

Canape – Bytes your Bits



James Forshaw and Michael Jordon



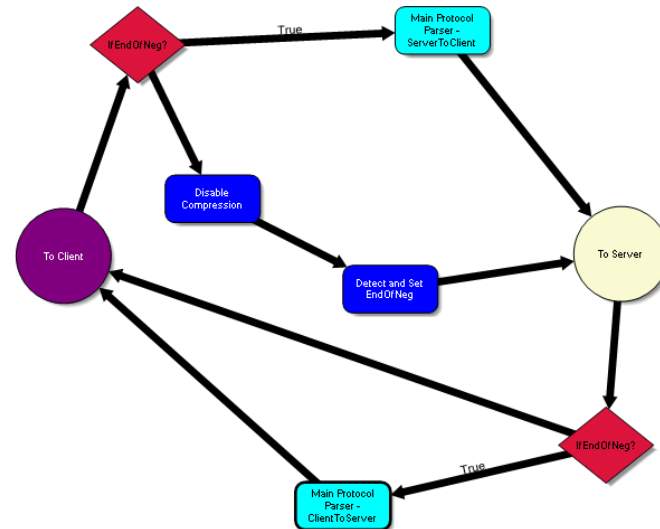
What we are going to talk about?

- New tool released at Blackhat – Canape
- What is Citrix ICA?
- In Canape:
 - MitM ICA
 - Fuzz ICA
 - Exploit ICA
- 0 Day



What is Canape?

- Binary Network Application Testing Tool
- Existing tools:
 - HTTP proxies (e.g. CAT)
 - Echo Mirage
 - Python libraries
 - Custom code
 - Wireshark
- Why a new tool?
 - Has these features and more
 - All driven through a GUI
- And it's free!





How does it MitM?

- MitM support:
 - SOCKS
 - Port forwarding
 - TCP, UDP, HTTP, Broadcast
 - SSL
- Pipelines



What is ICA?

- Protocol used for Citrix XenApp and XenDesktop products
- Remote desktop and applications
- Uses a bespoke client
- Needs a suitable configuration file to connect



Citrix Web Interface

The screenshot displays the Citrix Web Interface. At the top left is the Citrix logo. Below it is a navigation bar with 'Applications', 'Messages', and 'Preferences'. A status bar indicates 'Logged on as: administrator' and a 'Log Off' button. A search bar is located on the right. The main content area is titled 'Applications' and shows a 'Main' section with a 'Select view: Icons' dropdown. Three application icons are visible: Acrobat Reader, Mozilla Firefox, and Notepad. A hint at the bottom states: 'Hint: You can view your resources in several different ways. Use the Select view control to change the way that your resources are displayed.' A 'Problem Connecting?' link is in the bottom right corner.



The ICA File

```
[WFClient]
Version=2
TcpBrowserAddress=10.0.131.190
ICASOCKSProtocolVersion=0
ICASOCKSProxyHost=127.0.0.1
ICASOCKSProxyPortNumber=1080
```

```
[ApplicationServers]
10.0.131.190=
```

```
[10.0.131.190]
Address=10.0.131.190
InitialProgram=
```



Demo 1

- MitM ICA traffic





ICA Protocol

- Stream based protocol
- Single TCP stream
- Phases

– Hello

```
00000000  7F 7F 49 43 41 00  ..ICA.
```

– Negotiation

– Main stream

- Encrypted
- Compressed
- Multiplexed



Demo 2

- Handling state transitions





ICA Main Protocol

- Main protocol is wrapped in a simple frame
- 12 bit byte length
- 4 bit flags



Demo 3

- Parsing the framing





The 'Encryption'

