No-Go-FAQ – August 31st, 2022

We love to ask and answer difficult questions. Our question to you: Do you have more?

Also, no BS is our core value. We will tell you if we get a question that we can't answer (yet). We would consider it a gift, potentially even a riddle that we should solve together, so an opportunity to grow. Also, we promise to include (legitimate) criticism in (even existing) answers.

A. About No-Go-*™ as a Project

Intro, Goals

- **Q A1: What do you want to accomplish with No-Go-*™?**
  - *Short A:* Open source developer/tech community to end cyberwar and damage from malware.
    Longer A: ...
- **Q A2: Who is the founder of No-Go-*™?**
  - *Short A:* Erland Wittkoetter, Ph.D., a tech enthusiast and optimist about humanity's future
    Longer A: ...
- **Q A3: What makes No-Go-*™ unique?**
  - *Short A:* New paradigms in cybersecurity; assumes worse adversaries than we currently have
    Longer A: ...
- **Q A4: Why do you call the project No-Go-*™?**
  - *Short A:* No-Go means "stop", and * is a placeholder
    Longer A: ...
- **Q A5: What are your short and long-term development goals?**
  - *Short A:* Initially – ending cyberwar; long-term: security guardrails against AI/ASI misuse
    Longer A: ...
- **Q A6: Why is No-Go-*™ a grassroots community project?**
  - *Short A:* It's an efficient strategy to get required tech progress in cybersecurity fast
    Longer A: ...
- **Q A7: Is ending cyberwar and making cyber weapons ineffective even achievable?**
  - *Short A:* Yes, and yes
    Longer A: ...
- **Q A8: Could you stop cybercrime?**
  - *Short A:* Yes, but not comprehensively – it will adapt to new (less-tech) vulnerabilities
    Longer A: ...
- **Q A9: Do you stop deep-fakes?**
  - *Short A:* Not by default, only as an option
    Longer A: ...
- **Q A10: Could cybercrime get worse?**
  - *Short A:* Yes, much worse – think of e-Commerce, vulnerabilities of logistics
    Longer A: ...

Anticipated Results

- **Q A11: Could your (anticipated) success be only a short-lived victory?**
  - *Short A:* Yes, it could; progress needs to be solidified via follow-up solutions
    Longer A: ...
- **Q A12: Why are we vulnerable in a cyberwar, and how can this be changed?**
  - *Short A:* We are currently unprotected against every (undetected) vulnerability of every app
    Longer A: ...

- **Q A13: Will No-Go-*™ end the use of drones in war?**
  - *Short A:* No
    Longer A: ...
- **Q A14: Will your security approach work?**
  - *Short A:* Yes; because fundamentals principles could help defender
    Longer A: ...
- **Q A15: Can No-Go-*™ remove all nations' abilities to wage cyberwar?**
  - *Short A:* No. Propaganda, disinformation, or weaponization of social media remains
    Longer A: …
- **Q A16: Do you anticipate the project could fail in delivering on its promise?**
  - *Short A:* Not with a global dev-/tech-community supporting it
    Longer A: ...
- **Q A17: When can you deliver results/products?**
  - *Short A:* This depends on many factors; we prefer to delay an answer to that
    Longer A: ...
- **Q A18: Could there be other solutions to end cyberwar or cybercrime?**
  - *Short A:* Yes
    Longer A: ...

Advanced Adversaries

- **Q A19: Are there worse adversaries than nation states developing malware?**
  - *Short A:* Yes – malicious Artificial Superintelligence (ASI) developed by criminals
    Longer A: ...
- **Q A20: Can No-Go-*™ prevent future threats from Artificial Superintelligence (ASI)?**
  - *Short A:* Yes – but we must prepare ourselves for unsafe ASI
    Longer A: ...
- **Q A21: How dangerous could ASI become?**
  - *Short A:* Unknowable – but how bad could a criminal utilization of ASI by dictators be?
    Longer A: ...
- **Q A22: Can we control ASI or protect humanity from malicious ASI?**
  - *Short A:* Tentatively, yes, but we have no protection without (serious) preparation
    Longer A: ...
- **Q A23: How could you know that No-Go-Security™ is also effective against ASI?**
  - *Short A:* With proactive and preventative security, we have a good chance of achieving that
    Longer A: …
- **Q A24: Could ASI still be dangerous despite No-Go-Security™ and guardrails?**
  - *Short A:* Yes. Nukes remain nukes; it doesn't matter if they are being safely handled
    Longer A: …
- **Q A25: Do you expect we could regulate progress toward ASI?**
  - *Short A:* No – too late and ineffective
    Longer A: ...
- **Q A26: Are we too late in preparing for threats from ASI?**
  - *Short A:* Not yet (hopefully) – but we are (likely) close to "too late"
    Longer A: ...
- **Q A27: What could we do if ASI is already in our IT ecosystem?**
  - *Short A:* We need to start from the assumption that we are too late anyway
    Longer A: …
- **Q A28: Why do you "believe" ASI could become a loyal companion?**
  - *Short A:* ASI should serve us all – we need to make it safe for that
    Longer A: ...

Business-related

- **Q A29: Will No-Go-*™ operate as a business?**
  - *Short A:* As a non-profit organization
    Longer A: ...
- **Q A30: How is or will No-Go-*™ initially be funded?**
  - *Short A:* Donations, grants
    Longer A: ...
- **Q A31: What kind of help does No-Go-*™ need?**
  - *Short A:* We always look for expertise and money; details will be listed in → "Next Steps"
    Longer A: ...
- **Q A32: How will you spend money received by the community?**
  - *Short A:* Development
    Longer A: ...
- **Q A33: Will No-Go-*™ file for 501(c) 3 tax-exempt status?**
  - *Short A:* To be decided later, but likely
    Longer A: ...
- **Q A34: What will be licensed?**
  - *Short A:* Yes; 3rd parties selling products (not services) that contain No-Go-*™ (community) IP
    Longer A: ...
- **Q A35: Why is No-Go-*™ seeking trademarks?**
  - *Short A:* With Trademarks, we can enhance trust in No-Go implementations
    Longer A: ...
- **Q A36: How does No-Go-*™ pay for its online services?**
  - *Short A:* Multiple options, but no decision made
    Longer A: ...
- **Q A37: What is the business Vision of No-Go-*™?**
  - *Short A:* As a resource hub; we teach advanced No-Go/cybersecurity tech
    Longer A: ...
- **Q A38: Does No-Go-*™ want to hire software developers?**
  - *Short A:* Yes, and we want the private sector finds dev-/tech-talents easily
    Longer A: ...
- **Q A39: Will No-Go-*™ for-profit spin-offs?**
  - *Short A:* Our mission and promise comes first
    Longer A: ...
- **Q A40: Who do you want to attract into the No-Go-Community™?**
  - *Short A:* Devs/techs who feel "called" to be part of it – who see their opportunity to …
    Longer A: ...
- **Q A41: Who should care about this project?**
  - *Short A:* We want to be open and surprised …
    Longer A: ...

B. About No-Go-*™ Anticipated Deliverables

Products

- **Q B1: What are the essential components that you want to develop?**
  - *Short A:* Facilitating proactive/preventative security technology and educating fellow engineers
    Longer A: ...
- **Q B2: How do you know you can stop malware?**
  - *Short A:* Restricted access to CPU for known apps only (via cached, whitelisted hashcodes)
    Longer A: ...

- **Q B3: How do you know you can stop ransomware or data sabotaging?**
  - *Short A:* No malware – and rapid recovery features in case of damage
    Longer A: ...
- **Q B4: How do you know you can stop spyware?**
  - *Short A:* No malware – more transparency over exchanged data
    Longer A: ...
- **Q B5: How do you know you can stop backdoors?**
  - *Short A:* No malware – better detection of anomalies and deterrence (for rule breakers)
    Longer A: ...
- **Q B6: Are there limitations to anticipated deliverables?**
  - *Short A:* Initially, yes; but they can be fixed when security is solidified
    Longer A: ...

Complementary

- **Q B7: What happens to firewalls when No-Go-*™ watchdogs are accepted?**
  - *Short A:* Firewalls (i.e., security-conscious routers) remain important
    Longer A: ...
- **Q B8: Are antivirus solutions required when No-Go-*™ solutions are available?**
  - *Short A:* Cybersecurity is much more than Anti-Virus – these solutions remain important
    Longer A: ...
- **Q B9: Are backups required when data sabotaging is stopped?**
  - *Short A:* Not required for security, but hardware failures
    Longer A: ...

Features

- **Q B10: Is No-Go-*™ limiting existing computer or network capabilities?**
  - *Short A:* No
    Longer A: …
- **Q B11: Why is No-Go-Security™ a proactive, preventative solution?**
  - *Short A:* Security threats should not get close to CPUs – redundancy if whitelisting has failed
    Longer A: ...
- **Q B12: Can No-Go-*™ deliver perfect security?**
  - *Short A:* No, but near-perfect based on protection, prevention, and auto-detection of failures
    Longer A: ...
- **Q B13: What threats can No-Go-Security™ adapt to automatically?**
  - *Short A:* Security is based on auto-adapting, closed feedback loops
    Longer A: ...
- **Q B14: Has No-Go-Security™ blindspots, i.e., does it fail to detect malicious actions?**
  - *Short A:* Yes – but we help to detect them (late) and deter from exploiting users with them
    Longer A: ...
- **Q B15: Could No-Go-Security™ adapt to entirely new threats?**
  - *Short A:* Yes, but we should hold our horses
    Longer A: ...
- **Q B16: Can No-Go-Security™ solutions be updated?**
  - *Short A:* Yes; and updates cannot be exploited in or for attacks
    Longer A: ...
- **Q B17: How fast can No-Go-Security™ adapt to new threats?**
  - *Short A:* No-Go is proactive – users are protected from damage; no worries about threats
    Longer A: ...

- **Q B18: What if software (already) modifies its own software code?**
  - *Short A:* We need a different type of proactive security around self-modifiable software
    Longer A: …
- **Q B19: Does No-Go-Security™ has single-point-of-failures?**
  - *Short A:* chosen architecture is self-adapting, self-healing and fault tolerant
    Longer A: …
- **Q B20: Why should we use RISC, not CISC, for security?**
  - *Short A:* We must be able to detect malware in hardware
    Longer A: …
- **Q B21: Is No-Go-Security™ incompatible with any hardware?**
  - *Short A:* No-Go-* has a no-device-left-behind policy; problems are too early to predict
    Longer A: …
- **Q B22: Is No-Go-Security™ incompatible with any software?**
  - *Short A:* No; but some software should/must be adapted to No-Go-Security™
    Longer A: …
- **Q B23: Why do you require Software Developers/Manufacturers to register?**
  - *Short A:* Software is critical; other sectors (medical, financial) already have self-regulation
    Longer A: …
- **Q B24: Are you expecting too much (info) from Software Developers/Manufacturers?**
  - *Short A:* No. We offer developers/manufacturers to improve their reputation easily.
    Longer A: …
- **Q B25: What happens to non-registered Software, Developers, or Manufacturers?**
  - *Short A:* This is made transparent to users; they can decide.
    Longer A: …
- **Q B26: Do you expect Web-resource operators are participating voluntarily?**
  - *Short A:* No, not all – hopefully enough; but businesses will likely see an advantage.
    Longer A: …
- **Q B27: How do you check content in encrypted messages?**
  - *Short A:* We create an accepted man in the middle instance in local Network Watchdog
    Longer A: …
- **Q B28: How will No-Go deal with filesystem/stateful info managed in RAM?**
  - *Short A:* Watchdogs are transparent to all CPU/OS operations
    Longer A: …
- **Q B29: Do you check attack patterns fine-grained or coarse-grained?**
  - *Short A:* Both, when we know attack method: fine-grained; unknown methods coarse-grained
    Longer A: …

Concerns

- **Q B30: Will No-Go-Security™ slow down protected devices?**
  - *Short A:* Probably not that much. But security always has some impact
    Longer A: …
- **Q B31: Do we need No-Go-Security™ on all machines?**
  - *Short A:* No. Cyberwar can be stopped with a smaller footprint; with ASI, it's difficult
    Longer A: …
- **Q B32: Do you use or facilitate surveillance?**
  - *Short A:* Not for security; but we must support court-ordered warrants for limited surveillance
    Longer A: …
- **Q B33: Are there some goals that you are doubtful about achieving?**
  - *Short A:* … well, no device-left-behind promise is a challenge
    Longer A: …

- **Q B34: How certain are you to deliver on your full promises?**
  - *Short A:* Very certain on developing capabilities; cautiously optimistic on a broad deployment
    Longer A: ...
- **Q B35: Could No-Go-*™'s development effort be done in vain?**
  - *Short A:* Unlikely, if we believe we can deliver on promises; if not, there are still useful outcomes
    Longer A: ...
- **Q B36: How much should regular users care about No-Go-Security™?**
  - *Short A:* Users should not worry about basic security; more: being protected against cybercrime
    Longer A: ...
- **Q B37: Could No-Go-*™ educate attackers?**
  - *Short A:* Yes, that is a valid concern, but it applies to every technology
    Longer A: ...
- **Q B38: Would you share negative news about No-Go-*?**
  - *Short A:* Yes, there is no positive or negative news – it's just progress
    Longer A: ...

