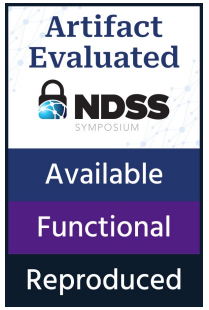


Detecting Ransomware Despite I/O Overhead: A Practical Multi-Staged Approach

Christian van Sloun, **Vincent Woeste**, **Konrad Wolsing**, **Jan Pennekamp**, and **Klaus Wehrle**
RWTH Aachen University

Christian van Sloun, sloun@comsys.rwth-aachen.de
32nd NDSS Symposium, February 2025
San Diego, California, USA

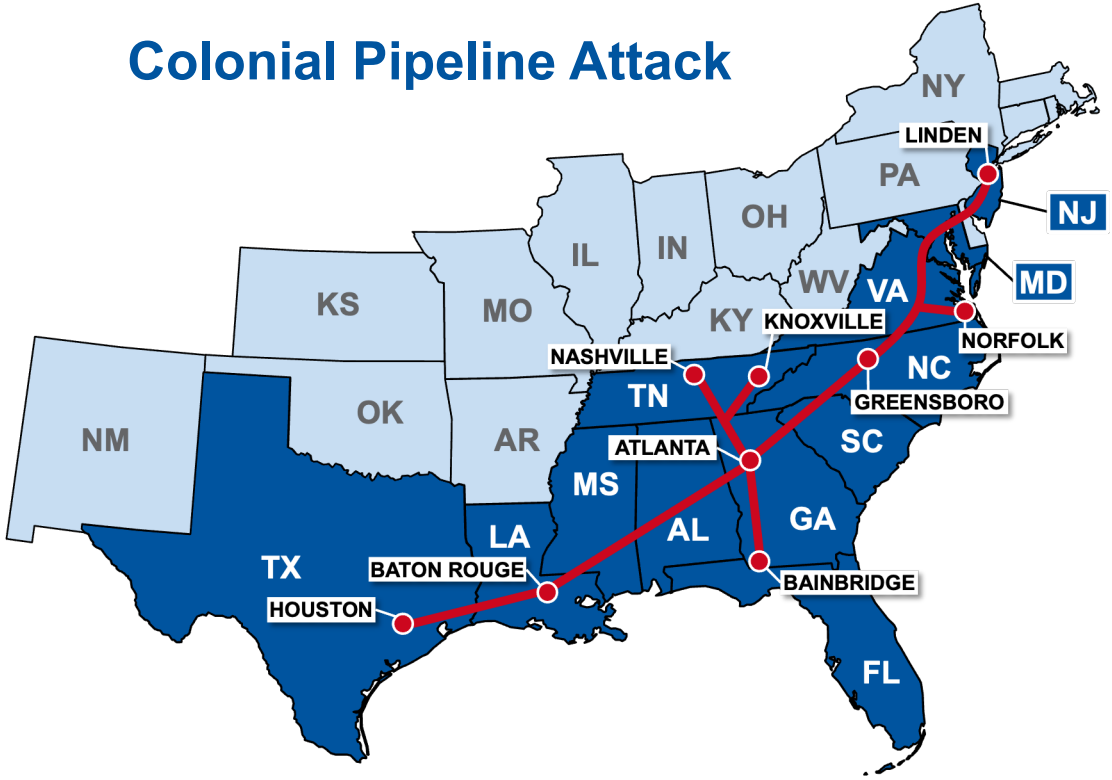
What is Ransomware and why do we care?

Ransomware 2nd biggest threat [ENISA 2024]

What is Ransomware and why do we care?

Ransomware 2nd biggest threat [ENISA 2024]

Colonial Pipeline Attack



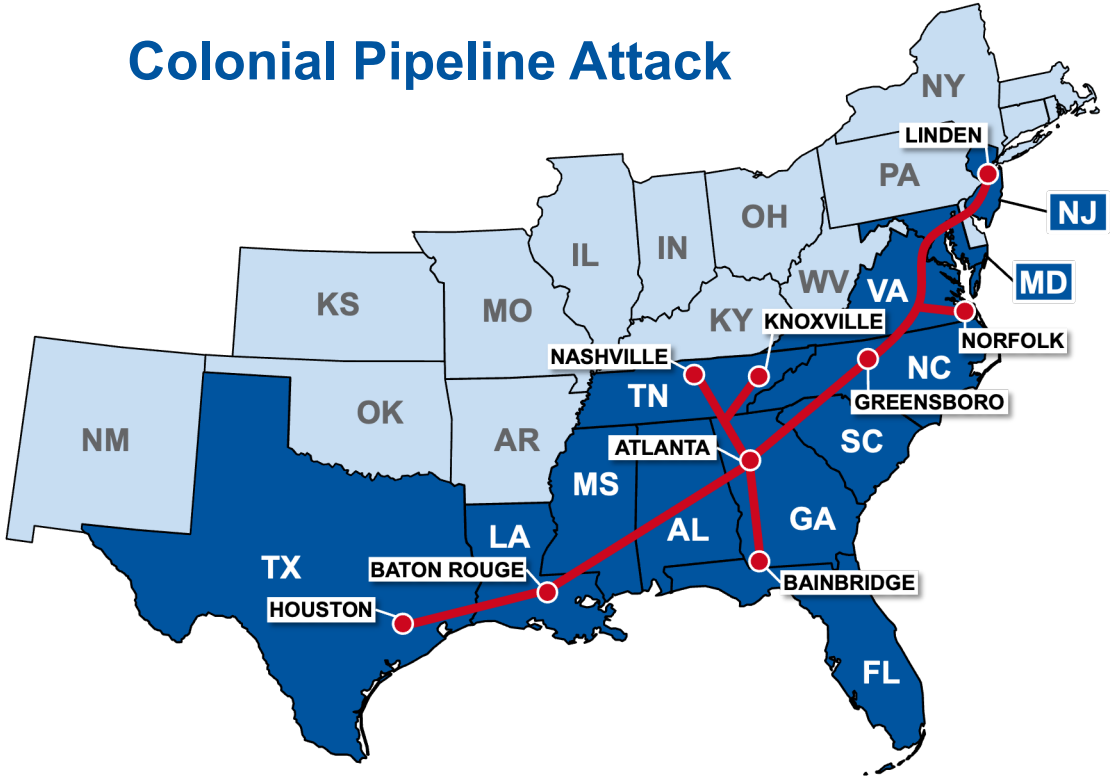
What is Ransomware and why do we care?

Ransomware 2nd biggest threat [ENISA 2024]

Cryptographic ransomware

- Encrypts user's files to prevent access
- Ransom demand to unlock files

Colonial Pipeline Attack

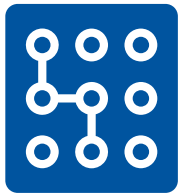


How does Ransomware Detection Work?

Cryptographic ransomware requires disk access

How does Ransomware Detection Work?

Cryptographic ransomware <u>requires</u> disk access

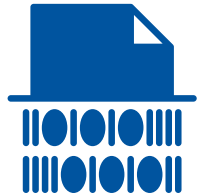


Access Patterns

- overwrite & encrypt
- read, write/encrypt, delete
- Read, encrypt, overwrite

File types

- Reads/Writes specific files (PDF, PNG, etc.)



Data Entropy

- Most files have lower entropy
- Encryption creates **<u>high</u>**-entropy data

Is there significant Monitoring Overhead?

- **Behavior-based detection has been around since 2016** **[Continella et al. 2016, Kharraz et al. 2016]**

Is there significant Monitoring Overhead?

- **Behavior-based detection has been around since 2016** **[Continella et al. 2016, Kharraz et al. 2016]**
- **Backing up files upon writing may introduce overhead of up to 3.8x** **[Continella et al. 2016]**
  - Real-world measurements found that overhead was not noticeable

Is there significant Monitoring Overhead?

- **Behavior-based detection has been around since 2016** [Continella et al. 2016, Kharraz et al. 2016]
- **Backing up files upon writing may introduce overhead of up to 3.8x** [Continella et al. 2016]
  - Real-world measurements found that overhead was not noticeable
- **Experiments were done on HDDs**
  - high latency, <100 IOPS
  - SSDs have up to 1.5mio IOPS

Is there significant Monitoring Overhead?

