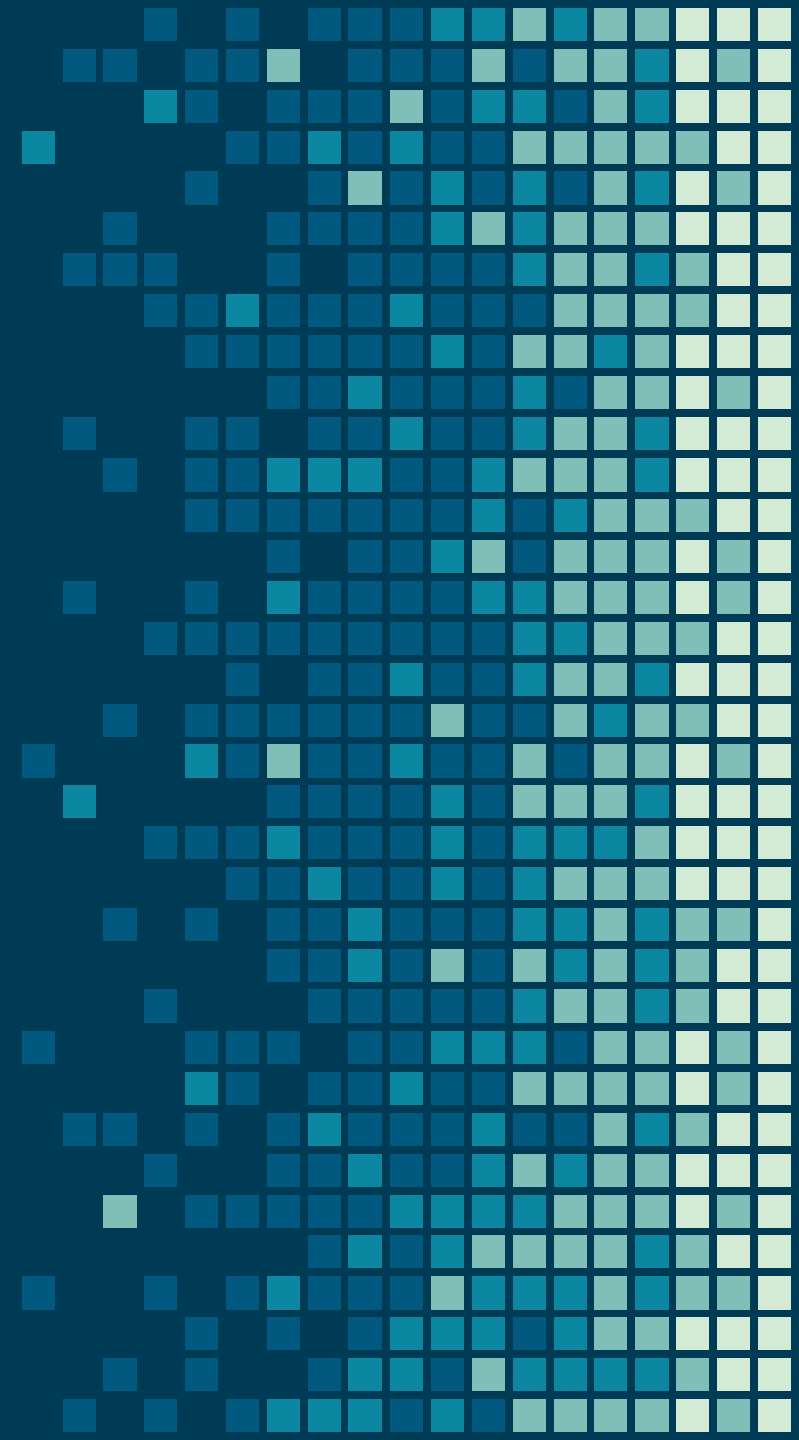


Security Features You've Never Heard of (but should)

Yarden Shafir

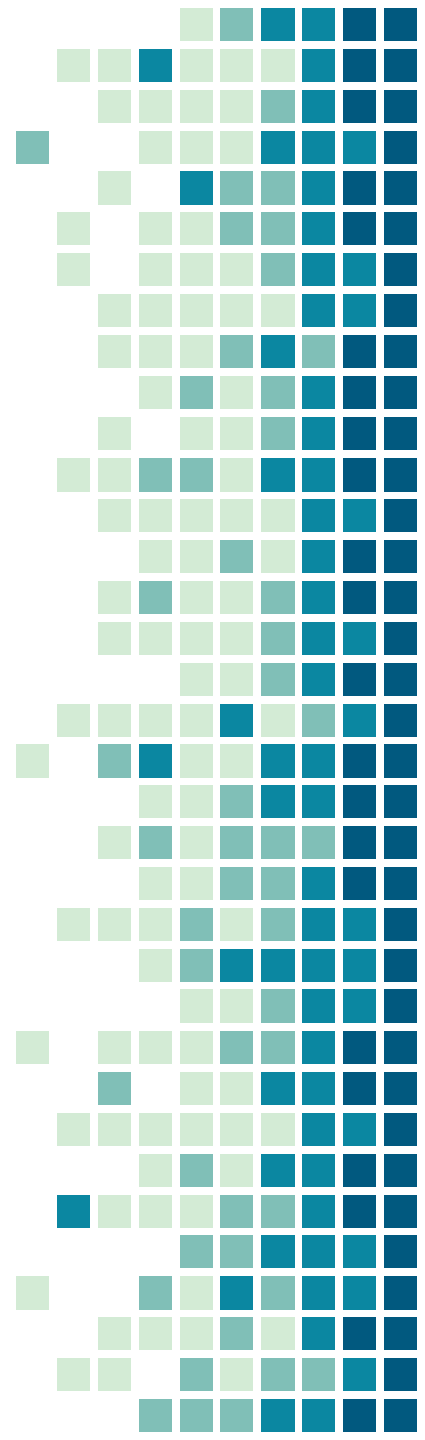


About Me

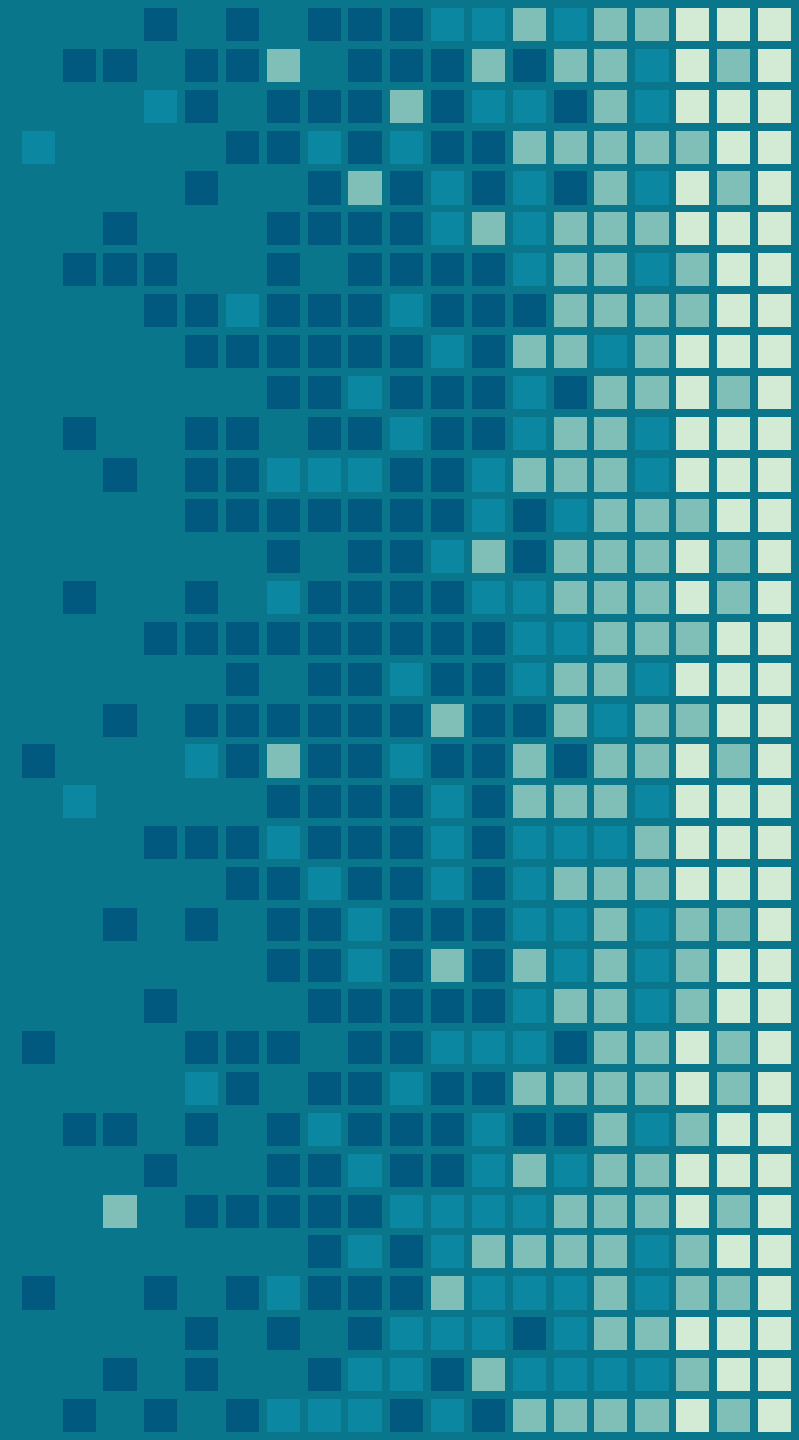
- Senior Software Engineer @CrowdStrike
- Previously Security Researcher @SentinelOne
- Windows Internals instructor
- Unemployed Circus Artist
- Retired Pastry Chef
- Blogging about Windows and Security: windows-internals.com
- @yarden_shafir

Another Talk About Mitigations?

