
MANICODE
SECURE CODING EDUCATION
The Abridged History of Application Security


MANICODE
SECURE CODING EDUCATION
extremely
The Abridged History of Application Security
Things are Getting a Lot Better

Jim Manico

jim@manicode.com

@manicode

- Former OWASP Global Board Member
- Project manager of the OWASP Cheat Sheet Series and several other OWASP projects
- 20+ years of software development experience
- Author of "Iron-Clad Java, Building Secure Web Applications" from McGraw-Hill/Oracle-Press
- Investor/advisor for KSOC, Nucleus Security, Signal Sciences, Secure Circle and BitDiscovery
- Based on Kauai, Aldie VA and Hotels worldwide

[redacted]

@manicode

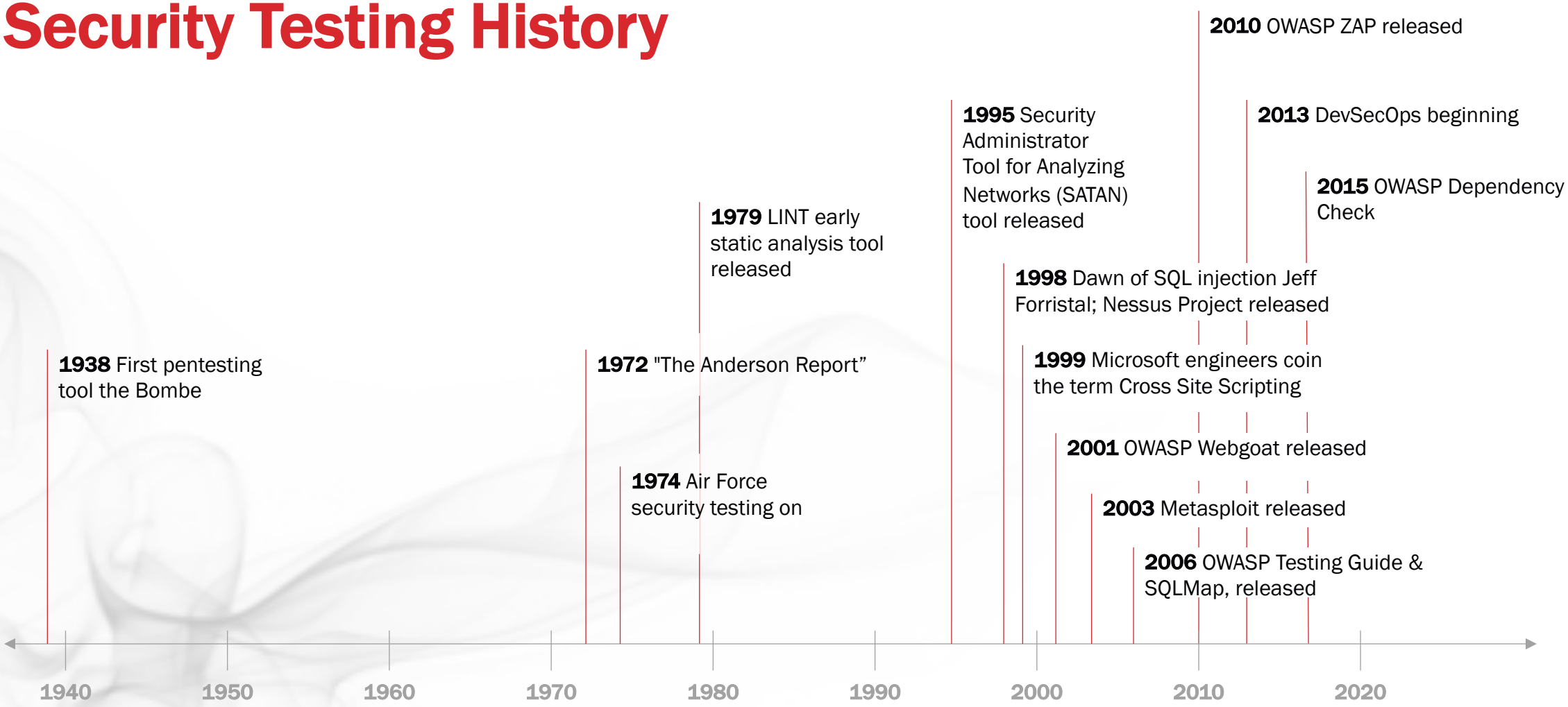
- Former [redacted]
- Project manager of the OWASP [redacted] OWASP projects

InfoSec Dark Ages

- October 1967 Task Force
- February 1970 R-609 Published
- October 1975 R-609 Declasified

IS MESSED UP!

