Helium, Argon, Krypton & Xenon: The Noble Gases of Windows Containers

Alex Ionescu,
Vice President of EDR Strategy
CrowdStrike, Inc.

SyScan 360 - May, 2017
About

• Vice President of EDR Strategy at CrowdStrike
• Reverse engineering NT since 2000
  • Lead kernel developer of ReactOS
  • Co-author of *Windows Internals*
  • Instructor of worldwide Windows internals classes
• Conference speaking:
  • SyScan 2012-2015, Infiltrate 2015
  • NoSuchCon 2014, 2013, Breakpoint 2012
  • Recon 2010-2017, 2006
• For more info, see [www.alex-ionescu.com](http://www.alex-ionescu.com)
• Twitter: @aionescu
Agenda

• Intro to Containers
• Container Types
• Contexts and Monitors
• Kernel Changes
• Boundary Violations (DEMO)
• Conclusion
Intro to Containers
What is a Container

• A low overhead wrapper around an application and/or set of services

• Allows easy deploy & migration

• Abstracts OS-level dependencies transparently

• Usually employs either virtualization as a boundary, or kernel-level isolation and redirection
Examples of Containers

• Control Groups (cgroups) / LXC
  ➢ Namespace boundaries & resource control
  ➢ Filesystem and network isolation and virtualization
  ➢ Used by Docker

• Job Objects
  ➢ Resource control
  ➢ Minimal namespace boundary (isolated atom table, clipboard, UI handle table)

• Silos
  ➢ Our big topic today
  ➢ “Jobs++”
  ➢ Leverages new kernel support and redirection drivers for namespace isolation
The Other “Container”

• Virtual Machines can be Containers too
  ➢ Typically heavyweight and slow, but can be made to rely on a trusted “root” OS
  ➢ True hardware-based isolation even against multi-tenant

• QubesOS uses this idea

• Bromium implements “micro-VMs” in this manner as well

• Windows Container technology supports this mode too
Container Types
It begins with a Silo...

- **A Silo** is an extension to a Job Object
  - **Application Silos** share Host State (“Global Data”)
  - **Server Silos** have their own Host State

- **A Silo** can be associated with virtual resources
  - Object Manager and Virtual Registry Namespace
  - File System (Managed by Wcifs.sys and Wcnfs.sys)
  - Network *Compartment* (Managed by Wnv.sys)

- Server Silos require TCB Privilege to Create & Manage
  - Created by VmCompute (Host Compute) Service

- Server Silos lead to the creation of a new Service Session
Helium

- Allows Win32 applications to be “ported” to Windows Store

- Creates a virtual file system and registry namespace on top of the existing Base OS on disk
  - File System and Virtual Registry Namespace provide lightweight mirroring and diffing
  - Not a hard boundary – real host FS and Registry still modifiable
  - C:\windows\containers\helium.def contains instructions for Application Silo creation

- Similar to UAC Virtualization, but scales to per-application needs

- Office on Windows 10 S will run like this
Example: Slack from Windows Store (File System)
Example: Windows Container Registry

lkd> !reg q \REGISTRY\WC\

Found KCB = ffffad0d1586e258 :: \REGISTRY\WC

Hive ffffad0cf9c31000
KeyNode ffffad0cf9c3a2cc

[SubKeyAddr] [SubKeyName]
fffad0cf9c3ae24 SILO_049d47df-1050-f852-6e54-687e0117e917_com
fffad0cf9c3aea4 SILO_7e68546e-1701-17e9-df47-9d04501052f8_com
fffad0d0e803114 SILO_7e68546e-1701-17e9-df47-9d04501052f8_software
fffad0cf9c3ad9c SILO_7e68546e-1701-17e9-df47-9d04501052f8_user_classes
fffad0d0e80319c SILO_7e68546e-1701-17e9-df47-9d04501052f8_user_sid
Call Stack: Launching Slack from Windows Store

