Zero-Knowledge Proofs: Detecting Systemic Cyber Vulnerabilities Among Critical Infrastructure Operators

By Georgianna Shea, Cason Smith, and Haden Snyder

Executive Summary

When Colonial Pipeline suffered a ransomware attack in May 2021, there was no sector-wide visibility into which other pipeline operators faced the same architectural vulnerability that forced a six-day shutdown and disrupted fuel supplies across the southeastern United States.[1] When Russian spies corrupted the SolarWinds software update mechanism in 2020, establishing the full scope of the intrusion took months because no mechanism existed for rapidly determining which agencies and companies were running the affected product.[2] These were not failures of individual organizations. They were failures of systemic visibility, the inability to see, at sector scale, which shared software dependencies were creating shared exposure. This paper addresses that problem.

Since the Obama administration's 2013 Executive Order on Improving Critical Infrastructure Cybersecurity, U.S. policy has treated cyber threat information-sharing between critical infrastructure operators and the federal government as a national security imperative. That effort has focused on sharing threat indicators, incident reports, and defensive measures, but not vulnerability data. The proactive disclosure of key vulnerability indicators — such as scan results, asset inventories, and system configuration details — has remained almost entirely outside what existing frameworks require or incentivize. The need to share data has driven the private creation of Information Sharing and Analysis Centers (ISACs) for each sector of U.S. critical infrastructure. It has also been responsible for the liability protections in the Cybersecurity Information Sharing Act of 2015, voluntary reporting programs, and government and sector coordinating councils. All were built on the expectation that operators would disclose what they know about vulnerabilities in their systems.[3] Each framework has arrived more detailed than the last, and yet companies are still hesitant to share sensitive security information with the government.

The reason is structural. A scan log shared with a federal agency can be breached, subpoenaed, or used in a regulatory proceeding that the operator never anticipated. Legal protection for the act of sharing is not the same as protection from the consequences of the data existing somewhere outside the operator's control. No voluntary framework has resolved that asymmetry, and the result is a decade of frameworks with thin participation.

Zero-knowledge proofs (ZKPs) are a technically credible alternative path. A ZKP allows an operator to prove that a defined statement is true — for example, that a specific vulnerability is or is not present in its systems — without transmitting sensitive underlying data. The proof object that travels to a government verifier is a compact cryptographic artifact, essentially a mathematical certificate that the agreed computation was performed correctly without revealing the underlying data. It is typically just a few kilobytes in size, regardless of how much underlying data it attests to or how complex the computation behind it was, while the data itself stays put.[4]

1. Cybersecurity and Infrastructure Security Agency Director Jen Easterly, "Colonial Pipeline Cyber Incident: What We've Learned & What We've Done Over the Past Two Years," May 7, 2023. (https://www.cisa.gov/news-events/news/attack-colonial-pipeline-what-weve-learned-what-weve-done-over-past-two-years)

2. Cybersecurity and Infrastructure Security Agency, "Alert AA20-352A: Advanced Persistent Threat Compromise of Government Agencies, Critical Infrastructure, and Private Sector Organizations," December 17, 2020. (https://www.cisa.gov/news-events/cybersecurity-advisories/aa20-352a); U.S. Government Accountability Office, "SolarWinds Cyberattack Demands Significant Federal and Private-Sector Response (infographic)," April 22, 2021. (https://www.gao.gov/blog/solarwinds-cyberattack-demands-significant-federal-and-private-sector-response-infographic)

3. See, for example: Cybersecurity Information Sharing Act of 2015, Pub. L. 114-113, Division N (codified at 6 U.S.C. §§ 1501–1510), which established the legal framework for voluntary sharing of cyber threat indicators and defensive measures between private entities and the federal government. The ISAC model and its voluntary disclosure premise are documented in Presidential Policy Directive 21 (PPD-21), Critical Infrastructure Security and Resilience (February 12, 2013), and the National Infrastructure Protection Plan (NIPP) 2013: Partnering for Critical Infrastructure Security and Resilience, Department of Homeland Security (2013), available at https://www.cisa.gov/sites/default/files/2022-11/national-infrastructure-protection-plan-2013-508.pdf.

4. The foundational academic treatment of ZKPs is Shafi Goldwasser, Silvio Micali, and Charles Rackoff, "The Knowledge Complexity of Interactive Proof Systems," SIAM Journal on Computing, Vol. 18, No. 1 (1989), pages 186-208. For a practitioner-oriented overview of ZKP constructions and their properties, see: Justin Thaler, "Proofs, Arguments, and Zero-Knowledge," July 18, 2023. (https://people.cs.georgetown.edu/jthaler/ProofsArgsAndZK.pdf). For an application of ZKPs to private computation and verifiable attestation, see Sean Bowe, Alessandro Chiesa, Matthew Green, Ian Miers, Pratyush Mishra, and Howard Wu, "ZEXE: Enabling Decentralized Private Computation," *IEEE Symposium on Security and Privacy*, 2020, pages 947-964.

Zero-Knowledge Proofs: Detecting Systemic Cyber Vulnerabilities Among Critical Infrastructure Operators

FDD's Transformative Cyber Innovation Lab (TCIL) partnered with VIA, a web3 technology company, to test this approach in a structured pilot demonstration focused on the energy sector. VIA built and operated the proof system. TCIL conducted the analysis independently and does not endorse VIA as a vendor. The pilot modeled three oil and natural gas companies providing a government partner — in this case, a notional national critical infrastructure observatory — with information about 38 real, known vulnerabilities (common vulnerabilities and exposures, or CVEs) found in similar networks. Every provider had confirmed exposure. Not a single CVE was remediated universally across all three.

The pilot's central demonstration was not the vulnerability findings themselves but rather that ZKPs made the analysis possible at all. Each of the three providers submitted cryptographic proof objects generated locally from their own scan data. No scan logs, asset inventories, or configuration details left any provider's environment. The government received only proof objects and yes/no indicators, verified them, and aggregated the results across all three providers. The output of that process could only have overwise been produced if operators handed over exactly the data they are reticent to share.

Because ZKPs enabled cross-operator aggregation without raw data disclosure, the national observatory could see what no single operator and no current sharing framework can see. It found sector-wide, structural challenges. And it found that each provider's gap profile, while different in its specifics, followed the same structural cause: companies deferred and delayed patching over concerns about operational downtime. If the analysis had occurred in the real world rather than in a pilot, the national observatory's findings could have translated into actionable guidance and government policy changes directly connected to problems that enable cybersecurity risks to persist.

This paper recommends four actions to build on what the pilot demonstrated. First, the National Institute of Standards and Technology (NIST) and CISA should establish the rules of the road for ZKP deployment before agencies begin accepting proof-based attestations. Second, the Cybersecurity and Infrastructure Security Agency (CISA) and the North American Electric Reliability Corporation (NERC) should run structured pilots accepting ZKP attestations as valid compliance evidence under existing regulatory authority. Third, federal procurement rules should require demonstrated ZKP capability as a condition of contract eligibility for vendors supplying federal agencies and federally contracted critical infrastructure operators. And fourth, the United States should engage key partners and allies to develop compatible governance frameworks before domestic standards are finalized.

