Host-based Intrusion Prevention on Windows and UNIX

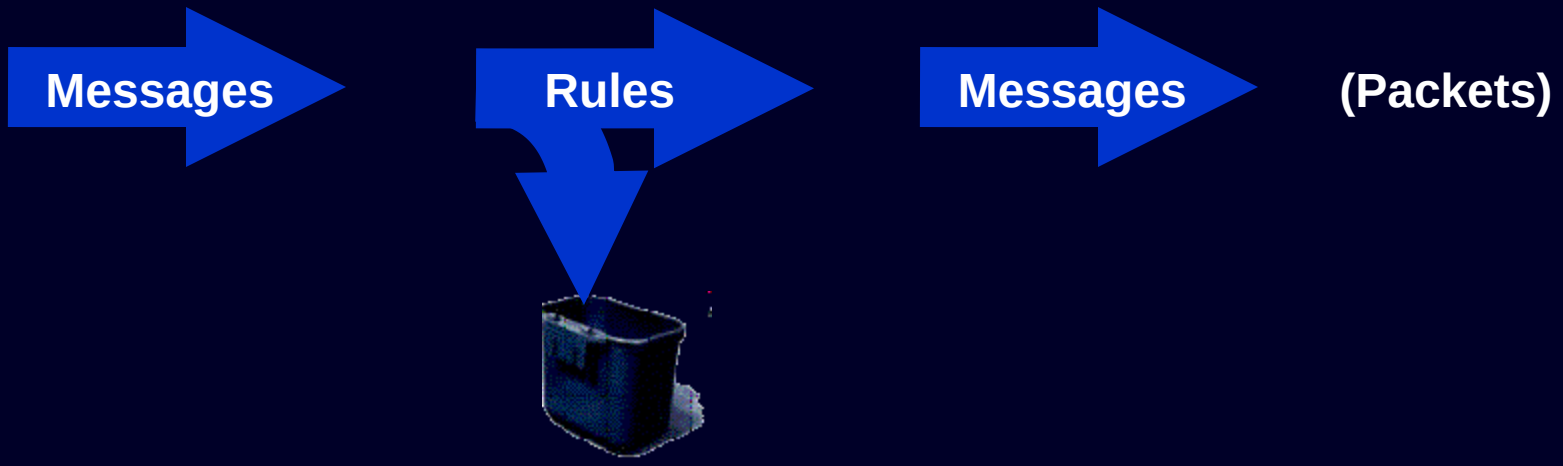
Dr. Rich Murphey
White Oak Labs

Acknowledgements

- Niels Provos – OpenBSD's systrace
- DT – suggested this thread last year
- Greg Hoglund – insights
- md5 at da ghettohackers – reviews

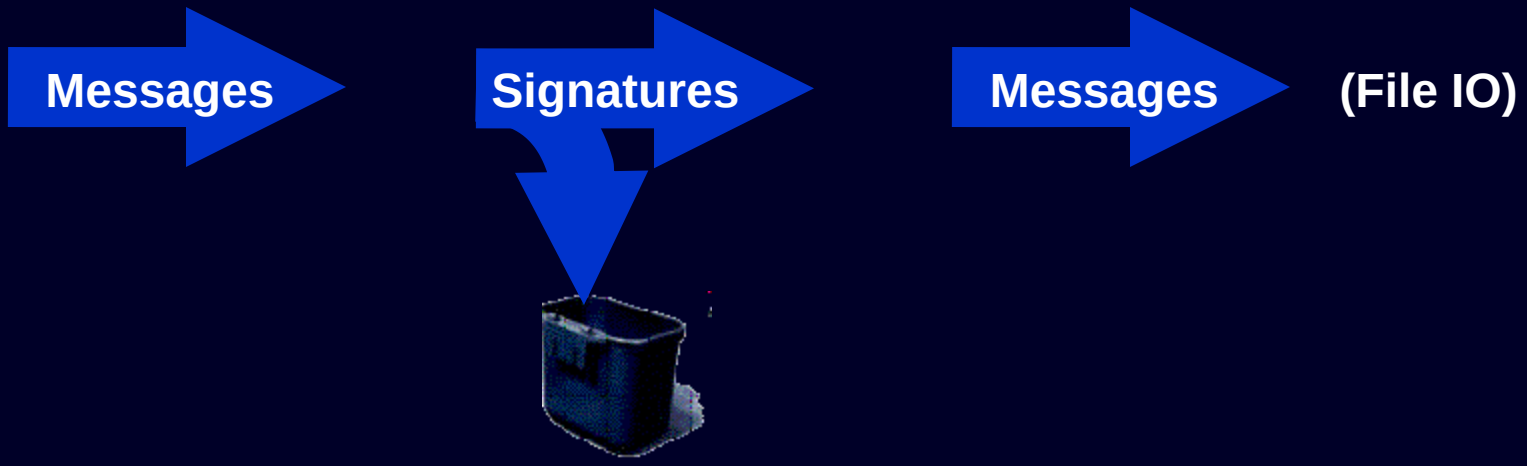
What is Intrusion Prevention?

