Man-At-The-End Attacks and Defenses I

Lecture #8

Christian Collberg
University of Arizona

1

```
main() {
  passwd = "rosebud";

  log_in(passwd);
}
```

Confidentiality violation.

2

```
main() {
   if (!paid_license)
      abort()
   else
      …
}
```

```
main() {
   if (!true)
      abort()
   else
      …
}
```

Integrity violation.

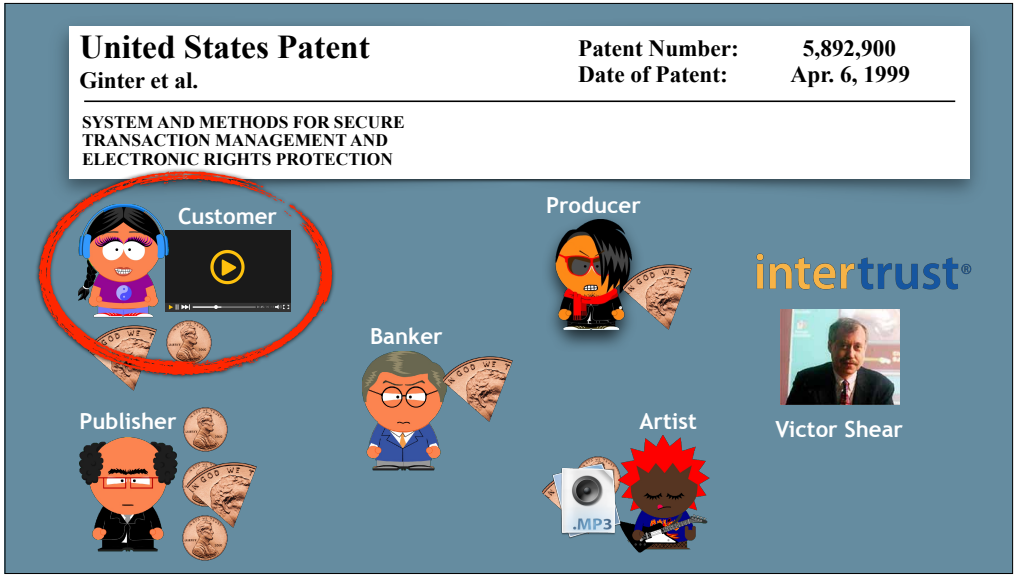
3

Man-At-The-End (MATE) attacks occur in any setting where an adversary has physical access to a device and compromises it by inspecting, reverse engineering, or tampering with its hardware or software.

