Uma Análise da Utilização de HTTPS no Brasil

Maurício M. Fiorenza[2,3], Diego Kreutz[1,2,3], Thiago Escarrone[1,3], Daniel Temp[2,3,4]

[1]Laboratório de Estudos Avançados (LEA)
[2] Mestrado Profissional em Engenharia de Software (MPES)
[3] Universidade Federal do Pampa (UNIPAMPA)
[4] Instituto Federal Farroupilha (IFFar)
{Nome.Sobrenome}@gmail.com[1,4], {Nome.Sobrenome}@unipampa.edu.br[2,3]

Abstract. *HTTPS is essential to grant the security of communications using the HTTP protocol on the Internet. However, although the growing adoption of HTTPS, many sites still don't implement digital certificates correctly and do not support version 1.3 of TLS. In this work we present an analysis of the use of HTTPS in Brazil. We have analyzed 5806 web sites of public and private institutions, including sites from the three governmental spheres, financial institutions, electronic commerce, and hundreds of IPs addresses from the Censys database, which includes addresses of telecom operators and Internet service providers. Our findings show that the majority of the analyzed sites use or support old versions of TLS/SSL, which contain known vulnerabilities and, thus, can be exploited by malicious agents. Only 30% of the web sites support TLS 1.3.*

Resumo. *O HTTPS é essencial para garantir a segurança das comunicações que utilizam o protocolo HTTP na Internet. Entretanto, apesar da crescente adoção do HTTPS, muitos sites ainda não implementam da maneira correta os certificados digitais e não suportam a versão 1.3 do TLS. Este trabalho apresenta uma análise da utilização do HTTPS no Brasil. A análise apresentada neste paper inclui 5806 sites de instituições públicas e privadas, incluindo sites das três esferas governamentais, instituições financeiras, comércio eletrônico, além de 994 endereços IP da base de dados do Censys, que englobam endereços de operadoras e provedores de serviços de Internet. Os resultados demonstram que a maioria dos serviços analisados utilizam ou suportam versões antigas do TLS/SSL, que contém vulnerabilidades conhecidas e passíveis de exploração por agentes maliciosos. Apenas 30% dos sites analisados suportam a versão 1.3 do TLS.*

1. Introdução

A maioria dos usuários acredita que está utilizando um canal de comunicação seguro quando navega na Internet utilizando o HTTPS (*Hyper Text Transfer Protocol Secure*), isto é, HTTP sobre SSL (*Secure Sockets Layer*)/TLS (*Transport Layer Security*). Na verdade, os próprios navegadores contribuem para esta sensação de segurança quando apresentam um cadeado verde ou fechado ao lado do endereço eletrônico, mais conhecido como URL (*Uniform Resource Locator*), que o usuário está acessando. Entretanto, pesquisas demonstram que muitas vezes o ecossistema do HTTPS é, diferentemente do imaginado (ou do senso comum), inseguro [Bokslag 2016, Frost et al. 2019, Samarasinghe and Mannan 2019]. Os desafios e problemas de segurança são muitos e podem

ocorrer em diferentes partes do ecossistema HTTPS, incluindo falhas na especificação ou implementação dos protocolos, falhas na configuração dos certificados digitais nos servidores Web, falhas na geração dos certificados digitais, vulnerabilidades nas ICPs (Infraestruturas de Chaves Públicas), entre outras vulnerabilidades [Bokslag 2016, Frost et al. 2019, Samarasinghe and Mannan 2019, Matsumoto and Reischuk 2015, Merzdovnik et al. 2016].

Quando o navegador inicia uma conexão HTTPS, é realizado o *handshake* do TLS entre navegador e o servidor Web [Rescorla and Dierks 2008]. Durante o *handshake*, o servidor apresenta o seu certificado digital X.509[1], que é utilizado para atestar que uma organização é realmente quem ela se diz ser através de uma chave pública.

Um certificado é atestado e emitido por uma AC (Autoridade Certificadora)[2]. As ACs fazem parte das ICPs [Durumeric et al. 2013]. No Brasil, há infraestruturas de chaves públicas mantidas por instituições como o SERPRO (`https://www.serpro.gov.br/`) e a RNP (`https://www.rnp.br`).

Considerando um serviço online que utiliza HTTPS, como é possível obter mais informações sobre a segurança do certificado digital e das futuras conexões com o site? Que versões do TLS/SSL o site suporta? As versões suportadas pelo site possuem vulnerabilidades conhecidas? Existem estudos que tentam responder a estes tipos de questões em contextos específicos, como sites da China, sites do Alexa Top 1 milhão e aplicativos bancários no Reino Unido [Samarasinghe and Mannan 2019, Vratonjic et al. 2013, Huang et al. 2019, Chothia et al. 2017]. Resultados das pesquisas apontam que ainda há um número significativo de sites e sistemas com vulnerabilidades relacionadas aos certificados digitais e às versões de protocolos suportados e utilizados na prática.

O presente estudo tem dois objetivos principais: (a) identificar ferramentas disponíveis livremente que permitam a análise dos certificados e protocolos utilizados pelo HTTPS; e (b) realizar um estudo abrangente da utilização do HTTPS no Brasil. Diferentemente dos trabalhos relacionados citados anteriormente, neste estudo são utilizadas ferramentas que proporcionam um diagnóstico mais detalhado, como a `testssl.sh`, que permitem identificar e catalogar também falhas dos protocolos e dos certificados utilizados pelos sites.

