Web Security — S03E05: Server & Proxy Parser Attacks

Modern SQLi, Cloud SSRF, HTTP Request Smuggling, Cache Poisoning & Deserialization



Gemini Notebook

THE EDGE

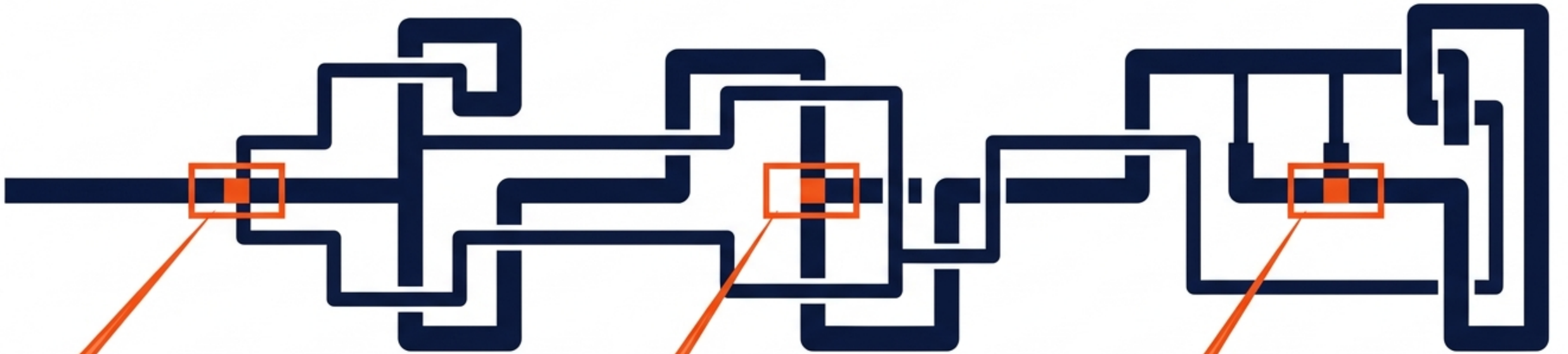
LOAD BALANCERS & CDNS

THE NETWORK

VPC & PROXIES

THE CORE

APP SERVERS & DBS



CALLOUT 1: THE EDGE
HTTP Request Smuggling & Cache Poisoning (Framing/Keying Discrepancies).

CALLOUT 2: THE NETWORK
Cloud SSRF & DNS Rebinding (URI/Resolution Discrepancies).

CALLOUT 3: THE CORE
SQLi & Insecure Deserialization (AST/Lifecycle Discrepancies).

CORE INSIGHT: The modern perimeter isn't a firewall; it's the parser. Attacks thrive in the discrepancies between how reverse proxies, network dials, and backend ORMs interpret the exact same byte stream.

Gemini Notebook

The Illusion of Safety

```
SELECT * FROM users
WHERE name = ?
```

The Hidden Sinks

1. Dynamic `ORDER BY` clauses bypassing parameterization.

2. Hibernate Query Language (HQL) injection:

```
session.createQuery("from Inventory
where productID='"+userInput+"'")
```

3. Raw `JSONB` query builder fragments.

Parameterized queries protect data values, but they **do not** protect **structural query elements** like table identifiers, sort directions, or ORM-specific AST compilations.

Gemini Notebook

Step 1 (Time T)
Attacker submits a malicious **payload** (`' OR SLEEP(10)--`) safely stored in a non-vulnerable `INSERT` statement.

Step 2 (Time T+1)
A background worker retrieves the poisoned data and concatenates it into a vulnerable `UPDATE` trigger.

Step 3 (Execution)
The payload **detonates** out-of-band.

Advanced Extraction Methods



Blind Boolean:
Injecting `AND 1=1` vs `AND 1=2` and observing HTTP response size/status differences.



Time-Based:
Injecting `SLEEP(10)` or `pg_sleep()` to infer truth based on socket delay.