4

Software Protection Applications

5



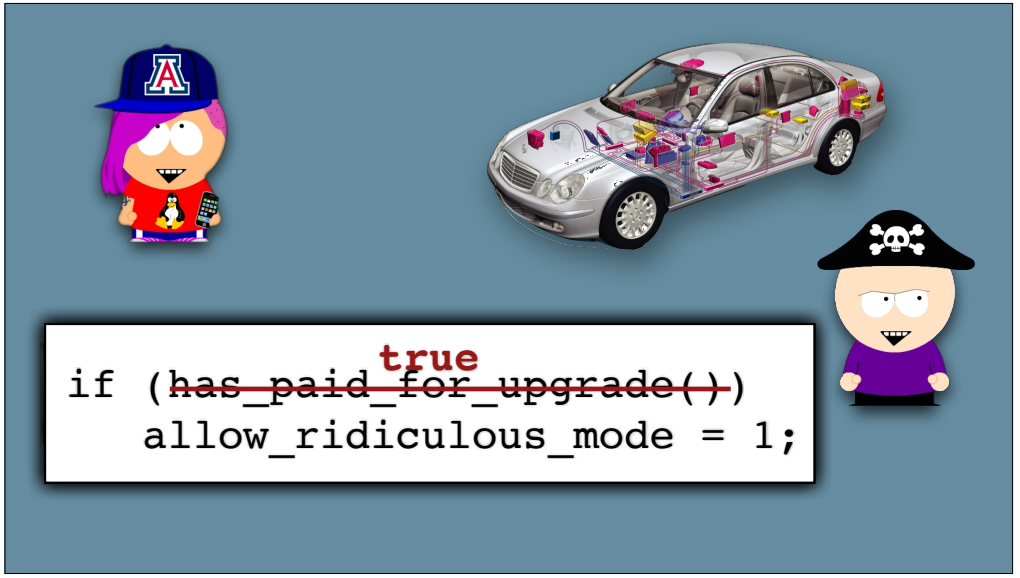
6

7

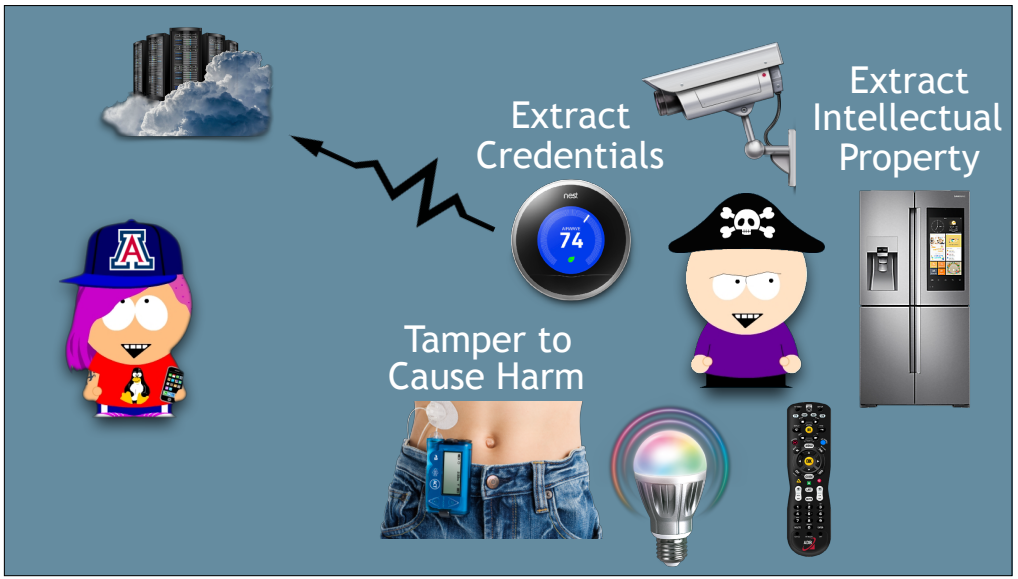
8


Off!
Cleemput, Mustafa, Preneel, *High Assurance Smart Metering*

9


true
if (has_paid_for_upgrade())
allow_ridiculous_mode = 1;

