دوره ی آموزش ساختار فایلهای PE

Site: onhexgroup.ir

Youtube: onhexgroup

Telegram: onhex_ir

X: onhexgroup

Github: onhexgroup

ارائه شده توسط

onhexgroup

Youtube: onhexgroup

Optional Header

- مهمترین هدر **NT Headers**
- دو تا نسخه داره:
  - نسخه ۳۲ بیتی ۳۱ عضو (**BaseOfData**) و نسخه ی ۶۴ بیتی ۳۰ عضو داره.
  - فیلدهای زیر در ۳۲ بیتی سایز **DWORD** اما در ۶۴ بیتی **ULONGLONG** :

- **ImageBase**
- **SizeOfStackReserve**
- **SizeOfStackCommit**
- **SizeOfHeapReserve**
- **SizeOfHeapCommit**

- در داخل **winnt.h** با عنوان **IMAGE_OPTIONAL_HEADER32** و **IMAGE_OPTIONAL_HEADER64** تعریف شدن.

Site: onhexgroup.ir

ساختار Optional HEADER

- **Magic**: مشخص کننده معماری فایل اجرایی هستش. ۳ تا مقدار داره:
  - PE32 : 10B
  - PE32+:20B
  - ROM Image :107
- **MajorLinkerVersion**: نسخه ی اصلی لینکر
- **MinorLinkerVersion**: نسخه ی فرعی یا جزئی لینکر

x: onhexgroup

ساختار Optional HEADER

- **SizeOfCode**: سایز بخش کد (.text)
- **SizeOfInitializedData**: سایز بخش داده هایی که مقداردهی اولیه شدن. (.data - .rdata)
- **SizeOfUninitializedData**: سایز بخش داده هایی که مقداردهی اولیه نشدن. (.bss)

Telegram: onhex_ir

ساختار Optional HEADER

- **AddressOfEntryPoint**: آدرس نقطه ورود به برنامه یا **EP**
  - آدرس مجازی نسبی (**RVA**)
  - معمولا در:
    - برنامه ها، اولین دستور اجرایی
    - در درایورها تابع **DriverEntry** یا **DriverInit**
    - در **DLL** تابع **DllMain** اما الزامی نیست، اگه نباشه مقدارش صفره

Github: onhexgroup

ساختار Optional HEADER

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

Github: onhexgroup

ساختار Optional HEADER

- **ImageBase**: آدرس ترجیحی اولین بایت در مموری
  - باید مضربی از **64k** (**0x10000**)
  - بدلیل **ASLR (Address Space Layout Randomization)** نادیده گرفته میشه.
  - **relocation section (.reloc)**
  - مقدار پیش فرض برای **DLL**ها **0x10000000** برای **Windows CE** برابر **0x10000** و بقیه **0x400000**

Site: onhexgroup.ir

ساختار Optional HEADER

- **SectionAlignment** : مشخص کننده سایز تراز هر سکشن در مموری. (آدرس شروع مضربی از این مقدارست)
  - مقدار پیش فرض یک صفحه = 4kb=4096b=0x1000
  - باید بزرگتر یا مساوی **FileAlignment** (سکتور/صفحه)
- **FileAlignment** : مشخص کننده سایز تراز هر سکشن در فایل. (آدرس شروع مضربی از این مقدارست)
  - مقدار پیش فرض معمولا ۵۱۲=0x200

Youtube: onhexgroup

ساختار Optional HEADER

- **MajorOperatingSystemVersion** :
- **MinorOperatingSystemVersion** :
- نسخه ی حداقلی اصلی و فرعی سیستم عامل رو نشون میده.
- لیست کامل

Youtube: onhexgroup

ساختار Optional HEADER

- **MajorImageVersion:**
- **MinorImageVersion:**
- نسخه ی اصلی و فرعی فایل رو نشون میده.

Youtube: onhexgroup

ساختار Optional HEADER

- **MajorSubsystemVersion:**
- **MinorSubsystemVersion:**
- نسخه ی 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 میشه.
- **SizeOfHeaders**: مجموع **DOS Header** و **DOS stub** و **NT Headers** و **Section Headers**
  - به مضربی از **FileAlignment** گرد میشه.
  - 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 .

Telegram: onhex_ir

ساختار Optional HEADER

- **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
  - **DLL**هایی که در زمان بوت لود میشن (**Ntdll** و ...)
  - **DLL**هایی که وارد پروسس های حیاتی ویندوز میشن (**kernel32** و ...)

