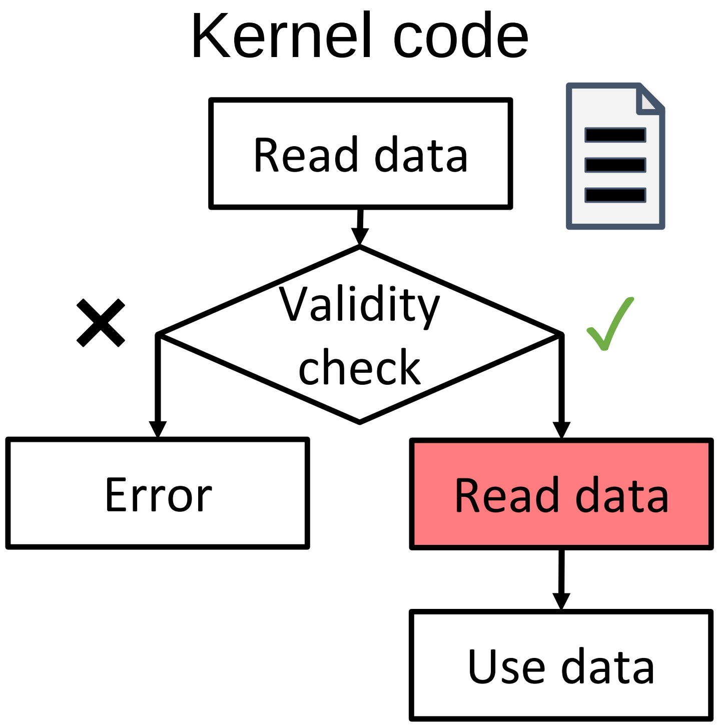
Midas
Systematic Kernel TOCTTOU Protection

Atri Bhattacharyya, Uros Tesic, Mathias Payer
EPFL
(Usenix Security ‘22)

Kernel TOCTTOU Bugs

- Kernels have double-fetch bugs
  - E.g., TOCTTOU (*Time-of-Check-to-Time-of-Use*)
- Common in security-critical software
  - Linux kernel, modules and SECCOMP
  - Hypervisors (KVM)
  - TEEs (TrustZone)
- Powerful CVEs for Linux
  - 2016-8438 – "Complete compromise"
  - 2020-25212 – "… information disclosure"



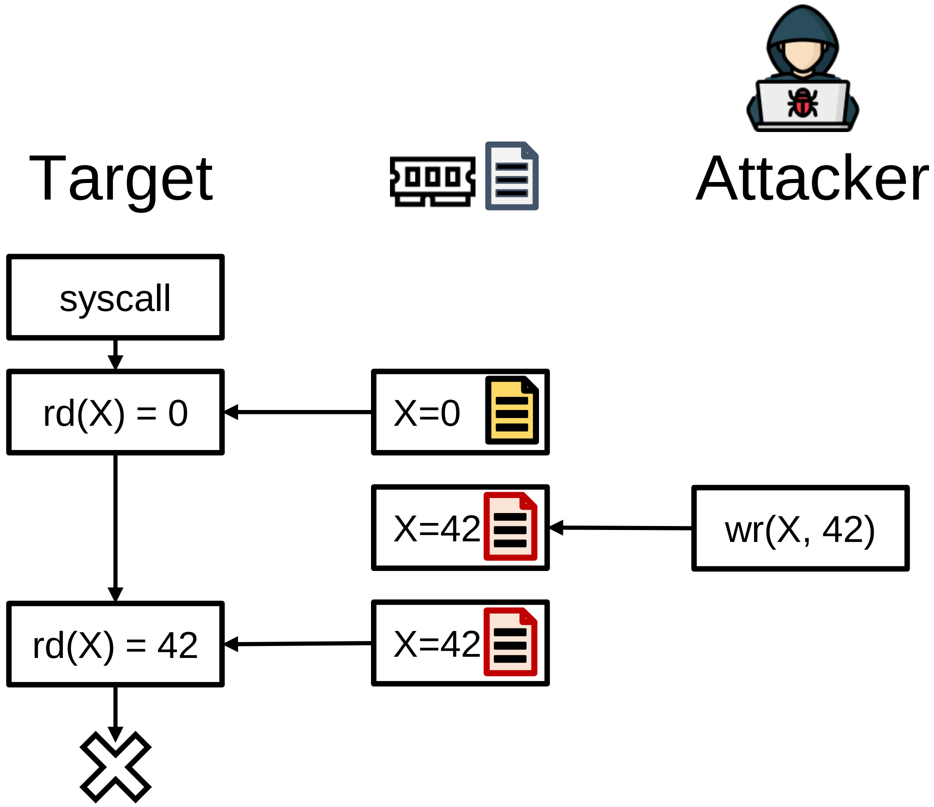
TOCTTOU bugs exist across security-critical interfaces

