INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-25 19:08 UTC

Underminr: CDN Infrastructure Vulnerability Enables Domain-Fronting-Style C2 Traffic Concealment

SECURITY ANALYSIS | HIGH | CVSS 7.5

SCC Item ID	SCC-STY-2026-0156
Type	Security Analysis
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Shared CDN infrastructure across major providers; estimated 88 million domains affected
Published	2026-05-23
Discovery Source	Gemini

Executive Summary

A newly disclosed vulnerability class called 'Underminr' allows attackers to route malicious command-and-control traffic through shared CDN infrastructure, making it appear legitimate to security tools that inspect only connection metadata. Because the technique abuses CDN architecture rather than exploiting a discrete software flaw, no patch exists, and an estimated 88 million domains fall within the affected scope. This signals a broader shift in attacker tradecraft: adversaries are increasingly hiding inside trusted cloud and delivery infrastructure, eroding the value of perimeter controls and DNS-based filtering.

Technical Analysis

Underminr is an architectural abuse technique, not a code-level exploit. It extends the well-documented domain fronting concept to shared CDN platforms, where multiple customer domains resolve to identical IP address ranges and share TLS termination endpoints. The attack works as follows: an adversary registers or compromises an account with a CDN provider, configures an origin pointing to attacker-controlled infrastructure, and routes C2 traffic through the CDN. From the perspective of a DNS filter or network inspection tool, the outbound connection appears to terminate at the CDN's trusted IP space. The SNI field in the TLS handshake and the DNS resolution both reflect a legitimate CDN domain. The malicious routing occurs after TLS termination, in the CDN-to-origin leg, which most enterprise security controls do not inspect.

The affected surface is substantial. Because major CDN providers consolidate millions of customer domains behind shared IP ranges, a defender cannot simply block the CDN's IP space without disrupting legitimate business traffic. This is the same constraint that made original domain fronting effective against censorship tools

before providers like Amazon and Google moved to enforce SNI-Host header matching. Underminr suggests that enforcement is inconsistent or incomplete across providers.

MITRE ATT&CK maps this technique directly to T1090.004 (Domain Fronting), T1071.001 (Web Protocols for C2), and T1102 (Web Service for C2). CWE-441 (Unintended Proxy or Intermediary) and CWE-923 (Improper Restriction of Communication Channel to Intended Endpoints) characterize the architectural flaw class. No CVE has been assigned, consistent with how security researchers and the CVE program treat architectural abuse issues that do not represent a discrete implementation defect.

The defensive gap this exploits is significant: most DNS filtering and network perimeter tools classify traffic by the SNI hostname or DNS query, not by the actual origin the CDN routes to after termination. Zero-trust architectures that rely on DNS categories or IP reputation for CDN-destined traffic are similarly blind to post-termination routing. Security teams that have invested in TLS inspection may have partial visibility, but CDN-mediated traffic is frequently excluded from inspection policies due to certificate pinning, performance concerns, or explicit allow-listing of CDN IP ranges.

No specific threat actors have been attributed to active exploitation of Underminr at the time of publication. However, the technique aligns with observed tradecraft from APT groups and ransomware operators who have previously abused Cloudflare, Fastly, and other CDN platforms to host or proxy malicious payloads. The absence of a CVE and the architectural nature of the issue mean that remediation depends on CDN providers enforcing stricter origin validation, not on enterprise defenders applying a patch.

