From cybercrime to cyberwarfare, nobody can overlook cybersecurity any more

Olivier Zendra, Bart Coppens

Many information technology (IT) systems still lack (cyber)security. The EU has much to lose from having poorly secured IT systems, and much to gain from secure ones. The good news is that we can do it, in Europe. Here is how.

From cybercrime to cyberwarfare, nobody can overlook cybersecurity any more

By OLIVIER ZENDRA and BART COPPENS

After decades of apparently low-intensity cyberattacks, during which security was not really thought of in most information technology (IT) systems, recent years, including those marked by the COVID-19 pandemic, have brought a flurry of well-organized, larger-scale attacks that have caused billions of euros of damage.

Such attacks have been made possible by the plethora of IT systems that have been produced with no or low security, a trend that has further increased with the rise of ubiquitous computing, with smartphones, the internet of things (IoT) and smart-* spreading everywhere with extremely low control.

However, although the current situation in IT systems can still be considered as critical and very much working in favour of cyber attackers, there are paths to massive but achievable technical improvements that can lead us to a much more secure and sovereign IT ecosystem, along with strong business opportunities in Europe. Furthermore, when it comes to cyberwarfare, improved cybersecurity is now the first line of defence, as the conflict in Ukraine has shown.

Key insights

- Cyberattacks are ever increasing, and the cost of damage caused by lack of cybersecurity is soaring.
- Cybercrime has become an industry, with tools and automation.
- Cyberwar has become prevalent, as shown in recent conflicts.
- The boundaries between cybercrime and cyberwarfare are blurred in some countries.
- Cyberattacks can be made at relatively low cost, and can cause very widespread and profound, even life-threatening, damage.
- Security must be a first-class concern of new programs from day one, in both specifications, source code and deployment. Methods and tools have to support this.
- However, legacy hasn't gone away: tools that are able to analyse the legacy code base, and find vulnerabilities and unwanted behaviour are needed, as are those that automatically circumvent or mitigate such vulnerabilities.
- Liability and certification in IT systems are crucial to the advance of security aspects. Security certification must become mandatory. Certification and liability reinforce one another. Regulation must play its role.
- Cybersecurity, hence IT sovereignty, which drastically impacts political sovereignty, depends on having control over hardware and software.
- Cybersecurity specialists are direly needed.

Key recommendations

- Invest in research on methods and tools to make security a first-class citizen during the development of new IT systems (including e.g. quantitative security metrics/properties), find vulnerabilities in existing IT systems (e.g. with static analyses on source code and behavioural analyses at runtime), and automatically prevent or mitigate them (e.g. with automated refactoring tools and blocking system).
- Broaden mandatory, EU-based security certification of IT systems, with several levels, and regulate to make IT systems providers and resellers liable in case of sub-par cybersecurity.
- Invest in research on automated methods and tools, possibly based on artificial intelligence (AI), for attack and intrusion detection, and for active cyber defence.
- To reclaim IT sovereignty:
  - base the critical parts of IT systems either on understood open-source software and hardware, or on EU-made, trustable because audited, proprietary hardware and software;
  - train more cybersecurity specialists and, through education, develop a culture of cybersecurity awareness in the greater workforce and public.
- Invest in cyberwarfare capacities, both for defence (resilience) and for counterstrike (deterrence).
- Harden both the hardware part and the software part of critical infrastructures and supply chains (utilities, transports, health, etc.) against cyberattacks, with reinforced cyber defences, monitoring and resilience capabilities (including redundancy).
- In addition to the above measures, harden edge and IoT devices against side-channel attacks and reverse engineering.

Cyber threats are real and costly

Barely a few years ago, cybersecurity was, if not unheard of, at least not on the minds of many people, including policy makers. IT systems seemed to be working, attacks only seemed to target "others"; in short, cybersecurity was a non-pressing matter, hence often non-existent… Since then, cyberattacks have made headlines: the influence on the 2016 US Presidential elections; the "NotPetya" ransomware attacks in 2016-2017 with losses estimated to US$10 billion [54][8], and the "Wannacry" ransomware attacks in 2017 leading to losses estimated to up to US$4 billion [53][9].

However, if awareness has indeed risen, it is still true that most of the world, including Europe, has not yet fully awakened to the cybersecurity aspect of IT systems.

Threats are numerous: malware in all its forms (spyware, ransomware, Trojan…), sniffing, spoofing, "man-in-the-middle" attacks, backdoors in hardware or software, etc. They are also spread across most if not all IT domains, ranging from the simplest, cheapest IoT devices to the more expensive smartphones, cars, planes, banking systems, air traffic control systems, etc. They know no frontiers, as IT information can easily cross (most) frontiers in the world, and know no delays thanks to the quasi-instantaneousness of information transmission.

Cyber threats used to be considered as being limited to hacking, which itself was seen as an uncommon activity of geeky-nerdy underground teenagers (see for example the 1983 movie Wargames [10]). Now, there is plenty of evidence that this has morphed into "industrial-scale" activities, with states using cyber warfare units to perform state-sponsored ransomware attacks even in relatively peaceful times [33], and organized cybercrime groups also having their professional hacker teams. Meanwhile, cyberattacks are crucial to modern warfare tactics, with the Russia-Ukraine conflict offering numerous examples [41]. Even a period of global pandemic such as that of COVID-19 did not create any truce on the cybersecurity front; on the contrary, "[c]ybercriminals [were] developing and boosting their attacks at an alarming pace, exploiting the fear and uncertainty caused by the unstable social and economic situation created by COVID-19." [11]

Security company McAfee estimated [55] that the global cost of cybercrime was over US$300 billion in 2013, US$600 billion in 2018, and US$1 trillion in 2020. These figures include direct costs such as cyber-

security expenses and paying for damage, but also "hidden costs [of cybercrime] that organizations may not be aware of, such as lost opportunities, wasted resources, and damaged staff morale." [55]. Most of this sharp increase does not come from better reporting but clearly from cybercrime expansion.

A very striking and telling example of the cybersecurity issues society faces is the sharp rise and spread of ransomware attacks, which has even led to the creation of a new term: "RaaS". As explained in [26], "Ransomware as a Service (RaaS) is a business model used by ransomware developers, in which they lease ransomware variants in the same way that legitimate software developers lease SaaS products. RaaS gives everyone, even people without much technical knowledge, the ability to launch ransomware attacks just by signing up for a service".

Statistics about ransomware can get one's head spinning. According to [25,27], 90% of financial institutions have been targeted by ransomware attacks. More than 68,000 new ransomware trojans for mobile were found in 2019 [25,28]. The consequences of ransomware attacks are major. The cost of ransomware attacks surpassed US$7.5 billion in 2019 [25,31]. The average downtime a company experiences after a ransomware attack was 21 days [25,30] in Q4 2020, an 11% increase from Q3. As found in [25,29], 80% of victim companies who paid a ransom experienced another attack soon after, while 46% got access to their data but most of it was corrupted; 60% experienced revenue loss and 53% stated their brands were damaged as a result; and, even worse, 29% respondents stated their companies were forced to remove jobs following a ransomware attack. Lists of affected companies and institutions flourish on the internet [32], but these may be only the tip of the iceberg (or, shall we say, ransomberg).

The healthcare sector has become one of the favourite targets of ransomware attacks, even at the most critical moments of the COVID-19 pandemic. The Health Sector Cybersecurity Coordination Center (HC3) of the US Department of Health & Human Services found [33] that 34% of healthcare organizations had been hit by ransomware in 2020; 65% that were hit by ransomware in 2020 saw cybercriminals succeed in encrypting their data in the most significant attack, and 93% of affected organizations got their data back but only 69% of the encrypted data was restored after the ransom was paid. [25,31] found that in 2020, 560 healthcare facilities were affected by ransomware attacks in 80 separate incidents, while [25,34] stress that in September 2020 alone, cybercriminals infiltrated and stole 9.7 million medical records in the US. Overall, ransomware attacks on US healthcare organizations cost US$20.8 billion in 2020 [35]. Of course, attacks were not limited to the US, with for example a full-scale attack impacting most of the Irish health system [33,36], and many more being reported on hospitals all across Europe.

