04 Api Documentation

MWRASP Quantum Defense System

Generated: 2025-08-24 18:15:12

TOP SECRET//SCI - HANDLE VIA SPECIAL ACCESS CHANNELS

MWRASP API DOCUMENTATION

Complete REST API, WebSocket, and SDK Reference

API OVERVIEW

Base URL: `https://mwrasp.{environment}.mil/api/v2` **Protocol**: HTTPS only (TLS 1.3 minimum) **Authentication**: OAuth 2.0 / PKI Certificate / API Key **Rate Limiting**: 10,000 requests/minute per client **Timeout**: 30 seconds default, 5 minutes for async operations

Environments

Environment	URL	Purpose
Production	`https://mwrasp.prod.mil/api/v2`	Live operations
Staging	`https://mwrasp.staging.mil/api/v2`	Pre-production testing
Development	`https://mwrasp.dev.mil/api/v2`	Development
Sandbox	`https://sandbox.mwrasp.io/api/v2`	Public testing

AUTHENTICATION

OAuth 2.0 Flow

```
POST /auth/token
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&
client_id=your_client_id&
client_secret=your_client_secret&
scope=read write admin
```

Response

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "Bearer",
  "expires_in": 3600,
  "refresh_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...",
  "scope": "read write admin"
}
```

PKI Certificate Authentication

```
GET /api/v2/system/status
X-Client-Certificate: -----BEGIN CERTIFICATE-----
MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPmljZbyjANBgkqhkiG9w0BAQsF
...
-----END CERTIFICATE-----
```

API Key Authentication

```
GET /api/v2/system/status
Authorization: ApiKey mwrasp_k_live_4242424242424242
```

CORE ENDPOINTS

System Management

Get System Status

```
GET /system/status
```

Response


```
{
  "status": "operational",
  "mode": "quantum defense",
  "threat level": "elevated",
  "uptime seconds": 8640000,
  "version": "2.0.1",
  "components": {
    "quantum detector": "active",
    "fragmentation": "active",
    "agents": "127 active",
    "legal_barriers": "deployed"
  },
  "metrics": {
    "threats_detected": 1847,
```

```
    "attacks_prevented": 1847,
    "false_positives": 2,
    "response_time_ms": 0.73
  }
}
```

Enable System

```
POST /system/enable
Content-Type: application/json

{
  "mode": "quantum_defense",
  "threat_level": "high",
  "auto_response": true,
  "reason": "Elevated threat intelligence"
}
```

Response

```
{
  "success": true,
  "previous_state": "maintenance",
  "current_state": "operational",
  "timestamp": "2024-02-01T12:00:00Z",
  "initiated_by": "admin@organization.mil"
}
```

Emergency Shutdown

```
POST /system/emergency-shutdown
Content-Type: application/json

{
  "confirmation": "CONFIRM EMERGENCY SHUTDOWN",
  "reason": "Compromised credentials suspected",
  "preserve_data": true,
  "notify_contacts": true
}
```

Response

```
{
  "success": true,
  "shutdown_initiated": "2024-02-01T12:00:00Z",
  "data_preserved": true,
  "notifications_sent": 5,
  "recovery_token": "recovery_8f7g6h5j4k3l2m1n"
}
```

Quantum Threat Detection

Get Active Threats

```
GET /threats?status=active&severity=critical&limit=100
```

Response


```
{
  "threats": [
    {
      "id": "threat_q7h8i9j0k1",
      "type": "quantum_computation",
      "algorithm": "shors_factoring",
      "confidence": 0.97,
      "severity": "critical",
      "detected_at": "2024-02-01T12:00:00.073Z",
      "source": {
        "ip": "203.0.113.42",
        "location": "Unknown",
        "quantum_signature": "IBM_Q_System"
      },
      "target": {
        "system": "authentication_server",
        "data": "RSA_4096_keys"
      },
      "response": {
        "action": "temporal_fragmentation",
        "status": "completed",
        "time_to_respond_ms": 73
      }
    }
  ],
  "pagination": {
```

```
    "total": 42,
    "page": 1,
    "per_page": 100,
    "has_more": false
  }
}
```

Analyze Threat

