

# Estatísticas de 40k+ sites do Ecossistema HTTPS no Brasil

Débora Patrícia Ströher<sup>1</sup>, Rodrigo Brandão Mansilha<sup>1</sup>, Diego Kreutz<sup>1</sup>

<sup>1</sup> Universidade Federal do Pampa (Unipampa)

deborastroher.aluno@unipampa.edu.br

{rodrigomansilha, diegokreutz}@unipampa.edu.br

***Resumo.** Estudos recentes demonstram que apenas utilizar HTTPS não garante, necessariamente, que os usuários irão navegar de forma segura no site. Neste trabalho apresentamos estatísticas de 40.406 sites HTTPS dos blocos de IPs do Brasil. Os resultados apontam que apenas 11,01% dos sites analisados suportam a versão TLS 1.3, a mais atual e a única sem vulnerabilidades conhecidas. Além disso, 9,26% utilizam certificados autoassinados, impossibilitando que um navegador o reconheça como confiável.*

## 1. Introdução

O protocolo HTTPS (*Hyper Text Transfer Protocol Secure*) é uma combinação do HTTP (*Hyper Text Transfer Protocol*) com o TLS (*Transport Layer Security*). O “S” do protocolo significa que ele oferece propriedades básicas de segurança (e.g., confidencialidade, integridade, autenticidade) às comunicações entre o cliente e o servidor web.

Estudos indicam recorrentemente que o fato de utilizar HTTPS não significa que um site oferece segurança à navegação dos usuários. Por exemplo, um estudo realizado em 2013 sobre um milhão dos sites mais populares de acordo com a classificação de Alexa Top [Vratonjic et al., 2013] constatou que apenas 16% dos sites que implementam HTTPS utilizam certificados válidos e implantados adequadamente nos respectivos domínios.

Em 2019 foi publicado um estudo sobre mais de 5 milhões de sites da China [Huang et al., 2019], no qual os autores constataram que 66,45% dos servidores web suportam apenas HTTP. Os autores identificaram também que mais de 40% dos certificados utilizam algoritmos de resumo criptográficos não recomendados (e.g., SHA1, MD5). Além disso, constataram que 49,19% dos servidores nas 10 principais regiões do país é vulnerável ao ataque Logjam (i.e., ainda suportam a versão 1.2 do TLS).

No Brasil, um estudo publicado em 2020 [Fiorenza et al., 2020] realizou um levantamento inicial de 5.510 sites HTTPS de governos federal, estadual e municipal, comércio eletrônico e instituições financeiras. Os resultados indicam que apenas 30% dos sites suportam a versão 1.3 do TLS, mais de 92% suportam a versão 1.2 do TLS e mais de 80% ainda suportam versões fortemente desaconselhadas do SSL/TLS (e.g., TLS 1.1). O estudo também identificou que 100% dos sites oferecem riscos de segurança aos seus usuários, isto é, suportam versões vulneráveis dos protocolos SSL/TLS.

O nosso objetivo é ampliar o estudo do ecossistema HTTPS do Brasil, utilizando uma amostra estatísticas mais representativa de sites. Para atingir o objetivo, optamos por realizar uma varredura dos blocos de IPs do Brasil, identificando sites que operam com HTTPS. A partir de uma primeira varredura, identificamos 40.406 sites HTTPS<sup>1</sup>.

---

<sup>1</sup>O número de sites não foi maior devido a diversos problemas de bloqueio das varreduras por parte de

Como contribuições deste trabalho podemos destacar: (a) um levantamento detalhado sobre ferramentas livremente disponíveis para análise de sites HTTPS; (b) uma análise mais abrangente do ecossistema HTTPS do Brasil; (c) identificação de estatísticas que apontam para um cenário bastante preocupante (e.g., apenas 11,01% dos sites suportam a versão 1.3 do TLS).

O restante deste trabalho está organizado como segue. Na Seção 2 apresentamos alguns conceitos relacionados aos certificados de domínio. Na Seção 3 introduzimos as principais ferramentas disponíveis livremente para a análise de sites HTTPS. Nas seções 4 e 5 descrevemos a metodologia adotada no trabalho e os resultados, respectivamente. Finalmente, apresentamos as considerações finais na Seção 6.

