The Economist

Computer security is broken from top to bottom

As the consequences pile up, things are starting to improve

OVER a couple of days in February, hundreds of thousands of point-of-sale printers in restaurants around the world began behaving strangely. Some churned out bizarre pictures of computers and giant robots signed, “with love from the hacker God himself”. Some informed their owners that, “YOUR PRINTER HAS BEEN PWND’D”. Some told them, “For the love of God, please close this port”. When the hacker God gave an interview to Motherboard, a technology website, he claimed to be a British secondary-school pupil by the name of “Stackoverflowin”. Annoyed by the parlous state of computer security, he had, he claimed, decided to perform a public service by demonstrating just how easy it was to seize control.

Not all hackers are so public-spirited, and 2016 was a bonanza for those who are not. In February of that year cyber-crooks stole $81m directly from the central bank of Bangladesh—and would have got away with more were it not for a crucial typo. In August America’s National Security Agency (NSA) saw its own hacking tools leaked all over the internet by a group calling themselves the Shadow Brokers. (The CIA suffered a similar indignity this March.) In October a piece of software called Mirai was used to flood Dyn, an internet infrastructure company, with so much meaningless traffic that websites such as Twitter and Reddit were made inaccessible to many users. And the hacking of the Democratic National Committee’s e-mail servers and the subsequent leaking of embarrassing communications seems to have been part of an attempt to influence the outcome of the American elections.

Away from matters of great scale and grand strategy, most hacking is either show-off vandalism or simply criminal. It is also increasingly easy. Obscure forums oil the trade in stolen credit-card details, sold in batches of thousands at a time. Data-dealers hawk “exploits”: flaws in code that allow malicious attackers to subvert systems. You can also buy “ransomware”, with which to

encrypt photos and documents on victims' computers before charging them for the key that will unscramble the data. So sophisticated are these facilitating markets that coding skills are now entirely optional. Botnets—flocks of compromised computers created by software like Mirai, which can then be used to flood websites with traffic, knocking them offline until a ransom is paid—can be rented by the hour. Just like a legitimate business, the bot-herders will, for a few dollars extra, provide technical support if anything goes wrong.

The total cost of all this hacking is anyone's guess (most small attacks, and many big ones, go unreported). But all agree it is likely to rise, because the scope for malice is about to expand remarkably. "We are building a world-sized robot," says Bruce Schneier, a security analyst, in the shape of the "Internet of Things". The IoT is a buzz-phrase used to describe the computerisation of everything from cars and electricity meters to children's toys, medical devices and light bulbs. In 2015 a group of computer-security researchers demonstrated that it was possible to take remote control of certain Jeep cars. When the Mirai malware is used to build a botnet it seeks out devices such as video recorders and webcams; the botnet for fridges is just around the corner.