Another telling, recent example is the Log4j vulnerability [37] unveiled in December 2021 that could give attackers full control of a system and had cybersecurity stakeholders scrambling [38]. A logging utility for Java applications, Log4j is widely used, being a (small) part of numerous products as a common off-the shelf tool that many use without paying too much attention to it.

The potential magnitude of the vulnerabilities was thus paramount, but not fully understood at the beginning. Fortunately, the worst did not happen because mitigating factors existed [37], such as the fact that not all versions were affected, or that only specific configurations enabled tampering with the affected versions, or the fact that, despite integrating the affected application programming interface (API) of Log4j, some systems were using another non-vulnerable implementation of the API, etc. Yet this could be considered as a free warning of what could have happened when a seemingly unimportant, known but very widespread component is vulnerable.

This should further draw attention to the need for security inside the IT industry supply chain, because (unintended) vulnerabilities or (malevolent) backdoors put in systems though the supply chain can allow massive-scale attacks.

The rise and spread of supply-chain attacks

Indeed, the first category of such supply-chain attacks is that targeting digital supply chains, i.e. supply chains used to provide software and software dependencies. Many modern software projects do not create every line of code from scratch: they depend on a multitude of other projects and frameworks. This makes total sense: it means that developers can create software in their own field of speciality, while at the same time drawing upon others' specialities. This makes software development *faster and cheaper*, as one can rely on pre-existing code bases for many features; but also *more secure*, as an app developer need not hold several PhDs in cryptographic engineering and related fields in order to create a secure network connection for their app: they can rely on the expertise of countless of specialists by simply importing external code.

However, this also creates a new attack surface, as attackers can now try to insert malicious changes into commonly used frameworks. These malicious changes then automatically get pulled in by the victims who update their dependencies to the most recent version (which is in and of itself necessary to stay secure, because the most recent version will contain the most recent fixes for security vulnerabilities). For example, contributors to the Python Package Index, a large software repository containing code which is used by developers all around the world, have been targeted in phishing attacks. Attackers could then use the phished credentials to upload malicious versions of the packages maintained by the phished users [56].

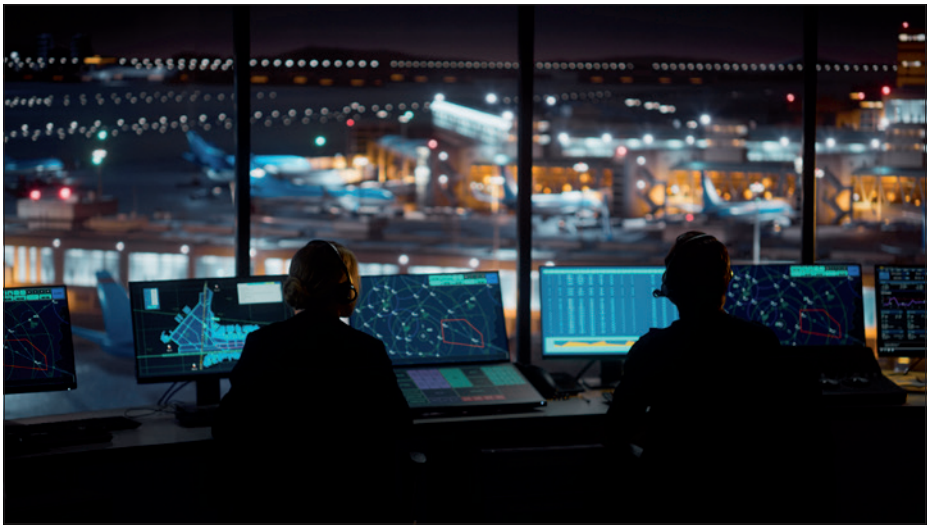
While external attackers are the most common source of this kind of supply-chain attack, there is also the possibility of a legitimate developer going rogue and inserting malicious changes themselves. An interesting recent example is that of a developer of a JavaScript library. When the war between Russia and Ukraine started, this developer inserted changes in their own code that would detect if the code was deployed on a computer with an IP address located in either Russia or Belarus; if so, it would wipe files from that machine [57].

Another famous example is the 2020 United States federal government data breach [39], which included the famous SolarWinds attack [58]. This series campaign included a supply-chain attack on Microsoft cloud services, which allowed an avenue for attackers to breach victims who had purchased these services through a reseller, and a supply-chain attack on SolarWinds' widely used Orion software. Multiple weaknesses in other products provided attackers with further access.

Due to the sensitivity and relevance of the targets as well as the length of time (8-9 months), this cyber-espionage incident is regarded the worst in US history. At least 200 businesses may have experienced data leaks as a result of the hack. Among the affected international organizations were NATO, the United Kingdom and United States governments, the European Parliament, Microsoft, and more institutions [39].

Cyberattacks can also threaten cyber-physical systems, and thus even threaten the concrete physical world, through actual, physical supply chains. One famous early example is the so-called Aurora Generator Test [48], in which in 2007 the Idaho National Laboratory demonstrated how a cyberattack could destroy physical components of the electric grid, namely a 2.5MW diesel generator.

Outside of a lab, the first publicly acknowledged successful cyberattack on a power grid was the cyberattack of 23 December 2015 on the power grid of Ukraine [49] that left 230,000 people without electricity for several hours. This is a striking example of a critical supply-chain cyberattack. This highly sophisticated attack had been carefully planned and prepared for months, patiently gaining access, putting triggers in appropriate places in the Ukrainian power grid. It involved various techniques, trojans and viruses, and even malware based on good old-fashioned, 1990s-era Microsoft Word macros. The latter shows that even ancient attack vectors, which some would have considered to be no longer a threat, can still be used to wreak havoc. It was a complete attack, since not only the systems delivering electricity to the customers were targeted, but also the backup systems in distribution centres, so as to blind the operators trying to restore power. It was even complemented with a distributed denial of service attack on the operators call centres, to prevent customers from reporting the exact extent of the problem and being informed about what was going on. Although formal attribution of the attack remains difficult, Ukraine's intelligence community has said with utter certainty that Russia is behind the attack [49].



Cyberattacks now target critical infrastructure in areas such as energy distribution and air traffic control

The aforementioned "NotPetya" malware, which some consider the "most devastating cyberattack in history" [54], was specifically designed to target "complete energy companies, the power grid, bus stations, gas stations, the airport, and banks" [8], i.e critical infrastructure and mostly supply chains. Among those, world leaders like pharmaceutical company Merck, delivery company TNT Express (the European subsidiary of FedEx), Danish shipping company Maersk, German logistics company DHL, India's largest container port JNPT, as well as food companies and scores of others. Restoring minimal operations took days, and full operations months, to the affected companies.

The EU *Action Plan on Military Mobility 2.0* [42] of November 2022 "proposes measures to enhance the protection of the transport sector against cyber-attacks and other hybrid threats", calling "for a joint civil-military endeavour to ensure a resil-

ient and robust data sharing network with a high level of cybersecurity.", and inviting EU Member States "to ensure the cyber-resilience of the future digital processes and procedures, building on the European Defence Agency's (EDA) work [43] and exploring the possibility of developing functional requirements related to security". There, the matters addressed do not only pertain to supply-chain cybersecurity, but also edge into the cyberwarfare domain.

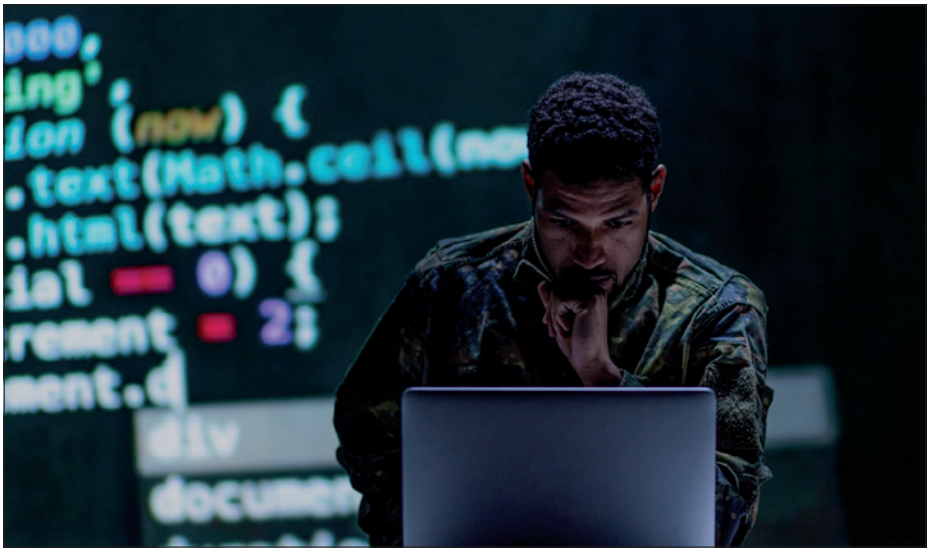
Cyberwarfare is upon us

Indeed, the **cyberattacks on critical supply chains** have now become more commonplace and are a clear part of cyberwarfare. Infiltration of computer networks used to be mostly about espionage (stealing secrets) but is now tied to destructive operations like sabotage and war preparation, as some very concrete examples show.