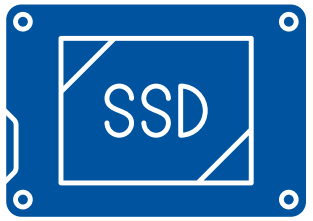
- **Behavior-based detection has been around since 2016** **[Continella et al. 2016, Kharraz et al. 2016]**
- **Backing up files upon writing may introduce overhead of up to 3.8x** **[Continella et al. 2016]**
  - Real-world measurements found that overhead was not noticeable
- **Experiments were done on HDDs**
  - high latency, <100 IOPS
  - SSDs have up to 1.5mio IOPS
- **Since: Research focused on improving detection and overhead of ML approaches**

  [Hirano et al. 2017, Shaukat and Ribeiro 2018, Jethva et al. 2020, Ahmed et al. 2021, Ayub et al. 2023]

Is there significant Monitoring Overhead?

- **Behavior-based detection has been around since 2016** [Continella et al. 2016, Kharraz et al. 2016]
- **Backing up files upon writing may introduce overhead of up to 3.8x** [Continella et al. 2016]
  - Real-world measurements found that overhead was not noticeable
- **Experiments were done on HDDs**
  - high latency, <100 IOPS
  - SSDs have up to 1.5mio IOPS
- **Since: Research focused on improving detection and overhead of ML approaches**

  [Hirano et al. 2017, Shaukat and Ribeiro 2018, Jethva et al. 2020, Ahmed et al. 2021, Ayub et al. 2023]



SATA SSD (sustained I/O) → **25%**

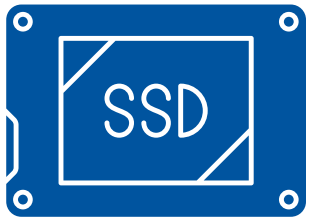
530 MB/s

Read/Write speed → **130 MB/s**

Is there significant Monitoring Overhead?

- **Behavior-based detection has been around since 2016** [Continella et al. 2016, Kharraz et al. 2016]
- **Backing up files upon writing may introduce overhead of up to 3.8x** [Continella et al. 2016]
  - Real-world measurements found that overhead was not noticeable
- **Experiments were done on HDDs**
  - high latency, <100 IOPS
  - SSDs have up to 1.5mio IOPS
- **Since: Research focused on improving detection and overhead of ML approaches**

  [Hirano et al. 2017, Shaukat and Ribeiro 2018, Jethva et al. 2020, Ahmed et al. 2021, Ayub et al. 2023]



SATA SSD (sustained I/O) → **25%**

530 MB/s

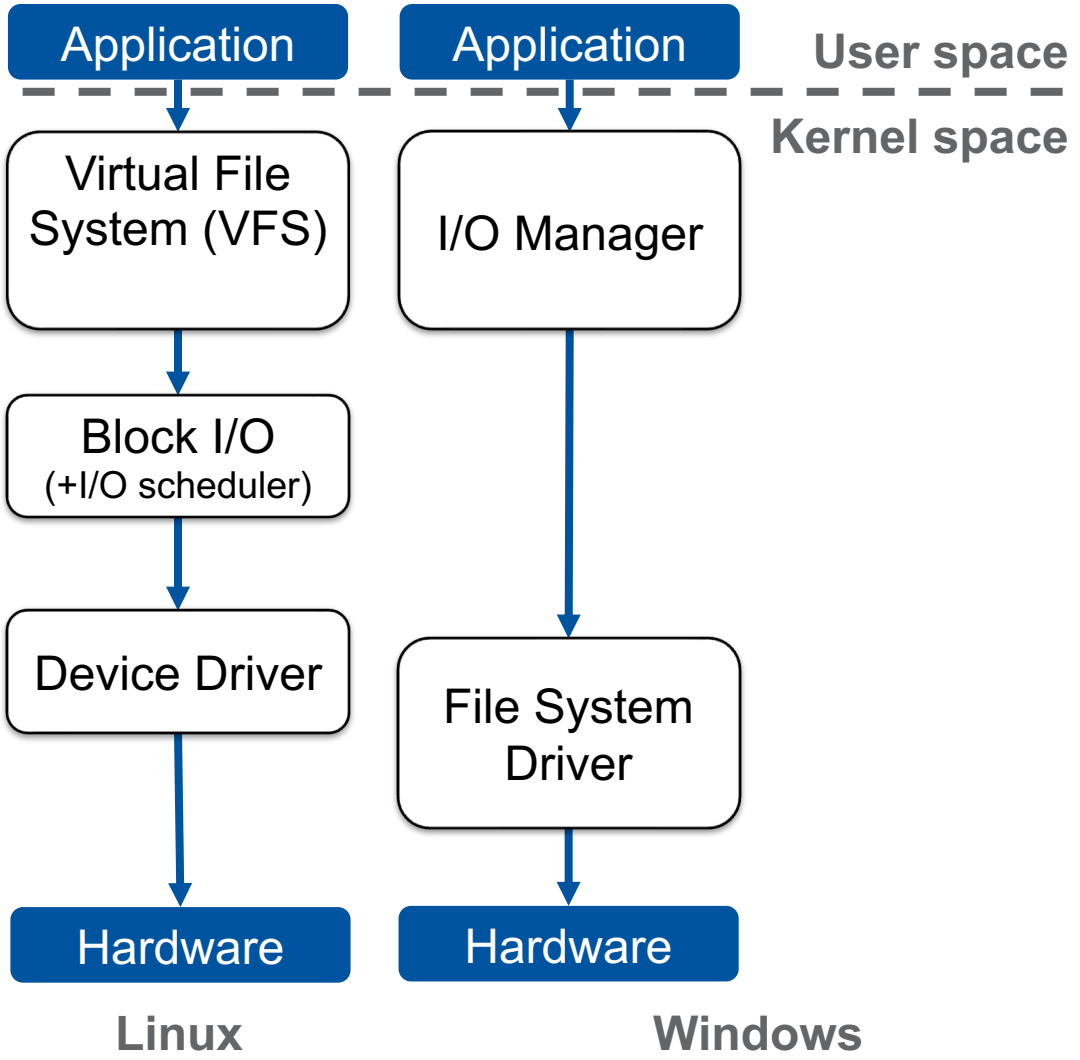
Read/Write speed →

130 MB/s

The impact of individual features on I/O overhead is unknown!