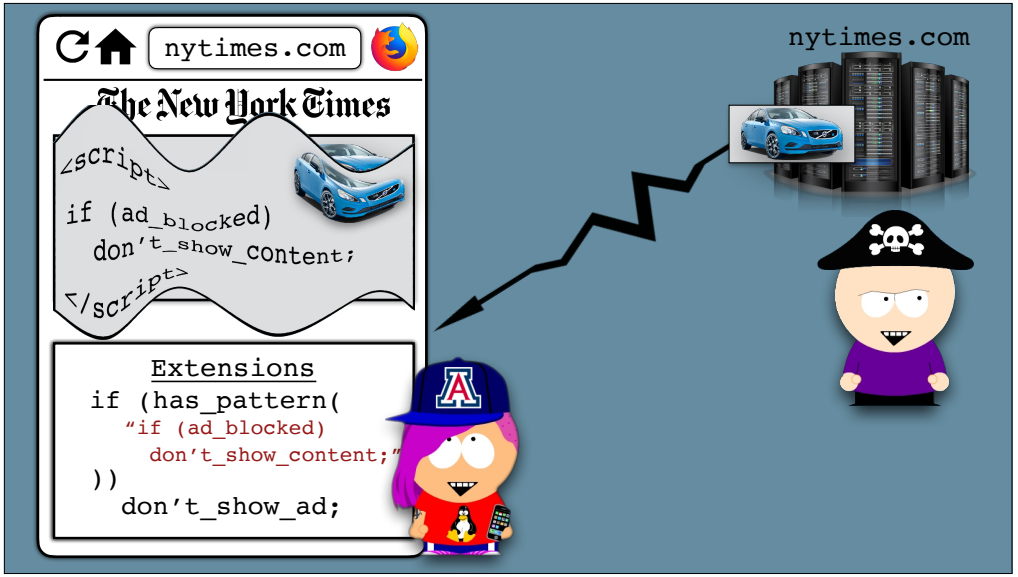
10


Extract Credentials
Extract Intellectual Property
Tamper to Cause Harm

11


NEW YORK POST
Queen B @ Doctor!
HACK!

12



13

Describe, in your own words, a few of the MATE scenarios we've seen.

1.

2.

14

Describe another few situations where a MATE attack can occur!

1.

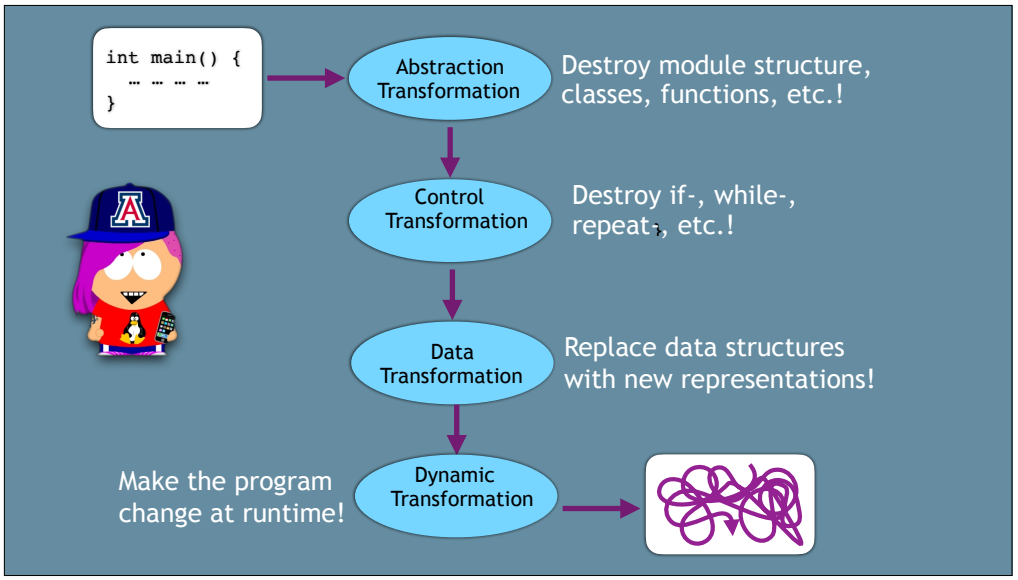
2.

15

Code Obfuscation

What is it?

16



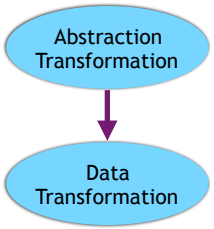
- The goal of code obfuscation is to make a program
  1. harder to understand for a human
  2. harder to analyze automatically for a reverse engineering tool
- in order to hide some asset in the program.


```
int main() {
   int y = 6;
   y = foo(y);
   bar(y,42);
}

int foo(int x)
   return x*7;
}

void bar(int x, int z) {
   if (x==z)
      printf("%i\n",x);
}
```


