How to Investigate Data Breaches Using OSINT: A Step-by-Step Tutorial

Maria Cattini | 12/02/2026 | CYBERSECURITY

The Substack breach of February 2025 exposed 700,000 user records. Within hours, independent researchers had mapped the attack surface, identified the threat actor's profile, and traced leaked data across multiple forums—all without breaking a single law.

This tutorial walks you through the exact OSINT methods used to investigate real breaches. You'll learn how to gather intelligence ethically, validate leaked data claims, and build a comprehensive breach timeline using only public sources.

Why OSINT Matters for Breach Investigation

Traditional breach response relies on internal logs and forensic tools. But by the time your security team detects an incident, threat actors have often been selling stolen data for weeks.

OSINT flips this model. Public sources—cybercrime forums, paste sites, search engine caches—frequently contain early breach indicators. The Substack case proves this: forum posts appeared before official disclosure, giving researchers a three-month head start.

Three reasons OSINT beats reactive security:

Speed. Threat intelligence from public sources updates in real-time. No waiting for vendors or internal analysis.

Scope. You see the attacker's full campaign—not just what touched your network. Previous victims, attack patterns, tool preferences.

Cost. Every technique here uses free tools. No enterprise licenses required.

The Substack Breach: A Real-World Case Study

On February 3rd, 2026, Substack disclosed a breach affecting 700,000 accounts. The company revealed email addresses, phone numbers, and internal metadata were accessed in October 2025—four months before discovery.

CEO Chris Best's notification included careful language: "credit card numbers, passwords, and financial information were not accessed." But security researcher Arvid Kahl highlighted the real risk: email + phone number combinations enable SIM swapping attacks and sophisticated phishing campaigns.

The breach timeline:

• October 2025: Initial unauthorized access (company claims)
• Late January 2026: Threat actor posts data sample on cybercrime forum
• February 3rd, 2026: Substack confirms breach and notifies users

• February 5th, 2026: Full dataset discussion appears across security communities

This four-month gap between compromise and detection is typical. OSINT practitioners monitoring breach forums spotted the Substack data weeks before official confirmation.

Let's reconstruct this investigation using only public methods.

Step 1: Google Dorking for Initial Indicators

Google Dorking uses advanced search operators to find information companies didn't mean to expose. When investigating potential breaches, these operators uncover vulnerable systems, leaked credentials, and exposed databases.

Finding Exposed Substack Infrastructure

Start with basic reconnaissance to map the attack surface:

```
site:substack.com filetype:pdf confidential
site:substack.com inurl:admin
site:substack.com intext:"index of" "backup"
```

These queries reveal administrative panels, backup directories, or sensitive documents. Nothing here constitutes hacking—you're simply using Google's index more precisely than the average user.

Each operator serves a specific purpose. The site: operator limits results to a single domain. The filetype: operator finds specific document types—PDFs often contain internal presentations or reports never meant for public viewing. The inurl: operator searches within web addresses, where administrators sometimes leave test environments or staging servers accessible.

For the Substack investigation, researchers likely used temporal operators to find cached pages:

```
site:substack.com after:2025-10-01 before:2025-11-01
```

This narrows results to the breach window. Cached pages sometimes preserve evidence removed from live sites. When companies discover exposed information, they remove it quickly. But Google's cache and the Internet Archive preserve snapshots. A page deleted on November 1st might still exist in Google's cache from October 15th—right in the middle of the breach period.

Advanced operators combine for precision targeting:

```
site:substack.com (inurl:admin OR inurl:dashboard) -login
site:substack.com intitle:"index of" +"parent directory"
site:substack.com ext:sql OR ext:db OR ext:log
```

The first query finds administrative interfaces that don't require login. The second locates directory listings exposing file structures. The third searches for database files or logs mistakenly left in web-accessible locations.

Real breach investigations often start with these reconnaissance queries. You're not exploiting vulnerabilities—you're documenting what's already publicly indexed. The distinction matters both legally and ethically.

Searching for Data Leaks

When breach claims surface, validate them before spreading misinformation:

```
"@substack.com" site:pastebin.com
"@substack.com" site:ghostbin.com
intext:"substack" intext:"database" filetype:sql
```

Legitimate breach data rarely appears on paste sites immediately. But threat actors testing stolen credentials often drop small samples to prove authenticity. Finding these samples confirms breach claims before official disclosure.

Ethical boundary: Looking at publicly indexed paste sites is legal. Downloading databases to verify passwords crosses into gray areas. Stay on the research side of this line.

Step 2: Monitoring Breach Forums and Marketplaces

Cybercrime forums are where stolen data gets traded. Monitoring these sources requires understanding their culture and communication patterns.

Where to Look

The Substack data appeared on a known breach marketplace (exact name withheld). Similar platforms include:

- BreachForums successors: The original shut down, but multiple clones operate
- Exploit.in: Russian-language forum with English section
- Nulled.to: Large community with data leak sections
- Telegram channels: Increasingly popular for quick data drops

Access without participation: Most forums allow read-only access or require minimal engagement. Never purchase data, trade information, or contribute to criminal activity. Observation for research purposes falls under OSINT.

Understanding forum culture matters. These communities have hierarchies, trusted members, and verification processes. New members claiming major breaches without proof get dismissed. Established actors post partial samples first—enough to prove authenticity without giving away the full dataset for free.

Forum structure varies. Some require cryptocurrency payments to access premium sections. Others use reputation systems where members earn trust through contributions. For OSINT purposes, free tiers usually suffice. Major breach announcements appear in public sections to attract buyers.

Registration often requires operational security awareness. Using your real email address or work account is amateurish and potentially dangerous. Researchers create compartmentalized identities—separate email addresses, VPN connections, and browsing environments. This protects both your organization and your personal safety.

