KCon

纵横

THE ACHILLES' HEEL OF CFI

演讲者：张云海

关于 CFI

CFI 是什么

2005年微软研究院联合学术界提出的一项漏洞利用缓解技术

用于防御利用内存破坏漏洞来获得软件行为控制权的外部攻击

确保程序执行时的控制流转移符合事先确定的控制流图

关于 CFI

CFI 的实现

Clang CFI

Microsoft Control Flow Guard

Intel Control-Flow Enforcement Technology

Microsoft eXtended Flow Guard

Clang CFI

Clang CFI 如何工作

-fsanitize=cfi-cast-strict: Enables strict cast checks.

-fsanitize=cfi-derived-cast: Base-to-derived cast to the wrong dynamic type.

-fsanitize=cfi-unrelated-cast: Cast from void* or another unrelated type to the wrong dynamic type.

-fsanitize=cfi-nvcall: Non-virtual call via an object whose vptr is of the wrong dynamic type.

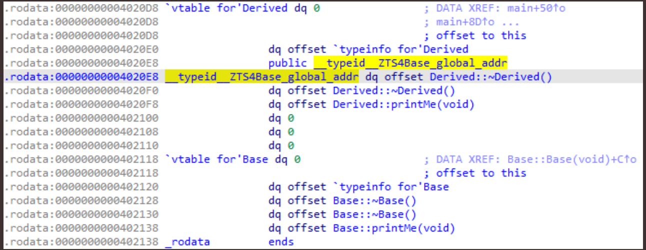
-fsanitize=cfi-vcall: Virtual call via an object whose vptr is of the wrong dynamic type.

-fsanitize=cfi-icall: Indirect call of a function with wrong dynamic type.

-fsanitize=cfi-mfcall: Indirect call via a member function pointer with wrong dynamic type

Clang CFI

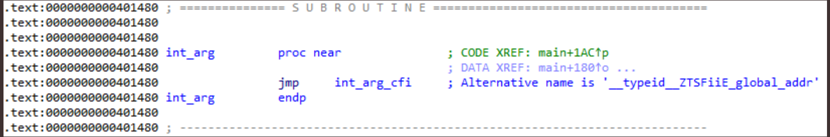
Clang CFI 如何工作



```
.rodata:00000000004020D8 `vtable for'Derived dq 0                ; DATA XREF: main+50↑o
.rodata:00000000004020D8                                         ; main+8D↑o ...
.rodata:00000000004020D8                                         ; offset to this
.rodata:00000000004020E0                 dq offset `typeinfo for'Derived
.rodata:00000000004020E8                 public __typeid__ZTS4Base_global_addr
.rodata:00000000004020E8 __typeid__ZTS4Base_global_addr dq offset Derived::~Derived()
.rodata:00000000004020F0                 dq offset Derived::~Derived()
.rodata:00000000004020F8                 dq offset Derived::printMe(void)
.rodata:0000000000402100                 dq 0
.rodata:0000000000402108                 dq 0
.rodata:0000000000402110                 dq 0
.rodata:0000000000402118 `vtable for'Base dq 0                   ; DATA XREF: Base::Base(void)+C↑o
.rodata:0000000000402118                                         ; offset to this
.rodata:0000000000402120                 dq offset `typeinfo for'Base
.rodata:0000000000402128                 dq offset Base::~Base()
.rodata:0000000000402130                 dq offset Base::~Base()
.rodata:0000000000402138                 dq offset Base::printMe(void)
.rodata:0000000000402138 _rodata         ends
```

Clang CFI

Clang CFI 如何工作



```
.text:0000000000401480 ; =============== S U B R O U T I N E =======================================
.text:0000000000401480
.text:0000000000401480
.text:0000000000401480 int_arg         proc near               ; CODE XREF: main+1AC↑p
.text:0000000000401480                                         ; DATA XREF: main+180↑o ...
.text:0000000000401480                 jmp     int_arg_cfi     ; Alternative name is '__typeid__ZTSFiiE_global_addr'
.text:0000000000401480 int_arg         endp
.text:0000000000401480
.text:0000000000401480 ; ---------------------------------------------------------------------------
```

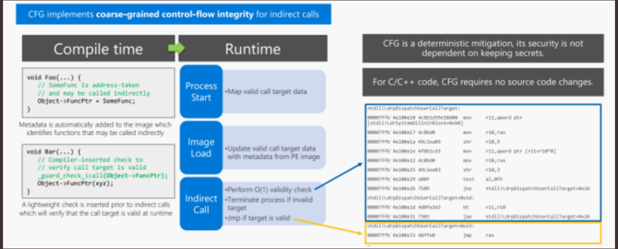
Clang CFI

Clang CFI 的问题

- 适用的场合受限
- 缺少对 Backward-Edge 的保护

Microsoft Control Flow Guard

CFG 如何工作



CFG implements **coarse-grained control-flow integrity** for indirect calls

Compile time
Runtime
Process Start
•Map valid call target data
Image Load
•Update valid call target data with metadata from PE image
Indirect Call
•Perform O(1) validity check
•Terminate process if invalid target
•Jmp if target is valid

```
void Foo(...) {
    // SomeFunc is address-taken
    // and may be called indirectly
    Object->FuncPtr = SomeFunc;
}
```

Metadata is automatically added to the image which identifies functions that may be called indirectly

```
void Bar(...) {
    // Compiler-inserted check to
    // verify call target is valid
    _guard_check_icall(Object->FuncPtr);
    Object->FuncPtr(xyz);
}
```

A lightweight check is inserted prior to indirect calls which will verify that the call target is valid at runtime

CFG is a deterministic mitigation, its security is not dependent on keeping secrets.

