Unsafe at Any Copy: Name Collisions from Mixing Case Sensitivities

Aditya Basu[+], John Sampson[+], Zhiyun Qian[■], and Trent Jaeger[+]

[+] *The Pennsylvania State University*

[■] *University of California, Riverside*

February 22, 2023

About Me

Aditya Basu
Mail: aditya.basu@psu.edu
Web: adityabasu.me

Research interests:
Systems,
Filesystem security,
Kernel security

PennState

Advisor: Trent Jaeger

Outline

- Background
- Problem of Name Collisions
- Detection
- Results
- Defenses

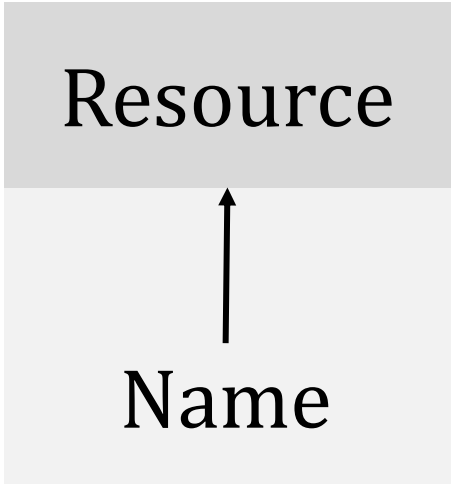
File systems use *names* to store and retrieve resources

Bad/Unexpected name ⟷ resource mapping:

- Data loss and/or corruption
- Leak confidential data

Additionally,

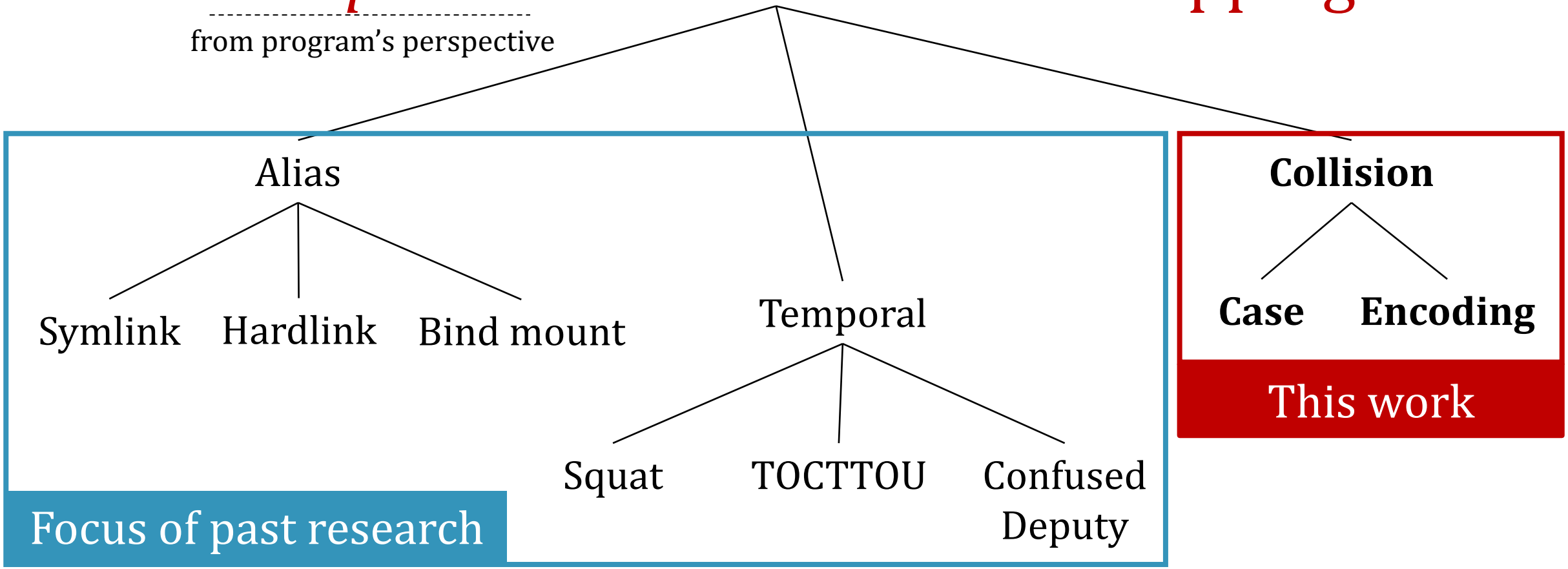
- File squatting
- Symbolic link traversal
- Time-of-check to time-of-use (TOCTTOU)
- Confused deputy