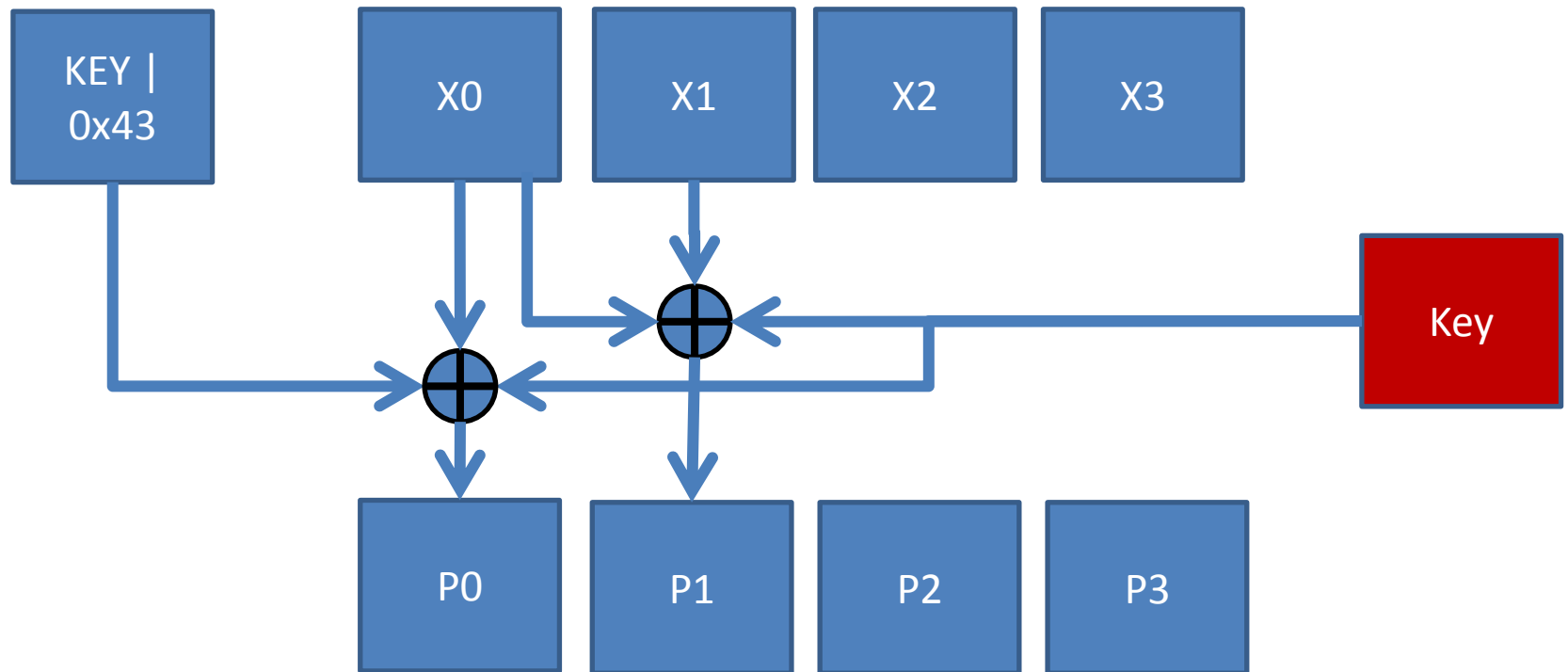
```
public EncryptProtocolDriver()
{
    super(false, g);
    h = false;
    i = false;
    l = (byte) (new Random()).nextInt();
    j = (byte) (l | 0x43);
    k = (byte) (l | 0x43);
}

private final void b(byte abyte0[], int i1, int j1)
{
    int k1 = (i1 + j1) - 1;
    byte byte0 = abyte0[k1];
    byte byte1 = l;
    for(int l1 = k1; l1 > i1; l1--)
        abyte0[l1] ^= abyte0[l1 - 1] ^ byte1;

    abyte0[i1] ^= j ^ byte1;
    j = byte0;
}
```



Encryption Diagram





Demo 4

- MitM the encrypted XOR stream





Compression

- Registry key

- HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\ICA
Client\Engine\Configuration\Advanced\Modules
\TCP/IP\Compress = Off