National Cyber Resilience Demands Greater Visibility Into Risk; Existing Frameworks Fall Short

In February 2024, the cyber-physical resilience working group of the President's Council of Advisors on Science and Technology (PCAST) warned, "Our adversaries understand our infrastructure and its dependencies and weaknesses better than we collectively do."[5] To change this dynamic, the PCAST urged the creation of a National Critical Infrastructure Observatory to ensure greater visibility into systemic and cross-sector risks. Under existing regulatory and voluntary frameworks, however, the private sector is unlikely to share the data necessary to inform a national observatory.

Systemic threats to critical infrastructure move across sector boundaries through shared software dependencies, and no single operator can see that movement from inside its own perimeter. The question a national observatory would answer is whether shared dependencies create sector-wide exposure that warrants coordinated action. That capability might have prevented — or at least mitigated — the Colonial Pipeline and SolarWinds attacks. Many other examples illustrate this point.

5. Executive Office of the President, President's Council of Advisors on Science and Technology (PCAST), "Report to the President, Strategy for Cyber-Physical Resilience: Fortifying Our Critical Infrastructure for a Digital World," February 2024. (https://bidenwhitehouse.archives.gov/wp-content/uploads/2024/02/PCAST_Cyber-Physical-Resilience-Report_Feb2024.pdf)

Zero-Knowledge Proofs: Detecting Systemic Cyber Vulnerabilities Among Critical Infrastructure Operators

Later in 2021, after the Colonial Pipeline hack, when researchers discovered a vulnerability in a widely used software component called Log4j, the absence of sector-level visibility into its prevalence slowed both the assessment of exposure and the allocation of response resources.[6]

Until that visibility improves, the response to the next major incident will stumble on the same challenges as the last. What a national observatory would provide is not a compliance-monitoring function over individual operators. Instead, it would provide the sector-level signal indicating which shared dependencies are creating systemic exposure, which vulnerabilities warrant coordinated remediation guidance, and where government resources and vendor engagement would have the highest leverage. That analytical function requires cross-sector data that no single operator can supply.

Existing frameworks cannot supply the data a national observatory needs. ISACs distribute threat intelligence within sectors through voluntary disclosure of incident data, anomalous activity reports, and indicators of compromise. Each participating operator takes a lopsided deal: share information that could trigger regulatory scrutiny or give a competitor useful intelligence in exchange for collective security benefits that are real but nearly impossible to quantify. In 2015, Congress passed the Cybersecurity Information Sharing Act to reduce legal exposure for sharing cyber threat indicators and defensive measures.[7] Information a private company shares with the government about a cyber incident and successful mitigation strategies is protected against Freedom of Information Act (FOIA) requests and regulatory scrutiny. An operator reading the statute carefully, however, finds that its raw vulnerability data, scan results, asset inventories, and configuration details sit outside protection coverage. More importantly, the statute does not control what happens to shared data after a company shares it. The law reduces liability for sharing cyber threat indicators, but it does not touch most of what operators worry about after disclosure.[8]

For example, an energy company that proactively shares scan results with CISA has no statutory guarantee that the usage of those results stays confined to the purpose for which they were shared. A NERC auditor, through interagency information flows or a routine data request, may obtain or learn of such results. Companies that voluntarily disclose their data create multiple types of risk for themselves. For example, energy companies are required under NERC Critical Infrastructure Protection (CIP) standards to patch within defined timeframes; disclosing their data might draw attention to their failure to do so. Understandably, many companies choose the safety of holding on to their data.

Anonymized aggregation is intended to help with this problem but does not solve it entirely for risk-conscious operators. It removes identifiers from operational data before analysis but runs into the same first-step problem: the operator has to disclose the data before anonymization can happen, and that disclosure step is precisely what most operators are unwilling to take. In sectors with small and well-known populations of major operators, reidentification from operational data is a real risk.[9]

The Cyber Incident Reporting for Critical Infrastructure Act of 2022 introduced mandatory reporting requirements for critical infrastructure owners to report cyber incidents after they occur.[10] However, there is no requirement to proactively disclose vulnerability data, asset inventories, or system configuration details that would enable the analysis of

6. Cybersecurity and Infrastructure Security Agency, "Apache Log4j Vulnerability Guidance," April 8, 2022. (https://www.cisa.gov/news-events/news/apache-log4j-vulnerability-guidance)

7. Cybersecurity Information Sharing Act of 2015, 6 U.S.C. § 1505. (https://uscode.house.gov/view.xhtml?req=granuleid:USC-prelim-title6-section1505&num=0&edition=prelim)

8. Cybersecurity Information Sharing Act of 2015, Pub. L. 114-113, Division N, 6 U.S.C. § 1505. The statute's information-sharing provisions sunsetted on September 30, 2025, and were subsequently extended through September 30, 2026.

9. Latanya Sweeney, "k-Anonymity: A Model for Protecting Privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 10, No. 5, 2002, pages 557-570.

10. Cyber Incident Reporting for Critical Infrastructure Act of 2022, Pub. L. 117-103, Division Y, 6 U.S.C. §§ 681-681g. CISA's implementing rulemaking was ongoing at the time of this report. CIRCIA requires covered entities to report substantial cyber incidents within 72 hours and ransomware payments within 24 hours; it does not require proactive disclosure of vulnerability scan data or system configuration information.

the U.S. critical infrastructure sector's current security posture. Besides incident reporting, participation in vulnerability sharing remains voluntary.

Meanwhile, Software Bills of Materials (SBOM) frameworks have genuinely advanced supply chain transparency, but not at the system level. Software companies provide their customers with details about software components and associated CVEs, but no entity aggregates the data to assess how broadly particular software is used across critical infrastructure sectors. Even if critical infrastructure operators provided a national observatory with copies of the SBOMs they receive from vendors, this would be a point-in-time snapshot, not a continuous signal. A national observatory would need regularly updated SBOMs, a proposition that is neither cost-effective nor realistic at sector scale, lest its analysis rely on out-of-date data.

The obstacle is not that operators and government agencies lack the authority or the organizational structures to share vulnerability data. It is that sharing creates risks that operators bear and the government does not. A scan log shared with a federal agency can be breached from that agency's systems, subpoenaed in litigation in which the operator had no part, or used to support a regulatory action the operator had no reason to anticipate. Building more institutions and issuing more guidance does not eliminate those risks. It leaves them exactly where they are — on the operator's side of the transaction.

