Crypto Done Right (0.2)

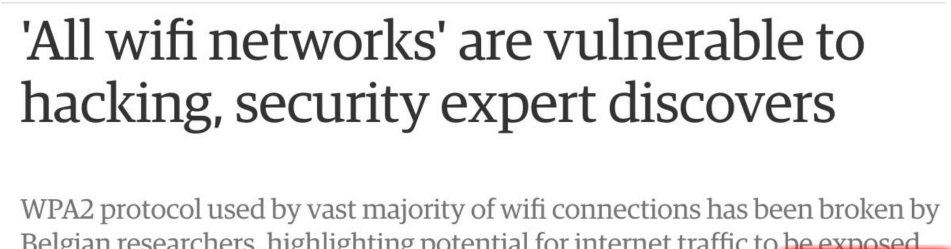
One Year Later, Lessons Learned

Good Morning! Introductions?

- Dr. Seth Nielson
- Founder of Crimson Vista, Inc
- Adjunct faculty at Johns Hopkins

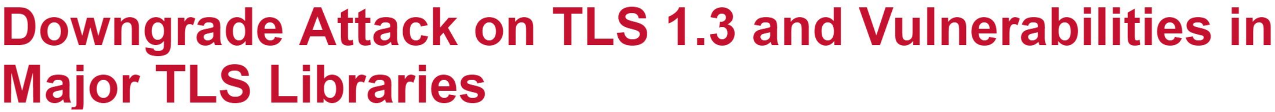
- Tell me about yourselves

Crypto Troubles (Still going!)

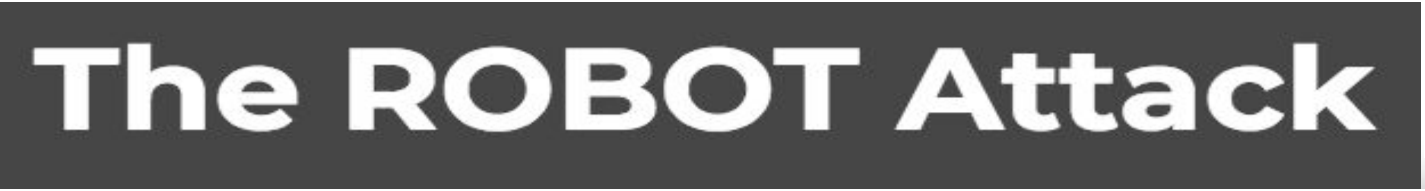


'All wifi networks' are vulnerable to hacking, security expert discovers

WPA2 protocol used by vast majority of wifi connections has been broken by Belgian researchers, highlighting potential for internet traffic to be exposed

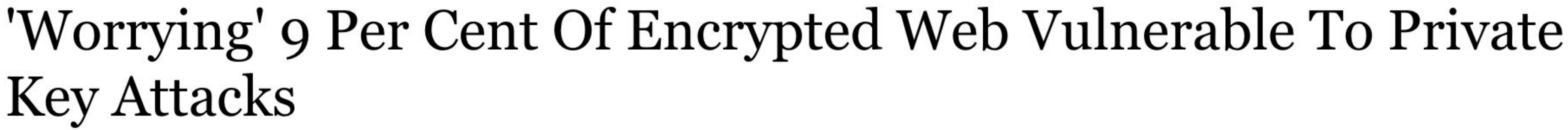


Downgrade Attack on TLS 1.3 and Vulnerabilities in Major TLS Libraries



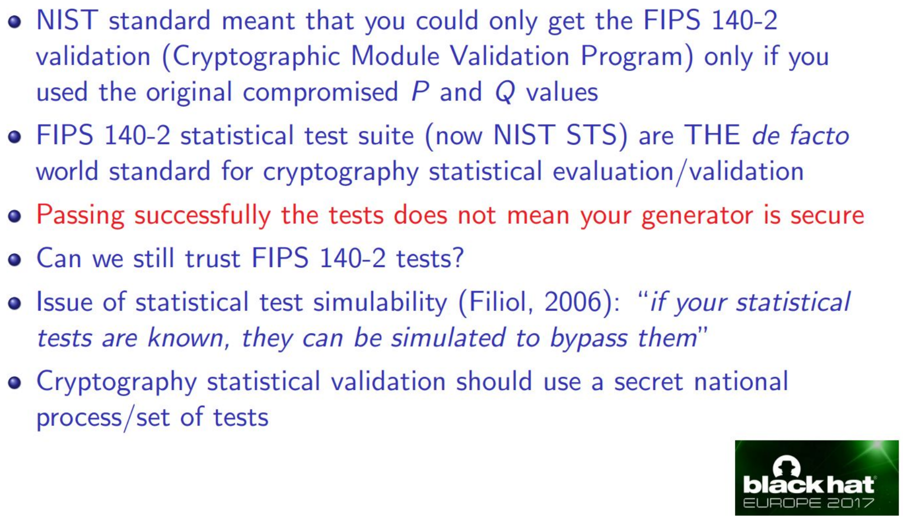
The ROBOT Attack

Timing vulnerabilities with CBC-mode symmetric decryption using padding



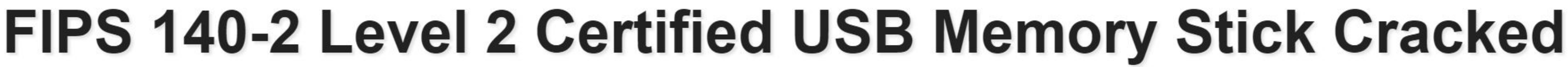
'Worrying' 9 Per Cent Of Encrypted Web Vulnerable To Private Key Attacks

Even Certified Modules are not Immune



- NIST standard meant that you could only get the FIPS 140-2 validation (Cryptographic Module Validation Program) only if you used the original compromised P and Q values
- FIPS 140-2 statistical test suite (now NIST STS) are THE *de facto* world standard for cryptography statistical evaluation/validation
- Passing successfully the tests does not mean your generator is secure
- Can we still trust FIPS 140-2 tests?
- Issue of statistical test simulability (Filiol, 2006): "*if your statistical tests are known, they can be simulated to bypass them*"
- Cryptography statistical validation should use a secret national process/set of tests