00000180	57	44	49	43	41	00	00	00	00	0B	3F	49	5C	A8	C1	26	WDICA.....?I\`Á&
00000190	00	BF	60	29	4B	1C	02	DC	07	0D	03	2E	3C	0F	00	00	.z`)K..Ü....<...
000001A0	00	00	00	00	00	88	13	2C	00	2C	00	FF	FF	FF	FF	01,.,.ÿÿÿÿ.
000001B0	00	00	00	00	00	1A	00	56	00	10	12	00	00	03	00	4FV.....0
000001C0	01	5A	01	43	54	58	54	57	20	20	00	09	00	43	54	58	.Z.CTXTW ...CTX
00000180	57	44	49	43	41	00	00	00	00	0B	3F	49	5C	A8	C1	26	WDICA...../I\ A&
00000190	00	BF	60	29	4B	1C	02	DC	07	0D	16	20	01	0F	00	00	.z`)K..Ü... ..
000001A0	00	00	00	00	00	88	13	2C	00	2C	00	FF	FF	FF	FF	01,.,.ÿÿÿÿ.
000001B0	00	00	00	00	00	1A	00	56	00	00	00	00	00	00	00	4FV.....0
000001C0	01	5A	01	43	54	58	54	57	20	20	00	09	00	43	54	58	.Z.CTXTW ...CTX



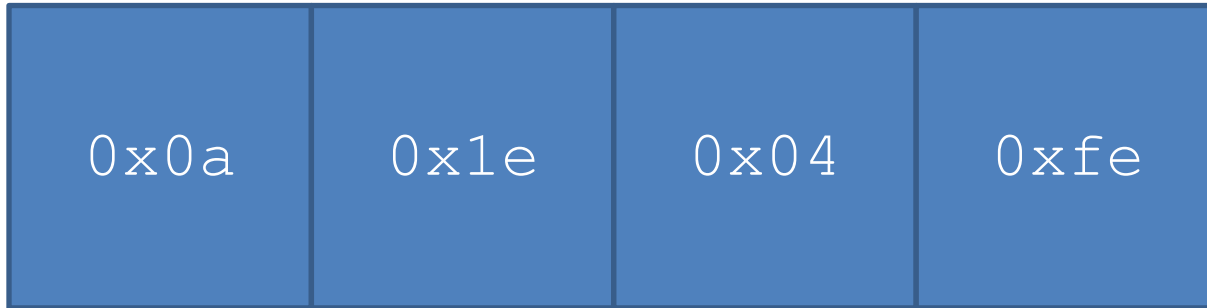
Demo 5

- Downgrade to no compression
- Replace:
 - 0x00 0x10 0x12 =>
0x00 0x00 0x00





Key Press



Type

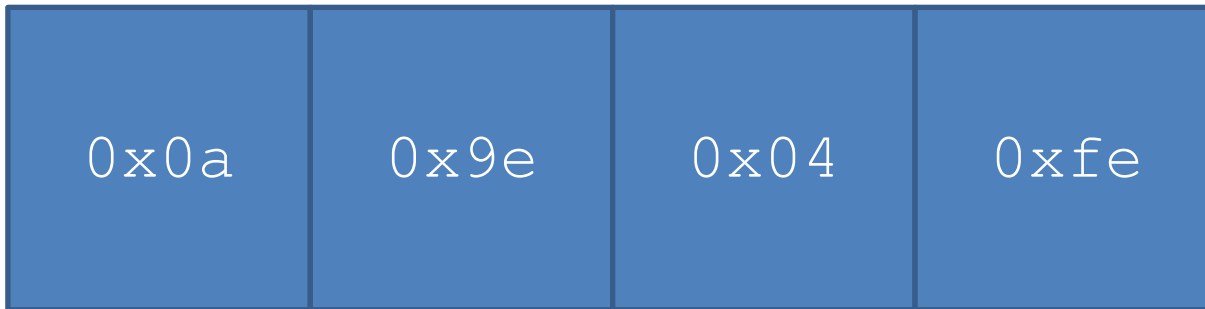
Scan Code

?

End Marker



A





Mouse Movement

0x0d	0x2acd	0x1fa7	0x01	0x0C	0xfe
Type	X Coordinate	Y Coordinate	Button State	?	End Marker

Button State

- 01 – No Buttons
- 02 – Press Left
- 04 – Release left
- 08 – Press Right
- 10 – Release Right



Fuzzing

- Standard fuzzing
 - But we are in the encrypted and compressed stream
- Byte fuzzing



Demo 6

- Fuzz the contents of the encrypted stream





OllyDbg - wfica32.exe - [CPU - main thread, module VDTW30N]

