# Chapter 3

# Psychology and Usability

**Humans are incapable of securely storing high-quality cryptographic keys, and they have unacceptable speed and accuracy when performing cryptographic operations. (They are also large, expensive to maintain, difficult to manage, and they pollute the environment. It is astonishing that these devices continue to be manufactured and deployed. But they are sufficiently pervasive that we must design our protocols around their limitations.)**
**– KAUFMANN, PERLMAN AND SPECINER [769]**

**Only amateurs attack machines; professionals target people.**
**– BRUCE SCHNEIER**

**Metternich told lies all the time, and never deceived any one; Talleyrand never told a lie and deceived the whole world.**
**– THOMAS MACAULAY**

## 3.1 Introduction

Many real attacks exploit psychology at least as much as technology. We saw in the last chapter how some online crimes involve the manipulation of angry mobs, while both property crimes and espionage make heavy use of *phishing*, in which victims are lured by an email to log on to a website that appears genuine but that's actually designed to steal their passwords or get them to install malware.

Online frauds like phishing are often easier to do, and harder to stop, than similar real-world frauds because many online protection mechanisms are neither as easy to use nor as difficult to forge as their real-world equivalents. It's much easier for crooks to create a bogus bank website that passes casual inspection than to build an actual bogus bank branch in a shopping street.

We've evolved social and psychological tools over millions of years to help us deal with deception in face-to-face contexts, but these are less effective when we get an email that asks us to do something. In an ideal technology, good use would

be easier than bad use. We have many examples in specialised physical objects: a potato peeler is easier to use for peeling potatoes than a knife is, but a lot harder to use for murder. But we've not always got this right for computer systems yet. Much of the asymmetry between good and bad on which we rely in our daily business doesn't just depend on formal exchanges – which can be automated easily – but on some combination of physical objects, judgment of people, and the supporting social protocols. So, as our relationships with employers, banks and government become more formalised via online communication, and we lose both physical and human context, the forgery of these communications becomes more of a risk.

Deception, of various kinds, is now the principal mechanism used to defeat online security. It can be used to get passwords, to compromise confidential information or to manipulate financial transactions directly. Hoaxes and frauds have always happened, but the Internet makes some of them easier, and lets others be repackaged in ways that may bypass our existing controls (be they personal intuitions, company procedures or even laws).

Another driver for the surge in attacks based on social engineering is that people are getting better at technology. As designers learn how to forestall the easier technical attacks, psychological manipulation of system users or operators becomes ever more attractive. So the security engineer absolutely must understand basic psychology, as a prerequisite for dealing competently with everything from passwords to CAPTCHAs and from phishing to social engineering in general; a working appreciation of risk misperception and scaremongering is also necessary to understand the mechanisms underlying angry online mobs and indeed terrorism. So just as research in security economics led to a real shift in perspective between the first and second editions of this book, research in security psychology has made much of the difference to how we view the world between the second edition and this one.

In the rest of this chapter, I'll first survey relevant research in psychology, then work through how we apply the principles to make password authentication mechanisms more robust against attack, to security usability more generally, and beyond that to good design.

## 3.2 Insights from Psychology Research

Psychology is a huge subject, ranging from neuroscience through to clinical topics, and spilling over into cognate disciplines from philosophy through artificial intelligence to sociology. Although it has been studied for much longer than computer science, our understanding of the mind is much less complete: the brain is so much more complex. There's one central problem – the nature of consciousness – that we just don't understand at all. We know that 'the mind is what the brain does', yet the mechanisms that underlie our sense of self and of personal history remain obscure.

Nonetheless a huge amount is known about the functioning of the mind and the brain, and we're learning interesting new things all the time. In what follows I can only offer a helicopter tour of three of the themes in psychology

research that are very relevant to our trade: cognitive psychology, which studies topics such as how we remember and what sort of mistakes we make; social psychology, which deals with how we relate to others in groups and to authority; and behavioral economics, which studies the heuristics and biases that lead us to make decisions that are consistently irrational in measurable and exploitable ways.

### 3.2.1 Cognitive psychology

Cognitive psychology is the classical approach to the subject – building on early empirical work in the nineteenth century. It deals with how we think, remember, make decisions and even daydream. Twentieth-century pioneers such as Ulric Neisser discovered that human memory doesn't work like a video recorder; our memories are stored in networks across the brain, from which they are reconstructed, so they change over time and can be manipulated [1057]. There are many well-known results. For example, it's easier to memorise things that are repeated frequently, and it's easier to store things in context. Many of these insights are misunderstood or just ignored by system developers.

For example, most of us have heard of George Miller's result that human short-term memory can cope with about seven (plus or minus two) simultaneous choices [983] and, as a result, many designers limit menu choices to about five. But this is not the right conclusion. People search for information first by recalling where to look, and then by scanning; once you've found the relevant menu, scanning ten items is only twice as hard as scanning five. The real limits on menu size are screen size, which might give you ten choices, and with spoken menus, where the average user has difficulty dealing with more than three or four [1147]. Here, too, Miller is misused because spatio-structural memory is a different faculty from echoic memory. This illustrates why a broad idea like 7+/-2 can be hazardous; you need to look at the detail.

So, in recent years, the centre of gravity in this field has been shifting from applied cognitive psychology to the human-computer interaction (HCI) research community, because of the huge amount of empirical know-how gained not just from lab experiments, but from the iterative improvement of fielded systems. As a result, HCI researchers not only model and measure human performance, including perception, motor control, memory and problem-solving; they have also developed an understanding of how users' mental models of systems work, how they differ from developers' mental models, and of the techniques (such as task analysis and cognitive walkthrough) that we can use to explore how people learn to use systems and understand them.

Security researchers need to find ways of turning these ploughshares into swords (the bad guys are already working on it). There are some low-hanging fruit; for example, the safety research community has put a lot of effort into studying the errors people make when operating equipment [1174]. It's said that 'to err is human' and error research confirms this: the predictable varieties of human error are rooted in the very nature of cognition. The schemata, or mental models, that enable us to recognise people, sounds and concepts so much better than computers, also make us vulnerable when the wrong model gets activated.

Human errors made while operating equipment fall into broadly three categories, depending on where they occur in the 'stack': slips and lapses at the level of skill, mistakes at the level of rules, and mistakes at the cognitive level.

- Actions performed often become a matter of skill, but we can slip when a manual skill fails – for example, pressing the wrong button – and we can also have a lapse where we use the wrong skill. For example, when you intend to go to the supermarket on the way home from work you may take the road home by mistake, if that's what you do most days (this is also known as a *capture error*). Slips are exploited by typosquatters, who register domains similar to popular ones, and harvest people who make typing errors; other attacks exploit the fact that people are trained to click 'OK' to pop-up boxes to get their work done. So when designing a system you need to ensure that dangerous actions, such as installing software, require action sequences that are quite different from routine ones. Errors also commonly follow interruptions and perceptual confusion. One example is the *post-completion error*: once they've accomplished their immediate goal, people are easily distracted from tidying-up actions. More people leave cards behind in ATMs that give them the money first and the card back second.

- Actions that people take by following rules are open to errors when they follow the wrong rule. Various circumstances – such as information overload – can cause people to follow the strongest rule they know, or the most general rule, rather than the best one. Phishermen use many tricks to get people to follow the wrong rule, ranging from using `https` (because 'it's secure') to starting URLs with the impersonated bank's name, as `www.citibank.secureauthentication.com` – since looking for a name is a stronger rule for many people than parsing its position.

- The third category of mistakes are those made by people for cognitive reasons – either they simply don't understand the problem, or pretend that they don't, and ignore advice in order to get their work done. The seminal paper on security usability, Alma Whitten and Doug Tygar's "Why Johnny Can't Encrypt", demonstrated that the encryption program PGP was simply too hard for most college students to use as they didn't understand the subtleties of private versus public keys, encryption and signatures [1489]. And there's growing realisation that many security bugs occur because most programmers can't use security mechanisms either. Both access control mechanisms and cryptographic APIs are hard to understand and fiddly to use; security testing tools are often not much better. Programs often appear to work even when using protection mechanisms in quite mistaken ways. Engineers then copy code from each other, and from online code sharing sites, so misconceptions and errors are propagated widely [9]. They probably know this is bad, but there's just not the time to do better.

What makes security harder than safety is that we have a sentient attacker who will try to provoke exploitable errors.

What can the defender expect attackers to do? They will use errors whose effect is predictable, such as capture errors; and to look for, or create, exploitable dissonances between users' mental models of a system and its actual logic. To look for these, you should try a cognitive walkthrough aimed at identifying attack points, just as a code walkthough can be used to search for software vulnerabilities. Attackers also learn by experiment and share techniques with each other, and develop tools to look efficiently for known attacks. So it's important to be aware of the attacks that have already worked. (That's one of the functions of this book.)

### 3.2.2 Gender, diversity and interpersonal variation

Most information systems are designed by men, and young geeky men at that. Yet over half their users may be women. The realisation that software can create barriers to females has led to some research on *gender HCI* – on how software should be designed so that women as well as men can use it effectively. For example, it's known that women navigate differently from men in the real world, using peripheral vision more, and it duly turns out that larger displays reduce gender bias. Other work has focused on female programmers, especially end-user programmers working with tools like spreadsheets. It turns out that women tinker less than males, but more effectively [160]. US women who program appear to be more thoughtful, but lower self-esteem and higher risk-aversion leads them to use fewer features. Societal factors matter; while about a sixth of computer science students are women in the USA and the UK, it's closer to a third in Eastern Europe and near to a half in India. So we need to differentiate between sex (whether you have a Y chromosome) and gender (which subsumes the social factors as well as the physiological ones). Gender matters for product design.

There's some interesting work on attitudes to risk. In US surveys, risks are judged lower by white people and by men, and on closer study this is because about 30% of white males judge risks to be extremely low. This bias is consistent across a wide range of hazards but is particularly strong for handguns, second-hand cigarette smoke, multiple sexual partners and street drugs. Asian males show similarly low sensitivity to some hazards, such as motor vehicles. White males are more trusting of technology, and less of government [512].

No-one seems to have done much work on gender and security usability, yet reviews of work on gender psychology (such as [1116]) suggest many points of leverage. One formulation, by Simon Baron-Cohen, gives people separate scores as systemisers (good at geometry and some kinds of symbolic reasoning) and as empathisers (good at intuiting the emotions of others and social intelligence generally) [139]. Most men score higher at systematising, while most women do better at empathising. Of course, innate abilities can be modulated by many developmental and social factors, but we have to work with the world as it is, not as it might be if our education system had less bias. Tyler Moore and I surveyed 132 volunteers (77 female, 55 male) who did Baron-Cohen's test and also a test of their ability to spot phishing scams by parsing the URL in the browser toolbar. It turned out that systemisers are more likely to spot scams using this technique, and (statistically independently of this) so are men. This

raises the question of whether it is unlawful sex discrimination for a bank to expect its customers to detect phishing attacks by parsing URLs [997].

More generally, systems tend to be designed by young fit white straight men who may not think hard or at all about the various forms of prejudice and disability that they do not encounter directly. Diversity is valuable in research, in development teams and in testing. As many of these factors are partly or wholly of social origin, this brings us to social psychology.

### 3.2.3 Social psychology

This attempts to explain how the thoughts, feelings, and behaviour of individuals are influenced by the actual, imagined, or implied presence of others. It has many aspects, from the identity that people derive from belonging to groups, through the self-esteem we get by comparing ourselves with others. The results that put it on the map were three early papers that laid the groundwork for understanding the abuse of authority and its relevance to propaganda, interrogation and aggression. They were closely followed by work on the bystander effect which is also highly relevant to crime and security.

#### 3.2.3.1 Authority and its abuse

In 1951, Solomon Asch showed that people could be induced to deny the evidence of their own eyes in order to conform to a group. Subjects judged the lengths of lines after hearing wrong opinions from other group members, who were actually the experimenter's stooges. Most subjects gave in and conformed, with only 29% resisting the bogus majority [109].

Stanley Milgram was inspired by the 1961 trial of Nazi war criminal Adolf Eichmann to investigate how many experimental subjects were prepared to administer severe electric shocks to an actor playing the role of a 'learner' at the behest of an experimenter while the subject played the role of the 'teacher' – even when the 'learner' appeared to be in severe pain and begged the subject to stop. This experiment was designed to measure what proportion of people will obey an authority rather than their conscience. Most did – Milgram found that consistently over 60% of subjects would do downright immoral things if they were told to [980]. This experiment is now controversial but had real influence on the development of the subject.

The third was the Stanford Prisoner Experiment which showed that normal people can behave wickedly even in the absence of orders. In 1971, experimenter Philip Zimbardo set up a 'prison' at Stanford where 24 students were assigned at random to the roles of 12 warders and 12 inmates. The aim of the experiment was to discover whether prison abuses occurred because warders (and possibly prisoners) were self-selecting. However, the students playing the role of warders rapidly became sadistic authoritarians, and the experiment was halted after six days on ethical grounds [1529]. This experiment is also controversial now and it's unlikely that a repeat would get ethical approval today. But abuse of authority, whether real or ostensible, is a real issue if you are designing operational security measures for a business.