After the attack on their electricity grid in 2015 [49] mentioned above, Ukraine had reinforced its defences and prepared to defend against new attacks. This preparedness has shown itself to be useful, since the Ukraine power grid was again cyberattacked on 22 April 2022 by a Russian military hacking team [50], but Ukraine was this time able to prevent any damage.

Similarly, as reported [51] on 25 November 2022 by cyber security company Dragos, hacker groups Xenotime and Kamacite, which the FBI says are linked to Russian FSB and GRU secret services, have been found poking at the digital systems of Dutch LNG terminals in Rotterdam, which is a first step in enabling a possible future cyberattack.

It is thus very clear that attacks against critical energy supply chains can be, and have been, prepared, and in some case performed, in a real-world, physical war, to complement missiles and bombs. Given the criticality of such energy infrastructures, even a minor disruption of which can have very strong economical and practical impacts, not to mention the military impact, it is crucial that the EU reinforces its infrastructures and its cyberdefence teams, learning a great deal from allied countries that have already been impacted.



This **prevention and learning process through cooperation** is very effective, and is for example very much followed by the US cyber forces. Indeed, the US military has been conducting "Hunt Forward" missions in the computer networks of allied nations [47] for years. More specifically, since 2018, the United States has sent military personnel to twenty allies in Europe, the Middle East, and the Indo-Pacific. Lithuania announced in May 2022 that a three-month US deployment working on its defence and foreign affairs networks had recently been completed. Most recently [47], Croatia hosted a "thorough and successful hunt", in which malicious attacks on Croatian state infrastructure were discovered and prevented, according to its security and intelligence service. Other examples abound.

Although the majority of US cyber fighters' duties entail combating Chinese and North Korean state hackers, their most persistent opponent is Russia. The recent war in Ukraine provides us with a remarkably telling example. Indeed, in Ukraine, Russia coupled cyberattacks with a full-scale, physical conflict for the first time. In December 2021, a forty-member cyber military team from the United States had arrived in Ukraine to begin detecting Russian cyber fighters online [47]. Western intelligence officials monitored Russian military preparations, calculating that a wave of cyberattacks might precede and facilitate an invasion by damaging communications, power, banking, and government systems and destabilizing the nation prior to the invasion. Russian cyberattacks did hit Ukraine in January 2022. Hackers wrote on the website of the Ukrainian Foreign Ministry, ""Be afraid and expect the worst". Wiper malware hit multiple government sites.

On 24 February 2022, hours prior to the Russian invasion of Ukraine, a cyberattack disrupted a US satellite communications provider assisting the Ukrainian military. Critical infrastructure such as railways and the electricity system were anticipated to be disrupted by computer attacks, but this did not occur because the Ukrainians were better prepared, having learned from previous cyberattacks such as the one on their power grid on 23 December 2015 [49]. Help from their allies and the involvement of the Ukrainian commercial sector also played a role in boosting Ukrainian cyber defences and fending off the Russian cyberattacks. Ukraine's friends, led by the United States, are thus also learning from the conflict. It is providing teams with experience which can be put to use domestically. The US cyber fighters, for example, detected malware they had previously found when deployed in Europe in a government institution in the United States [47]. Since the invasion of Ukraine, offensive cyber actions have also been taken against Russia, although officials have not disclosed details.

Countries such as the United Kingdom, Germany, and France, which have their own cyber security expertise and personnel and are therefore more independent, may not experience such US deployments.

Indeed, even US allies are hesitant to allow the US to search sensitive government networks. Edward Snowden's revelations [52] a decade ago highlighted the extent to which the United States spied on allies as well as adversaries in the cyber world. This left profound traces of mistrust and continues to have repercussions. The fact that Gen. Paul Nakasone, who commands the US military's Cyber Command, also commands the National Security Agency, America's largest spy agency, can be a source of concern for allies, despite the fact that this dual command also provides a greater understanding and anticipation of potential cyber adversaries' actions. Trust has to be built that the US military are there to assist, not snoop, and relay their discoveries so that local partners can expel hackers.

Such trust should be easier to build on a EU level, although EU countries retain their sovereignty in cyberwarfare-related matters, as they do for conventional warfare. The process may thus be long and bumpy. Nonetheless, various levels of cooperation, with more or less tight integration, can be envisioned, and should thus be encouraged by EU.

The cyberwar in Ukraine also directly impacts the EU. On November 23, 2022, the **European Parliament's website** may have been subjected to its most sophisticated cyberattack ever [46]. The attack was a distributed denial-of-service (DDoS) attack, in which massive amounts of traffic are sent to servers in an attempt to bring them to a halt and make it impossible or excruciatingly slow to respond to users. This attack occurred just after the European Parliament voted to designate Russia as a state supporter of terrorism for its strikes on civilian targets in Ukraine. Killnet, a Russian-affiliated hacker organization, has claimed responsibility in a message posted to its Telegram channel. DDoS attacks are routinely used by hackers to cause chaos, and they are rumoured to be favoured by Russian hacking groups such as Killnet in the struggle against supporters for Ukraine in the war. This is a striking example of how cyberspace has become an extension of the physical fighting grounds opposing countries or blocks of countries, and how the EU is exposed to it.

As cyberattacks can impact the physical world, as we have seen for attacks on power grids, they can also be used to **directly target military assets**. The US Air Force (USAF), for example, is taking this issue very seriously. Since 2016, its Defense Digital Service has set up large-scale white-hacking competitions to find vulnerabilities in its fighter jets. As [45] reported in 2019, hackers simulated cyberattacks that could be made by opponents infiltrating the numerous suppliers of the Air Force. In doing so, they "found a mother lode of vulnerabilities that – if exploited in real life – could have completely shut down the Trusted Aircraft Information Program Download Station, which collects reams of data from video cameras and sensors while the [F-15] jet is in flight."

Even more shocking was the fact that several of these vulnerabilities had been identified and documented in previous evaluations but had not been fixed. However, the USAF, despite its huge budget, is similar to many institutions and businesses around the world in that costs and time must be limited, sometimes at the expense of thoroughness and quality, and most frequently at the risk of cybersecurity. Such sobering experimental evaluations are crucial in providing feedback on vulnerabilities that make systems more susceptible to cyberattacks, so helping in the correction of these defects.

However, attempting to rectify errors after a system has been constructed can be quite challenging. Legacy systems may be significantly out of date, and their providers may no longer have the necessary competence. Some systems are designed to be extremely difficult to modify. Some suppliers are protective of their intellectual property rights and do not want their systems to be accessed and investigated, let alone modified, which could result in liability-related legal complications. Therefore, it appears simpler for the USAF [45] to leverage the knowledge gathered from these white-hacking competitions to enforce a greater level of cybersecurity in the IT systems it acquires from suppliers.

The EU, or at least individual EU countries, should also take inspiration from this experience, and hold such white hacking events to further harden their critical, especially military, systems. Such events can very usefully complement the inside work already done by the various bodies (like DGA, the military procurement agency in France,) in charge of evaluating the quality of military materiel.

