Unclassified

May 26, 2021

INCD Position Paper on standards and guidelines to enhance software supply chain cyber security

The Israeli National Cyber Directorate wishes to present this position paper re relevant NIST publication and Executive Order 14028. Most articles in this position paper relate to item 5 in NIST call for submissions- *Guidelines for software integrity chains and provenance.*

1. Existing common methodologies do not specify nor recognize "implants" as an attack vector that merits the development of methods to detect identify and mitigate. "Implants " means any malicious intervention in the code during the product life cycle (e.g. design or production or updates). For example, OWASP testing guide provides guidelines to write a secure code and to verify that the code correlates to the guidelines, however, the implant can use secure code methodology and by this would not be recognized by common security tools such as Static Application Security Testing (SAST). Another example would be security tools like the Software Composition Analysis (SCA) that is limited in detecting "implants". We claim that defining this attack vector better will accelerate the development of end user awareness and solutions for such malicious intervention.
2. "Inside-Out" Monitoring- existing common approach\paradigm is that cyber-attacks will be initiated from outside of the organization, so efforts are directed to detect and alert in that direction. As supply chain or implants attacks are becoming more common, malicious activities could spread from within the organization network by lateral movement and communicate with external Command and Controls (CnCs). To mitigate such threat, a continuous monitoring and holistic approach should be adopted rather than periodic checks of the security posture.
3. Common current legal arrangements rely on a Shared responsibility model where the supplier (a software vendor for example), the integrator and the customer share responsibility for cyber security. Such arrangements lead to a situation where the end user, the customer is not fully aware of his responsibilities. It would be advisable to consider including in the standard a recommendation to define clearly the responsibilities of each side of the contract. (It should be mentioned that there is a higher level of clarity when we look at the public cloud services arrangements).
4. Time to Rollback\Version forward - one of the incident response methods would be to rollback to the previous and safe version of the software (take Solar winds for example) or to upgrade to a newer and safer version. A crucial factor would be the time and complexity to return to the previous version or to upgrade to a newer one. Standards for secure software development should include a metrics that will relate to the Time to Rollback or Upgrade, and the metrics should be presented to the customers before-hand.
5. Advanced zero trust- Currently there are no common techniques to allow an organization to use untrusted software. The current zero trust approach rely on monitoring and controlling user and

computer behavior. It would be advisable to develop a more advanced model that would rely on techniques such as Moving target Defense (MTD) that allows end-users to "live with the enemy" (e.g. Symbiotic defense).

6. Use of Denial and Deception (D&D) as a cyber security means– it would be suggested to develop standards for effective D&D methodologies. By using integrated D&D capabilities with continuous monitoring approach, the organization would obtain a shorter Mean time to detect (MTTD) rate and more accurate situational awareness, and by this reducing the potential impact of cyber-attacks.
7. Nano-segmentation – The current segmentation methodologies and solutions do not provide a deep segmentation capability. By using Nano-segmentation principles, such as application process tracking and linking, organizations can enforce a stricter access policy. For instance, the firewall should enforce a strict Allow-list that is based on the following parameters:
   Source IP, Source Port Number, Source Username, Source Process (and its child process\threats, file dependencies such as DLL's etc.), Destination IP, Destination Port Number, Destination Username, Destination Process (and its child process\threats, file dependencies such as DLL's etc.), Time, Action (e.g. Allow, Block, Allow+ Log, Log Only, Reroute traffic).

INCD would be glad to join the relevant working groups.

This paper was written by Mr. Yuval Sinai and Mr. Yosi Aviram.
For further inquiries, please contact Ms. Gali Levakov, INCD Attache to the Israeli Embassy, Washington DC at Glevakov@cyber.gov.il