## 2. Certificados de Domínio

Os certificados de domínio permitem estabelecer conexões seguras (e.g., através de HTTPS) entre os clientes (e.g., navegadores) e os servidores (e.g., Web). Os certificados de domínio são emitidos por uma Autoridade Certificadora (AC), que é uma entidade da Infraestrutura de Chave Pública (ICP). As ICPs foram criadas com o objetivo de oferecer serviços (e.g., emissão e revogação de certificados) que permitem garantir propriedades de segurança aos envolvidos na comunicação, como autenticação e confidencialidade dos dados [IBM Corporation, 2020]. Essas propriedades são asseguradas a partir dos certificados de domínio no ecossistema HTTPS. Na prática, um navegador consegue verificar a autenticidade e confiabilidade de um site HTTPS através do seu certificado de domínio.

Na ICP, as ACs são responsáveis pela emissão, validação e revogação de certificados de domínio. Um certificado desse tipo contém informações sobre o nome do domínio (site), chave pública associada e proprietário, AC emissora e validade do certificado. Os certificados representam a base da segurança do ecossistema HTTPS.

Os sistemas operacionais e navegadores possuem uma lista interna de assinaturas confiáveis, isto é, certificados raiz. Para verificar um certificado de um site, o navegador analisa o caminho de certificação, que é uma sequência de certificados conectando a raiz da AC ao certificado do servidor [Fu et al., 2018]. Certificados ditos como confiáveis possuem uma validade, são emitidos por uma AC reconhecida e possuem AC raiz.

Problemas de segurança associados aos certificados de domínio incluem validade (e.g., certificado expirado), nome de domínio (e.g., DNS do site diferente do nome contido no certificado), e AC emissora do certificado. Um exemplo comum relacionado ao último caso é o de certificados autoassinados, ou seja, que não possuem assinatura de uma AC conhecida e acreditada. Os certificados autoassinados levam tipicamente a custos operacionais (e.g., gestão da ICP e ACs locais) e riscos de segurança (e.g., não reconhecimento nativo pelos navegadores, facilitando a atuação de usuários maliciosos) maiores e, por isso, devem ser evitados [Kappenberger, 2012].

## 3. Ferramentas

Há diversas ferramentas de análise de sites HTTPS. Neste trabalho levamos em consideração apenas aquelas livremente disponíveis e que permitem identificar

---

provedores de Internet. Chegamos a receber notificações do CAIS da RNP (<https://www.rnp.br/en/sistema-rnp/cais>).

**Tabela 1. Ferramentas de análise de sites HTTPS**

Ferramenta	Modo de operação		Informações do certificado				Protocolo		Classificação
	Navegador	Terminal	Emissor	Validade	Domínio	Cadeia do certificado	Versão	Vulnerabilidades	Possui
SSL Labs	x		x	x	x	x	x	x	x
ImmuniWeb	x		x	x	x	x	x	x	x
Digicert	x		x	x	x	x	x	x	
Observatory	x		x	x	x		x		x
SSL Checker	x		x	x	x				
Wormly	x			x	x		x		x
pentest-tools	x						x	x	x
CryptCheck	x						x		x
Geekflare TLS Scanner	x						x		
Cipherscan		x					x		
TestSSLServer		x	x				x		
SSLyze		x					x	x	
OpenSSL		x	x			x	x		
testssl.sh		x	x	x	x	x	x	x	
<b>Extensão Navegador</b>									
IndicateTLS	x		x	x	x		x		
Certainly Something	x		x	x	x		x		
Certificate Pinner	x			x					

informações dos sites, como dados sobre os certificados, versões e vulnerabilidades dos protocolos. Na Tabela 1 apresentamos um resumo das ferramentas selecionadas e suas respectivas funcionalidades, agrupadas em categorias.