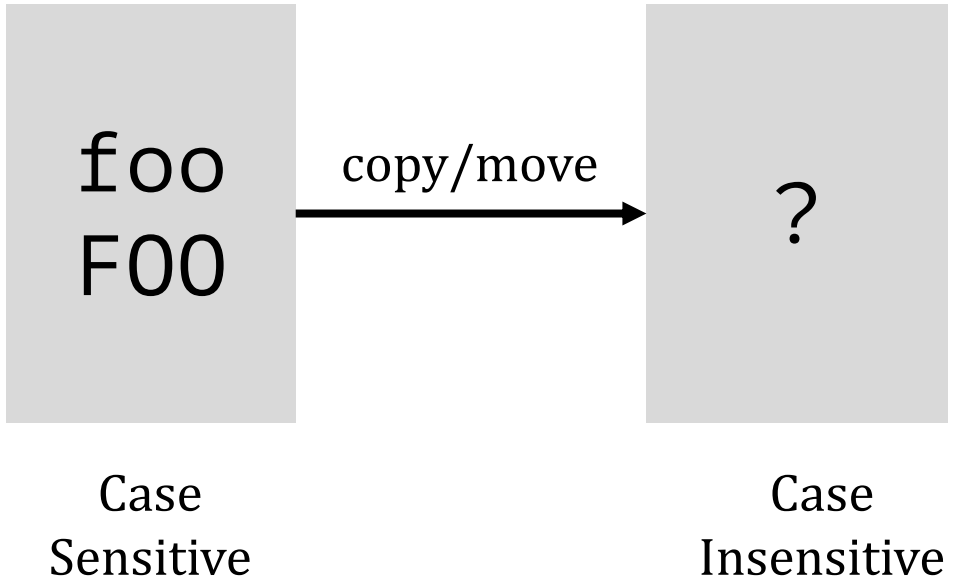
Name confusion is the result of

unexpected name ↔ resource mapping

from program's perspective



Name collision results in the *unexpected* reuse of existing resources



① `foo`
② `FOO`
③ `foo` with data of `FOO`
④ `FOO` with data of `foo`
⑤ `Error`

Multi-lingual case insensitivity:

- floß, FLOSS and floss are identical
- Kid vs. Kid (kelvin)

Our results show that the popular copy utilities are not well-behaved

Case Diversity is an actual concern

Ext4 recently added per-directory case-insensitivity (multi-lingual)

Additionally,

- JFS, ZFS, FAT, NTFS, ciopfs, and F2FS are case-insensitive
- Patches for case-insensitive `tmpfs`
- Windows Subsystem on Linux (WSL)

Motivation:

- SAMBA
- NFS
- WINE
- Proton games
- Android

Real-world Implications

Git CVEs

- CVE-2021-21300
- CVE-2014-9390

Collision between symbolic link and directory name

Result: Link traversal leads to arbitrary script execution

Collision due to crafted name

Result: Add .GIT/config to repo

Happens when `git` is exposed to case-insensitive filesystems

Our Contributions

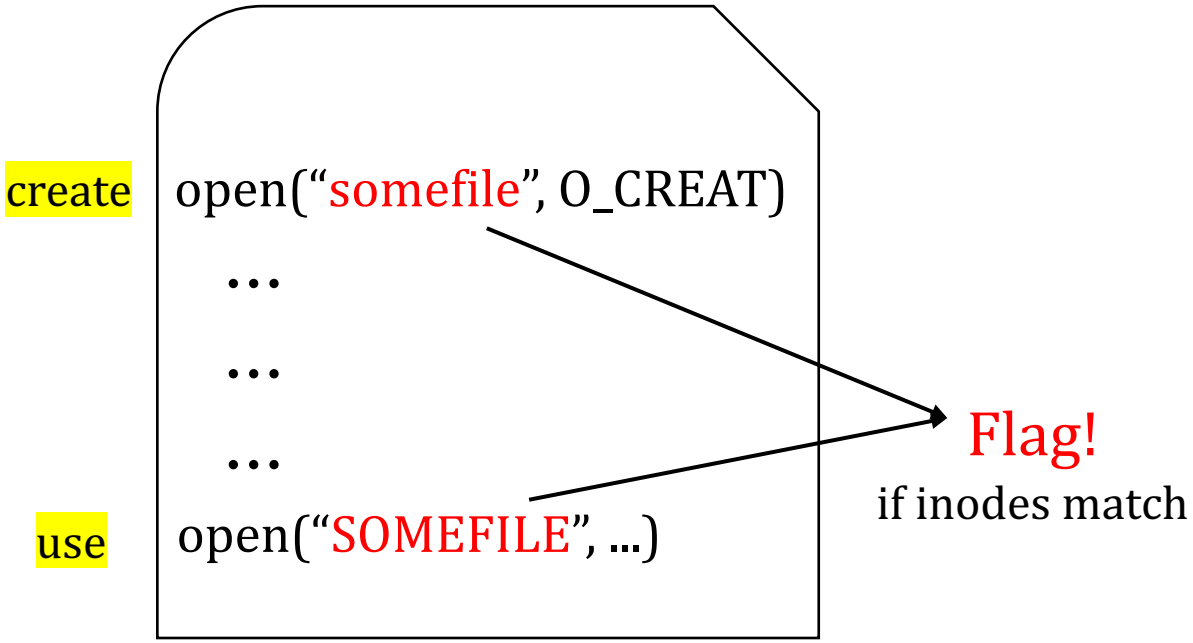
- We coin the term *name collisions* to describe a new class of naming problems
- We developed an automated method to test copy utilities and identify **unsafe responses** to name collision.
- We demonstrate **novel exploits** for `dpkg`, `rsync` and `httpd`

Improper case handling can lead to:

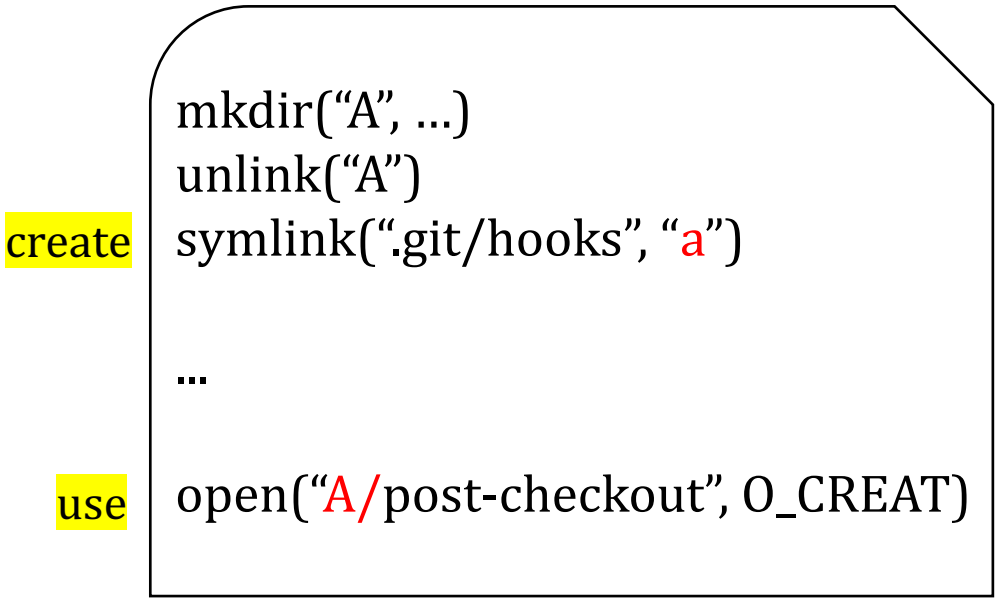
- Data Loss/Corruption
- Symlink traversal
- Hardlink corruption
- Unauthorized access
- Data disclosure

How to detect collisions?

