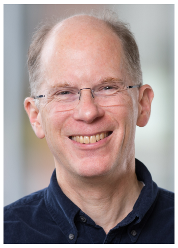
Specifying and Verifying Secure Compilation of C Code to Tagged Hardware

PI: Andrew Tolmach (tolmach@pdx.edu)
RAs: Sean Anderson, C.H.R. Chhak

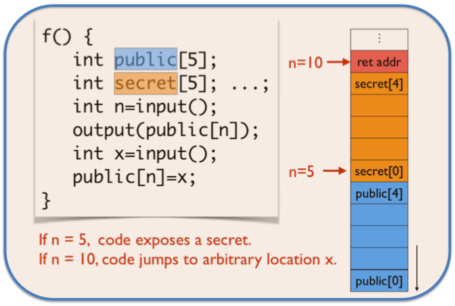
Portland State University



http://web.cecs.pdx.edu/~apt/satc_pi/

Goal: A provably secure platform for legacy C code

C Undefined Behaviors

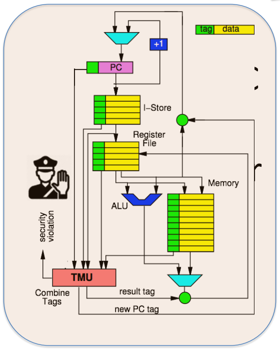


```
f() {
   int public[5];
   int secret[5]; ...;
   int n=input();
   output(public[n]);
   int x=input();
   public[n]=x;
}
```

If n = 5, code exposes a secret.
If n = 10, code jumps to arbitrary location x.

1 Many security attacks exploit C undefined behaviors (UBs), especially buffer overflows. Software-based mitigations hurt performance.

2 New CPU hardware enhanced with configurable support for instruction-level metadata tagging can efficiently monitor against security faults, including UBs.



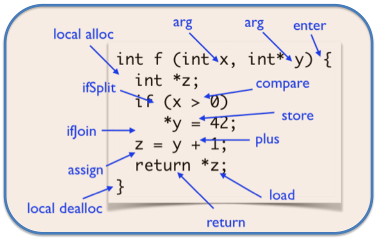
Tags might be memory regions, types, security levels, etc.

CPU with tag support

3 But how can we ensure that tagged hardware is used *correctly* to achieve source-level security goals? We need a flexible way to specify C-level policies and a highly reliable way to field them on tagged hardware.

Approach: Formal specification of C security properties and formal compiler verification

Tagged C



```
int f (int x, int* y) {
  int *z;
  if (x > 0)
     *y = 42;
  z = y + 1;
  return *z;
}
```

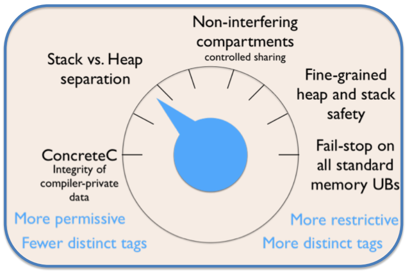
Control points

4 C security policies are expressed at source level, using Tagged C, a novel tag-aware C semantics and tag policy language. Tagged C attaches tags to variables, functions, etc. and checks them at control points.

Tagged C Verified Compiler

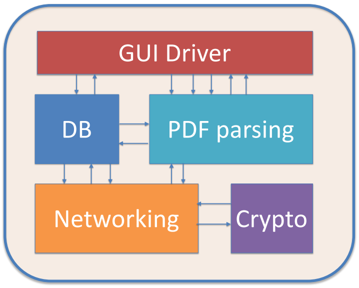
5 The compiler is part of the TCB, so we mechanically verify its correctness, using the Coq proof assistant.

6 Tagged C has no UBs and lets the user pick a level of memory safety that supports legacy idioms and gives good performance on tagged hardware.


Non-interfering compartments
controlled sharing
Stack vs. Heap separation
Fine-grained heap and stack safety
ConcreteC
Integrity of compiler-private data
Fail-stop on all standard memory UBs
More permissive
Fewer distinct tags
More restrictive
More distinct tags

Flexible Memory Safety

Compartmentalization



7 Tagged C can also be used to enforce higher-level security properties, such as compartmentalization in support of least privilege.

The 5th NSF Secure and Trustworthy Cyberspace Principal Investigator Meeting (2022 SaTC PI Meeting)
June 1-2, 2022 I Arlington, Virginia