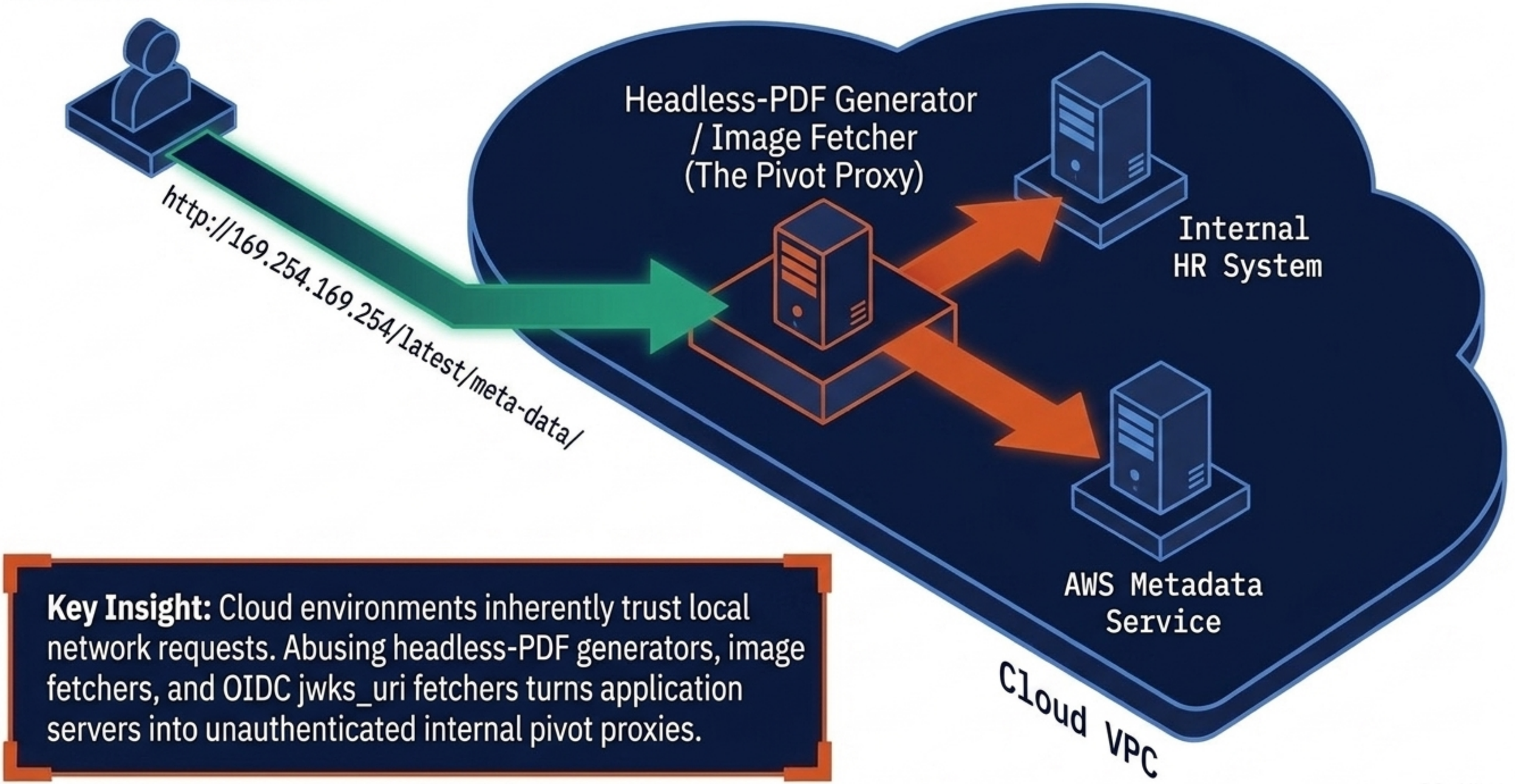
Out-of-Band (OAST):
Triggering a DNS lookup to an attacker-controlled domain containing the exfiltrated data.

Gemini Notebook

SECURE QUERY PATTERNS: A COMPARISON MATRIX

Unsafe Database Patterns	Secure Equivalents
String Concatenation `"SELECT * FROM " + tableName`	**Strict Allow-lists for Identifiers** Map requested string to fixed, app-owned `table_name_a`.
Manual Escaping / Regex Filters Fails against edge cases like `NO_BACKSLASH_ESCAPES`.	**Wire-Protocol Parameter Binding** PreparedStatement separating SQL structure from data values.
HQL/JPQL Injection `session.createQuery(..." + input)`	**ORM Named Parameters** `query.setParameter("id", input)`

Gemini Notebook



Key Insight: Cloud environments inherently trust local network requests. Abusing headless-PDF generators, image fetchers, and OIDC jwks_uri fetchers turns application servers into unauthenticated internal pivot proxies.

Gemini Notebook

```
GET http://169.254.169.254/latest/meta-data/iam/security-credentials/production-role

{
   "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
   "SecretAccessKey": "wJaIrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
   "Token": "IQoJb3JpZ2luX2VjEBAaCXVzLWVhc3QtMSJHMEUCI..."
}
```

The **169.254.169.254** IPv4 Link-Local address is a deadly target in AWS, GCP, and Azure because it exposes instance metadata and IAM tokens directly to the local machine without authentication.