CREATE-USE pairs are a succinct yet powerful way of capturing collisions



Syscall Trace



Syscall Trace for Git CVE-2021-21300

To identify **CREATE-USE** pairs, we need to capture system-level behavior

`strace` – system calls only

`Auditd` – system calls + *filesystem context*

Single event in Auditd

```
time->Wed Jul  7 14:54:51 2021
type=PROCTITLE msg=audit(1625684091.351:709): proctitle=6370...
type=PATH msg=audit(1625684091.351:709): item=1
name="/mercury/research/casefolding/tmp/ROOT" inode=667 dev=00:39 ...
type=PATH msg=audit(1625684091.351:709): item=0 name="/mercury/research/casefolding/tmp/"
inode=642 dev=00:39 mode=040775 ...
type=CWD msg=audit(1625684091.351:709): cwd="/mercury/research/casefolding/name-
confusion"
type=SYSCALL msg=audit(1625684091.351:709): arch=c000003e syscall=133 success=yes exit=0
a0=5638fffe3da0 a1=11c0 a2=0 a3=5638fffe1010 items=2 ppid=10395 pid=10396 auid=1000 uid=0
gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts7 ses=7962 comm="cp"
exe="/bin/cp" key="icase"
```

Structure of Auditd Logs

Record Type

/path/name

inode & dev

Syscall

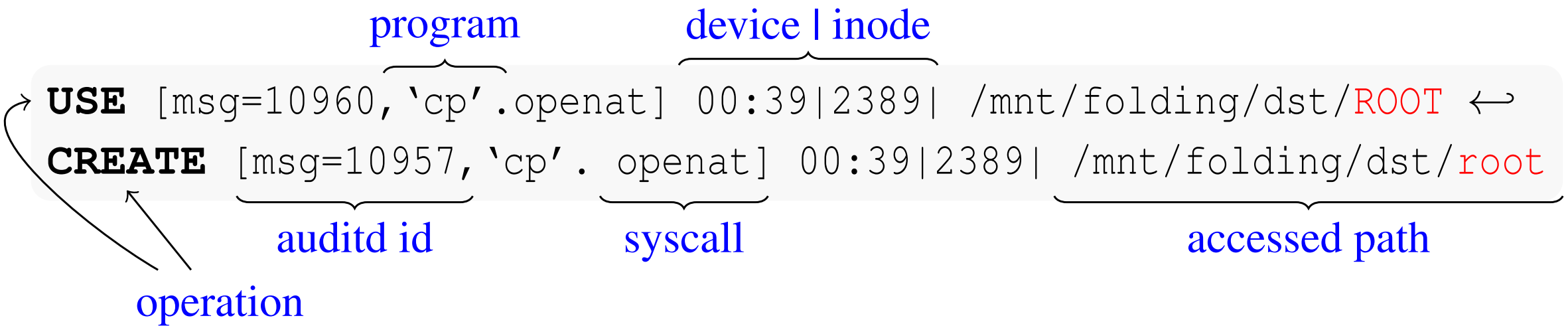
Executable

We need to extract meaningfully context from this!

- time->Wed Jul 7 14:54:51 2021
- type=PROCTITLE msg=audit(1625684091.351:709): proctitle=6370...
- type=PATH msg=audit(1625684091.351:709): item=1 name="/mercury/research/casefolding/tmp/ROOT" inode=667 dev=00:39 ...
- type=PATH msg=audit(1625684091.351:709): item=0 name="/mercury/research/casefolding/tmp/" inode=642 dev=00:39 mode=040775 ...
- type=CWD msg=audit(1625684091.351:709): cwd="/mercury/research/casefolding/name-confusion"
- type=SYSCALL msg=audit(1625684091.351:709): arch=c000003e syscall=133 success=yes exit=0 a0=5638fffe3da0 a1=11c0 a2=0 a3=5638fffe1010 items=2 ppid=10395 pid=10396 auid=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts7 ses=7962 comm="cp" exe="/bin/cp" key="icase"

Single event in auditctl

Extracting **CREATE-USE** pairs from Auditd logs



Test Suite to drive utilities

Generate all combinations of **collisions**

to test various copying utilities:

- cp
- tar
- zip
- rsync
- Dropbox

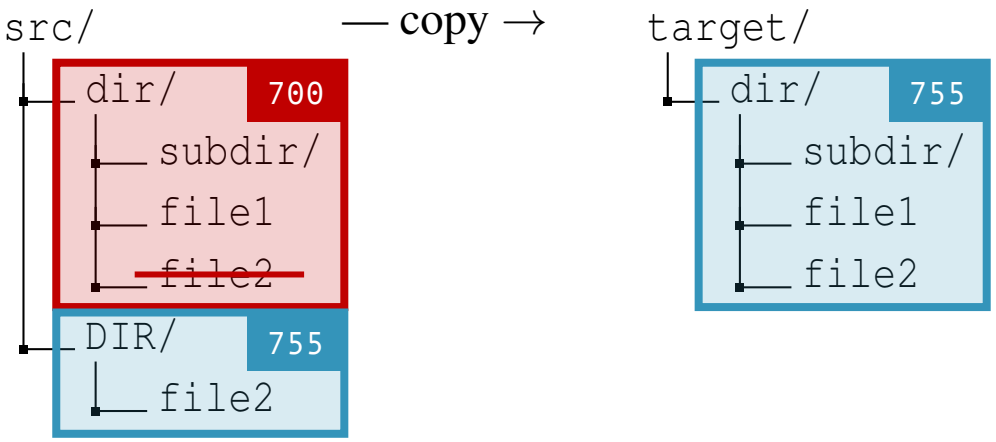
Case-sensitive filesystem

Case-insensitive filesystem

Here, * is a regular file and | is a named pipe

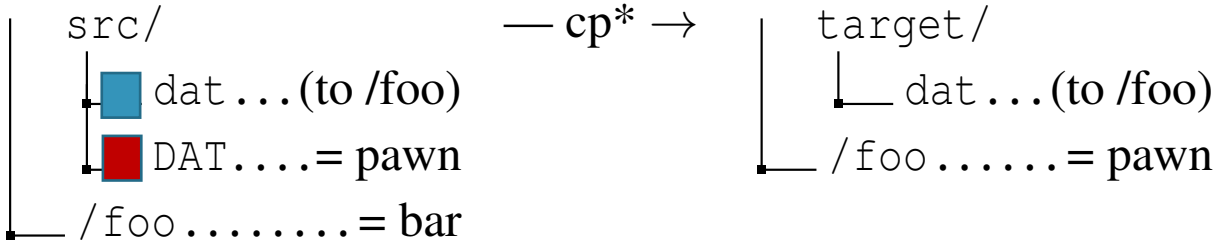
Results

Merging Directories



- Silent data loss
- Incorrect metadata can lead to security concerns
- `tar`, `zip`, `cp` and `rsync` are impacted

Overwrite existing files



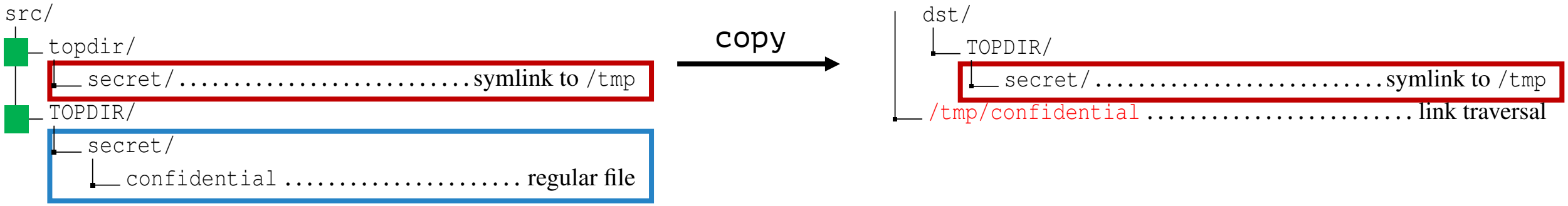
- Copy **`dat`**
- Overwrite **`dat`** with contents of **`DAT`** leading to symlink traversal

Problem: Breaks outside the target/

rsync

Parent directories collide!

Link Traversal



Symbolic Link to a folder outside the target **Directory** containing a file

Symbolic Link was followed, and **confidential file** was created outside dst/

Defense Strategies

Program-only defenses are unreliable because the underlying case folding rules are unknown.

System-only defense will suffer from false positives without any programmer intent.

A sound defense strategy will potentially involve creating system call APIs that allow programmers to convey intent.

Defense Strategies

- Flags for `openat2()`
  - O_CREAT: Create file if it does not exist
  - O_NOFOLLOW: Do not follow symbolic links at target
  - O_EXCLNAME : Fail open if the filename's case differs

- Add system call to retrieve the **canonicalized name**
- Add system call to query the file system about **equality of strings**

Thanks!

Summary

- Name collisions are becoming prevalent but are under-studied.
- We developed an automated method to test for collision.
- Our testing revealed different types of unsafe responses to collisions.
- We demonstrate novel exploits using name collisions.

Reach me at aditya.basu@psu.edu

Artifacts: https://github.com/mitthu/name-confusion