Not OK, computer

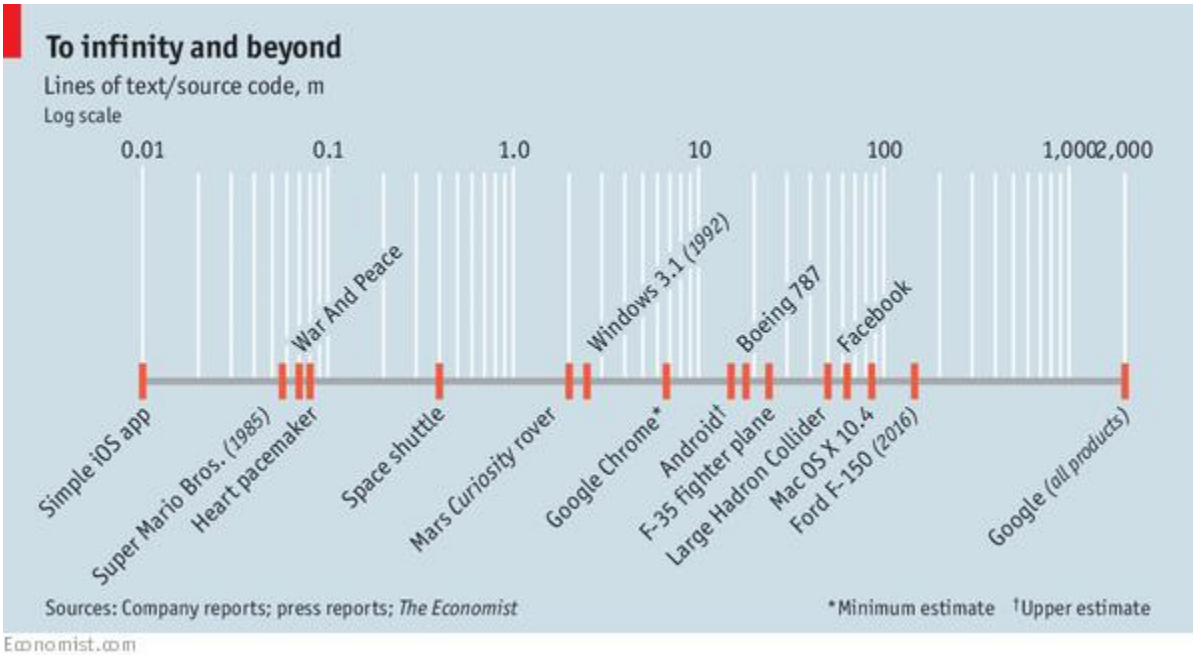
"The default assumption is that everything is vulnerable," says Robert Watson, a computer scientist at the University of Cambridge. The reasons for this run deep. The vulnerabilities of computers stem from the basics of information technology, the culture of software development, the breakneck pace of online business growth, the economic incentives faced by computer firms and the divided interests of governments. The rising damage caused by computer insecurity is, however, beginning to spur companies, academics and governments into action.

Modern computer chips are typically designed by one company, manufactured by another and then mounted on circuit boards built by third parties next to other chips from yet more firms. A further firm writes the lowest-level software necessary for the computer to function at all. The operating system that lets the machine run particular programs comes from someone else. The programs themselves from someone else again. A mistake

at any stage, or in the links between any two stages, can leave the entire system faulty—or vulnerable to attack.

It is not always easy to tell the difference. Peter Singer, a fellow at New America, a think-tank, tells the story of a manufacturing defect discovered in 2011 in some of the transistors which made up a chip used on American naval helicopters. Had the bug gone unspotted, it would have stopped those helicopters firing their missiles. The chips in question were, like most chips, made in China. The navy eventually concluded that the defect had been an accident, but not without giving serious thought to the idea it had been deliberate.

Most hackers lack the resources to mess around with chip design and manufacture. But they do not need them. Software offers opportunities for subversion in profusion. In 2015 Rachel Potvin, an engineer at Google, said that the company as a whole managed around 2bn lines of code across its various products. Those programs, in turn, must run on operating systems that are themselves ever more complicated. Linux, a widely used operating system, clocked in at 20.3m lines in 2015. The latest version of Microsoft's Windows operating system is thought to be around 50m lines long. Android, the most popular smartphone operating system, is 12m.



Economist.com

Getting each of those lines to interact properly with the rest of the program they are in, and with whatever other pieces of software and hardware that program might need to talk to, is a task that no one can get right first time. An oft-cited estimate made by Steve McConnell, a programming guru, is that people writing source code—the instructions that are compiled, inside a machine, into executable programs—make between ten and 50 errors in every 1,000 lines. Careful checking at big software companies, he says, can push that down to 0.5 per 1,000 or so. But even this error rate implies thousands of bugs in a modern program, any one of which could offer the possibility of exploitation. “The attackers only have to find one weakness,” says Kathleen Fisher, a computer scientist at Tufts University in Massachusetts. “The defenders have to plug every single hole, including ones they don’t know about.”

All that is needed is a way to get the computer to accept a set of commands that it should not. A mistake may mean there are outcomes of a particular command or sequence of commands that no one has foreseen. There may be ways of getting the computer to treat data as instructions—for both are represented inside the machine in the same form, as strings of digits. “Stackoverflowin”, the sobriquet chosen by the restaurant-printer hacker, refers to such a technique. If data “overflow” from a part of the system allocated for memory into a part where the machine expects instructions, they will be treated as a set of new instructions. (It is also possible to reverse the process and turn instructions into unexpected streams of data. In February researchers at Ben-Gurion University, in Israel, showed that they could get data out of a compromised computer by using the light that shows whether the hard drive is working to send those data to a watching drone.)