- To a netsec person it looks like a firewall.



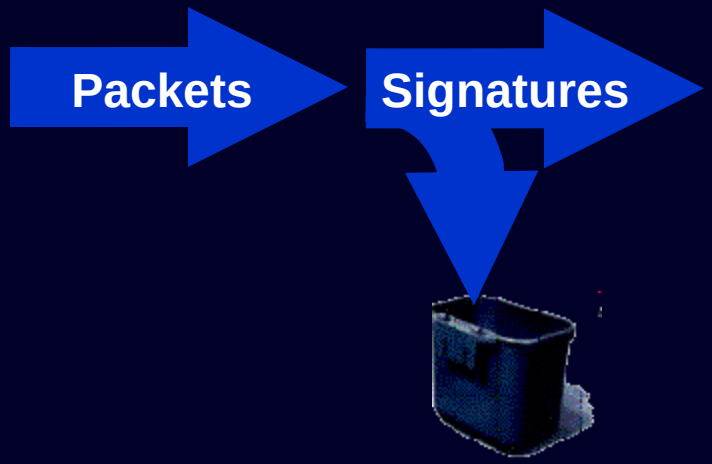
What is Intrusion Prevention?

- To a AV person it looks like an AV.

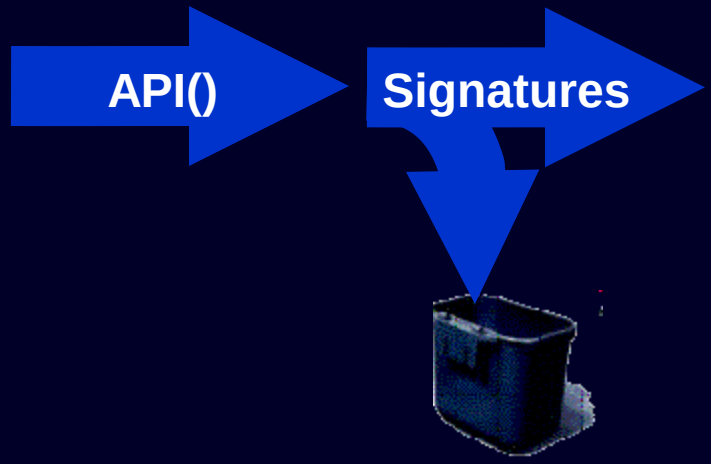


What is Intrusion Prevention?

