Memory Leak 결함 자동 수정 기술 연구

홍성준, 이준희, 오학주
Korea University

29 August 2018 @Samsung Research

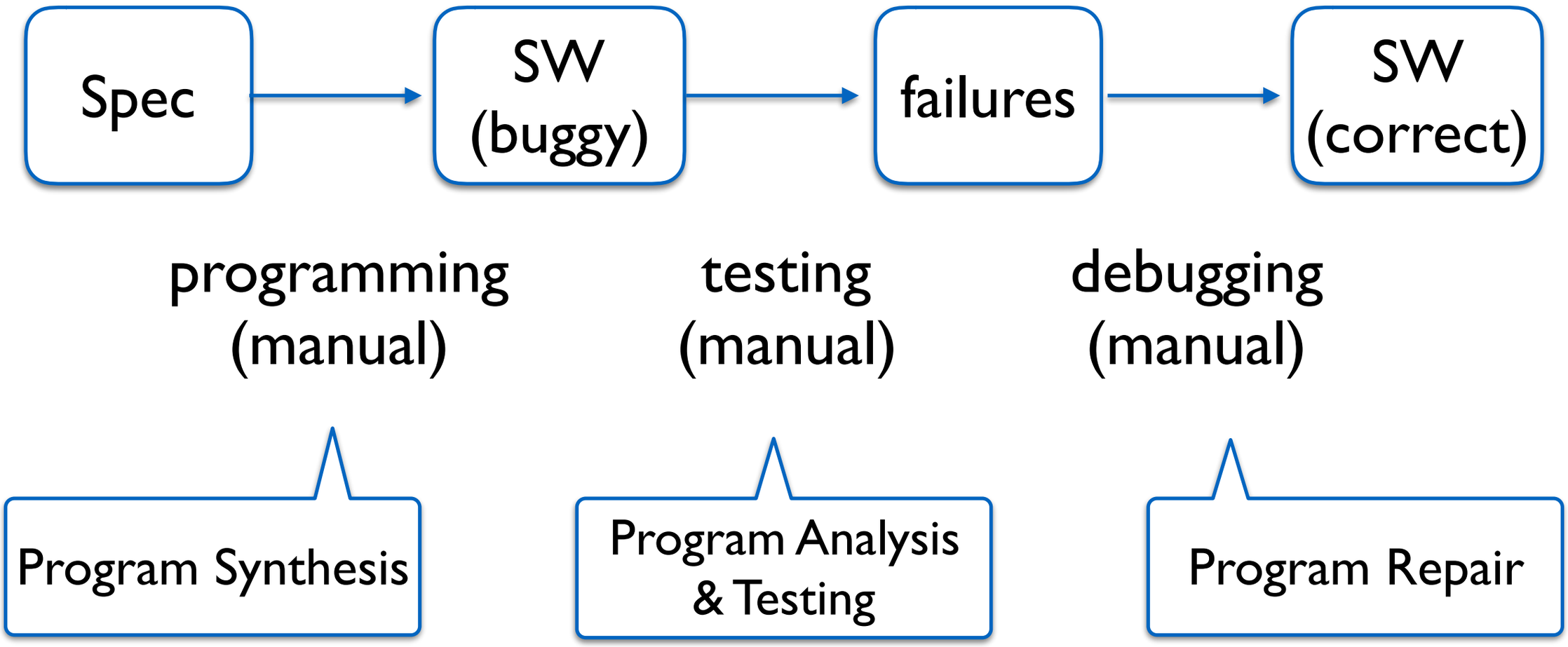
고려대학교 소프트웨어 분석 연구실

- Research areas: programming languages, software engineering, software security
  - program analysis and testing
  - program synthesis and repair
- Publication: top-venues in PL, SE, Security, and AI:
  - PLDI('12,'14), OOPSLA('15,'17,'17,'18,'18), TOPLAS('14,'16,'17,'18), ICSE('17,'18), FSE'18, ASE'18, S&P'17, IJCAI('17,'18), etc

http://prl.korea.ac.kr

Research Directions

- Automated programming / testing / debugging



연구 개발 개요

- C 프로그램의 Memory Leak 결함을 자동으로 수정
- 요구사항:
  1. 생성된 패치는 오류 수정 이외의 부분에서 프로그램의 의미를 바꾸면 안되고,
  2. 대상 결함을 반드시 제거해야 하고,
  3. 새로운 오류 (Double-Free / Use-After-Free 등)를 일으켜서는 안됨.
- 일정

연구 내용	일정 4	5	6	7	8	9	10	11	12
결함 자동 수정 기술 조사	■	■							
패치 후보를 모으는 분석기 디자인 및 개발			■	■	■				
올바른 후보를 찾는 알고리즘 디자인 및 개발						■	■		
구현 테스트 및 평가								■	■

Memory Leak 결함

- 메모리 관리 결함
  - Memory Leak: 할당한 메모리를 너무 늦게 해제
  - Use-After-Free: 너무 빨리 해제
  - Double-Free: 중복 해제

Memory Leak

```
p = malloc(1);
...
return;
```

Use-After-Free

```
p = malloc(1);
...
free(p);
...
use(p);
```

Double-Free

```
p = malloc(1);
...
free(p);
...
free(p);
```

Why Memory Leak?