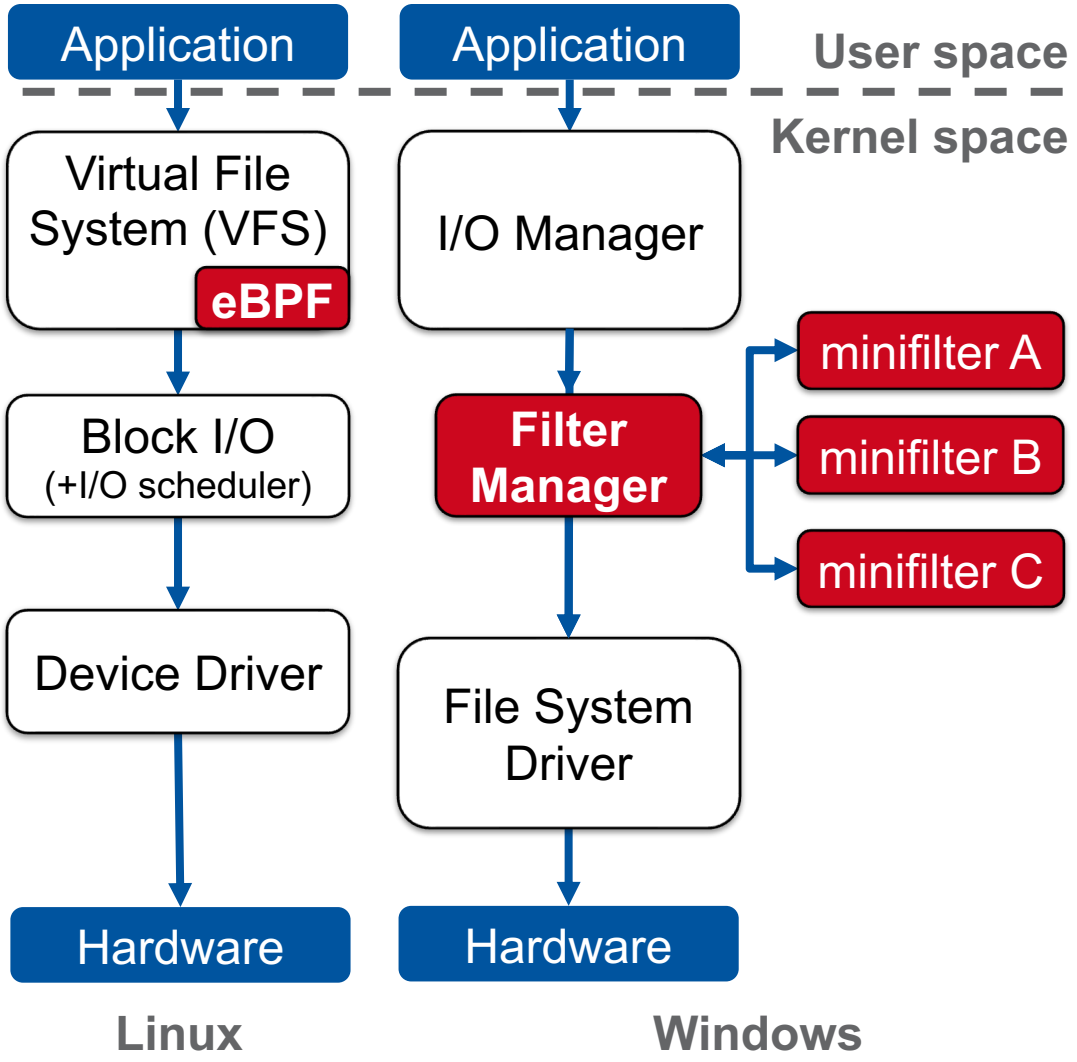
Quick Detour – Inner Workings of the I/O-Stack

- **Applications use OS functions to access the file system**
- **Hardware file layout abstracted through the file system**



Quick Detour – Inner Workings of the I/O-Stack

- **Applications use OS functions to access the file system**
- **Hardware file layout abstracted through the file system**



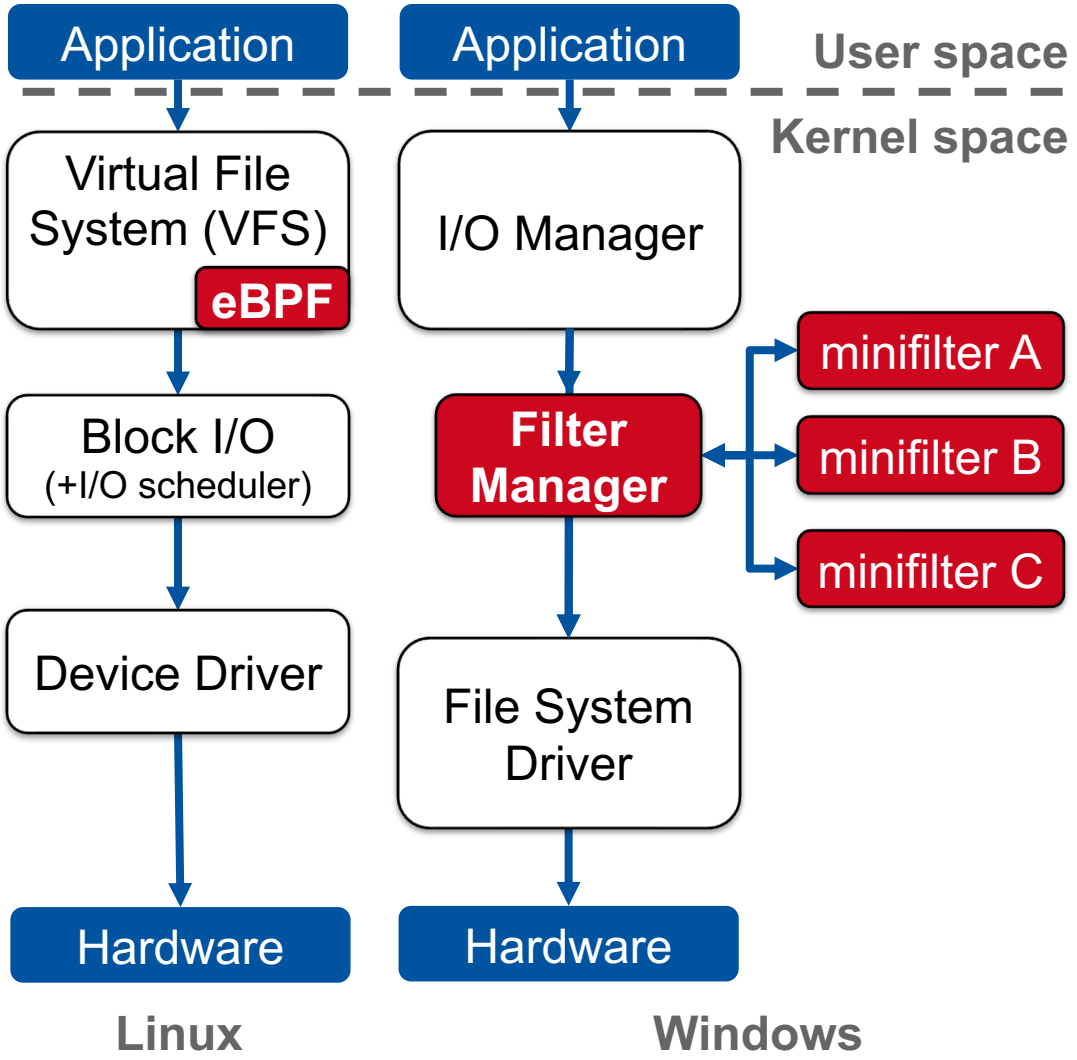
Quick Detour – Inner Workings of the I/O-Stack

- **Applications use OS functions to access the file system**
- **Hardware file layout abstracted through the file system**