Security Testing History



2020

- Security Testing Integrated Into GitHub
- SAST, DAST, 3rd Party Scanning and IAST
- Pentesting Still Expensive (not enough professionals out there)
- Compliance is a growing security testing driver

HTTP/S History

1994 Netscape creates initial version of HTTPS

1999 TLS 1.0 released

2006 TLS 1.1 released2008 TLS 1.2 released

2009 SSLLabs released to public as a way to verify security configuration of HTTPS websites

2010 Chrome starts to HSTS preload some sites

2011 Forward secrecy live in modern browsers

2013 TLS 1.2 live in modern browsers

2015 Lets Encrypt starts!

2016 Overt ½ the web HTTPS; Chrome 51 defaults to HTTP/2 and only allows TLS

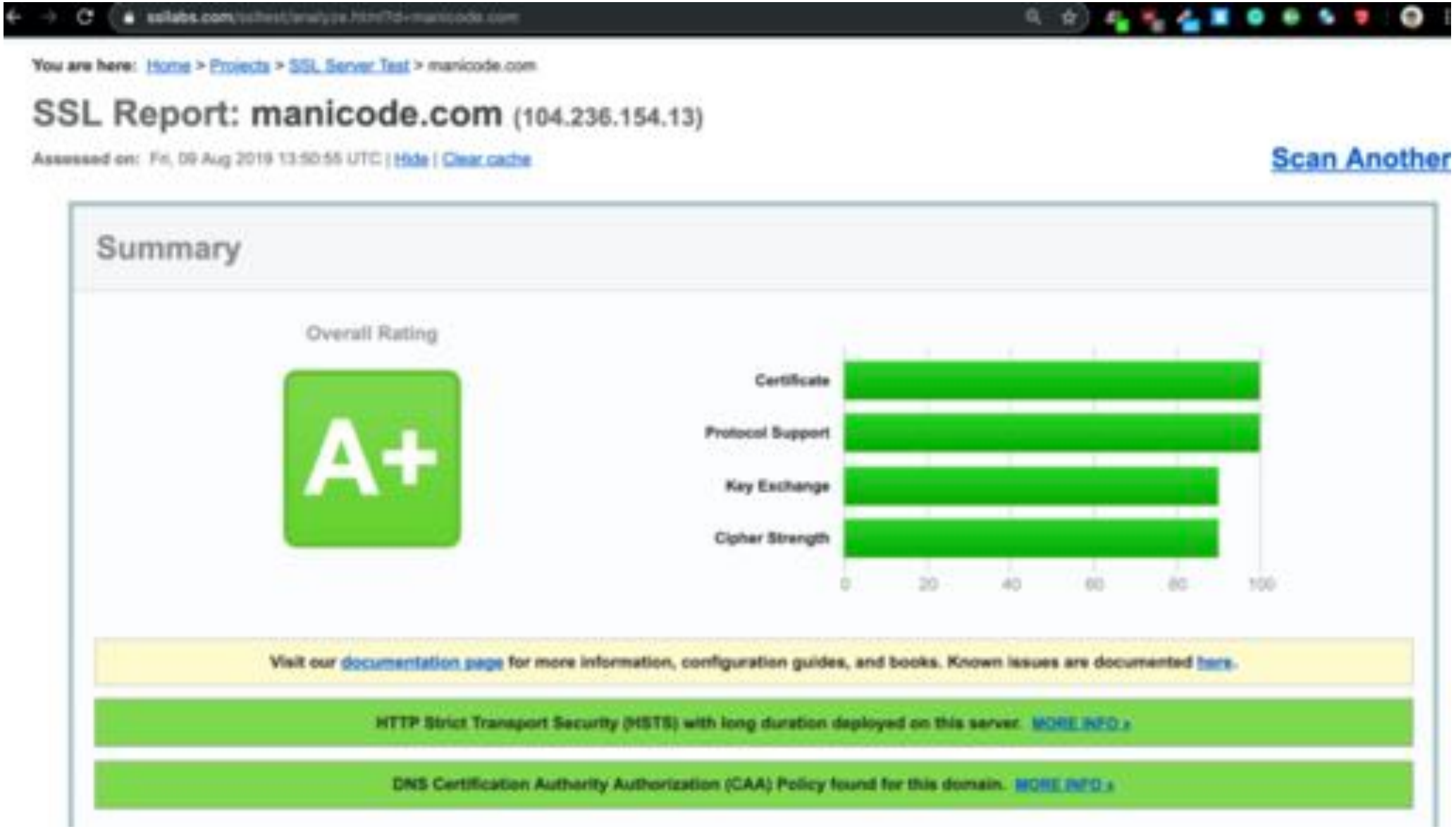
2017 CAA becomes mandatory

2018 Let's Encrypt offers wildcard; TLS 1.3 published as RFC 8446; CT required for new certs

1995 2010 2015 2020

2020

- https://transparencyreport.google.com/https/overview
- 90% or more of the web is HTTP/S


You are here: Home > Projects > SSL Server Test > manicode.com
SSL Report: manicode.com (104.236.154.13)
Assessed on: Fri, 09 Aug 2019 13:50:55 UTC | Hide | Clear cache
Scan Another
Summary
Overall Rating
A+
Certificate
Protocol Support
Key Exchange
Cipher Strength
0
20
40
60
80
100
Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.
HTTP Strict Transport Security (HSTS) with long duration deployed on this server. MORE INFO »
DNS Certification Authority Authorization (CAA) Policy found for this domain. MORE INFO »

Password History

1961 MIT's CTSS (https://en.wikipedia.org/wiki/Compatible_Time-Sharing_System) MOTD bug

1970s Crypt(3) released in Unix uses old M-209 code from WW2 (unix up to 6th edition)

