

# How Double-Fetch Situations turn into Double-Fetch Vulnerabilities:

---

## A Study of Double Fetches in the Linux Kernel

Pengfei Wang, Jens Krinke, Kai Lu, Gen Li, Steve Dodier-Lazaro

College of Computer  
National University of Defense Technology, China  
Centre for Research on Evolution, Search and Testing  
University College London



國防科學技術大學  
National University of Defense Technology



# OUTLINE

- **What is a double fetch ?**
- **A static pattern-based double fetch analysis.**
- **Results and Findings.**

# Double Fetch

## First Appearance - Fermin J. Serna, CVE-2008-2252

### Microsoft Security Bulletin MS08-061 – Important

#### Vulnerabilities in Windows Kernel Could Allow Elevation of Privilege (954211)

Published: October 14, 2008

Version: 1.0

#### General Information

##### Executive Summary

This security update resolves one publicly disclosed and two privately reported vulnerabilities in the Windows kernel. A local attacker who successfully exploited these vulnerabilities could take complete control of an affected system. The vulnerabilities could not be exploited remotely or by anonymous users.

This security update is rated Important for all supported editions of Microsoft Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008. For more information, see the subsection, **Affected and Non-Affected Software**, in this section.

The security update addresses the vulnerabilities by correcting window property validation passed during the new window creation process, correcting the manner in which system calls from multiple threads are handled, and correcting validation of parameters passed to the Windows Kernel from user mode. For more information about the vulnerabilities, see the Frequently Asked Questions (FAQ) subsection for the specific vulnerability entry under the next section, **Vulnerability Information**.

**Recommendation.** Microsoft recommends that customers apply the update at the earliest opportunity.

**Known Issues.** [Microsoft Knowledge Base Article 954211](#) documents the currently known issues that customers may experience when installing this security update. The article also documents recommended solutions for these issues.

##### Affected and Non-Affected Software

The following software have been tested to determine which versions or editions are affected. Other versions or editions are either past their support life cycle or are not affected. To determine the support life cycle for your software version or edition, visit [Microsoft Support Lifecycle](#).

