Static Analysis on the FreeBSD MAC Framework Using Mygcc*

Zhouyi Zhou[1 2], Robert N.M. Watson[3], Yeping He[1], and Hongliang Liang[1]

[1] Institute of Software, Chinese Academy of Sciences, Beijing 100080, PRC
[2] Graduate School of the Chinese Academy of Sciences, Beijing 100049, PRC
[3] Computer Laboratory, University of Cambridge, Cambridge CB3 0FD, UK
[4] Seccuris Incorporation, MB R3A 0A3, CA
zhouzhouyi@ercist.iscas.ac.cn, robert.watson@cl.cam.ac.uk,
hongliang@ios.cn, yphe@ercist.iscas.ac.cn

1 The Enhanced Algorithm

proc *check*(CFG, *condate*)
  $substs \leftarrow \emptyset$
  foreach node $t \in CFG$ **do**
    $global\_store \leftarrow \emptyset$
    if $condate.from.format\_spec =$ "entry"
      then if *match*(t, *condate.to*)
          then
          $substs \leftarrow substs \cup \{global\_store\}$
        fi // *match*(t, *condate.to*)
      else if $\neg$ *condate.from* **or** *match*(t, $condate.from$)
          then
          $substs \leftarrow substs \cup \{global\_store\}$
        fi //$\neg$ *condate.from* **or** *match*(t, $condate.from$)
    fi //$condate.from.format\_spec =$ "entry"
  end //for
  for $subst \in substs$ **do**
    $global\_store \leftarrow subst$
    $list \leftarrow []$
    if $condate.from.format\_spec =$ "entry"
      then
        foreach node $t \in CFG$ **do**
          $list \leftarrow [t|list]$
        end
      else
        foreach node $t \in CFG$ **do**
          if *match*(t,$from$) **then** $list \leftarrow [t|list]$ **fi**
        end

* This work is jointly supported by Google Summer of Code 2007 and National Basic Research Program of China (973) under Grant No. G1999035802.

```
        fi //condate.from.format_spec = "entry"
        while list = [t|rest] do
            list ← rest
            if ¬ visited(t)
                then
                    visited(t) ← true
                    if match(t,condate.to)
                        then print "reached t"
                        else foreach edge e = t → t' do
                                if ¬ match(e,avoid)
                                    then list ← [t'|list]
                                fi
                            end
                    fi// match(t,condate.to)
            fi //¬ visited(t)
        end//while
    end //for
end //proc
```