Este trabalho é uma evolução de um paper curto publicado no WRSeg 2019 [Escarrone et al. 2019], onde o escopo da análise foi limitado a apenas 44 sites. As principais contribuições deste trabalho podem ser resumidas em: (a) uma comparação e discussão sobre as principais ferramentas de análise de sites HTTPS disponíveis gratuitamente na Internet; (b) a análise de um número significativo e representativo de sites, cobrindo, por exemplo, todas as instituições financeiras filiadas a Federação Brasileira de Bancos (Febraban), um grupo de 4458 sites de comércio eletrônico ligados a Associação Brasileira de Comércio Eletrônico (ABCOMM), além de um conjunto com 994 IPs brasileiros disponível na base de dados do Censys [Durumeric et al. 2015] (`https://censys.io/`); e (c) uma discussão sobre ataques e desafios do ecossistema HTTPS.

O restante do trabalho está organizado como segue. Na seção 2 são apresentadas as ferramentas de análise de ecossistemas HTTPS. Os resultados das análises realizadas

[1] `https://tools.ietf.org/html/rfc2459`
[2] `https://tools.ietf.org/html/rfc5280.html`

sobre os 5806 sites estão disponíveis na seção 3. Finalmente, a seção 4 apresenta uma discussão sobre alguns dos ataques e desafios do ecossistema HTTPS.

2. Ferramentas de análise de sites HTTPS

Existe um variado número de ferramentas projetadas especificamente para analisar problemas na instalação de certificados digitais em sites HTTPS e detectar falhas de protocolos e na emissão dos certificados. Ferramentas como a SSL Checker[3] e Observatory[4], apresentam informações sucintas sobre os certificados, como a AC que emitiu e a validade do certificado. Por outro lado, ferramentas como a SSL Labs[5] e `testssl.sh`[6], realizam uma análise mais detalhada dos certificados e dos servidores, apresentando também as versões dos protocolos suportados e as vulnerabilidades conhecidas.

A Tabela 1 resume as principais características identificadas para cada umas das dez ferramentas analisadas. A categoria **Modo de operação** é relativa à interface utilizada para a execução das análises. Existem ferramentas que podem ser utilizadas via Web, diretamente no navegador, que tendem a ser mais simples de utilizar. Entretanto, essas ferramentas limitam o processo de análise pelo fato de permitir apenas testes manuais, isto é, um site por vez. Outras ferramentas, como a `testssl.sh`, são operadas via terminal, o que aumenta o grau de dificuldade de utilização para usuários leigos. Por outro lado, esse tipo de ferramenta permite a criação e execução de testes em lote, isto é, para centenas ou até milhares de sites sem a necessidade de intervenção manual.

Tabela 1. Ferramentas de análise de sites HTTPS

	Modo de operação		**Informações do certificado**			**Protocolo**		
Ferramenta	Navegador	Terminal	Emissor	Validade	Domínio	Versão	Vulnerabilidades	Cadeia do certificado
SSL Labs [5]	✓		✓	✓	✓	✓	✓	✓
SSL Checker [3]	✓		✓	✓	✓			
ImmuniWeb[7]	✓		✓	✓	✓	✓	✓	✓
Digicert[8]	✓		✓	✓	✓	✓	✓	✓
Wormly[9]	✓			✓	✓	✓		
Geekflare TLS Scanner[10]	✓					✓		
CryptCheck[11]	✓					✓		
Observatory[4]	✓		✓	✓	✓	✓		
Cipherscan[12]		✓				✓		
testssl.sh[6]		✓	✓	✓	✓	✓	✓	✓

A categoria **Informações do certificado** apresenta as informações do certificado identificadas por cada ferramenta, onde constam a AC emissora, validade e os domínios ao qual o certificado pertence. Neste cenário destacam-se negativamente as ferramentas

[3] `https://www.sslshopper.com/ssl-checker.html`
[4] `https://observatory.mozilla.org/`
[5] `https://www.ssllabs.com/ssltest`
[6] `https://testssl.sh`
[7] `https://www.immuniweb.com/ssl/`
[8] `https://www.digicert.com`
[9] `https://www.wormly.com/`
[10] `https://gf.dev/tls-test`
[11] `https://tls.imirhil.fr/`
[12] `https://github.com/mozilla/cipherscan`

Wormly, Geekflare, CryptCheck e Cipherscan, que apresentam de forma limitada (ou não apresentam) as informações do certificado.

Na categoria **Protocolo** são identificadas as informações como versões e vulnerabilidades conhecidas para os protocolos suportados pelos sites. Por fim, a **Cadeia do certificado** é analisada através da validade do certificado, AC emissora e AC raiz (órgão que autoriza uma AC a emitir certificados) para determinar se o certificado é confiável ou não. Nestas duas últimas categorias, apenas as ferramentas SSL Labs, ImmuniWeb, Digicert e `testssl.sh` informam as vulnerabilidades das versões do TLS/SSL suportadas pelo site e a cadeia do certificado.

Em termos de profundidade técnica das análises, as quatro ferramentas que se destacaram são SSL Labs, ImmuniWeb, Digicert e a `testssl.sh`. Estas são as únicas ferramentas que apresentam em seus relatórios informações sobre todas as categorias da Tabela 1. Essas quatro ferramentas passaram por uma segunda fase de testes, onde foram analisadas outras características, como (a) o tempo médio de execução, (b) o suporte à análise em lote de sites, (c) a atribuição de notas aos resultados, facilitando a utilização e compreensão por parte de usuários leigos, e (d) a informação do algoritmo e tamanho de chave utilizado para criptografia.

