Chapter 1

History of Computer Security

Those who do not learn from the past will repeat it.

George Santanya

Security is a journey, not a destination. Computer security has been travelling for 40 years, and counting. On this journey, the challenges faced have kept changing, as have the answers to familiar challenges. This first chapter will trace the history of computer security, putting security mechanisms into the perspective of the IT landscape they were developed for.

OBJECTIVES

- Give an outline of the history of computer security.
- Explain the context in which familiar security mechanisms were originally developed.
- Show how changes in the application of IT pose new challenges in computer security.
- Discuss the impact of disruptive technologies on computer security.

1.1 THE DAWN OF COMPUTER SECURITY

New security challenges arise when new – or old – technologies are put to new use. The code breakers at Bletchley Park pioneered the use of electronic programmable computers during World War II [117, 233]. The first electronic computers were built in the 1940s (Colossus, EDVAC, ENIAC) and found applications in academia (Ferranti Mark I, University of Manchester), commercial organizations (LEO, J. Lyons & Co.), and government agencies (Univac I, US Census Bureau) in the early 1950s. *Computer security* can trace its origins back to the 1960s. Multi-user systems emerged, needing mechanisms for protecting the system from its users, and the users from each other. Protection rings (Section 5.6.4) are a concept dating from this period [108].

Two reports in the early 1970s signal the start of computer security as a field of research in its own right. The RAND report by Willis Ware [231] summarized the technical foundations computer security had acquired by the end of the 1960s. The report also produced a detailed analysis of the policy requirements of one particular application area, the protection of classified information in the US defence sector. This report was followed shortly after by the Anderson report [9] that laid out a research programme for the design of secure computer systems, again dominated by the requirement of protecting classified information.

> In recent years the Air Force has become increasingly aware of the problem of computer security. This problem has intruded on virtually any aspect of USAF operations and administration. The problem arises from a combination of factors that includes: greater reliance on the computer as a data-processing and decision-making tool in sensitive functional areas; the need to realize economies by consolidating ADP [automated data processing] resources thereby integrating or co-locating previously separate data-processing operations; the emergence of complex resource sharing computer systems providing users with capabilities for sharing data and processes with other users; the extension of resource sharing concepts to networks of computers; and the slowly growing recognition of security inadequacies of currently available computer systems. [9]

We will treat the four decades starting with the 1970s as historical epochs. We note for each decade the leading innovation in computer technology, the characteristic applications of that technology, the security problems raised by these applications, and the developments and state of the art in finding solutions for these problems. Information technologies may appear in our time line well after their original inception. However, a new technology becomes a real issue for computer security only when it is sufficiently mature and deployed widely enough for new applications with new security problems to materialize. With this consideration in mind, we observe that computer security has passed through the following epochs:

- 1970s: age of the mainframe,
- 1980s: age of the PC,
- 1990s: age of the Internet,
- 2000s: age of the web.

1.2 1970s – MAINFRAMES

Advances in the design of memory devices (IBM's Winchester disk offered a capacity of 35–70 megabytes) facilitated the processing of large amounts of data (for that time). Mainframes were deployed mainly in government departments and in large commercial organizations. Two applications from public administration are of particular significance. First, the defence sector saw the potential benefits of using computers, but classified information would have to be processed securely. This led the US Air Force to create the study group that reported its finding in the Anderson report.

The research programmes triggered by this report developed a formal state machine model for the multi-level security policies regulating access to classified data, the Bell–LaPadula model (Chapter 11), which proved to be highly influential on computer security research well into the 1980s [23]. The Multics project [187] developed an operating system that had security as one of its main design objectives. Processor architectures were developed with support for primitives such as segmentations or capabilities that were the basis for the security mechanisms adopted at the operating system level [92].