1978 Crypt(3) released in Unix now DES based (7th edition); first stretching, salting and password policy

1980s Un… passwd dat… limited to …

1991 MD5 message-digest algori… h…

2007 PHP apps start using …sword storage, …ilable

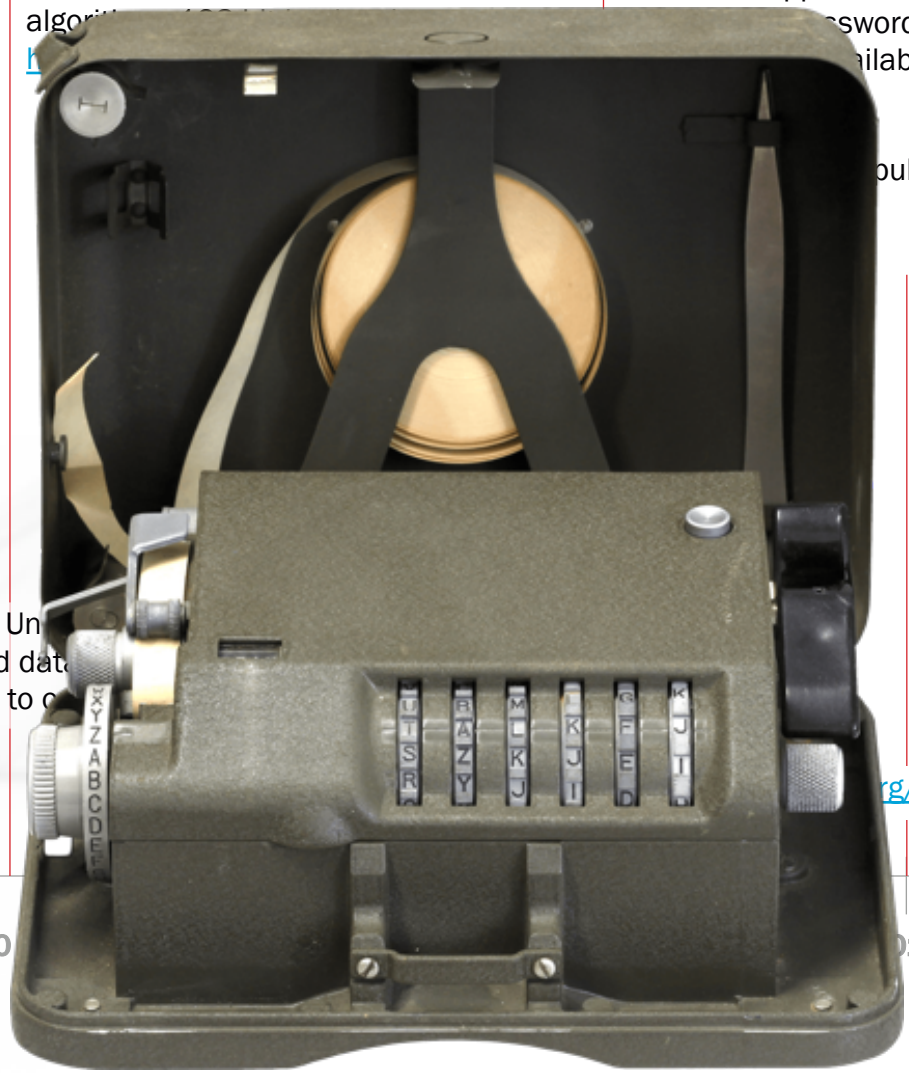
…ublished

2015 Argon2 wins password hashing competition https://password-hashing.net/

2016 Dr. Akhawe from Dropbox publishes password storage strategy

…rg/wiki/SHA-2

1960 1965 1970 1975 1980 1990 …015 2020



M-209B, cryptography collection of the Swiss Army headquarters. Photographed by Rama, Wikimedia Commons, licensed under CeCILL v2 and CC-BY-SA-2.0-FR