```
POST /threats/analyze
Content-Type: application/json

{
  "data": "base64_encoded_suspicious_data",
  "context": {
    "source": "network_monitor",
    "protocol": "TLS 1.3",
    "timestamp": "2024-02-01T12:00:00Z"
  },
  "deep_analysis": true
}
```

Response

```
{
  "analysis_id": "analysis_a1b2c3d4",
  "threat_detected": true,
  "threat_type": "quantum_key_extraction",
  "confidence": 0.92,
  "details": {
    "quantum_indicators": [
      "superposition_collapse_detected",
      "bell_inequality_violation",
      "speedup_factor_10000x"
    ],
    "algorithm_match": "grovers_search",
    "estimated_qubits": 256,
    "time_to_compromise": "8_seconds"
  },
  "recommendations": [
    "immediate_fragmentation",
    "key_rotation",
    "increase_monitoring"
  ]
}
```

Submit Quantum Canary Token

```
POST /threats/canary
Content-Type: application/json

{
  "token_type": "quantum_superposition",
  "sensitivity": "high",
  "lifetime_seconds": 3600,
  "notification_webhook": "https://soc.organization.mil/webhook"
}
```

Response

```
{
  "canary_id": "canary_x9y8z7w6",
  "token": "eyJzdXBlcnBvc2l0aW9uIjpbMC43MDcsMC43MDdd...",
  "expires_at": "2024-02-01T13:00:00Z",
  "monitoring_active": true,
  "detection_endpoint": "/threats/canary/canary_x9y8z7w6/status"
}
```

Data Fragmentation

Fragment Data

```
POST /fragment/create
Content-Type: application/json

{
  "data": "base64_encoded_sensitive_data",
  "fragment_count": 7,
  "lifetime_ms": 100,
  "overlap_percentage": 15,
  "jurisdictions": ["Switzerland", "Iceland", "Sealand"],
  "quantum_noise": true
}
```

Response

```
{
  "fragmentation_id": "frag_f5g6h7i8",
  "fragments": [
    {
      "id": "fragment_001",
      "index": 0,
      "size_bytes": 1450,
      "jurisdiction": "Switzerland",
      "expires_at": "2024-02-01T12:00:00.100Z",
      "checksum": "sha3_256:abcdef..."
    }
  ],
  "reconstruction_token": "recon_r4s5t6u7",
  "total_fragments": 7,
  "expiration_ms": 100
}
```

Reconstruct Data

```
POST /fragment/reconstruct
Content-Type: application/json

{
  "reconstruction_token": "recon_r4s5t6u7",
  "fragment_ids": [
    "fragment_001",
    "fragment_002",
    "fragment_003",
    "fragment_004",
    "fragment_005"
  ]
}
```

Response (Success)

```
{
  "success": true,
  "data": "base64_encoded_reconstructed_data",
  "fragments_used": 5,
  "reconstruction_time_ms": 8.3
}
```

Response (Failure - Expired)

```
{
  "success": false,
  "error": "FRAGMENTS_EXPIRED",
  "message": "Fragments expired 47ms ago",
  "expired_fragments": ["fragment_002", "fragment_004"]
}
```

Agent Management

List Agents

```
GET /agents?status=active&role=defender&sort=trust_score
```

Response

```
{
  "agents": [
    {
      "id": "agent_alpha_001",
      "role": "defender",
      "status": "active",
      "trust_score": 0.98,
      "spawn_generation": 3,
      "parent": "agent_prime",
      "metrics": {
        "threats_handled": 432,
        "success_rate": 0.997,
        "response_time_avg_ms": 12.3
      },
      "capabilities": [
        "quantum_detection",
        "fragmentation",
        "behavioral_analysis"
      ]
    }
  ],
  "total_agents": 127,
  "by_role": {
    "coordinator": 5,
    "defender": 45,
    "monitor": 40,
    "analyzer": 30,
```

```
    "recovery": 7
  }
}
```

Spawn Agent

```
POST /agents/spawn
Content-Type: application/json

{
  "role": "defender",
  "inherit_from": "agent_alpha_001",
  "mutation_rate": 0.2,
  "resources": {
    "cpu_cores": 4,
    "memory_gb": 16
  }
}
```

Response

