COURSE 👏 REVIEW 👏

Regarding the course textbook provided in the exam

- Please don't have a specific answer to questions
    - Richard will have to make the exam harder if you have too much information in the textbook
- Also avoid putting TOO MUCH detail
    - This will make it hard for students during the exam to read and understand what you've written under pressure
- Make sure to proof-read and check for errors

Finish working on the textbook **BEFORE TUESDAY NIGHT**

- Any changes made after this won't make it to the final version
- PDF will be released on OpenLearning

The movie, 'The China Syndrome' will be in the Exam

Just Culture

When something goes wrong, instead of acknowledging the root cause, you simply:

- Instead of blaming the individual who was the approximate cause of the problem, you look at the whole system, and figure out what needs to be changed
- When Qantas brought **'Just Culture'** in, it reduced their accidents by 50%!
- Blaming the individual: self gratification
    - The problem with that is that you'll simply encourage the cycle of firing people if the system is broken
- 'Don't shoot the messenger'
    - No one wants to be the deliverer of bad news (since they usually get the blame for it; i.e. Darth Vader)
- By analysing the problem, you're more able to discuss clearly and figure out the root cause of the problem

Look up: how the NHS used it in England

Security Engineering

Recall back to the first question Richard asked:

****What is it to be secure in an Engineering sense?***

> What are the properties that we can learn from Engineering and apply to Security?

1. Learning from the past

- This has allowed us to fail and learn from them
- Reflecting on past experiences

2. Methodologies to follow

- For example Civil Engineering and building a bridge:
    - Australian Standards**
    - What materials to use
    - Reports to write///

Engineering has this notion called *"Best Practice"*, where whenever someone has a good idea, we add to this. This is like a **Checklist**, which is useful because it **stops you from overlooking things**.
Google: the Tacoma bridge

People don't notice things
- If you missed something once, it's very hard to see it the following times
- Fresh pair of eyes is more successful
- Research says: if you miss something you'll find it difficult to spot it the next time

That's why pilots have checklists
Checklists are just a baseline, there are possibilities of things happening
- So engineers need to have processes to be creative
- That's why we don't just use robots who are very good at checklists

3. Redundancy and Testing

- Defense in depth
- Dual control: multiple things have to vote on the same thing (parallel)

Carl Popper: a scientist is always someone who is trying to prove themselves wrong
- It's not what is blindly following the procedure that is science, it's trying to **falsify yourself** (you're following it to prove yourself wrong)
- That's why we like theories that can be proved wrong

The same thing occurs in Engineering

- We test things to make sure things don't go wrong
- Such that we keep the logs and adapt principle from the results
- So that we break things in the lab

4. Engineers take pride in what they do

- They wish to do a good job
- They revel in craftsmanship

Even if your Product Manager is constantly saying *"SHIPSHIPSHIPHISHIP IT"* but you feel like the product isn't ready yet, an engineer with pride in their position will stand their ground and say no. By rushing to send something out, you usually drop the most time-consuming thing, which is **Security**.

5. Focus on process

- Understanding the process rather than what you're building

6. Review

- Review your process
- Review your tests
- Having others review your work (independent, peer review)
    - Removes bias and blindness to own work

7. Professionalism

- You're doing this job because your duty to the profession
- This is different to your duty to your position (i.e. your company asks you to do something)

Conflicts of interest

- Plan for it and address them
- Have systems in place to deal with this

Quantify things

- Come up with metrics and such
- Have specific numbers so you know
- These should be meaningful numbers
    - They should be tested and understood

“Closing the Loop”

- So many systems are simply just ‘feed forward’ (like high school)
    - There’s no feedback coming back
- Closing the loop is simply the idea of **checking that you are right**
- The Art of being an Engineer is the Art of Closing the Loop
- We don’t want wishful thinking, we want **WISE thinking**
- This means that you are often listening for feedback

Aerospace Safety Conference
Story: Accident on the oil rig

- 350 separate alarms went off
- Oil spilled, started fires
- Like Deepwater Horizon or **‘Three Mile Island’**
- Of those 350 alarms that went off, he surmised that only about **8 of them were important**
- So if you imagine that you’re in a control room, and 350 alarms are going off, it’s very hard to figure out what’s going on
    - What are the key ones?
    - This is what Security Engineering is about

System Properties

- It’s easy to be reactive like the barbarian who boxes
- What you need to do is understand and think about systems
    - What goes wrong
- E.g. pilot leaves a wrench in a socket when the airplane leaves
    - Yes that is a mistake
    - However the fault comes from the system which allows people to make mistakes
        - This is like the barbarian
    - A problem will arise next week and someone else will have to go down for it

Russia sub in Cuba

- Needs to surface for air
- Radio signals blocked out
- They’re allowed to launch nuclear missiles because they’re the last line of defence if things go to poop
- When they went down the last they knew was
- Needed a unanimous vote
- One commanding officer was onboard who kept voting against launching
    - Credited to saving the world

What is Coherence?

