Mapeando a BEAM: Um Censo Global de Nós Erlang Expostos na Internet

Fernando Areias
Programa de Pós-Graduação em Computação Aplicada (PPGCA), Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba, Brasil
fernandoareias@alunos.utfpr.edu.br

Ana Cristina Kochem Vendramin
Programa de Pós-Graduação em Computação Aplicada (PPGCA), Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba, Brasil
criskochem@utfpr.edu.br

Adolfo Neto
Programa de Pós-Graduação em Computação Aplicada (PPGCA), Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba, Brasil
adolfo@utfpr.edu.br

Resumo

A BEAM, máquina virtual de Erlang, é uma plataforma essencial para sistemas concorrentes, servindo de base para uma série de linguagens de programação funcional, como Erlang, Elixir e Gleam, além de diversos sistemas distribuídos. Apesar dessa ampla adoção, um mapeamento da presença da BEAM em escala global permanece ausente na literatura, assim como uma análise das implicações de segurança decorrentes dessa exposição. Nesse contexto, este artigo apresenta um levantamento de nós Erlang expostos na Internet, com o objetivo de mapear a presença da máquina virtual BEAM em escala global. Foram coletados 681.326 endereços IP únicos por meio da API do Shodan para Python, cujos dados foram processados a fim de responder às perguntas de pesquisa. A China concentra a maior parcela de nós (30,79%), seguida pelos Estados Unidos (20,83%) e pela Alemanha (10,07%). O RabbitMQ mostrou-se o serviço dominante, respondendo por 60,10% das ocorrências, o que evidencia o papel central da BEAM na infraestrutura de mensageria moderna. A maior parte dos nós está implantada em provedores de nuvem, que respondem conjuntamente por mais de 34% dos endereços identificados. Os resultados obtidos contribuem tanto para a comunidade técnica, ao oferecer um mapeamento até então inexistente na literatura, quanto para a área de segurança no desenvolvimento de software, ao quantificar a superfície de ataque associada à exposição do EPMD.

Palavras-chave

BEAM, Erlang, EPMD, Internet Exposure, Attack Surface, Shodan

1 Introdução

Erlang é uma linguagem funcional e orientada a mensagens, criada por Joe Armstrong, Robert Virding e Mike Williams na Ericsson durante a década de 1980, originalmente projetada para sistemas de comutação de telecomunicações [3, 4]. A linguagem é executada sobre a máquina virtual BEAM (*Bogdan/Björn's Erlang Abstract Machine*), cuja arquitetura, baseada em garantias de concorrência massiva, tolerância a falhas e distribuição nativa, a estabeleceu como plataforma fundamental para outras linguagens funcionais como Elixir[1] e Gleam[2]. A BEAM também sustenta uma ampla gama de sistemas distribuídos, bancos de dados, *brokers* de sistemas de mensageria e infraestruturas de telecomunicações em larga escala [8]. Apesar dessa ampla adoção, um mapeamento da presença da BEAM em escala global permanece ausente na literatura, assim como uma análise das implicações de segurança decorrentes dessa exposição.

Nesse contexto, este artigo apresenta um levantamento de nós Erlang expostos na Internet, com o objetivo de mapear a presença da máquina virtual BEAM em escala global, identificar os principais serviços em execução nesses nós e evidenciar a exposição do EPMD (*Erlang Port Mapper Daemon*), serviço responsável por registrar e mapear as portas utilizadas pelos nós Erlang em um *host*, permitindo a descoberta e o estabelecimento de conexões entre processos distribuídos. Por não empregar criptografia nem mecanismos de autenticação em sua comunicação padrão, o EPMD possibilita que agentes externos consultem livremente os nós registrados, ampliando significativamente a superfície de ataque de infraestruturas baseadas em Erlang [12].