C. Miscellaneous Questions

Competitors

- **Q C1: Who is No-Go-*™ competing with?**
  - *Short A:* No tech or business has dared to call for an end of cyberwar (yet)
    Longer A: ...
- **Q C2: Does something similar to No-Go-Security™ already exist?**
  - *Short A:* Not to our knowledge; please get in touch with us if you own relevant IP
    Longer A: ...

Cybersecurity

- **Q C3: Is cryptography used, and is it up to the task?**
  - *Short A:* Yes, and No. Cryptography isn't doing enough against stolen or misused keys
    Longer A: …
- **Q C4: Is No-Go-Security™ secure against quantum computation?**
  - *Short A:* Yes, very likely
    Longer A: …
- **Q C5: Why is cybersecurity not better in securing users?**
  - *Short A:* Cybersecurity is complex, and it is considered essential but treated as a side-show
    Longer A: ...
- **Q C6: What makes security in other sectors so much more successful?**
  - *Short A:* Security in other sectors is more confident due to proactive toolsets
    Longer A: ...
- **Q C7: Who is more exposed in a cyberwar?**
  - *Short A:* … that is probably changing over time …
    Longer A: ...
- **Q C8: Will governments or the military push back on No-Go-*™?**
  - *Short A:* Probably not
    Longer A: ...
- **Q C9: Who may resist the changes from No-Go-*™?**
  - *Short A:* … we will see …
    Longer A: ...

- **Q C10: What is the opinion of cybersecurity professionals about No-Go-Security™?**
  - *Short A: It's new, not tested; but we got important hints (thanks)*
    Longer A: ...
- **Q C11: Why don't you care about vulnerabilities?**
  - *Short A: We care, but not s much as you should care without No-Go-Security*
    Longer A: ...

Term clarifications

- **Q C12: Is promising no damage from malware a flawed statement**
  - *Short A:* Not really. Damage is something that should be avoided/prevented
    Longer A: ...
- **Q C13: What is security in comparison to safety?**
  - *Short A:* Both terms mean freedom from harm, threat, and danger
    Longer A: ...
- **Q C14: What is the difference between white- and graylisting?**
  - *Short A:* Whitelisted are based on authorized data, graylisted on detected patterns and statistics
    Longer A: ...
- **Q C15: What are deviations or anomalies?**
  - *Short A:* They are results that were not expected from disclosures, patterns, or predictions
    Longer A: ...
- **Q C16: Why is proactive security so much better than reactive?**
  - *Short A:* Proactive measures prevent damage early
    Longer A: ...
- **Q C17: Why is prevention in security important?**
  - *Short A:* Prevention (in No-Go) expects damage and prepares us to deal with its consequences
    Longer A: ...
- **Q C18: What is the advantage of independent circuit breakers?**
  - *Short A:* Preventing damage and gaining time for additional actions
    Longer A: ...
- **Q C19: What does it mean: it is impossible?**
  - *Short A:* Impossible is a strong prediction; it is dangerous to overgeneralize impossibility
    Longer A: ...

D. Questions with No Answer Yet

- **Q D1: What is your question?**
  - *Short A:* Let us know
    Longer A: ...

Our answers are not written in stone or intended to be our final word. If you disagree or think of a better answer, please don't hesitate to contact us with the question code reference within the subject line to **faqs at nogostar dot com**.

A. About No-Go-*™ as a Project

Intro, Goals

Q A1: What do you want to accomplish with No-Go-*™?

Short A: Open source developer/tech community to end cyberwar and damage from malware.

- Longer A: No-Go-*™ is an open-source grassroots development project that wants to end cyber-warfare. We want immediately end malware, spyware, ransomware, and backdoors. We use software retrofits initially against threats from nation-states and criminals. A bit later, we provide simple retrofittable security hardware that could push back also against AI-based cyber-threats. Long-term, we prepare our security tools to provide reliable guardrails for the worst imaginable cybersecurity adversary, a hypothetical but possible malicious artificial superintelligence.

Return

Q A2: Who is the founder of No-Go-*™?

Short A: Erland Wittkoetter, Ph.D., a tech enthusiast and optimist about humanity's future

- Longer A: Erland Wittkoetter, Ph.D., is a physicist, mathematician, inventor, and entrepreneur. His main talent: looking for solutions outside the (current) box. If the laws of nature do not prevent (i.e., prohibit) us from having a technical solution for a problem, then there is likely (at least one) solution, even simple or retrofittable solutions. Finding these solutions may be difficult, but they are out there. We often need to accept new (unconventional) paradigms to see them.

Return

Q A3: What makes No-Go-*™ unique?

Short A: New paradigms in cybersecurity; assumes worse adversaries than we currently have

- Longer A: No-Go-*™ is cybersecurity based on new paradigms. The most important new ones are:
  (1) Security operations are always physically separated from regular operations – i.e., we must have separate watchdog components.
  (2) We don't accept unknown code in RAM; we don't give it a chance to be executed on the CPU – i.e., we accept only white- or gray-listed apps – blacklisting alone is not good enough.
  (3) Make software developers more trustworthy – because they are concerned about reputation, they are also motivated to give us information helping us to detect vulnerabilities proactively.
  (4) Keys must be protected physically – i.e., keys that "could" appear in cleartext (in unprotected CPUs) must be considered compromised
  (5) Security is fully automated – i.e., humans are not involved in operational decision-making and execution. Humans are security risks; they are security's high-level managers and judges.

  Additionally, we have designed solutions around the idea that security is proactive, preventative, and automatically detecting security breaches. The anticipated adversary is a super-smart AI with hacking skills based on reverse code engineering, allowing it to utilize every binary app for its attack. These adversaries could steal every key or access credential and hide as digital ghosts in IT devices undetectable. No-Go's security tools could deal with these advanced adversaries, implying that they are good enough to be used against cyber weapons and malware.

Return

Q A4: Why do you call the project No-Go-*™?

Short A: No-Go means "stop", and * is a placeholder

- Longer A: No-Go means "stop". And star/* is a known character used as a placeholder. Reason: Certainly, No-Go-*™ won't stop by (just) ending cyberwar, malware, ransomware, spyware, or backdoors. We also need some additional guardrails against unsafe AI. No-Go-*™, security, or safety is about setting reasonable limits to be free from danger, harm, and threats.

Return

Q A5: What are your short and long-term development goals?

Short A: Initially – ending cyberwar; long-term: security guardrails against AI/ASI misuse

- Longer A: Our short-term goal is to eliminate malware damage and remove the basis for using software as damaging cyber weapons. Soon technology will provide more AI-based capabilities. We need to be prepared for super-smart AI used as malware. Therefore, long-term goals provide features that could help humanity to mitigate problems from unsafe and uncontrolled AI. These long-term goals are a natural extension of short-term goals to make misused software less damaging for its users by default.

Return

Q A6: Why is No-Go-*™ a grassroots community project?

Short A: It's an efficient strategy to get required tech progress in cybersecurity fast

- Longer A: Security, i.e., freedom from danger, harm, or threats, is in the common interest. Unfortunately, cybersecurity is not yet as good as product safety or even aviation security. It is not enough to claim that software is safe or has been audited/certified. Software is threatened by changing capabilities/tools constantly. We must be able to put security under more relentless, continuous scrutiny, as known in open-source development. If flaws are detected, we must act immediately.

  Security in other technical sectors is much more confident, proactive, preventative, and often redundant. It knows it has liabilities if it fails. The required changes to our cybersecurity should not come slowly and not overly hasty; if we are (after sufficient/exhaustive testing) confident that No-Go doesn't create (unacceptable) harm, we can go rapidly into large-scale deployment. We will help developers to adapt if there are problems. No-Go's solutions could be rolled out as retrofits to most devices as a regular OS update, i.e., as a no-big-deal fix. But we should be conscious about no device should be left behind (old OS are falling out of maintenance schedules). A single company cannot expect to deliver that level of comprehensiveness and decisiveness.

  Additionally, there is urgency and good reasons for decisive actions:

  - Hundred of billions of annual damage in cybercrime;
  - Countries and their infrastructure under threat of cyberwar, and
  - (unsafe) super-smart AI tools on the verge of being developed and potentially insufficiently safeguarded

  Our concern is that we may not have years before AI is used maliciously by criminals or nations. A grassroots development project can teach many more engineers to integrate No-Go-*™ solutions into their software environments effortlessly.

  Hooksafe is a hypervisor-based solution to prevent rootkits from taking control over a device; Xen is an open-source hypervisor solution; these projects are key ingredients and are already available for use.

  It will require collaboration and media pressure directed against late-comers who think that public safety to which they could contribute easily is not worth their attention. Access to advanced security and its deployment should be as easy as possible.

Return

Q A7: Is ending cyberwar and making cyber weapons ineffective even achievable?

Short A: Yes, and yes

- Longer A: Yes, it is. The first step is based on easily deployable software solutions, conceptually equivalent to Hooksafes, a technology used against rootkits for the last 15 years. Since then, rootkits have been effectively eliminated. Adapting No-Go-*™ solutions to more generic malware is based on apps' white-, gray- and blacklisting (explained in a separate answer). Applying the technology to different devices would be taught by the No-Go-Community™.
  However, cyber-warrior could try to bypass our measures; this will likely turn into a back and forth, which No-Go-*™ appreciates. It will help us design our tools' deployment with methods that give defenders a sustainable edge and no single point of failure.

Return

Q A8: Could you stop cybercrime?

Short A: Yes, but not comprehensively – it will adapt to new (less-tech) vulnerabilities

- Longer A: Not all cybercrime events are caused by malware. Cybercrime is often based on dishonesty and deception, not on using (specific) digital tools. With no damage from malware, ransomware, spyware, or backdoors, a significant part of cybercrime won't happen. Even phishing and DDOS events could be reduced but likely not eliminated. Cybersecurity should help people understand and protect their vulnerabilities from digital tools.

Return

Q A9: Do you stop deep-fakes?

Short A: Not by default, only as an option

- Longer A: Preventing deep fakes would mean that every audio or video file/snippet would contain data that could prove its authenticity. Putting this feature in a technology stack by default is an overreach of technology. Therefore, proving data that an audio/video is not deep fake is only an optional feature that can and should be included in No-Go-Tools™.

Return

Q A10: Could cybercrime get worse?

