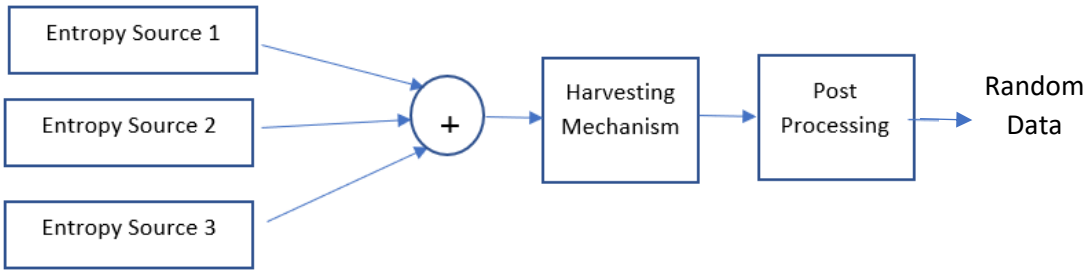
**Vulnerability**: Exploitation of harvesting mechanism may lead to less randomly generated number.

- **Description:** The randomness of a True Random Number Generator (TRNG) generated numbers depend on three components- 1) Entropy Source, 2) Harvesting mechanisms, and 3) Post-processing. The entropy source is the most critical component as it determines the available entropy. Even if entropy sources may exhibit biases: these should be eliminated during harvesting and post-processing steps [1]. Otherwise, various applications of TRNG, such as- key generation, chip manufacturing, authentication, nonce generation, etc., can suffer from different side-channel and fault injection attacks due to less randomly generated numbers. However, the harvesting mechanism and post-processor may possess biasness if they are controllable by a test and debug infrastructure. Then, the TRNG generated number will be less random than expected.




**Related Security Property:** Debug interface should not have access to the control registers associated with harvesting mechanism of a TRNG.

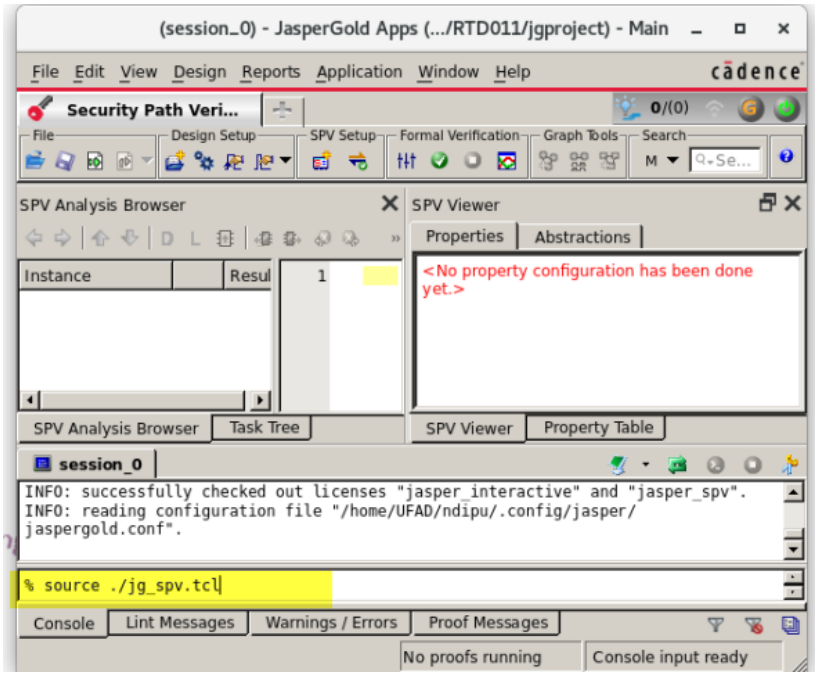
- It should be checked in design time, whether the FSM control registers of harvesting mechanism and post-processor of TRNG can be accessed from debug access port or not.

**Design:**
The Cryptech TRNG is a hybrid design with entropy providers connected to physical entropy sources are used to seed a cryptographically safe pseudo random number generator (CSPRNG). In order to combine the entropy from the providers, the TRNG contains a mixer stage (Harvesting mechanism + Post Processor) between the providers and the CSPRNG. Besides the three stages of the datapath, the TRNG contains a control part that provides the functionality needed to test and debug the TRNG in a secure manner, even in a running system.
RTL codes for the CryptoTech TRNG are included in ./src folder.

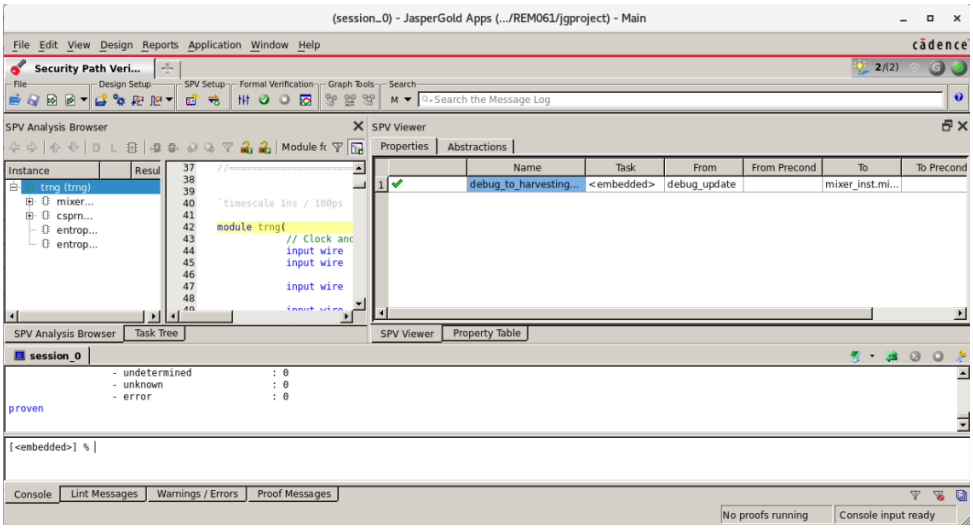
**Converted Assertion:**
```
check_spv -create -from {debug_update} -to {mixer_inst.mixer_ctrl_reg} -name
"debug_to_harvesting_mixer"
```

**Which tool to use:** Cadence JasperGold Security Path Verification (SPV).

**How to use:** Use “jg -spv” command to invoke JaspaerGold SPV. Utilize “jg_spv.tcl” file to check the assertion.
[N.B: Utilize “jg_spv_21.tcl” file to check the assertion if you are using JasperGold 2021.06 FCS release.]




**Result:**
For the current design of the TRNG, the property is proven, indicating that the harvesting mechanism is not vulnerable to provide less randomness during random number generation.



Reference:

1. Sunar, Berk, William J. Martin, and Douglas R. Stinson. "A provably secure true random number generator with built-in tolerance to active attacks." IEEE Transactions on computers 56.1 (2006): 109-119.