- C/C++ 프로그램에서 매우 빈번하게 발생

Tizen

Subject	Status	Owner	Updated	Size
Fix memory leak		Semun Lee	9:31 AM	
Fix a memory leak		Sangyoon Jang	Jul 13	
Fix a memory leak	Merged	Sangyoon Jang	Jul 13	
memory leak of mutex removed		manoj gupta	Jul 13	
Fix double free issue (WGID-348564)	Merged	cheoleun moon	Jul 10	
Fix use-after-free bug issued by ASAN	Merged	Seungbae Shin	Jul 10	
[Tizen 5.0] omx memory leak in tv product because of null check		hyuntae kim	Jul 9	
ecore_drm: fix coverity issues	Merged	JeongHyun Kang	Jul 6	
Fix the memory leak	Merged	Yang	Jul 4	
Fix use-after-free bug for idxset free	Merged	Seungbae Shin	Jul 4	
Release version 0.8.9	Merged	HwanKyu Jhun	Jul 3	
Avoid double free of g_hash_iter_list GSList	Merged	Abhishek Sansanwal	Jul 3	
Fix memory leak	Merged	HwanKyu Jhun	Jul 2	
Fix use-after-free bug issued by ASAN	Merged	Seungbae Shin	Jul 2	
fixup! Enable GLRenderer. fixup! Adjust message ids to communicate with ...	Merged	Suyambulingam R M	Jun 29	
[CONPRO-1303] Memory leak for equal cred contents	Merged	Amit K Sharma	Jun 29	
Memory leak for equal cred contents	Merged	Amit K Sharma	Jun 29	
Memory leak for equal cred contents	Merged	Amit K Sharma	Jun 29	
Fix use-after-free bug for idxset free	Merged	Seungbae Shin	Jun 28	
Missing shader/program delete of fill_rectangles_shader - migration from ...	Merged	moonhee choi	Jun 23	
task-factory: fixed name, fixed memory leak	Merged	Michal Kolodziejski	Jun 22	
Combobox: Remove memory leak, unused variable & function & unreachable coc		Anil Kumar Nahak	Jun 22	
Release version 1.3.19	Merged	jusung son	Jun 20	
Fix memory leak	Merged	jusung son	Jun 19	
Remove handling for device found event in obex	Merged	Atul Rai	Jun 19	

Why Memory Leak?

- C/C++ 프로그램에서 매우 빈번하게 발생

Tizen

Subject	Status	Owner	Updated	Size
Resolve possibe Memory Leak This patch resolve posible memory leak.	Merged	Nilesh Trimbake	Jun 11	
cbhm_helper: fixed the memory leak for coverity	Merged	Taehyub Kim	Jun 11	
efl_ui_win: fix double free bug related with fake window	Merged	Jiyoun Park	Jun 11	
Fixed double free problem for hashes of tizen_policy and display_policy.	Merged	Gwanglim Lee	Jun 5	
Fixed double free problem for hashes of tizen_policy and display_policy.	Merged	Gwanglim Lee	Jun 5	
Release version 0.12.11	Merged	Junghyun Yeon	Jun 5	
[Coverity] Fix the Double free issue	Merged	moonhee choi	Jun 1	
Fix Memory Leak - missing free on string buffer	Merged	moonhee choi	Jun 1	
[Coverity] Fix the Double free issue	Abandoned	moonhee choi	Jun 1	
fix memory leak	Merged	Youngjae Shin	Jun 1	
gadget: f_thor: Fix memory leaks of usb request and its buffer	Merged	Seung-Woo Kim	Jun 1	
Release version 0.4.7	Merged	Junghyun Yeon	May 31	
[M63 Migration][NaCl] Antialiasing for NaCl Graphics 3D and Compositor	Merged	zhu yong	May 30	
Sync-up with Tizen branch	Merged	saerome kim	May 30	
Release version 1.3.1	Merged	jusung son	May 30	
Release version 1.3.1	Merged	jusung son	May 29	
Fix double free	Merged	jusung son	May 29	
Fix double free	Merged	jusung son	May 29	
Fix memory leak reported by coverity	Abandoned	Lukasz Stelmach	May 25	
[Rendering] Prevent double free of MailboxManager unique pointer	Merged	Chandan Padhi	May 22	
Release version 1.0.6	Merged	MyungKi Lee	May 21	
Fix memory leak	Merged	Hyunho Kang	May 21	
efl_ui_win: fix double free bug related with fake window	Merged	Jiyoun Park	May 18	
Fix Memory leak	Merged	Nagaraj D R	May 18	
Release version 1.3.18	Merged	jusung son	May 18	

Why Memory Leak?

