In case you missed it, Downfall, the newest massive Intel bug, was publicly announced on Aug 8, 2023 and presented at Blackhat by Daniel Moghmi (sidechannels expert at Google Research).

How is the TCB Recovery process going so far? What's different sinceAepicLeak last year?

Downfall affects SGX like we'll focus on here, but more broadly its biggest impact is that on an unpatched machine, even unprivileged malware or malicious tenants on a multitenant machine could steal data and keys. In terms of SGX, this is pretty much the same situation as AepicLeak, with the researchers able to demonstrate breaching sealing keys. In fact the main difference is a much larger class of processors is affected, including many Xeon processors in cloud environments. The public disclosure came after a nearly 12-month embargo, so this was known even while we were still dealing with AepicLeak. We can't easily tell whether it was exploited in the wild before (e.g., if it were independently discovered by hackers).

Downfall and TCB Recovery

Just like with AepicLeak, above all it's important to understand the TCB Recovery process. TCB Recovery roughly means revoking sensitive data access to affected processors until they can re-attest that they have updated microcode. Projects relying on TEE remote attestation need to take an action based on this disclosure

.

One improvement since last time is that Intel has made the data necessary to carry out a TCB Recovery, called the "Verification Collateral," available on the date of public disclosure, through a new "update=early"

flag in their API. (See Intel's page onTCB Recovery in Q3 2023) Although there is an overall TCB Recovery date scheduled for later in the year, meaning that Intel Attestation Service (IAS) will enforce this on every new attestation, TEE projects don't have to wait. The "Verification Collateral" is basically a database of all the Intel processor types, and the current list of advisories affecting which microcode versions. In other words, if you are maintaining an SGX application, you need to check attestations using the updated VC database, not the old one. You can fetch individual records from Intel's API here, though to download the entire database at once requires an account. For example, this "tcbInfo" object is for one Xeon processor type:

https://api.trustedservices.intel.com/sgx/certification/v4/tcb?fmspc=00606a000000&update=early

To explain a few salient parts of a tcbInfo

record (abridged below), the fsmpc

is the processor type, each tcb

corresponds to one microcode configuration for this processor, identified by svn

and pcesvn

which are 17 total "Security Version Numbers" (each may get incremented with new security critical microcode release) associated with a configuration. The tcbLevels

lists the configurations currently thought to be secure.

{"tcbInfo": { ... "fmspc": "00606a000000", ... "tcbLevels": [{ "tcb": { "sgxtcbcomponents": [{"svn":12,...},...12,3,3,255,255,1,0,0,..], "pcesvn": 13 }, "tcbDate": "2023-08-09T00:00:00Z", ... }, ...]}, "signature": "b8b67e06.....7869b8f"}

So, after a disclosure like, projects are supposed to fetch new TCB records, and when checking a remote attestation should compare against the current list of acceptable configurations. Finally note that this comes with a signature from Intel.

Improving transparency of Verification Collateral

TLDR: we are missing an independent mirror of the processor status database (the "verification collateral") that Intel publishes.

Although making Verification Collateral updates available at the time of public disclosure is great, we noticed two anomalies when trying to follow along:

- Briefly, for different case strings of the same processor type (a hex number), one would provide the updated value, the other would provide the old insecure value. https://api.trustedservices.intel.com/sgx/certification/v4/tcb?fmspc=00606A000000&update=early
- We tried to use internet archive to take a snapshot, but some http parameters are incompatible, so this didn't work.

These are simple issues, but they highlight a transparency gap where we (the TEE-based decentralized blockchain communities) could provide improvement. This is a familiar pattern like certificate transparency. The tcbInfo

records already come with digital signatures from Intel, so in principle we could hold them accountable for errors we find or if they sign inconsistent versions. However, if an individual node used this service to automatically fetch a tcbInfo

record, unless they go out of their way to store this for later and compare with others at some point, there would be no way to notice if an invalid record was provided.

The general goal should be to provide a mirroring / data availability / transparency service for the Verification Collateral databases published by Intel. A crude option is just to place this Verification Collateral on-chain or otherwise in some data availability layer, and timestamp it.

Some actionable ideas that we could do is:

1. Short term: We might try to provide a minimal blockchain mirroring service for the Verification Collateral. Maybe ipfs is a sufficient structure to use. What else?

2. Medium term: In SUAVE we might look at how to automatically apply such updates without active intervention from blockchain developers needed.

Thoughts on these?

Acknowledgements: thanks Jernej Kos (Oasis) and Wentao Xiao (Clique) for explaining several aspects of verification collateral and tcbinfo records to me. The writing here reflects my best understanding and attempt to be readable, any misuse of terminology remains my fault.