
Patchwork

Stitching against malware families with IDA Pro

Daniel Plohmann

plohmann@cs.uni-bonn.de



Some words about myself

■ Personal background

- PhD student and researcher at University of Bonn & Fraunhofer FKIE
 - Research focus: Efficiency of Reverse Engineering
 - Work focus: malware analysis and botnet mitigation

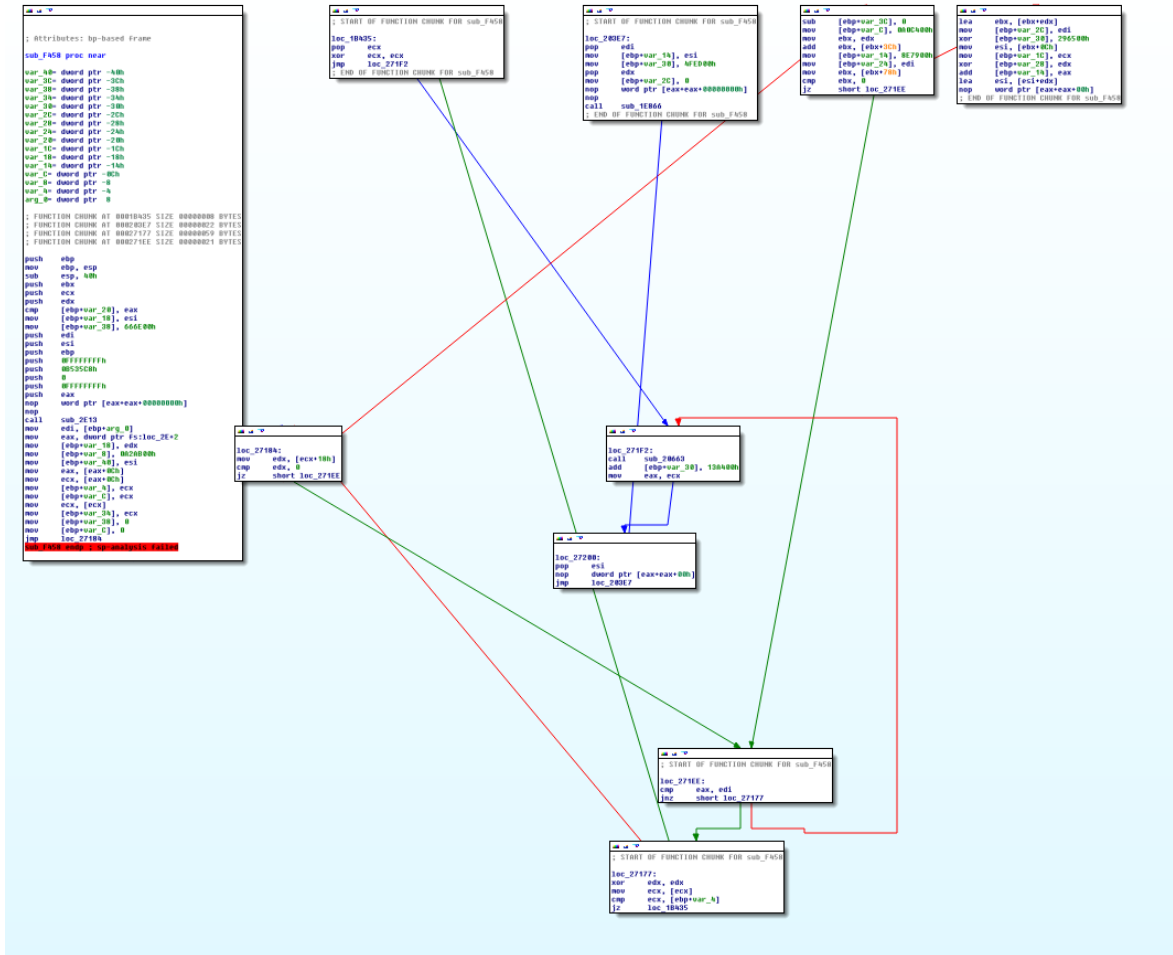
■ Related projects

- [1] PyBox (python sandboxing toolkit)
- [2] IDAScope (IDA Pro enhancements for malware RE)

[1] <http://code.google.com/p/pyboxed>

[2] <https://idascope.pnx.tf>

Patchwork Motivation



Patchwork

... in a nutshell

- Patchwork = refurbished (IDA)PyEmu [1] + set of convenience functions

 Work in progress

- Driving ideas:

- A flexible framework that allows data transformations aiding static analysis
- Get away from the (throwaway-)snippets-per-case approach

 Sharing is caring!