# Double Fetch

## First Study: Jurczyk & Coldwind - 2013

### Bochspwn: Exploring Conditions Found via Pattern

Mateusz "j00ru" Jurczyk  
SysSec  
Sing

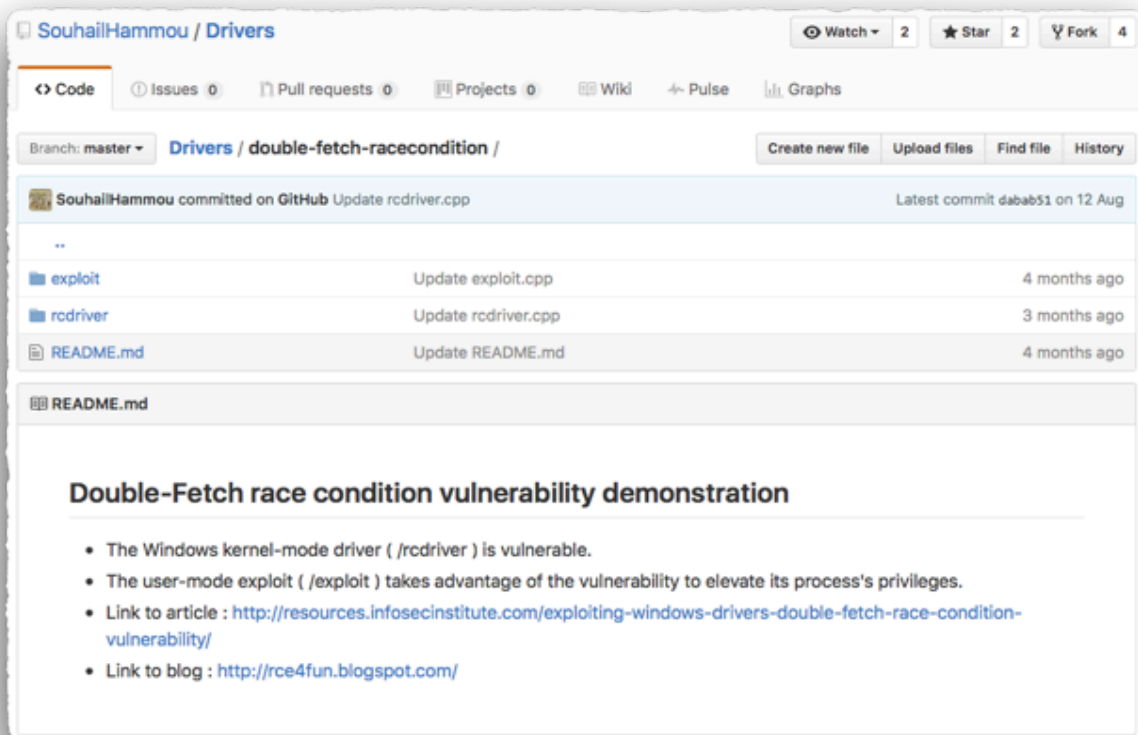
### Stats: **bochspwn** vs Windows

- **89 potential** new issues discovered
  - + part of the initial 27 bugs were also rediscovered
  - All were reported to Microsoft (Nov 2012 - Jan 2013)
- **36 EoPs** (+3 variants) addressed by: MS13-016, MS13-017, MS13-031, MS13-036
- **13** issues have been classified as **Local DoS** only
- **7** more are being analyzed / are scheduled to be fixed
- The rest were unexploitable / non-issues / etc

Tested: Windows 7 32-bit, Windows 8 32-bit and Windows 8 64-bit.

# Double Fetch

## Exploit Instructions on GitHub - 2016



The screenshot shows a GitHub repository page for 'SouhailHammou / Drivers'. The repository has 2 watches, 2 stars, and 4 forks. The current branch is 'master', and the selected file is 'Drivers / double-fetch-racecondition /'. The commit history shows the following updates:

File	Update	Time
exploit	Update exploit.cpp	4 months ago
rcdriver	Update rcdriver.cpp	3 months ago
README.md	Update README.md	4 months ago

The README.md file contains the following content:

### Double-Fetch race condition vulnerability demonstration

- The Windows kernel-mode driver ( /rcdriver ) is vulnerable.
- The user-mode exploit ( /exploit ) takes advantage of the vulnerability to elevate its process's privileges.
- Link to article : <http://resources.infosecinstitute.com/exploiting-windows-drivers-double-fetch-race-condition-vulnerability/>
- Link to blog : <http://rce4fun.blogspot.com/>

# Double Fetch Vulnerabilities Today: Where are they?

---

**Bochspwn** Is dynamic, slow and limited code coverage.

---

**Weakness** Did not show why double fetches happen.

---

Only workable for Windows.

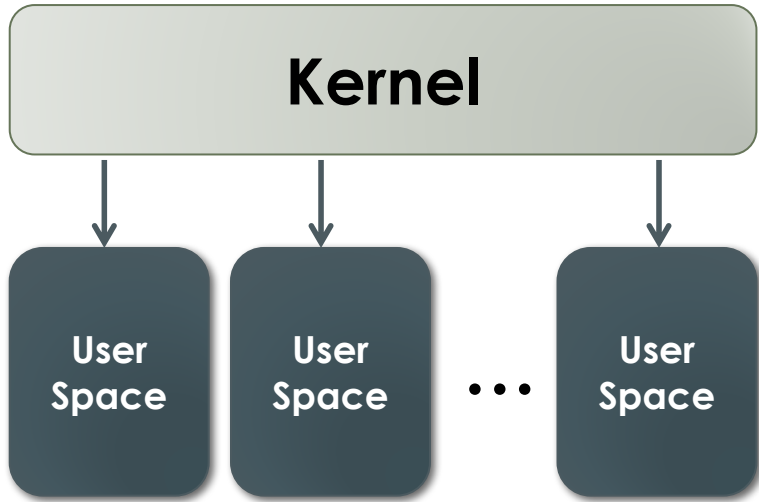
---

Cannot analyze driver code without hardware.