The goal isn't deep infiltration. You're monitoring, not participating. Think of it like watching a public street from a window—legal observation, not trespassing.

What to Look For

When a threat actor claims a breach, they post proof—usually a small data sample. The Substack post likely included:

• Record count (700,000 in this case)
• Data types (emails, phone numbers, metadata)
• Breach date claim (October 2025)
• Sample records (partial or full)

Cross-reference these claims with known Substack user patterns. Do email formats match? Are phone numbers in plausible formats? Experienced researchers spot fabricated breaches quickly.

Validating Claims

Before accepting breach claims as legitimate:

1. Check sample data authenticity. Do emails follow company patterns? Real Substack addresses typically look like "newsletter-name@substack.com" or "username@customdomain.com" for paid creators.
2. Search for corroboration. Has anyone else verified these records? Security researchers often share validation (without exposing full records) on Twitter or security forums.
3. Timeline plausibility. Does the claimed breach date align with system downtime, maintenance windows, or public incidents?

The Substack case passed all checks. Researchers verified email patterns, and the October 2025 date aligned with retrospective analysis of company communications.

Step 3: Analyzing Metadata and Attribution

Understanding *who* conducted the breach is as important as validating *what* was stolen. Threat actor attribution relies on pattern analysis across multiple breaches.

Techniques for Attribution

Language analysis. Forum posts contain clues. Grammar mistakes, idioms, and time zone references narrow possible origins. The Substack actor's English proficiency suggests native speaker or experienced operative.

Historical patterns. Search the threat actor's username across platforms:

```
"[username]" site:reddit.com
"[username]" site:twitter.com
"[username]" breach
```

Experienced actors reuse handles. Finding previous breaches credited to the same user reveals their specialization, target preferences, and sophistication level.

Technical indicators. How was data obtained? The metadata column in leaked Substack records suggests API exploitation rather than SQL injection. This detail matters—API breaches indicate different vulnerability classes than database dumps.

Building a Threat Profile

For the Substack breach, OSINT analysis revealed:

• Actor sophistication: Medium-high (maintained access for months undetected)
• Motivation: Financial (data posted for sale, not freely leaked)
• Previous activity: Linked to at least two similar breaches in 2024-2025
• TTPs: Prefers API exploitation over brute force methods

This profile helps other potential targets assess their risk. Companies using similar platforms or API architectures should review access controls immediately.

Building comprehensive threat profiles requires patience and cross-referencing. Start with the basics: username, posting patterns, language preferences, time zones. A threat actor posting consistently between 9 AM and 5 PM Moscow time likely operates from that region or uses it as cover.

Look for technical tells in their posts. Do they make mistakes revealing infrastructure knowledge? When discussing the breach, do they demonstrate insider understanding of specific systems? The Substack actor mentioned "internal metadata" before official disclosure specified what that meant—suggesting either access to company documents or technical knowledge of their database structure.

Previous breach connections matter most. If the same actor sold data from three newsletter platforms in 2024-2025, they're specializing. They've likely developed specific tools or techniques for this sector. Other newsletter platforms become obvious targets. This pattern recognition helps security teams prioritize defensive measures.

Track their business model. Some actors sell data immediately for quick profits. Others hold datasets, waiting for bidding wars. The Substack actor posted samples in late January but the full dataset wasn't widely available for weeks—suggesting selective sales to high-paying buyers first. This behavior indicates experience and business sophistication.

Step 4: Timeline Reconstruction Using Web Archives

The Internet Archive's Wayback Machine and archive.today preserve website snapshots. These archives help reconstruct events when companies delete evidence or revise statements.

Checking Substack's Response Evolution

Compare archived security pages before and after breach disclosure:

```
https://web.archive.org/web/*/substack.com/security
```

Did Substack update security documentation after October 2025? Major changes might indicate awareness of the breach before public disclosure.

Similarly, capture and archive the CEO's breach notification email. Companies sometimes edit official communications after backlash. Having the original version matters for accountability and accurate reporting.

Finding Deleted Evidence

Cached pages and archives sometimes preserve:

- Removed blog posts discussing security incidents
- Deleted social media responses to user complaints
- Earlier versions of privacy policies or data handling practices

Archive everything immediately after a breach becomes public. Within days, relevant pages often disappear or get modified.

Step 5: Correlating Multiple Intelligence Sources

Professional OSINT investigations combine multiple sources to build complete pictures. No single method reveals the full story.

Creating an Investigation Matrix

Track findings across channels:

Source	Information Found	Validation Status	Date Discovered
Breach forum	Initial claim, 700k records	Sample verified	Jan 28, 2026
Google cache	Possible vuln disclosure	Circumstantial	Feb 1, 2026
Twitter	User complaints about spam	Supports timeline	Oct-Nov 2025
Wayback Machine	Security page updates	Confirmed changes	Feb 4, 2026

This matrix helps spot contradictions and gaps. For example, if users complained about increased spam in November 2025, the October breach date gains credibility.

Red Flags and False Positives

Not every data leak is real. Watch for:

- Recycled breaches: Old databases reposted as "new" breaches
- Fabricated samples: AI-generated email addresses passing casual inspection
- Attention seekers: Unverified claims from accounts with no breach history

The Substack breach avoided these red flags. Multiple independent researchers verified the data, samples matched real user patterns, and the threat actor had credible history.

Practical Tools for OSINT Breach Investigation

Beyond Google Dorking, these tools streamline investigation workflows:

Free Intelligence Platforms

Have I Been Pwned (haveibeenpwned.com): Check if specific email addresses appear in known breaches. The creator, Troy Hunt, often adds newly confirmed breaches within days.

