بسمه تعالی

گزارش تمرین عملی دوم – سوال سوم – درس سیستم‌های عامل دکتر اسدی – نیمسال اول 405-404

نویسندگان:

1. سیداحمد موسوی‌اول – 402106648

2. عرفان تیموری – 402105813

گزارشکار.

1. مفهوم و نحوه عملکرد Fork Bomb:

Fork Bomb یک نوع حمله محروم‌سازی از سرویس (Denial-of-Service یا DOS) است. این حمله با استفاده از یک فرآیند (process) انجام می‌شود که به صورت مداوم از خودش کپی می‌گیرد (عملیات fork). هر کپی جدید نیز بلافاصله شروع به کپی گرفتن از خودش می‌کند. این فرآیند باعث رشد نمایی تعداد فرآیندها در سیستم می‌شود (1، 2، 4، 8، 16، 32 و ...). این رشد سریع، دو منبع حیاتی سیستم را به سرعت مصرف می‌کند:

1. جدول فرآیندهای سیستم (Process Table): هر سیستم‌عامل یک محدودیت برای حداکثر تعداد فرآیندهایی که می‌توانند همزمان در حال اجرا باشند (PID max) دارد. Fork Bomb این جدول را به سرعت پر می‌کند.

2. حافظه و CPU: هر فرآیند جدید، هرچند کوچک، به مقداری حافظه و زمان CPU برای مدیریت نیاز دارد. هزاران فرآیند به سرعت تمام RAM و 100% توان CPU را اشغال می‌کنند.

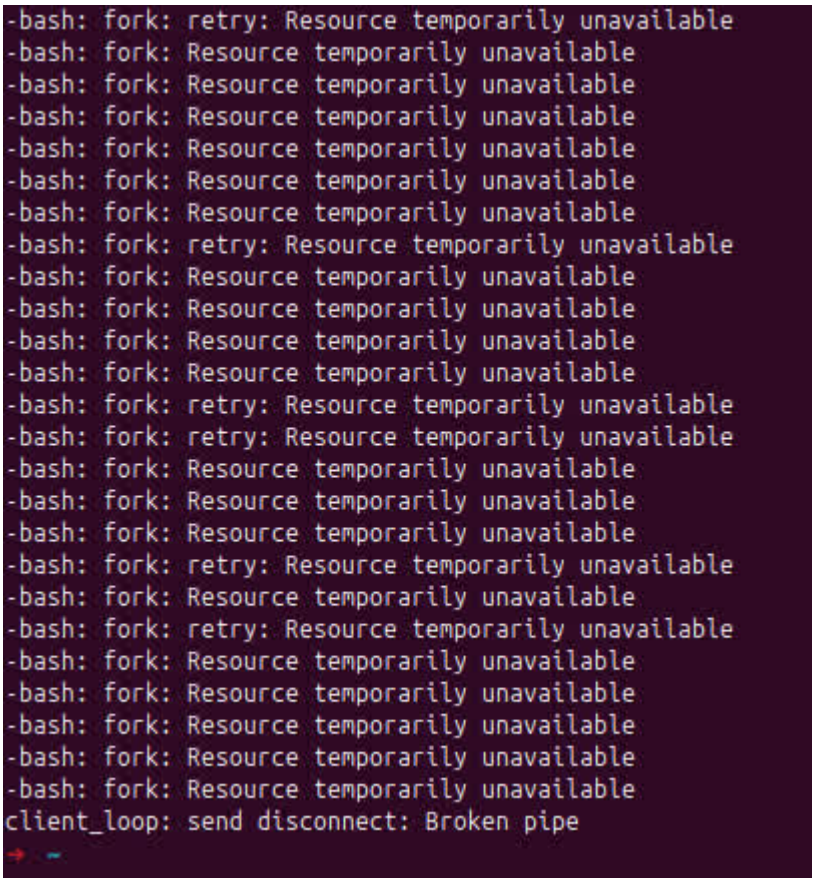
بنابراین سیستم به شدت کند شده و در نهایت فریز یا هنگ می‌کند. از آنجایی که جدول فرآیندها پر شده است، هیچ فرآیند جدیدی (حتی فرآیندهای مدیریتی سیستم یا دستورات شما مانند kill) نمی‌تواند اجرا شود، چون برای اجرای هر دستور جدیدی نیاز به fork کردن یک فرآیند جدید است.