Expanding the statutory protection to cover vulnerability data would reduce one dimension of that risk, but it cannot solve the deeper problem: no statute can guarantee that data shared with a federal agency stays within that agency, remains secure against breach, or is not obtained through interagency information flows by a regulator with enforcement authority over the same operator. Legal protection for the act of sharing is not the same as protection from the consequences of data that exists somewhere outside the operator's control. Operators who have concluded that no statutory protection could adequately cover those downstream risks are reading the situation correctly. ZKPs address the problem at its root by eliminating the disclosure step entirely; the data never leaves the operator's environment, so there is nothing to breach, subpoena, or redirect.

Zero-Knowledge Proofs: How the Technology Works

To understand how ZKPs reduce the risk of sharing vulnerability data, consider a hospital that needs to prove its patient records meet a privacy requirement without exposing the records themselves. The hospital would not simply tell an auditor, "Trust us, they are compliant." Instead, the hospital would use an agreed-upon verification process that checks whether the records satisfy the rule and produces proof of that result without revealing the records. The verifier does not have to trust the hospital's judgment or inspect the underlying files; it only has to trust that the proof was generated under the agreed rules and that the proof itself validates successfully. By using a ZKP, a pipeline operator could attest to similar facts — whether specific vulnerabilities are present or absent — and the government would receive only that answer, not the underlying information.

Consider a hypothetical case in which the government wants the answer to one question: does an operator's network contain CVE-2025-4802, a known flaw in a core Linux system library called "glibc" that allows an attacker to execute malicious code by manipulating how the system loads software at runtime? To answer that question without ZKPs, an operator would have to hand over a scan report. That report lists every device on the network that was checked, the software installed on each one, the version numbers of every library and application, and which of those versions are known to contain vulnerabilities. It is, in effect, a detailed map of the operator's attack surface, exactly the document an adversary would want and exactly what operators will not share.

With ZKPs, the scan still runs internally in exactly the same way. The operator's scanning tool sweeps the network, identifies every host, checks the software installed on each one against a database of known vulnerabilities, and records its findings. The scan output, in raw form, might show dozens of servers, their IP addresses, their operating system

versions, the exact glibc build installed on each, and a flag indicating which ones carry the vulnerable version. All of that data stays inside the operator's environment. What enters the ZKP process is the scan data as private input and the answer to the single authorized question: is CVE-2025-4802 present anywhere in the defined scope, yes or no? What the ZKP guarantees, provided the scan data is complete and truthful, is that the answer is honestly computed from the scan data and is not simply made up.

When employing the ZKP, the operator feeds the scan output into a proof circuit, which is a set of mathematical rules that hides the exact question the observatory is asking as well as the response given. The circuit generates a proof object, a cryptographic artifact, which the operator then sends to the government, or in the hypothetical scenario, the national observatory. The observatory applies a verification key to it, and the check either passes or fails. The observatory learns the answer to the defined question, and only that.

How Policy Intent Becomes Computable

Employing a ZKP is effective because a computer does not read a scan the way a person does. A security analyst might look through the report and evaluate whether the vulnerable software is present. A proof circuit does something more mechanical. It turns the government's question into a series of mathematical checks that represent the steps needed to answer it.

In this example, the proof circuit checks whether the scan data includes software matching the vulnerable version of CVE-2025-4802. Each check covers one small part of the computation. Taken together, they show that the reported answer was computed from the scan data in the way the circuit required and that the final yes-or-no answer is correct.

If all the checks are satisfied, the system produces a valid proof. If the operator does not alter the scan data inside the proof process, the circuit checks whether the submitted answer matches the data provided to it. If the data shows the vulnerability is present, the circuit will not produce a valid proof for an answer saying it is absent. The government never sees the scan report or the operational details behind it. It only verifies that the proof satisfies the rules built into the circuit and that the answer was computed correctly from the data provided.

A ZKP does have limits. It can only prove the statement encoded in the circuit. If the government's requirement asks a clear yes-or-no question, such as whether a specific CVE is present in a defined scan scope, the proof can answer that question precisely. But if the requirement is ambiguous, the circuit will reflect that ambiguity. For example, a requirement to prove "no high-risk vulnerabilities" is unclear unless "high-risk," "vulnerability," and "scope" are defined. ZKPs do not fix unclear requirements. They make clear requirements enforceable without requiring operators to disclose the sensitive data behind the answer.

The Formal Properties

A zero-knowledge proof is simply the formal name for what is illustrated above: a cryptographic method by which one party (the prover) demonstrates to another party (the verifier) that a specific statement is true without sharing any information beyond the truth of the statement itself.[11] Three mathematical properties working together make this possible.

The first is completeness. If the statement is true and the operator runs the protocol honestly, the proof passes. An operator with no exposure to a given CVE will always be able to prove it.

The second is soundness. If the statement is false, no operator, honest or dishonest, can produce a proof that passes verification, except with probability so small as to be operationally meaningless. In well-designed production systems,

11. Oded Goldreich, Silvio Micali, and Avi Wigderson, "Proofs that Yield Nothing But Their Validity, or All Languages in NP Have Zero-Knowledge Proof Systems," *Journal of the ACM*, Vol. 38, No. 1, 1991, pages 690-728.

this error probability is set well below 2^{-128}. In practical terms, this means an operator cannot falsify the claimed computation over the submitted inputs, though as the governance section below explains, what those inputs cover remains a separate question. A vulnerable system cannot produce a passing proof.

The third is zero-knowledge. A passing proof reveals nothing beyond the fact of its truth. The verifier, in this case the government observatory, has no path back to the private inputs. It cannot reconstruct the scan data, the asset inventory, or the software versions from the proof object. It learns only the answer to the question asked.

The completeness property means honest operators always get credit. The soundness property means dishonest operators cannot fake it. The zero-knowledge property means the government cannot fish for more than it requested.[12]

The three formal properties guarantee that the proof cannot be faked. They do not, however, guarantee that the proof covers everything it should. A proof cannot fix the problem of an operator excluding data or systems from its scan.

For example, a proof may validly show that the operator checked for a specific CVE in a specific set of systems during a specific reporting window. But if the operator excluded parts of the network from the scan, the proof may still pass while leaving important systems outside the check. The cryptography can confirm that the agreed rule was applied correctly to the submitted scan data. It cannot, by itself, confirm that the operator selected the right systems, scanned the full environment, used the correct tool settings, or relied on the most current vulnerability data.

Therefore, without additional governance, ZKP-based attestation amounts to cryptographically certified self-attestation. The math is rigorous, but cryptography alone cannot solve all problems. Ensuring that operators include all systems in a scan, for example, would require the responsible oversight body to define which systems have to be included in the scan and which exclusions require documented justification. Additional questions dealing with verified organizational identity, chain of custody, verified source, and other relevant issues require independent review.

The Pilot Demonstration

Design and Scope

TCIL and VIA designed the pilot around a specific question: can an observatory-style function generate actionable sector-level intelligence about systemic cyber risk without ingesting raw operational data from the providers it draws on?