[1] <https://github.com/codypierce/pyemu> (2009 / 2012)

Patchwork

Wishlist

- Seamless integration with IDA
- Instrumentalization of analysis target's native code
 - But don't actually run code (= no debugging, AppCall, PIN, ...)
- Reusability / generalization

- Notable emulation solutions compatible with IDA:
 - [1] Ida-x86emu: standalone plugin, no extendability
 - [2] (IDA)PyEmu: python-based, fully scriptable
 - Incomplete (limited to most common opcodes)
Outdated (state of 2009)

[1] <http://www.idabook.com/ida-x86emu>

[2] <https://github.com/codypierce/pyemu>

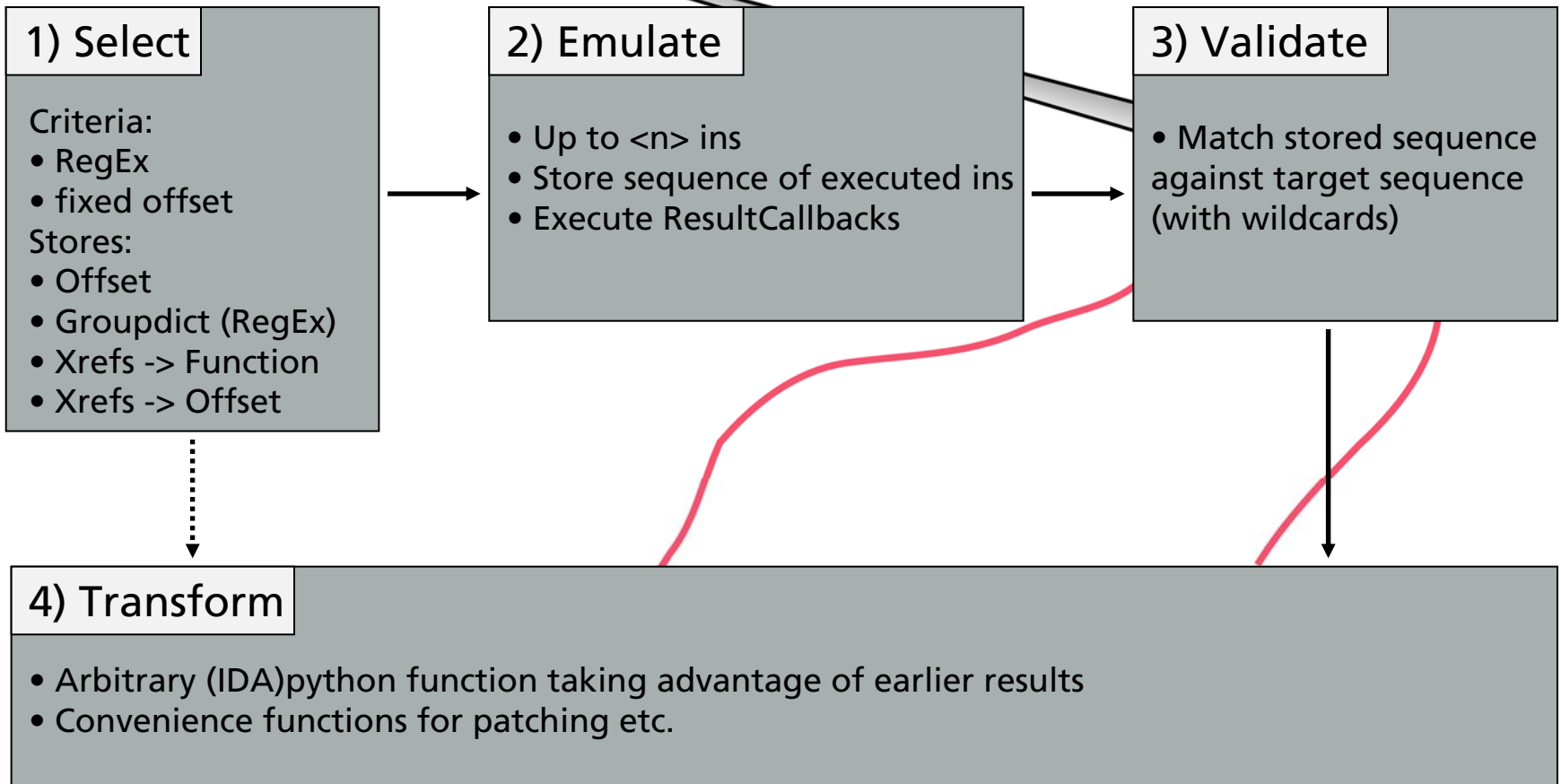
(IDA)PyEmu

Workflow

- Pretty straight-forward :)
- Set initial emulation state
 - Allocate + fill virtual memory
 - Create context (stack / registers + EIP)
- Emulate step by step
 - Proceed with modified state (algorithmic results, transformed memory)

Patchwork Workflow

■ A „stitch“:



Patchwork

Example: Nymaim

- Dropper / Ransom malware family [1]
 - Written in assembler, heavily obfuscated
 - Control flow obfuscation (call/jmp redirection)
 - Obfuscated stack/register usage (delegated to subfunction)
 - Obfuscated stack usage (introduction of many irrelevant fields)
- Hashed API calls

