Mining Software Repositories for Accurate Authorship

Dyninst

University of Maryland

University of Wisconsin–Madison

Current Repositories Record Line-Level Authorship

- Git, svn, and mercurial only report the most recent author to change a line of code
- A line of code may be modified by several developers through the development history
- We want to extract the whole history of a line to produce accurate authorship

Challenges

- There are noisy changes to code that should not be attributed for authorship
- Current version control systems represent changes as additions and deletions
- Branching and merging makes the relationship between commits complex
- Current version control systems do not extract history of a line of code

```
git author -L 440,449 -n dyninstAPI/src/BPatch_process.C
  440: (bd8b6a3,bernat,319) (88fbc08,James Waskiewicz,423) (fd7c419,Daniel McNulty,426)
  441: (88fbc08,James Waskiewicz,424) (fd7c419,Daniel McNulty,427)
  442: (bd8b6a3,bernat,320) (88fbc08,James Waskiewicz,425) (fd7c419,Daniel McNulty,428)
  443: (88fbc08,James Waskiewicz,426) (fd7c419,Daniel McNulty,429)
  444: (bd8b6a3,bernat,321) (fd7c419,Daniel McNulty,430)
  445: (bd8b6a3,bernat,322) (fd7c419,Daniel McNulty,431)
  446: (88fbc08,James Waskiewicz,429) (fd7c419,Daniel McNulty,432)
  447: (bd8b6a3,bernat,323) (fd7c419,Daniel McNulty,433)
  448: (bd8b6a3,bernat,324) (fd7c419,Daniel McNulty,434)
  449: (bd8b6a3,bernat,325) (fd7c419,Daniel McNulty,435)
```

Commit 6
Author: Jim

```
@@ -20,7 +20,7 @@
static voi
__bad_are
    struc
    unsig
- unsigne
+ unsigne
si_code)

@@ -39,7
__bad_are
   (struc
    unsig

+tsk->thr
+tsk->thr
+    erro
+    (add

-  tsk->t
+  tsk->t
```

Commit 4
Author: Bob

```
@@ -2,7 +2,6 @@

-     if (regs == NULL)
-     warning("Null Poi

@@ -14,16 +13,24 @@
-  if (address == 0)
-     warning("Zero add
+  if (is_prefetch(reg
error_code, address))
+      return;
+  if (is_errata100(
+          regs, addres
+      return;

@@ -31,8 +38,8 @@

-   if (is_f00f_bug(address))
+   if (is_f00f_bug(regs,
+                   address))
          return;
```

Commit 7
Author: Tom

```
@@ -1,5 +1,5 @@

static inline void
-  struct pt_regs *regs,
+  struct pt_regs *regs,
unsigned long error_code,

@@ -8,10 +8,10 @@
-  printk("%s%s[%d]: segfault
at %lx ip %p sp %p",

task_pid_nr(tsk) > 1 ?
KERN_INFO : KERN_EMERG,
tsk->comm, task_pid_nr(tsk),
address,

-  (void *)regs->ip, (void
*)regs->sp);
+  (void *)regs->ip, (void
*)regs->sp, error_code);
```

Commit 3
Author: Ji

```
@@ -1,10 +1,16 @@

static inline void
show_signal_msg(struct pt_regs
*regs,
-   unsigned long address)
+   unsigned long address,
struct task_struct *tsk)
 {
-  printk("%s segfault at %lx
-  KERN_INFO
-  address,

+  if (!unhandled_signal(tsk))
+     return;
+
+  if (!printk_ratelimit())
+     return;
+  printk("%s%s[%d]: segfault
at %lx ip %p sp %p",
+  tsk->comm,task_pid_nr(tsk),
```

