COMP6441 Week 1 Notes

Youhan Cheery

March 20, 2017

1 Introduction

Two ways of thinking about security:

1. Attack
2. Defence

Asymmetry of attack and defence implies a defender needs to clog every single point entry, while an attacker only needs to find a single point of entry to win.

Thinking as a 'defender' is pointless. Need to think like an attacker to create a better idea of the security of the system. Think like an attacker to make a good defence. Never be impressed by your own defence. Think like an engineer and be systematic.

Why can't we make computer programs secure?
How can we use engineering and other techniques to build *more* secure software?

We still don't know how to make software secure. Incorrect code does not fix itself and is available to others. Adding a little security increases the complexity and consumption of resources by orders of magnitude.

Bridges vs. computer programs:

1. Bridges, just like computer programs, can also collapse. However, we can be sure of the types of collapses, their likelihood and how to test them and make bridges safe in a systematic and thoughtful way. People are confident about bridges - faith in the civil engineers.
2. A bridge doesn't have attackers in the same way computers have attackers. Computers have malicious and normal failures.
3. Bridges can be fully understood. A holistic understanding of a bridge is much more accessible than a computer program, irrespective of the scale. Computer systems have many levels more of abstraction - just knowing the code is not enough.

4. Bridges have existed for the longest time - computers are infantile in comparison

2 History of Hacking

- 1970s were the golden age of hacking - the world really didn't seem to care about security.
- 'Phreaking' involved hacking the most complex/accessible systems at the time - phones. The telephone network began to become a little too big and complex for operators to have a really holistic understanding of it.
- A single band that dictated control and user access made them vulnerable.
- People were writing their own cryptography.
- Roughly around 2000, MS introduced Microsoft money which allowed a computer to access your bank account - now computer security becomes extremely meaningful. Hackers can now gain money. Hacking now becomes criminal.

When your system is too big to understand it will have security issues.

Code follows a specification that is then tested against... "valid inputs -¿ valid outputs". Hackers look for things that may have been out of specification.

Solution: move the behaviour you don't expect into the specification. But for a complex system there are really an infinite amount of variables that can be exploited.

3 Case Study: Wep

Used crypto random number generator called RC4: two series of 0s and 1s were XOR'd. XOR: 2 strings of data and 1 string is random can result in output string that is random. Two consecutive XORs using the same key results in the original string that was XOR'd.

Problem with WEP: when the WEP message was the that the control data was in-band with the message data - meaning if you get access to even some of the control data, you can manipulate it enough to get the actual data.

4 Miscellaneous

- Control systems begin to become connected to the Internet, meaning hackers can now access meaningful machines.
- Security is the weakest link property - you are only as strong as your weakest link. This contradicts human nature, which tends to look at the strongest link. Cryptography does not tend to be the weakest link, the system is not limited to simply the cryptography - there are many more aspects to "security" than encryption.
- If you think about one thing all the time, e.g. hashing, then your focus becomes on that one aspect. Need to consider the entirety of the system.