Intrusion Detection & Analysis Final Project

Prep by: Haileab Tadele Bekele

Intrusion Detection and Analysis

School of professional Studies

Department of Cybersecurity

Sub to: Prof. Jeff Robertson

Sub Date: Dec 8th, 2025

Background

On July 5th, 2024, a major security flaw at one of the largest cloud communications providers was revealed. This flaw led to a breach of its Authy service due to an unsecured API endpoint. (Security Boulevard, 2024).

Twilio Inc. is an American cloud communications company based in San Francisco, California. It was founded in 2008 by Jeff Lawson, Evan Cooke, and John Wolthuis (Twilio, n.d.). The company offers a platform-as-a-service (PaaS) that allows developers and businesses to add voice, messaging, video, email, and authentication features to their applications through APIs. Twilio's main products include Programmable Messaging, Voice, Flex (a cloud contact center platform), Verify (for two-factor authentication), Lookup (for identity verification), and Segment (a customer data platform) (Twilio, n.d.).

Reported Incident

Twilio, a top cloud communications provider, disclosed that hackers took advantage of an unsecured API endpoint linked to Authy, its well-known two-factor authentication (2FA) app. The breach was initially revealed by the cybercriminal group ShinyHunters, which leaked a CSV file with the compromised phone numbers on BreachForums. (Security Boulevard, 2024).

The CSV file contained 33,420,546 rows of data. This included account IDs, phone numbers, an "over-the-top" column, account statuses, and device counts.

Twilio confirmed the incident. “They said that the unsecured endpoint has been secured. They also found no further access to Twilio's internal systems or other sensitive data.” (Security Boulevard, 2024; TechCrunch, 2024).

Vulnerability / Catalyst for the Incident

Event	Vulnerability	How it was discovered (INT source)
1. Unauthenticated API endpoint (Info-Disclosure)	• The Authy API took requests from phone numbers without even checking who was asking. It told people whether a number was registered or not, which meant people could just try a ton of numbers	• Exposed, attackers could collect phone numbers of Authy users. These numbers are personal information that can be used for phishing or targeted social engineering, including phishing and smishing. • If someone could get in, they could grab Authy users' phone numbers. This is personal info that can be used to trick or scam people.

2. Inadequate Rate Limiting / Lack of Abuse Protection on API	• The exploit involved submitting a large list of phone numbers to the unsecured endpoint. • The hackers just sent a bunch of phone numbers to the unprotected spot.	• Since there wasn't any restriction, the bad guys could easily make a list of Authy users. This allowed them to get data from millions of users instead of just a few single accounts. Instead of being a small problem, this turned it into a big privacy mess.
3. Sensitive Metadata Exposure	• The leaked file had more than just phone numbers. It also had account IDs and other info, such as account statuses and device counts. This gave the hackers even MORE details about the accounts. Like whether they were in use or not and how many devices were tied to them.	• The file also had account IDs and other info, such as account statuses and device counts. This gave the hackers even MORE details about the accounts. Like whether they were in use or not and how many devices were tied to them.

How it All Came Together

- Because the API wasn't verified, hackers could freely get info without any proper checks.
- There weren't any protections, so hackers could easily send long lists of phone numbers and get info for each one, letting them find a LOT of registered users.
- The API didn't just say yes or no if a number was registered. It also gave out personal info such as account IDs, device details, and statuses.

In result, a simple misconfiguration, authenticated endpoint in conjunction with missing protections like iteration limit and protection, along with substantial amount of data exposure through detailed metadata, made the simple security incident into a major breach.

The technical vulnerability record describes this as an information-disclosure vulnerability with a CVSS score of medium (5.3). However, in practical application, particularly regarding 2FA user data, the scale and context can significantly amplify its impact.

