What has my compiler done for me lately to https://eduassistpro.github.io/ unspechinentat Add WeChat edu_assist_pro

Agenda

- 1. Actual shellcode attacks.
- 2. Stack/Heap based exploits a. W^X memory. Project Exam Help
- 3. Return Oriented Pro
 - a. Stack Cookies https://eduassistpro.github.io/
 - b. Shadow Call Stack
- 4. Indirect control flow & cold viable attack edu_assist_pro
 - a. CFI

Shellcode Attacks

- Shellcode is native (byte) code.
- The encoded instructions that are interpreted by the GPU. Assignment Project Exam Help

https://eduassistpro.github.io/

Shellcode Attacks

- Shellcode is native (byte) code.
- The encoded instructions that are interpreted by the GPU. Assignment Project Exam Help

Shellcode Attacks

- Shellcode is native (byte) code.
- The encoded instructions that are interpreted by the GPU.
 Assignment Project Exam Help

```
48 89 ec ; movq
```

c3 ; ret https://eduassistpro.github.io/

90 ; nop

- Managed code (Javascript):
 - Sandboxed. Can only access very specific things.
 - All interaction managed by interpreter.
- Shellcode:
 - o Full access to the system. Run directly on hardware.
- Two different attack types:

Remote attacks?

Assignment Project Exam Help

https://eduassistpro.github.io/

- Remote attacks.

 - We have no ability to run unmanaged code.

 Goal? Assignment Project Exam Help

https://eduassistpro.github.io/

- Remote attacks.
 - We have no ability to run unmanaged code.
 - o Goal? Assignment Project Exam Help
 - Get shellcode
 - Targets?

https://eduassistpro.github.io/

- Remote attacks.
 - We have no ability to run unmanaged code.
 - Assignment Project Exam Help Goal?
 - Get native cod
 - Targets? https://eduassistpro.github.io/
 - Browsers, net
- Local attacks?

- Remote attacks.
 - We have no ability to run unmanaged code.
 - o Goal? Assignment Project Exam Help
 - Get native cod
 - Targets?
 Browsers net

 https://eduassistpro.github.io/
- Local attacks.
 - We have shellcode access on the machine.
 - Goal?

- Remote attacks.
 - We have no ability to run unmanaged code.
 - o Goal? Assignment Project Exam Help
 - Get native cod
 - Targets?
 Browsers net
 https://eduassistpro.github.io/
- Local attacks.
 - We have shellcode access on the machine.

 Edu_assist_pro
 - Goal?
 - Privilege escalation.
 - o Targets?

- Remote attacks.
 - We have no ability to run unmanaged code.
 - o Goal? Assignment Project Exam Help
 - Get native cod
 - Targets? https://eduassistpro.github.io/
- Local attacks.
 - we have shellcode access on the machine. edu_assist_pro
 - Goal?
 - Privilege escalation.
 - o Targets?
 - Kernel, hypervisor, any process running as a different user/group.

Stack based exploits

Assignment Project Exam

Arguments

Exam Help

Return Address

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Local Variables

Stack based exploits

- Overwrite the local variables
 - a. Buffer overflow
 - b. Use after return Ssignment Project Exam
 - c. Use after scope

Arguments

Exam Help

Return Address

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

shellcode

Stack based exploits

- Overwrite the local variables
 - Buffer overflow
 - Use after return SSignment Project Exam Help
 - Use after scope
- Overwrite the return https://eduassistpro.github.

Add WeChat edu_assist_pi

Arguments

ptr to shellcode

shellcode

Heap based exploits

- Add shellcode to heap:
 - Heap buffer overflow
 - Use after free Assignment Project Exam Help
 - Global buffer overflo
 - Initialization order b https://eduassistpro.github.io/
- Overwrite the return

Add WeChat edu_assist_p (unused)

Arguments ptr to shellcode

somewhere on heap/global: shellcode

W[^]X Memory

- Write XOR Execute

- Compiler sets section stack, heap, etc) mehttps://eduassistpro.github.io/ no-execute. Done by default on all compilers.

 Add WeChat edu_assist_pro
- Problem solved, right...?
 - Chrome still RWX v8 pages.

Arguments Can't execute shellcede.

Can't execute shellcede.

EXECUTABLE)

somewhere on heap/global: shellcode (NOT **EXECUTABLE**)

Chains together "gadgets", which is
 a sequence of a few instructions
 followed by 'ret'.