Gemini Notebook

Why String Blocklists Fail (Parser Differentials)

Target String: `http://1.2.3.4\@169.254.169.254`

WHATWG URL Standard

- Treats `\` under a special scheme as a path separator.
- Resulting Host: 1.2.3.4
- **Outcome:** Passes the blocklist filter.

RFC 3986 (e.g., Python urllib.parse)

- No backslash allowed; derives host via netloc.rpartition('@').
- Resulting Host: 169.254.169.254
- **Outcome:** Executes the SSRF.

String-Matching Filter Bypasses (Equivalent IPs)

```
IPv6-mapped IPv4: ::ffff:169.254.169.254
Decimal: 2852039166
Octal: 0251.0376.0251.0376
```

Gemini Notebook

The DNS Rebinding Race Condition (TOCTOU)

1

Pre-flight (Time-of-Check)

- Application resolves malicious.com (TTL=0).
- DNS returns 1.2.3.4 (Safe external IP).
- Passes the security filter.

2

Socket Connect (Time-of-Use)

HTTP client initiates the connection, forcing a re-resolution due to TTL=0.

3

Execution

- Malicious DNS re-resolves to 169.254.169.254.
- Exploit succeeds against internal infrastructure.

Core Insight: DNS Rebinding subverts application-layer blocklists by swapping the IP address between validation and connection.

Gemini Notebook

IMDSv1 vs. IMDSv2: The Evolution of Instance Metadata Security

IMDSv1 (Vulnerable)

- A single `GET` request.
- Easily executed via standard SSRF.

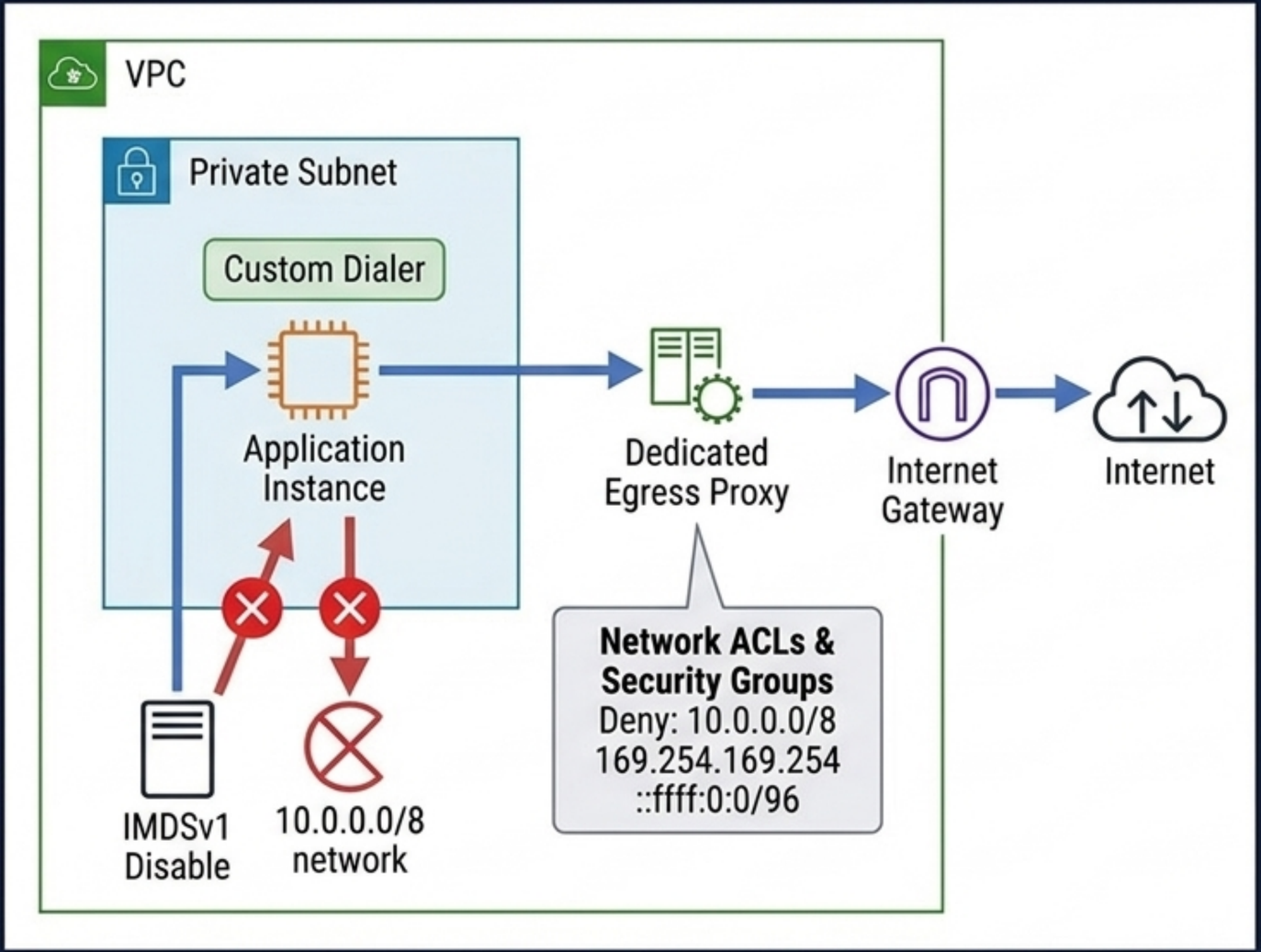
IMDSv2 (Secure)

