CMPSCI 630 Systems **Spring 2018**

Lecture 19

Lecturer: Emery Berger *Scribe: Breanna Devore-McDonald*

19.1 Information Flow

Security problems with information flow include scecret information leaking out and malicious input coming in. In fact, the dual of trying to prevent secret information leakage is trying to prevent malicious input coming in.

Emery followed up on his example from a few weeks ago:

```
x = secret // x is some secret information
print x // leaks information, but is pretty simple to prevent from happening

// this is called control dependency on classified data
if secret == foo: // leaks info too, harder to prevent
    print a
else:
    print b
```

This kind of attack is not covert, but rather an implicit flow attack. A way to prevent this is with taint analysis. An example of a covert attack would be a timing attack.

Another dual that Emery talked about was declassification and sanitization. Declassifying a secret (encryption) vs looking for bad information and sanitize out.

19.1.1 Taint Analysis

We specifically talked about dynamic taint analysis and what Perl does.

The way this works is as follows: every object has a taint bit that represents if it is clean (0) or dirty (1). For example, input from the internet would be dirty. During execution, the taint bits get propagated and anything that ”touches” a dirty taint bit gets turned into a 1. If data is tainted, then it can't be worked with (or the type of work with it is limited).

Emery also mentioned that data is generally considered ”good” if it has gone through a regexp.

Dynamic taint analysis is reasonably efficient, but dealing with implict flow is hard. One problem is ”taint explosion” where everything gets tainted (too conservative) and then the program is basically meaningless. Static taint analysis is more efficient, can deal with implicit flow, occurs at compile time, but also has problems with taint explosion.

19.2 Bitcoin

We talked about a few different random things about Bitcoin to supplement the other Bitcoin lecture.

Bitcoin is a decentralized cryptocurrency based on consensus over a shared ledger (blockchain). Using the blockchain solves problems like double spending, where someone tries to use the same coins to pay for different things. However, if you own the majority of bitcoin, you can disregard things from the ledger since ledger transactions are decided by a majority consensus.

Homo economicus is the idea that humans are rational, self-interested, and usually maximize utility by pursuing what is most optimal. Wikipedia is a counterexample. Bitcoin follows this principle: bitcoin mining consists of approving transactions and adding blocks to the blockchain, which takes a lot of energy but the reward is some amount of payment. This reward/incentive will stop at some point, since there is a cap on the number of bitcoins that exist. If there is a problem, everyone can agree to disregard a part of the ledger. This is called a hard fork and has happened before.

Bitcoin mining started out on CPUs, transitioned to GPUs, and now is usually done on ASICs (app specific). Mining pools are also common; they consist of a group of miners who want some steady income. The steady income is determined by their percentage of computation power in the pool. For the person in charge of the pool, to calculate the percentage of power of each miner, they require their miners to send in their "low hashes" to show that they are in fact performing computations.

Emery also said the following:

- Zero day vulnerability- an attack that is publicly known for n days, 0 day is unknown
- internet money crime
- instant bug bounty- company pays money to find bug
- liquidity- hard with bitcoin exchanges
- bitcoin just generates pollution
- selfish mining- where a miner (or a mining pool) does not publish a validated block so they have a head start on the next block. Caused a lot of outrage.
- Ethereum - smart contracts
- Concurrency- need to reason about interleaving of concurrent code -¿ race conditions.
- Race conditions are not usually a security attack, except TOCTTOU (time of change to time of use) (file system race)
- currency- universal exchangability
- bitcoin is a store of value
- Island of Yap - stones as a form of money