Para avaliação e comparação das quatro ferramentas, como resumido na Tabela 2, foram utilizados dez sites. Como pode ser observado, apenas a ferramenta `testssl.sh` permite a realização de testes em lotes, motivo pelo qual foi escolhida para realizar a análise dos 5806 sites selecionados. Vale ressaltar que, para análises pontuais ou utilização por parte de usuários leigos, as ferramentas SSL Labs e ImmuniWeb são as mais recomendadas. A ferramenta Digicert não apresenta atribuição de nota qualitativa ao site HTTPS.

Tabela 2. Análise das ferramentas

Ferramenta	Tempo médio de execução	Execução em lote	Atribuição de notas	Algoritmo e chave
SSL Labs	204s		✓	✓
ImmuniWeb	67s		✓	✓
Digicert	16s			✓
testssl.sh	92s	✓		✓

3. Análise do Ecossistema HTTPS no Brasil

Para as coletas de dados e análises, foram implementados *scripts*, cujos detalhes e repositório online pode ser encontrados na versão estendida do paper [Fiorenza et al. 2020]. Os *scripts* realizam a coleta dos dados online, as análise dos sites com HTTPS habilitado utilizando a ferramenta `testssl.sh` e o processamento automático dos relatórios de saída da ferramenta. A coleta de dados online (e.g., endereços IP) resultou em 5806 sites brasileiros com HTTP ou HTTPS habilitado.

Os 5806 sites foram divididos em seis conjuntos: (a) 20 sites do governo federal (e.g., presidência da república, ministérios, agências); (b) 108 sites oficiais dos 26 estados da federação mais o Distrito Federal (e.g., site principal, mais três secretarias de cada

estado); (c) 127 sites de prefeituras brasileiras (e.g., capitais dos estados, mais quatro cidades por estado escolhidas pseudo-aleatoriamente); (d) 99 sites de bancos e instituições financeiras filiados à Febraban[13]; (e) 4458 sites de *e-commerce* filiados à Associação Brasileira de Comércio Eletrônico[14]; (f) 994 endereços IP brasileiros registrados na base de dados da plataforma Censys[15].

Na primeira etapa da análise, foi identificado que 296 (5,1%) dos 5806 sites não utilizam HTTPS, sendo 49 deles portais de instituições públicas, 15 sites ligados a organizações financeiras, 74 sites de *e-commerce* e 157 endereços da base de dados do Censys. Para os 5510 sites remanescentes, foram realizadas diferentes análises, conforme discriminado a seguir.

3.1. Protocolos TLS/SSL

Nesta etapa, o objetivo foi identificar as versões dos protocolos TLS/SSL suportadas por cada site HTTPS. Diferentes versões dos protocolos são geralmente suportadas para garantir compatibilidade com navegadores mais antigos ou desatualizados, mas, por outro lado, isto pode levar a riscos de segurança desnecessários. Na Tabela 3 pode ser observado que a maioria dos sites suportam a versão 1.2 do TLS, que possui vulnerabilidades conhecidas como o Logjam e ainda é recomendada pelo *National Institute of Standards and Technology* (NIST) [McKay and Cooper 2019]. Novamente, o cenário apresentado é crítico pelo fato de grande parte dos sites (80%) ainda suportar versões fortemente desaconselhadas do TLS/SSL, como a versão 1.1 do TLS.

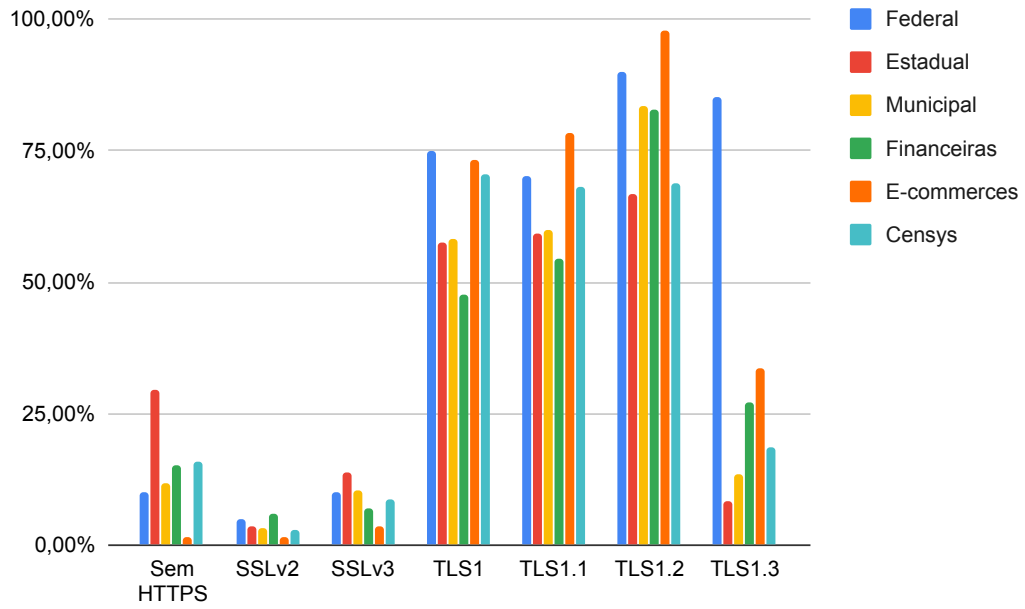
Tabela 3. Versões dos protocolos e respectivas vulnerabilidades

Versão	Sites que suportam	Ataques conhecidos
SSLv2	106	DROWN
SSLv3	290	POODLE, BEAST
TLS 1.0	4197	BEAST
TLS 1.1	4413	POODLE
TLS 1.2	5364	Logjam
TLS 1.3	1754	-

O gráfico da Figura 1 resume a utilização de diferentes versões dos protocolos por grupo analisado. Como pode ser observado, versões não recomendadas, como a SSLv2 e SSLv3, ainda continuam sendo utilizadas em todos os grupos analisados. Curiosamente, os sites de *e-commerce* apresentam o menor índice de utilização de versões SSLv2 e SSLv3, algo esperado de instituições financeiras. É interessante observar ainda que o grupo da esfera de entidades federais se destacou no suporte da versão 1.3 do TLS. Uma explicação para esta discrepância significativa no suporte ao TLS 1.3 pode ser a amostragem de sites, que foi comparativamente pequena na esfera federal quando comparada aos grupos financeiros e de *e-commerce*, por exemplo.