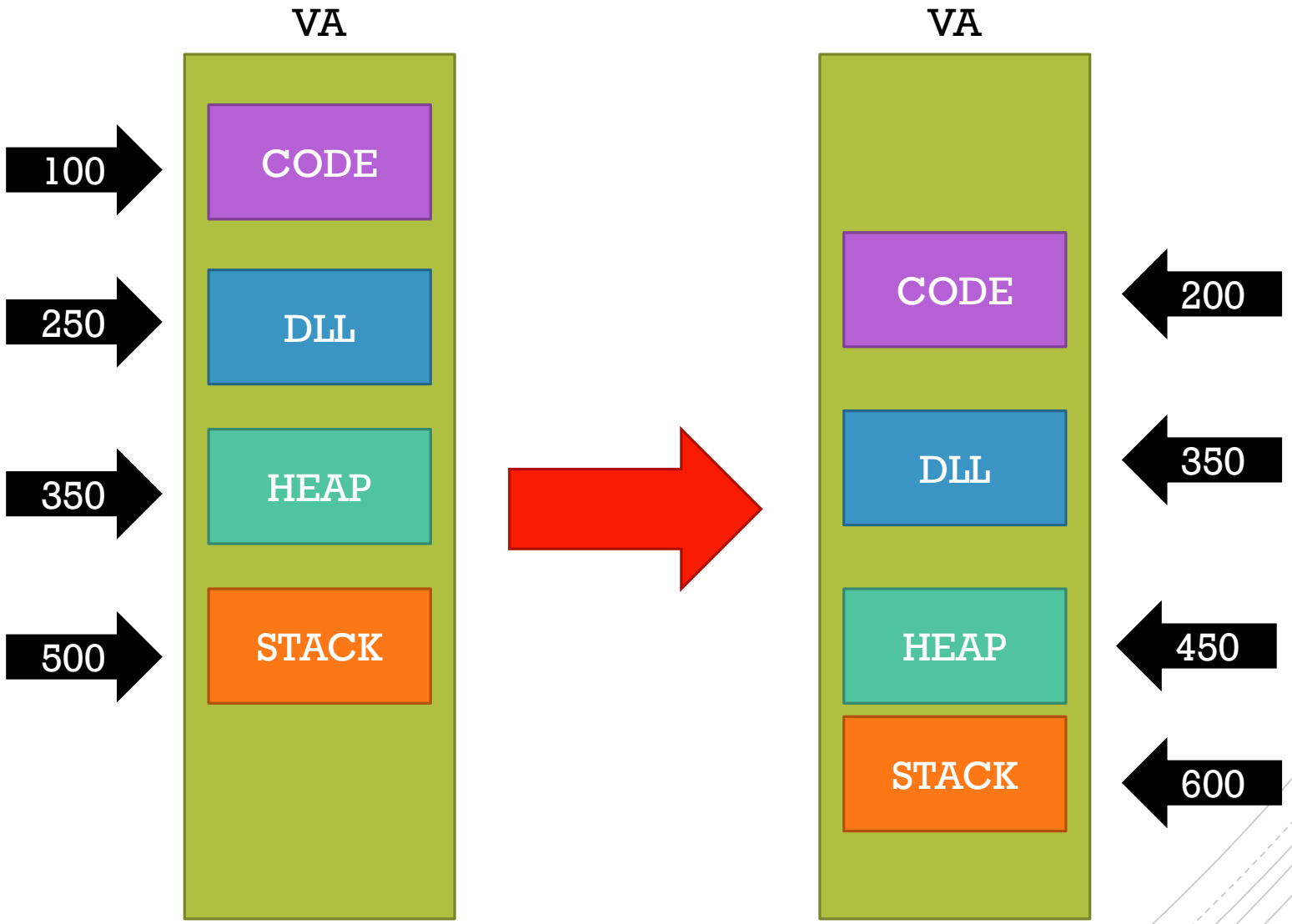
Youtube: onhexgroup

ساختار Optional HEADER

- **Subsystem** : محیط اجرای باینری رو مشخص میکنه.
  - لیست
- **DLLCharacteristics**:مشخص کننده یسری ویژگی امنیتی و اجرایی فایل.
  - لیست
  - بصورت BitField

