# 👋 Dasharo User Group #10 🎉 and Developers vPub 0xF 🍻

# Enhancements in Dasharo CI: Automatic DBX and microcode refresh

# Agenda

- Introduction
- DBX and microcode
- Automations
- Closing thoughts

# Introduction


## Michał Kopeć

*Firmware Engineer*

🔑 `869E 9AE8 AFDB 5FAE 6068 338B 99BD 2EEE E2D0 CE31`

✉️ michal.kopec@3mdeb.com

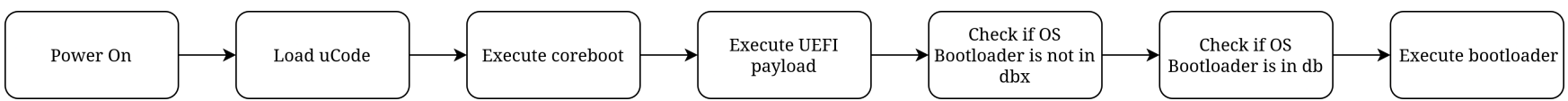
🔗 LinkedIn

💻 GitHub


- Firmware Engineer working primarily with coreboot and EDK2 but also Heads and Linux
- Have been at 3mdeb for 4 years now
- Free and open source software enthusiast
- ThinkPad collector

# Key Security Components & Trust

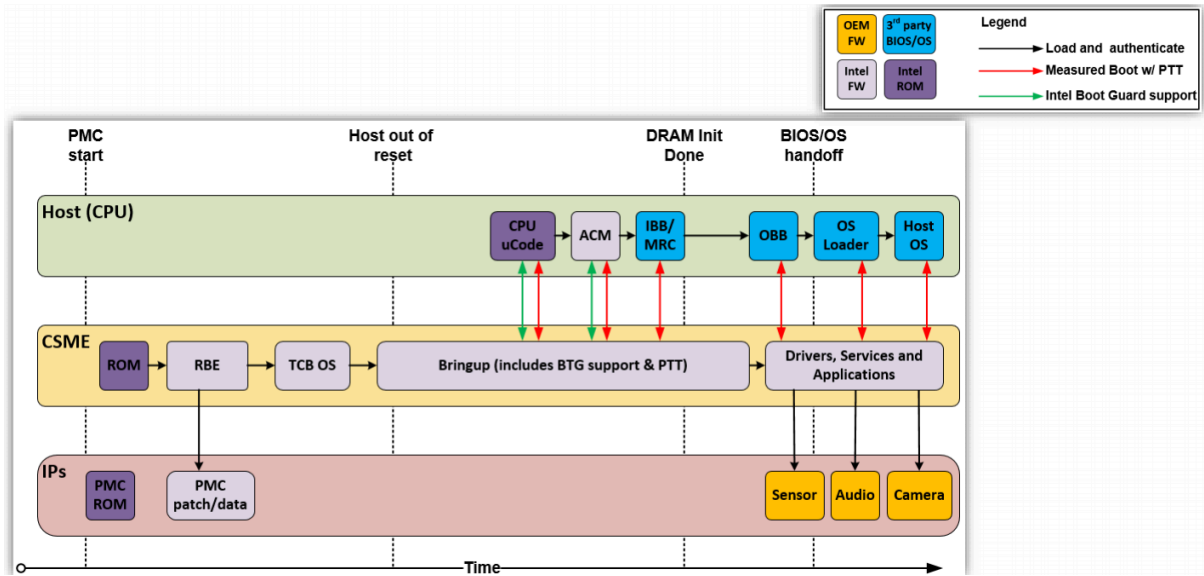
- **Microcode Updates:**
  - Processor-level fixes/mitigations (e.g., Spectre, Meltdown).
  - Applied very early by coreboot.
- **UEFI DBX (Revocation Database):**
  - Part of UEFI Secure Boot.
  - Revokes compromised binaries (e.g., BootHole exploit) and signing certificates.
- **Firmware Chain of Trust:**
  - Microcode -> coreboot -> UEFI -> Bootloader -> OS.
  - Each stage verifies the next.



# Our Goal: Consistent & Timely Security

- Ensure critical components (Microcode, DBX) are **always** up-to-date.
- Reduce risk of oversight.
- Free up developer resources.
- **Solution:** Automation!

