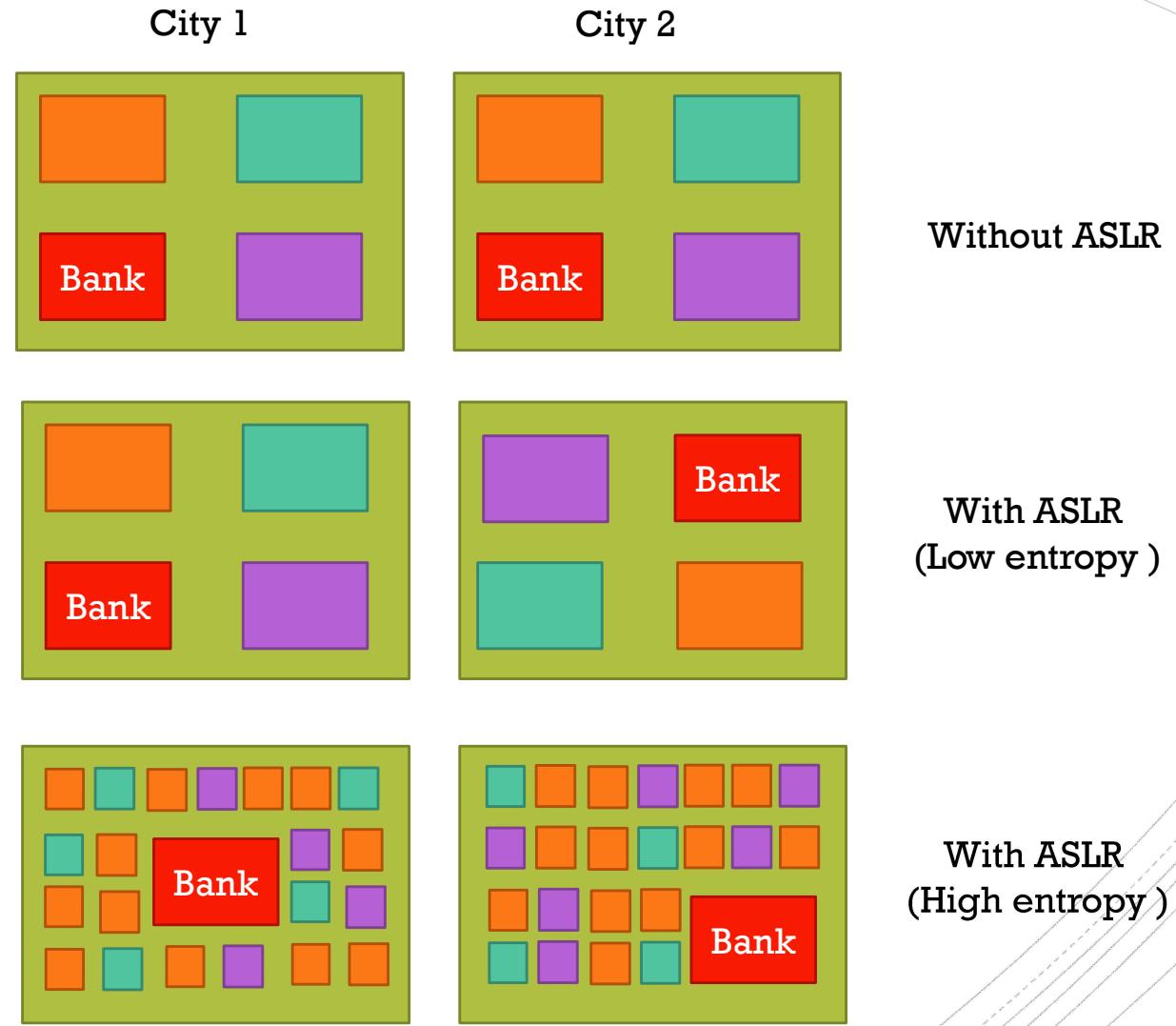
**DLLCharacteristics** : مشخص کننده یسری ویژگی امنیتی

