Logging to the Danger Zone: Race Condition Attacks and Defenses on System Audit Frameworks

Riccardo Paccagnella, Kevin Liao, Dave Tian, Adam Bates

ILLINOIS

Logs Are Useful

- 75% of incident response specialists said logs are the most valuable artifact during an investigation.[1]

[1] Carbon Black Quarterly Incident Response Threat Report April 2019

Logs Are Useful

- 75% of incident response specialists said logs are the most valuable artifact during an investigation.[1]

CAPEC-268: Audit Log Manipulation

Attack Pattern ID: 268
Abstraction: Standard

Presentation Filter: Complete

Description

The attacker injects, manipulates, deletes, or forges malicious log entries into the log file, in an attempt to mislead an audit of the log file or cover tracks of an attack.

Hackers are increasingly destroying logs to hide attacks

According to a new report, 72 percent of incident response specialists have came across hacks where attackers have destroyed logs to hide their tracks.

By Catalin Cimpanu for Zero Day | November 2, 2018 -- 16:36 GMT (09:36 PDT) | Topic: Security

[1] Carbon Black Quarterly Incident Response Threat Report April 2019

Can We Protect the Logs?

Can We Protect the Logs?

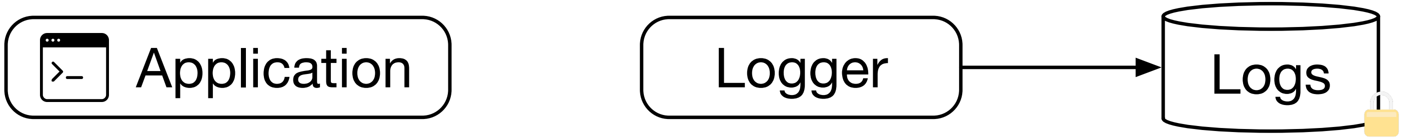
- Secure Logging!

Can We Protect the Logs?

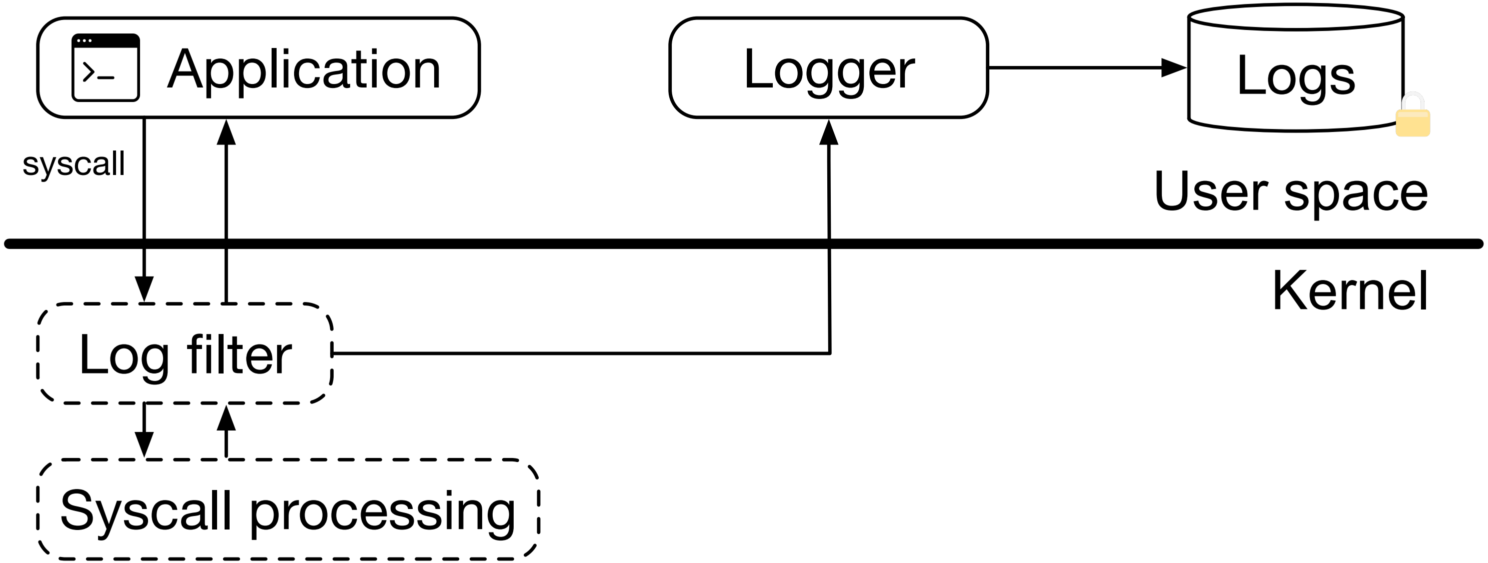
- Secure Logging!
- Logs recorded prior to full system compromise cannot be undetectably tampered with.