Este artigo está estruturado da seguinte forma. A Seção 2 apresenta o referencial teórico, abordando os conceitos fundamentais sobre a linguagem Erlang, a plataforma Shodan e a técnica de *banner grabbing*. A Seção 3 detalha a metodologia empregada, incluindo as perguntas de pesquisa formuladas e o processo de obtenção e tratamento dos dados coletados. A Seção 4 apresenta a análise dos resultados obtidos, respondendo às perguntas de pesquisa propostas e discutindo as limitações do estudo. Por fim, a Seção 5 apresenta as conclusões do trabalho, sintetizando as principais contribuições e implicações dos resultados, bem como as possibilidades de trabalhos futuros identificadas.

2 Referencial Teórico

Esta seção apresenta os principais conceitos e tecnologias que fundamentam este trabalho. Inicialmente, descrevem-se a linguagem Erlang e os sistemas distribuídos construídos a partir dela. Em seguida, apresenta-se o Shodan, um mecanismo de busca utilizado para a identificação de dispositivos expostos na rede. Por fim, aborda-se a técnica de *banner grabbing*, empregada na coleta de informações sobre serviços de rede.

2.1 Erlang

Erlang é uma linguagem funcional orientada à concorrência e nativamente distribuída. Devido a essas características, ela serve de base para uma ampla variedade de sistemas distribuídos, que vão desde *brokers* de mensagens até bancos de dados [3].

[1] https://elixir-lang.org
[2] https://gleam.run/

O CouchDB desenvolvido pela Apache Software Foundation em 2005, é um banco de dados NoSQL (*Not Only SQL*) orientado a documentos JSON (*JavaScript Object Notation*) que utiliza JavaScript para consultas e possui aplicações em diversos setores do mercado [2, 5, 16]. Ainda no ecossistema Erlang, o ejabberd [1] é um servidor de mensagens instantâneas criado em 2002, que serviu como base para o Messenger do Facebook e o WhatsApp. De forma semelhante, o MongooseIM [7], desenvolvido pela Erlang Solutions, é um servidor XMPP (*Extensible Messaging and Presence Protocol*) robusto e escalável, voltado a plataformas de mensagens instantâneas de grande escala em ambientes corporativos, servindo como base para soluções similares ao Microsoft Teams ou ao Slack.

No âmbito de servidores web, o Cowboy e o Mochiweb [9, 13] são soluções leves também escritas na linguagem, amplamente adotados como servidores HTTP (*Hypertext Transfer Protocol*) em aplicações Erlang e Elixir. Na área de IoT (*Internet of Things*), o EMQX [6] é uma plataforma MQTT (*Message Queuing Telemetry Transport*) altamente escalável e flexível, utilizada por grandes empresas como BMW e SAIC Volkswagen para a comunicação segura entre veículos em tempo real.

Por fim, o Riak [14] é um banco de dados NoSQL do tipo chave-valor, distribuído, que oferece alta disponibilidade, tolerância a falhas, simplicidade operacional e escalabilidade por concepção.

2.2 Shodan

O Shodan[3] é um mecanismo de busca semelhante ao Google, porém focado em dispositivos e endereços IP (*Internet Protocol*) expostos publicamente na rede. Devido à elevada precisão de seus dados, tornou-se uma ferramenta fundamental para especialistas em segurança na etapa inicial de identificação de vulnerabilidades. Ao realizar uma busca por palavra-chave, a plataforma analisa cabeçalhos HTTP e metadados, exibindo informações como o endereço IP, portas abertas e localização geográfica (país, cidade e coordenadas), as quais são atualizados constantemente para refletir a dinamicidade da Internet [10].

Além da interface web, o Shodan oferece uma API (*Application Programming Interface*) paga que permite a automação por meio de *scripts*. Através da biblioteca proprietária `shodan` para Python é possível utilizar métodos como o `search_cursor` para consultas em massa ou funções de *streaming* de dados, cujo uso é gerenciado por créditos disponibilizados mediante assinatura [15].