- Yes!
- Some mitigations are well known, others are not
 - Even well-known ones aren't used by every application
 - Other than DEP, ASLR, CFG, which ones can you think of?
- Knowing about modern mitigations can help developers, security products, forensics...
- There are different ways to enable mitigations
 - And they aren't always documented well (or at all)

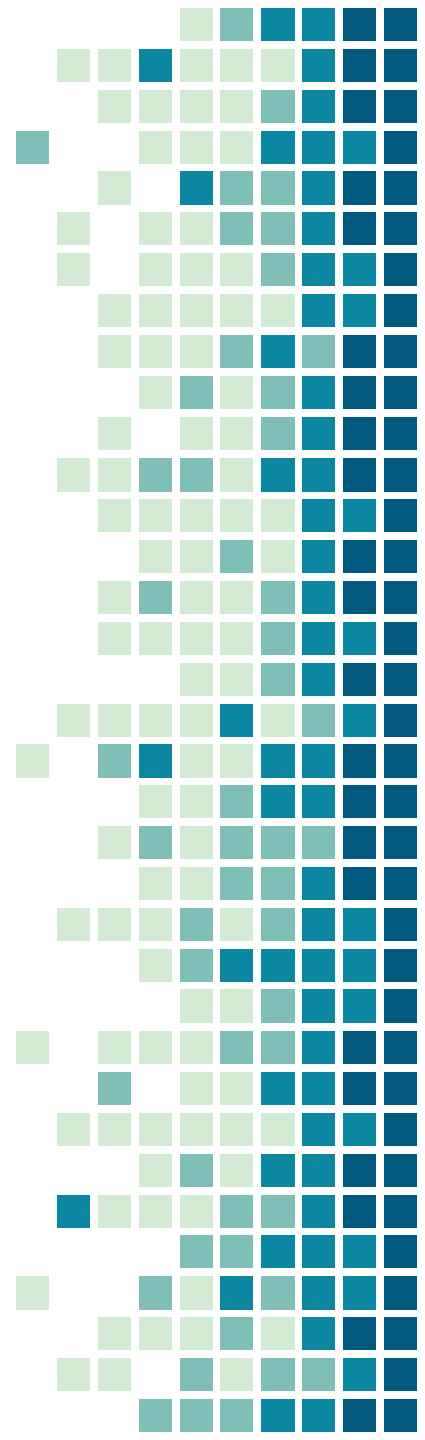


Module Tampering Protection

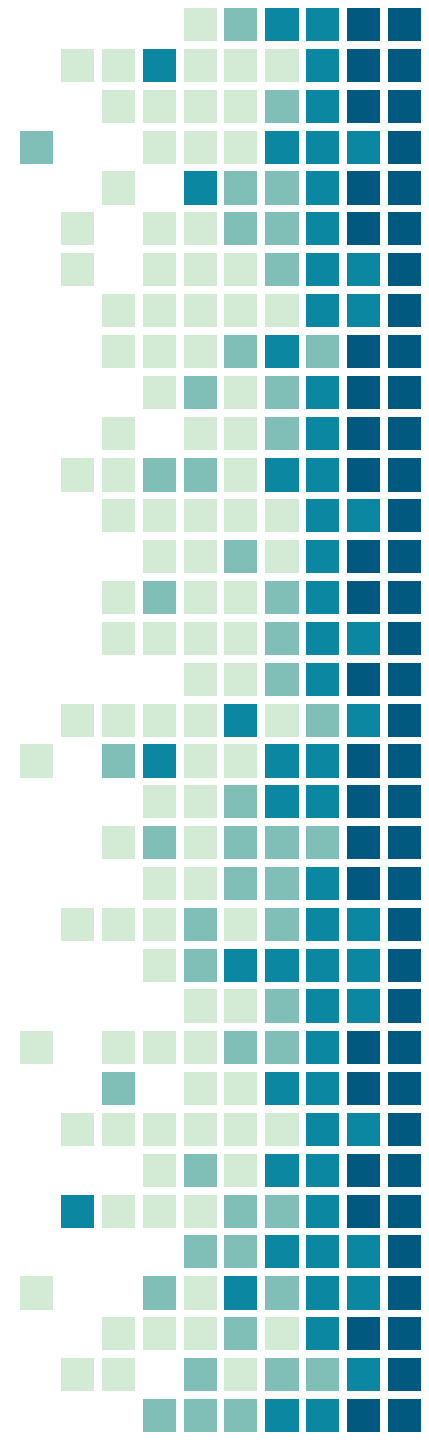
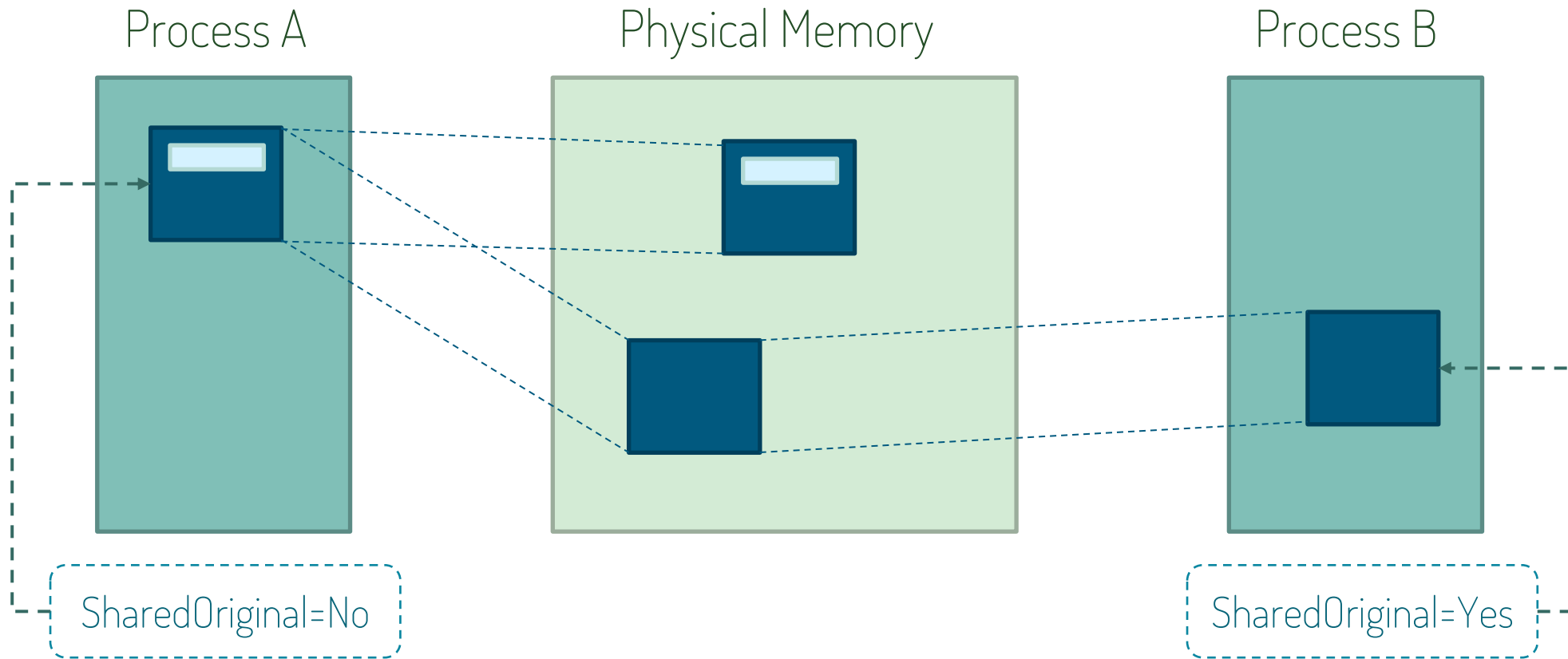