Exploiting TOCTTOU Bugs

- Vulnerable syscall reads userspace data

```
sigaction(signum,*act,*oldact)
   if(*(act->X) < len){…}
   …
   access(array[*(act->X)]);
```

- Attacker needs two userspace threads
  - One thread calls syscall
  - Second thread modifies data



TOCTTOU bugs are easy to exploit

image: Flaticon.com

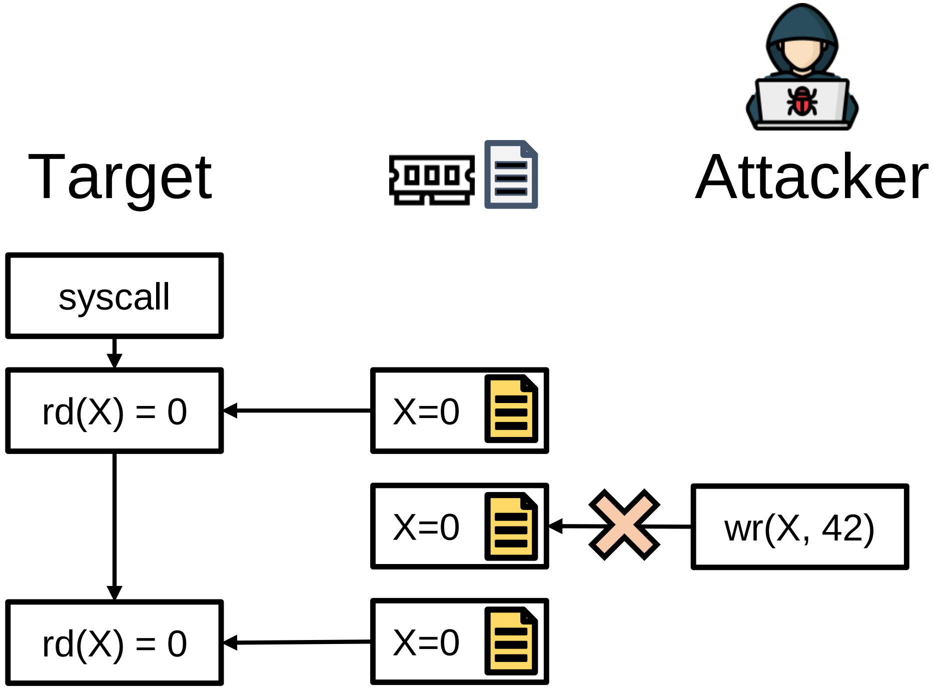
Mitigating TOCTTOU Exploitation