For C/C++ code, CFG requires no source code changes.

```
ntdll!LdrpDispatchUserCallTarget:
00007ffb`4e100e10 4c8b1d59e50d00  mov     r11,qword ptr
[ntdll!LdrSystemDllInitBlock+0xb0]
00007ffb`4e100e17 4c8bd0          mov     r10,rax
00007ffb`4e100e1a 49c1ea09        shr     r10,9
00007ffb`4e100e1e 4f8b1cd3        mov     r11,qword ptr [r11+r10*8]
00007ffb`4e100e22 4c8bd0          mov     r10,rax
00007ffb`4e100e25 49c1ea03        shr     r10,3
00007ffb`4e100e29 a80f            test    al,0fh
00007ffb`4e100e2b 7509            jne     ntdll!LdrpDispatchUserCallTarget+0x26
ntdll!LdrpDispatchUserCallTarget+0x1d:
00007ffb`4e100e2d 4d0fa3d3        bt      r11,r10
00007ffb`4e100e31 7303            jae     ntdll!LdrpDispatchUserCallTarget+0x26
ntdll!LdrpDispatchUserCallTarget+0x23:
00007ffb`4e100e33 48ffe0          jmp     rax
```

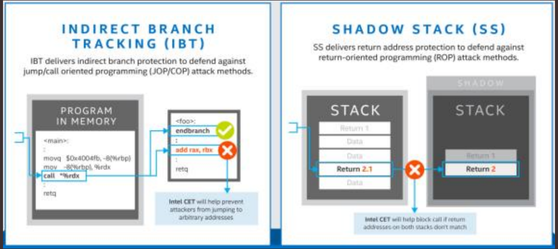
Microsoft Control Flow Guard

CFG 的问题

- CFG 是一个粗粒度的 CFI 实现
- 已知多种针对 CFG 的绕过技术
- 缺少对 Backward-Edge 的保护

Intel Control-Flow Enforcement Technology

CET 如何工作


INDIRECT BRANCH TRACKING (IBT)
IBT delivers indirect branch protection to defend against jump/call oriented programming (JOP/COP) attack methods.
PROGRAM IN MEMORY
<main>:
movq $0x4004fb, -8(%rbp)
mov -8(%rbp), %rdx
call *%rdx
retq
<foo>:
endbranch
add rax, rbx
retq
Intel CET will help prevent attackers from jumping to arbitrary addresses
SHADOW STACK (SS)
SS delivers return address protection to defend against return-oriented programming (ROP) attack methods.
STACK
Return 1
Data
Data
Return 2.1
Data
SHADOW
STACK
Return 1
Return 2
Intel CET will help block call if return addresses on both stacks don't match

Intel Control-Flow Enforcement Technology

CET 的问题

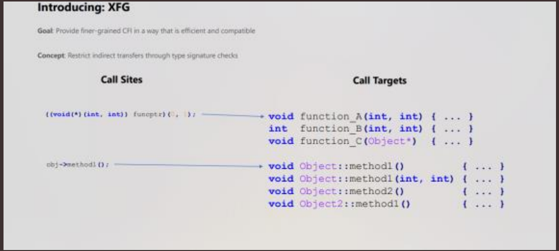
依赖特定的硬件

IBT 也是一个粗粒度的 CFI 实现

多数针对 CFG 的绕过技术也适用于 IBT

Microsoft eXtended Flow Guard

XFG 如何工作



Introducing: XFG

Goal: Provide finer-grained CFI in a way that is efficient and compatible

Concept: Restrict indirect transfers through type signature checks

Call Sites

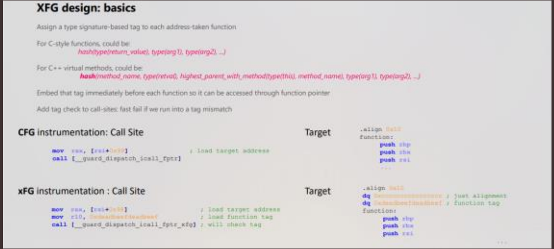
```
((void(*)(int, int)) funcptr)(0, 1);

obj->method1();
```

Call Targets

```
void function_A(int, int) { ... }
int  function_B(int, int) { ... }
void function_C(Object*)  { ... }

void Object::method1()          { ... }
void Object::method1(int, int) { ... }
void Object::method2()          { ... }
void Object2::method1()         { ... }
```

Microsoft eXtended Flow Guard

XFG 如何工作



XFG design: basics

Assign a type signature-based tag to each address-taken function

For C-style functions, could be:
hash(type(return_value), type(arg1), type(arg2), ...)

For C++ virtual methods, could be:
hash*(method_name, type(retval), highest_parent_with_method(type(this), method_name), type(arg1), type(arg2), ...)*

Embed that tag immediately before each function so it can be accessed through function pointer

Add tag check to call-sites: fast fail if we run into a tag mismatch

CFG instrumentation: Call Site

```
mov  rax, [rsi+0x98]                     ; load target address
call [__guard_dispatch_icall_fptr]
```

Target

```
.align 0x10
function:
      push rbp
      push rbx
      push rsi
      ...
```

xFG instrumentation : Call Site

```
mov  rax, [rsi+0x98]                      ; load target address
mov  r10, 0xdeadbeefdeadbeef              ; load function tag
call [__guard_dispatch_icall_fptr_xfg] ; will check tag
```

Target

```
.align 0x10
dq 0xcccccccccccccccc ; just alignment
dq 0xdeadbeefdeadbeef ; function tag
function:
      push rbp
      push rbx
      push rsi
                          ...
```

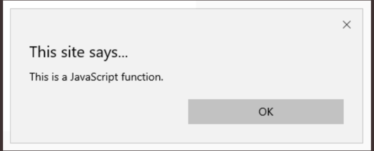
Microsoft eXtended Flow Guard

如何绕过 XFG?

