Manifesto: Let's Remove the Foundation to Wage Cyberwar

Erland Wittkoetter, Ph.D., September 5th 2022

In today's world, and more so in today's IT/cybersecurity ecosystem, the goal of removing the capacity to wage cyberwar sounds ridiculous - impossible. Though, technically, it is feasible. Why? No known law (of nature) tells us this is impossible. There are memes like: "there is no perfect security", but let us agree that memes have no evidential weight, and second, we do not need to be perfect. The goal of ending cyberwar sounds difficult to achieve, but actually, it is not.

This proposal is written because **ending cyberwar is achievable**. Not in 20 years, 10 years, or even in 5 years; it could be done much faster via **software-based** (relatively simple) **retrofits within a few months**. A **grassroots community of open-source developers** can do that. Then, a semi-soft/hardware retrofit follows, e.g., a USB security stick rolled out over the next few years would solidify this progress until hardware security components are part of newly sold devices. Ultimately, we need automated security independent of any detrimental human involvement. Cybersecurity defines what we need in concrete situations to be safe, and the rest should come from reliable toolboxes. Does that sound still impossible? Reminder: Do we still have rootkits (digital ghosts …) in our IT ecosystem? Now, they are irrelevant, almost gone. What can we learn from that?

More than cyberwar, being a victim of cybercrime is on many more people's minds. Cybercrime is responsible for Trillions (Dollars) of damages annually – with a rising trend. **Getting rid of cybercrime is a worthy call to action.** However, many forms of cybercrime are based on deception and dishonesty and not on flaws within the technical implementation of software or hardware products. Therefore, cybercrime like love-scams, crypto-, or money laundry scams should be fought on the monetary front, i.e., when money is exchanged, and not on the technical level via features provided to our IT devices. **Cybersecurity should guide/help/protect people with their inherent (often unavoidable) vulnerabilities** and how they can be protected from being scammed. Without going on a tangent, technology could do more against phishing. Solutions should warn people before becoming victims.

Cybersecurity or cyberdefense deals with data spying, sabotage, device/system misuse, and disinformation. As much as we hate fake news or propaganda, detecting or preventing disinformation and the weaponization of social media, i.e., all these fringe activities, should better not be handled technically by default. The same applies to **deepfakes**: automated authentication of audio/video streams via embedded cryptographic information and external information could make deepfakes identifiable reliably. But including mandatory anti-deepfake tools in a technology stack to detect deepfakes by default could lead to serious privacy or total surveillance issues and, therefore, have to be optional.

How about no (autonomous) drones or robotic soldiers? Unfortunately, no. But it would mean that these weapons are (likely) under full command/control of responsible/accountable humans.

Using software products requires trust in developers and manufacturers. Distrust or fear of malicious software consequences is often mitigated via independent audits or certifications. The quality of hardware products is much easier to determine than software security claims. Moreover, extensive code review is soon useless when supersmart adversaries use reverse code engineering on compiled software. These adversaries, likely using AI, could steal crypto-keys or misuse crypto-device; all these activities could be undetectable and untraceable. Software operates in a non-trustworthy environment.

After making these clarifications, we limit our **proposed solution scope** to detect/prevent software modifications and misuse of software **with a focus on stopping cyber weapons: malware, ransomware (no data sabotaging), spyware, and hidden backdoors**. The declared goal is to eliminate damage from all types of malware. Our adversaries are nation-states that soon use AI with scary hacker skills. Under these (artificially) aggregated conditions, the end of waging cyberwars is our goal.

 info@nogostar.com

For our goal, it is **essential to define** what we mean by malware, spyware, backdoors, and ransomware as representative of data sabotaging. **Malware** is software developed by unfriendly actors (cyber-criminals, etc.) to steal or maliciously manipulate data, damage or destroy the device's normal operation or utilize the device without permission against the expected intention or benevolent assistance of its user or owner. **Ransomware** or data sabotaging is malware focusing on damage to user-generated content. We usually consider spyware as software that tries to get user data covertly from devices to remote locations. Backdoors are features that offer attackers covert and secretive interfaces to operate against users' intentions or benefits. But these definitions are related to events that could be covered up; they are therefore not operational; instead, we define spyware and backdoors around covertness: Every software that sends out (not-required) information covertly is **spyware**. Software receiving covertly (not-required) data, which it uses, has **backdoors**. Both definitions are better suited than the damage-related ones, and manufacturers have a simple way out: be transparent/detectable – avoid covertness.