Coherence: Everything in the unit has to be doing the right thing

- Everything that is related comes together (Object-Oriented principles)
    - Coherence is a system property we want
- Don’t want it to be complex
- Want dont want the system to be tightly coupled
    - Then changes in one affects changes in many others

Youtube: Most unexpected gold medal in history
- Example of a tightly coupled system
- System had a common node of failure

Wargames clip
- Why are humans in the loop?
- Can we replace humans with ASD
    - Automatic decision makers
- ASDs don’t always work (failed in
    - It’s difficult to cut the head
- What if there’s a different problem that happens in which the designer didn’t design the system for?
    - Attackers attack where the system isnt designed for
        - They put the system in a state where the preconditions are not right
    - This is what security engineers are interested in
    - Systems have flaws and vulnerabilities
    - HUMANS design these systems

We need humans in the loop
- All these films are the consequences of such
- So we need to understand **SYSTEMS** as well as **HUMANS**
- So we need psychology, anthropology

Society itself is a system
- The system can be hacked
    - Russians in election
    - Brexit interference

Security is **END TO END**
- In software things are secure in the middle
- Just like the fence (fence post abused, or missing at the end)
- You need it secure all the way around
- E.g. typing password into system
    - Typing the password in is END
    - Attack vector is keyloggers

Work to undermine limits

RESEARCH: How the Romans and the Greeks transferred power

- Anyone in power works to increase their power and subvert the system
- As soon as someone is in power, they work to undo the systems that restrict their power
- Key idea is ensuring that checks and balances exist

This is just like the wareWolf (Buffy da Vampire Slayer)

- Every night they know they'll be turning into a werewolf
- So they lock themselves in before to prevent them from attacking anyone
- Warewolf spends the time trying to get out and undo the restrictions (cage)
- So we're hoping that the cage (system) is strong enough to stop people in power from undoing/overcome these systems

Multilateral security designs

- Like the Bell-Lapadula System
  - Person on top controlling the rest
- If the person on top goes crazy you're in trouble (Single point of failure)

Assange and Free Speech

- People fighting for free speech
- But trying to lock Assange up
- You want free speech, except that is against you (Conflict of interest)

Richard is in INDIA NEXT WEEK

- Adam running on privacy
- Then lachlan is doing his thing
- Guest lecture :(

Cars and Trolly Cars

- When Google/Uber went to parliament and talked about Self-driving cars
- They did some misdirection
- Someone asked the question "What if it the system gets hacked?"
  - Yes there are dangers
  - But we have top people working on it
  - We have men on the moon and planes
  - Not an impossible problem
  - "Just trust us"
  - Distraction 1: probably sometimes cars will kill people
    - But if you look at the road, people die all the time
    - Random statistics about car deaths as a result of human error

    - Distraction 2: the really interesting thing is the decisions that the cars are going to make in an ethical decision
        - Ethical dilemmas are so hard we really don't know the answer to that question (the trolley cart problem)
- Let's not think about the hacker problem (MISDIRECTED)

How can cars be safe, if one of the biggest software companies in the world can't fully secure their phones, email systems, etc.

- If it gets hacked, they have control over the system
- There are systemic flaws
- How do the cars know what the speed limit is?
    - Centralised data-base from GPS can be hacked and changed
    - What's the end-to-end secure solution?

