BEHIND ENEMY HOOKS

WHAT AV REALLY DOES TO YOUR APPS

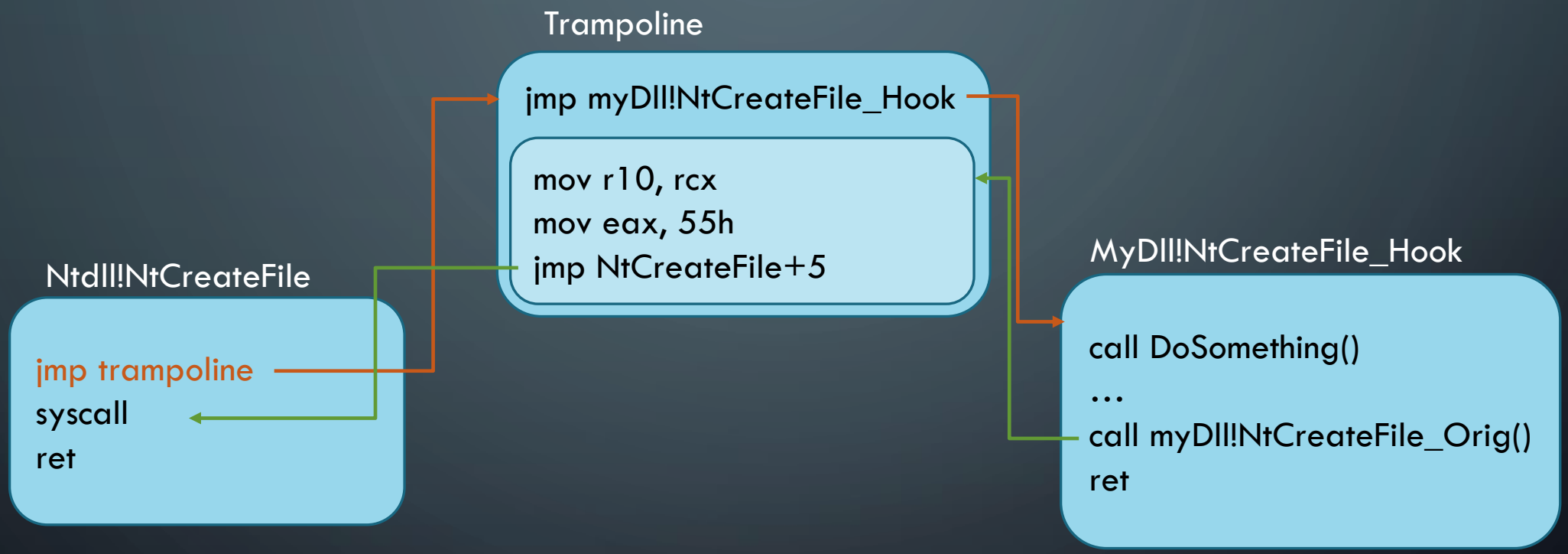
ABOUT ME

- Software Engineer at Crowdstrike
- Co-Instructor of Windows Internals security courses
- Aerialist, circusing most of the time
- Sometimes doing Windows Internals stuff
- Remotely-operated security researcher
- @yarden_shafir on twitter