[1] <http://www.welivesecurity.com/2013/08/26/nymaim-obfuscation-chronicles/>

Patchwork

Example: Nymaim (Control Flow Obfuscation)

```
000BA2BB    push    eax
000BA2BC    push    0B0B48F89h
000BA2C1    push    4F4AD544h
000BA2C6    call   sub_9D009
000BA2CB    mov     eax, ebx
```

```
000BA2C3    nop
000BA2C4    nop
000BA2C5    nop
000BA2C6    call   sub_B0798
000BA2CB    mov     eax, ebx
```

```
; Attributes: bp-based frame
sub_9D009 proc near
arg_0= dword ptr  8
arg_4= dword ptr  0Ch
arg_8= dword ptr  10h
; FUNCTION CHUNK AT 000B4327 SIZE 00000008 BYTES
push    ebp
mov     ebp, esp
push    eax
mov     eax, [ebp+8]
mov     [ebp+arg_8], eax
mov     eax, [ebp+arg_4]
add     eax, [ebp+arg_0]
jmp     loc_B4327
sub_9D009 endp

; START OF FUNCTION CHUNK FOR sub_9D009
loc B4327:
add     [ebp+8], eax
pop     eax
leave
retn   8
; END OF FUNCTION CHUNK FOR sub_9D009
```

Arg_0: Displacement offset part 0 (0x4F4AD544)
Arg_4: Displacement offset part 1 (0xB0B48F89)
Arg_8: Placeholder for original return address

Function prologue

Save original return address

Calculate displacement

Apply displacement to return address

Clean up and detour to displaced return address

Patchwork

Example: Nymaim (Control Flow Obfuscation)

OllyDbg - svchost.exe

File View Debug Trace Plugins Options Windows Help

CPU - thread 2. (00000960)

Address	Hex dump	Assembly	Comment
000BA2A1	8B5D 10	MOV EBX, DWORD PTR SS:[EBP+10]	
000BA2A4	39C3	CMP EBX, EAX	
000BA2A6	72 25	JB SHORT 000BA2C0	
000BA2A8	89C3	MOV EBX, EAX	
000BA2AA	6A 33	PUSH 33	
000BA2AC	E8 2861FEFF	CALL 000A03D9	
000BA2B1	6A 36	PUSH 36	
000BA2B3	E8 2161FEFF	CALL 000A03D9	
000BA2B8	FF75 0C	PUSH DWORD PTR SS:[EBP+0C]	
000BA2B8	30	POP EAX	
000BA2B8	68 898FB400	PUSH 80B48F89	
000BA2C1	68 44D54A4F	PUSH 4F4A0544	
000BA2C6	E8 3E2DFEFF	CALL 00090009	
000BA2CB	8908	MOV EAX, EBX	
000BA2CD	59	POP ECX	
000BA2CE	5F	POP EDI	
000BA2CF	5E	POP ESI	
000BA2D0	5A	POP EDX	
000BA2D1	5B	POP EBX	
000BA2D2	C9	LEAVE	
000BA2D3	C2 0C00	RETN 0C	
000BA2D6	8B45 0C	MOV EAX, DWORD PTR SS:[EBP+0C]	
000BA2D9	8B4D 08	MOV ECX, DWORD PTR SS:[EBP+8]	
000BA2E1	E9 EED1FFFF	JMP 000B74CF	
000BA2E1	55	PUSH EBP	
000BA2E2	89E5	MOV EBP, ESP	
000BA2E4	57	PUSH EDI	
000BA2E5	833D 31FA0C00	CMP DWORD PTR DS:[0CF831], 1	
000BA2EC	0F84 08CFDFF	JE 00092FC2	
000BA2F2	68 E0930000	PUSH 3E8	
000BA2F7	6A 64	PUSH 64	
000BA2F9	57	PUSH EDI	
000BA2FA	68 15803229	PUSH 29328015	
000BA2FF	68 93E33429	PUSH 2934E393	
000BA304	E8 8CEBFFFF	CALL 000B8E95	
000BA309	EB DA	JMP SHORT 000BA2E5	
000BA309	8B4D 08	MOV ECX, DWORD PTR SS:[EBP+8]	

Stack [00A8FEC4]=000CE166
EAX=0

Address	Hex dump	ASCII
01004000	70 75 73 68 20 70 6C 61	push placeholder
01004010	66 6F 72 20 6F 72 69 67	for original
01004020	72 65 74 75 72 6E 20 61	return address
01004030	00 00 00 00 00 00 00 00	
01004040	00 00 00 00 00 00 00 00	
01004050	00 00 00 00 00 00 00 00	
01004060	00 00 00 00 40 BB 00 00	
01004070	00 00 00 00 00 00 00 00	
01004080	00 00 00 00 00 00 00 00	
01004090	00 00 00 00 00 00 00 00	
010040A0	00 00 00 00 00 00 A8 10	
010040B0	BF 44 FF FF 00 00 00 00	

Registers (32Now!)