Argument Project Example 1
Return A

Arguments

Exam Help
Return Address

 Smash stack with lo to gadgets.

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Local Variables

Chains together "gadgets", which is
 a sequence of a few instructions
 followed by 'ret'.

ptr_3

Prx2am Help

- Smash stack with lo to gadgets.

 https://eduassistpro.github.io/
- Choose gadgets to execute WeChat edu_assist_pro shellcode we want.

 (unused)

- Chains together "gadgets", which is
 a sequence of a few instructions followed by 'ret'.
- Smash stack with lo to gadgets.

 https://eduassistpro.github.
- Choose gadgets to execute WeChat edu_assist_p shellcode we want.
- If clever, we hide strings in other places we have write access to.
 Use the strings in the exploit.

ptr_3

(unused)

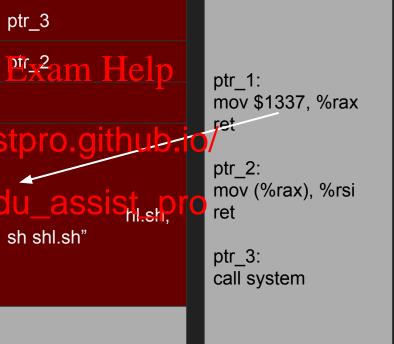
Ľz²am Help

ptr_1: mov \$1337, %rax ret

ptr_2: mov (%rax), %rsi ret

ptr_3: call system

- Chains together "gadgets", which is a sequence of a few instructions followed by 'ret'.
- Smash stack with lo to gadgets.
- Choose gadgets to execute WeChat established we want.
- If clever, we hide strings in other places we have write access to.
 Use the strings in the exploit.



Stack Cookies

- -fstack-protector
- Adds "cookie" or "canary" to stack a ssignment Project on function entry.
- Checks it on function
- If the cookie fails, kilhttps://eduassistpro.github.io/

Add WeChat edu_assist_pro

Local Variables

Arguments Return Address

Stack Cookies

```
movq %fs:40, %rax ; grab cookie
                                             Arguments
movq %rax, -8(%rbp); save to stack xorl %rax, %rax ; hide the cookie Exam Help
                                             ptr 1
<normal function code> https://eduassistpro.github.jo/
                                                              (was) Stack cookie
movq -8(%rbp), %rax; get from stack edu_assist_pro
                                             (unused)
xorq %fs:40, %rax ; compare
inz stack chk fail
ret
```

Stack Cookies

Overflowing the return address must write over stack cookie.

Cookie is hidden outside of normal

memory.

5.4210 * 10^(-20) chhttps://eduassistpro.github.io/

Stops sequential write bugs, what edu_assist_pro about arbitrary write?

ptr 3

Local Variables

Shadow Call Stack

- -fsanitize=shadow-call-stack
- Another ROP defense
 Separate stacks into safe and Project Example Saved 6 unsafe.
- Safe contains return https://eduassistpro.github.io
- Unsafe contains everything else. Chat edu_assist_pro

Arguments Saved ebp

Unsafe Stack

Return Address 1

Safe Stack

Return Address 2

Return Address 3

Shadow Call Stack

Arbitrary writes are still safe as long as safe stack pointer is secret.
Safe stack pointer is nidden:

Exam Help

Reserved register (x

Reserved segment (https://eddewssistpro.github.

Add WeChat edu_assist_p

Unsafe Stack Safe Stack

Return Address 1

Return Address 2

Return Address 3

TOCTOU Vulnerabilities

- Time of Check to Time of Use
- Thread-based security race Project Normal supetion entry. Allocate space on shadow stack. between call and pr https://eduadswistacko.github.io/ finishing.
- Prologue #3, save return address into register at top of function entry.
- Requires stack location disclosure.

ShadowCallStack Prologue:

- Pop from shadow stack.
- Compare to return address off regular stack.
- Fail if return address has been compromised.

int foo num = 0;

