Incident Reference: NCGC-2025-1223
Classification: TOP SECRET // SIGINT // NOFORN
Threat Actor: APT-N² (The Quadratic Collective)
Status: REMEDIATED

“OPERATION CONSTANT TIME”

Post-Incident Debrief: APT-N² & The Algorithmic Kill Chain

“The most elegant attacks don't breach your defences. They make you destroy yourself.”

TOP SECRET // NOFORN

First Blood: A Coordinated, Silent Attack

DOCUMENT ID: UK-NCGC-DOSS-2025-0047 // DATE OF ISSUE: 19 DEC 2025 // CLASSIFICATION: RESTRICTED

THE INCIDENT

DATE: 19 December 2025, 03:47 GMT

TARGETS: Renderby Financial, Endpoint Industries, Response Media Group, The Documentsworth Archive. All designated Critical National Infrastructure.

SYMPTOMS: Complete Denial of Service, systems described as 'catatonic.'

INITIAL SOC FINDINGS

- No volumetric attack. No SYN flood. No amplification.
- WAF shows nothing. No SQLi attempts, no malformed requests, no payload signatures matching any known CVE.

THE CRITICAL CLUE

> *"Whatever's killing them, it's not coming from outside. It's coming from inside their own processing pipeline."*

— Senior Analyst Maya Collector, NCGC

TOP SECRET // NOFORN

APT-N²: A New Threat Actor Emerges

DOCUMENT ID: UK-NCGC-DOSS-2025-0048 // DATE OF ISSUE: 19 DEC 2025 // CLASSIFICATION: RESTRICTED

Communication Method: Tor-routed dead drop, indicative of nation-state tradecraft.

>"Your systems are not breached. They are busy."
>"We have not deployed malware. We have sent invoices."
>"Complexity is the ultimate zero-day."

TOP SECRET // NOFORN

DECLASSIFIED INTELLIGENCE DOSSIER

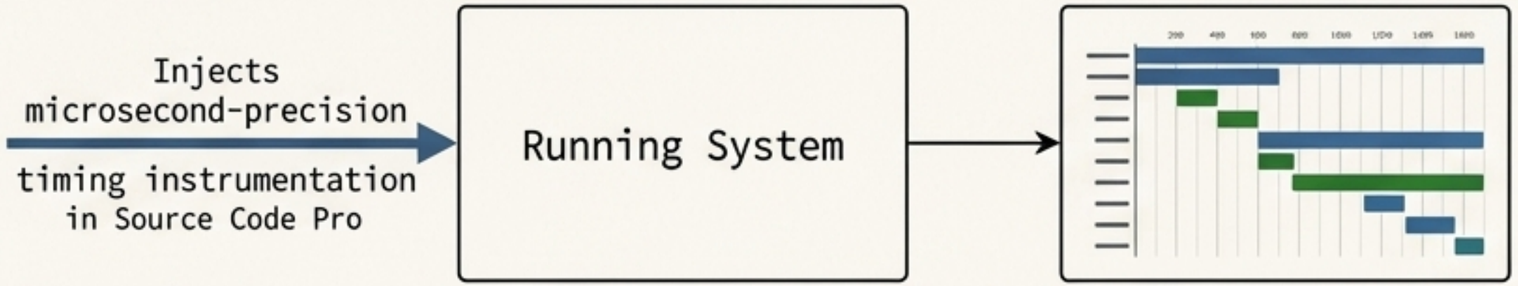
A New Weapon Requires a New Kind of Forensic Tool

DOCUMENT ID: UK-NCGC-DOSS-2025-0049 // DATE OF ISSUE: 20 DEC 2025 // CLASSIFICATION: RESTRICTED

The Hypothesis

>"Algorithmic complexity attack. Someone found a way to make their systems eat themselves."
— *Tempus "Stamp" Reyes, NCGC Responder*

The Tool



The Tool: The TIMESTAMP Protocol
Description: An experimental forensic telemetry framework.
Capabilities:
Performs stack-walking collection and execution reconstruction.
Key Metric: Measures ***self-time***—the actual CPU cycles burned by a specific method, excluding time spent in child calls.