During the period 1995–2005, a telephone hoaxer calling himself 'Officer Scott' ordered the managers of over 68 US stores and restaurants in 32 US states (including at least 17 McDonald's stores) to detain some young employee on suspicion of theft and strip-search her or him. Various other degradations were ordered, including beatings and sexual assaults [1501]. A former prison guard was tried for impersonating a police officer but acquitted. At least 13 people who obeyed the caller and did searches were charged with crimes, and seven were convicted. McDonald's got sued for not training its store managers properly, even years after the pattern of hoax calls was established; and in October 2007, a jury ordered them to pay $6.1 million dollars to one of the victims, who had been strip-searched when she was an 18-year-old employee. It was a nasty case, as she was left by the store manager in the custody of her boyfriend, who then committed a further indecent assault on her. The boyfriend got five years, and the manager pleaded guilty to unlawfully detaining her. McDonald's argued that she was responsible for whatever damages she suffered for not realizing it was a hoax, and that the store manager had failed to apply common sense. A Kentucky jury didn't buy this and ordered McDonald's to pay up. The store manager also sued, claiming to be another victim of the firm's negligence to warn her of the hoax, and got $1.1 million [815]. So as of 2007, US employers risk heavy damages if they fail to train their staff to resist the abuse of authority.

#### 3.2.3.2 The bystander effect

On March 13, 1964, a young lady called Kitty Genovese was stabbed to death in the street outside her apartment in Queens, New York, and the press reported that thirty-eight separate witnesses had failed to help or even call the police, although the assault lasted almost half an hour. Although these reports were later found to be exaggerated, the crime led to the nationwide 911 emergency number, and also to research on why bystanders often don't get involved.

John Darley and Bibb Latané reported experiments in 1968 on what factors modulated the probability of a bystander helping someone who appeared to be having an epileptic fit. They found that a lone bystander would help 85% of the time, while someone who thought that four other people could see the victim would help only 31% of the time; group size dominated all other effects. Whether another bystander was male, female or even medically qualified made essentially no difference [387]. The diffusion of responsibility has visible effects in many other contexts. If you want something done, you'll email one person to ask, not three people. Of course, security is usually seen as something that other people deal with.

However, if you ever find yourself in danger, the real question is whether at least one of the bystanders will help, and here the recent research is much more positive. Lasse Liebst, Mark Levine and others have surveyed CCTV footage of a number of public conflicts in several countries over the last ten years, finding that in 9 out of 10 cases, one or more bystanders intervened to de-escalate a fight, and that the more bystanders intervene, the more successful they are [877]. So it would be wrong to assume that bystanders generally pass by on the other side; so the bystander effect's name is rather misleading.

### 3.2.4 The social-brain theory of deception

Our second big theme, which also fits into social psychology, is the growing body of research into deception. How does deception work, how can we detect and measure it, and how can we deter it?

The modern approach started in 1976 with the social intelligence hypothesis. Until then, anthropologists had assumed that we evolved larger brains in order to make better tools. But the archaeological evidence doesn't support this. All through the old stone age, while our brains evolved from chimp size to human size, we used the same simple stone axes. They only became more sophisticated in the new stone age, by which time our ancestors were anatomically modern homo sapiens. So why, asked Nick Humphrey, did we evolve large brains if we didn't need them yet? Inspired by observing the behaviour of both caged and wild primates, his hypothesis was that the primary function of the intellect was social. Our ancestors didn't evolve bigger brains to make better tools, but to use other primates better as tools [694]. This is now supported by a growing body of evidence, and has transformed psychology as a discipline. Social psychology had been a poor country cousin until then and was seen as non-rigorous; since then, people have realised it was probably the driver of evolution. Almost all intelligent species developed in a social context. (One exception is the octopus, but it has to understand how predators and prey react.)

The primatologist Andy Whiten then collected much of the early evidence on tactical deception, and renamed it the Machiavellian brain hypothesis: we became smart in order to deceive others, and to detect deception too [286]. Not everyone agrees completely with this characterisation, as the positive aspects of socialisation, such as empathy, also matter. But Hugo Mercier and Dan Sperber have recently collected a lot of evidence that the modern human brain is more a machine for arguing than anything else [965]. Our goal is persuasion rather than truth; rhetoric comes first, and logic second.

The second thread coming from the social intellect hypothesis is theory of mind, an idea due to David Premack and Guy Woodruff in 1978 but developed by Heinz Wimmer and Josef Perner in a classic 1983 experiment to determine when children are first able to tell that someone has been deceived [1499]. In this experiment, the Sally-Anne test, a child sees a sweet hidden under a cup by Sally while Anne and the child watch. Anne then leaves the room and Sally switches the sweet to be under a different cup. Anne then comes back and the child is asked where Anne thinks the sweet is. Normal children get the right answer from about age five; this is when they acquire the ability to discern others' beliefs and intentions. Simon Baron-Cohen, Alan Leslie and Uta Frith then showed that children on the Aspergers / autism spectrum acquire this ability significantly later [140].

Many computer scientists and engineers appear to be on the spectrum to some extent, and we're generally not as good at deception as neurotypical people are. This has all sorts of implications! We're under-represented in politics, among senior executives and in marketing. Oh, and there was a lot less cyber-crime before underground markets brought together geeks who could write wicked code with crooks and spooks who could use it for wicked purposes. Geeks are also more likely to be whistleblowers; we're less likely to keep quiet about an

uncomfortable truth just to please others, as we place less value on the opinions of others. But this is a complex field. Some well-known online miscreants who are on the spectrum were hapless more than anything else; Gary McKinnon claimed to have hacked the Pentagon to discover the truth about flying saucers and didn't anticipate the ferocity of the FBI's response. And other kinds of empathic deficit are involved in many crimes. Other people with dispositional empathy deficits include psychopaths who disregard the feelings of others but understand them well enough to manipulate them, while there are many people whose deficits are situational, ranging from Nigerian scammers who think that any white person who falls for their lure deserves it as they must be a racist, to soldiers and terrorists who consider their opponents to be less than human or to be morally deserving of death. I'll discuss radicalisation more later in section 24.4.2.

The third thread is self-deception. Robert Trivers argues that we've evolved the ability to deceive ourselves in order to better deceive others: "If deceit is fundamental in animal communication, then there must be strong selection to spot deception and this ought, in turn, select for a degree of self-deception, rendering some facts and motives unconscious so as to not betray – by the subtle signs of self-knowledge – the deception being practiced" [670]. We forget inconvenient truths and rationalise things we want to believe. There may well be a range of self-deception abilities from honest geeks through to the great salesmen who have a magic ability to believe completely in their product. But it's controversial, and at a number of levels. For example, if Tony Blair really believed that Iraq had weapons of mass destruction when he persuaded Britain to go to war in 2003, was it actually a lie? How do you define sincerity? How can you measure it? And would you even elect a national leader if you expected that they'd be unable to lie to you? There is a lengthy discussion in [670], and the debate has led to further work on motivated reasoning. Russell Golman, David Hagman and George Loewenstein survey research on how people avoid information, even when it is free and could lead to better decision-making: people at risk of illness avoid medical tests, managers avoid information that might show they made bad decisions, and investors look at their portfolios less when markets are down [584]. This links up with filter-bubble effects on social media. People prefer to listen to others who confirm their beliefs and biases, and this can be analysed in terms of the hedonic value of information. Oh, and criminologists use the term *neutralisation* to describe the strategies that criminals use to minimise the guilt that they feel about their actions; there's an overlap with both filter effects and self-deception. A further link is to Hugo Mercier and Dan Sperber's work on the brain as a machine for argument, which I mentioned above.

The fourth thread is intent. The detection of hostile intent was a big deal in our ancestral evolutionary environment; in pre-state societies, perhaps a quarter of men and boys die of homicide, and further back many of our ancestors were killed by animal predators. So we've evolved a sensitivity to sounds and movements that might signal the intent of a person, an animal or even a god. This can lead us to spend too much defending against threats that involve hostile intent, such as terrorism, and not enough against epidemic disease, which kills many more people. At the other end of the scale are the philosophers of mind, who now build on the idea of intent that's inherent once we realise that

people use theories of mind to understand each other. Dan Dennett derived the intentional stance in philosophy, arguing that the propositional attitudes we use when reasoning – beliefs, desires and perceptions – come down to the intentions of people and animals. In our own field, we use logics of belief to analyse the security of authentication protocols, and have to deal with statements such as 'Alice believes that Bob believes that Charlie controls the key $K$'; we'll come to this in the next chapter.

A related matter is socially-motivated reasoning: people do logic much better if the problem is set in a social role. In the Wason test, subjects are told a rule such as "If a student has a grade D on the front of their card, then the back must be marked with code 3" and shown four cards displaying (say) D, F, 3 and 7. They are then asked "Which cards do you have to turn over to check that all cards are marked correctly?" Most subjects get this wrong; in the original experiment, only 48% of 96 subjects got the right answer of D and 7. However the evolutionary psychologists Leda Cosmides and John Tooby found the same problem becomes easier if the rule is changed to 'If a person is drinking beer, he must be 20 years old' and the individuals are a beer drinker, a coke drinker, a 25-year-old and a 16-year old. Now three-quarters of subjects deduce that the bouncer should check the age of the beer drinker and the drink of the 16-year-old [369]. Cosmides and Tooby argue that our ability to do logic and perhaps arithmetic evolved as a means of policing social exchanges.

The next factor is rationalisation or minimsation – the process by which people justify bad actions or make their harm appear to be less. I mentioned Nigerian scammers who think that white people who fall for their scam deserve it, as they must think Africans are stupid; there are many more examples of scammers seeing foreign targets as fair game. The criminologist Donald Cressey developed a *Fraud Triangle* theory to explain the factors that lead to fraud: as well as motive and opportunity, there must be a rationalisation. People may feel that their employer has underpaid them so it's justifiable to fiddle expenses, or that the state is wasting money on welfare when they cheat on their taxes. This is very common in cybercrime; kids operating DDoS-for-hire services reassured each other that offering a 'web stresser' service was legal, and said on their websites that the service could only be used for legal purposes. The countermeasure was for the UK National Crime Agency to buy Google ads so that anyone searching for a web stresser service would get an official warning that DDoS was a crime. A mere £3,000 spent between January and June 2018 suppressed demand growth; DDoS revenues remained constant in the UK while they grew in the USA.

Finally, the loss of social context is a factor in online disinhibition. People speak more frankly online, and this has both positive and negative effects; shy people can find partners but we also see vicious flame wars. John Suler analyses the factors as anonymity, invisibility, asynchronicity and the loss of symbols of authority and status; in addition there are effects relating to psychic boundaries and self-imagination which lead us to drop our guard and express feelings from affection to aggression that we normally rein in for social reasons [1361]. You might say that we are less motivated to deceive.

### 3.2.5 Heuristics, biases and behavioural economics

One field of psychology that has been applied by security researchers since the mid-2000s has been *decision science*, which sits at the boundary of psychology and economics and studies the heuristics that people use, and the biases that influence them, when making decisions. It is also known as *behavioural economics*, as it examines the ways in which people's decision processes depart from the rational behaviour modeled by economists. An early pioneer was Herb Simon – both a computer scientist and an economist – who noted that classical rationality meant doing whatever maximizes your expected utility regardless of how hard that choice is to compute; so how would people behave in a realistic world of bounded rationality? The real limits to human rationality have been explored extensively in the years since, and Daniel Kahneman won the Nobel prize in economics in 2002 for his major contributions to this field (along with the late Amos Tversky) [746].

#### 3.2.5.1 Prospect theory and risk misperception

Kahneman and Tversky did extensive experimental work on how people made decisions faced with uncertainty. They first developed *prospect theory* which models risk appetite: in many circumstances, people dislike losing $100 they already have more than they value winning $100. Framing an action as avoiding a loss can make people more likely to take it; phishermen hook people by sending messages like 'Your PayPal account has been frozen, and you need to click here to unlock it.' We're also bad at calculating probabilities, and use all sorts of heuristics to help us make decisions:

- we often base a judgment on an initial guess or comparison and then adjust it if need be – the *anchoring effect*;
- we base inferences on the ease of bringing examples to mind – the *availability heuristic*, which was OK for lion attacks 50,000 years ago but gives the wrong answers when mass media bombard us with images of terrorism;
- we're more likely to be sceptical about things we've heard than about things we've seen, perhaps as we have more neurons processing vision;
- we worry too much about events that are very unlikely but have very bad consequences;
- we're more likely to believe things we've worked out for ourselves rather than things we've been told.

Behavioral economics is not just relevant to working out how likely people are to click on links in phishing emails, but to the much deeper problem of the perception of risk. Many people perceive terrorism to be a much worse threat than epidemic disease, food poisoning or road traffic accidents: this is wrong, but hardly surprising to a behavioural economist. We overestimate the small risk of dying in a terrorist attack not just because it's small but because of the visual effect of the 9/11 TV coverage, the ease of remembering the event, the

outrage of an enemy attack, and the effort we put into thinking and worrying about it at the time. (There are further factors, which we'll delve into in Chapter 24 when we discuss terrorism.)

The misperception of risk underlies many other public-policy problems. The psychologist Daniel Gilbert, in an article provocatively entitled 'If only gay sex caused global warming', compares our fear of terrorism with our fear of climate change. First, we evolved to be much more wary of hostile intent than of nature; 100,000 years ago, a man with a club (or a hungry lion) was a much worse threat than a thunderstorm. Second, global warming doesn't violate anyone's moral sensibilities; third, it's a long-term threat rather than a clear and present danger; and fourth, we're sensitive to rapid changes in the environment rather than slow ones [570]. There are many more risk biases: we are less afraid when we're in control, such as when driving a car, as opposed to being a passenger in a car or airplane; and we are more afraid of uncertainty, that is, when the magnitude of the risk is unknown (even when it's small) [1246, 1250]. We also indulge in *satisficing* which means we go for an alternative that's 'good enough' rather than going to the trouble of trying to work out the odds perfectly, especially for small transactions. (The misperception here is not that of the risk taker, but of the economists who ignored the fact that real people include transaction costs in their calculations.)