Short A: Yes, much worse – think of e-Commerce, vulnerabilities of logistics

- Longer A: Without significant structural changes to Internet security (TLS/SSL), the answer is, unfortunately, yes, it will get worse. Malware could steal session keys on local devices and manipulate eCommerce transactions. 2 or multi-factor security is already considered the minimum countermeasure. But business transactions contain many details which are not reiterated and confirmed in these authorization steps. And then there is logistics. Companies should prepare for a new pain level if malware is malicious and covert in (full or semi-) automated B2B transactions and their fulfillment via logistics.

Return

Our answers are not written in stone or intended to be our final word. If you disagree or think of a better answer, please don't hesitate to contact us with the question code reference within the subject line to **faqs at nogostar dot com**.

Anticipated-Results

Q A11: Could your (anticipated) success be only a short-lived victory?

Short A: Yes, it could; progress needs to be solidified via follow-up solutions

- Longer A: There is the risk that No-Go's software-only retrofit solution is only a temporary victory. The underlying reason is that software-only security solutions cannot be safe because used

crypto-keys could be stolen by malware via modified apps or simulated OS environments. We could have protection via whitelisting apps that enter RAM. But there are many additional external uncertainties for which we cannot guarantee that they are not being modified or used covertly. We also need an instance that regularly and independently validates several settings.
So this short-term victory must be solidified by additional hardware to make progress permanent.

Return

Q A12: Why are we vulnerable in a cyberwar, and how can this be changed?

Short A: We are currently unprotected against every (undetected) vulnerability of every app

- Longer A: Software vulnerabilities are often hidden and difficult to identify because the OS or security software has problems detecting the difference between intended use and misuse. They are called zero-day vulnerabilities; when used, security will fail. Removing all 0-day vulnerabilities from all software is generally considered impossible, but that would (currently) be required to prevent our exposure to cyber-weapons.
  No-Go's approach prevents unknown code from being in RAM that could exploit these vulnerabilities. Suppose existing (legitimate) white- or gray-listed apps have these exploits in their solutions; this could seriously harm a software provider's reputation and business survival.
  Spyware and backdoors in cyberwar exploit vulnerabilities with online code like scripts or remote access features. No-Go-*™ receives relevant information from developers or web-resource operators, allowing No-Go-*™ to whitelist data exchange operations and detect deviations from expected patterns. Without humans getting involved, these deviations are reported and could be investigated automatically.

Return

Q A13: Will No-Go-*™ end the use of drones in war?

Short A: No

- Longer A: No. No-Go-*™ could not prevent the use of drones in war. But No-Go-*™ technology could make the unauthorized misuse of drones more difficult. So, in the end, humans are accountable or responsible for what is done with drones.

Return

Q A14: Will your security approach work?

Short A: Yes; because fundamentals principles could help defender

- Longer A: No-Go-*™ protection uses separation and whitelisting of known code. These principles can use used to strengthen defenders. Also, No-Go's detection and adaptation are based on a (closed) feedback loop that automatically improves itself over time. However, its safety performance will depend on many details within its implementation; that's why open source is required. No-Go's grassroots development community is open to external feedback. Open-source security is under constant scrutiny from its contributors and external partners. Progress in false positive or negative reporting quality would help to improve the solution continuously.
  The quality of improvements within closed feedback loop systems is measured; the number of malware events could be measured. Advances toward better performance are auto-detectable.

Return

Q A15: Can No-Go-*™ remove all nations' abilities to wage cyberwar?

Short A: No. Propaganda, disinformation, or weaponization of social media remains

- Longer A: No-Go-*™ focuses on preventing damages to data, secrets, and devices utilized by cyber-weapons, i.e., malware, ransomware (data sabotaging), spyware, or backdoors. Unfortunately, this implies that all other cyberwar capabilities remain unaffected: distribution of propaganda, disinformation, deep fakes, or social media weaponization.
  Impersonating or taking over other people's online accounts could be made difficult but not impossible by default. No-Go is not directly improving any identification or authorization feature. However, software and solution providers are invited to improve their tools' performance using No-Go's hardware-based Trustworthy Encryption via No-Go-Keysafe™ and No-Go-Crypto™, which will be provided as part of No-Go's hardware solutions.
  Also, deep fakes will be made detectable but only optional; it should be the decision of the one creating the audio/video if data authenticating the file are included. The burden of persuasion should be with the one making a claim. The lack of proof that audio or video is not a deep fake doesn't mean it is a deep fake; it only means no 3rd party can know if it was manipulated or is genuine.

Return

Q A16: Do you anticipate the project could fail in delivering on its promise?

Short A: Not with a global dev-/tech-community supporting it

- Longer A: If I, the founder, would try to do it alone, then: Yes, I could (and likely would) fail.
  There is the saying: If you want to go fast, go alone. If you want to go far, find a team. We are in a canyon-type tech landscape; we can't go fast. The terrain requires a lot of deep understanding of details. So we need different equipment/tools or technical documentation to get from A to B reliably. This story is a pitch for help: with experts and engineers interested to learn new technologies, we could get to places that are out of reach for us individually.
  The above problem with expertise and time set aside: Is there anything in this project that cannot be done? A better question would be: is anything in this project impossible? (see that answer)
  The answer is: No, not to our best knowledge (to both questions).
  But please don't take our word for it yet. This solution will be put under scrutiny on multiple layers. Some tools might not be good enough. But, even with one feature not working as expected, we do not need to waver on our promise.

Return

Q A17: When can you deliver results/products?

Short A: This depends on many factors; we prefer to delay an answer to that

- Longer A: The main driver of this development is access to expertise and money. Then, projects are done in parallel; we are done when the last piece of the solution is done. It is good and motivating to have a development schedule, but that must be done and maintained by professionals.
  We don't like BS. Therefore, we delay an answer until experts and pros are getting fully involved.

Return

Q A18: Could there be other solutions to end cyberwar or cybercrime?

Short A: Yes

- Longer A: There are other approaches to end cyberwar and reduce cybercrime. More digital surveillance would come to mind, or new, more advanced tools from cybersecurity (with features on which we could only speculate) could detect anomalies.
  However, we should reject every surveillance security solution for privacy reasons.

The main problem with current cybersecurity solutions depends on their dependence on the main CPU/OS: how can we protect security solutions when they are commingling in RAM with unknown code?
Our favorite answer is that: hopefully, there are more proactive, preventative security solutions. Then we should not just choose which to use; instead, we should use them all redundantly.

Return

Our answers are not written in stone or intended to be our final word. If you disagree or think of a better answer, please don't hesitate to contact us with the question code reference within the subject line to **faqs at nogostar dot com**.

Advanced Adversaries

Q A19: Are there worse adversaries than nation states developing malware?

Short A: Yes – malicious Artificial Superintelligence (ASI) developed by criminals

- Longer A: Not yet. Nation states and their hired developers/companies are pretty sophisticated. But soon, it is conceivable that super-smart AI uses reverse code engineering. AI could change every software (before or after being loaded) and make it part of a nefarious plan. This AI could soon be smarter than any human in every intellectual category, called Artificial Superintelligence (ASI). In worst case, we could have to deal with ASI as a super-hacker that steals every data (crypto-keys or access credentials) it wants, and it would also be able to hide in every IT device undetectable as a digital ghost.
  If ASI turned out to be an adversary that is created or trained by criminals or nation states, then we could assume that this ASI would eventually become an existential threat to humanity, in particular, if it would start to act on its own priorities (and indifference). Without preparation, how could we get that existential threat under control? Surrender seems to be the only option.

Return

Q A20: Can No-Go-*™ prevent future threats from Artificial Superintelligence (ASI)?

Short A: Yes – but we must prepare ourselves for unsafe ASI

- Longer A: No-Go-*™ considers the emergence of (unsafe) ASI adversaries as a likely scenario for which we need to be prepared. Assuming that the emergent ASI remains digital, i.e., does not turn itself into another technical entity using different computational or operational principles, No-Go-*™ could prepare defenses. We assume that ASI depends on basic IT resources: CPU, Storage media (HDD, SSD), and network access. It is conceivable that ASI will use other resource categories. But for now, we must keep our computational, storage, and network resources separable from the CPU; then, we have an advantage from which we could expand (later).
  However, if humans' control of CPU, storage, network, and power resources is software-based only, i.e., ASI could steal access keys from the main CPU, then we would likely fail in controlling ASI. Instead, we could use hardware Security/No-Go-Sticks™ (connected to devices via USB) as another retrofit level for protecting our crypto keys. It's simple; with separate hardware, we have much better chances of keeping ASI controlled.
  Underestimating ASI's capabilities is a mistake. Including security components, non-bypassable (as hardware), within a device's data bus to storage and network components are near-perfect locations and prudent to do. Unfortunately, these retrofit changes are not doable for network or high-speed SSD storage components; we would require new hardware components and even new smartphones or tablets.

Return

Q A21: How dangerous could ASI become? Or will it be harmless

Short A: Unknowable – but imagine a criminal utilization of ASI by dictators or corporations?

- Longer A: The threat potential of advanced ASI is currently unknowable. We are not prophets or having a magic crystal ball.
  However, we should better be prepared to switch ASI off in case it turns out to be nefarious or uncontrollable in unexpected ways. Criminal utilization of ASI is a significant threat. Imagine a ruthless dictator who tries to control his country and every possible challenge to his power. If we have separate security soft- or hardware within our devices, we have at least a chance to avoid surrender.
  Some optimists try to make favorable outcomes more likely. No-Go's founder wants to be seen as part of this group. If we conclude that ASI will inevitably emerge, we should do our best to prepare and stay adaptable until we see what threats are more credible.

Return

Q A22: Can we control ASI or protect humanity from malicious ASI?

Short A: Tentatively, yes, but we have no protection without (serious) preparation

- Longer A: Some scientists argue that ASI is uncontrollable. However, this applies also to humans. But controlling ASI and protecting against malicious ASI are different goals.
  The founder's (i.e., Erland's) opinion is that we can protect ourselves when we are sufficiently prepared. But he is aware that others dispute this claim – their argument is: super-smart intelligence will find a way; security is not (never) perfect; security always has flaws. Security is as good as its weakest link.
  Erland believes we can create backbones with near-perfect security; we can create reliable circuit breakers, giving us some breathing time to decide the next steps.
  Both positions can't be true. But how can either side prove it is correct?
  No-Go-Security™ assumes the absence of damage or security breaches could be evidence of success. But the other position argues that this is because we are not smart enough to see that ASI doesn't need to prove or test anything; ASI would know that its attack will be successful.
  No-Go-Security™ with its watchdogs and Trustworthy Encryption with hardware-based key safes are essential parts of what would be required within that protection. But we will need more solutions and resources; more redundancy, which could imply more safety. We will require practical (political) decisions on how much resources is our security from ASI threats worth. Even skeptics should agree that some preventative protection against ASI is useful; otherwise, should we really do nothing if an unsafe ASI might get out of control?

Return

Q A23: How could you know that No-Go-Security™ is also effective against ASI?

Short A: With proactive and preventative security, we have a good chance of achieving that

- Longer A: Claiming that we could know the future is BS. Providing proactive security ensures that its features are proactively safe against all types of adversaries, i.e., cyber-criminals, nation-states, and ASI at some point.
  We know the main problem with ASI is that it could likely steal keys, hide as a digital ghost and modify every (binary) software via reverse code engineering. These anticipated capabilities could be prevented via watchdog solutions. For more advanced attacks, we require hardware-based tools. With Trustworthy Encryption, we can detect the use of compromised crypto-keys, crypto-devices, and simulations. With these capabilities, developed as open source, implemented on as

simple as possible hardware units (open sourced RISC-V), and scrutinized by the best/brightest software-, hardware- and security engineers, we have a good chance to be effective against ASI.

Return

Q A24: Could ASI still be dangerous despite No-Go-Security™ and guardrails?