Action Checklist

1. Step 1: Assess CDN exposure, inventory which CDN providers (Cloudflare, Akamai, Fastly, AWS CloudFront, and others) your organization relies on, and identify whether your DNS filtering or network controls allow broad CDN IP ranges by default (CIS Controls 1.1, Establish and Maintain Detailed Enterprise Asset Inventory)
2. Step 2: Review perimeter inspection policies, audit whether TLS inspection is applied to CDN-destined traffic or whether CDN IP ranges are blanket allow-listed in your firewall or proxy rules; tighten where operationally feasible (NIST SC-7, Boundary Protection; CIS Controls 4.4, Implement and Manage a Firewall on Servers)
3. Step 3: Evaluate DNS filtering coverage gaps, DNS-based controls see only the CDN hostname, not the true origin; assess whether your DNS security tool provides any post-termination origin visibility or behavioral analysis, and document the gap if it does not (NIST SI-4 via D3-PBWSAM [Proxy-based Web Server Access Mediation], Proxy-based Web Server Access Mediation)
4. Step 4: Update threat model and detection rules, add T1090.004 (Domain Fronting) and T1102 (Web Service for C2) to your threat register; tune SIEM or NDR rules to flag unusual beacon patterns to CDN-associated IP ranges, particularly low-and-slow traffic with consistent intervals (NIST AU-6, Audit Record Review, Analysis, and Reporting)
5. Step 5: Engage CDN vendor on origin enforcement, contact your CDN provider(s) to understand whether they enforce SNI-Host header consistency and what controls exist to prevent unauthorized origin routing through their shared infrastructure; document the response for risk register purposes
6. Step 6: Communicate risk to leadership, brief security leadership on the limitation this creates for perimeter-based controls; frame the risk in terms of C2 detection gap, not patch status, since no patch is available (NIST IR-6, Incident Reporting)

7. Step 7: Monitor for researcher and provider disclosures, track SecurityWeek, CISA advisories, and individual CDN provider security bulletins for follow-up guidance, enforcement changes, or confirmed exploitation tied to Underminr

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate to immediate priority and invoke the IR plan (NIST IR-4) if: Zeek ssl.log or proxy logs reveal SNI-Host header mismatches to CDN IP ranges from internal endpoints not expected to communicate with those CDNs, Rita identifies statistically significant beacon regularity in CDN-destined connections, or a CDN provider or CISA publishes confirmed in-the-wild exploitation of Underminr-style techniques affecting your confirmed CDN providers; additionally, if your organization is subject to HIPAA, PCI-DSS, or SOX and the C2 detection gap cannot be mitigated within 30 days, escalate to GRC for regulatory disclosure risk assessment.
Recovery Notes	Because no patch exists for Underminr and remediation is architectural rather than procedural, recovery is defined as achieving a defensible detection posture rather than eliminating the vulnerability: verify that TLS inspection covers at least ingress/egress to your highest-risk CDN providers, that beacon detection rules (T1090.004, T1102) are operational and generating alerts in your monitoring platform, and that CDN vendor SNI-Host enforcement status is documented. Maintain enhanced monitoring of CDN-destined traffic — specifically Zeek conn.log beacon scoring via Rita and ssl.log SNI-Host mismatch alerts — for a minimum of 90 days given the architectural nature of the exposure and the lack of a definitive remediation signal. Reassess the risk register entry and detection rule effectiveness at 30, 60, and 90 days, updating based on any new researcher disclosures, CDN provider enforcement changes, or confirmed exploitation IOCs tied to Underminr.
Forensic Artifacts	Zeek ssl.log: SNI field (server_name column) vs. destination IP — in Underminr-style abuse, SNI will resolve to a legitimate CDN hostname (e.g., *.cloudfront.net, *.fastly.net) while the actual C2 routing occurs post-CDN-termination; preserve 30-day rolling logs for retrospective correlation against any published Underminr IOCs | Zeek http.log / proxy access logs: HTTP Host header values in requests destined for CDN IP ranges — a Host header referencing an unrecognized or attacker-controlled origin domain while the connection terminates at a CDN IP is the defining forensic indicator of domain-fronting-style C2 concealment as described in Underminr | Zeek conn.log processed through Rita: connection duration, bytes transferred, and inter-connection interval distributions for CDN-destined flows — Underminr C2 beacons will exhibit statistically anomalous regularity (low jitter, consistent packet sizes) compared to legitimate CDN content delivery traffic, which is characteristically bursty and variable | DNS resolver query logs (Windows DNS debug log at C:\Windows\System32\dns\dns.log or BIND query log at /var/log/named/): query frequency and TTL response values for CDN apex domains — C2 implants using Underminr-style techniques may query CDN hostnames at regular intervals inconsistent with normal browser or application behavior, and abnormally short TTLs may indicate CDN IP rotation to evade IP-based blocklists | Firewall and proxy allow-list configuration exports: timestamped snapshots of rules permitting CDN CIDR blocks without TLS inspection — these are not attack artifacts but are the primary control-gap evidence that would explain why an active Underminr C2 channel was not detected, and are essential for post-incident review, regulatory inquiry, and insurance documentation