Shutting down every risk of abuse in millions of lines of code before people start to use that code is nigh-on impossible. America’s Department of Defence (DoD), Mr Singer says, has found significant vulnerabilities in every weapon system it examined. Things are no better on civvie street. According to Trustwave, a security-research firm, in 2015 the average phone app had 14 vulnerabilities.

Karma police

All these programs sit on top of older technologies that are often based on ways of thinking which date back to a time when security was barely a concern at all. This is particularly true of the internet, originally a tool whereby academics shared research data. The first versions of the internet were policed mostly by consensus and etiquette, including a strong presumption against use for commercial gain.

When Vint Cerf, one of the internet's pioneers, talked about building encryption into it in the 1970s he says his efforts were blocked by America's spies, who saw cryptography as a weapon for nation-states. Thus, rather than being secure from the beginning, the net needs a layer of additional software half a million lines long to keep things like credit-card details safe. New vulnerabilities and weaknesses in that layer are reported every year.

The innocent foundations of many computer systems remain a source for concern. So does the innocence of many users. Send enough people an innocuous-looking e-mail that asks for passwords or contains what look like data, but is in fact a crafty set of instructions, and you have a good chance that someone will click on something that they should not have done. Try as network administrators might to instil good habits in their charges, if there are enough people to probe, the chances of trust, laziness or error letting a malefactor get in are pretty high.

Good security cultures, both within software developers and between firms and their clients, take time to develop. This is one of the reasons to worry about the Internet of Things. "Some of the companies making smart light bulbs, say, or electricity meters, are not computing companies, culturally speaking," says Graham Steel, who runs Cryptosense, a firm that carries out automated cryptographic analysis. A database belonging to Spiral Toys, a firm that sells internet-connected teddy bears through which toddlers can send messages to their parents, lay unprotected online for several days towards the end of 2016, allowing personal details and toddlers' messages to be retrieved.

Even in firms that are aware of the issues, such as car companies, nailing down security can be hard. "The big firms whose logos are on the cars you buy, they don't really make cars," points out Dr Fisher. "They assemble lots of components from smaller suppliers, and increasingly, each of those has

code in it. It's really hard for the car companies to get an overview of everything that's going in."

On top of the effects of technology and culture there is a third fundamental cause of insecurity: the economic incentives of the computer business. Internet businesses, in particular, value growth above almost everything else, and time spent trying to write secure code is time not spent adding customers. "Ship it on Tuesday, fix the security problems next week—maybe" is the attitude, according to Ross Anderson, another computer-security expert at the University of Cambridge.

The long licence agreements that users of software must accept (almost always without reading them) typically disclaim any liability on the part of a software firm if things go wrong—even when the software involved is specifically designed to protect computers against viruses and the like. Such disclaimers are not always enforceable everywhere. But courts in America, the world's biggest software market, have generally been sympathetic. This impunity is one reason why the computing industry is so innovative and fast-moving. But the lack of legal recourse when a product proves vulnerable represents a significant cost to users.

If customers find it hard to exert pressure on companies through the courts, you might expect governments to step in. But Dr Anderson points out that they suffer from contradictory incentives. Sometimes they want computer security to be strong, because hacking endangers both their citizens and their own operations. On the other hand, computers are espionage and surveillance tools, and easier to use as such if they are not completely secure. To this end, the NSA is widely believed to have built deliberate weaknesses into some of its favoured encryption technologies.

Increasingly paranoid android

The risk is that anyone else who discovers these weaknesses can do the same. In 2004 someone (no authority has said who) spent months listening to the mobile-phone calls of the upper echelons of the Greek government—including the prime minister, Costas Karamanlis—by subverting surveillance capabilities built into the kit Ericsson had supplied to Vodafone, the pertinent network operator.

