Abusing privileged file operations

Privilege escalation low-hanging fruits

Clément Lavoillotte (@clavoillotte)
Provadys

TROOPERS 2019
Lightning Talk

Privileged file operation bugs

provadys

- **File operations by a privileged process**
  - Problems occur when an unprivileged user/process has some control over that file
  - Works with all kinds of resources, files are just an easy target
- **Examples**
  - Service started from a user-writable EXE file
  - DLL loaded in a privileged process from a user-writable location
- **Quite common in (security) software**
  - Access rights misconfiguration
  - Access to user-owned files without impersonation or restrictions
  - Time Of Check vs. Time Of Use (TOCTOU)
- **Logic bugs**
  - Very stable (no memory corruption)
  - Can survive code refactoring
  - Cross-architecture

How to find these bugs

provadys

- **No assembly required**
  - for the low-hanging ones
- **Process Monitor** 
  - Filters on the product's privileged processes
  - Useful to find triggers
    - Perform actions as unprivileged user, look at the effects
  - Fast and effective
  - Userland only
- **Debugger, API Monitor, etc**
- **Explorer / icacls / AccessChk / Get-Acl**
  - Any way to view ACLs on files / folders

Techniques & tools

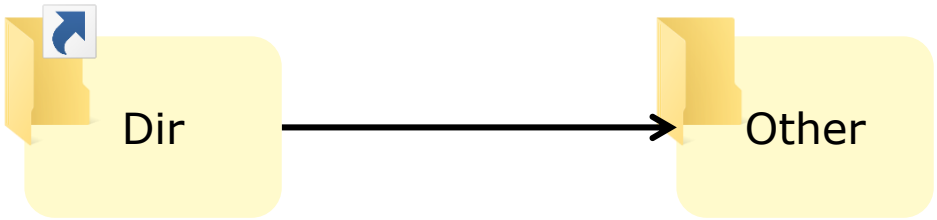
provadys

- **Useful techniques as an unprivileged user**
  - NTFS mount points (junctions)
  - Object manager symbolic links
  - Opportunistic Locks
  - Combinations
  - Courtesy of James Forshaw (@tiraniddo)
    - "A Link to the Past - Abusing Symbolic Links on Windows" at SyScan & Infiltrate 2015 (must watch!)
    - Following descriptions are shameless over-simplifications
- **Tools**
  - James' purpose-built tools & libraries
    - https://github.com/googleprojectzero/symboliclink-testing-tools
    - https://github.com/googleprojectzero/sandbox-attacksurface-analysis-tools
  - Windows built-in tools (powershell, cmd, filesystem utilities)
  - SysInternals
- **Many filesystem-level attacks are now low-hanging fruits**

Techniques & tools (cont.)

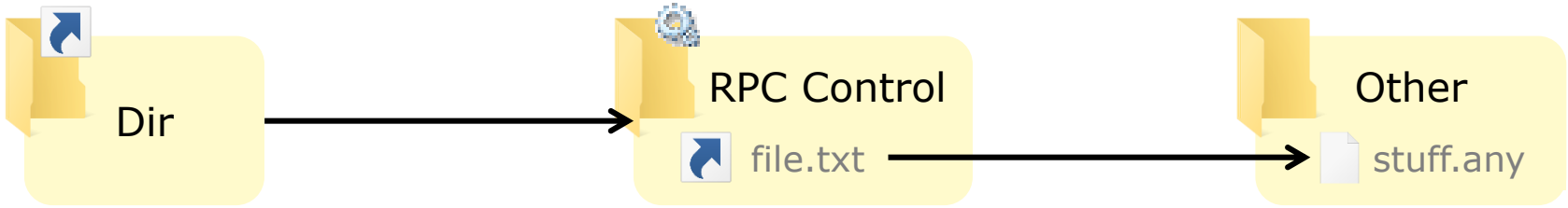
provadys

- **NTFS mount points (junctions)**
  - Redirects a directory to another directory
    - CreateMountPoint.exe, junction.exe, mklink /j, New-Item –Type Junction



`C:\Dir\file.exe` resolves to `C:\Other\file.exe`

- **Object manager symbolic links**
  - Links in the object manager namespace that can point to files, even if the file does not exist
    - NativeSymlink.exe, CreateDosDeviceSymlink.exe, WinObj.exe
  - Junction + Object Manager symbolic link = pseudo-symlink
    - CreateSymlink.exe