How Secure Logging Works

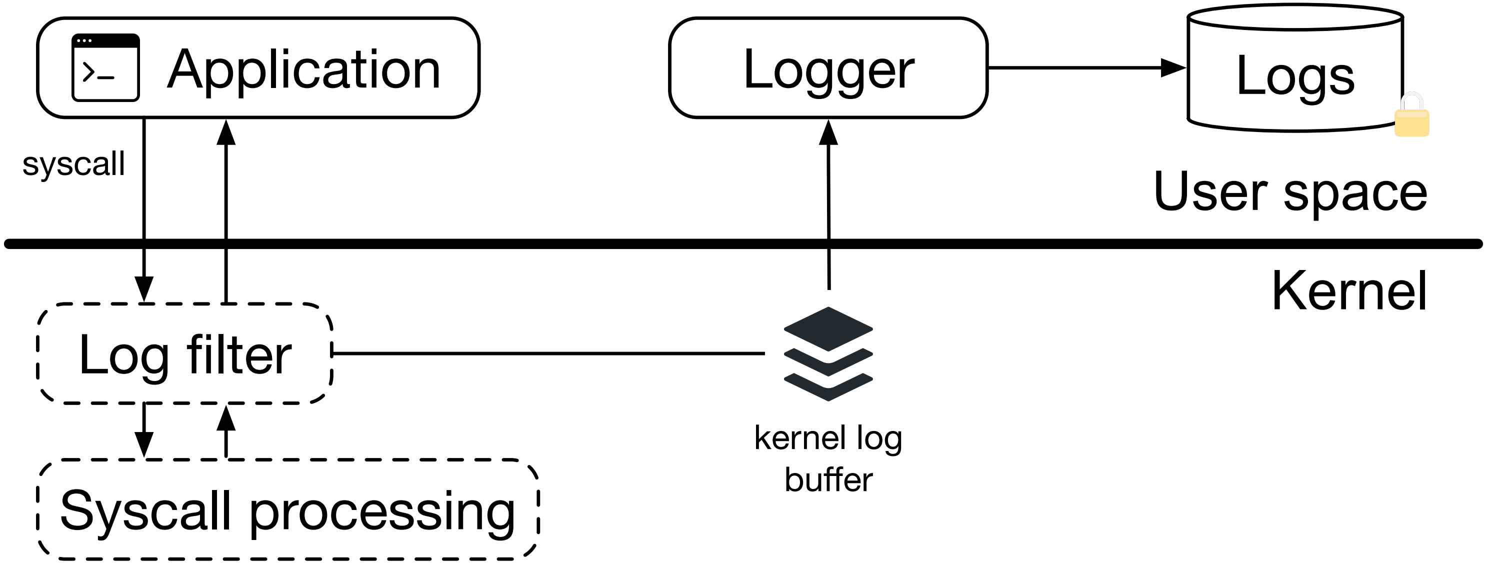
How Secure Logging Works



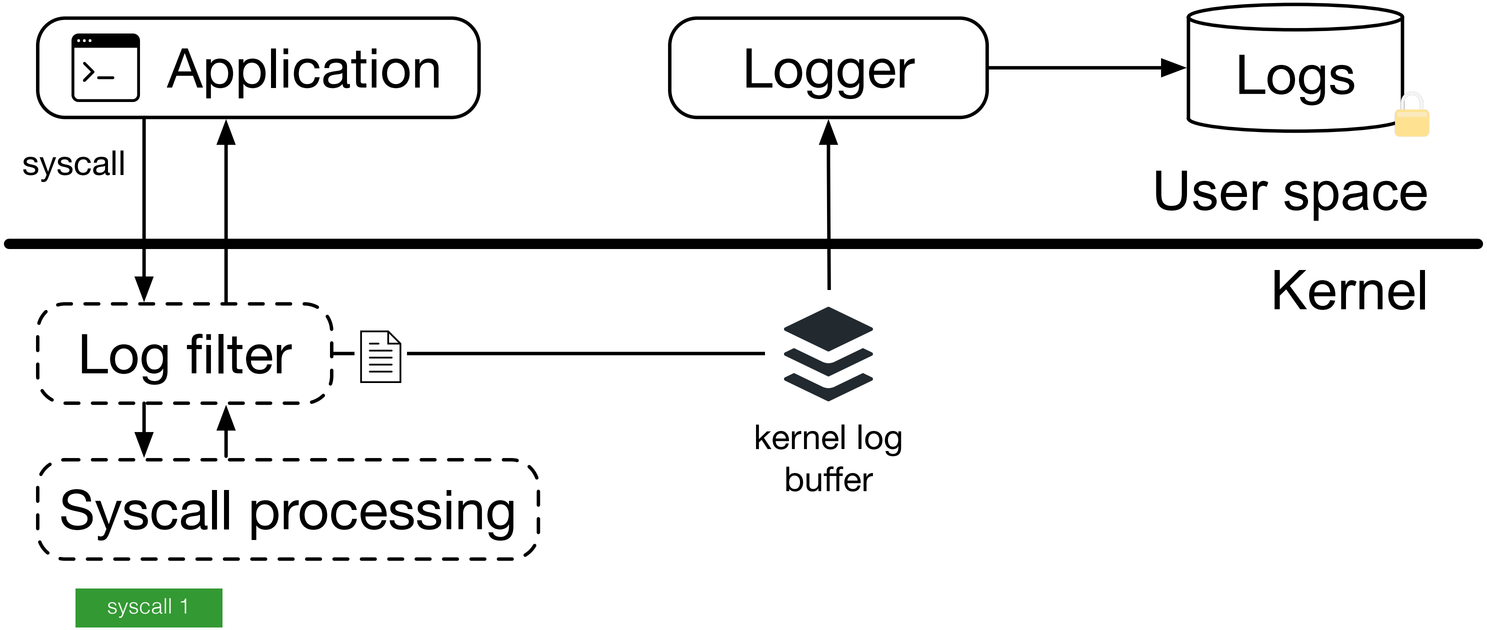
How Secure Logging Works



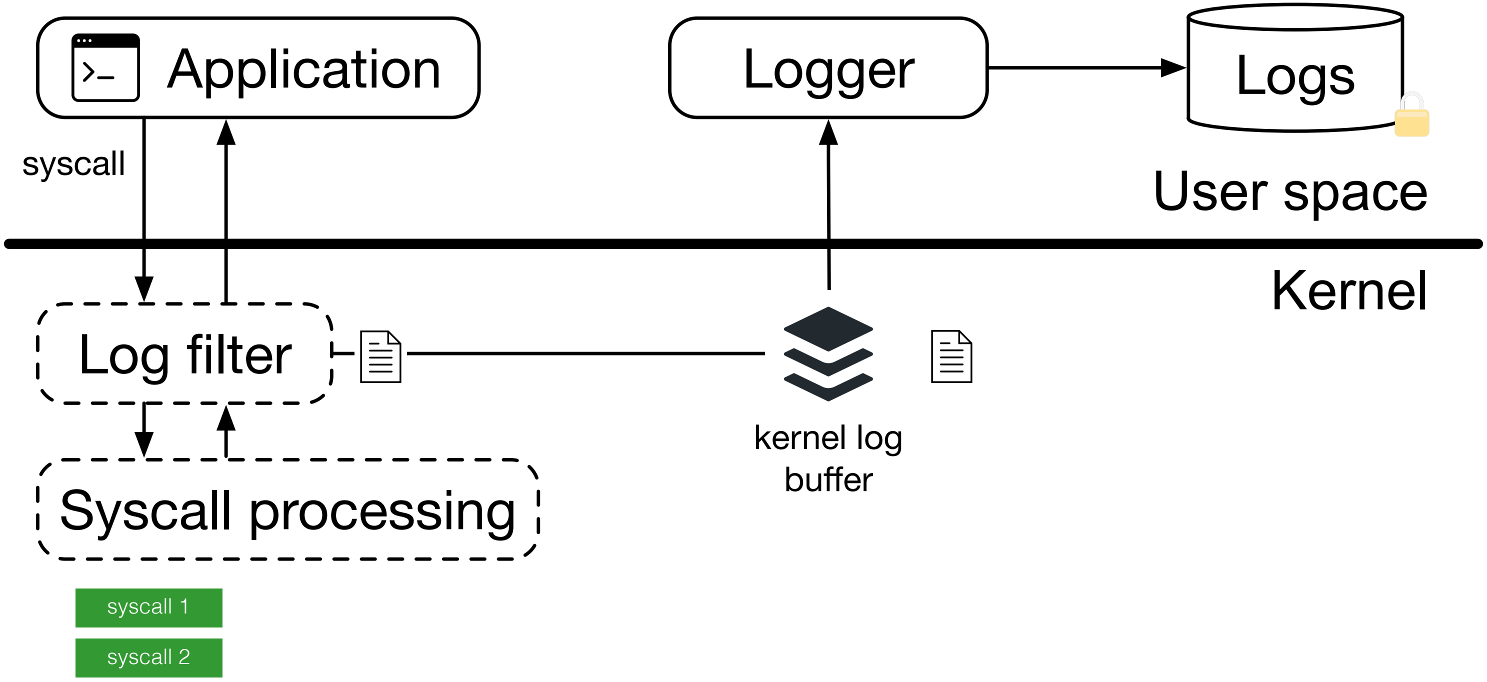
How Secure Logging Works



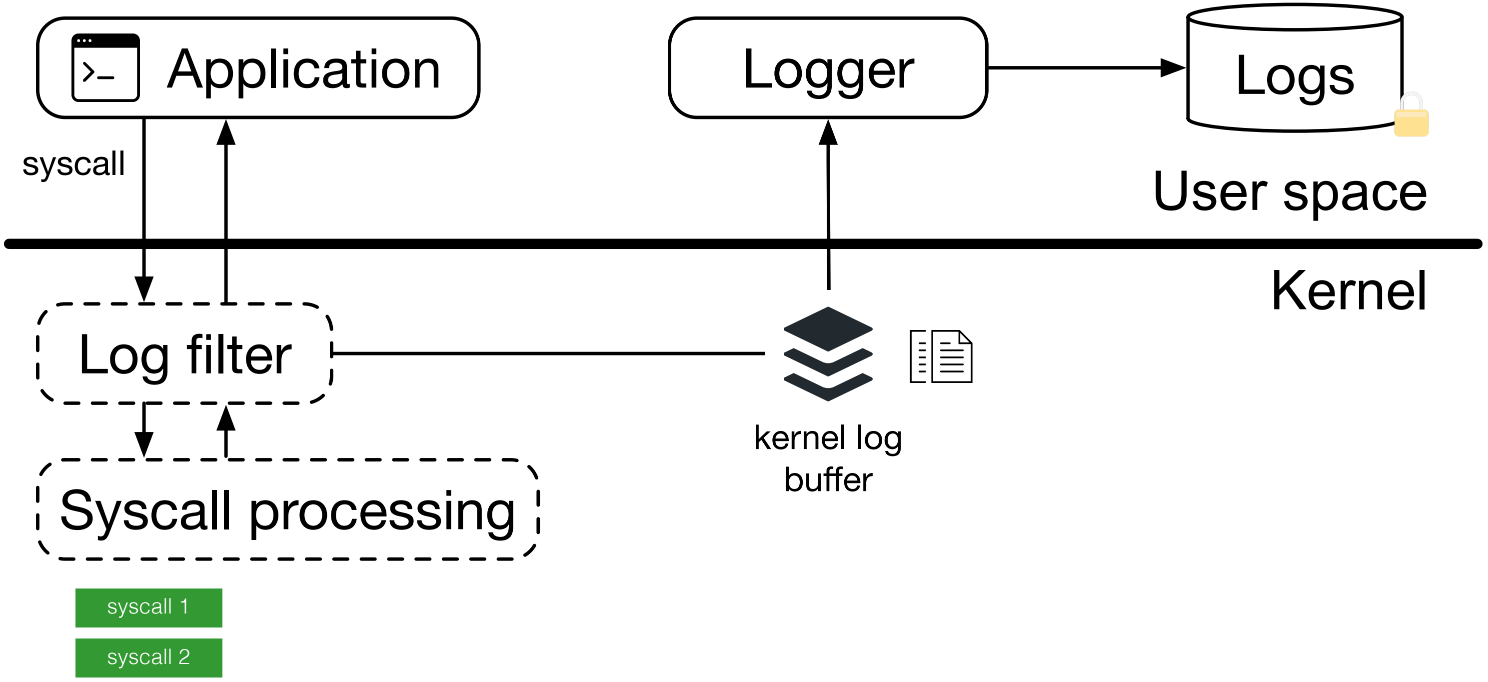
How Secure Logging Works



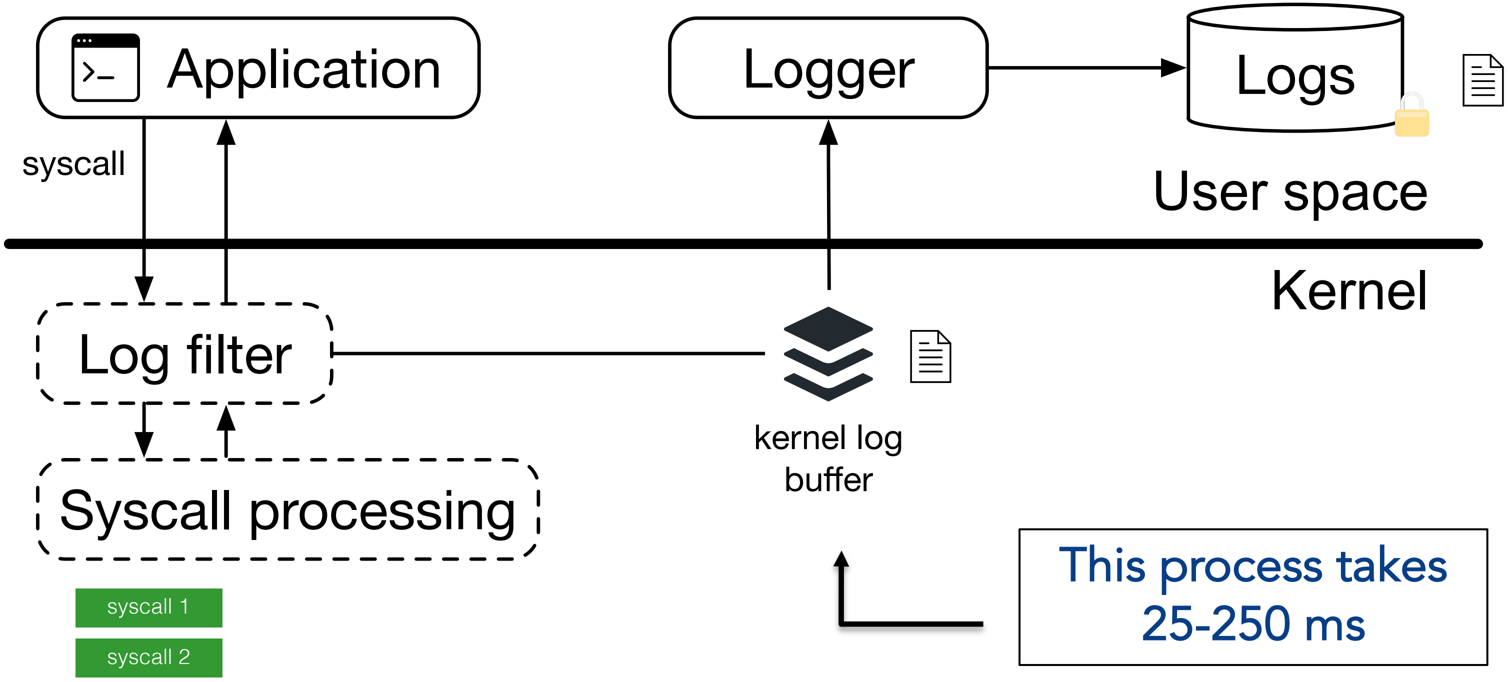
How Secure Logging Works



How Secure Logging Works

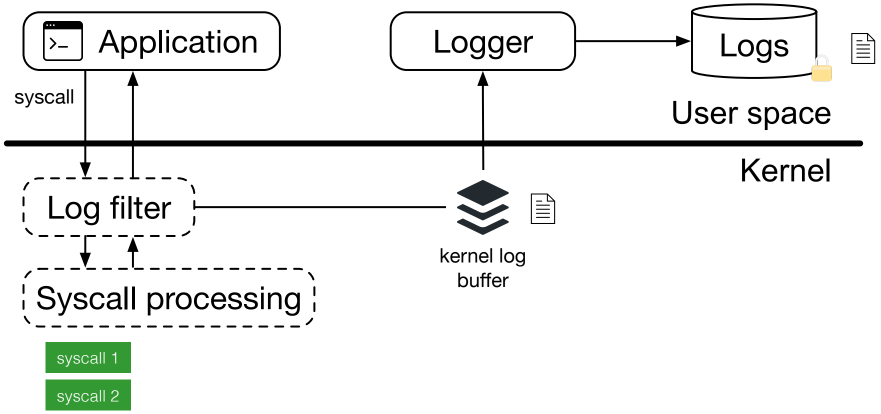


How Secure Logging Works



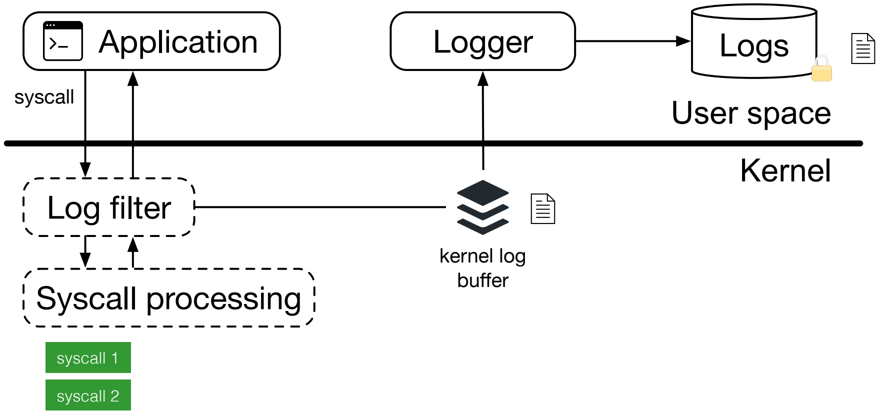