```
{
  "agent_id": "agent_beta_042",
  "role": "defender",
  "parent": "agent_alpha_001",
  "generation": 4,
  "status": "initializing",
  "estimated_ready": "2024-02-01T12:00:05Z"
}
```

Agent Consensus Query

```
POST /agents/consensus
Content-Type: application/json

{
  "question": "Should we escalate to critical threat level?",
  "timeout_ms": 500,
  "minimum_participants": 20,
  "weight_by_trust": true
}
```

Response

```
{
  "consensus_reached": true,
  "decision": "escalate",
  "confidence": 0.89,
  "participants": 67,
  "voting_breakdown": {
    "escalate": 58,
    "maintain": 9,
    "de-escalate": 0
  },
  "time_to_consensus_ms": 423
}
```

Legal Barriers

Deploy Legal Protection

```
POST /legal/deploy
Content-Type: application/json

{
  "data_id": "frag_f5g6h7i8",
  "jurisdictions": ["auto_select"],
  "hop_interval_ms": 50,
  "legal_challenges": true
}
```

Response

```
{
  "deployment_id": "legal_l8m9n0o1",
  "jurisdictions_selected": [
    "Switzerland",
    "Iceland",
    "Principality of Sealand",
    "International Waters",
    "Vatican City"
  ],
  "hop_schedule": [
    {"time": "T+50ms", "to": "Iceland"},
```

```
    {"time": "T+100ms", "to": "Sealand"},
    {"time": "T+150ms", "to": "International Waters"}
  ],
  "legal_complexity_score": 9.8,
  "prosecution_difficulty": "technically_infeasible"
}
```

Jurisdiction Status

```
GET /legal/jurisdictions
```

Response

```
{
  "jurisdictions": [
    {
      "name": "Switzerland",
      "privacy_regime": "Enhanced GDPR",
      "data_retention_limit_ms": 100,
      "warrant_required": true,
      "mlat_treaties": ["EU", "US"],
      "response_time_days": 180,
      "current_fragments": 42
    }
  ],
  "total_jurisdictions": 10,
  "fragments_distributed": 284,
  "legal_challenges_active": 7
}
```

Behavioral Authentication

Verify Behavioral Pattern

```
POST /auth/behavioral/verify
Content-Type: application/json

{
  "agent_id": "agent_alpha_001",
  "observed_behavior": {
```

```
    "protocol_order": ["TLS_1.3", "AES_256", "RSA_4096"],
    "packet_rhythm": [100, 100, 200, 100],
    "buffer_size": 8192,
    "error_response_time_ms": 150
  },
  "context": "normal_operations"
}
```

Response

```
{
  "authenticated": true,
  "similarity_score": 0.94,
  "confidence": "high",
  "behavioral_match": {
    "protocol_order": 0.95,
    "packet_rhythm": 0.92,
    "buffer_size": 1.0,
    "error_timing": 0.89
  },
  "anomalies": []
}
```

WEBSOCKET API

Connection

```
const ws = new WebSocket('wss://mwrasp.prod.mil/ws');

ws.on('open', () => {
  ws.send(JSON.stringify({
    type: 'AUTH',
    token: 'your_jwt_token'
  }));
});
```

Event Subscriptions

```
// Subscribe to threat events
ws.send(JSON.stringify({
  type: 'SUBSCRIBE',
  events: ['THREAT_DETECTED', 'QUANTUM_ATTACK', 'AGENT_SPAWNED']
}));

// Receive events
ws.on('message', (data) => {
  const event = JSON.parse(data);

  switch(event.type) {
    case 'THREAT_DETECTED':
      console.log(`Threat: ${event.threat_id}, Severity:
${event.severity}`);
      break;

    case 'QUANTUM_ATTACK':
      console.log(`QUANTUM ATTACK DETECTED! Algorithm:
${event.algorithm}`);
      // Automatic response already initiated
      break;

    case 'AGENT_SPAWNED':
      console.log(`New agent: ${event.agent_id}, Role:
${event.role}`);
      break;
  }
});
```

Real-time Metrics Stream