`C:\Dir\file.txt` resolves to `C:\Other\stuff.any`

Exploiting arbitrary file writes

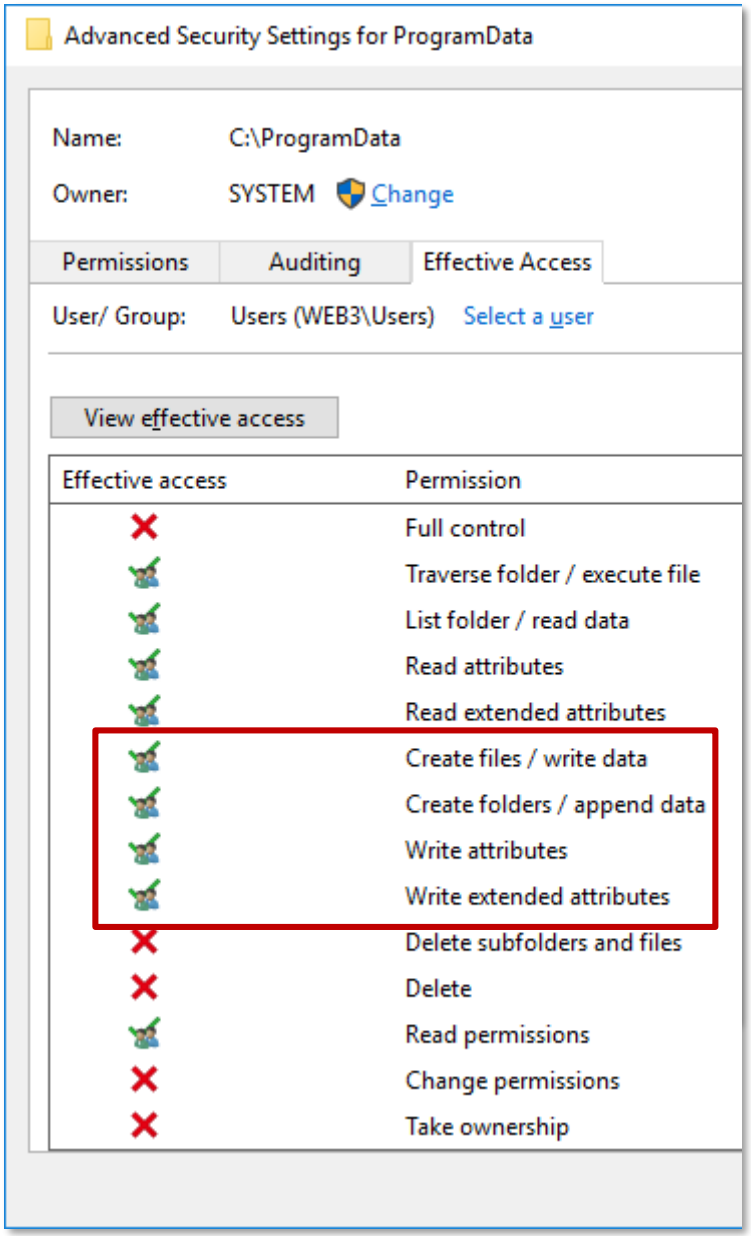
provadys

- **Replace an existing binary / config**
  - if overwrite is possible
- **Drop a DLL somewhere in the PATH of a privileged process**
  - needs a process that can be (re)started
  - example : Wow64Log.dll + privileged 32-bits process
    - 64-bits DLL loaded by WoW64 in all 32-bits processes if present in System32
    - Documented by Walied Assar http://waleedassar.blogspot.com/2013/01/wow64logdll.html
    - Not present by default
    - Can't import Kernel32, use NTDLL only
- **Drop/replace any file in System32**
  - Diagnostics Hub Service
    - Helpful trick by James Forshaw https://googleprojectzero.blogspot.com/2018/04/windows-exploitation-tricks-exploiting.html
    - The privileged DiagHub service can be made to load from System32 a file with any extension as a DLL
    - Microsoft added a ProcessImageLoadPolicy to DiagHub Windows 10 19H1 https://twitter.com/x9090/status/1090860643429736448