The second application field was the processing of 'unclassified but sensitive' data such as personal information about citizens in government departments. Government departments had been collecting and processing personal data before, but with mainframes data-processing at a much larger scale became a possibility. It was also much easier for staff to remain undetected when snooping around in filesystems looking for information they had no business in viewing. Both aspects were considered serious threats to privacy, and a number of protection mechanisms were developed in response.

Access control mechanisms in the operating system had to support multi-user security. Users should be kept apart, unless data sharing was explicitly permitted, and prevented from interfering with the management of the mainframe system. The fundamental concepts for access control in Chapter 5 belong to this epoch.

Encryption was seen to provide the most comprehensive protection for data stored in computer memory and on backup media. The US Federal Bureau of Standards issued a call for a data encryption standard for the protection of unclassified data. Eventually, IBM submitted the algorithm that became known as the Data Encryption Standard [221]. This call was the decisive event that began the public discussion about encryption algorithms and gave birth to cryptography as an academic discipline, a development deeply resented at that time by those working on communications security in the security services. A first key contribution from academic research was the concept of public-key cryptography published by Diffie and Hellman in 1976 [82]. Cryptography is the topic of Chapter 14.

In the context of statistical database queries, a typical task in social services, a new threat was observed. Even if individual queries were guaranteed to cover a large enough query

set so as not to leak information about individual entries, an attacker could use a clever combination of such 'safe' statistical queries to infer information about a single entry. Aggregation and inference, and countermeasures such as randomization of query data, were studied in database security. These issues are taken up in Section 9.4.

Thirdly, the legal system was adapted and data protection legislation was introduced in the US and in European countries and harmonized in the OECD privacy guidelines [188]; several legal initiatives on computer security issues followed (Section 9.6).

Since then, research on cryptography has reached a high level of maturity. When the US decided to update the Data Encryption Standard in the 1990s, a public review process led to the adoption of the new Advanced Encryption Standard. This 'civilian' algorithm developed by Belgian researchers was later also approved in the US for the protection of classified data [68]. For the inference problem in statistical databases, pragmatic solutions were developed, but there is no perfect solution and the data mining community is today re-examining (or reinventing?) some of the approaches from the 1970s. Multi-level security dominated security research into the following decade, posing interesting research questions which still engage theoreticians today – research on non-interference is going strong – and leading to the development of high-assurance systems whose design had been verified employing formal methods. However, these high-assurance systems did not solve the problems of the following epochs and now appear more as specialized offerings for a niche market than a foundation for the security systems of the next epoch.

1.3 1980s – PERSONAL COMPUTERS

Miniaturization and integration of switching components had reached the stage where computers no longer needed to be large machines housed in special rooms but were small enough to fit on a desk. Graphical user interfaces and mouse facilitated user-friendly input/output. This was the technological basis for the personal computer (PC), the innovation that, indirectly, changed the focus of computer security during the 1980s. The PC was cheap enough to be bought directly by smaller units in organizations, bypassing the IT department. The liberation from the tutelage of the IT department resounded through Apple's famous launch of the Macintosh in 1984. The PC was a single-user machine, the first successful applications were word processors and spreadsheet programs, and users were working on documents that may have been commercially sensitive but were rarely classified data. At a stroke, multi-level security and multi-user security became utterly irrelevant. To many security experts the 1980s triggered a retrograde development, leading to less protected systems, which in fairness only became less secure when they were later used outside their original environment.

While this change in application patterns was gathering momentum, security research still took its main cues from multi-level security. Information-flow models and

non-interference models were proposed to capture aspects not addressed in the Bell–LaPadula model. The Orange Book [224] strongly influenced the common perception of computer security (Section 13.2). High security assurance and multi-level security went hand in hand. Research on multi-level secure databases invented polyinstantiation so that users cleared at different security levels could enter data into the same table without creating covert channels [157].

We have to wait for the Clark–Wilson model (1987) [66] and the Chinese Wall model (1989) [44] to get research contributions influenced by commercial IT applications and coming from authors with a commercial background. Clark and Wilson present well-formed transactions and separation of duties as two important design principles for securing commercial systems. The Chinese Wall model was inspired by the requirement to prevent conflicts of interest in financial consultancy businesses. Chapter 12 covers both models.