Artifacts

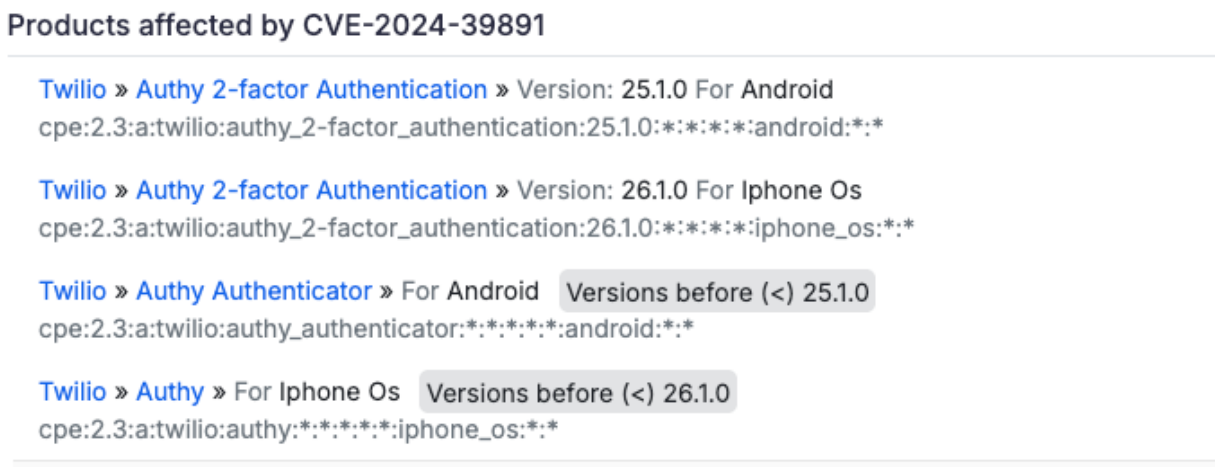
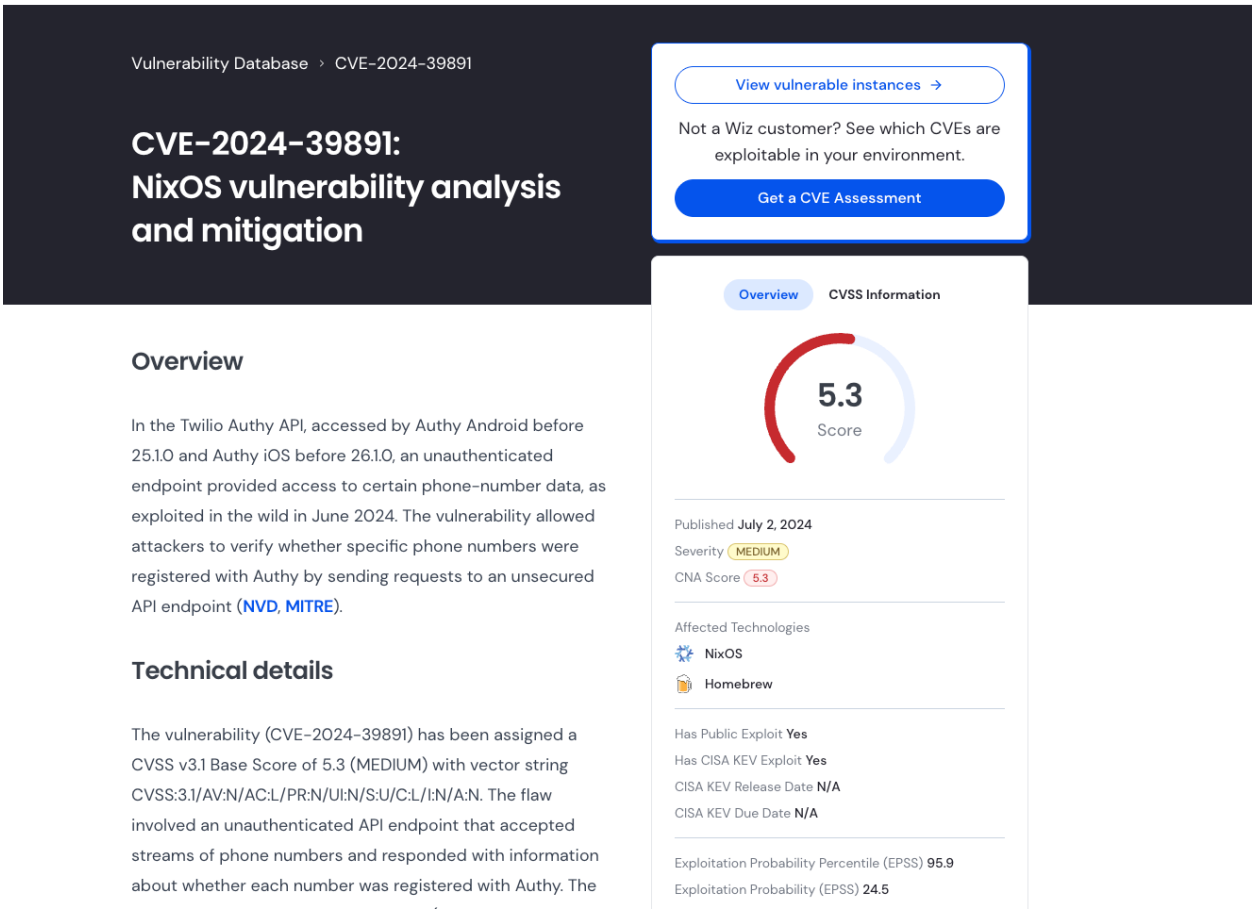