Module Tampering Protection

- Anti-Hollowing mitigation
- Detects when image headers or import table were modified and re-maps the original image
- Mitigation is set on process creation
 - Use (undocumented) mitigation policy bit
`PROCESS_CREATION_MITIGATION_POLICY2_MODULE_TAMPERING_PROTECTION_MASK` (defined in `WinBase.h`)
- Implemented in the loader (`Ntdll.dll`)
- Currently no processes enable this mitigation

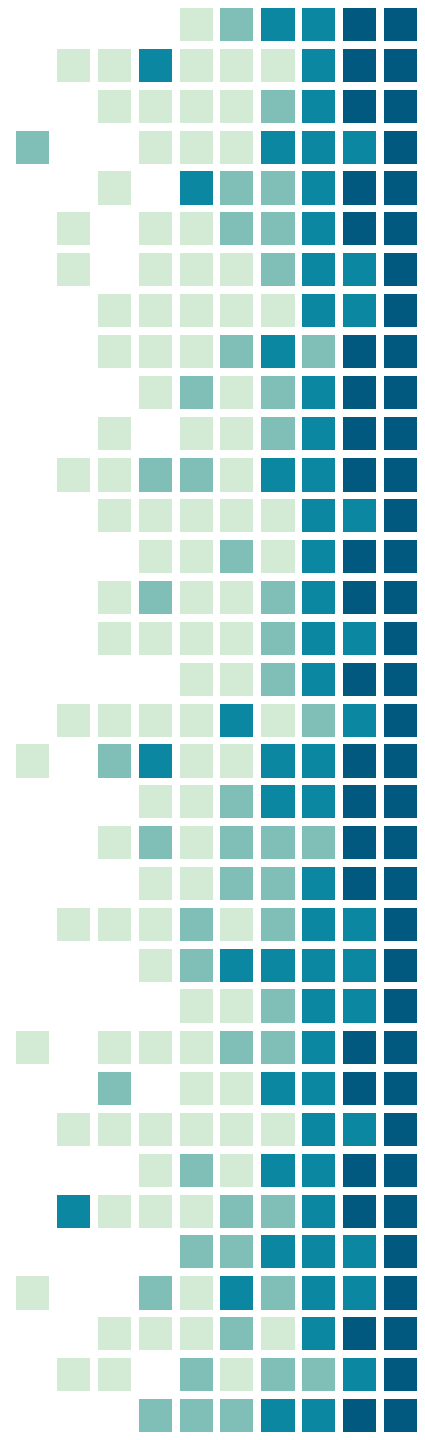


How Does It Work: Shared Sections



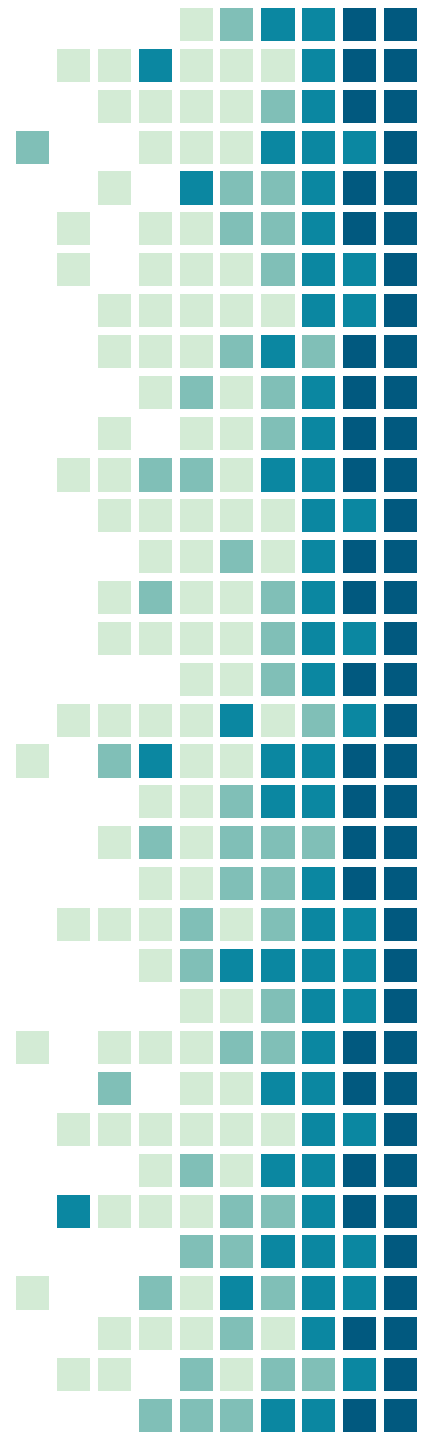
How Does It Work?

- LdrpCheckPagesForTampering checks page status
 - Calls NtQueryVirtualMemory with class MemoryWorkingSetExInformation
 - Returns information about the physical page:
 - Protection
 - ShareCount
 - SharedOriginal
 - If SharedOriginal is not set – page has been modified

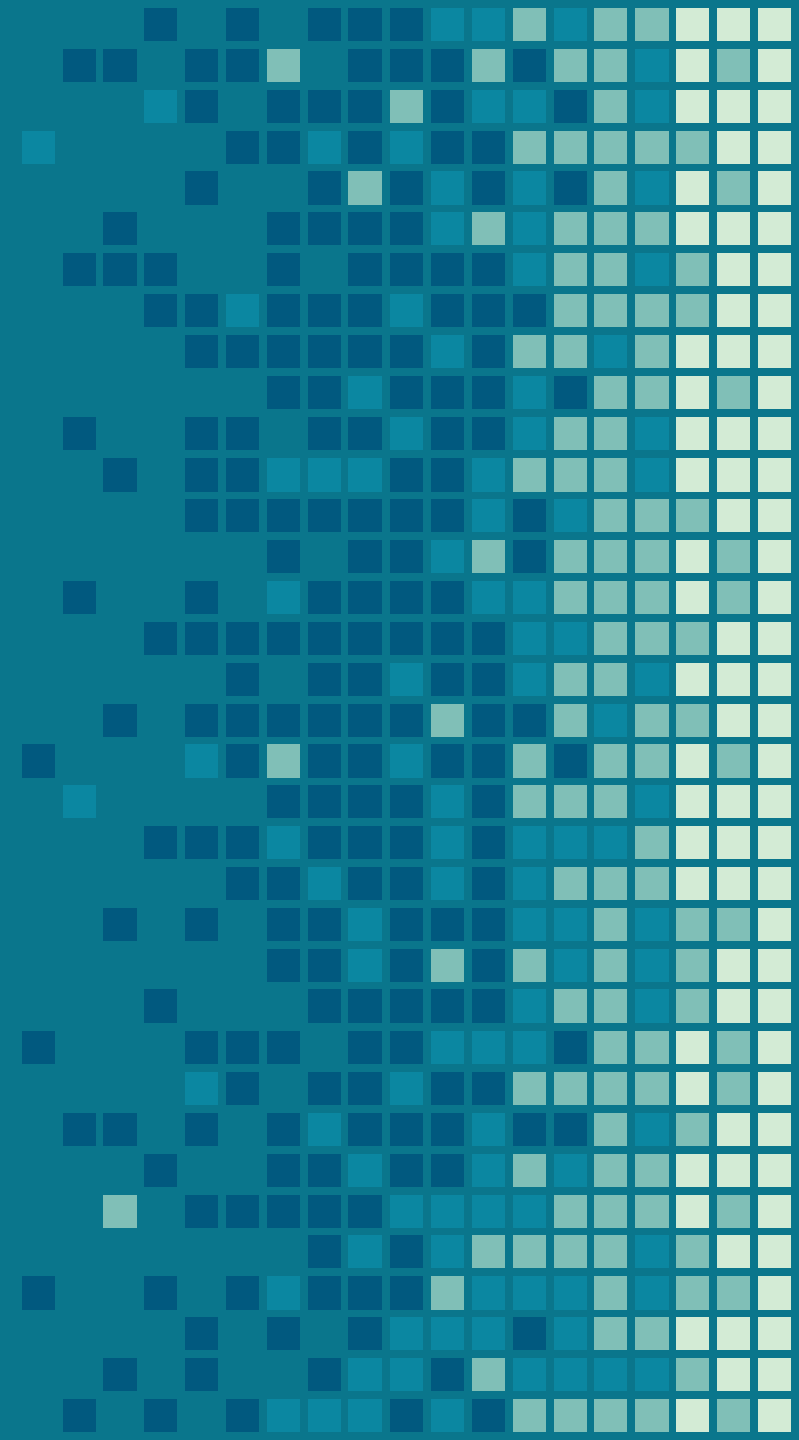


Remapping Image Section

- If image headers or IAT were modified, fresh copy of the image gets mapped
- LdrpMapCleanModuleView calls NtQueryInformationProcess
 - Uses new class ProcessImageSection that returns main image section handle
 - Calls NtMapViewOfSection to map clean copy



Process Redirection Trust



Follow the Link: Exploiting Symbolic Links with Ease

Tuesday, August 25, 2015

Windows 10^H^H Symbolic Link Mitigations

VDB-162605 · CVE-2020-16877

MICROSOFT WINDOWS UP TO SERVER 2019 REPARSE POINT ACCESS CONTROL

forshaw, abusing symbolic links like it's 1999.

