

[PATCH] split security_key_alloc into two functions

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-03/msg09557.html>

- *From:* "Serge E. Hallyn" <serue@xxxxxxxxxx>
 - *Date:* Tue, 28 Mar 2006 07:05:33 -0600
-

The security_key_alloc() function acted as both an authorizer and security structure allocation function. These roles should be separated. There are two reasons for this.

First, if two modules are stacked, the first module might grant permission and allocate security data, after which the second module refuses permission.

Second, by adding a security_post_alloc() function after the serial number has been assigned, security modules can append useful info.

Note that currently there is no LSM using these hooks, so the question of whether an LSM needs to record the serial number can't really be answered.

An alternative to this patch, supported by the historical approach to LSM hooks, would be to remove all these hooks. However as the keystore starts being used – in particular by, eg, ecryptfs – one might expect LSMs to be more interested in key activity.

```
:100644 100644 aaa0a5c... 3d8602e... M include/linux/security.h
:100644 100644 fd99429... 1eff777... M security/dummy.c
:100644 100644 a057e33... 6be6269... M security/keys/key.c
```

Signed-off-by: Serge Hallyn <serue@xxxxxxxxxx>

```
--- a/include/linux/security.h
+++ b/include/linux/security.h
@@ -844,10 +844,14 @@ struct swap_info_struct;
* Security hooks affecting all Key Management operations
*
* @key_alloc:
- * Permit allocation of a key and assign security data. Note that key does
- * not have a serial number assigned at this point.
+ * Check permission to allocate a key and assign security data. Note
+ * that key does not have a serial number assigned at this point.
* @key points to the key.
```

[PATCH] split security_key_alloc into two functions

```
* Return 0 if permission is granted, -ve error otherwise.
+ * @key_post_alloc:
+ * Allocate and attach a security structure to a key structure.
+ * At this point there is a serial number attached to the key.
+ * @key points to the key.
* @key_free:
* Notification of destruction; free security data.
* @key points to the key.
@@ -1312,6 +1316,7 @@ struct security_operations {
/* key management security hooks */
#ifdef CONFIG_KEYS
int (*key_alloc)(struct key *key);
+ void (*key_post_alloc)(struct key *key);
void (*key_free)(struct key *key);
int (*key_permission)(key_ref_t key_ref,
struct task_struct *context,
@@ -3001,6 +3006,11 @@ static inline int security_key_alloc(str
return security_ops->key_alloc(key);
}

+static inline void security_key_post_alloc(struct key *key)
+{
+ security_ops->key_post_alloc(key);
+}
+
static inline void security_key_free(struct key *key)
{
security_ops->key_free(key);
@@ -3020,6 +3030,10 @@ static inline int security_key_alloc(str
return 0;
}

+static inline void security_key_post_alloc(struct key *key)
+{
+}
+
static inline void security_key_free(struct key *key)
{
}
diff --git a/security/dummy.c b/security/dummy.c
index fd99429..1eff777 100644
--- a/security/dummy.c
+++ b/security/dummy.c
@@ -860,6 +860,10 @@ static inline int dummy_key_alloc(struct
return 0;
}

+static inline void dummy_key_post_alloc(struct key *key)
+{
+}
+
```

[PATCH] split security_key_alloc into two functions

```
static inline void dummy_key_free(struct key *key)
{
}
@@ -1036,6 +1040,7 @@ void security_fixup_ops (struct security
#endif /* CONFIG_SECURITY_NETWORK_XFRM */
#ifdef CONFIG_KEYS
set_to_dummy_if_null(ops, key_alloc);
+ set_to_dummy_if_null(ops, key_post_alloc);
set_to_dummy_if_null(ops, key_free);
set_to_dummy_if_null(ops, key_permission);
#endif /* CONFIG_KEYS */
diff --git a/security/keys/key.c b/security/keys/key.c
index a057e33..6be6269 100644
--- a/security/keys/key.c
+++ b/security/keys/key.c
@@ -325,6 +325,7 @@ struct key *key_alloc(struct key_type *t
/* publish the key by giving it a serial number */
atomic_inc(&user->nkeys);
key_alloc_serial(key);
+ security_key_post_alloc(key);
```

error:

return key;

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>