The service works through aggregation. When major breaches occur and data becomes public, Hunt verifies authenticity and adds records to his database. You can check individual email addresses or use the API to monitor multiple accounts. The Notify Me feature sends alerts when your email appears in new breaches—crucial for both personal security and organizational monitoring.

For researchers, HIBP provides breach timelines and affected service lists. If you're investigating a potential breach, checking whether similar services appear in HIBP's database reveals patterns. Three newsletter platforms breached in 2025? That's not coincidence—it's a campaign.

DeHashed (dehashed.com): Searches multiple breach databases simultaneously. Free tier limited but useful for quick checks.

Unlike HIBP's email-focused approach, DeHashed searches across multiple data points: usernames, email addresses, phone numbers, addresses, IP addresses. This broader search capability helps when investigating breaches where email addresses aren't the primary identifier. The Substack breach included phone numbers—DeHashed could correlate those numbers with other breached services, revealing the full scope of affected users' exposure.

BreachDirectory: Community-maintained breach aggregator. Less reliable than HIBP but sometimes faster to index new leaks.

This grassroots database accepts submissions from security researchers. Quality varies, but speed matters in breach investigations. When racing to validate claims, BreachDirectory sometimes surfaces data hours before major services. Cross-reference anything found here with more authoritative sources before relying on it.

Social Media Monitoring

TweetDeck/Twitter Advanced Search: Track breach-related keywords in real-time:

```
substack breach OR leak OR hacked -retweet
```

Remove retweets to see original reports. Security researchers often discuss breaches hours before mainstream media coverage.

Twitter's Advanced Search supports complex queries. Combine keywords with date ranges, language filters, and account specifications. Search tweets from verified security researchers only: from:troyhunt OR from:GossiTheDog substack breach. This filters signal from noise.

Set up saved searches or use TweetDeck columns for continuous monitoring. When a breach claim surfaces, you'll see it in real-time—not hours later when it's trending. Those early hours matter most for investigation and user notification.

Reddit monitoring: Subreddits like r/cybersecurity, r/netsec, and r/privacy frequently discuss breaches early. Search subreddit history for related discussions.

Reddit's community-driven nature means technical discussions happen before polished articles appear. Security professionals share initial analysis, debate authenticity, and post validation methods. The comment sections often contain more valuable intelligence than the original posts.

Use Reddit's search with timestamp filters: timestamp:1704067200..1706745600 searches posts between specific Unix timestamps. This temporal filtering helps reconstruct who knew what when—crucial for timeline analysis.

Archive and Documentation

archive.today: Faster than Wayback Machine for immediate archival. Create snapshots of breach claims, official statements, and related pages.

When investigating breaking breaches, archive everything immediately. Companies edit press releases, threat actors delete posts, forum moderators remove evidence. Archive.today captures pages in seconds—much faster than waiting for the Wayback Machine to crawl.

The service also bypasses some paywalls and anti-scraping measures, though this functionality exists primarily for archival purposes. Each archived page receives a unique URL you can share with other researchers without directing them to potentially malicious sites.

Hunchly (web archive tool): Designed for investigators. Automatically captures browsing history with searchable archives. Paid tool but worth it for serious investigations.

Hunchly runs as a browser extension, silently recording every page you visit during investigations. Later, search your entire browsing history by keyword, date, or domain. This proves invaluable when you remember reading something relevant but can't recall where.

The tool generates timestamped PDF reports of your investigation path—useful for court cases,

internal reviews, or simply documenting your methodology. For professional investigators, the $130 annual cost pays for itself in saved time during the first investigation.

Ethical and Legal Boundaries

OSINT operates in a gray zone between public research and potential criminal activity. Understanding these boundaries protects you legally and ethically.

What's Legal

- Viewing publicly accessible websites and forums
- Using search engines to find indexed information
- Reading breach notifications and security advisories
- Analyzing publicly posted data samples (not downloading databases)
- Archiving public web pages for documentation

What's Not Legal

- Accessing systems without authorization (even if credentials are public)
- Downloading or distributing stolen databases
- Using breached credentials to test access
- Participating in data trading or sales
- Doxing individuals using leaked information

The Substack investigation stayed ethical. Researchers verified claims using small samples, documented findings publicly, and notified potentially affected users—never distributing the full database.

Responsible Disclosure

When OSINT uncovers unreported breaches:

1. Verify thoroughly. False claims damage your credibility and cause unnecessary panic.
2. Notify the company privately first. Give them reasonable time to respond (typically 7-30 days).
3. Document everything. Your communications, their responses, and your findings.
4. Disclose responsibly. Share enough detail to inform users without enabling further exploitation.

The information security community respects researchers who follow these guidelines. Companies sometimes reward ethical disclosure through bug bounty programs.

Lessons from the Substack Breach

This case study reveals several broader security patterns worth understanding:

Why Detection Took Four Months

The October to February gap isn't unusual. Sophisticated actors maintain persistent access while exfiltrating data slowly to avoid triggering rate limits or anomaly detection. Substack's systems likely showed nothing obviously wrong until someone noticed the data for sale.

This delayed detection makes OSINT crucial. Breach forums provided earlier warning than internal security tools.

The Email + Phone Number Risk

Substack's reassurance about "no passwords or financial data" missed the real threat. Combining

email addresses with phone numbers enables:

• SIM swapping attacks: Convince carriers to port numbers, then bypass 2FA
• Targeted phishing: Messages personalized with real email and SMS channels
• Account takeover chains: Use one compromised service to attack others

Security researcher Arvid Kahl correctly identified this risk. When companies downplay "minor" data exposure, threat actors know better.