Abstraction Transformation

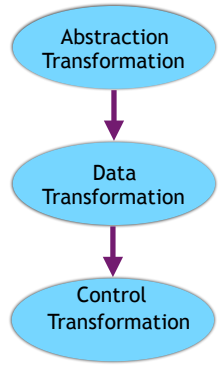
```
int main() {
   int y = 6;
   y = foobar(y,99,1);
   foobar(y,42,2);
}

int foobar(int x, int z, int s) {
   if (s==1)
      return x*7;
   else if (s==2)
      if (x==z)
         printf("%i \n",x);
}
```

```
int main () {
   int y = 12;
   y = foobar(y,99,1);
   foobar(y,36,2);
}
int foobar(int x, int z, int s) {
   if (s==1)
      return (x*37)%51;
   else if (x==z) {
      int x2=x*x%51,x3=x2*x%51;
      int x4=x2*x2%51,x8=x4*x4%51;
      int x11=x8*x3%51;
      printf("%i\n",x11);
   }
}
```



21

```
int foobar(int x, int z, int s){
   char* next=&&cell0;
   int retVal = 0;
   cell0: {next=(s==1)?&&cell1:&&cell2;
          goto *next;}
   cell1: {retVal=(x*37)%51; goto end;}
   cell2: {next=(s==2)?&&cell3:&&end;
          goto *next;}
   cell3: {next=(x==z)?&&cell4:&&end;
          goto *next;}
   cell4: {
      int x2=x*x%51,x3=x2*x%51;
      int x4=x2*x2%51,x8=x4*x4%51;
      int x11=x8*x3 % 51;
      printf("%i \n",x11); goto end;
   }
   end: return retVal;
}
```



22

Tamperproofing

What is it?

23

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

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

24

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

int main () {
  if (hash(foo,1000) != 0x4C49F346)
    ◆crash the program
    ◆phone home
    ◆refuse to run
    ◆run slower
    ◆make wrong results
  foo();
}
```

```
int foo() {
   … … … …
}
```

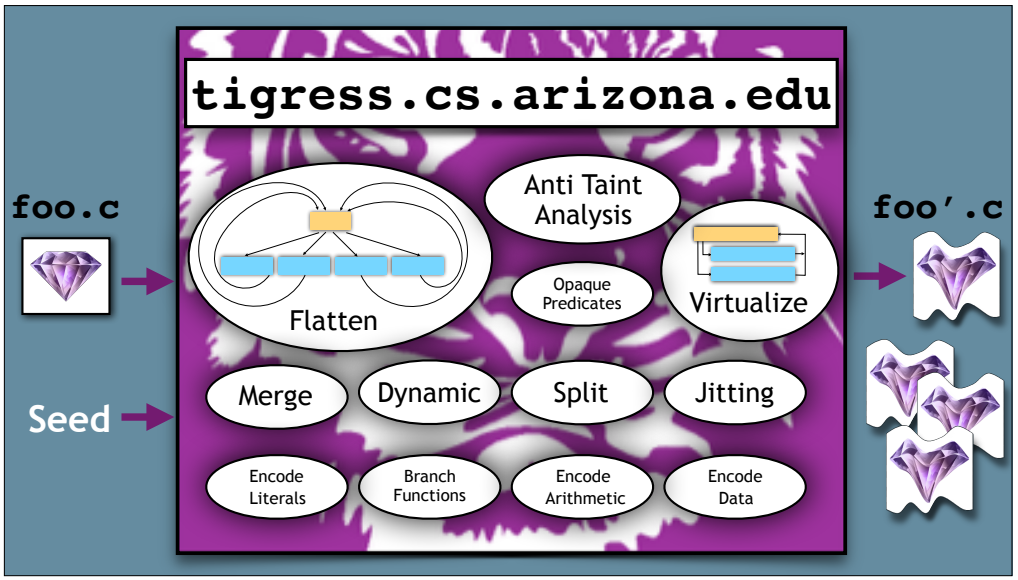
Detect tampering

Respond to tampering

25

Tigress

26



27

Make sure you have the following installed:

- `gcc` (or a drop-in replacement such as `clang`)
- `perl`
- `bash`

28

- Download Tigress from http://tigress.cs.arizona.edu/#download
- **GET THE UNSTABLE VERSION!**
- Unzip the .zip-file:
  `unzip tigress-ARCH-x86_64-VERSION.zip`

1. Set the TIGRESS_HOME environment variable to the directory in which the tigress binary resides, and add this directory to your path.

2. If your shell is csh, do

