IMMUNIO

How Does IMMUNIO Work

Your IMMUNIO Questions Answered

How does IMMUNIO work

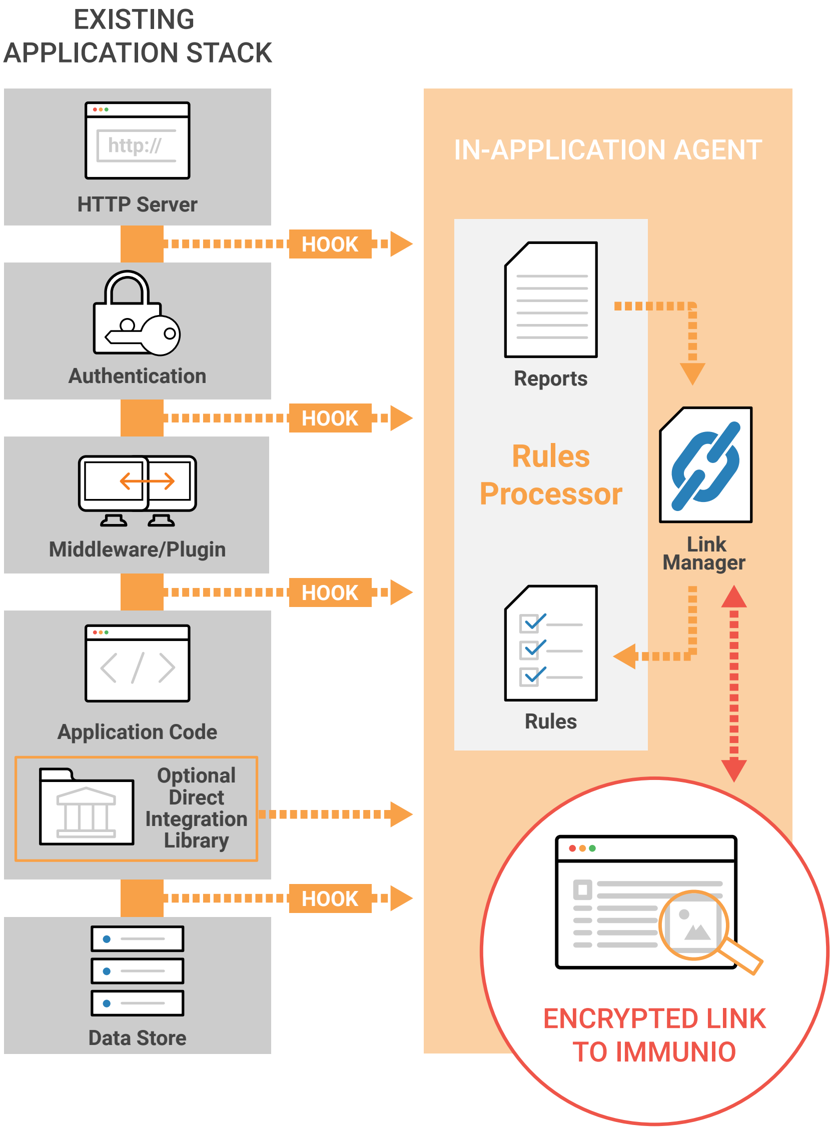
IMMUNIO's technology is designed to enable companies to release apps into production, secure in the knowledge that those applications will be immune to exploitation and that vulnerabilities can be quickly and efficiently remediated. IMMUNIO ensures application security from the inside – monitoring to detect vulnerabilities and immediately blocking unwanted activity in real time. The result is unprecedented protection, keeping web application owners and their users safe from hacks.

As with many advanced and innovative technologies, questions abound, so we have prepared a short series of documents to answer the most common questions we're asked after introducing the concept. We hope you find them useful.

HOW DOES IMMUNIO ACTUALLY WORK?

The IMMUNIO agent software installs in the same way as any other third party package used by your app - for Ruby, it's a Ruby Gem; for Java, it's a jar file; for Python, it's a Python package.