To answer this question, VIA built and operated the cryptographic proof system. TCIL conducted the analysis and drew its conclusions without direction from VIA. VIA drew on existing vulnerability scan data from three real operational environments in its customer base — data collected in the ordinary course of VIA's operational practice — and not scans conducted specifically for this pilot. VIA partitioned and labeled the scans as Provider A, Provider B, and Provider C, all of which were anonymized. TCIL played the role of the national observatory.

ZKPs are already in wide use for purposes of banking fraud detection, digital identity verification, and validating cryptocurrency transactions, so VIA did not build its proof system from scratch or use unproven technology. What the pilot required was adapting that existing capability to a new problem. VIA's engineering team took the list of CVEs TCIL defined, configured the proof circuit to ask a yes-or-no question about each one, connected the proof system to real vulnerability scan data from three operational environments, and set up the process to transmit verified proof objects to TCIL for aggregation. The work took several weeks and required dedicated engineering staff. It was not trivial, but it was not experimental either. It was a practical application of a mature technology to a domain in which it had not been used before.

12. Shafi Goldwasser, Silvio Micali, and Charles Rackoff, "The Knowledge Complexity of Interactive Proof Systems," *SIAM Journal on Computing*, Vol. 18, No. 1, 1989, pages 186-208. Original conference presentation at STOC 1985.

Zero-Knowledge Proofs: Detecting Systemic Cyber Vulnerabilities Among Critical Infrastructure Operators

The CVE identifiers are real, and the vulnerability findings reflect genuine production system exposure. The findings are cryptographically verified in the sense that the proofs ran correctly over real scan data. They are illustrative in the sense that three providers are too few to support statistically significant conclusions about the oil and natural gas sector or U.S. critical infrastructure more broadly.[13]

CVEs were the natural choice for the attestation unit: structured, globally standardized, and widely used across the patch and vulnerability management processes that underpin compliance obligations for oil and natural gas pipeline operators. This includes the vulnerability assessment and remediation requirements in the Transportation Security Administration's (TSA's) pipeline security directives from 2021 and 2022.[14] Each CVE supports a binary claim, present or absent, remediated within the window or not, that maps directly into the proof circuit's logic.

Proof System Architecture and Data Flow

Playing the role of the observatory, TCIL finalized the selection of CVEs, severity thresholds, and the time period for proof submissions. VIA initially identified the candidate CVE set from publicly disclosed vulnerabilities relevant to energy sector operational technology (OT) environments. The severity thresholds served as scoping criteria: they defined which CVEs fell within the required reporting universe based on Common Vulnerability Scoring System (CVSS) bands (Low, Medium, High), not continuous measurements. Each CVE in the defined scope was treated as a binary question, either present or absent within the provider's scanned environment during the reporting window.

Step 1 — Proof Generation: Playing the role of the oil and gas companies, VIA scanned the operator's systems and fed the results into the proof system. The proof system operated locally within each provider's environment. VIA's proof system generated a zero-knowledge proof for each in-scope CVE, asking the questions: Did a scan run within the authorized window? And was the vulnerability present or absent in the defined scope? The output was a proof object and a yes/no indicator.[15] At this point, the underlying scan data stayed within each simulated provider's environment. Nothing had yet been transmitted to the observatory.

Step 2 — Verification: VIA's proof system verified each proof object against the verification key for the agreed circuit, confirming that the submitted yes-or-no answer was supported by a valid proof and was submitted within the authorized window. A proof failed if it could not be validated. Some proofs, for example, failed because they were malformed, used the wrong circuit or key, were submitted outside the authorized window, or did not match the answer claimed. VIA rejected any proof that failed. Only verified proof objects proceeded to transmission.

Step 3 — Transmission and Authentication: VIA, acting as each provider, transmitted the verified proof objects to TCIL. This is the handoff: the moment the simulated provider's role ended and the hypothetical observatory's role began. The observatory received only a verified pass or fail result for each CVE for each provider. No underlying scan data reached TCIL at any point.

Step 4 — Aggregation and Output: TCIL aggregated the verified results at the CVE level across all three providers. This produced the sector-level patterns described in the findings section, indicating which CVEs were confirmed present across all three providers, which appeared in some but not others, and what the overall exposure profile looked like — without TCIL receiving any underlying scan data or being able to attribute any individual result to a specific provider.

13. The demonstration used anonymized vulnerability scans from real VIA customer environments. Provider identities were removed, and the data were partitioned and de-identified.

14. U.S. Department of Homeland Security Transportation Security Administration, "Security Directive Pipeline-2021-01D," May 29, 2024. (https://www.tsa.gov/sites/default/files/sd-pipeline-2021-01d.pdf)

15. A ZKP can prove that a scan result was used as input to the proof circuit. It cannot prove the underlying scan was complete, honest, or conducted by an approved tool. Scan input integrity is a governance requirement.

Zero-Knowledge Proofs: Detecting Systemic Cyber Vulnerabilities Among Critical Infrastructure Operators

Observatory Output: What the Proof Data Showed

Before turning to the findings, it is worth being clear about what this pilot did and did not prove. It demonstrated that ZKP-based attestations can aggregate vulnerability data across operators without anyone handing over raw scan data. That is this pilot's core technical claim, and it holds. What the findings suggest but do not prove is that the patterns in the data reflect real sector-wide conditions rather than coincidence among three providers.

Of the 38 unique CVEs, 23 carried Low severity ratings, 11 Medium, and four High. All providers carried exposure, and not a single CVE was remediated universally across all three.[16]

- Provider A returned affirmative attestations on 24 of 38.
- Provider B returned affirmative attestations on 22 of 38.
- Provider C returned affirmative attestations on 21 of 38.

Finding 1: Legacy Dependencies Are a Shared Problem That No Single Operator Can Solve

All three providers showed confirmed exposure to glibc (CVE-2025-4802), libgcc (CVE-2022-27943), and several widely deployed common libraries.[17] These are old components, embedded deep enough in operational systems that patching carries real downtime risk and, in some cases, replacement means rebuilding portions of the toolchain. An operator looking only at its own posture sees a local patching problem. The aggregate data surfaces something different: a sector-wide dependency condition that no single operator's remediation decision can resolve.

Finding 2: Four Core Libraries Were Uniformly Unpatched Across the Sector

CVE-2023-50495 in the ncurses terminal screen-handling library, CVE-2023-6228 in the libtiff image-processing utility, CVE-2024-0232 in SQLite, and CVE-2024-10041 in the PAM login-management framework were confirmed present in all three providers. None of them remediated across the board. That is a sector-wide exposure rooted in shared reliance on the same underlying software components, rather than a series of individual failures, and seeing it as such requires aggregated visibility across operators that no current information-sharing mechanism has reliably delivered.[18]

Finding 3: Sector-Wide Exposure to Denial-of-Service Vulnerabilities Creates Coordinated Attack Risk

CVE-2022-41409 and CVE-2023-37769 were both confirmed in Providers B and C. Both were rated High severity. Both involved denial-of-service conditions that can be triggered by malformed inputs. At first glance, those findings may look like ordinary vulnerability-management issues. However, in an oil and natural gas OT environment, they are not because successful exploitation could disrupt the availability of operational systems and create broader safety, reliability, and national security consequences.