- 控制流图中 fan-in fan-out 的数量会显著影响 CFI 的有效性
  - Variable Arguments
  - Generic Function Object

JavaScript Function

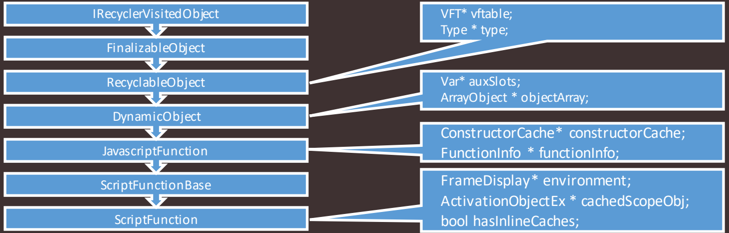
```
function f() {
   alert("This  is  a  JavaScript  Function.");
}

var o  =  f;
o();
```


This site says...

This is a JavaScript function.

JavaScript Function

Js::ScriptFunction



```
VFT* vftable;
Type * type;
```

```
Var* auxSlots;
ArrayObject * objectArray;
```

```
ConstructorCache* constructorCache;
FunctionInfo * functionInfo;
```

```
FrameDisplay* environment;
ActivationObjectEx * cachedScopeObj;
bool hasInlineCaches;
```

JavaScript Function

如何调用

```
template <class T> void OP_ProfiledCallI(const unaligned OpLayoutDynamicProfile<T>* playout) {
    OP_ProfileCallCommon(playout, OP_CallGetFunc(GetRegAllowStackVar(playout->Function)), Js::CallFlags_None, playout->profileId);
}

template <typename RegSlotType> Var InterpreterStackFrame::GetRegAllowStackVar(RegSlotType localRegisterID) const {
    Var value = m_localSlots[localRegisterID];
    ValidateRegValue(value, true);
    return value;
}

RecyclableObject * InterpreterStackFrame::OP_CallGetFunc(Var target) {
    return JavascriptOperators::GetCallableObjectOrThrow(target, GetScriptContext());
}
```

JavaScript Function

如何调用

```
template <class T> void InterpreterStackFrame::OP_ProfileCallCommon(const unaligned T * playout, RecyclableObject * function
    , unsigned flags, ProfileId profileId, InlineCacheIndex inlineCacheIndex, const Js::AuxArray<uint32> *spreadIndices) {
    FunctionBody* functionBody = this->m_functionBody;
    DynamicProfileInfo * dynamicProfileInfo = functionBody->GetDynamicProfileInfo();
    FunctionInfo* functionInfo = function->GetTypeId() == TypeIds_Function ?
        JavascriptFunction::FromVar(function)->GetFunctionInfo() : nullptr;
    bool isConstructorCall = (CallFlags_New & flags) == CallFlags_New;
    dynamicProfileInfo->RecordCallSiteInfo(functionBody, profileId, functionInfo, functionInfo ?
        static_cast<JavascriptFunction*>(function) : nullptr, playout->ArgCount, isConstructorCall, inlineCacheIndex);
    OP_CallCommon<T>(playout, function, flags, spreadIndices);
    if (playout->Return != Js::Constants::NoRegister) {
        dynamicProfileInfo->RecordReturnTypeOnCallSiteInfo(functionBody, profileId, GetReg((RegSlot)playout->Return));
    }
}
```

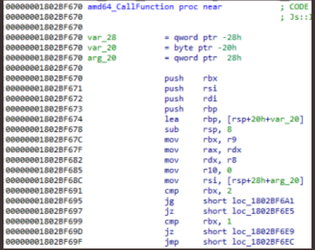
JavaScript Function

如何调用

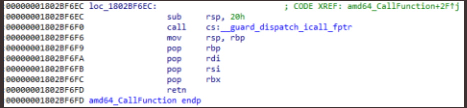
```
void InterpreterStackFrame::OP_CallCommon(const unaligned T * playout, RecyclableObject * function, unsigned flags
    , const Js::AuxArray<uint32> *spreadIndices){
    ...
    flags |= CallFlags_NotUsed;
    Arguments args(CallInfo((CallFlags)flags, argCount), m_outParams);
    AssertMsg(static_cast<unsigned>(args.Info.Flags) == flags, "Flags don't fit into the CallInfo field?");
    argCount = args.GetArgCountWithExtraArgs();
    if (spreadIndices != nullptr) {
        JavascriptFunction::CallSpreadFunction(function, args, spreadIndices);
    } else {
        JavascriptFunction::CallFunction<true>(function, function->GetEntryPoint(), args);
    }
    ...
}
```