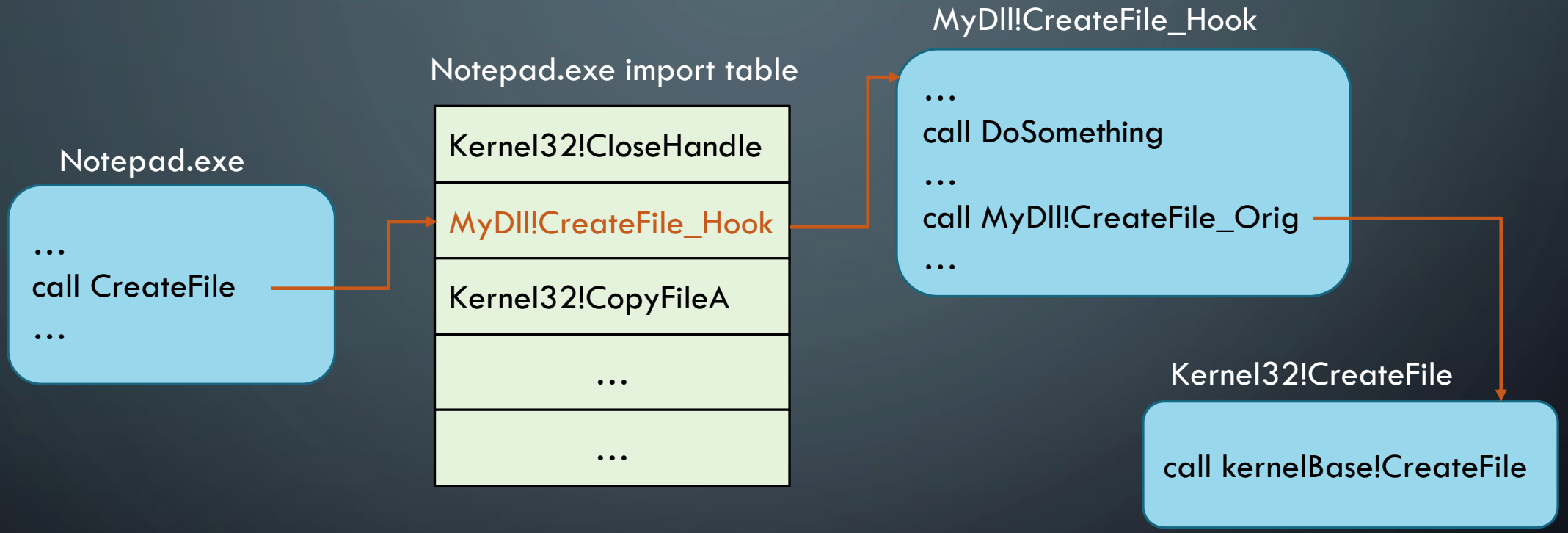
USER-MODE HOOKS

- Used by almost every AV
- Require injecting a DLL into every process in order to hook it
- Get every API call as it happens
- Can block or modify call
- In most cases, there is no other way to get the same information
- Lots of known, documented ways to do this

INLINE HOOKING



IAT HOOKING


Notepad.exe
…
call CreateFile
…
Notepad.exe import table
Kernel32!CloseHandle
MyDll!CreateFile_Hook
Kernel32!CopyFileA
…
…
MyDll!CreateFile_Hook
…
call DoSomething
…
call MyDll!CreateFile_Orig
…
Kernel32!CreateFile
call kernelBase!CreateFile