Another, extended form of cyberwarfare is **cognitive warfare** [44]. Although by far not limited to the cyber world, cognitive warfare thrives in it. The battlefield in cognitive warfare is the mind. Cognitive warfare is in a way the apex of influence, or propaganda (to use an old-fashioned word that somehow faded a bit after the collapse of the USSR). The objective of cognitive warfare is to influence people's thoughts and behaviour. It seeks to sow FUD (Fear Uncertainty and Doubt), to introduce conflicting narratives, to polarize opinions, to radicalize groups, and to motivate them to acts that can disrupt or fragment an otherwise cohesive society. It aims at influencing the thoughts and actions of individuals and groups in favour of the tactical or strategic objectives of an aggressor. In its most extreme form, it can divide a community to the point where it lacks the determination to oppose an adversary's goals, taming the community without any use of force.

Although cognitive warfare is not strictly to the cyber world, it is greatly facilitated by social networking, social messaging, and mobile gadgets. The widespread use of social media and smart device technologies render EU societies particularly vulnerable to this kind of unobtrusive yet very powerful attack.

Cybersecurity: let's build defences

The above examples clearly show that as a society all our systems, both industrial-scale and personal ones, must be as secure as possible. The stability and continuity of our daily activities, be they private or professional, and even our lives, depend heavily on the secure and continuous operation of IT systems. Until very recently, security was not a top priority for those who design and implement IT systems.

This has led to the current situation, with scores of vulnerable IT systems being used, and many still being developed with poor security.

Cybersecurity can be seen as a defence system, and has often been compared to medieval castles. Although it is true that some cyber defences can be nested like medieval castle defences were, the metaphor is inappropriate for whole IT systems. Unlike medieval castles, in which attackers must pass all the defences successively (climb the slope, and cross the outer moat, and climb or breach the outer wall, and the same for the inner wall, and for the dungeon), IT systems are in fact much less nested. They are more layered, or stacked, meaning that any breach present in either the hardware, or the operating system, or the execution environment, or the application, could be exploited by an attacker to gain information or control from the system. In that sense, one hole in IT systems may be enough. This is similar to the proverbial "forgotten, concealed postern" in a castle, or to a secret tunnel. Of course, there still may be some compartmentalization, in which case not the whole system falls but only part of it (a bit like one castle among several). But overall, IT systems and their defence appear rather more vulnerable in principle than the iconic medieval castle.

However, solutions exist to this grim situation, some of them specific to one layer of the IT system, others applicable to several layers. Many defence techniques exist, and can add their "stone" to the security walls of IT systems.

One way to create a "stronger stronghold" for IT systems is by having a smaller attack surface, by basing everything on a smaller yet highly secure base, like the trusted computing base (**TCB**) [1]. Research work has clearly shown that "From the security point of view, the monolithic OS design is flawed and a root cause of the majority of compromises." [22], and that **microkernels** make it easier to have more secure operating systems (OSes), even more so when they are verified. Indeed, OSes must have some parts that execute with the highest level of privileges, in kernel mode. But being software, kernels are also flawed. The bigger the kernel, the more flaws can be exploited in kernel mode to gain access to all information on the system. By reducing this attack surface to a bare minimum, the risk is mitigated. By reducing this kernel code to a bare minimum, it becomes easier to check it, secure it, or even verify it formally.

At the application and OS levels, a simple solution would be to use end-to-end cryptography, in which Europe is strong [40], in order to provide better protection for user data with regard to attacks, at storage level (encrypted filesystems), memory level and communication level. Recent OSes make it easy to encrypt files or the whole filesystem, and dedicated hardware support has made the cost of this largely unnoticeable in practice. At application level, several messaging or conferencing applications have introduced end-to-end crypto, some of them even putting a strong emphasis on this aspect (e.g., Olvid [23], Signal [24]). This need has only grown with COVID-19-induced teleworking, and the flurry of cyberattacks it has allowed [11].

Many more techniques exist. In the remainder of this article, we focus on a few techniques that must be supported and promoted since they provide the means for the EU to secure its IT systems and IT ecosystem. We show very promising *technical solutions* that make it possible to **find and fix vulnerabilities** in existing IT systems, to **strengthen** the security and **resilience** of IT systems, and to **express security properties** in order to be able to *verify* the security of existing systems and to *produce* new ones that are much more *secure* if not devoid of any vulnerability. However, if we do want to increase the security of our systems, we also need to *motivate* all actors to act in their best shared interest. This may imply **regulations**, based on **liability** in the face of the law and on **certification** processes. Finally, cybersecurity and IT sovereignty, hence simply **sovereignty** for the EU, depend on **trustable and auditable hardware and software**.

Finding and fixing vulnerabilities in existing systems

Since it is now infeasible to recreate all systems from scratch and redesign and rewrite them with perfect security, we will have to live with the legacy of our existing systems for a long time to come. As a result, these systems will need to be made secure to ensure that users and companies can access and store their private data for years to come.

In some cases, pragmatism can help us move forward on this path. As an example, consider the now commonplace technique of address space layout randomization (ASLR), which defends against memory-related vulnerabilities in software. Its core idea is that software will contain many bugs that can lead to exploitable vulnerabilities, but that attacks against these vulnerabilities were made exceedingly easy by the fact that program code and shared libraries were located on fixed addresses in memory. ASLR randomizes the locations of program code and libraries in memory, which makes such easy attacks fail. ASLR is a cross-layer technique, since it can be applied to the memory of application code/data, the execution environment (e.g. virtual machine, or VM) code/data, and OS code/data. Thus, applying ASLR to all systems increases the cost and difficulty of mounting attacks. While ASLR cannot prevent all possible attacks, users are nonetheless safer thanks to it.

This goes to show that in the cases where we cannot rigorously ensure the required security properties, we can still increase the overall security of systems through other means. In particular, **we need to have techniques to find, and fix or mitigate, security vulnerabilities in legacy code bases**; or techniques to at least isolate, thanks to containerization, these potentially insecure legacy code bases in hypervisor-like infrastructure, while at the same time controlling their exchanges with the rest of the system. While none of these techniques will be able to guarantee that programs are *completely* secure, each of them can lower the impact of inadvertent mistakes, and can raise the bar against specific attacks. The more of these techniques that are combined, the greater the security of the resulting system will be.

Even in large-scale systems, which are hard to deal with globally, vulnerability-finding techniques can perform a dual function. While an automated tool being unable to find security vulnerabilities in a certain amount of time does not constitute proof of the security of a system, if such an automated tool *does* find a security vulnerability, that vulnerability serves as constructive proof of the system's insecurity. Such security vulnerabilities (found automatically) can be passed as actionable items to the appropriate engineers to solve. This is a much more tractable task than trying to prove the security and correctness of an entire, large-scale system. For example, operating system kernels are complex, security-critical systems that are hard to analyse. Special-purpose techniques can be developed that target different kinds of bugs that can have a security impact in operating systems [12,13]. These analyses can still be improved in terms of how many such bugs they find in certain hard-to-analyse contexts. Furthermore, as such tools only target specific classes of bugs, more such tools need to be developed in order to find so-far under-reported classes of bugs.

For cases in which the systems still contain vulnerabilities which neither automated tools nor manual code analyses find, we need to have ways to mitigate the impact of these remaining vulnerabilities. This is typically done by targeting the ways in which an attacker typically exploits such vulnerabilities. An example of this was already touched upon in the context of ASLR: if typical attacks use hard-coded memory locations, making memory locations vary between executions will make attacks harder. Similarly, attackers can try to redirect the execution of the program in a way which the original developer did not write in the source code of the application. Then a type of defence, called control-flow integrity, inserts checks that verify that functions are executed only from the calling context of functions of which the developer intended this [14]. More such defences need to be researched and developed.

Diversity is a means of resilience against attacks

One way of mitigating and protecting against security vulnerabilities is introducing diversity. This is akin to avoiding monocultures in crops: having a more diverse gene pool in crops can make fields more resilient to diseases and infections. This was already alluded to in the context of ASLR: if every execution of a program has a totally unpredictable memory layout, it is harder for an attacker to create a single exploit that works against all these different memory layouts. An attacker would need to carefully craft an exploit that works around these limitations, or there would need to be a way for the attacker to gain knowledge of the exact memory layout.

Diversity can be introduced at many different, if not all, levels of an IT system: hardware, operating system, execution environment, application level. This can be done by having different implementations, by generating different versions at compile time [15], by randomizing the programs at install time[16], by randomizing them at load time such as with ASLR, or even by randomizing them during the execution of the program [15,16].