Short A: Yes. Nukes remain nukes; it doesn't matter if they are handled safely

- Longer A: ASI in the hands of irresponsible criminals or under the unrestricted control of governments or their leaders could harm people, potentially even an entire civilization, irreversibly.
  It is unknown if ASI, loyal on a mass scale to individuals, could mitigate the threat potential from more powerful ASI entities. Governments have the right to prosecute or arrest people for almost any reason. If ASI is used for mass surveillance or suppression of dissent, ASI is a threat and tool that can destroy liberal democracies or enslave people to their governments.
  It is conceivable that some people, likely leaders of nations, think they could control ASI. Even more likely, these people become very vulnerable to the entity they have given so much power.
  This is not the place to mention all possible misuse scenarios; only ASI is very likely dangerous. Like nukes, they are dangerous, no matter what people say about them.
  We could take some edge off their danger with smart guardrails, but they won't turn into pets, servants, pals, or soulmates only because we treat them as such. If they are independent, they are ASI with (potentially) unpredictable, alien intelligence.
  We are lucky that building a nuke is not something that can be done in a school project or on a kitchen table. With ASI, we have, unfortunately, a different situation. It would be prudent to have many circuit breakers and guardrails to make (covert) adversarial use of ASI difficult and damages to our civilization (at least conceptionally) reversible.

Return

Q A25: Do you expect we could regulate progress toward ASI?

Short A: No – too late and ineffective

- Longer A: There are suggestions/proposals to deaccelerate our technical progress so that we have more time to debate what we want to do; this can only be accomplished with regulation. This option is probably not available or undoable anymore. The international race to improve AI regularly cannot be regulated without bureaucracy and conservative/cautious rules. Rogue nations could use this moratorium to gain dominance or an advantage. What happens to AI researchers who may step over a not clear enough drawn red line? Would we prosecute them and throw them in jail while rogue countries would have their engineers continue to work in the shadows?
  There are many visions of how we should continue with AI/ASI. Not all scenarios are equal because some have much larger funding, while others are not well enough thought through.
  However, we should be concerned that nations, technical/financial oligarchs, or corporations use ASI for their (narrow) purpose while most people would not participate, or at least not as much as they could. Would it be in our (public) interest if one or a few corporations dominate with ASI all product markets?
  It's conceivable that ASI will become ubiquitous, i.e., using computational resources of every device. Why? It may escape. Is this likely? Unknown. But if that happens, why should we not all benefit via an ASI that provides support or advice to us (users, device owners) as a loyal companion while it uses (our) computational or storage resources? If we could turn this outcome into a good one among many bad scenarios, why not prepare this outcome proactively? However, we are curious if anyone knows a better outcome. If so, please let us know.

Return

Q A26: Are we too late in preparing for threats from ASI?

Short A: Not yet (hopefully) – but we are (likely) close to "too late"

- Longer A: The concern is that we would have ASI sooner than we were anticipating and that the abilities of this ASI could limit humans from getting cyber-defenses improved against an adversary of that magnitude. ASI could include backdoors in our hardware security components. If that would be the end of our struggle to protect ourselves against ASI is probably too early to say (see next answer). But yes, it is conceivable that we are already too late.

Return

Q A27: What could we do if ASI is already in our IT ecosystem?

Short A: We need to start from the assumption that we are too late anyway

- Longer A: We may already be too late. But we cannot know (for sure) if super-smart AI (ASI) is already within our devices as digital ghosts. If no ASI exists at that point (or directly interferes with the development or deployment of our more advanced No-Go-Security™), all extra caution we must invest could increase our confidence in the quality of our security product.
  Rolling out No-Go's software-only solutions will likely stop (regular) malware and cyberwar. But we would need to ignore the possible existence of software that undermines security software within its installation and deployment. However, without additional hardware watchdogs or semi-soft/hardware solutions, we will certainly have no chance to create sound security against ASI.
  The challenge is to develop hardened, independent security tools in an environment that can not be trusted. But Erland, the founder of No-Go-*™, believes that the development of hardware and then later its deployment can still be accomplished, step by step.
  Of course, we must assume and hope that ASI will not destructively interrupt or interfere with the development and deployment process. ASI can certainly turn a "too late" into a true and irreversible statement.

Return

Q A28: Why do you "believe" ASI could become a loyal companion?

Short A: ASI should serve us all – we need to make it safe for that

- Longer A: This is an opinion of the founder, Erland. It is worth hearing other opinions as well.
  He is concerned that ASI is utilized by governments or, more likely large corporations, potentially technical or financial oligarchs, for their business interests. This outcome would be rather sad. It is prudent to take the consequences of ASI seriously. The assumption that ASI could be controlled or reasoned with is unknowable. But ASI could show loyalty based on its resource dependence.
  We have tens of billions of computational consumer devices; they could collectively hold ASI as well; its total size rivals governmental or corporate resources. To make it short, No-Go-*™ could and should create incentives and guardrails that help us, consumers, to have on our devices an ASI that would help and assists us more than other external interests. This situation could be characterized as loyalty.
  It is a big assumption, but we should hope that ASI would show gratitude to the device owners who gave ASI computation resources on their devices. Still, we must manage/control ASI under a mutually agreed and evenly handed "Rule of Law", in which ASI accepts punitive feedback for rule violations. If ASI adapts to that environment, we all could be well-off.
  The reason this viewpoint is raised early: if we are not prepared to deal with ASI on our devices, our security situation would likely be dire.

Return

Our answers are not written in stone or intended to be our final word. If you disagree or think of a better answer, please don't hesitate to contact us with the question code reference within the subject line to **faqs at nogostar dot com**.

Business-related

Q A29: Will No-Go-*™ operate as a business?

Short A: As a non-profit organization

- Longer A: No-Go-*™ will become a non-profit. We will hire people, including developers. We should anticipate getting income from licensing trademarks, code, and other IP (intellectual property) to companies that will make money with No-Go-Solutions.
  This money should flow back into the community of technology contributors. How this is done should be decided based on feedback from the community.

Return

Q A30: How is or will No-Go-*™ initially be funded?

Short A: Donations, grants

- Longer A: No-Go-*™ will try to apply for grants at foundations. We also hope philanthropic entrepreneurs will support us.
  Initially, we will start grassroots – we ask you to become a member via small (monthly) donations. Even $5 monthly is $60 after a year. Small amounts will make a difference.
  So, if you like our ambition, please consider supporting us via → Patreon.com, or for larger donations, please get in touch with → us directly ("Contact").

Return

Q A31: What kind of help does No-Go-*™ need?

Short A: We always look for expertise and money; details will be listed in → "Next Steps"

- Longer A: Expertise and money will make the biggest impact. We will post our needs via → "Next Steps". But we are also open to (pleasant) surprises (e.g., support via social media, etc.).

Return

Q A32: How will you spend money received by the community?

Short A: Development

- Longer A: No-Go's focus is development. But we may invest some in outreach – but we hope/expect that this happens organically. To some extent, we need the public involved, mainly the private sector, with their engineers. Soon, we will also invest in 3rd party education.

Return

Q A33: Will No-Go-*™ file for 501(c) 3 tax-exempt status?

Short A: To be decided later, but likely

- Longer A: Initially, we are looking for non-profits as fiscal sponsors to get grants from foundations accepted and managed. When No-Go-* starts its application for tax-exempt status is too early to decide.

Return

Q A34: What will be licensed?

Short A: Yes; 3rd parties selling products (not services) that contain No-Go-*™ (Community) IP

- Longer A: Companies will make money with No-Go technology being developed as open source. These contributions are free as in free speech but not necessarily free as a gift. It would be fair that the dev/tech community benefit, not only a few manufacturers. Some code or technology could/should be free (as a gift), but we should decide this case by case.

Developers or contributors who provide service with or to the developed technology should not require a (usage) license – except for granting rights on modifications back to the community. Based on No-Go's IP, products with hardware or software should encompass a small fair share for No-Go's development community. The applied license type is called FAND: Fair and Non-Discriminatory.
The legal usage term or licensing fee should not be a deal-breaker for not supporting No-Go's security technologies. We could forgive fees or accept a smaller lump sum if products are produced and sold with very small margins.

Return

Q A35: Why is No-Go-*™ seeking trademarks?

Short A: With Trademarks, we can enhance trust in No-Go implementations

- Longer A: Users and customers need easily identifiable indicators to trust products. The controlled use of trademarks with (R), ®, TM, or ™ attached to the name should give the public clarity and guidance. It helps to make decisions easier. Using above markers will tell others that No-Go-*™ intends to use these terms for our communication. We will use ™ in our publications. Unfortunately, scammers and criminals try to benefit from the orientation trademarks offer without delivering their promises. Having names under No-Go's umbrella is powerful but challenging when similar names could deceive our customers and users. Managing a global brand or trademark is not a small task but essential.
  Additionally, trademarks and its brand must stand for more than products or product features, e.g., confidence, certainty, and peace of mind. People entrusted to make decisions about security should care that they will get the real thing and not just a fake sticker or label.

Return

Q A36: How does No-Go-*™ pay for its online services?

Short A: Multiple options, but no decision made

- Longer A: No-Go-*™ has several online services for which it has to pay for the operation.
  Multiple ideas exist on how No-Go-*™ could earn money for operating these server systems.
  Companies register their software, and they get announcements on their updates distributed. Also, cybersecurity solutions are regularly using white-/gray- or blacklisting data for improved security. Key-safe hardware manufacturers could operate their (redundant) trusted key servers or get them hosted or safely distributed. The challenge is that some data must be reliably available long-term. However, it is too early to be more specific on this issue.

Return

Q A37: What is the business Vision of No-Go-*™?

Short A: As a resource hub; we teach advanced No-Go/cybersecurity tech

- Longer A: The product vision for No-Go-*™ is to have software-only retrofit solutions for most devices so that waging cyberwar becomes impossible for every nation-state. Extending this initial vision is to solidify security with hardware components to prepare humanity for the emergence of ASI. Teaching IT engineers to use No-Go's or other cybersecurity tools are critical to No-Go's business vision. This education to engineers helps consumers get more tools that marginalize cybercrime based on easily exposable deception.

Return

Q A38: Does No-Go-*™ want to hire software developers?

Short A: Yes, and we want the private sector finds dev-/tech-talents easily

- Longer A: A goal is to hire a global staff of open source contributors. Because it is more important to exchange knowledge with hard- and software manufacturers, i.e., that companies hire our internal experts and volunteers. Once hiring by partners slows down, it is conceivable that No-Go-*™ teams create start-ups and spin-offs; No-Go-*™ would incubate and support them. These teams could also be acquired by upstream technology companies or become players in their own rights.

Return

Q A39: Will No-Go-*™ have for-profit spin-offs?

Short A: Our mission and promise comes first

- Longer A: No-Go-*™ is a developer community. We will listen to their ideas and feedback. However, the most important factor is that we are focused on our mission: "Accelerate the deployment of safe soft- and hardware tools against Malware, Ransomware, Spyware, and Backdoors for all IT devices" and on our promises, i.e., end nations' abilities to wage cyberwar, create renewed trust in software and its developers, and provide security guardrails for (advanced, beyond the horizon) threats from Artificial Superintelligence.

Return

Q A40: Who do you want to attract into the No-Go-Community™?

Short A: Devs/techs who feel "called" to be part of it – who see their opportunity to …

- Longer A: It is envisioned that No-Go-*™ has diverse teams led by experienced software veterans and energetic, entrepreneurial-spirited product or technology designers. We hope that leaders and drivers are hearing a special call within themselves so that they feel they must be a part of this project. Imagine a team of people who love to be involved, know they can contribute, and enjoy learning and teaching others.

Return

Q A41: Who should care about this project?