Per-Action IR Details

Step 1: Assess CDN exposure — inventory which CDN providers (Cloudflare, Akamai, Fastly, AWS CloudFront, and others) your organization relies on, and identify whether your DNS filtering or network controls allow broad CDN IP ranges by default (CIS 1.1 — Establish and Maintain Detailed Enterprise Asset Inventory)

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Establishing IR capability and asset visibility baselines before an incident occurs

Controls: CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory), NIST RA-3 (Risk Assessment), NIST CM-8 (System Component Inventory)

Compensating: Run 'dig +short ' and cross-reference returned IPs against published CDN CIDR ranges (Cloudflare: https://www.cloudflare.com/ips/, AWS CloudFront: aws ip-ranges.json, Fastly: https://api.fastly.com/public-ip-list). Use a bash one-liner: 'for domain in $(cat domains.txt); do dig +short $domain; done | sort -u' to enumerate resolved CDN IPs. Cross-reference your firewall allow-list exports (e.g., 'iptables -L -n' or Windows Firewall 'netsh advfirewall firewall show rule name=all') to identify which CDN CIDR blocks are blanket-permitted without deep inspection.

Evidence: Before inventorying, capture a snapshot of current DNS resolution records for all outbound-communicating assets: export DNS query logs from your resolver (e.g., Windows DNS debug log at C:\Windows\System32\dns\dns.log, or Pi-hole/BIND query logs at /var/log/named/queries.log) to establish a pre-assessment baseline. Also export current firewall allow-list rules referencing CDN IP ranges so you can document the pre-change state for the risk register.

Step 2: Review perimeter inspection policies — audit whether TLS inspection is applied to CDN-destined traffic or whether CDN IP ranges are blanket allow-listed in your firewall or proxy rules; tighten where operationally feasible (NIST SC-7 — Boundary Protection; CIS 4.4 — Implement and Manage a Firewall on Servers)

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Configuring network defenses and inspection policies as a pre-incident hardening measure

Controls: NIST SC-7 (Boundary Protection), NIST SC-8 (Transmission Confidentiality and Integrity), NIST SI-4 (System Monitoring), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: For teams without a commercial SSL inspection proxy, deploy mitmproxy (free, open-source) in transparent mode on the network egress path to selectively intercept and log TLS traffic to CDN IP ranges. Use Wireshark with a capture filter such as 'host 104.16.0.0/12' (Cloudflare range) to observe TLS SNI fields in ClientHello packets — SNI will reveal the CDN hostname but NOT the true origin in Underminr-style abuse. Document any traffic where the SNI hostname does not match an expected internal or approved SaaS domain as a gap requiring escalation.

Evidence: Capture TLS ClientHello packets to CDN-associated IP ranges before making any rule changes: use 'tcpdump -i eth0 -w cdn_tls_baseline.pcap net 104.16.0.0/12 or net 151.101.0.0/16 or net 13.32.0.0/15' (adjust CIDRs per your CDN inventory from Step 1) for a 24-hour baseline. Preserve existing firewall rule exports and proxy exception lists as pre-change evidence. These captures will reveal whether any current traffic already exhibits Underminr-consistent patterns (CDN IP destination, CDN SNI, but anomalous beacon timing or payload size distributions).