Register	Value
EAX	00000000
ECX	00A8FEC8
EDX	7C90E4F4 ntdll.KiFastSystemCallRet
EBX	000A2A37
ESP	00A8FEC8
EBP	00A8FEE4
ESI	095C0003
EDI	7C9115EF ntdll.7C9115EF
EIP	000BA2B8
C 0	ES 0023 32bit 0(FFFFFFFF)
P 1	CS 001B 32bit 0(FFFFFFFF)
A 0	SS 0023 32bit 0(FFFFFFFF)
Z 1	DS 0023 32bit 0(FFFFFFFF)
S 0	FS 003B 32bit 7FFD0000(FFF)
T 0	GS 0000 NULL
D 0	
O 0	LastErr 00000000 ERROR_SUCCESS
EFL	00010246 (NO, NB, E, BE, NS, PE, GE, LE)
MM0	5.995635e+36 5.981526e+36
MM1	8.399271e-40 5.989963e+36
MM2	5.981526e+36 5.989963e+36
MM3	6.011055e+36 9.934225e-40
MM4	5.989963e+36 6.011055e+36
MM5	8.399131e-40 1.404938e-38
MM6	6.011055e+36 5.647233e-43
MM7	1.404938e-38 5.995635e+36
XMM0	7C903400 7C90B5FD 00000193 00000017
XMM1	000B0000 0098FD44 7C9056F4 7C900000
XMM2	00001000 0098F0BC 7C906469 7C90B5FD
XMM3	7C90CF50 00000040 0098F0C4 00000000
XMM4	7C90B5FD 00000193 00000150 0098FC44
XMM5	0098FC60 7C9056F4 7C900000 7C903400
XMM6	00000193 00092560 7C903400 00092556
XMM7	7C9056F4 7C900000 7C903400 7C90B5FD
MXCSR	00001F80 FZ 0 DZ 0 Err 0 0 0 0 0 Rnd NEAR Mask 1 1 1 1 1 1

00A8FEC8 7C9001FC RETN to ntdll.ZwDelayExecution+0C

00A8FEC8 7C9115EF ntdll.ZwDelayExecution+0C

00A8FED0 00000000

00A8FED4 00A8FF00

00A8FED8 00000700

00A8FEDC 095C0003

00A8FEE0 00A8FFA4

00A8FEE4 00A8FF03

00A8FEE8 000AD0E2 RETURN from 000908D7 to 000AD0E2

00A8FEEC 7C9001FB ntdll.ZwDelayExecution

00A8FEF0 00000002

00A8FEF4 000BA3C0 RETURN from 000908BF to 000BA3C0

00A8FEF8 00000000

Patchwork

Example: Nymaim (Control Flow Obfuscation)

The screenshot displays the OllyDbg interface for the process 'svchost.exe'. The main window shows assembly code for CPU-thread 2. (00000960). The registers window on the right shows the current state of the CPU registers, with EIP pointing to 000BA2C6. The hex dump window at the bottom shows the memory dump for the instruction at address 01004000, which is a call instruction to 4F4AD544.

Address	Hex dump	ASCII
01004000	63 61 6C 6C 20 20 20 20 20 20 20 20 20 20 20 20	call
01004010	64 65 74 6F 75 72 20 20 20 20 20 20 20 20 20 20	detour
01004020	66 75 6E 63 74 69 6F 6E 20 20 20 20 20 20 20 20	function
01004030	20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004060	00 00 00 00 40 BB 00 00 00 00 00 00 00 00 00 00	
01004070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004100	BF 44 FF FF 00 00 00 00 00 00 00 00 00 00 00 00	

Patchwork

Example: Nymaim (Control Flow Obfuscation)

The screenshot displays the OllyDbg interface for the process 'svchost.exe'. The main window shows assembly code for CPU thread 2 (0000960). The code includes instructions like POP, MOV, JMP, PUSH, and LEA, with various registers and memory addresses. The registers window on the right shows the current state of registers, including EAX, ECX, EDX, EBX, ESP, EBP, ESI, EDI, and EIP. The hex dump at the bottom shows the raw bytes of the code, with some bytes highlighted in red and green. The ASCII column shows the corresponding characters for the hex dump.

Address	Hex dump	ASCII
01004000	66 75 6E 63 74 69 6F 6E 20 20 20 20 20 20 20 20	function
01004010	70 72 6F 6C 6F 67 75 65 20 20 20 20 20 20 20 20	prologue
01004020	20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
01004030	20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004060	00 00 00 00 40 BB 00 00 00 00 00 00 00 00 00 00	
01004070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040B0	BF 44 FF FF 00 00 00 00 00 00 00 00 00 00 00 00	

Patchwork

Example: Nymaim (Control Flow Obfuscation)

The screenshot displays the OllyDbg interface for the process 'svchost.exe'. The main window shows assembly code for CPU thread 2 (ID: 00000960). The code includes instructions such as POP EDI, MOV ECX, DWORD PTR DS:[ECX], and various jumps (JMP) to different addresses. The registers window on the right shows the current state of registers, with EIP at 00090013. The stack window shows the current stack frame, with EAX at 000BA2CB. The memory dump window at the bottom shows a hex dump of memory starting at address 01004000, with ASCII characters 'f', 'd', 'i', 's', 'p', 'l', 'a', 'c', 'e', 'm', 'e', 'n', 't', 'o', 'p', 'e', 'r', 'a', 'n', 'd' visible.