Bug: Arbitrary file write in <Product> logging

provadys

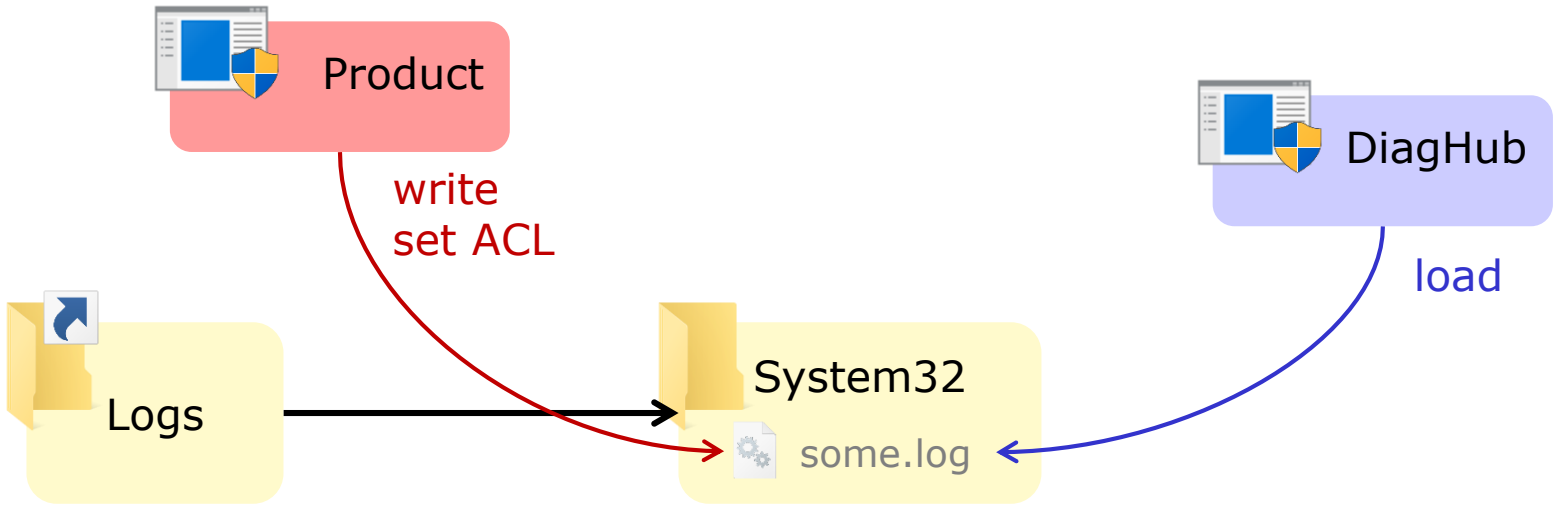
- **Log file with over-permissive ACL**
  - Explicit permissive ACL, e.g. Everyone has full control over the log files
  - File created with no ACL inherited / set
- **Created in user-writable location**
  - Users can also add files in / set properties of its parent folder
  - Subfolder of `C:\ProgramData`, with default inherited access rights
    - Unprivileged users can create files & directories
    - But not to modify existing files
- **Write from a privileged service/process**
  - Without impersonation



Bug: Arbitrary file write in <Product> logging

provadys

- **Exploitation: via the DiagHub service**
  - Delete existing log files
  - Replace `C:\ProgramData\Product\Logs` by a junction to `C:\Windows\System32\`
  - Trigger/wait for log creation
    - `some.log` is created in the target folder with the permissive ACL
  - Replace content of `some.log` by payload
  - Start DiagHub and trigger load of `some.log` as a DLL → payload runs as SYSTEM