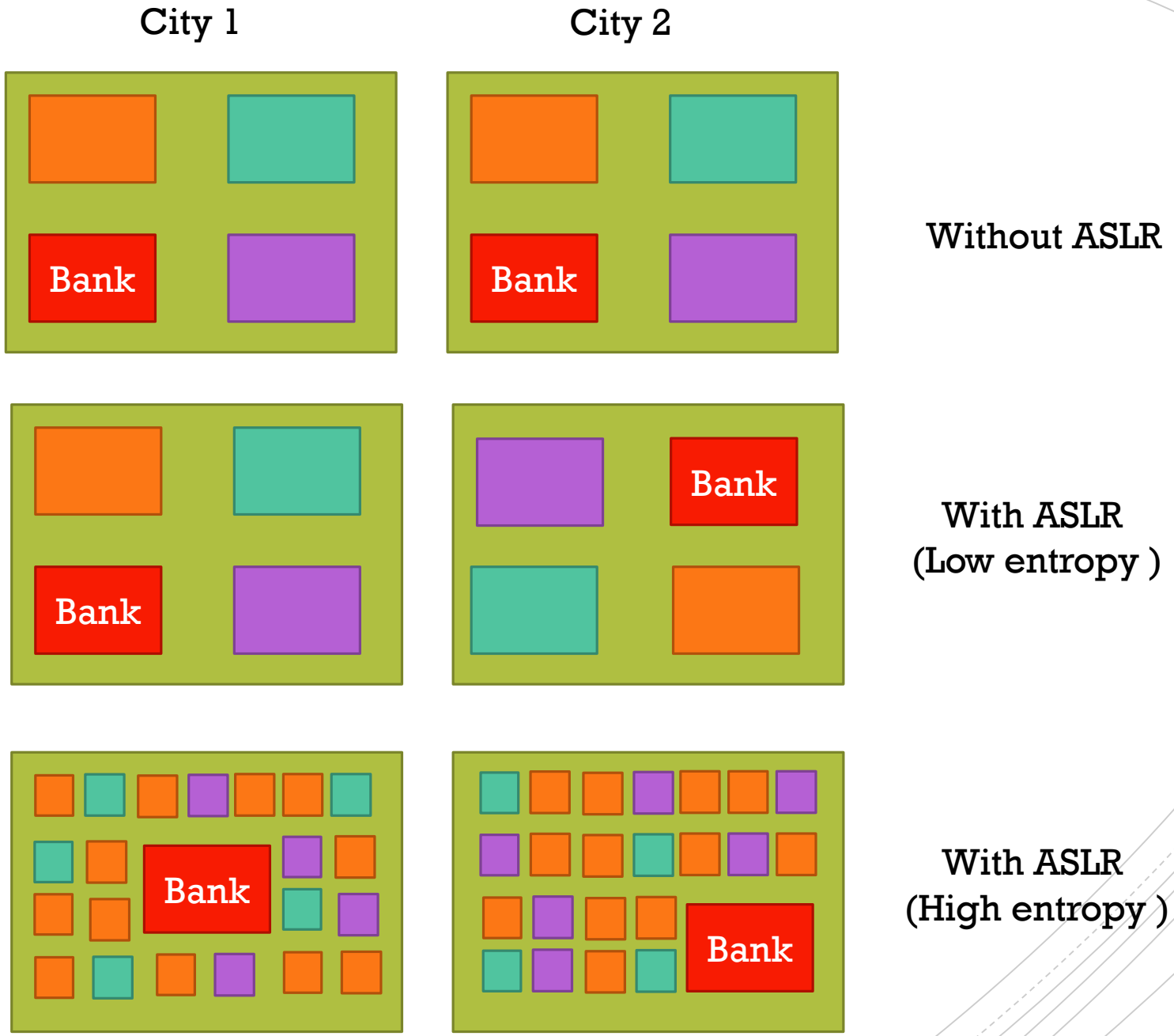
Github: onhexgroup

Address Space Layout Randomization (ASLR)



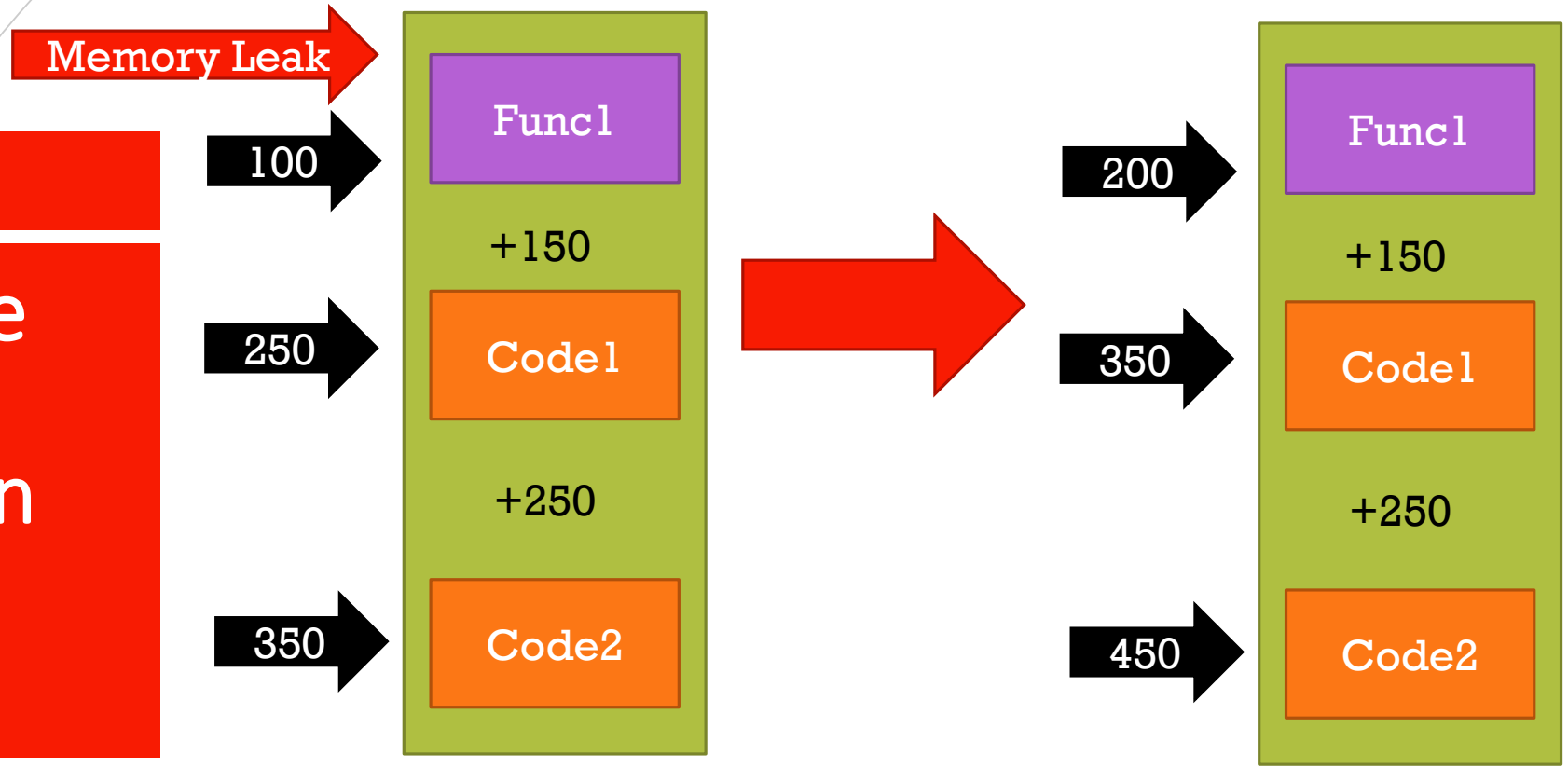
Github: onhexgroup

Address Space Layout Randomization (ASLR)



Github: onhexgroup

Address Space Layout Randomization (ASLR)



Youtube: onhexgroup

Address Space Layout Randomization (ASLR)

- از ویندوز **Vista** معرفی و رفته رفته بهبود داده شد.
- **EXE** و **DLL** و **HEAP** و **STACK** با هر اجرا (و ریبوت) آدرسشون تغییر پیدا میکنه.
- درایورها و کرنل (**ntoskrnl.exe**) با هر **Reboot**
- نکته: تصمیم گیرنده نهایی لوودر هستش.