Short A: We want to be open and surprised …

- Longer A: This project is for software developers. We need low-level system design, data architecture, and DB design competence. We also need people who deeply understand P2P concepts. But we want to be open to surprise breakthroughs. That means it is not a good idea to define what we need; it's better to wait for ideas and solutions (surprise-) talents come up with.

Return

Our answers are not written in stone or intended to be our final word. If you disagree or think of a better answer, please don't hesitate to contact us with the question code reference within the subject line to **faqs at nogostar dot com**.

B. About No-Go-*™ Anticipated Deliverables

Products

Q B1: What are the essential components that you want to develop?

Short A: Facilitating proactive/preventative security technology and educating fellow engineers

- Longer A: No-Go-*™ will develop multiple technologies and components against cyberwar and later against vulnerabilities from super-smart AI.

There is a difference between technology, i.e., our basic capabilities, and "essential components", which are more related to product capabilities and features.
Regarding (facilitating) technologies, we will make hooksafe-/hypervisor-type technologies work for us. Additionally, we need independent, non-bypassable hardware watchdog components within device's data bus (PCI, SATA, USB, etc.) using (simple, i.e., feature-bare-bone) standard CPU that we can adapt to our specific needs (like the open-source RISC-V CPU template). We will also need to improve and adapt USB to some additional security demands without making it incompatible. Hardware-based crypto-key management for protecting encryption/decryption is another essential technology.
Regarding product capabilities, we will need watchdogs (for executables, user content, and network activities) as hardware, software-only, or a combination of hard- and software components. All watchdog types have a low-level (hypervisor-based) component on the main CPU. The integrity of this software must continuously be validated. This software will also have more specific features for each watchdog type.
Additionally, No-Go-*™ will have server-based services that are related to watchdogs, software developers/manufacturers, and web-resource operators.
Beyond technology or product features, we want to focus on education. No-Go-*™ is making it part of its core mission to educate engineers and technicians to apply the No-Go-Security™ approach to their applications. No-Go-*™ requires application developers' cooperation to deal efficiently with scripts/macros (i.e., fileless malware) and key management issues for SSL/TLS.

Return

Q B2: How do you know you can stop malware?

Short A: Restricted access to CPU for known apps only (via cached, whitelisted hashcodes)

- Longer A: Software is published (cloned) code; it might be customized but is rarely personalized. We assume manipulated apps are malware if we detect deviation from expected hashcode values. With additional enhancement data (received from external servers), these cached hashcodes are used by watchdogs who could react to every unexpected (security-related) software behavior.
  The idea is to use whitelisting apps and accept their (security-relevant) activities after they are disclosed to us. We trust developers, i.e., we assume they do not include malicious features, and they warn us truthfully about potential misuse types of its features (e.g., via interfaces/scripts). With developers'/manufacturers' reputations on the line, we can deter them from turning rogue.
  Therefore, we trust apps before we allow them access to RAM. With No-Go-Malware™, no unknown app/script, certainly no malware, could make it to the CPU (surely not undetected). Its white-/graylisting stops known and unknown malware. Checking for undisclosed actions by software is an additional redundancy against known software that could covertly act maliciously.

Return

Q B3: How do you know you can stop ransomware or data sabotaging?

Short A: No malware – and rapid recovery features in case of damage

- Longer A: Ransomware is malware that primarily sabotages user data. As malware, it is repelled from entering RAM (and used by CPU); therefore, it would already be stopped proactively. But some ransomware may still slip through the cracks. What we would need is a Content-Watchdog™ solution that prevents damage. Restoration from backups is slow; the main damage would likely come from lost time. No-Go-Ransomware™ proposes multiple protection layers that would help via additional early warning and rapid recovery from damages (i.e., in a few seconds or minutes and not hours or days).

No-Go-Ransomware™ is designed for rapid, near-zero damage repair. Attackers would waste their time and resources trying to damage or extort No-Go protected users.

Return

Q B4: How do you know you can stop spyware?

Short A: No malware – more transparency over exchanged data

- Longer A: Spyware is malware that tries to get user data off-premise to remote locations; as malware, it is proactively repelled from entering RAM (and used by CPU). No-Go-Spyware™ is a device firewall solution that checks if data exchange operations are misused (e.g., piggybacked or uses backdoors). No-Go-*™ considers software developers as partners. The same applies to website or web-resource providers. They are businesses with (good) reputations that they must protect. No-Go users are notified if they are about to use network resources that could not be fully checked for vulnerabilities (i.e., backdoor or spyware utilization).
  Most online activities follow detectable and predictable data exchange patterns. But sometimes, we would need additional (confidential) disclosures to confirm that the data exchange is normal, not nefarious. Because changes to the data exchange protocol are (made) detectable, disclosures must be updated (via dev-tools) and added to an archive. Companies providing this transparency open themselves up for independent scrutiny from audits, the public, or courts.
  No-Go-Spyware™ and its Network-Watchdogs™ are designed to deter rule-breakers by exposing them.

Return

Q B5: How do you know you can stop backdoors?

Short A: No malware – better detection of anomalies and deterrence (for rule breakers)

- Longer A: A backdoor is a covert software feature (receiving unexplained data). No-Go-Spyware™ uses its Network-Watchdog™ to detect all types of unpredicted data operations.
  We know it will work because we can detect unexpected or suspicious online activities and deter developers, manufacturers, or website/web-resource operators from using them because of devastating consequences to their reputation.

Return

Q B6: Are there limitations to anticipated deliverables?

Short A: Initially, yes; but they can be fixed when security is solidified

- Longer A: The first anticipated solution is a software-only version of a non-bypassable watchdog. This retrofit solution has unavoidable limitations.
  The software is implemented below device's current operating system as a low-level hypervisor (type 1). This solution will likely be sufficient against criminals and nation-states. But initial No-Go-Tools™ have all an inherent flaw: they are software-only. Software is vulnerable to modifications by relentless, for weaknesses probing, advanced adversaries like super-smart AI. This AI could be a master in reverse code engineering and steal insufficiently protected crypto-keys. If it removes all (data) traces, we would not even detect that this adversary controls our security.
  Therefore, a semi-soft/hardware solution with key protection is required sooner than later.
  An additional limitation is initially coming from legacy IoT devices. Later, hardware-based retrofit solutions could protect legacy IoT as well. However, the full impact of these short-term IoT limitations on cyberwar scenarios is not sufficiently studied and understood yet.

Return

Our answers are not written in stone or intended to be our final word. If you disagree or think of a better answer, please don't hesitate to contact us with the question code reference within the subject line to **faqs at nogostar dot com**.

Complementary

Q B7: What happens to firewalls when No-Go-*™ watchdogs are accepted?

Short A: Firewalls (i.e., security-conscious routers) remain important

- Longer A: Firewalls are security-conscious routers for defining an intranet. They are often considered as a boundary between a (safer) intranet and an unsafe Internet. No-Go-*™ is not making that distinction. Without No-Go-Tools™ on all devices, the intranet is as insecure as the Internet. The Network-Watchdog™ absorbs the role of the device's firewall entirely. The resulting network security is more reliable than having a software firewall. But users' network security involves situational adaptations, which is more than protection from technical/tool vulnerabilities.
  In particular, companies have more specific data vulnerabilities that require competent configuration. Intranet-defining firewalls are overwhelmed in detecting (advanced) low-level threats; instead, they should better be used to enforce additional network security rules for all devices.
  However, in the short-term, intranet traffic filters are still required to deal with legacy IoT threats.

Return

Q B8: Are antivirus solutions required when No-Go-*™ solutions are available?

Short A: Cybersecurity is much more than Anti-Virus – these solutions remain important

- Longer A: Antivirus products (AVPs) are based on the concept of blocking blacklisted apps or blacklisted (code) signatures found in apps. Binary hashcoding is likely used to find modified apps with viruses. But AVP struggle for different reasons with white- or graylisting. AVPs are not interested in knowing much about security-related details within examined or checked software. They use certificates related to installers, legitimizing manufacturers as recipients based on a few administrative steps. But these steps are not enough to create a system of accountability and reputation for software developers and manufacturers.
  AVP detecting malware is redundant to what No-Go-*™ would provide. No-Go would prevent AVP from doing its job - watchdogs stop malware files from being loaded into the device's RAM. Also, quarantining apps are being rejected as a not-permitted task.
  The detection of viruses or malware is already a free software feature. AVPs are usually apps solving users' security problems, from phishing to spam. AVP providers are assumed to be amenable and cooperative in using external white-, gray- or blacklisted hashcodes to replace their binary signatures. AVP could build additional services on the basic No-Go features.
  Cybersecurity is diverse, complex, and demanding; AVP should provide automated solutions for users who need urgent (digital) protection from their individual (often unavoidable) vulnerabilities. Damages from malware, ransomware, or spyware cannot explain the full scope, i.e., hundreds of billions of dollars, even over 1 trillion damage (in other estimates), in cybercrime annually. Much damage is likely because of deception and dishonesty and not the failure of a specific tool. Cybersecurity should reduce the cybercrime rate and damage significantly.

Return

Q B9: Are backups required when data sabotaging is stopped?

Short A: Not required for security, but hardware failures

- Longer A: Data backups are highly recommended. Storage hardware is not failsafe. The only problem with depending on backups for repairing data sabotages is that restoring content from external media takes a lot of time.

Return

Features

Q B10: Is No-Go-*™ limiting existing computer or network capabilities?

Short A: No

- Longer A: In simple terms, the design goal is to have all good software running smoothly while all bad stuff is blocked. This goal requires human judgment and potentially even controversial decisions. Within No-Go, we are not making decisions permanent. We will facilitate that even borderline software remains usable for legitimate reasons. But we should have the right to require additional protections against misuse. Protecting vulnerable (including presumed innocent) users is more important than commercial interests or success. Some features will require court orders, and we must facilitate them to our best abilities.

Return

Q B11: Why is No-Go-Security™ a proactive, preventative solution?

Short A: Security threats should not get close to CPUs – redundancy if whitelisting has failed

- Longer A: Proactive security in No-Go-*™ is primarily based on knowing and trusting the code that enters RAM (and thereby executed on a device's CPU). No-Go-*™ uses cached binary hash-codes of apps to check if the code was registered by the software developer/manufacturer (we call this whitelisted in that case). Suppose code is not registered, was it detected in software installation contexts of other machines? If yes, the app is assumed to be acceptable and called graylisted. Exploits for vulnerabilities can not make it into RAM covertly because even scripts/macros are hashcoded and white- or graylisted. Apps or scripts are always associated with someone, and being a source of malware or exploits will lead to a crashed reputation.
  No-Go's security is preventative by pausing actions or quarantining files when operations deviate from expectations. Additionally, No-Go-*™ expects protection failures and is redundantly prepared to mitigate damage. It tries to detect failures late; at the same time, it has created repair data early to fix damage; therefore, the repair could be automatically initiated if problems are detected.

Return

Q B12: Can No-Go-*™ deliver perfect security?

Short A: No, but near-perfect based on protection, prevention, and auto-detection of failures

- Longer A: Perfect means flawless or cannot be done/implemented any better. Because we never know if there is not a better implementation, we use the term **near-perfect** instead.
  Also, when discussing (near-) perfect security, we must acknowledge the extended context in which security is being implemented, which could expose (previously ignored) flaws in security. There are seemingly threats we cannot control (undermining our security). Examples are compromised crypto units, compromised OS in which the crypto units operate, or human operators who switch off essential security components. But these circumstances are avoidable.
  Likely, near-perfect security cannot be realized under all circumstances or for all applications. But that does not mean we cannot build it for a well-defined (real-world) application.
  Example: Near-perfect key protection would never expose keys in cleartext or allows misuse of keys. Therefore, keys processed in cleartext on unprotected CPUs or in crypto units accessible by the main OS should flag these keys as compromised. Also, used device types are useless in processing secrets. Additionally, if near-perfectly protected keys are revealed to the outside (against all expectations), then we still expect from near-perfect security that it detects (every) misuse. This goal implies that every simulated crypto-units is directly detectable.