So, starting out from the folk saying that a bird in the hand is worth two in the bush, we can develop quite a lot of machinery to help us understand and model people's attitudes towards risk.

#### 3.2.5.2 Present bias and hyperbolic discounting

Saint Augustine famously prayed 'Lord, make me chaste, but not yet.' We find a similar sentiment with applying security updates, where people may pay more attention to the costs as they're immediate and determinate in time, storage and bandwidth, than the unpredictable future benefits. This *present bias* causes many people to decline updates, which is probably the major source of technical vulnerability online. One way software companies try to fix this is allowing people to delay updates: Windows has 'restart / pick a time / snooze'. Reminders cut the ignore rate from about 90% to about 34%, and may ultimately double overall compliance [533]. A better design is to make updates so painless that they can be made mandatory, or nearly so; this is the approach now followed by some web browsers.

*Hyperbolic discounting* is a model used by decision scientists to quantify present bias. Intuitive reasoning may lead people to use utility functions that discount the future so deeply that immediate gratification seems to be the best course of action, even when it isn't. Such models have been applied to try to explain the *privacy paradox* – why people say in surveys that they care about privacy but act otherwise online. I discuss this in more detail in section8.6.6: other factors, such as uncertainty about the risks and about the efficacy of privacy measures, play a part too. Taken together, the immediate and determinate positive utility of getting free stuff outweighs the random future costs of disclosing too much personal information, or disclosing it to dubious websites.

#### 3.2.5.3 Defaults and nudges

This leads to the importance of defaults. Many people usually take the easiest path and use the standard configuration of a system, as they assume it will be good enough. In 2009, Richard Thaler and Cass Sunnstein wrote a best-seller *'Nudge'* exploring this, pointing out that governments can achieve many policy goals without infringing personal liberty simply by setting the right defaults [1379]. For example, if a firm's staff are enrolled in a pension plan by default, most will not bother to opt out, while if it's optional most will not bother to opt in. A second example is that many more organs are made available for transplant in Spain, where the law lets a dead person's organs be used unless they objected, than in Britain where donors have to consent actively. A third example is that tax evasion can be cut by having the taxpayer declare that the information in the form is true when they start to fill it out, rather than at the end. The set of choices people have to make, the order in which they make them, and the defaults if they do nothing, are called the *choice architecture*. Sunnstein got a job in the Obama administration implementing some of these ideas while Thaler won the 2017 economics Nobel prize.

Defaults matter in security too, but often they are set by an adversary so as to trip you up. For example, Facebook defaults to fairly open information sharing, and whenever enough people have figured out how to increase their privacy settings, the architecture is changed so you have to opt out all over again. This exploits not just hazardous defaults but also the *control paradox* – providing the illusion of control causes people to share more information. We like to feel in control; we feel more comfortable driving in our cars than letting someone else fly us in an airplane – even if the latter is an order of magnitude safer. "Privacy control settings give people more rope to hang themselves," as behavioral economist George Loewenstein puts it. "Facebook has figured this out, so they give you incredibly granular controls." [1140]

#### 3.2.5.4 The default to intentionality

Behavioral economics follows a long tradition in psychology of seeing the mind as composed of interacting rational and emotional components – 'heart' and 'head', or 'affective' and 'cognitive' systems. Studies of developmental biology have shown that, from an early age, we have different mental processing systems for social phenomena (such as recognising parents and siblings) and physical phenomena. Paul Bloom argues that the tension between them explains why many people believe that mind and body are basically different [216]. Children try to explain what they see using physics, but when their understanding falls short, they explain phenomena in terms of intentional action. This has survival value to the young, as it disposes them to get advice from parents or other adults about novel natural phenomena. Bloom suggests that it has an interesting side-effect: it predisposes humans to believe that body and soul are different, and thus lays the ground for religious belief. This argument may not overwhelm the faithful (who will retort that Bloom simply stumbled across a mechanism created by the Intelligent Designer to cause us to have faith in Him). But it may have relevance for the security engineer.

First, it goes some way to explaining the *fundamental attribution error* – people often err by trying to explain things from intentionality rather than from context. Second, attempts to curb phishing by teaching users about the gory design details of the Internet – for example, by telling them to parse URLs in emails that seem to come from a bank – will be of limited value once they get bewildered. If the emotional is programmed to take over whenever the rational runs out, then engaging in a war of technical instruction and counter-instruction with the phishermen is unsound, as they'll be better at it. Safe defaults would be better.

#### 3.2.5.5 The affect heuristic

Nudging people to think in terms of intent rather than of mechanism can exploit the *affect heuristic*, explored by Paul Slovic and colleagues [1313]. The idea is that while the human brain can handle multiple threads of cognitive processing, our emotions remain resolutely single-threaded, and they are even less good at probability theory than the rational part of our brains. So by making emotion salient, a marketer or a fraudster can try to get you to answer questions using emotion rather than reason, and using heuristics rather than calculation. A common trick is to ask an emotional question (whether 'How many dates did you have last month?' or even 'What do you think of President Trump?') to make people insensitive to probability.

So it should not surprise anyone that porn websites have been used to install a lot of malware – as have church websites, which are often poorly maintained and easy to hack. Similarly, events that evoke a feeling of dread – from cancer to terrorism – not only scare people more than the naked probabilities justify, but also make those probabilities harder to calculate, and deter people from even making the effort.

Other factors that can reinforce our tendency to explain things by intent include cognitive overload, where the rational part of the brain simply gets tired. Our capacity for self-control is also liable to fatigue, both physical and mental; some mental arithmetic will increase the probability that we'll pick up a chocolate rather than an apple. So a bank that builds a busy website may be able to sell more life insurance, but it's also likely to make its customers more vulnerable to phishing.

#### 3.2.5.6 Cognitive dissonance

Another interesting offshoot of social psychology is cognitive dissonance theory. People are uncomfortable when they hold conflicting views; they seek out information that confirms their existing views of the world and of themselves, and try to reject information that conflicts with their views or might undermine their self-esteem. One practical consequence is that people are remarkably able to persist in wrong courses of action in the face of mounting evidence that things have gone wrong [1370]. Admitting to yourself or to others that you were duped can be painful; hustlers know this and exploit it. A security professional should 'feel the hustle' – that is, be alert for a situation in which recently established social cues and expectations place you under pressure to 'just do' something

about which you'd normally have reservations. That's the time to step back and ask yourself whether you're being had. But training people to perceive this is hard enough, and getting the average person to break the social flow and say 'stop!' is hard. There have been some experiments, for example with training health-service staff to not give out health information on the phone, and training people on women's self-defence classes to resist demands for extra personal information. The problem with mainstreaming such training is that the budget that might be available for it is orders of magnitude less than the budgets available to firms whose business model is to hustle their customers.

## 3.3 Deception in Practice

This takes us from the theory to the practice. Deception often involves an abuse of the techniques developed by *compliance professionals* – those people whose job it is to get other people to do things. While a sales executive might dazzle you with an offer of a finance plan for a holiday apartment, a police officer might nudge you by their presence to drive more carefully, a park ranger might tell you to extinguish campfires carefully and not feed the bears, and a corporate lawyer might threaten you into taking down something from your website.

The behavioural economics pioneer and apostle of 'nudge', Dick Thaler, refers to the selfish use of behavioural economics as 'sludge' [1378]. But it's odd that economists ever thought that the altruistic use of such techniques would ever be more common than the selfish ones. Not only do marketers push the most profitable option rather than the best value, but they use every other available trick too. Stanford's Persuasive Technology Lab has been at the forefront of developing techniques to keep people addicted to their screens, and one of their alumni, ex-Googler Tristan Harris, has become a vocal critic. Sometimes dubbed 'Silicon valley's conscience', he explains how tech earns its money by manipulating not just defaults but choices, and asks how this can be done ethically [641]. Phones and other screens present menus and thus control choices, but there's more to it than that. Two techniques that screens have made mainstream are the casino's technique of using intermittent variable rewards to create addiction (we check our phones 150 times a day to see if someone has rewarded us with attention) and bottomless message feeds (to keep us consuming even when we aren't hungry any more). But there are many older techniques that pre-date computers.

### 3.3.1 The sabre-toothed salesman

Deception is the twin brother of marketing, so one starting point is the huge literature about sales techniques. One writer who has influenced the security psychology community is Robert Cialdini, a psychology professor who took summer jobs selling everything from used cars to home improvements and life insurance in order to document the tricks of the trade. His book *'Influence, science and Practice'* is widely read by sales professionals and describes six main classes of technique used to influence people and close a sale [329].

These are:

1. Reciprocity: most people feel the need to return favours;
2. Commitment and consistency: people suffer cognitive dissonance if they feel they're being inconsistent;
3. Social proof: most people want the approval of others. This means following others in a group of which they're a member, and the smaller the group the stronger the pressure;
4. Liking: most people want to do what a good-looking or otherwise likeable person asks;
5. Authority: most people are deferential to authority figures (recall the Milgram study mentioned above);
6. Scarcity: we're afraid of missing out, if something we might want could suddenly be unavailable.

All of these are psychological phenomena that are the subject of continuing research. They are also traceable to pressures in our ancestral evolutionary environment, where food scarcity was a real threat, strangers could be dangerous and group solidarity against them (and in the provision of food and shelter) was vital. All are used repeatedly in the advertising and other messages we encounter constantly.

### 3.3.2 Scam psychology

Frank Stajano and Paul Wilson built on this foundation to analyse the principles behind scams. Wilson researched and appeared in nine seasons of TV programs on the most common scams – 'The Real Hustle' – where the scams would be perpetrated on unsuspecting members of the public, who would then be given their money back, debriefed and asked permission for video footage to be used on TV. The know-how from experimenting with several hundred frauds on thousands of marks over several years was distilled into the following seven principles [1343]:

1. Distraction – the fraudster gets the mark to concentrate on the wrong thing. This is at the heart of most magic performances;
2. Social compliance – society trains us not to question people who seem to have authority, leaving people vulnerable to conmen who pretend to be from their bank or from the police;
3. The herd principle – people let their guard down when everyone around them appears to share the same risks. This is a mainstay of the three-card trick, and a growing number of scams on social networks;
4. Dishonesty – if the mark is doing something dodgy, he's less likely to complain. Many are attracted by the idea that 'you're getting a good deal because it's illegal', and whole scam families – such as the resale of fraudulently obtained plane tickets – turn on this;

5. Kindness – this is the flip side of dishonesty, and an adaptation of Cialdini's principle of reciprocity. Many social engineering scams rely on the victims' helpfulness, from tailgating into a building to phoning up with a sob story to ask for a password reset;

6. Need and greed – sales trainers tell us we should find what someone really wants then show him how to get it. A good fraudster can help the mark dream a dream and use this to milk them;

7. Time pressure – this causes people to act viscerally rather than stopping to think. Normal marketers use this all the time ('only 2 seats left at this price'); so do crooks.

The relationship with Cialdini's principles should be obvious. A cynic might say that fraud is just a subdivision of marketing; or perhaps that, as marketing becomes ever more aggressive, it comes to look ever more like fraud. When we investigated online accommodation scams we found it hard to code detectors, since many real estate agents use the same techniques. Oh, and we find the same in software, where there's a blurry dividing line between illegal malware and just-about-legal 'Potentially Unwanted Programs' (PUPs) such as browser plugins that replace your ads with different ones (in fact, the best distinguisher seems to be technical: malware is distributed by many small botnets because of the risk of arrest, while PUPs are mostly distributed by one large network [710]). Fraudsters use regular marketing channels too: Ben Edelman found in 2006 that while 2.73% of companies ranked top in a web search were bad, 4.44% of companies that appeared alongside in the search ads were bad [453]. Bad companies were also more likely to exhibit cheap trust signals, such as TRUSTe privacy certificates on their websites; similarly, bogus landlords often send reference letters or even copies of their ID to prospective tenants, something that genuine landlords never do.

And then there are the deceptive marketing practices of 'legal' businesses. To take just one of many studies, a 2019 crawl of 11K shopping websites by Arunesh Mathur and colleagues found 1,818 instances of 'dark patterns' – manipulative marketing practices such as hidden subscriptions, hidden costs, pressure selling, sneak-into-basket tactics and forced account opening. Of these at least 183 were clearly deceptive [931]. What's more, the bad websites were among the most popular; perhaps a quarter to a third of websites you visit, weighted by traffic, try to hustle you. This constant pressure from scams that lie just short of the threshold for a fraud prosecution has a chilling effect on trust generally. People are less likely to believe security warnings if they are mixed with marketing, or smack of marketing on any way. And we even see some loss of trust in software updates; people say in surveys that they're less likely to apply a security-plus-features upgrade than a security patch, though the field data on upgrades don't (yet) show any difference [1177].

### 3.3.3 Social engineering

Hacking systems through the people who operate them may be growing rapidly but is not new. Military and intelligence organisations have always targeted

each other's staff; most of the intelligence successes of the old Soviet Union were of this kind [94]. Private investigation agencies have not been far behind.

Investigative journalists, private detectives and fraudsters developed the false-pretext phone call into something between an industrial process and an art form in the latter half of the twentieth century. An example of the industrial process was how private detectives tracked people in Britain; given that the country has a National Health Service with which everyone's registered, the trick was to phone up someone with access to the administrative systems in the area you thought the target was, pretend to be someone else in the health service, and ask. Colleagues of mine did an experiment in England in 1996 where we trained the staff at a local health authority to identify and report such calls[1]. We detected about 30 false-pretext calls a week, which would scale to 6000 a week or 300,000 a year for the whole of Britain. That eventually got sort-of fixed but it took over a decade. The real fix wasn't privacy law but that administrators simply stopped answering the phone.