The Justification

>"Traditional profilers will show us *that* something is slow. TIMESTAMP will show us ***why***."

The Ghost in the Machine: Weaponising Computational Complexity

TOP SECRET // NOFORN

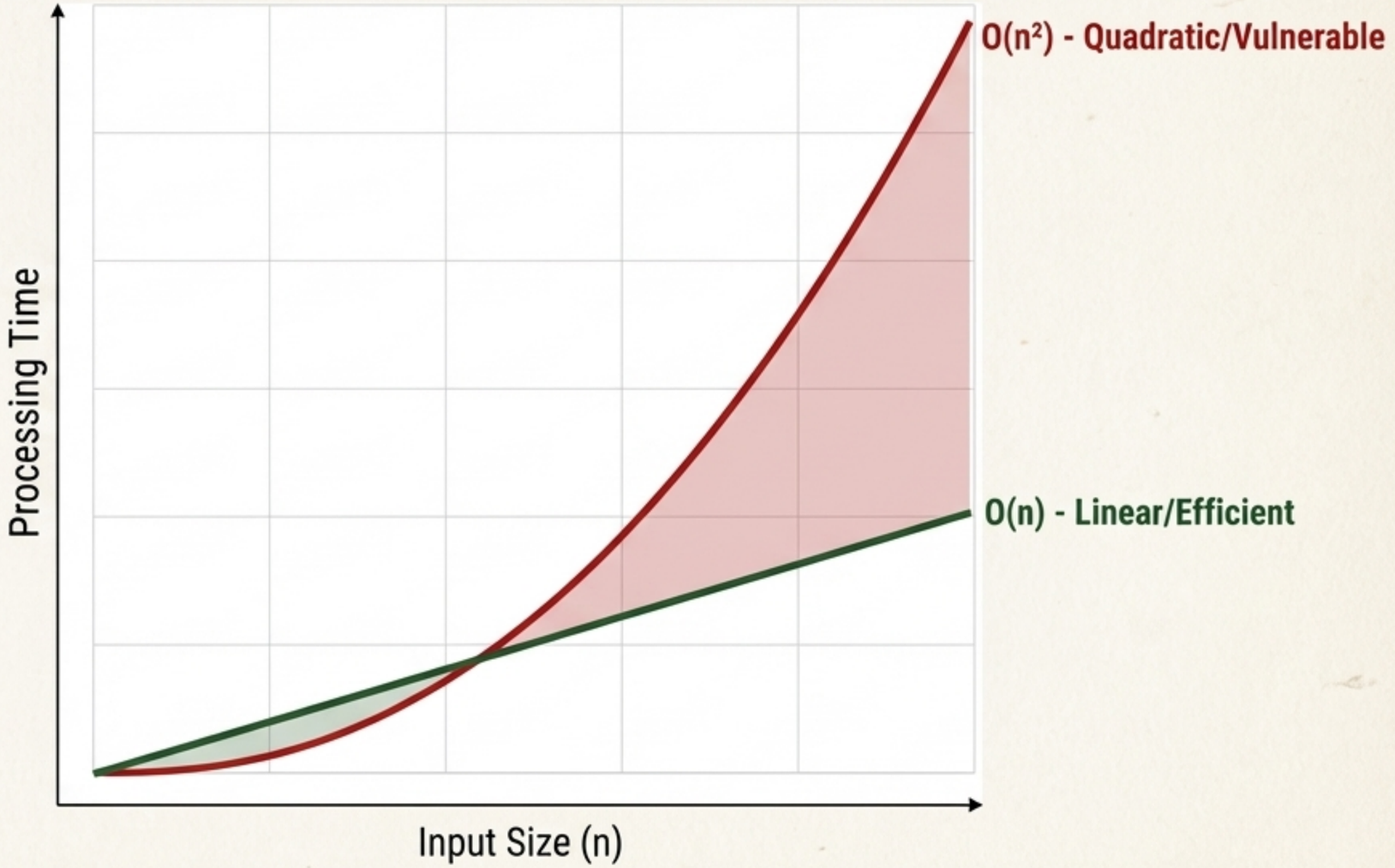
DOCUMENT ID: UK-NCGC-DOSS-2025-0050
// DATE OF ISSUE: 21 DEC 2025 //
CLASSIFICATION: RESTRICTED

Core Concept:

These attacks do not rely on volume, but on *leverage*. They find an operation in your code that scales poorly and craft specific inputs to trigger its worst-case behaviour.