Work on improving filesystems and I/O stacks is ongoing

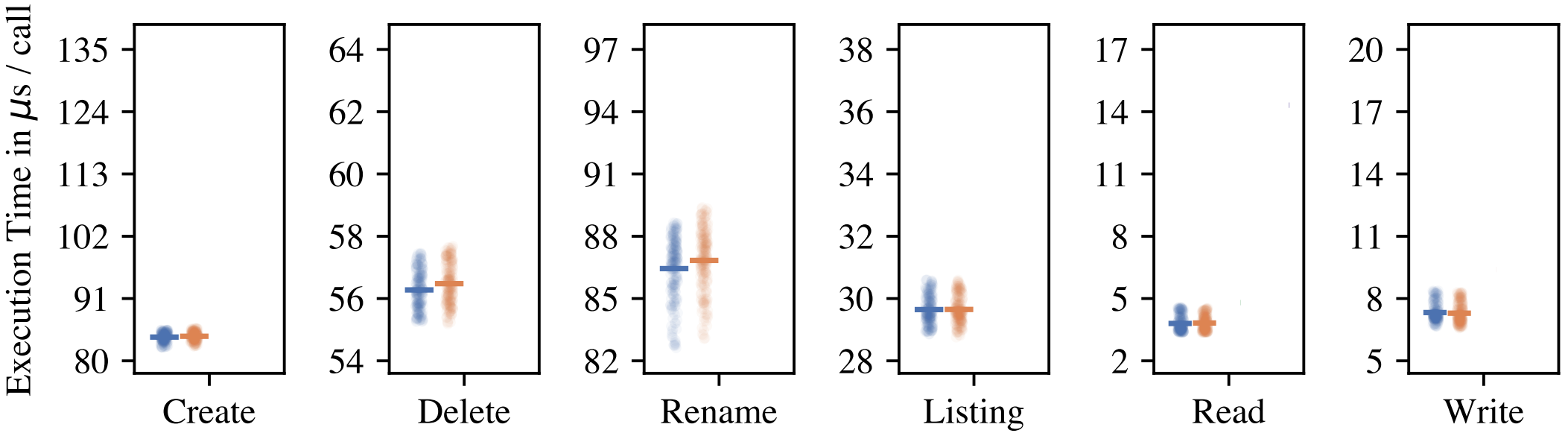
Microsoft FastIO

Microsoft DirectStorage 2020



I/O Overhead Measurements on Windows 11

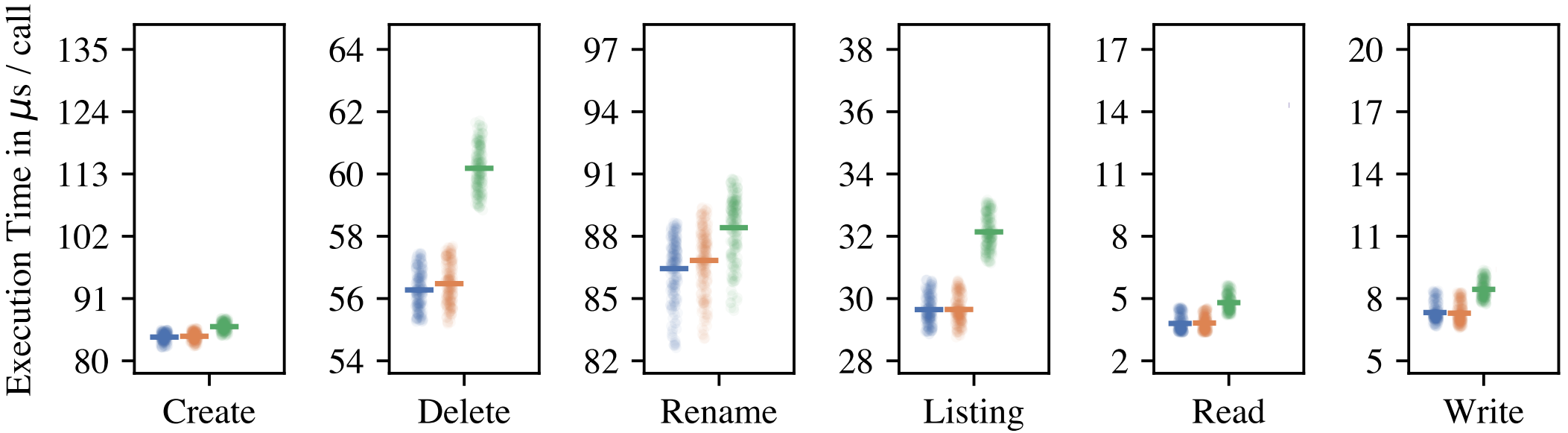
Hardware: Intel i7 4770,
16GB RAM, 500GB SATA SSD



1. No driver
2. Monitoring, but no feature is recorded

I/O Overhead Measurements on Windows 11

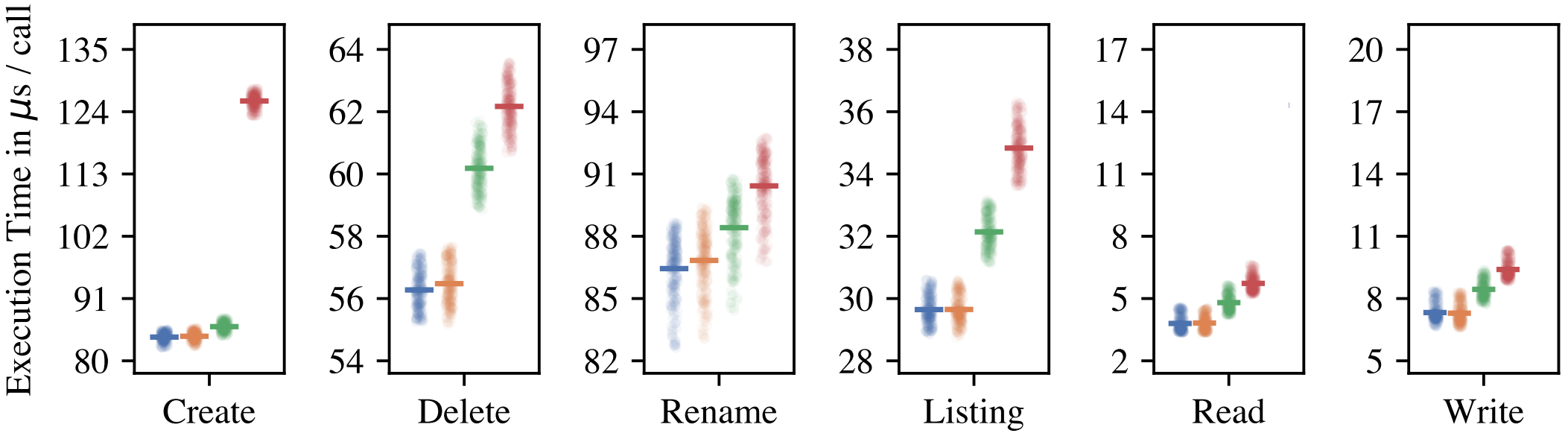
Hardware: Intel i7 4770,
16GB RAM, 500GB SATA SSD



1. No driver
2. Monitoring, but no feature is recorded
3. Record call, PID, arguments

I/O Overhead Measurements on Windows 11

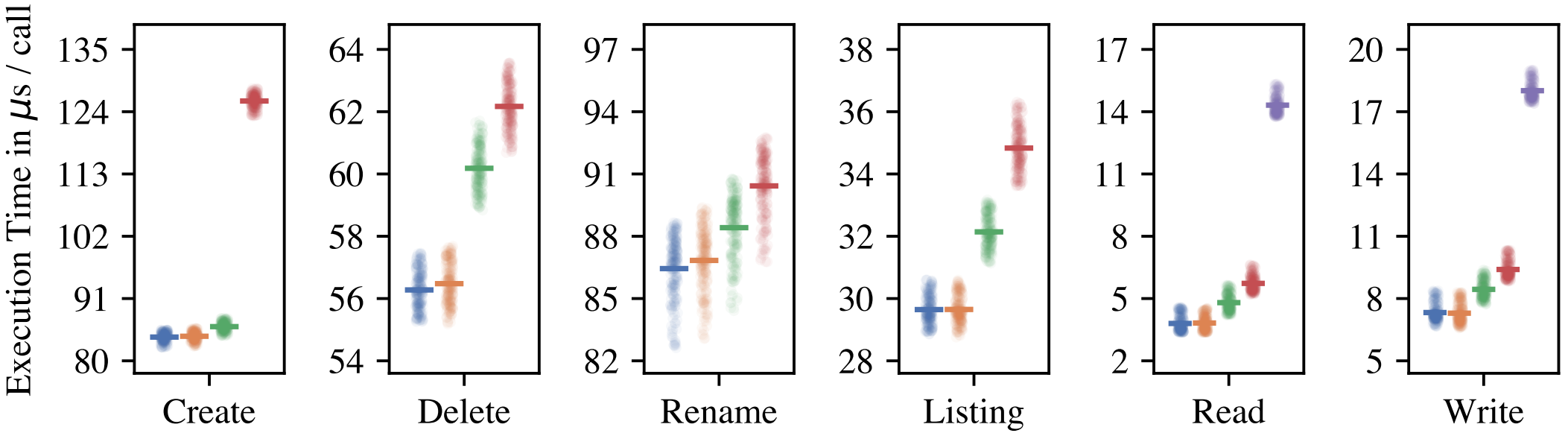
Hardware: Intel i7 4770,
16GB RAM, 500GB SATA SSD



1. No driver
2. Monitoring, but no feature is recorded
3. Record call, PID, arguments
4. Record call, PID, arguments, resolve filename

