Dear Sean,
Here is the copy of the Project Pitch with reference number : **00061602** submitted to the **Cybersecurity and Authentication (CA)** on **5/18/2023**.

1. Submitter Name

   Sean Brennan

2. Submitter Email

   sean.brennan@tekfive.com

3. Submitter Phone

   5059011074

4. Company Name

   TekFive, Inc.

5. State

   AL

6. Zip Code

   35805

7. Corporate Website

   tekfive.com

8. SBIR/STTR topic that best fits your projects technology area

   Cybersecurity and Authentication (CA)

9. Is this Project Pitch for a technology or project concept that was previously submitted as a full proposal by your company to the NSF SBIR/STTR Phase I Program – and was not awarded ?

   No

10. Has your company received a prior NSF SBIR or STTR award?

    No

11. Does your company currently have a full Phase I SBIR or STTR proposal under review at NSF?

    No

12. Briefly Describe the Technology Innovation?

AutonomousTrust is TekFive's high-trust cooperative computing concept -- a data messaging framework that allows for dynamic composibility, requesting and serving encrypted data only with trusted peers and only to the extent of that fine-grained trust. Said trust is dynamically evaluated in real time to rapidly eliminate incoming threats and even reclassify existing peers as their behavior changes and thus protect resources. Increased risk requires a greater trust threshold. An autonomous agent using this framework can adaptively: 1) refuse communications from severely untrusted peers, conserving bandwidth; 2) communicate with but refuse computation services to faintly trusted peers, protecting CPU time; 3) offer services but refuse data-sharing to moderately trusted peers, protecting data; and 4) offer data-sharing to well trusted peers; all with a configurable gradient of access at every level, and all within the same application. In more concrete terms, AutonomousTrust is an operations framework for a vast distributed system that dynamically composes numerous individual microservices into a coherent application on-demand -- with security at its core. We follow the Unix philosophy: do one thing well, work together, use a universal (text) interface; yet implemented such that each microservice can choose its level of participation. Our system presumes that the best viewpoint of security is local, situational and ever-changing, thereby requiring decentralized control, detection and response. As such, our system replicates certain aspects of human social behavior in seven key facets. Firstly, we limit communication to direct contact among peers within a resource-based hierarchy. Second, we utilize a domain-specific, extensible markup language to define well-constrained APIs. We use a strict negotiation protocol to dynamically mediate peer interactions. We provide for consistent, immutable identity with a crypto-ledger --- shared among cohorts only. This allows reputation scoring derived from shared transaction scores. Reputation is bootstrapped through game-theoretic strategies. Lastly, the opposing strategies of paranoid security and vulnerable opportunity are dynamically self-balanced. In combination, these facets are the backbone of a software ecosystem supporting a community of autonomous agents which perform decentralized computing in a way that minimizes security risks and optimizes cooperation. We strongly restrict the capabilities of any one node of the system; no general computing here. But, through micro-service composability, the system as a whole is generally capable yet highly secure because the interfaces are very specific and continuously, autonomously monitored. AutonomousTrust is tangentially related to advances in Internet-of-Things (IoT) and machine-to-machine (M2M) messaging technologies, but explicitly addresses a prevalent lack of security in those paradigms, as well as cybersecurity shortcomings in general. This concept is also a highly-distributed answer to the extreme centralization of Cloud computing, in which we perceive foundational vulnerabilities. Although the Web3 principles of decentralization and knowledge-proofing is incorporated here, our concept is a concrete step beyond, emphasizing blockchain's roots in distributed consensus. Our intended realization of this project into a product would be no mere incremental step, but a sharp departure from current computation practices, much like Cloud computing was before it.

13. Briefly Describe the Technical Objectives and Challenges?

This proof-of-concept project will necessarily require some engineering development to rough-out the operational viability of the above mechanisms, but primarily it will research and explore the effectiveness of the concept itself. To this end, we will study three things: 1) the structure, efficiency and resiliency of self-formed network topologies, 2) effective patterns of negotiation and reputation-building at scale, and 3) the resistance of the network at each level to failure and attack. This last item is our primary metric for success, ultimately comparing against the 2019--2020 SunBurst supply-chain attack on the SolarWinds Orion product. We will intentionally include and then exploit several classes of known CVE vulnerabilities against individual agents, run established attacks such as DoS, 51% and Sybil against our blockchains, and conduct system-specific attacks. We have identified five attack vectors specific to our system: 1) service deception (falsifying results), 2) trust betrayal (building up reputation then attacking), 3) reputation trashing (colluding to ruin a target), 4) atomization (collusion to isolate a target), and 5) authority corruption (using one's position in the hierarchy to isolate or mislead a group of targets). Additional possible attacks may be discovered as development proceeds. Statistical methods will be used to determine our effectiveness compared to both naïve and state-of-the-art implementations. The strongest challenge for our system lies in network bootstrapping and insertion of valid new nodes. While nominal communications between nodes are fully encrypted, initial contact cannot be. We anticipate using specialized nodes -- more capable and hardened -- to act as diplomats to vet and shepherd new nodes, as a protective ring around the core of the system. These diplomatic nodes will have separate open and encrypted channels, to interface with either side of the barrier, but to an inevitable extent the open channel presents an attack vector. We will have to pay particular attention to the potential vulnerabilities in this outward-facing ring. Studying this aspect will begin at the outset of the project because this is also how the network as a whole is created from scratch. That will also help ensure our emphasis is on testing security from step one. Our mature product will likely be written in C, C++, or Rust for performance and deployed in containers or virtual machines, but at this stage we will be using Python for rapid prototyping and little or no sandboxing. We will develop a testbed environment that will enable network behavior visualization, injection of errors and attacks, and general debugging. This tool can later be used as a development environment for rapidly prototyping instances of the ultimate product as well.