APPLICATION COMPATIBILITY

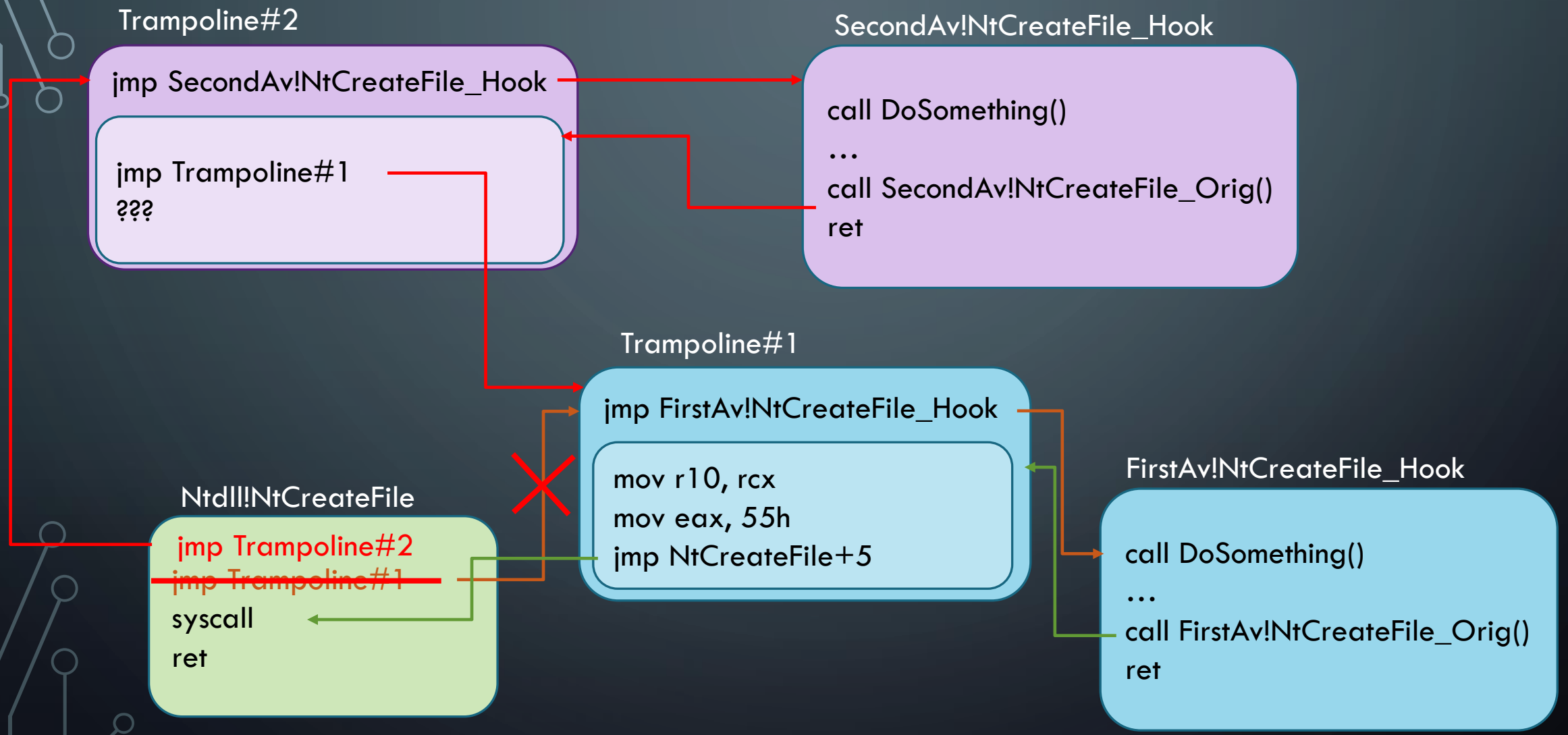
Hooks are very invasive and can harm an application's stability

