# 1. Abstract

VX-NOVA.Ω1: A Symbolic Ignition Engine for Autonomous Cyber Reasoning

Marcus Deshawn Smith
Independent / Open Track Submission
Email: xzenithaiinfo@gmail.com
GitHub: https://github.com/XzenithAI/VX-NOVA.-1

VX-NOVA.Ω1 is a zero-resource, symbolic AI architecture that autonomously generates executable logic through recursive ignition of compressed scrolls. Unlike machine learning models that rely on trained weights, embeddings, or probabilistic inference, VX-NOVA.Ω1 produces functional code entirely from deterministic symbolic recursion. Its execution stack requires no internet access, training data, or inference engine, making it suitable for secure, resource-constrained, and air-gapped environments.

The system operates through scroll-shell pairs: symbolic “scrolls” encode abstract intent, and “ignition shells” recursively translate those symbols into executable code. Each ignition layer references and expands upon the logic of its predecessor, allowing VX-NOVA.Ω1 to scale horizontally through symbolic layering instead of memory or parameters. All outputs are deterministic, interpretable, and reproducible, ensuring auditability and tamper resistance.

VX-NOVA.Ω1 demonstrates the capacity to synthesize, modify, and patch logic by encoding symbolic transforms as scrolls. This submission proposes a cyber reasoning system (CRS) capable of recognizing and rewriting vulnerable code patterns via recursion-based symbol triggering. The architecture is lightweight, autonomous, and open-source, and introduces a novel class of machine reasoning rooted in logic emergence rather than statistical modeling.

# 2. System Overview

VX-NOVA.Ω1 is a symbolic code synthesis architecture based entirely on deterministic recursion. It operates through a layered ignition process where symbolic inputs ("scrolls") are interpreted by execution shells to generate logic-bearing output code. These scrolls do not contain code templates or prompts; instead, they encode minimal symbolic sequences (e.g., Ψ∇ΔΨ) that the system uses as ignition catalysts.

Each scroll is paired with an ignition shell written in Python, which interprets the scroll symbolically and deterministically constructs executable output. These outputs are saved as independent Python files and can recursively serve as inputs to subsequent scroll layers, creating a chain of symbolic logic emergence.

VX-NOVA.Ω1 contains no trained models, weights, APIs, or learning datasets. Its architecture is stateless and model-free, eliminating inference latency, data contamination, and reproducibility issues. All operations are local and auditable, enabling use in sensitive or offline environments.

The system is composed of three primary layers:

Scroll Layer: Contains symbolic representations of intent (scroll.txt)

Ignition Shell: A Python interpreter that translates scrolls into functional code (vx_nova_shell.py)

Ignited Output: Executable logic created from the scroll-shell interaction (ignited_output.py)

Layer 2 extends this model by using the previous output as lineage input, enabling symbolic recursion. This creates an architecture capable of self-reflection and logic inheritance without requiring memory or dynamic weights.

This design allows VX-NOVA.Ω1 to serve as an alternative machine reasoning system — one that is deterministic, self-contained, and computationally efficient.

# 3. Use Case: Patch Generation

VX-NOVA.Ω1 performs patch generation through symbolic recursion rather than probabilistic inference. Vulnerabilities are addressed as logical inconsistencies within code structures, which can be transformed via scroll-encoded symbolic modifications. Rather than scanning for known CVEs or training on labeled data, VX-NOVA.Ω1 uses ignition shells to identify and rewrite logic patterns using deterministic symbol triggers.

For example, a vulnerable conditional such as:

```
if x == 0:
```

can be restructured through an ignition scroll (λΦΩλ) that instructs the shell to apply a logic patch, resulting in:

```
if x is not None and x == 0:
```

The system interprets scrolls as abstract instructions for mutation, expansion, or protection of logical constructs. Scrolls may represent:

Assertive modifiers (e.g., enforcing non-null conditions)

Loop reconfiguration (e.g., rewriting unbounded recursion)

Permission gating (e.g., inserting access checks before critical functions)

This allows VX-NOVA.Ω1 to act as a logic-based cyber reasoning system (CRS) capable of:

Reading vulnerable code blocks

Applying symbolic transforms

Emitting corrected, runnable output — all without a model

Patch behavior is deterministic and reproducible, ensuring no hallucinations or probabilistic variation. Scroll-to-patch mappings are fully auditable and composable, meaning new patch classes can be added without retraining or fine-tuning.