#945122

Arbitrary file creation via symlink attack on syncagentsrv (Acronis Sync Agent Service)



Windows Sandboxed Mount Reparse Point Creation Mitigation Bypass

Authored by [Google Security Research](#), forshaw

Posted Oct 15, 2015

Privilege/Dangerous Behavior

Bypass/Elevation of

Sandboxed Mount Reparse Point Creation Mitigation Bypass Redux (MS16-008) (1)

Windows NTFS Global Reparse Point Elevation of Privilege Vulnerability

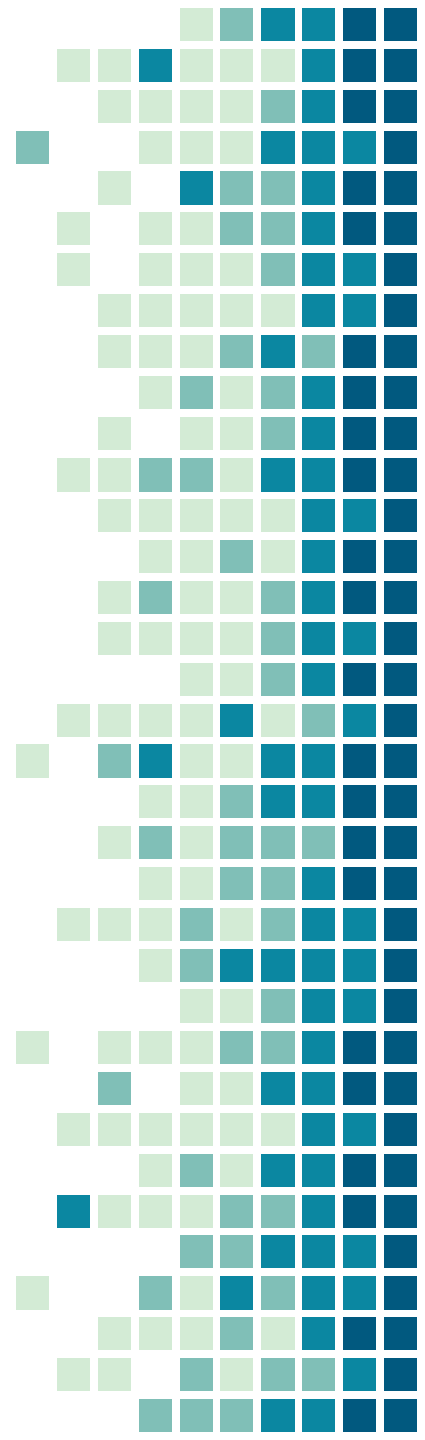
SET_REPARSE_POINT_EX Mount Point Security Feature Bypass

Feature Bypass/Elevation

of Privilege

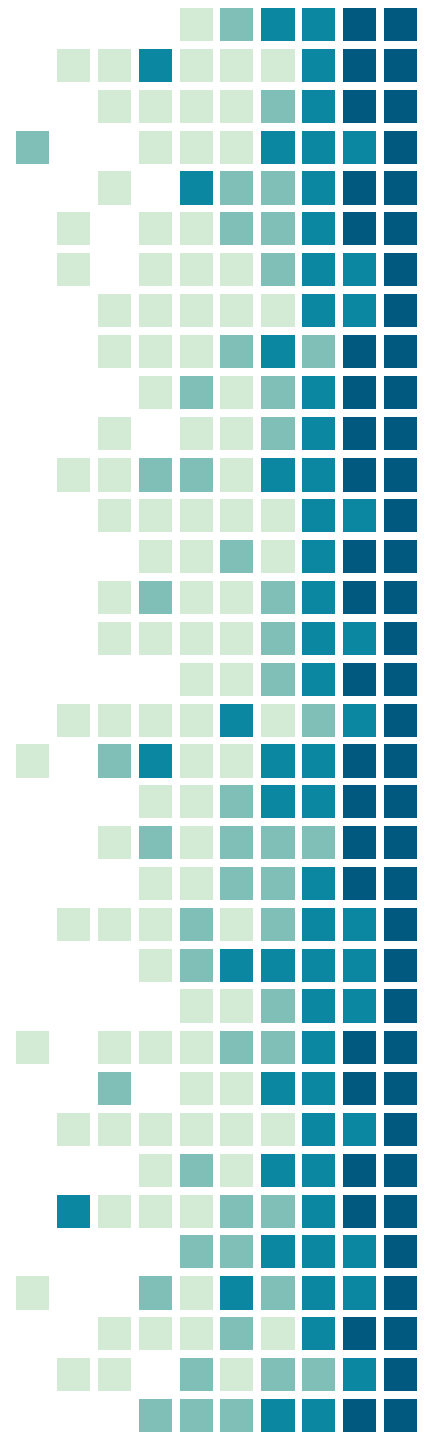
Path Redirection Bugs

- There are a lot of them
- Allow a low integrity process to “confuse” a high integrity process to use an incorrect path
 - Can lead to arbitrary file creation in privileged location
 - Or perform actions based on data read from user-controlled files and directories
 - Many eventually leading to privilege escalation



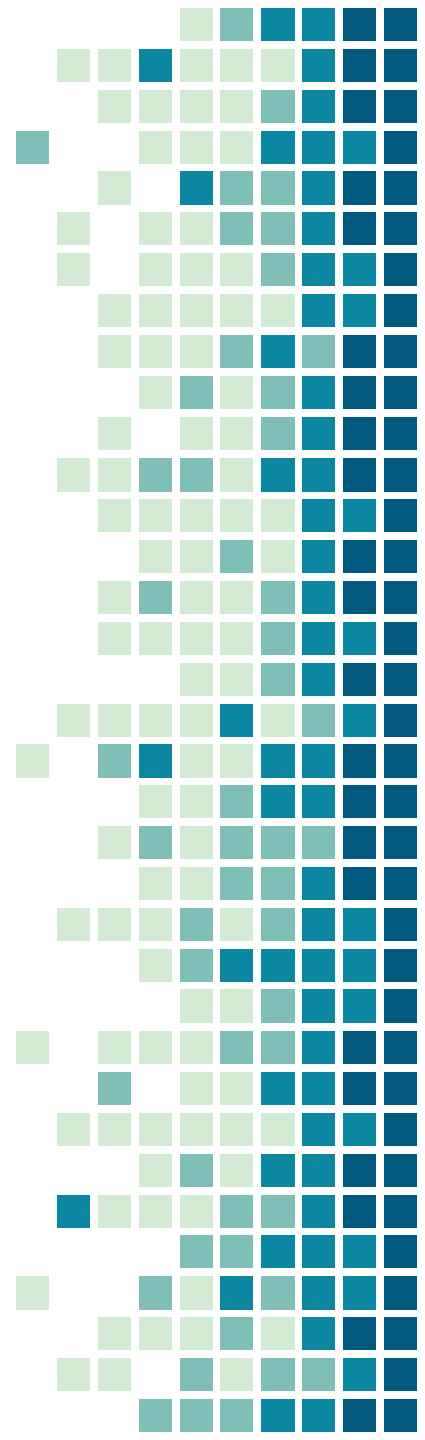
Redirection Trust Mitigation Policy

- Mitigation is set on a token
 - Which means it gets inherited by the process' children
 - And applies on any process using token for impersonation
- Can be set in "Enabled" or "Audit" mode
- Set during process creation or in runtime using `SetProcessMitigationPolicy`
 - `ProcessRedirectionTrustPolicy`



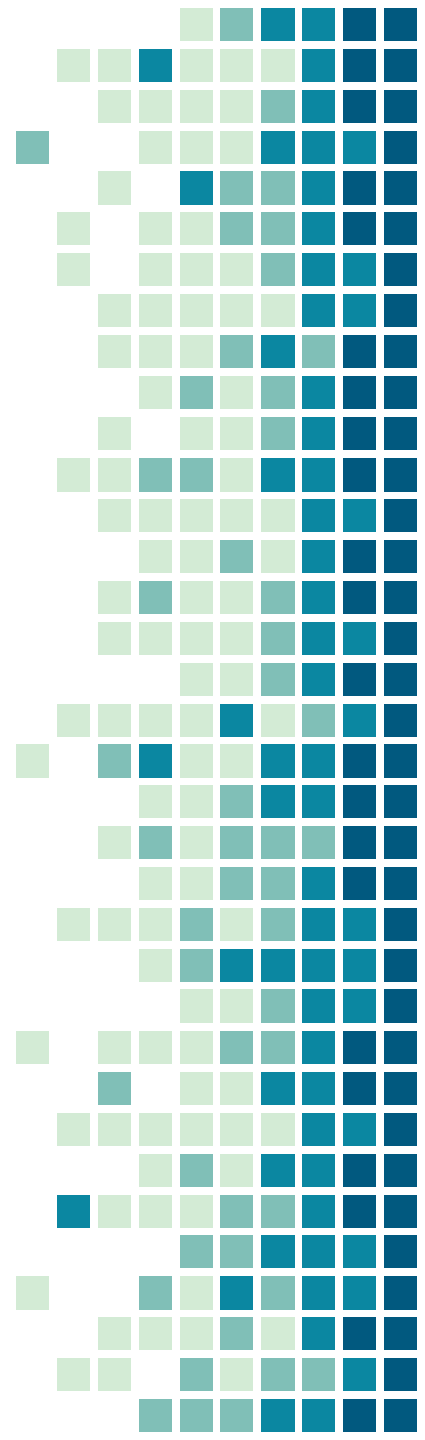
What Does It Do?