- C/C++ 프로그램에서 매우 빈번하게 발생

Tizen

Subject	Status	Owner	Updated	Size
Handled the memory leak	Merged	Mayank Haarit	May 17	
Fix double free	Merged	jusung son	May 16	
Fix memory leak	Merged	Vyacheslav Cherkashin	May 16	
Release version 1.0.4	Merged	Hyunho Kang	May 16	
Fix heap-use-after-free error detected by Adress Sanitizer.	Merged	Lukasz Stelmach	May 16	
Fix memory leak	Merged	MyungKi Lee	May 15	
[MM][TTS] Removed EWK_BRINGUP flag for TTS	Merged	joseph lolak	May 14	
Release version 1.0.2	Merged	Hyunho Kang	May 14	
Fix coverity issues	Merged	seolheui kim	May 9	
evas_out : fix a memory leak.	Abandoned	junsu choi	May 9	
Fix Bundle memory leak	Merged	Hyunho Kang	May 9	
genlist: prevent memory leak in item class update	Merged	SangHyeon Lee	May 8	
genlist: prevent memory leak in item class update	Merged	SangHyeon Lee	May 8	
[M63 Migration] Fix memory leak for evas and ecore event register	Merged	chen shurong	May 7	
Fix heap-use-after-free error detected by Adress Sanitizer.	Merged	Elmurod Talipov	May 4	
cbhm_helper: fixed the memory leak for coverity	Merged	Taehyub Kim	May 4	
Fix memory leak	Merged	MyungKi Lee	May 4	
Fix memory leak in app control	Merged	Chang Joo Lee	May 3	
Fix memory leak issue	Merged	Jihoon Kim	May 2	

1,700 error fixing commits in 3 years

Why Memory Leak?

- C/C++ 프로그램에서 매우 빈번하게 발생

Tizen

Subject	Status	Owner	Updated	Size
Handled the memory leak	Merged	Mayank Haarit	May 17	
Fix double free	Merged	jusung son	May 16	
Fix memory leak	Merged	Vyacheslav Cherkashin	May 16	
Release version 1.0.4	Merged	Hyunho Kang	May 16	
Fix heap-use-after-free error detected by Adress Sanitizer.	Merged	Lukasz Stelmach	May 16	
Fix memory leak	Merged	MyungKi Lee	May 15	
[MM][TTS] Removed EWK_BRINGUP flag for TTS	Merged	joseph lolak	May 14	
Release version 1.0.2	Merged	Hyunho Kang	May 14	
Fix coverity issues	Merged	seolheui kim	May 9	
evas_out : fix a memory leak.	Abandoned	junsu choi	May 9	
Fix Bundle memory leak	Merged	Hyunho Kang	May 9	
genlist: prevent memory leak in item class update	Merged	SangHyeon Lee	May 8	
genlist: prevent memory leak in item class update	Merged	SangHyeon Lee	May 8	
[M63 Migration] Fix memory leak for evas and ecore event register	Merged	chen shurong	May 7	
Fix heap-use-after-free error detected by Adress Sanitizer.	Merged	Elmurod Talipov	May 4	
cbhm_helper: fixed the memory leak for coverity	Merged	Taehyub Kim	May 4	
Fix memory leak	Merged	MyungKi Lee	May 4	
Fix memory leak in app control	Merged	Chang Joo Lee	May 3	
Fix memory leak issue	Merged	Jihoon Kim	May 2	

1,700 error fixing commits in 3 years

Open SW

Repository	#commits	#memory leaks	#buffer/integer-overflows
php	105,613	1,129	649
git	49,475	350	258
openssl	21,009	220	61

Why Memory Leak?

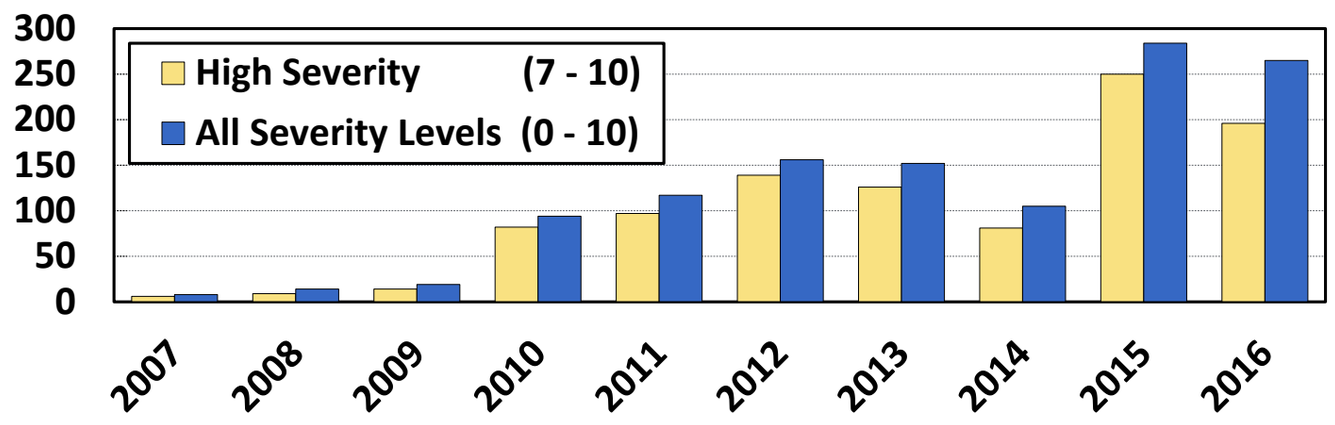
- 심각한 보안 취약점의 원인
  - CWE-401 (Memory Leak), CWE-415 (Double-Free), CWE-416 (Use-After-Free)

