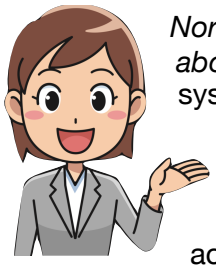
Quantitative Information Flow:

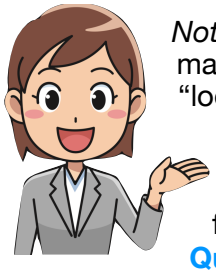
What is it, what does it do, and why should we use it?

Why is designing security for computer systems so hard?



Normally when you design a computer system, you have some idea about what you want it to do. For example if you want to create a system that accepts online payments for purchases, you know that it involves a communication between a customer and a store. This means that the design must include ways to allow that to happen. But to make that secure, you have also to consider "adversaries" who don't follow those rules and are actually trying to find workarounds (aka "vulnerabilities") that they can exploit. When designing a system the designer really has to think like a hacker and come up with all the ways the system can be exploited.

I see… so it looks like the hackers will always win, right?



Not necessarily. Computer Scientists have discovered that using mathematical models to describe what an adversary can do enables many of these "loopholes" that attract unsavoury behaviour to be discovered and then closed before a system is released for public use. This doesn't prevent everything, but it does prevent a lot of things, and at least it makes it much harder for a hacker to inflict damage. **Quantitative Information Flow** (QIF) analysis uses just such a mathematical model to determine how hackers can use any accidental information leaks of secrets (such as password credentials) while a system is in operation. QIF can actually measure the severity and impact of any damage caused by information leaks.

But that seems impossible! All my friends do different things when they try to break in… I mean my friends tell me that hacker groups are like that.

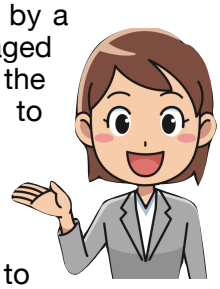
Good point — all hackers do go about their "work" in different ways. I would certainly like to meet your friends to discuss these ideas… However for designing secure systems it's definitely wise to model many different methods of gathering information from leaks. The most important indication of a security vulnerability is whether the adversary can actually use any leaked information. For that it's best to create a model of an adversary that includes things like how hard or costly is it for them to obtain leaked information, and what exactly they are trying to do with it. For instance if someone gets exactly one try to break into a system (by say guessing the password) then that's one thing; but if the someone has say three tries then they can use some of the information they learn in their previous failed tries to help them in their later attempts.

Ok I get that, but still it seems like an impossible task to be able to say "well I've analysed everyone". There will still be some hackers you haven't thought of.

Well perhaps, but as your friends might be able to tell you, hackers are in fact limited in what they can do — to a certain extent. In fact using the theory of quantitative information flow we can determine pretty much the maximum that even the best hacker can learn from any information leaks. We do though need to make some reasonable assumptions about what they might already know (aka prior knowledge), and about their intent (eg guessing any password or guessing a specific one) and, of course, the way the system can be interrogated. In fact what we're most concerned about is not really how effective is the best hacker, working under the most optimal conditions to break the system — such a hacker might already have an awful lot of prior knowledge — but rather how the information vulnerabilities in the system contributed to that über hacker's effort. We can actually estimate how much additional information that über hacker can learn.

What?! Wait — You know about The Über? Hang on — you are saying that there's some upper bound on how much information can leak?

Well the mathematical theory of QIF contains an important variation of an idea that was first used in engineering and invented by a great computer Scientist called Claude Shannon. Shannon envisaged a communication channel that was "noisy" in the sense that the information going through the channel was perturbed due to physical distortions in radio waves.



And for security analysis we can similarly model the information leaks in terms of a noisy channel but there is a *big difference* between trying to get as much information through a noisy channel and trying to prevent information going through a noisy channel. In the first instance you're looking at a very specific scenario: that of choosing a rate of transmission to deliver the message. In the latter you have to consider all the scenarios we talked about above, and we want to make sure that whoever the hacker is, they don't get enough benefit from the system to make it worth their while to attack it. However just as in Shannon's theory QIF also has an important notion of Capacity — the **QIF multiplicative capacity** says that for a security system modelled as a noisy channel there is a precise maximum amount of extra information that can be usefully learned. And it applies to everyone — even, erm, The Über.

So it's just Shannon capacity, right?

Good guess, but actually totally wrong! In fact if you use Shannon's Information Theory that will just give you the answer to the problem that Shannon was trying to solve. And in fact the Shannon capacity gives a very optimistic view of security in the sense of protecting the system from hackers. There are many examples however that show how the Shannon estimate would judge a system to leak hardly any useful information at all when *really* it's leaking all of the secret often enough to be problematic for a lot of people. It's like saying that the system is 99% secure, but 1% of users will have their identity stolen. That sounds ok but 1% of millions of people is still a lot of people with stolen identities. In fact QIF Multiplicative capacity is given by something called **Bayes Vulnerability**, and it would pick up that 1% of people and alert the system designer to redo their design.

So are you saying that QIF can be used by designers as a way to evaluate *their security system BEFORE they put it out?*

But wait — that means that QIF could tell the designers something about hacking…I mean that seems so…

… unfair? Well, maybe: yet I like to think of it as levelling the playing field. But actually you are quite right! QIF is a way of thinking about security vulnerabilities and their potential impact. It's an example of a Formal Method for security, and it addresses one of the trickiest questions of how to think about security when confidential information is only partially released. That might not seem so problematic for a single specific person, but actually when a system is used by a lot of people, a lot of people could be affected, even if we don't know who they will turn out to be at the outset.