When your app is deployed, the IMMUNIO library starts up and adds sensors to those parts of your application that are relevant to the security of the application. These sensors enable IMMUNIO to "see" incoming request details - URL, headers, parameters, etc. The agent also hooks within the application to inspect SQL queries, template rendering, file access, redirects, logins, and so forth in order to track the activity of each request.



As the requests flow through the system, IMMUNIO agents inspect how the system reacts to each incoming HTTP request in order to determine whether its intentions are good or bad. Ill-intentioned requests will trigger the agent to take one of two courses of action, depending on how you choose to configure IMMUNIO:

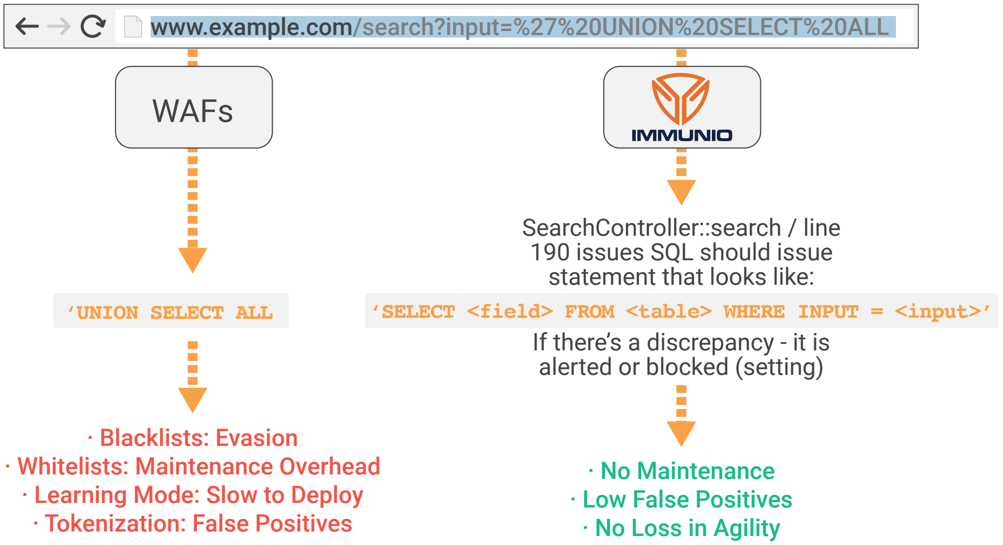
- activate tracking of the event on the IMMUNIO dashboard, so that you can watch its progress to determine what action should be taken
- initiate proactive protection of the application immediately to prevent any security breaches from taking place

The IMMUNIO agent handles the request inspections on your servers, which ensures that the analysis happens quickly and without introducing network response latency. This approach also minimizes the amount of data transmitted to IMMUNIO's servers.

IS IMMUNIO SIGNATURE-BASED?

Signature-based detection, as we all know from 25+ years of fighting viruses, is prone to false positives and evasion by attackers, aka zero-day threats. Rather than rely on signatures to separate legitimate and non-legitimate actions on the codebase, IMMUNIO learns what activity is normal for each line of code and, more importantly, what is not. Based on this information, the software automatically generates strong rules that can't be bypassed to permit activity that's known to be acceptable while barring previously unseen and/or unexpected behavior. Once tuned over a period of working with a particular application, this approach generates almost no false positives.

Here's what the IMMUNIO agent "sees", compared with a WAF


www.example.com/search?input=%27%20UNION%20SELECT%20ALL
WAFs
IMMUNIO
SearchController::search / line 190 issues SQL should issue statement that looks like:
'UNION SELECT ALL
'SELECT <field> FROM <table> WHERE INPUT = <input>'
If there's a discrepancy - it is alerted or blocked (setting)
· Blacklists: Evasion
· Whitelists: Maintenance Overhead
· Learning Mode: Slow to Deploy
· Tokenization: False Positives
· No Maintenance
· Low False Positives
· No Loss in Agility

IMMUNIO does use signatures at the first level of filtering, to help classify scanners and for some basic detection, but it is primarily used to process graphical displays within the dashboard, not for alerting or protection. In each situation, IMMUNIO applies the most appropriate algorithm available for each class of attack; this is determined by examining the internal details of each request and action, which can be accessed because the agent is running within the protected application space.