[13] `https://portal.febraban.org.br/pagina/3164/12/pt-br/associados`
[14] `https://abcomm.org/associados/`
[15] `https://censys.io/ipv4?q=location.country\%3A+Brazil`

Figura 1. Protocolos utilizados por grupos



Os ataques destacados na Tabela 3 exploram vulnerabilidades conhecidas nas versões dos protocolos para interceptar as informações trocadas entre cliente e servidor, comprometendo a confidencialidade e a integridade dos dados. O ataque Beast (abreviação de *Browser Exploit Against SSL/TLS*), foi descoberto em 2002 e aplicado na prática em 2011 [Duong and Rizzo 2011]. Este ataque explora uma vulnerabilidade no *cipher block chaining* (CBC) do protocolo TLS v1.0 e permite acesso a informações essenciais da conexão, como os *cookies*, permitindo que a sessão seja sequestrada pelo atacante.

A vulnerabilidade POODLE (*Padding Oracle On Downgraded Legacy Encryption*) acontece quando sistemas que utilizam diferentes versões de protocolos (e.g., SSLv3 e TLS 1.1) para fazer o *handshake* da comunicação. Esta troca de informações, para evitar problemas de interoperabilidade, é realizada através de um protocolo anterior, o SSLv3, o que possibilita o ataque através das vulnerabilidades desta versão [Möller et al. 2014].

Logjam [Kerner 2015], vulnerabilidade descoberta em 2015, pode ser utilizada para ataques específicos ao algoritmo Diffie-Hellman. Este algoritmo, imprescindível para muitos protocolos de segurança como o TLS, é utilizado na geração e compartilhamento de chaves secretas entre os participantes que pretendem estabelecer uma conexão segura [Bokslag 2016]. O ataque, que possibilita realizar o *downgrade* de conexões TLS para um nível de criptografia inseguro, permite ao atacante ler e modificar os dados transmitidos pela conexão.

Descoberto em 2016, o ataque DROWN (*Decrypting RSA with Obsolete and Weakened eNcryptation*) [Aviram et al. 2016], afeta conexões TLS/SSL baseadas em SSLv2. Com o ataque DROWN, após a descoberta da chave, é possível decifrar as mensagens trocadas entre cliente e servidor via HTTPS. Na época da descoberta, estimava-se que 33 % dos servidores WEB estavam vulneráveis. Dos sites HTTPS do Brasil analisados,

apenas 1,92% suporta SSLv2.

3.2. Análise dos certificados

Um certificado digital somente é considerado como confiável quando ele apresenta: (a) o nome da AC que o assinou; (b) o domínio ao qual pertence; e (c) a data de validade [Vratonjic et al. 2013]. Obviamente, as três informações devem estar corretas. As análises realizadas mostram que 12,83% dos sites possuem a cadeia do certificado quebrada, impedindo que navegador reconheça o site como confiável, e causando mensagens de riscos de segurança para o usuário ao acessar o site.

A cadeia do certificado também é quebrada quando a própria empresa assina o certificado, caracterizando-o como auto-assinado. Em 5,49% dos sites foram identificados certificados que são assinados pelo próprio órgão, o que também impede os navegadores de validar de forma automática e transparente a autenticidade do certificado.

Quando o nome do domínio do certificado difere do nome do domínio do site, o certificado também é identificado como não-confiável. Este caso ocorre em 9,5% dos certificados analisados. Já a data de validade do certificado é apresentada da seguinte forma: *Not before and Not after*. Se for identificada qualquer data fora deste intervalo explícito, o certificado é considerado como expirado, como foi o caso de 4,22% dos certificados analisados.

Em resumo, 18% dos sites analisados apresentam algum problema relacionado ao certificado digital, seja relacionado à utilização incorreta ou má implementação. Isto demonstra a necessidade de revisão e atualização dos certificados digitais das instituições, com especial atenção para novas leis como a Lei Geral de Proteção de Dados Pessoais (LGPD), nº 13.709[16], que entrará em vigor em breve.

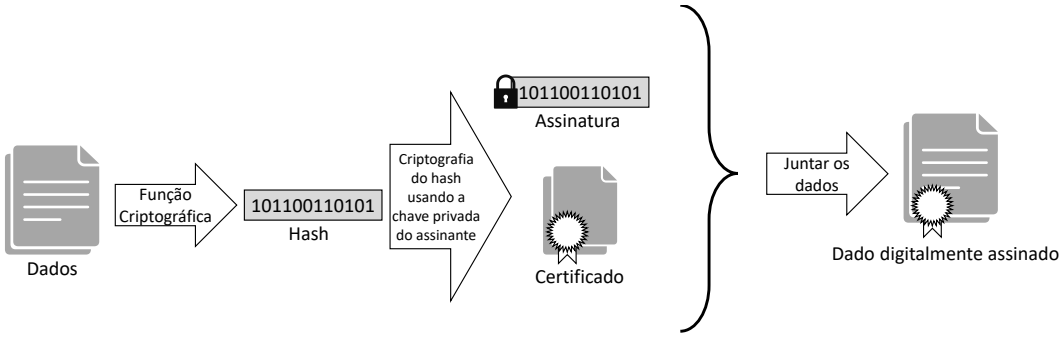
3.3. Algoritmos de assinatura, funções criptográficas e tamanhos de chaves