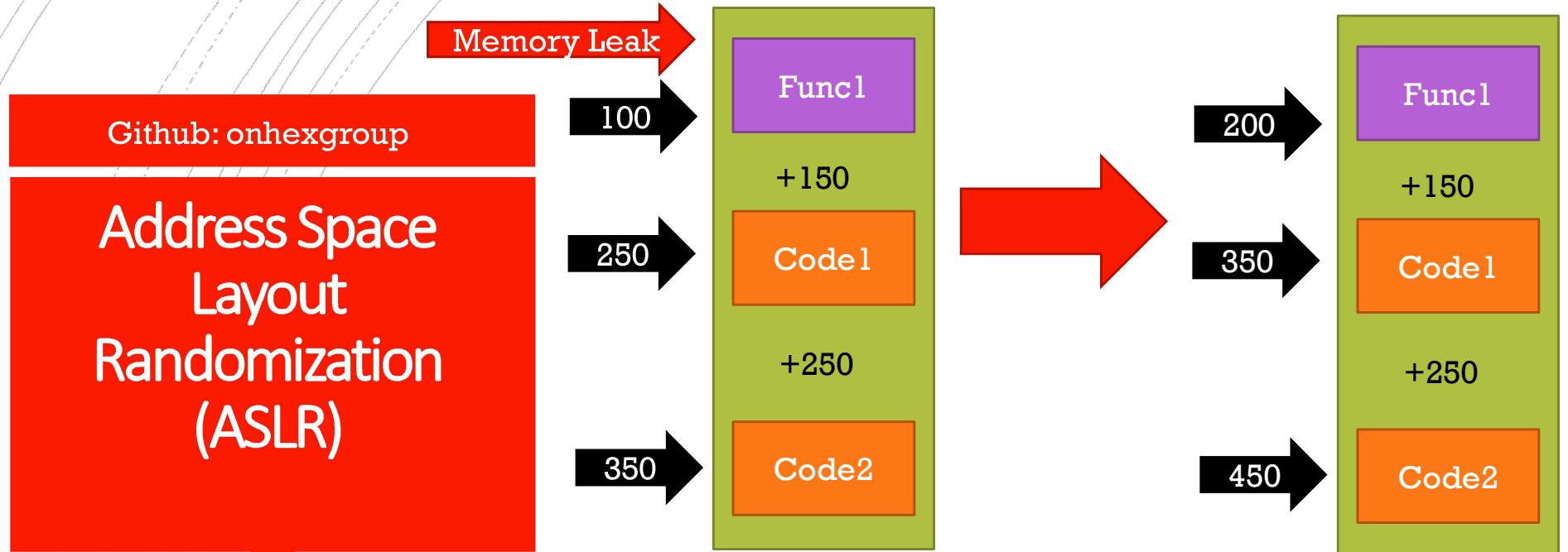
و اجرایی فایل.

لیست

**BitField** بصورت







Youtube: onhexgroup

## Address Space Layout Randomization (ASLR)

■ از ویندوز Vista معرفی و رفته رفته بهبود داده شد.

■ STACK و HEAP و DLL و EXE با هر اجرا (و ریبوت) آدرسشنون تغییر پیدا میکنه.

■ درایورها و کرنل (ntoskrnl.exe) با هر Reboot (با هر

■ نکته: تصمیم گیرنده نهایی لوودر هستش.

Youtube: onhexgroup

## Address Space Layout Randomization (ASLR)

▪ باید سیستم عامل و باینری این ویژگی را پشتیبانی کن.