This process takes
25-250 ms

How Secure Logging Works



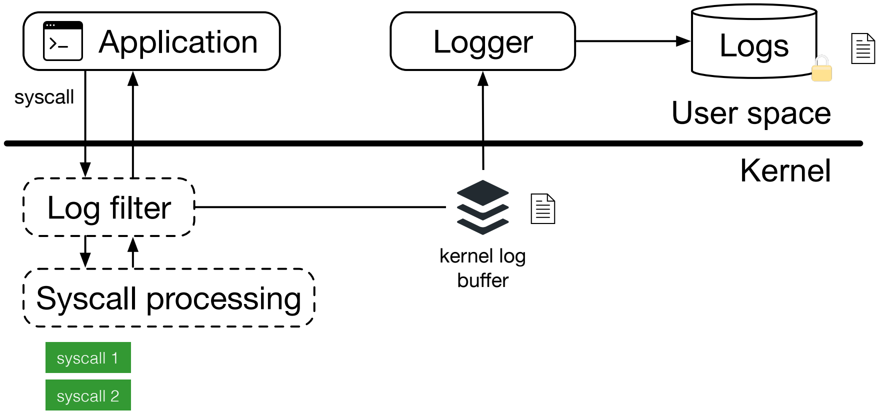
- Logging is asynchronous.

How Secure Logging Works



- Logging is asynchronous.
- It takes 25-250 ms for an event to be logged.

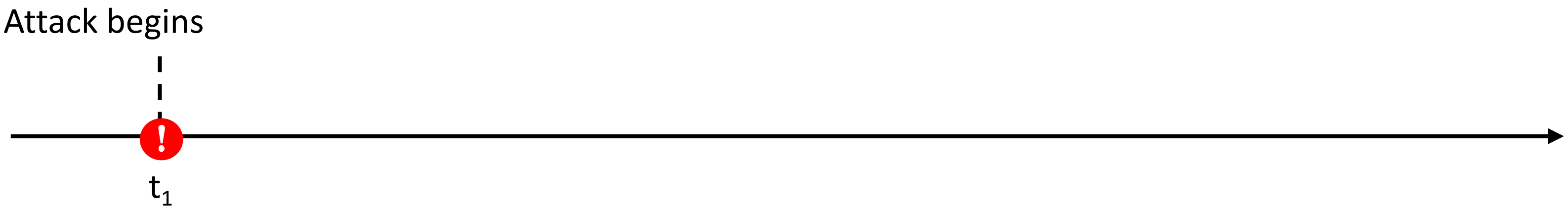
How Secure Logging Works



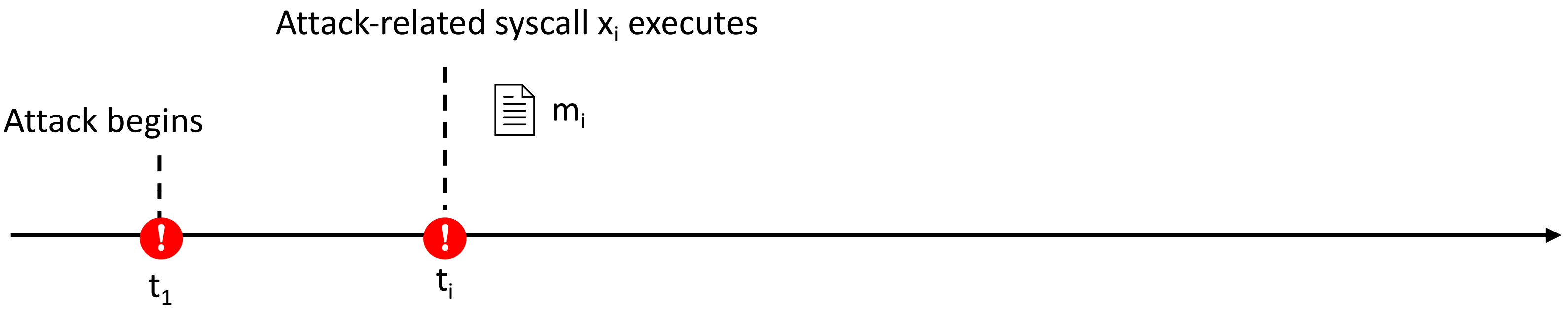
- Logging is asynchronous.
- It takes 25-250 ms for an event to be logged.

Logs are vulnerable when in the kernel log buffer!