---

- **Linux had double fetch vulnerabilities, but no dedicated audit has been done.**
- **We need a static analysis to cover the complete kernel including all drivers(44%).**

# Operating Systems: Separate Address Spaces



- Each user process has its own virtual memory space
- User spaces are isolated.
- Only the kernel can access all user spaces.

# Operating Systems: System Call Interface

**Kernel**

```
graph TD; Kernel[Kernel] --> Syscall[Syscall]; Syscall --> UserSpace[User Space];
```



**Syscall**



**User  
Space**

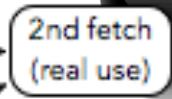
- Fundamental Interface between application and kernel
- Arguments are copied
  - either directly or
  - as pointers to data structures
- The kernel cannot trust any data coming from the application!



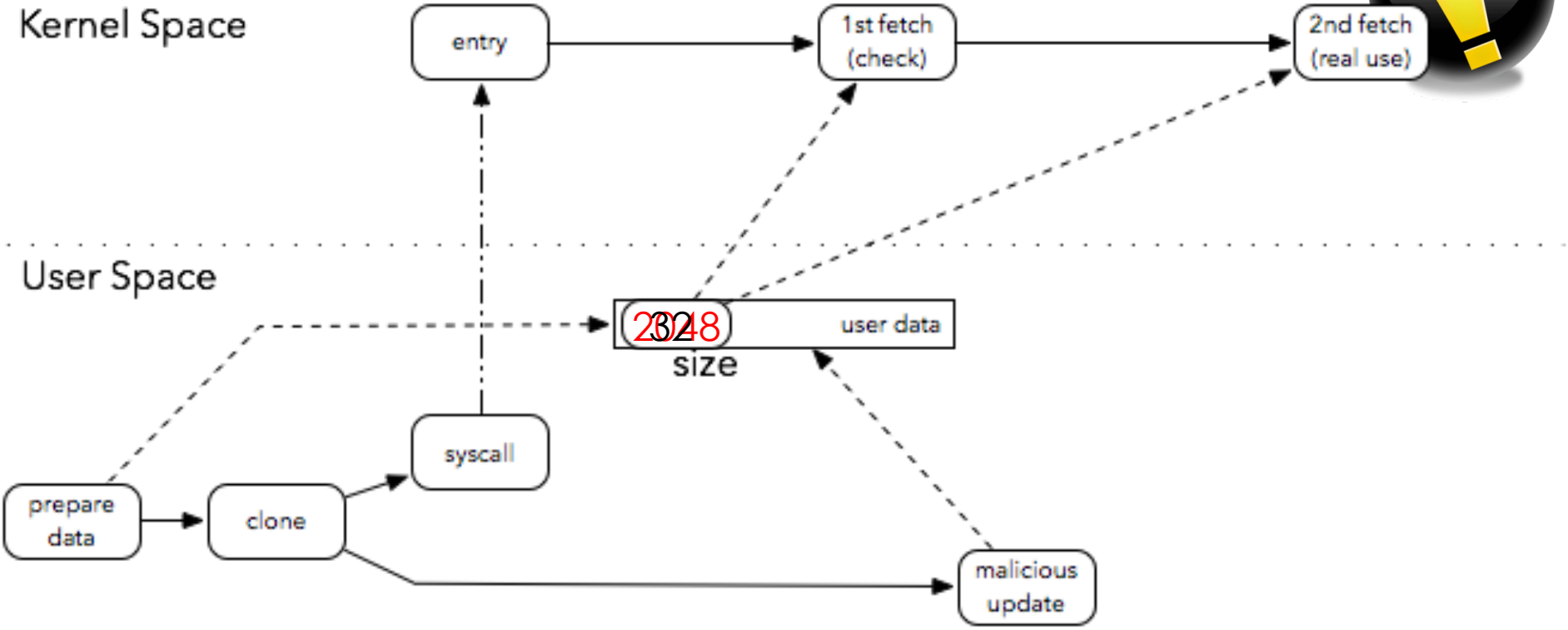
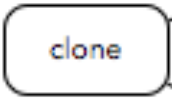
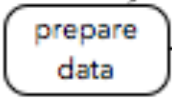
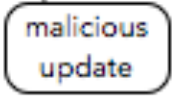
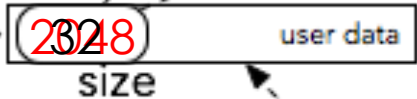
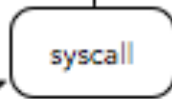
# Anatomy of a Double Fetch