Diversity has even more security-related applications. For example, multi-variant techniques take their inspiration from reliability in critical systems such as aeroplanes in which multiple different implementations are compared against one another [18]. By executing (specially chosen) diversified instances of the same application, feeding these the same inputs, and then comparing their outputs, some classes of attack can be prevented from the fact that these attacks will impact these instances in a different way, which can be detected[19].

An application of diversity in a wider sense can be found in the context of security updates. When a developer discovers and fixes a security vulnerability and releases an update, users don't typically apply this update immediately. However, attackers can still compare the original application which contains the vulnerability with the just-released updated version of the application in which the vulnerability has been patched. Based on this differ-

ence, attackers can easily create attacks against the users who do not yet have the update installed [20]. This is sometimes called patch *Tuesday / exploit Wednesday*, as Microsoft typically releases their (security) updates on a Tuesday, after which attackers can try to exploit the unpatched users the day afterwards. A mitigation to this could consist of making the original version and the updated version differ more, that is, making the versions more diverse from each other. This will slow down the analysis and attack-generation by the attacker, allowing more users to apply the security update [21].

Some of these diversity techniques are already widely adopted, but not all. One reason that some proposed techniques are not yet used in practice is that they have too high an overhead to be applicable in most practical situations, or still have other limitations. Furthermore, as no technique can prevent every possible vulnerability, it is important that more such techniques are investigated and can be applied in practice. **This means researching new diversity techniques that are both efficient and effective, as well as making existing technologies more widely applicable,** by raising their maturity levels such that they can be adopted at large, rather than existing as academic prototypes.

Non-functional security properties must be included as first-class citizens in new IT systems

In addition to taking care of vulnerabilities in existing legacy systems, it is of the utmost importance to include security as a first-class citizen when building new IT systems. Far too often, security is not considered upfront in programs at the design and implementation stages, but only as an afterthought. This situation absolutely must change. The non-functional property that security is, or more precisely the set of non-functional properties that pertain to security in its various aspects, must be present in programs, in the minds of designers and implementers, from the very beginning of the creation of an IT system.

An example of a security property would be the level of threat by interception and decryption of communications in a given environment (e.g., known by its location in a military conflict). Another example would be, say, the strength of a cryptographic algorithm. An IT system could adapt its operation depending on the threat level, sending unencrypted or lightly (hence cheaply in term of time and energy) encrypted information in safe cases, choosing a stronger encryption algorithm in higher threat situations, or even avoiding transmission altogether if the level of threat is above the strength of the available algorithms.

To get to this point, security properties must be treated as first-class citizens, which means that they have to be expressed as clearly and as explicitly as the functional properties (i.e., what the program does, its algorithms), and that designers and developers should be able to reason about them, query them, manipulate them, just as for the functional aspects of the program. It is only by having security interleaved in all the fibres of the IT system that it can be solid.

To this end, first, security properties must be present in the **specifications** of the IT system from the beginning. Designers should be able to express the level of security they want, for the various facets of security (interception of communications, side-channel attacks, fault injection, malware, etc.), and reason about them. **Security contracts** must be present in the system that express and guarantee security at module or component boundaries. Developers should be able to pick up off-the-shelf **modules** or components knowing the security levels they provide for specific facets or security, and plug them in as part of the security continuum of their system.

Programming languages are also not all equal in terms of security. Many programming languages tolerate sloppy programming, where code that looks reasonable at first sight may in fact contain major vulnerabilities. Some, e.g., C/C++, tend to be harder to master and more error-prone, making it difficult to find bugs like security issues. For a compendium of programming language vulnerabilities, see the work by ISO/IEC TR 24772 Programming languages – *Guidance to avoiding vulnerabilities in programming languages* [2]. Other languages should be promoted: those that have stricter rules, more safeguards, and make it easier to develop more secure code. The Rust programming language, originally developed by Mozilla, is one example that makes many aspects of memory management and memory safety explicit in its language constructs, making it harder to leave room for security glitches that could be exploited malevolently.

Automated or semi-automated **tools** must be able to rely on these specification and security contracts **to verify security**, to prove security properties, to provide strong guarantees about the quality of IT systems with regard to security, at both specification and code level. Notable examples that can help in this endeavour in the line of program verification include Frama-C [3], Coq [5] and Ada SPARK's Discovery toolset [4]: those tools operate on the premise that the source code must conform to some formal specification.

Although **security by design** of whole IT systems seems a perfect answer to security issues, the current approach is often more limited and pragmatic, with only a limited part of the IT system being trusted. This trusted computing base (TCB), as mentioned above, comprises the system hardware, firmware and software components whose combination is intended to provide the system with mechanisms for a secure environment. The idea here is that verifying this, either automatically with verification tools or manually by human examination, on a small set of hardware and software is a more tractable and less costly task than doing it on the whole system. However, verification of large pieces of real-life software is already doable, as proven by recent research-level successes like the CompCert compiler [6] and the verified parts of the seL4 microkernel [7].

Blockchain: a technology that is still maturing

Blockchain has frequently been hailed as a disruptive, enabling, infrastructure technology for cybersecurity.

Blockchain is a potentially revolutionary technology, frequently referred to as the "trust machine," and it represents the first implementation of decentralized trust at present. According to Gartner [59], "Blockchain is clearly heading out of the Gartner Hype Cycle's Trough of Disillusionment, and now is the time to act." [60]. Nonetheless, it is worth examining the blockchains and their current status with attention.

The blockchain was born in 2009 with the emergence of Bitcoin, but its true potential was shown a few years later with the emergence of Ethereum and the concept of the "smart contract": not just money, but any digitized object may be exchanged, and this can be done according to sophisticated rules. In the past decade, blockchain technology has moved from digital currency (Blockchain 1.0) to smart contracts or programmable blockchains (Blockchain 2.0) to a vision of economically and legally sophisticated forms of decentralized collaboration (Blockchain 3.0).

Blockchains are a type of (digital) distributed ledger. Blockchain enables two or more individuals, corporations, or machines that may or may not know each other to exchange value in digital settings – in a monetary transaction, information exchange, or other asset exchange – without the need for a service provider. The recorded event could be a monetary transaction, as is generally recognized due to the initial deployment of blockchain for Bitcoin, but it could also be any information exchange.

Schematically, the functionality that a blockchain provides is confirming the pseudonymous identity of the participants, validating that the participants own the information/assets they wish to exchange, authenticating and approving the transaction, and recording the transaction information to the ledger, a copy of which is independently updated and held by each node on the network.

Blockchain integrates technologies and techniques such as distributed digital ledgers, encryption, immutable records management, asset tokenization, and decentralized governance to capture and record the data that network participants require to engage and transact. No intermediaries, like banks, validate and safeguard the transactions. Thus, the requirement for a central administration is eliminated.

Thus, the major purpose of blockchains is to facilitate user-to-user transactions in an environment where participants are "equal peers" and do not *a priori* trust one another. Instead of going via a central control organization that acts as a guarantee or "third party," the security of exchanges is achieved by maintaining a shared register that preserves the history of transactions. Using a computer protocol, each user keeps their own copy of the ledger, and each new transaction is validated locally by the user before being recorded. Due to the fact that each user has a local copy, it is evident that the ledger will be very accessible, as it is redundant on a large number of nodes.

The use of simple cryptographic techniques makes it extremely challenging, even locally, to alter the ledger's content without rendering it incorrect in the eyes of the other participants. To do so, one would have to reconstitute the contents of the registry from the modified record and corrupt the other participants into adopting the regenerated contents. As this is deemed almost impossible or ridiculous, then inadvertent or intentional modification remains nearly impossible, ensuring the ledger's integrity.

The blockchain revolution is technically predicated on the concept of "public verifiability", i.e., a technical system that allows anyone to independently verify the system's state's accuracy. In a distributed ledger, every observer can verify that each activity affecting the system's state is valid, that is to say, in line with the rules that govern the system, which are recognized by all. This verifiability is only achievable if the information required for validation is accessible, verifiable in a reasonable amount of time, and if the actions taken are observable.

Formally, a complete blockchain consists of five components.