▪ ویندوزهای مدرن پشتیبانی میکن:

- Windows Security > App & Browser Control > Exploit Protection
- Get-ProcessMitigation -System

بررسی باینری:

### ▪ در DLLCharacteristics فیلد Optional Header :

- IMAGE\_DLLCHARACTERISTICS\_DYNAMIC\_BASE
- IMAGE\_DLLCHARACTERISTICS\_HIGH\_ENTROPY\_VA

▪ پاورشل (پروسس/باینری):

- Get-Process Your\_process | Get-ProcessMitigation
- رجیستری: (آنالیز)
- HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\

▪ در ویژوال استدیو (پیش فرض فعل):

- Project Properties → Linker → Advanced → Randomized Base Address = "Yes (/DYNAMICBASE)"
- /HIGHENTROPYVA

Address Space Layout Randomization (ASLR)

Youtube: onhexgroup

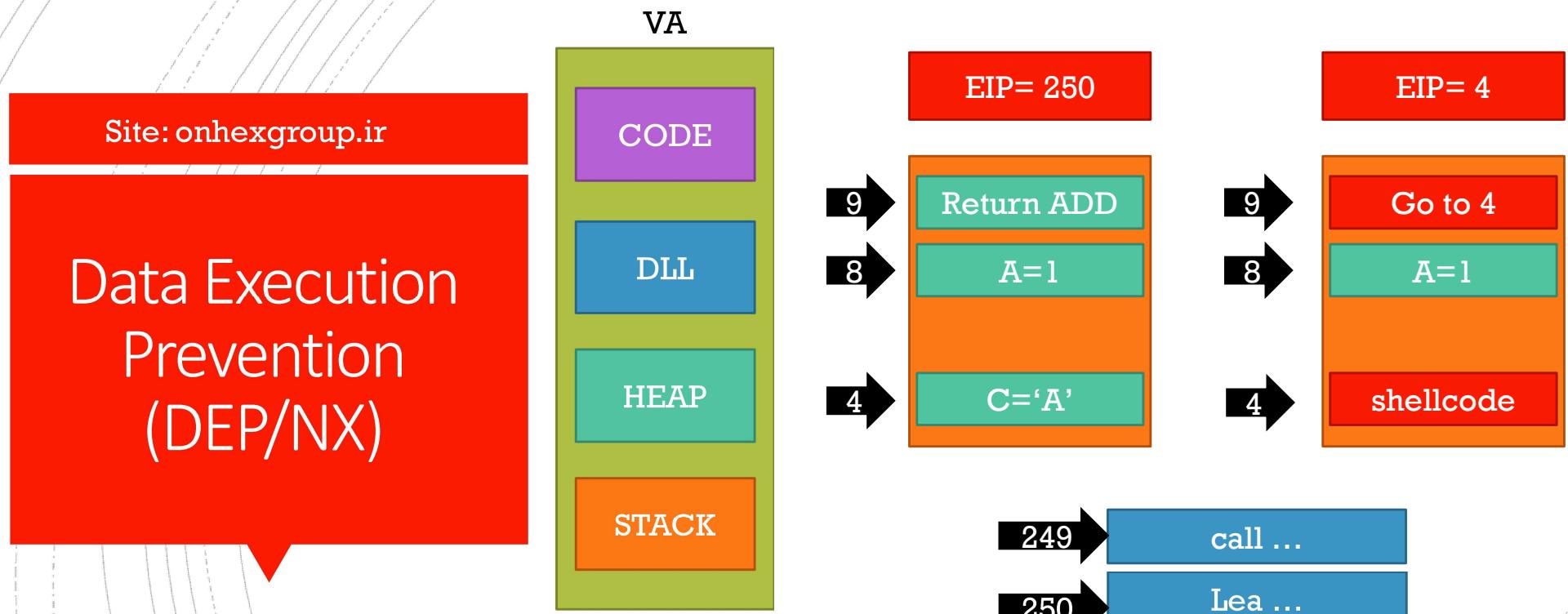
Site: onhexgroup.ir

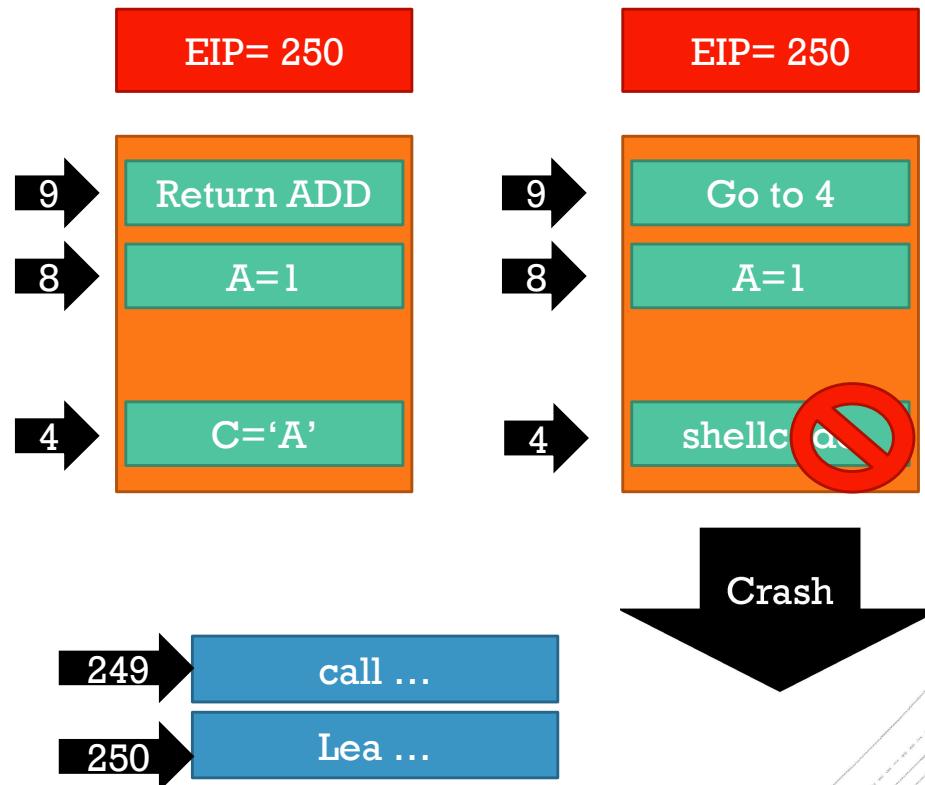
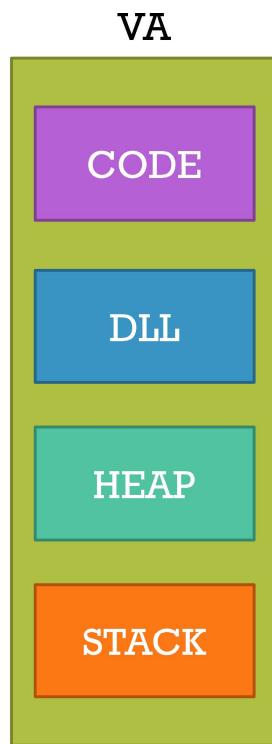
FORCE  
INTEGRITY

## IMAGE\_DLLCHARACTERISTICS\_FORCE\_INTEGRITY

سیستم عامل هنگام بارگذاری فایل PE امضا دیجیتالش رو بررسی میکنه و در صورت نامعتبر بودن امضا از اجرаш جلوگیری میکنه.

/INTEGRITYCHECK در ویژوال استدیو:





# Data Execution Prevention (DEP/NX)

Youtube: onhexgroup

▪ مخفف NX (Never Execute) یا No-eXecute) یک پیاده

سازی سخت افزاری هستش.

▪ مخفف DEP (DATA Execution Prevention) پیاده

سازی مايكروسافت از NX

▪ باید سیستم عامل و باينری این ويژگی رو پشتيبانی کن.

▪ ويندوزهاي مدرن پشتيبانی ميکنن:

- Windows Security > App & Browser Control >  
Exploit Protection
- Get-ProcessMitigation -System

بررسی باینری:

: DLLCharacteristics فیلد Optional Header در

- **IMAGE\_DLLCHARACTERISTICS\_NX\_COMPAT**

▪ پاورشل (پروسس/باینری):

- Get-Process Your\_process | Get-ProcessMitigation

▪ رجیستری: (آنالیز)

- HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\

▪ در ویژوال استدیو (پیش فرض فعل):

- Project Properties → Linker → Advanced → Data Execution Prevention (DEP) = "Yes (/NXCOMPAT)"

Youtube: onhexgroup

# Data Execution Prevention (DEP/NX)

Youtube: onhexgroup

# Structured Exception Handling (SEH)

▪ استثناء: رویدادهای غیرمنتظره که در طول اجرای برنامه رخ میده و برنامه نمیتونه مدیریتش کنه.