Another old scam from the 20th century is to steal someone's ATM card then phone them up pretending to be from the bank asking whether their card's been stolen. On hearing that it has, the conman says 'We thought so. Please just tell me your PIN now so I can go into the system and cancel your card.' The most rapidly growing recent variety is the 'authorised push payment', where the conman again pretends to be from the bank, and persuades the customer to make a transfer to another account, typically by confusing the customer about the bank's authentication procedures, which most customers find rather mysterious anyway[2].

As for art form, one of the most disturbing security books ever published is Kevin Mitnick's *'Art of Deception'*. Mitnick, who was arrested and convicted for breaking into US phone systems, related after his release from prison how almost all of his exploits had involved social engineering. His typical hack was to pretend to a phone company employee that he was a colleague, and solicit 'help' such as a password. Ways of getting past a company's switchboard and winning its people's trust have been taught for years in sales-training courses, and hackers apply these directly. A harassed system administrator is called once or twice on trivial matters by someone who claims to be a very senior manager's personal assistant; once he has accepted her as an employee, she calls and demands a new password for her boss. Mitnick became an expert at using such tricks to defeat company security procedures, and his book recounts a fascinating range of exploits [987].

Social engineering became world headline news in September 2006 when it emerged that Hewlett-Packard chairwoman Patricia Dunn had hired private investigators who used pretexting to obtain the phone records of other board members of whom she was suspicious, and of journalists she considered hostile. She was forced to resign. The detectives were convicted of fraudulent wire communications and sentenced to do community service [112]. In the same

[1]The story is told in detail in chapter 9 of the second edition of this book, available free online.

[2]Very occasionally, a customer can confuse the bank; a 2019 innovation was the 'callhammer' attack, where someone phones up repeatedly to 'correct' the spelling of 'his name' and changes it one character at a time into another one.

year, the UK privacy authorities prosecuted a private detective agency that did pretexting jobs for top law firms [858].

Amid growing publicity about social engineering, there was an audit of the IRS in 2007 by the Treasury Inspector General for Tax Administration, whose staff called 102 IRS employees at all levels, asked for their user IDs, and told them to change their passwords to a known value; 62 did so. What's worse, this happened despite similar audit tests in 2001 and 2004 [1248]. Since then, a number of audit firms have offered social engineering as a service; they phish their audit clients to show how easy it is. Since the mid-2010s, opinion has shifted against this practice, as it causes distress to staff without changing behaviour much.

Social engineering isn't limited to stealing private information. It can also be about getting people to believe bogus public information. The quote from Bruce Schneier at the head of this chapter appeared in a report of a stock scam, where a bogus press release said that a company's CEO had resigned and its earnings would be restated. Several wire services passed this on, and the stock dropped 61% until the hoax was exposed [1245]. Fake news of this kind has been around forever, but the Internet has made it easier to promote and social media seem to be making it ubiquitous. We'll revisit this issue when I discuss censorship in section 24.4.

### 3.3.4 Phishing

While phone-based social engineering was the favoured tactic of the twentieth century, online phishing seems to have replaced it as the main tactic of the twenty-first. The operators include both spooks and crooks, while the targets are both your staff and your customers. It is difficult enough to train your staff; training the average customer is even harder. They'll assume you're trying to hustle them, ignore your warnings and just figure out the easiest way to get what they want from your system. And you can't design simply for the average. If your systems are not safe to use by people who don't speak English well, or who are dyslexic, or who have learning difficulties, you are asking for serious legal trouble. So the easiest way to use your system had better be the safest.

The word 'phishing' appeared in 1996 in the context of the theft of AOL passwords. By then, attempts to crack email accounts to send spam had become common enough for AOL to have a 'report password solicitation' button on its web page; and the first reference to 'password fishing' is in 1990, in the context of people altering terminal firmware to collect Unix logon passwords [339]. Also in 1996, Tony Greening reported a systematic experimental study: 336 computer science students at the University of Sydney were sent an email message asking them to supply their password on the pretext that it was required to 'validate' the password database after a suspected break-in. 138 of them returned a valid password. Some were suspicious: 30 returned a plausible looking but invalid password, while over 200 changed their passwords without official prompting. But very few of them reported the email to authority [603].

Phishing attacks against banks started seven years later in 2003, with half-a-dozen attempts reported [337].The early attacks imitated bank websites, but

were both crude and greedy; the attackers asked for all sorts of information such as ATM PINs, and their emails were also written in poor English. Most customers smelt a rat. By about 2008, the attackers learned to use better psychology; they often reused genuine bank emails, with just the URLs changed, or sent an email saying something like 'Thank you for adding a new email address to your PayPal account' to provoke the customer to log on to complain that they hadn't. Of course, customers who used the provided link rather than typing in `www.paypal.com` or using an existing bookmark would get their accounts emptied. By now, phishing was being used by state actors too; I described in section 2.2.2 how Chinese intelligence compromised the Dalai Lama's private office during the 2008 Olympic games. They did this using crimeware tools that had been developed for use by fraud gangs.

Fraud losses grew rapidly but stabilised by about 2015. A number of countermeasures helped bring things under control, including more complex logon schemes (using two-factor authentication, or its low-cost cousin, the request for some random letters of your password); a move to webmail systems that filter spam better; and back-end fraud engines that look for cashout patterns. The competitive landscape was rough, in that the phishermen would hit the easiest targets at any time in each country, both in terms of stealing their customer credentials and using their accounts to launder stolen funds. Concentrated losses caused the targets to wake up and take action. Since then, we've seen large-scale attacks on non-financial firms like Amazon; in the late 2000s, the crook would change your email and street address, then use your credit card to order a wide-screen TV. Since about 2016, the action has been in gift vouchers.

As we noted in the last chapter, phishing is also used at scale by botmasters to recruit new machines to their botnets, and in targeted ways both by crooks aiming at specific people or firms, and by intelligence agencies. There's a big difference between attacks conducted at scale, where the economics dictate that the cost of recruiting a new machine to a botnet can be at most a few cents, and targeted attacks, where spooks can spend years trying to hack the phone of a rival head of government, or a smart crook can spend weeks or months of effort stalking a chief financial officer in the hope of a large payout. The lures and techniques used are different, even if the crimeware installed on the target's laptop or phone comes from the same stable. Cormac Herley argues that this gulf between the economics of targeted crime and volume crime is one of the reasons why cybercrime isn't much worse than it is [655]. After all, given that we depend on computers, and that all computers are insecure, and that there are attacks all the time, how come civilisation hasn't collapsed? Cybercrime can't always be as easy as it looks.

Well, it took seven years for the bad guys to catch up with Tony Greening's 1995 phishing work. A 2007 paper by Tom Jagatic and colleagues showed how to make phishing much more effective by automatically personalising each phish using context mined from the target's social network [720]. I cited that in the second edition of this book, and in 2016 we saw it in the wild: a gang sent hundreds of thousands of phish with US and Australian banking Trojans to individuals working in finance departments of companies, with their names and job titles apparently scraped from LinkedIn [969]. This seems to have been crude and hasn't really caught on, but once the bad guys figure it out we may

see spearphishing at scale in the future, and it's interesting to think of how we might respond. The other personalised bulk scams we see are blackmail attempts where the victims get email claiming that their personal information has been compromised and including a password or the last four digits of a credit card number as evidence, but the yield from such scams seems to be low.

### 3.3.5 Opsec

Getting your staff to resist attempts by outsiders to inveigle them into revealing secrets, whether over the phone or online, is known in military circles as *operational security* or Opsec. Protecting really valuable secrets, such as unpublished financial data, not-yet-patented industrial research or military plans, depends on limiting the number of people with access, and also having strict doctrines about with whom they may be discussed and how. It's not enough for rules to exist; you have to train all the staff who have access to the confidential material, explain the reasons behind the rules, and embed them socially in the organisation. In our medical privacy case, we educated health service staff about pretext calls and set up a strict callback policy: they would not discuss medical records on the phone unless they had called a number they had got from the health service internal phone book rather than from a caller. And once the staff have detected and defeated a few false-pretext calls, they talk about it and the message gets embedded in the way everybody works.

Another example comes from a large Silicon Valley service firm, which suffered intrusion attempts when outsiders tailgated staff into buildings on campus. Stopping this with airport-style ID checks, or even card-activated turnstiles, would have changed the ambience and clashed with the culture. The solution was to create and embed a social rule that when someone holds open a building door for you, you show them your badge. The critical factor, as with the bogus phone calls, is social embedding rather than just training.

Often the hardest people to educate are the most senior; a consultancy sent the finance directors of 500 publicly-quoted companies a USB memory stick as part of an anonymous invitation saying 'For Your Chance to Attend the Party of a Lifetime', and 46% of them put it into their computers [774]. In my own experience in banking, the people you couldn't train were those who were paid more than you, such as traders in the dealing rooms.

Some operational security measures are common sense, such as not tossing out sensitive stuff in the trash. Less obvious is the need to train the people you trust, from suppliers to old friends. A leak of embarrassing emails that appeared to come from the office of the UK Prime Minister and was initially blamed on 'hackers' turned out to have been fished out of the trash at his personal pollster's home by a private detective [911].

People operate systems in the way they have to in order to get their work done. Designing systems so that staff can't disclose data they shouldn't, or at least so that disclosures involve talking to other staff members or jumping through other hoops, is a more reliable way to stop them being tricked into disclosing data they shouldn't. Research shows that company staff have only so much *compliance budget*, that is, they're only prepared to put so many hours

a year into tasks that are not obviously helping them achieve their goals [155]. You need to figure out what this budget is, and use it wisely.

But what about a firm's customers? There is a lot of scope for phishermen to simply order bank customers to reveal their security data, and this happens at scale, against both retail and business customers. There are also the many small scams that customers try on when they find vulnerabilities in your business processes. I'll discuss both types of fraud further in the chapter on Banking and Bookkeeping.

### 3.3.6 Deception research

Finally, a word on deception research. Since 9/11, huge amounts of money have been spent by governments trying to find better lie detectors, and deception researchers are funded across about five different subdisciplines of psychology. The polygraph measures stress via heart rate and skin conductance; it has been around since the 1920s and is used by some US states in criminal investigations, as well as by the Federal government in screening people for Top Secret clearances. The evidence on its effectiveness is patchy at best, and surveyed extensively by Aldert Vrij [1450]. While it can be an effective prop in the hands of a skilled interrogator, the key factor is the skill rather than the prop. When used by unskilled people in a lab environment, against experimental subjects telling low-stakes lies, its output is little better than random. As well as measuring stress via skin conductance, you can measure distraction using eye movements and guilt by upper body movements. In a research project with Sophie van der Zee, we used body motion-capture suits and also the gesture-recognition cameras in an Xbox and got slightly better results than a polygraph [1523]. However such technologies can at best augment the interrogator's skill; the government dream of an interrogation robot for automated border crossings is some way off.

A second approach to dealing with deception is to train a machine-learning classifier on real customer behaviour. This is what credit-card fraud engines have been doing since the late 1990s, and recent research has pushed into other fields too. For example, Noam Brown and Tuomas Sandholm have created a poker-playing bot called Pluribus that beat a dozen expert players over a 12-day marathon of 10,000 hands of Texas Hold 'em. It doesn't use psychology but game theory, playing against itself millions of times and tracking regret at bids that could have given better outcomes. That it can consistently beat experts without access to 'tells' such as its opponents' facial gestures or body language is itself telling. Dealing with deception using statistical machine learning rather than physiological monitoring also intrudes less into privacy.

## 3.4 Passwords

The management of passwords gives an important and instructive context in which usability, applied psychology and security meet. Passwords have been one of the biggest practical problems facing security engineers since perhaps the 1970s. In fact, as the usability researcher Angela Sasse puts it, it's hard to think of a worse authentication mechanism than passwords, given what we know

about human memory: people can't remember infrequently-used, frequently-changed, or many similar items; we can't forget on demand; recall is harder than recognition; and non-meaningful words are more difficult.

To place the problem in context, the password system used by a large modern service firm has a number of components:

1. The visible part is the logon page, which asks you to choose a password when you register, probably checks its strength in some way, and then asks for the password when you log on;

2. There will be recovery mechanisms that enable you to deal with a forgotten password or even a compromised account, typically by asking further security questions, or via your primary email account, or by sending an SMS to your phone;

3. Behind this lie technical protocol mechanisms for password checking, typically routines that encrypt your password when you enter it at your laptop or phone, and then either compare it with a local encrypted value, or take it to a remote server for checking;

4. There are often protocol mechanisms to synchronise passwords across multiple platforms, so that if you change your password on your laptop, your phone won't let you use that service until you enter the new one there too. And these mechanisms may enable you to blacklist a stolen phone without having to reset the passwords for all the services it was able to access;

5. There will be intrusion-detection mechanisms to propagate an alarm if one of your passwords is used somewhere it probably shouldn't be;

6. There are single-signon mechanisms to use one logon on many websites, as when you use your Google or Facebook account to log on to a newspaper.

Let's work up from the bottom. Developing a full-feature password management system can be a lot of work, and providing support for password recovery also costs money (a few years ago, the UK phone company BT had two hundred people in its password-reset centre). So outsourcing 'identity management' can make business sense. In addition, intrusion detection works best at scale; if someone uses my gmail password in an Internet cafe in Peru while Google knows I'm in Scotland, they send an SMS to my phone to check; a small website can't do that. The main cause of attempted password abuse is when one firm gets hacked, disclosing millions of email addresses and passwords, which the bad guys try out elsewhere; big firms spot this quickly while small ones don't. The big firms also help their customers maintain situational awareness, by alerting you to logons from new devices or from strange places. Again, it's hard to do that if you're a small website or one that people visit infrequently.