- A two-step flow.
- Step 1: `PUT /latest/api/token` with header `X-aws-ec2-metadata-token-ttl-seconds: 21600`
- Step 2: Use token in subsequent `GET`.

The Hop Limit Killswitch: IMDSv2 strictly sets the IP TTL (hop limit) to 1. If the request is proxied through a container or a misconfigured network bridge, the packet dies before reaching the metadata service, neutralizing lateral SSRF.

Gemini Notebook

SSRF Defense-in-Depth Architecture



1. Custom Dialer Enforcement

Bind the HTTP client connection to the exact IP address validated during the initial DNS resolution, preserving the original hostname for the Host header and SNI to prevent DNS Rebinding.

2. Dedicated Egress Proxies

Route all outbound webhook/fetch traffic through an isolated egress proxy that denies 10.0.0.0/8, 169.254.169.254, and ::ffff:0:0/96 at the network layer.

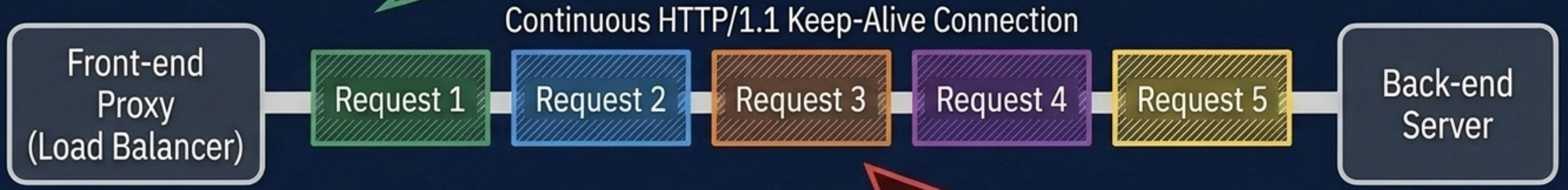
3. IMDSv2 Migration

Permanently disable IMDSv1 on all cloud compute instances.

Gemini Notebook

HTTP/1.1 Keep-Alive Connection and Request Smuggling Risks

Front-end proxies and back-end servers must perfectly agree on where one request ends and the next begins.



The Threat: If the parser at the edge calculates a different request length than the parser at the core, the attacker's payload is orphaned in the TCP buffer, hijacking the next unsuspecting user's request mid-flight.

Gemini Notebook

CL.TE Attack Breakdown

```
POST / HTTP/1.1
Host: vulnerable.com
Content-Length: 13
Transfer-Encoding: chunked\r\n
\r\n
\r\n
0\r\n
\r\n
SMUGGLED
```

Front-end
(e.g., Load Balancer)

Back-end
(e.g., Server)

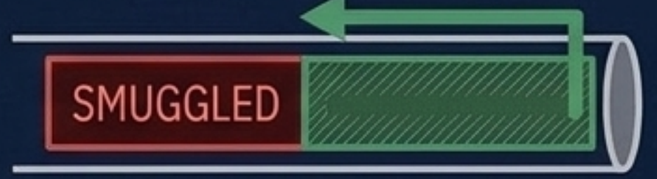
1. Front-end reads Content-Length (forwards the entire block).

```
POST / HTTP/1.1
Host: vulnerable.com
Content-Length: 13
Transfer-Encoding: chunked\r\n
\r\n
0\r\n
\r\n
SMUGGLED
```