CVE-2017-10810

English

The MITRE CVE dictionary describes this issue as:

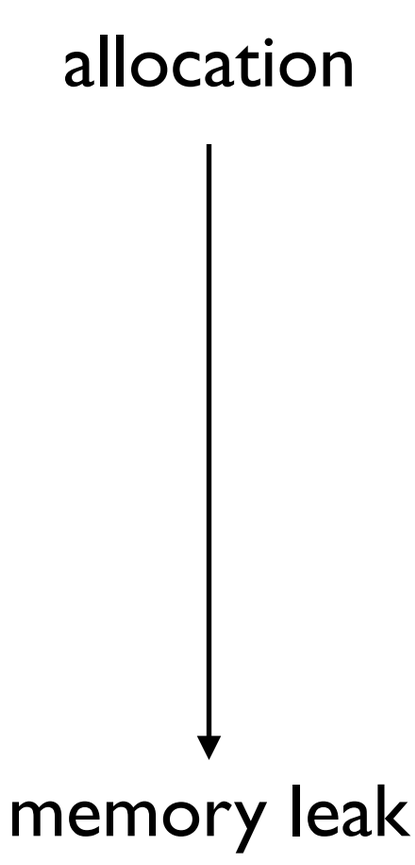
Memory leak in the virtio_gpu_object_create function in drivers/gpu/drm/virtio/virtgpu_object.c in the Linux kernel through 4.11.8 allows attackers to cause a denial of service (memory consumption) by triggering object-initialization failures.



(Yan et al. ICSE 2018)

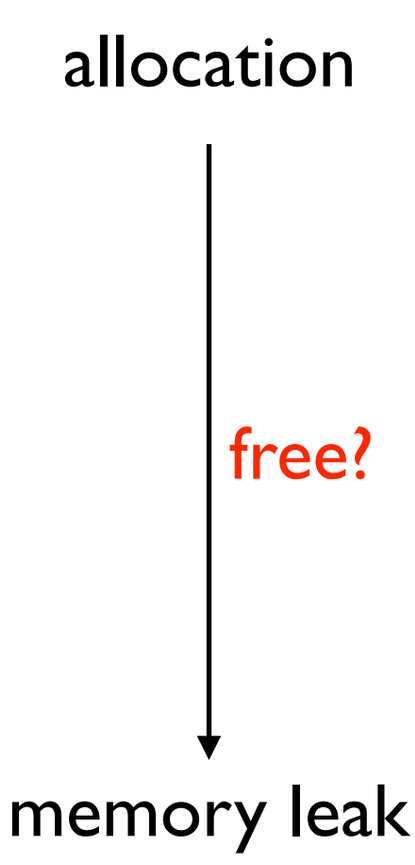
Why Memory Leak?

- 정확하게 수정하기가 매우 어려움
- 잘못 수정하면 더욱 심각한 문제(UAF, DF) 발생



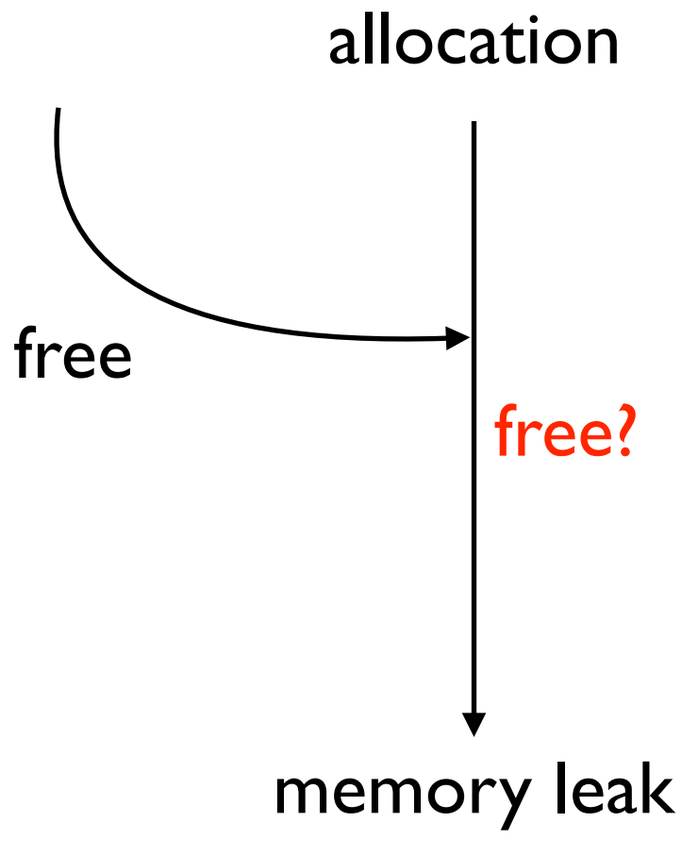
Why Memory Leak?