As for syncing passwords between devices, only the device vendors can really do that well; and the protocol mechanisms for encrypting passwords in transit to a server that verifies them will be discussed in the next chapter. That brings us to password recovery.

### 3.4.1 Password recovery

The experience of the 2010s, as the large service firms scaled up and people moved en masse to smartphones, is that password recovery is often the hardest aspect of authentication. If people you know, such as your staff, forget their passwords, you can get them to interact with an administrator or manager who knows them. But for people you don't know such as your online customers it's harder. And as a large service firm will be recovering tens of thousands of accounts every day, you need some way of doing it without human intervention in the vast majority of cases.

During the 1990s and 2000s, many websites did password recovery using 'security questions' such as asking for your favourite team, the name of your pet or even that old chestnut, your mother's maiden name. Such near-public information is often easy to guess so it gave an easier way to break into accounts than guessing the password itself. This was made even worse by everyone asking the same questions. In the case of celebrities – or abuse by a former intimate partner – there may be no usable secrets. This was brought home to the public in 2008, when a student hacked the Yahoo email account of US Vice-Presidential candidate Sarah Palin via the password recovery questions – her date of birth and the name of her first school. Both of these were public information. Since then, crooks have learned to use security questions to loot accounts when they can; at the US Social Security Administration, a common fraud was to open an online account for a pensioner who's dealt with their pension by snail mail in the past, and redirect the payments to a different bank account. This peaked in 2013; the countermeasure that fixed it was to always notify beneficiaries of account changes by snail mail.

In 2015, five Google engineers published a thorough analysis of security questions, and many turned out to be extremely weak. For example, an attacker could get a 19.7% success rate against 'Favourite food?' in English. Some 37% of people provided wrong answers, in some cases to make them stronger, but sometimes not; so the attacker could guess the answer to 'Frequent flyer number?' 4.2% of the time. Fully 16% of people's answers were public. In addition to being insecure, the 'security questions' turned out to be hard to use: 40% of English-speaking US users were unable to recall the answers when needed, while twice as many could recover accounts using an SMS reset code [234].

Given these problems with security and memorability, most websites now let you recover your password by an email to the address with which you first registered. But if someone compromises that email account, they can get all your dependent accounts too. While this may be adequate for websites where a compromise is of little consequence, for important accounts – such as banking and email – standard practice is now to use a second factor, typically a code sent to your phone by SMS, or better still using an app that can encrypt the code and tie it to a specific handset. Service providers that allow email recovery nudge people towards using such a code instead where possible. Google research shows that SMSs stop all bulk password guessing by bots, 96% of bulk phishing and 76% of targeted attacks [433]. However this depends on phone companies taking care over who can get a replacement SIM card, and many don't. The problem in 2019 is rapid growth in attacks based on intercepting SMS authentication codes,

which could use Android malware or hacks to the SS7 mobile-phone signaling system but mostly seen to involve SIM swap, where the attacker pretends to be you to your mobile phone company and gets a replacement SIM card for your account. SIM-swap attacks have become big time where phone companies make it easy, including Nigeria (where it's the main form of bank fraud) and California (where it's used to loot people's accounts at bitcoin exchanges) [819]. I'll discuss this in more detail in the chapter on phones. The next step in the arms race looks like being moving customers from SMS messages for authentication and account recovery to an app; the same Google research shows that this improves these last two figures to 99% for bulk phishing and 90% for targeted attacks [433].

Email notification is the default for telling people not just of suspicious login attempts, but of logins to new devices that succeeded with the help of a code. That way, if someone plants malware on your phone, you have some chance of detecting it. If all else fails, a service provider may have to let the customer get through to a person. But when designing such a system, never forget that it's only as strong as the weakest fallback mechanism – be it a recovery email loop with an email provider you don't control, a phone code that's vulnerable to SIM swapping or mobile malware, or a human who's open to social engineering.

### 3.4.2 Password choice

Many accounts are compromised by guessing PINs or passwords. There are botnets constantly breaking into online accounts by guessing passwords and password-recovery questions, as I described in 2.3.1.4, in order to use email accounts to send spam and to recruit machines to botnets. And as people invent new services and put passwords on them, the password guessers find new targets. A recent example is cryptocurrency wallets: an anonymous 'bitcoin bandit' managed to steal $50m by trying lots of weak passwords for ethereum wallets [600]. Meanwhile, billions of dollars' worth of cryptocurrency has been lost because passwords were forgotten. So passwords matter, and there are basically three broad concerns, in ascending order of importance and difficulty:

1. Will the user enter the password correctly with a high enough probability?
2. Will the user remember the password, or will they have to either write it down or choose one that's easy for the attacker to guess?
3. Will the user break the system security by disclosing the password to a third party, whether accidentally, on purpose, or as a result of deception?

### 3.4.3 Difficulties with reliable password entry

The first human-factors issue is that if a password is too long or complex, users might have difficulty entering it correctly. If the operation they're trying to perform is urgent, this might have safety implications. If customers have difficulty entering software product activation codes, this can generate expensive calls to your support desk. And the move from laptops to smartphones during the 2010s has made password rules such as 'at least one lower-case letter, upper-case letter, number and special character' really fiddly and annoying. This is one of the

factors pushing people toward longer but simpler secrets, such as passphrases of three or four words. But will people be able to enter them without making too many errors?

An interesting study was done for the STS prepayment meters used to sell electricity and other utilities to about 400 million households in over 100 mostly less developed countries. The customer hands some money to a sales agent, and gets a 20-digit number printed out on a receipt. She takes this receipt home, enters the numbers at a keypad in her meter, and the lights come on. The STS designers worried that since a lot of the population was illiterate, and since people might get lost halfway through entering the number, the system might be unusable. But illiteracy was not a problem: even people who could not read had no difficulty with numbers ('everybody can use a phone', as one of the engineers said). The biggest problem was entry errors, and these were dealt with by printing the twenty digits in two rows, with three and two groups of four digits respectively [74]. I'll describe this in detail in section 12.2.

A quite different application is the firing codes for US nuclear weapons. These consist of only 12 decimal digits. If they are ever used, the operators will be under extreme stress, and possibly using improvised or obsolete communications channels. Experiments suggested that 12 digits was the maximum that could be conveyed reliably in such circumstances. I'll discuss how this evolved in 13.2.

### 3.4.4 Difficulties with remembering the password

Our second psychological issue is that people often find passwords hard to remember [1532]. Twelve to twenty digits may be easy to copy from a telegram or a meter ticket, but when customers are expected to memorize passwords, they either choose values that are easy for attackers to guess, or write them down, or both. In fact, standard password advice has been summed up as: "Choose a password you can't remember, and don't write it down."

The problems are not limited to computer access. For example, one chain of cheap hotels in France introduced self service. You'd turn up at the hotel, swipe your credit card in the reception machine, and get a receipt with a numerical access code to unlock your room door. To keep costs down, the rooms did not have en-suite bathrooms. A common failure mode was that you'd get up in the middle of the night to go to the bathroom, forget your access code, and realise you hadn't taken the receipt with you. So you'd have to sleep on the bathroom floor until the staff arrived the following morning.

Password memorability can be discussed under five main headings: naïve choice, user abilities and training, design errors, operational failures and vulnerability to social-engineering attacks.

#### 3.4.4.1 Naïve choice

Since the mid-1980s, people have studied what sort of passwords people choose, and the results are depressing. People use spouses' names, single letters, or even just hit carriage return giving an empty string as their password. Fred

Grampp and Robert Morris's classic paper on Unix security [596] reports that after software became available which forced passwords to be at least six characters long and have at least one nonletter, they made a file of the 20 most common female names, each followed by a single digit. Of these 200 passwords, at least one was in use on each of several dozen machines they examined. At the time, Unix systems kept encrypted passwords in a file `/etc/passwd` that all system users could read, so any user could verify a guess of any other user's password. Other studies showed that requiring a non-letter simply changed the most popular password from 'password' to 'password1' [1247].

In 1990, Daniel Klein gathered 25,000 Unix passwords and found that 21–25% of passwords could be guessed depending on the amount of effort put in [791]. Dictionary words accounted for 7.4%, common names for 4%, combinations of user and account name 2.7%, and so on down a list of less probable choices such as words from science fiction (0.4%) and sports terms (0.2%). Other password guesses used patterns, such as by taking an account *'klone'* belonging to the user 'Daniel V. Klein' and trying passwords such as klone, klone1, klone123, dvk, dvkdvk, leinad, neilk, DvkkvD, and so on. The following year, Alec Muffett released 'crack', software that would try to brute-force Unix passwords using dictionaries and patterns derived from them by a set of mangling rules.

The largest academic study of password choice of which I am aware is by Joe Bonneau, who in 2012 analysed tens of millions of passwords in leaked password files, and also interned at Yahoo where he instrumented the login system to collect live statistics on the choices of 70 million users. He also worked out the best metrics to use for password guessability, both in standalone systems and where attackers use passwords harvested from one system to crack accounts on another [233]. This work informed the design of password strength checkers and other current practices at the big service firms.

#### 3.4.4.2 User abilities and training

Sometimes you can train the users. Password checkers have trained them to use longer passwords with numbers as well as letters, and the effect spills over to websites that don't use them [340]. But you do not want to drive customers away, so the marketing folks will limit what you can do. In fact, research shows that password rule enforcement is not a function of the value at risk, but of whether the website is a monopoly. Such websites typically have very annoying rules, while websites with competitors, such as Amazon, are more usable, placing more reliance on back-end intrusion-detection systems.

In a corporate or military environment you can enforce password choice rules, or password change rules, or issue people with random passwords. But then people will have to write them down. So you can insist that passwords are treated the same way as the data they protect: bank master passwords go in the vault overnight, while military 'Top Secret' passwords must be sealed in an envelope, in a safe, in a room that's locked when not occupied, in a building patrolled by guards. You can send guards round at night to clean all desks and bin everything that hasn't been locked up. But if you want to hire and retain good people, you'd better think things through a bit more carefully. For

example, one Silicon Valley firm had a policy that the root password for each machine would be written down on a card and put in an envelope taped to the side of the machine – a more human version of the rule that passwords be treated the same way as the data they protect.

While writing the first edition of this book, I could not find any account of experiments on training people in password choice that would hold water by the standards of applied psychology (i.e., randomized controlled trials with adequate statistical power). The closest I found was a study of the recall rates, forgetting rates, and guessing rates of various types of password [280]; this didn't tell us the actual effects of giving users various kinds of advice. We therefore decided to see what could be achieved by training, and selected three groups of about a hundred volunteers from our first-year science students [1515]:

- the red (control) group was given the usual advice (password at least six characters long, including one nonletter)
- the green group was told to think of a passphrase and select letters from it to build a password. So 'It's 12 noon and I am hungry' would give `'I'S12&IAH'`
- the yellow group was told to select eight characters (alpha or numeric) at random from a table we gave them, write them down, and destroy this note after a week or two once they'd memorized the password.

What we expected to find was that the red group's passwords would be easier to guess than the green group's which would in turn be easier than the yellow group's; and that the yellow group would have the most difficulty remembering their passwords (or would be forced to reset them more often), followed by green and then red. But that's not what we found.

About 30% of the control group chose passwords that could be guessed using Alec Muffett's 'crack' software, versus about 10 percent for the other two groups. So passphrases and random passwords seemed to be about equally effective. When we looked at password reset rates, there was no significant difference between the three groups. When we asked the students whether they'd found their passwords hard to remember (or had written them down), the yellow group had significantly more problems than the other two; but there was no significant difference between red and green.

The conclusions we drew were as follows.

- For users who follow instructions, passwords based on mnemonic phrases offer the best of both worlds. They are as easy to remember as naively selected passwords, and as hard to guess as random passwords.
- The problem then becomes one of *user compliance*. A significant number of users (perhaps a third of them) just don't do what they're told.

So when the army gives soldiers randomly-selected passwords, its value comes from the fact that the password assignment compels user compliance, rather

than from the fact that they're random (as mnemonic phrases would do just as well).

But centrally-assigned passwords are often inappropriate. When you are offering a service to the public, your customers expect you to present broadly the same interfaces as your competitors. So you must let users choose their own website passwords, subject to some lightweight algorithm to reject passwords that are 'clearly bad'. (GCHQ suggests using a 'bad password list' of the 100,000 passwords most commonly found in online password dumps.) In the case of bank cards, users expect a bank-issued initial PIN plus the ability to change the PIN afterwards to one of their choosing (though again you may block a 'clearly bad' PIN such as 0000 or 1234). Over half of cardholders keep a random PIN, but about a quarter choose PINs such as children's birth dates which have less entropy than random PINs would, and have the same PIN on different cards. The upshot is that a thief who steals a purse or wallet may have a chance of about one in eleven to get lucky, if he tries the most common PINs on all the cards first in offline mode and then in online mode, so he gets six goes at each [237].

#### 3.4.4.3 Design errors

Attempts to make passwords memorable are a frequent source of severe design errors. The classic example of how not to do it is to ask for 'your mother's maiden name'. A surprising number of banks, government departments and other organisations still authenticate their customers in this way, though nowadays it tends to be not a password but a password recovery question. You could always try to tell 'Yngstrom' to your bank, 'Jones' to the phone company, 'Geraghty' to the travel agent, and so on; but data are shared extensively between companies, so you could easily end up confusing their systems – not to mention yourself. And if you try to phone up your bank and tell them that you've decided to change your mother's maiden name from Yngstrom to `yGt5r4ad` – or even Smith – then good luck.