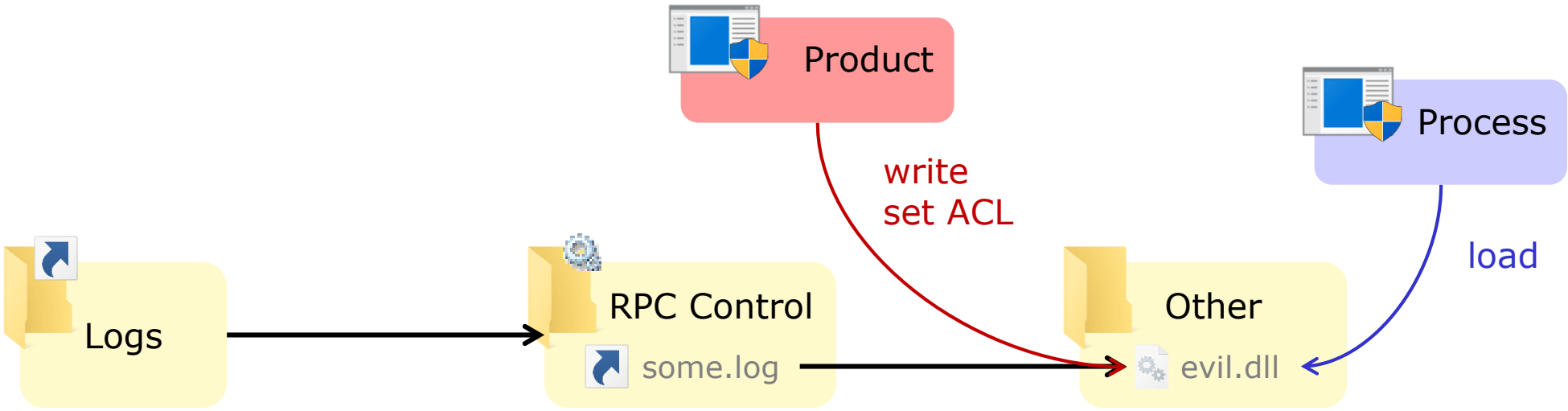


`C:\ProgramData\Product\Logs\some.log` resolves to `C:\Windows\System32\some.log`

Bug: Arbitrary file write in <Product> logging

provadys

- **Exploitation: generic DLL hijacking**
  - Delete existing log files
  - Replace `C:\ProgramData\Product\Logs` by a junction to the `\RPC Control\` object directory
  - Create a `some.log` symlink in `\RPC Control\` that points to the target path `C:\Other\evil.dll`
  - Trigger/wait for log creation
    - `evil.dll` is created in the target folder with the permissive ACL
  - Replace content of `evil.dll` by payload
  - Trigger start of target privileged process → payload runs with its privileges



`C:\ProgramData\Product\Logs\some.log` resolves to `C:\Other\evil.dll`

Bug: Arbitrary file write in <Product> logging

provadys

- **Unsecure implementation of a common need**
  - Logs in `C:\ProgramData` with default access rights
  - Logs meant to be world-writable → permissive ACL set on created logs
  - Privileged component writes log and forgets impersonation

  ➔ Exploitable condition

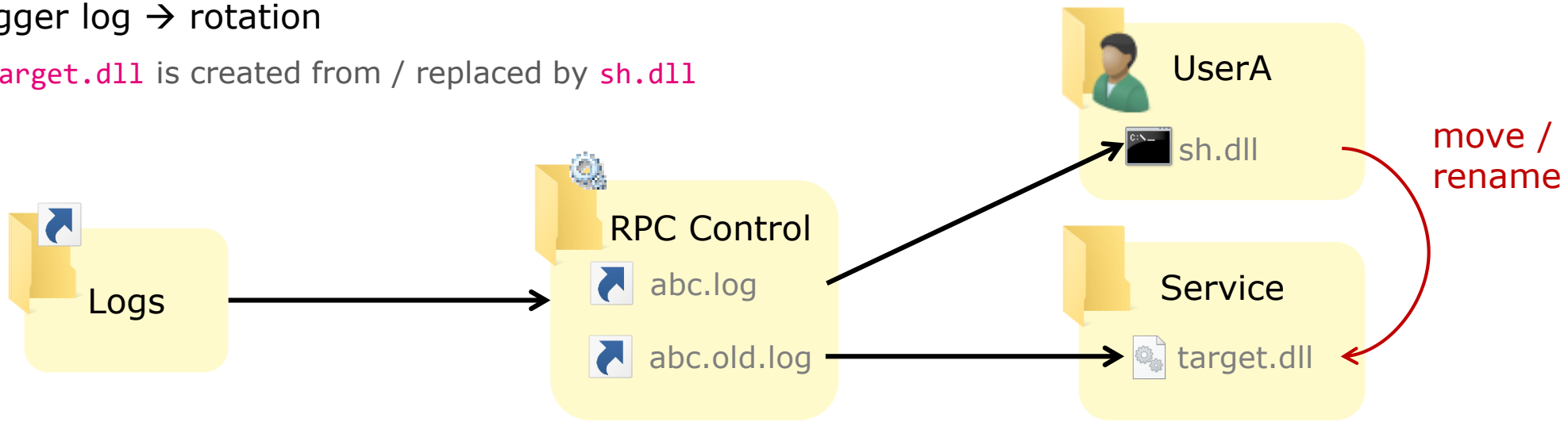
- **Instances found in multiple <Products>**
  - In Cylance by Ryan Hanson (@ryhanson)
  - In McAfee Endpoint Security (patched) found 05/2018
  - In NVIDIA and Intel utilities by Mark Barnes (@incanus)
  - Pulse Secure VPN client (unpatched), found 06/2018 (collision w/ Matt Bush @3xocyte)
  - Other products (fix still pending)

