TwizSec Library Crate

Surendra Jammishetti

Twizzler

1 Early Goals

The TwizSec crate aims to provide an external library for the Twizzler kernel that has the following goals (summarized from me and Daniels meeting on 11/26/24).

1. storing and receiving capabilities
2. signing and verifying capabilities
3. programming the mmu / io to reflect security policy data

2 Implementation

2.1 Needs

The plan is to work on the second item first, as its the path of least resistance. Ideally expose two functions.

1. Takens in capability and signature, returns if they are correct or not
2. Given a capability, construct a signature

2.2 Deps

The kernel has crypto libraries already integrated, use those to build these features currently this is all we got
p256 : https://crates.io/crates/p256
sha2 : https://crates.io/crates/sha2
Which, atleast right now, should have everything we need.

2.3 Capabilites

Currently we dont have a capability struct, so Im going to use what was in the security paper as an example.

Additionally I'm considering making the two functions impl'd onto the struct, so that way they can be called on any capability struct, as I think it would be nice and ergonomic but not sure what others would think.

This is the spec inside the paper

```
CAP := {
    target, accessor : ObjectId,
    permissions, flags : BitField, // bitfield of permissions
    gates: Gates, // going to not have this
    revocation : Revoc, // not going to have this
    siglen: Length, // not really sure what "Length" means here but im assuming a u16 should be good en
    sig: u8[], // signature
}
```