```
setenv TIGRESS_HOME /home/bob/bin/tigress-VERSION
setenv PATH "$PATH":/home/bob/bin/tigress-VERSION
rehash
```

3. If your shell is bash do

```
export TIGRESS_HOME=/home/bob/bin/tigress-VERSION
export PATH=$PATH:/home/bob/bin/tigress-VERSION
```


- Download these two files:
  - http://tigress.cs.arizona.edu/exercises.txt
  - http://tigress.cs.arizona.edu/fib.c
- The file **exercises.txt** contains tigress commands I want you to run. The commands can be long and I don't want you to have to type them yourselves.
- The file **fib.c** contains the C file I want you to work on.

- exercises.txt:

```
************************************************************
* 1) Opaque Predicates
************************************************************
tigress --Environment=x86_64:Linux:Gcc:4.6 --Seed=0 \
   --Transform=InitEntropy \
      --InitEntropyKinds=vars \
   --Transform=InitOpaque \
      --Functions=main\
      --InitOpaqueCount=2\
      --InitOpaqueStructs=list,array \
   --Transform=AddOpaque\
      --Functions=fib\
      --AddOpaqueKinds=question \
      --AddOpaqueCount=10 \
    --out=fib1.c fib.c

************************************************************
* 2) Flatten
************************************************************
```

```c
// #include "apple.h"
#include<stdio.h>
#include<stdlib.h>
// #include “tigress_unstable/jitter-amd64.c”

int fib(int n) {
   int a = 1;  int b = 1; int i;
   for (i = 3; i <= n; i++) {
      int c = a + b; a = b; b = c;
   };
   return b;
}

int main(int argc, char** argv) {
  if (argc != 2) {
     printf("Give one argument!\n”); abort();
  };
  long n = strtol(argv[1],NULL,10);
  int f = fib(n);
  printf("fib(%li)=%i\n",n,f);
}
```


- Try this simple transformation to make sure everything works!

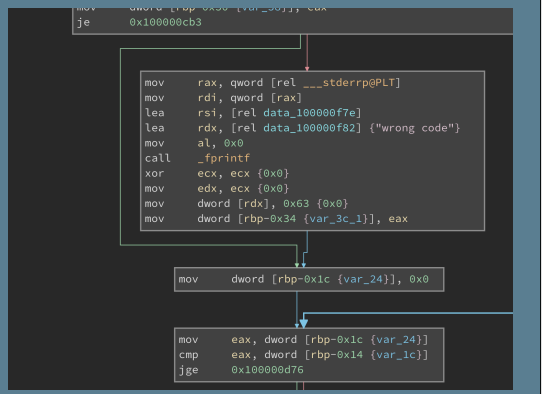
```
*****************************************
* 3) Virtualize
*****************************************
tigress --Environment=x86_64:Linux:Gcc:4.6\
   --Transform=Virtualize\
     --Functions=fib\
     --VirtualizeDispatch=switch\
   --out=fib3.c fib.c
```


Reverse Engineering

Exercise!

Work with your friends!!!

- Go to cloud.binary.ninja
- Load some binary file
- Click around!



Reverse Engineering Tools

- Angr: angr.io
- KLEE: https://klee.github.io
- Ghidra: https://ghidra-sre.org

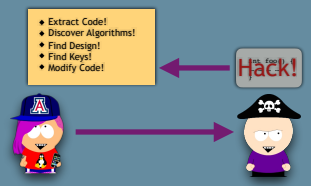
Wanna Become a Reverse Engineer?

- Let's start a CTF!
- CTF: Capture The Flag

Summary

Exercise

- *Why do we obfuscate?*
- *Why do we tamperproof?*



Discuss with your friends!!!

Exercise

- *Can obfuscation be used to tamperproof a program?*

Discuss with your friends!!!

Exercise

- *Should you both obfuscate and tamperproof a program?*

If so, why?

Discuss with your friends!!!