Step 3: Evaluate DNS filtering coverage gaps — DNS-based controls see only the CDN hostname, not the true origin; assess whether your DNS security tool provides any post-termination origin visibility or behavioral analysis, and document the gap if it does not (NIST SI-4 via D3-PBWSAM — Proxy-based Web Server Access Mediation)

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Identifying and documenting detection capability gaps that would prevent identification of Underminr-style C2 concealment

Controls: NIST SI-4 (System Monitoring), NIST AU-2 (Event Logging), NIST CA-7 (Continuous Monitoring), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Deploy PassiveDNS logging via Zeek (free) on your network tap to record all DNS query/response pairs including TTLs — Underminr C2 channels may exhibit abnormally short TTLs as CDN operators rotate IPs, or unusually consistent query intervals characteristic of C2 beaconing. Use the Zeek dns.log to run: 'cat dns.log | zeek-cut query answers TTL | awk '$3 < 60' to flag sub-60-second TTL responses from CDN hostnames. Document formally in your risk register that DNS-layer controls cannot distinguish between a legitimate SaaS request and an Underminr C2 channel terminating at the same CDN hostname.

Evidence: Export your DNS security tool's block/allow logs and exception lists before the assessment — specifically any CDN hostname categories that are blanket-allowed (e.g., 'content delivery' or 'cloud infrastructure' categories in Cisco Umbrella, Infoblox, or Pi-hole). Preserve Zeek or DNS server query logs (Windows DNS at HKLM\SYSTEM\CurrentControlSet\Services\DNS\Parameters for debug logging config; Linux BIND at /var/log/named/) covering the prior 30 days to establish a behavioral baseline of CDN-destined DNS query patterns.

Step 4: Update threat model and detection rules — add T1090.004 (Domain Fronting) and T1102 (Web Service for C2) to your threat register; tune SIEM or NDR rules to flag unusual beacon patterns to CDN-associated IP ranges, particularly low-and-slow traffic with consistent intervals (NIST AU-6 — Audit Record Review, Analysis, and Reporting)

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: Implementing detection logic and correlating indicators specific to CDN-abusing C2 techniques

Controls: NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-2 (Event Logging), NIST SI-4 (System Monitoring), NIST RA-3 (Risk Assessment), CIS 8.2 (Collect Audit Logs)

Compensating: Without a SIEM, deploy the Sigma rule for T1090.004 (available in SigmaHQ repository under rules/network/net_connection_domain_fronting.yml) converted to your available tooling via sigma-cli. For manual detection, use Zeek's http.log and ssl.log to identify mismatches between the TLS SNI field and the HTTP Host header — a defining artifact of domain fronting and Underminr-style abuse: 'cat ssl.log | zeek-cut server_name | sort | uniq -c | sort -rn' cross-referenced with 'cat http.log | zeek-cut host | sort | uniq -c | sort -rn'. Flag any session where SNI resolves to a CDN apex domain (e.g., cloudfront.net, fastly.net) but HTTP Host header references an unexpected or unrecognized origin. Use Rita (Real Intelligence Threat Analytics, free) to automatically score beacon regularity in Zeek conn.log output.

Evidence: Before deploying new detection rules, preserve a 30-day baseline of Zeek ssl.log and conn.log (or equivalent proxy logs) showing normal CDN connection patterns — connection duration distributions, bytes-transferred distributions, and inter-connection interval distributions — for all CDN-destined traffic. This baseline is essential to tune beacon detection thresholds and will serve as comparative evidence if a retrospective hunt identifies pre-existing Underminr C2 activity that predates the detection rule deployment.

Step 5: Engage CDN vendor on origin enforcement — contact your CDN provider(s) to understand whether they enforce SNI-Host header consistency and what controls exist to prevent unauthorized origin routing through their shared infrastructure; document the response for risk register purposes

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Establishing third-party coordination mechanisms and documenting vendor security posture as part of IR readiness

Controls: NIST IR-7 (Incident Response Assistance), NIST SA-9 (External System Services), NIST SC-7 (Boundary Protection), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Without a formal vendor management program, use each CDN provider's public security documentation and bug bounty/security contact pages to request written clarification on SNI-Host enforcement: Cloudflare Security (security@cloudflare.com), AWS Security (aws-security@amazon.com), Fastly Security (security@fastly.com). Log the inquiry date, response date, and content in a simple risk register spreadsheet. As an independent verification step, use curl with explicit Host header overrides against your own CDN-hosted domains to test whether the provider rejects SNI-Host mismatches: 'curl -v --resolve legitimate-cdn-domain.com:443:

https://legitimate-cdn-domain.com -H "Host: attacker-controlled-origin.com"' and document the response behavior.