How Dropbox securely stores your passwords

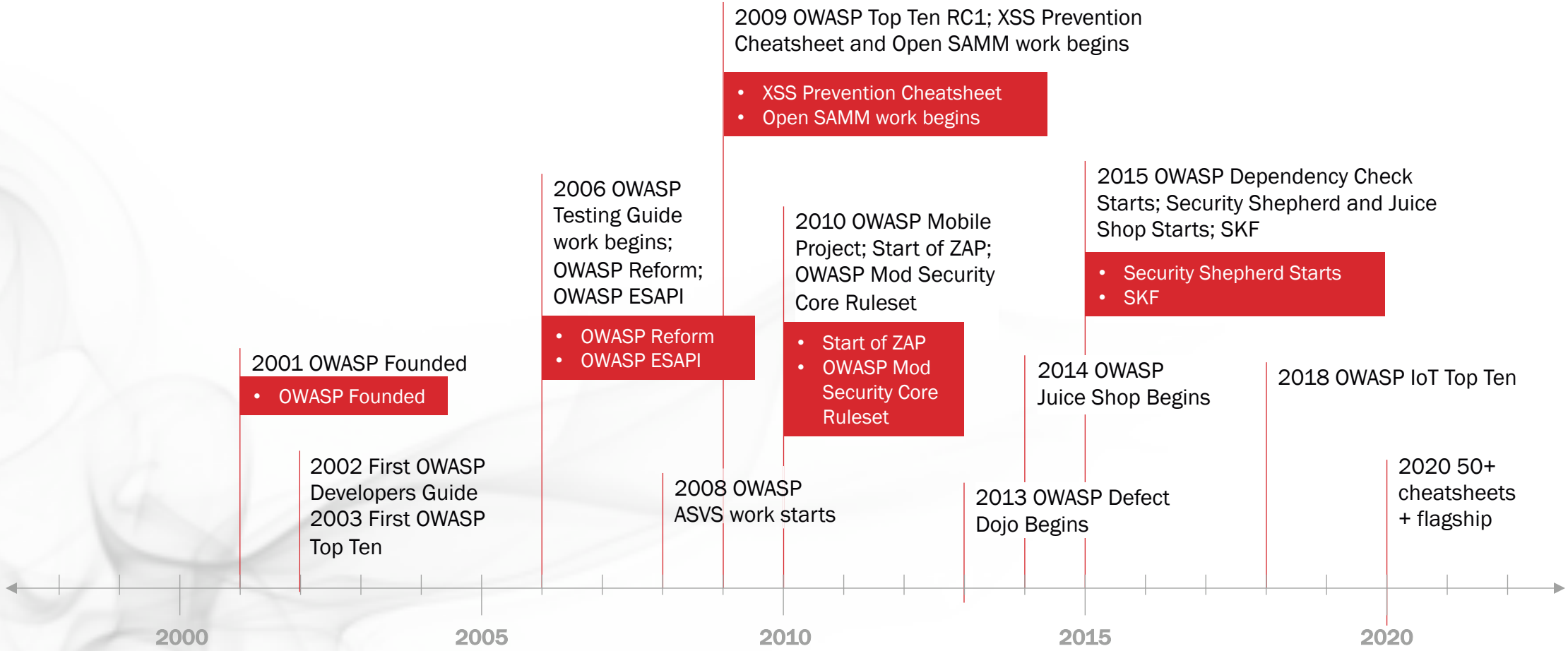
Devdatta Akhawe | September 21, 2016

It's universally acknowledged that it's a bad idea to store plain-text passwords. If a database containing plain-text passwords is compromised, user accounts are in immediate danger. For this reason, as early as 1976, the industry standardized on storing passwords using secure, one-way hashing mechanisms (starting with Unix Crypt). Unfortunately, while this prevents the direct reading of passwords in case of a compromise, all hashing mechanisms necessarily allow attackers to brute force the hash offline, by going through lists of possible passwords, hashing them, and comparing the result. In this context, secure hashing functions like SHA have a critical flaw for password hashing: they are designed to be fast. A modern commodity CPU can generate millions of SHA256 hashes per second. Specialized GPU

2020

- Argon2 supported everywhere

OWASP Project History



Flagship Projects