black hat
EUROPE 2017



FIPS 140-2 Level 2 Certified USB Memory Stick Cracked

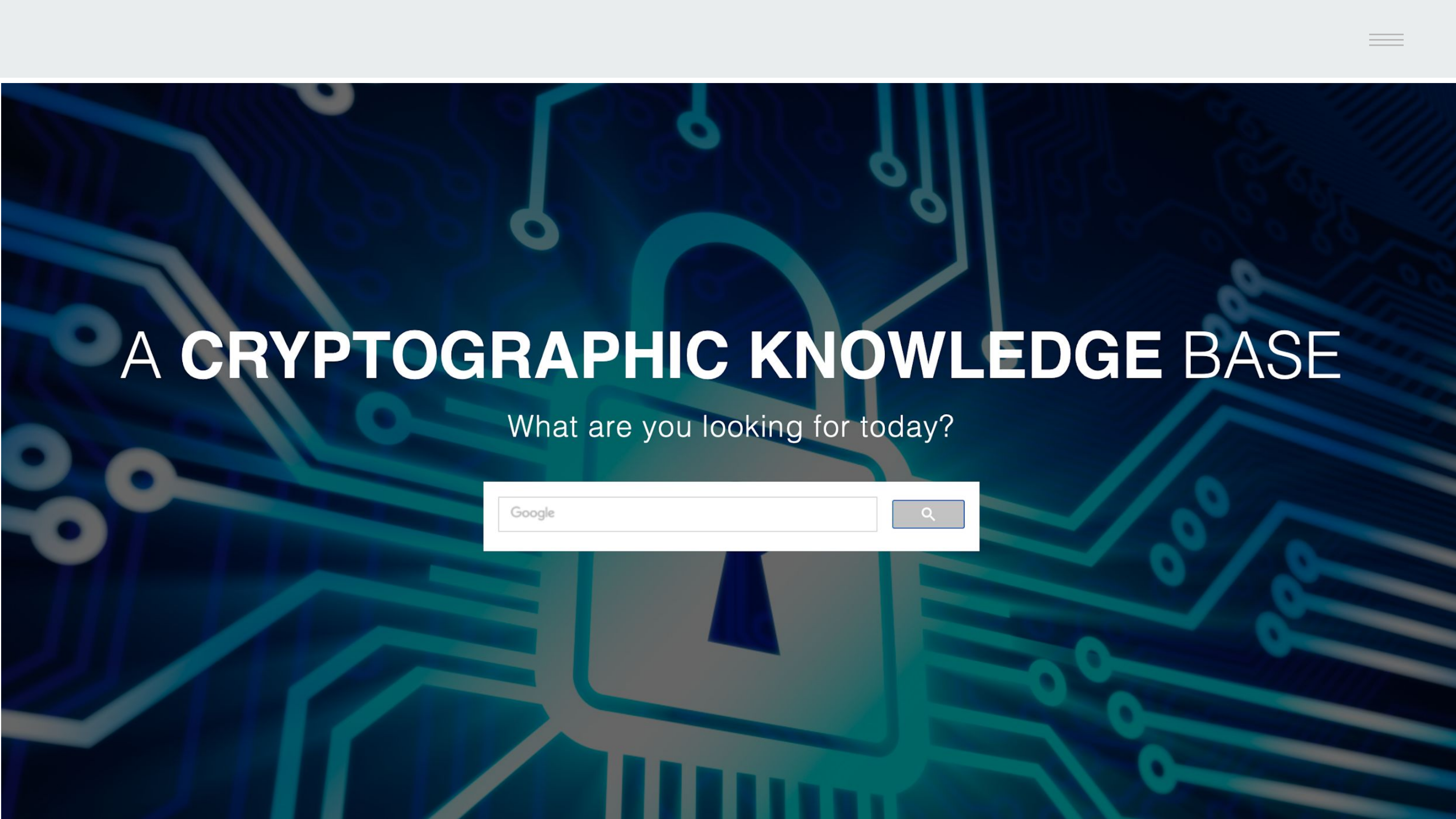
Security Beyond the Boundary

- Crypto must be configured correctly
- Crypto must be maintained

- Often by *non-experts*

Solutions?

- Prohibit crypto without license?
- Make crypto easier to use?
- **Educate the masses?**



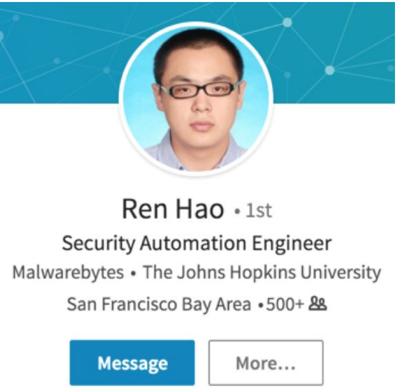
A CRYPTOGRAPHIC KNOWLEDGE BASE

What are you looking for today?

Inception

Is there one place to find best practices cryptographic configuration information?

- Original idea by Debra Baker, CISSP, CCSP
- The idea came out of a presentation Debra did on cryptographic best practices.
- Industry needs this because even with the DUAL_EC_DRBG debacle, it took 7 years before industry made a change away from using it.



Ren Hao • 1st

Security Automation Engineer

Malwarebytes • The Johns Hopkins University

San Francisco Bay Area • 500+

9/8/15: Apple's iMessage defense against spying has one flaw, CS's Matt Green, WIRED

Online Privacy In The Trump Era, JHUISI' Avi Rubin, WYPR

MAY 10, 2017

Cisco

- Initial funding gift to Hopkins
- No strings attached
- Additional collaboration and ideas
- Committed to an independent community

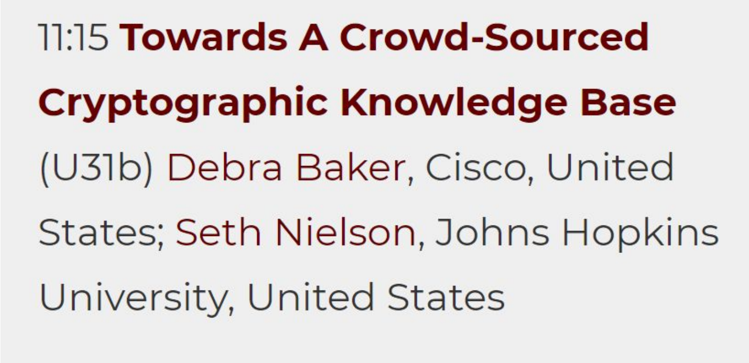
Getting Started

- Understanding the audience
- Developing a philosophy
- Creating initial content

Crypto Done Right

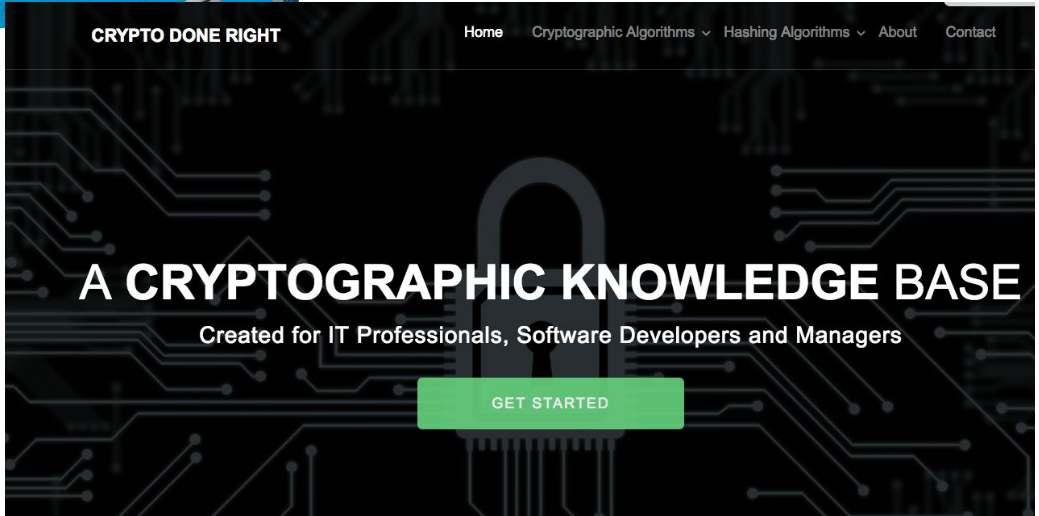
First Reveal:

CONFERENCE AGENDA 2018



11:15 **Towards A Crowd-Sourced Cryptographic Knowledge Base**

(U31b) Debra Baker, Cisco, United States; Seth Nielson, Johns Hopkins University, United States



A CRYPTOGRAPHIC KNOWLEDGE BASE

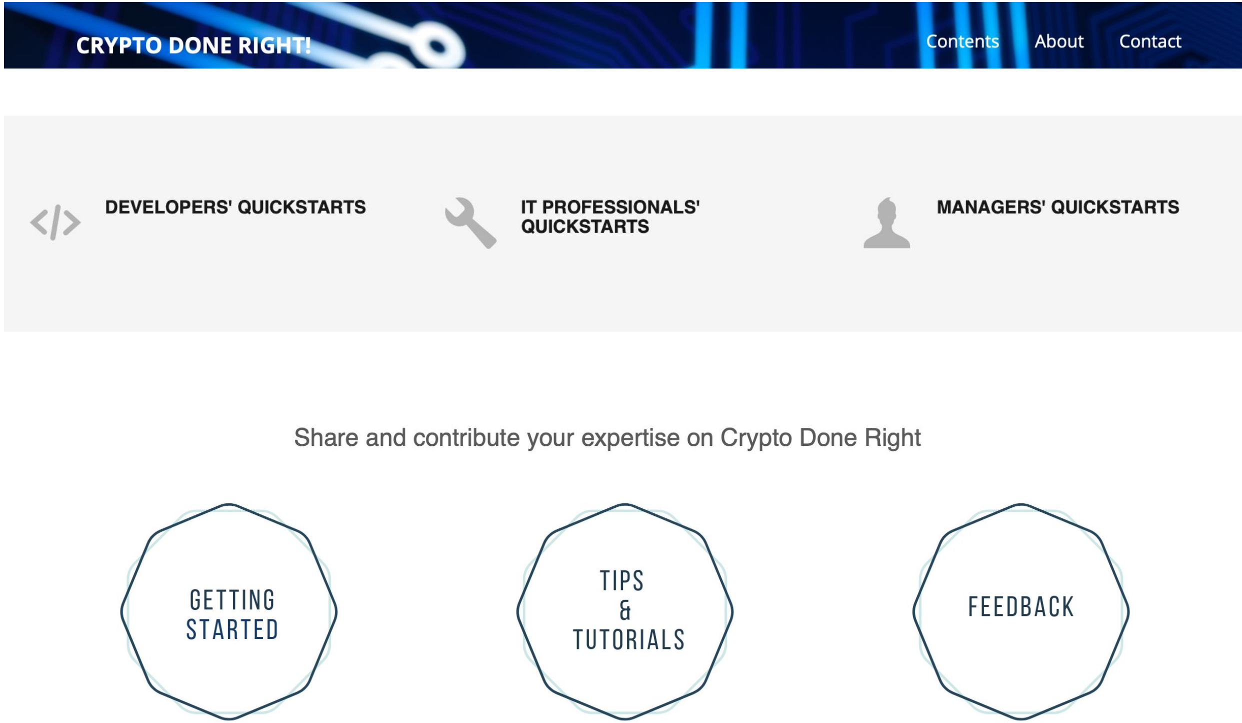
Created for IT Professionals, Software Developers and Managers

Version 0.2
ICMC 2019

- Spent a year testing and tweaking
- Learned many lessons
- Prepared for next stage

Landing Page

- Simplified / Refactored
- Better Entry Points
  - Site search
  - A-Z index
  - Tag/Topic
- Contribution Guides



CRYPTO DONE RIGHT!

Contents About Contact

DEVELOPERS' QUICKSTARTS

IT PROFESSIONALS' QUICKSTARTS

MANAGERS' QUICKSTARTS

Share and contribute your expertise on Crypto Done Right

GETTING STARTED

TIPS & TUTORIALS

FEEDBACK

Updated Layout & Legends

- Additional banners
- Standardization/consistency
- Better visual cues

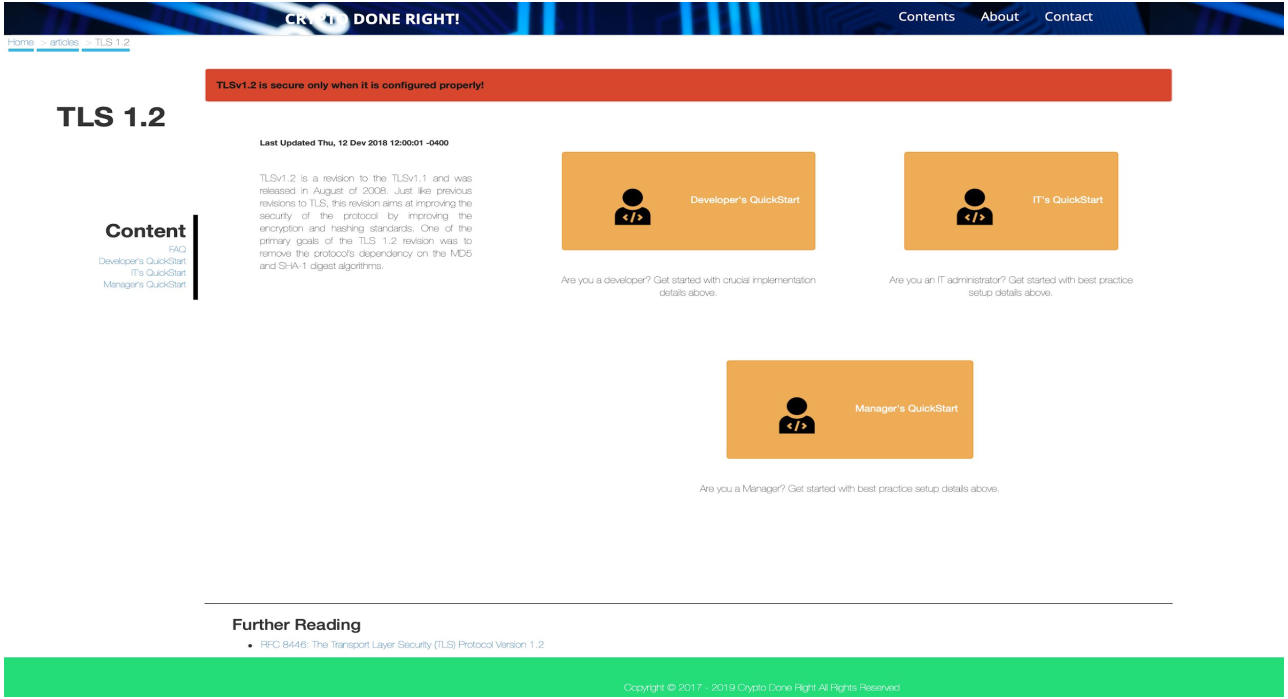
Implementation

Configuration

Upgrade/Patch Management

Protocol

Polished Layout



TLSv1.2 is secure only when it is configured properly!