Exploiting arbitrary file writes (cont.)

provadys

- **Controlling the content**
  - Previous example assumes a permissive ACL set on the created file
  - By default: ACL inherited from parent directory → no write access to file
  - How to exploit more generic cases?

- **Target a different operation**
  - Example: log file rotation → move/rename operation
  - Replace moved/renamed files by pseudo-symlinks
    - Replace the source (`abc.log`) by a link to a controlled (big) file (`sh.dll`)
    - Replace the destination (`abc.old.log`) by a link to the target file (`target.dll`)
  - Trigger log → rotation
    - `target.dll` is created from / replaced by `sh.dll`



File layout before move/rename of `C:\Dir\abc.log` *to* `C:\Dir\abc.old.log`

File operations as exploit primitives

provadys

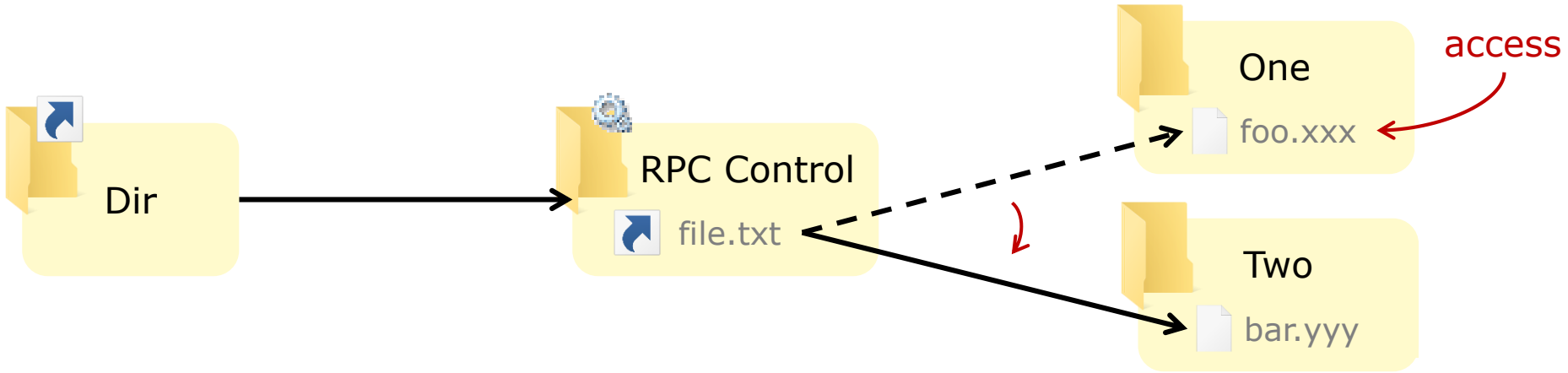
- **Some operations are very powerful when you can control them**
  - And not only on log files
- **Controlled move/rename = arbitrary file write**
  - move payload into `System32`
  - move `cmd.exe` to `sethc.exe`
- **Controlled copy = arbitrary file read/write**
  - copy with controlled source & destination = arbitrary file write
  - copy with controlled source and user-readable destination = arbitrary file read
    - read `SAM` / `SECURITY` / `SYSTEM` hives to dump SAM db, cached creds, etc
- **ProcMon as a ~~poor man's~~ file operation debugger**
  - An actual debugger is also very useful
- **Some operations require precise timing**

Techniques & tools (cont.)