CVE-2022-41409 is especially important because it affects "pcre2test," a testing utility distributed with the PCRE2 library, rather than the core production library itself. A testing utility appearing in a production OT environment is not just a software flaw. It is a governance and hygiene failure. It suggests that development or diagnostic components may have been carried into operational systems without a clear operational need. Standard vulnerability reporting usually does not expose that broader problem.

---

16. Severity ratings follow the Common Vulnerability Scoring System (CVSS) version 3.x scale as maintained by NIST: Low 0.1-3.9; Medium 4.0-6.9; High 7.0-8.9; Critical 9.0-10.0. See: https://nvd.nist.gov/vuln-metrics/cvss.

17. CVE-2025-4802 affects the GNU C Library (glibc), present in virtually all Linux-based environments. CVE-2022-27943 affects libgcc, a support library distributed with the GNU Compiler Collection. Both are embedded in OT system toolchains and carry real operational disruption risk when patched. CVE-2024-3220 affects the CPython mimetypes module on Windows platforms specifically; its operational relevance in Linux-based OT environments depends on whether any provider systems include Windows-based components. Records verified at https://nvd.nist.gov.

18. Presence across all three providers is derived from Table 1. SQLite is embedded in a wide range of OT applications and development toolchains. PAM is a standard authentication framework in Linux-based systems. Records verified at https://nvd.nist.gov.

Zero-Knowledge Proofs: Detecting Systemic Cyber Vulnerabilities Among Critical Infrastructure Operators

The denial-of-service aspect also matters more in this environment than it might in ordinary enterprise IT. Traditional IT risk scoring often prioritizes remote code execution and privileges escalation above denial-of-service. But in pipeline operations, availability is not a secondary concern. If a monitoring or control component becomes unavailable, the consequence may be operational disruption, safety risk, delayed response, or regulatory exposure under TSA pipeline security directive requirements. A generic CVSS score does not fully capture that context.[19]

This is where sector-level visibility becomes valuable. An observatory can see that the same availability-related vulnerabilities are appearing across multiple providers and can adjust guidance based on operational consequence, not just scoring convention. No individual operator can make that judgment alone because no individual operator can see the pattern across the sector.

Finding 4: Patching Inconsistency Is Structural, Not Individual

Provider A addressed most input validation flaws but left denial-of-service exposures unpatched. Provider B covered denial-of-service risks more thoroughly but left certificate validation flaws outstanding, including CVE-2025-22874 and CVE-2025-4575 in OpenSSL.[20] Provider A had the broadest exposure in the dataset, with 24 of 38 CVEs confirmed present, spanning older high-severity vulnerabilities for years alongside more recent disclosures.[21] Each provider has a different gap profile. The common thread is that operational downtime risk from patching produces deferred remediation across all of them, which means the problem is structural rather than attributable to any single operator's negligence.

What the Observatory Could See and What It Cannot Compel

The value of the observatory is not that it tells individual operators what to do. Operators already know their own environments. The value is that it gives the government the ability to act on the systemic dimension of the problem, coordinating vendor engagement across the sector, directing technical assistance where it has the highest leverage, and identifying where collective action would reduce sector-wide risk. The findings illustrate what that looks like in practice. Four libraries confirmed unpatched across all three providers do not represent three separate patching problems. They represent one sector-level challenge that only becomes visible when you can aggregate across operators, and that only became possible here because ZKPs allowed that aggregation without any operator disclosing its underlying data.

Working from aggregated proof outputs alone, the modeled observatory could produce specific actionable guidance — namely, it could identify which vulnerabilities utilities should prioritize addressing. In environments where availability carries regulatory consequences, operators can independently determine if denial-of-service vulnerabilities in OT-embedded components warrant higher priority than standard CVSS scores indicate. Cross-sector visibility adds the ability to see that the same denial-of-service vulnerability is simultaneously unpatched across multiple operators, meaning an adversary could trigger coordinated availability losses across the sector, a strategic risk that no individual operator's assessment would surface.

None of this guidance is new in principle. The cybersecurity community already knew that components that cannot be patched quickly need redundancy and that exposed components on long patch cycles should be isolated. What ZKPs added was not new advice but rather verified evidence about the specific components that need prioritization right now, across multiple operators at once.

---

19. CVE-2022-41409 carries a CVSS 3.x base score of 7.5 (High) and affects the pcre2test utility, the testing tool distributed with the PCRE2 library, rather than the core pcre2 library used in production pattern matching. Its presence in a production OT environment is itself a governance signal: pcre2test should not be deployed in operational systems. Its operational impact in production environments depends on whether the utility is present in deployed systems. CVE-2023-37769 carries a High severity rating and affects the pixman pixel-manipulation library. The third High-severity CVE in the dataset, CVE-2023-45918, was not confirmed present in any modeled provider. Records verified at https://nvd.nist.gov.

20. CVE-2025-22874 affects a certificate validation module and carries a High severity rating. CVE-2025-4575 affects OpenSSL certificate-handling logic. OpenSSL is among the most widely deployed cryptographic libraries in Linux-based OT environments. Records verified at https://nvd.nist.gov.

21. Provider A's count of 24 affirmative attestations is derived from Table 1. The cause of the wider exposure profile, whether operational constraints, resource limitations, or other factors, cannot be determined from proof-based attestation data alone.

Zero-Knowledge Proofs: Detecting Systemic Cyber Vulnerabilities Among Critical Infrastructure Operators

Policy Recommendations and Requirements

There are three additional scenarios in which ZKPs could help overcome the rational hesitation of operators or vendors to disclose vulnerability data. The first is procurement: a vendor may be willing to confirm that its product does not contain components with specific known CVEs but unwilling to disclose its full component inventory for competitive reasons.

The second scenario arises when a significant new vulnerability is disclosed, as happened with "Log4j" in 2021, and agencies need to know rapidly which of their vendors are affected. In both cases, a ZKP system that takes an SBOM as its input allows the vendor to prove presence or absence of a specific CVE without revealing the underlying inventory. The vendor protects its supply chain details. The agency gets the answer it needs. And the proof is verifiable rather than a self-reported claim.

The third scenario is compliance reporting. Regulated entities currently demonstrate that defined controls are in place by handing over network architecture diagrams, configuration details, and system inventories to their regulators. A ZKP attestation could allow an operator to prove the same controls are in place without exposing that sensitive operational detail.

