Erlang Solutions

Security Audit for Erlang & Elixir (SAFE)

Validate your code.
Protect your business

What is it?

The Security Audit for Erlang & Elixir (SAFE) is an intensive investigation into Erlang or Elixir code to identify vulnerabilities that would make the code susceptible to cyber attacks.

Who is it for?

- All users of Erlang and Elixir, but especially those businesses that require high availability and resilience of their mission-critical applications for many different reasons:
    - **Business continuity**—where there is an impact to the operation of the business if an Erlang/ Elixir node, system or application goes down
    - **Legislation**—industries where system availability is mandatory and failure to meet requirements would lead to non-compliance and significant financial consequences
    - **Customer experience**—guaranteeing systems are available for customers to carry out their daily lives
    - **Corporate reputation**—where system downtime or any form of security breach negatively impacts the reputation of the business, affecting its trading performance and potentially its responsibility to shareholders
- Businesses that have already been impacted by a security vulnerability within their Erlang or Elixir code base and want to investigate what other parts of their system may be vulnerable
- Anyone launching new Erlang or Elixir code, that would benefit from the reassurance that an audit can provide before going live
- Anyone concerned about the impact of a cyber attack

How does SAFE work?

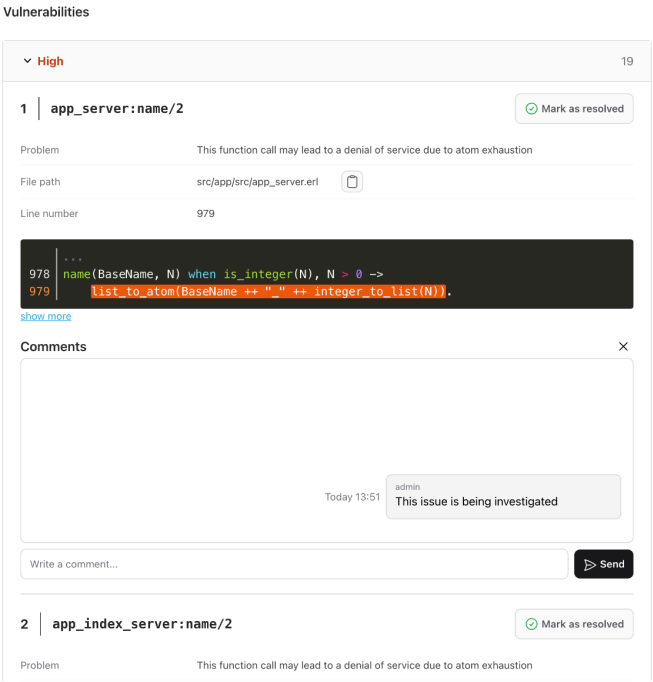
Security Audit for Erlang & Elixir is an audit procedure, carried out by our experts, aided by software analyzing tools. The findings are delivered in a comprehensive online report detailing:

- Any security vulnerabilities found
- The level of risk identified - categorised into low, moderate or high
- Detail of each vulnerability including a description of the issue, the file path and the code in question
- Ability to comment on issues and mark them resolved

It includes:

A security overview

The security overview gives a summary of the vulnerabilities found in the software. The overview aims to help team leads and managers to easily identify the major security weaknesses in the code base.



Denial of Service

We manage unique atom values through a global atom table. As your system runs, new atom values are automatically added to this table as needed. It's like expanding a dictionary with new words.

However, here's the technical bit: we don't remove entries from this table once they're in there. The table's size is initially set based on system configuration, usually around a million entries. If you ever attempt to add a new value when the table is already full, it can cause the virtual machine to crash. So, it's important to configure this table size wisely to ensure the smooth operation of yourErlang or Elixir system.

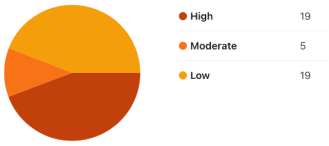
Denial of Service

Each unique atom value in use in the virtual machine takes up an entry in the global atom table. New atom values are appended to this table as needed, but entries are never removed. The size of the table is determined at startup, based on the +t emulator flag, with a default of 1.048.576 entries. If an attempt is made to add a new value while the table is at capacity, the virtual machine crashes.

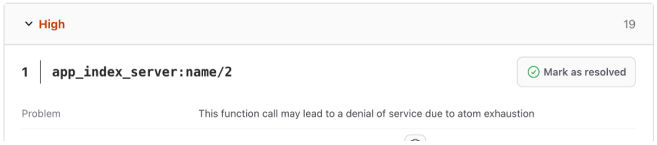
Preventing atom exhaustion | EEF Security WG

8134 functions were analysed, of which 19 has high, 5 has moderate and 19 has low severity vulnerabilities.

Distribution of vulnerable functions



Vulnerabilities



Race condition

Even Erlang and Elixir are not immune to race conditions, despite its model. Race conditions can happen, for example, in databases such as Mnesia.

Deprecated calls

As many programming languages, Erlang and Elixir also need to deprecate certain functions from time to time. Some function depreciations, like the crypto module, are based on security concerns, while others are needed for the improvement of the ecosystem. These deprications need to be monitored to ensure business continuity.

Injections

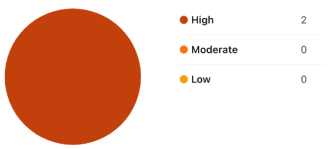
Erlang and Elixir are known for the ability to dynamically upgrade code while the system is running, allowing new versions of modules to be loaded and executed without interrupting the system's operation. However, this feature is considered potentially vulnerable as it allows incompatible, erroneous, or even malicious code to be loaded into the system, leading to data leaks or crashes.

Man in the middle

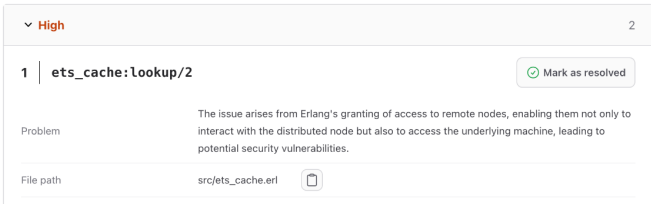
Man in the middle attacks are when instead of the expected communication route, the cyberattcker diverts the route and joins the communication, to see or even replace the data sent from one component to another.

8134 functions were analysed, of which 2 has high, 0 has moderate and 0 has low severity vulnerabilities.

Distribution of vulnerable functions



Vulnerabilities



Man in the middle

A man-in-the-middle attack occurs when a cyber attacker diverts the expected communication route, inserting themselves into the communication to intercept, view, or even replace the data being sent between two components.

Why Erlang Solutions?

Expert collaboration

SAFE has been developed in collaboration with the Eötvös Loránd University, a prominent institution in Computer Science, and is based on a solid theoretical foundation and extensive research.

Erlang & Elixir proficiency

Erlang Solutions is the world leading consultancy for Erlang and Elixir. Our specialists have extensive experience supporting businesses across the world with their mission-critical systems and architecture.

Additional support

Beyond the report, we are the perfect team to help with any further support you might need. The findings of your report may indicate a bigger issue requiring a thorough code and architecture review. We can also offer training in developing secure Erlang and Elixir code, should any of your team need upskilling

Timely service

Most security audits are completed within 2-5 business days, depending on the size and complexity of the system.

www.erlang-solutions.com
Erlang Solutions
Europe | USA | South America

To find out more, or talk to one of our team, please contact us on security-audit@erlang-solutions.com