- 정확하게 수정하기가 매우 어려움
- 잘못 수정하면 더욱 심각한 문제(UAF, DF) 발생



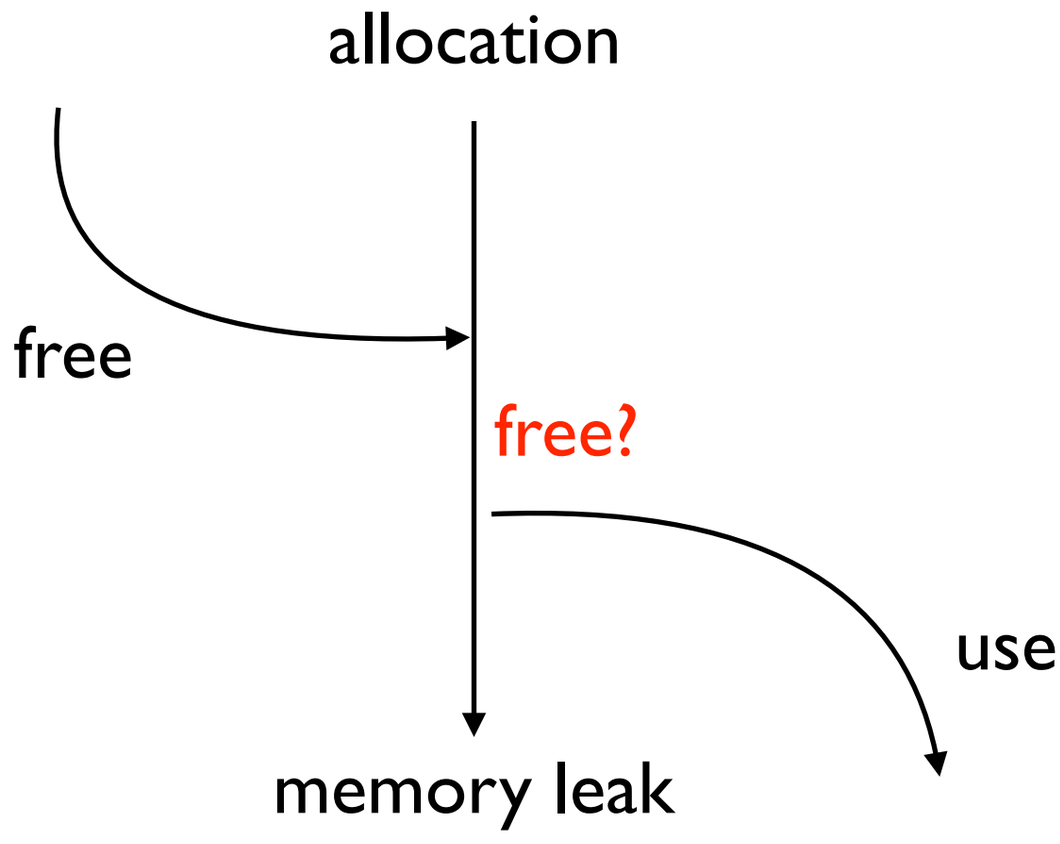
Why Memory Leak?

- 정확하게 수정하기가 매우 어려움
- 잘못 수정하면 더욱 심각한 문제(UAF, DF) 발생



Why Memory Leak?

- 정확하게 수정하기가 매우 어려움
- 잘못 수정하면 더욱 심각한 문제(UAF, DF) 발생



Example (Linux Kernel)

```
in = malloc(1);
out = malloc(1);
... // use in, out
free(out);
free(in);

in = malloc(2);
if (in == NULL) {

  goto err;
}

out = malloc(2);
if (out == NULL) {
  free(in);

  goto err;
}
... // use in, out
err:
  free(in);
  free(out);
  return;
```

Example (Linux Kernel)

```
in = malloc(1);
out = malloc(1);
... // use in, out
free(out);
free(in);

in = malloc(2);
if (in == NULL) {
  goto err;
}

out = malloc(2);
if (out == NULL) {
  free(in);

  goto err;
}
... // use in, out
err:
  free(in);
  free(out);
  return;
```

double-free

Example (Linux Kernel)

```
in = malloc(1);
out = malloc(1);
... // use in, out
free(out);
free(in);

in = malloc(2);
if (in == NULL) {

  goto err;
}

out = malloc(2);
if (out == NULL) {
  free(in);

  goto err;
}
... // use in, out
err:
  free(in);
  free(out);
  return;
```

Example (Linux Kernel)

USB: fix double frees in error code paths of ipaq driver

the error code paths can be enter with buffers to freed buffers.
Serial core would do a kfree() on memory already freed.