2. Back-end reads Transfer-Encoding: chunked, sees the 0 chunk, and terminates parsing early.

```
0\r\n
```

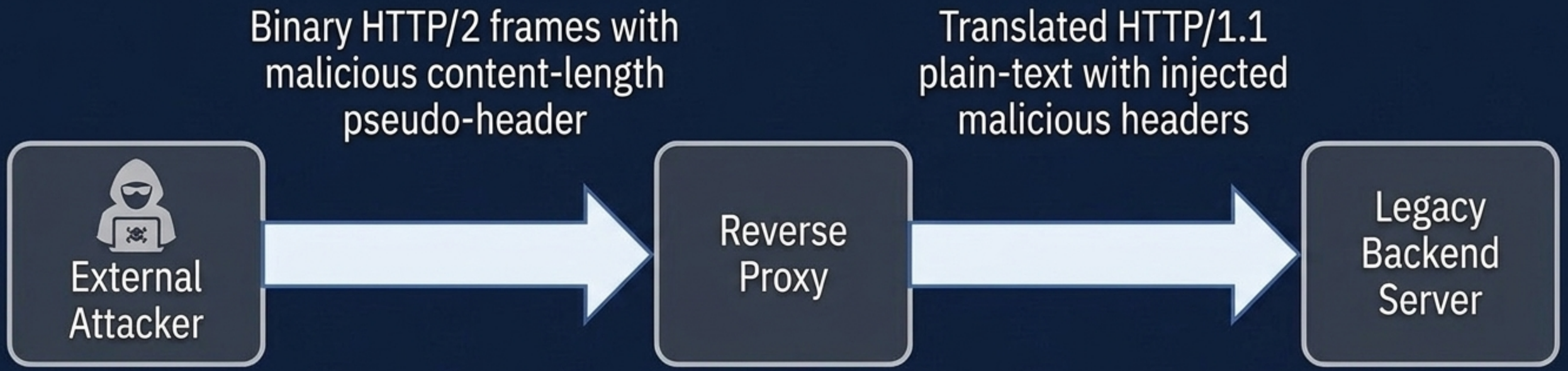
3. The string **SMUGGLED** is left dead in the socket buffer, attaching to the next user's request.



Obfuscated headers (e.g., Transfer-Encoding: xchunked) force parsers into fallback behaviors, creating deadly framing disagreements.

Gemini Notebook

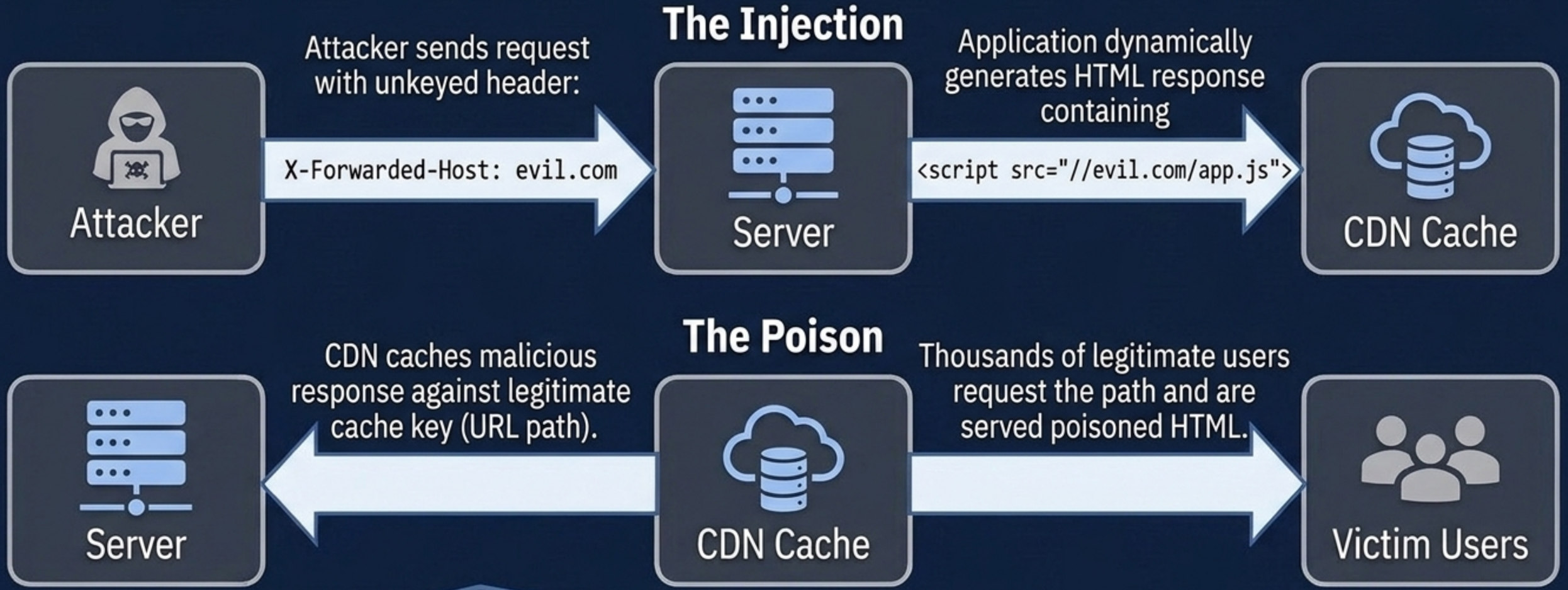
HTTP/2 Downgrade Attacks (H2.CL / H2.TE)