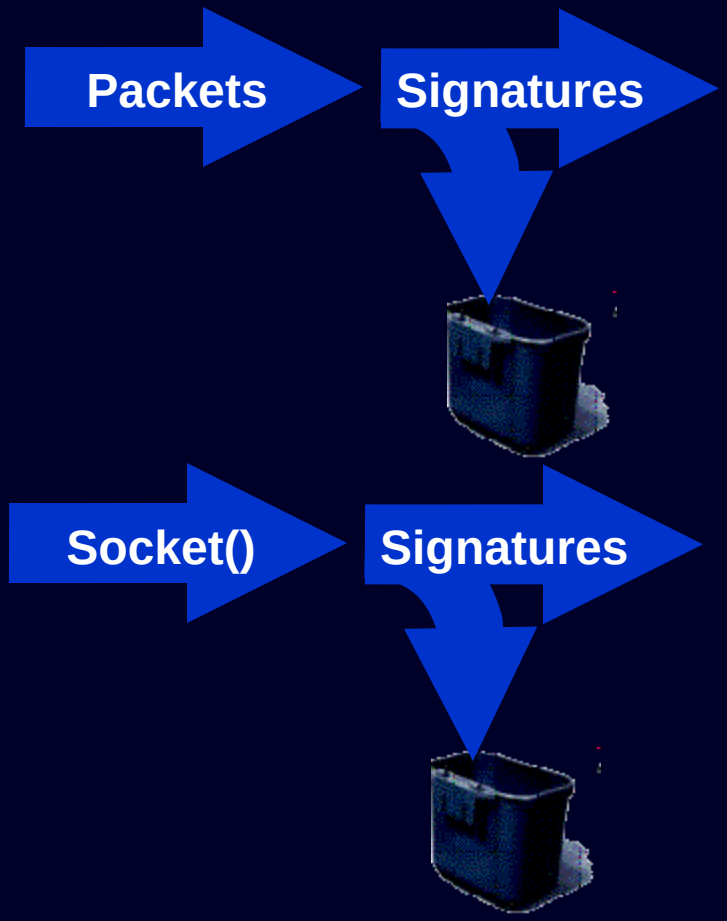
Network-based



Host-based



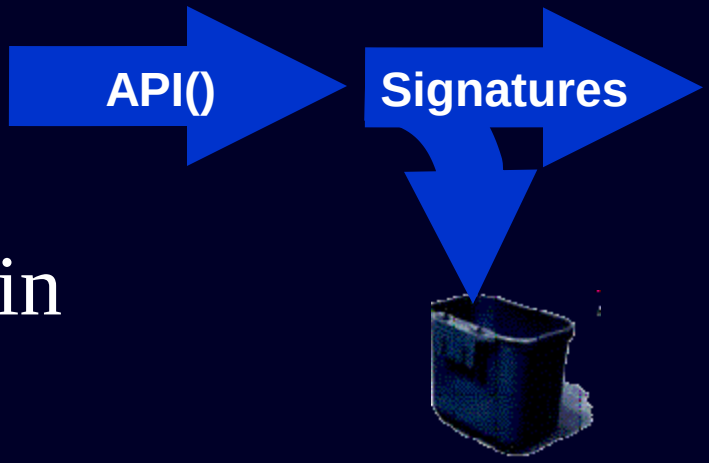
What is Intrusion Prevention?



Consider personal firewalls that combine host and network based filtering.

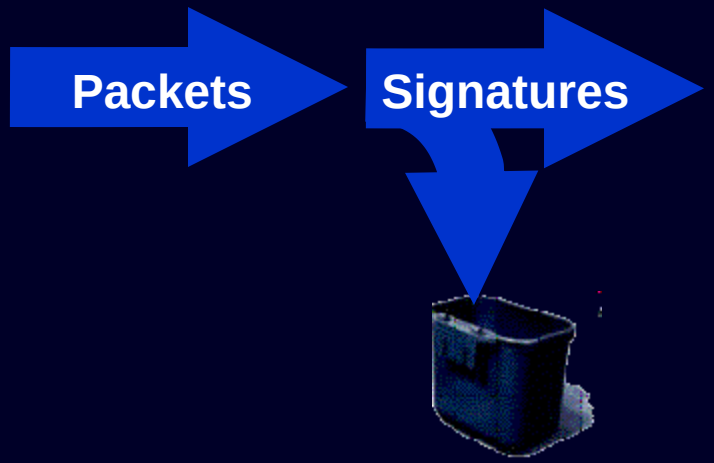
How is IP different?

- Rather than rules, it uses signatures.
- But these aren't the same signatures you might run in an Intrusion Detection Systems (IDS)
- Signatures as access controls.



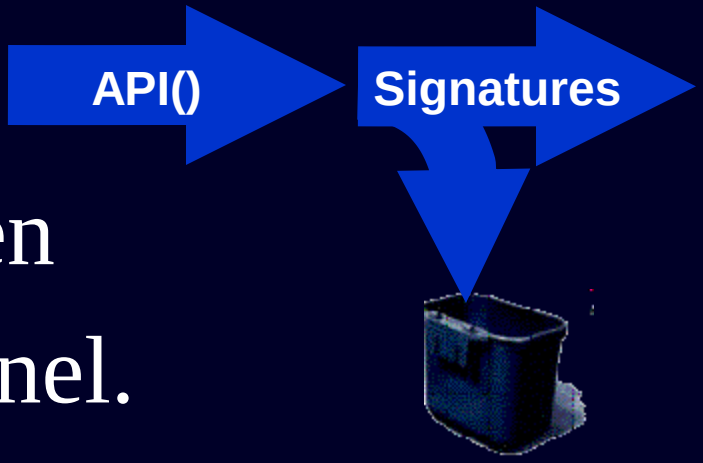
How is it different?

- Consider SNORT/Hogwash..
- Signature-based Firewall
- IDS vendors call this "Gateway IDS"



What is Intrusion Prevention?

- It's complementary to AV & Firewall



  - Filters messages between applications and the kernel.
  - Uses signatures to recognize payload behavior or injection mechanisms.

Why the heck should we care?

- Encryption, fragmentation and re-encoding, can prevent application layer filtering on the wire.

- Data resides in the clear in the application layer. So do exploits.

Why the heck should we care?

- Visibility into the application layer provides capability for better contextual discrimination.

= Stops certain kinds of exploits.

So, why do we need another tool?

- Network security
  - App level Firewalls
- Attack through services/daemons.
  - crunchy on the outside?
- Application state is complex.
  - State of memory, disk, clients…

IP Signatures

For signatures that:

- Are application state specific
- Are system state specific
- Use contextual clues
- Block from the inside