WHAT INFORMATION DOES IMMUNIO COLLECT, AND HOW IS IT USED?

IMMUNIO is equipped with sensors to inspect HTTP request and response details (status, URLs, headers, payloads, etc.). The agent also inspects SQL queries, template rendering, redirects, file access, shell command execution, calls to eval(), and authentication events (login, logout, current_user, failed logins, password reset requests, and more.) You can also provide IMMUNIO with information about custom events that are relevant to expected behavior in your app, like failed credit card transactions and other business errors.

Because the agent runs within the customer's application, IMMUNIO is able to analyze all the available data without moving sensitive information through any external services, preserving privacy and confidentiality for the customer. IMMUNIO does synchronize some data through its cloud service to keep all its agents in sync, including some sanitized fingerprints on SQL query structure, HTML template structure, normal file access patterns, and other non-sensitive information. Significant user activity events are also tracked in order to monitor tolerance thresholds for logins, logouts, account creation, etc.

IMMUNIO analyzes all this information for every request to decide if it should be allowed to proceed, or if some protective measures should be taken. It does all of this with a <5% impact on application performance.

HOW IS IMMUNIO DIFFERENT FROM A WAF?

The fundamental difference between IMMUNIO and web application firewalls is that IMMUNIO operates from inside the protected application, with the attendant specialized knowledge that comes along with that position. WAFs operate in front of the protected application, with no knowledge of the inner workings of the app.

From its privileged position inside the app, IMMUNIO has direct access to all the rich context of the application itself, whereas a WAF has to make guesses and assumptions as to how a request might affect the operation and security of the app. And, because the WAF doesn't know much about the application it's protecting, it relies heavily on attack signatures to identify and block bad input.

What this means in practice is that an attacker can bypass WAF protection simply by adding extra encoding to a request so it no longer matches any of the signatures the WAF is configured to block. The payload arrives at the application, the additional encoding is stripped away, and the damage is done. With IMMUNIO, the protection is inside the application, so all the encoding is stripped away by the app itself before the agent checks for bad payloads. Because IMMUNIO is not dependent on signatures, it doesn't matter how many attack variants the bad guys throw at the app – they won't get through.

HOW DOES THE IMMUNIO AGENT USE THIS INFORMATION TO MAKE DECISIONS?

The agent contains a JIT-enabled embedded runtime engine that applies IMMUNIO's proprietary security algorithms to every request. As noted earlier, this patented technology is fast enough that the full suite of algorithms can be run on every request without impacting application performance.

