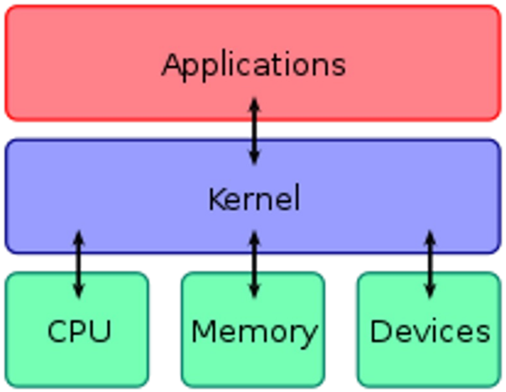
# Security Kernel

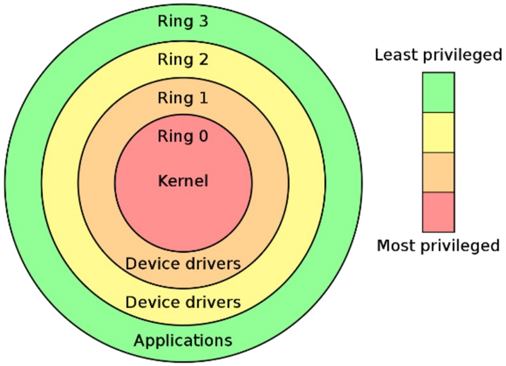
**DSCI 519: Foundations and Policy for Information Security**

Crisrael Lucero
Shruti Krishna Kumar

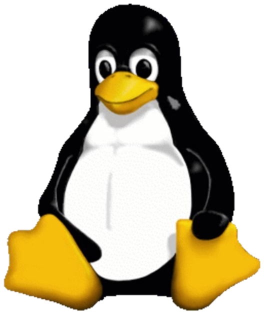
# Operating System Kernel

- Central component of an operating system that manages operations of computer and hardware.
- The major aim of kernel is to manage communication between software i.e. user-level applications and hardware i.e., CPU and disk memory.



# Linux - Kernel Security



Security features provided by Kernels (Linux):

- Discretionary Access Control (DAC)
- Extended DAC
- Process isolation

Provides system admins ability to:

- Ability to remove unnecessary and potentially insecure parts of kernel
- Specify encryption algorithms
- Customize Linux authentication

# Linux Kernel – Security Bugs

Linux kernel security bugs in recent years:

- **CVE-2017-18017:** Present in netfilter tcpmss_mangle_packet function; susceptible to overflow issues and DoS attacks
- **CVE-2016-10229:** allows a remote attacker to execute arbitrary code via UDP traffic
- **CVE-2016-10150:** use-after-free vulnerability
- **CVE-2015-8812:** enables a remote attacker to execute arbitrary code attack via crafted packets
- **Several Wi-Fi vulnerabilities patched (CVE-2022-41674, 42719, 42720…)**

# Linux - Kernel Vulnerabilities & Mitigations

| Vulnerabilities | Mitigation Options |
| --- | --- |
| Kernel pointer leaks | kernel ASLR |
| direct kernel overwrite | Executable memory cannot be writable (CONFIG_STRICT_KERNEL_RWX) |
| function pointer overwrite | read-only function tables (e.g. PAX_CONSTIFY_PLUGIN) |
| userspace execution | hardware segmentation<br>emulated memory segmentation via page table swap |