Para usuários leigos, ferramentas como SSL Labs, SSL Checker, ImmuniWeb e Digicert podem ser mais indicadas. Elas são relativamente simples de serem utilizadas, pois funcionam pelo navegador e possuem uma interface mais amigável. Por exemplo, algumas dessas ferramentas trazem uma classificação do site intuitiva (e.g., A, B, C, D) que facilita o entendimento e a comparação. Para usuários avançados que necessitam analisar um conjunto grande de sites, ferramentas como Cipherscan, TestSSLServer, SSLyze e testssl.sh são mais recomendadas.

Ferramentas que extraem informações do certificado identificam tipicamente a AC emissora, a validade e o domínio do certificado. Das ferramentas analisadas, apenas SSL Labs, SSL Checker, ImmuniWeb, Digicert, Observatory e testssl.sh recuperam informações detalhadas sobre os certificados. Algumas das ferramentas identificam também a cadeia do certificado, permitindo classificá-los como confiável ou não.

Existem também ferramentas que funcionam como extensões de navegador, como é o caso da IndicateTLS, Certainly Something e Certificate Pinner. Dependendo da finalidade (e.g., verificar dados básicos dos sites acessados), elas podem ser suficientes e mais indicadas, pois funcionam de forma simples e integradas ao navegador.

#### 4. Metodologia

A Figura 1 resume o processo adotado para a coleta e análise dos dados. Na primeira etapa foram coletados os dados referentes aos blocos de IPs do Brasil, utilizando a lista de 2.286 blocos de IPs da NirSoft (<https://www.nirsoft.net/countryip/br.html>). A partir dos blocos geramos as sequências de mais de 77 milhões de IPs.

Na segunda etapa realizamos a varredura dos IPs em busca de conexões HTTP ou HTTPS. Para identificar portas com HTTP ou HTTPS habilitado, utilizamos a ferramenta `wget` (<https://www.gnu.org/software/wget/>) do Linux. Ao final dessa etapa, identificamos mais de 40 mil sites com HTTPS habilitado. É importante des-



**Figura 1. Processo de coleta e análise dos dados**

tacar que diversos provedores de Internet acabaram por bloquear nossas varreduras, pois interpretaram as requisições aos IPs dos respectivos blocos como varreduras pré-ataque. Sem os bloqueios certamente aumentaríamos o número de IPs identificados com HTTPS habilitado. Um dos nossos objetivos futuros é executar novas varreduras utilizando estratégias e ferramentas para evitar os bloqueios.

Na terceira etapa realizamos o teste da conexão HTTPS dos IPs identificados na etapa anterior. Para automatizar a análise, utilizamos a ferramenta `testssl.sh` (<https://testssl.sh/>), que identifica as versões dos protocolos SSL/TLS suportadas pelos sites, bem como extrai informações detalhadas sobre os certificados do domínio.

## 5. Resultados

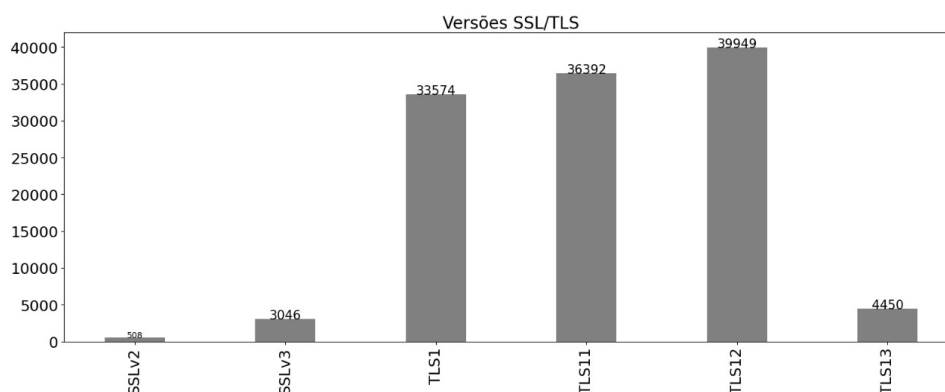
As saídas da ferramenta `testssl.sh`, para os 40.406 sites, estão disponíveis no GitHub (<https://github.com/HTTPS-TLS-BR/WRSeg21>). Para cada endereço IP há um arquivo contendo os dados completos de saída dos testes da ferramenta, incluindo informações associadas aos certificados, algoritmos de assinatura, tamanhos de chaves, versões e vulnerabilidades associadas aos protocolos.