JavaScript Function

如何调用



```
00000001802BF670 amd64_CallFunction proc near                    ; CODE
00000001802BF670                                                 ; Js::1
00000001802BF670
00000001802BF670 var_28          = qword ptr -28h
00000001802BF670 var_20          = byte ptr -20h
00000001802BF670 arg_20          = qword ptr  28h
00000001802BF670
00000001802BF670                 push    rbx
00000001802BF671                 push    rsi
00000001802BF672                 push    rdi
00000001802BF673                 push    rbp
00000001802BF674                 lea     rbp, [rsp+20h+var_20]
00000001802BF678                 sub     rsp, 8
00000001802BF67C                 mov     rbx, r9
00000001802BF67F                 mov     rax, rdx
00000001802BF682                 mov     rdx, r8
00000001802BF685                 mov     r10, 0
00000001802BF68C                 mov     rsi, [rsp+28h+arg_20]
00000001802BF691                 cmp     rbx, 2
00000001802BF695                 jg      short loc_1802BF6A1
00000001802BF697                 jz      short loc_1802BF6E5
00000001802BF699                 cmp     rbx, 1
00000001802BF69D                 jz      short loc_1802BF6E9
00000001802BF69F                 jmp     short loc_1802BF6EC
```



```
00000001802BF6EC loc_1802BF6EC:                          ; CODE XREF: amd64_CallFunction+2F↑j
00000001802BF6EC                 sub     rsp, 20h
00000001802BF6F0                 call    cs:__guard_dispatch_icall_fptr
00000001802BF6F6                 mov     rsp, rbp
00000001802BF6F9                 pop     rbp
00000001802BF6FA                 pop     rdi
00000001802BF6FB                 pop     rsi
00000001802BF6FC                 pop     rbx
00000001802BF6FD                 retn
00000001802BF6FD amd64_CallFunction endp
```

JavaScript Function

如何调用

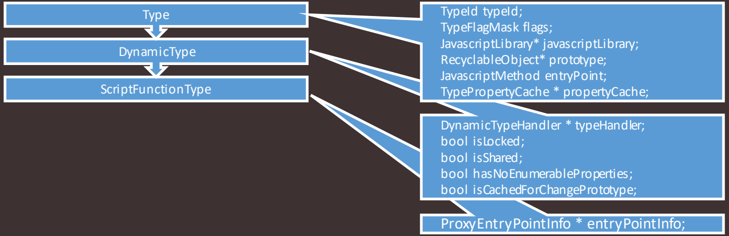
```
JavascriptMethod RecyclableObject::GetEntryPoint() const {
    return this->GetType()->GetEntryPoint();
}

inline Type * GetType() const {
    return type;
}

JavascriptMethod GetEntryPoint() const {
    return entryPoint;
}
```

KCon

纵横

JavaScript Function

Js::ScriptFunctionType



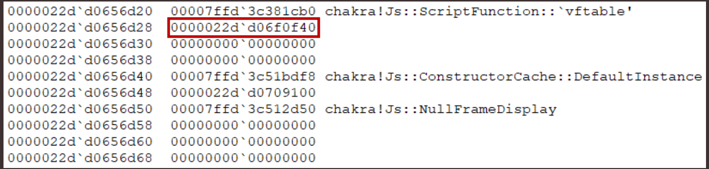
```
TypeId typeId;
TypeFlagMask flags;
JavascriptLibrary* javascriptLibrary;
RecyclableObject* prototype;
JavascriptMethod entryPoint;
TypePropertyCache * propertyCache;

DynamicTypeHandler * typeHandler;
bool isLocked;
bool isShared;
bool hasNoEnumerableProperties;
bool isCachedForChangePrototype;

ProxyEntryPointInfo * entryPointInfo;
```

KCon

纵横

JavaScript Function

Js::ScriptFunction



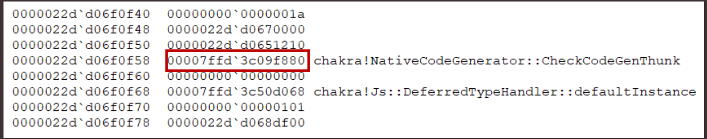
```
0000022d`d0656d20  00007ffd`3c381cb0 chakra!Js::ScriptFunction::`vftable'
0000022d`d0656d28  0000022d`d06f0f40
0000022d`d0656d30  00000000`00000000
0000022d`d0656d38  00000000`00000000
0000022d`d0656d40  00007ffd`3c51bdf8 chakra!Js::ConstructorCache::DefaultInstance
0000022d`d0656d48  0000022d`d0709100
0000022d`d0656d50  00007ffd`3c512d50 chakra!Js::NullFrameDisplay
0000022d`d0656d58  00000000`00000000
0000022d`d0656d60  00000000`00000000
0000022d`d0656d68  00000000`00000000
```

KCon

纵横

JavaScript Function

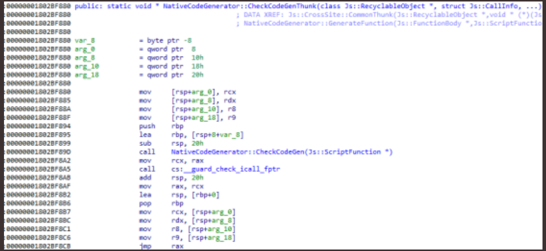
Js::ScriptFunctionType



```
0000022d`d06f0f40  00000000`0000001a
0000022d`d06f0f48  0000022d`d0670000
0000022d`d06f0f50  0000022d`d0651210
0000022d`d06f0f58  00007ffd`3c09f880 chakra!NativeCodeGenerator::CheckCodeGenThunk
0000022d`d06f0f60  00000000`00000000
0000022d`d06f0f68  00007ffd`3c50d068 chakra!Js::DeferredTypeHandler::defaultInstance
0000022d`d06f0f70  00000000`00000101
0000022d`d06f0f78  0000022d`d068df00
```