The OWASP Flagship designation is given to projects that have demonstrated strategic value to OWASP and application security as a whole. After a major review process More info here the following projects are considered to be flagship candidate projects. These project have been evaluated more deeply to confirm their flagship status:

Tools

- OWASP Zed Attack Proxy
- OWASP Web Testing Environment Project
- OWASP OWTF
- OWASP Dependency Check
- OWASP Security Shepherd
- OWASP DefectDojo Project
- OWASP Juice Shop Project

XSS History

1999 Microsoft engineers coin the term Cross Site Scripting

2002 HTTPOnly supported in IE 6 SP1

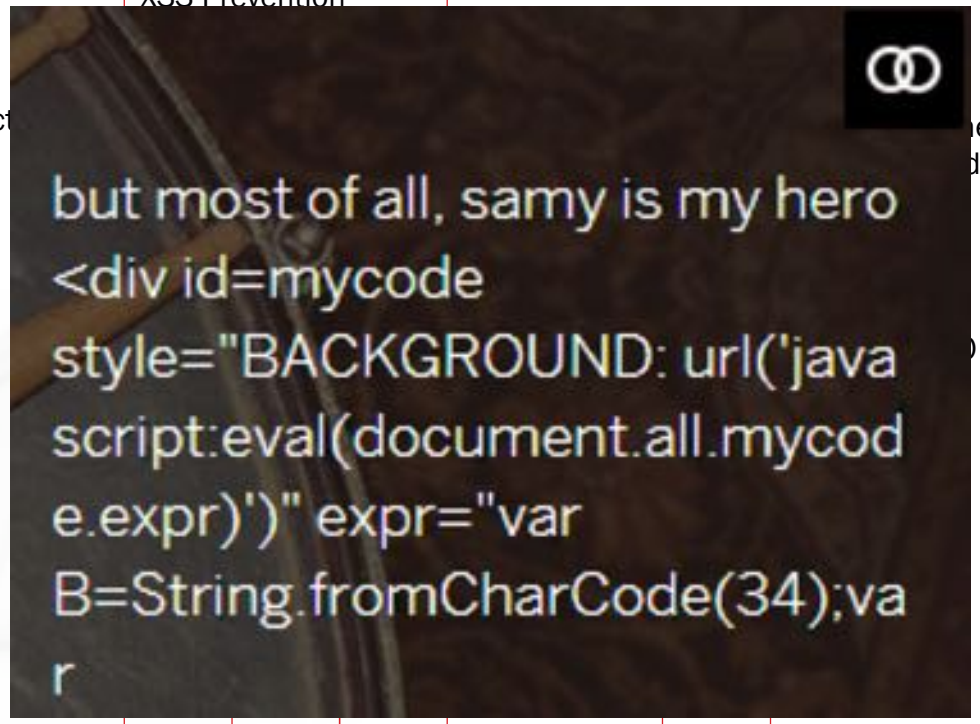
2005 But most of all, samy is my hero

2005 Amit Klein first DOM XSS publication

2006 OWASP Reform project starts

2009 OWASP XSS Prevention

2012 CSP 1.0 published



2019 Trusted Types Live in Chrome, React

2000 2005 2010 2015

Trusted Types help prevent Cross-Site Scripting

☆ ☆ ☆ ☆ ☆

TL;DR

We've created a new experimental API that aims to prevent DOM-Based Cross Site Scripting in modern web applications.