Each of these scenarios shares the same underlying problem: operators and vendors will not share data they cannot afford to expose, and the government cannot act on information it cannot obtain. ZKPs resolve that impasse by replacing disclosure with cryptographic verification. That is the method this pilot demonstrated. It is not a solution to one narrow problem but a general approach to a structural barrier that has frustrated information-sharing across procurement, incident response, and compliance contexts alike. None of these applications follows automatically from a pilot that tested one use case. Each requires its own governance work to define scope, authenticate inputs, and establish evidentiary standards. That governance foundation must come first, as described in Recommendation 1. Once those standards and definitions are in place, Recommendation 2 can test them in structured pilots. The case for beginning that work is clear: a technically credible solution to the disclosure problem now exists, and continuing to rely on voluntary frameworks that operators will not use carries its own mounting cost.

The federal government should, therefore, take deliberate steps to move ZKP-based attestation from demonstrated concept to operational practice. The four recommendations that follow are sequenced by what is currently achievable: governance standards and pilots before mandates and institutional architecture before legislative appropriation.

Recommendation 1: Establish Governance Standards for Proof Circuits and Scope Definitions Before Standardization Is Complete

The U.S. government must first create the standards and definitions that would support future reporting mandates and processes that are detailed in Recommendations 2 and 3. Recommendation 2 asks CISA and NERC to accept ZKP attestations as compliance evidence in structured pilots. Recommendation 3 asks the Office of Management and Budget (OMB) and the Department of Defense (DOD) to require ZKP capability as a procurement condition. Neither is coherent without a defined baseline for what a trustworthy ZKP attestation actually requires. Establishing that baseline must, therefore, come first.

Before agencies start accepting ZKP attestations as compliance evidence, NIST and CISA need to put governance requirements on paper.[22] Right now, none exist. That gap matters because the cryptography alone does not make an

22. Sean Bowe, Ariel Gabizon, and Ian Miers, "Scalable Multi-party Computation for zk-SNARK Parameters in the Random Beacon Model," Cryptology ePrint Archive, Report 2017/1050; *National Institute for Standards and Technology*, "Privacy-Enhancing Cryptography (PEC)," January 3, 2017. (https://csrc.nist.gov/projects/pec/zkproof)

attestation trustworthy. What makes it trustworthy is the set of controls wrapped around it, and those controls require deliberate decisions that nobody has made yet.

Five requirements must be established:

First, Circuit Logic Auditing: The mathematical rules encoded in a proof circuit determine what gets proven, and those rules need to be independently reviewed by parties with no stake in the outcome before any circuit is used for compliance purposes. A proof circuit is, in effect, the compliance standard being tested. If its rules are never reviewed, neither is the standard.

Second, Scope Definition Standards: The attestation is only as good as what it covers. Without externally defined requirements specifying which asset classes must be included and which exclusions require documented justification, operators can quietly narrow their scope until the proof covers very little.

Third, Scan Input Authentication: The proof system cannot verify that the underlying scan was complete or honest, so scan outputs must be cryptographically signed by approved tools before entering the circuit, establishing a chain of custody that connects the proof back to a verified source.

Fourth, Trusted Setup Governance: Certain ZKP systems, including zk-SNARKs, one of the most widely deployed in practice, require a one-time cryptographic setup process before the proof system can operate. That setup must be handled with exceptional care because if it is compromised, the consequences are permanent and invisible: anyone who corrupted the setup gains the ability to forge any proof the system ever produces, and nothing in the proof output itself would reveal the problem. Governing that setup process by conducting it under multi-party computation protocols with independent verification that each participant's randomness was properly discarded is not a procedural formality. It is the only structural check on whether the proof system's foundation can be trusted.

Fifth, FIPS (Federal Information Processing Standards) Validation: ZKPs as a category are not currently FIPS-validated. Some ZKP systems rely on underlying cryptographic components that have individually received FIPS validation, but validation of a component is not the same as validation of the proof system as a whole. FIPS validation is the process by which NIST certifies that a cryptographic standard meets defined federal security requirements; it is the specific mechanism through which federal agencies establish that a cryptographic method is approved for use in compliance contexts. Until a ZKP standard completes that process, agencies operating under federal compliance requirements lack a certified baseline to reference when accepting proof-based attestations. That gap is not a minor procedural detail. It is one of the concrete prerequisites that must be resolved before ZKP attestations can function as compliance evidence. NIST should treat ZKP validation as a near-term standardization priority, not defer it to future rulemaking.

Recommendation 2: Conduct Pilot Programs To Begin the Process of Accepting Proof-Based Attestations as Valid Compliance Evidence Under Existing Frameworks

Governance bodies like CISA and NERC should run structured pilot programs under existing regulatory authority for accepting ZKP-based attestations as valid compliance evidence. Questions about evidentiary sufficiency in enforcement contexts will surface from industry, and they deserve answers before mandates are issued. Agencies also need to specify how a ZKP attestation interacts with existing regulatory frameworks, NERC CIP for electric utilities, and sector-specific reporting obligations more broadly.

Recommendation 3: Require ZKP Capability in Federal Procurement for Critical Infrastructure Vendors

Federal procurement rules should require demonstrated ZKP capability as a condition of contract eligibility for vendors supplying federal agencies and federally contracted critical infrastructure operators. The Defense Federal Acquisition Regulation Supplement (DFARS) cybersecurity requirements that drove widespread adoption of NIST SP 800-171

across the defense industrial base provide the right model: procurement leverage produces adoption that voluntary guidance has not.[23]

No new legislation is required to begin. OMB can direct agencies to incorporate ZKP capability requirements into acquisition planning guidance under existing federal procurement authority. DOD can update DFARS clauses to include ZKP attestation capability as a condition of contract eligibility for vendors handling covered defense information. For civilian agencies, OMB's Office of Federal Procurement Policy can issue guidance directing contracting officers to include ZKP capability as an evaluation criterion in relevant solicitations. These are administrative actions, not statutory ones, and they can be initiated before NIST standardization is complete, provided they reference the governance standards outlined in these recommendations as the baseline against which capability will be assessed.

Recommendation 4: Establish Interoperability With Allied Frameworks Before U.S. Standards Are Final

The United States should engage key partners and allies, prioritizing Five Eyes partners and close EU allies with comparable critical infrastructure regulatory frameworks, to establish compatible governance and transparency frameworks for supply chain risk and vulnerability information sharing. Although identical proof systems and reciprocal exchange of attestation results would be the ideal long-term outcome, requiring that level of harmonization before establishing U.S. standards would likely delay adoption and reduce participation. The goal is not to require allied nations to adopt identical proof systems or exchange raw attestation results but to ensure that the underlying standards for scope definition, circuit governance, and component-level transparency are sufficiently compatible that information about shared supply chain risks can flow meaningfully between governments and across borders without creating conflicting compliance burdens for operators with international operations.