A less visible change occurred in the development of processor architectures. The Intel 80286 processor supported segmentation, a feature used by multi-user operating systems. In the 80386 processor this feature was no longer present as it was not used by Microsoft's DOS. The 1980s also saw the first worms and viruses, interestingly enough first in research papers [209, 69] before they later appeared in the wild. The damage that could be done by attacking computer systems became visible to a wider public. We will briefly describe two incidents from this decade. Both ultimately led to convictions in court.

1.3.1 An Early Worm

The Internet worm of 1988 exploited a number of known vulnerabilities such as brute force password guessing for remote login, bad configurations (*sendmail* in debug mode), a buffer overrun in the *fingerd* daemon, and unauthenticated login from trusted hosts identified by their network address which could be forged. The worm penetrated 5–10% of the machines on the Internet, which totalled approximately 60,000 machines at the time. The buffer overrun in the *fingerd* daemon broke into VAX systems running Unix 4BSD. A special 536-byte message to the *fingerd* was used to overwrite the system stack:

```
pushl   $68732f         push '/sh, <NUL>'
pushl   $6e69622f       push '/bin'
movl    sp, r10         save address of start of string
pushl   $0              push 0 (arg 3 to execve)
pushl   $0              push 0 (arg 2 to execve)
pushl   r10             push string addr (arg 1 to execve)
pushl   $3              push argument count
movl    sp, ap          set argument pointer
chmk    $3b             do "execve" kernel call
```

The stack is thus set up so that the command `execve("/bin/sh",0,0)` will be executed on return to the *main* routine, opening a connection to a remote shell via

TCP [213]. Chapter 10 presents technical background on buffer overruns. The person responsible for the worm was brought to court and sentenced to a $10,050 fine and 400 hours of community service, with a three-year probation period (4 May 1990).

1.3.2 The Mad Hacker

This security incident affected ICL's VME/B operating system. VME/B stored information about files in *file descriptors*. All file descriptors were owned by the user `:STD`. For classified file descriptors this would create a security problem: system operators would require clearance to access classified information. Hence, `:STD` was not given access to classified file descriptors. In consequence, these descriptors could not be restored during a normal backup. A new user `:STD/CLASS` was therefore created who owned the classified file descriptors. This facility was included in a routine systems update.

The user `:STD/CLASS` had no other purpose than owning file descriptors. Hence, it was undesirable and unnecessary for anybody to log in as `:STD/CLASS`. To make login impossible, the password for `:STD/CLASS` was defined to be the RETURN key. Nobody could login because RETURN would always be interpreted as the delimiter of the password and not as part of the password. The password in the user profile of `:STD/CLASS` was set by patching hexadecimal code. Unfortunately, the wrong field was changed and instead of a user who could not log in, a user with an unrecognizable security level was created. This unrecognizable security level was interpreted as 'no security' so the designers had achieved the opposite of their goal.

There was still one line of defence left. User `:STD/CLASS` could only log in from the master console. However, once the master console was switched off, the next device opening a connection would be treated as the master console.

These flaws were exploited by a hacker who himself was managing a VME/B system. He thus had ample opportunity for detailed analysis and experimentation. He broke into a number of university computers via dial-up lines during nighttime when the computer centre was not staffed, modifying and deleting system and user files and leaving messages from *The Mad Hacker*. He was successfully tracked, brought to court, convicted (under the UK Criminal Damage Act of 1971), and handed a prison sentence. The conviction, the first of a computer hacker in the United Kingdom, was upheld by the Court of Appeal in 1991.

