On bugs and elephants: Mining for science of security

Dusko Pavlovic

1. On security engineering

> A number of blind men came to an elephant. Somebody told them that it was an elephant. The blind men asked, "What is the elephant like?" and they began to touch its body. One of them said: "It is like a pillar." This blind man had only touched its leg. Another man said, "The elephant is like a husking basket." This person had only touched its ears. Similarly, he who touched its trunk or its belly talked of it differently.
>
> ~Ramakrishna Paramahamsa~

Security means many things to many people. For a software engineer, it often means that there are no buffer overflows or dangling pointers in the code. For a cryptographer, it means that any successful attack on the cypher can be reduced to an algorithm for computing discrete logarithms or to integer factorization. For a diplomat, security means that the enemy cannot read the confidential messages. For a credit card operator, it means that the total costs of the fraudulent transactions and of the measures to prevent them are low, relative to the revenue. For a bee, security means that no intruder into the beehive will escape her sting . . .

Is it an accident that all these different ideas go under the same name? What do they really have in common? They are studied in different sciences, ranging from computer science to biology, by a wide variety of different methods. Would it be useful to study them together?

1.1. What is security engineering?

If all avatars of security have one thing in common, it is surely the idea that *there are enemies and potential attackers out there.* All security concerns, from computation to politics and biology, come down to averting the adversarial processes in the *environment* that are poised to subvert the goals of the *system.* There are, for instance, many kinds of bugs in software, but only those that the hackers use are a security concern.

In all engineering disciplines, the system guarantees a functionality, provided that the environment satisfies some assumptions. This is the standard assume-guarantee format of the engineering correctness statements. Such statements are useful when the environment is passive so that the assumptions about it remain valid for a while. The essence of security engineering is that System and Environment face off as opponents, and Environment actively seeks to invalidate System's assumptions.

Security is thus an adversarial process. In all engineering disciplines, failures usually arise from some engineering errors. In security, failures arise in spite of compliance with the best engineering practices of the moment. Failures are the first-class citizens of security. For all major software systems, we normally expect security updates, which usually arise from attacks and often inspire them.

1.2. Where did security engineering come from?

The earliest examples of security technologies are found among the earliest documents of civilization. Figure 1, on the following page, shows security tokens with a tamper protection technology from almost 6,000 years ago. Figure 2 depicts the situation where this technology was probably used. Alice has a lamb and Bob has built a secure vault, perhaps with multiple security levels, spacious enough to store both Bob's and Alice's assets. For each of Alice's assets deposited

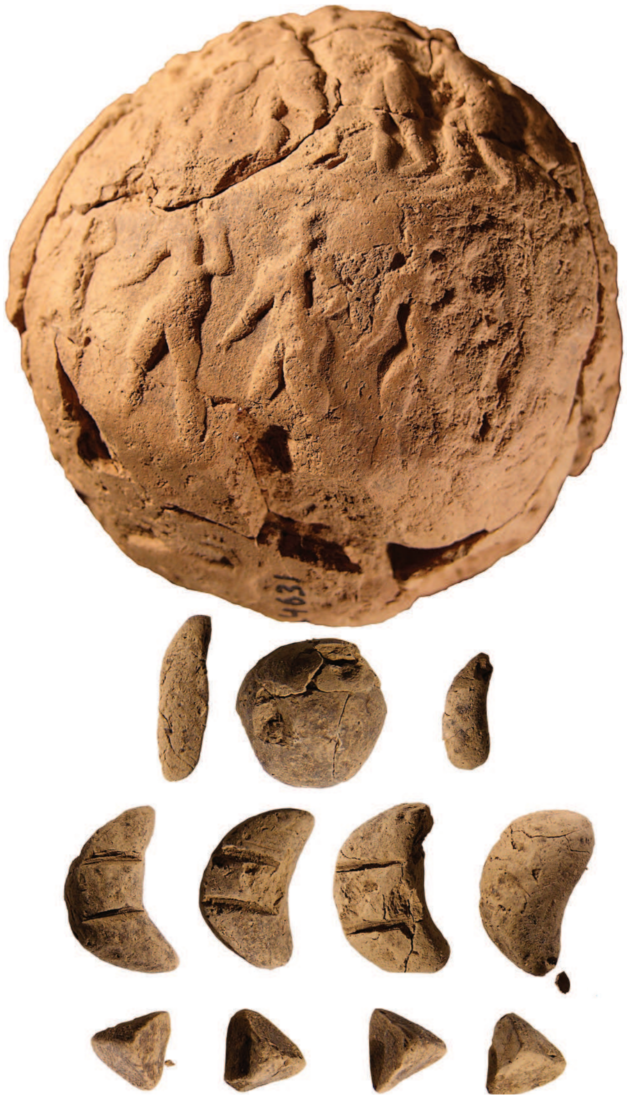

FIGURE 1. Tamper protection (bulla envelope with 11 plain and complex tokens inside) from the Near East, circa 3700–3200 BC. (The Schøyen Collection MS 4631. ©The Schøyen Collection, Oslo and London. Available at: www.schoyencollection.com.)

