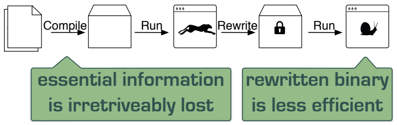
ENCORE: Enhanced program protection through Compiler-Rewriter cooperation

Challenge:

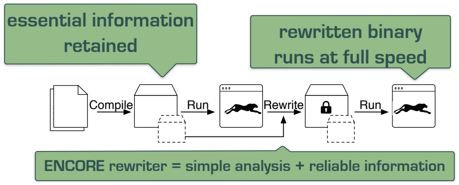
- Compilers discard all **structural information** from programs: speed and size is all that matters.
- Binary rewriters try to recover structural information using **complex analysis plus unreliable guesses.**
- Rewritten programs take up more space and run slower than their original counterparts.

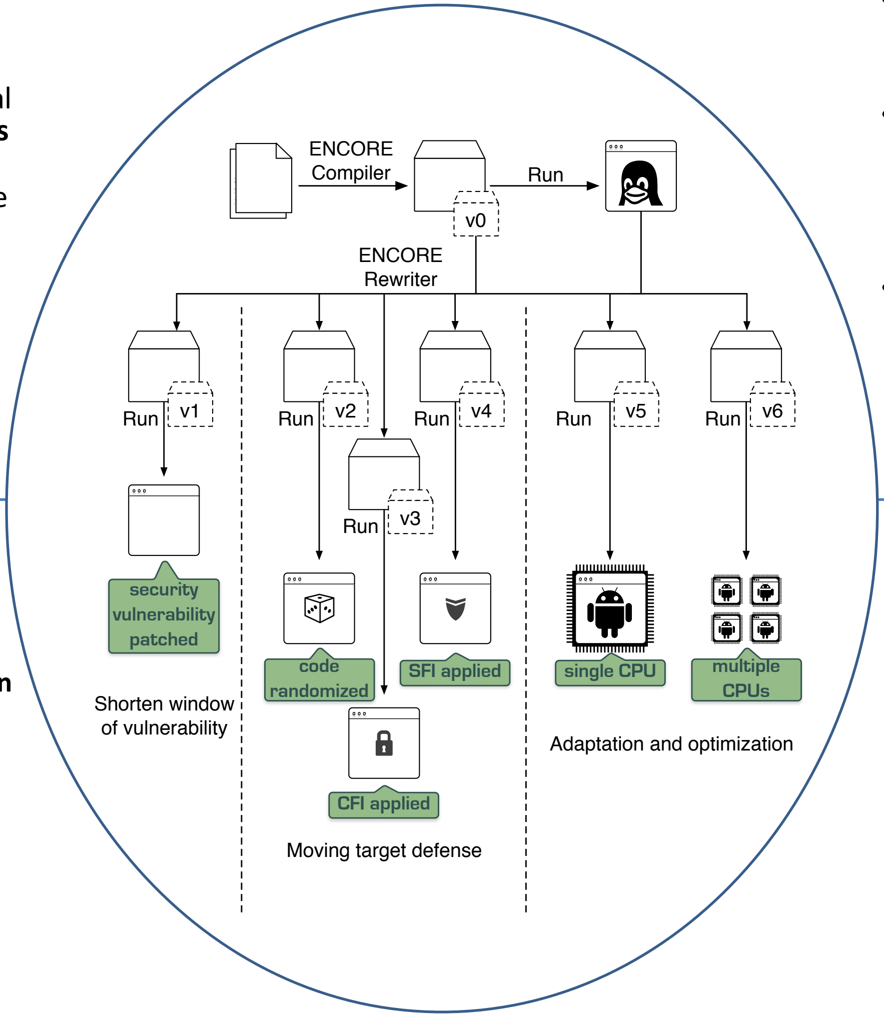

Compile
Run
Rewrite
Run
essential information is irretrievably lost
rewritten binary is less efficient

Solution:

Change the way compilers have been constructed during the past sixty-odd years:

- Compiler **retains structural information.**
- Simple binary analysis identifies **information that can be reliably recovered**.
- Residual "hard to recover" **information is embedded in output** ENCORE binary.


essential information retained
rewritten binary runs at full speed
Compile
Run
Rewrite
Run
ENCORE rewriter = simple analysis + reliable information



NSF Award **CNS-1513837**

Lead PI: Michael Franz (UC Irvine)

Co-PIs: Kevin Hamlen (UT Dallas), Mathias Payer (Purdue University)

PM: Sol J. Greenspan

Scientific Impact:

- Enables consumers of software to fix vulnerable programs themselves rather than having to wait for vendor's fix.
- In 2014, the top 5 **zero-day vulnerabilities took 59 days to patch** on average because of slow vendor response time. Total window of vulnerability = 295 days.
- Compiler supported client-side binary rewriting **closes or shortens the window of vulnerability and presents adversaries with a moving target.**

Broader Impact:

- ✓ Lessen the impact of zero-day vulnerabilities.
- ✓ Shorten time to deployment for new defenses.
- ✓ Support consumer-driven security policies.
- ✓ Increase robustness of cyber infrastructure.
- Project has a strong educational component.