# The Intel boot process




- PMC starts executing -> CSME loads code from ROM -> CSME executes boot extensions and begins bringup -> Releases CPU from reset
- CPU loads ucode and ACMs (optional) from FIT table -> BtG ACM verifies BIOS -> BIOS is allowed to execute

# Component Deep Dive: Intel Microcode

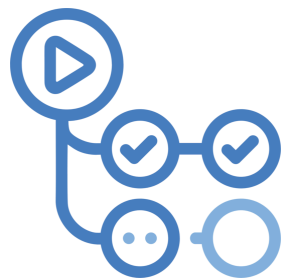
- **Source:** Intel's GitHub
- **Purpose:** Fix security advisories, functional issues, and errata.
  - Often essential for basic CPU operation.
- **Loading:**
  - By coreboot via Firmware Interface Table (FIT) – *before coreboot itself runs!*
  - OS can load updates, but firmware-level is earlier & more comprehensive.
- **Example Vulnerability Patched:**
  - **INTEL-SA-01139:** Privilege escalation via UEFI module input validation.

# Component Deep Dive: UEFI DBX

- **Source:** UEFI Forum / Microsoft GitHub
  - We use Microsoft's Git repo for easier automation & prompt updates.
  - Discrepancy highlights need for reliable, scriptable sources.
- **Purpose:** Revokes signatures of previously approved firmware/software.
  - Critical for mitigating bootkits (e.g., BlackLotus) and compromised bootloaders.
  - Revoking certificates is powerful: invalidates many binaries with one update.
- **Example Vulnerability Mitigated:**
  - **CVE-2022-21894 (BlackLotus):** Persistent UEFI bootkit bypassing Secure Boot.

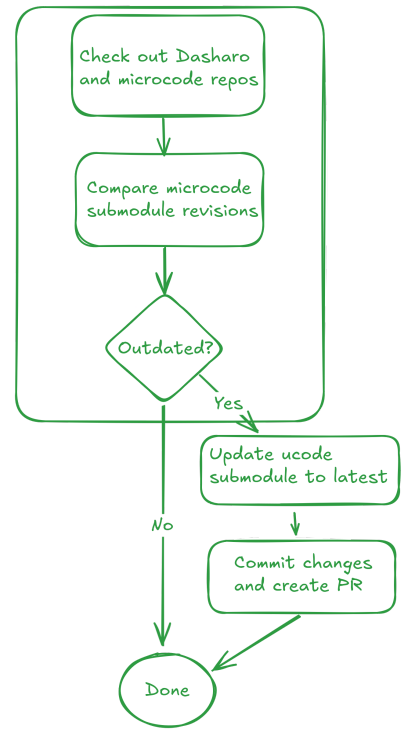
# The Automation: GitHub Actions

- Why GitHub Actions?
  - Tight integration with our GitHub codebase.
  - Declarative YAML syntax.
  - Wide range of community-supported actions (e.g., for creating PRs).
- Daily checks for updates to Microcode and DBX.



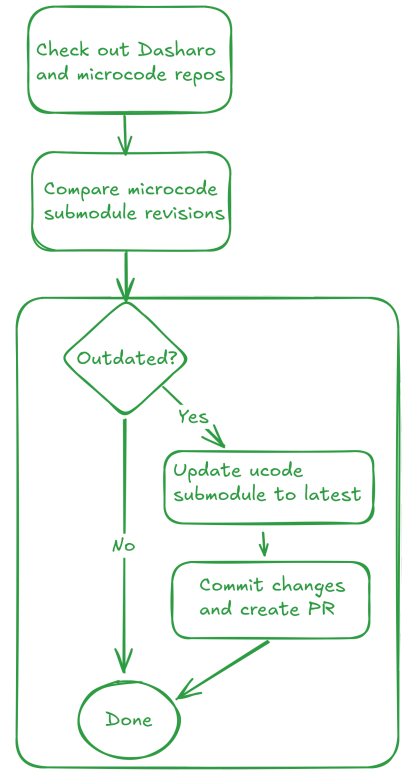
# Automation: Microcode Update Workflow

```
# Microcode Check Snippet
- name: Check if µcode submodule is up to date
  run: |
    git submodule update --init --checkout 3rdparty/intel-microcode
    pushd 3rdparty/intel-microcode
    current=<span class="math-inline">\(git log \-1 \-\-pretty\=format\:"%H"\)
git checkout main \# Switch to main to get the latest
new\=</span>(git log -1 --pretty=format:"%H")
    if [[ $current == $new ]]; then
      echo "Intel µcode submodule is up-to-date."
    else
      echo "Intel µcode submodule is out of date!"
      exit 1
    fi
    popd
```



