The Great MFA Distribution Plan

Agenda:
https://docs.google.com/document/d/1K4zD0_rmMp3VHG2kJVz-UBSFDxgbKrEUNy6Gf-_WknA/edit#

Other doc:
https://docs.google.com/document/d/17-kbHo8RY-G9cTUHqlar749c1SPuoGRjRTZqJYAYazo/edit#

This is a draft plan to improve OSS security by increasing the use of MFA (multi-factor) tokens used by OSS developers. The OpenSSF has been offered a number of MFA keys (in thousands) to distribute to OSS developers. In the future we may have more keys to distribute, so while we'll probably start by focusing on the keys we've already been offered, we don't need to restrict ourselves to just that.

MFA tokens are not "the" solution - they are a way to harden against attacks on stolen passwords. They won't apply in all circumstances. However, they can help in many circumstances.

Our first meeting was Sep 28 URL = https://meet.jit.si/mfa-distribution-planning-session

We need to work out:

- Timeline - Google token coupon codes expire 2021-12-31, need to get them out before then. Here's the phase I timeline:
    - 2021-12-02: Critical Projects WG provides a list of ~100 critical projects (there are always more that could be added, we just need *a* list).
    - 2021-12-02: Great MFA Project completes first draft of documentation
        - How to get token of each type
        - How to verify token
        - How to install
        - How to use in common circumstances
        - Ideally, tries them out on a willing guinea pig project
    - 2021-12-03..09: Critical projects notified (typically via a GitHub issue) that they're eligible by our great-mfa-plan notifiers (John Naulty, David A. Wheeler, CRob, etc.). See the "invitation.md" file for the invitation.
    - 12-02..31: When a project accepts, the notifier will tell a sender (David A. Wheeler or Jory Burson) where to send the coupon codes (email address), what project, and a copy of how they found out (for our records). The sender will send the coupon codes using the "coupon_sending.md" template.
    - 2021-12-31: Tokens distributed (at least all Google tokens)
- Who to distribute keys to?

    - Current plan is to use a list being developed by the OpenSSF Critical Project Working Group of ~100 critical projects. They're using the Preliminary Census II report (from Harvard), criticality score, and other data to identify the list, then having humans review it to determine what's really critical (using the data for insight)
    - We'll use repo URLs to identify projects (otherwise it's not clear what a name means)
    - In the longer-term we probably want to distribute keys at some conferences - which ones? That's been pushed off to a future phase.
- We need some simple rules (requirements to get a free MFA token), e.g.:
        - Must be a developer of OSS
            - David A. Wheeler has instead proposed maintainer or contributor of the critical project OR anything it depends on (transitively)
        - Must commit to *trying* to use an MFA key when developing OSS
        - A token must not be reused between different people (a token has 1 owner/user)
        - Users will consider providing feedback (so we can fix problems or highlight what's working well)
- How to distribute keys (we need to give them confidence that the keys are not subverted!)
    - OpenSSF will not distribute keys directly. Instead, we'll distribute coupon codes (Google) or validation codes (GitHub). Receivers will "buy" them at no cost from Google Store / GitHub Shop.
    - Google: will give out "coupon codes" that allow a dev to get a token. Dev goes to a google site, enters code and gets token
    - GitHub will do something similar.
    - GitHub is ok to give away 500 coupon codes to buy these tokens.
        - GitHub provides the token and its shipping
        - International shipping except in the countries under sanctions (China, Afghanistan, Russia, Ukraine, North Korea, Iran, Sudan, Syria)
        - GitHub needs to know what coupons have been distributed, to report back which ones have been actually used to buy the token
    - Please note that any organization sending tokens from the US can't send them to certain countries due to sanctions https://home.treasury.gov/policy-issues/financial-sanctions/sanctions-programs-and-country-information
        - @team find out international shipping considerations [where can these be sent?]
- How can receivers verify the key on receipt?
    - Non-Profit Education group 'Hashbang' creates content on verifying yubikeys: https://book.hashbang.sh/docs/security/personal-hsms/yubikey/#verifying-authenticity
- Need a communication plan - make it clear this is legitimate