1.4 1990s – INTERNET

At the end of 1980s it was still undecided whether fax (a service offered by traditional telephone operators) or email (an Internet service) would prevail as the main method of document exchange. By the 1990s this question had been settled and this decade became without doubt the epoch of the Internet. Not because the Internet was created

in the 1990s – it is much older – but because new technology became available and because the Internet was opened to commercial use in 1992. The HTTP protocol and HTML provided the basis for visually more interesting applications than email or remote procedure calls. The World Wide Web (1991) and graphical web browsers (Mosaic, 1993) created a whole new 'user experience'. Both developments facilitated a whole new range of applications.

The Internet is a communications system so it may be natural that Internet security was initially equated with communications security, and in particular with strong cryptography. In the 1990s, the 'crypto wars' between the defenders of (US) export restrictions on encryption algorithms with more than 40-bit keys and advocates for the use of unbreakable (or rather, not obviously breakable) encryption was fought to an end, with the proponents of strong cryptography emerging victorious. Chapter 16 presents the communications security solutions developed for the Internet in the 1990s.

Communications security, however, only solves the easy problem, i.e. protecting data in transit. It should have been clear from the start that the real problems resided elsewhere. The typical end system was a PC, no longer stand-alone or connected to a LAN, but connected to the Internet. Connecting a machine to the Internet has two major ramifications. The system owner no longer controls who can send inputs to this machine; the system owner no longer controls what input is sent to the machine. The first observation rules out traditional identity-based access control as a viable protection mechanism. The second observation points to a new kind of attack, as described by Aleph One in his paper on 'Smashing the Stack for Fun and Profit' (1996) [6]. The attacker sends intentionally malformed inputs to an open port on the machine that causes a buffer overrun in the program handling the input, transferring control to shellcode inserted by the attacker. Chapter 10 is devoted to software security.

The Java security model addressed both issues. Privileges are assigned depending on the origin of code, not according to the identity of the user running a program. Remote code (applets) is put in a sandbox where it runs with restricted privileges only. As a type-safe language, the Java runtime system offers memory safety guarantees that prevent buffer overruns and the like. Chapter 20 explores the current state of code-based access control.

With the steep rise in the number of exploitable software vulnerabilities reported in the aftermath of Aleph One's paper and with several high profile email-based virus attacks sweeping through the Internet, 'trust and confidence' in the PC was at a low ebb. In reaction, Compaq, Hewlett-Packard, IBM, Intel, and Microsoft founded the Trusted Computing Platform Alliance in 1999, with the goal of 'making the web a safer place to surf'.

Advances in computer graphics turned the PC into a viable home entertainment platform for computer games, video, and music. The Internet became an attractive new distribution

channel for companies offering entertainment services, but they had to grapple with technical issues around copy protection (not provided on a standard PC platform of that time). Copy protection had been explored in the 1980s but in the end deemed unsuitable for mass market software; see [110, p. 59). In computer security, digital rights management (DRM) added a new twist to access control. For the first time access control did not protect the system owner from external parties. DRM enforces the security policy of an external party against actions by the system owner. For a short period, DRM mania reached a stage where access control was treated as a special case of DRM, before a more sober view returned. DRM was the second driving force of trusted computing, introducing remote attestation as a mechanism that would allow a document owner to check the software configuration of the intended destination before releasing the document. This development is taken up in Sections 15.6 and 20.7.

Availability, one of the 'big three' security properties, had always been of paramount importance in commercial applications. In previous epochs, availability had been addressed by organizational measures such as contingency plans, regular backup of data, and fall-back servers preferably located at a distance from a company's main premises. With the Internet, on-line denial-of-service attacks became a possibility and towards the end of the 1990s a fact. In response, firewalls and intrusion detection systems became common components of network security architectures (Chapter 17).

The emergence of on-line denial-of-service attacks led to a reconsideration of the engineering principles underpinning the design of cryptographic protocols. Strong cryptography can make protocols more exploitable by denial-of-service attacks. Today protocols are designed to balance the workload between initiator and responder so that an attacker would have to expend the same computational effort as the victim.