JavaScript Function

NativeCodeGenerator::CheckCodeGenThunk

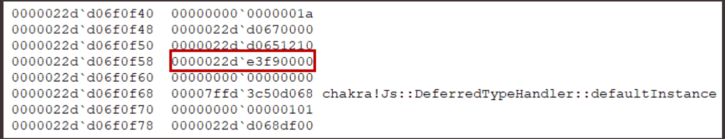


```
:00000001802BF880 public: static void * NativeCodeGenerator::CheckCodeGenThunk(class Js::RecyclableObject *, struct Js::CallInfo, ...)
:00000001802BF880                                          ; DATA XREF: Js::CrossSite::CommonThunk(Js::RecyclableObject *,void * (*)(Js
:00000001802BF880                                          ; NativeCodeGenerator::GenerateFunction(Js::FunctionBody *,Js::ScriptFunctio
:00000001802BF880
:00000001802BF880 var_8           = byte ptr -8
:00000001802BF880 arg_0           = qword ptr  8
:00000001802BF880 arg_8           = qword ptr  10h
:00000001802BF880 arg_10          = qword ptr  18h
:00000001802BF880 arg_18          = qword ptr  20h
:00000001802BF880
:00000001802BF880                 mov     [rsp+arg_0], rcx
:00000001802BF885                 mov     [rsp+arg_8], rdx
:00000001802BF88A                 mov     [rsp+arg_10], r8
:00000001802BF88F                 mov     [rsp+arg_18], r9
:00000001802BF894                 push    rbp
:00000001802BF895                 lea     rbp, [rsp+8+var_8]
:00000001802BF899                 sub     rsp, 20h
:00000001802BF89D                 call    NativeCodeGenerator::CheckCodeGen(Js::ScriptFunction *)
:00000001802BF8A2                 mov     rcx, rax
:00000001802BF8A5                 call    cs:__guard_check_icall_fptr
:00000001802BF8AB                 add     rsp, 20h
:00000001802BF8AF                 mov     rax, rcx
:00000001802BF8B2                 lea     rsp, [rbp+0]
:00000001802BF8B6                 pop     rbp
:00000001802BF8B7                 mov     rcx, [rsp+arg_0]
:00000001802BF8BC                 mov     rdx, [rsp+arg_8]
:00000001802BF8C1                 mov     r8, [rsp+arg_10]
:00000001802BF8C6                 mov     r9, [rsp+arg_18]
:00000001802BF8CB                 jmp     rax
```

纵横

JavaScript Function

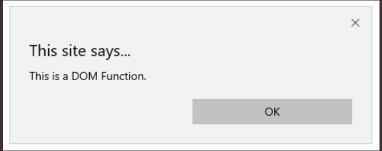
Js::ScriptFunctionType



```
0000022d`d06f0f40  00000000`0000001a
0000022d`d06f0f48  0000022d`d0670000
0000022d`d06f0f50  0000022d`d0651210
0000022d`d06f0f58  0000022d`e3f90000
0000022d`d06f0f60  00000000`00000000
0000022d`d06f0f68  00007ffd`3c50d068 chakra!Js::DeferredTypeHandler::defaultInstance
0000022d`d06f0f70  00000000`00000101
0000022d`d06f0f78  0000022d`d068df00
```

DOM Function

```
window.alert("This  is a DOM Function.");
```

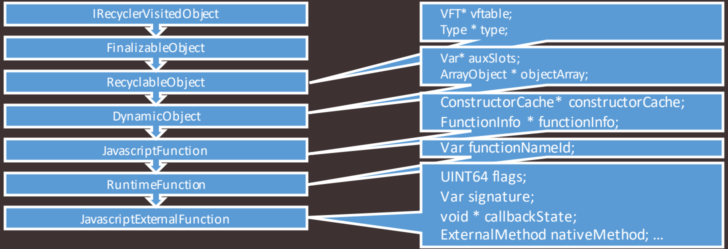


This site says...

This is a DOM Function.

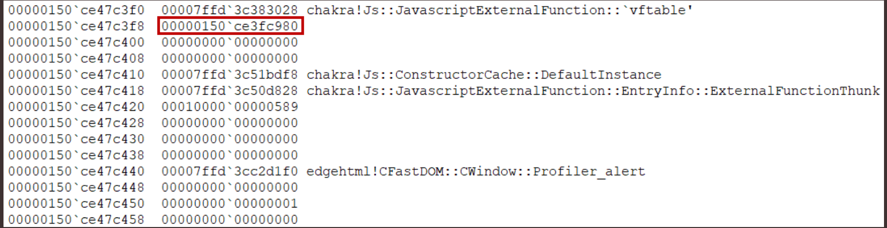
DOM Function

Js::JavascriptExternalFunction



DOM Function

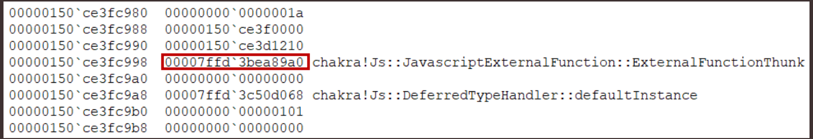
Js::JavascriptExternalFunction



```
00000150`ce47c3f0  00007ffd`3c383028 chakra!Js::JavascriptExternalFunction::`vftable'
00000150`ce47c3f8  00000150`ce3fc980
00000150`ce47c400  00000000`00000000
00000150`ce47c408  00000000`00000000
00000150`ce47c410  00007ffd`3c51bdf8 chakra!Js::ConstructorCache::DefaultInstance
00000150`ce47c418  00007ffd`3c50d828 chakra!Js::JavascriptExternalFunction::EntryInfo::ExternalFunctionThunk
00000150`ce47c420  00010000`00000589
00000150`ce47c428  00000000`00000000
00000150`ce47c430  00000000`00000000
00000150`ce47c438  00000000`00000000
00000150`ce47c440  00007ffd`3cc2d1f0 edgehtml!CFastDOM::CWindow::Profiler_alert
00000150`ce47c448  00000000`00000000
00000150`ce47c450  00000000`00000001
00000150`ce47c458  00000000`00000000
```

