ExecVus

Alexandru Totolici

a way to visualize control-flow in software execution

planned for:

- code collapse
- code reordering
- improved selection of exec. path
- “zoom” in/out of control flow
- scented graph for func. callers/callees

managed:

- ✓ code collapse
- x code reordering
- … improved selection of exec. path
- … “zoom” in/out of control flow
- ✓ scented graph for func. callers/callees

bonus, learned:

- <canvas>
- javascript / jquery
- mercurial / hgweb
- python
- mice in IKBLC!

improved selection of exec. path

rewrite to underlying engine required

animating on <canvas> might be slow

“zoom” in/out of control flow

some backend code written

had issues because of dataset

demo

finish zooming functionality

improve scents

performance tweaks

questions? comments.

(answers optional)

```
470 EXPORT_SYMBOL(key_instantiate_and_link);
476 int key_negate_and_link(struct key *key,
531 EXPORT_SYMBOL(key_negate_and_link);
538 static void key_cleanup(struct work_struct *work)
603 void key_put(struct key *key)
604 {
605   if (key) {
606           key_check(key);
607
608           if (atomic_dec_and_test(&key->usage))
609                   schedule_work(&key_cleanup_task);
610   }
611
```

/Developer/dsg/dist/linux-2.6.30

security/security.c

line source

```
  1 static __initdata char chosen_lsm[SECURITY_NAME_MAX + 1];
 24 extern struct security_operations default_security_ops;
 25 extern void security_fixup_ops(struct security_operations *ops);
 27 struct security_operations *security_ops; /* Initialized to NULL */
 30 unsigned long mmap_min_addr = CONFIG_SECURITY_DEFAULT_MMAP_MIN_ADDR;
 32 static inline int verify(struct security_operations *ops)
 41 static void __init do_security_initcalls(void)
 56 int __init security_init(void)
 68 static int __init choose_lsm(char *str)
 73 __setup("security=", choose_lsm);
 90 int __init security_module_enable(struct security_operations *ops)
112 int register_security(struct security_operations *ops)
130 int security_ptrace_may_access(struct task_struct *child, unsigned int mode)
135 int security_ptrace_traceme(struct task_struct *parent)
140 int security_capget(struct task_struct *target,
148 int security_capset(struct cred *new, const struct cred *old,
157 int security_capable(int cap)
163 int security_real_capable(struct task_struct *tsk, int cap)
174 int security_real_capable_noaudit(struct task_struct *tsk, int cap)
185 int security_acct(struct file *file)
190 int security_sysctl(struct ctl_table *table, int op)
195 int security_quotactl(int cmds, int type, int id, struct super_block *sb)
200 int security_quota_on(struct dentry *dentry)
205 int security_syslog(int type)
210 int security_settime(struct timespec *ts, struct timezone *tz)
215 int security_vm_enough_memory(long pages)
221 int security_vm_enough_memory_mm(struct mm_struct *mm, long pages)
227 int security_vm_enough_memory_kern(long pages)
234 int security_bprm_set_creds(struct linux_binprm *bprm)
239 int security_bprm_check(struct linux_binprm *bprm)
244 void security_bprm_committing_creds(struct linux_binprm *bprm)
249 void security_bprm_committed_creds(struct linux_binprm *bprm)
254 int security_bprm_secureexec(struct linux_binprm *bprm)
259 int security_sb_alloc(struct super_block *sb)
264 void security_sb_free(struct super_block *sb)
269 int security_sb_copy_data(char *orig, char *copy)
273 EXPORT_SYMBOL(security_sb_copy_data);
275 int security_sb_kern_mount(struct super_block *sb, int flags, void *data)
280 int security_sb_show_options(struct seq_file *m, struct super_block *sb)
285 int security_sb_statfs(struct dentry *dentry)
290 int security_sb_mount(char *dev_name, struct path *path,
296 int security_sb_check_sb(struct vfsmount *mnt, struct path *path)
301 int security_sb_umount(struct vfsmount *mnt, int flags)
306 void security_sb_umount_close(struct vfsmount *mnt)
311 void security_sb_umount_busy(struct vfsmount *mnt)
316 void security_sb_post_remount(struct vfsmount *mnt, unsigned long flags, void *data)