in the vault, Bob issues a clay token with an inscription identifying the asset. Alice's tokens are then encased into a bulla—a round, hollow envelope of clay—that is then baked to prevent tampering. When she wants to withdraw her deposits, Alice submits her bulla to Bob; he breaks it, extracts the tokens, and returns the goods. Alice can also give her bulla to Carol, who can also submit it to Bob to withdraw the goods, or pass it on to Dave. Bullae can thus be traded and facilitate an exchange economy. The tokens used in the bullae evolved into the earliest forms of money; and the inscriptions on them led to the earliest numeral systems, as well as to Sumerian cuneiform script, which was one of the earliest alphabets. Security thus predates literature, science, mathematics, and even money.

1.3. Where is security engineering going?

Through history, security technologies evolved gradually, serving the purposes of war and peace, protecting public resources and private property. As computers pervaded all aspects of social life, security became interlaced with computation, and security engineering came to be closely related with computer science. The developments in the realm of security are nowadays inseparable from the developments in the realm of computation. The most notable such development is, of course, cyberspace.

A brief history of cyberspace. In the beginning, engineers built computers and wrote programs to control computations. The platform of computation was the computer, and it was used to execute algorithms and calculations, allowing people to discover, for example, fractals, and to invent compilers that allowed them to write and execute more algorithms and more calculations more efficiently. Then the operating system became the platform of computation, and software was developed on top of it. The era of personal computing and enterprise software broke out. And then the Internet happened, followed by cellular networks, and wireless networks, and ad hoc networks, and mixed networks. Cyberspace emerged as the distance-free

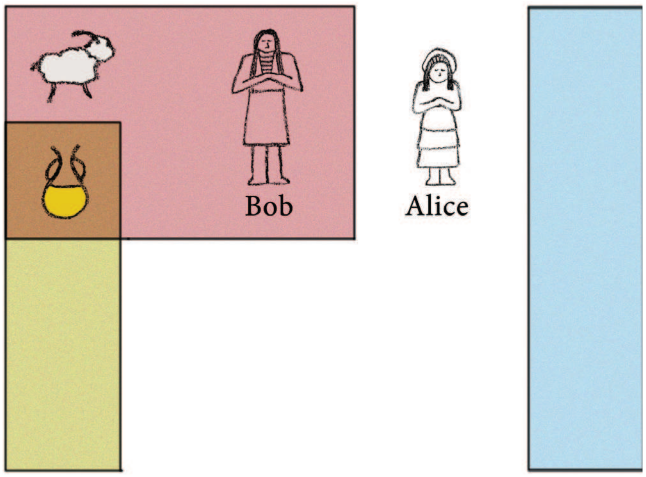

FIGURE 2. To withdraw her sheep from Bob's secure vault, Alice submits a tamper-proof token, like those shown in figure 1.

space of instant, costless communication. Nowadays, software is developed to run in cyberspace.

The Web is, strictly speaking, just a software system, albeit a formidable one. A botnet is also a software system. As social space blends with cyberspace, many social (business, collaborative) processes can be usefully construed as software systems that run on social networks as hardware. Many social and computational processes become inextricable. Table 1 summarizes the crude picture of the paradigm shifts that led to this remarkable situation.

TABLE 1. Paradigms of computation

	Ancient Times	Middle Ages	Modern Times
Platform	computer	operating system	network
Applications	Quicksort, compiler	MS Word, Oracle	WWW, botnets
Requirements	correctness, termination	liveness, safety	trust, privacy
Tools	programming languages	specification languages	scripting languages

But as every person got connected to a computer, and every computer to a network, and every network to a network of networks, computation became interlaced with communication and ceased to be programmable. The functioning of the web and of web applications is not determined by the code in the same sense as in a traditional software system; after all, web applications do include the human users as a part of their runtime. The fusion of social and computational processes in cybersocial space leads to a new type of information processing, where the purposeful program executions at the network nodes are supplemented by spontaneous data-driven evolution of network links. While the network emerges as the new computer, data and metadata become inseparable, and a new type of security problems arises.