provadys

- **Opportunistic Locks (OpLocks)**
  - Placed on a file/directory to trigger an action (callback) when it is accessed
    - SetOpLock.exe
  - Can turn some race conditions into reliable exploit
  - Some limitations : one-shot, not all types of access

- **Combined**
  - Pseudo-symlink + OpLock = “BaitAndSwitch”
    - BaitAndSwitch.exe
  - Useful for TOCTOU



`C:\Dir\file.txt` resolves to `C:\One\foo.xxx`

then to `C:\Two\bar.yyy`

AV

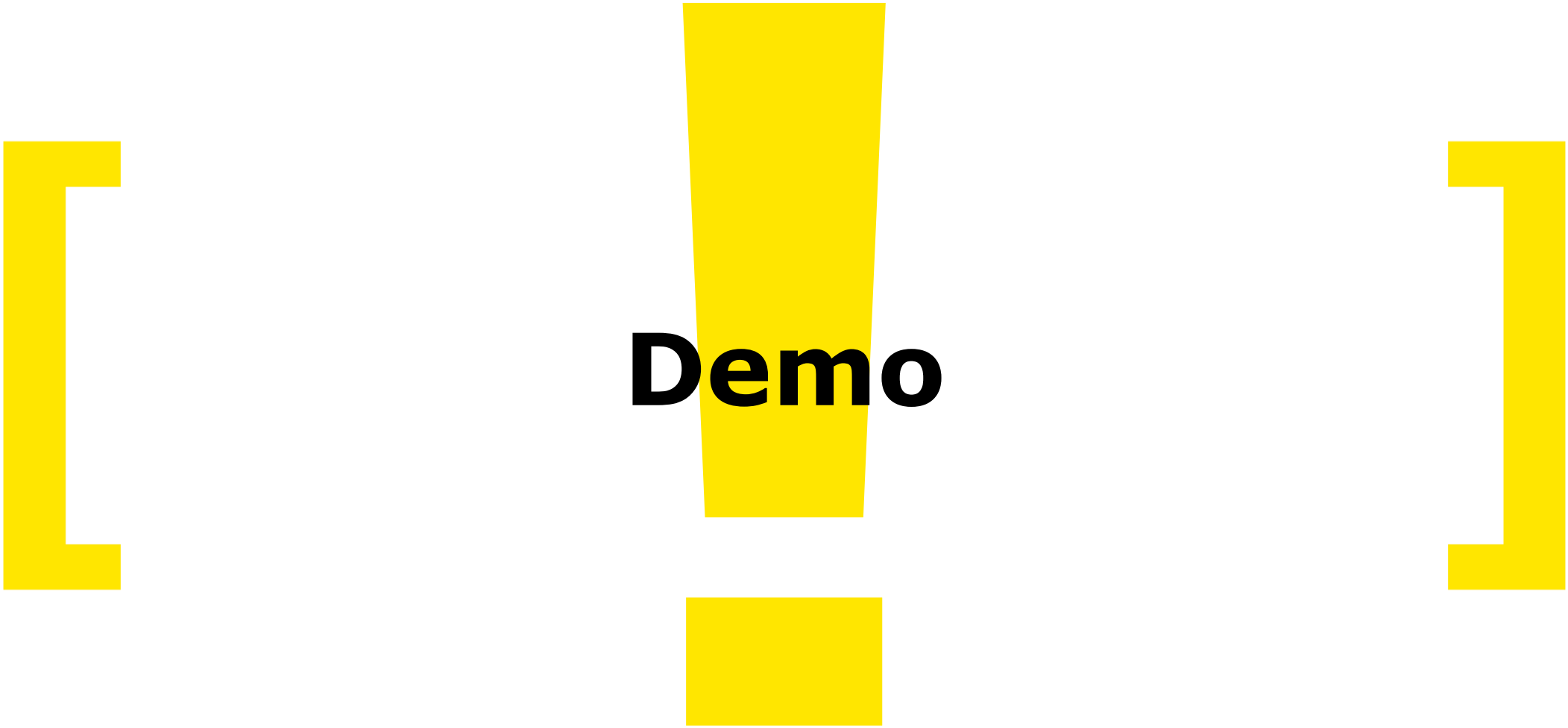
provadys

- **Files are usually scanned / removed / restored by a privileged process**
  - Sometimes without impersonation
  - Can be triggered by unprivileged users
    - EICAR or malicious file, manual quarantine (sometimes), etc.
    - Or disabled in the UI but accessible via COM hijacking, as shown by Bálint Varga-Perke (@buherator)