00 ntdll!NtSetInformationJobObject
01 container!container::Job::JobNode::Execute+0x12a
02 container!container_runtime::CreateContainerObject+0x1a9
03 container!container::CreateContainer+0x113
04 daxexec!helium::Container::CreateArgonContainer+0x80
05 daxexec!helium::Container::Start+0x1b5
06 daxexec!helium::Container::Create+0xcc
07 daxexec!helium::AddProcess+0x5ba
08 daxexec!PostCreateProcessDesktopAppXActivation+0x28b
09 appinfo!RAiLaunchProcessWithIdentity+0xa3f

lkd> !process @$proc 0
PROCESS ffff9d025327f7c0
    SessionId: 0  Cid: 0ed4  Peb: 40f2a7000  ParentCid: 0330
    DirBase: 8098c000  ObjectTable: ffffad0d0e41c040  HandleCount: 233.
    Image: svchost.exe [AppInfo]
lkd> !silo

Address Type ProcessCount Identifier
ffff9d025a492270 Silo 5 {0124cdf6-458e-11e7-b1f0-548ca0c7f848} (0n1256)

1 active Silo(s)

lkd> !silo ffff9d025a492270
Silo ffff9d025a492270:
  Job : ffff9d025a492270
  Type : Silo
  Identifier : {0124cdf6-458e-11e7-b1f0-548ca0c7f848} (0n1256)
  Processes : 5

lkd> !job ffff9d025a492270 f
Job at ffff9d025a492270
Child jobs:
  Job at ffff9d0257d37060
    Child jobs:
      Job at ffff9d0257b4f9f0
        Processes assigned to this job:
        PROCESS ffff9d02578cb080
          SessionId: 1  Cid: 3730  Pid: 74b7235000  ParentCid: 1b60
          DirBase: 359267000  ObjectTable: ffffad0d18186380  HandleCount: 994.
          Image: Slack.exe
Argon

- Lightweight Container for **Server SKU only**, enables cheap Docker
  - **Host Compute Service** creates Server Silo for each Container
  - C:\windows\system32\container\wsc.def defines full namespace boundaries

- Complex set of changes to the kernel to allow isolation, redirection, and virtualization
  - Object Manager per-Silo Namespace
  - Network Compartments / Multitenancy

- Breaks a number of assumptions about the system
  - Multiple Console Sessions that are != 0
  - “Shutdown” isn’t really shutdown

- Requires a **Base OS Image** (WIM), with possible “layers” on top
Argon Architecture
Call Stack: Launching Cmd from Docker

02 ntdll!NtSetInformationJobObject+0x14
03 container!container_runtime::CreateContainerObject+0x46
04 container!container::CreateContainer+0x12c
05 vmcompute!ComputeService::ContainerUtilities::CreateWindowsContainer+0x27c
06 vmcompute!ComputeService::Management::WindowsContainerOrchestrator::Construct+0x412
07 vmcompute!ComputeService::Management::ComputeSystemManager::CreateComputeSystem+0x16b
08 vmcompute!HcsRpc_CreateSystem+0x202

0: kd> !process @$proc 0
PROCESS ffff9e07d24f6840
  SessionId: 0  Cid: 0bf8  Peb: cba625e000  ParentCid: 0248
  DirBase: 34d86000  ObjectTable: ffffb78e42523740  HandleCount: <Data Not Accessible>
  Image: vmcompute.exe
Example: Server Silo (Cmd.exe on Docker)

1: kd> !silo

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>ProcessCount</th>
<th>Identifier</th>
<th>(0n60)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ffff9e07d2af99f0</td>
<td>ServerSilo</td>
<td>17</td>
<td>{ad56ad7b-4607-11e7-8611-00155d52a9a0}</td>
<td>(0n60)</td>
</tr>
</tbody>
</table>

1 active Silo(s)

1: kd> !silo ffff9e07d2af99f0

Silo ffff9e07d2af99f0:

<table>
<thead>
<tr>
<th>Job</th>
<th>: ffff9e07d2af99f0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>: ServerSilo</td>
</tr>
<tr>
<td>Identifier</td>
<td>: {ad56ad7b-4607-11e7-8611-00155d52a9a0} (0n60)</td>
</tr>
<tr>
<td>Processes</td>
<td>: 17</td>
</tr>
</tbody>
</table>

