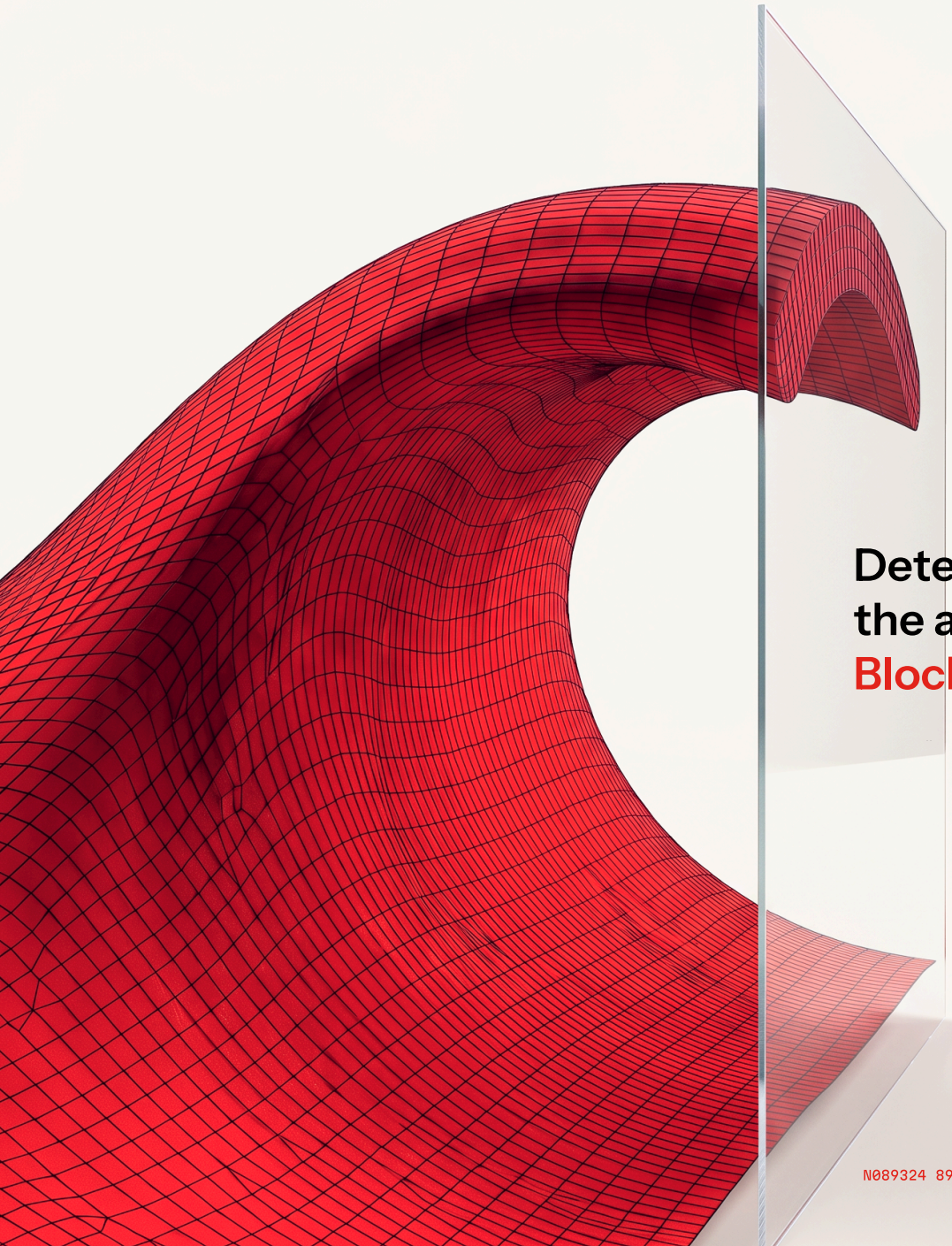
OLIGO

Threat Brief

Zero-Days Are Being Exploited in the Wild. Is Your Blocking Enabled?

In recent weeks, Oligo has detected zero-day exploitation in first-party code, where there's no CVE and no patch. Blocking is what stops it.



Detection shows you the attack.
Blocking ends it.

oligo.security

N089324 8923759882 7712 393432 H2384762 II98238

Since September, Oligo has detected multiple attempts to exploit zero-day vulnerabilities in live workloads. With blocking on, customers can stop these attacks at the point of execution.