1.5 2000s – THE WEB

When we talk about the web, there is on one side the technology: the browser as the main software component at the client managing the interaction with servers and displaying pages to the user; HTTP as the application-level communications protocol; HTML and XML for data formats; client-side and server-side scripting languages for dynamic interactions; WLAN and mobile phone systems providing ubiquitous network access. On the other side, there are the users of the web: providers offering content and services, and the customers of those offerings.

The technology is mainly from the 1990s. The major step forward in the 2000s was the growth of the user base. Once sufficiently many private users had regular and mobile Internet access, companies had the opportunity of directly interacting with their customers and reducing costs by eliminating middlemen and unnecessary transaction steps. In the travel sector budget airlines were among the first to offer web booking of flights, demonstrating that paper tickets can be virtualized. Other airlines

followed suit. In 2008, the International Air Transport Association (IATA) abandoned printed airline tickets in favour of electronic tickets as part of its 'Simplifying the Business' initiative.

Similarly, the modern traveller can arrange hotel reservations, car rentals, and conference registrations on the Internet. Other successful commercial applications are the bookseller Amazon, the mail-order business in general, e-banking, and the auction site eBay. The latter is particularly interesting as it enables transactions between private citizens where identities only need to be revealed to the extent of giving a shipping address.

The application-level software implementing the services offered on the web has become a main target for attacks. Major attack patterns are SQL injection (Section 10.5.2), cross-site scripting (Chapter 18), and attacks against the domain name system (Section 17.2). Application software accounts for an increasing number of reported vulnerabilities and real attacks. Attacks have stolen contact data from Gmail users,[1] and a worm spread to over a million users on MySpace.[2] Cross-site scripting overtook buffer overruns as the number one software vulnerability in the Common Vulnerabilities and Exposures list in 2005 and ranked first in the 2007 OWASP Top Ten vulnerabilities.[3] In 2006 SQL injection ranked second in the CVE list.[4]

In line with the growth of commercial activities on the web, the picture of the attacker has changed. The hackers of the 1990s often matched the stereotype of a male in his teens or twenties with limited social skills. One could discuss whether they were laudable whistle blowers exposing flaws in commercial software or whether they were creating wanton damage simply in order to bolster their self-esteem. In rare cases, attacks were made for financial gain. Today, criminal organizations have moved into the web. Criminals have no interest in high profile fast spreading worm attacks. They prefer to place trojans on their victims' machines to harvest sensitive data such as passwords, PINs, or TANs, or to use the victims' machines as part of a botnet.

A further aspect of commercial life has gained momentum because of the availability of the Internet as a high bandwidth global communications infrastructure. Outsourcing, virtual organizations, grid and cloud computing describe facets of a business world where companies merge, split, form joint enterprises, and move part of their activities to subcontractors or subsidiaries abroad on a regular basis. Sensitive information has to be protected among these recurring changes. At the same time information is becoming ever more crucial to a company's success. Security policies have to be

[1] http://jeremiahgrossman.blogspot.com/2006/01/advanced-web-attack-techniques-using.html
[2] http://www.betanews.com/article/CrossSite_Scripting_Worm_Hits_MySpace/1129232391
[3] http://www.owasp.org/index.php/OWASP_Top_Ten_Project
[4] Steve Christey and Robert A. Martin. Vulnerability type distributions in CVE, May 2007. http://cve.mitre.org/docs/vuln-trends/vuln-trends.pdf

defined, enforced, and managed in distributed heterogeneous settings. Policy administration, policy decisions, and policy enforcement become separate activities, potentially carried out at different sites and by different partners. Policy languages should provide support for controlling the effects of merging policies or of importing local policies into an enterprise-wide policy. Policy management systems may present a console for getting a comprehensive view of the various policies coexisting in an enterprise and for setting those policies, so that management rather than the local system owners are in control of policy. Having an accurate and up-to-date view of the current state of a dynamic and global enterprise is a challenge not only for management but also for supervisory authorities. Compliance with regulations that ask management to be truly in control of their companies, e.g. the Sarbanes–Oxley Act, is a major task in today's enterprises.