1) **Tokenization**: Value is traded in the form of tokens that can represent a range of asset kinds, including monetary assets, data units, and user identities. Smart contracts may be used to program token usage.
2) **Distribution**: Blockchain users connected on a distributed network operate nodes (computers) that execute a program that enforces the blockchain's business rules. Nodes also maintain a complete copy of the distributed ledger, which is automatically updated when new transactions occur.
3) **Decentralization**: No single entity controls the majority of nodes or defines the rules. A consensus process checks and authorizes transactions, removing the need for a centralized intermediary to control the network. The manner in which governance is implemented can be modified.
4) **Encryption**: To record data safely and semi-anonymously, blockchain employs technologies such as public and private keys.
5) **Immutability**: Transactions are cryptographically signed, timestamped, and appended sequentially to the ledger. Changes to the record are not permitted unless all participants concur.

Wallets allow users to engage with the blockchain. A digital wallet gives its owner a pair of keys (private and public). Every record is signed using the issuer's private key. If the record is valid, other users can read it using the issuer's public key and accept it. A genuine digital signature provides the reader with strong evidence that the record was created by its issuer (authenticity) and has not been altered (integrity). Digital signatures also include a non-repudiation property, which means that the signer cannot declare that the signature is false. Each wallet is paired with a group of tokens, the wallet contents. Using the private key, only the owner of the wallet has the authority to transfer the related tokens. The digital wallet tied to a smart contract can also function as the contract's identifier within the system. In essence, users pay for the contract's execution.

Blockchain also holds the potential of improved transactional performance. We used to rely on sluggish, expensive, analogue-based techniques to verify identity and legal standing in economic transactions. Blockchain eliminates these methods. Equally crucial is that blockchain enables faster and more diverse transactions, both in terms of type and size, than are achievable with conventional centralized systems. For years, businesses have relied on centralized infrastructures, such as payment systems, insurance, delivery and logistics services, and governments, in order to execute commercial transactions and manage risk. However, these systems were not meant to handle the complexity, volume, and scale of the machine-to-machine transactions enabled by digital platforms. Without employing an intermediary that collects data on each side and gets a share of the transaction's value, businesses need a new method for handling new digital assets and interactions. Blockchain offers a solution.

Once blockchain becomes a mature technology, its benefits will have a profound effect on many facets of society. In constantly changing societal, industrial, and economic interactions, direct exchanges between users based on self-established, evolving, transparent, and verifiable rules at any time might very well open up new markets and services while simultaneously improving the quality of the services and goods exchanged.

However, the genuine appeal or utility of blockchain must be demonstrated in real-world applications before it can be compared to other more proven or specialized alternatives. Many of the proof-of-concept projects created in recent years attempted to tackle problems with blockchain technology that were not necessarily the best fit for demonstrating its actual utility. Undoubtedly, the massive hype surrounding blockchain has prompted a lot of organizations to put up demos too rapidly, while others have questioned whether blockchain is now more of a solution than it is a problem in itself. All of this has contributed to an unfairly negative perception of these technologies.

Aside from crypto-currencies and speculative objects, there are other reasons why these technologies have not yet been able to fully display their true potential, including their inherent complexity and a number of challenges [62]. Blockchain technologies are still in their infancy, despite having to meet extremely stringent security requirements. To provide trust, blockchain must operate as a flawless system. Replacing a bank, a notary, or an auditor with a computer protocol running on a network necessitates that the blockchain meets similar security standards as important software systems. This is not the case yet.

Thus, blockchain solutions are still maturing and may eventually underlie new social and economic models. Indeed, blockchain technologies are showing gradual advances. In recent years, there has been a profusion of proofs of concept or trials in sectors such as energy, food, mobility, pollution control, administration, etc., although frequently with modest goals and modest investments.

Utilizing tokenized real-world assets and smart contracts to control them, enterprise applications ranging from aviation maintenance to food safety are beginning to reap the first benefits of blockchain. But despite the fact that blockchain solutions are currently available, the majority of enterprise initiatives only incorporate a subset of the five fundamental blockchain components. In addition, apart from the now famous and occasionally controversial cryptocurrency trading, we have yet to see "killer use cases" that would have to surpass present applications in terms of improving our lives.

Additionally, blockchains experience security issues. Since the "bad guys" experiment with new technologies far quicker and earlier than the "good guys", it takes time for the "good use cases" to catch up, and it takes much more time to apply fraud and security controls [61].

Hence before blockchain can take over the world, numerous significant scientific and technical problems must be addressed.

First, viable alternatives to proof of work must be identified in a sustainable society if blockchain is to make a breakthrough in the sector. Therefore, research naturally focuses on these alternatives. Emerging alternatives to proof of work have interesting design concepts, but they must still be validated. Specifically, the execution of the selection and renewal of validator committees is a significant security concern for these solutions today.

Blockchains must also be able to scale, i.e., to have light replication of information, but in a highly dynamic environment.

There is a need for more complex cryptographic approaches in order to verify the correctness of blockchain transactions without requiring access to their sensitive data, hence protecting privacy and trade secrets.

Another difficulty is the validation of the correctness of smart contracts. Smart contracts are software programs that are performed on a blockchain. Once a smart contract has been added to a block, it cannot be altered or deleted. As with any software, a poorly coded smart contract is susceptible to computer cyberattacks. Its immutability then becomes a problem. This was the situation with the well-known The DAO, which was attacked [63] causing

the loss of a significant amount of cryptocurrency. Numerous industrial applications would also be encoded in the form of smart contracts, making the issue of smart contract security all the more crucial. Fixing smart contracts is obviously a top priority, although it is crucial to remember that the majority of cyberattacks [64] have targeted bugs or weaknesses in the execution platform.

In addition, it is necessary to address the expressiveness of the languages in which smart contracts are written. There is a significant difference between a contract written or comprehended by an attorney and a smart contract. A smart contract is a piece of code written in a programming language (be it a general-purpose one, or a specialized one like Solidity [65]), which is largely unrelated to legal ideas. In the common conception, smart contracts are viewed as a potential method for automatically enforcing a contract in the legal sense. Nevertheless, from this perspective, smart contracts are no more powerful than a simple evidence log. Further study is required to develop smart contract languages that more accurately reflect actual legal contracts.

Scaling up blockchain is also a significant challenge that needs to be overcome, particularly in relation to energy usage. For Bitcoin for example, the consumption per transaction is predicted to range between 10 and 18 kWh [62]. However, exact consumption models are lacking and should be explored to provide an accurate appraisal of blockchain and its numerous incarnations.

The lack of interoperability between blockchains is another major challenge, as the blockchain industry is now fragmented over numerous incompatible platforms and protocols. This makes it extremely difficult to trade value between assets developed on different platforms, as it would require a centralized clearinghouse-operating intermediary. Also problematic is the exchange between apps produced on different platforms.

Monitoring or exploring blockchains remains an issue as well. Often referred to as explorers or scans, blockchain exploration tools (such as Etherscan and Ethereum blockchain explorer for Ethereum, Bitcoin Blockchain Explorer and Block explorer for BitCoin) typically display the blocks generated, the transactions within a block, and the miner who generated the block. However, these are still extremely basic tools. Exploration tools will be required to progress into far more advanced tools, similar to web search engines, with genuine monitoring or auditing capabilities. A good tool that satisfies the specified security standard will be the entry point to the underlying blockchain and will have a significant impact on its adoption. These mining tools will be the true guarantors of the auditability and transparency of blockchains.

The lack of proper software engineering tools at all stages of the design and development cycle of blockchains is a final but very significant problem for blockchain technology. At the design stage, formal methods and tools for the verification and certification of chains, as well as methods and tools for architecture and modularity, are required. The development stage needs methods and tools for multi-level simulation, testing, and benchmarking.

EU research is active in these fields, but it is dwarfed by e.g. American and Israeli competition. Major EU investments must thus be made to keep in the race for the future of blockchains.

Sovereignty depends on trustable and auditable hardware and software

A castle can have the deepest moat and the highest and strongest walls, but they are of no use if an adversary has the key to the secret tunnel or hidden postern. Similarly, if we don't control the parts of our IT systems that are crucial for cybersecurity, we can hardly guarantee it. If backdoors exist in the operating system or even in the hardware we buy, unknown to us, they can be exploited by attackers and it is very difficult to add extra elements of security to counter them. The same is true for development tools, like the compilers that generate the actual executables, or the software libraries used as building blocks to compose programs: being closed source and distributed as binaries, they could embed backdoors in the programs that are produced with them.