I/O Overhead Measurements on Windows 11

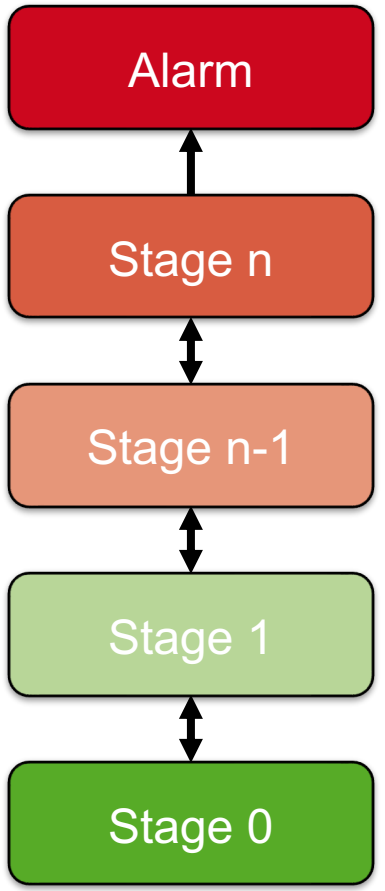
Hardware: Intel i7 4770,
16GB RAM, 500GB SATA SSD



1. No driver
2. Monitoring, but no feature is recorded
3. Record call, PID, arguments
4. Record call, PID, arguments, resolve filename
5. Record call, PID, arguments, resolve filename, calculate entropy (read/write)

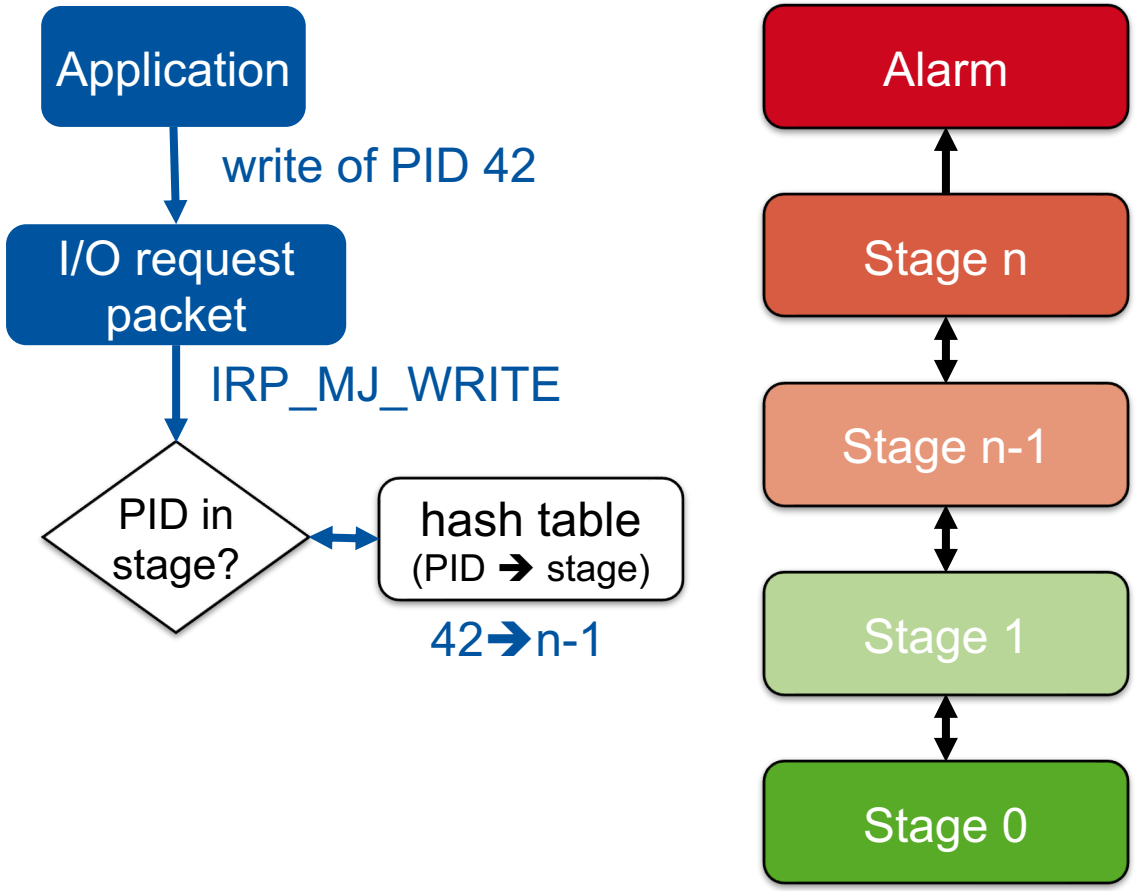
Multi-Staged Ransomware Detection (MS-IDS)

- **Can we intelligently adjust real-time monitoring on a per-process level?**
  - Most processes are benign!
  - Why not monitor only what is required?
- **Add/remove features according to the process's behavior**



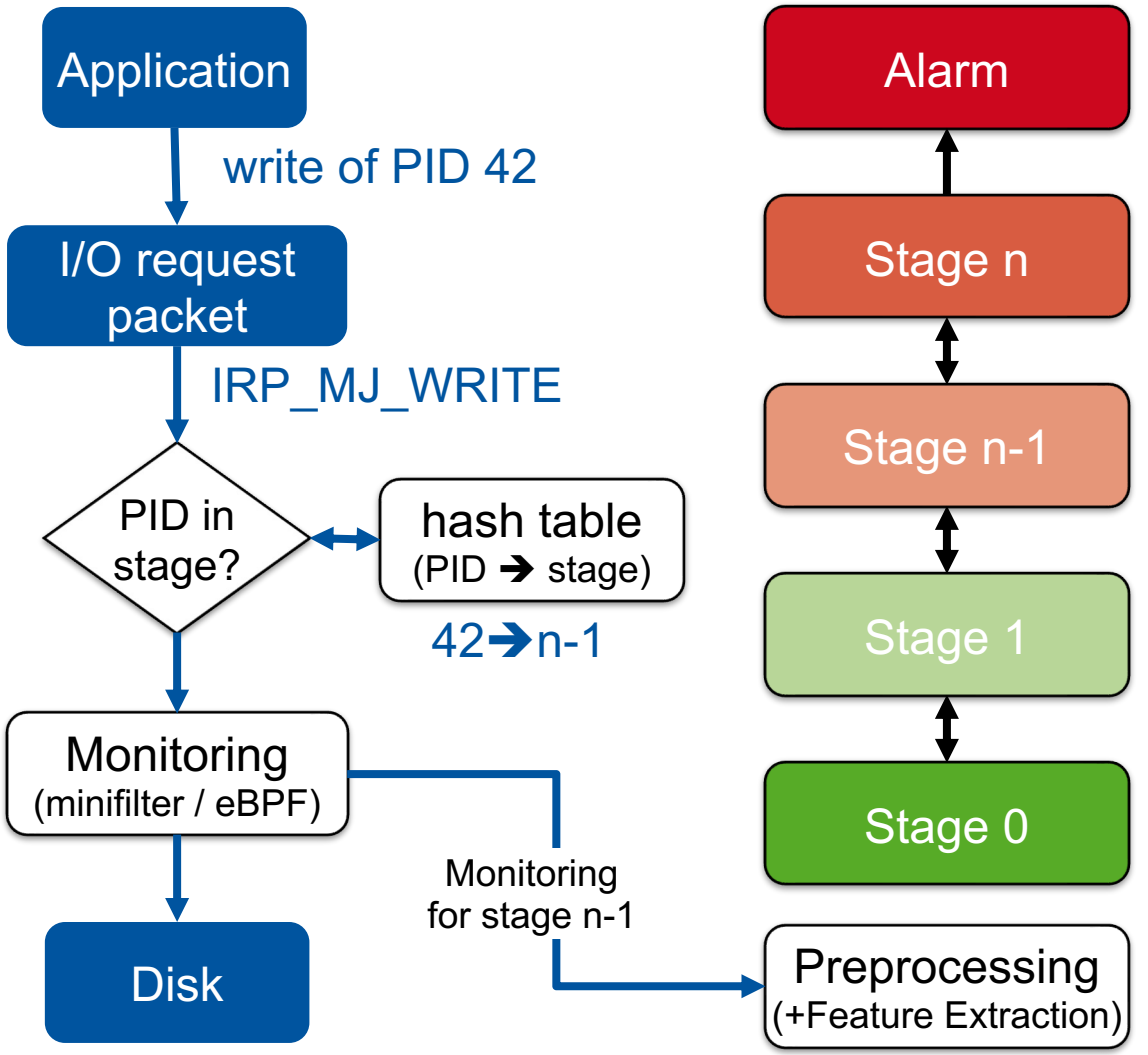
Multi-Staged Ransomware Detection (MS-IDS)