321 void security_sb_post_addmount(struct vfsmount *mnt, struct path *mountpoint)
326 int security_sb_pivotroot(struct path *old_path, struct path *new_path)
331 void security_sb_post_pivotroot(struct path *old_path, struct path *new_path)
336 int security_sb_set_mnt_opts(struct super_block *sb,
341 EXPORT_SYMBOL(security_sb_set_mnt_opts);
343 void security_sb_clone_mnt_opts(const struct super_block *oldsb,
348 EXPORT_SYMBOL(security_sb_clone_mnt_opts);
350 int security_sb_parse_opts_str(char *options, struct security_mnt_opts *opts)
354 EXPORT_SYMBOL(security_sb_parse_opts_str);
356 int security_inode_alloc(struct inode *inode)
362 void security_inode_free(struct inode *inode)
367 int security_inode_init_security(struct inode *inode, struct inode *dir,
374 EXPORT_SYMBOL(security_inode_init_security);
377 int security_path_mknod(struct path *path, struct dentry *dentry, int mode,
384 EXPORT_SYMBOL(security_path_mknod);
386 int security_path_mkdir(struct path *path, struct dentry *dentry, int mode)
393 int security_path_rmdir(struct path *path, struct dentry *dentry)
400 int security_path_unlink(struct path *path, struct dentry *dentry)
407 int security_path_symlink(struct path *path, struct dentry *dentry,
415 int security_path_link(struct dentry *old_dentry, struct path *new_dir,
423 int security_path_rename(struct path *old_dir, struct dentry *old_dentry,
433 int security_path_truncate(struct path *path, loff_t length,
442 int security_inode_create(struct inode *dir, struct dentry *dentry, int mode)
448 EXPORT_SYMBOL_GPL(security_inode_create);
450 int security_inode_link(struct dentry *old_dentry, struct inode *dir,
458 int security_inode_unlink(struct inode *dir, struct dentry *dentry)
465 int security_inode_symlink(struct inode *dir, struct dentry *dentry,
473 int security_inode_mkdir(struct inode *dir, struct dentry *dentry, int mode)
479 EXPORT_SYMBOL_GPL(security_inode_mkdir);
481 int security_inode_rmdir(struct inode *dir, struct dentry *dentry)
488 int security_inode_mknod(struct inode *dir, struct dentry *dentry, int mode, dev_t dev)
495 int security_inode_rename(struct inode *old_dir, struct dentry *old_dentry,
505 int security_inode_readlink(struct dentry *dentry)
512 int security_inode_follow_link(struct dentry *dentry, struct nameidata *nd)
519 int security_inode_permission(struct inode *inode, int mask)
526 int security_inode_setattr(struct dentry *dentry, struct iattr *attr)
532 EXPORT_SYMBOL_GPL(security_inode_setattr);
534 int security_inode_getattr(struct vfsmount *mnt, struct dentry *dentry)
541 void security_inode_delete(struct inode *inode)
548 int security_inode_setxattr(struct dentry *dentry, const char *name,
556 void security_inode_post_setxattr(struct dentry *dentry, const char *name,
564 int security_inode_getxattr(struct dentry *dentry, const char *name)
571 int security_inode_listxattr(struct dentry *dentry)
578 int security_inode_removexattr(struct dentry *dentry, const char *name)
585 int security_inode_need_killpriv(struct dentry *dentry)
590 int security_inode_killpriv(struct dentry *dentry)
595 int security_inode_getsecurity(const struct inode *inode, const char *name, void **buffer, bool alloc)
602 int security_inode_setsecurity(struct inode *inode, const char *name, const void *value, size_t size, int flags)
609 int security_inode_listsecurity(struct inode *inode, char *buffer, size_t buffer_size)
616 void security_inode_getsecid(const struct inode *inode, u32 *secid)
621 int security_file_permission(struct file *file, int mask)
626 int security_file_alloc(struct file *file)
631 void security_file_free(struct file *file)
636 int security_file_ioctl(struct file *file, unsigned int cmd, unsigned long arg)
641 int security_file_mmap(struct file *file, unsigned long reqprot,
654 int security_file_mprotect(struct vm_area_struct *vma, unsigned long reqprot,
659 int security_file_lock(struct file *file, unsigned int cmd)