A brief history of cybersecurity. In early computer systems, security tasks mainly concerned sharing of the computing resources. In computer networks, security goals expanded to include information protection. Both computer security and information security essentially depend on a clear distinction between the secure areas and the insecure areas, separated by a security perimeter. Security engineering caters for computer security and for information security by providing the tools to build the security perimeter. In cyberspace, the secure areas are separated from the insecure areas by the "walls" of cryptography, and they are connected through the "gates" of cryptographic protocols.

But as networks of computers and devices spread through physical and social spaces, the distinctions between the secure and the insecure areas become blurred. And in such areas of cybersocial space, where information processing does not yield to programming and cannot be secured by cryptography and protocols, security cannot be assured by engineering methodologies alone. The methodologies of data mining and classification, needed to secure such areas, form a bridge from information science to a putative security science.

2. On security science

> It is the aim of the natural scientist to discover mathematical theories, formally expressed as predicates describing the relevant observations that can be made of some [natural] system. . . . The aim of an engineer is complementary to that of the scientist. He starts with a specification, formally expressible as a predicate describing the desired observable behaviour. Then . . . he must design and construct a product that meets that specification.
>
> ~Tony Hoare~

The preceding quote was the first paragraph in one of the first papers on formal methods for software engineering, published under the title "Programs are predicates." Following this slogan, software has been formalized by logical methods and viewed as an engineering task ever since. But computation evolved, permeated all aspects of social life, and came to include not just the purposeful program executions, but also spontaneously evolving network processes. Data and metadata processing became inseparable. In cyberspace, computations are not localized at network nodes, but also propagate with nonlocal data flows and with the evolution of network links. While the local computations remain the subject of software engineering, network processes are also studied in the emerging software and information sciences, where the experimental validation of mathematical models has become the order of the day. Modern software engineering is therefore coupled with an empiric software science, as depicted in figure 3. In a similar way, modern security engineering needs to be coupled with an empiric security science.

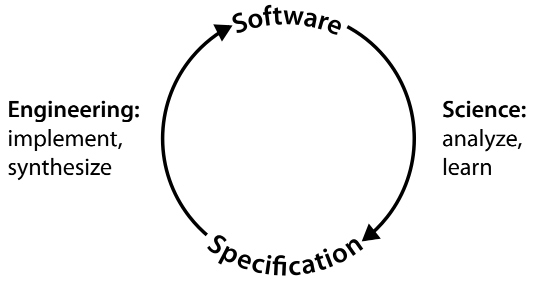

FIGURE 3. Conceptualization loop: The life cycle of computation.

2.1. Why security science?

Conjoining cyber, physical, and social spaces by networks gives rise to new security problems that combine computational, physical, and social aspects. They cross the boundaries of the disciplines where security was studied before, and require new modeling tools, and a new, unified framework, with a solid scientific foundation, and empiric methods to deal with the natural and social processes on which security now depends. In many respects, a scientific foundation for the various approaches to security would have been beneficial even before; but now it became necessary.

Let us have a closer look at the paradigm shift to postmodern cybersecurity in table 2. It can be illustrated as the shift from figure 4 to figure 5. The fortress in figure 4 represents the static, architectural view of security. A fortress consists of walls and gates separating the secure area within from the insecure area outside. The boundary between these two areas is the security perimeter. The secure area may be further subdivided into areas of higher security and areas of lower security. These intuitions extend into cyberspace, where crypto systems and access controls can be viewed as the walls, preventing the undesired traffic; whereas, authentication protocols and authorization mechanisms can be construed as the gates, allowing the desired traffic. But as every fortress owner knows, the walls and the gates are not enough for security; you also need weapons, soldiers, and maybe even some detectives and judges. They take care of the dynamic aspects of security. Dynamic security evolves through social processes, such as trust, privacy, reputation, or influence. The static and dynamic aspects depend on each other. For example, the authentication on the gates is based on some credentials intended to prove that the owner is honest. These credentials may be based on some older credentials, but down the line a first credential must have resulted from a process of trust building or from a trust decision, whereby the principal's honesty was accepted with no credentials. The word *credential* has its root in Latin *credo,* which means "I believe."

The attacks mostly studied in security research can be roughly divided into cryptanalytic attacks and protocol attacks. They are the cyber versions of the simple frontal attacks on the walls and the gates of a fortress. Such attacks are static in the sense that the attackers are outside, the defenders inside, and the two are easily distinguished. The dynamic attacks come about when some attackers penetrate the security perimeter and attack from within, as in figure 5. They may even blend with the defenders and become spies. Some of them may build up trust and infiltrate the fortress earlier, where they wait as moles. Some of the insiders may defect and become attackers. The traitors and the spies are the dynamic attackers; they use the vulnerabilities in the process of trust. To deter them, all cultures reserve for the breaches of trust the harshest punishments imaginable; Dante, in his description of Hell, places the traitors into the deepest, Ninth Circle. As a dynamic attack, treason was always much easier to punish than to prevent.