DOM Function

Js::Type

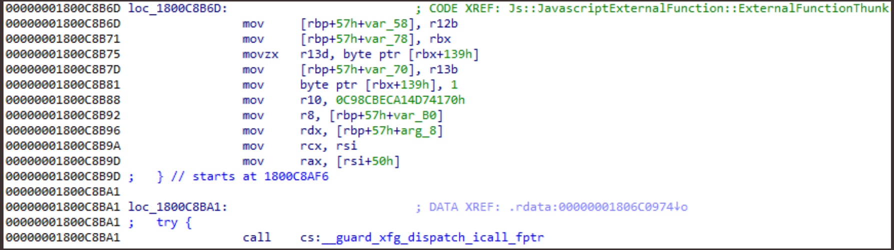


```
00000150`ce3fc980  00000000`0000001a
00000150`ce3fc988  00000150`ce3f0000
00000150`ce3fc990  00000150`ce3d1210
00000150`ce3fc998  00007ffd`3bea89a0 chakra!Js::JavascriptExternalFunction::ExternalFunctionThunk
00000150`ce3fc9a0  00000000`00000000
00000150`ce3fc9a8  00007ffd`3c50d068 chakra!Js::DeferredTypeHandler::defaultInstance
00000150`ce3fc9b0  00000000`00000101
00000150`ce3fc9b8  00000000`00000000
```

KCon

DOM Function

Js::JavascriptExternalFunction::ExternalFunctionThunk

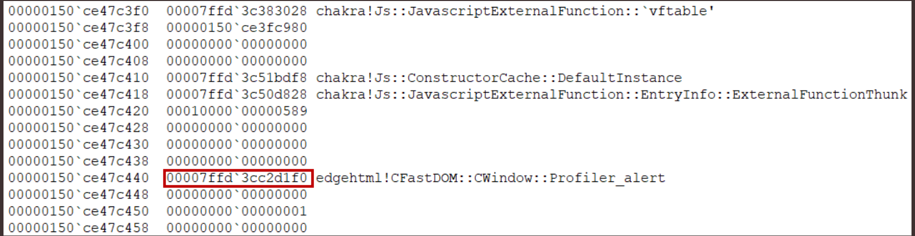


```
00000001800C8B6D loc_1800C8B6D:                          ; CODE XREF: Js::JavascriptExternalFunction::ExternalFunctionThunk
00000001800C8B6D                 mov     [rbp+57h+var_58], r12b
00000001800C8B71                 mov     [rbp+57h+var_78], rbx
00000001800C8B75                 movzx   r13d, byte ptr [rbx+139h]
00000001800C8B7D                 mov     [rbp+57h+var_70], r13b
00000001800C8B81                 mov     byte ptr [rbx+139h], 1
00000001800C8B88                 mov     r10, 0C98CBECA14D74170h
00000001800C8B92                 mov     r8, [rbp+57h+var_B0]
00000001800C8B96                 mov     rdx, [rbp+57h+arg_8]
00000001800C8B9A                 mov     rcx, rsi
00000001800C8B9D                 mov     rax, [rsi+50h]
00000001800C8B9D ;   } // starts at 1800C8AF6
00000001800C8BA1
00000001800C8BA1 loc_1800C8BA1:                          ; DATA XREF: .rdata:00000001806C0974↓o
00000001800C8BA1 ;   try {
00000001800C8BA1                 call    cs:__guard_xfg_dispatch_icall_fptr
```

纵横

DOM Function

Js::JavascriptExternalFunction::ExternalFunctionThunk



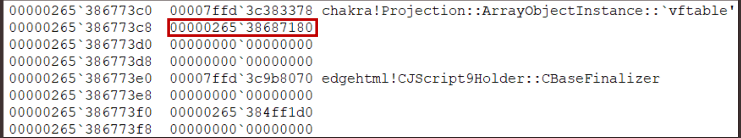
```
00000150`ce47c3f0  00007ffd`3c383028 chakra!Js::JavascriptExternalFunction::`vftable'
00000150`ce47c3f8  00000150`ce3fc980
00000150`ce47c400  00000000`00000000
00000150`ce47c408  00000000`00000000
00000150`ce47c410  00007ffd`3c51bdf8 chakra!Js::ConstructorCache::DefaultInstance
00000150`ce47c418  00007ffd`3c50d828 chakra!Js::JavascriptExternalFunction::EntryInfo::ExternalFunctionThunk
00000150`ce47c420  00010000`00000589
00000150`ce47c428  00000000`00000000
00000150`ce47c430  00000000`00000000
00000150`ce47c438  00000000`00000000
00000150`ce47c440  00007ffd`3cc2d1f0 edgehtml!CFastDOM::CWindow::Profiler_alert
00000150`ce47c448  00000000`00000000
00000150`ce47c450  00000000`00000001
00000150`ce47c458  00000000`00000000
```

DOM Getter/Setter Function

```
var s = document.createElement("script");
s.async = true;
```

DOM Getter/Setter Function

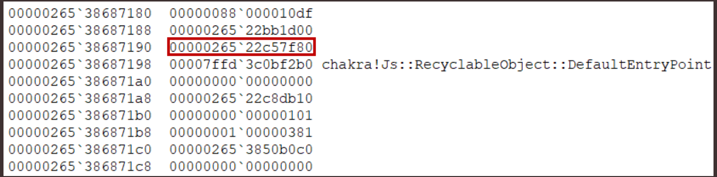
DOM Object



```
00000265`386773c0  00007ffd`3c383378 chakra!Projection::ArrayObjectInstance::`vftable'
00000265`386773c8  00000265`38687180
00000265`386773d0  00000000`00000000
00000265`386773d8  00000000`00000000
00000265`386773e0  00007ffd`3c9b8070 edgehtml!CJScript9Holder::CBaseFinalizer
00000265`386773e8  00000000`00000000
00000265`386773f0  00000265`384ff1d0
00000265`386773f8  00000000`00000000
```

