Harry Langford
hjel2@cam.ac.uk

Software and Security Engineering Supervision 1

1 Bell LaPadula

- unclassified
- classified
- secret
- top secret

Bell LaPadula is secure:

- You can get running at high level by writing executables hihger or traitors or idiots.
- Someone at unclassified can write to top secret etc.
- In reality you may end up with integrity over code. There can be separate policies for separate types of data. This is only about confidentiality. You don't allow anything that's been odified by an untrusted actor to run code.
- Bell LaPadula does not provide integrity. It's not intended to. You could allow anyone to write to top secret and delete anything.
- The reason you don't allow write down is malware. Mainly malware. This model should be secure even under malware. You ruser being incompetend should not break Bell LaPadula.
- If your security policy is broken by the user downloading some weird software then it's probably not a very good security model.

The BiBa model is for integrity.

2 Buffer Overflow

Buffer overflow is where you write into a buffer and it fills up and writes into adjacent memory. In the worst case you overwrite the adjacent bytes. And that adjacent bytes is all the stuff on your stack. You end up overwriting maybe the program counter of the function. This causes you to return to other parts of the function. You've overwritten the function this now messes up and you can execute whatever you want.

You can make this harder by preventing the program from executing anything that's ever been written. By making pages either executable or writeable. You can no longer execute code. You've now got an integrity guarantee. You want to make data impossible to execute. You can have confidentiality and integrity guarantees in a system – they may just be in different things.

3 Covert Channels

Covert channels on single systems work by leaving shared resources in a particular state.

In a distributed system, a read request is a write. You can use this to send data back to the recipient. In practice this is almost always a good covert channel.

You could read to different files to contain bits "read from file A is 0 and read from file 1 is B".

It's usually impossible to get down below a few hundred bytes per second. This is usually not problematic. The strategy to fight covert channels is to reduce the size of them.

3.1 Difference between Covert Channels and Side Channels

Covert channels and side channels are very similar. They both use the same mechanisms. The difference is:

A covert channel is where two parties are deliberately communicating with each other.

A side channel uses the exact same mechanism – but where one party is a victim and the other is an attacker.

4 Obtaining Passwords

You can send emails from whoever you want. There is not proper authentication. This is usually a feature not a bug. You can just say your email comes from xyz. Almost everyone uses googles or microsofts mail servers. This causes subtle things to happen – for example google can block you from spoofing an email address. There is way less authentication and integrity than you think. Headers are not authenticated in any way.

Don't click links from someone you don't trust. This is wrong. You can spoof email addresses.

You can get infections from pdf's etc. It's firefox/adobe being broken.

Not everyone patches zero days. Not everyone keeps their software up to date.

4.1 Passwords

Systems should remain secure even if users are incompetent – if they use crappy passwords.

Different things have different threat models. For example

4.2 Why are larger companies insecure?

If someone wants to buy stuff on your website then you want the transaction to be as easy as possible. If you make buying as easy as possible then they will do it as possible. Even if you occasionally have to refund someone, then you make more money than you would otherwise.

You deliberately reduce security in the hope that it will cause people to buy stuff.

If you can cost shift then security isn't usually very good. One way this meifests is that the banks will say when you send a bank transfer "make sure this is who you mean to send it to as we won't reverse it if it's a fraud". Bank transfers are free and they don't take a commission for it. Banks don't take commission on bank transfers. Banks don't refund fraud on bank transfer as they don't pay for fraud on bank transfer.

4.3 Actionable Advice

Two factor authentication is good. Although text messages are not too secure. Phones never say they're identity providers.

SS7 hacking; phone networks can claim they're serving your phone now and will then see your text messages.

Alternatives are microsoft Authenticator. This can be awkward if the phone is being authenticated by having the phone.

UB keys (little USB's which hold keys) work. They keep people secure. They're really good if your userbase is somewhat technically competent.

Password policies that change passwords every 30 days are terrible. Good tech companies will make you alter your password only if there has been a breach.

Firefox or chrome will check your password against known leaks and will warn you. However, many people ignore this.

Passwords with elaborate symbols are not that high entropy and are probably not too hard to guess.

Long passwords are usually pretty good.

Don't rely on passwords. They're not that good. They should be one mechanism in a multifactor authentication for anything you seriously care about security in.

4.4 Fault tree analysis vs Failure modes and effects analysis

In Fault trees you work down.

In failure modes and effects analysis you work up. This is bad for human systems. Good for tehnical systems. For example passwords should be used with fault trees and the aeroplane should be failure modes analysis since you have a long list of things that can go wrong.

4.5 Nonces

Nonces can be sequences – since you encrypt them they're random.

You count up. If you've got 64 bits then you'll never wrap around. If you use smaller bits then you just wrap around.

You can just store the range. This also will remove old nonces if you miss some.

4.6 Salts

Salts should be as random as possible. We do not need to reproduce them ever.

We append the salt onto the password and hash that. We then store the salt and the hash.

4.7 Interactive Protocol

These are for keyless protocols. You can use relay attacks. You just relay the nonce to the key and relay the response to the car. The security policy in this case is that they key is nearby. You can violate this assumption by relaying the signal to the car. If you can amplify the signal enough then you can start the car or open the boot etc.

Interactive protocols are not secure. They are only for the purposes of buttonless keys.

You can make them more secure by working out the time between the time you send the signal and the time the key responds. This is really complicated and the propogation time is much smaller than computation time. Alternatively put an accellerometer in a key. So they key doesn't move unless it's moving.

4.8 Cross Protocol Attack

Making some protocols secure can make other protocols less secure. For example, card readers being secure made bank security less secure.

4.9 Long form questions

With these long form questions (8 marks or so), Ross wants structured prose. You should **not** use bullet points. Say the point you're making, then state the evidence.

4.10 Smart meters

The real tricky part is that there's misaligned incentives for smart meters. Smart meters are designed to be confusing and don't do what they're designed to do.

4.11 LASCAD

Management should be leadership. Leadership is about someone's neck being on the line if the project fails. If someone's job matters then they are highly incentivised to make it work. Management's job should depend on the projects they are running. This is leadership. This is necessary, to make sure things go right, you make sure that someone will take the fall if stuff goes wrong.