    - Wiorking Draft - https://docs.google.com/document/d/1cdZAXBB9_E3T2QrKQOfmgSI5heqwF9mUwQV3qynLwOc/edit#
    - Press release
    - Blog
    - twitter/social plan
        - Seed tweets
        - Retweeters
        - Other social (tiktok-like vids?)
- Need documentation on why it's secure ("this is not the big supply chain attack")
    - Need more details on distribution process to write this
    - We're only giving out coupon codes, the distribution is handled separately.
- EASY to apply guidelines/tutorials/best practices for using them in common uses for OSS.
    - High-quality 'one-stop shop' for documentation
        - Thought through the use-cases for maintainers
        - Highlighting caution for 'where things can go wrong'
    - Must be SHORT and SIMPLE
    - E.g., ((commit to GitHub|GitLab|?)|(release package on repo)) using my platform (Ubuntu|Fedora|MacOS|Windows)
    - Where possible, point to existing good tutorials
        - John Naulty: The most exhaustive guide I've come across is http://github.com/drduh/YubiKey-Guide - I refer to this almost anytime I refresh a workshop/docs on Yubikey setup for Linux/Mac.
    - Make things as easy as possible (to get started & to use actively)
    - Make some videos - for some steps it's easiest to SHOW what to do
    - Who'd be willing?
        - CRob (also comm plan) <crob@intel.com>
        - David A. Wheeler
        - John Naulty
        - Arnaud J Le Hors <lehors@us.ibm.com>
    - OpenSSF creates and curates a 'Securing Supply Chains with Hardware Tokens' repository or directory in an existing repository
        - Plan to start with pages within the Best Practices WG repo
        - this will serve as the landing page for anyone interested in securing their supply chains with a hardware token (e.g., someone who just received a free Yubikey from OpenSSF)
    - provide FAQ or some type of checklist to guide people through the various defenses a hardware token can provide
        - using hardware token for 2FA for 'critical' accounts (github, e-mail used for package manager accounts like npm, pypi, etc)
        - commit signing
        - release signing (e.g. with sigstore)
        - for each item, provide

            - a simple page describing how attaching this action to a hardware token improves the supply chain security
            - official links to Yubikey docs (if any exist)
            - instructions to perform operation for Windows/Mac/Linux
            - Link to OpenSSF Youtube video performing a screen recording of the steps
            - Additional community resources/references
        - provide a doc/section describing the vision
            - a rising tide lifts all boats, and it's useful to describe and share how using hardware tokens can protect a lot of 'boats' in the opensource sea.
            - definitely give some praise to Github + Google for donating these tokens and explicitly share their intent/cause for doing so
            - describe how the initial yubikey recipients were chosen and how the process for the token giveaway looks like
- Marketing? (How to announce publicly the project as well as the selection of the first 'round' of recipients?)
    - Twitter memes? (sign_all_the_thingz...with hardware!)
    - OpenSSF blog / LF blog
- Need to beta test the process with a few people first
    - Make sure we get the (expected) tokens, can use them using the guidance we create, etc.

David A. Wheeler proposes working this as a task within the best practices WG as the "lead working group" & using its mailing list. However, this task crosses multiple WG areas, so we really need to coordinate with:

- the Digital Identity Attestation WG (hi!) [to ensure we get confidence]
- Critical projects WG (ID some projects we really want to use tokens)
- Best Practices WG (to create best practices)

TAC agreed.

Once we've worked out the plan (this document), we'll assign people to parts (aka beg for volunteers), and start executing.

What could go wrong?