DOM Getter/Setter Function

Type



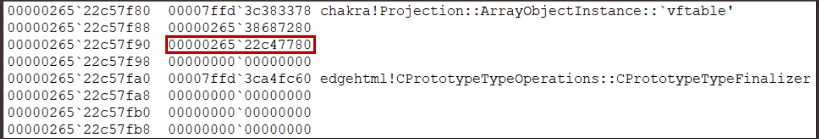
```
00000265`38687180  00000088`000010df
00000265`38687188  00000265`22bb1d00
00000265`38687190  00000265`22c57f80
00000265`38687198  00007ffd`3c0bf2b0 chakra!Js::RecyclableObject::DefaultEntryPoint
00000265`386871a0  00000000`00000000
00000265`386871a8  00000265`22c8db10
00000265`386871b0  00000000`00000101
00000265`386871b8  00000001`00000381
00000265`386871c0  00000265`3850b0c0
00000265`386871c8  00000000`00000000
```

KCon

纵横

DOM Getter/Setter Function

Prototype

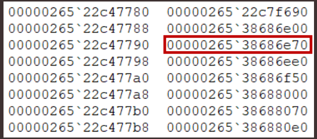


```
00000265`22c57f80  00007ffd`3c383378 chakra!Projection::ArrayObjectInstance::`vftable'
00000265`22c57f88  00000265`38687280
00000265`22c57f90  00000265`22c47780
00000265`22c57f98  00000000`00000000
00000265`22c57fa0  00007ffd`3ca4fc60 edgehtml!CPrototypeTypeOperations::CPrototypeTypeFinalizer
00000265`22c57fa8  00000000`00000000
00000265`22c57fb0  00000000`00000000
00000265`22c57fb8  00000000`00000000
```

KCon

DOM Getter/Setter Function

Functions



```
00000265`22c47780  00000265`22c7f690
00000265`22c47788  00000265`38686e00
00000265`22c47790  00000265`38686e70
00000265`22c47798  00000265`38686ee0
00000265`22c477a0  00000265`38686f50
00000265`22c477a8  00000265`38688000
00000265`22c477b0  00000265`38688070
00000265`22c477b8  00000265`386880e0
```

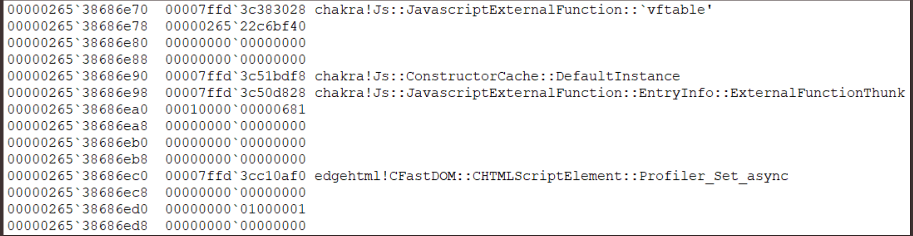
纵
横

KCon

纵横

DOM Getter/Setter Function

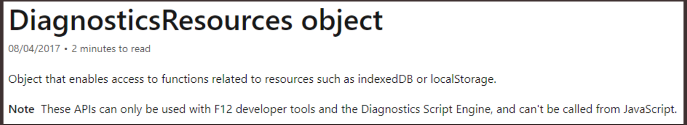
Setter Function



```
00000265`38686e70  00007ffd`3c383028 chakra!Js::JavascriptExternalFunction::`vftable'
00000265`38686e78  00000265`22c6bf40
00000265`38686e80  00000000`00000000
00000265`38686e88  00000000`00000000
00000265`38686e90  00007ffd`3c51bdf8 chakra!Js::ConstructorCache::DefaultInstance
00000265`38686e98  00007ffd`3c50d828 chakra!Js::JavascriptExternalFunction::EntryInfo::ExternalFunctionThunk
00000265`38686ea0  00010000`00000681
00000265`38686ea8  00000000`00000000
00000265`38686eb0  00000000`00000000
00000265`38686eb8  00000000`00000000
00000265`38686ec0  00007ffd`3cc10af0 edgehtml!CFastDOM::CHTMLScriptElement::Profiler_Set_async
00000265`38686ec8  00000000`00000000
00000265`38686ed0  00000000`01000001
00000265`38686ed8  00000000`00000000
```

如何利用

DiagnosticsResources



DiagnosticsResources object

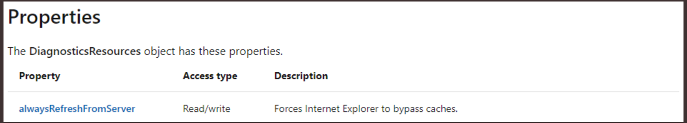
08/04/2017 • 2 minutes to read

Object that enables access to functions related to resources such as indexedDB or localStorage.

Note These APIs can only be used with F12 developer tools and the Diagnostics Script Engine, and can't be called from JavaScript.

如何利用

alwaysRefreshFromServer 属性



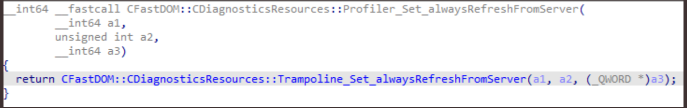
Properties

The **DiagnosticsResources** object has these properties.

Property	Access type	Description
alwaysRefreshFromServer	Read/write	Forces Internet Explorer to bypass caches.