- **Can we intelligently adjust real-time monitoring on a per-process level?**
  - Most processes are benign!
  - Why not monitor only what is required?
- **Add/remove features according to the process's behavior**



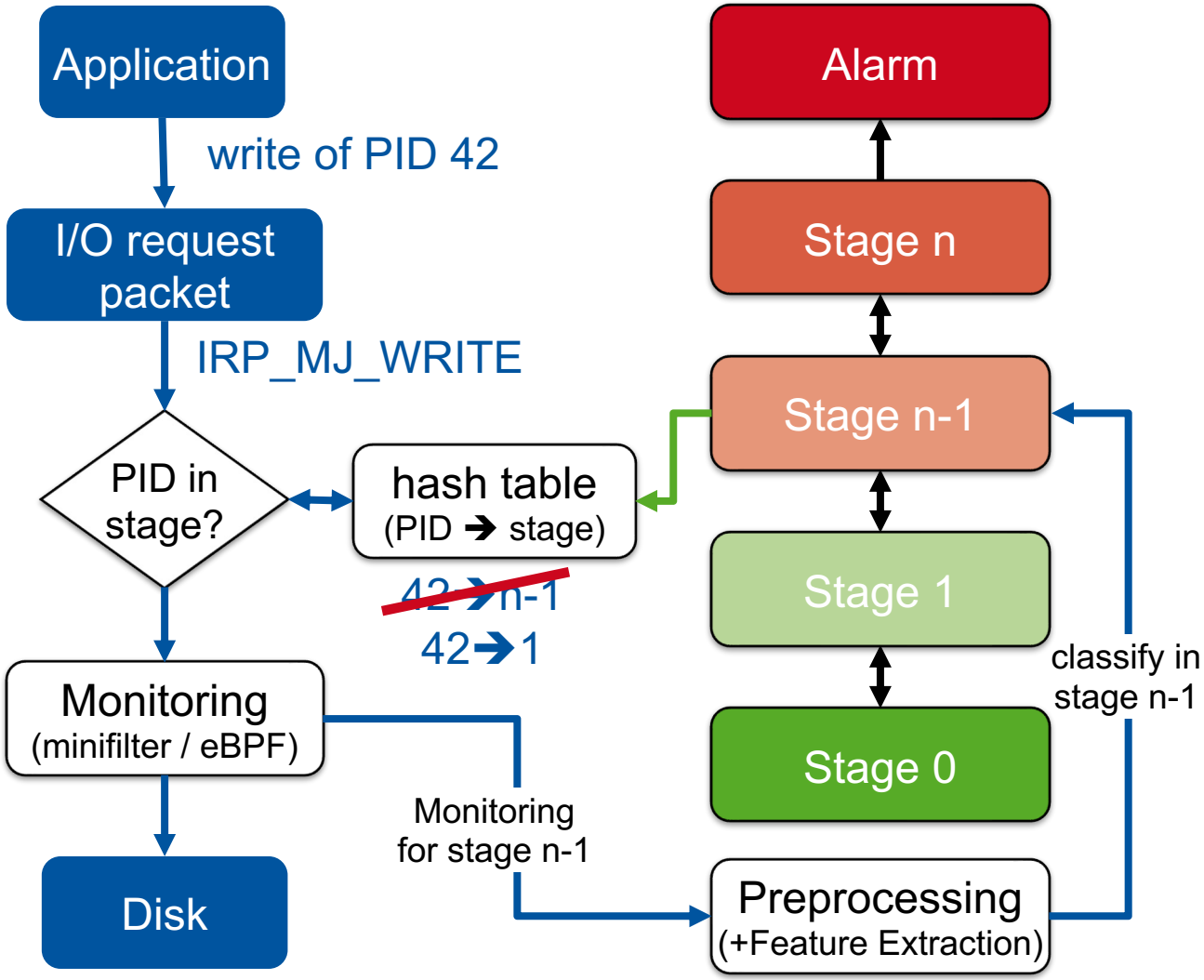
Multi-Staged Ransomware Detection (MS-IDS)

- **Can we intelligently adjust real-time monitoring on a per-process level?**
  - Most processes are benign!
  - Why not monitor only what is required?
- **Add/remove features according to the process's behavior**



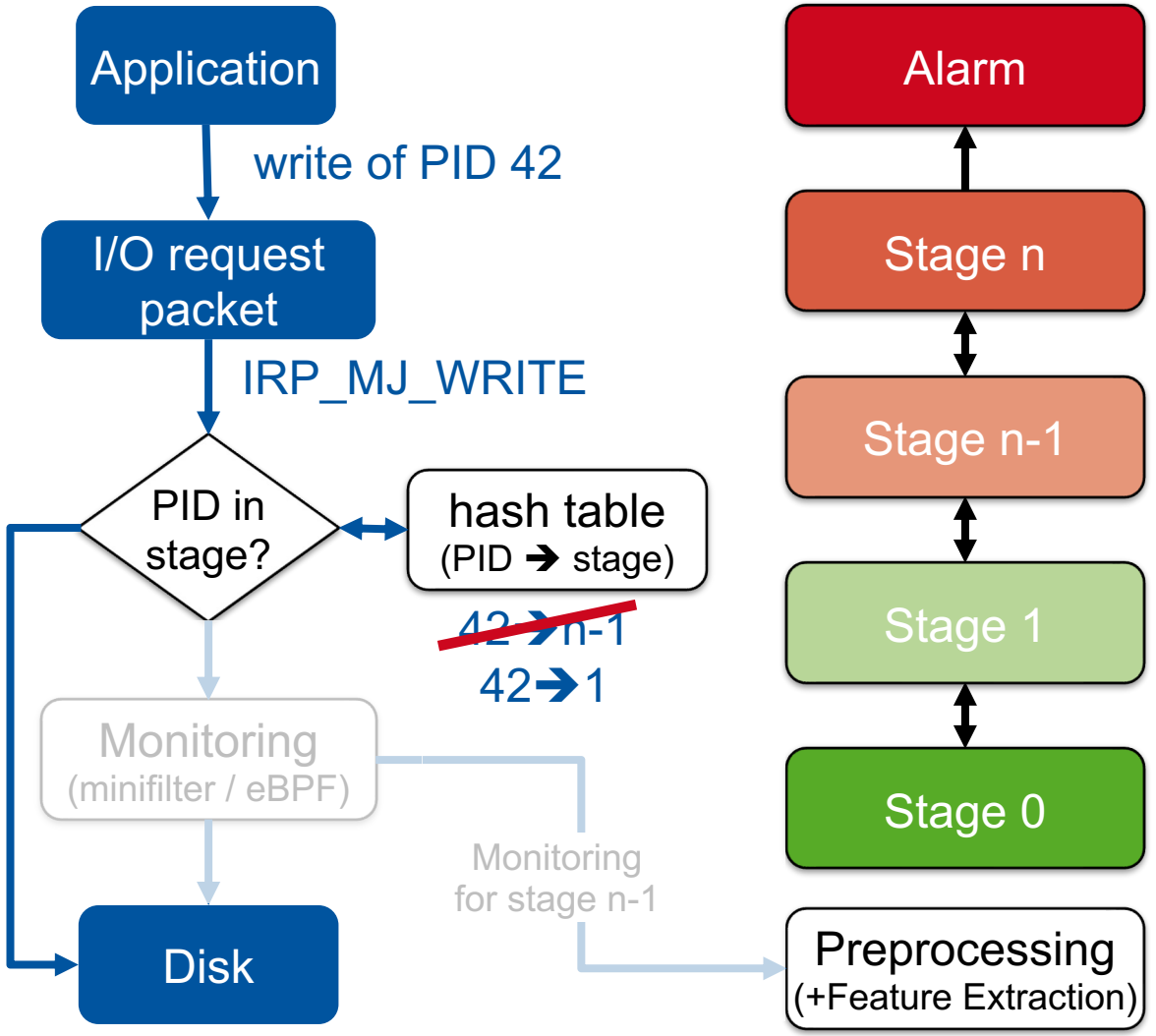
Multi-Staged Ransomware Detection (MS-IDS)