Commit 2
Author: Tom

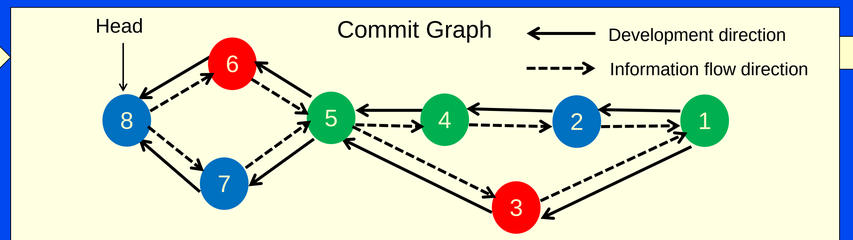
```
c void
_area_nosemaphore
igned long error_code,

 (address == 0)
g("Zero address");
 (error_code & PF_USER){
local_irq_enable();

tsk->thread.cr2=address;
tsk->thread.error_code
r_code | (address >=
IZE);
tsk->thread.trap_no=20;

return;


 (is_f00f_bug(address))
return;
```

Commit 1
Author: Bob

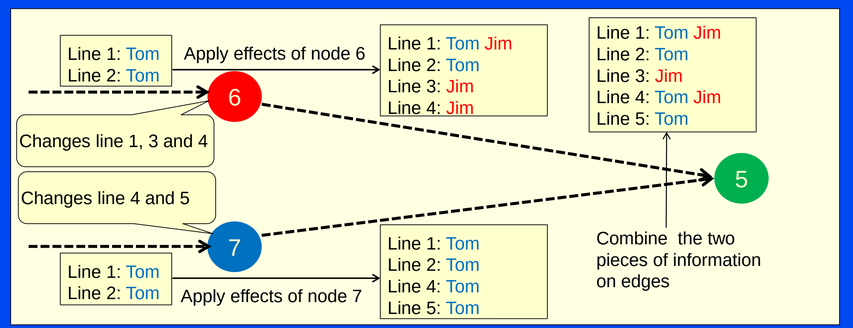
```
uct

signed long


 segfault
,
RN_INFO
dress,
oid *)regs-
sp);

KERN_CONT "


N_CONT
```

Create Character-Level Authorship

- Start from git's commit graph from the repository
  - Nodes represent commits
  - Edges represent development dependency
- Perform a backward flow analysis
  - Edges are annotated with the authors who changed the lines
  - Nodes add author information to lines



- For each commit
  - Combine the author information from analyzed commits
  - Use ldiff[1] to identify changed lines from added lines and deleted lines
  - Apply the ldiff result to the combined information to produce new information



- The analysis for a line stops when the line is first added into the repository

[1] G. Canfora, L. Cerulo and M. D. Penta. "Ldiff: an Enhanced Line Differencing Tool."


```
static inline void
show_signal_msg(struct pt_regs *regs, unsigned long error_code,
                unsigned long address, struct task_struct *tsk)
{
    if (!unhandled_signal(tsk, SIGSEGV))        Author Jim
        return;

    if (!printk_ratelimit())
        return;

    printk("%s%s[%d]: segfault at %lx ip %p sp %p error %lx",
           task_pid_nr(tsk) > 1 ? KERN_INFO : KERN_EMERG,
           tsk->comm, task_pid_nr(tsk), address,
           (void *)regs->ip, (void *)regs->sp, error_code);

    print_vma_addr(KERN_CONT " in ", regs->ip);     Author Bob

    printk(KERN_CONT "\n");
}

static void
__bad_area_nosemaphore(struct pt_regs *regs, unsigned long error_code,
                       unsigned long address, int si_code)
{
    struct task_struct *tsk = current;          Author Tom

    if (error_code & PF_USER) {
        local_irq_enable();

        if (is_prefetch(regs, error_code, address))
            return;

        if (is_errata100(regs, address))
            return;

        if (unlikely(show_unhandled_signals))
            show_signal_msg(regs, error_code, address, tsk);

        tsk->thread.cr2          = address;
        tsk->thread.error_code   = error_code | (address >= TASK_SIZE);
        tsk->thread.trap_no      = 14;

        force_sig_info_fault(SIGSEGV, si_code, address, tsk, 0);

        return;
    }

    if (is_f00f_bug(regs, address))
        return;

    no_context(regs, error_code, address);
}
```