Patchwork

Example: Nymaim (Control Flow Obfuscation)

The screenshot displays the OllyDbg interface for the process svchost.exe. The main window shows assembly code for CPU thread 2 (00000960). The registers window on the right shows the current state of registers, with EAX at 00000000 and EIP at 0009D019. The memory dump at the bottom shows a sequence of bytes, with a comment indicating a jump to a basic block.

Address	Hex dump	ASCII
01004000	67 6F 20 74 6F 20 73 65 63 6F 6E 64 20 20 20 20	go to second basic block
01004010	62 61 73 69 63 20 62 6C 6F 63 68 20 20 20 20 20	
01004020	20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
01004030	20 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004060	00 00 00 00 40 BB 00 00 00 00 00 00 00 00 00	
01004070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040C0	BF 44 FF FF 00 00 00 00 00 00 00 00 00 00 00	

Patchwork

Example: Nymaim (Control Flow Obfuscation)

The screenshot displays the OllyDbg interface for the process svchost.exe. The main window shows assembly code for CPU thread 2 (00000960). The registers window on the right shows the current state of registers, with EAX at 00000000 and EIP at 000B4327. The memory dump at the bottom shows a sequence of bytes with hex and ASCII representations.

Assembly Code:

```
000B42D2 8B07 MOV EAX, DWORD PTR DS:[EDI]
000B42D4 81FD 203C4D39 CMP EBP, 394D3C20
000B42DA 8B40 14 MOV ECX, DWORD PTR SS:[EBP+14]
000B42DD 81FD 203D4D09 CMP EBP, 94D3D20
000B42E3 81F9 6EFA0C00 CMP ECX, 0CFA6E
000B42E9 0F84 3015FEFF JE 0009581F
000B42EF 81F9 4FFC0C00 CMP ECX, 0CFC4F
000B42F5 0F84 4544FFFF JG 000A8740
000B42FB 8935 B4FCFFFF MOV DWORD PTR SS:[EBP-34C], EDX
000B4301 8985 ACFCFFFF MOV DWORD PTR SS:[EBP-354], EAX
000B4307 C785 B8FCFFFF MOV DWORD PTR SS:[EBP-343], 0
000B4311 E9 3544FFFF JMP 000A874B
000B4316 89E5 MOV EBP, ESP
000B4318 50 PUSH EAX
000B4319 8B45 04 MOV EAX, DWORD PTR SS:[EBP+4]
000B431C 8945 10 MOV DWORD PTR SS:[EBP+10], EAX
000B431F 8B45 08 MOV EAX, DWORD PTR SS:[EBP+8]
000B4322 E9 2B8DFE9F JMP 0009D052
000B4327 0145 04 ADD DWORD PTR SS:[EBP+4], EAX
000B4329 58 POP EAX
000B432B C9 LEAVE
000B432C C2 0800 RETN 8
000B432F 0145 04 ADD DWORD PTR SS:[EBP+4], EAX
000B4332 58 POP EAX
000B4334 C9 LEAVE
000B4337 C2 0800 RETN 8
000B4339 6A 03 PUSH 3
000B433B E9 3973FF9F JMP 000AB6C7
000B433E 55 PUSH EBP
000B4340 89E5 MOV EBP, ESP
000B4342 83EC 24 SUB ESP, 24
000B4344 8955 E4 MOV DWORD PTR SS:[EBP-1C], EDX
000B4347 8145 EC 00F877 ADD DWORD PTR SS:[EBP-14], 77F800
000B434E 56 PUSH ESI
000B4350 57 PUSH EDI
000B4352 8175 E8 003524 XOR DWORD PTR SS:[EBP-18], 00243500
000B4354 8955 E4 MOV DWORD PTR SS:[EBP-1C], EDX
```

Registers (32Now!):

```
EAX 00000000
ECX 00000000
EDX 7C90E4F4 ntdll.KiFastSystemCallRet
EBX 00000000
ESP 0008FEB0
EBP 0008FEB4
ESI 095C0003
EDI 7C9115EF ntdll.7C9115EF
EIP 000B4327
C 0 ES 0023 32bit 0(FFFFFFFF)
P 0 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 0 DS 0023 32bit 0(FFFFFFFF)
S 1 FS 003B 32bit 7FFD0000(FFF)
T 0 GS 0000 NULL
D 0
O 0 LastErr 00000000 ERROR_SUCCESS
EFL 00000282 (NO, NB, NE, A, S, PO, L, LE)
```