Some organisations use contextual security information. My bank asks its business customers the value of the last check from their account that was cleared. In theory, this could be helpful: if someone overhears me doing a transaction on the telephone, then it's not a long-term compromise. The details bear some attention though. When this system was first introduced, I wondered whether a supplier, to whom I'd just written a check, might impersonate me, and concluded that asking for the last three checks' values would be safer. But the problem I actually had was unexpected. Having given the checkbook to our accountant for the annual audit, I couldn't talk to the bank. I also don't like the idea that someone who steals my physical post can also steal my money.

The sheer number of applications demanding a password nowadays exceeds the powers of human memory. A 2007 study by Dinei Florêncio and Cormac Herley of half a million web users over three months showed that the average user has 6.5 passwords, each shared across 3.9 different sites; she has about 25 accounts that require passwords, and types an average of 8 passwords per day. Bonneau published more extensive statistics in 2012 [233] but since then the frequency of user password entry has fallen, thanks to smartphones. Modern

web browsers also cache passwords; see the discussion of password managers at section 3.4.11 below. But many people use the same password for many different purposes and don't work out special processes to deal with their high-value logons such as to their bank, their social media accounts and their email. So you have to expect that the password chosen by the customer of the electronic banking system you've just designed, may be known to a Mafia-operated porn site as well. (There's even a website, `http://haveibeenpwned.com`, that will tell you which security breaches have leaked your email address and password.)

One of the most pervasive and persistent design errors has been forcing users to change passwords regularly. Indeed, this has almost become a religious war between security researchers and auditors. When I first came across enforced monthly password changes in the 1980s, I observed that it led people to choose passwords such as 'julia03' for March, 'julia04' for April, and so on, and said as much in the first (2001) edition of this book (chapter 3, page 48). However, in 2003, Bill Burr of NIST wrote password guidelines recommending regular update [822]. This was adopted by the Big Four auditors, who pushed it out to all their audit clients[3]. Meanwhile, security usability researchers conducted survey after survey showing that monthly change was suboptimal; and finally a survey was written by usability expert Lorrie Cranor while she was Chief Technologist at the FTC [374]. The following year, NIST recanted; they now recommend long passphrases that are only changed on compromise[4]. Other government agencies such as GCHQ have said likewise, and Microsoft finally announced the end of password-expiration policies in Windows 10 from April 2019. The big four auditors, who dictate compliance to many companies, will catch up eventually.

The current fashion, in 2019, is to invite users to select passphrases of three or more random dictionary words. This was promoted by a famous xkcd cartoon which suggested 'correct battery horse staple' as a password. Empirical research, however, shows that real users select multi-word passphrases with much less entropy than they'd get if they really did select at random from a dictionary; they tend to go for common noun bigrams, and moving to three or four words brings rapidly diminishing returns [238].

#### 3.4.4.4 Operational issues

The most pervasive operational error is failing to reset default passwords. This has been a chronic problem since the early dial access systems in the 1980s attracted attention from mischievous schoolkids. A particularly bad example is where systems have default passwords that can't be changed, checked by software that can't be patched. We see ever more such devices in the Internet of Things; they remain vulnerable for their operational lives. The Mirai botnets have emerged to recruit and exploit them, as I described in Chapter 2 above.

[3]Our university's auditors wrote in their annual report for three years in a row that we should have monthly enforced password change, but couldn't provide any evidence to support this and weren't even aware that their policy came ultimately from NIST. Unimpressed, we asked the chair of our Audit Committee to appoint a new lot of auditors, and eventually that happened.

[4]NIST SP 800-63-3

Passwords in plain sight are another long-running problem, whether on sticky notes or some electronic equivalent. A famous early case was R v Gold and Schifreen, where two young hackers saw a phone number for the development version of an early public email service run by British Telecom in a note stuck on a terminal at an exhibition. They dialed in later, and found the welcome screen had a maintenance password displayed on it. They tried this on the live system too, and it worked! They proceeded to hack into the Duke of Edinburgh's electronic mail account, and sent mail 'from' him to someone they didn't like, announcing the award of a knighthood. This heinous crime so shocked the establishment that when prosecutors failed to persuade the courts to convict the defendants under the laws then in force, parliament passed Britain's first Computer Misuse Act.

A third operational issue is asking for passwords when they're not really needed. Many websites nudge you or even compel you to set up an account as an excuse to get your email address or to give you the feeling of belonging to a 'club' [236]. So it may be perfectly rational for users who never plan to visit that site again to express their exasperation by entering '123456' or even 'password' in the password field.

A fourth is atrocious password management systems: some don't encrypt passwords at all, and there are reports from time to time of enterprising hackers smuggling back doors into password management libraries [331].

But perhaps the biggest operational issue is vulnerability to social-engineering attacks.

#### 3.4.4.5 Social-engineering attacks

Careful organisations communicate security context in various ways to help staff avoid making mistakes. The NSA, for example, had different colored internal and external telephones, and when an external phone in a room is off-hook, classified material can't even be discussed in the room – let alone on the phone.

Yet while many banks and other businesses maintain some internal security context, they often train their customers to act in unsafe ways. Because of pervasive phishing, it's not prudent to try to log on to your bank by clicking on a link in an email, so you should always use a browser bookmark or type in the URL by hand. Yet bank marketing departments send out lots of emails containing clickable links. Indeed much of the marketing industry is devoted to getting people to click on links. Many email clients – including Apple's, Microsoft's, and Google's – make plaintext URLs clickable, so their users may never see a URL that isn't. Bank customers are well trained to do the wrong thing.

A prudent customer should also be cautious if a web service directs him somewhere else – yet bank systems use all sorts of strange URLs for their services. A spam from the Bank of America directed UK customers to `mynewcard.com` and got the certificate wrong (it was for `mynewcard.bankofamerica.com`). There are many more examples of major banks training their customers to practice unsafe computing – by disregarding domain names, ignoring certificate warnings, and merrily clicking links [438]. As a result, even security experts have

difficulty telling bank spam from phish [339].

It's not prudent to give out security information over the phone to unidentified callers – yet we all get phoned by bank staff who demand security information. Banks also call us on our mobiles now and expect us to give out security information to a whole train carriage of strangers, rather than letting us text a response. It's also not prudent to put a bank card PIN into any device other than an ATM or a PIN entry device (PED) in a store; and Citibank even asks customers to disregard and report emails that ask for personal information, including PIN and account details. So what happened? You guessed it – it sent its Australian customers an email asking customers 'as part of a security upgrade' to log on to its website and authenticate themselves using a card number and an ATM PIN [814]. And in one 2005 case, the Halifax sent a spam to the mother of a student of ours who contacted the bank's security department, which told her it was a phish. The student then contacted the ISP to report abuse, and found that the URL and the service were genuine [930]. The Halifax disappeared during the crash of 2008, and given that their own security department couldn't tell spam from phish, perhaps that was justice (though it cost UK taxpayers a shedload of money).

#### 3.4.4.6 Customer education

After phishing became a real threat to online banking in the mid-2000s, banks tried to train their customers to look for certain features in websites. This has been partly a bona fide attempt at risk reduction, but partly risk dumping – seeing to it that customers who don't understand or can't follow instructions can be held responsible for the resulting loss. The general pattern has been that as soon as customers are trained to follow some particular rule, the phishermen exploit this, as the reasons for the rule are not adequately explained.

At the beginning, the advice was 'Check the English', so the bad guys either got someone who could write English, or simply started using the banks' own emails but with the URLs changed. Then it was 'Look for the lock symbol', so the phishing sites started to use SSL (or just forging it by putting graphics of lock symbols on their web pages). Some banks started putting the last four digits of the customer account number into emails; the phishermen responded by putting in the first four (which are constant for a given bank and card product). Next the advice was that it was OK to click on images, but not on URLs; the phishermen promptly put in links that appeared to be images but actually pointed at executables. The advice then was to check where a link would really go by hovering your mouse over it; the bad guys then either inserted a non-printing character into the URL to stop Internet Explorer from displaying the rest, or used an unmanageably long URL (as many banks also did).

This sort of arms race is most likely to benefit the attackers. The countermeasures become so complex and counterintuitive that they confuse more and more users – exactly what the phishermen need. The safety and usability communities have known for years that 'blame and train' is not the way to deal with unusable systems – the only real fix is to design for safe usability in the first place [1072].

#### 3.4.4.7 Phishing warnings

Part of the solution is to give users better tools, and modern browsers alert you to wicked URLs. There's a range of mechanisms under the hood. First, there are lists of bad URLs collated by the anti-virus and threat intelligence community. Second, there's logic to look for expired certificates and other compliance failures (as the majority of those alerts are false alarms).

There has been a lot of research, in both industry and academia, about how you get people to pay attention to warnings. We see so many of them, most are irrelevant, and many are designed to shift risk to us from someone else. So when do people pay attention? In our own work, we tried a number of things and found that people paid most attention when the warnings were not vague and general (*'Warning - visiting this web site may harm your computer!'*) but specific and concrete (*'The site you are about to visit has been confirmed to contain software that poses a significant risk to you, with no tangible benefit. It would try to infect your computer with malware designed to steal your bank account and credit card details in order to defraud you*) [989]. Subsequent research by Adrienne Porter Felt and Google's usability team has tried many ideas including making warnings psychologically salient using faces (which doesn't work), simplifying the text (which helps) and making the safe defaults both attractive and prominent (which also helps). Optimising these factors improves compliance from about 35% to about 50% [500]. However, if you want to stop the great majority of people from clicking on known-bad URLs, then voluntary compliance isn't enough. You either have to block them at your firewall, or block them at the browser (as both Chrome and Firefox do for different types of certificate error – a matter to which we'll return in 21.4.6).

### 3.4.5 System Issues

Not all phishing attacks involve psychology. Some involve technical mechanisms to do with password entry and storage together with some broader system issues.

As we already noted, a key question is whether we can restrict the number of password guesses. Security engineers sometimes refer to password systems as 'online' if guessing is limited (as with ATM PINs) and 'offline' if it is not (this originally meant systems where a user could fetch the password file and take it away to try to guess the passwords of other users, including more privileged users). But the terms are no longer really accurate. Some offline systems can restrict guesses, such as payment cards which use physical tamper-resistance to limit you to three PIN guesses, while some online systems cannot. For example, if you log on using Kerberos, an opponent who taps the line can observe your key encrypted with your password flowing from the server to your client, and then data encrypted with that key flowing on the line; so she can take her time to try out all possible passwords. The most common trap here is the system that normally restricts password guesses but then suddenly fails to do so, when it gets hacked and a one-way encrypted password file is leaked, together with the encryption keys. Then the bad guys can try out their entire password dictionary against each account at their leisure.

Password guessability ultimately depends on the entropy of the chosen pass-

words and the number of allowed guesses, but this plays out in the context of a specific threat model, so you need to consider the type of attacks you are trying to defend against. Broadly speaking, these are:

**Targeted attack on one account:** an intruder tries to guess a specific user's password. He might try to guess a rival's logon password at the office, in order to do mischief directly.

**Attempt to penetrate any account on any system belonging to a specific target:** an enemy tries to hack any account you own, anywhere, to get information that might might help take over other accounts, or do harm directly

**Attempt to penetrate any account on a target system:** the intruder tries to get a logon as any user of the system. This is the classic case of the phisherman trying to hack any account at a target bank so he can launder stolen money through it.

**Attempt to penetrate any account on any system:** the intruder merely wants an account at any system in a given domain but doesn't care which one. Examples are bad guys trying to guess passwords on any online email service so they can send spam from the compromised account, and a targeted attacker who wants a logon to any random machine in the domain of a target company as a beachhead.

**Attempt to use a breach of one system to penetrate a related one:** the intruder has got a beachhead and now wants to move inland to capture higher-value targets.

**Service denial attack:** the attacker may wish to block one or more legitimate users from using the system. This might be targeted on a particular account or system-wide.

This taxonomy helps us ask relevant questions when evaluating a password system.

### 3.4.6 Can you deny service?

There are basically three ways to deal with password guessing when you detect it: lockout, throttling, and protective monitoring. Banks may freeze your card after three wrong PINs; but if they freeze your online account after three bad password attempts they open themselves up to a denial-of-service attack. Service can also fail by accident; poorly-configured systems can generate repeat fails with stale credentials. So many commercial websites nowadays use throttling rather than lockout. In in a military system, you might not want even that, in case an enemy who gets access to the network could jam it with a flood of false logon attempts. In this case, protective monitoring might be the preferred option, with a plan to abandon rate-limiting if need be in a crisis. Joe Bonneau and Soren Preibusch collected statistics of how many major websites use account locking versus various types of rate control [236]. They found that popular, growing, competent sites tend to be more secure, as do payment sites, while content sites

do worst. Microsoft Research's Yuan Tian, Cormac Herley and Stuart Schechter investigated how to locking or throttling properly; among other things, it's best to penalise guesses of weak passwords (as otherwise an attacker gets advantage by guessing them first), to be more aggressive when protecting users who have selected weak passwords, and to not punish IPs or clients that repeatedly submit the same wrong password [1386].

### 3.4.7 Protecting oneself or others?

Next, to what extent does the system need to protect users and subsystems from each other? In global systems on which anyone can get an account – such as mobile phone systems and cash machine systems – you must assume that the attackers are already legitimate users, and see to it that no-one can use the service at someone else's expense. So knowledge of one user's password will not allow another user's account to be compromised. This has both personal aspects, and system aspects.