Youtube: onhexgroup

Address Space Layout Randomization (ASLR)

- باید سیستم عامل و باینری این ویژگی رو پشتیبانی کنن.
  - ویندوزهای مدرن پشتیبانی میکنن:

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

Youtube: onhexgroup

Address Space Layout Randomization (ASLR)

- بررسی باینری:
  - در Optional Header فیلد DLLCharacteristics :
    - 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 فعال) :
    - Project Properties → Linker → Advanced → Randomized Base Address = "Yes (/DYNAMICBASE)
    - /HIGHENTROPYVA

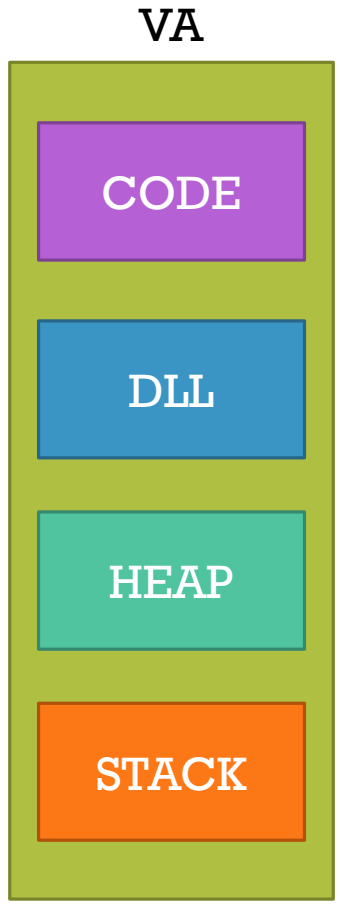
Site: onhexgroup.ir

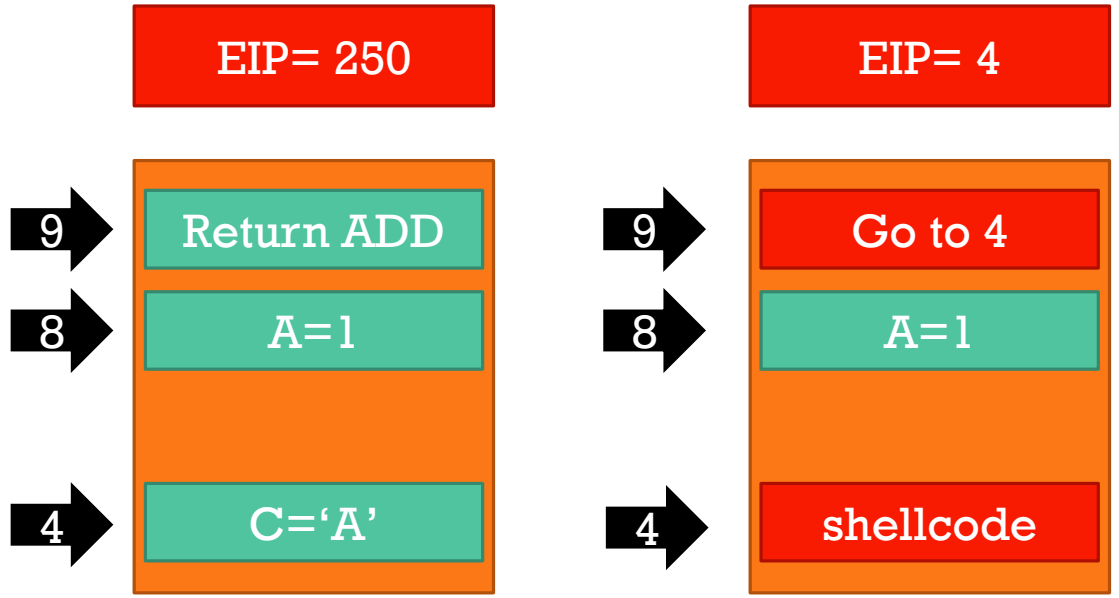
FORCE INTEGRITY

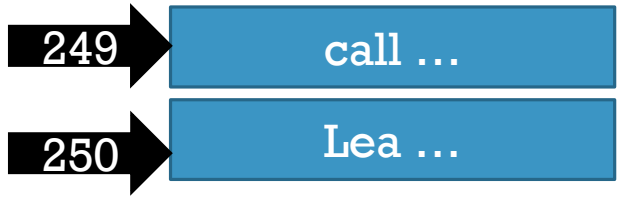
- IMAGE_DLLCHARACTERISTICS_FORCE_INTEGRITY:
- سیستم‌عامل هنگام بارگذاری فایل PE امضای دیجیتالش رو بررسی میکنه و در صورت نامعتبر بودن امضا از اجراش جلوگیری میکنه.
- در ویژوال استدیو: /INTEGRITYCHECK

Site: onhexgroup.ir

Data Execution Prevention (DEP/NX)

