

[PATCH 1/3] Audit: break up execve argument lists into multiple records

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-10/msg02460.html>

- *From:* Eric Paris <eparis@xxxxxxxxxx>
 - *Date:* Mon, 08 Oct 2007 17:34:01 -0400
-

Break the auditing of a list of execve arguments into smaller records if there are a too many. The limit is currently around 7.5k of arguments as userspace has an 8k buffer limit and will drop messages which are longer.

Signed-off-by: Eric Paris <eparis@xxxxxxxxxx>

Basically the same patch as last time, used a #define, cleaned up a memory leak on a malloc failure code path. Other than that it's the same.

7500 also is a good size because it means we never need more than a 2 page allocation. I'd say this is a good thing as even if we used a full netlink message of 32k its just making it harder on the kernel to have the memory it needs. I think userspace will want to get fixed eventually to handle a full 32k just in case, but keeping the kernel under 8k when we know we can seems like a good idea.

kernel/auditsc.c | 39 ++++++-----
1 files changed, 33 insertions(+), 6 deletions(-)

diff --git a/kernel/auditsc.c b/kernel/auditsc.c

index 04f3ffb..4176db6 100644

--- a/kernel/auditsc.c

+++ b/kernel/auditsc.c

@@ -78,6 +78,9 @@ extern struct list_head audit_filter_list[];

/* Indicates that audit should log the full pathname. */

#define AUDIT_NAME_FULL -1

+/* no execve audit message should be longer than this (userspace limits) */

+#define MAX_EXECVE_AUDIT_LEN 7500

+

/* number of audit rules */

int audit_n_rules;

@@ -819,11 +822,12 @@ static int audit_log_pid_context(struct audit_context *context, pid_t pid,

return rc;

}

-static void audit_log_execve_info(struct audit_buffer *ab,

[PATCH 1/3] Audit: break up execve argument lists into multiple records

```
+static void audit_log_execve_info(struct audit_context *context,
+ struct audit_buffer **ab,
struct audit_aux_data_execve *axi)
{
int i;
- long len, ret;
+ long len, ret, len_sent = 0;
const char __user *p;
char *buf;

@@ -833,7 +837,11 @@ static void audit_log_execve_info(struct audit_buffer *ab,
p = (const char __user *)axi->mm->arg_start;

for (i = 0; i < axi->argc; i++, p += len) {
+ char tmp_buf[12];
+ /* how many digits are in i? */
+ int i_len = sprintf(tmp_buf, 12, "%d", i);
len = strlen_user(p, MAX_ARG_STRLEN);
+
/*
* We just created this mm, if we can't find the strings
* we just copied into it something is _very_ wrong. Similar
@@ -862,9 +870,28 @@ static void audit_log_execve_info(struct audit_buffer *ab,
send_sig(SIGKILL, current, 0);
}

- audit_log_format(ab, "a%d=", i);
- audit_log_untrustedstring(ab, buf);
- audit_log_format(ab, "\n");
+ /*
+ * If there are a lot of args just break them into multiple
+ * messages. the last ab started will get closed by the
+ * caller.
+ *
+ * + 3 + i_len because we know at least a = and \n will be sent
+ * as well as the number of digits in i (i_len).
+ */
+ len_sent += (len + 3 + i_len);
+ if (len_sent > MAX_EXECVE_AUDIT_LEN) {
+ len_sent = len + 3 + i_len;
+ audit_log_end(*ab);
+ *ab = audit_log_start(context, GFP_KERNEL, AUDIT_EXECVE);
+ if (!*ab) {
+ kfree(buf);
+ return;
+ }
+ }
+
+ audit_log_format(*ab, "a%d=", i);
+ audit_log_untrustedstring(*ab, buf);
+ audit_log_format(*ab, "\n");
```

[PATCH 1/3] Audit: break up execve argument lists into multiple records

```
kfree(buf);
}
@@ -1010,7 +1037,7 @@ static void audit_log_exit(struct audit_context *context, struct task_struct *ts

case AUDIT_EXECVE: {
struct audit_aux_data_execve *axi = (void *)aux;
- audit_log_execve_info(ab, axi);
+ audit_log_execve_info(context, &ab, axi);
break; }

case AUDIT_SOCKETCALL: {
```

—
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@xxxxxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>