Some big companies, and also some governments, are now trying to solve security problems in a systematic way. Freelance bug-hunters can often claim bounties from firms whose software they find fault with. Microsoft vigorously nags customers to ditch outdated, less-secure versions of Windows in favour of newer ones, though with only limited success. In an attempt to squash as many bugs as possible, Google and Amazon are developing their own versions of standard encryption protocols, rewriting from top to bottom the code that keeps credit-card details and other tempting items secure. Amazon's version has been released on an "open-source" basis, letting all comers look at the source code and suggest improvements. Open-source projects provide, in principle, a broad base of criticism and improvement. The approach only works well, though, if it attracts and retains a committed community of developers.

More fundamental is work paid for by the Defence Advanced Research Projects Agency (DARPA), a bit of the DoD that was instrumental in the development of the internet. At the University of Cambridge, Dr Watson has been using this agency's money to design CHERI, a new kind of chip that attempts to bake security into hardware, rather than software. One feature, he says, is that the chip manages its memory in a way that ensures data cannot be mistaken for instructions, thus defanging an entire category of vulnerabilities. CHERI also lets individual programs, and even bits of programs, run inside secure "sandboxes", which limit their ability to affect other parts of the machine. So even if attackers obtain access to one part of the system, they cannot break out into the rest.

Sandboxing is already used by operating systems, web browsers and so on. But writing sandboxing into software imposes performance penalties. Having a chip that instantiates the idea in hardware gets around that. "We can have a web browser where every part of a page—every image, every ad, the text, and so on—all run in their own little secure enclaves," says Dr Watson. His team's innovations, he believes, could be added fairly easily to the chips designed by ARM and Intel that power phones and laptops.

Another DARPA project focuses on a technique called "formal methods". This reduces computer programs to gigantic statements in formal logic.

Mathematical theorem-proving tools can then be applied to show that a program behaves exactly as its designers want it to. Computer scientists have been exploring such approaches for years, says Dr Fisher, but it is only recently that cheap computing power and usable tools have let the results be applied to pieces of software big enough to be of practical interest. In 2013 Dr Fisher's team developed formally verified flight-control software for a hobbyist drone. A team of attackers, despite being given full access to the drone's source code, proved unable to find their way in.

"It will be a long time before we're using this stuff on something as complicated as a fully fledged operating system," says Dr Fisher. But she points out that many of the riskiest computing applications need only simple programs. "Things like insulin pumps, car components, all kinds of IoT devices—those are things we could look at applying this to."

Most fundamental of all, though, is the way in which markets are changing. The ubiquity of cyber-attacks, and the seeming impossibility of preventing them, is persuading big companies to turn to an old remedy for such unavoidable risks: insurance. "The cyber-insurance market is worth something like $3bn-4bn a year," says Jeremiah Grossman of SentinelOne, a company which sells protection against hacking (and which, unusually, offers a guarantee that its solutions work). "And it's growing at 60% a year."

As the costs of insurance mount, companies may start to demand more from the software they are using to protect themselves, and as payouts rise, insurers will demand the software be used properly. That could be a virtuous alignment of interests. A report published in 2015 by PwC, a management consultancy, found that a third of American businesses have cyber-insurance cover of some kind, though it often offers only limited protection.

But it is the issue of software-makers' liability for their products that will prove most contentious. The precedents that lie behind it belong to an age when software was a business novelty—and when computers dealt mostly with abstract things like spreadsheets. In those days, the issue was less pressing. But in a world where software is everywhere, and computerised cars or medical devices can kill people directly, it cannot be ducked for ever.

“The industry will fight any attempt to impose liability absolutely tooth and nail,” says Mr Grossman. On top of the usual resistance to regulations that impose costs, Silicon Valley’s companies often have a libertarian streak that goes with roots in the counterculture of the 1960s, bolstered by a self-serving belief that anything which slows innovation—defined rather narrowly—is an attack on the public good. Kenneth White, a cryptography researcher in Washington, DC, warns that if the government comes down too hard, the software business may end up looking like the pharmaceutical industry, where tough, ubiquitous regulation is one reason why the cost of developing a new drug is now close to a billion dollars. There is, then, a powerful incentive for the industry to clean up its act before the government cleans up for it. Too many more years like 2016, and that opportunity will vanish like the contents of a hacked bank account.