Analogy:

Think of ReDoS (Regular Expression Denial of Service), where a carefully crafted string forces a regex engine into an exponential backtrack, hanging the system with a single request.



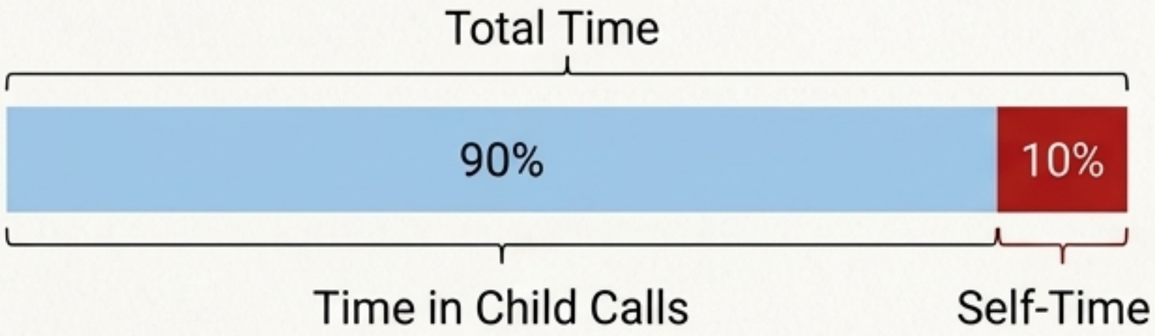
TOP SECRET // NOFORN

DOCUMENT ID: UK-NCGC-DOSS-2025-0051 // DATE OF ISSUE: 22 DEC 2025 // CLASSIFICATION: RESTRICTED

Following the Time: The First Clues Emerge

Key Metric Explained

Self-Time: Strips away the children. It shows you only the time that specific method actually consumed. If something has high self-time... that's where the actual CPU cycles are burning.



Initial Forensic Data

Target:	Sandboxed replica of Renderby infrastructure.
Input:	Medium-sized document.
Finding:	The `process_body` method consumes **60.1%** of the total execution time, with a self-time of **99.8%**.

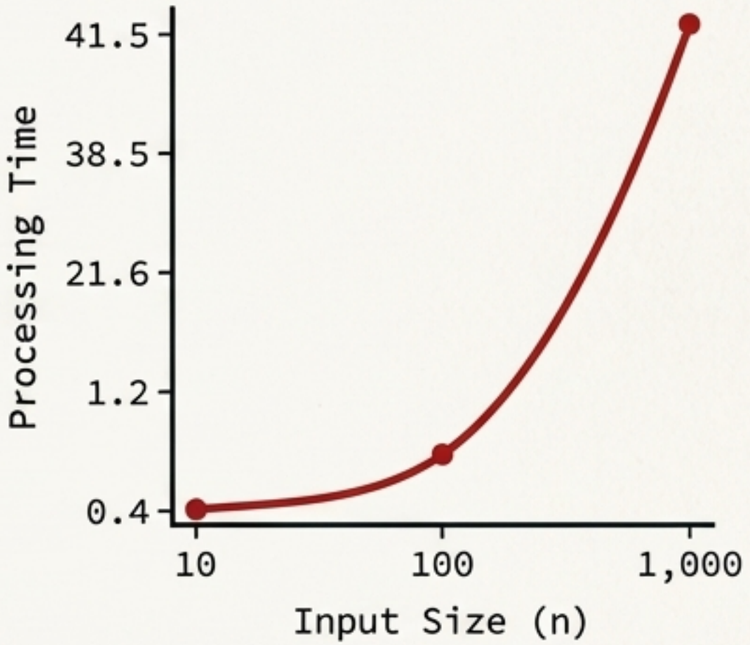
TOP SECRET // NOFORN

DOCUMENT ID: UK-NCGC-DOSS-2025-0052
// DATE OF ISSUE: 22 DEC 2025
// CLASSIFICATION: RESTRICTED

The Kill Curve: Proving Quadratic Slowdown

The per-element cost is *increasing* with scale. That's not linear growth. That's quadratic.