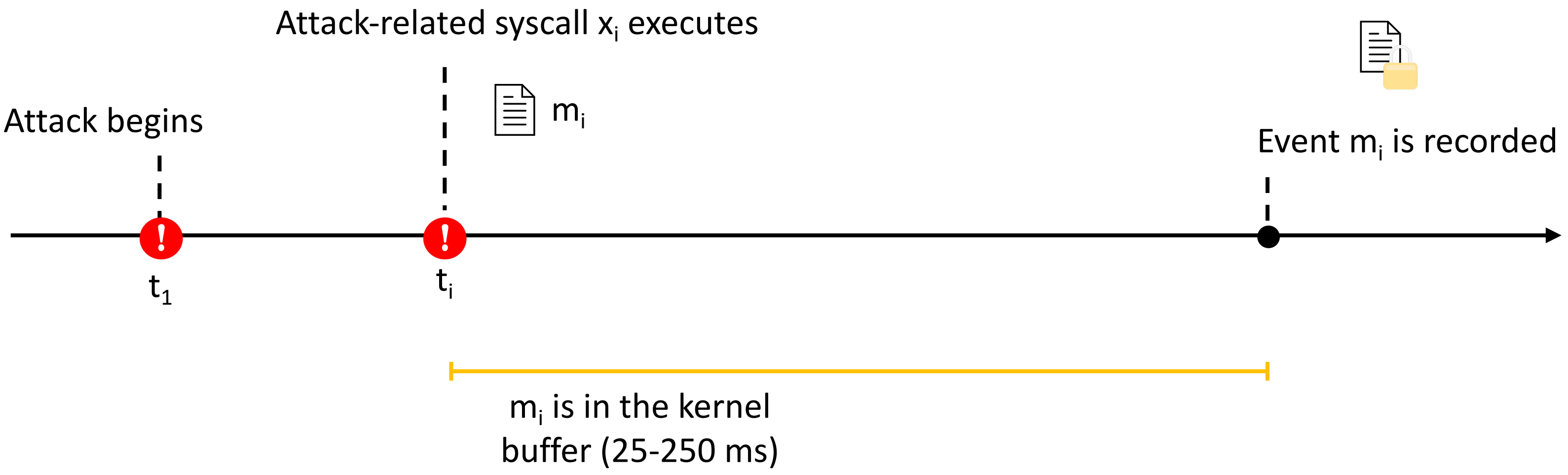
Attack Timeline



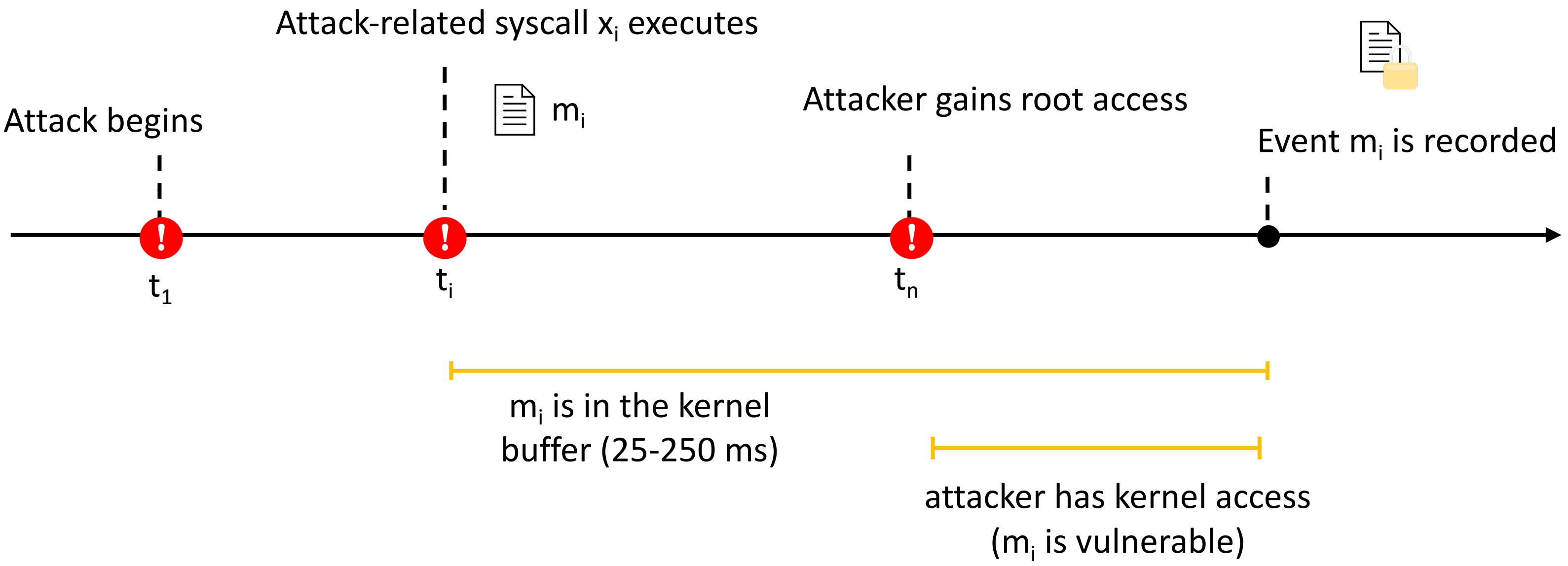
Attack Timeline



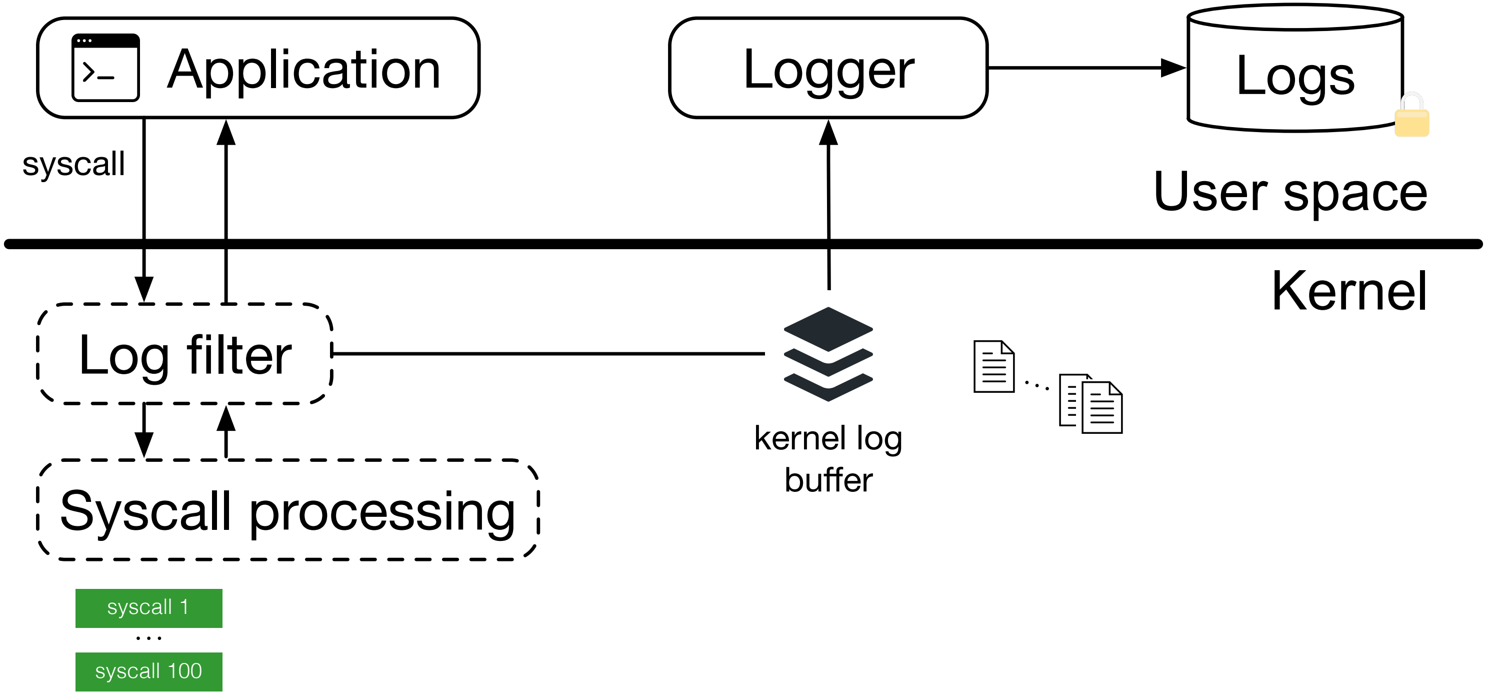
Attack Timeline



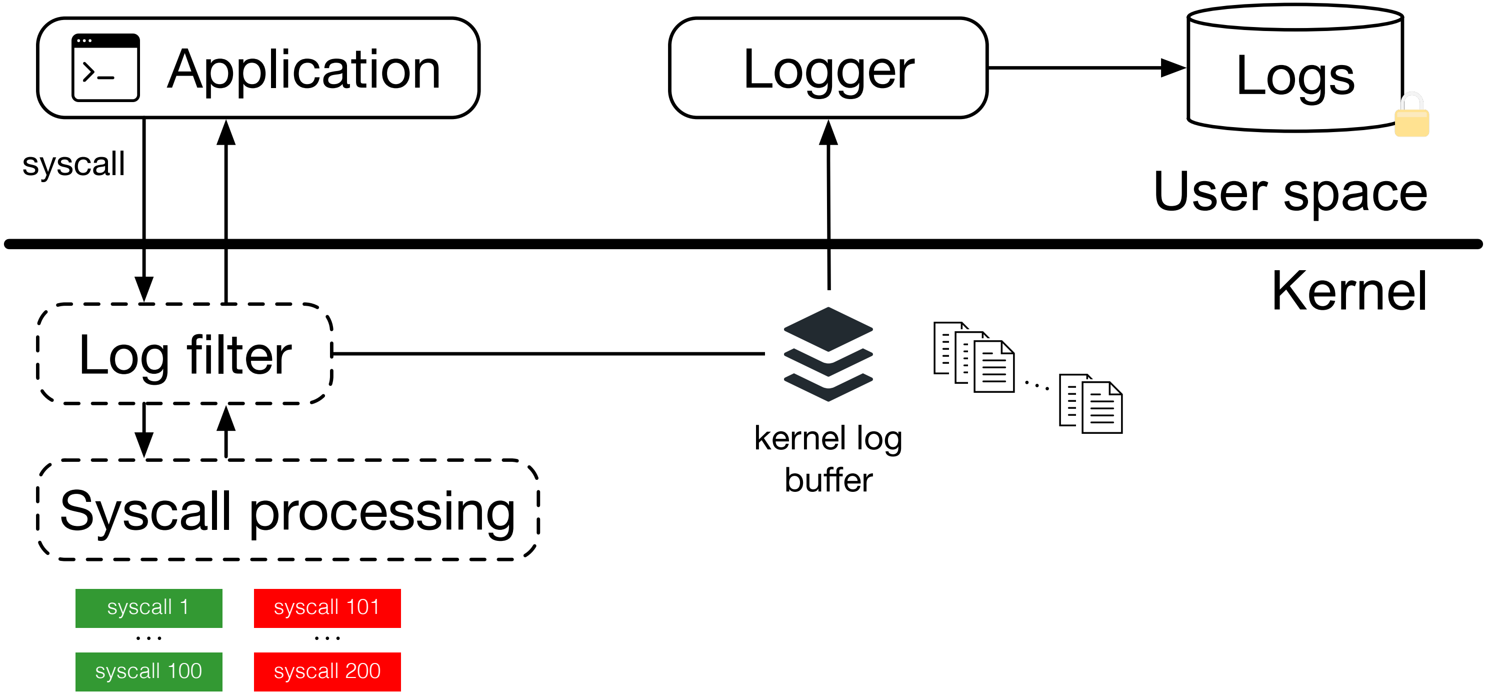
Attack Timeline



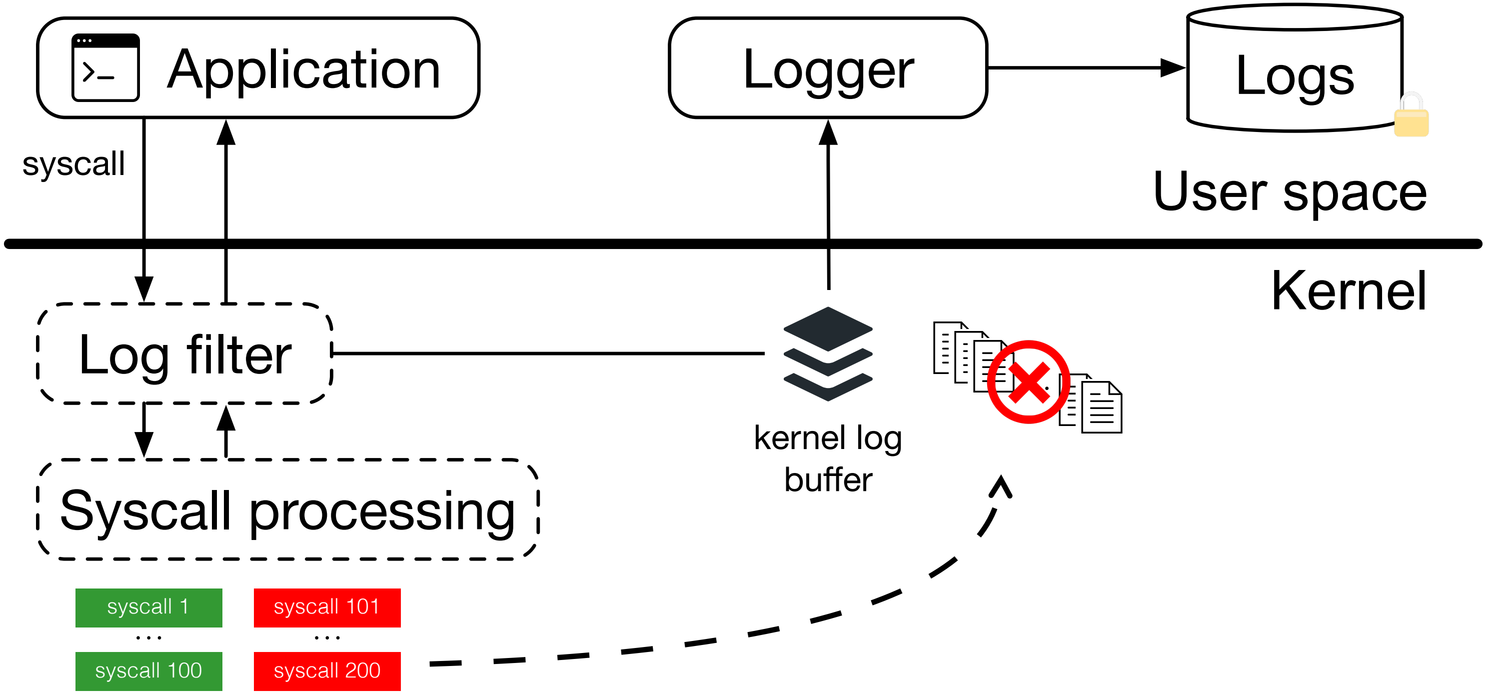
Race Condition Attack



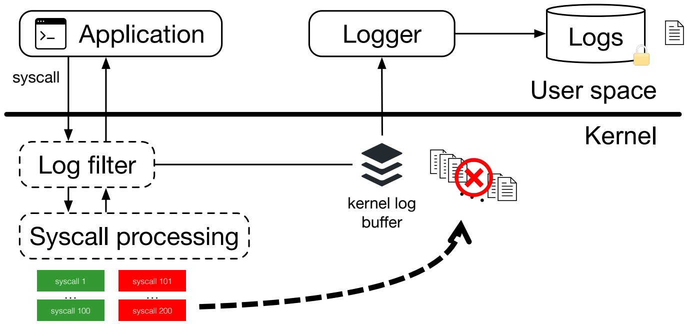
Race Condition Attack



Race Condition Attack

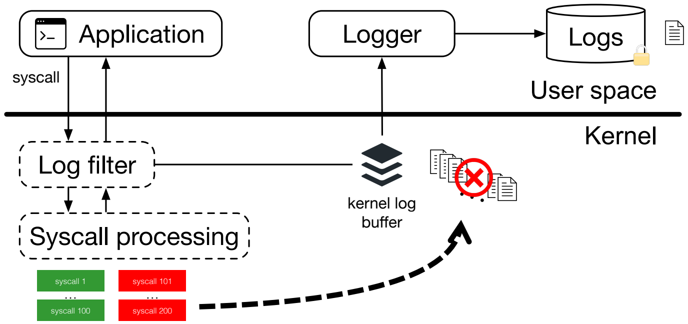


Realistic Attack Scenario



Realistic Attack Scenario

1. Remote code execution: CVE-2014-6271 (Shellshock)
2. Privilege escalation: CVE-2017-16995
3. Log tampering: log-interceptor (details in the paper)



Realistic Attack Scenario

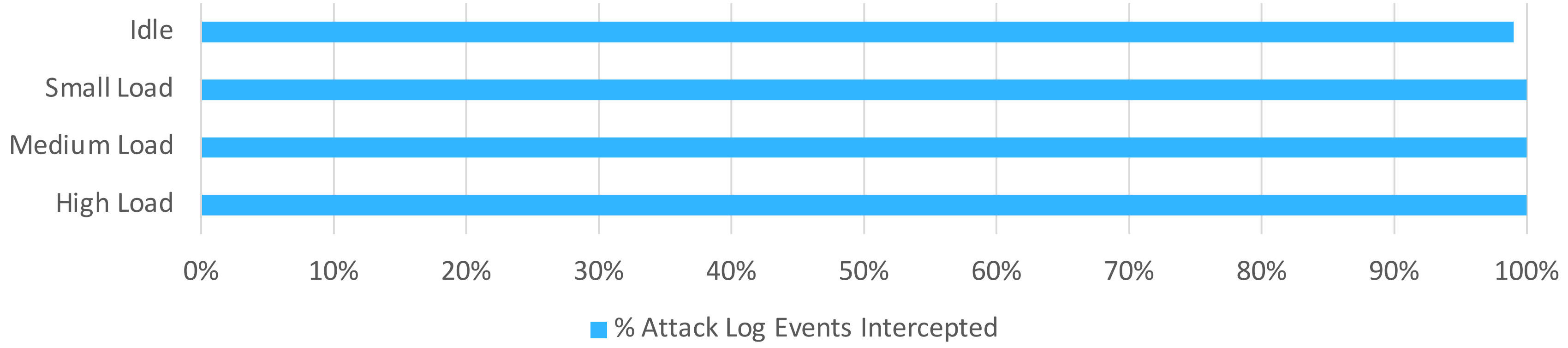
1. Remote code execution: CVE-2014-6271 (Shellshock)
2. Privilege escalation: CVE-2017-16995
3. Log tampering: log-interceptor (details in the paper)

How many of the 176 log events describing the attack steps above get intercepted?