Platform-Wide Implications

Substack isn't unique. Similar newsletter platforms, content management systems, and creator economy tools handle identical data types. Any of them could face parallel breaches.

If you use these services, the Substack case offers clear lessons: enable all available security features, use unique passwords, and monitor breach notification services actively.

Building Your OSINT Investigation Workflow

Investigating breaches requires systematic approaches. Here's a starter workflow based on real research patterns:

Phase 1: Initial Detection (Days 1-2)

• Monitor breach forums and Telegram channels daily
• Set up Google Alerts for "[company name] breach OR leak"
• Check HIBP and DeHashed for new breach additions
• Watch security researcher Twitter lists for early discussions

Detection speed determines everything. The gap between breach occurrence and discovery belongs to threat actors. The gap between threat actor disclosure and public awareness belongs to researchers monitoring the right channels.

Build a monitoring routine you can sustain. Check three breach forums every morning with coffee. Scan your Twitter security list during lunch. Review Google Alerts before leaving work. Consistency beats intensity—daily 15-minute checks outperform sporadic deep dives.

Automation helps. RSS feeds from security blogs, Telegram bots monitoring keywords, browser extensions alerting on breach mentions. The OSINT Framework lists dozens of monitoring tools. Pick three that fit your workflow and master them.

Phase 2: Validation (Days 3-5)

• Examine posted samples for authenticity markers
• Cross-reference claimed breach dates with public events
• Search for corroborating reports from multiple sources
• Archive all findings immediately

Validation separates professionals from amateurs. Anyone can screenshot a breach claim. Researchers verify before amplifying. The Substack case included multiple validation layers: sample email formats matched known patterns, metadata fields aligned with Substack's database structure, and timing correlated with user complaints about increased spam.

Create a validation checklist. Does the sample data look real? Do field names match the company's known database structure? Is the claimed breach date plausible given maintenance windows or public incidents? Have other researchers verified independently?

Document your validation process. Screenshots of original claims, notes on verification methods, timestamps of when you confirmed each element. This documentation protects you from accusations of spreading misinformation and provides a paper trail if your findings later become legal evidence.

Phase 3: Deep Analysis (Days 6-10)

• Attempt threat actor attribution through username research
• Build timeline using web archives and cached pages
• Analyze technical details to understand attack vectors
• Document methodology for potential publication

Deep analysis distinguishes basic reporting from intelligence work. Now you're not just confirming a breach happened—you're understanding how, why, and what it means for future security.

Threat actor attribution requires patience. Search their username across platforms going back months or years. Read their posting history. What services do they target? What tools do they mention? Do they sell data quickly or hold it? Understanding their methods and motivations helps predict their next moves.

Timeline reconstruction combines multiple sources. Company status pages might show unusual maintenance during the breach period. User forums might have complaints about system behavior. Web archives preserve deleted security blog posts. Piece together the full story from fragments scattered across the internet.

Phase 4: Reporting (Days 11+)

• Compile findings into structured intelligence report
• Contact affected company if they haven't disclosed publicly
• Share appropriate details with security community
• Monitor company response and update findings

Reporting requires careful judgment. What information helps defenders without enabling attackers? The Substack investigation shared enough detail for users to assess their risk without posting the stolen database or detailed exploitation techniques.

Structure matters. Executive summary for time-constrained readers, detailed methodology for security professionals, actionable recommendations for affected users. Good breach reports inform multiple audiences simultaneously.

Time your disclosure carefully. If the company hasn't acknowledged the breach, give them reasonable warning—typically 7-30 days depending on severity. Coordinate with other researchers who've found the same breach. Unified disclosure carries more weight than scattered reports.

This workflow assumes part-time investigation. Full-time security researchers compress these phases into 48-72 hours. But even quick investigations follow these steps—detection, validation, analysis, reporting. Skip validation and you spread misinformation. Skip analysis and you're just a messenger. Complete all four and you're producing actionable intelligence.

Common Mistakes in OSINT Investigation

Learning from others' errors accelerates your skill development:

Mistake 1: Trusting single sources. Always corroborate claims across multiple channels. Threat actors sometimes post fake breaches for attention or to damage competitors.

Mistake 2: Ignoring legal boundaries. Downloading full databases crosses from research into potential evidence of criminal activity. Sample analysis suffices for validation.

Mistake 3: Poor operational security. Investigating breach forums from your real IP address or work accounts creates risk. Use VPNs and compartmentalized identities.

Mistake 4: Premature disclosure. Publishing unverified breach claims harms your reputation and potentially causes real damage if false. Verify first.

Mistake 5: Neglecting documentation. Without contemporaneous records of your investigation process, findings lose credibility. Screenshot everything, archive pages, and maintain clear notes.

Next Steps: Advancing Your OSINT Skills

This tutorial covered breach investigation fundamentals. To progress further:

Join the OSINT community. Follow researchers like Troy Hunt, Arvid Kahl, and Kevin Beaumont on Twitter. Participate in forums like Reddit's r/OSINT and specialized Discord servers.

Practice on closed cases. Investigate past breaches where full timelines are known. Compare your findings against published post-mortems to identify gaps in your methodology.

Learn complementary skills. Understanding basic web application security, API architecture, and database design helps interpret technical details in breach claims.

Track your own digital footprint. Run OSINT techniques on yourself. Finding your own exposed information teaches you what threat actors see.

The Substack breach demonstrates how much intelligence sits in public view. Companies take months to detect compromises that forum posts reveal in days. Your OSINT skills tip this balance toward defenders.

Want to dive deeper into OSINT techniques? Join thousands of researchers improving their skills daily.