- How to make sure we are giving the key to the ACTUAL developer of a repository?
    - Less important if it goes to the wrong developer.
- How do we help developers _verify_ that they are receiving a legitimate key?
    - What could go wrong? How can we defend against that?
    - Counter subverted tokens being sent by using coupon codes
    - Can't become a helpdesk for MFA for everyone worldwide. It's okay if we help people we specifically send tokens to, especially the first ones

Methods/Medium of collaboration?
- open up issues on github

- Use openssf-best-practices-wg mailing list
- Should we spin up our own little ‘working group’ tactically focused on this project?

- Best Practices WG meetings/mailing list

Repo names:
“great-mfa-project” was most-liked
+1 +1 +1 +1

Use cases

- You just got a token! What now? (OS differences, device differences)
- How to use token to log in and sign git commits: given token type, forge (GitHub, GitLab), platform
- How do I use it to execute a release
- What if it is lost or broken? Recovery mechanism / path

XKCD has a comic about 2FA (of course): https://xkcd.com/2522/

Connecting People with areas that need development

- Selection Problem
    - David Wheeler
        - At the least, use Alpha-Omega (at least Alpha) & the top projects identified by the Harvard Census II preliminary study
- Content + Curation Problem
    - John Naulty
- Jennifer Fernick: connecting project to security researchers if needed; helping writing guidance document on MFA to OSS devs/maintainers; writing OpenSSF blog post we can use to initially communicate to devs about the project; helping with the early work to lay out a number of Github issues in our repo to break out the project into a number of pieces
- Appu Goundan - Github automation?/titan keys stuff (figuring out if they can be 'verified')
- John Fontana (Yubico) - Distribution

People who plan to participate in the Great MFA Distribution:

- David A. Wheeler
- John Naulty
- Glenn Ten Cate
- CRob
- Marta R.
- Sergio Rojas
- Xavier Rene-Corail

Marta R. reports that GitLab does let you query if a user uses MFA.

Plan is to discuss MFA distribution at next Best Practices WG meeting,

On 2021-11-18 Xavier René-Corail (GitHub) sent the following:

I will be off for our next meeting, so I paste here the latest from GitHub on the Great MFA, specifically on the problem of measuring success vs. privacy concerns.

But my colleague Jose Palafox should be in the meeting to discuss

- People think that OSSF collecting GitHub handles and sending them to GitHub for explicit inquiry purposes will be very complicated from a legal perspective
- The counter-proposal is that instead of sending you coupon codes, we send you a link to a form. Then the developer fills out the form, with their GitHub handle. They get the coupon, and we can track if they bought the token, and if they used it
- This way we can tell you how many coupons were sent (you can compare this to the number of emails you sent), how many tokens were actually bought, how many of these users actually activated MFA. We'll give you just the numbers.
- GitHub legal will still need to get their consent to get and use their GitHub handle, but at least there is no transfer of private info from ossf to gh, which simplifies a lot

I think that gives us a nice and detailed success measure for the campaign: # of mails sent (from ossf) > # of coupons sent > # of token shipped > # of account activating MFA.

The form would contain:

Information about the program (TODO: copy to be provided by ossf)
Legal text to get consent for using the data collected below
input: email
input: GitHub handle
Input: verification code (GitHub provided the 500 verification codes)

Legal text: *Upon completing this form you will be e-mailed a coupon code valid at shop.github.com for (1) Yubikey. By requesting a coupon for a security device you consent to allowing GitHub to verify whether you have enabled MFA/2FA by checking your GitHub account settings. GitHub will check your account to verify MFA/2FA settings every 30 days for 1 year starting (12/3/21) and may send periodic automated emails during this period to help you configure your security token. The purpose of verifying MFA/2FA settings is to gauge the effectiveness of programs meant to encourage MFA/2FA adoption and report MFA/2FA utilization among high priority community members generally. You are not obligated to use the security token or to enable MFA/2FA on your GitHub account. User MFA/2FA settings will only be reported in the aggregate and not at the individual user level. You may*

opt out of the periodic emails at any time and you will not be enrolled in any long term mailing lists.