On the personal side, don't forget what we said about intimate partner abuse in 2.5.4: the passwords people choose are often easy for their spouses or partners to guess, and the same goes for password recovery questions: so some thought needs to be given to how abuse victims can recover their security.

On the system side, there are all sorts of passwords used for mutual authentication between subsystems, few mechanisms to enforce password quality in server-server environments, and many well-known issues (for example, the default password for the Java trusted keystore file is 'changeit'). Development teams often share passwords that end up in live systems, even 30 years after this practice led to the well-publicised hack of the Duke of Edinburgh's email described in section 3.4.4.4. Within a single big service firm you can lock stuff down by having named crypto keys and seeing to it that each name generates a call to an underlying hardware security module; or you can even use mechanisms like SGX to tie keys to known software. But that costs real money, and money isn't the only problem. Enterprise system components are often hosted at different service companies, which makes adoption of better practices a hard coordination problem too. As a result, server passwords often appear in scripts or other plaintext files, which can end up in Dropbox or Splunk. So it is vital to think of password practices beyond end users. In later chapters we'll look at protocols such as Kerberos and ssh; for now, recall Ed Snowden's remark that it was trivial to hack the typical large company: just spear-phish a sysadmin and then chain your way in. Much of this chapter is about the 'spear-phish a sysadmin' part; but don't neglect the 'chain your way in' part.

### 3.4.8 Attacks on password entry

Password entry is often poorly protected.

#### 3.4.8.1 Interface design

Thoughtless interface design is all too common. Some common makes of cash machine have a vertical keyboard at head height, making it simple for a pickpocket to watch a customer enter her PIN before lifting her purse from her handbag. The keyboards may have been at a reasonable height for the men who designed them, but women who are a few inches shorter are exposed.

When entering a card number or PIN in a public place, I usually cover my typing hand with my body or my other hand – but you can't assume that all your customers will. Many people are uncomfortable shielding a PIN as it's a signal of distrust, especially if they're in a supermarket queue and a friend is standing nearby. UK banks found that 20% of users never shield their PIN [102] – and then used this to blame customers whose PINs were compromised by an overhead CCTV camera, rather than designing better PIN entry devices.

#### 3.4.8.2 Trusted path, and bogus terminals

A *trusted path* is some means of being sure that you're logging into a genuine machine through a channel that isn't open to eavesdropping. False terminal attacks go back to the dawn of time-shared computing. A public terminal would be left running an attack program that looks just like the usual logon screen – asking for a user name and password. When an unsuspecting user did this, it would save the password, reply 'sorry, wrong password' and then vanish, invoking the genuine password program. The user assumed they'd made a typing error and just entered the password again. This is why Windows had a *secure attention sequence*; hitting `ctrl-alt-del` was guaranteed to take you to a genuine password prompt. But eventually, in Windows 10, this got removed to prepare the way for Windows tablets, and because almost nobody understood it.

ATM skimmers are devices that sit on an ATM's throat, copy card details, and have a camera to record the customer PIN. There are many variants on the theme. Fraudsters deploy bad PIN entry devices too, and have even been jailed for attaching password-stealing hardware to terminals in bank branches. I'll describe this world in much more detail in the chapter on Banking and Bookkeeping; the long-term solution has been to move from magnetic-strip cards that are easy to copy to chip cards that are much harder. In any case, if a terminal might contain malicious hardware or software, then passwords alone will not be enough.

#### 3.4.8.3 Technical defeats of password retry counters

Many kids find out that a bicycle combination lock can usually be broken in a few minutes by solving each ring in order of looseness. The same idea worked against a number of computer systems. The PDP-10 TENEX operating system checked passwords one character at a time, and stopped as soon as one of them was wrong. This opened up a *timing attack*: the attacker would repeatedly place a guessed password in memory at a suitable location, have it verified as part of a file access request, and wait to see how long it took to be rejected [852]. An

error in the first character would be reported almost at once, an error in the second character would take a little longer to report, and in the third character a little longer still, and so on. So you could guess the characters once after another, and instead of a password of $N$ characters drawn from an alphabet of $A$ characters taking $A^N/2$ guesses on average, it took $AN/2$. (Bear in mind that in thirty years' time, all that might remain of the system you're building today is the memory of its more newsworthy security failures.)

These same mistakes are being made all over again in the world of embedded systems. With one remote car locking device, as soon as a wrong byte was transmitted from the key fob, the red telltale light on the receiver came on. With some smartcards, it has been possible to determine the customer PIN by trying each possible input value and looking at the card's power consumption, then issuing a reset if the input was wrong. The reason was that a wrong PIN caused a PIN retry counter to be decremented, and writing to the EEPROM memory which held this counter caused a current surge of several milliamps – which could be detected in time to reset the card before the write was complete [831]. These implementation details matter. Timing channels are a serious problem for people implementing cryptography, as we'll discuss at greater length in the next chapter.

A recent high-profile issue was the PIN retry counter in the iPhone. My colleague Sergei Skorobogatov noted that the iPhone keeps sensitive data encrypted in flash memory, and built an adapter that enabled him to save the encrypted memory contents and restore them to their original condition after several PIN attempts. This enabled him to try all 10,000 possible PINs rather than the ten PINs limit that Apple tried to impose [1311][5].

### 3.4.9 Attacks on Password Storage

Passwords have often been vulnerable where they are stored. In MIT's 'Compatible Time Sharing System' `ctss` – a 1960s predecessor of Multics – it once happened that one person was editing the message of the day, while another was editing the password file. Because of a software bug, the two editor temporary files got swapped, and everyone who logged on was greeted with a copy of the password file! [365].

Another horrible programming error struck a UK bank in the late 1980s, which issued all its customers with the same PIN by mistake [40]. As the procedures for handling PINs meant that no one in the bank got access to anyone's PIN other than his or her own, the bug wasn't spotted until after thousands of customer cards had been shipped. Big blunders continue: in 2019 the security company that does the Biostar and AEOS biometric lock system for building entry control and whose customers include banks and police forces in 83 countries left a database unprotected online with over a million people's IDs, plaintext passwords, fingerprints and facial recognition data; security researchers who discovered this from an Internet scan were able to add themselves as users [1371].

[5]This was done to undermine an argument by then FBI Director James Comey that the iPhone was unhackable and so Apple should be ordered to produce an operating system upgrade that created a backdoor; see section 24.2.8.

Auditing provides another hazard. When systems log failed password attempts, the log usually contains a large number of passwords, as users get the 'username, password' sequence out of phase. If the logs are not well protected then someone who sees an audit record of a failed login with a non-existent user name of `e5gv*8yp` just has to try this as a password for all the valid user names.

#### 3.4.9.1 One-way encryption

Such incidents taught people to protect passwords by encrypting them using a one-way algorithm, an innovation due to Roger Needham and Mike Guy. The password, when entered, is passed through a one-way function and the user is logged on only if it matches a previously stored value. However, it's often implemented wrong. The right way to do it is to generate a random key, historically known in this context as a *salt*; combine the password with the salt using a slow, cryptographically strong one-way function; and store both the salt and the hash.

#### 3.4.9.2 Password cracking

Some systems that use an encrypted password file make it widely readable. Unix used to be the prime example – the password file `/etc/passwd` was readable by all users. So any user could fetch it and try to break passwords by encrypting all the passwords in his dictionary and comparing them with the encrypted values in the file. We already mentioned in 3.4.4.1 the 'Crack' software that people have used for years for this purpose.

Most modern operating systems have sort-of fixed this problem; in modern Linux distributions, for example, passwords are salted, hashed using 5000 rounds of SHA-512, and stored in a file that only the root user can read. But there are still password-recovery tools to help you if, for example, you've encrypted an Office document with a password you've forgotten [1249]. Such tools can also be used by a bad man who has got root access, and there are still lots of badly designed systems out there where the password file is vulnerable in other ways.

There is also the cross-domain risk: a system is hacked, some passwords are cracked (or were even found unencrypted), and are then tried out on other systems to catch people who reused them. So password cracking is still worth some attention. One countermeasure worth considering is deception, in the form of bogus encrypted passwords that will alarm if anyone cracks the password file and attempts to use them (*password canaries*). You can have honeypot systems, that alarm if anyone ever logs on to them, honeypot user accounts on a live system, or *honeywords* – bogus encrypted passwords for genuine accounts [741].

#### 3.4.9.3 Remote password checking

Many systems check passwords remotely, using cryptographic protocols to protect the password in transit, and the interaction between password security and network security can be complex. Local networks often use a protocol called Kerberos, where a server sends you a key encrypted under your password; if

you know the password you can decrypt the key and use it to get tickets that give you access to resources. I'll discuss this in the next chapter, on protocols; it doesn't always protect weak passwords against an opponent who can wiretap encrypted traffic. Web servers mostly use a protocol called TLS to encrypt your traffic from the browser on your phone or laptop; I discuss TLS in the following chapter, on cryptography. TLS does not protect you if the server gets hacked. However there is a new protocol called Simultaneous Authentication of Equals (SAE) which is designed to set up secure sessions even where the password is guessable, and which has been adopted from 2018 in the WPA3 standard for wifi authentication. I'll discuss this later too.

And then there's OAuth, a protocol which allows access delegation, so you can grant one website the right to authenticate you using the mechanisms provided by another. Developed by Twitter from 2006, it's now used by the main service providers such as Google, Microsoft and Facebook to let you log on to media and other sites; an authorisation server issues access tokens for the purpose. We'll discuss the mechanisms later too. The concomitant risk is cross-site attacks; we are now (2019) seeing OAuth being used by state actors in authoritarian countries to phish local human-rights defenders. The technique is to create a malicious app with a plausible name (say 'Outlook Security Defender') and send an email, purportedly from Microsoft, asking for access. If the target responds they end up at a Microsoft web page where they're asked to authorise the app to have access to their data [32].

### 3.4.10 Absolute limits

If you have confidence in the cryptographic algorithms and operating-system security mechanisms that protect passwords, then the probability of a successful password guessing attack is a function of the entropy of passwords, if they are centrally assigned, and the psychology of users if they're allowed to choose them. Military sysadmins often prefer to issue random passwords, so the probability of password guessing attacks can be managed. For example, if $L$ is the maximum password lifetime, $R$ is login attempt rate, $S$ is the size of the password space, then the probability that a password can be guessed in its lifetime is $P = LR/S$, according to the US Department of Defense password management guideline [414].

There are issues with such a 'provable security' doctrine, starting with the attackers' goal. Do they want to crack a target account, or just any account? If an army has a million possible passwords and a million users, and the alarm goes off after three bad password attempts on any account, then the attacker can just try one password for every different account. If you want to stop this, you have to do rate control not just for every account, but for all accounts.

To take a concrete example, some UK government systems used to issue passwords randomly selected with a fixed template of consonants, vowels and numbers designed to make them easier to remember, such as CVCNCVCN (e.g. `fuR5xEb8`). If passwords are not case sensitive, the guess probability is only one in $21^4.5^2.10^2$, or about $2^{-29}$. So if an attacker could guess 100 passwords a second – perhaps distributed across 10,000 accounts on hundreds of machines on a network, so as not to raise the alarm – then he'd need about 5 million seconds,

or two months, to get in. So if you're defending such a system, you might find it prudent to do rate control: set a limit of say one password guess per ten seconds per user account, and perhaps by source IP address (though the attacker might use a botnet). You might also count the failed logon attempts and analyse them: is there a constant series of guesses that suggests an attempted intrusion? And what will you do once you notice one? Will you close the system down? Welcome back to the world of service denial.

With a commercial website, 100 passwords per second may translate to one compromised user account per second, because of poor user password choices. That may not be a big deal for a web service with 100 million accounts – but it may still be worth trying to identify the source of any industrial-scale password-guessing attacks. If they're from a small number of IP addresses, you can block them, but then the attacker may use a botnet. But if an automated guessing attack does emerge, then another way of dealing with it is the CAPTCHA, which I'll describe in section 3.5.

### 3.4.11 Using a password manager

Since the 1980s, companies have been selling single sign-on systems that remember your passwords for multiple applications, and when browsers came along in the mid-1990s and people started logging into dozens of websites, password managers became a mass-market product. Browser vendors noticed, and started providing much the same functionality for free.

Choosing random passwords and letting your browser remember them can be a pragmatic way of operating. The browser will only enter the password into a web page with the right URL (IE) or the same hostname and field name (Firefox). Browsers let you set a master password, which encrypts all the individual site passwords and which you only have to enter when your browser is updated. The main drawbacks of password managers in general are that you might forget the master password; and all your passwords may be compromised at once, since malware writers can work out how to hack common products. This is a particular issue when using a browser, and another is that a master password is not always the default so many users don't invoke it. (The same holds for other security services you get as options with platforms, such as encrypting your phone or laptop.) An advantage of using the browser is that you may be able to sync passwords between the browser in your phone and that in your laptop.

Third-party password managers can offer more, such as choosing long random passwords for you, identifying passwords shared across more than one website, and providing more controllable ways for you to manage the backup and recovery of your password collection. (With a browser, this comes down to backing up your whole laptop or phone.) They can also help you track your accounts, so you can see whether you had a password on a system that's announced a breach.