- Every reparse point created or modified by a non-Admin process gets “tagged”
- Processes running with a token that enables redirection trust mitigation will ignore reparse points created by non-Admin processes
- Kernel exports functions to compute and check redirection trust level
 - Called by NTFS.sys when getting and setting reparse points



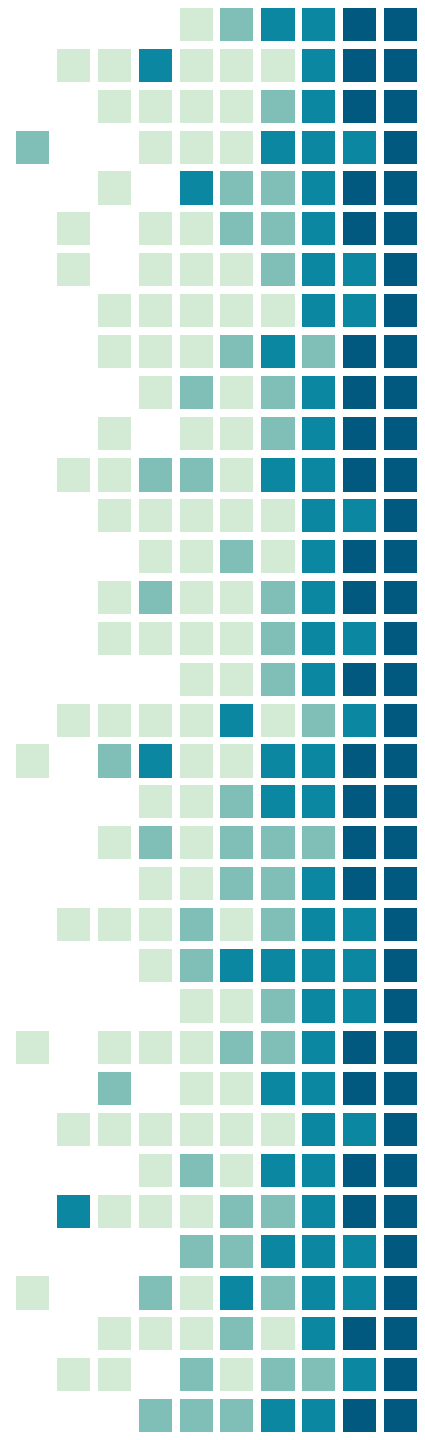
Process Redirection Trust Audit Mode

- In audit mode access to the non-admin reparse point won't be blocked
- Instead, an ETW event will be thrown to the ETW Threat Intelligence channel
 - Accessible to security products running as PPLs
- Most processes that enable this mitigation use audit mode only

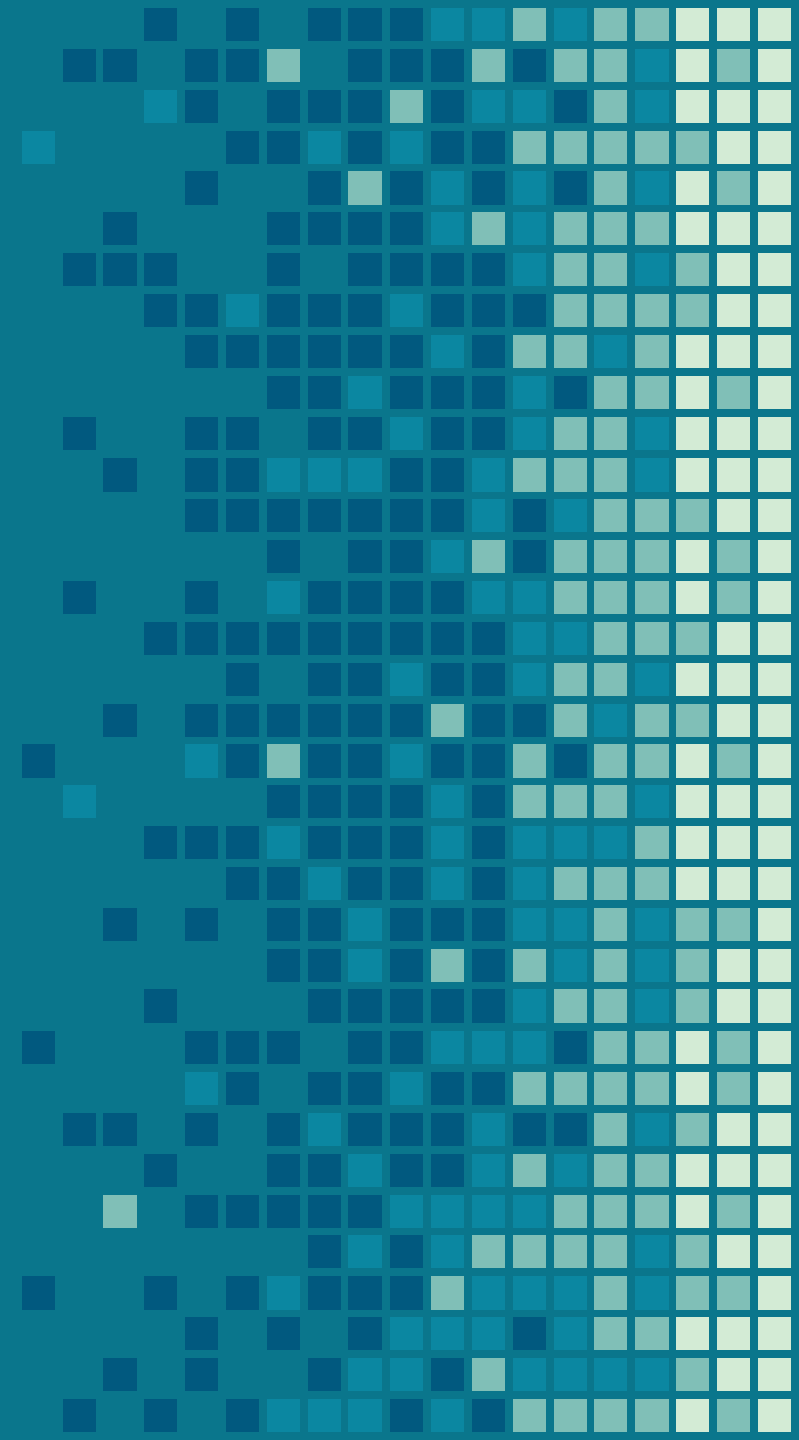


Who Uses This Mitigation?

	Name	RedirectionTrust	RedirectionTrustAudit
[0x11f4]	AggregatorHost.exe	false	true
[0x2524]	TabTip.exe	false	true
[0x1194]	conhost.exe	false	true
[0x1110]	spoolsv.exe	false	true
[0x1528]	svchost.exe	false	true
[0x1a50]	svchost.exe	false	true
[0x1ad4]	svchost.exe	false	true
[0x19a4]	svchost.exe	false	true
[0x6b8]	svchost.exe	false	true
[0x294c]	svchost.exe	false	true
[0x12e8]	svchost.exe	false	true
[0x83c]	svchost.exe	false	true
[0x520]	svchost.exe	false	true
[0x1834]	svchost.exe	false	true
[0x1680]	svchost.exe	false	true
[0x5b0]	svchost.exe	false	true
[0x5dc]	svchost.exe	false	true
[0x1958]	svchost.exe	false	true
[0x5a8]	svchost.exe	false	true
[0x65c]	svchost.exe	false	true
[0x1044]	svchost.exe	false	true
[0x6c0]	svchost.exe	false	true
[0x1760]	svchost.exe	false	true
[0x76c]	svchost.exe	false	true