TLS 1.2

Last Updated Thu, 12 Dev 2018 12:00:01 -0400

TLSv1.2 is a revision to the TLSv1.1 and was released in August of 2008. Just like previous revisions to TLS, this revision aims at improving the security of the protocol by improving the encryption and hashing standards. One of the primary goals of the TLS 1.2 revision was to remove the protocol's dependency on the MD5 and SHA-1 digest algorithms.

Content

FAQ
Developer's QuickStart
IT's QuickStart
Manager's QuickStart

Are you a developer? Get started with crucial implementation details above.

Are you an IT administrator? Get started with best practice setup details above.

Are you a Manager? Get started with best practice setup details above.

Further Reading

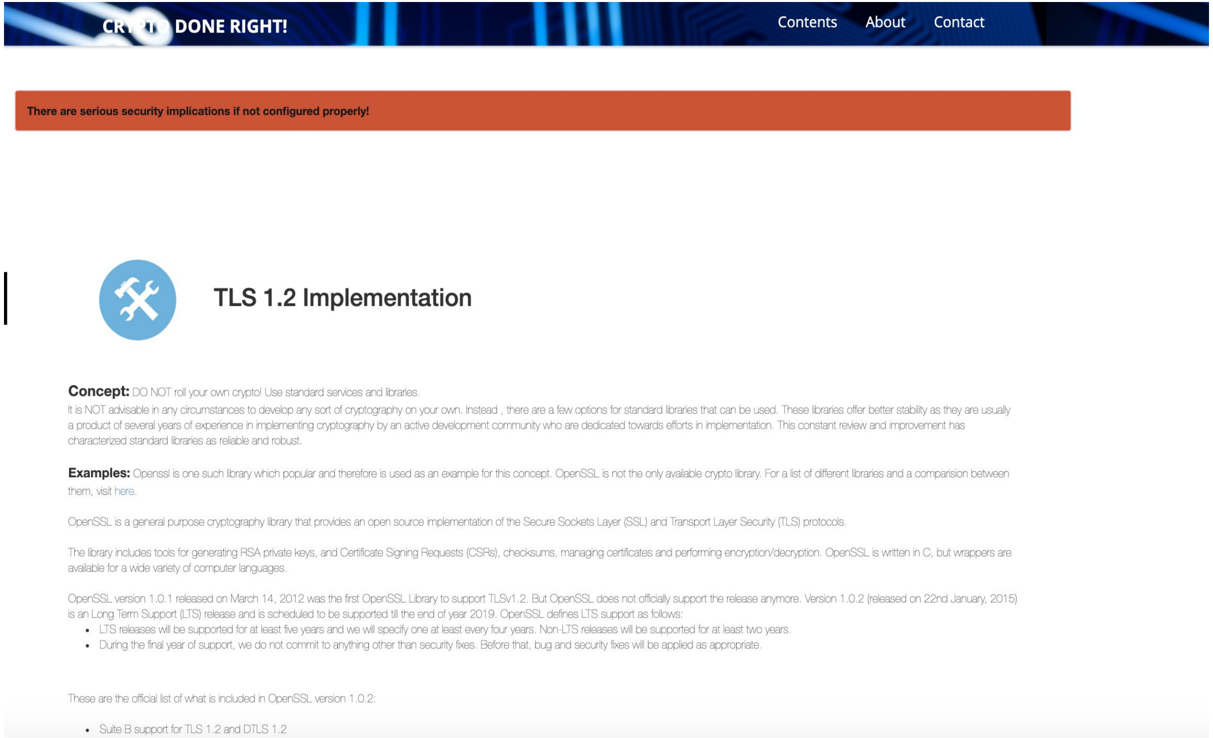
- RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.2

Quick Starts - Developer

- One-stop place for all developers to upgrade and gain knowledge of their systems

- Guidelines for immediate patching of system

- Code is provided for easy implementation



There are serious security implications if not configured properly!

TLS 1.2 Implementation

Concept: DO NOT roll your own crypto! Use standard services and libraries.
It is NOT advisable in any circumstances to develop any sort of cryptography on your own. Instead , there are a few options for standard libraries that can be used. These libraries offer better stability as they are usually a product of several years of experience in implementing cryptography by an active development community who are dedicated towards efforts in implementation. This constant review and improvement has characterized standard libraries as reliable and robust.

Examples: Openssl is one such library which popular and therefore is used as an example for this concept. OpenSSL is not the only available crypto library. For a list of different libraries and a comparision between them, visit here.

OpenSSL is a general purpose cryptography library that provides an open source implementation of the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols.

The library includes tools for generating RSA private keys, and Certificate Signing Requests (CSRs), checksums, managing certificates and performing encryption/decryption. OpenSSL is written in C, but wrappers are available for a wide variety of computer languages.

OpenSSL version 1.0.1 released on March 14, 2012 was the first OpenSSL Library to support TLSv1.2. But OpenSSL does not officially support the release anymore. Version 1.0.2 (released on 22nd January, 2015) is an Long Term Support (LTS) release and is scheduled to be supported till the end of year 2019. OpenSSL defines LTS support as follows:

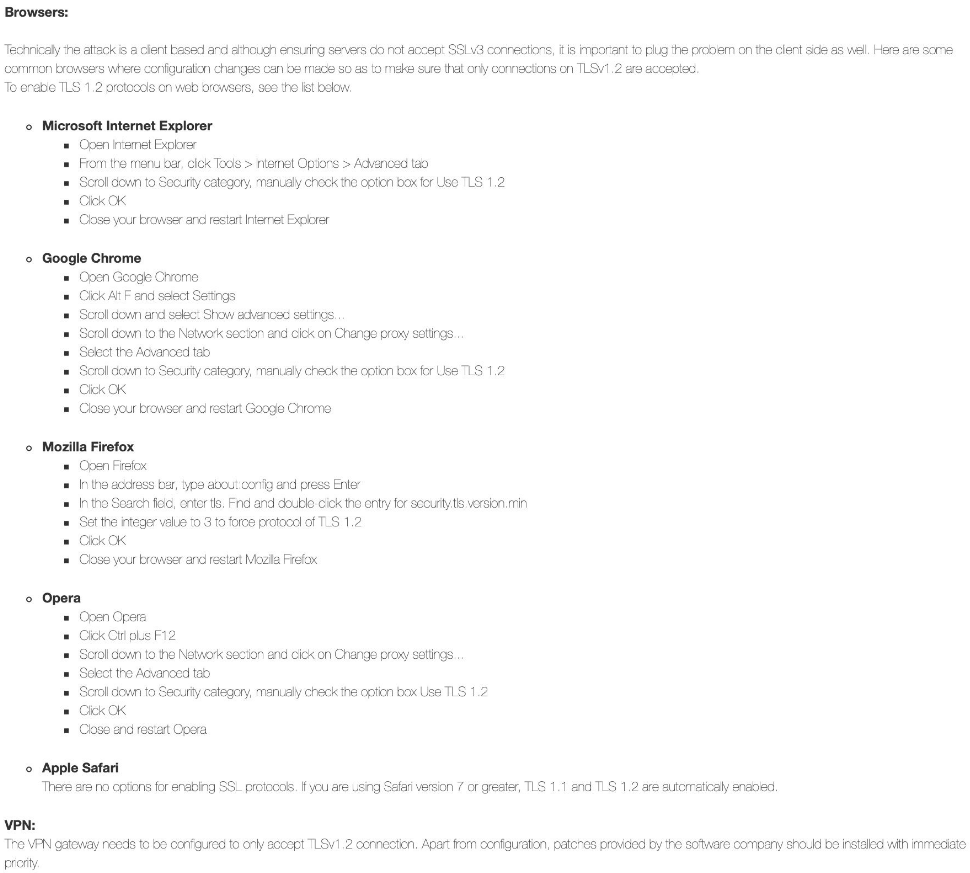
- LTS releases will be supported for at least five years and we will specify one at least every four years. Non-LTS releases will be supported for at least two years.
- During the final year of support, we do not commit to anything other than security fixes. Before that, bug and security fixes will be applied as appropriate.