Signed-off-by: Oliver Neukum <oneukum@suse.de>
Signed-off-by: Greg Kroah-Hartman <gregkh@suse.de>

master v4.15-rc1 ... v2.6.24-rc1

Oliver Neukum committed with **gregkh** on 18 Sep 2007 1 par

```
in = malloc(1);
out = malloc(1);
... // use in, out
free(out);
free(in);

in = malloc(2);
if (in == NULL) {
  out = NULL;
  goto err;
}

out = malloc(2);
if (out == NULL) {
  free(in);
  in = NULL;
  goto err;
}
... // use in, out
err:
  free(in);
  free(out);
  return;
```

Example (Linux Kernel)

USB: fix double kfree in ipaq in error case

in the error case the ipaq driver leaves a dangling pointer to already freed memory that will be freed again.

Signed-off-by: Oliver Neukum <oneukum@suse.de>
Signed-off-by: Greg Kroah-Hartman <gregkh@suse.de>

master v4.15-rc1 ... v2.6.27-rc1

Oliver Neukum committed with **gregkh** on 30 Jun 2008 1 parent 35

```
in = malloc(1);
out = malloc(1);
... // use in, out
// removed
free(in);

in = malloc(2);
if (in == NULL) {
  out = NULL;
  goto err;
}
free(out);
out = malloc(2);
if (out == NULL) {
  free(in);
  in = NULL;
  goto err;
}
... // use in, out
err:
  free(in);
  free(out);
  return;
```

Example (Linux Kernel)

USB: fix double kfree in ipaq in error case

in the error case the ipaq driver leaves a dangling pointer to already freed memory that will be freed again.

Signed-off-by: Oliver Neukum <oneukum@suse.de>
Signed-off-by: Greg Kroah-Hartman <gregkh@suse.de>

master v4.15-rc1 ... v2.6.27-rc1

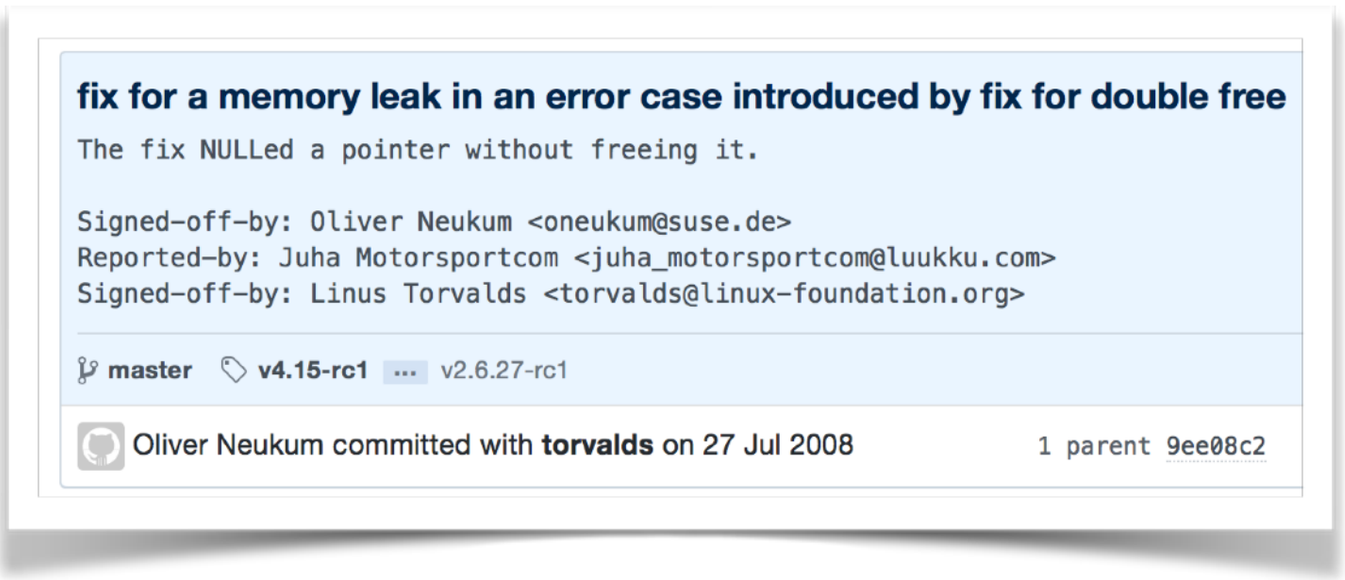
Oliver Neukum committed with **gregkh** on 30 Jun 2008 1 parent 35

memory leak

```
in = malloc(1);
out = malloc(1);
... // use in, out
// removed
free(in);

in = malloc(2);
if (in == NULL) {
  out = NULL;
  goto err;
}
free(out);
out = malloc(2);
if (out == NULL) {
  free(in);
  in = NULL;
  goto err;
}
... // use in, out
err:
  free(in);
  free(out);
  return;
```