Cause: Different values read over time

Insights

- *Transfer functions* to read from user
- Page tables control access to pages

Fix: Ensure kernel reads the same value

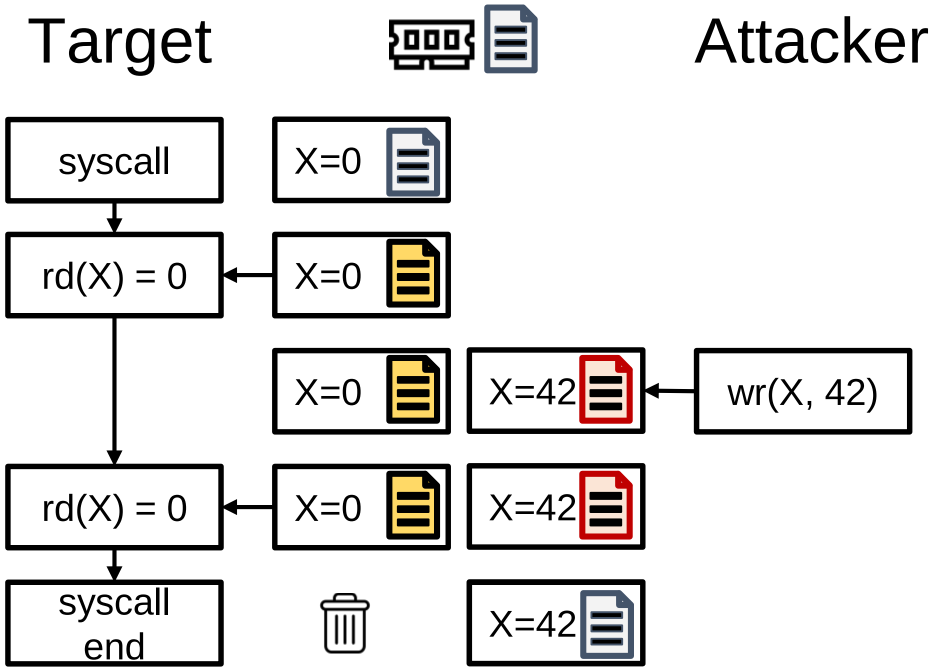


Midas mitigates TOCTTOU bugs throughout the kernel

Midas' Invariant

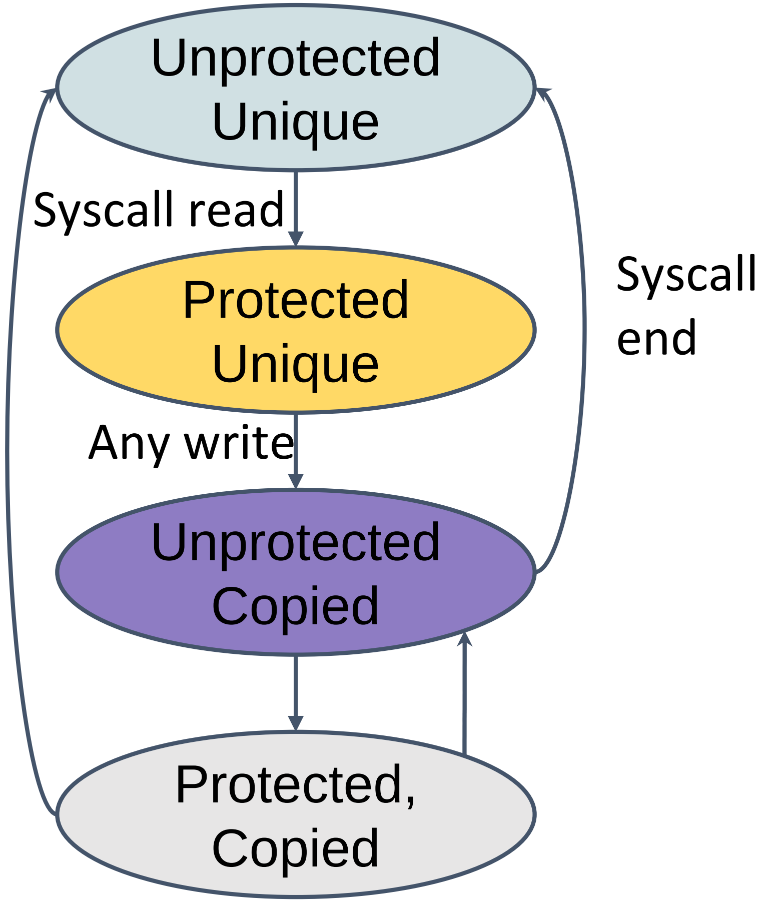
> "Through a *syscall's lifetime,*
> *every read* to a userspace object
> will return the *same value.*"