2.3 *Banner Grabbing*

Banner grabbing é uma técnica de reconhecimento amplamente difundida, presente em ferramentas largamente utilizadas no mercado, como Nmap, ZMap, WhatWeb, entre outras [10, 11]. A ideia consiste em extrair informações sobre serviços de rede por meio de mensagens de saudação ou dados de versão exibidos pelo *host*. Essas informações, como o nome, o título e a edição do software, podem ser utilizadas por um atacante para mapear a superfície de ataque disponível. A captura de *banner* pode ser realizada de forma manual ou automatizada, com o uso de ferramentas de OSINT (*Open-Source Intelligence*), sendo uma etapa essencial tanto em testes de penetração de natureza ofensiva quanto defensiva [10, 11].

Existem duas estratégias de *banner grabbing*: a primeira consiste no mapeamento ativo através da abertura e envio de pacotes TCP (Transmission Control Protocol) diretamente aos alvos, o que pode ser facilmente detectado e bloqueado por sistemas de detecção de intrusão, como o Snort. A outra abordagem seria a captura passiva de *banners*, realizada via *sniffing* de rede ou por meio de serviços intermediários que abstraem a comunicação direta com o *host* alvo [11].

3 Metodologia

Este capítulo descreve a metodologia utilizada no desenvolvimento deste artigo, bem como a estratégia de busca e o processamento dos dados. Os dados utilizados, assim como os *scripts* em Python, estão disponíveis no repositório do GitHub[4].

3.1 Perguntas de Pesquisa

O objetivo deste trabalho é compreender onde e como o Erlang está sendo utilizado no mundo real, a partir da identificação de nós acessíveis publicamente na Internet. Para isso, formulamos quatro perguntas de pesquisa que orientam a investigação:

- **PP1:** Quais países concentram o maior número de nós Erlang em ambientes de produção?
- **PP2:** Quais são os tipos predominantes de aplicações executadas sobre a BEAM?
- **PP3:** Qual é a quantidade de instâncias do EPMD expostas na internet?
- **PP4:** Quais provedores de internet (do inglês *Internet Service Providers* – ISP) e Sistemas Autônomos (do inglês *Autonomous Systems* – AS) concentram a hospedagem de nós Erlang expostos publicamente?

Ao responder essas perguntas, pretendemos traçar um panorama da presença do Erlang em infraestruturas de produção, contribuindo tanto para a comunidade técnica quanto para a discussão sobre a superfície de ataque associada a serviços BEAM expostos sem as devidas medidas de proteção.

3.2 Obtenção e Tratamento dos Dados

Foram coletados 681.326 endereços IP únicos por meio da API do Shodan para Python, utilizando o método `search_cursor` e a string de busca "Erlang", com o objetivo de identificar o maior número possível de *hosts* expostos na internet que executam potenciais nós Erlang.

Após a coleta inicial, foi realizado o processamento dos *banners* de cada *host*. Para cada IP único obtido na etapa anterior, o *script* estabelece conexões diretas em múltiplas portas associadas a serviços Erlang, além da porta original reportada pelo Shodan. Para cada porta, é utilizado um *probe* específico de acordo com o protocolo:

- **EPMD** (porta 4369): envia o comando `NAMES_REQ` para obter a lista de nós Erlang registrados;
- **AMQP** (porta 5672): realiza o *handshake* do protocolo AMQP 0-9-1 para identificar instâncias RabbitMQ;

[3] https://www.shodan.io

[4] https://github.com/fernandoareias/erlang_nodes_discovery

- **MQTT** (portas 1883/8883): envia um pacote `CONNECT` e aguarda o `CONNACK` para identificar servidores MOMs (*Message Oriented Middleware*) escritos em Erlang que utilizam o protocolo MQTT;
- **HTTP/HTTPS**: realiza uma requisição `GET /` e analisa os *headers* e corpo da resposta;
- **TCP genérico**: estabelece conexão e captura o *banner* inicial.

Os *banners* obtidos são, então, classificados por meio de expressões regulares que identificam serviços Erlang conhecidos como CouchDB, RabbitMQ, ejabberd, EMQX, Cowboy, Mochiweb, VerneMQ, Riak e MongooseIM. O processamento é paralelizado, utilizando um sistema de *checkpoint* que permite retomar a execução em caso de interrupção. Um *timeout* de 3 segundos é aplicado a cada conexão.

