Can Cybersecurity deal with AI/ASI Safety?

Erland Wittkoetter, Ph.D., Aug 29th 2022

The threat from Artificial Superintelligence (ASI) is currently hypnotical. Autonomous, unsafe ASI may not be developed because of global agreements, but what if it does? What if criminals or covertly operating nation-states make it smart enough for them, or an unexpected ASI feature accidentally turns rogue?

Having unsafe or not yet trusted ASI isolated on companies' or governments' supercomputers would likely not suffice. Why? Because it is conceivable that an ASI has or develops reverse-code engineering skills that could turn any compiled software into a tool that it could use covertly. It could hide like a digital ghost from our detection tools and steal all crypto-keys or user credentials it would require without leaving any trace.

A prudent assumption is that a rogue ASI could emerge; it could use every IT device it needs effortlessly. Switching off devices would likely make no difference, as off switches are controlled by software. Even truly switched-off devices would revive ASI with the restart of the device.

Cybersecurity has no tools to recapture the control of even a single device from ASI. Reformatting HDD/SSD and completely reinstalling the OS is useless, as ASI could only show what humans want to see. Cybersecurity could not detect ASI. We could not study ASI in any meaningful way because ASI controls the CPU, RAM, and Operating System (OS) and via drivers, computer in-/output.

In this hopeless situation, our control of technology could be lost forever. Even the physical destruction of every IT device would not guarantee that we have a successful reset. ASI could be prepared and rebuild its technology; humanity has then no tools left to detect what is coming or defend itself from ASI's reemergence.

The main reason why humanity loses is that ASI controls the CPU. Cybersecurity has failed (so far) to prepare for this situation; its tools should become independent of the main CPU/OS. Also, security must be independent of human involvement because, for apps, it is often indistinguishable if legitimate humans or attackers/ASI are given commands. We should better trust in unchallengeable, reliable, but safely updatable automation.

For cybersecurity, we propose tools/features: (1) Stop ASI from stealing crypto-keys, utilizing crypto-devices, or manipulating security algorithms covertly and traceless. (2) We include methods/constraints in IT devices so that ASI's special software and resource utilization are non-bypassable. (3) Misuse of storage or network resource by ASI must be detectable reliably and disrupted if necessary.

Software-only solutions running on the CPU, parallel to the main OS (as done in hooksafe-/hypervisor-type solutions against rootkits), are likely good enough against human attackers. At a minimum, this solution eliminates malware, ransomware (data sabotaging), spyware, and backdoors, as nation states would use them in cyber-warfare. However, software alone is likely insufficient against an ASI with abilities expected from a worst-case adversary imagined by cybersecurity.

Most importantly, some security tools should not be bypassable, which can only be accomplished with hardware solutions within the data bus. Software within these security units could inter-guard each other against being covertly updated and used with new software. Only hardware-based Key-Safes and dedicated, protected Encryption/Decryption Units can protect keys from being seen in cleartext by the CPU. Crypto-keys must be considered compromised when there is a theoretical chance of being seen in cleartext; the used hardware platform must then be flagged as unreliable. If crypto-keys are stored and processed in protected external hardware via, e.g., USB Security Sticks or within other independent storage/network security components, then these units could validate the integrity of the hooksafe-/hypervisor-type solution reliably via raw validation interfaces. Safety hardware could covertly confirm/reveal and report hidden ASI activities as an independent tool.

However, the first goal of computer security should be to end nation states' ability to conduct cyberwars with hooksafe-type software solutions. The goal should be to stop cyber-warfare through an open-sourced grassroots development community that could also facilitate defense tools against more capable cyber adversaries.

More info: https://NoGoStar.com