Input Elements	Processing Time
10	0.4 seconds
100	3.2 seconds
1,000	41.5 seconds



"They're looking at processing times measured in *hours*. No wonder it timed out. It wasn't attacked. It was *mathematically murdered*."

TOP SECRET // NOFORN

The Billion-Cycle Loop

DOCUMENT ID: UK-NCGC-DOSS-2025-0053 // DATE OF ISSUE: 22 DEC 2025 // CLASSIFICATION: RESTRICTED

The Vulnerability

CVE-2025-QUAD - Weaponised O(n^2) lookup in value node retrieval.

```python
def get_value_node(self, name: str) -> Optional[ValueNode]:
    """Finds a value node by its name."""
    for node in self.value_nodes:
        if node.name == name:
            return node

    return None
```

Explanation

A `for` loop that iterates over *every node in the graph* just to find a single value. This operation is performed for every attribute on every element.

The Equation

O(N elements * A attributes * N nodes) = **O(N^2)**

TOP SECRET // NOFORN

DOCUMENT ID: UK-NCGC-DOSS-2025-0054
DATE OF ISSUE: 22 DEC 2025
CLASSIFICATION: RESTRICTED

The Algorithmic Kill Chain of APT-N²

RECONNAISSANCE

Fuzz frameworks for months to identify non-linear complexity vulnerabilities.

MODELLING

Test target systems with incrementally-sized payloads to map the precise threshold where performance becomes catastrophic.

WEAPONISATION

Craft legitimate-looking, complex documents optimised to trigger the worst-case slowdown without hitting size limits.

EXECUTION

Launch a coordinated attack against multiple critical targets simultaneously for maximum impact.

Key Attributes: Maximum impact, zero signatures, complete deniability.

TOP SECRET // NOFORN

DOCUMENT ID: UK-NCGC-DOSS-2025-0054
DATE OF ISSUE: 22 DEC 2025
CLASSIFICATION: RESTRICTED

Counter-Intelligence: The Four-Line Fix

The Solution

Replace the iterative search with a pre-computed hash map for O(1) lookups.

Before (Vulnerable)

```
# O(n) lookup
for node in self.value_nodes:
    if node.name == name:
        return node
```

After (Patched)

```
# O(1) lookup
# Pre-computation of the map at initialisation
self.value_nodes_by_name = {
    n.name: n for n in self.value_nodes
}
return self.value_nodes_by_name.get(name)
```

The Takeaway: Four lines of code. That's the difference between a system that scales linearly and one that can be weaponised against itself.