Figure 1 Affected Products



Detailed Analysis: Twilio Authy API Breach

1. Unauthenticated API Endpoint (Info-Disclosure via iteration/enumeration)

Vulnerability Description

The Authy API had an unauthenticated endpoint that allowed phone number queries and returned registration status without needing authentication or authorization. In short, the Authy API let people ask about phone numbers and get back registration info without needing to log in or prove

who they were. This flaw let attackers check if specific phone numbers were registered with Authy, leading to mass enumeration attacks.

Risk Assessment

Risk: Attackers could easily check if phone numbers were on Authy and get many numbers at once.

MITRE ATT&CK Framework Mapping:

- Technique: T1589.002 - Gather Victim Identity Information: Email Addresses (adapted for phone numbers)
- Tactic: Reconnaissance
- Sub-technique: Phone number enumeration for account validation

Threat Intelligence Analysis:

- Threat Actors: Likely cybercriminal groups looking for databases for phishing campaigns, smishing operations, or credential stuffing attacks
- Attack Vector: Automated enumeration via API requests
- Data Sensitivity: High - phone numbers are PII and can be used for multi-vector attacks
- Exploit Complexity: Low - requires basic scripting knowledge
- Attack Surface: External-facing API with no authentication barrier

Business Impact:

- Regulatory compliance violations (GDPR Article 32, CCPA)
- Reputational damage to Twilio brand
- Potential liability for downstream attacks on users
- Loss of customer trust in 2FA security provider

2. Lack of Abuse Protection on API

Vulnerability Description

The Authy API did not have enough rate limiting controls. This allowed attackers to send many phone number queries without throttling or blocking. The lack of abuse protection made it possible for enumeration attacks to grow from targeting individual accounts to compromising data for millions of users.

Risk Assessment

MITRE ATT&CK Framework Mapping: (MITRE, 2024)

- Technique: T1499.004 - Endpoint Denial of Service: Application or System Exploitation
- Tactic: Impact (though primarily used for reconnaissance in this case)
- Sub-technique: API abuse through unlimited requests

Business Impact Multiplier:

- Turns an isolated incident into a widespread breach affecting over 33 million users
- Causes a sharp rise in regulatory penalties (per-record fines under GDPR/CCPA)
- Creates exposure to class-action lawsuits due to scale
- Puts Twilio at a disadvantage in a security-sensitive 2FA market

Threat Intelligence Analysis:

- Attack Amplification: Changes a single-account vulnerability into a massive data breach