A Tabela 1 apresenta o conjunto completo de portas escaneadas durante o processo de *banner grabbing*.

Tabela 1: Portas escaneadas durante o *banner grabbing*

Porta	Protocolo	Serviço
80	HTTP	HTTP
161	TCP	SNMP
443	HTTPS	HTTPS
554	TCP	RTSP
1883	MQTT	MQTT
2082	HTTP	HTTP
2083	HTTPS	HTTPS
3000	HTTP	HTTP
4280	HTTP	CouchDB
4369	EPMD	EPMD
5222	TCP	ejabberd XMPP
5280	HTTP	ejabberd HTTP
5672	AMQP	RabbitMQ AMQP
5984	HTTP	CouchDB
6443	HTTPS	K8s API
6984	HTTPS	CouchDB SSL
8000	HTTP	HTTP
8001	HTTP	HTTP
8080	HTTP	Erlang HTTP
8443	HTTPS	Erlang HTTPS
8883	MQTT	MQTT SSL
9100	TCP	Erlang Distribution
15672	HTTP	RabbitMQ Management
25672	TCP	RabbitMQ Cluster

A próxima seção apresenta a análise dos resultados obtidos a partir dos dados coletados e classificados, buscando responder às perguntas de pesquisa formuladas e discutir as implicações da exposição de serviços Erlang na internet.

4 Análise dos Resultados

Dos 681.326 endereços IP coletados por meio do Shodan, 136.218 (19,99%) não responderam, enquanto 545.108 (80,01%) responderam com sucesso. Esse resultado já era esperado, considerando o modo de funcionamento do Shodan, que rastreia dispositivos dinâmicos mapeados na Internet, os quais podem ser desligados a qualquer momento ou ter seus endereços IP públicos reatribuídos. Desses 545.108 *hosts* que responderam, 58.374 (10,71%) estão executando nós Erlang, os quais analisaremos a seguir, a fim de responder às perguntas de pesquisa apresentadas anteriormente.

4.1 Concentração de Nós Erlang por País

A Figura 1 apresenta a distribuição dos nós Erlang por país. A China concentra a maior parcela, com 17.976 (30,79%), seguida pelos Estados Unidos, com 12.162 (20,83%), e pela Alemanha, com 5.881 (10,07%). A França ocupa a quarta posição, com 2.175 (3,73%), e a Holanda aparece em quinto lugar, com 1.564 (2,68%).

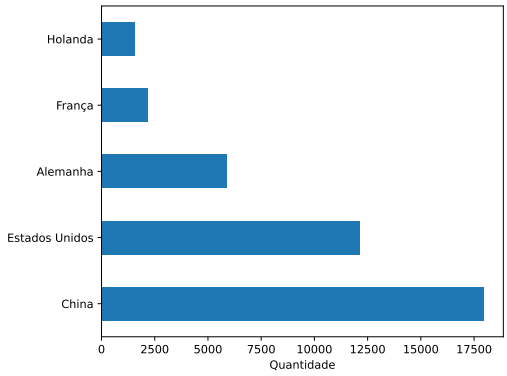

Figura 1: Distribuição de nós Erlang por país.

4.2 Aplicações Executadas na BEAM

A Figura 2 apresenta os cinco serviços mais frequentes identificados pelo mapeamento de portas. O RabbitMQ representa a maior parcela, com 73.721 ocorrências (60,10%), seguido pelo EPMD, com 21.715 (17,70%), e pelo Erlang genérico, com 16.503 (13,45%). Os MOMs escritos em Erlang que utilizam o protocolo MQTT aparecem na quarta posição, com 7.140 (5,82%), e o CouchDB em quinta, com 1.326 (1,08%). Os demais serviços, não exibidos na figura, apresentam participação inferior a 1%: ejabberd, com 723 (0,59%), Cowboy, com 711 (0,58%), EMQX, com 566 (0,46%), MongooseIM, com 145 (0,12%), Mochiweb, com 72 (0,06%), e Riak, com 49 (0,04%).