## ▪ انواع استثناء:

- سخت افزاری: مثلا دسترسی به یک آدرس نامعتبر توسط CPU
- نرم افزاری: توسط برنامه یا سیستم عامل. مثلا تابع `RaiseException`

```
open myfile.txt  
...
```

```
Try{  
    open myfile.txt  
    ....  
}  
Catch{  
    Show("File Not Find!")  
}
```

# Structured Exception Handling (SEH)

Youtube: onhexgroup

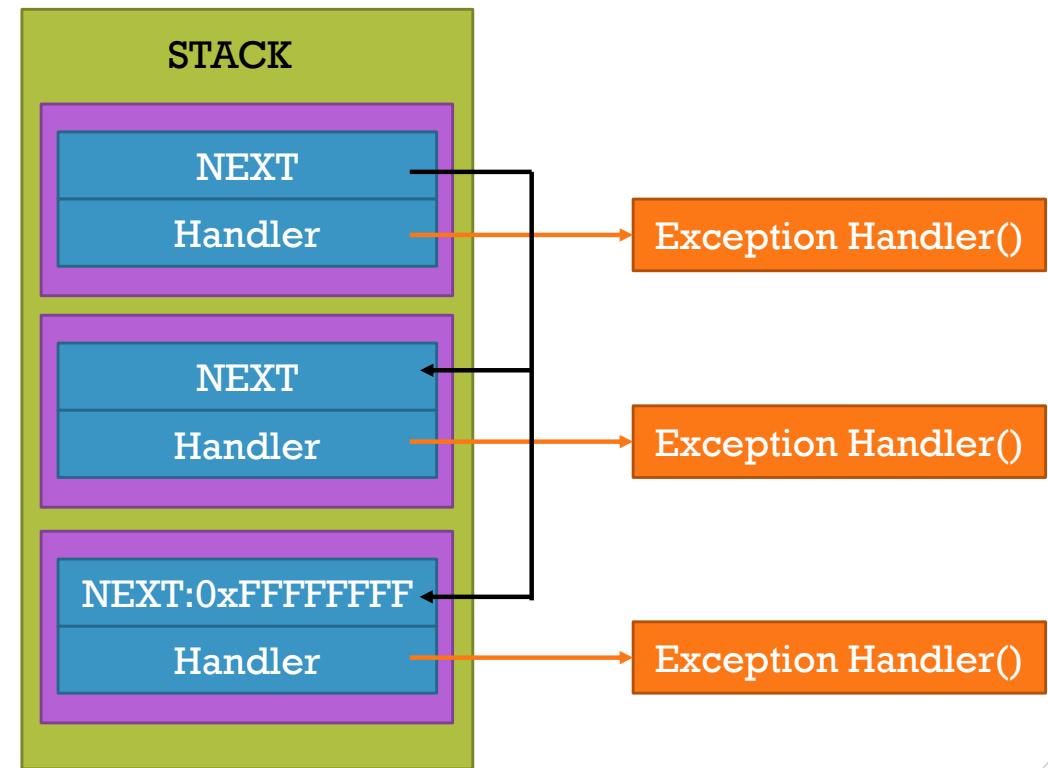
■ یک مکانیسم مدیریت استثناء در سطح سیستم عامل هستش.

■ وقتی در طول اجرای برنامه، یک **Thread** با یک استثناء مواجه میشه، سیستم عامل مجموعه مشخصی از توابع رو فراخوانی میکنه (اون خطا رو اصلاح کنن یا **Exception Handlers**) اطلاعات بیشتری در خصوصش بدن.