Kernel Space



User Space



## Transfer Functions in Linux

- Linux uses dedicated functions to copy data between user and kernel space:

```
get_user(src)
copy_from_user(dst, src, size)
put_user(dst)
copy_to_user(dst, src, size)
```

- Data in user space is not accessed directly:  
Ensures that the access is valid.

# Double-fetch bug in Linux (CVE-2016-5728)

```
522 static int mic_copy_dp_entry(...) {
533     ...
534     if(copy_from_user(&dd, argp, sizeof(dd))) {
535         ...
536         return -EFAULT;
537     }
538     dd_config = kmalloc(mic_desc_size(&dd), GFP_KERNEL);
539     if (dd_config == NULL) {
540         ...
541         return -ENOMEM;
542     }
543     if(copy_from_user(dd_config, argp, mic_desc_size(&dd))) {
544         ret = -EFAULT;
545         ...
546     }
547     for ( i = sizeof(struct mic_bootparam);
548          i < MIC_DP_SIZE - mic_total_desc_size(dd_config);
549          i += mic_total_desc_size(devp)) {
550         devp = mdev->dp + i;
551         ...
552     }
553     memcpy(devp, dd_config, mic_desc_size(dd_config));
554     ...
555 }
556 }
```

Allocate buffer  
use 'size' from first fetch

Use 'size' from  
second fetch

# Static Pattern-Based Approach

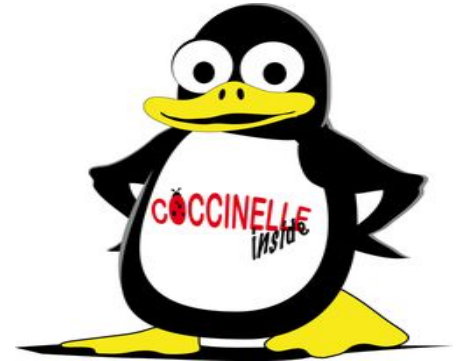
# Pattern-based Double Fetch Analysis

Based on Coccinelle (Julia Lawall, LIP6 – France)