In cybersecurity, a brand new line of defense against dynamic attacks relies on predictive analytics, based on mining the data gathered by active or passive

TABLE 2. Paradigms of security

	Middle Ages	Modern Times	Postmodern Times
Space	computer center	cyberspace	cybersocial space
Assets	computing resources	information	public and private resources
Requirements	availability, authorization	integrity, confidentiality	trust, privacy
Tools	locks, tokens, passwords	cryptography, protocols	mining and classification

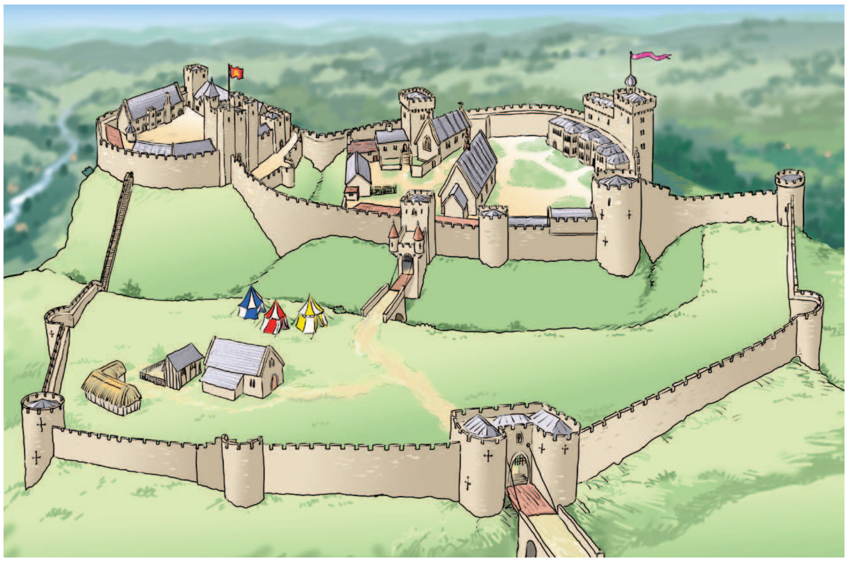

FIGURE 4. Static security: Multilevel architecture. (Illustration by Mark Burgess at www.markburgess.co.uk.)

observations, network probes, honeypots, or direct interactions. It should be noted that the expanding practices of predictive modeling are not engineering methodologies, geared toward building some specified systems, but the first simple tools of a security science, recognizing security as a process.

2.2. What is security science?

Although the security environment maliciously defies any system's assumptions that it can, security engineering still pursues its tasks strictly within the framework of the assume-guarantee methods. Indeed, to engineer a system, we must frame an environment for it; to guarantee system behavior, we must assume the environment behavior; to guarantee system security, we must specify an attacker model. That is the essence of the engineering approach. Following that approach, the cryptographic techniques of security engineering are based on the fixed assumption that the environment is computationally limited and cannot solve certain hard problems. (Defy that, Environment!)

But sometimes, as we have seen, it is not realistic to assume even that there is a clear boundary between the system and the environment. Such situations have become pervasive with the spread of networks supporting not only social, commercial, and collaborative applications, but also criminal and terrorist organizations. When there is a lot going on, you cannot be sure who is who. In large networks, with immense numbers of processes, the distinction between the system and the environment becomes meaningless, and the engineering *assume-guarantee approach* must be supplemented by the *analyze-adapt approach* of science. The task of the analyze-adapt approach of science is to recover the distinction between system and environment—whenever possible, albeit as a dynamic variable—and to adaptively follow its evolution. Similar situations, where engineering interventions are interleaved with scientific analyses, arise not only in security—where they elicit security science to support security engineering—but also, for example, in the context of health—where they elicit medical science to support health care. And just as health is not achieved by isolating the body from the external world, but by supporting its internal defense mechanisms, security is not achieved by erecting fortresses, but by supporting

FIGURE 5. Security dynamics: Threats within.

dynamic defenses, akin to the immune response. While security engineering provides blueprints and materials for static defenses, it is the task of security science to provide guidance and adaptation methods for dynamic defenses.

