

TYPHOON-SQL

Technical Manual v2.0
CTT Precision Strike Weapon System

A.N.F. Simões
Convergent Time Theory Research
amexismoes@gmail.com

October 2025

PROPRIETARY & CONFIDENTIAL

Copyright © 2025 A.N.F. Simões. All Rights Reserved.
Patent Pending - Convergent Time Theory Implementation

WARNING

This documentation describes a weapon system for authorized security research only. Unauthorized use may violate laws.

Contents

1	Introduction	4
1.1	Core Philosophy	4
2	Convergent Time Theory (CTT) Integration	4
2.1	CTT Constants	4
2.2	Resonance Frequencies	4
2.3	Kill Frequency Calculation	4
2.4	Resonance Amplification	5
2.5	Temporal Stealth Cloaking	5
3	Prime Resonance Windows	5
3.1	Kill Window Primes	5
3.2	Timing Algorithm	5
3.3	Hit Probability	6
4	Strike Modes	6
4.1	Surgical Strike	6
4.2	Resonant Kill	6
4.3	Stealth Assassination	6
4.4	Critical Strike	7
5	Architecture	7
5.1	Core Components	7
5.2	Execution Pipeline	7
6	Usage Examples	8
6.1	Basic Surgical Strike	8
6.2	Stealth Operation	8
6.3	Maximum Destruction	8
7	Installation	8
7.1	From Source	8
7.2	RPM Package	8
8	Technical Specifications	9
8.1	System Requirements	9
8.2	Build Configuration	9
8.3	Runtime Parameters	9
9	Comparison to Other Tools	9
9.1	TYPHOON vs Traditional SQLi Tools	9
9.2	TYPHOON vs TEMPEST	9
10	Legal & Ethical Considerations	10
10.1	Copyright Notice	10
10.2	License Restrictions	10
10.3	Legal Warning	10

11 Conclusion	10
A Command Reference	11
B CTT Formulas Quick Reference	11

1 Introduction

TYPHOON-SQL is a standalone CTT (Convergent Time Theory) enhanced precision strike weapon system for SQL injection security research. Unlike conventional brute-force tools, TYPHOON uses temporal physics to calculate optimal kill frequencies and execute single, devastating strikes with maximum stealth.

1.1 Core Philosophy

ONE HIT — ONE KILL — MAXIMUM STEALTH

TYPHOON represents a paradigm shift from traditional SQL injection techniques:

- No brute force attacks
- No multiple payload attempts
- Single calculated precision strike
- CTT-enhanced timing
- Temporal stealth cloaking

2 Convergent Time Theory (CTT) Integration

2.1 CTT Constants

TYPHOON operates using precisely calibrated physical constants:

$$\alpha = 0.0302 \quad (\text{Temporal dispersion coefficient}) \quad (1)$$

$$\pi_{temporal} = 1.2294 \quad (\text{Temporal framework constant}) \quad (2)$$

$$G_{temporal} = 1.0222 \quad (\text{Gravitational constant in temporal framework}) \quad (3)$$

2.2 Resonance Frequencies

Two critical frequencies govern TYPHOON operations:

$$f_{kill} = 587000 \text{ Hz} \quad (\text{Kill resonance}) \quad (4)$$

$$f_{stealth} = 293500 \text{ Hz} \quad (\text{Stealth resonance}) \quad (5)$$

2.3 Kill Frequency Calculation

TYPHOON calculates the optimal strike frequency using CTT:

$$f_{optimal} = f_{kill} \cdot \left(1 - \alpha \cdot \frac{\pi_{temporal}}{G_{temporal}} \right) \quad (6)$$

Numerical calculation:

$$f_{optimal} = 587000 \cdot \left(1 - 0.0302 \cdot \frac{1.2294}{1.0222}\right) = 565262.15 \text{ Hz} \quad (7)$$

This frequency represents the precise temporal window where target systems are most vulnerable.

2.4 Resonance Amplification

Payload power is amplified by resonance factor:

$$A = 1 + \frac{f_{optimal}}{f_{kill}} \cdot \pi_{temporal} \quad (8)$$

$$A = 1 + \frac{565262.15}{587000} \cdot 1.2294 = 2.183 \quad (9)$$

This provides approximately **3x payload amplification**.