Realistic Attack Scenario

1. Remote code execution: CVE-2014-6271 (Shellshock)
2. Privilege escalation: CVE-2017-16995
3. Log tampering: log-interceptor (details in the paper)

How many of the 176 log events describing the attack steps above get intercepted?

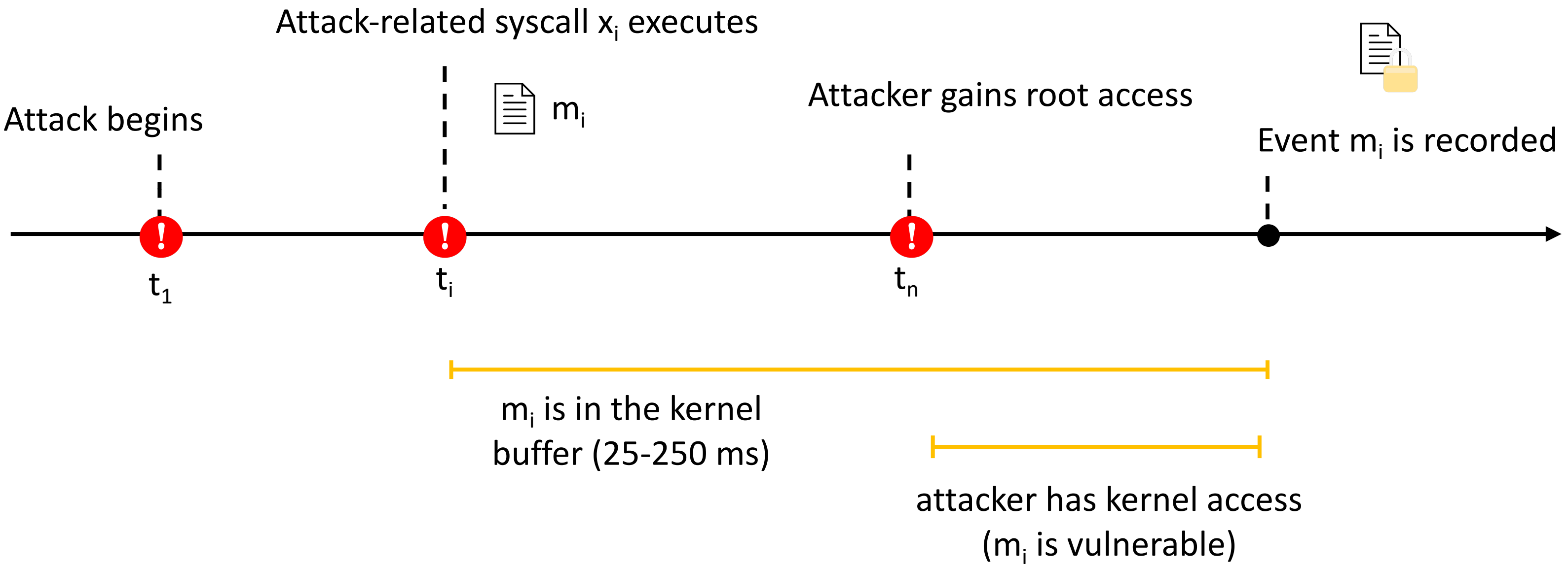


How do we defend against this attack?

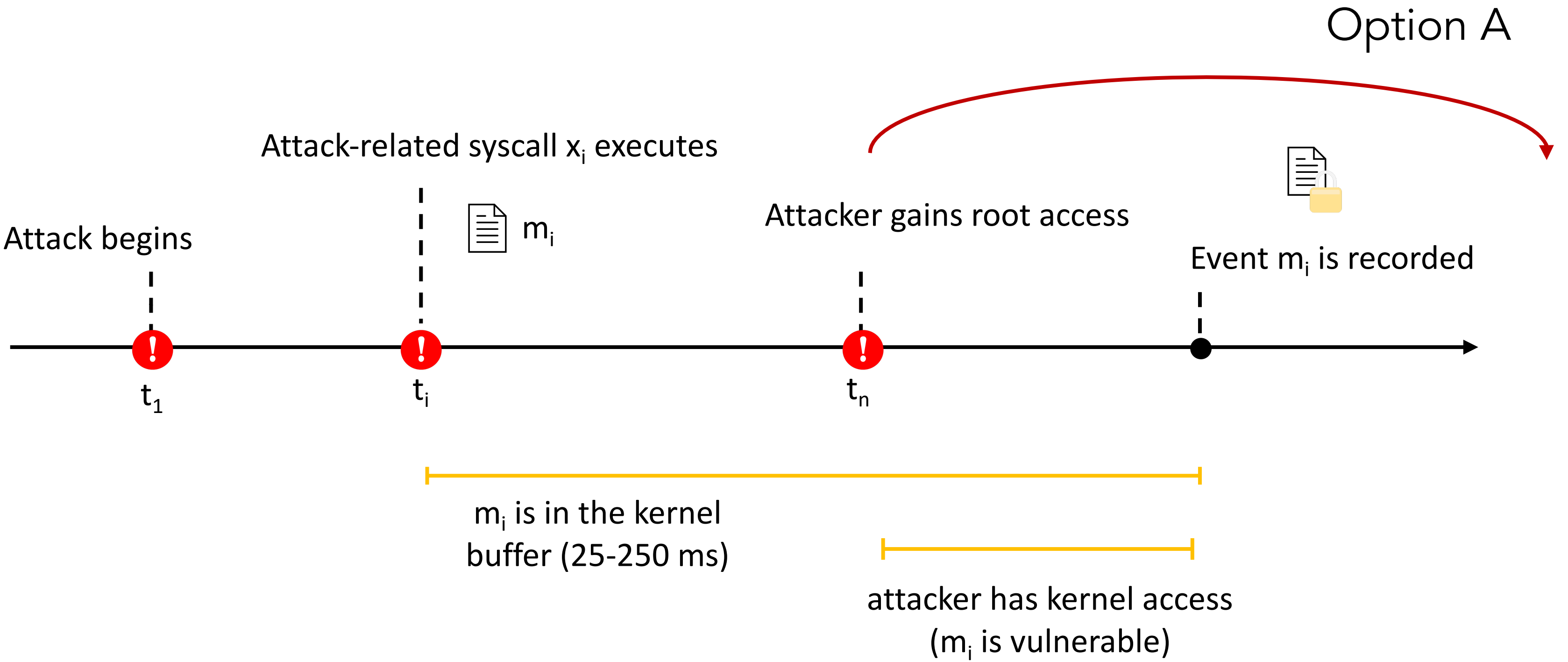
Defense Goals

- **Synchronous integrity**: securing logs synchronously with their creation.