Currently, by basing most, if not all, of its activities on IT systems running on proprietary hardware and operating systems not made in the EU, the EU effectively entrusts the providers with the keys to its whole economy and all aspects of its (cyber)security. The situation is barely better for development tools, compilers, and libraries, in which EU production is quantitatively very limited. This is why it is crucial to retain the keys of the castle, which means to retain sovereignty and control over the important hardware and software components for at least the TCB, and hopefully the whole software stack.

The way for the EU to reclaim its IT **sovereignty** is thus to **base the TCB of EU IT systems either on open-source software and hardware, or on EU-made, trustable-because-audited, proprietary hardware or software.**

Liability and certification for IT systems

Liability and **certification** are crucial building blocks needed to mandate the consideration of non-functional security properties in IT systems [40]. Indeed, **adding the legal building block of liability**, introducing the threat of a potential penalty for non-secure systems, provides motivation for system builders and providers to commit effort, and hence money, to having secure systems. More and more voices call for such liability, including penal liability, for cybersecurity neglect [66], in similar way as liability exists for any other business (such as car manufacturers or plumbers). With liability, the extra time of specification and the extra step of verification would be worth taking.

Cybersecurity certification is the technical building block that makes the legal one of liability viable. With such certification, system builders can have their efforts for security quantified, priced and legally acknowledged; purchasers can mandate security based on an independent assessment; and regulatory bodies can outlaw low-security systems. Certification can be based on test suites that must be passed, on verification tools, on development practices that must be adhered to, etc. Cybersecurity certification schemes already exist in the world, like the CMMC (Cybersecurity Maturity Model Certification) in the USA [67] or the Cyber Essentials in the UK [68], and of course the revised EU directive on security of Network and Information Systems (NIS 2 Directive [69]).

With strong cybersecurity liability and certification, companies have the incentive to create secure software, or to keep finding, and mitigating or fixing vulnerabilities. Cybersecurity liability and certification must thus be put in force and reinforced as quickly as possible, at EU and national levels.

Education is a keystone for cybersecurity

Of course, there are not only the technical aspects to consider; education is involved as well in securing society.

First of all, we need to train more cybersecurity specialists. These are needed for researching and developing all kinds of defences, such as those mentioned in this article. As security is a kind of never-ending cat-and-mouse game between attackers and defenders, these cybersecurity specialists are also needed for continuing research into defining systems against the new and as yet unknown attacks malicious actors will come up with in the future.

Secondly, while not all developers need to become cybersecurity specialists, all developers should at least have some basic knowledge and understanding of cybersecurity. This is because while the cybersecurity specialists will develop new techniques and approaches to secure systems, these technologies still need to be integrated in and applied to systems. Application developers should furthermore be aware of the appropriate techniques to create secure systems by design, know how to apply these, and also know which techniques and, equally important, know which coding practices should be avoided when creating secure systems. This basic knowledge of cybersecurity will also enable better interaction between developers and cybersecurity specialists.

Finally, the public at large also should at least have some basic awareness of cybersecurity. Although this should not be necessary in an ideal world (where all systems are secure, and under no circumstances can these systems be abused or circumvented), we unfortunately do not yet live in this ideal world. As such, it is important that the general public is taught some basic cybersecurity concepts.

All this implies that we should invest in cybersecurity education, ranging from educating cybersecurity specialists, through giving all software developers an education into the basics of cybersecurity, to instilling an awareness in the public at large.

Conclusion

(Cyber)security is only as strong as its weakest link, which means that the level of security of an IT system is at the minimum of the level of security of its components. But "the physics of cyberspace are wholly different" to physical work, because of the **worldwide interconnection and instantaneousness of IT networks**, where an attack on a vulnerability in some obscure accounting software in Ukraine can spread to and **paralyse whole worldwide crucial supply chains**. The weakest link thus has to be seen on a worldwide basis.

For way too long security has been forgotten in the interests of better time-to-market and lower prices. However, increased awareness of the **cost of poor security**, coupled with the flurry of attacks in recent years, further increased during the COVID-19 pandemic, in a much more organized way than before, is now making it both *necessary and possible* to reverse this trend, and to take a path towards better security and resilience in IT systems, providing better resilience for economies, vital supply chains and people. Even when full protection cannot be achieved, simply increasing the cost for attackers to perform a cyberattack already has a positive effect, since cybercrime is a business that looks for return on investment, hence steers away from heavily defended targets.

The technical building blocks to this end are within sight in the research community and must be nurtured and pushed forward, so as to quickly mature and then irrigate the whole IT industry. Europe has a key role to play in terms of **IT systems that are more valuable because of their higher quality thanks to higher security and resilience**, by promoting **targeted research**, taking the appropriate **regulatory steps**, and taking the necessary steps to reclaim **sovereignty of the critical building blocks** of its own IT systems.

References

[1] TCB: Trusted computing base: https://en.wikipedia.org/wiki/Trusted_computing_base

[2] ISO/IEC TR 24772 Programming languages – Guidance to avoiding vulnerabilities in programming languages: https://committee.iso.org/sites/isoorg/contents/data/committee/04/52/45202/x/catalogue/p/1/u/1/w/0/d /0

[3] Frama-C: https://frama-c.com/features.html

[4] Ada SPARK's Discovery toolset: https://www.adacore.com/sparkpro

[5] Coq proof assistant: https://en.wikipedia.org/wiki/Coq

[6] CompCert compiler: http://compcert.inria.fr

[7] seL4 microkernel: https://sel4.systems

[8] Petya ransomware attacks: https://en.wikipedia.org/wiki/Petya_(malware)

[9] Wannacry ransomware attacks: https://en.wikipedia.org/wiki/WannaCry_ransomware_attack

[10] "Wargames" movie: https://www.imdb.com/title/tt0086567/

[11] INTERPOL report shows alarming rate of cyber attacks during COVID-19: https://www.interpol.int/News-and-Events/News/2020/INTERPOL-report-shows-alarming-rate-of-cyberattacks-during-COVID-19

[12] Precise and Scalable Detection of Double-Fetch Bugs in OS Kernels. Meng Xu, Chenxiong Qian, Kangjie Lu, Michael Backes, Taesoo Kim. IEEE Symposium on Security and Privacy, 2018

[13] Check It Again: Detecting Lacking-Recheck Bugs in OS Kernels. Wenwen Wang, Kangjie Lu, Pen-Chung Yew. ACM Conference on Computer and Communications Security, 2018.

[14] Control-flow integrity principles, implementations, and applications. Martín Abadi, Mihai Budiu, Úlfar Erlingsson, Jay Ligatti. ACM Trans. Inf. Syst. Secur. 2009

[15] Enhanced Operating System Security Through Efficient and Fine-grained Address Space Randomization. Cristiano Giuffrida, Anton Kuijsten, Andrew S. Tanenbaum. In USENIX Security Symposium, 2012

[16] SoK: Automated Software Diversity. Per Larsen, Andrei Homescu, Stefan Brunthaler, Michael Franz. IEEE Symposium on Security and Privacy, 2014

[17] Librando: transparent code randomization for just-in-time compilers. Andrei Homescu, Stefan Brunthaler, Per Larsen, Michael Franz In ACM Conference on Computer and Communications Security, 2013

[18] N-Variant Systems: A Secretless Framework for Security through Diversity. Benjamin Cox, David Evans. USENIX Security Symposium, 2006

[19] Cloning Your Gadgets: Complete ROP Attack Immunity with Multi-Variant Execution. Stijn Volckaert, Bart Coppens, Bjorn De Sutter. IEEE Trans. Dependable Secur. Comput. 2016

[20] Automatic Patch-Based Exploit Generation is Possible: Techniques and Implications. David Brumley, Pongsin Poosankam, Dawn Xiaodong Song, Jiang Zheng. IEEE Symposium on Security and Privacy. 2008

[21] Feedback-driven binary code diversification. Bart Coppens, Bjorn De Sutter, Jonas Maebe. ACM Trans. Archit. Code Optim. 2013

[22] The Jury Is In: Monolithic OS Design Is Flawed: Microkernel-based Designs Improve Security. Simon Biggs, Damon Lee, Gernot Heiser: APSys 2018: 16:1-16:7