659 int security_file_fcntl(struct file *file, unsigned int cmd, unsigned long arg)
664 int security_file_set_fowner(struct file *file)
669 int security_file_send_sigiotask(struct task_struct *tsk,
675 int security_file_receive(struct file *file)
680 int security_dentry_open(struct file *file, const struct cred *cred)
685 int security_task_create(unsigned long clone_flags)
690 void security_cred_free(struct cred *cred)
695 int security_prepare_creds(struct cred *new, const struct cred *old, gfp_t gfp)
700 void security_commit_creds(struct cred *new, const struct cred *old)
705 int security_kernel_act_as(struct cred *new, u32 secid)
710 int security_kernel_create_files_as(struct cred *new, struct inode *inode)
715 int security_task_setuid(uid_t id0, uid_t id1, uid_t id2, int flags)
720 int security_task_fix_setuid(struct cred *new, const struct cred *old,
726 int security_task_setgid(gid_t id0, gid_t id1, gid_t id2, int flags)
731 int security_task_setpgid(struct task_struct *p, pid_t pgid)
736 int security_task_getpgid(struct task_struct *p)
741 int security_task_getsid(struct task_struct *p)
744 void security_task_getsecid(struct task_struct *p, u32 *secid)
750 EXPORT_SYMBOL(security_task_getsecid);
752 int security_task_setgroups(struct group_info *group_info)
757 int security_task_setnice(struct task_struct *p, int nice)
762 int security_task_setioprio(struct task_struct *p, int ioprio)
767 int security_task_getioprio(struct task_struct *p)
772 int security_task_setrlimit(unsigned int resource, struct rlimit *new_rlim)
777 int security_task_setscheduler(struct task_struct *p,
783 int security_task_getscheduler(struct task_struct *p)
788 int security_task_movememory(struct task_struct *p)
793 int security_task_kill(struct task_struct *p, struct siginfo *info,
798 int security_task_wait(struct task_struct *p)
804 int security_task_prctl(int option, unsigned long arg2, unsigned long arg3,
810 void security_task_to_inode(struct task_struct *p, struct inode *inode)
815 int security_ipc_permission(struct kern_ipc_perm *ipcp, short flag)
820 void security_ipc_getsecid(struct kern_ipc_perm *ipcp, u32 *secid)
825 int security_msg_msg_alloc(struct msg_msg *msg)
830 void security_msg_msg_free(struct msg_msg *msg)
835 int security_msg_queue_alloc(struct msg_queue *msq)
840 void security_msg_queue_free(struct msg_queue *msq)
845 int security_msg_queue_associate(struct msg_queue *msq, int msqflg)
850 int security_msg_queue_msgctl(struct msg_queue *msq, int cmd)
855 int security_msg_queue_msgsnd(struct msg_queue *msq,
861 int security_msg_queue_msgrcv(struct msg_queue *msq, struct msg_msg *msg,
867 int security_shm_alloc(struct shmid_kernel *shp)
872 void security_shm_free(struct shmid_kernel *shp)
877 int security_shm_associate(struct shmid_kernel *shp, int shmflg)
882 int security_shm_shmctl(struct shmid_kernel *shp, int cmd)
887 int security_shm_shmat(struct shmid_kernel *shp, char __user *shmaddr, int shmflg)
892 int security_sem_alloc(struct sem_array *sma)
897 void security_sem_free(struct sem_array *sma)
902 int security_sem_associate(struct sem_array *sma, int semflg)
907 int security_sem_semctl(struct sem_array *sma, int cmd)
912 int security_sem_semop(struct sem_array *sma, struct sembuf *sops,
918 void security_d_instantiate(struct dentry *dentry, struct inode *inode)
924 EXPORT_SYMBOL(security_d_instantiate);
926 int security_getprocattr(struct task_struct *p, char *name, char **value)
931 int security_setprocattr(struct task_struct *p, char *name, void *value, size_t size)
936 int security_netlink_send(struct sock *sk, struct sk_buff *skb)
941 int security_netlink_recv(struct sk_buff *skb, int cap)
946 EXPORT_SYMBOL(security_netlink_recv);
947 int security_secid_to_secctx(u32 secid, char **secdata, u32 *seclen)
951 EXPORT_SYMBOL(security_secid_to_secctx);
953 int security_secctx_to_secid(const char *secdata, u32 seclen, u32 *secid)
957 EXPORT_SYMBOL(security_secctx_to_secid);
959 void security_release_secctx(char *secdata, u32 seclen)