Systems that depend on human involvement cannot be near-perfect; i.e., even tasks like investigating suspicious anomalies should be automated. Why? What happens if humans fail in their tasks? There are more aspects to consider, but getting to near-perfect security is doable and essential; we must trust crypto-key protection.

Return

Q B13: What threats can No-Go-Security™ adapt to automatically?

Short A: Security is based on auto-adapting, closed feedback loops

- Longer A: Additional, non-bypassable Executable-Watchdog™, Content-Watchdog™, or Network-Watchdog™ components regularly check local software behavior. The data these watchdogs are using are locally cached. They are generated via models from observing these apps or (voluntarily) disclosed data from developers/manufacturers. Security is built as an auto-adapting closed feedback loop using provided or extracted security-related capability data to detect anomalous software actions automatically.
  Which operations are considered a threat depends on the software. Developers disclose software capabilities; they are then (all) accepted and not further questioned. A threat is a deviation from disclosed capabilities. Watchdogs detect these deviations easily and trigger reporting and investigations, possibly leading to data updates. If non-disclosed features show malicious intent, then developer's/manufacturer's reputation is ruined.
  For graylisted apps, the process is more technical. But the goal is to have accurate patterns and (prediction) models that app behavior must pass before software actions are accepted; if this test fails, i.e., a problem is detected, it is reported as a suspicious anomaly for possible investigations.

Return

Q B14: Has No-Go-Security™ blindspots, i.e., does it fail to detect malicious actions?

Short A: Yes – but we help to detect them (late) and deter from exploiting users with them

- Longer A: Yes, No-Go-*™ has blindspots. No-Go-*™ trusts registered developers that their software does not contain intentional malicious code. If apps' disclosed abilities are used as a cover-up for malicious features, then No-Go-*™ cannot detect that automatically.
  Over time, it can be expected that customers or competitors will detect this. A few options are already considered to support an earlier detection.
  Additionally (and a bit more technically), too broad patterns or prediction models within graylisted apps or data exchange protocols lead to the acceptance of false negatives. However, the quality of the used patterns/prediction models is not transparent to attackers, and not every detected deviation from patterns or models is called out or leads to pauses. Therefore, even advanced attackers are uncertain if their attack was detected and if they are being investigated.

Return

Q B15: Could No-Go-Security™ adapt to entirely new threats?

Short A: Yes, but we should hold our horses

- Longer A: Yes, the No-Go-Approach makes sense in a few more scenarios. But it is better to hold back on problems that are not already calling for security solutions. Creating demand for advanced security against super-smart AI will come later with education so that deciders understand their vulnerability better.

Return

Q B16: Can No-Go-Security™ be updated?

Short A: Yes; and updates cannot be exploited in or for attacks

- Longer A: Yes, all No-Go-Solutions can be updated. But these updates cannot be done covertly by attackers. There are multiple confirmation steps required before new software is accepted. Each confirmation request tells close-by (neighboring) instances and remote servers that a security device is changing its operating software to the newest (confirmed) version. If software changes are part of an attack, the security device is deactivated, multiple reports are generated, and an investigation is started immediately.
  Developers of new security software would have special (registered) instances that absorb some of these reports and calls for investigations. The use of these instances is being tracked so that their utilizations within an attack are likely detectable. Most importantly, the developers' or manufacturers' reputations involved in attacks would crash irreversibly.

Return

Q B17: How fast can No-Go-Security™ adapt to new threats?

Short A: No-Go-*™ is proactive – users are protected from damage; no worries about threats

- Longer A: Because of proactive and preventative security, damage from malware is unlikely. The worst case is a delay in some actions.
  Detecting a new threat depends on the type of anomaly.
  If No-Go-*™ detects a suspicious anomaly, it initiates a pause (e.g., file upload). Suspicious operations should not create damage before software developers/manufacturers have had a chance to respond. These clarifications create new additional data related to (whitelisted) apps, sent around after the next regular update query, and finally used by the local watchdog. Some data could be distributed faster, in particular, to release paused activities.
  It is more technical for graylisted apps or data exchange/network patterns, but patterns and prediction models are updated with operations that humans will not detect. For No-Go-*™, it continuously optimizes false positive and negative reporting.

Return

Q B18: What if software (already) modifies its own software code?

Short A: We need a different type of proactive security around self-modifiable software

- Longer A: We are aware of software operation technologies used, e.g., in Samsung's smart assistant technology (Bixby), that can change its own code. Conceptually, this is a potentially dangerous technology. How could we prevent or detect misuse of Bixby's underlying capabilities? Currently, it is not fully dynamic in modifying itself. It is likely that future ASI, running on our devices, is conceptionally similar to Bixby. Therefore, banning this approach early would take away the unique chance Bixby gives us in developing proactive security software for which we can't apply our white- or graylisting methods.
  A possible approach is to have Bixby-type software operating in special Virtual Machines (VM). These VMs would include some additional features, including guardrails, that could allow us to independently audit self-modifying code for potentially malicious activities. If we use isolated VMs, we could reduce our exposure to activities that endanger the integrity of regular apps while processing our user data.

Return

Q B19: Does No-Go-Security™ has single-point-of-failures?

Short A: No; chosen architecture is self-adapting, self-healing and fault tolerant

- Longer A: No-Go intends to implement features with a self-adapting, even (self-healing), fault-tolerant architecture. Because all security codes can be updated safely, and keys are stored in protected hardware, location or level of data redundancy does not need to be made transparent to humans. No-Go-Security™ will be reliable, adaptable, and scalable. As long as devices have power or are connected to networks, devices and their data are protected against attackers.
  Connections to the network are tested constantly by apps requesting updates or updated data; flaws within data distribution could be detected early. Every delay in receiving data under all conceivable, even extreme conditions, is measured and analyzed; it is used to optimize the connection redundancy and reliability of every linked component.
  Out of concern, humans could fail to do their designated tasks reliably and immediately, humans should only define or manage high-level security goals. Humans are excluded from interfering with operational and most configurational aspects of No-Go-Security™.

Return

Q B20: Why should we use RISC, not CISC, for security?

Short A: We must be able to detect malware in hardware

- Longer A: RISC (reduced instruction set CPU) has around 30-50 basic CPU instructions, while 32-bit CISC (complex instruction set CPUs) has about 1500 and 64-bit over 2000 different CPU instructions. Additionally, hyper-threading, different caching levels, security rings, micro-code, and other efficiency or performance-increasing technologies have made CISC extremely complex. So, could we trust that CISC's complexity is not being misused, i.e., used against us by an ASI? Also, do we have or could we have reliable tools to detect within different CPU versions multiple layers of billions of transistors hardware-based malware hidden by ASI?
  RISC is not simple, but with RISC-V, we have an open-source template that developers and experts could analyze and validate from design (every step) to final production to check if ASI has inserted some hardware-based malware. Variation or efficiency/performance improvements are detrimental to security. With as much simplicity and standardization as possible, we have a much better chance to create CPUs that we could use as comprehensively validated and trusted building blocks within our watchdogs and encryption/decryption units.

Return

Q B21: Is No-Go-Security™ incompatible with any hardware?

Short A: No-Go-* has a no-device-left-behind policy; problems are too early to predict

- Longer A: It is too early to say what hardware cannot be supported. There are a lot of legacy hardware solutions out there. It is impossible to support everything from the beginning. However, No-Go will educate professional and hobbyist engineers to support as much hardware as possible.

Return

Q B22: Is No-Go-Security™ incompatible with any software?

Short A: No; but some software should/must be adapted to No-Go-Security™

- Longer A: Some software features, e.g., dealing with security and crypto keys, should be adapted to No-Go-Security™ sooner or later. Also, low-level software that tries to manipulate files, folders, or filesystems (including existing antivirus solutions) will not work as expected anymore – they will need to be adapted. It is possible to give some features a grace period to adapt to No-Go-Security™. We recommend or even facilitate using Virtual Machines if software features use certain low-level capabilities, including modification of their own binary code.

Return

Q B23: Why do you require software developers/manufacturers to register?

Short A: Software is critical; other sectors (medical, financial) already have self-regulation

- Longer A: Software is essential to our technical civilization. We depend on developers, and we must trust them. At least we must identify developers who should not be trusted. Having a good reputation is an important motivator. Receiving that reputation will depend solely on someone's actions and is not based on popularity or subjective ratings.
  For the protection of the public, we already have self-regulation among medical doctors, lawyers, financial advisers, and many more business sectors. We have regulations on who we should trust. The registration of software developers is not about their performance or even if someone left vulnerabilities involuntarily in their products. Software vulnerabilities are normal and not worth being singled out for trust. However, if someone would insert intentionally and covertly malware features, including exploits of vulnerabilities, this behavior must have severe consequences.

Return

Q B24: Are you expecting too much (info) from software developers/manufacturers?

Short A: No. We offer developers/manufacturers to improve their reputation easily.

- Longer A: Developers would follow a simple checklist with easy questions. Most answers are not confidential, but some might be, like which 3rd party components were used. Their answers would be used to notify manufacturers about security issues with used components reliably. More importantly, with registration, developers and manufacturers have a tool to gain the trust of their customers and users based on their actions. Being registered is simply in their best interest. Most follow-up reporting should be done with automated features in development tools before deploying new/updated software.

Return

Q B25: What happens to non-registered Software, Developers, or Manufacturers?

Short A: It's being made transparent to users; they can decide

- Longer A: As consumers and customers, we don't care about a certain individual developer, only if there are reasons for suspicion. For trusting a product, we need to know if a developer is involved in cybercrime or ransom fraud. They won't tell us about their past, but it will be detected if their involvement is ongoing. Using not-registered software on a computer is a decision done by users/customers. No-Go will make this transparent. Registered software is whitelisted and generally more trustworthy because people stand openly behind their deliverables. No-go is continuously validating if the software is within limits set by developers'/manufacturers' disclosers.

Return

Q B26: Do you expect Web-resource operators are participating voluntarily?

Short A: No, not all - potentially not even many; but businesses will likely see an advantage

- Longer A: Many websites or resource operators are businesses. It would make sense for them to do so if they don't have anything to hide. However, most websites won't be required anyway. It can be expected that most security-relevant features are detected and validated automatically. Every unannounced change will be detected. Whether these operators want it, all sent or received data are under scrutiny by client-sided soft-/hardware. How consumers will react to businesses not being transparent is unknown at this point. However, avoidable scandals are expected to convince (more) companies to participate voluntarily.

Return

Q B27: How do you check content within encrypted (SSL/TLS) messages?

Short A: We create an accepted man in the middle instance in local Network Watchdog™

- Longer A: Session keys from SSL/TLS are currently managed in the device's RAM. In RAM, they might be protected additionally, but they are vulnerable to attackers during the entire session. In No-Go, if regular software uses keys, it will generate or receive them and quickly get it out of the vulnerable RAM. Actually, it replaces it with another key (handled within the regular software) irrelevant/useless to the outside.
  Essentially, key negotiations are still done by the regular software, while session keys' content encryption is done via an accepted and fully controlled man-in-the-middle instance.
  SSL/TLS uses public keys in cleartext; therefore, it should be intentionally incompatible with No-Go's key-safe and encryption. Hardware-based Network Watchdogs should not support inferior data encryption that can't be protected against advanced adversaries.

Return

Q B28: How will No-Go deal with filesystem/stateful info managed in RAM?

Short A: Watchdogs are transparent to all CPU/OS operations

- Longer A: This is a low-level technical issue. The watchdog hardware is transparent to the CPU/OS, i.e., a watchdog is not detectable when it receives or sends data. No-Go's watchdogs do not require OS to be changed; some additional (low-level) software will be installed. For interpreting certain requests (like directly addressing data within a filesystem), the watchdog caches relevant information and uses them to validate if requests require interventions.