Memory Dump:

Address	Hex dump	ASCII
01004000	61 64 64 20 64 69 73 70 6C 61 63 65 60 65 6E 74	add displacement
01004010	74 6F 20 72 65 74 75 72 6E 20 20 20 20 20 20 20	to return
01004020	61 64 64 72 65 73 73 20 20 20 20 20 20 20 20 20	address
01004030	20 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004060	00 00 00 00 40 BB 00 00 00 00 00 00 00 00 00	
01004070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040A0	00 00 00 00 00 00 00 00 08 10 00 01	
010040B0	BF 44 FF FF 00 00 00 00 00 00 00 00 00 00 00	

Patchwork

Example: Nymaim (Control Flow Obfuscation)

The screenshot displays the OllyDbg interface for the process svchost.exe. The main window shows assembly code for CPU thread 2 (00000960). The registers window on the right shows the current state of the CPU registers, with EIP pointing to 000B432A. The memory dump window at the bottom shows the stack content, including a return address of 000B432A and a return value of 00000000.

Assembly Code:

```
000B42D2 8B07 MOV EAX,DWORD PTR DS:[EDI]
000B42D4 81FD 203C4D39 CMP EBP,394D3C20
000B42DA 8B40 14 MOV ECX,DWORD PTR SS:[EBP+14]
000B42DD 81FD 203D4D09 CMP EBP,94D3D20
000B42E3 81F9 6EFA0C00 CMP ECX,0CFA6E
000B42E9 0F84 3015FEFF JE 0009581F
000B42EF 81F9 4FFC0C00 CMP ECX,0CFC4F
000B42F5 0F84 4544FFFF JE 000A8740
000B42FB 8935 B4FCFFFF MOV DWORD PTR SS:[EBP-34C],EDX
000B4301 8985 ACFCFFFF MOV DWORD PTR SS:[EBP-354],EAX
000B4307 C785 B8FCFFFF MOV DWORD PTR SS:[EBP-343],0
000B4311 E9 3544FFFF JMP 000A874B
000B4316 89E5 MOV EBP,ESP
000B4318 50 PUSH EAX
000B4319 8B45 04 MOV EAX,DWORD PTR SS:[EBP+4]
000B431C 8945 10 MOV DWORD PTR SS:[EBP+10],EAX
000B431F 8B45 08 MOV EAX,DWORD PTR SS:[EBP+8]
000B4322 E9 2B8DFE9F JMP 0009D052
000B4327 0145 04 ADD DWORD PTR SS:[EBP+4],EAX
000B432A 58 POP EAX
000B432B C9 LEAVE
000B432C C2 0800 RETN 8
000B432F 0145 04 ADD DWORD PTR SS:[EBP+4],EAX
000B4332 58 POP EAX
000B4333 C9 LEAVE
000B4334 C2 0800 RETN 8
000B4337 6A 03 PUSH 3
000B4339 E9 3973FFFF JMP 000AB6C7
000B433E 55 PUSH EBP
000B433F 89E5 MOV EBP,ESP
000B4341 83EC 24 SUB ESP,24
000B4344 8955 E4 MOV DWORD PTR SS:[EBP-1C],EDX
000B4347 8145 EC 00F877 ADD DWORD PTR SS:[EBP-14],77F800
000B434E 56 PUSH ESI
000B434F 57 PUSH EDI
000B4350 8175 E8 003524 XOR DWORD PTR SS:[EBP-18],00243500
000B4353 8955 E4 MOV DWORD PTR SS:[EBP-1C],EDX
```

Registers (32Now!):

```
EAX FFFF64CD
ECX 00A8FEC8
EDX 7C90E4F4 ntdll.KiFastSystemCallRet
EBX 000A2037
ESP 00A8FEB0
EBP 00A8FEB4
ESI 095C0003
EDI 7C9115EF ntdll.7C9115EF
EIP 000B432A
C 1 ES 0023 32bit 0(FFFFFFFF)
P 0 CS 001B 32bit 0(FFFFFFFF)
A 1 SS 0023 32bit 0(FFFFFFFF)
Z 0 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FFD0000(FFF)
T 0 GS 0000 NULL
D 0
O 0
0 0 LastErr 00000000 ERROR_SUCCESS
EFL 00000213 (NO,B,NE,BE,NS,PO,GE,G)
MM0 5.995635e+36 5.981526e+36
MM1 8.399271e-40 5.989963e+36
MM2 5.981526e+36 5.989963e+36
MM3 6.011055e+36 9.934225e-40
MM4 5.989963e+36 6.011055e+36
MM5 8.399131e-40 1.404938e-38
MM6 6.011055e+36 5.647233e-43
MM7 1.404938e-38 5.995635e+36
XMM0 7C903400 7C90B5FD 00000193 00000017
XMM1 000B9D80 0098FD44 7C9056F4 7C908000
XMM2 00001000 0093FD0C 7C906460 7C90B5FD
XMM3 7C90CFE0 00000040 0098FD04 00000000
XMM4 7C90B5FD 00000193 00000150 0098FC44
XMM5 0098FC60 7C9056F4 7C900000 7C903400
XMM6 00000193 00092560 7C903400 00092556
XMM7 7C9056F4 7C900000 7C903400 7C90B5FD
MXCSR 00001F80 FZ 0 DZ 0 Err 0 0 0 0 0
P U O Z D I
Rnd NEAR Mask 1 1 1 1 1 1
```