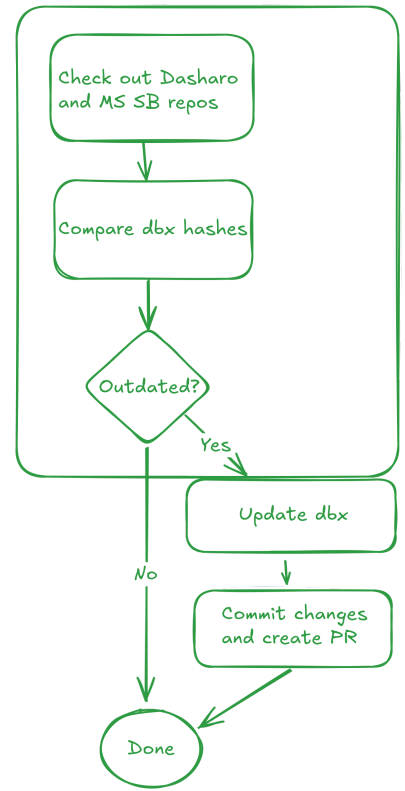
# Automation: Microcode Update Workflow (Cont.)

- Checkout Dasharo code.
- Update intel-microcode submodule to its main branch.
- Use peter-evans/create-pull-request action to submit a PR.
- Design Rationale: Git submodule pins specific versions & tracks changes transparently.



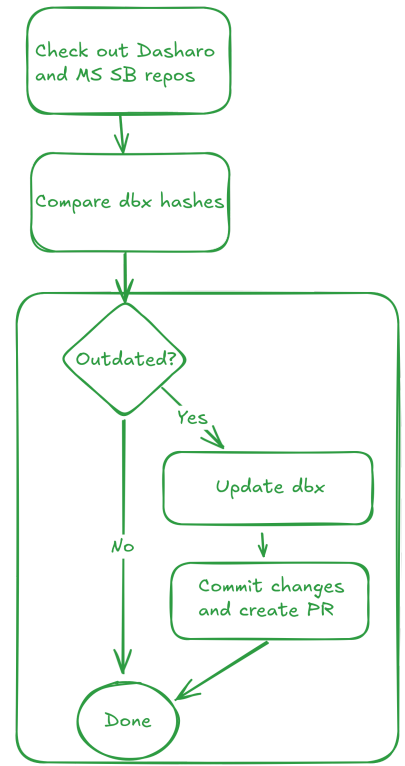
# Automation: UEFI DBX Update Workflow

```
# DBX Check Snippet
- name: Check if DBX is out-of-date
  run: |
    old=$(sha256sum edk2/DasharoPayloadPkg/SecureBootDefaultKeys/DBXUpdate.bin
    | awk '{ print <span class="math-inline">1 \}'\)
    new=(sha256sum secureboot_objects/PostSignedObjects/DBX/amd64/DBXUpdate.bin
    | awk '{ print $1 }')
    if [ "$old" = "$new" ]; then
      echo 'UEFI DBX is up-to-date.'
    else
      echo 'UEFI DBX is out of date.'
      exit 1
    fi
```



# Automation: UEFI DBX Update Workflow (Cont.)

- Copy the new DBXUpdate.bin from secureboot_objects into Dasharo edk2 tree.
- Use peter-evans/create-pull-request action to submit a PR.
- Design Rationale: Checksum comparison is straightforward for single-file artifacts.



# Human Oversight: The PR Review Process

- Automated PRs are not automatically merged!
- Standard Dasharo Review Process:
  - Developers review changes.
  - Ensure correct integration.
  - Test on relevant hardware platforms.
- Result: Balances timeliness with stability.

# Benefits: Security & Transparency

- Enhanced Security:
  - Regular, automated updates minimize vulnerability windows.
- Increased Transparency:
  - Open repositories and CI workflows.
  - Users can see update status and history.
  - Community can audit processes and adapt methods.
- Contrast: Many proprietary firmware solutions are opaque about update contents and schedules.