These are the official list of what is included in OpenSSL version 1.0.2:

- Suite B support for TLS 1.2 and DTLS 1.2

Quick Starts - IT

- Information is provided on the acceptable Web servers and Browsers
- Patch management and upgrade information if applicable
- Lists of recommended tools and configurations are provided



Browsers:

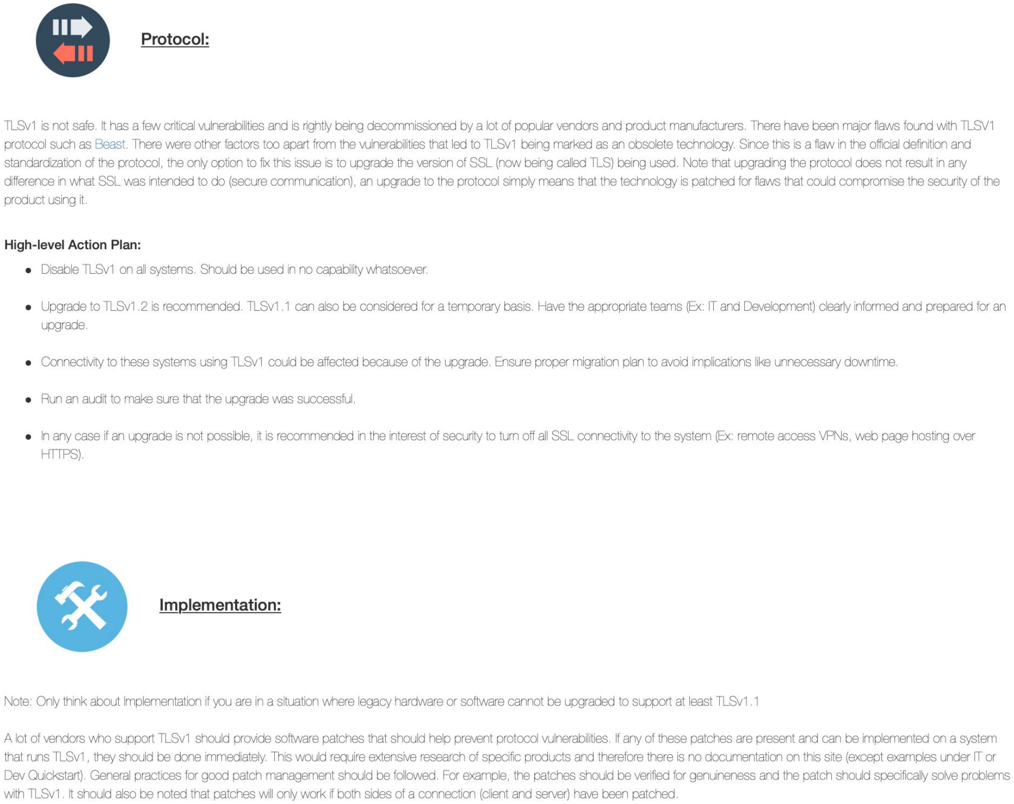
Technically the attack is a client based and although ensuring servers do not accept SSLv3 connections, it is important to plug the problem on the client side as well. Here are some common browsers where configuration changes can be made so as to make sure that only connections on TLSv1.2 are accepted.
To enable TLS 1.2 protocols on web browsers, see the list below.

- **Microsoft Internet Explorer**
  - Open Internet Explorer
  - From the menu bar, click Tools > Internet Options > Advanced tab
  - Scroll down to Security category, manually check the option box for Use TLS 1.2
  - Click OK
  - Close your browser and restart Internet Explorer
- **Google Chrome**
  - Open Google Chrome
  - Click Alt F and select Settings
  - Scroll down and select Show advanced settings...
  - Scroll down to the Network section and click on Change proxy settings...
  - Select the Advanced tab
  - Scroll down to Security category, manually check the option box for Use TLS 1.2
  - Click OK
  - Close your browser and restart Google Chrome
- **Mozilla Firefox**
  - Open Firefox
  - In the address bar, type about:config and press Enter
  - In the Search field, enter tls. Find and double-click the entry for security.tls.version.min
  - Set the integer value to 3 to force protocol of TLS 1.2
  - Click OK
  - Close your browser and restart Mozilla Firefox
- **Opera**
  - Open Opera
  - Click Ctrl plus F12
  - Scroll down to the Network section and click on Change proxy settings...
  - Select the Advanced tab
  - Scroll down to Security category, manually check the option box Use TLS 1.2
  - Click OK
  - Close and restart Opera
- **Apple Safari**
  There are no options for enabling SSL protocols. If you are using Safari version 7 or greater, TLS 1.1 and TLS 1.2 are automatically enabled.

VPN:
The VPN gateway needs to be configured to only accept TLSv1.2 connection. Apart from configuration, patches provided by the software company should be installed with immediate priority.

Quick Starts - Manager

- Managers must know the high-level action plan
- Overview of the techniques that must be used to ensure security
- Protocol, implementation, and configuration of the relevant topic is available



Protocol:

TLSv1 is not safe. It has a few critical vulnerabilities and is rightly being decommissioned by a lot of popular vendors and product manufacturers. There have been major flaws found with TLSV1 protocol such as Beast. There were other factors too apart from the vulnerabilities that led to TLSv1 being marked as an obsolete technology. Since this is a flaw in the official definition and standardization of the protocol, the only option to fix this issue is to upgrade the version of SSL (now being called TLS) being used. Note that upgrading the protocol does not result in any difference in what SSL was intended to do (secure communication), an upgrade to the protocol simply means that the technology is patched for flaws that could compromise the security of the product using it.

High-level Action Plan:

- Disable TLSv1 on all systems. Should be used in no capability whatsoever.
- Upgrade to TLSv1.2 is recommended. TLSv1.1 can also be considered for a temporary basis. Have the appropriate teams (Ex: IT and Development) clearly informed and prepared for an upgrade.
- Connectivity to these systems using TLSv1 could be affected because of the upgrade. Ensure proper migration plan to avoid implications like unnecessary downtime.
- Run an audit to make sure that the upgrade was successful.
- In any case if an upgrade is not possible, it is recommended in the interest of security to turn off all SSL connectivity to the system (Ex: remote access VPNs, web page hosting over HTTPS).