# Linux - Kernel Security

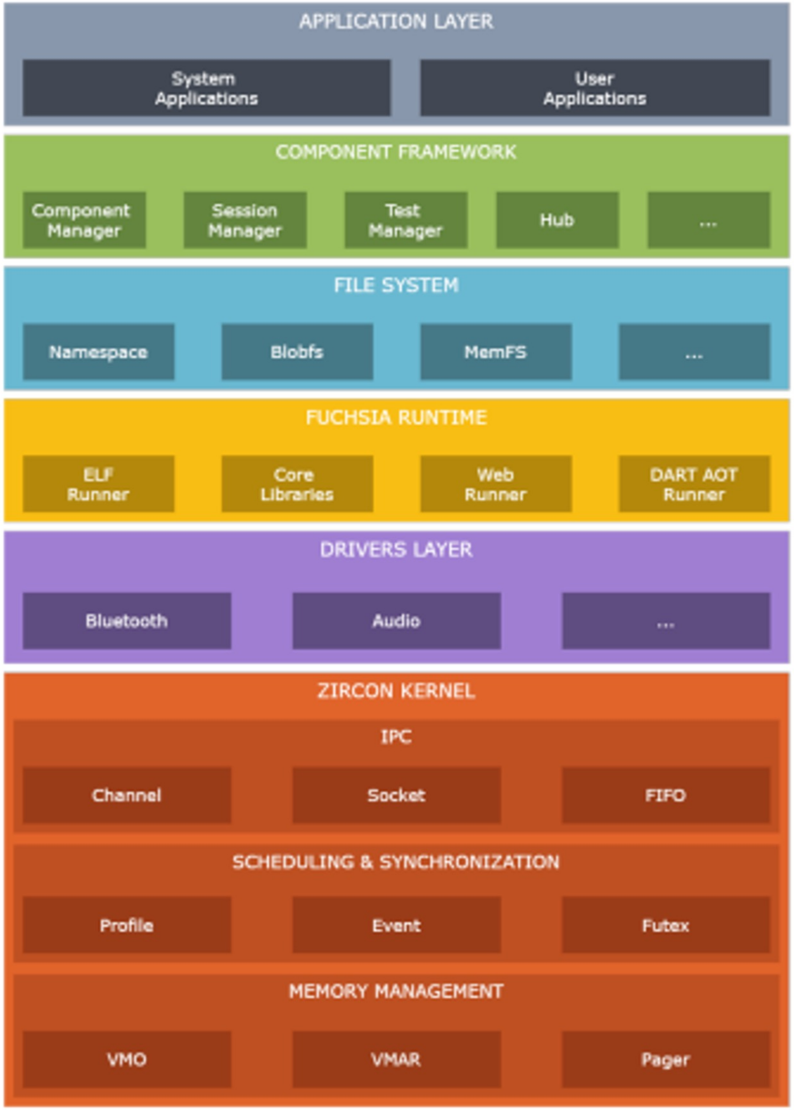
Security features provided by Kernels (Linux):

- Extended DAC
- POSIX Access Control Lists: Allowing separate permissions for individual users and different groups. They're managed with the setfacl and getfacl commands.
- Linux Security Modules - Allows different security models to be plugged into the kernel
- *SELinux*(Security Enhanced Linux) – MAC implementation
- Secure computing mode (seccomp): Restricts access to system calls by processes.

# Fuchsia OS and Zircon Kernel Overview

- Fuchsia OS is a general-purpose open source OS developed by Google.
- The OS was designed for the IoT ecosystem, so it has a very different security architecture than Linux/GNU.

- Fuchsia is based on the Zircon microkernel. Compared to Linux, a lot of functionality is moved outside of the Zircon microkernel, which decreases the kernel's attack surface.
- No concept of a **user**, but instead Fuchsia is **capabilitiy-based**, which means the kernel resources are exposed to processes (subjects) as objects that require corresponding capabilities.


APPLICATION LAYER
System Applications
User Applications
COMPONENT FRAMEWORK
Component Manager
Session Manager
Test Manager
Hub
...
FILE SYSTEM
Namespace
Blobfs
MemFS
...
FUCHSIA RUNTIME
ELF Runner
Core Libraries
Web Runner
DART AOT Runner
DRIVERS LAYER
Bluetooth
Audio
...
ZIRCON KERNEL
IPC
Channel
Socket
FIFO
SCHEDULING & SYNCHRONIZATION
Profile
Event
Futex
MEMORY MANAGEMENT
VMO
VMAR
Pager

# Zircon Services vs Other Kernels

### Typical OS Kernel Services

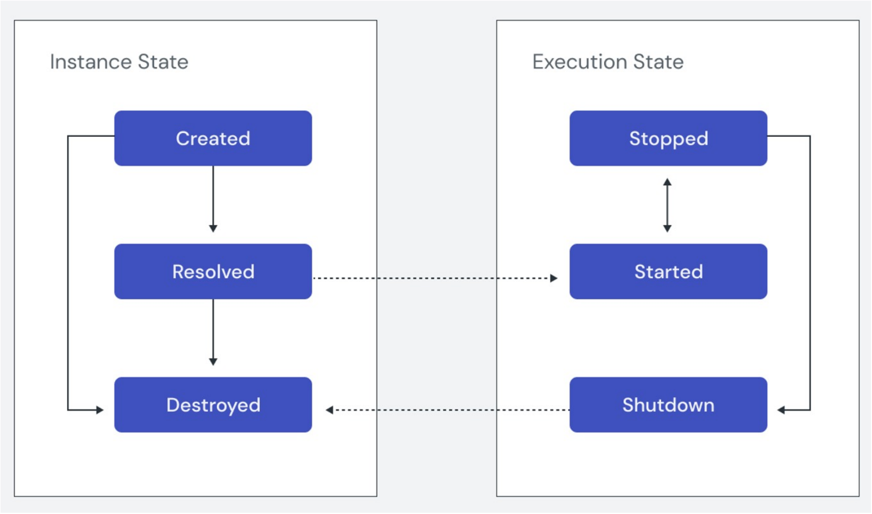
- IPC
- Memory Management
- Scheduler
- Process Management
- User Permissions
- Networking
- Filesystems
- Device Drivers