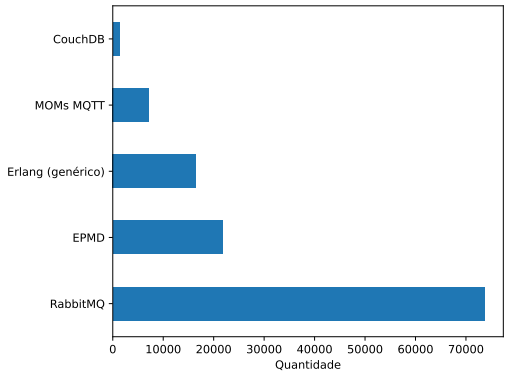

Figura 2: Distribuição dos serviços.

Ao analisar a distribuição desses serviços por país, alguns padrões se destacam. O RabbitMQ e o EPMD concentram-se predominantemente na China e nos Estados Unidos. Hong Kong apresenta uma proporção atipicamente elevada de nós Erlang genéricos, com 5.939 ocorrências, superando todos os demais países nessa categoria. Os MOMs escritos em Erlang que utilizam o protocolo MQTT possuem presença expressiva na China, e o CouchDB destaca-se nos Estados Unidos, com 411 registros.

4.3 Instâncias EPMD

Também analisamos a quantidade de nós Erlang com a porta do EPMD (4369) exposta na rede. Por meio dessa porta, enviamos uma requisição EPMD_NAMES com o objetivo de coletar os nomes dos nós e suas respectivas portas na máquina, evidenciando um problema de segurança já conhecido pela comunidade, porém ainda em aberto.

A Figura 3 apresenta os resultados obtidos desse procedimento. Dos 58.374 *hosts* analisados, 23.400 (40,09%) possuíam a porta do EPMD acessível, enquanto 34.974 (59,91%) não responderam à requisição. Nos 23.400 *hosts* com EPMD exposto, foi possível identificar um total de 37.589 nós Erlang em execução.

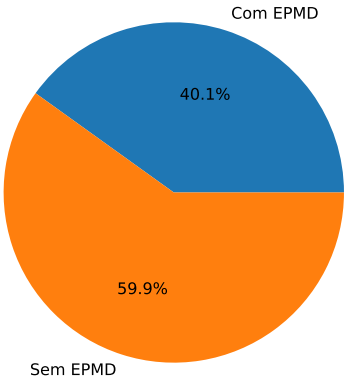

Figura 3: Distribuição dos EPMDs expostos.

4.4 Provedores de Internet e Sistemas Autônomos

A análise dos endereços IP identificados durante a varredura revelou uma concentração expressiva em provedores de infraestrutura de nuvem e serviços de hospedagem. A fim de evitar distorções decorrentes de variações nominais associadas a uma mesma organização, como diferentes entidades jurídicas que operam sob a mesma marca, os nomes dos provedores foram normalizados por meio da remoção de sufixos corporativos (*Inc.*, *LLC*, *Corporation*, entre outros) e da consolidação manual dos casos identificados, como as distintas razões sociais vinculadas à Amazon (*Amazon.com, Inc.* e *Amazon Technologies Inc.*).

A Figura 4 apresenta os quinze ISPs mais frequentes após esse processo de normalização. A Amazon lidera com 61.708 endereços (11,33% do total), seguida pela Deutsche Telekom, com 40.189 endereços (7,38%), e pela Internet Rimon, com 32.931 endereços (6,04%). A presença de provedores de nuvem e hospedagem no topo da lista, como Amazon, Hetzner, DigitalOcean, Oracle, Google, Microsoft e GoDaddy, que acumulam conjuntamente mais de 34% dos endereços identificados, indica que a maior parte dos nós Erlang expostos está implantada em ambientes de nuvem ou servidores dedicados, e não em redes residenciais ou corporativas tradicionais.