Implementation:

Note: Only think about Implementation if you are in a situation where legacy hardware or software cannot be upgraded to support at least TLSv1.1

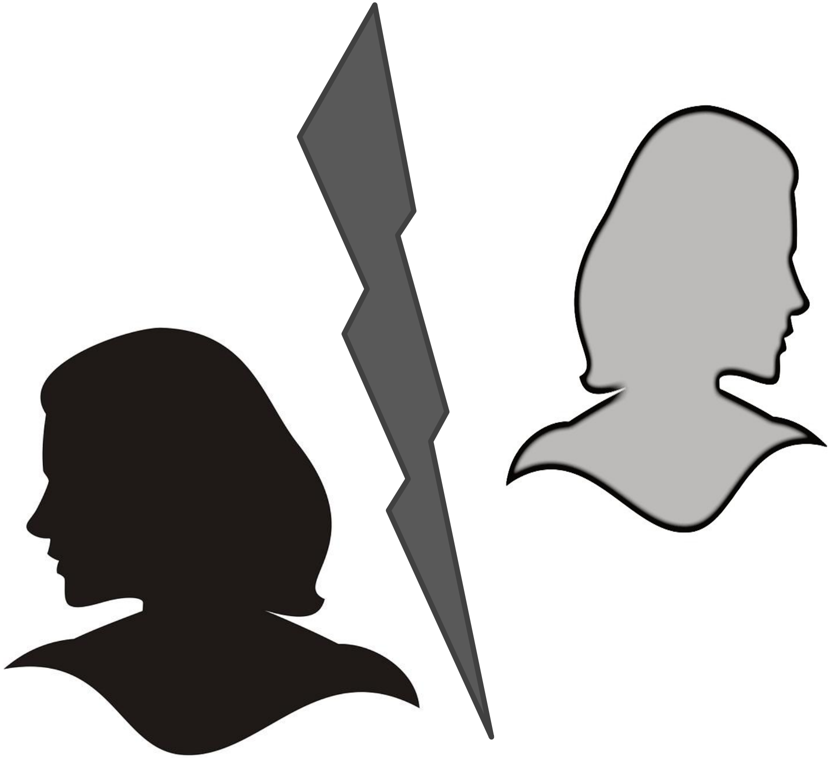
A lot of vendors who support TLSv1 should provide software patches that should help prevent protocol vulnerabilities. If any of these patches are present and can be implemented on a system that runs TLSv1, they should be done immediately. This would require extensive research of specific products and therefore there is no documentation on this site (except examples under IT or Dev Quickstart). General practices for good patch management should be followed. For example, the patches should be verified for genuineness and the patch should specifically solve problems with TLSv1. It should also be noted that patches will only work if both sides of a connection (client and server) have been patched.

The REAL Update: Behind the Scenes

- Site updates are relatively minor
- Greater insight from collaboration

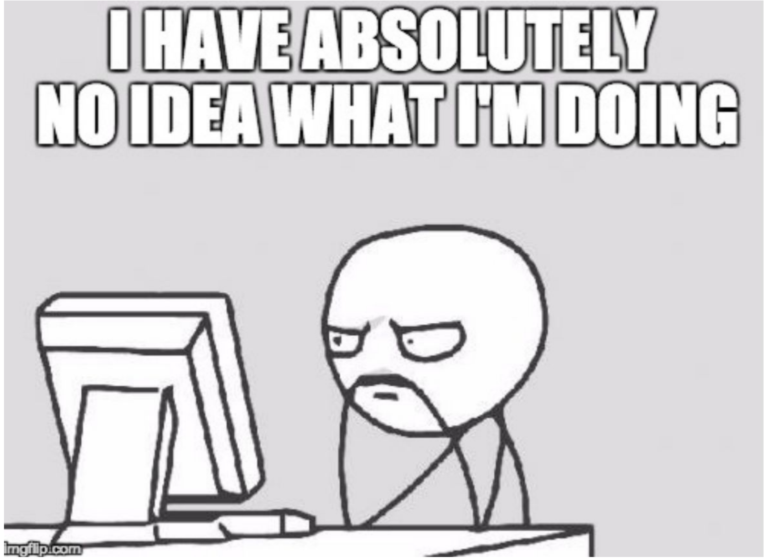
Lesson #1:
The Divide is REAL

- Cryptographers vs Users
- Example:
  “Why do you talk about DES?”



Lesson #2:
Usability Studies

- Attempted to recreate Usenix study
- Failed to find a similar population
- Received helpful feedback (+/-)



“I HAVE NO IDEA WHAT I’M DOING” – ON THE USABILITY OF DEPLOYING HTTPS

Katharina Krombholz, Wilfried Mayer, Martin Schmiedecker, Edgar Weippl

Lesson #3: Collaboration Challenges

- Initial collaboration struggled
- *Don’t Roll Your Own!*

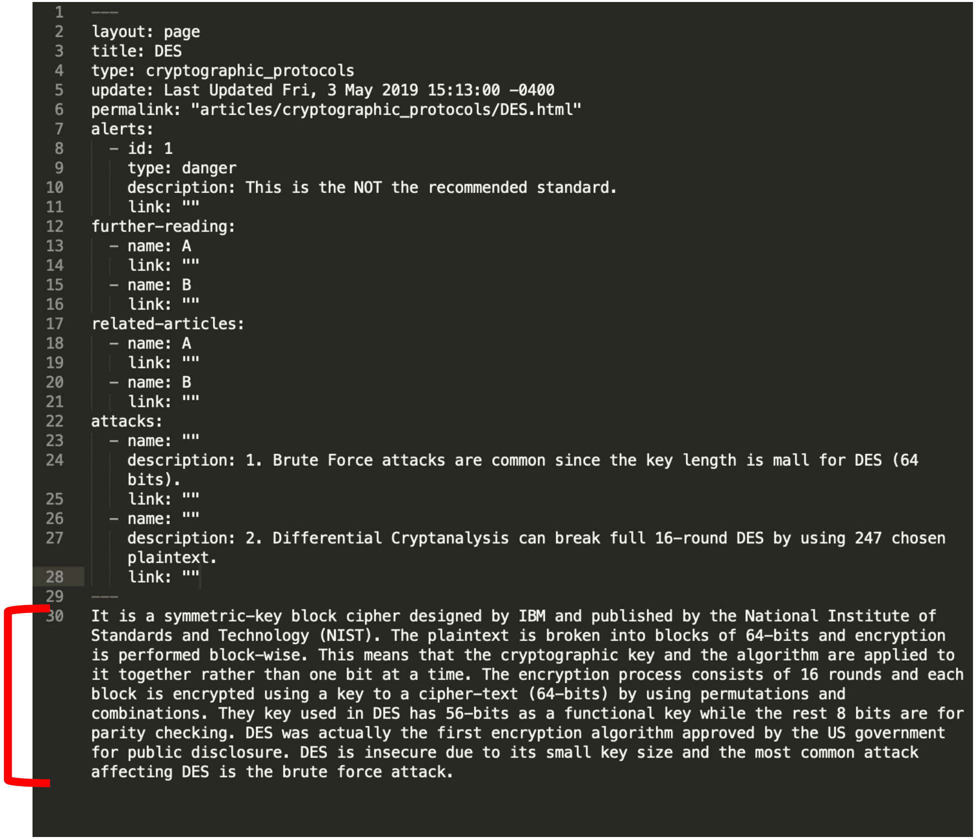
Putting it Together: Collaboration v2

- All standard communications tools
- Git manages change controls
- Jekyl for page development