14. Briefly Describe the Market Opportunity?

Although potentially widely applicable, this concept is perfect for coordination in hostile environments and contentious inter-organizational data sharing. For USDOD customers, this product could be a key element for reliable sixth generation warfare (6GW) [non-contact] or 7GW [total automation]. Numerous U.S. federal agencies are informationally siloed from one another, either intentionally or through a combination of competition for funding, conflicting expertise, and overlapping domain authority. These structural barriers are highly detrimental to data sharing and collaboration which is often crucial to mission objectives. The private sector is no less contentious and under-threat, yet commerce requires some level of sharing. Analysis of the SunBurst attack suggests that, aside from the initial SolarWinds network breach, state-of-the-art cybersecurity techniques and tools would have been ineffective at preventing or even detecting this hack. Once inside the network, attacker activity would have been indistinguishable from that of valid developers, and client companies would have no reason to cut-off Orion's access to the Internet. In short, while Zero-Trust would have likely prevented the initial network hack, neither it nor most intrusion detection systems could have halted the rest of this attack. As we will show in this project, AutonomousTrust makes this attack impossible. Using our system, even code developers would not have direct access to the build system -- eliminating the binary injection. Clients that used an AutonomousTrust-based product would immediately know if that product was violating the scope of its specification, when the Trojan Horse either phones home or tries to control the client system.

15. Briefly Describe the Company and Team?

Founded in 2007, TekFive is a veteran-owned, veteran centric, innovative and agile services company. We provide comprehensive federal IT domain experience, while offering the latest commercial industry insight, a highly motivated full-stack development team and a network of successful industry partners to rapidly respond to your enterprise IT challenges. Sean Brennan is the principal investigator for this effort. He has over 20 years of research and engineering experience, specializing in large-scale, resource-constrained networks and software correctness. He has been involved in 12 DOE-funded projects related to nuclear nonproliferation, and was the technical lead for five of these. Dr. Brennan received a Ph.D. in Computer Science from the University of New Mexico, and is an author on 15 published papers. Corey Baswell is TekFive's Chief Technology Officer. He has over 20 years of enterprise IT experience as a full stack engineer in the areas of application development, enterprise architecture, and platform development. He has developed numerous enterprise software services including the NASA OpenESB, the NASA and VA DevSecOps pipeline, the NASA and VA Pulse app analytics, and the NASA Application Portfolio Management tool. Mr. Baswell received a Bachelor of Science degree in Computer Engineering from Auburn University where he graduated summa cum laude.

16. How did you first hear about our program?

My network (personal or professional contact sent information)

NSF SBIR/STTR Phase I Eligibility Information:

In addition to receiving an invitation to submit a full proposal from the NSF SBIR/STTR Phase I Program based upon the review of their submitted Project Pitch,potential proposers to the program must also qualify as a small business concern to participate in the program (see SBIR/STTR Eligibility Guidefor more information).

The firm must be in compliance with the SBIR/STTR Policy Directive(s) and the Code of Federal Regulations (13 CFR 121).

- Your company must be a small business (fewer than 500 employees) located in the United States. Please note that the size limit of 500 employees includes affiliates.
- At least 50% of your company’s equity must be owned by U.S. citizens or permanent residents, and all funded work needs to take place in the United States (including work done by consultants and contractors).
- Primary employment is defined as at least 51 percent employed by the small business. NSF normally considers a full-time work week to be 40 hours and considers employment elsewhere of greater than 19.6 hours per week to be in conflict with this requirement.
- The Principal Investigator needs to commit to at least one month (173 hours) of effort to the funded project, per six months of project duration.

For more detailed information, please refer to the SBIR/STTR Eligibility Guide by using https://www.sbir.gov/sites/default/files/elig_size_compliance_guide.pdf. Please note that these requirements need to be satisfied at the time an SBIR/STTR award is made, and not necessarily when the proposal is submitted.