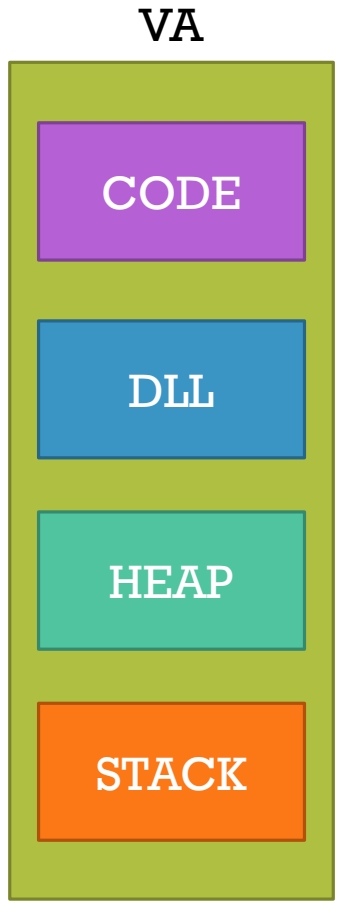


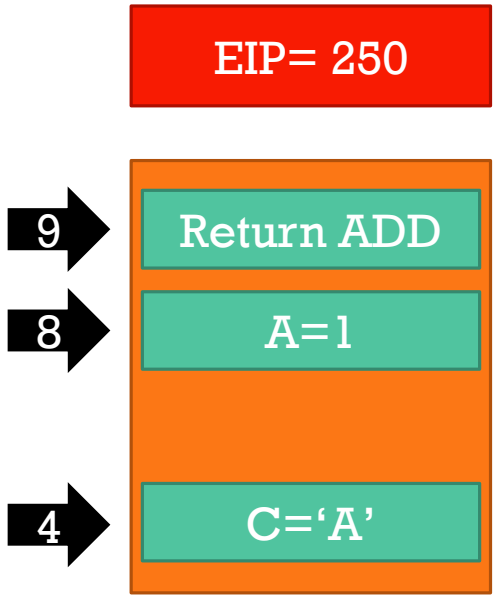


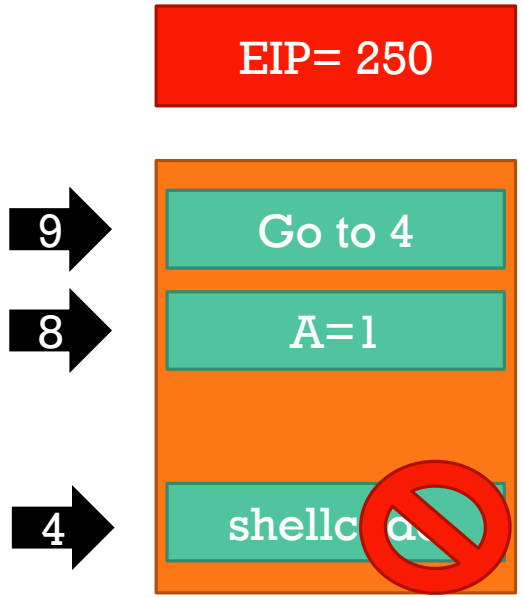


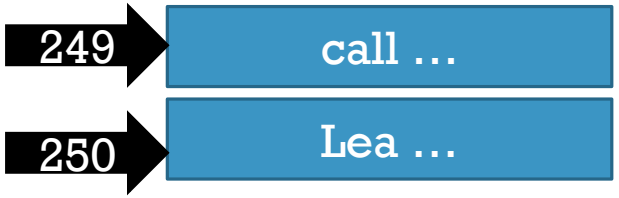
twitter: onhexgroup

Data Execution Prevention (DEP/NX)









Youtube: onhexgroup

Data Execution Prevention (DEP/NX)

- **NX** مخفف (**No-eXecute** یا **Never Execute**) یک پیاده سازی سخت افزاری هستش.
- **DEP** مخفف (**DATA Execution Prevention**) پیاده سازی مایکروسافت از **NX**.
- باید سیستم عامل و باینری این ویژگی رو پشتیبانی کنن.
  - ویندوزهای مدرن پشتیبانی میکنن:
- **Windows Security > App & Browser Control > Exploit Protection**
- **Get-ProcessMitigation -System**

Youtube: onhexgroup

Data Execution Prevention (DEP/NX)

- بررسی باینری:
  - در Optional Header فیلد DLLCharacteristics :
- 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 فعال) :
- Project Properties → Linker → Advanced → Data Execution Prevention (DEP) = "Yes (/NXCOMPAT)

Youtube: onhexgroup

Structured Exception Handling (SEH)