Using Jekyll

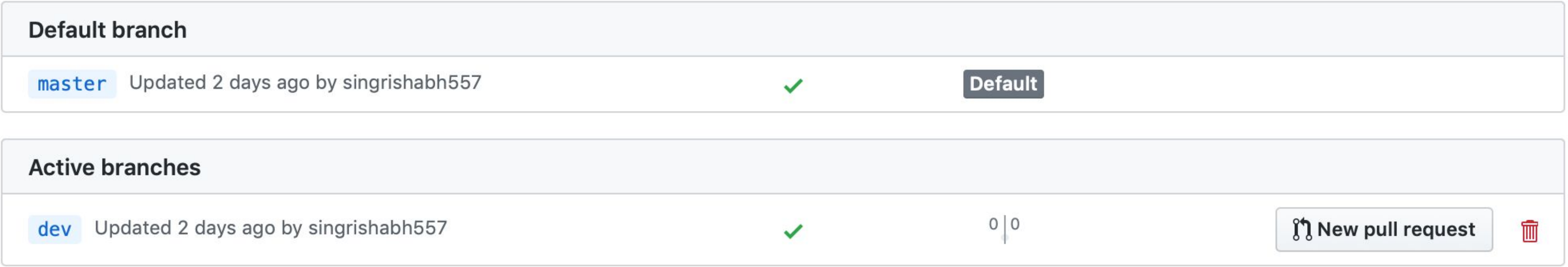
WHY ARE WE USING IT?

- Static site generator for creating webpages
- Renders markdown files
- Easy integrations
- Widely used with Github



```
---
layout: page
title: DES
type: cryptographic_protocols
update: Last Updated Fri, 3 May 2019 15:13:00 -0400
permalink: "articles/cryptographic_protocols/DES.html"
alerts:
  - id: 1
    type: danger
    description: This is the NOT the recommended standard.
    link: ""
further-reading:
  - name: A
    link: ""
  - name: B
    link: ""
related-articles:
  - name: A
    link: ""
  - name: B
    link: ""
attacks:
  - name: ""
    description: 1. Brute Force attacks are common since the key length is mall for DES (64 bits).
    link: ""
  - name: ""
    description: 2. Differential Cryptanalysis can break full 16-round DES by using 247 chosen plaintext.
    link: ""
---
It is a symmetric-key block cipher designed by IBM and published by the National Institute of Standards and Technology (NIST). The plaintext is broken into blocks of 64-bits and encryption is performed block-wise. This means that the cryptographic key and the algorithm are applied to it together rather than one bit at a time. The encryption process consists of 16 rounds and each block is encrypted using a key to a cipher-text (64-bits) by using permutations and combinations. They key used in DES has 56-bits as a functional key while the rest 8 bits are for parity checking. DES was actually the first encryption algorithm approved by the US government for public disclosure. DES is insecure due to its small key size and the most common attack affecting DES is the brute force attack.
```

Branches



Default branch

master Updated 2 days ago by singrishabh557 ✓

Active branches

dev Updated 2 days ago by singrishabh557 ✓

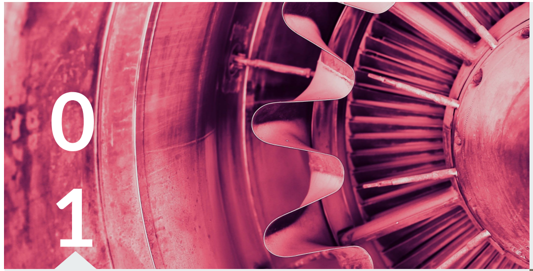
“Master”

- Default branch for deployment
- Changes made on development branch don’t affect the master branch

“Dev”

- Used for testing in production before merging to master.
- Can roll it back by deploying the existing master into production if any new change causes issues

Flow



Create Branch

You're creating an environment where you can try out new ideas.

Add commits >> open pull/issue request >> Discuss/Review

From your environment to development branch

Deploy >> Merge

Once changes have been verified in development branch, it is time to merge your code into the master branch!

Collaboration Guidelines!

Contributing Guideline

Provides instructions on how to properly contribute:

- Branches
- Templates
- Review & deploy process
- Ground rules

Pull Request Template

Standards for a pull request:

- Description
- Pull request type
- Changes made
- Related issue
- How has it been tested or verified

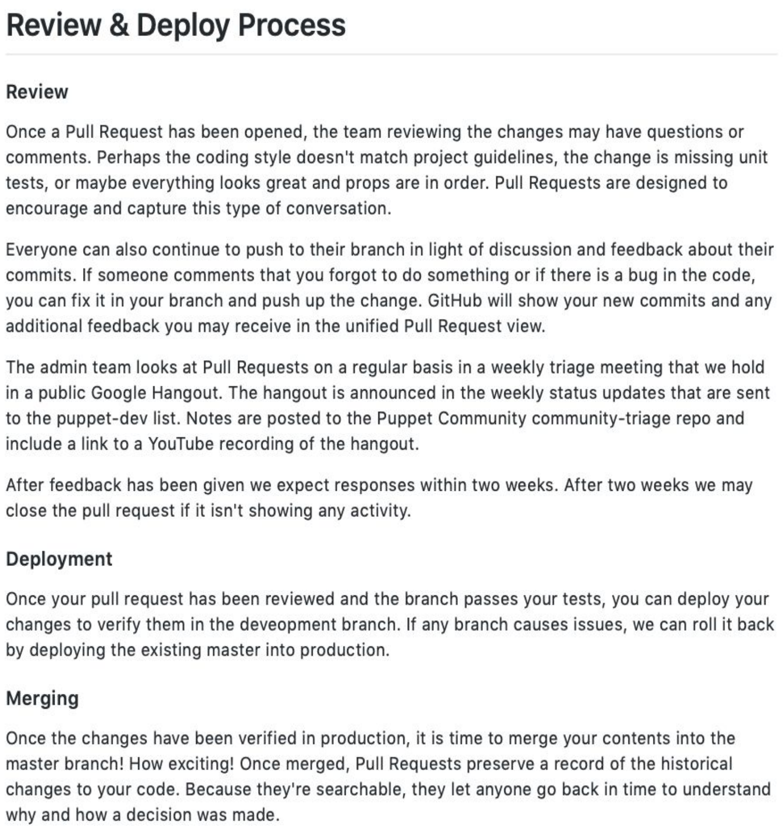
Issue Template

Standards for an issue report:

- Types of issues
- Environment (i.e., browser, os)
- Descriptions and steps to reproduce the issue

Contributing Template

- Walkthrough of how to submit a contribution
  - fork and commit
  - Review, deployment, and merging
- Proper markdown files to use for pull request and issue
- Expected behaviors from all of us



Review & Deploy Process

Review

Once a Pull Request has been opened, the team reviewing the changes may have questions or comments. Perhaps the coding style doesn't match project guidelines, the change is missing unit tests, or maybe everything looks great and props are in order. Pull Requests are designed to encourage and capture this type of conversation.

Everyone can also continue to push to their branch in light of discussion and feedback about their commits. If someone comments that you forgot to do something or if there is a bug in the code, you can fix it in your branch and push up the change. GitHub will show your new commits and any additional feedback you may receive in the unified Pull Request view.