963 EXPORT_SYMBOL(security_release_secctx);
967 int security_unix_stream_connect(struct socket *sock, struct socket *other,
972 EXPORT_SYMBOL(security_unix_stream_connect);
974 int security_unix_may_send(struct socket *sock,  struct socket *other)
978 EXPORT_SYMBOL(security_unix_may_send);
980 int security_socket_create(int family, int type, int protocol, int kern)
985 int security_socket_post_create(struct socket *sock, int family,
991 int security_socket_bind(struct socket *sock, struct sockaddr *address, int addrlen)
997 int security_socket_connect(struct socket *sock, struct sockaddr *address, int addrlen)
1002 int security_socket_listen(struct socket *sock, int backlog)
1007 int security_socket_accept(struct socket *sock, struct socket *newsock)
1012 int security_socket_sendmsg(struct socket *sock, struct msghdr *msg, int size)
1017 int security_socket_recvmsg(struct socket *sock, struct msghdr *msg,
1023 int security_socket_getsockname(struct socket *sock)
1028 int security_socket_getpeername(struct socket *sock)
1033 int security_socket_getsockopt(struct socket *sock, int level, int optname)
1038 int security_socket_setsockopt(struct socket *sock, int level, int optname)
1043 int security_socket_shutdown(struct socket *sock, int how)
1048 int security_sock_rcv_skb(struct sock *sk, struct sk_buff *skb)
1052 EXPORT_SYMBOL(security_sock_rcv_skb);
1054 int security_socket_getpeersec_stream(struct socket *sock, char __user *optval,
1060 int security_socket_getpeersec_dgram(struct socket *sock, struct sk_buff *skb, u32 *secid)
1064 EXPORT_SYMBOL(security_socket_getpeersec_dgram);
1066 int security_sk_alloc(struct sock *sk, int family, gfp_t priority)
1071 void security_sk_free(struct sock *sk)
1076 void security_sk_clone(const struct sock *sk, struct sock *newsk)
1081 void security_sk_classify_flow(struct sock *sk, struct flowi *fl)
1085 EXPORT_SYMBOL(security_sk_classify_flow);
1087 void security_req_classify_flow(const struct request_sock *req, struct flowi *fl)
1091 EXPORT_SYMBOL(security_req_classify_flow);
1093 void security_sock_graft(struct sock *sk, struct socket *parent)
1097 EXPORT_SYMBOL(security_sock_graft);
1099 int security_inet_conn_request(struct sock *sk,
1104 EXPORT_SYMBOL(security_inet_conn_request);
1106 void security_inet_csk_clone(struct sock *newsk,
1112 void security_inet_conn_established(struct sock *sk,
1122 int security_xfrm_policy_alloc(struct xfrm_sec_ctx **ctxp, struct xfrm_user_sec_ctx *sec_ctx)
1126 EXPORT_SYMBOL(security_xfrm_policy_alloc);
1128 int security_xfrm_policy_clone(struct xfrm_sec_ctx *old_ctx,
1134 void security_xfrm_policy_free(struct xfrm_sec_ctx *ctx)
1138 EXPORT_SYMBOL(security_xfrm_policy_free);
1140 int security_xfrm_policy_delete(struct xfrm_sec_ctx *ctx)
1145 int security_xfrm_state_alloc(struct xfrm_state *x, struct xfrm_user_sec_ctx *sec_ctx)
1149 EXPORT_SYMBOL(security_xfrm_state_alloc);
1151 int security_xfrm_state_alloc_acquire(struct xfrm_state *x,
1163 int security_xfrm_state_delete(struct xfrm_state *x)
1167 EXPORT_SYMBOL(security_xfrm_state_delete);
1169 void security_xfrm_state_free(struct xfrm_state *x)
1174 int security_xfrm_policy_lookup(struct xfrm_sec_ctx *ctx, u32 fl_secid, u8 dir)
1179 int security_xfrm_state_pol_flow_match(struct xfrm_state *x,
1185 int security_xfrm_decode_session(struct sk_buff *skb, u32 *secid)
1190 void security_skb_classify_flow(struct sk_buff *skb, struct flowi *fl)
1196 EXPORT_SYMBOL(security_skb_classify_flow);
1202 int security_key_alloc(struct key *key, const struct cred *cred,
1208 void security_key_free(struct key *key)
1213 int security_key_permission(key_ref_t key_ref,
1219 int security_key_getsecurity(struct key *key, char **_buffer)
1228 int security_audit_rule_init(u32 field, u32 op, char *rulestr, void **lsmrule)
1233 int security_audit_rule_known(struct audit_krule *krule)
1238 void security_audit_rule_free(void *lsmrule)
1243 int security_audit_rule_match(u32 secid, u32 field, u32 op, void *lsmrule,
```