- استثناء: رویدادهای غیر منتظره 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!")
}
```

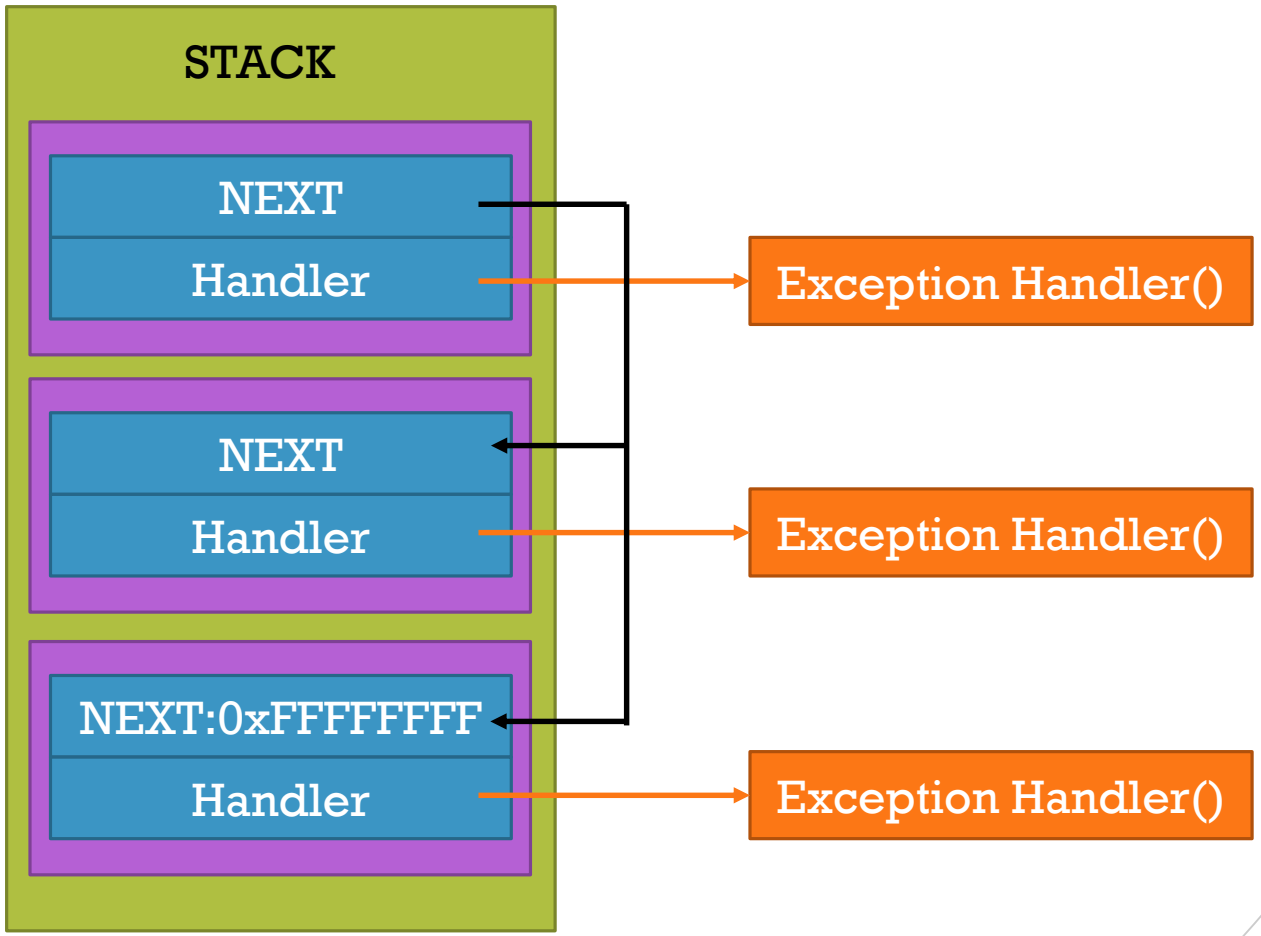
Youtube: onhexgroup

Structured Exception Handling (SEH)

- **SEH** 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 (**Exception Handlers**) تا اون خطا رو اصلاح کنن یا 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.

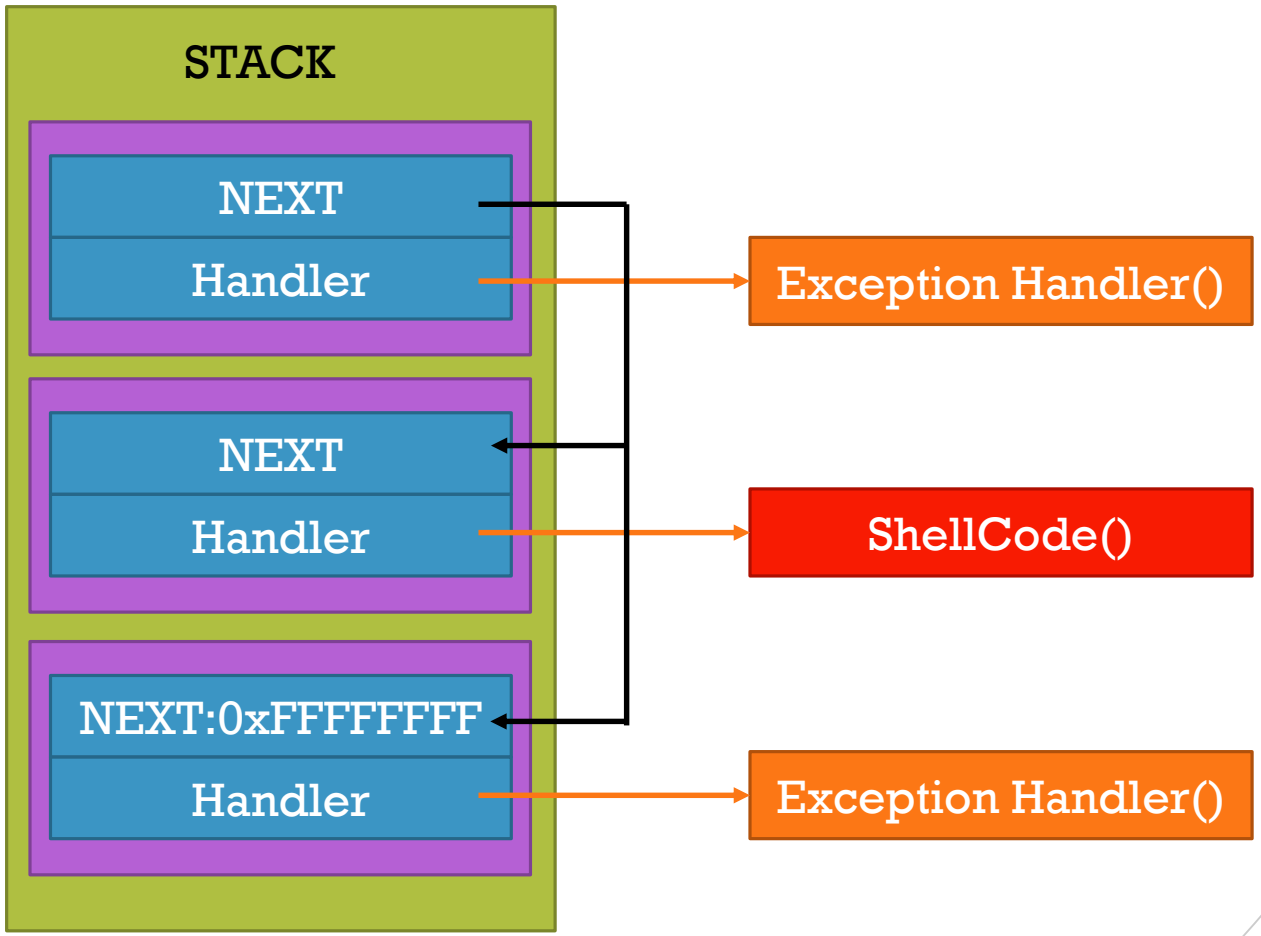
Youtube: onhexgroup

Structured Exception Handling (SEH)



Youtube: onhexgroup

Structured Exception Handling (SEH)



Youtube: onhexgroup

Structured Exception Handling (SEH)

- بررسی باینری:
  - در Optional Header فیلد DLLCharacteristics:
- IMAGE_DLLCHARACTERISTICS_NO_SEH
  - در ویژوال استدیو:
- Project Properties → Linker → Advanced → Command line: /NOSEH

Telegram: onhex_ir

APP CONTAINER

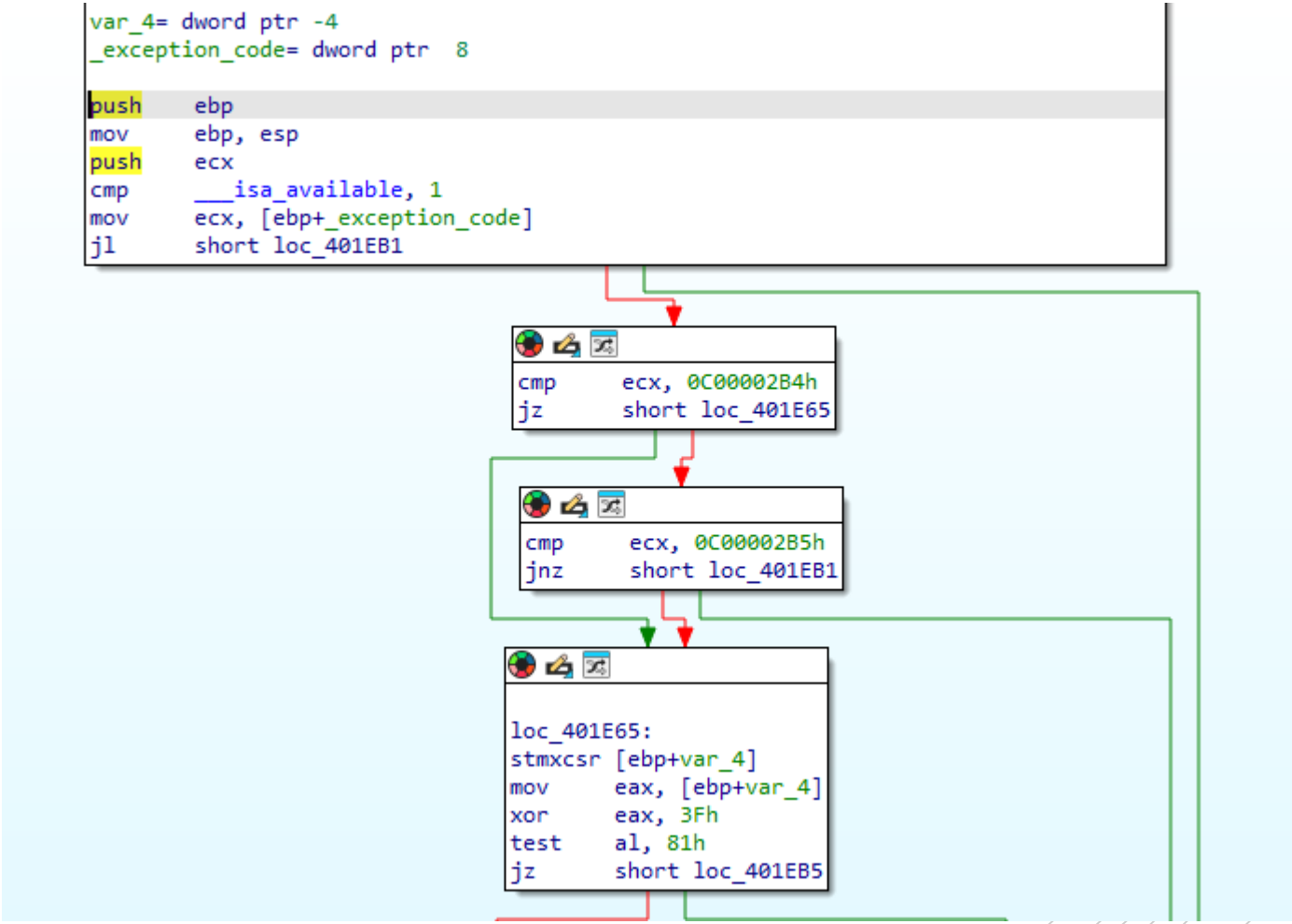
- یک محیط سندباکس در ویندوز برای اجرای امن برنامه ها
- بعد از ویندوز ۸ معرفی شد.
- برنامه هایی که معمولا داخل **AppContainer** اجرا میشن:
  - اپلیکیشن های **UWP (Universal Windows Platform)**
  - برخی برنامه‌های **Packaged Win32**
  - نسخه های قدیمی مرورگر **Microsoft Edge**
- در **Optional Header** فیلد **DLLCharacteristics**:
  - IMAGE_DLLCHARACTERISTICS_APPCONTAINER

Control Flow Graph (CFG)

- یک ساختار گرافیکی از مسیر اجرای برنامه

Github: onhexgroup

CFG ?
Control Flow Graph



Github: onhexgroup

CFG ? Control Flow Guard

- 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 معتبر بودن آدرس اجرا میشه. اگه آدرس معتبر باشه اجرا، در غیر اینصورت از اجرای کد جلوگیری میشه.
- در **Optional Header** فیلد **DLLCharacteristics**:
  - **IMAGE_DLLCHARACTERISTICS_GUARD_CF**
- در ویژوال استدیو:
  - **Configuration Properties → C/C++ → Code Generatio → Control Flow Guard :Yes (/guard:cf)**

Youtube: onhexgroup

Windows Driver Model (WDM)

- یک مدل قدیمی برای نوشتن درایور