Server silo globals ffff9e07d2c7e440:

<table>
<thead>
<tr>
<th>Default Error Port</th>
<th>: ffff9e07d2d5a080</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceSessionId</td>
<td>: 3</td>
</tr>
<tr>
<td>Root Directory</td>
<td>: 0000005fc92fe3a4 '\Silos\60'</td>
</tr>
<tr>
<td>State</td>
<td>: Running</td>
</tr>
</tbody>
</table>
Example: Network Compartment

PS C:\Users\Administrator> Get-NetCompartment
CompartmentId : 1
CompartmentDescription : Default Compartment
CompartmentGuid : {b1062982-2b18-4b4f-b3d5-a78d9b9cdd49}
CompartmentId : 2
CompartmentDescription : \Container_bafcd1fbc6f08357d63ca99cc28c614b8b44992a19a3c30358864fee9fca854b
CompartmentGuid : {ad56ad7b-4607-11e7-8611-00155d52a9a0}

PS C:\Users\Administrator> ipconfig /allcompartments
Ethernet adapter vEthernet (Container NIC ae6669b8):
    Description . . . . . . . . . . . . : Hyper-V Virtual Ethernet Adapter #2
    Physical Address . . . . . . . . . . : 00-15-5D-52-A5-70
    IPv4 Address . . . . . . . . . . . : 172.21.183.78(Preferred)
    Default Gateway . . . . . . . . . . : 172.21.176.1
    DNS Servers . . . . . . . . . . . . : 172.21.176.1
<table>
<thead>
<tr>
<th>Hash Address</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ffff9e07d2ce0500</td>
<td>Mutant</td>
<td>PendingRenameMutex</td>
</tr>
<tr>
<td>fffff8e425f6e70</td>
<td>SymbolicLink</td>
<td>ObjectType</td>
</tr>
<tr>
<td>ffff8e4378f080</td>
<td>SymbolicLink</td>
<td>SystemRoot</td>
</tr>
<tr>
<td>fffff8e462f85f0</td>
<td>Directory</td>
<td>Sessions</td>
</tr>
<tr>
<td>fffff8e43783620</td>
<td>Directory</td>
<td>NLS</td>
</tr>
<tr>
<td>fffff8e4378ba00</td>
<td>Directory</td>
<td>Windows</td>
</tr>
<tr>
<td>ffffb78e43794ea0</td>
<td>Directory</td>
<td>GLOBAL??</td>
</tr>
<tr>
<td>ffffb78e420f1d50</td>
<td>Directory</td>
<td>RPC Control</td>
</tr>
<tr>
<td>ffffb78e4378d5f0</td>
<td>SymbolicLink</td>
<td>PdcPort</td>
</tr>
<tr>
<td>ffffb78e4378bf0d</td>
<td>SymbolicLink</td>
<td>Dfs</td>
</tr>
<tr>
<td>ffffb78e4378d5f0</td>
<td>SymbolicLink</td>
<td>CExecSvcStarted</td>
</tr>
<tr>
<td>ffffb78e4378bf0d</td>
<td>SymbolicLink</td>
<td>CsrSbSyncEvent</td>
</tr>
<tr>
<td>ffffb78e43783d50</td>
<td>SymbolicLink</td>
<td>SeRmCommandPort</td>
</tr>
<tr>
<td>ffffb78e437a5780</td>
<td>SymbolicLink</td>
<td>DosDevices</td>
</tr>
<tr>
<td>ffffb78e434fab30</td>
<td>Directory</td>
<td>KnownDlls32</td>
</tr>
<tr>
<td>ffffb78e43783570</td>
<td>SymbolicLink</td>
<td>Registry</td>
</tr>
<tr>
<td>ffffb78e4368ba80</td>
<td>Directory</td>
<td>BaseNamedObjects</td>
</tr>
<tr>
<td>ffffb78e4386d310</td>
<td>Section</td>
<td>Win32kCrossSessionGlobals</td>
</tr>
<tr>
<td>ffffb78e4375c430</td>
<td>ALPC Port</td>
<td>SmSsWinStationApiPort</td>
</tr>
<tr>
<td>ffffb78e437a78a0</td>
<td>Directory</td>
<td>KnownDlls</td>
</tr>
<tr>
<td>ffffb78e43783e70</td>
<td>SymbolicLink</td>
<td>FileSystem</td>
</tr>
<tr>
<td>ffffb78e4316b080</td>
<td>SymbolicLink</td>
<td>Driver</td>
</tr>
<tr>
<td>ffffb78e4316b080</td>
<td>Event</td>
<td>SAM_SERVICE_STARTED</td>
</tr>
</tbody>
</table>
Demo