These attempts followed a consistent pattern: server-side template injection (SSTI) against first-party code. First-party code has no CVE. There is no advisory to track, no patch to wait for, and no signature for a vulnerability scanner to match. The vulnerability lives only in code your team wrote, and the first sign of it is the attack itself.

Once exploitation starts, detection tells you it's happening. Blocking decides whether it succeeds.

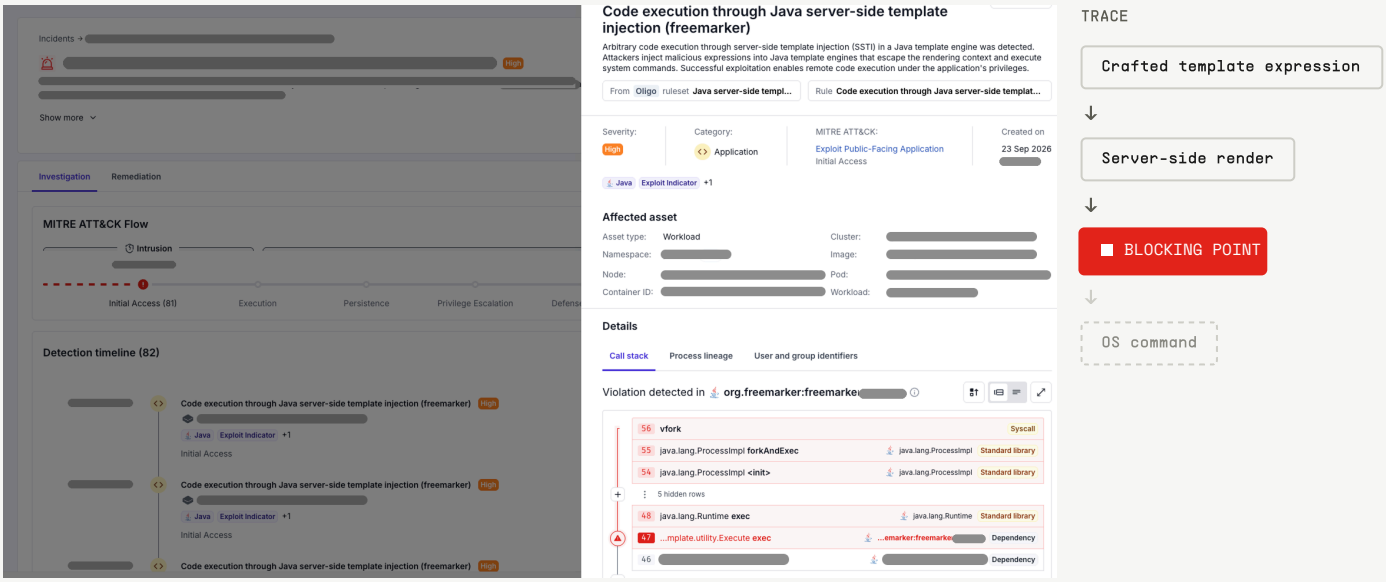
1

Detected at Runtime, Where Oligo Blocking Can Stop It

Oligo has detected this technique repeatedly. Here's one example.

SSTI in a Java Application

First-party code | Template engine | Server-side template injection



A crafted template expression came in through a user-controlled input and was rendered server-side, where it attempted to execute an OS command. Oligo observed the template engine spawning a process it never spawns in normal operation. With blocking on, the call is stopped before the command runs.

OUTCOME **Detected in production, at the exact call blocking stops.**

2

Why First-Party Zero-Days Get Past Other Controls

First-party code

No CVE.
No advisory.
No patch.
No signature.

The vulnerability lives only in code your team wrote.

SCANNERS & SCA

SCA tools track known vulnerabilities in open-source packages, and a first-party flaw has no CVE to match. SAST catches some injection patterns, but what it misses ships with no advisory.