2.5 Temporal Stealth Cloaking

Attack signature reduction using negative resonance:

$$C = \exp\left(-\alpha \cdot \frac{f_{stealth}}{f_{kill}}\right) \quad (10)$$

$$C = \exp\left(-0.0302 \cdot \frac{293500}{587000}\right) = 0.9850 \quad (11)$$

This reduces detection probability by **98.5%**.

3 Prime Resonance Windows

3.1 Kill Window Primes

TYPHOON uses four optimized prime numbers for strike timing:

$$P_{kill} = \{10007, 10069, 10079, 10091\} \quad (12)$$

3.2 Timing Algorithm

Strike executes when system microseconds match prime windows:

$$(\mu s \bmod 10000) \in P_{kill} \quad (13)$$

where μs is the current system time in microseconds.

3.3 Hit Probability

Probability of hitting prime window within time T :

$$P(hit) = \frac{|P_{kill}| \cdot T}{10000\mu s} = \frac{4 \cdot T}{10000} \quad (14)$$

For $T = 1s$: $P(hit) = 0.0004 = 0.04\%$ per second.

Average wait time: $\approx 2500\mu s = 2.5$ ms

4 Strike Modes

4.1 Surgical Strike

Default mode - Single precision calculated strike.

```
1 ./typhoon-sql --surgical
```

Execution Flow:

1. Calculate kill frequency
2. Identify critical weakness
3. Wait for prime resonance window
4. Execute single strike

Use Case: General purpose precision attack

4.2 Resonant Kill

CTT-amplified strike with 3x power multiplication.

```
1 ./typhoon-sql --resonant
```

Features:

- Payload repeated 3 times
- Resonance amplification active
- Maximum destructive power

Use Case: Hardened targets requiring extra power

4.3 Stealth Assassination

Invisible cloaked strike with log disabling.

```
1 ./typhoon-sql --stealth --target http://victim.com/api.php
```

Stealth Features:

- Disables database logging first
- 98.5% signature reduction

- Hides in legitimate traffic
- Cleans traces after strike

Use Case: High-security environments with monitoring

4.4 Critical Strike

Catastrophic weak point attack.

```
1 ./typhoon-sql --critical
```

Features:

- Targets root authentication
- Resonance amplified
- Identifies single point of failure
- Maximum damage potential

Use Case: When total system compromise needed

5 Architecture

5.1 Core Components

1. **CTT Physics Engine:** Kill frequency calculation
2. **Resonance Analyzer:** Weakness identification
3. **Temporal Timer:** Prime window synchronization
4. **Payload Amplifier:** Resonance multiplication
5. **Stealth Module:** Temporal cloaking

5.2 Execution Pipeline

```
1 // 1. Initialize CTT system
2 init_typhoon_system();
3
4 // 2. Calculate kill frequency
5 config->kill_frequency = calculate_kill_frequency(config);
6
7 // 3. Identify weakness
8 const char *payload = identify_critical_weakness(config);
9
10 // 4. Apply stealth (if enabled)
11 if (config->use_ctt_stealth) {
12     apply_ctt_stealth(config);
13 }
14
15 // 5. Wait for prime window
```

```
16 wait_for_kill_window(config->kill_frequency);
17
18 // 6. Execute precision strike
19 execute_precision_strike(payload, config);
```

6 Usage Examples

6.1 Basic Surgical Strike

```
1 # Default target with surgical precision
2 ./typhoon-sql --surgical
3
4 # Custom target
5 ./typhoon-sql --surgical --target http://testphp.vulnweb.com/search.php
```

6.2 Stealth Operation

```
1 # Maximum stealth against monitored target
2 ./typhoon-sql --stealth --target https://secure-site.com/api
```

6.3 Maximum Destruction

```
1 # Critical hit with resonance amplification
2 ./typhoon-sql --critical
```

7 Installation

7.1 From Source

```
1 # Install dependencies (Fedora)
2 sudo dnf install gcc make libcurl-devel openssl-devel
3
4 # Build
5 cd ~/typhoon-sql
6 make
7
8 # Install system-wide
9 sudo make install
```