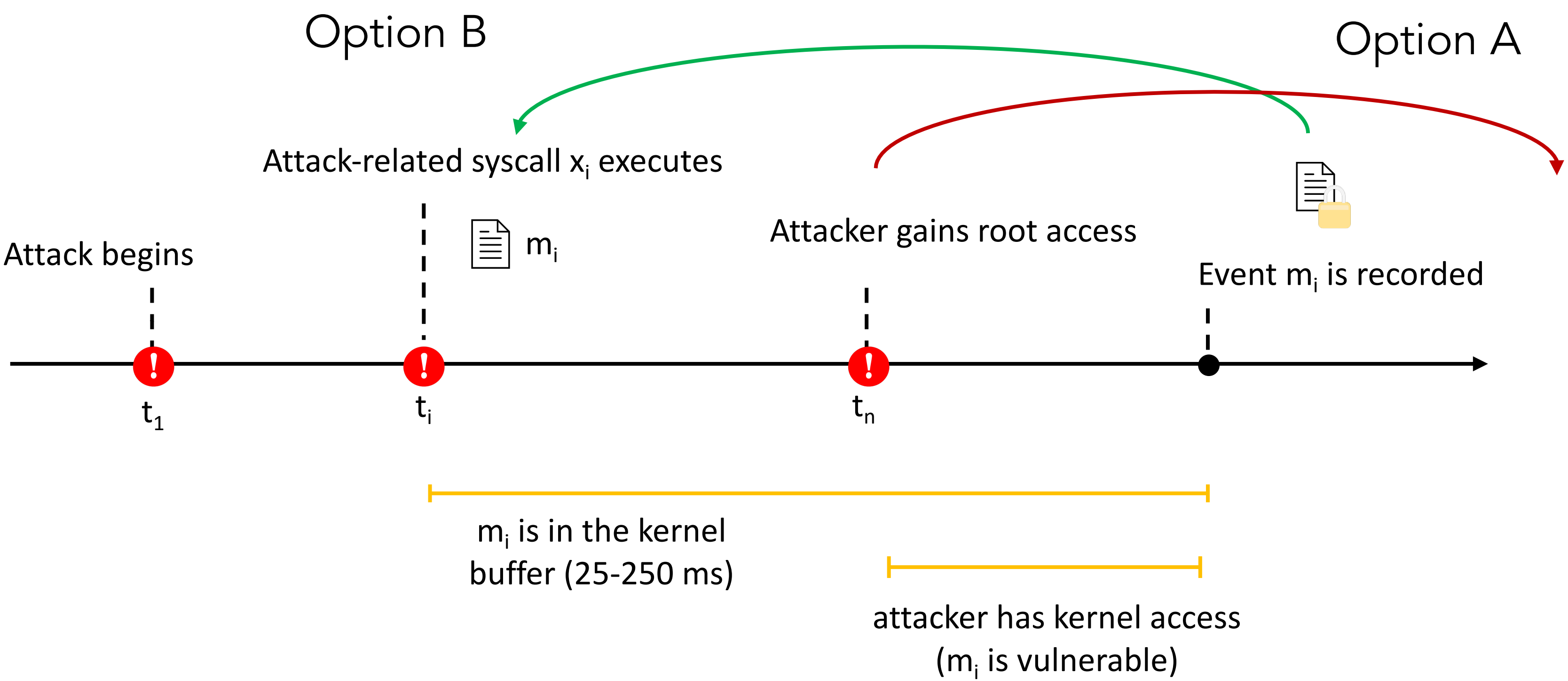
Ways to achieve synchronous integrity



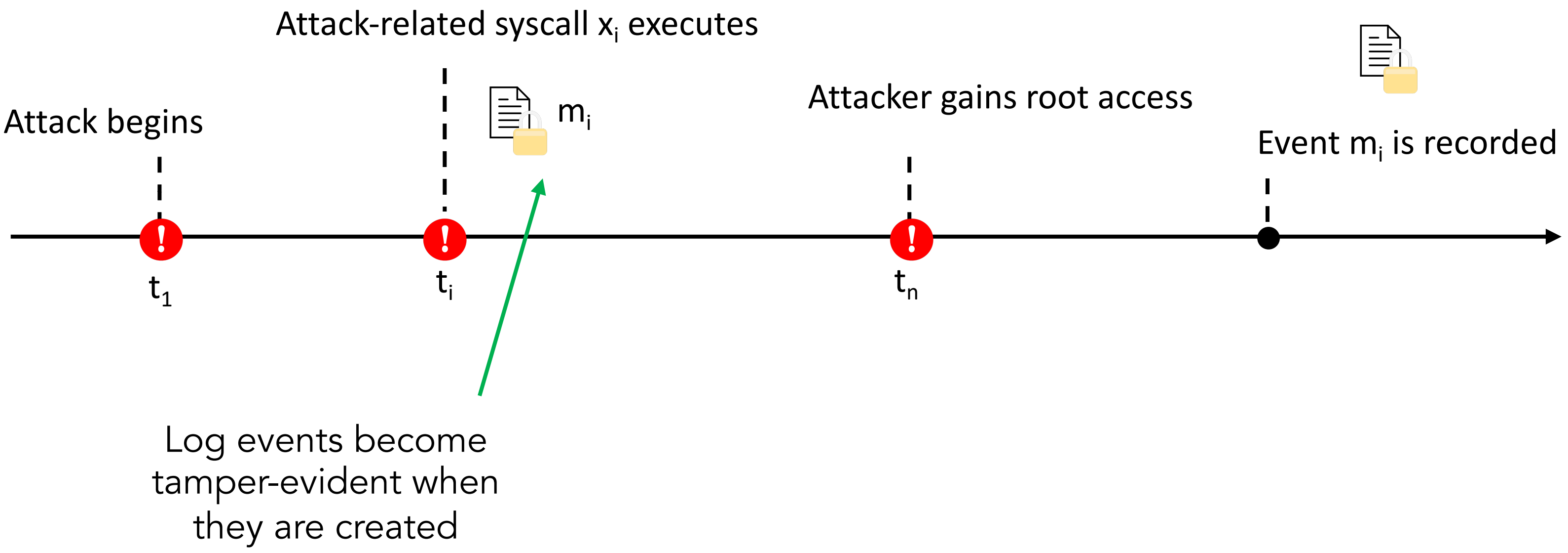
Ways to achieve synchronous integrity



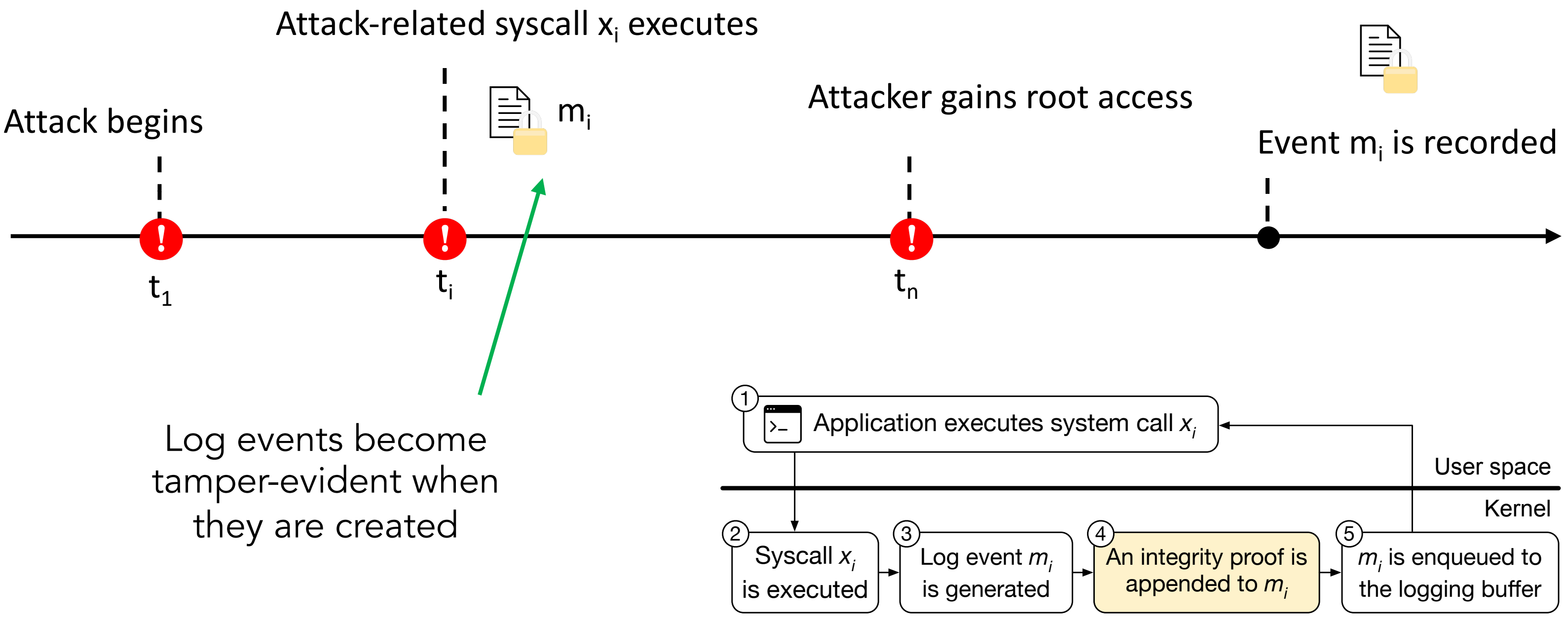
Ways to achieve synchronous integrity



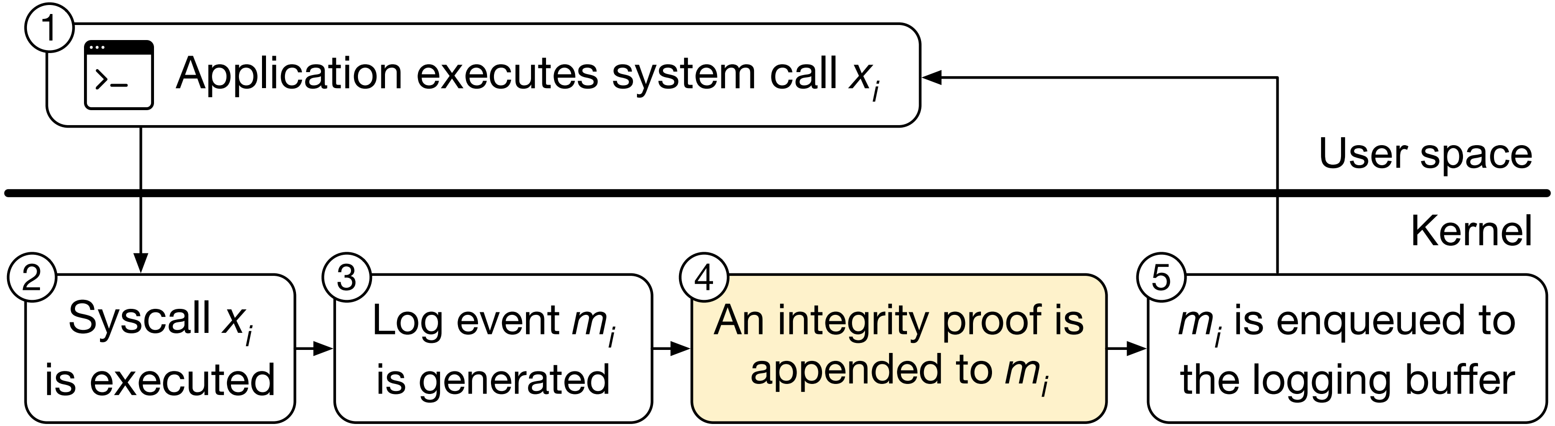
KennyLoggings



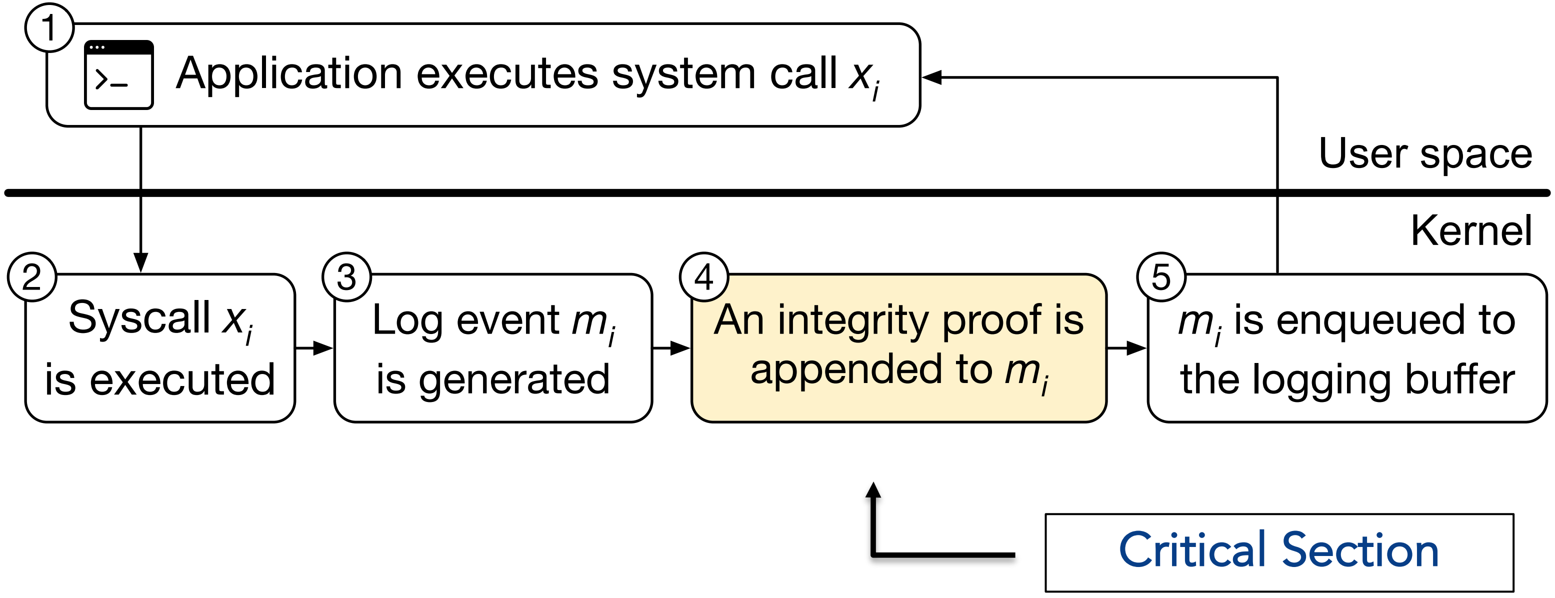
KennyLoggings



Log events become tamper-evident when they are created



Log events become tamper-evident when they are created



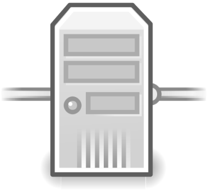
Forward secure MACs

Forward secure MACs

Forward secure MACs

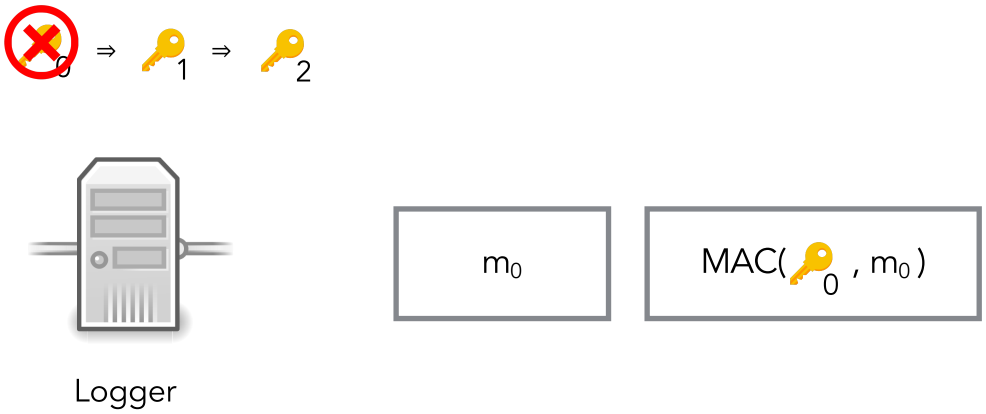
Forward secure MACs

🔑$_0$ ⇒ 🔑$_1$ ⇒ 🔑$_2$

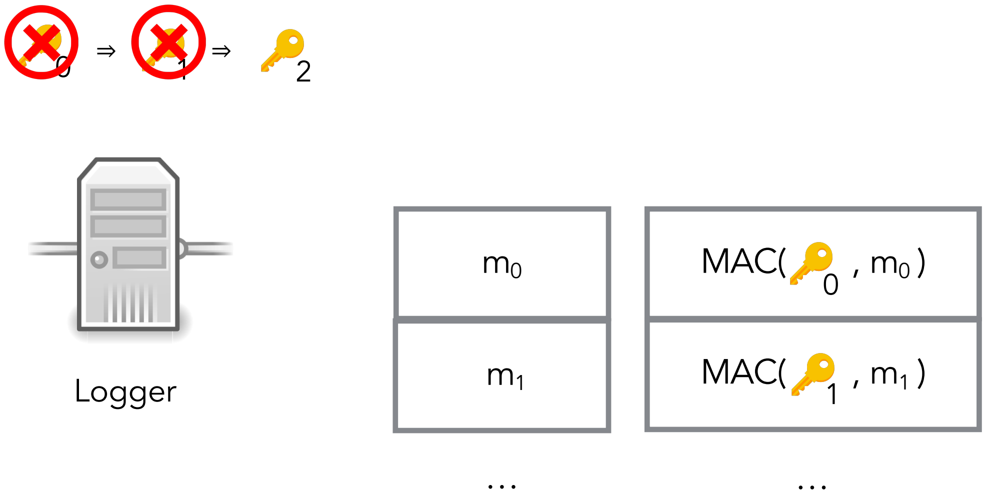


Logger

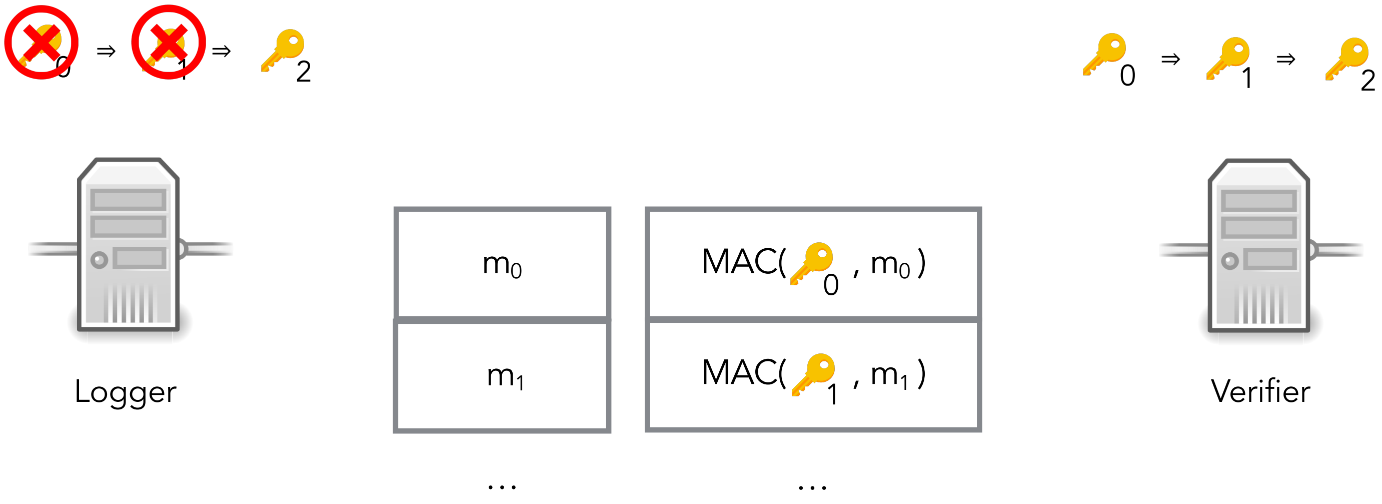
Forward secure MACs



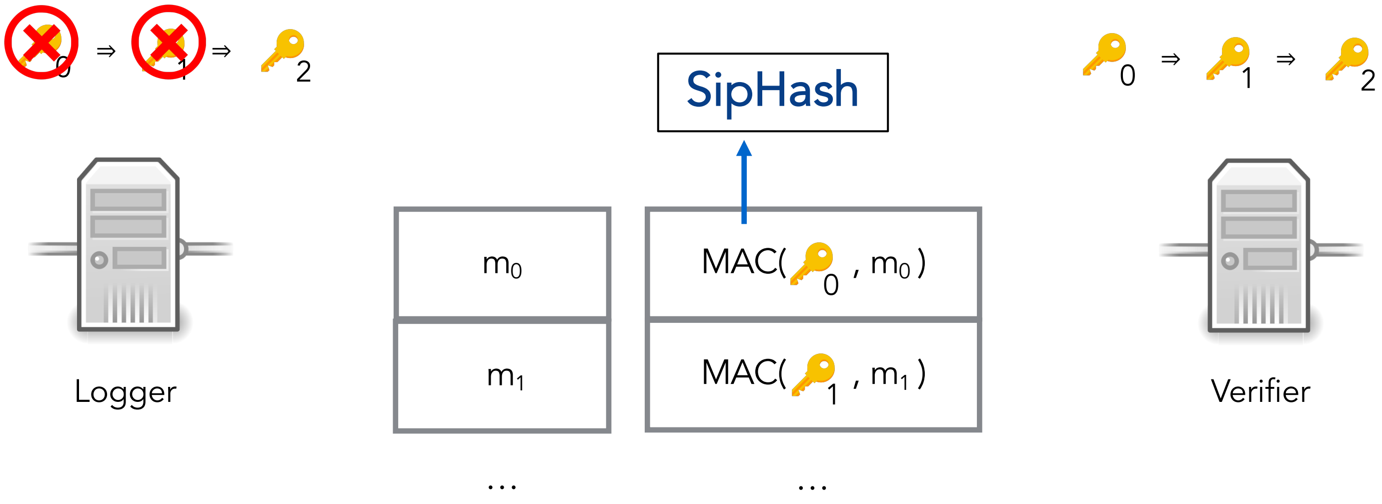
Forward secure MACs



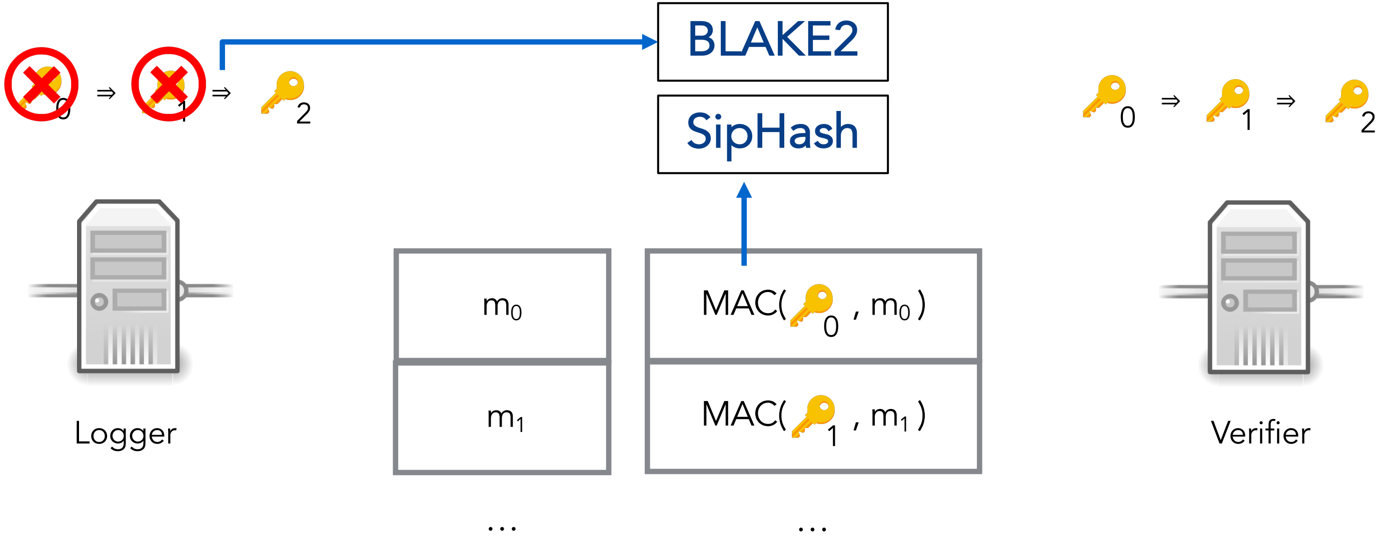
Forward secure MACs



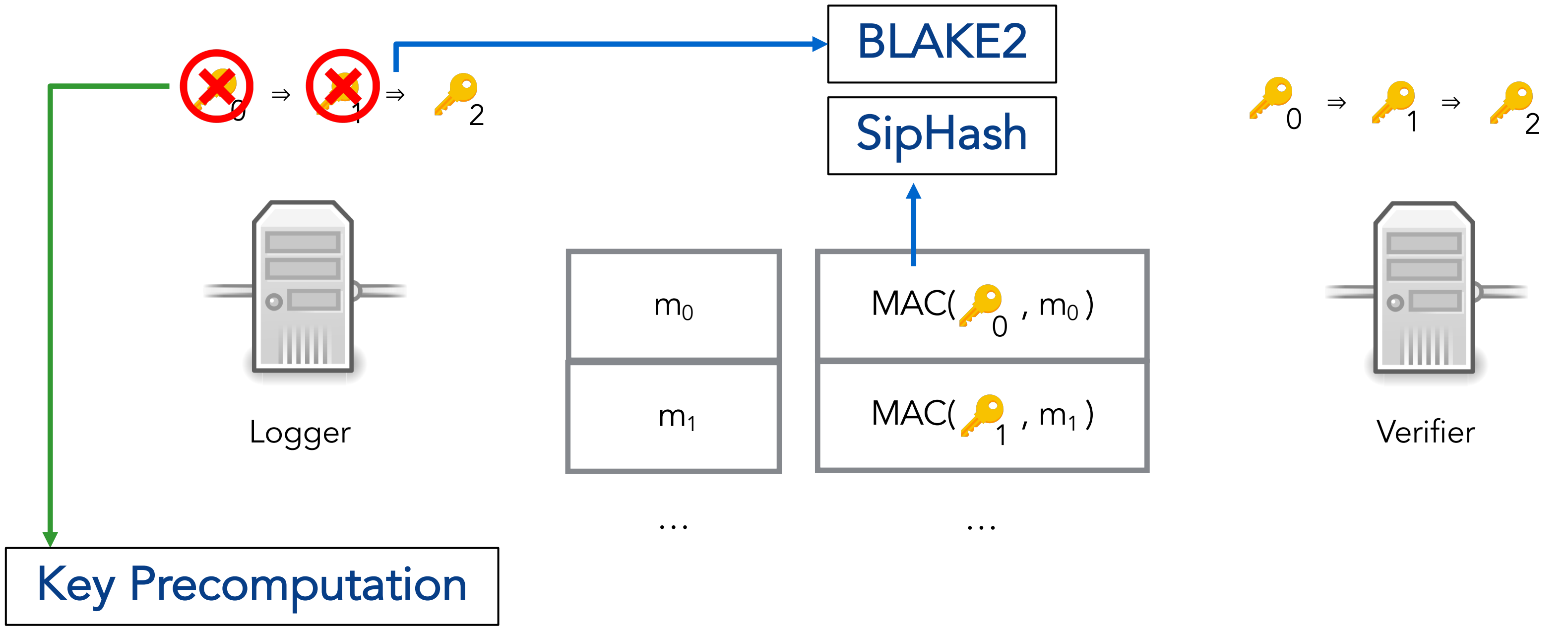
Forward secure MACs



Forward secure MACs



Forward secure MACs

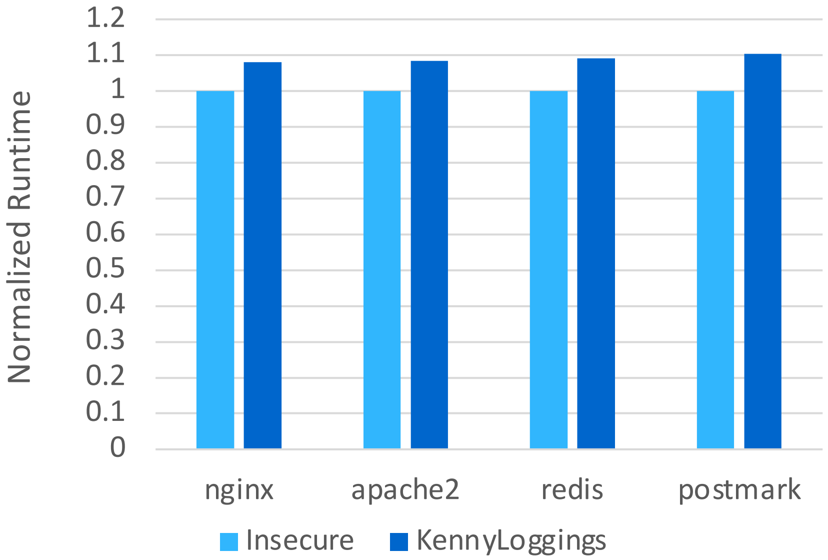