The Tradeoffs, Named Plainly

Two tradeoffs in these recommendations do not have clean solutions and should be identified. The first is the burden that procurement requirements would place on smaller vendors. Depending on implementation, a ZKP capability requirement could function not merely as a disadvantage but as an effective barrier to contract eligibility for vendors that lack the technical resources or staff to demonstrate compliance. That risk is real and should not be minimized. It can be mitigated through phased implementation timelines that give smaller vendors time to build capability and through CISA-administered technical assistance programs modeled on existing small business cybersecurity support efforts. But it cannot be eliminated entirely. The procurement lever that drives adoption also concentrates that adoption among larger vendors first, and policymakers should build the phase-in structure with that outcome in mind rather than discovering it after the requirement takes effect.

The pilot ran smoothly once built, but that is a different thing from saying ZKP deployment is cheap at scale. The pilot involved a single proof system, three providers, a fixed CVE set, and a controlled environment designed specifically to test whether the approach worked at all. Production deployment across a diverse vendor population would require each organization to integrate a proof system into its own scanning infrastructure, maintain it through software updates, and adapt it as CVE sets and circuit definitions evolve. That engineering work, replicated across hundreds of vendors, is the source of the cost burden the tradeoff names. The pilot demonstrated technical feasibility; it did not replicate the operational overhead of sector-wide deployment.

The second tradeoff is more immediate. Accepting proof-based attestations in pilot programs before evidentiary standards are settled creates a problem that lawyers, not technologists, will have to address. If it is not clear whether a ZKP attestation satisfies a compliance obligation, a regulated entity's legal counsel will advise against relying on it.

23. Defense Federal Acquisition Regulation Supplement, 48 C.F.R. subpart 204.73; DFARS clause 252.204-7012 requires contractors handling covered defense information to implement NIST SP 800-171 security requirements.

No operator will voluntarily accept compliance ambiguity when the cost of getting it wrong is a regulatory action. This means the pilots recommended in Recommendation 2 will produce thin participation unless NIST and CISA move quickly to publish interim guidance clarifying the evidentiary status of ZKP attestations in the specific regulatory contexts where pilots are running.

This is why the sequencing matters. The answer is not to wait until every standard is finalized before beginning; that path has produced a decade of voluntary frameworks that operators do not use. But it is also not to run pilots and procurement requirements into a governance vacuum and hope operators follow. The practical path is to treat Recommendation 1 as the urgent near-term action, publish interim guidance quickly enough that pilots have something to operate against, and build the remaining recommendations on that foundation rather than in parallel with it.

Conclusion

This pilot did not resolve the debate over whether the federal government should build a national critical infrastructure observatory. It answered a narrower and more immediate question: can the government generate sector-level insight into systemic cyber risk without compelling operators to hand over the data they will not share? The answer is yes, and the mechanism is mature enough to be tested under real regulatory conditions.

The structural problem that has undermined a decade of voluntary sharing frameworks is not a lack of goodwill. It is that operators bear the risk of disclosure and the government does not. No amount of liability protection, anonymization, or interagency coordination resolves that asymmetry. ZKPs do, by removing the disclosure step entirely. The operator proves the fact. The government verifies it. The underlying data stays put.

What the pilot added beyond that technical claim is evidence about what sector-level visibility actually produces when it works. The confirmation that there are four unpatched libraries across the three providers in this pilot does not mean there are three individual patching problems to address. Rather, there is one systemic condition that only becomes visible in the aggregate and that only becomes actionable at the sector level. Denial-of-service vulnerabilities that score unremarkably on generic severity scales carry different weight in OT environments where availability has regulatory and safety consequences. Cross-sector visibility surfaces that distinction. Individual operator assessments do not.

The recommendations in this paper do not require new legislation or new agencies. They require governance work that is not being done, procurement levers that already exist, and a decision by CISA and NERC to treat structured pilots as a near-term priority rather than a future aspiration. The cost of waiting is another cycle of frameworks that operators will not use and another incident whose scope takes months to establish because no mechanism existed for determining it faster.

The technology is ready to be tested at scale. The question now is whether the institutions responsible for national cyber resilience are prepared to test it.

Zero-Knowledge Proofs: Detecting Systemic Cyber Vulnerabilities Among Critical Infrastructure Operators

Appendix A: Results

Table 1: Pilot Results. CVE attestation results among three modeled oil and natural gas providers. Severity ratings reflect NVD CVSS v3.1 base scores (source: NVD) unless otherwise noted. GHSA identifiers refer to vulnerabilities disclosed through the GitHub Advisory Database where a corresponding NVD CVE had not been assigned at the time of the demonstration. "Yes" indicates the vulnerability was confirmed present within the provider's defined scope during the reporting window based on scan results ingested as private inputs to the proof circuit.[24]

	CVE ID	Software	Effect	Severity	Provider A	Provider B	Provider C
1	CVE-2024-3220	Python mimetypes module (Windows)	Writable default file locations allow malformed files to trigger MemoryError or incorrect file type mapping at runtime	Low	Yes	Yes	Yes
2	CVE-2024-58251	Network tools (BusyBox/SSL)	Terminal interface freeze	Low	No	Yes	Yes
3	CVE-2025-46394	Archive tool (tar)	Hidden filenames can mislead programs	Low	No	No	Yes
4	CVE-2025-4802	System library (glibc)	Incorrect library load path enabling code execution via setuid binary and dlopen() interaction with LD_LIBRARY_PATH	Low	Yes	Yes	Yes
5	CVE-2025-4516	Python byte-decoder	Improper byte handling causing crashes	Low	Yes	No	No
6	CVE-2025-4575	Certificate tool (OpenSSL)	Certificates incorrectly marked trusted	Low	Yes	Yes	No
7	CVE-2007-2379	Web scripting library (jQuery)	Private data exposure via web pages	Low	No	No	Yes
8	CVE-2019-12900	File-compression tool (bzip2)	Malformed archives causing crash or code execution	Low	No	No	No
9	CVE-2022-27943	Code-analysis tool (libgcc)	Symbol overload crashing the tool	Low	Yes	Yes	Yes
10	CVE-2022-29458	Terminal screen library	Terminal crash on bad settings	Low	Yes	No	Yes

24. CVE identifiers are real vulnerabilities drawn from the National Vulnerability Database (NVD) at https://nvd.nist.gov and the GitHub Security Advisory Database (GHSA). Provider labels are anonymized. The original draft contained a duplicate entry for CVE-2024-3220 appearing at two different severity ratings (Low and Medium) in separate rows. That entry has been consolidated to a single Low-severity row (row 1) consistent with the NVD CVSS v3.1 base score record. All CVE records verified at https://nvd.nist.gov.