The admin team looks at Pull Requests on a regular basis in a weekly triage meeting that we hold in a public Google Hangout. The hangout is announced in the weekly status updates that are sent to the puppet-dev list. Notes are posted to the Puppet Community community-triage repo and include a link to a YouTube recording of the hangout.

After feedback has been given we expect responses within two weeks. After two weeks we may close the pull request if it isn't showing any activity.

Deployment

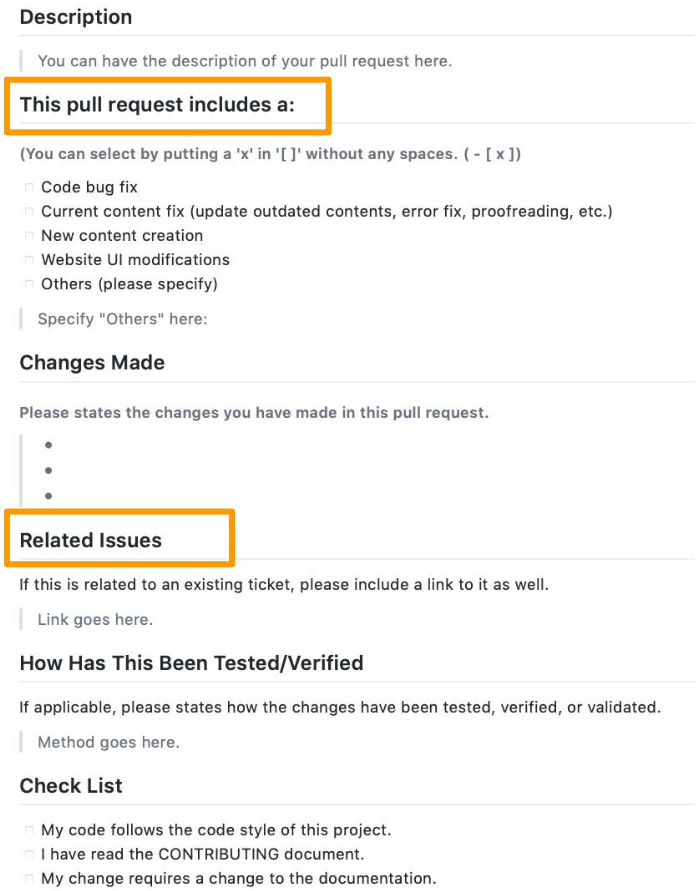
Once your pull request has been reviewed and the branch passes your tests, you can deploy your changes to verify them in the deveopment branch. If any branch causes issues, we can roll it back by deploying the existing master into production.

Merging

Once the changes have been verified in production, it is time to merge your contents into the master branch! How exciting! Once merged, Pull Requests preserve a record of the historical changes to your code. Because they're searchable, they let anyone go back in time to understand why and how a decision was made.

Pull Request Template

- Standardize what should be included in a pull request.
- Initiate at any point during the development process.
- Track changes



Description

> You can have the description of your pull request here.

This pull request includes a:

(You can select by putting a 'x' in '[ ]' without any spaces. (- [x])

- □ Code bug fix
- □ Current content fix (update outdated contents, error fix, proofreading, etc.)
- □ New content creation
- □ Website UI modifications
- □ Others (please specify)

> Specify "Others" here:

Changes Made

Please states the changes you have made in this pull request.

> -
> -
> -

Related Issues

If this is related to an existing ticket, please include a link to it as well.

> Link goes here.

How Has This Been Tested/Verified

If applicable, please states how the changes have been tested, verified, or validated.

> Method goes here.

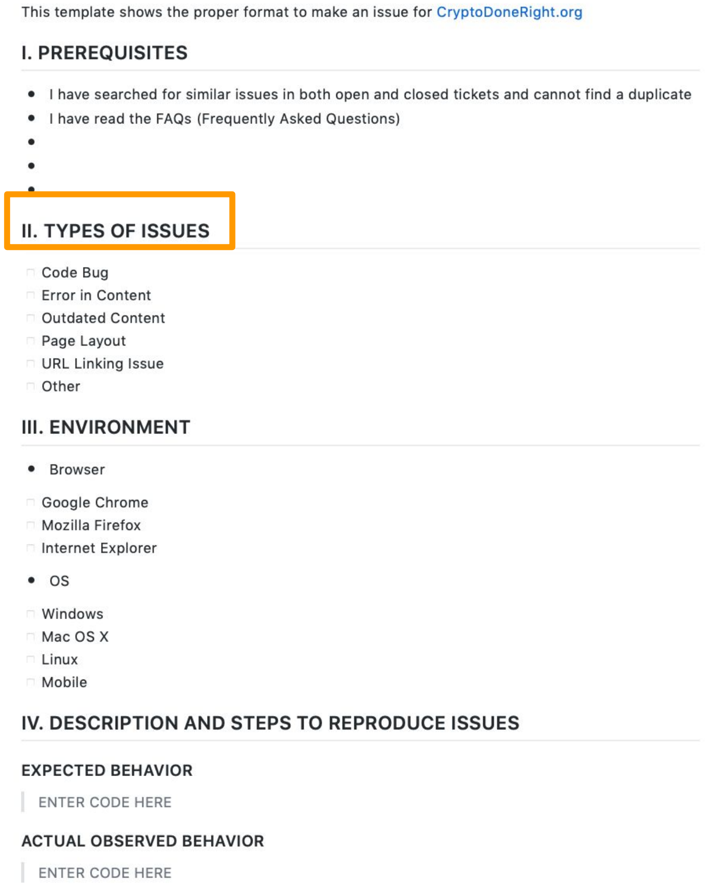
Check List

- □ My code follows the code style of this project.
- □ I have read the CONTRIBUTING document.
- □ My change requires a change to the documentation.

Issue Template

With issues, we can:

- Reference other issues or pull requests (so that your issue automatically closes when you merge a pull request)
- Pin important issues to make them easier to find, preventing duplicate issues and reducing noise.
- Report comments that violate GitHub's Community Guidelines.



This template shows the proper format to make an issue for CryptoDoneRight.org

I. PREREQUISITES

- I have searched for similar issues in both open and closed tickets and cannot find a duplicate
- I have read the FAQs (Frequently Asked Questions)
-
-
-

II. TYPES OF ISSUES

□ Code Bug
□ Error in Content
□ Outdated Content
□ Page Layout
□ URL Linking Issue
□ Other

III. ENVIRONMENT

- Browser

□ Google Chrome
□ Mozilla Firefox
□ Internet Explorer

- OS

□ Windows
□ Mac OS X
□ Linux
□ Mobile

IV. DESCRIPTION AND STEPS TO REPRODUCE ISSUES

EXPECTED BEHAVIOR

> ENTER CODE HERE

ACTUAL OBSERVED BEHAVIOR

> ENTER CODE HERE

Pages Under Construction (Coming Soon!)

Feel free to contribute to the pages that are ‘under construction’ with a Pull Request!

- **Topics that are currently under construction :**
  - Certification
  - AES Manager Quickstart
  - TLS 1.3 Manager Quickstart
  - SHA3 Landing Page, Dev QS, IT QS, Manager QS

The Future of Crypto Done Right

- Spin off from JHU
- Independent non profit
- Nexus of community, industry, academia partnership

Join our Community

- Contribute content
- Corporate sponsors
- Spread the word