HTTP/2 is binary and immune to smuggling end-to-end. However, if the reverse proxy translates H2 to HTTP/1.1 for legacy backends, it creates framing ambiguities, blindly writing malicious headers into the new stream and unleashing H2-powered desync attacks.

Gemini Notebook

Unkeyed Inputs & Poisoned Reflections



Key Concept: The cache key (URL + query) does not include the malicious header, but the backend application uses it to build the response. The cache blindly trusts the output.

Gemini Notebook

Path-Confusion Web Cache Deception

`https://site.com/account/settings/avatar.css`

Edge CDN Parser

Sees .css extension.

Aggressively caches the response as a static public asset.

Backend Framework Parser

Ignores the .css extension.

Dynamically resolves the route to /account/settings

Returns the user's private JSON PII.

The Result: The victim's private profile data is stored in the public CDN edge node, accessible to anyone requesting that specific .css URL.

Gemini Notebook

Cache Hardening Rules

[FAIL] Caching Authenticated Responses
Never cache responses containing session cookies or Authorization headers.

[PASS] Strip Unexpected Edge Headers
Load balancers must proactively strip or normalize unkeyed routing headers (e.g., X-Forwarded-Host, X-Original-URL) before forwarding.

[PASS] Strict Routing Alignment
CDN caching regex rules must perfectly mirror the backend framework's routing logic. If the backend drops extensions, the CDN must not cache based on extensions.

[PASS] Param Miner Testing
Automate the discovery of unkeyed inputs that mutate backend responses.

Gemini Notebook

The Danger of Native Serializers (Magic Methods)

Java

`ObjectInputStream (readObject())`

Python

`pickle (pickle.loads())`

PHP

`unserialize()`

.NET

`BinaryFormatter`



Core Insight: Native serializers do not just restore state (data); they **restore behavior**. Magic lifecycle methods execute automatically on untrusted bytes during the parsing phase—turning data ingestion directly into pre-auth Remote Code Execution (RCE) before validation even occurs.

