Engineering Code Obfuscation

ISSISP 2017 - Tamperproofing

Christian Collberg

Department of Computer Science
University of Arizona

http://collberg.cs.arizona.edu

collberg@gmail.com

Supported by NSF grants 1525820 and 1318955 and
by the private foundation that shall not be named

What is Tamperproofing?

Bob wants to modify the program binary so that it does something different than we want:

- remove functionality (license check)
- change data (password, cryptographic key)
- add functionality (print, save game)

Tamperproofing the code makes it stop working if Bob changes as little as a byte of the binary!

Tamperproofing has to do two things:

1. detect tampering
2. respond to tampering

Essentially:

```
if (tampering-detected())
    respond-to-tampering()
```

but this is too unstealthy!

```
int foo() {
   … … … …
}
```

Detect tampering

```
int main () {
   if (foo-has-changed-in-any-way())
      foo();
}
```

- ***crash the program***
- ***phone home***
- ***refuse to run***
- ***run slower***
- ***make wrong results***

Respond to tampering

```
int hash (addr_t addr,int words){
   int h = *addr;
   for(int i=1; i<words; i++) {
        addr++;
        h ^= *addr;
    }
    return h;
}
```

```
int foo() {
  … … … …
}
```

Detect tampering

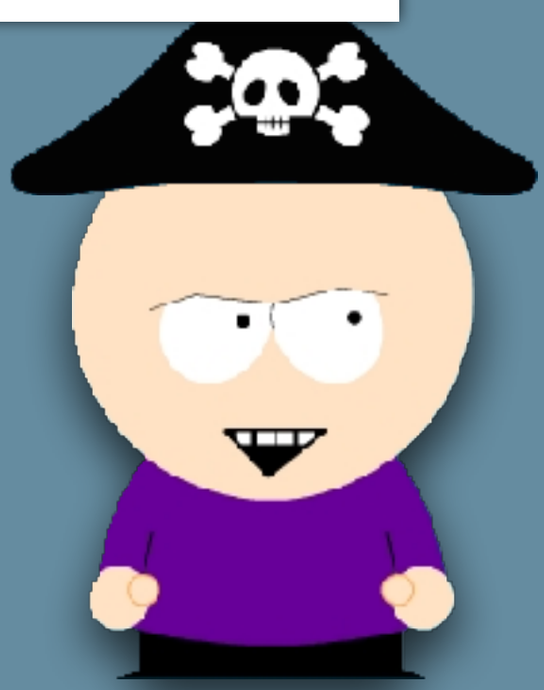
```
int main () {
    if (hash(foo,1000) != 0x4C49F346)
        foo();
}
```

- ***crash the program***
- ***phone home***
- ***refuse to run***
- ***run slower***
- ***make wrong results***

Respond to tampering

```
int foo () {
    if (today > “Aug 17,2016”){
      printf(“License expired!”);
      abort;
    }
}
```

```
check(){
 if (hash(foo)!=42)
   abort()
}
```



```
int foo() {
   … … … …
}
```

```
int foo_copy() {
   … … … …
}
```

Repair foo!!!

Checker 1

```
if (foo-has-changed-in-any-way())
```

copy

```
int foo_copy() {
  … … … …
}
```

```
int foo() {
  … … … …
}
```

Repair Checker 1!