Os algoritmos de assinatura são utilizados durante a negociação entre o cliente e o servidor, e garantem o quão segura será a criptografia utilizada na transmissão dos dados entre eles. No âmbito dos certificados para TLS/SSL, são utilizados essencialmente dois tipos de algoritmos, o RSA (*Rivest, Shamir, and Adelman*) e EDCDA (*Elliptic Curve Digital Signature Algorithm*). Enquanto o RSA é o padrão consolidado e utilizado na maioria dos certificados, o EDCDA surge como uma opção promissora pelo fato de implementar o ECDLP (*Elliptic Curve Discrete Logarithm Problem*), o que torna o algoritmo mais difícil de ser quebrado.

Segundo a RFC3279 [Housley et al. 2002], todo algoritmo de assinatura deve vir acompanhado de uma função criptográfica que será responsável por gerar o *hash* da mensagem a ser enviada, antes de passar pelo processo de assinatura do algoritmo. A Figura 2 ilustra resumidamente o processo de assinatura de uma informação antes de ser enviada. Primeiro, é gerado o *hash* da mensagem. Em seguida, é realizada a assinatura da mensagem a partir da chave privada do certificado.

[16] http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/Lei/L13709.htm

Figura 2. Processo de assinatura



Todo certificado possui uma chave privada associada, que é utilizada no processo de assinatura das informações. Esta chave tem tamanho medido em bits e deve seguir alguns padrões recomendados por entidades como o NIST para garantir a segurança. O algoritmo de assinatura utilizado impacta diretamente na relação tamanho da chave *versus* segurança da informação. Para o algoritmo RSA, atualmente o NIST recomenda chaves de no mínimo 2048 bits, enquanto para o EDCDA chaves de 256 bits garantem o mesmo nível de proteção.

Os dois gráficos da Figura 3 apresentam os resultados para os conjuntos de algoritmo + função criptográfica e tamanhos de chaves utilizados nos 5510 sites analisados. A análise mostra que 97,4% dos sites utilizam algoritmos de assinaturas recomendados, enquanto 143 sites utilizam funções obsoletas como MD5 e SHA-1. Quanto ao tamanho das chaves, 98,1% dos sites utilizam chaves RSA e EDCDA dentro dos padrões recomendados.

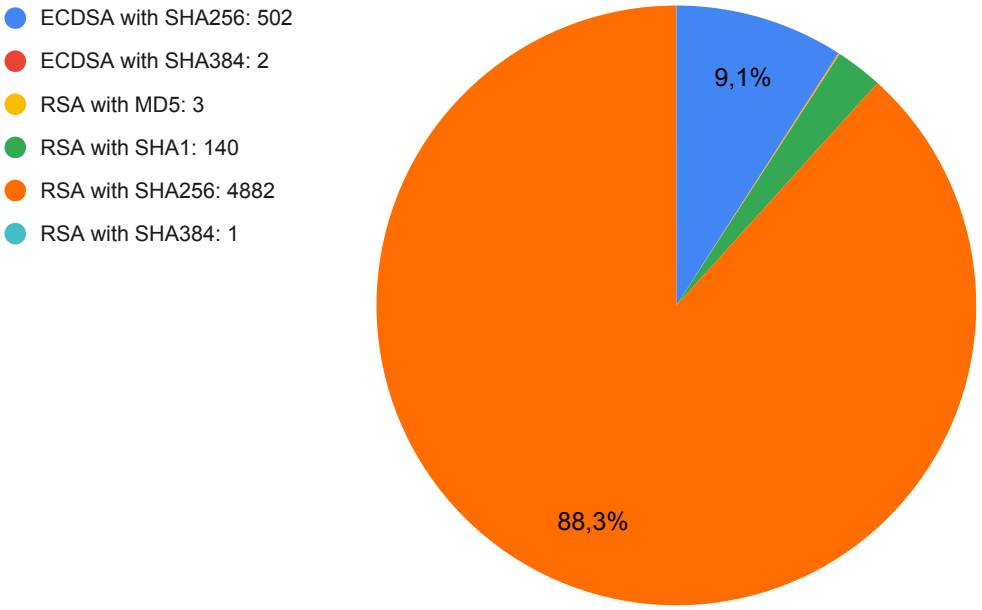
3.4. Navegadores e o uso de cifras

Um dos objetivos do *handshake* é permitir que o cliente e o servidor definam uma cifra comum a ser utilizada no restante da comunicação. Uma cifra é composta por um conjunto de algoritmos criptográficos (e.g., algoritmo de assinatura) que irão ser utilizados sessão TLS. Uma cifra forte deve ser definida para que o restante da comunicação ocorra com segurança. Neste trabalho, foi utilizado o documento de referência do NIST como referência para definir se uma cifra é segura ou não [McKay and Cooper 2019]. Todas as cifras não recomendadas pelo NIST são classificadas como não recomendadas.