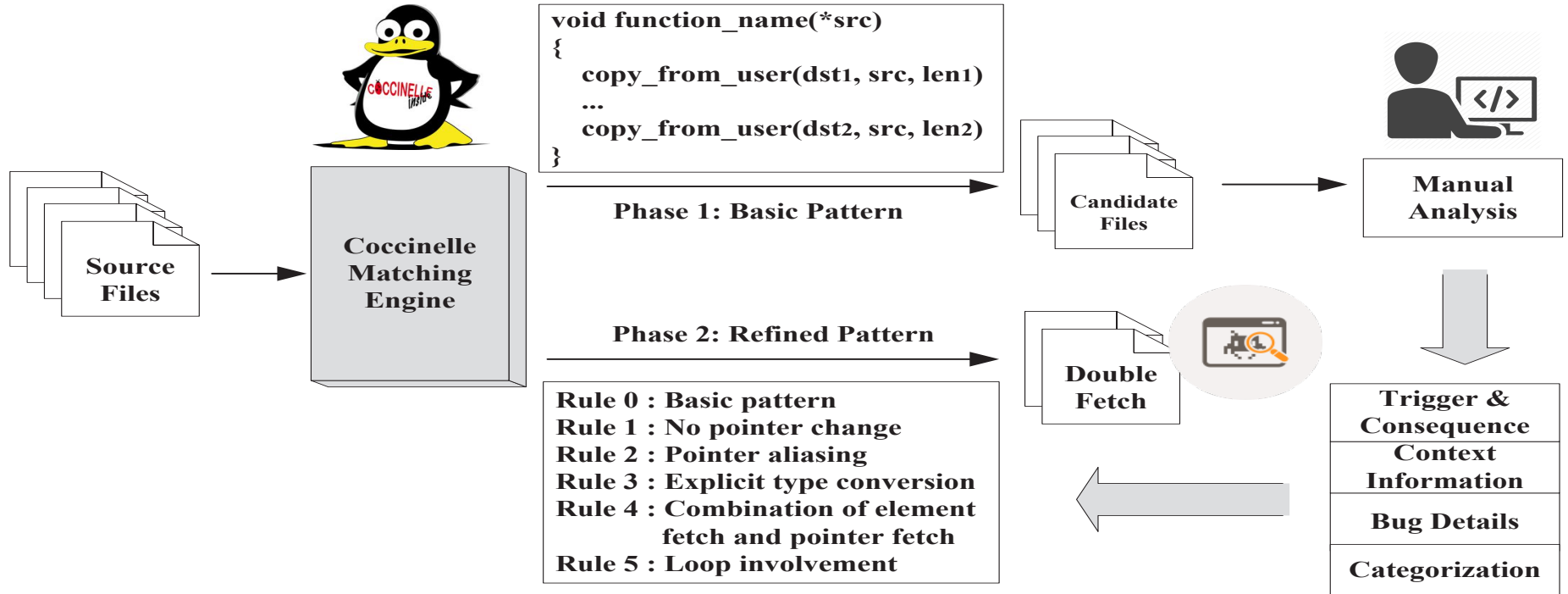
Program matching and transformation engine  
used for Linux checking

Developed two analyses:

1. A simple analysis to identify double-fetch situations
2. A refined analysis to discover double-fetch bugs



# Pattern-based Double Fetch Analysis



# Manual Analysis

## Characteristics

- How user data is transferred and used in the kernel
- Trigger and consequence

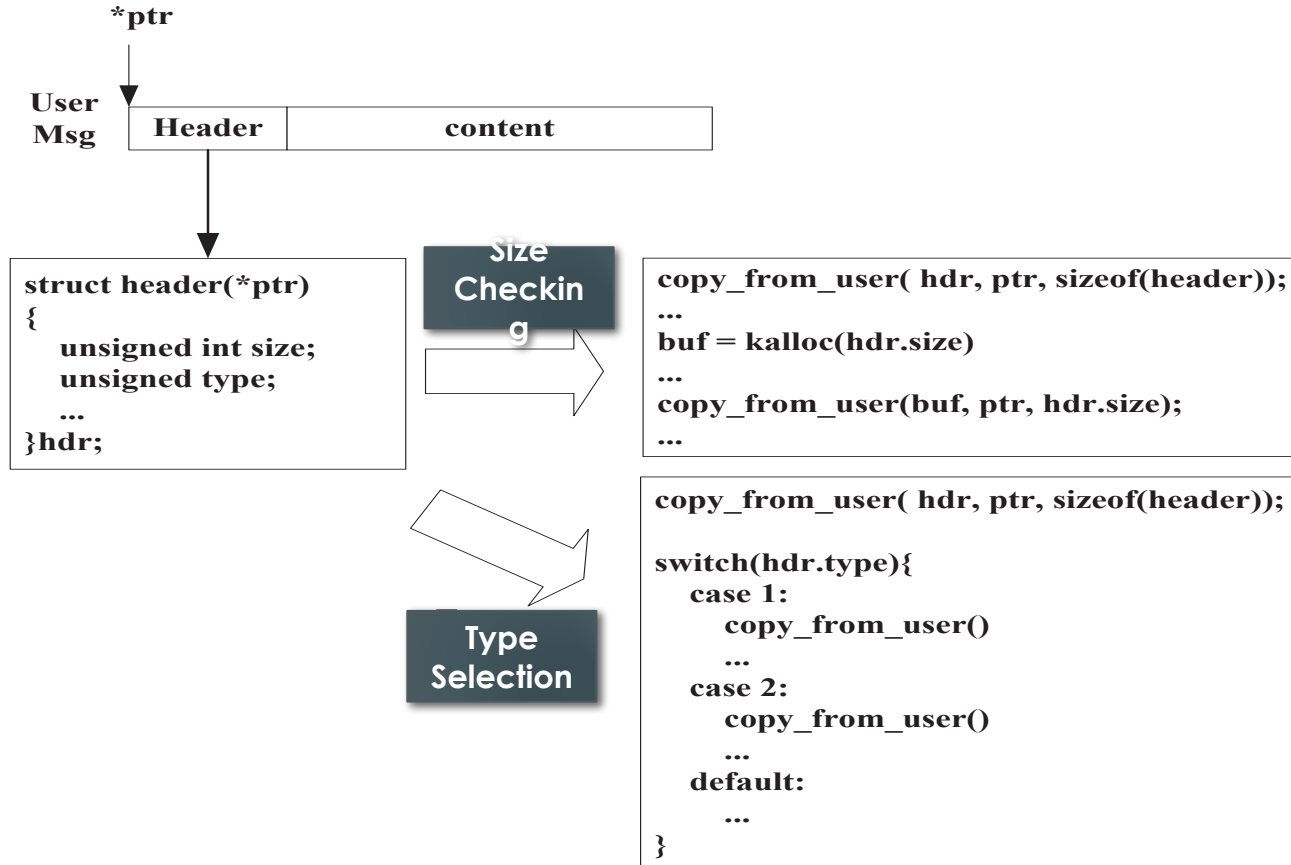
## Details at C code level

- Context information
- Implementation details
- Add rules to refine the pattern

## Categorization

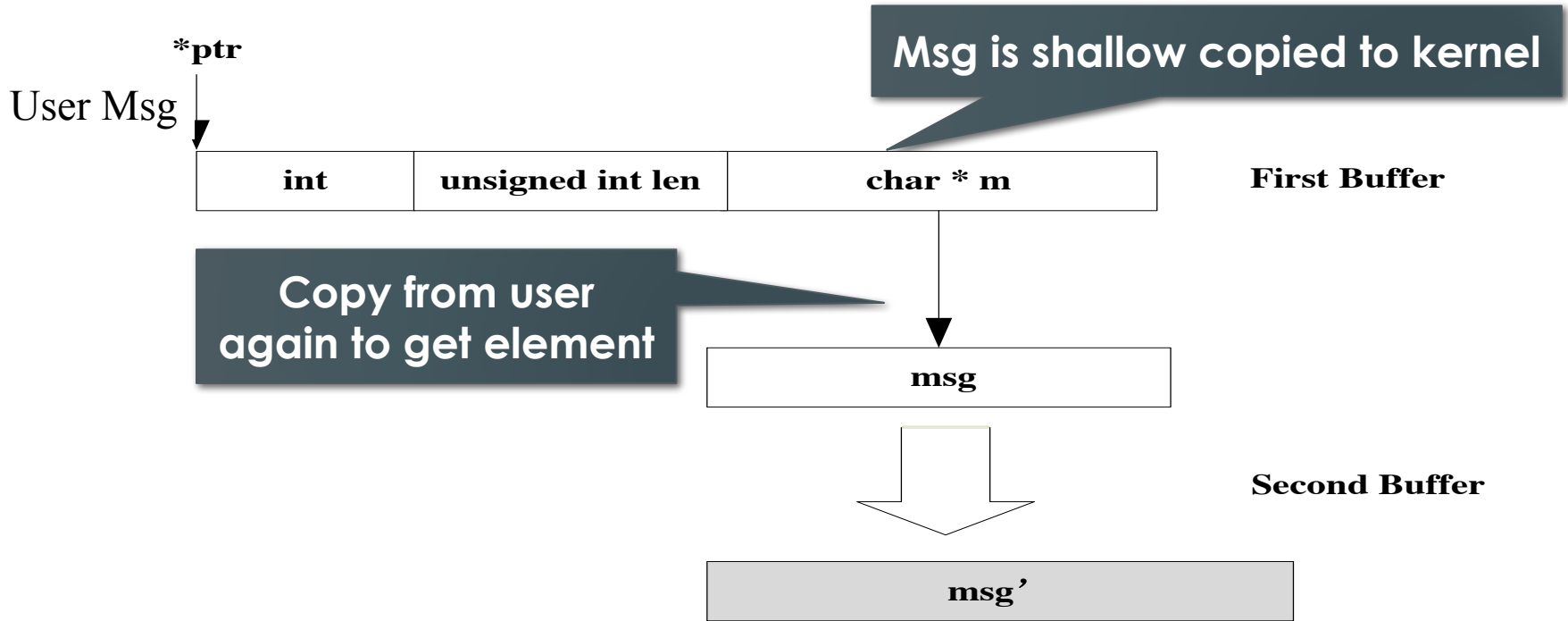
- Size Checking
- Type Selection
- Shallow Copy