File View Debug Plugins Options Window Help

LEMTWHC / KBR ... S

6692730B	8B7C24 1C	MOV EDI,DWORD PTR SS:[ESP+1C]	
6692730F	8B7424 24	MOV ESI,DWORD PTR SS:[ESP+24]	
669273E3	8B4C24 28	MOV ECX,DWORD PTR SS:[ESP+28]	
669273E7	EB 05	JMP SHORT VDTW30N.669273EE	
669273E9	66:85DB	TEST BX,BX	
669273EC	75 5C	JNZ SHORT VDTW30N.6692744A	
669273EE	66:8907	MOV WORD PTR DS:[EDI],AX	
669273F1	66:895F 02	MOV WORD PTR DS:[EDI+2],BX	
669273F5	66:8B81 C817936	MOV AX,WORD PTR DS:[ECX+669317C8]	
669273FC	66:8947 08	MOV WORD PTR DS:[EDI+8],AX	
66927400	8B4C24 2C	MOV ECX,DWORD PTR SS:[ESP+2C]	
66927404	25 FFFF0000	AND EAX,0FFFF	
66927409	C1E0 04	SHL EAX,4	
6692740C	8B4408 08	MOV EAX,DWORD PTR DS:[EAX+ECX+8]	
66927410	85C0	TEST EAX,EAX	
66927412	74 08	JE SHORT VDTW30N.6692741C	
66927414	8B5424 3C	MOV EDX,DWORD PTR SS:[ESP+3C]	
66927418	57	PUSH EDI	

Registers (FPU)

EAX E10BFFB7
ECX 000BFC94
EDX 000B00B7
EBX DCDC0000
ESP 0012EEEC
EBP 00E108AF
ESI 017B6CA8
EDI 0187693C
EIP 669273F5 VDTW30N.66927

C 0 ES 0023 32bit 0(FFFFFF)
P 1 CS 001B 32bit 0(FFFFFF)
A 0 SS 0023 32bit 0(FFFFFF)
Z 1 DS 0023 32bit 0(FFFFFF)
S 0 FS 003B 32bit 7FFDF00
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_SUCCESS
EFL 00000246 (NO_NB_E_RE_N

Address	Hex dump	ASCII
669317C8	00 00 00 00 01 00 02 00	...0.0.
669317D0	F8 45 37 00 00 00 00 00	°E7....
669317D8	02 00 0A 00 10 49 37 00	0...I7.
669317E0	01 00 00 00 00 00 00 00	0.....
669317E8	00 00 00 00 00 00 00 00

Access violation when reading [669F145C] - use Shift+F7/F8/F9 to pass exception to program

Paused

Access violation when reading [669F145C] -



Example Citrix ICA Client Bug

- Old, reported February 2008
- Fixed August 2010
- Affected clients on:
 - Windows
 - Mac
 - Linux
 - Solaris
 - Windows Mobile
- Demo on Windows XP SP2