Many banks try to disable storage, whether by setting `autocomplete="off"` in their web pages or using other tricks that block password managers too. Banks think this improves security, but I'm not at all convinced. Stopping people using

password managers or the browser's own storage will probably make most of them use weaker passwords. The banks may argue that killing autocomplete makes compromise following device theft harder, and may stop malware stealing the password from the database of your browser or password manager, but the phishing defence provided by that product is disabled – which probably exposes the average customer to much greater risk [1008]. It's also inconvenient; one bank that suddenly disabled password storage had to back down the following day, because of extremely severe reaction from customers [956]. People manage risk in all sorts of ways. I personally use different browsers for different purposes, and let them store low-value passwords; for important accounts, such as email and banking, I always enter passwords manually, and always navigate to them via bookmarks rather than by clicking on links. (And be sure to think through backup and recovery, and to exercise it to make sure it works. What happens when your laptop dies? When your phone dies? When someone persuades your phone company to link your phone number to their SIM? When you die – or when you fall ill and your partner needs to manage your stuff? Does he or she know where to find the envelope with the master passwords? Very few people get this right.)

### 3.4.12 Will we ever get rid of passwords?

Passwords are annoying, so many people have discussed getting rid of them, and the move from laptops to phones gives us a chance. The proliferation of IoT devices that don't have keyboards will force us to do without them for some purposes. A handful of firms have tried hard to get rid of passwords. One example is the online bank Monzo, which operates exclusively via an app. They leave it up to the customer whether they protect their phone using a fingerprint, a pattern lock, a PIN or a password. However they still use email to prompt people to upgrade. The most popular app that uses SMS to authenticate rather than a password may be WhatsApp. I expect that this will become more widespread; as it does, we'll see more and more attacks based on phone takeover, from SIM swaps through Android malware and SS7 hacking to simple physical theft. In such cases, recovery may come down to an email loop, making your email password more critical than ever – or to phoning a call centre and telling them your mother's maiden name.

Bonneau and colleagues analysed the options in 2012 [235]. There are many security criteria against which an authentication system can be evaluated, and we've worked through them in this section: its resilience to theft, to physical observation, to guessing, to malware and other internal compromise, to leaks from other verifiers, to phishing and to targeted impersonation. Other factors include ease of use, ease of learning, whether you need to carry something extra, error rate, ease of recovery, cost per user, and whether it's an open design that anyone can use. They concluded that most of the schemes involving net benefits were variants of the idea of single sign-on – and OpenID has indeed become widespread, with many people logging in to their favourite newspaper using Google or Facebook, despite the obvious privacy cost[6]. Beyond that, any

[6]Government attempts to set up single sign-on for public services have been less successful, with the UK 'Verify' program due to be shuttered in 2020 [1029]. There have been many

security improvements involve giving up one or more of the benefits of passwords, namely that they're easy, efficient and cheap.

Bonneau's survey gave high security ratings to physical authentication tokens such as the CAP reader, which enables people to use their bank cards to log on to online banking; bank regulators have already mandated two-factor authentication in a number of countries. Using something tied to a bank card gives a more traditional root of trust, at least with traditional high-street banks; a customer can walk into a branch and order a new card[7]. Firms that are targets of state-level attackers, such as Google and Microsoft, now give authentication tokens of some kind or another to all their staff.

Did the survey miss anything? Well, the old saying is 'something you have, something you know, or something you are' – or, as Simson Garfinkel engagingly puts it, 'something you had once, something you've forgotten, or something you once were'. The third option, biometrics, has started coming into wide use since high-end mobile phones started offering fingerprint readers. Some countries, like Germany, issue their citizens with ID cards containing a fingerprint, which may provide an alternate root of trust for when everything else goes wrong. We'll discuss biometrics in its own chapter later in the book.

Both tokens and biometrics are still mostly used with passwords, first as a backstop in case a device gets stolen, and second as part of the process of security recovery. So passwords remain the (shaky) foundation on which much of information security is built. What may change this is the growing number of devices that have no user interface at all, and so have to be authenticated using other mechanisms. One approach that's getting ever more common is trust on first use, also known as the 'resurrecting duckling' after the fact that a duckling bonds on the first moving animal it sees after it hatches. We'll discuss this in the next chapter, and also when we dive into specific applications such as security in vehicles.

Finally, you should think hard about how to authenticate customers or other people who exercise their right to demand copies of their personal information under data-protection law. In 2019, James Pavur sent out 150 such requests to companies, impersonating his fiancée [1385]. 86 firms admitted they had information about her, and many had the sense to demand her logon and password to authenticate her. But about a quarter were prepared to accept an email address or phone number as authentication; and a further 16 percent asked for easily forgeable ID. He collected full personal information about her, including her credit card number, her social-security number and her mother's maiden name. A threat intelligence firm with which she'd never interacted sent a list of her accounts and passwords that had been compromised. Given that firms

problems around attempts to entrench government's role in identity assurance, which I'll discuss further in the chapter on biometrics, and which spill over into issues from online services to the security of elections. It was also hard for other private-sector firms to compete because of the network effects enjoyed by incumbents. However in 2019 Apple announced that it would provide a new, more privacy-friendly single sign-on mechanism, and use the market power of its app store to force websites to support it. Thus the quality and nature of the privacy on offer is becoming a side-effect of battles fought for other motives. We'll analyse this in more depth in the chapter on economics.

[7]This doesn't work for branchless banks like Monzo; but they do take a video of you when you register so that their call centre can recognise you later.

face big fines in the EU if they don't comply with such requests within 30 days, you'd better work out in advance how to cope with them, rather than leaving it to an assistant in your law office to improvise a procedure. If you abolish passwords, and a former customer claims their phone was stolen, what do you do then? And if you hold personal data on people who have never been your customers, how do you identify them?

## 3.5 CAPTCHAs

Can we have protection mechanisms that use the brain's strengths rather than its weaknesses? The most successful innovation in this field is probably the CAPTCHA – the 'Completely Automated Public Turing Test to Tell Computers and Humans Apart.' These are the little visual puzzles that you often have to solve to post to a blog, to register for a free online account, or to recover a password. The idea is that people can solve such problems easily, while computers find them hard.

CAPTCHAs first came into use in a big way in 2003 to stop spammers using scripts to open thousands of accounts on free email services, and to make it harder for attackers to try a few simple passwords with each of a large number of existing accounts. They were invented by Luis von Ahn and colleagues [1449], who were inspired by the test famously posed by Alan Turing as to whether a computer was intelligent: you put a computer in one room and a human in another, and invite a human to try to tell them apart. The test is turned round so that a computer can tell the difference between human and machine.

Early versions set out to use a known 'hard problem' in AI such as the recognition of distorted text against a noisy background. The idea is that breaking the CAPTCHA was equivalent to solving the AI problem, so an attacker would actually have to do the work by hand, or come up with a real innovation in computer science. Humans were good at reading distorted text, while programs were less good. It turned out to be harder than it seemed.

Many of the image recognition problems posed by early systems turned out not to be too hard at all once smart people tried hard to solve them. There are also protocol-level attacks; von Ahn mentioned that in theory a spammer could get people to solve them as the price of access to free porn [1448]. In October 2007, this actually started to happen: spammers created a game in which you undress a woman by solving one CAPTCHA after another [153]. That's also when we saw the first commercial CAPTCHA-breaking tools arrive on the market [625]. There are also generic attacks that use signal-processing techniques inspired by the operation of the human visual system, which have become fairly efficient at solving at least a subset of most types of text CAPTCHA [555]. And security-economics research in underground markets has shown that by 2011 the action had moved to using humans; people in countries with incomes of a few dollars a day will solve CAPTCHAs for about 50c per 1000.

From 2014, the CAPTCHA has been superseded by the ReCAPTCHA, another of Luis von Ahn's inventions. Here the idea is to get a number of users to do some useful piece of work, and check their answers against each other. The

service initially asked people to transcribe fragments of text from Google books that confused OCR software; more recently you get a puzzle with eight pictures asking 'click on all images containing a shop front', which helps Google train its vision-recognition AI systems[8]. It pushes back on the cheap-labour attack by putting up two or three multiple-choice puzzles and taking tens of seconds over it, rather than allowing rapid responses.

The implementation of CAPTCHAs is often thoughtless, with accessibility issues for users who are visually impaired. And try paying a road toll in Portugal where the website throws up a CAPTCHA asking you to identify pictures with an object, if you can't understand Portuguese well enough to figure out what you're supposed to look for!

## 3.6 Summary

Psychology matters to the security engineer, because of deception and because of usability. Most real attacks nowadays target the user. Various kinds of phishing are the main national-security threat, the principal means of developing and maintaining the cybercrime infrastructure, and one of the principal threats to online banking systems. Other forms of deception account for much of the rest of the cybercrime ecosystem, which is roughly equal to legacy crime in both volume and value.

Part of the remedy is security usability, yet research in this field was long neglected, being seen as less glamorous than cryptography or operating systems. That was a serious error on our part, and from the mid-2000s we have started to realise the importance of making it easier for ordinary people to use systems in safe ways. Since the mid-2010s we've also started to realise that we also have to make things easier for ordinary programmers; many of the security bugs that have broken real systems have been the result of tools that were just too hard to use, from cryptographic APIs that used unsafe defaults to the C programming language. Getting usability right also helps business directly: PayPal has built a $100bn business through being a safer and more convenient way to shop online[9].

In this chapter, we took a whistle-stop tour through psychology research relevant to deception and to the kinds of errors people make, and then tackled authentication as a case study. Much of the early work on security usability focused on password systems, which raise dozens of interesting questions. We now have more and more data not just on things we can measure in the lab such as guessability, memorability, and user trainability, but also on factors that can only be observed in the field such as how real systems break, how real attacks scale and how the incentives facing different players lead to unsafe equilibria.

At the end of the first workshop on security and human behavior in 2008, the psychologist Nick Humphrey summed up a long discussion on risk. "We're all agreed," he said, "that people pay too much attention to terrorism and not enough to cybercrime. But to a psychologist this is obvious. If you want people

[8]There's been pushback from users who see a ReCAPTCHA saying 'click on all images containing a helicopter' and don't want to help in military AI research. Google's own staff protested at this research too and the program was discontinued.

[9]Full disclosure: I consult for them.

to be more relaxed in airports, take away the tanks and guns, put in some nice sofas and Mozart in the loudspeakers, and people will relax soon enough. And if you want people to be more wary online, make everyone use Jaws as their screen saver. But that's not going to happen as the computer industry goes out of its way to make computers seem a lot less scary than they used to be." And of course governments want people to be anxious about terrorism, as it bids up the police budgets and helps politicians get re-elected. So we give people the wrong signals as well as spending our money on the wrong things. Understanding the many tensions between the demands of psychology, economics and engineering is essential to building robust systems at global scale.

## Research Problems

Security psychology is one of the hot topics in 2019. In the second edition of this book, I wrote '*We also need more fundamental thinking about the relationship between psychology and security. Between the first edition of this book in 2001 and the second in 2007, the whole field of security economics sprang into life; now there are two regular conferences and numerous other relevant events. So far, security usability is in a fairly embryonic state. Will it also grow big and prosperous? If so, which parts of existing psychology research will be the interesting areas to mine?*'

Since then security psychology and the behavioral economics of security have attracted hundreds of researchers. As well as the established symposium on usable privacy and security – SOUPS – we now have other workshops, and there are regular security-psychology papers at the mainstream events.

My meta-algorithm for finding research topics is to look first at applications and then at neighbouring disciplines. An example of the first is safe usability: as safety-critical products from cars to medical devices acquire not just software and Internet connections, but complex interfaces and even their own apps, how can we design them so that they won't harm people by accident, or as a result of malice?

An example of the second, and the theme of the Workshop on Security and Human Behaviour, is what we can learn from disciplines that study how people deal with risk, ranging from anthropology and criminology to sociology, history and philosophy. The papers and liveblogs of these workshops are available online and contain links to a lot of fascinating things I've not had space to discuss here.

## Further Reading

The Real Hustle videos are probably the best tutorial on deception; a number of episodes are on YouTube. Meanwhile, the best book on social engineering is still Kevin Mitnick's *'The Art of Deception'* [987].

For how social psychology gets used and abused in marketing, the must-read book is Tim Wu's *'The Attention Merchants'* which tells the history of advertising [1513].

In the computer science literature, perhaps a good starting point is James Reason's *'Human Error'*, which tells us what the safety-critical systems community has learned from many years studying the cognate problems in their field [1174]. Then there are standard HCI texts such as [1147], while early papers on security usability appeared as [375] and on phishing appeared as [726].

As for behavioral economics, I get my students to read Danny Kahneman's Nobel prize lecture. For more technical detail, there's a volume of papers Danny edited just before that with Tom Gilovich and Dale Griffin [574], or the pop science book *'Thinking, Fast and Slow'* that he wrote afterwards [747]. An alternative view, which gives the whole history of behavioral economics, is Dick Thaler's *'Misbehaving: The Making of Behavioural Economics'* [1377]. For the applications of this theory in government and elsewhere, the standard reference is Dick Thaler and Cass Sunnstein's *'Nudge'* [1379]. Dick's later second thoughts about 'Sludge' are at [1378].

For a detailed history of passwords and related mechanisms, as well as many empirical results and an analysis of statistical techniques for measuring both guessability and recall, I strongly recommend Joe Bonneau's thesis [233], a number of whose chapters ended up as papers I cited above.

Finally, if you're interested in the dark side, *'The Manipulation of Human Behavior'* by Albert Biderman and Herb Zimmer reports experiments on interrogation carried out after the Korean War with US Government funding [184]. Known as the Torturer's Bible, it describes the relative effectiveness of sensory deprivation, drugs, hypnosis, social pressure and so on when interrogating and brainwashing prisoners. As for the polygraph and other deception-detection techniques used nowadays, the standard reference is Vrij [1450].