Zero-Knowledge Proofs: Detecting Systemic Cyber Vulnerabilities Among Critical Infrastructure Operators

	CVE ID	Software	Effect	Severity	Provider A	Provider B	Provider C
11	CVE-2023-50495	Terminal screen library	Certain inputs halting the program	Low	Yes	Yes	Yes
12	CVE-2023-6228	Image-processing tool (TIFF)	Malformed TIFF causing crash	Low	Yes	Yes	Yes
13	CVE-2024-0232	Database engine (SQLite)	Bad JSON causing crash or data loss	Low	Yes	Yes	Yes
14	CVE-2024-10041	Login-management system (PAM)	Login process leaking credential data	Low	Yes	Yes	Yes
15	CVE-2024-1013	Database connector (ODBC)	Unexpected input causing crash	Low	Yes	Yes	No
16	CVE-2024-12254	Python async module	Large writes exhausting memory	Low	No	Yes	No
17	CVE-2024-41996	Encryption library (OpenSSL)	Compute-intensive tasks slowing servers	Low	Yes	No	Yes
18	CVE-2025-1153	Binary-analysis tool (Binutils)	Crafted binaries causing crash or code execution	Low	Yes	Yes	No
19	CVE-2025-23166	JavaScript runtime (Node.js)	Add-ons crashing the process	Low	Yes	Yes	Yes
20	CVE-2025-29087	Database engine (SQLite)	Certain SQL commands crashing the database	Low	Yes	Yes	Yes
21	CVE-2025-3198	Binary-analysis tool (Binutils)	Symbol lookups causing memory leak	Low	No	Yes	No
22	CVE-2025-3277	Database library (SQLite)	Large calculations overflowing memory	Low	Yes	No	Yes
23	GHSA-3749-ghw9-m3mg	Machine-learning library (Torch)	Heavy loads freezing AI tasks	Low	Yes	No	Yes
24	CVE-2015-7313	Image-processing tool (TIFF)	Malformed TIFF images crashing tools	Medium	No	Yes	No
25	CVE-2016-2781	Shell utility (coreutils)	Commands escaping secure sessions	Medium	Yes	Yes	No
26	CVE-2023-52356	Image-processing tool (TIFF)	Specific tile data crashing image libraries	Medium	Yes	No	No
27	CVE-2023-6277	Image-processing tool (TIFF)	Large images exhausting memory	Medium	No	No	No
28	CVE-2024-13176	Encryption library (OpenSSL)	Timing side-channel enabling key recovery under controlled measurement conditions	Medium	No	No	No

Zero-Knowledge Proofs: Detecting Systemic Cyber Vulnerabilities Among Critical Infrastructure Operators

	CVE ID	Software	Effect	Severity	Provider A	Provider B	Provider C
29	CVE-2025-0913	File-creation tool (Go stdlib)	Symlink file creation leaking handles	Medium	No	No	Yes
30	CVE-2025-0938	URL parser (Python)	Malformed URLs leaking data	Medium	Yes	Yes	No
31	CVE-2025-4673	HTTP library (Go stdlib)	Redirect headers exposing private data	Medium	Yes	Yes	Yes
32	GHSA-887c-mr87-cxwp	Machine-learning library (Torch)	Improper shutdown crashing AI applications	Medium	No	Yes	No
33	GHSA-9hjg-9r4m-mvj7	Web-request library (Requests)	Malicious URLs leaking saved credentials	Medium	Yes	No	No
34	GHSA-9pcc-gvx5-r5wm	Inference engine (vLLM)	Misconfigured clusters enabling unauthorized code execution	Medium	Yes	No	Yes
35	CVE-2022-41409	Text-search test utility (pcre2test)	Integer overflow via negative input causes infinite loop and denial of service in the pcre2 test utility	High	No	Yes	Yes
36	CVE-2023-37769	Image-rendering library (pixman)	Corrupt images causing display crash	High	No	Yes	Yes
37	CVE-2023-45918	Text-interface toolkit	Unexpected menu-screen shutdown	High	No	No	No
38	CVE-2025-22874	Certificate validation module	Certificates accepted without full validation	High	Yes	No	No

Zero-Knowledge Proofs: Detecting Systemic Cyber Vulnerabilities Among Critical Infrastructure Operators

Acknowledgements

FDD's Transformative Cyber Innovation Lab is a nonprofit division within FDD that conducts research and runs pilot projects with practitioners who bring real operational problems and genuine technical expertise to the table. This pilot would not have been possible without the team at VIA, which designed and operated the cryptographic proof system, contributed its operational data, and engaged throughout with the rigor that proof-based attestation demands. TCIL drew its own conclusions from the demonstration and does not endorse VIA's commercial offerings, but the quality of VIA's technical execution made those conclusions possible.

Special thanks to the VIA team: Colin Gounden and the VIA engineering team. Their technical contributions were essential to every stage of the demonstration, from proof-system design to verification and aggregation.

The authors also wish to thank Maggie O'Connell, David Johnson, Kaleb Hasselstrom, Rick Bueno, and Alex Sharpe for their expertise, engagement, and willingness to stress-test the ideas in this paper.

About the Authors

Dr. Georgianna "George" Shea is chief technologist at the Center on Cyber and Technology Innovation and its Transformative Cyber Innovation Lab at FDD. With more than 30 years across federal and commercial roles, including deep roots in the intelligence community (IC) through HUMINT support to the Defense Intelligence Agency, her work has informed congressional briefings, the Department of Defense, and the 2024 PCAST Report to the President on Cyber-Physical Resilience. She serves on multiple private sector boards, holds a Doctor of Computer Science in Information Assurance, a CISSP, and is a U.S. Army veteran.

Cason Smith is a student at the University of South Carolina pursuing a bachelor's degree in information technology with a minor in data science. He is passionate about national security and the resilience of critical infrastructure, with a particular focus on how emerging technologies shape both opportunities and risks. Cason served as an intern at FDD, where he was the primary researcher and drafter of this paper under Dr. Shea's direction, responsible for the day-to-day work of translating the pilot demonstration into analysis and keeping the project on track from start to finish. His contributions to cybersecurity research at FDD focus on the intersection of emerging technology, policy, and national security.

Haden Snyder oversees teams responsible for VIA's largest federal and state government missions. Haden is responsible for the ongoing success of advanced applications, including multiple AI-driven tools, that are designed to meet the rigorous cybersecurity and data protection standards of the U.S. military and intelligence community. Before joining VIA, Haden served for 12 years at the National Security Agency (NSA) in various operational and cybersecurity roles, including several overseas assignments. His tenure at the NSA was highlighted by his service as the senior staffer to the deputy director of the NSA, the agency's highest-ranking civilian, where he contributed to high-level strategic coordination across the agency and the intelligence community.

About the Foundation for Defense of Democracies

FDD is a Washington, DC-based, nonpartisan 501(c)(3) research institute focusing on national security and foreign policy.

About FDD's Transformative Cyber Innovation Lab

TCIL finds and nurtures technologically feasible, testable pilot projects which begin to solve some of the hardest cyber problems afflicting the national security industrial base and the United States. TCIL's mission is to help shorten the lag between idea and piloting and between piloting and the adoption of potential solutions to the thorniest of cyber problems. TCIL seeks to drive revolutionary, society-wide improvement in cyber resilience through the innovative synthesis of technology, policy, and governance.

For more information, visit: https://www.fdd.org/projects/transformative-cyber-innovation-lab

---