<http://support.citrix.com/article/CTX125975>

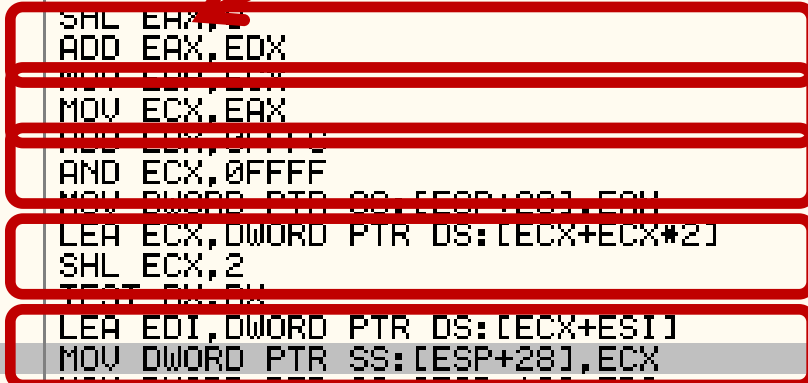


```
669273DB 8B7C24 1C MOV EDI,DWORD PTR SS:[ESP+1C]
669273DF 8B7424 24 MOV ESI,DWORD PTR SS:[ESP+24]
669273E3 8B4C24 28 MOV ECX,DWORD PTR SS:[ESP+28]
669273E7 v EB 05 JMP SHORT UDTW30N.669273EE
669273E9 66:85DB TEST BX,BX
669273EC v 75 5C JNZ SHORT UDTW30N.6692744A
669273EE 66:8907 MOV WORD PTR DS:[EDI],AX
669273F1 66:895F 02 MOV WORD PTR DS:[EDI+2],BH
669273F5 66:8B81 C817935 MOV AX,WORD PTR DS:[ECX+669317C8]
669273FC 66:8947 08 MOV WORD PTR DS:[EDI+8],AX
66927400 8B4C24 2C MOV ECX,DWORD PTR SS:[ESP+2C]
66927404 25 FFFF0000 AND EAX,0FFFF
66927409 C1E0 04 SHL EAX,4
6692740C 8B4408 08 MOV EAX,DWORD PTR DS:[EAX+ECX+8]
66927410 85C0 TEST EAX,EAX
66927412 v 74 08 JE SHORT UDTW30N.6692741C
66927414 8B5424 3C MOV EDX,DWORD PTR SS:[ESP+3C]
66927418 57 PUSH EDI
66927419 52 PUSH EDX
6692741A FF00 CALL EAX
6692741C 004424 44 MOV DL,BYTE PTR DS:[ECX+44]
```



```
66927324 8A4D 01 MOV CL, BYTE PTR SS:[EBP+1]
66927327 81E2 FFFF0000 AND EDX, 0FFFF
6692732D 03EA ADD EBP, EDX
6692732F 894424 30 MOV DWORD PTR SS:[ESP+30], EAX
66927333 84C9 TEST CL, CL
66927335 884C24 44 MOV BYTE PTR SS:[ESP+44], CL
66927339 v0F84 F1000000 JE VDTW30N.66927430
6692733F 8B4C24 48 MOV ECX, DWORD PTR SS:[ESP+48]
66927343 81E1 FFFF0000 AND ECX, 0FFFF
66927349 03C8 ADD ECX, EAX
6692734B 894C24 14 MOV DWORD PTR SS:[ESP+14], ECX
6692734F 8B4424 14 MOV EAX, DWORD PTR SS:[ESP+14]
66927353 8D4D 04 LEA ECX, DWORD PTR SS:[EBP+4]
66927356 3BC8 CMP ECX, EAX
66927358 v0F87 EC000000 JA VDTW30N.6692744A
6692735E 66:0FB65D 01 MOVZX BX, BYTE PTR SS:[EBP+1]
66927363 66:0FB655 00 MOVZX DX, BYTE PTR SS:[EBP]
66927368 66:0FB645 03 MOVZX AX, BYTE PTR SS:[EBP+3]
6692736D C1E3 08 SHL EBX, 8
66927370 03DA ADD EBX, EDX
66927372 66:0FB655 02 MOVZX DX, BYTE PTR SS:[EBP+2]
66927377 C1E0 08 SHL EAX, 8
6692737A 03C2 ADD EAX, EDX
6692737C 8BE9 MOV EBP, EAX
6692737E 8BC8 MOV ECX, EAX
66927380 81C3 FCFF0000 AND EDI, 0FFF
66927386 81E1 FFFF0000 AND ECX, 0FFFF
6692738C 894424 20 MOV DWORD PTR SS:[ESP+20], EAX
66927390 8D0C49 LEA ECX, DWORD PTR DS:[ECX+ECX*2]
66927393 C1E1 02 SHL ECX, 2
66927396 66:85DB TEST EBX, EBX
66927399 8D3C31 LEA EDI, DWORD PTR DS:[ECX+ESI]
6692739C 894C24 28 MOV DWORD PTR SS:[ESP+28], ECX
669273A0 897C24 1C MOV DWORD PTR SS:[ESP+1C], EDI
```