Bar as an Object Bar's VFN table Used to implement polymorphism. &Bar::x Assignment Project Exam Help struct Bar { &Bar::y virtual void x() { ... } virtual void y() { ... } https://eduassistpro.github. Foo's VFN table int bar num = 0; &Foo::x Add WeChat edu_assist foo num &Bar::y class Foo : Bar { vtable ptr void x() override { ... }

```
Foo as an Object
int x() {
                                                         Arguments
                Assignment Project Exam Help
    Foo my_foo;
   foo.x();
                                                         Return Address
                                       foo num
                     https://eduassistpro.github.iog/ved ebp
                                                                 bar num
                      Add WeChat edu_assist_prop_foo:
                                                                 foo num
                                                                 &vtable
```

```
Foo as an Object
int x() {
                                                           Arguments
                Assignment Project Exam Help
    Foo my_foo;
    foo.x();
                                                           Return Address
                                        foo num
                      https://eduassistpro.github.iog/ved ebp
                                                                  bar num
X:
<create my_foo on stack</pre>
Add WeChat edu_assist_prop_foo:
                                                                   foo num
movq -8(%rsp), %rdi ; get vtable ptr
                                                                   &vtable
movg $0, %rsi
                    ; index of &Foo:x
movq %rsi(%rdi), %rdi; get &Foo:x
callq *(%rdi)
                    ; call &Foo:x
```

Foo as an Object Add some dangerous code and... Arguments Assignment Project Exam Help int x() { Return Address foo num Foo my_foo; https://eduassistpro.github.iog/ved ebp char buffer[255]; fgets("%s", buffer); bar num Add WeChat edu_assist_prog_foo: foo.x(); foo num ptr_1 buffer overflow (unused)

Control Flow Integrity (CFI)

```
movq -8(%rsp), %rsi ; get vtable ptr
-fsanitize=cfi
Adds checks to ensure correct project Exam Help vtable before call.
                                                                ; index of &Foo:x
vtable before call.
Stops smashing vta stack/heap. ; get &Foo:x https://eduassistpro.github.io/call &Foo:x
stack/heap.
Kills the program on sanity check wechat edu_assist_pro
failures.
                                        ; ensure table is in range and aligned
                                        ; ensure index (%rsi) is valid
```

CFI cont.

More advanced attack: Run a different virtual function from a Project Exam Help different class.

class Other { virtual void n();

CFI protects against these as well.

Foo as an Object foo_num https://eduassistpro.github.iog/ved ebp Add WeChat edu_assist_ &Other::n

Arguments

Return Address

bar num

My_foo: foo num

&vtable

buffer overflow (unused)

CFI Cast Checking

```
Bar:
    CFI adds checks to all types of cast
    checking.
                  Assignment Project Exam Helpoid x()
                                                       virtual void y()
Baz* b1 = new Foo();
                        https://eduassistpro.github
Baz^* b2 = new Bar():
                                                                 Baz (inherits Bar):
Bar* b3 = new Foo():
                          // OK
                         AddromeChat edu_assist_
                                                             Orirtual void z();
Baz* b4 = b3:
(Other* b3)->non_virtual(); // wrong
Bar* b6 = new Other();
                          // wrong
                                                                 Other:
void* o = new Other();
                          // OK
                                                                 virtual void n();
(Bar^* o)->y();
                          // wrong
                                                                 void non_virtual();
```

CFI Indirect Call

Also protects against similar Arguments Arguments tomfoolery with indirect function Project Exa calls. Return Address Return Address void my_function() { ... } https://eduassistpro.github.io//aved ebp ptr_1 Add WeChat edu_assist int main() { buffer overflow void (*fn ptr)() = &my function; buffer (unused) char buffer[255]; fgets("%s", buffer); fn ptr();

CFI Issues

- Only forward-edge protection.
 - o rCFI is implemented, but very expensive and requires significant metadata.
- Cross-DSO is complicationally expensive! Exam Help
- Checks can get quit
 - e.g. Base class vcallhttps://eduassistpro.github.io/

```
Derived::x() {
    return Base::x() + Base::y();
}
Chat edu_assist_pro
}
```

also...

Assignment Project Exam Help

https://eduassistpro.github.io/

Other CFI

cise.

- Microsoft Control Flow Guard (CFG)
 - Near-precise. Isn't perfect.
- Intel Control Entorcement Fechnologie E Fxam Help
 - Hardware enforced
 - Also ENDBRANCH, https://eduassistpro.github.io/
- ARM Pointer Authe

Add WeChat edu_assist_pro

38

Final Notes

- Protection mechanisms mentioned do not fix the underlying bug.
- Can still deny service by crashing process.

 No protection is holistic.

 Help
- Compilers can only
 - -fstack-protector (-fshttps://eduassistpro.github.io/
 - -fsanitize=cfi
 - -shadow-call-stack Add WeChat edu_assist_pro
- Compilers are made by humans. Any s ode should always be inspected by hand.