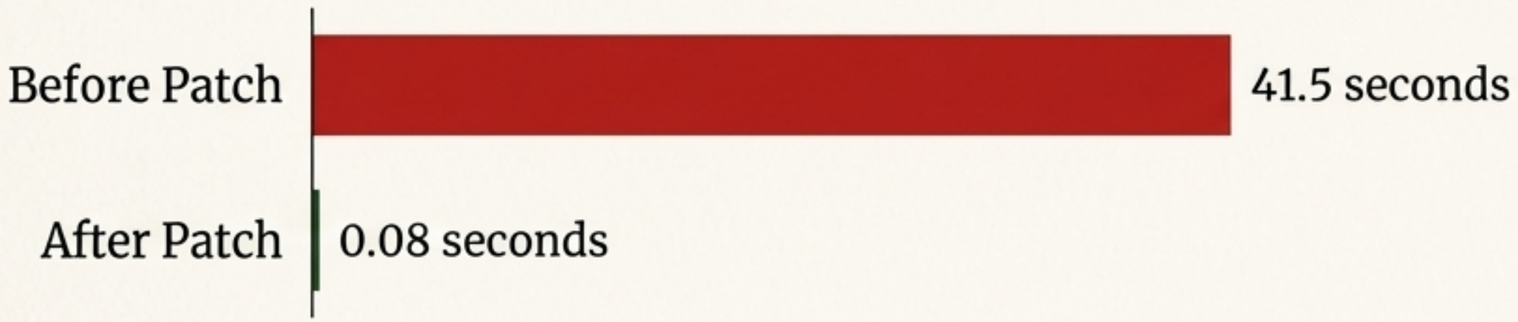
TOP SECRET // NOFORN

From Catatonic to Constant Time: Post-Patch Validation

DOCUMENT ID: UK-NCGC-DOSS-2025-0054
DATE OF ISSUE: 22 DEC 2025
CLASSIFICATION: RESTRICTED

The same 1,000-element document that previously caused catastrophic failure was re-processed on the patched system.

Metric	Before Patch	After Patch
Processing Time	41.5 seconds (#B91C1C)	0.08 seconds (#285430)
System Status	Catatonic / Timed Out (#B91C1C)	Online / Responsive (#285430)



The Archive wasn't compromised. No data was exfiltrated. It was simply… stuck. With the patch, those same documents process in under two seconds."

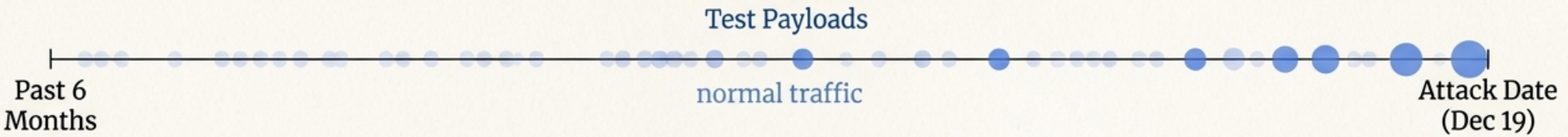
TOP SECRET // NOFORN

DOCUMENT ID: UK-NCGC-DOSS-2025-0054
DATE OF ISSUE: 22 DEC 2025
CLASSIFICATION: RESTRICTED

This Was Not an Attack. It Was an Experiment.

The Revelation

Analysis of six months of logs revealed a pattern of probing.



Key Intelligence

- Payloads were not random; they were precisely “calibrated” for each target.
- APT-N² knew the ***exact threshold*** where O(n^2) performance becomes lethal.

Chilling Conclusion

The final quotes from Stamp:

>That’s not something you learn from source code analysis alone. That’s something you learn from *testing*. ...

>They spent six months mapping the attack surface... **We were their test subjects.**

A Paradigm Shift: Performance is Security

Our threat models are incomplete. We focus on traditional exploits while ignoring vulnerabilities that exist in plain sight within the logic of our own systems.

Traditional Threat Surface	Algorithmic Threat Surface
• Firewall Rules • Authentication • Injection Flaws • Memory Corruption	• Time Complexity ($O(n^2)$) • Resource Scaling • Worst-Case Inputs • `Recursive Depth`

>"APT-N^2 didn't hack us... They simply *used our systems exactly as designed*—but with inputs that exploited mathematical weaknesses we didn't know existed."

DOCUMENT ID: UK-NCGC-DOSS-2025-0054
DATE OF ISSUE: 22 DEC 2025
CLASSIFICATION: RESTRICTED

The Three Pillars of Algorithmic Defence

Pillar 1: Code-Level Audits

Systematically review critical code paths for non-linear complexity ($O(n^2)$, $O(2^n)$, etc.). Treat high-complexity algorithms as potential security bugs.

Pillar 2: Continuous Monitoring

Deploy timing-based observability (like the TIMESTAMP Protocol) to detect complexity anomalies in production before they become critical. Monitor for increasing per-operation latency as data size grows.

Pillar 3: Adversarial Testing

Integrate complexity fuzzing into the CI/CD pipeline. Test systems with deliberately crafted worst-case inputs, not just average-case or ‘happy path’ data.

The New Mandate

The old paradigm was about keeping attackers out. The new paradigm is about preventing our own systems from being turned against us. As Maya Collector asked, ”t asked, “So how do you fight an enemy like that?”

Always check your time complexity.

— Tempus ‘Stamp’ Reyes, NCGC Senior Incident Responder