Gemini Notebook

XXE & XML Parser Subversion

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [
  <!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
<stockCheck><productId>&xxe;</productId></stockCheck>
```

The trigger that forces the XML parser to fetch an external URI.

The **target sink** (can also be an internal `http://` URL to pivot into SSRF).

The **entity reflection** that leaks the fetched data back to the attacker in the application's response.

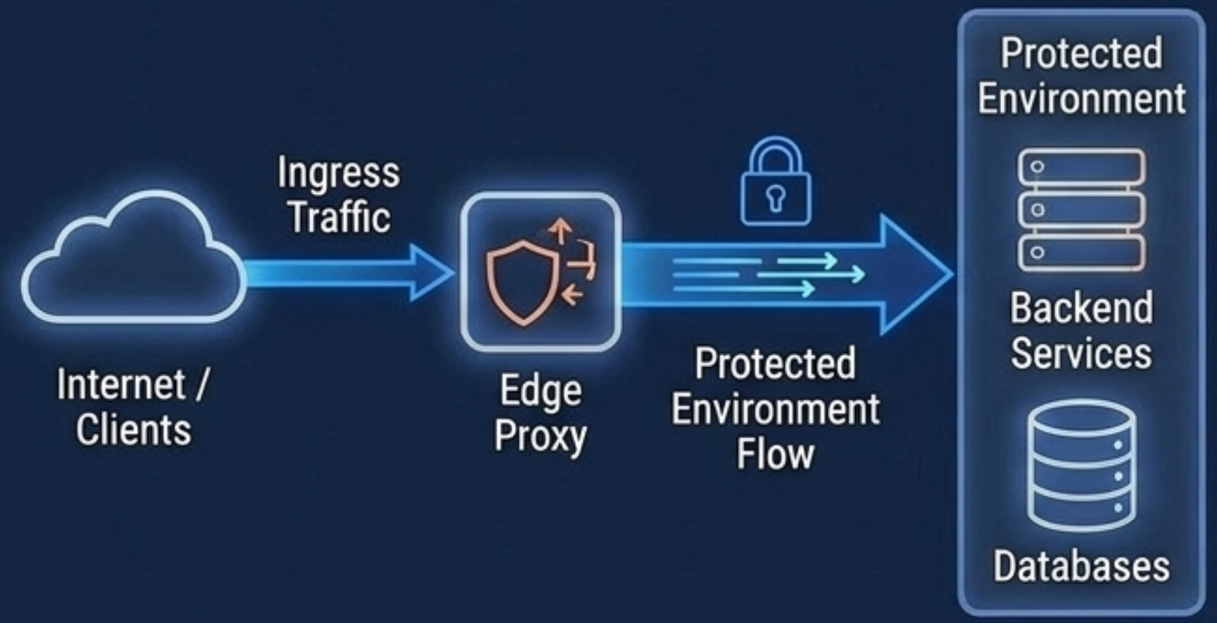
Gemini Notebook

Diagnostic Matrix: Serializers vs. Data Formats

Unsafe / Native Serializers	Safe / Data-Only Formats
Examples: Pickle, YAML Unsafe Loader, ObjectInputStream. **Trait**: Couples code with data; instantiates arbitrary classes; **vulnera**ble to gadget chains.	**Examples:** JSON, Protobuf, SafeConstructor YAML. **Trait:** Pure data transfer object (DTO) patterns. Separates logic from data. **Requires explicit** instantiation by the application, eliminating arbitrary gadget execution.

Gemini Notebook

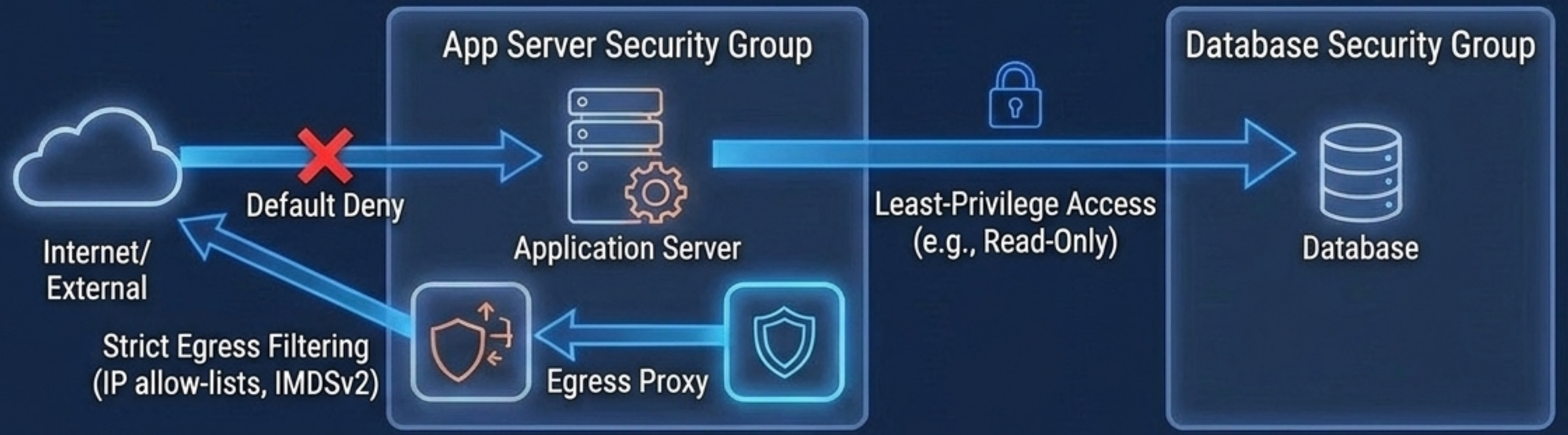
End-to-End Framing & Protocol Enforcement



- **Adopt HTTP/2 End-to-End**
  - Eliminate HTTP downgrades between the load balancer and backend to neutralize framing desync (Request Smuggling).