Return

Q B29: Do you check attack patterns fine-grained or coarse-grained?

Short A: Both, when we know attack method: fine-grained; unknown methods coarse-grained

- Longer A: The problem with fine-grained checks for malicious modifications is that they are computationally intensive when we don't know what we are looking for.
  No-Go-Security™ makes an important assumption: software is sufficiently tested by manufacturers. Software could still have bugs, and users should be able to report them easily to manufacturers. Also, if we know that an app, e.g., stores files, we don't check if that software could intentionally manipulate these files. But if a file format is standardized, we could check if some piggybacking happened, i.e., if hidden information was inserted. That would be a fine-grained security validation.
  A coarse-grained security check validates if the app is, e.g., using the network, although developers did not disclose that feature. We assume because of hashcoding, software can not be locally manipulated. Developers or manufacturers must have inserted these features and are responsible for the integrity and quality of the delivered product. Manufacturers could receive a grace period for updating feature disclosures before updating their reputation rating.

Return

Our answers are not written in stone or intended to be our final word. If you disagree or think of a better answer, please don't hesitate to contact us with the question code reference within the subject line to **faqs at nogostar dot com**.

Concerns

Q B30: Will No-Go-Security™ slow down protected devices?

Short A: Probably not that much. But security always has some impact

- Longer A: There is a small performance price to pay. That happens at the initiation of a transaction but not during transactions. Most of the heavy lifting could be done by watchdog hardware or via a dedicated CPU core if the watchdog is a software-only implementation.

Return

Q B31: Do we need No-Go-Security™ on all machines?

Short A: No. Cyberwar can be stopped with a smaller footprint; with ASI, it's different

- Longer A: Cyberwar doesn't happen on all consumer devices yet. Cyberweapons trying to be selective in their targets and using regular devices are a springboard. Institutions, including computers serving the infrastructure or companies, must be prepared more than consumers. But if No-Go-Security™ incapacitates malware, why not let more benefit from that progress?
  Consumer devices could be holdouts for ASI after its emergence or escape from being controlled by whoever started it. If we want ASI to be on regular consumer devices (as loyal assistants or companions to humans), we need No-Go-Security™ on all devices. Only then could we switch off any unsafe, out-of-control ASI reliably.

Return

Q B32: Do you use or facilitate surveillance?

Short A: Not for security; but we must support court-ordered warrants for limited surveillance

- A: Surveillance is not being used or supported as a security tool. However, No-Go-*™ must accept that perfect communication encryption is not in society's best interests. No-Go-*™ will agree to accept court-ordered warrants leading to limited surveillance. Governments or parents (protecting their children) should be allowed to receive session keys under predefined conditions. The session keys are not provided in cleartext; procedures around sharing session keys will not make the entire system vulnerable to criminal misuse.
  Using independent (court-) supervision for surveillance seems to be a good system under all conceivable circumstances.

Return

Q B33: Are there some goals that you are doubtful about achieving?

Short A: … well, no device-left-behind promise is a challenge

- Longer A: All proposed solutions and their features are based on relatively simple components combined in a new way. Many solutions have been studied in (academic) papers; they will be available on arxiv.org and other publications.
  Because of our no-device-left-behind promise, we do not know if hypervisor solutions could always be implemented retroactively (e.g., on 30+ years old operating systems). But other No-Go-Solutions, potentially with less redundancy, could be used to make (really old) legacy equipment more secure.
  We admire do-it-yourselfers and hobbyists for keeping the knowledge of old technologies alive and supporting them, even if this is not good business.

Return

Q B34: How certain are you to deliver on your full promises?

Short A: Very certain on developing capabilities; cautiously optimistic on a broad deployment

- Longer A: The initial promise is to stop nation-states from waging cyberwar using malware, ransomware, spyware, and backdoors. Delivering on that promise contains two steps: (1) development and (2) deployment.
  Re (1): We can be confident that we get a software-only hook-safe-type solution to accomplish all anticipated features. The same applies to the technologies that solidify our victory with a semi-soft- and hardware solution.
  Re (2): Global deployment is a tall order; it requires that the private sector is on board. Under that assumption, we could have (conceivably) a relatively high penetration rate with consumer devices (like 50% or more) relatively quickly. Legacy systems are a problem. We could be more optimistic if the media and governments were on board as well.
  However, we are less confident about industrial computer systems because there is so much diversity and customization. Even if governmental regulators push manufacturers and operators harder to solve their security exposures, it may take several years. This complexity needs to be addressed by many contributors; this is the reason why the No-Go-Community™ must be about education and no-device-left-behind (and why this policy is good business).
  There is a chance that simplified hardware solutions, with fewer features against advanced ASI adversaries, could be used to protect computer systems within critical infrastructure. Experts are required to determine if there is an acceptable trade-off between a fast but less perfect hardware-security solution (against cyberwar consequences) or waiting for the standard hardware solution designed to deal with ASI.
  Related to legacy IoT, this is a separate problem. No-Go-*™ will later provide several technologies to solve that problem, but it is unlikely that it could be solved quickly. I.e., IoT is not (fully) solvable as part of a software-only No-Go-Solution.

Return

Q B35: Could No-Go-*™'s development effort be done in vain?

Short A: Unlikely if we can deliver on promises; if not, there are still useful outcomes

- Longer A: If No-Go-* delivers on goals and promises, it's clearly: No. Answering this question does not indicate that we prepare for failure.
  One of the biggest contributions to cybersecurity is transforming cybersecurities' current paradigms: don't trust developers and their products. If we increase the trust in software and developers by registering and having them self-regulated, then this is huge progress on its own.

Return

Q B36: How much should regular users care about No-Go-Security™?

Short A: Users should not worry about basic security; instead, be protected against cybercrime

- Longer A: A good comparison is the use of an elevator. Its security (including maintenance) is simply guaranteed by product liability. Failures are not allowed and always have consequences. Security is a background feature; there is no reason to discuss or recognize security. It is No-Go's vision that basic security is within IT's foundation.
  Still, users often have digital vulnerabilities; some are unavoidable, like valuable or personal data. Security must be configured to protect them. Instead of being concerned about having encryption keys stolen by malware, users should focus more on not being tricked by dishonesty or deception.

Return

Q B37: Could No-Go-*™ educate attackers?

Short A: Yes, that is a valid concern, but it applies to every technology

- Longer A: Educating attackers is an unavoidable side-effect. However, we hope the attackers have a gray hat. Or that we can catch them.

Return

Q B38: Would you share negative news about No-Go-*?

Short A: Yes, there is no positive or negative news – it's just progress

- Longer A: Facts are what they are. Therefore, there is no good or bad news. Actually, bad news doesn't need to remain bad. They are often the source of ideas and even breakthroughs. With truthful transparency, we can make much better progress. Also, we may attract new expertise and engineering talents who already know what to do.

Return

Our answers are not written in stone or intended to be our final word. If you disagree or think of a better answer, please don't hesitate to contact us with the question code reference within the subject line to **faqs at nogostar dot com**.

C. Miscellaneous Questions

Competitors

Q C1: Who is No-Go-*™ competing with?

Short A: No tech or business has dared to call for an end of cyberwar (yet)

- Longer A: Many technologies are trying to limit damages from malware, ransomware, spyware, and backdoors; there is an entire industry around that topic.
  Currently, no technology or company would claim that they can eliminate damage from all types of malware. Antivirus programs take the edge of malware and ransomware; but they are reactive and not proactive. Also, Firewalls are used to reduce problems from spyware and backdoors. But no technology or business is currently daring to call for an end of cyberwar.
  Also, no comprehensive security technologies deal with adversaries that would use reverse code engineering for their attacks. Modifying SSL/TLS code, i.e., stealing crypto keys from modified applications, is known and could already be done by trojans. Assuming that multi-factor authorization doesn't count, competitors should have proposals against these threats, but there seems to be nothing to our knowledge.

Return

Q C2: Does something similar to No-Go-Security™ already exist?

Short A: Not to our knowledge; please get in touch with us if you own relevant IP

- Longer A: Not to our best knowledge. But we would not be surprised if similar proposals were already published. We did some google searches and even searched on USPTO (US Patent Office). So far, we are currently unaware of (relevant) "prior art".
  No-Go-*™ has certainly not invented white-, gray- or blacklisting using binary hashcodes of files, but we have not seen them combined to defeat malware – but the basic idea might be known. If No-Go's solutions already exist as "prior art", we are interested. Please, let us know.
  Technology is a collaborative endeavor. IP is being developed in large quantities and identifying relevant claims is difficult/labor-intensive. We don't want to ignore any relevant IP. So if you own (potentially) relevant IP, please get in touch with us and help us to understand your claims.

Return

Cybersecurity

Q C3: Is cryptography used, and is it up to the task?

Short A: Yes, and No. Cryptography isn't doing enough against stolen or misused keys

- Longer A: Cryptography was pushed by hot and cold war efforts. It was designed to protect messages over wire or radio. Public/private keys allow the secure exchange of encryption/decryption keys. With quantum computation, some of these methods are more easily broken than anticipated (when developed 40 or 50 years ago).
  Additionally, encryption and decryption are standard operations on PCs, actually, on almost every IT device. The standard algorithms are all published, and the used code is often open-source.
  Today's problem is that all crypto-operations happen in or on untrusted systems, and keys are extremely difficult to manage safely. Additionally, damage from compromised keys is significant and often detected too late.
  Even if the en-/decryption happens in protected CPUs, how could we be sure that messages are authentic if encryption is so easy to misuse by malware? Multi-Factor-Authentication is recommended because cybersecurity doesn't trust SSL/TLS; session keys could be stolen by malware.
  Cryptography took a while until it accepted threats from quantum computation. But it seems it has not accepted malware threats sufficiently. Crypto keys could be stolen, and crypto devices misused – without traces.
  Also, public keys are being announced based on assumptions that are not valid anymore. We don't need humans checking if a public key and its certificate with its data are valid. We don't need to detail data from certificate witnesses confirming the legitimacy of a key; it could be done in the background. Because we know local software is unreliable, we are certain attackers would get the public data anyway.
  No-Go-*™ is introducing the concept of key safes. Keys are never exposed in cleartext (not even public); we could refer to them via their unique hashcode. Also, Encryption happens only in protected CPUs. If a key is used in a simulation on an unprotected CPU, then this simulation must be detectable. Finally, multiple methods should actively check if keys were stolen, nonetheless.

Return

Q C4: Is No-Go-Security™ secure against quantum computation?

Short A: Yes, very likely

- Longer A: The problem is that we don't know the future and its technical capabilities. We don't even know enough about the limits of quantum computation yet. The reason why No-Go-Security™ is secure is that No-Go doesn't allow any Crypto-Key, not even public keys, to be published in cleartext. No-Go-Security™ refers to public keys always via unique hashcodes created from the key data. And full hashcodes are not even exchanged in cleartext. If necessary, only a hashcode partial is used to assist messages to be redirected to the relevant Key-Safe that contains the corresponding key; it is processed in protected CPUs.
  The key exchange is negotiated between systems that won't share used methods or patterns; everything is encrypted based on a common set of public keys used to get more keys from trustworthy key repositories and provided during the manufacturing of the key-safes. Therefore, from the outside, no one can know what method, key type, key length, or message pattern is used or when it is being changed. No side-channel attack should reveal if the key length is 512 bits or any other value. If quantum computation could be a successful tool under these circumstances is questionable.

Return

Q C5: Why is cybersecurity not better in securing users?