Server Silo Command Prompt
Krypton and Xenon

- These now add Hyper-V Virtualization to the mix

- Address the Hostile Multitenant scenario

- Krypton: Bromium/QubesOS: Run each application in a micro VM
  - Edge in Creators Fall Update (RS3)
  - Why stop there? 😊

- Xenon: Run entire OS instance in a lightweight VM
  - Used by Docker for Windows
  - Supported on Client SKU, unlike Argon
Container Technologies in Windows 10

Helium
- File System Virtualization (Wcsifs.sys, Wcnfs.sys)
- Registry Virtualization (Registry.sys + Native VReg)
- Runs on Host, Modifies Host
- “Application Silo”
  - Used by “Project Centennial”
  - Windows Bridge for Desktop
  - Increasing amount of apps

Argon
- Object Manager Isolation
- Network Virtualization (Wnv.sys)
- Isolated Service Session
- PID/TID Filtering
- Runs on Base OS Image
- “Server Silo”
  - Windows Server Container
  - Docker for Windows
  - Supported on Server SKU Only
  - Not a Security Boundary

Krypton
- Runs on Host, Cannot Modify
- Base Image Created On The Fly
- Thin Hyper-V Partition
- HVSI
  - Windows Defender Application Guard (WDAG)
  - “Project Barcelona”

Xenon
- Argon w/ Base OS Image
  - Thin Hyper-V Partition
- Windows Container
  - Docker For Windows
  - Supported on Client SKU
  - Security Boundary / No Bounty
Kernel Changes
Job Object Changes

• Jobs now have a Container GUID (even if not Silos)

• Jobs now have a Job ID (JID)
  ➢ This is in the PspUniqueJobIdTable – no way to open/enumerate, however
  ➢ Used when creating a Server Silo namespace

• New Silo flag indicates Job is an Application or Server Silo
  ➢ If Job has “ServerSiloGlobals”, then this is a Server Silo
New Silo Job Object Information Classes

JobObjectCreateSilo = 0n35
JobObjectSiloBasicInformation = 0n36
JobObjectSiloRootDirectory = 0n37
JobObjectServerSiloBasicInformation = 0n38
JobObjectServerSiloUserSharedData = 0n39
JobObjectServerSiloInitialize = 0n40
JobObjectServerSiloRunningState = 0n41
JobObjectContainerTelemetryId = 0n44
JobObjectSiloSystemRoot = 0n45
JobObjectThreadImpersonationInformation = 0n47
Applications in Server Silo see a different “OS”

- Filter Drivers redirect to Base OS WIM, Local Directory File System and Local Registry
- Kernel Isolates Previously Global Variables
- Drivers get Silo Enlightenments through Monitors
- Real Smss.exe creates a new non-0 System Session
Example: Host Server SiloGlobals

lkd> !silo -g Host
Server silo globals ffff80281608540:

State : Running

lkd> dx -r1 *((nt\_ESERVERSILO\_GLOBALS *)0xffff80281608540))
[+0x000] ObSiloState [Type: _OBP\_SILODRIVERSTATE]
[+0x2e0] SesioloState [Type: _SEP\_SILOSTATE]
[+0x310] SeRmSiloState [Type: _SEP\_RM\_LSA\_CONNECTION\_STATE]
[+0x360] EtwSiloState : 0xffff9d02454b4000 [Type: _ETW\_SILODRIVERSTATE *]
[+0x368] MiSessionLeaderProcess : 0xffff9d0250f77c0 [Type: _EPROCESS *]
[+0x370] ExpDefaultErrorPortProcess : 0xffff9d0251e627c0 [Type: _EPROCESS *]
[+0x378] ExpDefaultErrorPort : 0xffff9d0251eb5ad0 [Type: void *]
[+0x380] HardErrorState : 0x1 [Type: unsigned long]
[+0x388] WntSiloState [Type: _WNF\_SILODRIVERSTATE]
[+0x3c0] PsProtectedCurrentDirectory : "C:\WINDOWS" [Type: _UNICODE\_STRING]
[+0x3d0] PsProtectedEnvironment : "Path=C:\WINDOWS\System32" [Type: _UNICODE\_STRING]
[+0x3e0] ApiSetSection : 0x0 [Type: void *]
[+0x3e8] ApiSetSchema : 0x0 [Type: void *]
[+0x3f0] OneCoreForwardersEnabled : 0x0 [Type: unsigned char]
[+0x3f8] NtSystemRoot : "C:\WINDOWS" [Type: _UNICODE\_STRING]
[+0x408] SiloRootDirectoryName : "" [Type: _UNICODE\_STRING]
[+0x418] Storage : 0xffff9d02454cbc00 [Type: _PSP\_STORAGE *]
[+0x420] State : SERVERSILO\_STARTED (1) [Type: _SERVERSILO\_STATE]
[+0x424] ExitStatus : 259 [Type: long]
[+0x428] DeleteEvent : 0x0 [Type: _KEVENT *]
[+0x430] UserSharedData : 0xffff9d0245476db0 [Type: _SILO\_USER\_SHARED\_DATA *]
[+0x438] UserSharedSection : 0x0 [Type: void *]
[+0x440] TerminateWorkItem [Type: _WORK\_QUEUE\_ITEM]
Object and I/O Manager Changes

- Object Parsing can now use a Silo Context
  - OBJECT_TYPE_INITIALIZE has new field UseExtendedParameters
  - This enables usage of ParseRoutineEx which receives OB_EXTENDED_PARSE_PARAMETERS
  - This structure adds a Silo field

- This is used by File Object Parsing, and exposed to developers
  - IoCreateFileEx accepts IO_DRIVER_CREATE_CONTEXT which has a SiloContext field
  - Creates the file in the given silo – can also use IoGetSiloParameters to query

- Does the lookup in the correct \Silos namespace
Other System Changes

- **Smss.exe** now has a new ALPC API to “boot up” a Silo
  - Creates a copy of itself, assigning it to the Silo Job Object, which then creates the new service session, starts the Windows Subsystem (Csrss.exe), etc...

- **In Creators Update, PEB now has a SharedData field which contains the SILO_USER_SHARED_DATA**
  - Various Rtl APIs now exposed in Ntdll.dll which avoid hard-coding the SharedUserData (0x7FFE0000) address, and use this instead
  - Mapped by PspMapSiloSharedDataView (also in WoW64 PEB)

- **Conhost.exe** is not used – instead, CExecSvc.exe handles Console I/O as the host
  - Uses a combination of two named pipes to provide STDIN/STDOUT to dockerd.exe (Server), which then communicates with docker.exe (Client)
Contexts and Monitors
Silo Local Storage (SLS)

- Each piece of Silo-specific data consumes a Silo Slot, and receives a globally unique Slot Index
  - `PsAllocSiloContextSlot` allocates, `PsFreeSiloContextSlot` releases

- Each Silo has 32 static slot indices, plus up to 256 dynamic

- Each owner of silo-specific data stores it in their associated slot index inside of every silo

- Silo monitors receive their own slot index automatically
  - `PsGetSiloMonitorContextSlot` returns it
Silo Contexts

- One example of Silo-specific data is a Silo Context (paged / non)
  - PsCreateSiloContext allocates

- Once a Slot Index has been allocated, can be inserted into a Silo
  - PsInsertSiloContext inserts, PsRemoveSiloContext removes
  - Can also make Permanent contexts (PsMakeSiloContextPermanent)