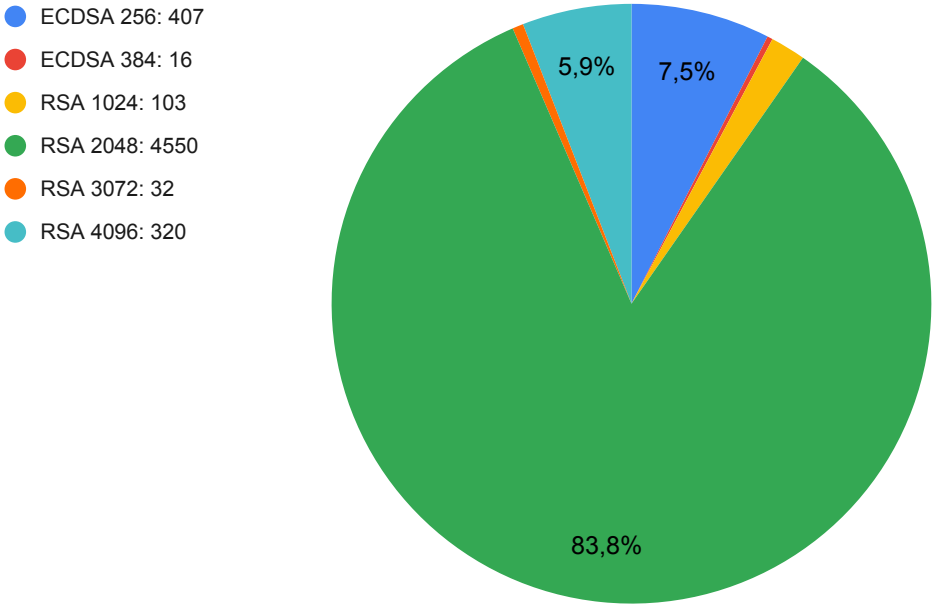
A ferramenta `testssl.sh` simula conexões com um número limitado de navegadores e captura as versões do TLS e cifras negociadas com o site. A tabela 4 resume os dados coletados para cada navegador simulado, exibindo as porcentagens de sites que utilizam cifras recomendadas, cifras não recomendadas, além dos sites que não aceitam conexões a partir daquele navegador.

Fica evidente que a utilização de navegadores atualizados impacta diretamente na segurança da comunicação. Para navegadores atualizados, mais de 90% dos sites analisados oferecem cifras recomendadas pelo NIST. Por outro lado, para navegadores mais antigos, como o Internet Explorer 8, 71,49% dos sites disponibilizam cifras não recomendadas. Outro aspecto a ser observado é o fato de navegadores mais atualizados, como o Google Chrome 79, não conseguirem realizar conexões com alguns sites. Isto se deve principalmente ao fato do o próprio navegador limitar o acesso a sites que suportem

Figura 3. Algoritmos de assinatura, funções criptográfica e tamanho das chaves



(a) Algoritmo + Função



(b) Tamanho da chave

Tabela 4. Navegadores atualizados e suas cifras

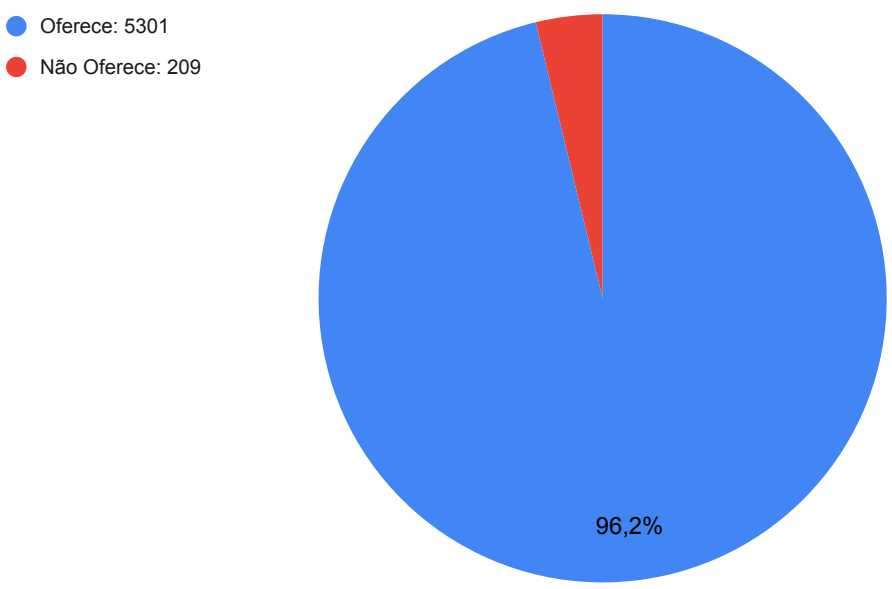
Navegador	Cifras recomendadas	Cifras não recomendadas	Sem conexão
Google Chrome 79	95,60%	4,39%	0,01%
Mozilla Firefox 71	93,32%	6,64%	0,04%
Internet Explorer 11	96,43%	3,57%	0%
Opera 66	95,61%	4,37%	0,02%
Safari 13	94,92%	5,06%	0,02%
Internet Explorer 8	0,90%	71,49%	27,61%

versões antigas do TLS/SSL, como SSv2 ou SSv3 [Constantin 2014], evitando ataques como DROW e BEAST.

3.5. Utilização de PFS

Perfect forward secrecy (PSF) é uma propriedade que visa garantir a segurança de comunicações passadas, ou seja, comunicações realizadas antes de um atacante comprometer a chave secreta utilizada na criptografia da informação [Barbosa et al. 2018, Kreutz et al. 2019]. Ao contrário de uma conexão convencional, uma comunicação utilizando PFS faz uso de uma chave secreta a cada nova seção entre o cliente e o servidor. Desta forma, caso um atacante descubra a chave secreta de uma sessão em andamento, não conseguirá comprometer a segurança das informações trocadas nas sessões anteriores. Felizmente, a Figura 4 apresenta um cenário animador. Pelas análises realizadas, 96,20% dos sites utilizam algoritmos que suportam PFS.