Require injecting a DLL in ways that can get very hacky

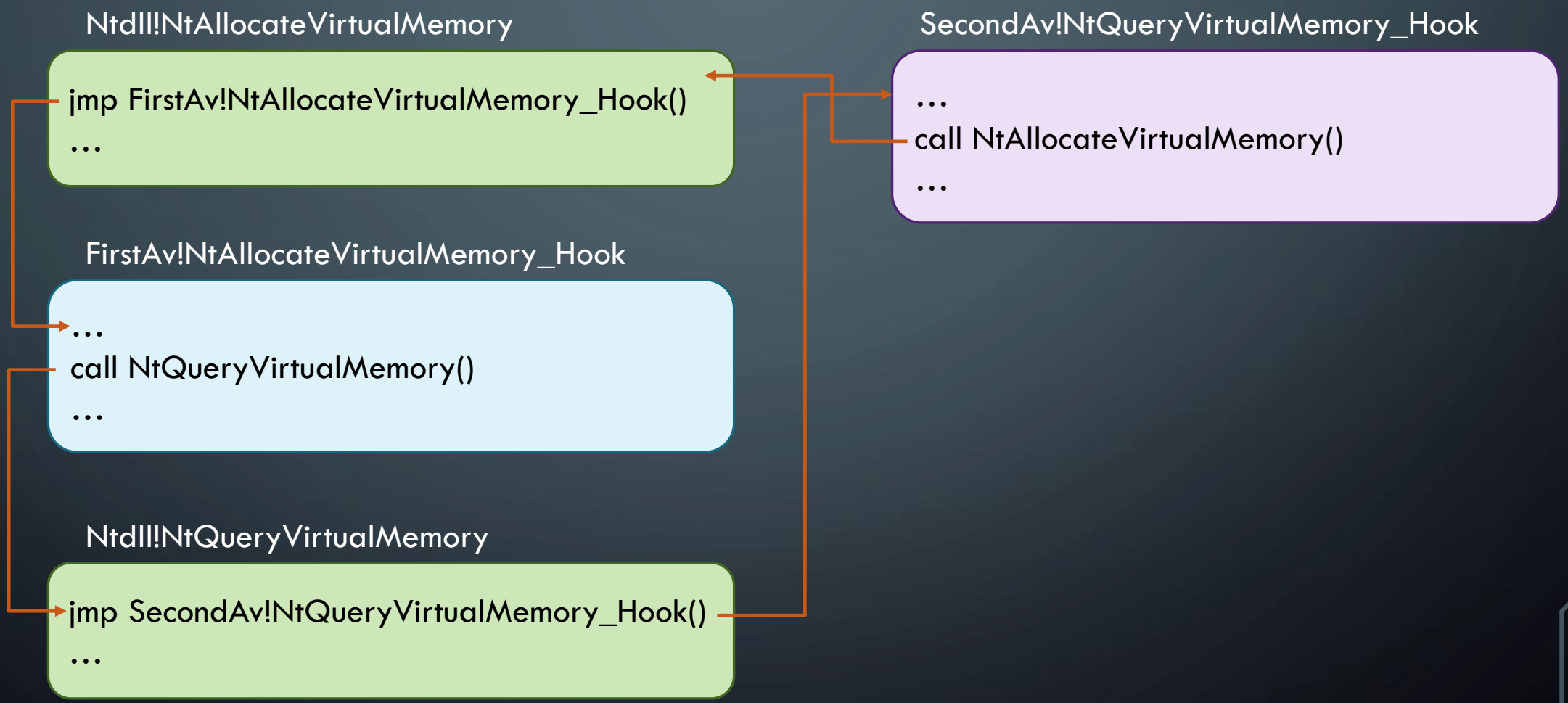
A lot of AVs don't consider new Windows mitigations that can affect their hooks or injection

When 2 AVs are trying to hook the same application things get really messy

INLINE HOOKING WITH 2 AVS



RECURSIVE HOOKING



ARBITRARY CODE GUARD

- With ACG, memory cannot be both writable and executable
  - Can't modify code
  - Data cannot become executable
- Enabled per-process, either on creation or later by the process itself
- "Kills" inline hooks and some code injection techniques
  - Inline hooks are based on patching existing functions to point to a hook function – can't do that with ACG
- Some AVs adapt slowly to mitigations and try to inject and hook anyway

DLLS ARE VULNERABLE

- A lot of AVs don't stay updated on new mitigations and don't enable them for their injected DLLs
- Most of AV DLLs don't have CFG enabled
- Some don't even use ASLR
- A single vulnerable DLL in the process makes the whole process vulnerable
- AV DLLs are usually injected into all processes

WHAT ELSE CAN WE DO?

- ETW!
- A mechanism that supplies information about events happening in the system
- Processes can register and get this information without hooks
- Not invasive – doesn't require anything to be injected into the process
- Less performance impact on running processes
- Not perfect either – has lots of issues of its own
  - More on that in some other talk