- Silo Contexts are Object Manager owned
  - PsReferenceSiloContext, PsDereferenceSiloContext

- Can be retrieved from a silo through the Slot Index
  - PsGetSiloContext returns it
<table>
<thead>
<tr>
<th>Slot</th>
<th>Object</th>
<th>Flags</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>ffffd0d16c1ea40</td>
<td>[RO]</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>ffffd0d17a53d40</td>
<td>[RO]</td>
<td>VReg</td>
</tr>
<tr>
<td>15</td>
<td>ffffd0257fc3900</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>ffffd0252c4f680</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: Cmd.exe Server Silo Storage

Storage ffff9e07d2afc980:

<table>
<thead>
<tr>
<th>Slot</th>
<th>Object</th>
<th>Flags</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>ffff878e436c86c0</td>
<td>[RO] : OB Root</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ffff9e07d0b70480</td>
<td>[RO] : 'ci.dll'</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ffff878e4633ce40</td>
<td>[RO] : 'NTOS_CMP'</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ffff9e07d19027c0</td>
<td>[RO] : 'NSI'</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ffff878e462fde40</td>
<td>[RO] : MountMgr</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ffff9e07d2c0f9c0</td>
<td>[RO] : '\Driver\KSecDD'</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ffff878e462fca80</td>
<td>[RO] : '\FileSystem\Mup'</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>ffff9e07d2c0f5e0</td>
<td>[RO] : '\FileSystem\Npfs'</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ffff9e07d2c0f0e0</td>
<td>[RO] : '\FileSystem\Msfs'</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ffff9e07d1548740</td>
<td>[RO] : '\Driver\AFD'</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ffff9e07d19037c0</td>
<td>[RO] : '\FileSystem\rdbss'</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>ffff9e07d258d900</td>
<td>[RO] : '\FileSystem\Dfsc'</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>ffff878e43731c20</td>
<td>[RO] : VReg</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>ffff9e07d2c1e040</td>
<td>[RO] : '\Driver\HTTP'</td>
<td></td>
</tr>
</tbody>
</table>
Silo Attach

• If performing operation on behalf of a silo, must “attach” to it
  ➢ PsAttachSiloToCurrentThread sets ETARGET->Silo to requested value
  ➢ 0xffffffff (-3) is NtCurrentSilo – non-attached → use Process’ Root Job Silo
  ➢ PsGetCurrentSilo returns one or the other

• Creators Update adds user-mode attach/detach
  ➢ NtSetInformationJobObject(JobObjectThreadImpersonationInformation)
  ➢ NtSetInformationThread(ThreadAttachContainer)

• Affects Object Manager Parse Routine
  ➢ It calls PsGetPermanentSiloContext to get the root Object Directory Pointer
Example: Attached Threads in Server Silo

Impersonating threads (16 total):

<table>
<thead>
<tr>
<th>Thread</th>
<th>Process</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ffff9e07d25e76c0</td>
<td>ffff9e07d0a4e040</td>
<td>'System'</td>
</tr>
<tr>
<td>ffff9e07d2c90040</td>
<td>ffff9e07d0a4e040</td>
<td>'System'</td>
</tr>
<tr>
<td>ffff9e07d211e800</td>
<td>ffff9e07d0a4e040</td>
<td>'System'</td>
</tr>
</tbody>
</table>

1: kd> !thread ffff9e07d211e800
Win32 Start Address nt!EtwpLogger (0x0ffff8014d386968)

1: kd> dt nt!_ETHREAD ffff9e07d211e800 Silo
   +0x7c0 Silo : 0xfffff9e07`d2af99f0 _EJOB

1: kd> !silo 0xfffff9e07`d2af99f0
Silo ffff9e07d2af99f0:
   Job : ffff9e07d2af99f0
   Type : ServerSilo
   Identifier : {ad56ad7b-4607-11e7-8611-00155d52a9a0} (0n60)
Silo Monitors