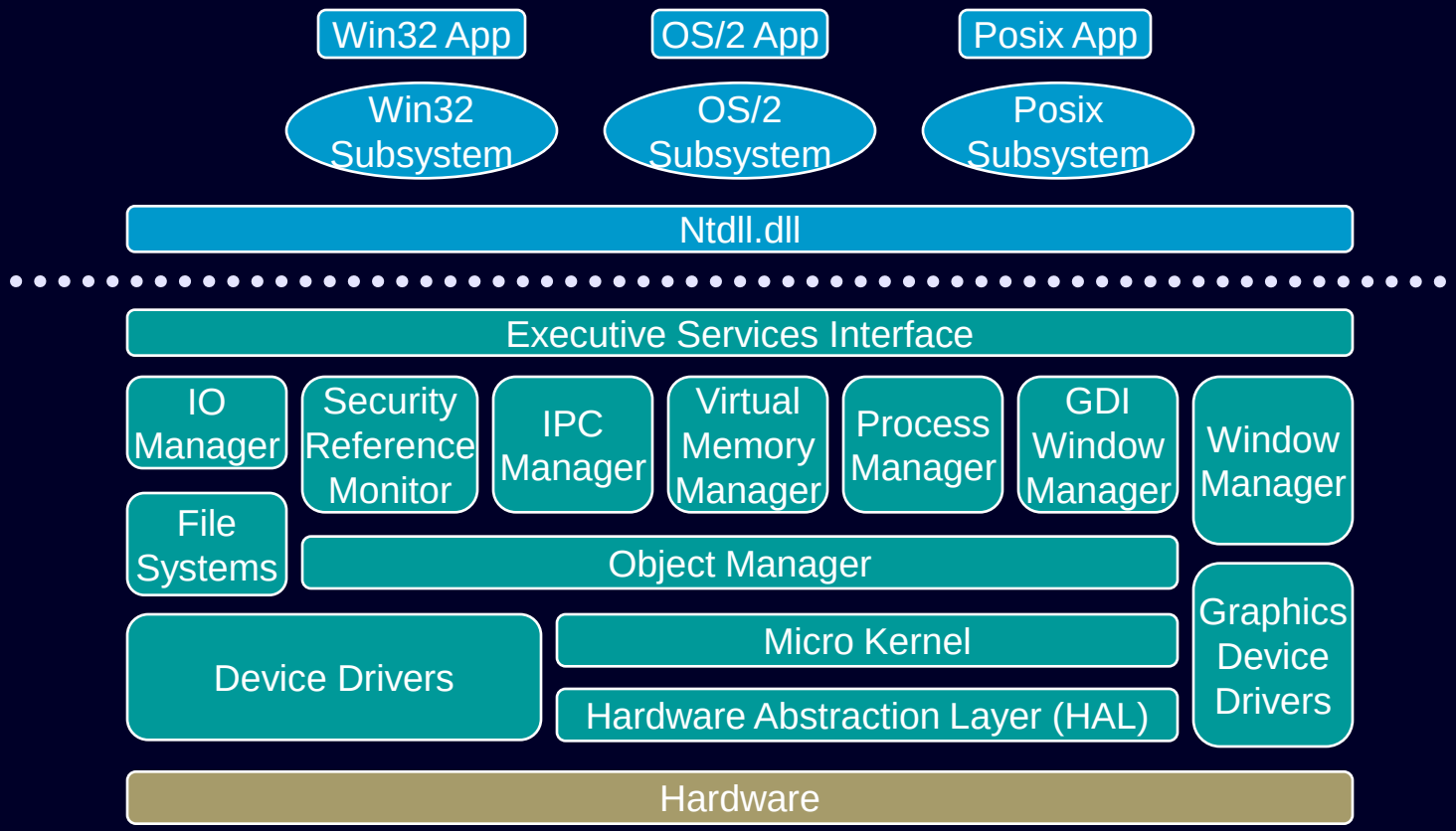
Well, OK, so how does it work?

Consider architectural layers in the OS.

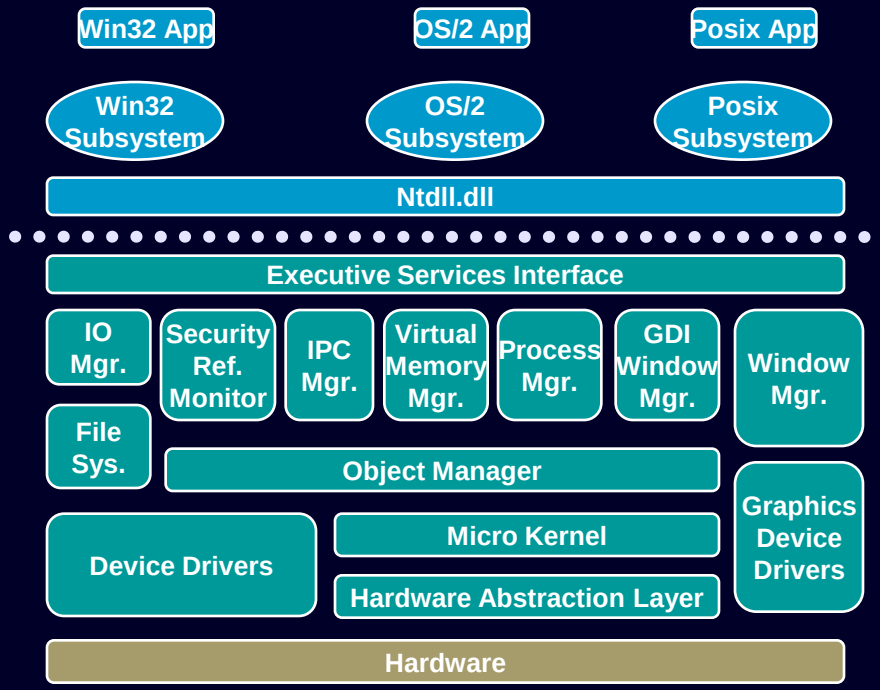
Let's take a look at:

- Layers in Windows architecture
- Layers in UNIX

Win2K System Architecture



Win2K System Architecture



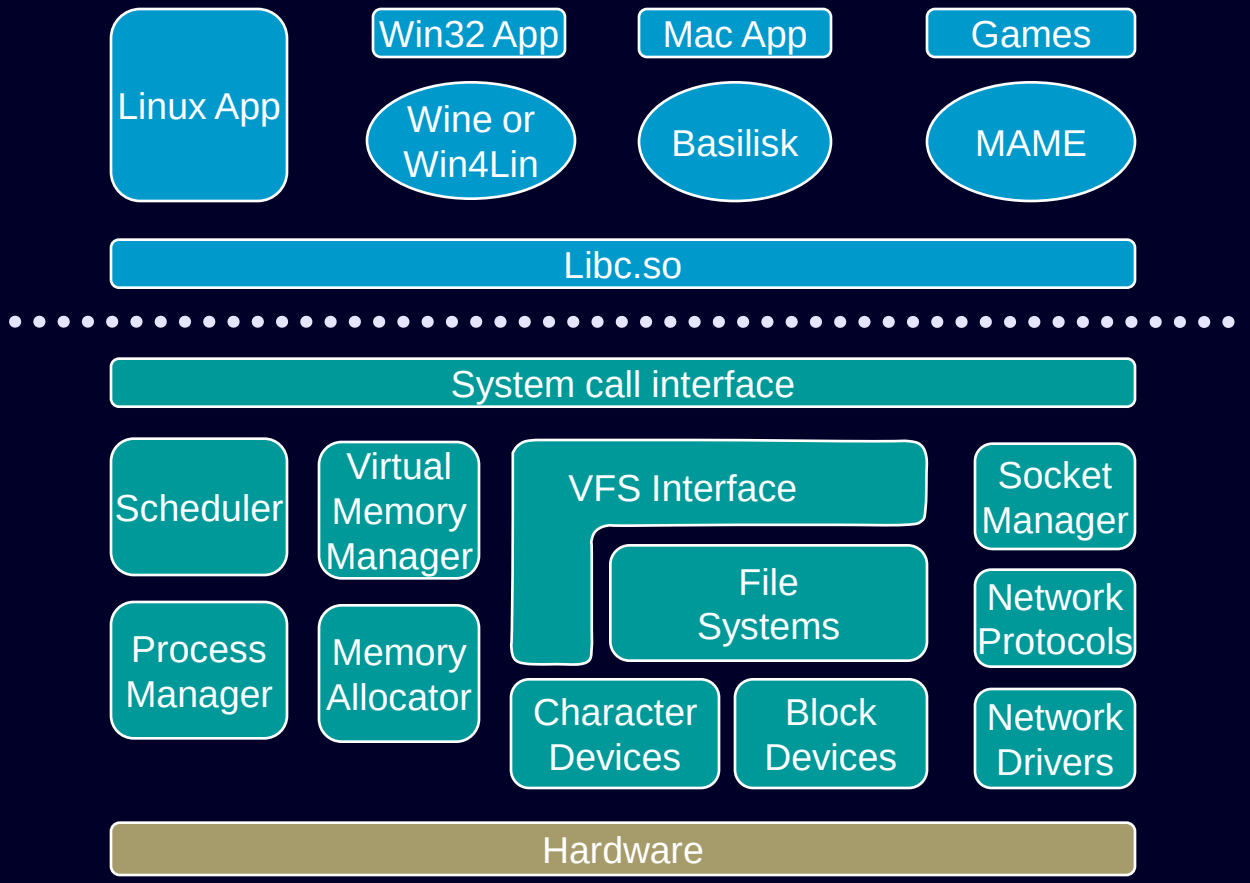
Two distinct user-land layers:

- Binary compatible app layer
- OS specific, native layer

Three distinct kernel layers:

- Executive services
- Object Manager
- Hardware Abstration Layer

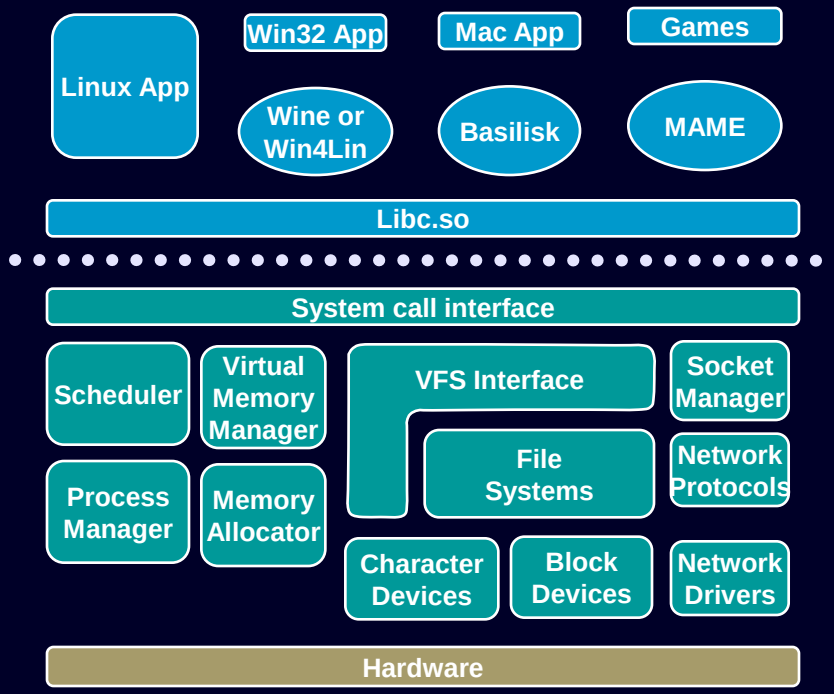
Linux System Architecture



Linux System Architecture

- One common abstraction layer for kernel.
- One common abstraction layer for user-land.

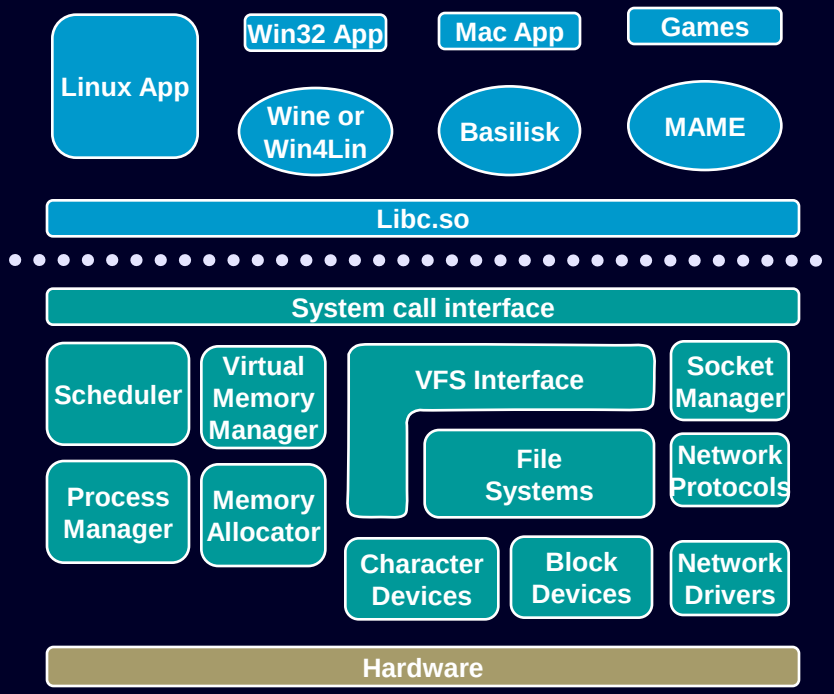
Is fewer layers bad?