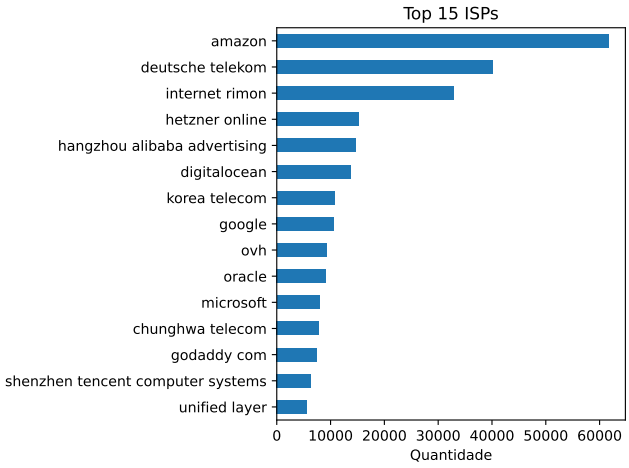

Figura 4: Quinze ISPs com maior número de endereços IP identificados.

Um mesmo ISP pode operar múltiplos ASs, cada um associado ao seu próprio bloco de endereços IP e às respectivas políticas de roteamento. A Figura 5 detalha essa relação para os quinze provedores mais representativos, apresentando, de forma empilhada, a contribuição de cada AS para o total de IPs correspondente a cada ISP. Os ASs foram ordenados em ordem decrescente de volume dentro de cada provedor, enquanto aqueles que excedem a terceira posição foram agrupados na categoria *outros*.

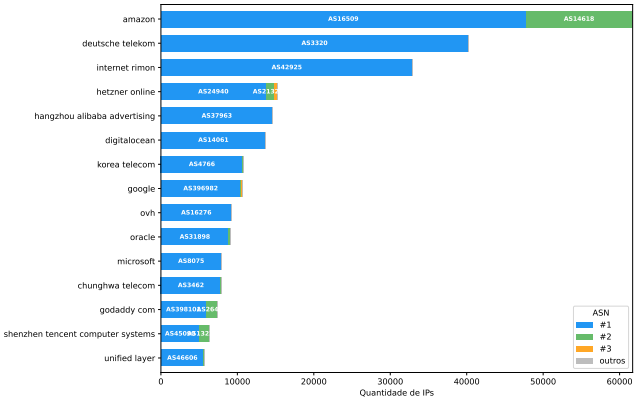

Figura 5: Distribuição de endereços IP por AS dentro dos quinze principais ISPs. Cada segmento é anotado com o número do AS quando o volume é suficiente para a exibição.

O gráfico evidencia dois padrões distintos de organização. Provedores como Deutsche Telekom, Internet Rimon e DigitalOcean

concentram toda a sua presença em um único AS, sugerindo uma infraestrutura de roteamento centralizada. Em contraste, a Korea Telecom distribui seus 10.774 endereços entre 19 ASs distintos, enquanto a Oracle opera por meio de 9 ASs, padrão típico de organizações que adquiriram infraestrutura de terceiros ao longo do tempo ou que mantêm divisões regionais autônomas. No caso da Amazon, o AS16509 responde por aproximadamente 77% dos endereços identificados, ao passo que o AS14618 contribui com os 23% restantes, refletindo a separação histórica entre as regiões da AWS nos Estados Unidos.

4.5 Limitações

A análise realizada possui algumas limitações que devem ser consideradas, tais como:

- A impossibilidade de reprodução exata dos resultados, em razão da natureza dinâmica do Shodan, que indexa dispositivos sujeitos a desligamento ou à reatribuição de endereços IP a qualquer momento;
- A identificação dos serviços baseia-se em *banner grabbing* sobre um conjunto fixo de portas, estando sujeita a falsos negativos em nós com portas não convencionais ou *banners* personalizados, bem como a eventuais falsos positivos;
- Nem todos os MOMs puderam ser identificados individualmente, pois, em parte dos *hosts*, o *banner* do serviço não revelava a implementação específica do *broker* MQTT. Esses casos foram agregados à categoria "MOMs MQTT", permitindo sua contabilização sem a atribuição do produto subjacente;
- A geolocalização por país depende de bases de dados *WHOIS*, que podem estar desatualizadas, e a contagem de nós via EPMD contempla apenas *hosts* com a porta 4369 publicamente acessível, o que implica que os números reportados representam um limite inferior da presença real de nós Erlang na Internet.