• Attack Pattern: Distributed enumeration from multiple IP addresses to avoid basic detection

• Automation Level: Fully automated - likely using botnets or cloud infrastructure

• Dwell Time: Unknown, but likely operated for a long period to collect over 33 million records

• Detection Evasion: Lack of rate limiting suggests no anomaly detection was triggered

3. Sensitive Metadata Exposure

Vulnerability Description

The compromised CSV data contained not only phone numbers but also rich metadata like account IDs, "over-the-top" (OTT) status flags, account status indicators, and device count information. This added context increased the risk by giving attackers actionable information for targeted attacks, account takeover attempts, and social engineering campaigns.

Risk Assessment

MITRE ATT&CK Framework Mapping: (MITRE, 2024)

- Technique: T1592.004 - Gather Victim Network Information: Client Configurations
- Tactic: Reconnaissance

Supporting Techniques: (MITRE, 2024)

- T1589.001 - Credentials (account status helps with credential stuffing prioritization)
- T1598 - Phishing for Information (metadata supports targeted social engineering)

Business Impact:

• Downstream Liability: If exposed metadata leads to attacks on user accounts beyond Authy

• Regulatory Scrutiny: Metadata exposure may break data minimization rules (GDPR Article 5)

• Trust Erosion: A security company leaking security-related metadata is particularly harmful

• Competitive Risk: Metadata about the customer base is visible to competitors

Threat Intelligence Analysis:

Metadata Risk Multipliers:

1. Account IDs:

- Enable correlation across data breaches
- Can lead to account takeover if IDs are predictable
- Facilitate targeted attacks on high-value accounts

2. Account Status (Active/Inactive):

- Active accounts are more valuable targets for attackers
- Inactive accounts might indicate abandoned email addresses (easier to hijack)
- Prioritize credential stuffing campaigns

3. Device Count:

- One device indicates an individual user, easier target
- Multiple devices suggest a business account or high-value individual

- Zero devices indicate a dormant account, possible target for account resurrection attacks

Recommended Response or Mitigation

While the technical vulnerability (CVE-2024-39891) is rated medium severity (CVSS 5.3), the business impact is severe because of the potential for exposure, damage to the reputation of a key security service, and an increased risk of phishing and smishing attacks against affected users. This incident highlights serious issues in API security management for a sensitive authentication service.

The recommended response and mitigation take in 2 steps

1. addressing the identified 3 vulnerabilities and events
2. addressing it from managerial point of view

Vulnerabilities and events

1. For the Unauthenticated API Endpoint (Information Disclosure)

Industry Best Practices (OWASP API Security Top 10, NIST SP 800-204):

- Implement Strict Authentication and Authorization: Follow the principle of "Never trust, always verify." All API endpoints, especially those that return user data, must require authentication. Use strong standards like OAuth 2.0 with scopes or API keys linked to limited access.
- Conduct Rigorous Security Testing: Integrate SAST (Static Application Security Testing) and DAST (Dynamic Application Security Testing) into CI/CD pipelines to identify misconfigured endpoints before deployment. Manual penetration testing should focus on "shadow" or undocumented APIs.

- Adopt a Zero-Trust Architecture: Treat all APIs as untrusted. Use consistent identity-aware proxy layers (like API Gateways) to enforce authentication and policy. This keeps business logic separate from security controls.

Immediate Actions:

- Audit all external-facing APIs for gaps in authentication, focusing on "lookup" or "check" functions.
- List and document all API endpoints; remove unused or undocumented ones.
- Require mandatory authentication schemas in API definitions (OpenAPI/Swagger).

2. For Inadequate Rate Limiting and Abuse Protection

Industry Best Practices (OWASP API8:2019 - Injection, NIST SP 800-190):

- Implement Multi-Factor Rate Limiting: Set limits based on user/API key, IP address, and location. Use algorithms like token bucket or fixed window. Limits should be strict for endpoints related to authentication or identity.
- Deploy Advanced Bot Management and Anomaly Detection: Use solutions that identify clients (like JA3, HTTP/2 fingerprints) and spot unusual activity (sudden spikes from one source) or behavioral patterns (like number enumeration).
- Leverage API Gateways: Use gateways (like AWS WAF/CloudFront, Azure API Management, Apigee) for centralized rate limiting, threat detection, and logging.