Efforts in the specification and design of relevant security mechanisms are under way in several areas. Web services security standards address cryptographic protection for XML documents, generic patterns for authentication (SAML), access control (XACML) and much more. The future will show which of these standards have stood the test of time. Federated identity management is a related topic, with applications in heterogeneous organizations but also for single sign-on systems for customers of federated companies. The integration of different authentication, authorization and accounting (AAA) systems, driven in particular by the convergence of Internet and mobile phone services, raises interesting challenges. On access control, research is pursuing ideas first introduced in the work on trust management. In the design of policy languages research is looking for the right balance between the expressiveness of a language and the strength of its formal foundations. Chapter 20 gives an introduction to these developments.

1.6 CONCLUSIONS – THE BENEFITS OF HINDSIGHT

Innovations developed in research laboratories – the mouse, graphical user interfaces, the Internet, the World Wide Web, mobile communications, public-key cryptography, worms, and viruses – have found their way into the mass market. These innovations are, however, not always used as originally envisaged by their inventors. For example, the creators of the Internet were surprised when email turned out to be their most popular service, the PC was turned into an Internet terminal, and SMS was not expected to grow into the major application of the mobile phone network it is today.

There is a lesson for security. Not only inventors are inventive, but also users. Proponents of new technologies are often asked to take a precautionary approach, study the impact of their technology and develop appropriate security mechanisms in advance. This approach can work if the use of the technology follows expectations, but is likely to fail in the face of user innovations. There is the added danger that familiarity with the 'old' security

challenges and their solutions inhibits the appreciation of new challenges where standard state-of-the-art solutions no longer work, and actually would be an impediment to the user. Multi-level secure operating systems and database management systems rarely fit commercial organizations. The recommendation to authenticate all messages in Internet protocols [2] gave way to privacy protection demands.

Innovations research defines disruptive technologies as cheap and comparatively simple technologies that do not meet the requirements of sophisticated users, who would not be seen using them, but are adopted by a wider public that does not need the advanced features [42]. Eventually, the new technology acquires more and more advanced features while remaining cheap as it serves a large enough market. In the end even sophisticated users migrate to the new technology. For example, there was once a market for workstations with much more powerful graphics processors than in a normal PC, but hardly any of the workstation manufacturers have survived. As another example, Internet protocols that were not designed to provide quality of service (QoS) are replacing more and more of the protocols traditionally used in telephone networks. Disruptive technologies may also be a problem for security. Initially, security features are neither required by their users nor by the applications for which they are used, but by the time they are a platform for sensitive applications it becomes difficult to reintegrate security.

1.7 EXERCISES

Exercise 1.1 Examine how end users' responsibilities for managing security have changed over time.

Exercise 1.2 *Full disclosure* asks for all details of a security vulnerability to be disclosed. Does this lead to an increase or a decrease in security?

Exercise 1.3 It has been frequently proposed to make software vendors liable for deficiencies in their products. Who would benefit from such regulations?

Exercise 1.4 Computer security is often compared unfavourably with car safety, where new models have to be approved before they can be brought to market and where vendors recall models when a design problem is detected. Is traffic safety a good model for computer security? Do we need the equivalent of driving licences, traffic rules, and traffic police?

Exercise 1.5 Social networks are a new application that has grown rapidly in recent years. What new security challenges are posed by social networks?

Exercise 1.6 'The net does not forget.' To what extent is it possible to delete information once it has been published on the Internet?

Exercise 1.7 Attacks can come from inside or outside an organization. Are there basic differences in the defences against insider and outsider threats? What is the relative importance of insider threats? Has the relative importance of insider threats changed as the modern IT landscape has been formed?

Exercise 1.8 Examine how security regulations and security mechanisms may be used as trade barriers.