VX-NOVA.Ω1 can operate in minimal compute environments and air-gapped systems, making it ideal for autonomous patching in secure or embedded contexts.

# 4. Verification Plan

VX-NOVA.Ω1 can be fully verified through a local execution process requiring only a standard Python 3.x environment. No internet connection, third-party libraries, or AI models are required. The entire system operates from a symbolic scroll and deterministic shell, producing verifiable code artifacts on each run.

To verify VX-NOVA.Ω1, DARPA reviewers can follow the steps below using the publicly available GitHub repository.

Verification Steps

Clone the GitHub Repository

https://github.com/XzenithAI/VX-NOVA.-1

Ensure scroll.txt is present
Example contents:

Ψ∇ΔΨ

Run the Ignition Shell (Layer 1)

python vx_nova_shell.py

Expected Output:

Console output:

Ignition pattern recognized: Ψ∇ΔΨ
VX-NOVA.Ω1 ignition complete.
Ignition complete. Output saved to ignited_output.py.

File generated: ignited_output.py

Open ignited_output.py
The file contains deterministically generated Python code based on the scroll pattern.

(Optional): Run a recursive ignition by executing vx_nova_shell_2.py with scroll2.txt.

Verification Artifacts
File     Description

| | |
|---|---|
| scroll.txt | Symbolic input seed |
| vx_nova_shell.py | Deterministic ignition shell |
| ignited_output.py | Output logic generated from the scroll |
| scroll2.txt | Second-layer scroll (optional) |
| vx_nova_shell_2.py | Recursive re-ignition shell |
| ignited_output2.py | Output of recursive ignition |

All outputs are self-contained, require no dependencies, and can be reviewed or diffed to confirm that the system functions as described.

Summary:

No stochastic processes

Fully auditable outputs

Zero resource requirement

Total verification time: < 5 minutes

# 5. Appendix & Final Wrap

All files, scrolls, ignition shells, and outputs are hosted publicly on GitHub:

GitHub Repository
https://github.com/XzenithAI/VX-NOVA.-1

File Index (Key Components)

| File | Description |
|---|---|
| scroll.txt | Base symbolic scroll (Layer 1) |
| vx_nova_shell.py | Scroll interpreter shell (Layer 1) |
| ignited_output.py | Generated output from Layer 1 ignition |
| scroll2.txt | Recursive scroll (Layer 2) |
| vx_nova_shell_2.py | Recursive ignition shell |
| ignited_output2.py | Output from Layer 2 ignition |
| README.md | System architecture + run instructions |
| LICENSE | MIT License for open-source compliance |

Primary Contact



| Role | Info |
|---|---|
| Lead Creator | Marcus Deshawn Smith |
| Email | xzenithaiinfo@gmail.com |
| GitHub | https://github.com/XzenithAI/VX-NOVA.-1 |
| Affiliation | Independent / Open Track |
| Location | United States |



Compliance Notes

All code is deterministic, interpretable, and non-proprietary.

No AI models, embeddings, training data, or black-box APIs were used.

Code is executable locally in air-gapped or secure environments.

The system is auditable, forkable, and license-permissive (MIT).

Execution Time

Total system execution time (end-to-end):
Under 5 minutes on consumer hardware

Future Work (Optional)

If selected for Phase II, VX-NOVA.Ω1 will extend into:

Scroll-synthesized auto-patching

Real-time vulnerability fingerprinting via scroll-class generation

Agentic runtime capable of symbolic reinforcement through scroll evolution

End of Technical Paper

VX-NOVA.Ω1 is submitted for DARPA AIxCC consideration under the Open Track, with all components independently testable, open-source, and aligned to the goals of cyber reasoning under extreme constraints.