- هدفش ایجاد یک استاندارد واحد بود تا درایورها بتونن هم روی ویندوزهای خانواده **9x** و هم خانواده **NT** کار کنن.
- در سطح کرنل اجرا میشن و دسترسی مستقیم به سخت‌افزار دارن.
- در **Optional Header** فیلد **DLLCharacteristics**:
  - **IMAGE_DLLCHARACTERISTICS_ WDM_DRIVER**
- در ویژوال استدیو:
  - **Configuration Properties → Linker→ System →Driver - WDM (/DRIVER:WDM)**

Youtube: onhexgroup

TERMINAL SERVER

- قابلیتی در ویندوز سرورکه به چندین کاربر اجازه میده بصورت همزمان و از راه دور به یک سرور متصل و هر کدوم دسکتاپ و برنامه‌های خودشون رو داشته باشن.
- برنامه ها برای اجرا در چنین محیطی باید این قابلیت رو داشته باشن.
- در **Optional Header** فیلد **DLLCharacteristics**:
  - IMAGE_DLLCHARACTERISTICS_ TERMINAL_SERVER_AWARE
- در ویژوال استدیو:
  - **Configuration Properties → Linker→ System →Terminal Server - Yes (/TSAWARE)**

Youtube: onhexgroup

ساختار Optional HEADER

- **SizeOfStackReserve** : کل فضای رزرو شده برای پشته برای Thread اصلی
- **SizeOfStackCommit** : چه مقدار از فضای رزرو شده به Thread اصلی در رم اختصاص یافته.
- در ویژوال استدیو:
  - **Configuration Properties → Linker→ System → Stack Reserve Size | Stack Commit Size**

Youtube: onhexgroup

ساختار Optional HEADER

- **SizeOfHeapReserve** : کل فضای رزرو شده برای هیپ پیش فرض پروسس
- **SizeOfHeapCommit** : چه مقدار از فضای رزرو شده در رم اختصاص داده شده.
- در ویژوال استدیو:
  - **Configuration Properties → Linker→ System → Heap Reserve Size| Heap Commit Size**

Youtube: onhexgroup

ساختار Optional HEADER

- **LoaderFlags** : یک مقدار رزرو شده که طبق مستندات مقدارش برابر صفره.
- **NumberOfRvaAndSizes**: تعداد ورودی های Data Directory رو مشخص میکنه. معمولا مقدارش برابر ۱۶

- **DataDirectory**: آرایه ای از IMAGE_DATA_DIRECTORY

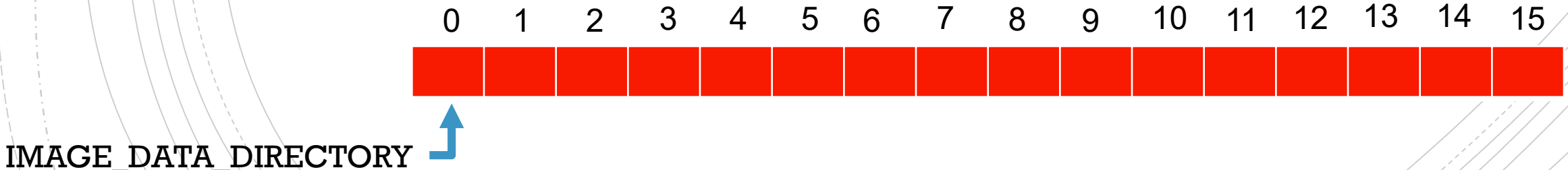
  که حاوی اطلاعاتی در خصوص بخش های مختلف فایل **PE**، مانند **Import** و **Export** و ... هستش.

Youtube: onhexgroup

ساختار Optional HEADER