Evidence: Before vendor engagement, document your current CDN configuration settings (origin pull rules, Host header forwarding policies, CNAME configurations) by exporting CDN dashboard configurations or capturing API responses (e.g., AWS CloudFront distribution config via 'aws cloudfront get-distribution-config --id ', Cloudflare zone settings via API). This snapshot establishes your pre-engagement configuration state and will be necessary if a future incident requires demonstrating due diligence in vendor coordination.

Step 6: Communicate risk to leadership — brief security leadership on the limitation this creates for perimeter-based controls; frame the risk in terms of C2 detection gap, not patch status, since no patch is available (NIST IR-6 — Incident Reporting)

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Communicating systemic risk findings and capability gaps to organizational leadership to drive policy and investment decisions

Controls: NIST IR-6 (Incident Reporting), NIST IR-8 (Incident Response Plan), NIST RA-3 (Risk Assessment), NIST PM-9 (Risk Management Strategy), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Produce a one-page risk summary using NIST SP 800-30 risk framing: threat source (adversaries adopting CDN-abusing C2 tradecraft), threat event (C2 traffic routed through shared CDN infrastructure appearing as legitimate HTTPS to Cloudflare/Akamai/Fastly/CloudFront), vulnerability (perimeter controls inspecting only connection metadata, not post-CDN-termination routing), and likelihood/impact ratings. Explicitly quantify the detection gap: 'Our DNS filtering and firewall rules allow all traffic to [X CDN provider] IP ranges without TLS inspection, which means an active Underminr-style C2 channel would be invisible to current controls.' This is more actionable than a generic 'patch unavailable' statement.

Evidence: Compile the outputs from Steps 1–5 as supporting evidence for the leadership brief: the CDN inventory (Step 1), the TLS inspection gap documentation (Step 2), the DNS filtering gap documentation (Step 3), the updated threat register entries for T1090.004 and T1102 (Step 4), and the CDN vendor response log (Step 5). These collectively constitute the evidentiary basis for the risk statement and should be preserved as supporting documentation in the risk register.

Step 7: Monitor for researcher and provider disclosures — track SecurityWeek, CISA advisories, and individual CDN provider security bulletins for follow-up guidance, enforcement changes, or confirmed exploitation tied to Underminr

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Maintaining situational awareness and integrating emerging threat intelligence to update detection and response posture

Controls: NIST SI-5 (Security Alerts, Advisories, and Directives), NIST IR-5 (Incident Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Subscribe to CISA Known Exploited Vulnerabilities (KEV) catalog RSS feed and CISA Cybersecurity Advisories RSS (https://www.cisa.gov/cybersecurity-advisories) for any Underminr-related exploitation confirmations. Configure free RSS/atom monitoring via RSS aggregators (e.g., Feedly free tier) for SecurityWeek CDN-related search terms. For each CDN provider, subscribe directly to their security bulletins: Cloudflare Blog security category, AWS Security Bulletins (https://aws.amazon.com/security/security-bulletins/), Fastly Security Advisories. Create a shared tracking document logging disclosure date, source, content summary, and required action — review weekly with the IR team lead.

Evidence: Maintain a running threat intelligence log that cross-references each new Underminr-related disclosure against your CDN inventory from Step 1 and your detection gap documentation from Steps 2–3. When a CDN provider announces SNI-Host enforcement changes or when confirmed exploitation IOCs are published (specific C2 domain patterns, CDN provider abused, beacon timing signatures, HTTP Host header values seen in confirmed attacks), immediately correlate against your preserved Zeek ssl.log and conn.log baselines from Steps 2–4 to determine if pre-existing traffic in your environment matches the published IOCs.