- **Abuse potential**
  - Scanning a file → privileged file read
  - Putting a file in Quarantine → privileged file read/copy
  - Deleting the original file → privileged file delete
  - Restoring a file → privileged file write

- **Some AVs perform operations before removing an infected file**
  - Create/delete temporary files in the same directory
  - Copy or move/rename the infected file in a user-writable location
  - Copy or move the infected file to a user-readable quarantine location

  - Fun: quarantine `C:\Windows\System32\config\SAM` then read the quarantine file


Demo

Abusing file deletion

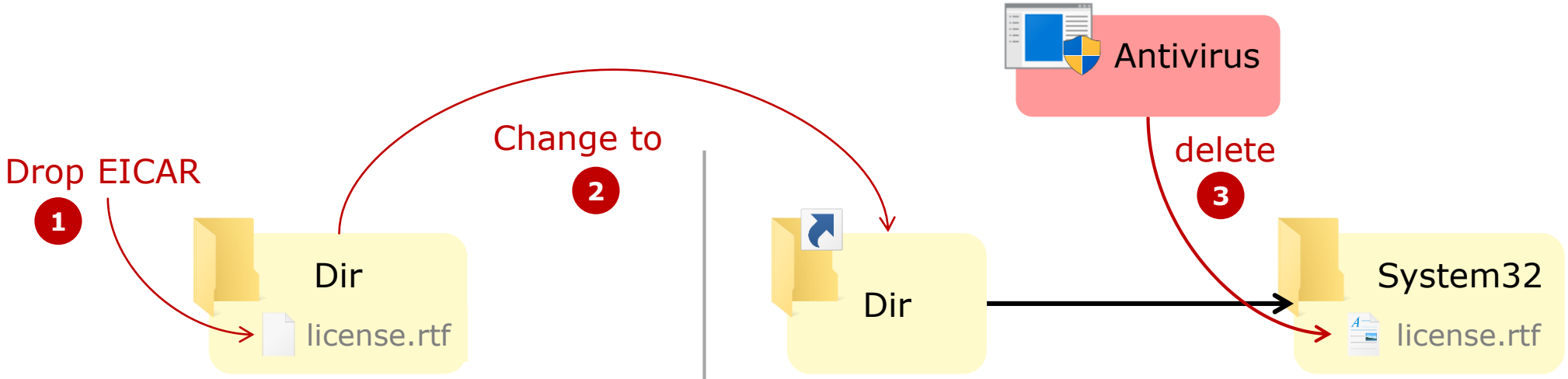
provadys

- **Files are removed when deemed malicious**
  - Manipulate the file and/or the deletion process
  - Remove arbitrary files
  - So… what?
- **Exploiting arbitrary delete**
  - Remove files that we can replace and that will be later used by a privileged process
    - `C:\ProgramData` and `C:\Windows\Temp`
  - Replace dirs / files → now these files are user-controlled
  - Now use the privileged file writes exploitation techniques on these files
- **AV software is an obvious target for these**
  - Similar technique to exploit installers (and others programs) that do not check for preexisting files

Redirected file deletion

provadys

- **What if the file AV wants to remove is no longer there?**
- **Divert a file deletion with a TOCTOU**
  - Drop EICAR in `C:\Dir\license.rtf`
  - Wait for it to be detected
  - Replace `C:\Dir` by a junction to `C:\Windows\System32`
  - AV deletes `C:\Dir\license.rtf` which reparses to `C:\Windows\System32\license.rtf`