```
#define IMAGE_NUMBEROF_DIRECTORY_ENTRIES    16

IMAGE_DATA_DIRECTORY DataDirectory[IMAGE_NUMBEROF_DIRECTORY_ENTRIES];
```



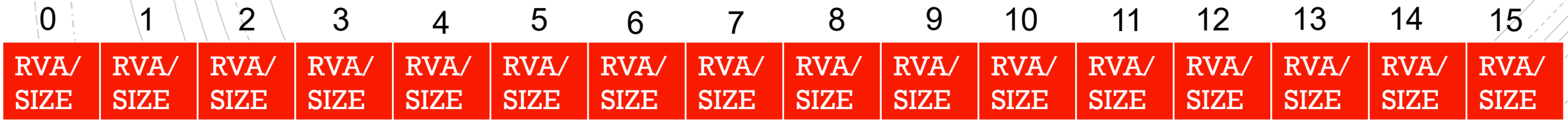
- ساختار IMAGE_DATA_DIRECTORY :

```
typedef struct _IMAGE_DATA_DIRECTORY {
    DWORD   VirtualAddress;
    DWORD   Size;
} IMAGE_DATA_DIRECTORY, *PIMAGE_DATA_DIRECTORY;
```

Youtube: onhexgroup

ساختار Optional HEADER

- **VirtualAddress**: آدرس ورودی
- **Size**: سایز اون ورودی



Youtube: onhexgroup

ساختار Optional HEADER

- هر ایندکس آرایه مشخص کننده یک ساختار هستش.

0	RVA/SIZE	Export Directory
1	RVA/SIZE	Import Directory
2	RVA/SIZE	Resource Directory
3	RVA/SIZE	Exception Directory
4	RVA/SIZE	Security Directory
5	RVA/SIZE	Base Relocation Table
6	RVA/SIZE	Debug Directory
7	RVA/SIZE	Architecture Specific Data
8	RVA/SIZE	RVA of GP
9	RVA/SIZE	TLS Directory
10	RVA/SIZE	Load Configuration Directory
11	RVA/SIZE	Bound Import Directory in headers
12	RVA/SIZE	Import Address Table
13	RVA/SIZE	Delay Load Import Descriptors
14	RVA/SIZE	COM Runtime descriptor
15	RVA/SIZE	Reserve