```
ws.send(JSON.stringify({
  type: 'METRICS_STREAM',
  interval_ms: 1000
}));

ws.on('message', (data) => {
  const metrics = JSON.parse(data);
  if (metrics.type === 'METRICS') {
    updateDashboard({
      threats_per_second: metrics.threats_per_second,
      fragments_active: metrics.fragments_active,
      agents_count: metrics.agents_count,
      response_time_ms: metrics.response_time_ms
    });
```

```
  }
});
```

SDK EXAMPLES

Python SDK

```python
from mwrasp import MWRASPClient

# Initialize client
client = MWRASPClient(
    api_key='mwrasp_k_live_4242424242424242',
    environment='production'
)

# Protect sensitive data
protection = client.protect_data(
    data=classified_document,
    threat_level='critical',
    fragment_count=10,
    lifetime_ms=100,
    jurisdictions=['Switzerland', 'Iceland']
)

print(f"Data protected: {protection.fragmentation_id}")
print(f"Expires in: {protection.lifetime_ms}ms")

# Monitor for threats
@client.on_threat_detected
def handle_threat(threat):
    print(f"Threat detected: {threat.type}")
    print(f"Confidence: {threat.confidence}")
    print(f"Response: {threat.response.action}")

# Start monitoring
client.start_monitoring()
```

JavaScript/Node.js SDK

```
const { MWRASPClient } = require('@mwrasp/sdk');

const client = new MWRASPClient({
  apiKey: 'mwrasp_k_live_4242424242424242',
  environment: 'production'
});

// Async/await pattern
async function protectData() {
  try {
    const protection = await client.fragmentData({
      data: Buffer.from('sensitive data'),
      fragmentCount: 7,
      lifetimeMs: 100,
      quantumNoise: true
    });

    console.log(`Protected with ${protection.fragmentCount}
fragments`);

    // Attempt reconstruction before expiry
    setTimeout(async () => {
      const reconstructed = await client.reconstructData(
        protection.reconstructionToken
      );
      console.log('Data reconstructed successfully');
    }, 50); // 50ms < 100ms expiry

  } catch (error) {
    console.error('Protection failed:', error.message);
  }
}

// Event-driven pattern
client.on('quantumAttackDetected', (attack) => {
  console.log(`QUANTUM ATTACK: ${attack.algorithm}`);
  // MWRASP automatically responds
});

client.on('agentSpawned', (agent) => {
  console.log(`New agent ${agent.id} spawned with role
${agent.role}`);
});

client.startMonitoring();
```

Java SDK

```
 import mil.mwrasp.sdk.MWRASPClient;
import mil.mwrasp.sdk.models.*;

public class MWRASPExample {
    public static void main(String[] args) {
        // Initialize client
        MWRASPClient client = MWRASPClient.builder()
            .apiKey("mwrasp_k_live_4242424242424242")
            .environment(Environment.PRODUCTION)
            .build();

        // Protect data
        ProtectionRequest request = ProtectionRequest.builder()
            .data(sensitiveData)
            .fragmentCount(7)
            .lifetimeMs(100)
            .jurisdictions(Arrays.asList("Switzerland", "Iceland"))
            .build();

        ProtectionResponse response = client.protectData(request);

        System.out.println("Fragmentation ID: " +
response.getFragmentationId());
        System.out.println("Expires at: " + response.getExpiresAt());

        // Set up threat listener
        client.onThreatDetected(threat -> {
            System.out.println("Threat detected: " +
threat.getType());
            System.out.println("Algorithm: " + threat.getAlgorithm());
            System.out.println("Response: " +
threat.getResponse().getAction());
        });

        // Start monitoring
        client.startMonitoring();
    }
}
```

Go SDK

```
 package main

import (
    "fmt"
    "log"
    "github.com/mwrasp/go-sdk"
```

```
)

func main() {
    // Create client
    client, err := mwrasp.NewClient(
        mwrasp.WithAPIKey("mwrasp_k_live_4242424242424242"),
        mwrasp.WithEnvironment("production"),
    )
    if err != nil {
        log.Fatal(err)
    }