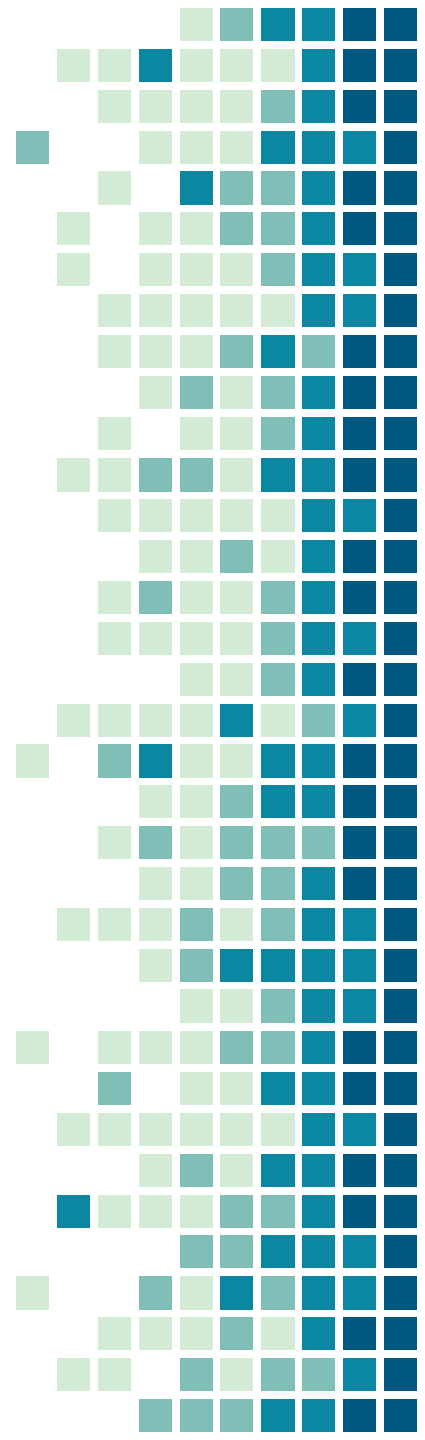


File Handle Revocation



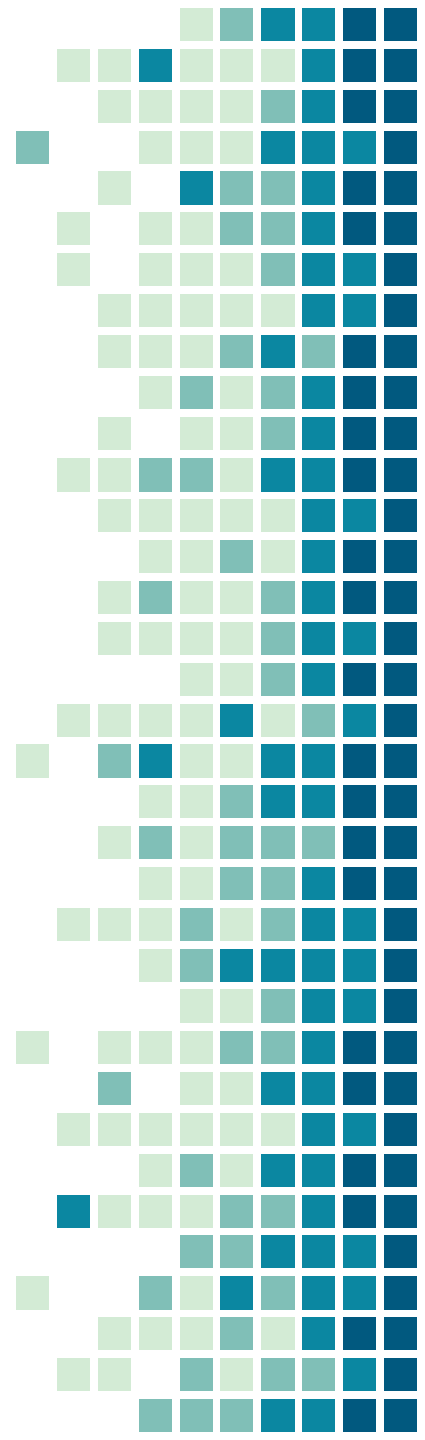
What are AppContainers?

- Microsoft container technology
- Restricts an application to very limited access to file system, registry, object manager...
- AppContainers can't access even most of the things that low integrity processes can access
- Requires complete rewrite of applications and was mostly abandoned in Windows 10
 - Used by a few Windows applications like fintDrvHost.exe, SearchHost.exe



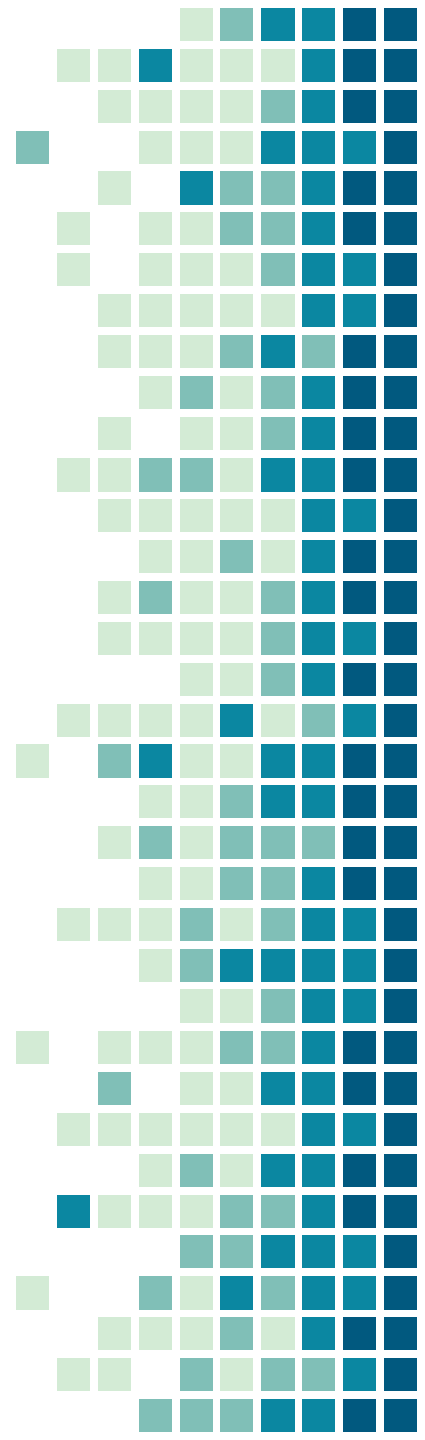
Handles and Access in Windows

- To access an object, a process has to open a handle
 - Handle specifies the object and the access granted
- Access rights to the object are only checked when handle is created, not when it is used
- Once a handle to an object is opened, process has access to that object
 - Even if security descriptor of the object changed



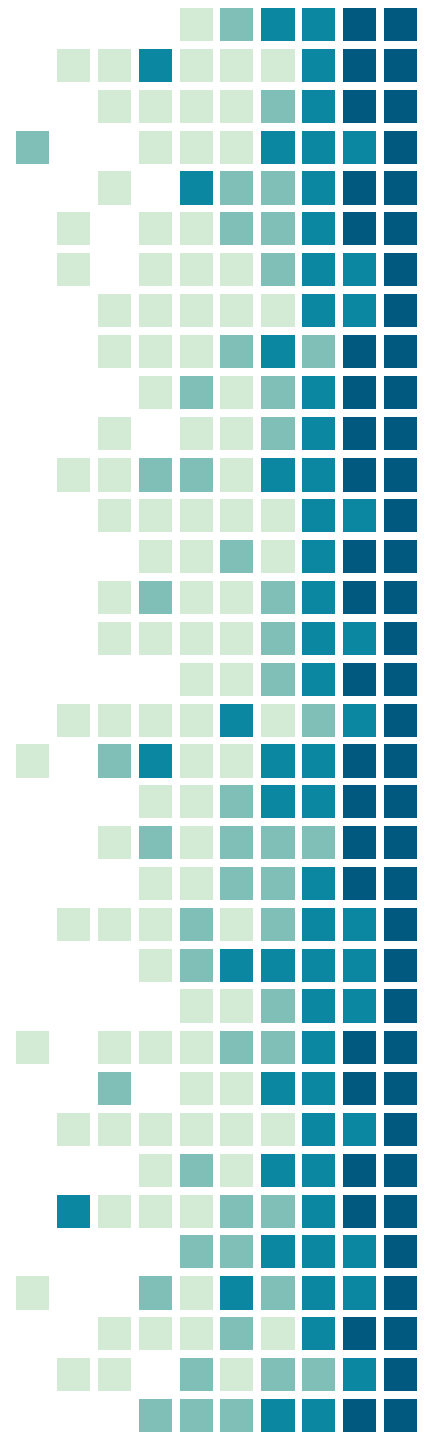
File Handle Revocation

- Allows revoking an AppContainer's access to a file after handle has been created
 - Can only be done for AppContainers
- Can be done with NtSetInformationProcess with class ProcessRevokeFileHandles
 - Supply full path of file to be revoked
- FILE_OBJECT will be marked as revoked
- Any attempt to use the handle will receive STATUS_FILE_HANDLE_REVOKED