### Zircon Kernel Services

- IPC
- Memory Management
- Scheduler
- Process Management

# Sandboxing Applications

- Applications and system services outside of the kernel exist as **components**.
- Each component runs in isolated sandboxes, any IPC between components must be explicitly declared. There isn't even a global filesystem, instead each sandbox has a local namespace it operates in.



# Zircon Vulnerabilities

- **KASLR bypass**
  - Very similar to a Linux vulnerability (CVE-2021-26708)
- **Planting a rootkit in userspace**
  - Allowed some kernelspace functionality to be called in userspace.
- **Fake vtables exploiting SMAP**
  - Supervisor Mode Access Prevention makes it so processes in kernelspace can't access userspace data.
  - Fake vtables allow you to subvert this policy.

# Vulnerability Trends

**Vulnerability Trends Over Time**

| Year | # of Vulnerabilities | DoS | Code Execution | Overflow | Memory Corruption | Sql Injection | XSS | Directory Traversal | Http Response Splitting | Bypass something | Gain Information | Gain Privileges | CSRF | File Inclusion | # of exploits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2022 | 4 | | 1 | 1 | | | | | | 1 | 1 | | | | |
| Total | 4 | | 1 | 1 | | | | | | 1 | 1 | | | | |
| % Of All | | 0.0 | 25.0 | 25.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 25.0 | 25.0 | 0.0 | 0.0 | 0.0 | |

Fuchsia

**Vulnerability Trends Over Time**

| Year | # of Vulnerabilities | DoS | Code Execution | Overflow | Memory Corruption | Sql Injection | XSS | Directory Traversal | Http Response Splitting | Bypass something | Gain Information | Gain Privileges | CSRF | File Inclusion | # of exploits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2015 | 57 | 4 | 19 | 6 | 6 | | | | | 10 | 5 | 26 | | | |
| 2016 | 172 | 6 | 47 | 23 | 7 | | | | | 19 | 31 | 82 | | | |
| 2017 | 262 | 32 | 49 | 15 | 2 | | 1 | | | 18 | 103 | 19 | | | |
| 2018 | 258 | 21 | 45 | 6 | 1 | | 1 | 1 | | 40 | 30 | 1 | | | |
| 2019 | 448 | 34 | 142 | 6 | 7 | | 1 | 1 | | 17 | 44 | 3 | | | |
| 2020 | 807 | 29 | 100 | 103 | 20 | | 1 | 1 | | 18 | 97 | 74 | | | |
| 2021 | 486 | 38 | 112 | 2 | 6 | | | | | 31 | 26 | | | | |
| 2022 | 463 | 35 | 129 | | | | | | | 24 | | | | | |
| Total | 2953 | 199 | 643 | 161 | 49 | | 4 | 3 | | 177 | 336 | 205 | | | |
| % Of All | | 6.7 | 21.8 | 5.5 | 1.7 | 0.0 | 0.1 | 0.1 | 0.0 | 6.0 | 11.4 | 6.9 | 0.0 | 0.0 | |