■ این مکانیسم موقع کامپایل به کدهای ما اضافه میشه و موقع اجرا با ساختارهایی در پشته (۳۲ بیتی) و جدول استثنائات (۶۴ بیتی) کار میکنه.

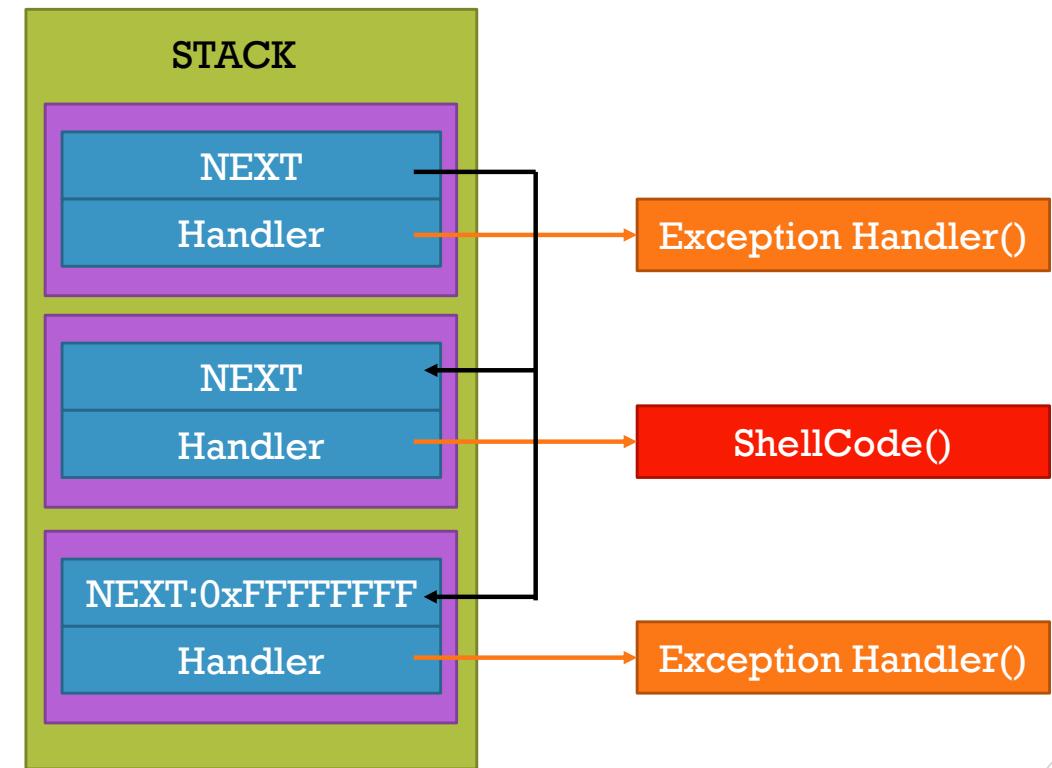
Youtube: onhexgroup

# Structured Exception Handling (SEH)



Youtube: onhexgroup

# Structured Exception Handling (SEH)



# Structured Exception Handling (SEH)

Youtube: onhexgroup

- در **DLLCharacteristics** فیلد **Optional Header**:
  - **IMAGE\_DLLCHARACTERISTICS\_NO\_SEH**
  - در ویژوال استدیو:
    - Project Properties → Linker → Advanced →
    - Command line: /NOSEH

Telegram: onhex\_ir

## APP CONTAINER

■ یک محیط سندباکس در ویندوز برای اجرای امن برنامه ها

■ بعد از ویندوز ۸ معرفی شد.