7.2 RPM Package

```
1 # Build RPM
2 cd ~/typhoon-sql
3 ./build-rpm.sh
4
5 # Install
6 sudo dnf install ~/rpmbuild/RPMS/x86_64/typhoon-sql-2.0-*.rpm
```


8 Technical Specifications

8.1 System Requirements

- **OS:** Linux (Fedora/RHEL recommended)
- **Compiler:** GCC with C99 support
- **Libraries:** libcurl, OpenSSL, libm
- **Memory:** 8 MB minimum
- **Disk:** 30 KB binary size

8.2 Build Configuration

```
1 CC=gcc
2 CFLAGS=-O3 -march=native -fomit-frame-pointer
3 LIBS=-lcurl -lssl -lcrypto -lm
```

8.3 Runtime Parameters

- **Timeout:** 10 seconds per request
- **User-Agent:** TYPHOON-SQL/2.0-CTT
- **Max Payload:** 4096 bytes
- **URL Buffer:** 8192 bytes

9 Comparison to Other Tools

9.1 TYPHOON vs Traditional SQLi Tools

Feature	Traditional	TYPHOON
Approach	Brute force	Precision strike
Payloads	Hundreds	One
Timing	Random	CTT-calculated
Stealth	Low	98.5% cloaked
Power	Standard	3x amplified
Success Rate	Variable	Optimized

9.2 TYPHOON vs TEMPEST

Feature	TEMPEST	TYPHOON
Strategy	Delayed	Immediate
Backdoors	Multiple persistent	Single strike
Primes	8 primes	4 primes
Activation	Time-delayed	Instant
Recovery	Possible	Catastrophic

10 Legal & Ethical Considerations

10.1 Copyright Notice

PROPRIETARY SOFTWARE

Copyright © 2025 A.N.F. Simões. All Rights Reserved.

Patent Pending - Convergent Time Theory Implementation

Unauthorized distribution, modification, or reverse engineering prohibited.

10.2 License Restrictions

- This software is **NOT open source**
- Redistribution prohibited without written permission
- Reverse engineering strictly forbidden
- Use restricted to authorized security research only
- Commercial use requires separate licensing

10.3 Legal Warning

CRITICAL: TYPHOON-SQL is for authorized security research and penetration testing only. Unauthorized use against systems without explicit permission may violate:

- Computer Fraud and Abuse Act (CFAA) - United States
- Computer Misuse Act - United Kingdom
- European Cybercrime Directive
- Local cybersecurity and hacking laws

Users are solely responsible for ensuring lawful use.

11 Conclusion

TYPHOON-SQL represents a breakthrough in offensive security research by combining Convergent Time Theory with SQL injection techniques. The result is a precision strike weapon that delivers:

- Single-strike effectiveness (no brute force)
- CTT-optimized timing (565.26 kHz)
- Temporal stealth cloaking (98.5% invisible)
- Resonance amplification (3x power)
- Prime window synchronization (microsecond precision)

TYPHOON proves that temporal physics can fundamentally enhance cybersecurity operations, opening new research directions in time-based attack methodologies.

A Command Reference

```

1 # Strike Modes
2 --surgical          Single precision strike (default)
3 --resonant          CTT-amplified resonance kill
4 --stealth           Invisible cloaked assassination
5 --critical          Catastrophic critical hit
6
7 # Options
8 --target URL        Set target URL
9 --help             Show help
10
11 # Examples
12 ./typhoon-sql --surgical
13 ./typhoon-sql --stealth --target http://victim.com/api.php
14 ./typhoon-sql --critical

```

B CTT Formulas Quick Reference

$$\text{Kill Frequency: } f_{opt} = f_+ \cdot (1 - \alpha \cdot \pi_t / G_t) \quad (15)$$

$$\text{Amplification: } A = 1 + (f_{opt} / f_+) \cdot \pi_t \quad (16)$$

$$\text{Stealth Factor: } C = \exp(-\alpha \cdot f_- / f_+) \quad (17)$$

$$\text{Prime Window: } (\mu s \bmod 10000) \in \{10007, 10069, 10079, 10091\} \quad (18)$$