Join our community:
□□ Newsletter https://projectosint.substack.com/
□□ Telegram Group https://t.me/osintprojectgroup

The Substack breach of February 2025 exposed 700,000 user records. Within hours, independent researchers had mapped the attack surface, identified the threat actor's profile, and traced leaked data across multiple forums—all without breaking a single law.

This tutorial walks you through the exact OSINT methods used to investigate real breaches. You'll learn how to gather intelligence ethically, validate leaked data claims, and build a comprehensive breach timeline using only public sources.

Why OSINT Matters for Breach Investigation

Traditional breach response relies on internal logs and forensic tools. But by the time your security team detects an incident, threat actors have often been selling stolen data for weeks.

OSINT flips this model. Public sources—cybercrime forums, paste sites, search engine caches—frequently contain early breach indicators. The Substack case proves this: forum posts appeared before official disclosure, giving researchers a three-month head start.

Three reasons OSINT beats reactive security:

Speed. Threat intelligence from public sources updates in real-time. No waiting for vendors or internal analysis.

Scope. You see the attacker's full campaign—not just what touched your network. Previous victims, attack patterns, tool preferences.

Cost. Every technique here uses free tools. No enterprise licenses required.

The Substack Breach: A Real-World Case Study

On February 3rd, 2026, Substack disclosed a breach affecting 700,000 accounts. The company revealed email addresses, phone numbers, and internal metadata were accessed in October 2025—four months before discovery.

CEO Chris Best's notification included careful language: "credit card numbers, passwords, and financial information were not accessed." But security researcher Arvid Kahl highlighted the real risk: email + phone number combinations enable SIM swapping attacks and sophisticated phishing campaigns.

The breach timeline:

• October 2025: Initial unauthorized access (company claims)
• Late January 2026: Threat actor posts data sample on cybercrime forum
• February 3rd, 2026: Substack confirms breach and notifies users
• February 5th, 2026: Full dataset discussion appears across security communities

This four-month gap between compromise and detection is typical. OSINT practitioners monitoring breach forums spotted the Substack data weeks before official confirmation.

Let's reconstruct this investigation using only public methods.

Step 1: Google Dorking for Initial Indicators

Google Dorking uses advanced search operators to find information companies didn't mean to expose. When investigating potential breaches, these operators uncover vulnerable systems, leaked credentials, and exposed databases.

Finding Exposed Substack Infrastructure

Start with basic reconnaissance to map the attack surface:

```
site:substack.com filetype:pdf confidential
site:substack.com inurl:admin
site:substack.com intext:"index of" "backup"
```

These queries reveal administrative panels, backup directories, or sensitive documents. Nothing here constitutes hacking—you're simply using Google's index more precisely than the average user.

Each operator serves a specific purpose. The site: operator limits results to a single domain. The filetype: operator finds specific document types—PDFs often contain internal presentations or reports never meant for public viewing. The inurl: operator searches within web addresses, where administrators sometimes leave test environments or staging servers accessible.

For the Substack investigation, researchers likely used temporal operators to find cached pages:

```
site:substack.com after:2025-10-01 before:2025-11-01
```

This narrows results to the breach window. Cached pages sometimes preserve evidence removed from live sites. When companies discover exposed information, they remove it quickly. But Google's cache and the Internet Archive preserve snapshots. A page deleted on November 1st might still exist in Google's cache from October 15th—right in the middle of the breach period.

Advanced operators combine for precision targeting:

```
site:substack.com (inurl:admin OR inurl:dashboard) -login
site:substack.com intitle:"index of" +"parent directory"
site:substack.com ext:sql OR ext:db OR ext:log
```

The first query finds administrative interfaces that don't require login. The second locates directory listings exposing file structures. The third searches for database files or logs mistakenly left in web-accessible locations.

Real breach investigations often start with these reconnaissance queries. You're not exploiting vulnerabilities—you're documenting what's already publicly indexed. The distinction matters both legally and ethically.

Searching for Data Leaks

When breach claims surface, validate them before spreading misinformation:

```
"@substack.com" site:pastebin.com
"@substack.com" site:ghostbin.com
intext:"substack" intext:"database" filetype:sql
```

Legitimate breach data rarely appears on paste sites immediately. But threat actors testing stolen credentials often drop small samples to prove authenticity. Finding these samples confirms breach claims before official disclosure.

Ethical boundary: Looking at publicly indexed paste sites is legal. Downloading databases to verify passwords crosses into gray areas. Stay on the research side of this line.

Step 2: Monitoring Breach Forums and Marketplaces

Cybercrime forums are where stolen data gets traded. Monitoring these sources requires understanding their culture and communication patterns.

Where to Look

The Substack data appeared on a known breach marketplace (exact name withheld). Similar platforms include:

- BreachForums successors: The original shut down, but multiple clones operate
- Exploit.in: Russian-language forum with English section
- Nulled.to: Large community with data leak sections
- Telegram channels: Increasingly popular for quick data drops

Access without participation: Most forums allow read-only access or require minimal engagement. Never purchase data, trade information, or contribute to criminal activity. Observation for research purposes falls under OSINT.

Understanding forum culture matters. These communities have hierarchies, trusted members, and

verification processes. New members claiming major breaches without proof get dismissed. Established actors post partial samples first—enough to prove authenticity without giving away the full dataset for free.

Forum structure varies. Some require cryptocurrency payments to access premium sections. Others use reputation systems where members earn trust through contributions. For OSINT purposes, free tiers usually suffice. Major breach announcements appear in public sections to attract buyers.

Registration often requires operational security awareness. Using your real email address or work account is amateurish and potentially dangerous. Researchers create compartmentalized identities—separate email addresses, VPN connections, and browsing environments. This protects both your organization and your personal safety.