### 5.1. Versões dos Protocolos SSL/TLS

Na Figura 2 apresentamos as diferentes versões dos protocolos SSL/TLS suportadas pelos sites. A soma dos valores das versões ultrapassa o número total de sites porque um site pode suportar múltiplas versões. Como podemos observar, 90,06% dos sites ainda suportam TLS 1.1, o que é um dado preocupante devido ao fato de que essa versão do protocolo ser fortemente desaconselhada desde 2008. Ainda mais preocupante é o fato de mais de 83% dos sites ainda suportarem a versão 1.0 do TLS, que deveria estar em desuso desde 2006. Isso significa que uma grande parcela dos sites HTTPS ainda continua colocando usuários em risco através de ataques conhecidos e bem documentados na Internet, como BEAST, POODLE e Logjam.

Outro aspecto preocupante é o baixíssimo número de sites que suportam o TLS 1.3 (apenas 11,01%). Desde 2018, utilizar TLS 1.3 é o método mais eficaz para mitigar diferentes vulnerabilidades e ataques de versões anteriores do protocolo. Entretanto, a maioria absoluta dos servidores Web ainda não suporta essa versão do protocolo. Consequentemente, mesmo que os navegadores dos usuários estejam atualizados, não será possível estabelecer uma conexão TLS 1.3 com os sites HTTPS.

Na Tabela 2 apresentamos os resultados de [Fiorenza et al., 2020] (5.510 sites HTTPS) e os nossos resultados para os 40.406 sites HTTPS. O dado que mais chama atenção é a porcentagem de sites que suportam o TLS 1.3. Num escopo mais limitado, chegou a 31,83% dos sites, entretanto, em uma análise mais abrangente (e.g., incluindo sites de diversos setores, como hotelaria e sustentabilidade), esse número reduziu para



**Figura 2. Versões dos Protocolos SSL/TLS**

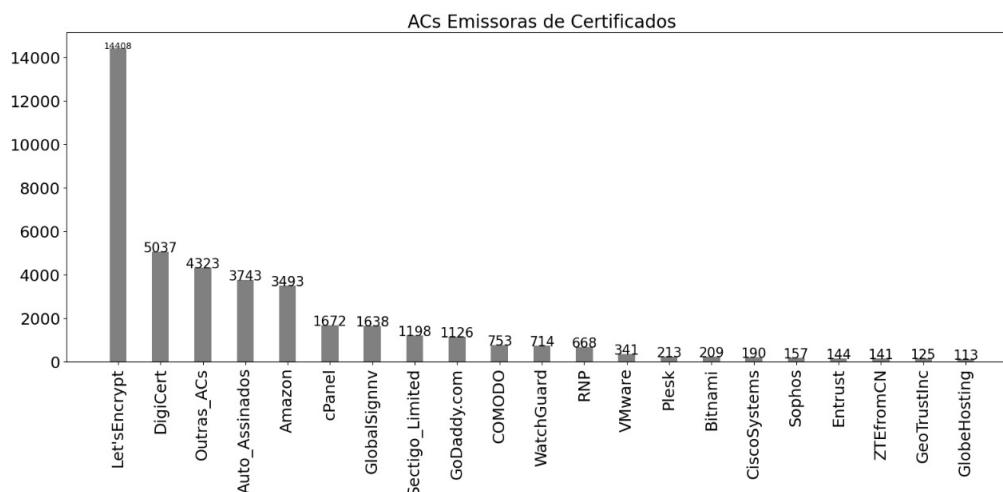
apenas 11,01%. Isto significa que o cenário é ainda mais preocupante do que o imaginado anteriormente.

**Tabela 2. Resultados em perspectiva**

Quantidade de sites	Ano	SSLv2	SSLv3	TLS 1	TLS 1.1	TLS 1.2	TLS 1.3
5.510	2020	1,92%	5,26%	76,17%	80,09%	97,35%	31,83%
40.406	2021	1,25%	7,53%	83,09%	90,06%	98,86%	11,01%

## 5.2. Emissores dos certificados

Na análise dos dados, identificamos mais de 4.300 ACs emissoras dos certificados. Apresentamos as 18 ACs mais frequentemente utilizadas, o número de certificados autoassinados e o número total de outras ACs na Figura 3.