Detection Guidance

Because Underminr abuses legitimate CDN routing rather than injecting malicious code, traditional signature-based detection is ineffective. Detection must focus on behavioral anomalies in traffic destined for CDN IP ranges.

Log sources to review: proxy and firewall egress logs, DNS query logs, and any available NetFlow or NDR telemetry covering CDN-bound traffic.

Behavioral patterns to hunt for:

- Beaconing to CDN IP ranges: regular, low-variance outbound connections to CDN-associated IP space at consistent intervals (e.g., every 60 or 300 seconds), particularly from hosts with no legitimate reason to poll CDN-hosted content continuously (maps to T1071.001 and T1102 detection logic)

- SNI/Host header mismatch: if TLS inspection is in scope, flag connections where the SNI value references a CDN domain but the HTTP Host header references an unexpected or unrecognized domain (core indicator of domain fronting mechanics)

- Unusual data volumes on CDN connections: outbound data volume disproportionate to the type of CDN-hosted resource implied by the SNI domain

- CDN connections from non-browser processes: proxy logs showing CDN-bound HTTPS traffic originating from processes that would not normally make such requests (scheduled tasks, services, scripting engines)

- Low-reputation or newly registered CDN customer domains: DNS queries resolving to CDN IP ranges for domains registered within the past 30-90 days with no established business relationship

SIEM/NDR tuning recommendations:

- Build a baseline of legitimate CDN destinations and volumes per host; alert on deviations exceeding two standard deviations from that baseline

- Cross-reference CDN-bound connections against your asset inventory (NIST AU-2, Event Logging; CIS Controls 8.2, Collect Audit Logs) to identify unexpected source hosts

- Apply D3-PBWSAM (Proxy-based Web Server Access Mediation) logic by routing all CDN-bound traffic through an inspecting proxy rather than allowing direct egress

Policy gap audit: verify that your zero-trust policy engine does not implicitly trust traffic solely because it resolves to a known CDN IP range; trust decisions should incorporate user identity, device posture, and destination context, not IP reputation alone (NIST SC-7, Boundary Protection).

Indicators of Compromise

Type	Value	Context	Confidence
TOOL	`Pending — refer to SecurityWeek Underminr coverage and Rescana active exploitation alert for published indicators`	Researcher-published indicators associated with Underminr CDN abuse technique, including any C2 domains or origin endpoints identified during disclosure research; actual values not available in provided source material	LOW

Framework Mappings

MITRE-ATTACK

- **T1090.004** — Domain Fronting
- **T1071.001** — Web Protocols
- **T1102** — Web Service

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities

CIS-V8

- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks
- **8.2** — Collect Audit Logs

NIST-800-53R5

- **AT-2** — Literacy Training and Awareness
- **SC-13** — Cryptographic Protection
- **SI-4** — System Monitoring

HIPAA-SECURITY

- **164.312(e)(1)** — Transmission Security

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1090.004	Domain Fronting	Command-And-Control
T1071.001	Web Protocols	Command-And-Control
T1102	Web Service	Command-And-Control

Sources

Source	URL	Tier
gemini	https://www.securityweek.com/underminr-vulnerability-lets-attackers...	T3
What is a Content Delivery Network (CDN)? - Akamai	https://www.akamai.com/glossary/what-is-a-cdn	T3

Source	URL	Tier
Top CDN Security Risks to Consider - Fastly	https://www.fastly.com/learning/cdn/top-cdn-security-risks-to-consider	T3
Underminr Vulnerability in Major CDN Providers Enables Attackers ...	https://www.rescana.com/post/active-exploitation-alert-underminr-vu...	T3
CDNs' Dark Side: Security Problems in CDN-to-Origin Connections	https://dl.acm.org/doi/10.1145/3499428	T3

Generated 2026-05-25 19:08 UTC by TJS Security Command Center