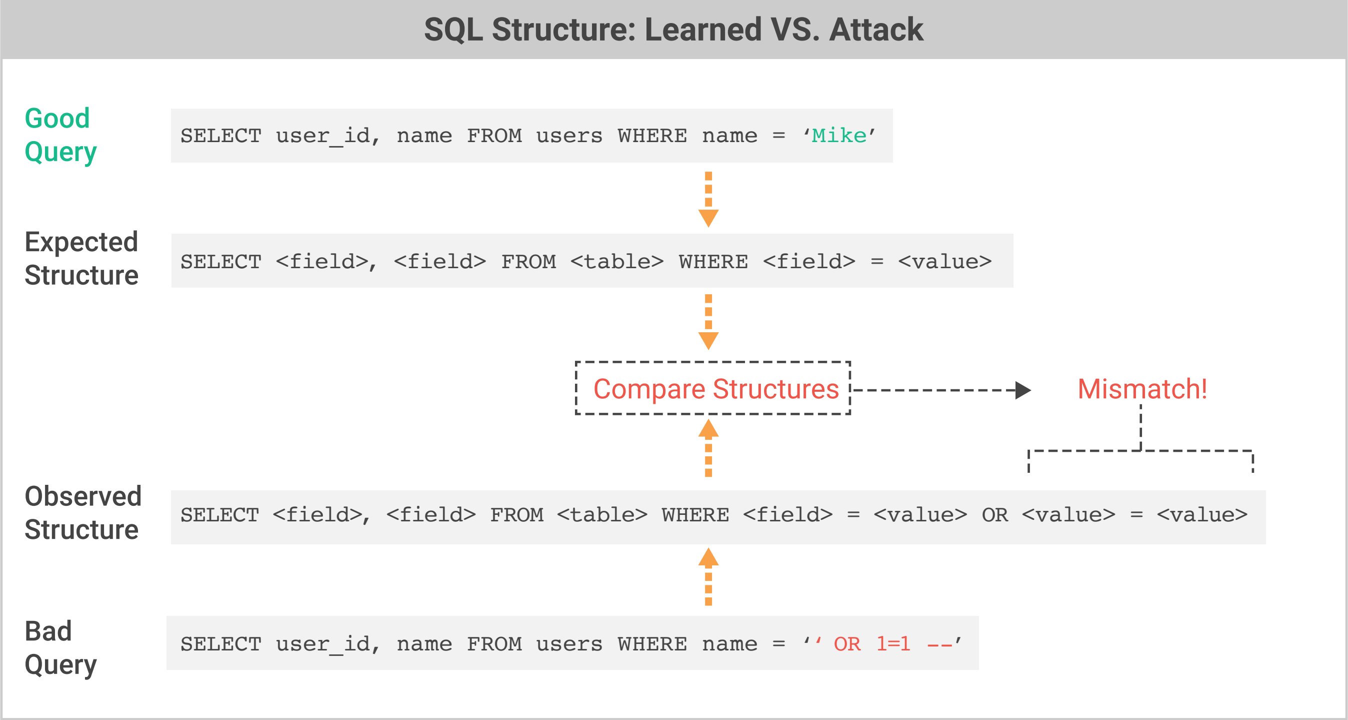
For most attack types, the full detection and protection process is run from within the agent. For detection that depends on rates over time, such as brute force attacks, the data is synchronized to IMMUNIO's hosted infrastructure to ensure the rate information from all of the customer's servers has been included.

WHAT OTHER ACTIONS DOES IMMUNIO TAKE TO DETECT AND STOP THREATS?

IMMUNIO applies different algorithms for each attack type, ensuring the most appropriate blocking technique is always deployed.

- For SQL Injection attacks, the agent examines every line of code involved in executing an SQL statement. Each time an SQL statement is run, IMMUNIO keeps track of where in the code that request originated so that it can quickly identify which each line of code should be executing, and what the structure of those SQL queries should look like. Once IMMUNIO has "seen" the expected structure, it can alert on any change resulting from an injection.

For example, if the agent determines that line 118 has a vulnerability and has seen what structure the SQL query should have based on good traffic, it can recognize when the structure is not as it should be and deduce that an SQLi attack is in progress. IMMUNIO does this same basic structure-change detection with all non-normal code interactions, including remote code execution and unauthorized redirects.