Memory Dump:

```
Address Hex dump ASCII
01004000 72 65 73 74 6F 72 65 20 65 61 78 20 20 20 20 20 restore eax
01004010 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
01004020 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
01004030 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01004040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01004050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01004060 00 00 00 00 40 BB 00 00 00 00 00 00 00 00 00
01004070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01004080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01004090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
010040A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
010040B0 BF 44 FF FF 00 00 00 00 00 00 00 00 00 00 00
```

Patchwork

Example: Nymaim (Control Flow Obfuscation)

OllyDbg - svchost.exe

File View Debug Trace Plugins Options Windows Help

CPU - thread 2. (00000960)

Address	Hex dump	Assembly	Comment
000B42D2	8B07	MOV EAX, DWORD PTR DS:[EDI]	
000B42D4	81FD 203C4D39	CMP EBP, 394D3C20	
000B42DA	8B40 14	MOV ECX, DWORD PTR SS:[EBP+14]	
000B42DD	81FD 203D4D09	CMP EBP, 94D3D20	
000B42E3	81F9 6EFA0C00	CMP ECX, 0CFA6E	
000B42E9	0F84 3015FEFF	JE 0009581F	
000B42EF	81F9 4FFC0C00	CMP ECX, 0CFC4F	
000B42F5	0F84 4544FFFF	JG 000A8740	
000B42FB	8935 B4FCFFFF	MOV DWORD PTR SS:[EBP-34C], EDX	
000B4301	8985 ACFCFFFF	MOV DWORD PTR SS:[EBP-354], EAX	
000B4307	C785 B8FCFFFF	MOV DWORD PTR SS:[EBP-343], 0	
000B4311	E9 3544FFFF	JMP 000A874B	
000B4316	89E5	MOV EBP, ESP	
000B4318	50	PUSH EAX	
000B4319	8B45 04	MOV EAX, DWORD PTR SS:[EBP+4]	
000B431C	8945 10	MOV DWORD PTR SS:[EBP+10], EAX	
000B431F	8B45 08	MOV EAX, DWORD PTR SS:[EBP+8]	
000B4322	E9 2B8DFEFF	JMP 0009D052	
000B4327	0145 04	ADD DWORD PTR SS:[EBP+4], EAX	
000B432B	58	POP EAX	
000B432B	C9	LEAVE	
000B432C	C2 0800	RETN 8	
000B432F	0145 04	ADD DWORD PTR SS:[EBP+4], EAX	
000B4332	58	POP EAX	
000B4333	C9	LEAVE	
000B4334	C2 0800	RETN 8	
000B4337	6A 03	PUSH 3	
000B4339	E9 3973FFFF	JMP 000A86C7	
000B433E	55	PUSH EBP	
000B433F	89E5	MOV EBP, ESP	
000B4341	83EC 24	SUB ESP, 24	
000B4344	8955 E4	MOV DWORD PTR SS:[EBP-1C], EDX	
000B4347	8145 EC 00F877	ADD DWORD PTR SS:[EBP-14], 77F800	
000B434E	56	PUSH ESI	
000B434F	57	PUSH EDI	
000B4350	8175 E8 003524	XOR DWORD PTR SS:[EBP-18], 00243500	
000B4353	8955 E4	MOV DWORD PTR SS:[EBP-1C], EDX	

Registers (3DNow!)

EAX	00000000
ECX	00A8FEC8
EDX	7C90E4F4 ntdll.KiFastSystemCallRet
EBX	000A2037
ESP	00A8FEB4
EBP	00A8FEB4
ESI	095C0003
EDI	7C9115EF ntdll.7C9115EF
EIP	000B432B

**Stack [00A8FEB4]=00A8FEE4
EBP=00A8FEB4**

Address	Hex dump	ASCII
01004000	63 6C 65 61 6E 20 75 70 20 20 20 20 20 20 20 20	clean up
01004010	73 74 61 63 68 20 66 72 61 6D 65 20 20 20 20 20	stack frame
01004020	20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
01004030	20 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004060	00 00 00 00 40 BB 00 00 00 00 00 00 00 00 00	
01004070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040B0	BF 44 FF FF 00 00 00 00 00 00 00 00 00 00 00	

Registers (3DNow!)