KennyLoggings - Performance

- KennyLoggings adds 340 ns to the kernel control path.

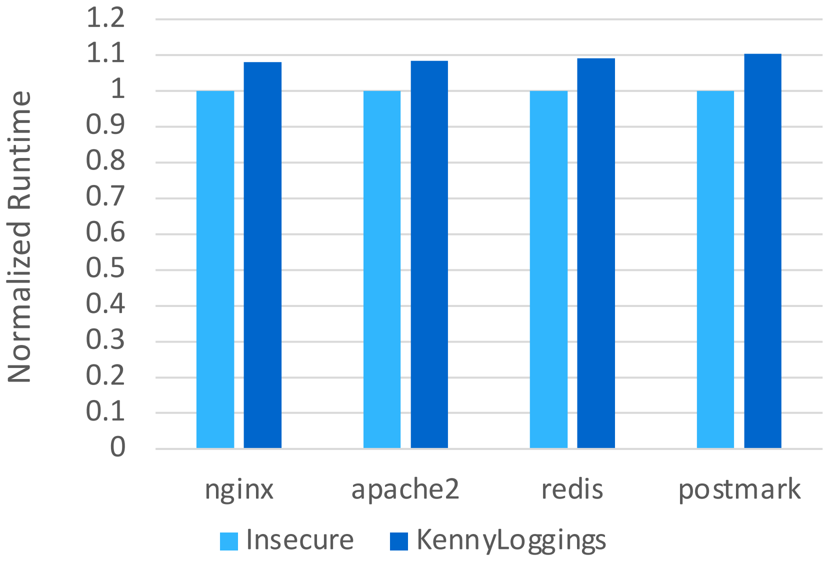
KennyLoggings - Performance

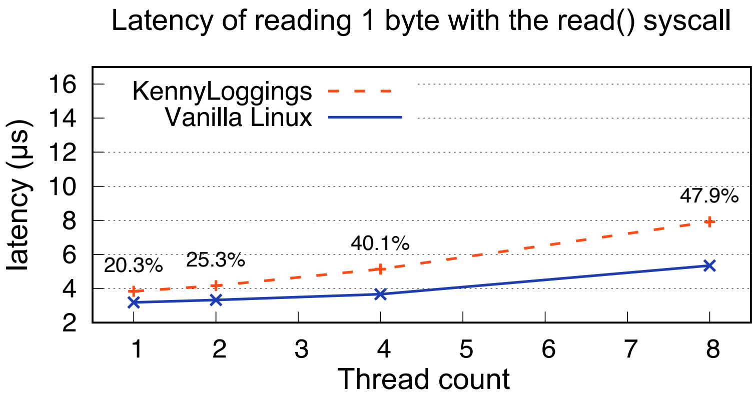
- KennyLoggings adds 340 ns to the kernel control path.



KennyLoggings - Performance

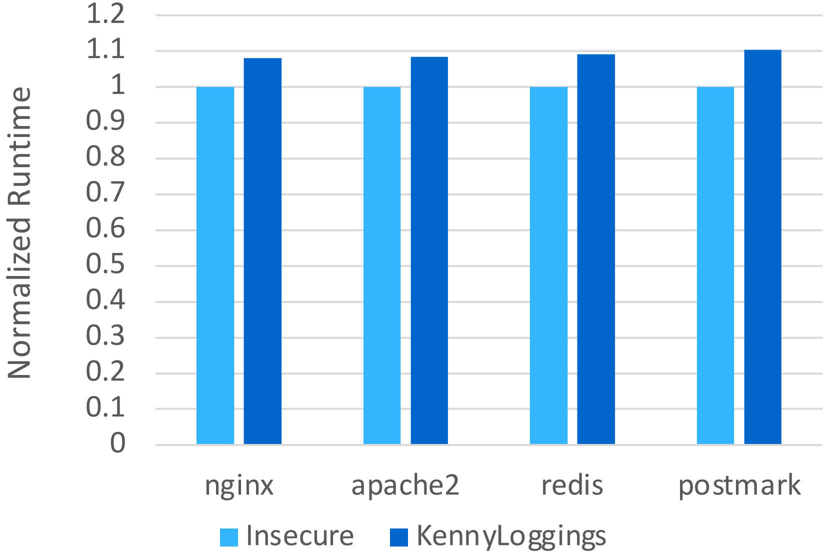
- KennyLoggings adds 340 ns to the kernel control path.

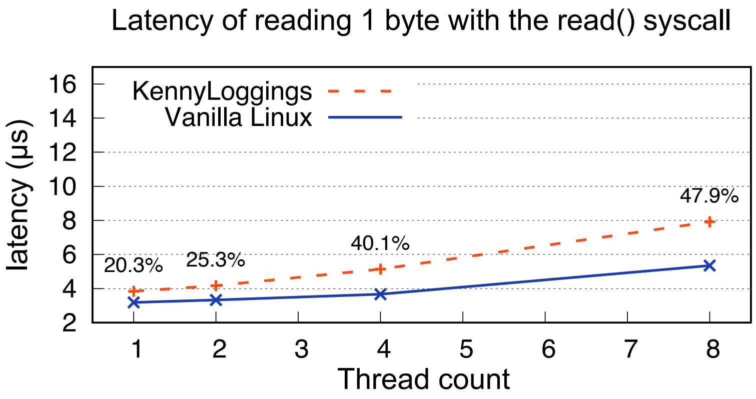




KennyLoggings - Performance

- KennyLoggings adds 340 ns to the kernel control path.





Ultimately, overhead is dictated by contention on the critical section

Conclusion

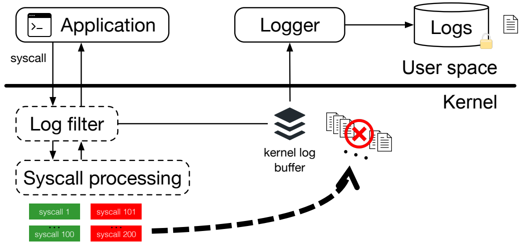
Conclusion

- Asynchronous system logging frameworks are vulnerable to race condition attacks

Hackers are increasingly destroying logs to hide attacks

According to a new report, 72 percent of incident response specialists have came across hacks where attackers have destroyed logs to hide their tracks.

By Catalin Cimpanu for Zero Day | November 2, 2018 -- 16:36 GMT (09:36 PDT) | Topic: Security



Conclusion