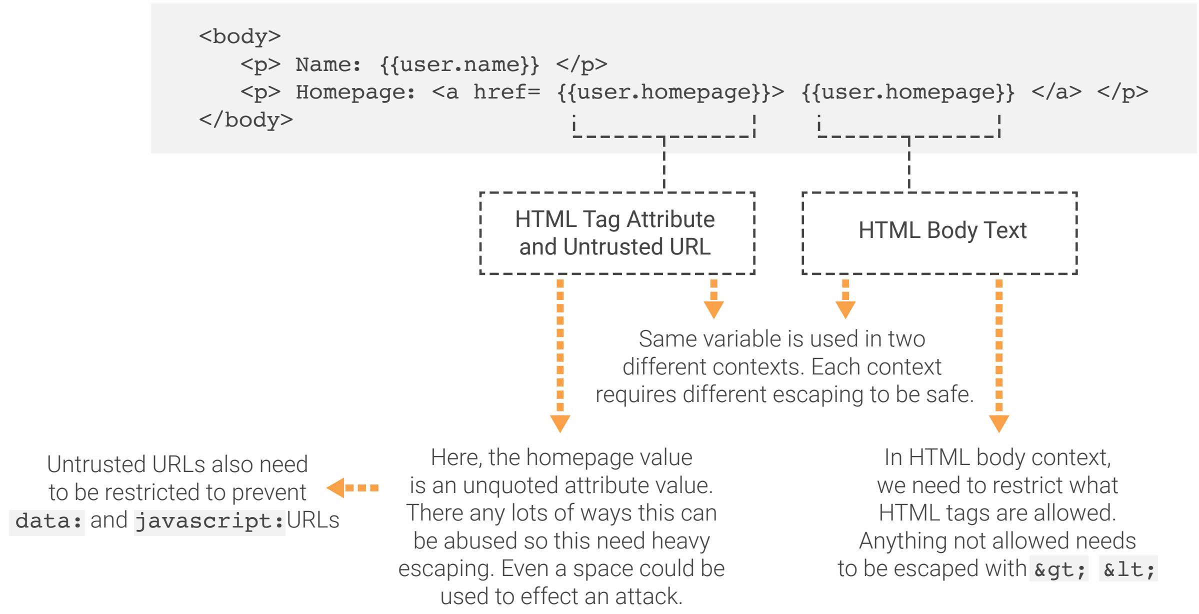


SQL Structure: Learned VS. Attack

```
SELECT user_id, name FROM users WHERE name = 'Mike'
SELECT <field>, <field> FROM <table> WHERE <field> = <value>
SELECT <field>, <field> FROM <table> WHERE <field> = <value> OR <value> = <value>
SELECT user_id, name FROM users WHERE name = '' OR 1=1 --'
```

- For XSS, the agent watches each template being rendered and tracks every variable insertion point, enabling it to learn the types of data and structure rendered into each variable placeholder. Once the template is rendered, IMMUNIO parses the result to identify the context of each variable. For each one, it knows if the variable is in HTML text, a tag attribute, CSS, Javascript, etc. and can escape the supplied data correctly based on that context. Again, once IMMUNIO knows what it should be seeing, it can start alerting on any suspicious input not normally present.

XSS Contexts



```
<body>
   <p> Name: {{user.name}} </p>
   <p> Homepage: <a href= {{user.homepage}}> {{user.homepage}} </a> </p>
</body>
```

Same variable is used in two different contexts. Each context requires different escaping to be safe.

Untrusted URLs also need to be restricted to prevent `data:` and `javascript:` URLs

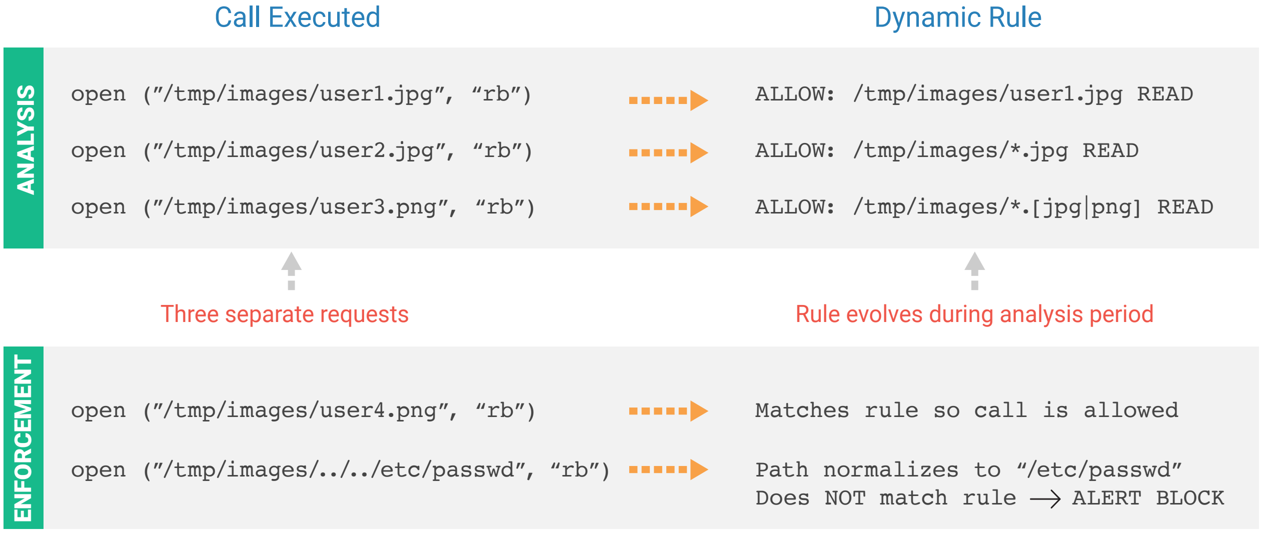
Here, the homepage value is an unquoted attribute value. There any lots of ways this can be abused so this need heavy escaping. Even a space could be used to effect an attack.