How to ace the exam:
good question! this reminds me of a really interesting problem to do with the number 7

No trusted third parties

- There are bad people everywhere
- If you have an M&M security you're at risk to insiders and such
- Security is all built on trust (Things work and operate on trust)
- If you can't trust anyone you need walls everywhere
- This breach of trust is the real problem
- So we set things up so when things do fail, they fail small (minimal impact)

Ideas for your future from Richard

1. Community
    a. Help each other
    b. Listen and learn
2. Self-directed learning
    a. Learn for yourself
    b. Think about what you need to do (be critical)
3. Exercise
    a. Practice, practice!
    b. If you don't exercise you get fat
    c. Don't just do what you're assigned to do
4. Professionalism
    a. Do what is right for Cyber Security
    b. Grow and become professionals

Cyber Education and Careers Conference and CTF (Sep 20)

- Contact Anatoli
- Print a business card of yours and hand it out

Last lecture with richard tonight :(

Reversing & Cracking Seminar

WannaCry malware

- Encrypted your data, demanded payment in Bitcoin
- Marcus Hutchins reversed it and found the killswitch

What is reverse engineering?

- Taking an executable program and analysing its inside
    - E.g. constructing a recipe of an already baked cake

How to Reverse Engineer

- **Static**
    - Look through code (Assembly)
    - Ghidra
    - Binary Ninja
    - Radare2
    - GDB
    - Ida
    - Hopper
- **Dynamic**
    - Walkthrough the program during runtime
        - GDB is useful for this
    - Useful for obfuscated programs

Typically used together to maximise effectiveness

How2Reversing

1. Run the program first (don't do this with malware!!!)
2. Put the program in a disassembler (e.g. BinaryNinja)
3. Look at what the program is doing
    a. MOVs before function call is setting up arguments
4. Find exploit
5. Profit???

Tips

- Look at the big picture
    - Don't get caught in the nitty gritty
- Many tools let you modify and patch the code to test ideas
- Run beforehand where possible
- Work your way up
- Focus on areas of interest only
    - E.g. reversing a password, look for where the authentication is done, and reverse your way back up the call tree

Practice

- Crackmes/CTFs
  - PicoCT
  - ioli Crackmes
  - Jazz’s crackmes
- Reversing programs on your computer like hello world

Youtube

- OA Labs
- Malware analysis for hedgehogs

Cracking/Patching

- A step after reversing (ILLEGAL - without permission)
- Patching is when you change the binary to make the software do what you want
  - E.g. removing the authentication step of the password

Further Knowledge

The Stack

- Grows upwards (except in Australia)
- Push, pop, ebp, esp
- Has entry and exit procedures (prologue, epilogue)
- Stack frames are like a contained stack for the function scope
  - This way stacks can be called and exited cleanly without manipulating the main stack

Preventing Reversing

1. Don’t Release the Debug Build!

- Remove symbol table from the binary
  - Gcc -s prog.c
- Dump symbols
- Disable asserts
  - Leaks information about the asserts
  - Gcc -DNDEBUG prog.c
- Watch videos of Tomb Raider on PS1

2. Trick the Diassembler

- Make the program as convoluted as possible
- Dummy instructions
- Excess jumps to make spaghet (**SOMEBODY STOLE MINE**)
- Overlapping instructions
- Self modifying code (runtime encryption)

- Jump halfway into add instruction (to hide the return call) <- embedded instructions

Main Lecture

Final Exam QA

- 3 hour exam :(
- No sub-cipher for the final exam
- Extra question for 6841
- FOI and GIPA requests is relevant to the course
- 6841 Seminars are also examinable
    - E.g. obfuscate to make reversing hard

Adam talking about Privacy next week (skipped)

“Canary in a coal mine”

- People used to take canaries in the coal mine in them
- The canary would die very quickly to indicate where the dangerous fumes are
- So it served as an indicator for danger

Opal is tracking your taps even when using credit cards

- Credit cards can now be used with DISCOUNTS!
- This means that they’re tracking our identities

Companies collect data and sell them to brokers for a lot

- “They’re de-identified of course”
- Our phones are constantly monitoring our data and sending it out
- ‘NORMAL’ PEOPLE CAN BUY THIS
- People pay for these surveillance systems (phones) and even more money to replace them

“Data needs to be free”

- They won’t use their data
- They’re talking about the data about you and me (the public)

Each individual bit isn’t dangerous

- The aggregation of the data is
- These pools of data are extremely valuable

Surveillance is making it harder for CIA agents to operate

- Just pervasive tracking and the ability to data-match helps to identify and expose them
- Fake identities need digital trails because a person without one is suspicious
- Facial tracking is pervasive, especially in airports
    - Airports now have facial recognition for checking in and passport control
-