# Categorization – Size Checking, Type Selection

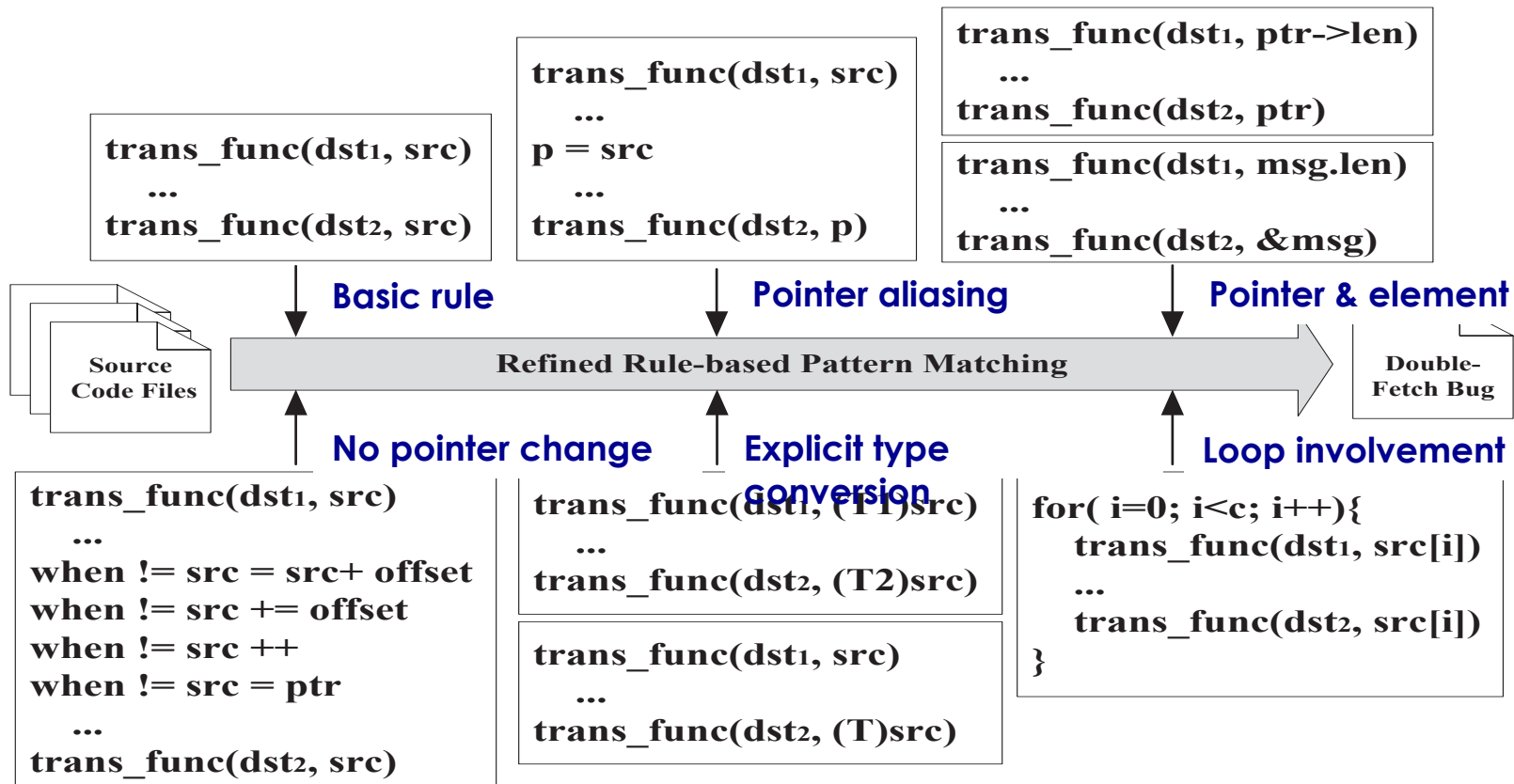




# Categorization – Shallow Copy



# Refined Double Fetch Detection



# Results and Findings

# Evaluation - Basic Double Fetch Analysis

Category	Occurrences		In Drivers	
Size Checking	30	33%	22	73%
Type Selection	11	12%	9	82%
Shallow Copy	31	34%	19	61%
Other	18	20%	7	39%
Total	90	100%	57	63%
True Bugs	5	6%	4	80%

- Most **double fetches** don't cause **double-fetch bugs**.
- Double fetches are more likely to **occur in drivers**.
  - About 63% (57 out of 90) of the cases were driver related.
  - About 80% (4 out of 5) of the true double-fetch bugs inside drivers.

# Evaluation – Refined Detection

Kernel	Total Files	Reported Files	True Bugs	Size Check.	Type Sel.
Linux 4.5	39,906	53	5	23	6
Android 6.0.1	35,313	48	3	18	6
FreeBSD	32,830	16	0	8	3

- **Totally 6 bugs found:**
  - 5 new bugs in newest Linux kernel 4.5.
  - 2 shared between Android and Linux.
  - 1 bug only showed in Android.
  - No bug found in FreeBSD.

# The Confirmed Bugs

**CVE-2016-5728**

- MIC VOP (Virtio Over PCIe) driver
- `Linux-4.5/drivers/misc/mic/host/mic_virtio.c`

**CVE-2016-6130**

- IBM (z-Series) s390 platform driver
- `Linux-4.5/drivers/s390/char/sclp_ctl.c`

**CVE-2016-6136**

- Auditing subsystem
- `Linux-4.5/kernel/auditsc.c`

**CVE-2016-6156**

- Expose the Chrome OS Embedded Controller to user-space
- `Linux-4.5/drivers/platform/chrome/cros_ec_dev.c`

**CVE-2016-6480**

- The aacraid driver (adds support for AdaptecRAID controllers)
- `Linux-4.5/drivers/scsi/aacraid/commctrl.c`

**CVE-2015-1420**

- File system
- `Android-6.0.1/fs/fhandle.c`

# Findings

## Double fetches have a long history

- Windows, Linux, Android, FreeBSD
- Some double-fetch bugs existed over 10 years (CVE-2016-6480).

## Some double fetches are inevitable

- Size checking, type selection, shallow copy
- Size checking is more likely to cause true bugs (5/6)

## Benign double fetches are not all safe

- Can turn into harmful ones by code update (CVE-2016-5728).
- Can cause performance issue.

# Conclusion

- **Double fetches occur in operating systems and can cause bugs and vulnerabilities.**
- **With a static pattern-matching analysis, we analyzed the complete kernel (all drivers) and categorized bug prone scenarios.**
- **We found 6 true bugs (vulnerabilities), all have been confirmed by the maintainers and patched already.**

Pengfei Wang      E-mail: [pfwang@nudt.edu.cn](mailto:pfwang@nudt.edu.cn)  
National University of Defense Technology, China  
Jens Krinke      E-mail: [j.krinke@ucl.ac.uk](mailto:j.krinke@ucl.ac.uk)  
University College London, UK



國防科學技術大學  
National University of Defense Technology