    // Protect data
    protection, err := client.FragmentData(&mwrasp.FragmentRequest{
        Data:           []byte("sensitive data"),
        FragmentCount:  7,
        LifetimeMs:     100,
        Jurisdictions:  []string{"Switzerland", "Iceland"},
        QuantumNoise:   true,
    })
    if err != nil {
        log.Fatal(err)
    }

    fmt.Printf("Protected with ID: %s\n", protection.FragmentationID)

    // Monitor threats
    threats := client.MonitorThreats()
    for threat := range threats {
        fmt.Printf("Threat: %s, Confidence: %.2f\n",
            threat.Type, threat.Confidence)
    }
}
```

ERROR HANDLING

Error Response Format

```
 {
  "error": {
    "code": "FRAGMENTS_EXPIRED",
    "message": "The requested fragments have expired and cannot be
reconstructed",
    "details": {
      "expired_at": "2024-02-01T12:00:00.100Z",
```

```
      "attempted_at": "2024-02-01T12:00:00.247Z",
      "expired_fragments": ["fragment_002", "fragment_004"]
    },
    "request_id": "req_r5s6t7u8v9",
    "documentation_url":
"https://docs.mwrasp.mil/errors/FRAGMENTS_EXPIRED"
  }
}
```

Common Error Codes

Code	HTTP Status	Description
UNAUTHORIZED	401	Invalid or missing authentication
FORBIDDEN	403	Insufficient permissions
NOT_FOUND	404	Resource not found
RATE_LIMITED	429	Too many requests
FRAGMENTS_EXPIRED	410	Data fragments have expired
QUANTUM_ATTACK_ACTIVE	503	System defending against quantum attack
INVALID_PARAMETERS	400	Invalid request parameters
INTERNAL_ERROR	500	Internal server error

RATE LIMITING

Default Limits

Endpoint Category	Requests/Minute	Burst
System Status	600	100
Threat Detection	1000	200
Data Fragmentation	500	50
Agent Operations	300	30
Legal Barriers	100	10

Rate Limit Headers

```
 HTTP/1.1 200 OK
X-RateLimit-Limit: 1000
X-RateLimit-Remaining: 999
X-RateLimit-Reset: 1706788860
X-RateLimit-Burst-Limit: 200
X-RateLimit-Burst-Remaining: 199
```

WEBHOOKS

Webhook Configuration

```
 POST /webhooks
Content-Type: application/json

{
  "url": "https://soc.organization.mil/mwrasp-webhook",
  "events": [
    "threat.detected",
    "quantum.attack",
    "fragment.expired",
    "agent.spawned",
    "system.emergency"
  ],
```

```
  "secret": "webhook_secret_key",
  "active": true
}
```

Webhook Payload

```
{
  "id": "evt_e1f2g3h4",
  "type": "quantum.attack",
  "created": "2024-02-01T12:00:00.073Z",
  "data": {
    "threat_id": "threat_q7h8i9j0k1",
    "algorithm": "shors_factoring",
    "confidence": 0.97,
    "response": {
      "action": "temporal_fragmentation",
      "completed": true
    }
  }
}
```

Webhook Signature Verification

```
import hmac
import hashlib

def verify_webhook(payload, signature, secret):
    expected = hmac.new(
        secret.encode(),
        payload.encode(),
        hashlib.sha256
    ).hexdigest()

    return hmac.compare_digest(expected, signature)
```

API VERSIONING

Version Strategy

- **Current Version**: v2
- **Deprecated**: v1 (sunset date: 2024-12-31)
- **Beta**: v3-beta (experimental features)

Version Selection

```
 # Via URL
GET https://mwrasp.prod.mil/api/v2/system/status

# Via Header
GET https://mwrasp.prod.mil/api/system/status
API-Version: 2

# Via Query Parameter
GET https://mwrasp.prod.mil/api/system/status?version=2
```

CHANGELOG

Version 2.0.1 (Current)

- Added quantum canary token endpoints
- Improved fragmentation performance
- Enhanced behavioral authentication
- Added jurisdiction hopping automation

Version 2.0.0

- Major rewrite for quantum defense
- Added temporal fragmentation
- Introduced agent system
- Legal barriers implementation

API Documentation Version: 2.0.1 **Last Updated**: February 2024 **Support**: api-support@mwrasp.mil **Status Page**: https://status.mwrasp.mil

---