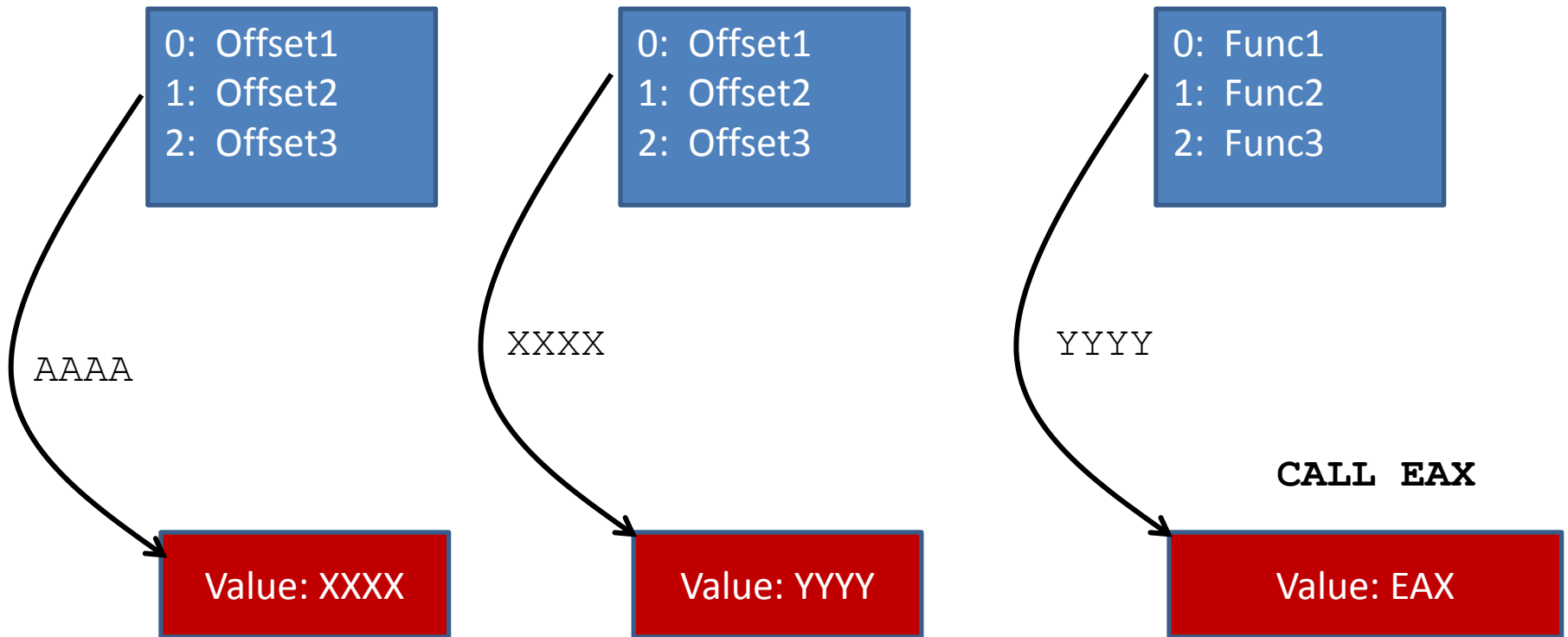
We Control





Offset Value

Control Offset: AAAA





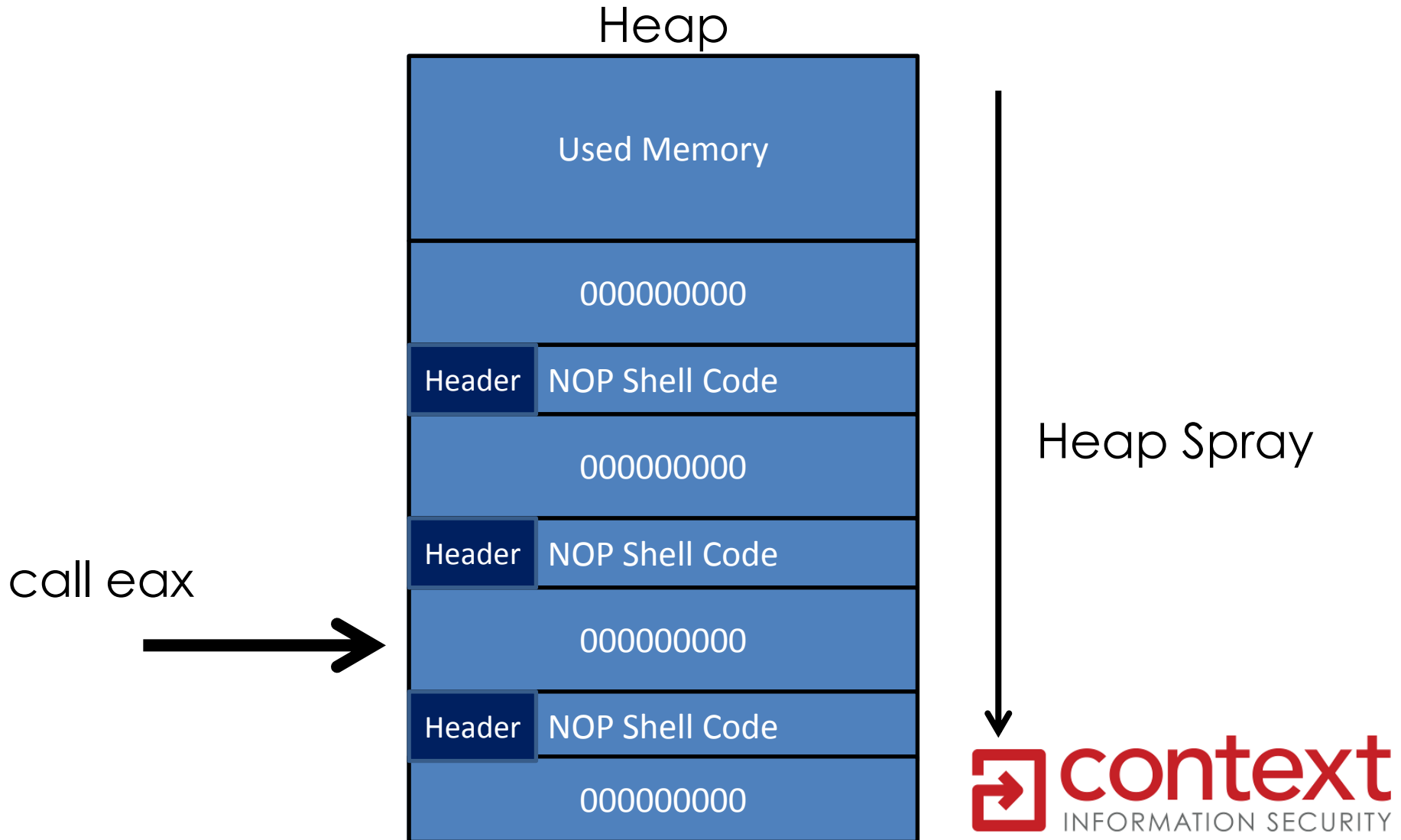
Demo 7

- Brute force the value to find a heap offset





Heap Spray





Easy Heap Spray

Packet Buffer →



Packet Copied



Heap Header

Valid Pointer



0000

=>

ADD BYTE PTR [EAX], AL



0



2

4



5

6

7

8

Control Heap Spray Size

Random

81 00 =>

ADD DWORD PTR DS:[EAX], PrevSize_Cookie_Flags



Exec Heap Header

```
01F2F24C 0000
01F2F24E 0000
01F2F250 0000
01F2F252 0000
01F2F254 0000
01F2F256 0000
01F2F258 0000
01F2F25A 0000
01F2F25C 0000
01F2F25E 0000
01F2F260 8100 B1016901
01F2F266 0C 03
01F2F268 90
01F2F269 90
01F2F26A 90
01F2F26B 90
01F2F26C 90
01F2F26D 90
01F2F26E 90
```

```
ADD BYTE PTR DS:[EAX],AL
ADD BYTE PTR DS:[EAX],AL
ADD BYTE PTR DS:[EAX],AL
ADD BYTE PTR DS:[EAX],AL
ADD BYTE PTR DS:[EAX],AL
ADD BYTE PTR DS:[EAX],AL
ADD BYTE PTR DS:[EAX],AL
ADD BYTE PTR DS:[EAX],AL
ADD BYTE PTR DS:[EAX],AL
ADD DWORD PTR DS:[EAX],16901B1
OR AL,3
NOP
NOP
NOP
NOP
NOP
NOP
NOP
```

EAX pointer to valid memory



Our NOP Sled and Shellcode



Demo 8 "Root"

- HTTP send ICA file
- Replay negotiation
- Prime the heap – large packet
- Spray the heap x 5000 – small packet big Len
- Send payload trigger packet





Demo 9 "Other Examples"

- The Power of Canape!





Demo 10 "0Day"

- Demo only, sorry 😞





Questions

- Please fill in your feedback forms



References

- <http://canape.contextis.com>
- Twitter: @ctxis
- Email: canape@contextis.com