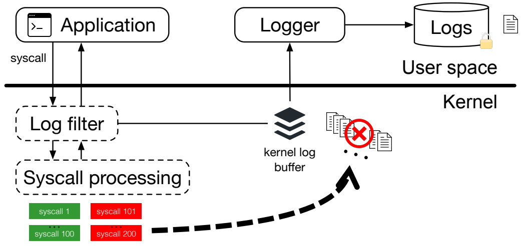
- Asynchronous system logging frameworks are vulnerable to race condition attacks

- Protecting against these attacks necessitates to redefine the requirements of secure logging

Hackers are increasingly destroying logs to hide attacks

According to a new report, 72 percent of incident response specialists have came across hacks where attackers have destroyed logs to hide their tracks.

By Catalin Cimpanu for Zero Day | November 2, 2018 -- 16:36 GMT (09:36 PDT) | Topic: Security



Conclusion

- Asynchronous system logging frameworks are vulnerable to race condition attacks

- Protecting against these attacks necessitates to redefine the requirements of secure logging

- https://bitbucket.org/sts-lab/kennyloggings

Hackers are increasingly destroying logs to hide attacks

According to a new report, 72 percent of incident response specialists have came across hacks where attackers have destroyed logs to hide their tracks.

By Catalin Cimpanu for Zero Day | November 2, 2018 -- 16:36 GMT (09:36 PDT) | Topic: Security