Figura 4. Utilização do PFS



3.6. Sites governamentais *versus* iniciativa privada

Tanto sites de órgãos governamentais como de empresas privadas permitem conexões utilizando versões dos protocolos TLS/SSL com vulnerabilidades conhecidas, expondo os usuários (e seus respectivos dados) a incidentes de segurança e privacidade.

Sites do Governo Federal. Os sites do governo federal, apresentam um cenário interessante quanto a segurança. Este grupo possui 85% dos seus sites suportando TLS 1.3, além da grande maioria suportar também a versão 1.2. Entretanto, 10% dos sites da alta esfera do governo federal sequer possuem HTTPS habilitado.

Sites dos governos estaduais. No contexto estadual, o cenário é pior. Primeiro, 29,66% dos sites sequer utiliza HTTPS. Segundo, 23 dos 108 sites analisados possuem problemas relacionados ao domínio indicado no certificado, que não corresponde ao domínio acessado. Outros 18,5% dos sites possuem a cadeia de certificados quebrada, onde a maioria deles são certificados auto-assinados, ou estão com data de validade expirada. Não bastasse isso, mais de 58,26% dos sites suporta TLS 1.0, ou seja, são vulneráveis a ataques como o BEAST.

Sites dos governos municipais. Nos sites de governos municipais a situação dos certificados não é diferente. O estudo mostrou que 11,81% dos sites deste grupo não utilizam HTTPS em suas conexões. Outros 8,66% dos sites apresentam o domínio diferente do domínio registrado no certificado, o que configura uma quebra na cadeia do certificado. Com relação aos ataques, 3,14% dos sites suportam SSLv2, isto é, são suscetíveis ao ataque DROWN, e 59,84% dos sites são vulneráveis aos ataques POODLE e BEAST, pois suportam versões anteriores ao TLS 1.1.

Sites da iniciativa privada. O cenário na iniciativa privada, considerando instituições financeiras e de *e-commerce*, também é preocupante. Em 54% dos sites bancários, os usuários podem estar em risco devido ao fato de suportar versões do TLS/SSL vulneráveis a ataques como o BEAST e o POODLE. Este número sobe para 78% nos sites de *e-commerce*, o que causa bastante preocupação, pois milhões de consumidores online podem estar em risco.

Por fim, vale ressaltar que nenhum dos sites analisados suporta exclusivamente TLS 1.3, o que aumentaria a segurança das comunicações. Entretanto, existem iniciativas internacionais para migrar e limitar navegadores e servidores Web ao TLS 1.3, aumentando a segurança dos usuários finais e instituições.

4. Discussão

A utilização de certificados digitais, para viabilizar conexões HTTPS, por si só, não garante segurança e privacidade aos usuários e instituições. Como visto nos resultados apresentados na Seção 3, a maioria dos sites analisados são vulneráveis a ataques conhecidos. Os problemas variam, indo desde falhas de configuração dos sistemas até problemas mais específicos, inerentes à geração dos certificados digitais.

Os profissionais de tecnologia, responsáveis pela instalação dos certificados e manutenção dos sites, precisam estar minimamente cientes da importância e da criticidade das suas tarefas. Por exemplo, um certificado digital qualquer, instalado sem o mínimo de cuidados técnicos, não traz segurança aos usuários do sistema. Como ocorre

frequentemente, o uso de HTTPS leva a uma falsa sensação de segurança aos usuários e instituições.

As análises realizadas neste trabalho, observando a utilização de certificados digitais e protocolos suportados nas conexões HTTPS do ecossistema da Internet do Brasil, mostram que ainda há um longo caminho pela frente para oferecer, efetivamente, segurança e privacidade aos usuários e instituições. Alternativas gratuitas para geração de certificados válidos e de qualidade, como o Let's Encrypt (`https://letsencrypt.org/`), podem representar uma opção atrativa para entidades que não possuem o conhecimento técnico ou os recursos financeiros para adquirir e gerenciar certificados de ACs tradicionais, reconhecidas pelos navegadores. Certificados emitidos pelo Let's Encrypt são reconhecidos e autenticados pela maioria dos navegadores.

Outro ponto que merece destaque são as novas formas de ataque ao HTTPS, que independem do certificado ou protocolo utilizado. Um exemplo é o *Malware Reductor* [Kaspersky researchers 2019], um *malware* disseminado através da Internet com objetivo de afetar a segurança do HTTPS diretamente na máquina cliente. Ao infectar o dispositivo da vítima, o *malware* manipula os certificados digitais utilizados para estabelecer conexões seguras, permitindo ao atacante acessar os dados do usuário.

Fica claro também, que o usuário tem a sua parcela de responsabilidade na segurança dos seus dados. A análise realizada neste trabalho demonstra que a utilização de navegadores atualizados é importante para a segurança da comunicação. As simulações realizadas pela ferramenta *testssl.sh* demonstram que uma grande parcela dos sites respondem às solicitações de conexão com cifras recomendadas pela NIST quando as requisições são realizadas por navegadores atualizados. Por outro lado, quando a requisição parte de navegadores desatualizados, o que prevalece é o uso de cifras não recomendadas. Este é o caso do Internet Explorer 8. Mais de 70% dos 5806 sites analisados ofereceram cifras consideradas fracas para as conexões deste navegador.

Atualmente, as empresas responsáveis pelo desenvolvimento dos principais navegadores estão unindo esforços para que a Internet se torne um lugar mais seguro. Como anunciado no início de 2020 [van der Merwe 2020], os navegadores irão, em breve, apresentar uma mensagem de advertência em sites que suportam apenas protocolos inseguros, ou seja, anteriores ao TLS 1.2, permitindo que o usuário prossiga ou não no site em questão. Neste caso, caberá ao usuário decidir por correr, ou não, o risco de ter os seus dados eventualmente comprometidos por agentes maliciosos.