- **Can we intelligently adjust real-time monitoring on a per-process level?**
  - Most processes are benign!
  - Why not monitor only what is required?
- **Add/remove features according to the process's behavior**



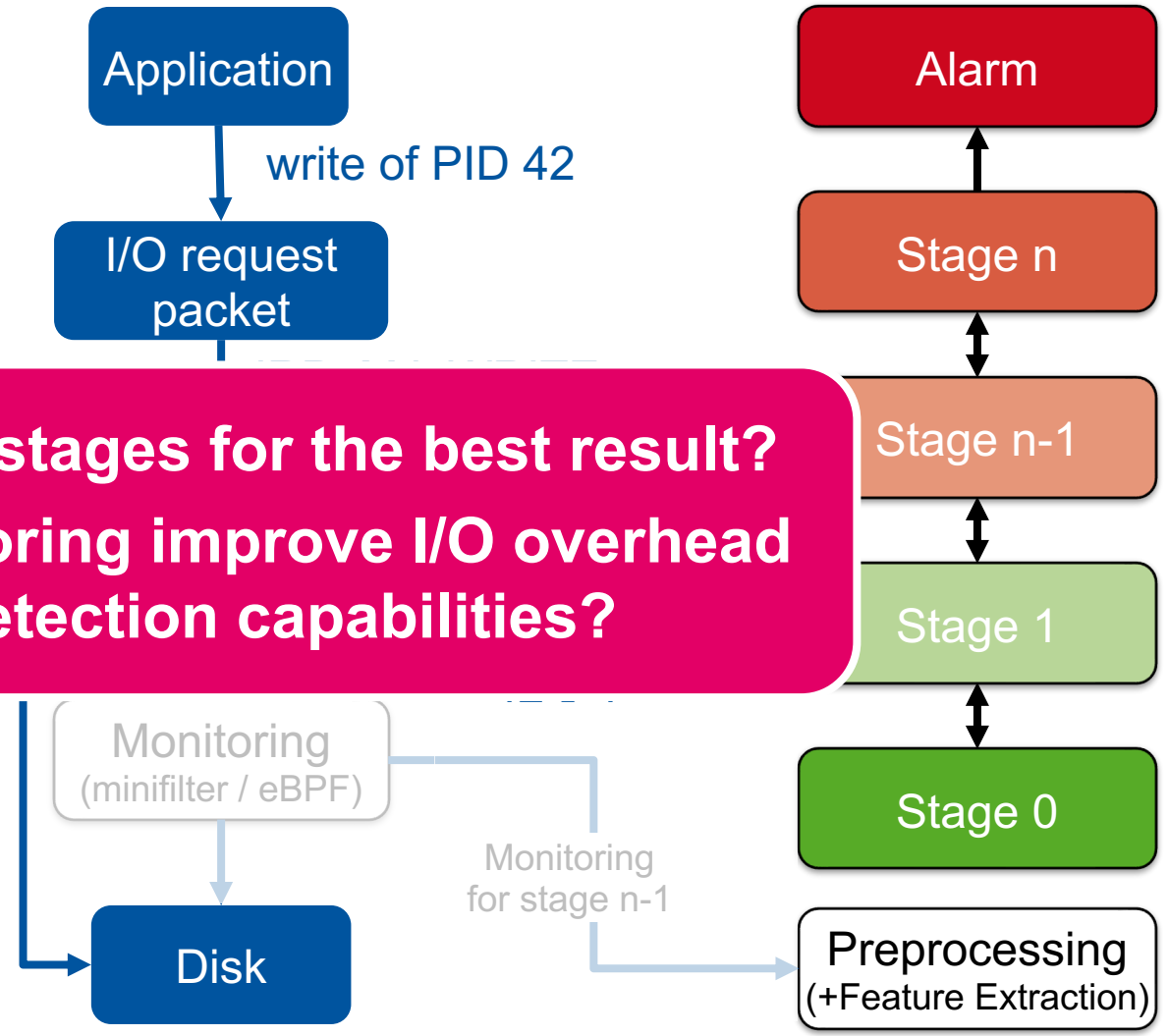
Multi-Staged Ransomware Detection (MS-IDS)

- **Can we intelligently adjust real-time monitoring on a per-process level?**
  - Most processes are benign!
  - Why not monitor only what is required?
- **Add/remove features according to the process's behavior**



Multi-Staged Ransomware Detection (MS-IDS)

- **Can we intelligently adjust real-time monitoring on a per-process level?**
  - Most processes are benign!
  - Why not monitor only what is required?
- **Add/remove fe… process's beh…**



- **How to connect the stages for the best result?**
- **Can dynamic monitoring improve I/O overhead while maintaining detection capabilities?**

Stage Designs Considerations

1. **Classification Performance**
   - Balance precision/recall per stage individually
   - Low stages require high recall, but **not** high precision

2. **Overhead / Classification Frequency**
   - Frequency influences overhead
   - Only recorded calls can trigger classification

3. **Detection Latency**
   - More stages ➔ higher latency to detect attacks
     - Minimize “shortest path to alarm”

Stage Designs Considerations

Paper: Details on stage-transitions and features per stage

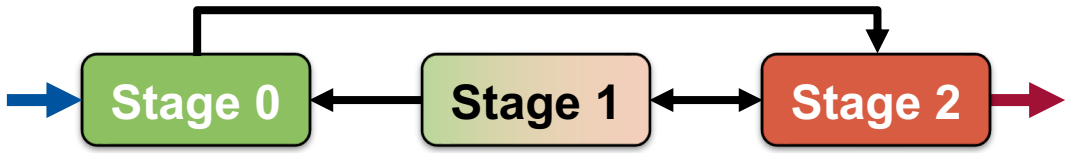
1. **Classification Performance**
   - Balance precision/recall per stage individually
   - Low stages require high recall, but **not** high precision

2. **Overhead / Classification Frequency**
   - Frequency influences overhead
   - Only recorded calls can trigger classification

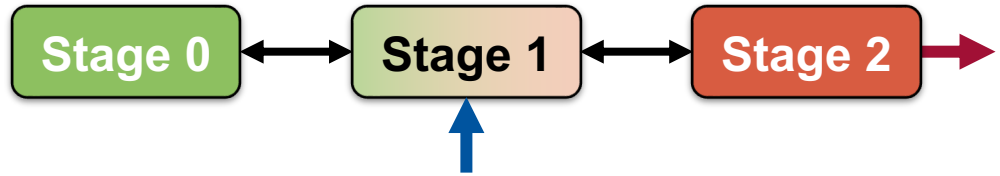
3. **Detection Latency**
   - More stages ➔ higher latency to detect attacks
     - Minimize “shortest path to alarm”

Naïve Monitoring (N)

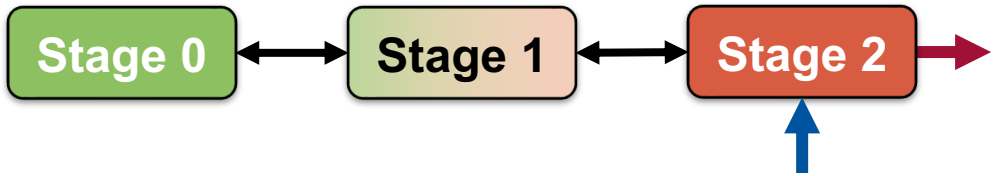
Reduced Monitoring (RM)



Initial Monitoring (IM)



Suspicious Monitoring (SM)



