License Locking

Andrei Coman
Yunzhou Li
Jacob Polatty

Estimated Total Hours: 15

Plans (Past Week)

- Adjust the current CryptoFP implementation to reduce
  a) collisions on different machines with identical HW
  b) fingerprint variations with CPU load
- Keep looking into Centauri

Tentative:

- Develop software with CPU fingerprint
- Explore fingerprinting via multithreading

What Actually Happened

Work completed: removed fingerprint variation with CPU load (more careful time measurements or use of time differences instead of quotients)

Major challenges+roadblocks:

- Still experiencing fingerprint collisions on identical HW (100% accuracy in distinguishing between 8 different machine types on GCP and native machines, but experiencing collisions between all GCP machines with the same hardware types)
- Thread-based fingerprinting has shown no patterns so far

Attribution:

- Andrei: removed fingerprint variation with CPU load
- Yunzhou: continued discovering shared memory fingerprinting
- Jacob: continued testing new versions of CryptoFP and memory-based fingerprinting on GCP, implementation of thread-based fingerprinting

Measurements with CPU load

```
$ ./taskset_test.sh … --stress=0
cpu 0: 4/1000 failed
cpu 1: 4/1000 failed
cpu 2: 3/1000 failed
cpu 3: 4/1000 failed
cpu 4: 2/1000 failed
cpu 5: 4/1000 failed
cpu 6: 6/1000 failed
cpu 7: 2/1000 failed
cpu 8: 0/1000 failed
cpu 9: 2/1000 failed
cpu 10: 0/1000 failed
cpu 11: 5/1000 failed
```

```
$ ./taskset_test.sh … --stress=1
cpu 0: 0/1000 failed
cpu 1: 1/1000 failed
cpu 2: 3/1000 failed
cpu 3: 0/1000 failed
cpu 4: 0/1000 failed
cpu 5: 0/1000 failed
cpu 6: 0/1000 failed
cpu 7: 1/1000 failed
cpu 8: 0/1000 failed
cpu 9: 0/1000 failed
cpu 10: 0/1000 failed
cpu 11: 0/1000 failed
```

Note: fingerprints at higher CPU load are more stable

Idea: artificially drive CPU load to 100% and hope this removes collisions

Plans (Next Week)

- Test CryptoFP in different language implementation
- Develop cryptographic algorithm for generating license key from a fingerprint
- Create basic structure for encrypting a binary with the key derived from the fingerprint and decrypting it on the user's machine
- Consider how to securely store the fingerprint on the user's machine so that it cannot be stolen to decrypt the binary

Summary/Overall Progress

Implementation of CryptoFP achieves

a) Stability on a single machine (for at least one week)
b) Stability across different power settings **AND CPU loads**
c) Fingerprint differentiation between machines with different HW

Improvements:

a) Collisions among different machines with same HW