We assume that no customer or user would use or want to use malware, ransomware, spyware, or software with backdoors intentionally or voluntarily. If companies are caught providing such software features, they are publicly shamed; these indiscretions should not be quickly forgotten. **Threats to (long-term) reputation** are quite effective. In our world, where we depend on software so critically, being involved covertly with malware, spyware, or backdoors should be a business or career-ending event like losing a law- or medical license. We could have much fewer problems with malware if manufacturers and software developers were tracked and treated like other professionals: medical doctors, lawyers, financial advisers, or journalists. The mentioned professions have written or unwritten rules, guaranteeing a minimum level of quality control and legal compliance in their contributions to protect the public via tough self-regulating instruments. Having flaws in software, even vulnerabilities, is normal. These events can be ignored, but if developers use exploits nefariously or covertly for an illegitimate advantage, these actions should ruin bad actors' reputations. But if manufacturers help us detect software misuse, we reward them with positive ratings.

What is left from cryptography if we assume attackers **steal crypto-keys** or **utilize crypto devices covertly**? The answer is very little (*nothing* of practical use).

Also, is it fair to say that the current state of cybersecurity, with blacklisting threats, blocking firewall ports, etc., is insufficient to eliminate damage from malware? New zero-day vulnerabilities are currently blindspots; it is impossible to predict them ahead of time or be prepared for them. Therefore, removing the basis for waging cyberwar is an unrealistic goal with current cybersecurity tools.

After reading through many books on computer security, network security, and cryptography and comparing them with sound security engineering in aviation or construction/building codes, there is an almost unbelievable steep culture gap that no one in cybersecurity dares to eliminate yet. We know from aviation or nuclear safety that a human doesn't need to be hurt before professionals study possible problems. Whenever we got sloppy and careless, someone got hurt. NASA paid with, e.g., the loss of Challenger in 1986 and 2003 with Columbia. Passengers died on Boinig 737Max due to flaws in MCAS systems. It seems it's human nature to get complacent. Despite zero-tolerance, hundreds of aviation issues annually are still a testimonial to the consistency of human failures. Translated to cybersecurity, our infrastructure remains vulnerable to malware if we don't fully automate cybersecurity tasks. This full automation must still adapt to innovations and not stifle progress. Security must always be updateable, while inter-guarding checks are used to discover covert modifications from attackers.

Many computer and software architectural decisions were made with efficiency, cost-effectiveness, and small performance gains in their minds; security was an afterthought. These decisions are now technical debt that will cost us in dealing with security-related implications. But we can leave this in the past and do much better from now on. Please let me get a bit more technical for two paragraphs.

Just mentioning a few issues: having more features done by the CPU, including all security- and crypto-related features done by the CPU, was intentional, but now it is problematic. E.g., any crypto-key, including public keys, appearing in cleartext within the main CPU should be considered compromised and replaced immediately. Encryption/decryption must be done by separate hardware using trustworthy algorithms, i.e., code updates are inter-guarded by other independent security-related components. But currently, all units on a device depend on the OS (like a single point of failure).

Security depends on reliable CPU/OS features, but how many vulnerabilities have not been discovered yet? Therefore, no app managed by OS, RAM, and CPU is safe. Blacklisting bad apples is not enough. Instead, every unknown code is suspicious. At the bare minimum, we should allow only known (white- or gray-listed) software in RAM, whereby gray-listed software is statistically inferred harmless but not registered via manufacturers as whitelisted software. Regular checking hashcodes of executables for code modification should be required before apps, scripts, or code libraries are loaded into RAM. As a direct consequence of whitelisting, undetected zero-day vulnerabilities cannot be exploited covertly. Software vulnerabilities become insignificant and, with additional info from manufacturers, easier to detect. On the other side, exploiting them covertly (and maliciously) could come with a heavy price tag for the reputation of registered developers/manufacturers daring to do that.

Our technical civilization depends on reliable software. Malware, spyware, backdoors, or ransomware is poisoning the trust in technology. The trustworthiness of the new security software will come from the trust in strict/isolated task execution and the integrity of its incorruptible execution and update process. All security-related soft and hardware must be open-sourced and under the scrutiny of never-ending audits. Software in security components is inspectable but not modifiable by humans.

The proposed project wants to attract the best software minds and specialists in diverse aspects of relevant solutions. Let's envision software being created by people who know what to do because a few changes to what they already have done are all it takes from them. Others may know they should provide their experience and support to enthusiastic new contributors. People who know their experience belongs in this project should join, i.e., people who know that they are the best choice.

This manifesto is **a call to form a grassroots community of engineers** that should drive a Manhattan-type anti-war effort project to a successful and rapid conclusion. There is urgency. We see technology evolving exponentially in its capabilities. Announcements that would be celebrated as AI breakthroughs a few years ago have become a (monthly) normal. We must have protection before AI is used in malware. This terrifying thought should give us the push to develop anti-cyberwar technology for free, unrestricted, widespread deployment. For the sake of our common technical progress, we must not fail.

More info: https://NoGoStar.com