The goal isn't deep infiltration. You're monitoring, not participating. Think of it like watching a public street from a window—legal observation, not trespassing.

What to Look For

When a threat actor claims a breach, they post proof—usually a small data sample. The Substack post likely included:

• Record count (700,000 in this case)
• Data types (emails, phone numbers, metadata)
• Breach date claim (October 2025)
• Sample records (partial or full)

Cross-reference these claims with known Substack user patterns. Do email formats match? Are phone numbers in plausible formats? Experienced researchers spot fabricated breaches quickly.

Validating Claims

Before accepting breach claims as legitimate:

1. Check sample data authenticity. Do emails follow company patterns? Real Substack addresses typically look like "newsletter-name@substack.com" or "username@customdomain.com" for paid creators.
2. Search for corroboration. Has anyone else verified these records? Security researchers often share validation (without exposing full records) on Twitter or security forums.
3. Timeline plausibility. Does the claimed breach date align with system downtime, maintenance windows, or public incidents?

The Substack case passed all checks. Researchers verified email patterns, and the October 2025 date aligned with retrospective analysis of company communications.

Step 3: Analyzing Metadata and Attribution

Understanding *who* conducted the breach is as important as validating *what* was stolen. Threat actor attribution relies on pattern analysis across multiple breaches.

Techniques for Attribution

Language analysis. Forum posts contain clues. Grammar mistakes, idioms, and time zone references narrow possible origins. The Substack actor's English proficiency suggests native speaker or experienced operative.

Historical patterns. Search the threat actor's username across platforms:

```
"[username]" site:reddit.com
"[username]" site:twitter.com
"[username]" breach
```

Experienced actors reuse handles. Finding previous breaches credited to the same user reveals their specialization, target preferences, and sophistication level.

Technical indicators. How was data obtained? The metadata column in leaked Substack records suggests API exploitation rather than SQL injection. This detail matters—API breaches indicate different vulnerability classes than database dumps.

Building a Threat Profile

For the Substack breach, OSINT analysis revealed:

- Actor sophistication: Medium-high (maintained access for months undetected)
- Motivation: Financial (data posted for sale, not freely leaked)
- Previous activity: Linked to at least two similar breaches in 2024-2025
- TTPs: Prefers API exploitation over brute force methods

This profile helps other potential targets assess their risk. Companies using similar platforms or API architectures should review access controls immediately.

Building comprehensive threat profiles requires patience and cross-referencing. Start with the basics: username, posting patterns, language preferences, time zones. A threat actor posting consistently between 9 AM and 5 PM Moscow time likely operates from that region or uses it as cover.

Look for technical tells in their posts. Do they make mistakes revealing infrastructure knowledge? When discussing the breach, do they demonstrate insider understanding of specific systems? The Substack actor mentioned "internal metadata" before official disclosure specified what that meant—suggesting either access to company documents or technical knowledge of their database structure.

Previous breach connections matter most. If the same actor sold data from three newsletter platforms in 2024-2025, they're specializing. They've likely developed specific tools or techniques for this sector. Other newsletter platforms become obvious targets. This pattern recognition helps security teams prioritize defensive measures.

Track their business model. Some actors sell data immediately for quick profits. Others hold datasets, waiting for bidding wars. The Substack actor posted samples in late January but the full dataset wasn't widely available for weeks—suggesting selective sales to high-paying buyers first. This behavior indicates experience and business sophistication.

Step 4: Timeline Reconstruction Using Web Archives

The Internet Archive's Wayback Machine and archive.today preserve website snapshots. These archives help reconstruct events when companies delete evidence or revise statements.

Checking Substack's Response Evolution

Compare archived security pages before and after breach disclosure:

```
https://web.archive.org/web/*/substack.com/security
```

Did Substack update security documentation after October 2025? Major changes might indicate awareness of the breach before public disclosure.

Similarly, capture and archive the CEO's breach notification email. Companies sometimes edit official communications after backlash. Having the original version matters for accountability and accurate reporting.

Finding Deleted Evidence

Cached pages and archives sometimes preserve:

- Removed blog posts discussing security incidents
- Deleted social media responses to user complaints
- Earlier versions of privacy policies or data handling practices

Archive everything immediately after a breach becomes public. Within days, relevant pages often disappear or get modified.

Step 5: Correlating Multiple Intelligence Sources

Professional OSINT investigations combine multiple sources to build complete pictures. No single method reveals the full story.

Creating an Investigation Matrix

Track findings across channels:

Source	Information Found	Validation Status	Date Discovered
Breach forum	Initial claim, 700k records	Sample verified	Jan 28, 2026
Google cache	Possible vuln disclosure	Circumstantial	Feb 1, 2026
Twitter	User complaints about spam	Supports timeline	Oct-Nov 2025
Wayback Machine	Security page updates	Confirmed changes	Feb 4, 2026

This matrix helps spot contradictions and gaps. For example, if users complained about increased spam in November 2025, the October breach date gains credibility.

Red Flags and False Positives

Not every data leak is real. Watch for:

- Recycled breaches: Old databases reposted as "new" breaches
- Fabricated samples: AI-generated email addresses passing casual inspection
- Attention seekers: Unverified claims from accounts with no breach history

The Substack breach avoided these red flags. Multiple independent researchers verified the data, samples matched real user patterns, and the threat actor had credible history.

Practical Tools for OSINT Breach Investigation

Beyond Google Dorking, these tools streamline investigation workflows:

Free Intelligence Platforms

Have I Been Pwned (haveibeenpwned.com): Check if specific email addresses appear in known

breaches. The creator, Troy Hunt, often adds newly confirmed breaches within days.