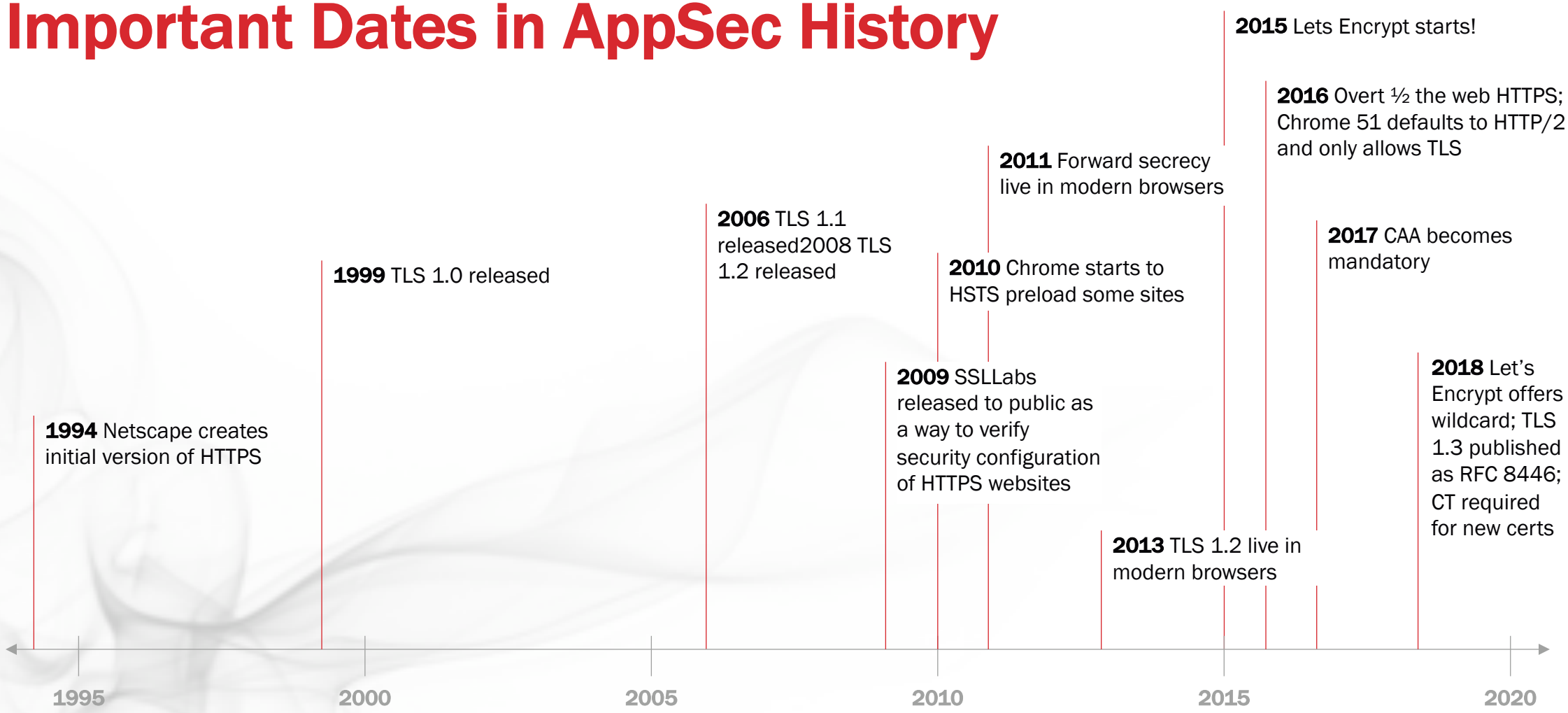
By Krzysztof Kotowicz
Software Engineer in the Information Security Engineering team at Google

We're currently working on the specification and implementation details for this API. We'll keep this post updated

2020

- AutoEscaping templates the norm
- CSP with strict-dynamic is easier to deploy
- Trusted Types is making its way into frameworks

Important Dates in AppSec History



2020

- February 21, 2020 OWASP New Zealand Day

AppSec is Global 260+ OWASP Chapters Worldwide

AMA-AA

(Ask Me Anything About AppSec)

Thank you to *YOU*...
for helping the world be more secure.

Have a great OWASP NZ CONFERENCE!