5 Conclusões

Este trabalho apresentou um levantamento inédito da presença de nós Erlang expostos na Internet, utilizando dados coletados por meio da API do Shodan e técnicas de *banner grabbing* sobre 681.326 endereços IP únicos. Dos *hosts* que responderam, 58.374 foram confirmados como nós Erlang, o que permitiu traçar um panorama da distribuição geográfica, dos serviços predominantes e da exposição do EPMD em escala global.

Em relação às perguntas de pesquisa, os resultados revelaram que a China concentra a maior parcela de nós (30,79%), seguida pelos Estados Unidos (20,83%) e pela Alemanha (10,07%). O RabbitMQ mostrou-se o serviço dominante, respondendo por 60,10% das ocorrências, o que evidencia o papel central da BEAM na infraestrutura de mensageria moderna. No que diz respeito à segurança, 40,09% dos *hosts* identificados possuíam a porta do EPMD publicamente acessível, o que expõe 37.589 nós Erlang a consultas externas sem autenticação, configurando um dado preocupante e reforçando a necessidade de políticas de proteção mais rigorosas para serviços baseados na BEAM.

A análise dos provedores de infraestrutura revelou que a maior parte dos nós Erlang expostos está implantada em ambientes de nuvem ou servidores dedicados. Provedores como Amazon, Hetzner, DigitalOcean, Oracle, Google, Microsoft e GoDaddy respondem conjuntamente por mais de 34% dos endereços identificados, com destaque para a Amazon, líder isolada com 11,33% do total. Essa concentração em infraestrutura de nuvem sugere que a exposição do EPMD não é um fenômeno restrito a ambientes mal configurados em redes residenciais ou corporativas, mas está presente em implantações profissionais de larga escala, ampliando o alcance e a relevância dos riscos identificados.

Os resultados obtidos contribuem tanto para a comunidade técnica, ao oferecer um mapeamento até então inexistente na literatura, quanto para a área de segurança, ao quantificar a superfície de ataque associada à exposição do EPMD. Espera-se que este trabalho incentive a adoção de boas práticas, tais como a restrição do acesso à porta 4369 por meio de *firewalls*, o uso de TLS (*Transport Layer Security*) na comunicação entre nós e a configuração adequada de *cookies* de autenticação em ambientes distribuídos Erlang.

5.1 Trabalhos Futuros

A principal limitação deste trabalho reside no caráter pontual da coleta, que oferece apenas um retrato estático da exposição de nós Erlang na Internet. Nesse sentido, o desenvolvimento de uma ferramenta de monitoramento contínuo de nós BEAM expostos, capaz de realizar varreduras periódicas, permitiria identificar tendências de crescimento ou redução, detectar novos nós expostos em tempo hábil e acompanhar a efetividade das medidas de mitigação adotadas pela comunidade.

Outro desdobramento relevante seria a expansão do escopo de identificação para além do Erlang, uma vez que a máquina virtual BEAM sustenta um ecossistema crescente de linguagens, como Elixir e Gleam, cada uma com frameworks e serviços próprios. A inclusão de assinaturas específicas dessas tecnologias, como o Phoenix Framework para Elixir ou o framework Wisp para Gleam, nas heurísticas de *banner grabbing* possibilitaria um mapeamento mais completo e fiel do ecossistema BEAM em produção.

Do ponto de vista de segurança, também permanece em aberto a investigação mais aprofundada das implicações da exposição identificada, por exemplo, avaliando quantos dos nós com EPMD exposto compartilham cookies de autenticação padrão ou previsíveis, o que representaria um risco ainda mais severo do que a simples exposição da porta 4369.

Adicionalmente, a correlação dos dados obtidos com bases de vulnerabilidades conhecidas, como o CVE (Common Vulnerabilities and Exposures), permitiria estimar com maior precisão o grau de risco efetivo das instâncias identificadas.