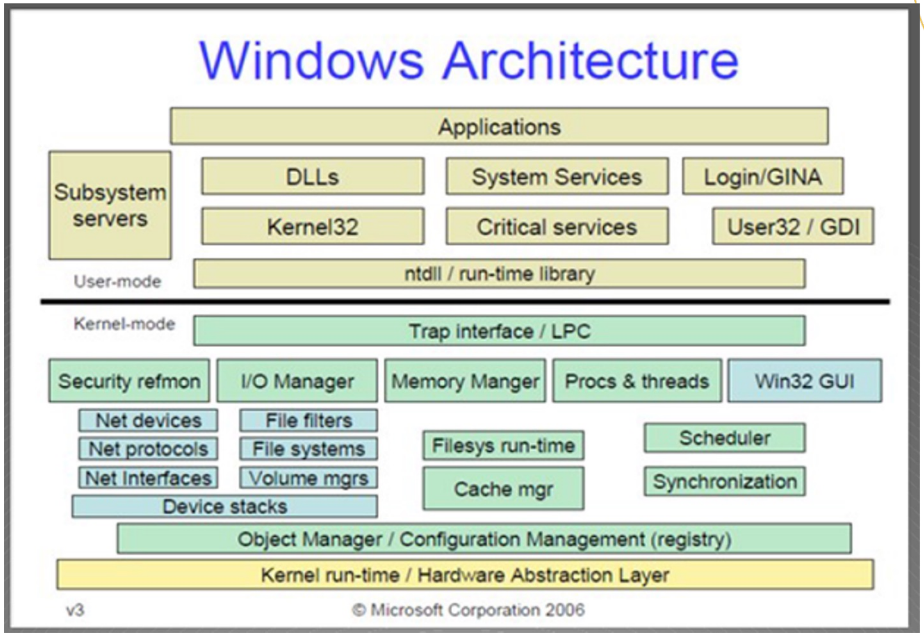
Windows 10

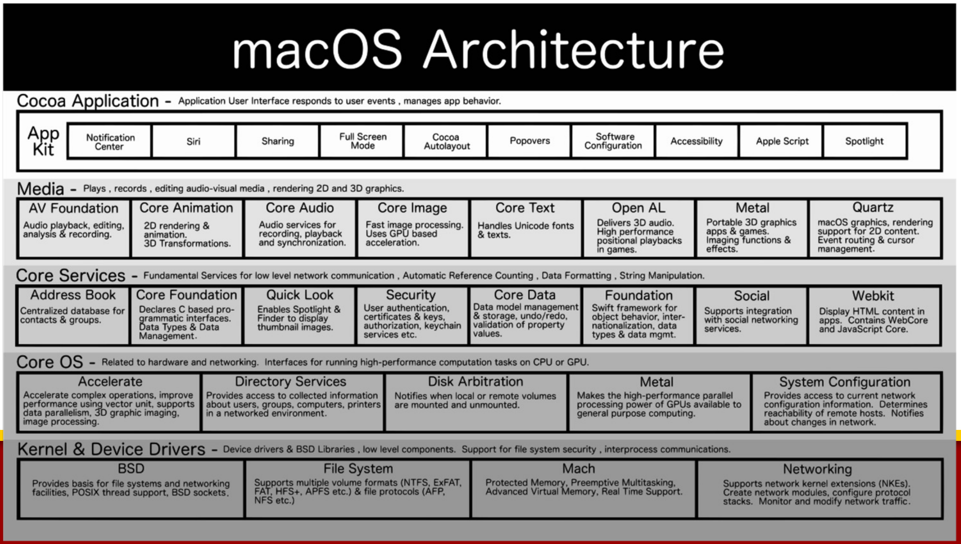
# Proprietary Kernels

**Both Windows and macOS use hybrid kernels**

Hybrid kernels attempt to combine benefits of microkernel and monolithic kernel architectures.

- Microkernels are typically more stable
- Monolithic kernels provide better performance



# Windows / macOS Vulnerabilities

Just because these kernels and operating systems are closed source doesn't mean they have less vulnerabilities than their open source counterparts.

**Windows**
https://www.cvedetails.com/vulnerability-list/vendor_id-26/product_id-32238/Microsoft-Windows-10.html

**macOS**
https://www.cvedetails.com/vulnerability-list/vendor_id-49/product_id-156/Apple-Mac-Os-X.html

# References


- Slide 2 : https://en.wikipedia.org/wiki/Kernel_(operating_system)
- Slide 3 : https://www.linux.com/training-tutorials/overview-linux-kernel-security-features/
- Slide 4: https://www.mend.io/resources/blog/top-10-linux-kernel-vulnerabilities/
- https://linuxsecurity.com/features/how-to-secure-the-linux-kernel
- Slide 4: https://www.zdnet.com/article/linux-dodges-serious-wi-fi-security-exploits/
- Slide 6 : https://docs.kernel.org/security/self-protection.html
- Slide 7, 8, 10: https://arxiv.org/pdf/2108.04183.pdf
- Slide 9, 10, 11: https://swarm.ptsecurity.com/a-kernel-hacker-meets-fuchsia-os/
- Slide 12: https://www.cvedetails.com/