Immediate Actions:

- Define and enforce strict rate limits for all public APIs (like 100 requests/hour/user, 1,000/day/IP).

- Use progressive responses: Implement CAPTCHA challenges after soft limits and hard blocks after reaching abusive thresholds.
- Check API logs for enumeration patterns (like sequential requests or number lists).

3. For Sensitive Metadata Exposure

Industry Best Practices (GDPR Principle of Data Minimization, ISO/IEC 27001:2022 Annex A):

- Apply Data Minimization: APIs should return the minimum data needed for the client's function. For phone number registration checks, return a boolean ({"registered": true/false}), not a full profile.
- Classify Data and Implement Tiered Access: Classify data fields (like phone number = PII, device count = internal metadata). Enforce stricter access controls for metadata, possibly requiring a higher authorization level.
- Mask or Hash Identifiers: When internal IDs must be exposed, use unpredictable, non-sequential tokens (UUIDs) instead of incremental integers.

Immediate Actions:

- Review all API responses with data classification in mind. Remove unnecessary metadata from public endpoints.
- Validate response shapes in testing to avoid data schema "creep."
- Encrypt sensitive fields both at rest and in transit, ensuring keys are managed separately.

STRATEGIC RECOMMENDATIONS FOR LEADERSHIP

Hence the leadership is the one on public scrutiny I recommend the leadership to take in the following steps for strategic leadership and governance/leadership

1. **Governance:** Set up an API Security Center of Excellence to define and enforce security standards across all development teams.

2. **Shift-Left Security:** Require security-by-design workshops for all product teams building APIs. Include threat modeling for new features.

3. **Continuous Monitoring:** Invest in dedicated API Security Posture Management (APM) tools to constantly discover, test, and monitor APIs for misconfigurations and unusual traffic.

4. **Incident Response Preparedness:** Update incident response playbooks to include API-specific breaches. Clearly define escalation paths and communication plans for when API abuse is detected.

5. **Third-Party Risk:** Apply the same standards to APIs from third parties. Understand their security posture as part of vendor risk assessments.

References

- Security Boulevard. (2024). Twilio's Authy breach: The attack via an unsecured API endpoint. https://securityboulevard.com/2024/07/twilios-authy-breach-the-attack-via-an-unsecured-api-endpoint/
- Whittaker, Z. (2024). Twilio says hackers identified cell phone numbers of two-factor app Authy users. TechCrunch. https://techcrunch.com/2024/07/03/twilio-says-hackers-identified-cell-phone-numbers-of-two-factor-app-authy-users/
- Wiz. (2024). CVE 2024 39891 — Impact, exploitability, and mitigation steps. Wiz Vulnerability Database. https://www.wiz.io/vulnerability-database/cve/cve-2024-39891

- MITRE. (2024). MITRE ATT&CK® framework. https://attack.mitre.org/

- OWASP. (2023). OWASP API security top 10. https://owasp.org/API-Security/editions/2023/en/0x00-header/
- National Institute of Standards and Technology. (2022). SP 800-204: Security strategies for microservices-based application systems. U.S. Department of Commerce. https://doi.org/10.6028/NIST.SP.800-204
- Twilio. (n.d.). Company overview. Retrieved from https://www.twilio.com/company
- YouTube Infostealer Logs Unmask Pedophiles, Twilio API Hack, Rockwell Device Vulnerabilities: https://www.youtube.com/watch?v=cEplej8E11kn
- YouTube Twilio Denies Massive Data Hack: Twilio is pushing back against claims that 848,000 customers were impacted by a breach: https://www.youtube.com/watch?v=64lVym98VPM