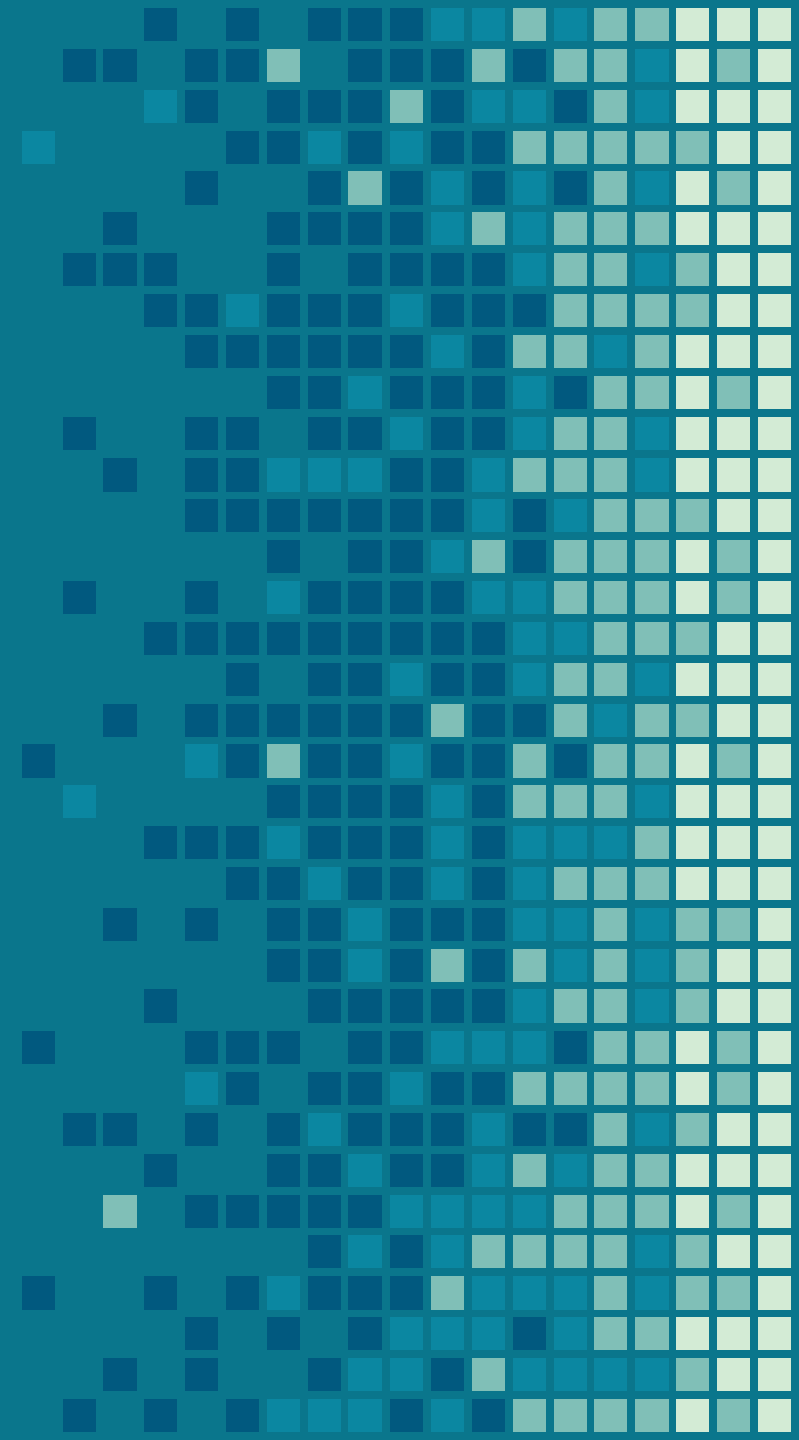


Who Uses This Mitigation?

- No one
- There isn't even a Win32 wrapper for this mitigation: only NT function is available
- Only available for AppContainers, and there aren't a lot of those anyway
- But since code is there, maybe will be available in the future in a wider context?

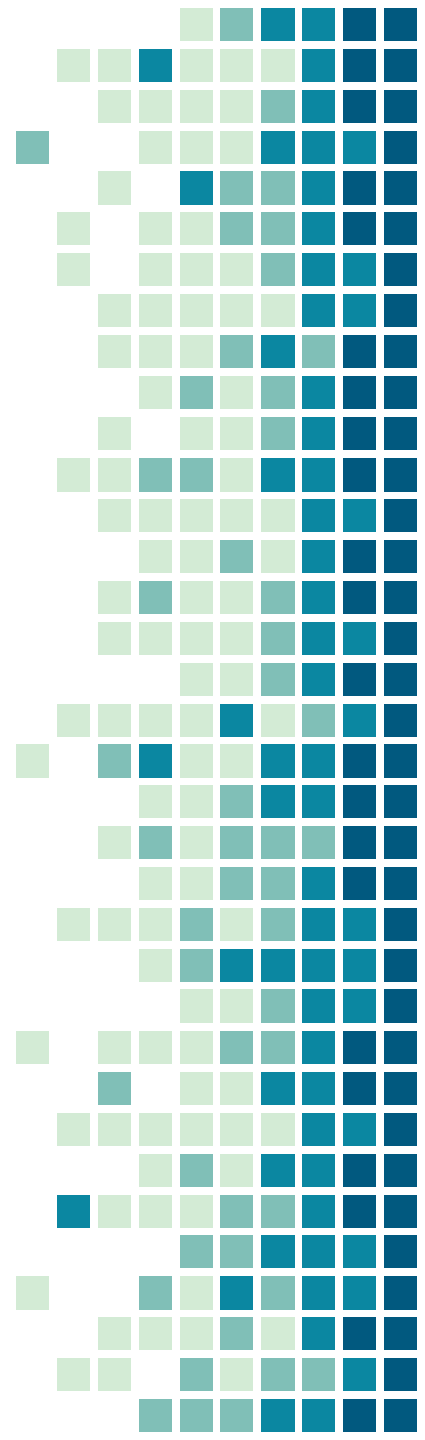


Other Mitigations



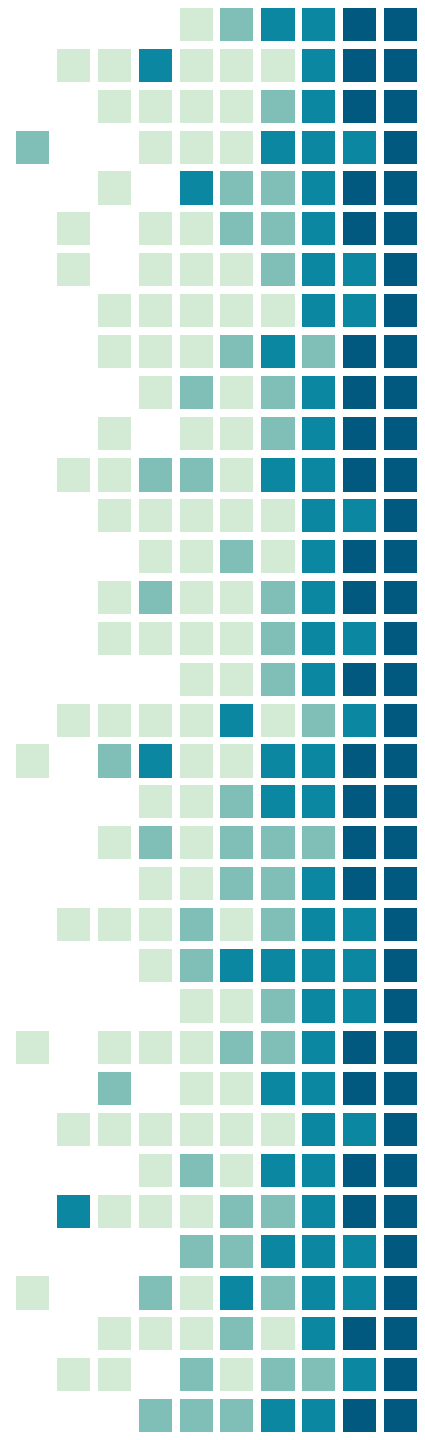
Lots of Other Mitigations Lying Around

- Win32k syscall filtering
- Disable dynamic code
- Prefer System32 images
- Block non-Microsoft binaries
- Font loading mitigations
- Side channel mitigations
- Child creation mitigation