Evaluation Results

Excludes processes with insufficient IO operations to raise false alarms

#Stages	Architecture	TN	FP	FN	TP	F1
1	Traditional	12746	10	0	91	**0.95**
3	Naïve Monitoring	2693	3	0	91	0.98
	Reduced Monitoring	2694	2	0	91	**0.99**
	Initial Monitoring	11638	9	0	91	0.95
	Suspicious Monitoring	12749	7	0	91	0.96
5	Naïve Monitoring	2696	0	0	91	**1.00**
	Reduced Monitoring	2694	2	0	91	0.99
	Initial Monitoring	11644	3	0	91	0.98
	Suspicious Monitoring	12745	11	0	91	0.94

Evaluated on ShieldFS dataset (2016)

- Real-world data of user sessions from 11 machines
- Over 380 distinct ransomware samples of 6 ransomware strains (91 contained in test set)

Evaluation Results

Excludes processes with insufficient IO operations to raise false alarms

#Stages	Architecture	TN	FP	FN	TP	F1
1	Traditional	12746	10	0	91	**0.95**
3	Naïve Monitoring	2693	3	0	91	0.98
	Reduced Monitoring	2694	2	0	91	**0.99**
	Initial Monitoring	11638	9	0	91	0.95
	Suspicious Monitoring	12749	7	0	91	0.96
5	Naïve Monitoring	2696	0	0	91	**1.00**
	Reduced Monitoring	2694	2	0	91	0.99
	Initial Monitoring	11644	3	0	91	0.98
	Suspicious Monitoring	12745	11	0	91	0.94

Even the naïve models detect all samples

Evaluated on ShieldFS dataset (2016)

- Real-world data of user sessions from 11 machines
- Over 380 distinct ransomware samples of 6 ransomware strains (91 contained in test set)

Evaluation Results

Excludes processes with insufficient IO operations to raise false alarms

#Stages	Architecture	TN	FP	FN	TP	F1
1	Traditional	12746	10	0	91	**0.95**
3	Naïve Monitoring	2693	3	0	91	0.98
	Reduced Monitoring	2694	2	0	91	**0.99**
	Initial Monitoring	11638	9	0	91	0.95
	Suspicious Monitoring	12749	7	0	91	0.96
5	Naïve Monitoring	2696	0	0	91	**1.00**
	Reduced Monitoring	2694	2	0	91	0.99
	Initial Monitoring	11644	3	0	91	0.98
	Suspicious Monitoring	12745	11	0	91	0.94

No increase in false-positives for suspicious monitoring

Evaluated on ShieldFS dataset (2016)

- Real-world data of user sessions from 11 machines
- Over 380 distinct ransomware samples of 6 ransomware strains (91 contained in test set)

Evaluation Results

#Stages	Architecture	TN	FP	FN	TP	F1
1	Traditional	12746	10	0	91	**0.95**
3	Naïve Monitoring	2693	3	0	91	0.98
	Reduced Monitoring	2694	2	0	91	**0.99**
	Initial Monitoring	11638	9	0	91	0.95
	Suspicious Monitoring	12749	7	0	91	0.96
5	Naïve Monitoring	2696	0	0	91	**1.00**
	Reduced Monitoring	2694	2	0	91	0.99
	Initial Monitoring	11644	3	0	91	0.98
	Suspicious Monitoring	12745	11	0	91	0.94

Evaluated on ShieldFS dataset (2016)

- Real-world data of user sessions from 11 machines
- Over 380 distinct ransomware samples of 6 ransomware strains (91 contained in test set)

Evaluation against novel ransomware (2020-2023)

Unmodified models with data from 2016:

- MS-IDS detects 12 out of 47 samples **(~25%)**
- Single-Stage IDS detects 12 samples

Manual adjustment of hyperparameters:

- MS-IDS detects 18 out of 47 samples **(~38%)**

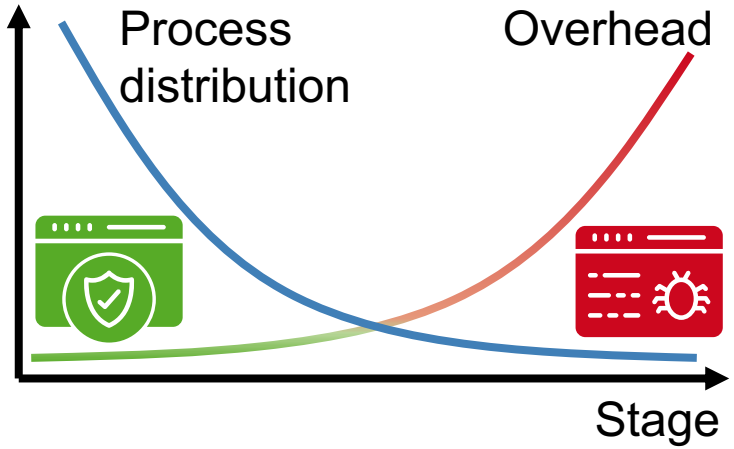
I/O Overhead Evaluation

- **Monitoring all features causes an overhead of 180%±103%**
  - Excluding entropy: still 35%±17% overhead
  - **Entropy is expensive but important**

I/O Overhead Evaluation

- **Monitoring all features causes an overhead of 180%±103%**
  - Excluding entropy: still 35%±17% overhead
  - **Entropy is expensive but important**

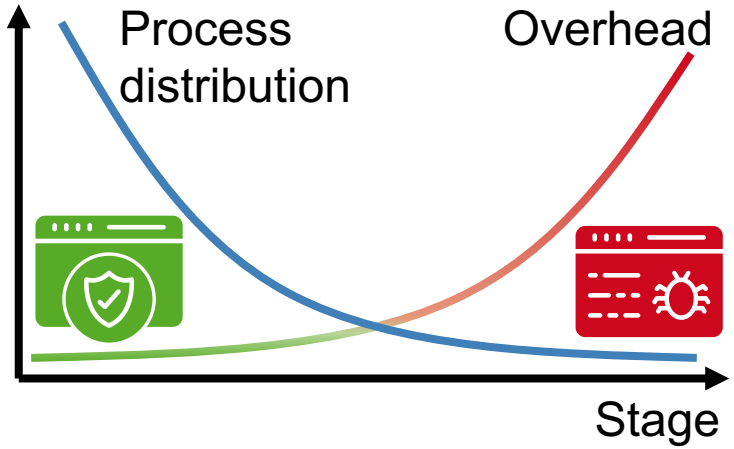
- **Using a Multi-Staged IDS:**
  - Benign processes quickly move to stages with low overhead
  - Ransomware processes move to high stages and are stopped



I/O Overhead Evaluation

Paper: Details on detection latency and I/O distribution across stages

- **Monitoring all features causes an overhead of 180%±103%**
  - Excluding entropy: still 35%±17% overhead
  - **Entropy is expensive but important**



- **Using a Multi-Staged IDS:**
  - Benign processes quickly move to stages with low overhead
  - Ransomware processes move to high stages and are stopped

- **An MS-IDS starting with <u>all</u> features already decreases average overhead by 10x**
  - Other architectures reduce overhead by a further factor of 2x

Conclusion

- **Ransomware detection interferes with optimized I/O stacks**
- **Behavior monitoring negatively impacts modern storage systems**

Conclusion

- **Ransomware detection interferes with optimized I/O stacks**
- **Behavior monitoring negatively impacts modern storage systems**

- **To detect ransomware, it is not necessary to monitor everything**
- **Intelligent monitoring can adjust overhead on a per-process level**

Conclusion

- Ransomware detection interferes with optimized I/O stacks
- Behavior monitoring negatively impacts modern storage systems

- To detect ransomware, it is not necessary to monitor everything
- Intelligent monitoring can adjust overhead on a per-process level

- Detection performance is on par with a single-staged model
- Multi-Staged design significantly reduces overhead and is effective against modern ransomware

Conclusion

- Ransomware detection interferes with optimized I/O stacks
- Behavior monitoring negatively impacts modern storage systems

- To detect ransomware, it is not necessary to monitor everything
- Intelligent monitoring can adjust overhead on a per-process level

- Detection performance is on par with a single-staged model
- Multi-Staged design significantly reduces overhead and is effective against modern ransomware

Thank you for your attention

Check out our code