**Figura 3. ACs Emissoras de Certificados**

A Let's Encrypt é a AC mais utilizada, representando uma fatia de 35,65% dos sites analisados. A segunda colocada é a AC comercial DigiCert, responsável pelos certificados de 12,46% dos sites. Como podemos perceber, os certificados autoassinados aparecem terceiro lugar, perfazendo 9,26% do total, chegando muito próximo ao somatório

de todas as outras ACs (terceira barra do gráfico) não representadas no gráfico. Esses números indicam que há uma tendência crescente por ACs gratuitas, como é o caso da Let's Encrypt. Mesmo com certificados que precisam ser renovados a cada 3 meses, a utilização desta AC vem claramente ganhando espaço no mercado.

## 6. Considerações Finais

Realizamos uma análise de 40.406 sites com o objetivo de diagnosticar a saúde do ecossistema HTTPS do Brasil. As descobertas indicam que mais de 98% dos sites ainda suporta versões inferiores a 1.3 do TLS, o que pode colocar em risco os usuários que acessam esses sites. Além disso, apenas 11,01% dos sites suporta a versão 1.3 do TLS, que desde 2018 é reconhecida como a única sem vulnerabilidades identificadas e imune aos ataques conhecidos (e.g., DROWN, POODLE e BEAST) entre as versões do TLS.

Os resultados indicam a necessidade de avanços quanto ao suporte de diferentes versões do TLS em sites HTTPS no Brasil. Apesar de a maioria dos navegadores já suportarem o TLS 1.3 há algum tempo, a maioria absoluta dos sites HTTPS ainda não suporta essa versão do protocolo.

**Limitações.** A varredura dos IPs incorreu em diversos bloqueios por parte dos provedores de Internet. Uma das formas de mitigar esse problema é implementar algum grau de aleatoriedade na distribuição da carga de trabalho das faixas de endereços IPs (de forma a desagregar sequências de endereços IPs de um mesmo provedor, por exemplo). Complementarmente, utilizar ferramentas que permitem uma varredura menos intrusiva e mais difícil de ser detectada, como o `nmap`<sup>2</sup> (<https://nmap.org>).

**Trabalhos futuros:** (a) aumentar a quantidade de sites analisados; (b) classificar os sites de acordo com setores da economia, buscando identificar eventuais peculiaridades de setores específicos; (c) notificar os sites sobre as vulnerabilidades encontradas; e (d) iniciar campanhas de boas práticas na utilização de HTTPS nos sites.

## Referências

- Fiorenza, M. M., Kreutz, D., Escarrone, T., and Temp, D. (2020). Uma análise da utilização de HTTPS no brasil. In *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 966–979. SBC.
- Fu, P., Li, Z., Xiong, G., Cao, Z., and Kang, C. (2018). SSL/TLS security exploration through X.509 certificate's life cycle measurement. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 00652–00655.
- Huang, J., Zhang, Z., Li, W., and Xin, Y. (2019). Assessment of the impacts of TLS vulnerabilities in the HTTPS ecosystem of china. *Procedia computer science*, 147:512–518.
- IBM Corporation (2020). Infraestrutura da chave pública (pki). <https://www.ibm.com/docs/pt-br/ibm-mq/8.0?topic=ssfksj-8-0-0-com-ibm-mq-sec-doc-q009900--htm>.
- Kappenberger, R. (2012). The true cost of self-signed SSL certificates. *Computer Fraud & Security*, 2012(9):14–16.
- Vratonjic, N., Freudiger, J., Bindschaedler, V., and Hubaux, J.-P. (2013). The inconvenient truth about web certificates. In *Economics of information security and privacy iii*, pages 79–117. Springer.

---

<sup>2</sup>Ferramentas como o `nmap` permitem definir padrões de temporização (e.g., `-T1 sneaky`) que dificultam o trabalho dos sistemas de detecção.