Disponibilidade de Artefatos

Os artefatos produzidos durante o desenvolvimento deste trabalho estão publicamente disponíveis no repositório do GitHub[5].

[5] https://github.com/fernandoareias/erlang_nodes_discovery

Referências

[1] Alexey Shchepin. 2003. ejabberd XMPP Server with MQTT Broker & SIP Service. https://www.ejabberd.im/. Acessado em 7 de abril de 2026.

[2] Apache Software Foundation. 2005. Apache CouchDB: Seamless multi-primary syncing database with an intuitive HTTP/JSON API, designed for reliability. https://github.com/apache/couchdb. Acessado em 7 de abril de 2026.

[3] Joe Armstrong. 2003. *Making reliable distributed systems in the presence of software errors.* Ph. D. Dissertation. Royal Institute of Technology, KTH, Stockholm, Sweden.

[4] Joe Armstrong. 2007. A history of Erlang. In *Proceedings of the Third ACM SIGPLAN Conference on History of Programming Languages* (San Diego, California) *(HOPL III)*. Association for Computing Machinery, New York, NY, USA, 6–1–6–26. doi:10.1145/1238844.1238850

[5] Inês Carvalho, Filipe Sá, and Jorge Bernardino. 2023. Performance Evaluation of NoSQL Document Databases: Couchbase, CouchDB, and MongoDB. *Algorithms* 16, 2 (2023). doi:10.3390/a16020078

[6] EMQ Technologies. 2013. EMQX: The Unified MQTT Platform for AI and IoT Data Streaming. https://www.emqx.com/en. Acessado em 7 de abril de 2026.

[7] Erlang Solutions. [s.d.]. MongooseIM: Robust, scalable and efficient XMPP server. https://github.com/esl/MongooseIM. Acessado em 7 de abril de 2026.

[8] Erlang Solutions Team. 2018. 20 Years of Open Source Erlang: Open-Erlang Interview with Anton Lavrik from WhatsApp. Erlang Solutions Blog. https://www.erlang-solutions.com/blog/20-years-of-open-source-erlang-openerlang-interview-with-anton-lavrik-from-whatsapp/ Acessado em: 31 mar. 2026.

[9] Loïc Hoguin. [s.d.]. Cowboy: Small, fast, modern HTTP server for Erlang/OTP. https://github.com/ninenines/cowboy. Acessado em 7 de abril de 2026.

[10] Gukang Hong et al. 2017. Study on Collecting Server Information through Banner Grabbing. *Journal of the Korea Institute of Information Security & Cryptology* 27, 6 (2017), 1317–1330. doi:10.13089/JKIISC.2017.27.6.1317 Acessado em 7 de abril de 2026.

[11] Keshav Kaushik, Ishika Punhani, Smita Sharma, and Mamta Martolia. 2022. An Advanced Approach for performing Cyber Fraud using Banner Grabbing. In *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*. 298–302. doi:10.1109/IC3I56241.2022.10072445

[12] Andreas Lindberg, Simon Merle, and Paul Stritzinger. 2019. Scaling Erlang Distribution: Going Beyond the Fully Connected Mesh. In *Proceedings of the 18th ACM SIGPLAN International Workshop on Erlang (Erlang 2019)*. Berlin, Germany, 48–55. doi:10.1145/3331542.3342572

[13] Mochi Media. [s.d.]. MochiWeb: An Erlang library for building lightweight HTTP servers. https://github.com/mochi/mochiweb. Acessado em 7 de abril de 2026.

[14] Riak. [n. d.]. Riak: Enterprise NoSQL Database – Scalable Database Solutions. https://riak.com/index.html. Acessado em 7 de abril de 2026.

[15] Shodan. [n. d.]. On-Demand Scanning. https://help.shodan.io/the-basics/on-demand-scanning. Acessado em 7 de abril de 2026.

[16] TheirStack. 2021. Empresas que usam CouchDB. https://theirstack.com/pt/technology/couchdb. Acessado em 7 de abril de 2026.