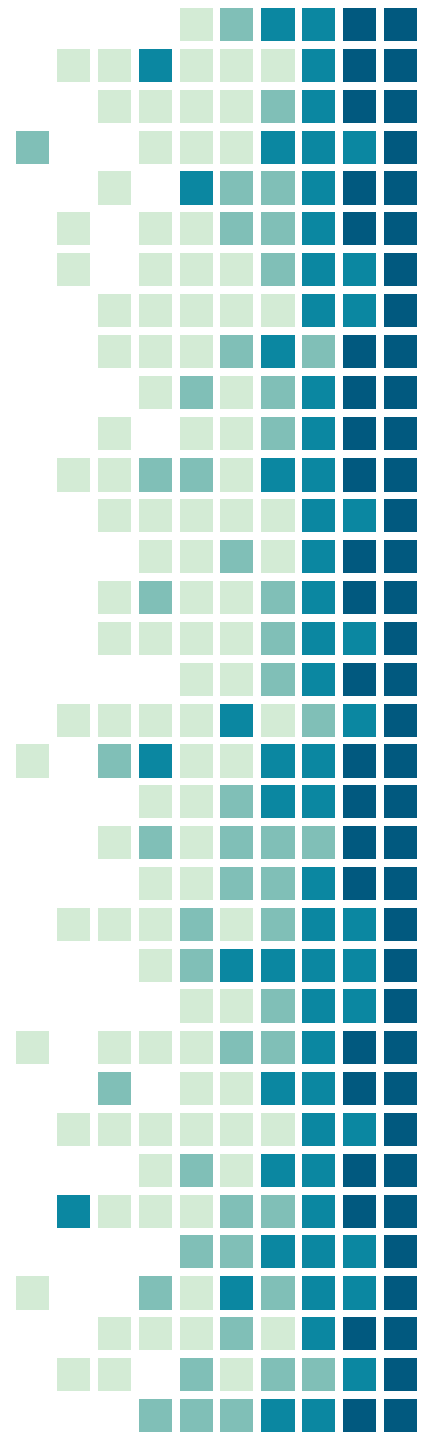
Who Uses All These Mitigations?

- Chrome, edge, spooler, fontdrvhost use some of the newer mitigations
 - Block/filter Win32k syscalls
 - Disable non-System fonts
 - Disable dynamic code
- Most 3rd party code uses none of the newer mitigations
- Even a lot of Microsoft code doesn't use them
 - Office, OneDrive...



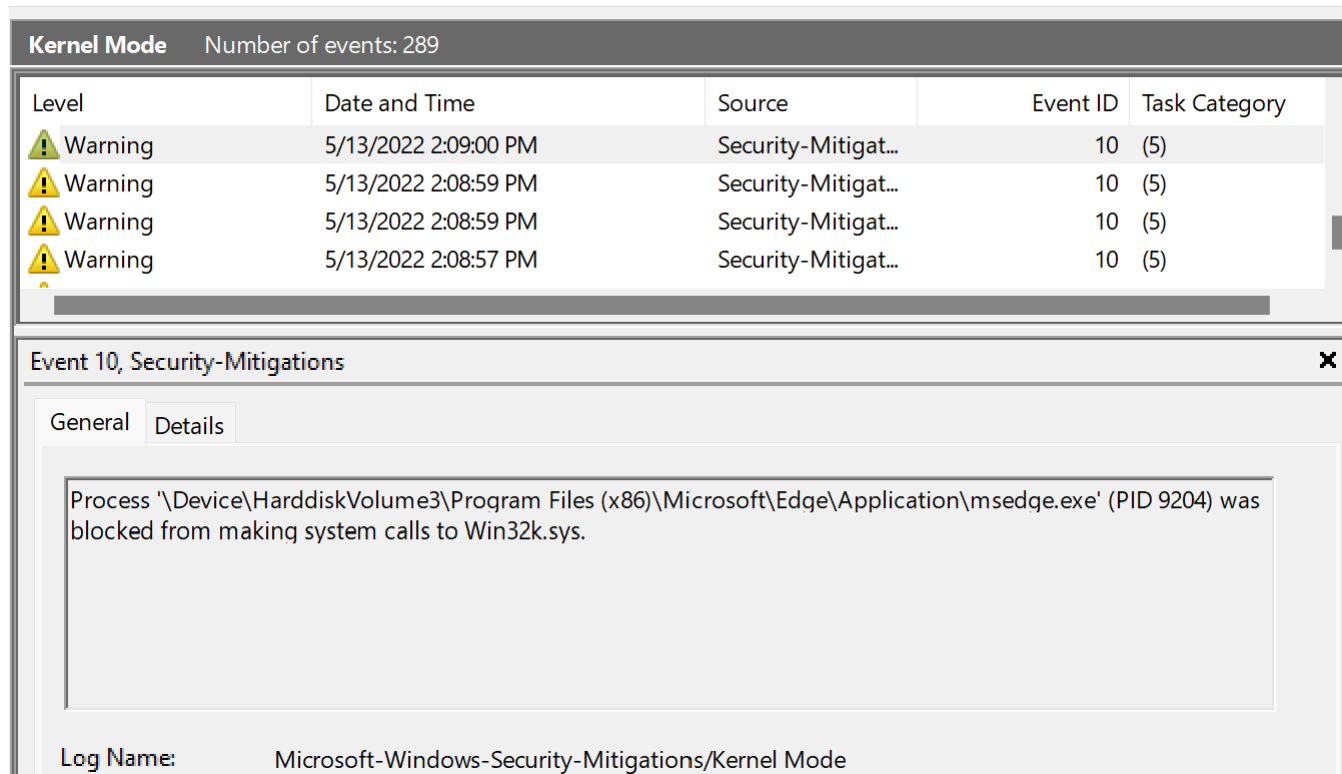
Forensics

- Mitigations in audit mode will generate ETW events
- Some Windows processes enable a lot of those
- For example, Print Spooler enables:
 - AuditDisableDynamicCode
 - AuditNonSystemFontLoading
 - AuditProhibitRemoteImageMap
 - AuditProhibitLowILImageMap
 - AuditBlockNonMicrosoftBinaries



Forensics

- Look for ETW events indicating potential security issues



The screenshot displays the Windows Event Viewer interface. The top pane shows a list of events under 'Kernel Mode' with a total of 289 events. The bottom pane provides details for 'Event 10, Security-Mitigations'.

Level	Date and Time	Source	Event ID	Task Category
Warning	5/13/2022 2:09:00 PM	Security-Mitigat...	10	(5)
Warning	5/13/2022 2:08:59 PM	Security-Mitigat...	10	(5)
Warning	5/13/2022 2:08:59 PM	Security-Mitigat...	10	(5)
Warning	5/13/2022 2:08:57 PM	Security-Mitigat...	10	(5)

Event 10, Security-Mitigations

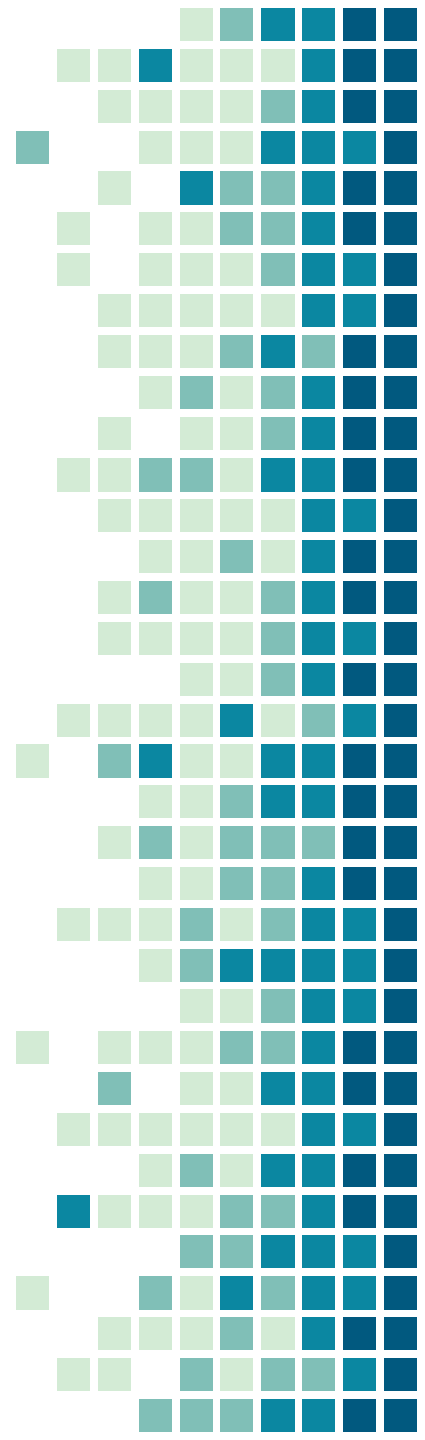
General Details

Process '\Device\HarddiskVolume3\Program Files (x86)\Microsoft\Edge\Application\msedge.exe' (PID 9204) was blocked from making system calls to Win32k.sys.

Log Name: Microsoft-Windows-Security-Mitigations/Kernel Mode

Conclusions

- There are a lot of process mitigations on Windows
 - So many that EPROCESS needs MitigationFlags3 in Windows 11
- Many are not documented at all, or not very well
- Enabling those can stop entire bug classes
 - But also break functionality, if you're not careful
- But enabling them in audit mode is safer and can still help get visibility into process actions



Questions?

