**Misdirection**

**Misdirection** is where the con-artist / magician / social-engineer, someone who is trying to trick another human, gets them to pay attention to the wrong thing and calms down their interest and alertness and arousal about the right thing, so you only notice the WRONG THING.

Misdirection Philosophy

* Don’t draw attention to things unnecessarily, unless it is the deliberately wrong thing that you want people to focus on.
* Show tricks / misdirection in a way that makes it LOOK **natural**, so that your attention is NOT directed towards what it should be directed.
* You need to misdirect in a very natural way so everyone is drawn towards the misdirection and **won’t become suspicious**.

Misdirection can happen in a lot of places.

* E.g. focusing on misdirecting airport security to the passengers coming into the airport, rather than the flight crew
* **Security Theatre:** We put all our attention and energy to make sure that the one spot is right, but the little obvious, natural thing on the side where people can just walk in and out as they want because they have a small plastic ID card, that doesn’t get any attention
* When that is discovered and solved, it doesn’t matter because now we’re focusing on those two things, we’re still not looking at the other 60 things because we are bad at being attentive to everything and can only focus on a few things at once.

**Kevin Mitnick Attack**

Mitnick’s hack of Tsutomu Shimomura.

* 1994 Christmas, Shim left his computer and has gone to do some Christmas stuff
* It involved TCP / IP
  + They are two protocols for networking
  + Everyone should learn a bit of TCP / IP, because a lot of attacks are launched over the internet and they require knowing level 2/3 of TCP/IP, so you know what a TCP / IP packet looks like etc.
* IP is lowdown
* TCP is one level higher than that.
* **Good book about TCP/IP: Internetworking with TCP/IP Volume One by Douglas Comer**

**TCP**

* One of the protocols used when a computer somewhere wants to talk to a computer somewhere else, by passing messages through lots of intermediate machines along the way.
* We only care about the source of the whole thing / destination.
  + The destination is possibly a web-server, you are serving up web pages
  + The source is possibly someone who wants to read a web-page, so sends a request from his/her browser by typing in a URL that bounces around the internet in all sorts of ways and eventually ends up at the web-server, telling it that he/she wants a particular web-page.
  + Imagine it is Wikipedia that the source is requesting, so Wikipedia sends the web-page back to him/her.
  + The source’s browser gets the web-page and displays it.
  + **^This is a high-level overview of what is happening.**
  + **With networking, we go from high-level 🡪 low-level**
* Suppose you are looking at a big-page, with many many kilobytes of data on it.
  + The source sends a packet up, telling the destination what they would like 🡪 the destination sends a packet of data back with the “answer”
  + **Packets are of a certain size, they can’t get bigger than the maximum size of X kilobytes**
  + **Anything bigger than X kilobytes, the data has to be broken down into multiple chunks of data**
  + All the packets travel a particular way back and forth from the source 🡪 destination
  + Each of the packets are posted through a “postal system” which has an address of it.
  + Intermediate people get the address of the packets to look at it and then it gets passed on all the way until it reaches its destination.
  + The structure of this whole thing / these packets travelling around are doing something called IP.
  + TCP sits on top of IP.
    - **TCP is about not just sending a packet, but having a connection**.
    - Different from someone sending a letter.
    - With a letter, you have to renegotiate and decide again if you want to talk or not.
    - With a connection, once you go through all the overhead of setting it up, its very lightweight to keep communication going.
  + **TCP has the same protocol to set up an initial connection by sending three packets**
    - A source wants to connect to a web-server, so they send one packet saying:
      1. **S-Y-N packet** **(synchronise)**  
         “Hey, I want to initiate a connection with you”
      2. **SYN + ACK packet** **(synchronise + acknowledge)**We want the convo to go two ways, so sends both SYN and ACK)  
         “Okay I will connect with you”