如何利用

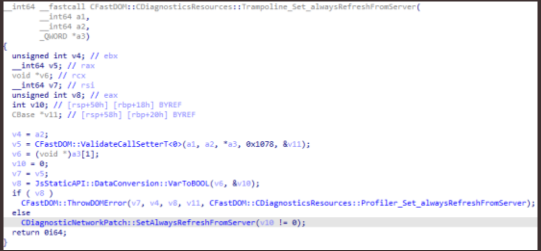
CFastDOM::CDiagnosticsResources::Profiler_Set_alwaysRefreshFromServer



```
__int64 __fastcall CFastDOM::CDiagnosticsResources::Profiler_Set_alwaysRefreshFromServer(
        __int64 a1,
        unsigned int a2,
        __int64 a3)
{
  return CFastDOM::CDiagnosticsResources::Trampoline_Set_alwaysRefreshFromServer(a1, a2, (_QWORD *)a3);
}
```

如何利用

CFastDOM::CDiagnosticsResources::Trampoline_Set_alwaysRefreshFromServer

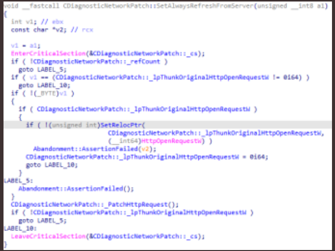


```
__int64 __fastcall CFastDOM::CDiagnosticsResources::Trampoline_Set_alwaysRefreshFromServer(
        __int64 a1,
        __int64 a2,
        _QWORD *a3)
{
  unsigned int v4; // ebx
  __int64 v5; // rax
  void *v6; // rcx
  __int64 v7; // rsi
  unsigned int v8; // eax
  int v10; // [rsp+50h] [rbp+18h] BYREF
  CBase *v11; // [rsp+58h] [rbp+20h] BYREF

  v4 = a2;
  v5 = CFastDOM::ValidateCallSetterT<0>(a1, a2, *a3, 0x1078, &v11);
  v6 = (void *)a3[1];
  v10 = 0;
  v7 = v5;
  v8 = JsStaticAPI::DataConversion::VarToBOOL(v6, &v10);
  if ( v8 )
    CFastDOM::ThrowDOMError(v7, v4, v8, v11, CFastDOM::CDiagnosticsResources::Profiler_Set_alwaysRefreshFromServer);
  else
    CDiagnosticNetworkPatch::SetAlwaysRefreshFromServer(v10 != 0);
  return 0i64;
}
```

纵
横

如何利用

CDiagnosticNetworkPatch::SetAlwaysRefreshFromServer

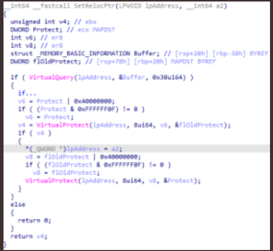


```
void __fastcall CDiagnosticNetworkPatch::SetAlwaysRefreshFromServer(unsigned __int8 a1)
{
  int v1; // ebx
  const char *v2; // rcx

  v1 = a1;
  EnterCriticalSection(&CDiagnosticNetworkPatch::_cs);
  if ( !CDiagnosticNetworkPatch::_refCount )
    goto LABEL_5;
  if ( v1 == (CDiagnosticNetworkPatch::_lpThunkOriginalHttpOpenRequestW != 0i64) )
    goto LABEL_10;
  if ( !(_BYTE)v1 )
  {
    if ( CDiagnosticNetworkPatch::_lpThunkOriginalHttpOpenRequestW )
    {
      if ( !(unsigned int)SetRelocPtr(
                            CDiagnosticNetworkPatch::_lpThunkOriginalHttpOpenRequestW,
                            (__int64)HttpOpenRequestW) )
        Abandonment::AssertionFailed(v2);
      CDiagnosticNetworkPatch::_lpThunkOriginalHttpOpenRequestW = 0i64;
      goto LABEL_10;
    }
LABEL_5:
    Abandonment::AssertionFailed();
  }
  CDiagnosticNetworkPatch::_PatchHttpRequest();
  if ( !CDiagnosticNetworkPatch::_lpThunkOriginalHttpOpenRequestW )
    goto LABEL_5;
LABEL_10:
  LeaveCriticalSection(&CDiagnosticNetworkPatch::_cs);
}
```

纵横

如何利用

SetRelocPtr



```
__int64 __fastcall SetRelocPtr(LPVOID lpAddress, __int64 a2)
{
  unsigned int v4; // ebx
  DWORD Protect; // ecx MAPDST
  int v6; // er8
  int v8; // er8
  struct _MEMORY_BASIC_INFORMATION Buffer; // [rsp+20h] [rbp-38h] BYREF
  DWORD flOldProtect; // [rsp+78h] [rbp+20h] MAPDST BYREF

  if ( VirtualQuery(lpAddress, &Buffer, 0x30ui64) )
  {
    if...
    v6 = Protect | 0x40000000;
    if ( (Protect & 0xFFFFFF0F) != 0 )
      v6 = Protect;
    v4 = VirtualProtect(lpAddress, 8ui64, v6, &flOldProtect);
    if ( v4 )
    {
      *(_QWORD *)lpAddress = a2;
      v8 = flOldProtect | 0x40000000;
      if ( (flOldProtect & 0xFFFFFF0F) != 0 )
        v8 = flOldProtect;
      VirtualProtect(lpAddress, 8ui64, v8, &Protect);
    }
  }
  else
  {
    return 0;
  }
  return v4;
}
```

纵
横

总结

CFI 是一项有效的漏洞利用缓解措施

目前的 CFI实现都只是某种程度上的近似

完整实现的 CFI 依然不能解决所有问题

感谢观看！

KCon 汇聚黑客的智慧

知道创宇 | KCon