■ برنامه هایی که معمولاً داخل **AppContainer** اجرا میشند:

■ اپلیکیشن های (**Universal Windows Platform**)

■ برخی برنامه های **Packaged Win32**

■ نسخه های قدیمی مرورگر **Microsoft Edge**

■ در **Optional Header** **DLLCharacteristics** فیلد

■ **IMAGE\_DLLCHARACTERISTICS\_APPCONTAINER**

Github: onhexgroup

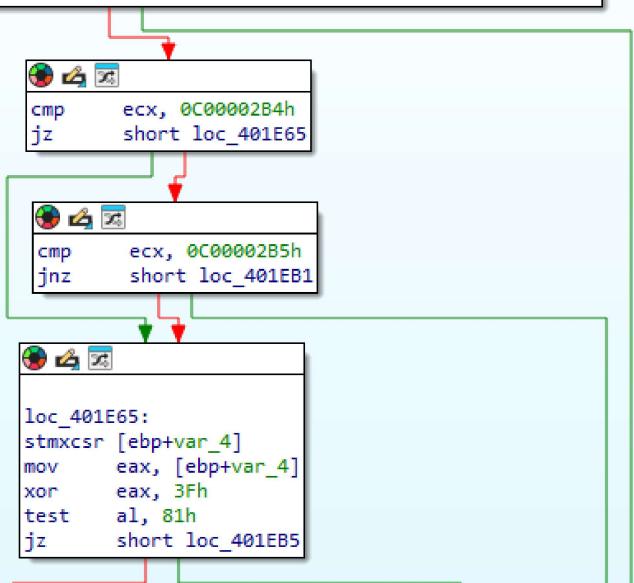
CFG ?  
Control Flow  
Graph

## Control Flow Graph (CFG) ■

■ یک ساختار گرافیکی از مسیر اجرای برنامه

```
var_4= dword ptr -4
_exception_code= dword ptr  8

push  ebp
mov   ebp, esp
push  ecx
cmp   __isa_available, 1
mov   ecx, [ebp+_exception_code]
j1    short loc_401EB1
```





- مکانیسم امنیتی برای جلوگیری از تغییر جریان اجرای برنامه که از ویندوز ۸.۱ معرفی شد.

- موقع کامپایل فراخوانی های غیر مستقیم شناسایی و کد بررسی قبلش اضافه میشه. مقصد پرش های غیرمستقیم مجاز، لیست میشه.

- موقع اجرا، در فراخوانی غیرمستقیم، کد اضافه شده برای معتر بودن آدرس اجرا میشه. اگه آدرس معتر باشه اجرا، در غیر اینصورت از اجرای کد جلوگیری میشه.

- در **DLLCharacteristics** فیلد **Optional Header**

- **IMAGE\_DLLCHARACTERISTICS\_GUARD\_CF**

- در ویژوال استدیو:

- **Configuration Properties → C/C++ → Code Generation**

- **Control Flow Guard : Yes (/guard:cf)**

Github: onhexgroup

CFG ?  
Control Flow  
Guard

# Windows Driver Model (WDM)

Youtube: onhexgroup

- یک مدل قدیمی برای نوشتن درایور

- هدفش ایجاد یک استاندارد واحد بود تا درایورها بتوانن هم روی ویندوزهای خانواده 9x و هم خانواده NT کار کنن.

- در سطح کرنل اجرا میشن و دسترسی مستقیم به سختافزار دارن.

- در **DLLCharacteristics** فیلد **Optional Header**

- **IMAGE\_DLLCHARACTERISTICS\_WDM\_DRIVER**

- در ویژوال استدیو:

- Configuration Properties → Linker → System → Driver - WDM  
(/DRIVER:WDM)

Youtube: onhexgroup

# TERMINAL SERVER

قابلیتی در ویندوز سرور که به چندین کاربر اجازه میده بصورت همزمان و از راه دور به یک سرور متصل و هر کدام دسکتاپ و برنامه های خودشون رو داشته باشند.

برنامه ها برای اجرا در چنین محیطی باید این قابلیت رو داشته باشند.

:**DLLCharacteristics** فیلد **Optional Header** در

- **IMAGE\_DLLCHARACTERISTICS\_TERMINAL\_SERVER\_AWARE**

در ویژوال استدیو:

- Configuration Properties → Linker → System → Terminal Server - Yes (/TSAWARE)