2. یک نمونه ساده Fork Bomb و اجرای آن:

معروف‌ترین، ساده‌ترین و فشرده‌ترین نمونه Fork Bomb، یک خط کد در ترمینال لینوکس است:

```
1.  :(){ :|:& };:
```

بعد از اجرای این دستور بعد از چند ثانیه ترمینال فریز می‌شود و هنگ می‌کند و حتی با بازکردن ترمینال جدید باز هم نمی‌توانیم دستور جدیدی را در ترمینال اجرا کنیم؛ همچنین از یک جا به بعد پردازه‌های جدید ایجاد شده نیز به محدودیت تعداد پردازه‌ها می‌خورند. خروجی این دستور در تصویر زیر قابل مشاهده است:



3. اثرات Fork Bomb و عدم کارکرد Ctrl+C:

همانطور که در بخش ۱ گفته شد، به محض اجرا:

1. CPU Usage: بلافاصله به 100% می‌رسد، چون تمام هسته‌ها درگیر زمان‌بندی و ایجاد هزاران فرآیند جدید می‌شوند.

2. Process count (تعداد پردازه‌ها): تعداد فرآیندها به صورت نمایی بالا می‌رود تا به حداکثر مجاز سیستم (که می‌توان با دستور cat /proc/sys/kernel/pid_max آن را دید) برسد.

3. Swap: مصرف حافظه به سرعت افزایش می‌یابد و سیستم به شدت وارد حالت Swap (استفاده از هارد دیسک به جای رم) می‌شود که این خود، سیستم را به شدت کند می‌کند.

در رابطه با عدم کارکرد Ctrl+C نیز موارد زیر قابل توجه است:

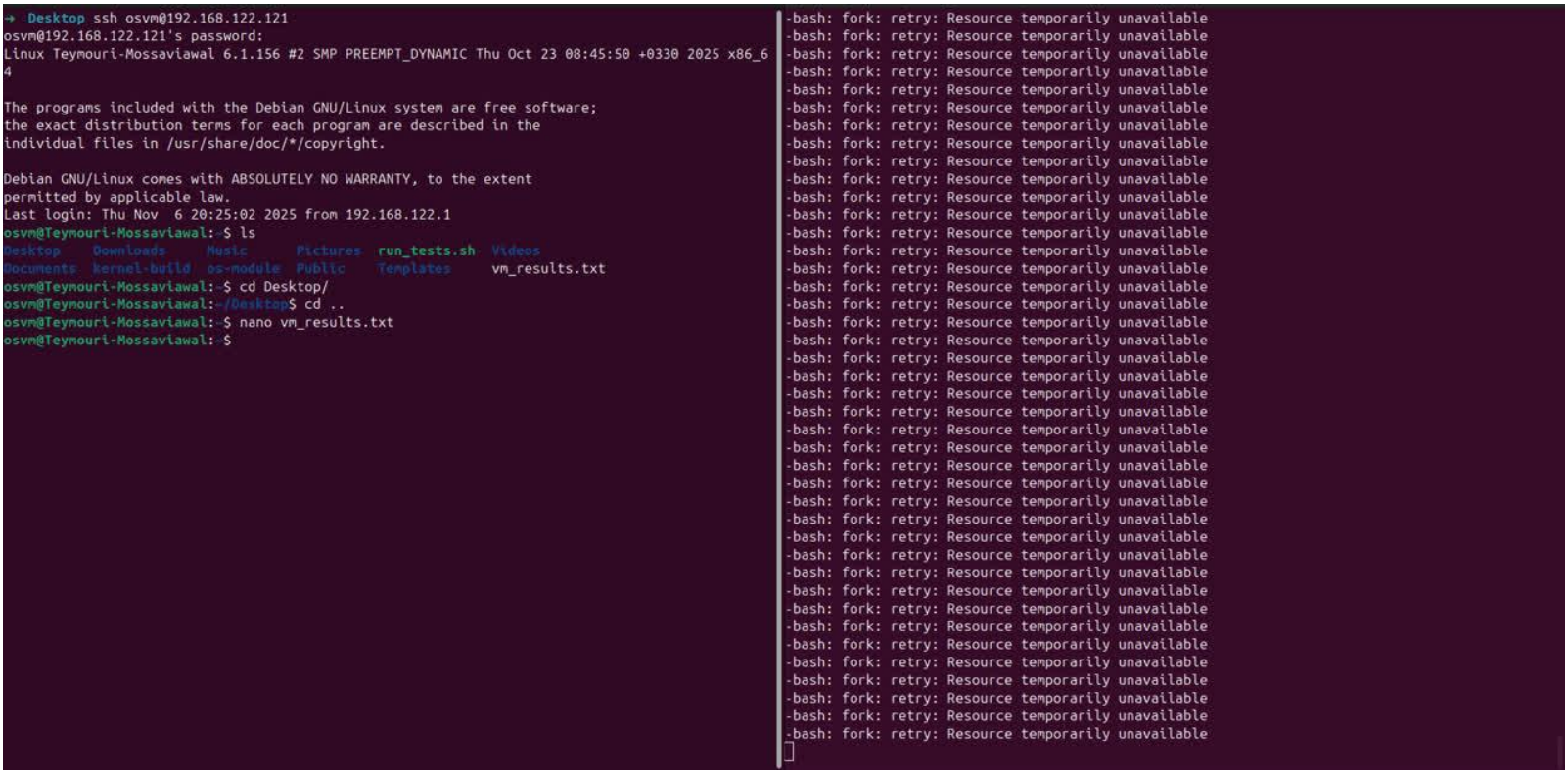
Ctrl+C سیگنال SIGINT (Interrupt) را ارسال می‌کند. این سیگنال فقط به فرآیندهایی ارسال می‌شود که در پیش‌زمینه ترمینال فعلی در حال اجرا هستند. همانطور که در کد بخش ۲ دیدیم، بمب با استفاده از علامت & تمام فرزندان خود را به پس‌زمینه می‌فرستد. بنابراین وقتی Ctrl+C را بزنیم، شاید (شاید!) بتوانیم اولین فرآیند والد را از بین ببریم، اما هزاران کپی دیگر که در پس‌زمینه در حال اجرا و تکثیر هستند، هیچ سیگنالی دریافت نمی‌کنند و به کار خود ادامه می‌دهند. آن‌ها دیگر تحت کنترل آن ترمینال نیستند.

4. راه‌های مقابله با Fork Bomb:

مقابله با Fork Bomb اساسا یک اقدام پیشگیرانه است. وقتی بمب فعال شود، دیگر خیلی دیر است (البته لازم بذکر است بعد از اجرای Fork Bomb می‌توانیم با کاربر root لاگین کرده و تمام پردازه‌های کاربری که بمب را اجرا کرده به کمک دستور killall از بین ببریم). راه اصلی، محدود کردن منابعی است که یک کاربر می‌تواند استفاده کند:

1. استفاده از ulimit:

این دستور (مخفف User Limit) در shell به شما اجازه می‌دهد منابع را برای نشست فعلی محدود کنید. مهم‌ترین پارامتر برای مقابله با Fork Bomb، پارامتر nproc (تعداد فرآیندها) است که با آپشن u- تنظیم می‌شود. برای مثال اگر قبل از اجرای بمب، در ترمینال خود دستور ulimit -u 100 را اجرا کنید، به سیستم می‌گویید که این کاربر حق ندارد بیش از 100 فرآیند به صورت همزمان داشته باشد و در نتیجه وقتی Fork Bomb اجرا شود، به سرعت به 100 فرآیند می‌رسد و پس از آن، هر تلاش جدید برای fork با خطای "Resource temporarily unavailable" مواجه شده و بمب متوقف می‌شود، در حالی که سیستم شما کاملاً پایدار باقی می‌ماند و دستورات ورودی شما را در یک شل دیگر اجرا می‌کند. خروجی این‌کار را می‌توانید در تصویر زیر مشاهده کنید:

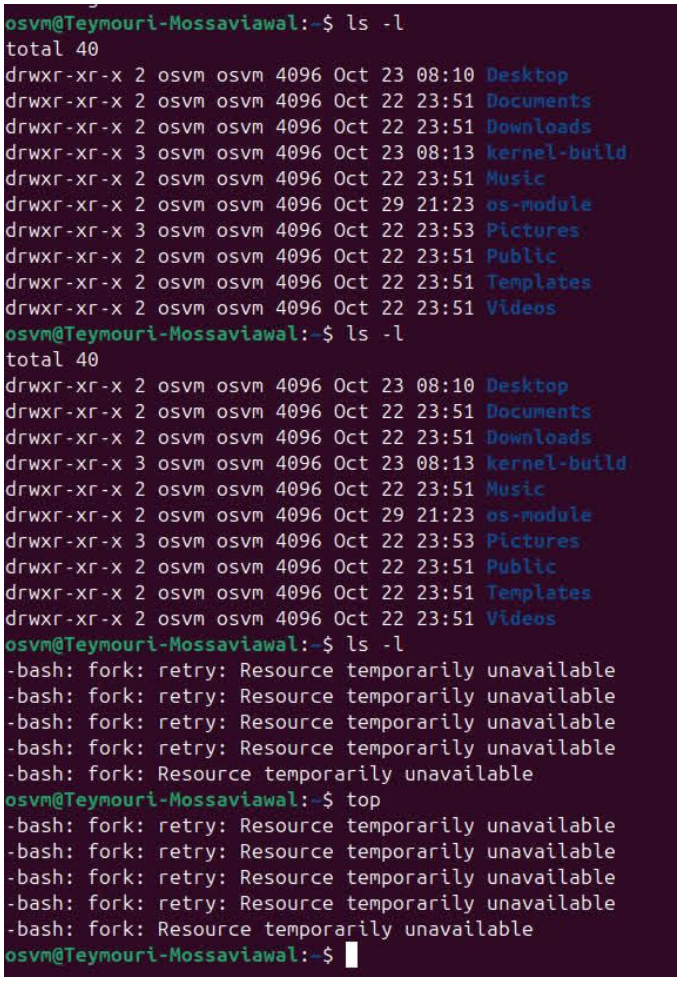


2. تنظیمات سیستمی (PAM):

برای اعمال دائمی این محدودیت‌ها برای کاربران یا گروه‌های خاص، کاربر root می‌تواند سیستم فایل /etc/security/limits.conf را ویرایش می‌کند. برای مثال اضافه کردن خط زیر به این فایل، باعث می‌شود هیچ کاربری در گروه students نتواند بیش از 200 فرآیند داشته باشد:

```
1.  @students hard nproc 200
```

5. اجرای عملی، نمایش عدم ایجاد فرآیند جدید و مهار بمب:



خروجی این قسمت تصویر زیر است:

همونطور که مشاهده می‌شود تا قبل از پر شدن کامل CPU توسط پردازه‌های کپی شده دستورهای ls اجرا شده و خروجی آنها چاپ می‌شود اما بعد از چند دستور پردازنده پر می‌شود و دیگر هیچ دستوری اجرا نمی‌شود و خطای

bash: fork: Resource temporarily unavailable

را دریافت می‌کنیم.

برای مهار کردن fork bomb نیز یا باید مشابه قسمت 4 برای کاربر لیمیت تعیین کنیم یا بعد از پر شدن سیستم راهی جز reboot کردن سیستم نداریم. همچنین برای از بین بردن پردازه‌های کاربر در این حالت می‌توانیم یک ترمینال جدید باز کرده و دستور زیر را اجرا کنیم:

```
1.  killall -9 :
```