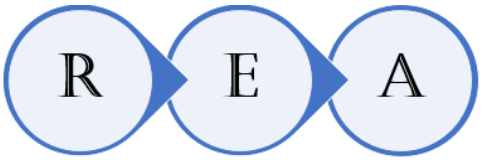

RELIABLE ENERGY ANALYTICS LLC

Testimony of Richard "Dick" Brooks

Co-Founder of Reliable Energy Analytics, LLC

Before the

Joint Committee on Advanced Information Technology, the Internet and Cybersecurity

September 8, 2021

On behalf of Reliable Energy Analytics, LLC (REA)[1], I want to thank the members of the Joint Committee on Advanced Information Technology, the Internet and Cybersecurity for the opportunity to present this testimony for your consideration.

REA is a small software company, co-founded by Dick and Joanne Brooks, located in Westfield, Massachusetts that specializes in NIST compliant Cyber Supply Chain Risk Management (C-SCRM)[2] software solutions for the Energy industry to meet NERC Cybersecurity requirements for software verification, specified in standard CIP-010-3 Part 1.6[3]. I have served in multiple standards development and leadership roles since 1995 within the North American Energy Standards Board (NAESB) in the development of Energy industry cybersecurity standards that have been adopted as FERC Regulations under Title 18 of the US Code of Federal Regulations. I currently serve as Vice Chairman, Executive Committee of the Wholesale Electric Quadrant (WEQ) within NAESB[4]. I am also actively participating in the development of industry standards within the Northeast Power Coordinating Council (NPCC) NERC Regional entity, Task Force on Infrastructure Security & Technology (*TFIST*)[5] and am an active participant in the DHS CISA ICT_SCRM Taskforce, Small and Medium Business Work Group[6] developing supply chain risk management guidance to support Executive Order 14028. For the past 12 months I have worked with the Department of Commerce NTIA Software Transparency initiative developing guidelines for the use of Software Bill of Materials (SBOM)[7]. I am the lead software engineer

---

[1] https://reliableenergyanalytics.com/

[2] https://csrc.nist.gov/publications/detail/sp/800-161/rev-1/draft

[3] https://www.nerc.com/_layouts/PrintStandard.aspx?standardnumber=CIP-010-3&title=Cyber%20Security%20%E2%80%94%20Configuration%20Change%20Management%20and%20Vulnerability%20Assessments&Jurisdiction=United%20States

[4] https://www.naesb.org/

[5] https://www.npcc.org/

[6] https://www.cisa.gov/ict-scrm-task-force

[7] https://www.ntia.gov/sbom

responsible for REA's flagship C-SCRM/SBOM risk assessment product, Software Assurance Guardian™ (SAG™) Point Man™ (SAG-PM ™)[8]. I have worked within the Energy industry in a software engineering/software architect capacity since 1990.

I'm providing this testimony today with two goals in mind: 1) Provide an understanding of the risks within the software supply chain[9], largely resulting from a lack of transparency into the ingredients that go into a software product and 2) Raise awareness as to the need for more effective software supply chain risk management controls to enable software consumers to verify the trustworthiness of a software package by verifying the trust relationship between a packaged software product, it's original software supplier and the parties that digitally sign software packages.[10]

Imagine the following scenario; you're sitting at a table with an un-labelled can in front of you, which you're expected to eat. Would you blindly open the un-labelled can and eat whatever it contains? I certainly hope not. But this happens every single day with software. A software consumer will download a software package from the Internet and install it in their digital ecosystems, without knowing what's in it or the risks that may be present. Today, we all know the risks that may be present in software, such as ransomware, illicit bitcoin mining, viruses and many other cyber attacks that arrive through the installation of a software product. New controls are needed to help software consumers, **proactively**, understand the risks and dangers within a software package before any attempt to install it in a digital ecosystem. Once bad software has been installed it can go about causing harm that can make for a very unpleasant experience for the victims. The most effective solution to stopping malware from causing harm is to prevent it from being installed – FULL STOP. Fortunately, a solution to this problem is now available, with help from a Software Bill of Materials (SBOM).

Adding an SBOM to software is equivalent to "adding a label" to the un-labelled can in the earlier metaphor. An SBOM provides a software consumer with vitally important information about what's in a software product, including the list of software components that were used in the construction of the software, many of which are open source, and the original supplier's identity of the software product and each of the components that are included in the product. This enables a software consumer to proactively conduct a software supply chain risk assessment on a software package to determine the trustworthiness of the software product, each of its embedded components and the parties that developed each software component contained in the product. A software consumer can use the vendor supplied SBOM as input to a cyber supply chain risk management control that follows NIST Cybersecurity (C-SCRM) Guidelines to proactively determine the trustworthiness of a software package and identify any risks that may be present, prior to any attempt at installation.

---

[8] https://www.einpresswire.com/article/549494383/sag-pm-version-1-1-3-simplifies-ntia-sbom-creation-and-consumption-for-both-software-vendors-and-consumers

[9] https://energycentral.com/c/ec/we-cannot-secure-software-supply-chain-without-sbom

[10] https://energycentral.com/c/ec/who-ya-gonna-trust

Simply stated, we cannot stop ransomware and secure the software supply chain without an SBOM as input to an effective Cyber Supply Chain Risk Management control that follows NIST C-SCRM guidelines.

Software objects that are not digitally signed, are simply not trustworthy. Therefore, it is imperative that each software package, that a software consumer wishes to install, must be digitally signed, using a trusted digital certificate, issued by a legitimate, recognized Certificate Authority. A software package is, frequently, developed by one party, i.e., the original software supplier, and is digitally signed by another party, whose name and identity may be different from that of the original supplier of the software. Current code signing practices allow any party to digitally sign a software package, without authorization by the original software supplier. This presents a dilemma for software consumers because there is no defined standard, method which a consumer can use to verify the trust relationship between a software supplier and the party that digitally signed a software package.

Imagine a scenario where two parties, A and B, engage in a contract and each one applies their own signature. Now imagine that a third party, C, comes along and replaces the signature applied by B, without B or A's knowledge, would the contract still be valid? And, party A has no way to verify which party, B or C, is authorized to sign. This is the problem with today's digital signature practices for code signing – a consumer has no effective, standard method to verify the trust relationship between an original software supplier and a code signing party on a digitally signed software package. Parties that rely on code signing practices as a means to identify the original software supplier, i.e., NERC/NATF software verification guidance, are aware of this flaw. Certificate Authorities have begun to address this problem by proposing a new code signing standard via the CAB Forum industry organization. Software consumers currently rely on proprietary methods to verify the trust relationship between a software supplier and a digital signer, which can be a highly manual process with substantial opportunity for human error to occur.

A nationwide standard that would allow a software consumer to verify the trust relationship between an original software supplier of a software product and the party that has been authorized to digitally sign a software product would provide a higher degree of trust within the software supply chain and, hopefully, eliminate the potential for human error to occur during verification.

In summary, ransomware and other forms of malware can be identified and stopped, proactively, through the transparency provided by an SBOM within a NIST C-SCRM best practice solution. Local and State government entities could improve their own software supply chain protections by requiring software vendors to supply SBOM's with their software products, as a first step. A nationwide, standard method which a software consumer can use to verify the trust relationship between an original software supplier and the party that has been authorized by the supplier to sign their software products is needed to address a known issue with digital code signing practices that exists today.

This concludes my testimony.