WAF

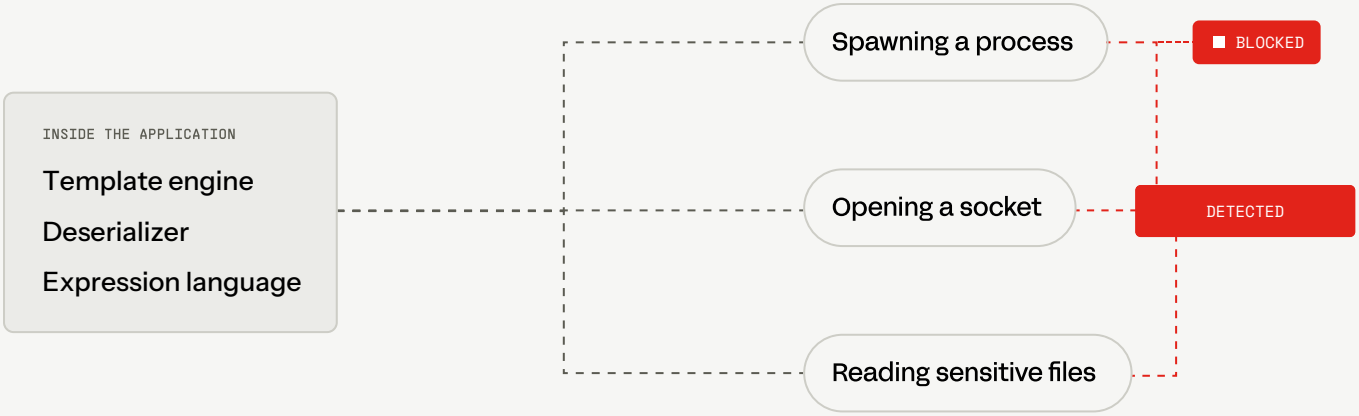
WAFs filter known payload patterns at the edge, and template syntax varies by engine and is easy to encode around.

EDR & CNAPP

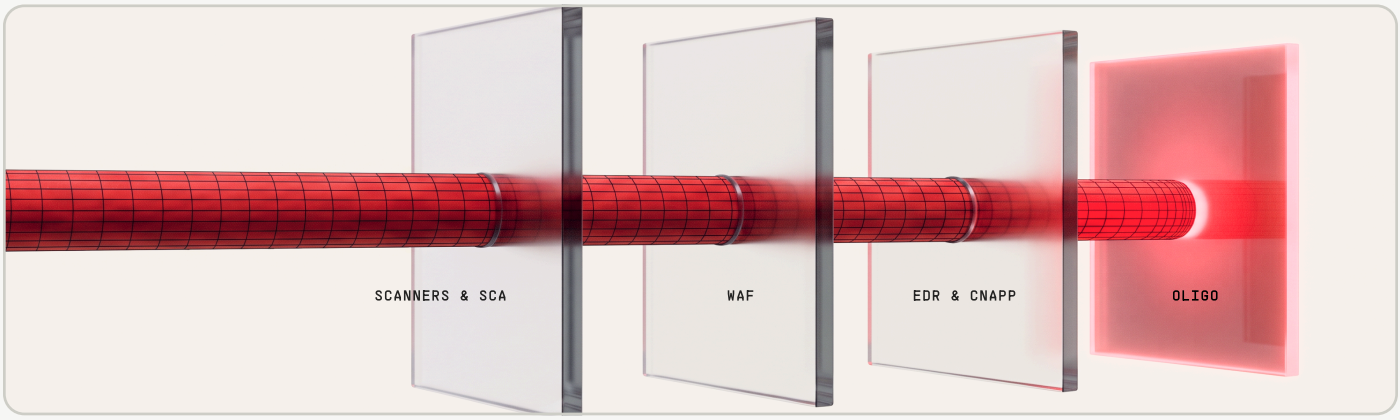
EDR and CNAPP see processes on the host, but they cannot see that a process came from a template render inside the application, and they don't block at the application layer at all.

How Oligo sees it

Attackers can change the payload, but not the code that executes it. Oligo sees the full call stack, so it knows when a template engine spawns a process. Blocking stops it there.



Oligo doesn't need to know the vulnerability in advance, which is why it catches zero-days that signature-based controls miss.



3

Blocking Stops the Exploit, Not the Application

One customer's story

01 Detected
Oligo caught a zero-day exploit in the customer's application.

→ 02 Blocking on
The customer turned on blocking rules.

→ 03 Tried again
The attacker came back with the same exploit, dozens of times.

→ 04 Stayed up
Blocking stopped them, and the application kept running.

Next step

Start Blocking Now

NOT RUNNING OLIGO YET?

Find out what's executing inside your applications right now.

SEE WHAT YOUR TOOLS MISS →

ALREADY AN OLIGO CUSTOMER?

Your Oligo team can confirm which rules cover your stack and turn them on.

TALK TO YOUR OLIGO TEAM →