Example (Linux Kernel)




```
in = malloc(1);
out = malloc(1);
... // use in, out
free(out);
free(in);
out = NULL;
in = malloc(2);
if (in == NULL) {
  out = NULL;
  goto err;
}
// removed
out = malloc(2);
if (out == NULL) {
  free(in);
  in = NULL;
  goto err;
}
... // use in, out
err:
  free(in);
  free(out);
  return;
```

목표 도구

```
in = malloc(1);
out = malloc(1);
... // use in, out
free(out);
free(in);

in = malloc(2);
if (in == NULL) {
  goto err;
}

out = malloc(2);
if (out == NULL) {
  free(in);

  goto err;
}
... // use in, out
err:
  free(in);
  free(out);
  return;
```

Our Tool

```
in = malloc(1);
out = malloc(1);
... // use in, out
// removed
free(in);

in = malloc(2);
if (in == NULL) {
  goto err;
}
free(out);
out = malloc(2);
if (out == NULL) {
  // removed

  goto err;
}
... // use in, out
err:
  free(in);
  free(out);
  return;
```

기존 결함 수정 기술들

- 범용 결함 수정 기술:
  - GenProg, PAR, SPR, Prophet, SketchFix, SemFix, etc
    - 일반적인 오류를 수정 가능하지만 Memory Leak오류 수정에는 적합하지 않음
    - 요구사항 (1), (2), (3) 위배 가능

기존 결함 수정 기술들

- 메모리 결함 특화 기술:
  - LeakFix [ICSE'15], FootPatch [ICSE'18]
  - 결과를 믿을 수 없거나, 새로운 오류를 유발 가능
  - 요구 사항 (2), (3) 위배 가능

```
1 p = malloc();
2 if(...) {
3 q = malloc();
4
5 } else
6 q = p;
7 ... // use q
8 free(p);
9 // M.L.
```

(a) 메모리 누수 코드

```
1 p = malloc();
2 if(...) {
3 q = malloc();
4
5 } else
6 q = p;
7 ... // use q
8 free(p);
9 free(q); // D.F.
```

(b) 잘못 수정된 코드

```
1 p = malloc();
2 if(...) {
3 q = malloc();
4 free(p); // +
5 } else
6 q = p;
7 ... // use q
8 // -
9 free(q); // +.
```

(c) 올바르게 수정된 코드

Memory Leak 수정 알고리즘

- 정적 분석을 이용하여 패치가 만족해야 하는 제약식 생성

Approx. by Static Aanalysis

```
1  p = malloc(1); /
2  if (...) {
3    q = malloc(1);
4
5  }
6  else
7     q = p;
8  ... = *q; // use
9  free(q);
10 free(p);
```

Buggy Program

	τ_1	τ_2	τ_3
$(3, \mathbf{p})$	0	**1**	0
$(8, p)$	0	1	1
$(8, \mathbf{q})$	**1**	0	**1**

Exact Cover Problem

SAT Solver

```
1  p = malloc(1);
2  if (...) {
3    q = malloc(1);
4    free(p); // +
5  }
6  else
7     q = p;
8  ... = *q;
9  free(q);
10 // -
```

Patched Program

정적 분석

- 할당된 객체 별로 아래 정보를 계산

반드시 가리키는 포인터들

안전하게 해제할 수 있는 해제문들

$\langle o, must, mustNot, patch, patchNot \rangle$

객체 할당지점 (allocation-site)

반드시 가리키지 않는 포인터들

안전하지 않은 해제문들

예제

```
1  p = malloc(1); // o1
2  if (...) {
3    q = malloc(1); // o2
4
5  }
6  else
7     q = p;
8  ... = *q; // use q
```

$< o_1, \{p\}, \emptyset, \{(1,p)\}, \emptyset >$

$< o_1, \{p\}, \{q\}, \{(1,p),(3,p)\}, \emptyset >$

$< o_2, \{q\}, \emptyset, \{(3,q)\}, \emptyset >$

$< o_1, \{p,q\}, \emptyset, \{(1,p),(7,p),(7,q)\}, \emptyset\}$

$\tau_1 = < o_1, \{q\}, \emptyset, \{(8,q)\}, \{(3,q)\} >$

$\tau_2 = < o_1, \{p\}, \{q\}, \{(1,p),(3,p),(8,p)\}, \emptyset >$

$\tau_3 = < o_1, \{p,q\}, \emptyset, \{(8,p),(8,q)\}, \{(1,p),(7,p),(7,q)\} >$

제약식 생성

- 최종 분석 결과

$$\tau_1 = < o_1, \{q\}, \emptyset, \{(8, q)\}, \{(3, q)\} >$$
$$\tau_2 = < o_1, \{p\}, \{q\}, \{(1, p), (3, p), (8, p)\}, \emptyset >$$
$$\tau_3 = < o_1, \{p, q\}, \emptyset, \{(8, p), (8, q)\}, \{(1, p), (7, p), (7, q)\} >$$