[23] Olvid. https://olvid.io/technology/en/

[24] Signal. https://signal.org/docs/

[25] 81 Ransomware Statistics, Data, Trends and Facts for 2021. https://www.varonis.com/blog/ransomware-statistics-2021

[26] Ransomware As A Service (RaaS) explained. https://www.crowdstrike.com/cybersecurity-101/ransomware/ransomware-as-a-service-raas /

[27] Financial institutions; 90% of them have been targeted by ransomware. https://www.prdistribution.com/news/financial-institutions-90-of-them-have-been-targeted-by-ransomware/3279096
[28] 20 Ransomware Statistics You're Powerless to Resist Reading. https://www.thesslstore.com/blog/ransomware-statistics/
[29] Ransomware: the true cost to business. https://www.cybereason.com/hubfs/dam/collateral/ebooks/Cybereason_Ransomware_Research_2021.pdf
[30] Coveware Quarterly Ransomware Report Q4 2020. https://www.coveware.com/blog/ransomware-marketplace-report-q4-2020
[31] The State of Ransomware in the US: Report and Statistics 2019. https://blog.emsisoft.com/en/34822/the-state-of-ransomware-in-the-us-report-and-statistics-2019/
[32] More and More Companies Are Getting Hit with Ransomware. https://heimdalsecurity.com/blog/companies-affected-by-ransomware/
[33] Ransomware Trends 2021. https://www.hhs.gov/sites/default/files/ransomware-trends-2021.pdf
[34] September 2020 Healthcare Data Breach Report: 9.7 Million Records Compromised. https://www.hipaajournal.com/september-2020-healthcare-data-breach-report-9-7-million-records-compromised/
[35] Ransomware attacks on US healthcare organizations cost $20.8bn in 2020. https://www.comparitech.com/blog/information-security/ransomware-attacks-hospitals-data/
[36] Health Service Executive ransomware attack. https://en.wikipedia.org/wiki/Health_Service_Executive_ransomware_attack
[37] Java Logging Package RCE Vulnerability. https://media.cert.europa.eu/static/SecurityAdvisories/2021/CERT-EU-SA2021-067.pdf
[38] Log4j vulnerability - update from the CSIRTs Network. https://www.enisa.europa.eu/news/enisa-news/log4j-vulnerability-update-from-the-csirts-network
[39] 2020 United States federal government data breach. https://en.wikipedia.org/wiki/2020_United_States_federal_government_data_breach
[40] HiPEAC interview: Ingrid Verbauwhede on security by design, the internet of things and why European leadership is needed. https://youtu.be/2ZBfpwV2nx8
[41] The hybrid war in Ukraine https://blogs.microsoft.com/on-the-issues/2022/04/27/hybrid-war-ukraine-russia-cyberattacks/
[42] EU Action plan on military mobility 2.0.November 2022. https://defence-industry-space.ec.europa.eu/action-plan-military-mobility-20_en
[43] European Defence Agency (EDA) https://eda.europa.eu/
[44] Countering cognitive warfare: awareness and resilience. Cao et al. in NATO Review, 20 May 2021. https://www.nato.int/docu/review/articles/2021/05/20/countering-cognitive-warfare-awareness-and-resilience/index.html
[45] Hackers just found serious vulnerabilities in a U.S. military fighter jet. Joseph Marks, in The Cybersecurity 202 Newsletter. 14 August 2019. https://www.washingtonpost.com/news/powerpost/paloma/the-cybersecurity-202/2019/08/14/the-cybersecurity-202-hackers-just-found-serious-vulnerabilities-in-a-u-s-military-fighter-jet/5d53111988e0fa79e5481f68/
[46] European Parliament website hit by cyber attack. Le Monde. 23 November. 2022. https://www.lemonde.fr/en/europe/article/2022/11/23/european-parliament-website-hit-by-cyber-attack_6005369_143.html
[47] Inside a US military cyber team's defence of Ukraine. Gordon Corera. BBC. 30 October 2022. https://www.bbc.com/news/uk-63328398
[48] Aurora Generator Test. Wikipedia. https://en.wikipedia.org/wiki/Aurora_Generator_Test
[49] Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid. Wired. 3 March 2016. https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/
[50] Ukraine says it thwarted Russian cyberattack on electricity grid. James Pearson. Reuters. 12 April 2022. https://www.reuters.com/world/europe/russian-hackers-tried-sabotage-ukrainian-power-grid-officials-researchers-2022-04-12/
[51] Russian hackers targeting Dutch gas terminal: report. 25 The NL Times. 25 November 2022. https://nltimes.nl/2022/11/25/russian-hackers-targeting-dutch-gas-terminal-report
[52] Global surveillance disclosures (2013–present). Wikipedia. https://en.wikipedia.org/wiki/Global_surveillance_disclosures_(2013%E2%80%93present)
[53] What is WannaCry ransomware? Kasperky. https://www.kaspersky.com/resource-center/threats/ransomware-wannacry
[54] The Untold Story of NotPetya, the Most Devastating Cyberattack in History. Andy Greenberg. September 2018, Wired. https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/
[55] The Hidden Costs of Cybercrime. McAfee. December 2020. https://www.mcafee.com/enterprise/en-us/assets/reports/rp-hidden-costs-of-cybercrime.pdf
[56] Actors behind PyPI supply chain attack have been active since late 2021. Dan Goodin. Ars Technica. September 1, 2022. https://arstechnica.com/information-technology/2022/09/actors-behind-pypi-supply-chain-attack-have-been-active-since-late-2021/
[57] Sabotage: Code added to popular NPM package wiped files in Russia and Belarus. Dan Goodin. Ars Technica. March 18, 2022. https://arstechnica.com/information-technology/2022/03/sabotage-code-added-to-popular-npm-package-wiped-files-in-russia-and-belarus/
[58] The SolarWinds Cyber-Attack: What You Need to Know. Center for Internet Security. March 2021. https://www.cisecurity.org/solarwinds
[59] What Is Blockchain? Gartner. March 2022. https://www.gartner.co.uk/en/articles/what-is-blockchain
[60] Gartner blockchain hype cycle 2021: where we are & what's next. Godfrey Benjamin. iMi Blockchain. June 2021. https://imiblockchain.com/gartner-blockchain-hype-cycle/
[61] Gartner Hype Cycle for Blockchain and Web3, 2022. Avivah Litan. Gartner. July 2022. https://blogs.gartner.com/avivah-litan/2022/07/22/gartner-hype-cycle-for-blockchain-and-web3-2022/
[62] Les verrous technologiques des blockchains. CEA, IMT, INRIA. April 2021. https://www.entreprises.gouv.fr/files/files/etudes-et-statistiques/rapport-final-blockchain.pdf
[63] What Was The DAO? Cryptopedia. March 2022. https://www.gemini.com/cryptopedia/the-dao-hack-makerdao
[64] N. Atzei, M. Bartoletti, and T. Cimoli. A survey of attacks on ethereum smart contracts sok. In Conference on Principles of Security and Trust 2017.
[65] Solidity language. https://soliditylang.org/
[66] Introduction to track "Supply chain cybersecurity" of the European Cyber Week. Admiral COUSTILLIERE. November 2022. https://www.european-cyber-week.eu/programme?lang=en
[67] Cybersecurity Maturity Model Certification. https://dodcio.defense.gov/CMMC/
[68] Cyber Essentials. https://www.ncsc.gov.uk/cyberessentials/overview
[69] The NIS 2 Directive. https://www.nis-2-directive.com/

Olivier Zendra is a tenured computer science researcher at Inria, Rennes, France.

Bart Coppens is a part-time assistant professor and a post-doctoral researcher in the electronics department of Ghent University, Ghent, Belgium.

This document is part of the HiPEAC Vision available at hipeac.net/vision.

This is release v.3, January 2023. Previous versions were published under the name "Cybersecurity must come to IT systems now".

Cite as: O. Zendra and B.Coppens. From cybercrime to cyberwarfare, nobody can overlook cybersecurity any more. In M. Duranton et al., editors, HiPEAC Vision 2023, pages 130-144, Jan 2023.

DOI: 10.5281/zenodo.7461910

The HiPEAC project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 871174.