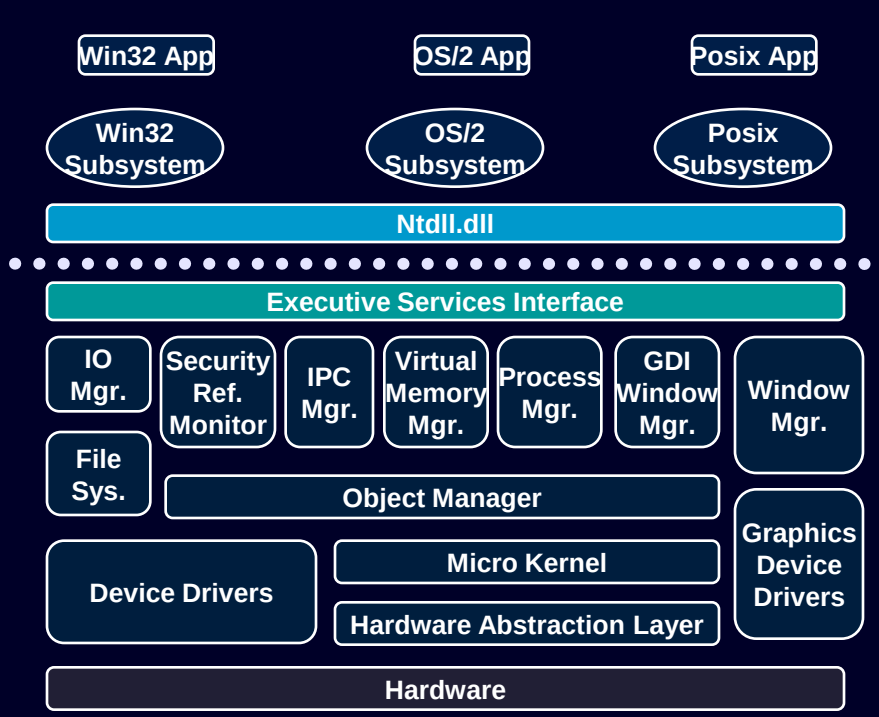
Linux System Architecture

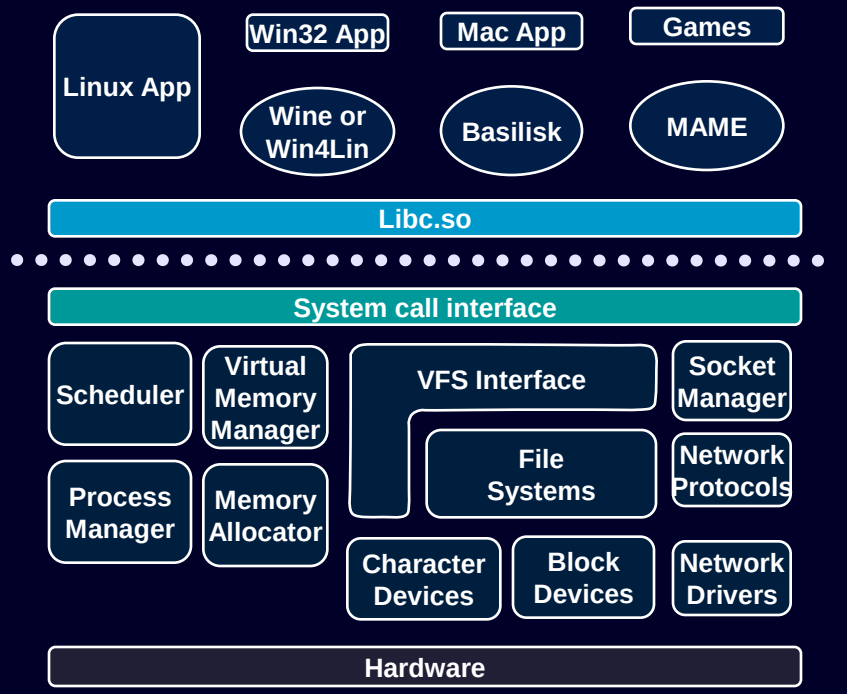
Each layer provides context info.

- What can that information provide?
- Given this information, what can we distinguish?
- How is this different than other forms of filtering?



Architecture Comparison

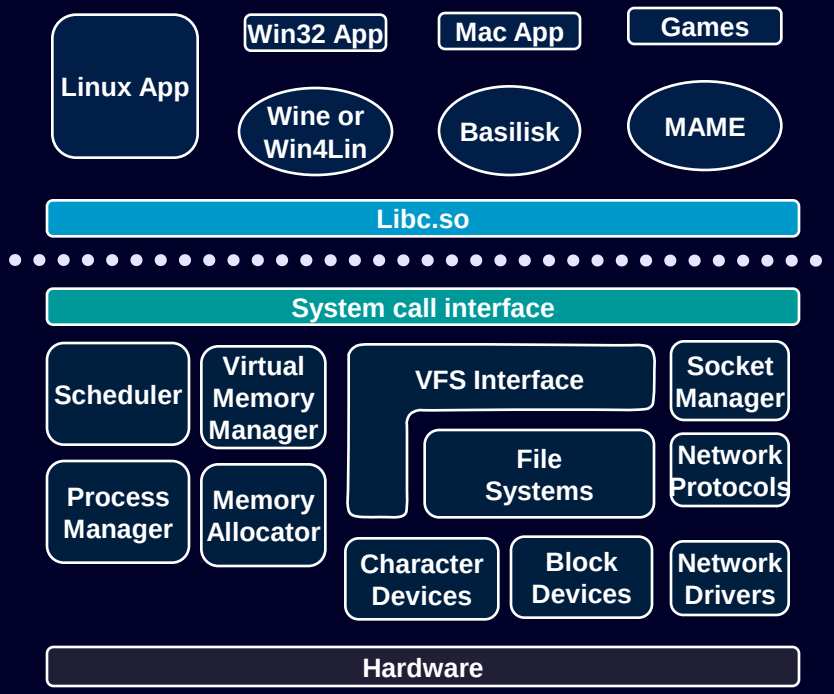




Linux System Architecture

Even considering the app. as a “black box” we can observe at multiple layers:

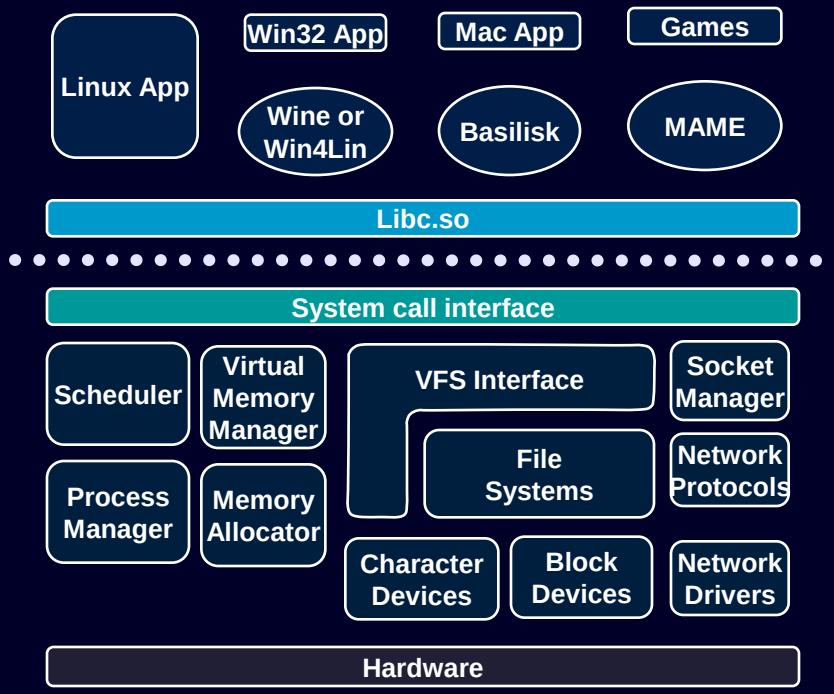
- API calls
- System calls
- Instruction level



Linux System Architecture

Exploits:
- focus on escaping IDS
- often cause exceptional behavior.

Rather than fix IDS, let's take an entirely different approach.



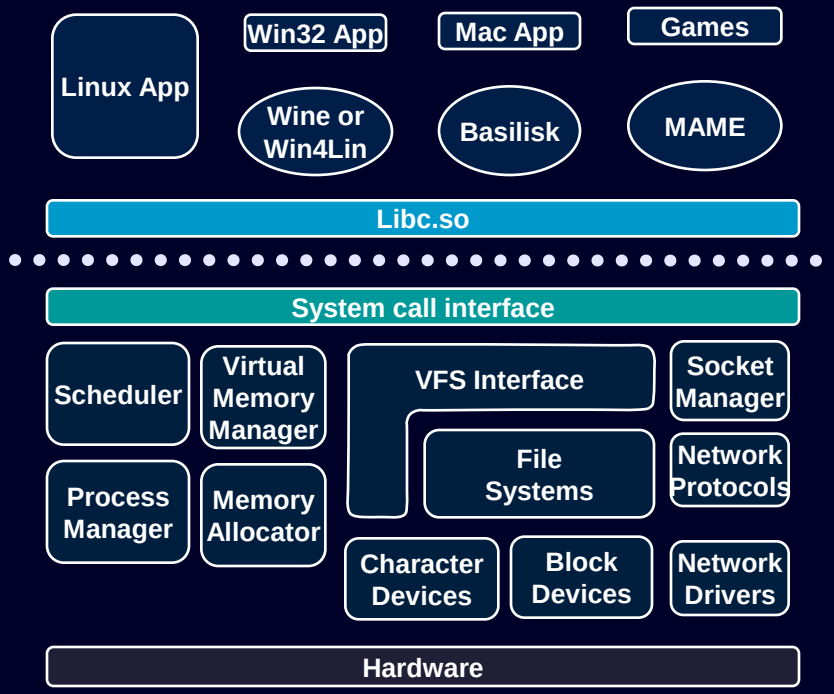
Linux System Architecture

Because it acts at these layers it is independent of:

- Method of transport
- Method of injection.

It is specific to:

- App state or behavior
- Payload behavior



Examples…

- Yes, let's consider a specific payload (and exploit for that matter) on FreeBSD.
- How does it get in and how do we distinguish it's behavior.
- Why is this important...

Demo of IP on FreeBSD

- Background of exploit: release date, authors, impact, subsequent variations.
- firewall coverage
- anti-virus coverage
- IDS and AV coverage

Demo of successful exploit

Review of injection and payload

Code excerpts:

- Vulnerability
- Injection
- Payload

How can we prevent it?

- Unique aspects of payload?
- How we can recognize it..
- Unique aspects of injection?
- How we can recognize it..

Demo of blocked exploit

Review of contextual information

- network identity
- user identity
- application state
- authentication
- workflow
- orthogonally to other methods

What's the state of the industry?

- Leading products in the market?
- What differentiates them?
- How do the integrate with others?
- Success stories?

The End

- Updated slides and tools will be available at:

  www.defcon.org

  www.murphey.org/dc-11/

- I can be contacted at:

  Rich@Murphey.org