Vale ressaltar também, como demonstrado na Seção 2, que existem ferramentas de fácil utilização. Algumas delas atribuem, inclusive, nota ao nível de segurança do site analisado, o que simplifica a análise para o usuário final. Além disto, os próprios navegadores já realizam notificações em casos críticos de segurança dos certificados, cabendo ao usuário desconfiar da confiabilidade do site em questão. Finalmente, cabe também aos profissionais de tecnologia da informação difundir o conhecimento e criar uma cultura saudável e positiva de segurança da informação dentro de suas instituições.

Por fim, uma análise ainda mais extensa do ecosistema HTTPS do Brasil pode ser encontrada na versão estendida do paper [Fiorenza et al. 2020]. Nesta versão também estão disponíveis informações sobre os *scripts* e ferramentas de automação desenvolvidas para a coleta e análise dos dados. Há um repositório no GitHub com os códigos e os dados

utilizados neste trabalho e na respectiva versão estendida.

Referências

Aviram, N., Schinzel, S., Somorovsky, J., Heninger, N., Dankel, M., Steube, J., Valenta, L., Adrian, D., Halderman, J. A., Dukhovni, V., et al. (2016). DROWN: Breaking TLS Using SSLv2. In *25th USENIX Security Symposium*, pages 689–706.

Barbosa, R. d. O., Winckler, S. C., Kreutz, D., and Chervinski, J. O. M. (2018). Proteção de comunicações passadas e futuras 101. In *Revista Brasileira de Computação Aplicada*, pages 1–4.

Bokslag, W. (2016). The problem of popular primes: Logjam. *arXiv preprint arXiv:1602.02396*.

Chothia, T., Garcia, F. D., Heppel, C., and Stone, C. M. (2017). Why Banker Bob (still) Can't Get TLS Right: A Security Analysis of TLS in Leading UK Banking Apps. In *Int. Conf. on Financial Cryptography and Data Security*, pages 579–597. Springer.

Constantin, L. (2014). Google to kill off SSL 3.0 in Chrome 40. `https://www.computerworld.com/article/2841821/google-to-kill-off-ssl-30-in-chrome-40.html`.

Duong, T. and Rizzo, J. (2011). Here come the ninjas. *Unpublished manuscript*, 320.

Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., and Halderman, J. A. (2015). A search engine backed by internet-wide scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 542–553.

Durumeric, Z., Kasten, J., Bailey, M., and Halderman, J. A. (2013). Analysis of the HTTPS certificate ecosystem. In *ACM IMC*, pages 291–304. ACM.

Escarrone, T., Kreutz, D., and Fiorenza, M. (2019). Uma Primeira Analise do Ecossistema HTTPS no Brasil. In *4o Workshop Regional de Segurança da Informação e de Sistemas Computacionais*, Alegrete-RS, Brasil. `http://errc.sbc.org.br/2019/wrseg/papers/escarrone2019uma.pdf`.

Fiorenza, M. M., Kreutz, D., Escarrone, T., and Temp, D. (2020). Uma Análise da Utilização de HTTPS no Brasil (versão estendida arXiv). `https://arxiv.kreutz.xyz/https_no_BR_arXiv2020.pdf`.

Frost, V., Tian, D. J., Ruales, C., Prakash, V., Traynor, P., and Butler, K. R. B. (2019). Examining DES-based Cipher Suite Support Within the TLS Ecosystem. In *2019 ACM AsiaCCS*, pages 539–546. ACM.

Housley, R., Polk, T., and III, L. E. B. (2002). Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3279.

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.

Kaspersky researchers (2019). Not so random: Reductor malware hijacks HTTPS traffic by manipulating browsers' "random"numbers generator. `shorturl.at/FMSY5`.

Kerner, S. (2015). Logjam SSL/TLS Vulnerability Exposes Cryptographic Weakness. `shorturl.at/amu69`.

Kreutz, D., Winckler, S. C., de Oliveira Barbosa, R., Chervinski, J. O., and Jenuario, T. (2019). Introdução a Propriedades Básicas e Avançadas de Segurança da Informação. In *17a Escola Regional de Redes de Computadores*, Alegrete-RS, Brasil. `http://errc.sbc.org.br/2019/mc/kreutz2019propriedades.pdf`.

Matsumoto, S. and Reischuk, R. M. (2015). Certificates-as-an-Insurance: Incentivizing accountability in SSL/TLS. In *USENIX NDSS WSENT*.

McKay, K. and Cooper, D. (2019). Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations (2nd Draft). Technical Report 52r2, National Institute of Standards and Technology.

Merzdovnik, G., Falb, K., Schmiedecker, M., Voyiatzis, A. G., and Weippl, E. (2016). Whom You Gonna Trust? A Longitudinal Study on TLS Notary Services. In *Data and Applications Security and Privacy*, pages 331–346. Springer.

Möller, B., Duong, T., and Kotowicz, K. (2014). This POODLE bites: exploiting the SSL 3.0 fallback. *Security Advisory*.

Rescorla, E. and Dierks, T. (2008). The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246.

Samarasinghe, N. and Mannan, M. (2019). Another look at TLS ecosystems in networked devices vs. Web servers. *Computers & Security*, 80:1 – 13.

van der Merwe, T. (2020). It's the boot for tls 1.0 and tls 1.1. `https://hacks.mozilla.org/2020/02/its-the-boot-for-tls-1-0-and-tls-1-1/`.

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.