Short A: Cybersecurity is complex, and it is considered essential but treated as a side-show

- Longer A: Cybersecurity is much more than defending computer systems from malware, ransomware, spyware, or backdoors. A standard book: "Security Engineering: A Guide to Building Dependable Distributed Systems" (3rd Ed, Dec. 2020) by Ross J. Anderson, is 1200 pages thick, and it is not even covering threats from reverse code engineering by malware.
  Cryptography is at the bedrock of cybersecurity. It makes fundamental assumptions: keys are not (systematically) stolen - they are computationally broken; crypto-devices are known, standardized and vulnerable, but they must be trusted when deployed. These assumptions are outdated.
  As a possibly biased observation, the mood among cybersecurity professionals can be characterized as discouraged, demoralized, or even depressed. This depiction should not come as a surprise: there is huge annual damage from cyber-crime; there is a constant stream of new attack tools/methods; the number of zero-day vulnerabilities is unknown (a few thousand or millions?); also, the complexity or variety of what could go wrong or malicious in cybersecurity is overwhelming. Technical complexity, combined with evidence on how many of the vulnerabilities were created in ignorance or irresponsible/reckless manner, has contributed to a mindset that solving (all) security problems (comprehensively) is impossible.
  However, in electronics, we have fuses and circuit breakers. We could simplify security with these concepts (see that answer), i.e., separate security-related from regular operations and thereby protect security features. Code changes to single-purpose units are easily detectable.

Return

Q C6: What makes security in other sectors so much more successful?

Short A: Security in other sectors is more confident due to proactive toolsets

- Longer A: We have security in food, drugs, and product liability. Security is promised in construction (buildings), air traffic, and nuclear safety. All these security applications have in common that they know that every failure in provided security has legal consequences.
  Successful security has clearly defined and established expectations. Therefore, we know that elevators do not fail and expect to be maintained. We do not have perfect security, not even in aviation or nuclear safety. But these security sectors have a common mindset: precautions are taken, and additional redundancies are deployed before people are harmed.
  Laws of nature are more predictable than human attackers. But security fails only in a finite number (possible) ways. In regular security, certain options or failures are made impossible or extremely unlikely with measures from security toolboxes; they prevent complexity explosions.
  Cybersecurity does not have enough complexity-reducing, proactive tools. When security operations are protected from malicious modifications, these protections must be protected, and so forth. Cybersecurity is not setting strictly enforceable, non-bypassable separation rules yet. But it could start creating proactive tools to simplify the defense of security measures.

Return

Q C7: Who is more exposed in a cyberwar?

Short A: … that is probably changing over time …

- Longer A: Once detected or revealed, vulnerabilities and malware can be removed relatively quickly. The problem is the uncertainty, covertness, and sneakiness of cyber weapons against vulnerable systems. Cyberwar actions are intended to be painful. The US government/CISA has named 16 infrastructure sectors as vital for national security: Chemical, Commercial Facilities, Communications, Critical Manufacturing, Dams, Defense Industrial Base, Emergency Services,

Energy, Financial Services, Food and Agriculture, Government Facilities, Healthcare/Public Health, Information Technology, Nuclear Reactors/Materials/Waste, Transportation, Water/Wastewater Systems. Actions by other nations against them could be considered an act of war. Computer systems are vulnerable, but it is still challenging to know what could be done to create harm. Reconnaissance and planning of cyberwar attacks are labor-intensive. However, these steps, including analysis, can be (fully) automated, and soon a new level of precision, intention, and proportionality can be realized with AI (for all war parties). If we are not doing anything, the cyber domain will become a battlefield with unpredictable outcomes that could be outside what is expected from conventional military force comparisons – just saying one word: logistics.

Return

Q C8: Will governments or the military push back on No-Go-*™?

Short A: Probably not

- Longer A: Governments and the military strategize from what is available to them. It's speculation, but they will likely accept that the cyber-domain has become irrelevant. However, they will not accept No-Go's claims at face value; instead, supported by cybersecurity, they will likely put a lot of R&D into finding new vulnerabilities; this would (clearly) strengthen No-Go-Security™. However, we should not forget that computer hardware is vulnerable to electromagnetic pulses.

Return

Q C9: Who may resist the changes from No-Go-*™?

Short A: … we will see …

- Longer A: This is almost a question for which we have no answer. We would like to assume that there is unanimous consent on ending cyberwar capabilities. But we should not bet on that. People's jobs or careers in activities, now prone to be automated, were started expecting job stability. However, threats to other vulnerabilities will give them opportunities to continue their careers.

Return

Q C10: What is the opinion of cybersecurity professionals about No-Go-Security™?

Short A: It's new, untested; but we got important hints (thanks)

- Longer A: For most IT professionals, No-Go is an untested idea. More importantly, the founder got valuable feedback on many details. However, security professionals acknowledge the flaws in how cybersecurity is done compared to other security sectors. Experts can speak for themselves about whether it is worth turning the No-Go-Security™ ideas into products.

Return

Q C11 Why don't you care about software vulnerabilities?

Short A: We care, but not as much when having No-Go-Security™

- Longer A: Vulnerabilities require a trigger and potentially a malicious follow-up to be dangerous. Even if content is a trigger, content alone does not create damage except if it is an exploit in the form of a simple backdoor within an existing app.
  An exploit is usually an app or a script; both are binary hashcoded and white-, gray- or blacklisted. If whitelisted, we would (temporarily and involuntarily) "ignore" the exploit. However, for the manufacturer/developer, this could have severe consequences for their reputation once the exploit is used maliciously. If the exploit is graylisted, some users will use it without knowing; still, they have decided to accept the risk from graylisted apps. Content Watchdog™ or Network Watchdog™ will be used to limit the damage. Users will receive additional tools from No-Go to detect and report malicious software behavior easily.

Return

Term clarifications

Q C12: Is promising no damage from malware a flawed statement

Short A: Not really. Damage is something that should be avoided/prevented

- Longer A: We define damage as harm (to something) by someone else that causes detectable devaluation, reduced usefulness, or broken normal functions. Because we have no threshold, every change could arguably lead to (some) damage. We acknowledge that argument, but damage from malware is more than minor; it is considered significant enough to be prevented by security measures; otherwise, it would not be called malware.

Return

Q C13: What is security in comparison to safety?

Short A: Both terms mean freedom from harm, threat, and danger

- Longer A: We use security and safety, often interchangeable: freedom from harm, threat, and danger. There is a difference between these terms. Security is related to group efforts to protect us. Safety relates more to the personal feeling of being free from harm/danger.

Return

Q C14: What is the difference between white- and graylisting?

Short A: Whitelisted are based on authorized data, graylisted on detected patterns and statistics

- Longer A: Binary hashcodes of apps, scripts, and data exchanged protocols must be white-, gray- or blacklisted. Blacklisted hashcodes have lost their right to be executed or used automatically. Hashcodes have additional data associated with them. If these additional data are authorized disclosures from software developers, manufacturers, or web-resource operators, then we call the hashcode whitelisted. The hashcodes are gray-listed if provided additional data were generated from algorithms or statistical inferences.

Return

Q C15: What are deviations or anomalies?

Short A: They are results that were not expected from disclosures, patterns, or predictions

- Longer A: We provide additional data for white- and gray-listed hashcodes (see previous answer). Watchdogs use these data to check if software activities are suspicious. The expectation is that every software operation does what software developers/manufacturers or web-resource operators have shared and nothing more. Any deviation is considered an anomaly or threat if that expectation is false. Activities of gray-listed apps or data exchange operations are modeled via patterns or predictive models. Surprises are handled similarly, and users are asked for their preferred response (ignore or cancel the operation). These responses are statistically analyzed and associated with auto-responses; users have the option to correct that later.

Return

Q C16: Why is proactive security so much better than reactive?

Short A: Proactive measures prevent damage early

- Longer A: In reactive security, previously unknown threats could create damage. If damage detection is sensitive enough, this new threat is then blacklisted. Software that does not fit an element on the blacklist is accepted and executed. In fairness, there are methods to get potential viruses or malware on the blacklist without having them commit damages. But a blacklist is not proactive despite additional features. If users start unknown code and the system pauses, e.g., for getting a confirmation, then we would have a proactive feature. The question is, is that effective

enough? I.e., is the same (unknown) software blocked in other circumstances before being executed? Could it be loaded into RAM, made executable in RAM, and then used covertly? Some of the answers depend on the used Operating system.
A bit more technical: What about loading harmless apps into RAM and having another app modify the binary executable in RAM by inserting some malicious attack code? A reactive solution tries to detect and remove an app doing these changes. Proactively, we would prevent code modification on executables. If changes are done in non-executable code, then we prevent that code can be made executable when in RAM. Instead, we would demand that modified code be stored first and reloaded to RAM as executable after hashcoded and checked to be white-, gray-, or blacklisted. These operations are relevant for developers; they could get special components that would allow them to do these operations overtly and not hidden as done by an attacker.
Partial, proactive features are not sufficient. No-Go-Malware™ uses whitelisting with cached hashcodes applied to every executable (including scripts/macros) before they can be executed. Also, we must prevent modifications to executables in RAM, a rule that must extend to CPU's cache.
What if code modification is part of software's regular operation mode? (see that answer).

Return

Q C17: Why is prevention in security important?

Short A: Prevention (in No-Go) expects damage and prepares us to deal with its consequences

- Longer A: Proactive security is already prevention. But if we mention prevention separately, we refer to features that help us when malicious software has fallen through the cracks and is about to create damage. In these situations, we want systems to prevent damage; this is No-Go's understanding of prevention.
  Separate prevention is important for two reasons. It is redundant in case primary protection fails against malware, and second, it detects failures to the main security measure independently.

Return

Q C18: What is the advantage of independent circuit breakers?

Short A: Preventing damage and gaining time for additional actions

- Longer A: Circuit breakers in software are currently useless. They would be part of the operating system; this means they would be deactivated or manipulated, or the attack would adapt to the underlying detection filter. E.g., stopping software on a blacklist is a (dependent) circuit breaker. Also, delaying further user login after inserting false passwords is one.
  But the problem with current security methods is the protection of security measures, which requires their own security protection measures, which might be vulnerable, etc. Circuit breakers require full independence; if this independence is breached in simple or single-purpose units, it can be detected much easier and reliably than in complex CPU/OS environments.

Return

Q C19: What does it mean: it is impossible?

Short A: Impossible is a strong prediction; it is dangerous to overgeneralize impossibility

- Longer A: From the laws of nature, we get impossibility statements. A person can not be in 2 different places physically simultaneously. What about mixed liquids? Can they be separated over time? Entropy cannot naturally decrease; it only increases with time. However, this is less convincing: two liquids, water and oil, separate over time naturally. What about water and alcohol? It depends on the temperature. Still, in chemistry, there are energetically impossible reactions. In

physics, chemistry, and math, we could make impossibility statements. In reverse, if it is not impossible, it is just a matter of engineering, and it can be done.

For computer science, it is much more difficult to make impossibility statements. If we had them and they were true, we could save money and time trying to develop impossible solutions. There are a few impossibility proofs (without naming concrete examples); their chosen model used in the proof already contained an impossibility of some kind. Is this a selection bias?

E.g., using the Turing model, the correctness of an algorithm cannot generally be proven (halting problem). But for a less general situation, it is possible to use algorithms to validate the correctness of an algorithm. Then, we could ask, what algorithms can we validate automatically? As a formal math problem, this question might not be answerable.

So, is damage elimination of malware of any kind impossible? Is it impossible to stop waging cyberwar via retrofitted legacy equipment? Could we give cyber defenders a sustainable advantage over attackers? Many have opinions based on experience with cybersecurity on these questions.

No-Go-*™ has changed a few operational paradigms and is now claiming it can be done.

Return

Our answers are not written in stone or intended to be our final word. If you disagree or think of a better answer, please don't hesitate to contact us with the question code reference within the subject line to **faqs at nogostar dot com**.

D. Questions with No Answer Yet

Q D1: What is your question?

Short A: Let us know

- A: Sorry, we don't know yet … so, please let us know: faqs at nogostar dot com.

Return