- Exact cover problem

Safe: {(1, p), (3, p), (8, p), (8, q)}

Unsafe: {(1, p), (3, q), (7, p), (7, q)}

Cand: {(3, p), (8, p), (8, q)}

	τ_1	τ_2	τ_3
$(3, \mathbf{p})$	0	**1**	0
$(8, p)$	0	1	1
$(8, \mathbf{q})$	**1**	0	**1**

제약식 생성

- 최종 분석 결과

$$\tau_1 = < o_1, \{q\}, \emptyset, \{(8, q)\}, \{(3, q)\} >$$
$$\tau_2 = < o_1, \{p\}, \{q\}, \{(1, p), (3, p), (8, p)\}, \emptyset >$$
$$\tau_3 = < o_1, \{p, q\}, \emptyset, \{(8, p), (8, q)\}, \{(1, p), (7, p), (7, q)\} >$$

- Exact cover problem

Safe: {(1, p), (3, p), (8, p), (8, q)}

Unsafe: {(1, p), (3, q), (7, p), (7, q)}

Cand: {(3, p), (8, p), (8, q)}

	τ_1	τ_2	τ_3
$(3, \mathbf{p})$	0	**1**	0
$(8, p)$	0	1	1
$(8, \mathbf{q})$	**1**	0	**1**

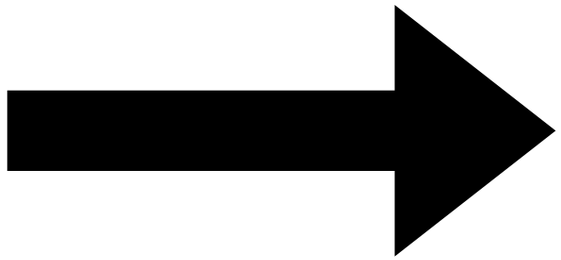
Linux Example

```
in = malloc(1);
out = malloc(1);
... // use in, out
free(out);
free(in);

in = malloc(2);
if (in == NULL) {
  goto err;
}

out = malloc(2);
if (out == NULL) {
  free(in);

  goto err;
}
... // use in, out
err:
  free(in);
  free(out);
  return;
```



Applying a patch:

(4, in), (12, out), (21, in), (21, out)

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6
(4, in)	1	0	0	0	0	0
(4, out)	0	1	1	0	0	0
(9, out)	0	0	1	0	0	0
(12, out)	0	1	0	0	0	0
(15, in)	0	0	0	0	1	0
(19, out)	0	0	0	1	0	0
(19, out)	0	0	0	0	0	1
(21, in)	0	0	0	1	1	0
(21, out)	0	0	1	0	0	1

Generated by static analysis

```
1  in = malloc(1);
2  out = malloc(1);
3  ...
4  // -
5  free(in);
6
7  in = malloc(2);
8  if (in == NULL) {
9
10   goto err;
11 }
12 free(out); // +
13 out = malloc(2);
14 if (out == NULL) {
15   // -
16
17   goto err;
18 }
19 ...
20 err:
21   free(in);
22   free(out);
23   return;
```

중간 보고서

Contents

1 개요 **3**

1.1 연구개발 목적 3
1.2 연구범위 3
1.3 요구사항 3
1.4 연구개발 현황 3

2 소프트웨어 자동 수정 기술 조사 **4**

2.1 범용 결함 수정 기술 4
2.1.1 시행착오 기반 방식 4
2.1.2 제약식 기반 방식 7
2.2 메모리 누수 결함 수정 기술 11
2.3 기존 기술의 한계 11

3 Memory Leak 자동 수정 기술 **13**

3.1 기술 개요 13
3.2 분석기 디자인 15
3.2.1 Language 15
3.2.2 Abstract Domain 15
3.2.3 Abstract Semantics 15
3.3 분석기 구현 17
3.3.1 Must-Alias Analysis 18
3.3.2 Partial Patch 19

최종 보고서 (예정)

- 제약식 생성 알고리즘
- 구현 테스트 및 평가
  - GNU Coreutils 대상 30%이상의 패치 생성률 / 90% 이상 정확도

Summary

- Memory Leak 결함을 자동으로 올바르게 수정하는 기법 개발
- 알고리즘 설계 완료 및 프로토타입 구현 중

감사합니다!