Bug1.c

(1). What kind of memory deallocation error is there in the source code? (0.5p)

1. memory leak 2. double free 3. use-after free

(2). What is the root cause of the memory deallocation error in the source code? (0.5p)

```
HEAP SUMMARY:
    in use at exit: 9 bytes in 1 blocks
  total heap usage: 3 allocs, 2 frees, 1,042 bytes allocated

9 bytes in 1 blocks are definitely lost in loss record 1 of 1
   at 0x4C31B0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
   by 0x108D89: match_filter_list (in /home/nicholasbear/Desktop/HW2/bug_1.out)
   by 0x108E87: main (in /home/nicholasbear/Desktop/HW2/bug_1.out)

LEAK SUMMARY:
   definitely lost: 9 bytes in 1 blocks
   indirectly lost: 0 bytes in 0 blocks
     possibly lost: 0 bytes in 0 blocks
   still reachable: 0 bytes in 0 blocks
        suppressed: 0 bytes in 0 blocks

For counts of detected and suppressed errors, rerun with: -v
ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

Main 함수의 match_filter_list 에서 malloc된 것이 free가 되지 않음

(3). What is the minimum correct patch for fixing the memory deallocation error? (1p)

```
int main()
{
    char *proposal = match_filter_list("proposal", "filter");
    if(proposal == NULL)
        printf("match_filter_list failed\n");
    printf("proposal : %s\n", proposal);
    free(proposal);
    return 0;
}
```

free(proposal) 추가

Bug2.c

(1). What kind of memory deallocation error is there in the source code? (0.5p)

1. memory leak 2. double free **3. use-after free**

(2). What is the root cause of the memory deallocation error in the source code? (0.5p)

```
if (cd->int_insn_p)
  {
    buf = (unsigned char *) xmalloc (4);
    cgen_put_insn_value (cd, buf, length, insn_int_value);
    base_insn = insn_int_value;
    free (buf);
  }
else
  {
    buf = insn_bytes_value;
    base_insn = cgen_get_insn_value (cd, buf, length);
  }

if (!insn)
  {
    const CGEN_INSN_LIST *insn_list;

    insn_list = cgen_dis_lookup_insn (cd, (char *) buf, base_insn);
    while (insn_list != NULL)
  {
      insn = insn_list->insn;
      insn_list = insn_list->next;
  }
  }
```

첫번째 if문에서 free를 하고 나서 두번째 if 문에서 buf를 쓰게 된다

(3). What is the minimum correct patch for fixing the memory deallocation error? (1p)

```
    insn_list = cgen_dis_lookup_insn (cd, (char *) buf, base_insn);
    while (insn_list != NULL)
  {
      insn = insn_list->insn;
      insn_list = insn_list->next;
  }
  }

free (buf);
return NULL;
```

Free(buf)의 위치를 return 하기전에 넣어준다

Bug3.c

(1). What kind of memory deallocation error is there in the source code? (0.5p)

1. memory leak 2. double free 3. use-after free

(2). What is the root cause of the memory deallocation error in the source code? (0.5p)

Cmd 변수가 free가 안되는 경우가 있다

(3). What is the minimum correct patch for fixing the memory deallocation error? (1p)

```
298         } else {
299             if (channel_request_remote_forwarding(fwd.listen_host,
300                 fwd.listen_port, fwd.connect_host,
301                 fwd.connect_port) < 0) {
302                 logit("Port forwarding failed.");
303                 goto out;
304             }
305         }
306
307         logit("Forwarding port.");
308     }
309
310     goto out;
```

맨마지막에 goto ou;을 해줘서 cmd 변수를 free 를 해줘야한다.

Bug4.c

(1). What kind of memory deallocation error is there in the source code? (0.5p)

1. memory leak 2. double free 3. use-after free

(2). What is the root cause of the memory deallocation error in the source code? (0.5p)

```
array = xcalloc(num_head, sizeof(*array));
for (p = heads, i = 0; p; p = p->next) {
    if (p->item->object.flags & STALE) {
        array[i++] = p->item;
        p->item->object.flags &= ~STALE;
    }
}
num_head = remove_redundant(array, num_head);
for (i = 0; i < num_head; i++)
    tail = &commit_list_insert(array[i], tail)->next;

return result;
```

Array가 free가 되지 않음

(3). What is the minimum correct patch for fixing the memory deallocation error? (1p)

```
array = xcalloc(num_head, sizeof(*array));
for (p = heads, i = 0; p; p = p->next) {
    if (p->item->object.flags & STALE) {
        array[i++] = p->item;
        p->item->object.flags &= ~STALE;
    }
}
num_head = remove_redundant(array, num_head);
for (i = 0; i < num_head; i++)
    tail = &commit_list_insert(array[i], tail)->next;

free(array);
return result;
```

free(array) 추가

Bug5.c

(1). What kind of memory deallocation error is there in the source code? (0.5p)

1. memory leak **2. double free** 3. use-after free

(2). What is the root cause of the memory deallocation error in the source code? (0.5p)

```
if (start_active_slot(slot)) {
    run_active_slot(slot);
    free(url);
    if (results.http_code == 404)
        ret = 0;
    else if (results.curl_result == CURLE_OK)
        ret = 1;
    else
        fprintf(stderr, "HEAD HTTP error %ld\n", results.http_code);
} else {
    free(url);
    fprintf(stderr, "Unable to start HEAD request\n");
}

free(url);
```

Free 가 2번됐다

If else문에서 한번 마지막에서 한번

(3). What is the minimum correct patch for fixing the memory deallocation error? (1p)

```
if (start_active_slot(slot)) {
    run_active_slot(slot);
    free(url);
    if (results.http_code == 404)
        ret = 0;
    else if (results.curl_result == CURLE_OK)
        ret = 1;
    else
        fprintf(stderr, "HEAD HTTP error %ld\n", results.http_code);
} else {
    free(url);
    fprintf(stderr, "Unable to start HEAD request\n");
}

return ret;
```

마지막 free를 지운다