Checker 2

```
if (foo-checker1-changed())
```

copy

```
Checker1_copy
```

```
Checker1
```

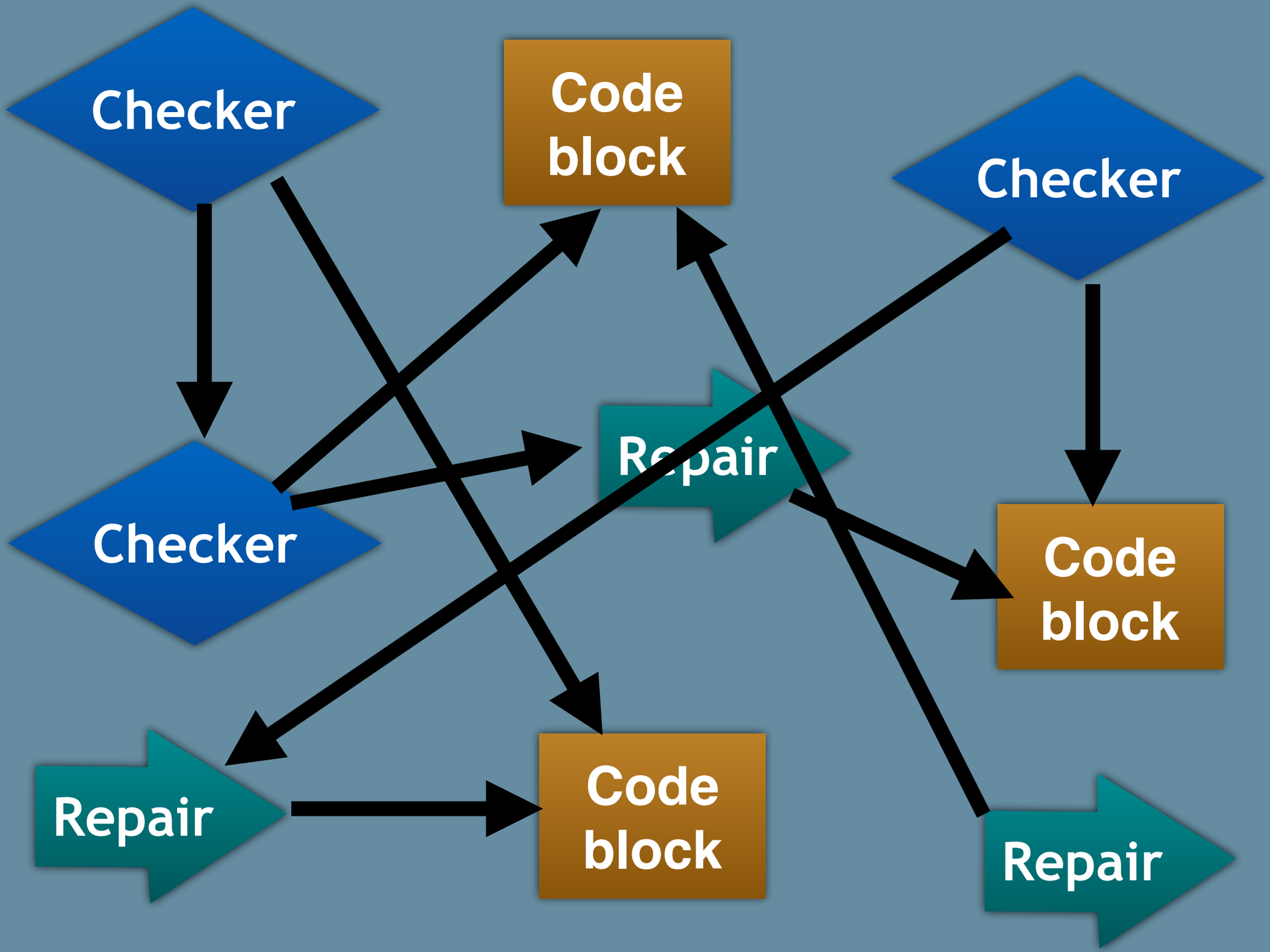
Checker 1

```
if (foo-has-changed-in-any-way())
```

copy

```
int foo_copy() {
  … … … …
}
```

```
int foo() {
  … … … …
}
```


Checker
Code block
Checker
Checker
Repair
Code block
Repair
Code block
Repair

```
uint32 Skypes_hash_function () {
  addr_t addr =(addr_t)((uint32)addr ^(uint32)addr);
  addr = (addr_t)((uint32) addr + 0 x688E5C);
  uint32 hash = 0x320E83 ^ 0x1C4C4 ;
  int bound = hash + 0 xFFCC5AFD ;

  do {
     uint32 data =*((addr_t)((uint32)addr + 0x10));
     goto b1; asm volatile (". byte 0x19"); b1:
     hash = hash ⊕ data ; addr -= 1; bound --;
  } while (bound !=0);

  goto b2;
   asm volatile (".byte 0x73");
  b2:
  goto b3;
   asm volatile (".word 0xC8528417,…”);
  b3:
  hash -= 0x4C49F346; return hash;
}
```

Questions?