The service works through aggregation. When major breaches occur and data becomes public, Hunt verifies authenticity and adds records to his database. You can check individual email addresses or use the API to monitor multiple accounts. The Notify Me feature sends alerts when your email appears in new breaches—crucial for both personal security and organizational monitoring.

For researchers, HIBP provides breach timelines and affected service lists. If you're investigating a potential breach, checking whether similar services appear in HIBP's database reveals patterns. Three newsletter platforms breached in 2025? That's not coincidence—it's a campaign.

DeHashed (dehashed.com): Searches multiple breach databases simultaneously. Free tier limited but useful for quick checks.

Unlike HIBP's email-focused approach, DeHashed searches across multiple data points: usernames, email addresses, phone numbers, addresses, IP addresses. This broader search capability helps when investigating breaches where email addresses aren't the primary identifier. The Substack breach included phone numbers—DeHashed could correlate those numbers with other breached services, revealing the full scope of affected users' exposure.

BreachDirectory: Community-maintained breach aggregator. Less reliable than HIBP but sometimes faster to index new leaks.

This grassroots database accepts submissions from security researchers. Quality varies, but speed matters in breach investigations. When racing to validate claims, BreachDirectory sometimes surfaces data hours before major services. Cross-reference anything found here with more authoritative sources before relying on it.

Social Media Monitoring

TweetDeck/Twitter Advanced Search: Track breach-related keywords in real-time:

```
substack breach OR leak OR hacked -retweet
```

Remove retweets to see original reports. Security researchers often discuss breaches hours before mainstream media coverage.

Twitter's Advanced Search supports complex queries. Combine keywords with date ranges, language filters, and account specifications. Search tweets from verified security researchers only: from:troyhunt OR from:GossiTheDog substack breach. This filters signal from noise.

Set up saved searches or use TweetDeck columns for continuous monitoring. When a breach claim surfaces, you'll see it in real-time—not hours later when it's trending. Those early hours matter most for investigation and user notification.

Reddit monitoring: Subreddits like r/cybersecurity, r/netsec, and r/privacy frequently discuss breaches early. Search subreddit history for related discussions.

Reddit's community-driven nature means technical discussions happen before polished articles appear. Security professionals share initial analysis, debate authenticity, and post validation methods. The comment sections often contain more valuable intelligence than the original posts.

Use Reddit's search with timestamp filters: timestamp:1704067200..1706745600 searches posts between specific Unix timestamps. This temporal filtering helps reconstruct who knew what when—crucial for timeline analysis.

Archive and Documentation

archive.today: Faster than Wayback Machine for immediate archival. Create snapshots of breach claims, official statements, and related pages.

When investigating breaking breaches, archive everything immediately. Companies edit press releases, threat actors delete posts, forum moderators remove evidence. Archive.today captures pages in seconds—much faster than waiting for the Wayback Machine to crawl.

The service also bypasses some paywalls and anti-scraping measures, though this functionality exists primarily for archival purposes. Each archived page receives a unique URL you can share with other researchers without directing them to potentially malicious sites.

Hunchly (web archive tool): Designed for investigators. Automatically captures browsing history with searchable archives. Paid tool but worth it for serious investigations.

Hunchly runs as a browser extension, silently recording every page you visit during investigations. Later, search your entire browsing history by keyword, date, or domain. This proves invaluable when you remember reading something relevant but can't recall where.

The tool generates timestamped PDF reports of your investigation path—useful for court cases, internal reviews, or simply documenting your methodology. For professional investigators, the $130 annual cost pays for itself in saved time during the first investigation.

Ethical and Legal Boundaries

OSINT operates in a gray zone between public research and potential criminal activity. Understanding these boundaries protects you legally and ethically.

What's Legal

• Viewing publicly accessible websites and forums
• Using search engines to find indexed information
• Reading breach notifications and security advisories
• Analyzing publicly posted data samples (not downloading databases)
• Archiving public web pages for documentation

What's Not Legal

• Accessing systems without authorization (even if credentials are public)
• Downloading or distributing stolen databases
• Using breached credentials to test access
• Participating in data trading or sales
• Doxing individuals using leaked information

The Substack investigation stayed ethical. Researchers verified claims using small samples, documented findings publicly, and notified potentially affected users—never distributing the full database.

Responsible Disclosure

When OSINT uncovers unreported breaches:

1. Verify thoroughly. False claims damage your credibility and cause unnecessary panic.
2. Notify the company privately first. Give them reasonable time to respond (typically 7-30 days).
3. Document everything. Your communications, their responses, and your findings.
4. Disclose responsibly. Share enough detail to inform users without enabling further exploitation.

The information security community respects researchers who follow these guidelines. Companies sometimes reward ethical disclosure through bug bounty programs.

Lessons from the Substack Breach

This case study reveals several broader security patterns worth understanding:

Why Detection Took Four Months

The October to February gap isn't unusual. Sophisticated actors maintain persistent access while exfiltrating data slowly to avoid triggering rate limits or anomaly detection. Substack's systems likely showed nothing obviously wrong until someone noticed the data for sale.

This delayed detection makes OSINT crucial. Breach forums provided earlier warning than internal security tools.

The Email + Phone Number Risk

Substack's reassurance about "no passwords or financial data" missed the real threat. Combining email addresses with phone numbers enables:

- SIM swapping attacks: Convince carriers to port numbers, then bypass 2FA
- Targeted phishing: Messages personalized with real email and SMS channels
- Account takeover chains: Use one compromised service to attack others

Security researcher Arvid Kahl correctly identified this risk. When companies downplay "minor" data exposure, threat actors know better.