In HTML body context, we need to restrict what HTML tags are allowed. Anything not allowed needs to be escaped with `&gt;` `&lt;`

- File Access is tackled in a similar way to the two above threat types. The agent tracks every place in the code that opens files, and watches which files are opened for reading and writing. Once IMMUNIO has identified where in the file system each line of code typically accesses, the agent can alert if there is a deviation from the anticipated action.

File Access

We track file access based on the application stacktrace and the filename and line number.



	Call Executed	Dynamic Rule
ANALYSIS	`open ("/tmp/images/user1.jpg", "rb")`	`ALLOW: /tmp/images/user1.jpg READ`
	`open ("/tmp/images/user2.jpg", "rb")`	`ALLOW: /tmp/images/*.jpg READ`
	`open ("/tmp/images/user3.png", "rb")`	`ALLOW: /tmp/images/*.[jpg|png] READ`
	Three separate requests	Rule evolves during analysis period
ENFORCEMENT	`open ("/tmp/images/user4.png", "rb")`	`Matches rule so call is allowed`

	`open ("/tmp/images/../../etc/passwd", "rb")`	`Path normalizes to "/etc/passwd"`
		`Does NOT match rule → ALERT BLOCK`

- For threshold-based attacks, if the agent sees a significant event like a failed login, a few details (username, IP, User-Agent) are sent back to IMMUNIO's hosted infrastructure. The number of failures from that IP and against that username is monitored to see if a particular IP is involved, or if a particular username is being targeted.

HOW ABOUT PROACTIVE PROTECTION?

IMMUNIO has a few different tools in its arsenal to protect against attackers; as with detection, the exact method varies depending on the attack type.

- For SQL Injection, File Access, RCE, and Open Redirect, if IMMUNIO protection is enabled, it simply blocks bad requests and returns a 403 Forbidden response.

- For XSS, because IMMUNIO is watching the templates as they are rendered, the correct encoding can be applied to the attack payload to prevent the vulnerability from being exploited, protecting the customer's users.

- For threshold-based attacks, IMMUNIO identifies the source of the bad requests and takes action against that source. At that point, it can either block that attacker (IP or Username) for a given period of time, or stop the attacker with a CAPTCHA to block automated bot attacks.

STILL HAVE QUESTIONS?

Additional *YOUR IMMUNIO QUESTIONS ANSWERED* documents are available on request:

- How does IMMUNIO impact application performance?
- How does IMMUNIO protect user data?
- IMMUNIO integration roadmap

Please don't hesitate to contact us at info@immun.io with any unanswered questions and for more information.

ABOUT IMMUNIO

IMMUNIO is a pioneer in real-time application self-protection (RASP), providing automatic detection and protection against application security vulnerabilities. The company's mission is to make truly effective real-time web protection technology easily available and widely deployed, and by doing so, stop the biggest source of breached data records. **For more information, visit www.immun.io.**