- Drivers may need server silo awareness to manage global data
- Named pipes & mailslots must be isolated
- Mapped Network Drives (DFS) must be isolated
- Some kernel drivers communicate up to user-space services
  - WSK DNS API (Afd.sys) must communicate to right DNS Client Service
  - Kernel Crypto (Ksecdd.sys) must communicate to right LSA Subsystem Service
  - Network Store Interface (Nsi.sys) must communicate to right NSI Service
**Example: Creator’s Update Client Silo Monitors**

```
1kd> !silo -m

<table>
<thead>
<tr>
<th>Address</th>
<th>Slot</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ffffad0cf9c1d110</td>
<td>3</td>
<td>ci.dll</td>
</tr>
<tr>
<td>ffffad0cfa36c770</td>
<td>5</td>
<td>NSI</td>
</tr>
<tr>
<td>ffffad0cf9c42850</td>
<td>7</td>
<td>\Driver\KSecDD</td>
</tr>
<tr>
<td>ffffad0cfadfe8c0</td>
<td>8</td>
<td>\FileSystem\Mup</td>
</tr>
<tr>
<td>ffffad0cfb1f27f0</td>
<td>9</td>
<td>\FileSystem\Npfs</td>
</tr>
<tr>
<td>ffffad0cfafa9a30</td>
<td>10</td>
<td>\FileSystem\Msfs</td>
</tr>
<tr>
<td>ffffad0cfae316f0</td>
<td>11</td>
<td>\Driver\AFD</td>
</tr>
<tr>
<td>ffffad0cfb37bec0</td>
<td>12</td>
<td>\FileSystem\rdbss</td>
</tr>
<tr>
<td>ffffad0cfad67cc0</td>
<td>13</td>
<td>\FileSystem\Dfsc</td>
</tr>
<tr>
<td>ffffad0cfc7f34d0</td>
<td>16</td>
<td>\Driver\HTTP</td>
</tr>
</tbody>
</table>

10 monitor(s) found.
Monitor API

- **PsRegisterSiloMonitor** registers a driver or internal component
  - Creation and Termination callbacks are provided
  - Notifications can be received for any existing Server Silo

- **PsStartSiloMonitor** begins monitoring

- **PsUnregisterSiloMonitor** to unregister

- Callback receives pointer to Silo (ESILO) which is really just EJOB

- Monitor can now associate Silo Context with this new Silo
Boundary Violations
PID / TID Filtering

• Unlike Linux Namespaces, Server Silos do not provide for isolated PID/TID (CID) tables

• C-level checks are used to filter out PID/TID information in
  ➢ ExpGetProcessInformation (used by NtQuerySystemInformation / ToolHelp API)
  ➢ PsLookupProcessByProcessId (used by NtOpenProcess)
  ➢ PsLookupThreadByThreadId (used by NtOpenThread)
  ➢ RULE: Given PID/TID must belong to the same Silo as requester

• Checks not present in **PsGetNextProcess/PsGetNextThread**
  ➢ Exposed to user-mode through NtGetNextProcess/NtGetNextThread
  ➢ Allows obtaining PROCESS_ALL_ACCESS handles to any PID on system, if Silo is running as
Other Bug Classes

- Obviously, bugs in kernel (like the Win32kSessionGlobals 0-day) allow for Server Silo Escapes

- 3rd party drivers that may be storing global state and talking with/to User Mode leading to confusion

- Some ALPC ports are globally visible to all Silos

- Only TCB can convert Job to Server Silo, but Admin can modify definition files in c:\windows\system32\containers
Demo

Server Silo Container Escape
Conclusion
Parting Thoughts

• Don’t run anything in a Windows Server Container (Server Silo) that you wouldn’t normally trust running as Administrator

• Multiple layers of virtualization/redirection/symbolic linking may result in Server Silo being used to escape/confuse products

• Control of Trusted Computing Base (TCB) boot chain (Smss- >Csrss- >Wininit, Lsass) is an interesting aspect of Server Silos

• Security Vendors should probably update their drivers to be Server Silo aware (using Monitors)
Further References

• More information is available in “Windows Internals, 7th Edition”

• See MSDN Documentation on Windows Containers
  https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/

• Also see Docker Source (now “Moby”) on GitHub
  https://github.com/moby/moby/
Thank You!

@aionescu