Platform-Wide Implications

Substack isn't unique. Similar newsletter platforms, content management systems, and creator economy tools handle identical data types. Any of them could face parallel breaches.

If you use these services, the Substack case offers clear lessons: enable all available security features, use unique passwords, and monitor breach notification services actively.

Building Your OSINT Investigation Workflow

Investigating breaches requires systematic approaches. Here's a starter workflow based on real research patterns:

Phase 1: Initial Detection (Days 1-2)

- Monitor breach forums and Telegram channels daily
- Set up Google Alerts for "[company name] breach OR leak"
- Check HIBP and DeHashed for new breach additions
- Watch security researcher Twitter lists for early discussions

Detection speed determines everything. The gap between breach occurrence and discovery belongs to threat actors. The gap between threat actor disclosure and public awareness belongs to researchers monitoring the right channels.

Build a monitoring routine you can sustain. Check three breach forums every morning with coffee. Scan your Twitter security list during lunch. Review Google Alerts before leaving work. Consistency beats intensity—daily 15-minute checks outperform sporadic deep dives.

Automation helps. RSS feeds from security blogs, Telegram bots monitoring keywords, browser extensions alerting on breach mentions. The OSINT Framework lists dozens of monitoring tools. Pick three that fit your workflow and master them.

Phase 2: Validation (Days 3-5)

• Examine posted samples for authenticity markers
• Cross-reference claimed breach dates with public events
• Search for corroborating reports from multiple sources
• Archive all findings immediately

Validation separates professionals from amateurs. Anyone can screenshot a breach claim. Researchers verify before amplifying. The Substack case included multiple validation layers: sample email formats matched known patterns, metadata fields aligned with Substack's database structure, and timing correlated with user complaints about increased spam.

Create a validation checklist. Does the sample data look real? Do field names match the company's known database structure? Is the claimed breach date plausible given maintenance windows or public incidents? Have other researchers verified independently?

Document your validation process. Screenshots of original claims, notes on verification methods, timestamps of when you confirmed each element. This documentation protects you from accusations of spreading misinformation and provides a paper trail if your findings later become legal evidence.

Phase 3: Deep Analysis (Days 6-10)

• Attempt threat actor attribution through username research
• Build timeline using web archives and cached pages
• Analyze technical details to understand attack vectors
• Document methodology for potential publication

Deep analysis distinguishes basic reporting from intelligence work. Now you're not just confirming a breach happened—you're understanding how, why, and what it means for future security.

Threat actor attribution requires patience. Search their username across platforms going back months or years. Read their posting history. What services do they target? What tools do they mention? Do they sell data quickly or hold it? Understanding their methods and motivations helps predict their next moves.

Timeline reconstruction combines multiple sources. Company status pages might show unusual maintenance during the breach period. User forums might have complaints about system behavior. Web archives preserve deleted security blog posts. Piece together the full story from fragments scattered across the internet.

Phase 4: Reporting (Days 11+)

• Compile findings into structured intelligence report
• Contact affected company if they haven't disclosed publicly
• Share appropriate details with security community
• Monitor company response and update findings

Reporting requires careful judgment. What information helps defenders without enabling attackers? The Substack investigation shared enough detail for users to assess their risk without posting the stolen database or detailed exploitation techniques.

Structure matters. Executive summary for time-constrained readers, detailed methodology for security professionals, actionable recommendations for affected users. Good breach reports inform

multiple audiences simultaneously.

Time your disclosure carefully. If the company hasn't acknowledged the breach, give them reasonable warning—typically 7-30 days depending on severity. Coordinate with other researchers who've found the same breach. Unified disclosure carries more weight than scattered reports.

This workflow assumes part-time investigation. Full-time security researchers compress these phases into 48-72 hours. But even quick investigations follow these steps—detection, validation, analysis, reporting. Skip validation and you spread misinformation. Skip analysis and you're just a messenger. Complete all four and you're producing actionable intelligence.

Common Mistakes in OSINT Investigation

Learning from others' errors accelerates your skill development:

Mistake 1: Trusting single sources. Always corroborate claims across multiple channels. Threat actors sometimes post fake breaches for attention or to damage competitors.

Mistake 2: Ignoring legal boundaries. Downloading full databases crosses from research into potential evidence of criminal activity. Sample analysis suffices for validation.

Mistake 3: Poor operational security. Investigating breach forums from your real IP address or work accounts creates risk. Use VPNs and compartmentalized identities.

Mistake 4: Premature disclosure. Publishing unverified breach claims harms your reputation and potentially causes real damage if false. Verify first.

Mistake 5: Neglecting documentation. Without contemporaneous records of your investigation process, findings lose credibility. Screenshot everything, archive pages, and maintain clear notes.

Next Steps: Advancing Your OSINT Skills

This tutorial covered breach investigation fundamentals. To progress further:

Join the OSINT community. Follow researchers like Troy Hunt, Arvid Kahl, and Kevin Beaumont on Twitter. Participate in forums like Reddit's r/OSINT and specialized Discord servers.

Practice on closed cases. Investigate past breaches where full timelines are known. Compare your findings against published post-mortems to identify gaps in your methodology.

Learn complementary skills. Understanding basic web application security, API architecture, and database design helps interpret technical details in breach claims.

Track your own digital footprint. Run OSINT techniques on yourself. Finding your own exposed information teaches you what threat actors see.

The Substack breach demonstrates how much intelligence sits in public view. Companies take months to detect compromises that forum posts reveal in days. Your OSINT skills tip this balance toward defenders.

Want to dive deeper into OSINT techniques? Join thousands of researchers improving their skills daily.

Join our community:
□□ Newsletterhttps://projectosint.substack.com/
□□ Telegram Grouphttps://t.me/osintprojectgroup