`C:\Dir\license.rtf` is a standard file

`C:\Dir\license.rtf` resolves to `C:\Windows\System32\license.rtf`

Reported bugs & vendor responses

provadys

Product	ID	Vulnerablity	Arbitrary file	Reported	Fix
Symantec Endpoint Protection 12 & 14	CVE-2017-13680	TOCTOU in the quarantine GUI	Deletion Read	09/2017	Available 11/2017
	CVE-2018-5236	TOCTOU during file deletion	Deletion	11/2017	Available 06/2018
	CVE-2018-5237	Check bypass in file restore	Write	11/2017	Available 06/2018
AV product A	TBD	Over-privileged file deletion	Deletion	03/2018	In progress
AV product B	TBD	Over-privileged file restore	Write	05/2018	In progress
McAfee Endpoint Security 10	CVE-2019-3582	Overpermissive access rights Over-privileged file creation	Write Deletion	05/2018	Available 10/2018 & 02/2019
AV product C	TBD	TOCTOU during file deletion	Deletion	05/2018	In progress
AV product D	TBD	TOCTOU during file deletion	Deletion	05/2018	In progress
F-Secure SAFE/CS/CP	(none)	Over-privileged file copy	Write, Read, Delete	07/2018	Available 08/2018
Pulse Secure VPN client	CVE-2018-11002 (collision)	Overpermissive access rights Over-privileged file creation	Write	06/2018	Unavailable
Product F	TBD	Over-privileged file creation	Write	07/2018	In progress
Product G	TBD	Over-privileged file creation	Write	07/2018	In progress
Product H	TBD	Over-privileged file creation	Write	08/2018	In progress
Intel PROset / Wireless	INTEL-SA-00182 / CVE-2018-12177 (collision)	Overpermissive access rights	DACL set	08/2018	Available 01/2019

Product names & additional details will be published as fixes become available

Prevention & detection

provadys

- **Least privilege**
  - Do not break the security boundary in the first place
  - Impersonate or use restricted tokens
- **Harden the product**
  - Work on fully resolved paths
  - Lock before check, release lock after use
  - Restrict access rights
    - Remove write permission to your `ProgramData` & `Windows\Temp` subfolders
    - Also remove read permissions when possible
- **Detection**
  - Some attempts will generate logs
    - Not necessarily alarming ones (e.g. EICAR, Threat mitigated / remediated)
    - Correlate with filesystem changes and privileged process creation
  - Behavioral analysis
    - Unusual processes creating junctions, object manager symlinks, OpLocks
    - Processes (even your own) replacing system files

Questions

More examples & PoCs and at:
https://offsec.provadys.com/

References

provadys

- **Research by James Forshaw / Google Project Zero**
  - https://googleprojectzero.blogspot.com/2015/08/windows-10hh-symbolic-link-mitigations.html
  - https://googleprojectzero.blogspot.com/2015/12/between-rock-and-hard-link.html
  - https://googleprojectzero.blogspot.com/2016/02/the-definitive-guide-on-win32-to-nt.html
  - https://googleprojectzero.blogspot.com/2017/08/windows-exploitation-tricks-arbitrary.html
  - https://googleprojectzero.blogspot.com/2018/04/windows-exploitation-tricks-exploiting.html
  - https://infocon.org/cons/SyScan/SyScan%202015%20Singapore/SyScan%202015%20Singapore%20presentations/SyScan15%20James%20Forshaw%20-%20A%20Link%20to%20the%20Past.pdf
  - https://vimeo.com/133002251
- **Vulnerabilities & other research**
  - https://bogner.sh/2017/11/avgater-getting-local-admin-by-abusing-the-anti-virus-quarantine/, Florian Bogner
  - https://blog.silentsignal.eu/2018/01/08/bare-knuckled-antivirus-breaking/, Bálint Varga-Perke / Silent Signal
  - https://www.atredis.com/blog/cylance-privilege-escalation-vulnerability and https://www.atredis.com/blog/cve-2018-0952-privilege-escalation-vulnerability-in-windows-standard-collector-service, Ryan Hanson / Atredis Partners
  - https://labs.nettitude.com/blog/cve-2018-5240-symantec-management-agent-altiris-privilege-escalation/, Ben Turner / Nettitude Labs
  - https://github.com/SandboxEscaper/randomrepo and https://twitter.com/SandboxEscaper and http://sandboxescaper.blogspot.com/2018/10/reversing-alpc-where-are-your-windows.html, @SandboxEscaper
  - https://www.themissinglink.com.au/security-advisories-cve-2018-11002, Matt Bush The Missing Link Security
  - https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00182.html, Thomas Hibbert / Insomnia Security
  - https://labs.mwrinfosecurity.com/advisories/intel-driver-and-support-assistant-dsa-lpe/ and https://labs.mwrinfosecurity.com/advisories/nvidia-geforce-experience-lpe/, Mark Barnes / MWR Labs