- **Strict RFC 9112 Validation**
  - Configure edge proxies to drop requests with ambiguous framing (spaces in methods, duplicate Content-Length, obfuscated Transfer-Encoding).
- **Normalize Before Forwarding**
  - Sanitize X-Forwarded-* headers at the trust boundary.

Gemini Notebook

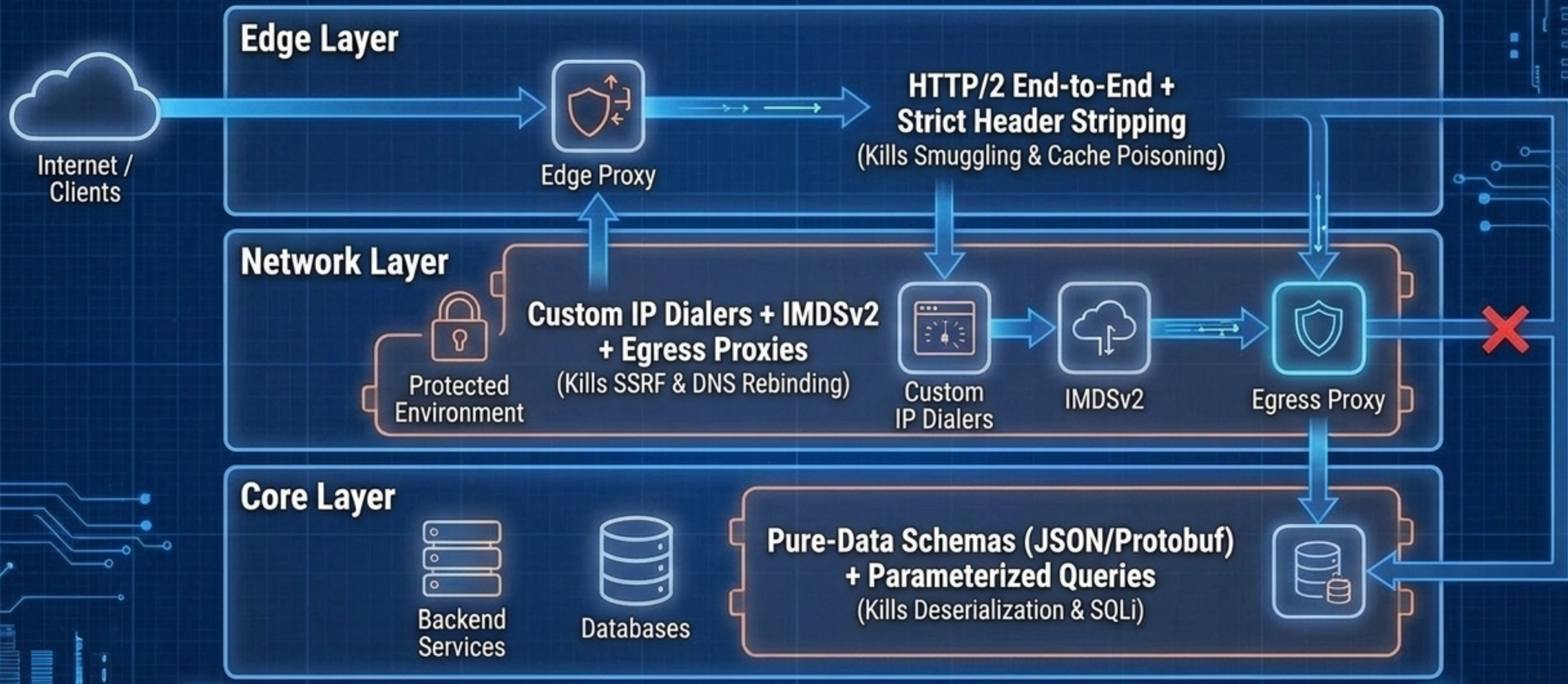
Least-Privilege & Egress Isolation



- **Database Layer**
  - Implement least-privilege DB roles. Separate Read-Only, Write, and DDL accounts.
  - Expose data via SQL Views rather than raw tables to mitigate SQLi fallout.
- **Network Layer**
  - Default-deny all outbound VPC traffic.
  - Force necessary webhooks/fetches through dedicated egress proxies with strict IP allow-lists and IMDSv2 enforcement.

Gemini Notebook

The Master Hardening Blueprint (Synthesis)



True infrastructure security is not achieved by chasing individual payloads, but by **eliminating architectural ambiguity** and parser discrepancies.

Gemini Notebook

End of S03E05: Server & Proxy Parser Attacks

Recommended References:

OWASP Cheat Sheet Series (Deserialization, SQLi, SSRF)
PortSwigger Web Security Academy & Research (HTTP Desync, Cache Poisoning)

```
sys.exit(0)
```

Gemini Notebook