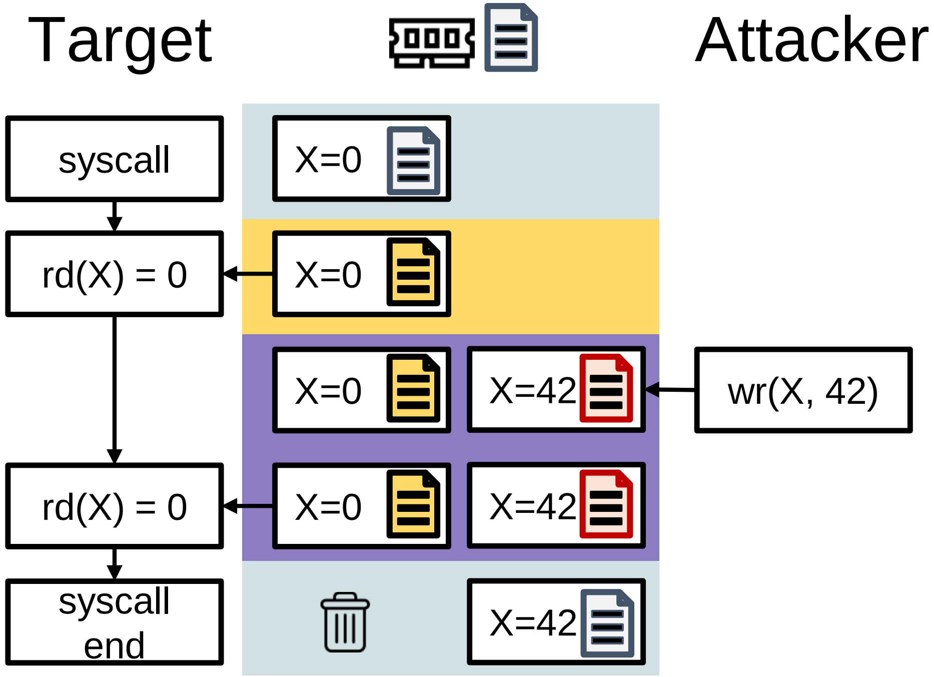
- Snapshot page on first read
- Read from snapshot on future reads
- Duplicate page on concurrent writes
- Discard snapshot when syscall finishes



Midas implements multi-versioning for userspace pages with a state machine

Page State Machine





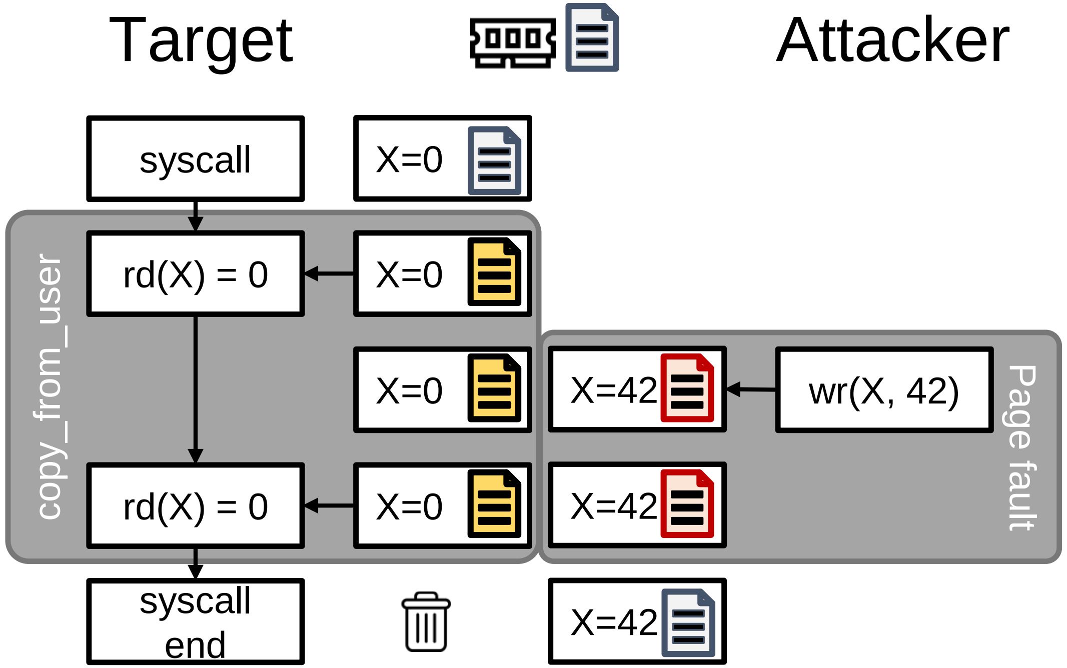
Protecting Golden Pages

Special function for userspace access

- OS explicitly knows userspace reads
- `copy_from_user` function
- Instrument interface to read same data

Hardware-enforced access control

- Permissions specified in page tables
- Writes to read-only pages raise faults
- OS handles page faults



Existing OS/hardware features enables Midas to protect snapshots

Conclusion

Midas systematically mitigates TOCTTOU bugs

- From userspace and kernel
- Leverages page tables and `copy_from_user`

Implements state machine

- Protected/unprotected
- Copied/unique

Low average overhead (average 3.4%)

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

🔗 https://hexhive.epfl.ch/midas

Midas provides comprehensive low-overhead double-fetch kernel protection