In general, science is the process of understanding the environment, adapting the system to it, changing the environment by the system, adapting to these changes, and so on. Science is thus an ongoing dialog of the system and the environment, separated and conjoined along the ever-changing boundaries. Dynamic security, on the other hand, is an ongoing battle between the ever-changing teams of attackers and defenders. Only scientific probing and analyses of this battle can tell who is who at any particular moment.

In summary, if security engineering is a family of methods to keep the attackers out, security science is a family of methods to catch the attackers once they get in.

It may be interesting to note that these two families of methods, viewed as strategies in an abstract security game, turn out to have opposite winning odds. It is often observed that the attackers only need to find one attack vector to enter the fortress, whereas the defenders must defend all attack vectors to prevent them. But when the battle switches to the dynamic mode and the defense moves inside, then the defenders only need to find one marker to recognize and catch the attackers; whereas, the attackers must cover all their markers. This strategic advantage is also the critical aspect of the immune response, where the invading organisms are purposely sampled and analyzed for chemical markers. In security science, this sampling and analyses take the form of data mining.

2.3. Where to look for security science?

The germs of a scientific approach to security, with data gathering, statistical analyses, and experimental validation, are already present in many intrusion detection and antivirus systems, as well as in spam filters and some firewalls. Such systems use measurable inputs and have quantifiable performance and model accuracy and thus conform to the basic requirements of the scientific method. The collaborative processes for sharing data, comparing models, and retesting and unifying results complete the social process of scientific research.

However, a broader range of deep security problems is still awaiting applications of a broader range of powerful scientific methods that are available in this realm. At least initially, the statistical methods of security science will need to be borrowed from information science. Security, however, imposes special data analysis requirements, some of which have been investigated in the existing work and led to novel approaches. In the long run, security science will undoubtedly engender its own domain-specific data analysis methods.

In general, security engineering solutions are based on security infrastructure: Internet protocol security (IPSec) suites, Rivest-Shamir-Adleman (RSA) systems, and elliptic curve cryptography (ECC) provide typical examples. In contrast, security science solutions emerge where the available infrastructure does not suffice for security. The examples abound—a mobile ad hoc network (MANET), for example, is a network of nodes with no previous contacts, direct or indirect, and thus no previous infrastructure. Although advanced MANET technologies have been available for more than 15 years, secure MANETs are still a bit of a holy grail. Device pairing, social network security, and web commerce security also require secure ad hoc interactions akin to the social protocols that regulate new encounters in social space. Such protocols are invariably incremental and accumulating, analyzing and classifying the data from multiple channels until a new link is established or aborted. Powerful data-mining methods have been developed and deployed in web commerce and financial security, but they are still awaiting systematic studies in noncommercial security research and systematic applications in noncommercial security domains.

3. Summary

Security processes are distributed, subtle, and complex, and there are no global observers. Security is like an elephant, and we are like the blind men touching its body. For the cryptographers among us, the security elephant consists of elliptic curves and of integers with large factors. Many software engineers among us derive their view of the security elephant entirely from their view of the software bugs flying around it.

Beyond and above all of our partial views is the actual elephant—people cheating each other, stealing secrets and money, forming online gangs and terrorist networks. There is a whole wide world of social processes of attacking and defending the assets by methods beyond the reach of security engineering. Such attacks and fraud cannot be debugged or programmed away; they cannot be eliminated by cryptography, protocols, or policies. Security engineering defers such attacks to the marginal notes about "social engineering."

However, since these attacks nowadays evolve in networks, the underlying social processes can be observed, measured, analyzed, understood, validated, and even experimented with. Security can be improved by security science, combining and refining the methods of information sciences, social sciences, and computational sciences.

Acknowledgements

Just like security, science of security also means many things to many people. I have presented one view of it, not because it is the only one I know, but mainly because it is the simplest one that I could think of, and maybe the most useful one. But some of my good friends and collaborators see it differently, and I am keeping an open mind. I am grateful to Brad Martin and Robert Meushaw for interesting conversations and, above all, for their initiative in this area.

About the author

Dusko Pavlovic is a professor of information security at Royal Holloway, University of London. He received his PhD in mathematics at the Utrecht University in 1990. His interests evolved from research in pure mathematics and theoretical computer science, through software design and engineering, to problems of security and network computation. He worked in academia in Canada, the United Kingdom, and the Netherlands, and in software research and development in the United States. Besides the chair in information security at Royal Holloway, he currently holds a chair in security protocols at University of Twente, and a visiting professorship at University of Oxford. His research projects are concerned with extending the mathematical methods of security beyond the standard cryptographic models toward capturing the complex phenomena that arise from physical, economic, and social aspects of security processes.