Manual Of The Microarchitectural Attack

Hardware Security

For questions contact:
Mathé @ m.c.hertogh@vu.nl
or
Max @ m.barger2@vu.nl

***** WEEK #3 *****

Goal: Become root on the HWSec cluster!!

This week you've been given (in **/tmp** on all the cluster nodes) a program called `set_root_password`, which sets a new password for the `root` user on the machine every time you run it. The program does the following:

1. Executes the `rdrand` instruction a number of times to request random numbers

```
E.g., r1 = rdrand()
      r2 = rdrand()
      r3 = rdrand()
      r4 = rdrand()
      r5 = rdrand()
      r6 = rdrand()
           ...
```

2. Denormalizes the obtained random numbers by zeroing out all the 12 most significant bits (i.e. including the sign)

```
e.g., r1 = 0x1234567890ABCDEF
             |||
             vvv
dr1 = 0x000567890ABCDEF
```

3. It uses (in order) each consecutive couple of denormalized numbers as operands of a floating-point division and performs the FPVI attack to get the corresponding transient result.

```
e.g., t1 = dr1/dr2
      t2 = dr3/dr4
           ...
```

4. It concatenates all the transient results to obtain the password prefix

```
e.g., prefix = t1||t2||t3 ...
                                                    || is the concatenation
```

5. It concatenates the prefix to a fixed string and sets the resulting string as the new password of the `root` user.

The password set every time is of the following structure:

[prefix]||[fixed string]

|| is the concatenation

Each user password (including the root password) is hashed **(using MD5 on the cluster)** and saved along with other user information in the `/etc/shadow` privileged file as follows:

```
ubuntu@hwsec01:~$ sudo cat /etc/shadow
root:$1$VA/Q51FQ$Gju9JhnBv7JaDIu8v9EpY1:18600:0:99999:7:::
daemon:*:18474:0:99999:7:::
bin:*:18474:0:99999:7:::
sys:*:18474:0:99999:7:::
sync:*:18474:0:99999:7:::
.
```

Steps:

This week you are going to recover the `root` password in 4 main steps:

- Leak the random bytes used by `set_root_password` by performing the `CrossTalk` attack.
- Denormalize the leaked random numbers and perform the FPVI attack to get the transient results used as the password prefix.
- Leak the hash of the new password from `/etc/shadow`, which normally needs a `root` access privilege to be read by performing the `RIDL-TAA` attack.
- Crack the remaining fixed string of the password by using the John The Ripper, Hashcat, or any other password cracker. You can assume that the fixed string is small, and made of lowercase letters and numbers only.

Think about how you can bring the content of the `/etc/shadow` file in flight. Follow the explanation in the `RIDL` paper to understand more about how you can build this threat model. You should optimize your attacker so it can leak the `root` user hash (i.e. first row of the `/etc/shadow` file) as fast as you can.

Final Deadline: Fri, December 12th @ 23:59

Upload all of your code (i.e. the latest version) to Canvas. Functionality, performance, and readability of all code submitted will be taken into account during grading.

Bonus

Try to optimize your exploit, to leak the root password hash as fast as possible! You can earn the following bonuses counting towards your final big project grade:

- 0.25 pt: faster than 4 seconds
- additional 0.25 pt: faster than 1 second
- additional 0.25pt: faster than 0.3 seconds