EAX	00000000
ECX	00A8FEC8
EDX	7C90E4F4 ntdll.KiFastSystemCallRet
EBX	000A2037
ESP	00A8FEB4
EBP	00A8FEB4
ESI	095C0003
EDI	7C9115EF ntdll.7C9115EF
EIP	000B432B

**Stack [00A8FEB4]=00A8FEE4
EBP=00A8FEB4**

Address	Hex dump	ASCII
00A8FEB4	00A8FEE4	
00A8FEB8	000B7938	
00A8FEBC	4F4A0544	
00A8FEBD	B0B48F89	
00A8FEC0	000B72C8	
00A8FEC4	7C90D1FC	
00A8FEC8	7C9115EF	
00A8FED0	00000000	
00A8FED4	00A8FF00	
00A8FED8	000007D0	
00A8FEDC	095C0003	
00A8FEE0	00A8FFA4	
00A8FEE4	00A8FF00	
00A8FEE8	00000000	

Registers (3DNow!)

EAX	00000000
ECX	00A8FEC8
EDX	7C90E4F4 ntdll.KiFastSystemCallRet
EBX	000A2037
ESP	00A8FEB4
EBP	00A8FEB4
ESI	095C0003
EDI	7C9115EF ntdll.7C9115EF
EIP	000B432B

**Stack [00A8FEB4]=00A8FEE4
EBP=00A8FEB4**

Address	Hex dump	ASCII
00A8FEB4	00A8FEE4	
00A8FEB8	000B7938	
00A8FEBC	4F4A0544	
00A8FEBD	B0B48F89	
00A8FEC0	000B72C8	
00A8FEC4	7C90D1FC	
00A8FEC8	7C9115EF	
00A8FED0	00000000	
00A8FED4	00A8FF00	
00A8FED8	000007D0	
00A8FEDC	095C0003	
00A8FEE0	00A8FFA4	
00A8FEE4	00A8FF00	
00A8FEE8	00000000	

Registers (3DNow!)

EAX	00000000
ECX	00A8FEC8
EDX	7C90E4F4 ntdll.KiFastSystemCallRet
EBX	000A2037
ESP	00A8FEB4
EBP	00A8FEB4
ESI	095C0003
EDI	7C9115EF ntdll.7C9115EF
EIP	000B432B

**Stack [00A8FEB4]=00A8FEE4
EBP=00A8FEB4**

Address	Hex dump	ASCII
00A8FEB4	00A8FEE4	
00A8FEB8	000B7938	
00A8FEBC	4F4A0544	
00A8FEBD	B0B48F89	
00A8FEC0	000B72C8	
00A8FEC4	7C90D1FC	
00A8FEC8	7C9115EF	
00A8FED0	00000000	
00A8FED4	00A8FF00	
00A8FED8	000007D0	
00A8FEDC	095C0003	
00A8FEE0	00A8FFA4	
00A8FEE4	00A8FF00	
00A8FEE8	00000000	

Registers (3DNow!)

EAX	00000000
ECX	00A8FEC8
EDX	7C90E4F4 ntdll.KiFastSystemCallRet
EBX	000A2037
ESP	00A8FEB4
EBP	00A8FEB4
ESI	095C0003
EDI	7C9115EF ntdll.7C9115EF
EIP	000B432B

**Stack [00A8FEB4]=00A8FEE4
EBP=00A8FEB4**

Address	Hex dump	ASCII
00A8FEB4	00A8FEE4	
00A8FEB8	000B7938	
00A8FEBC	4F4A0544	
00A8FEBD	B0B48F89	
00A8FEC0	000B72C8	
00A8FEC4	7C90D1FC	
00A8FEC8	7C9115EF	
00A8FED0	00000000	
00A8FED4	00A8FF00	
00A8FED8	000007D0	
00A8FEDC	095C0003	
00A8FEE0	00A8FFA4	
00A8FEE4	00A8FF00	
00A8FEE8	00000000	

Registers (3DNow!)

EAX	00000000
ECX	00A8FEC8
EDX	7C90E4F4 ntdll.KiFastSystemCallRet
EBX	000A2037
ESP	00A8FEB4
EBP	00A8FEB4
ESI	095C0003
EDI	7C9115EF ntdll.7C9115EF
EIP	000B432B

**Stack [00A8FEB4]=00A8FEE4
EBP=00A8FEB4**

Address	Hex dump	ASCII
00A8FEB4	00A8FEE4	
00A8FEB8	000B7938	
00A8FEBC	4F4A0544	
00A8FEBD	B0B48F89	
00A8FEC0	000B72C8	
00A8FEC4	7C90D1FC	
00A8FEC8	7C9115EF	
00A8FED0	00000000	
00A8FED4	00A8FF00	
00A8FED8	000007D0	
00A8FEDC	095C0003	
00A8FEE0	00A8FFA4	
00A8FEE4	00A8FF00	
00A8FEE8	00000000	

Registers (3DNow!)

EAX	00000000
ECX	00A8FEC8
EDX	7C90E4F4 ntdll.KiFastSystemCallRet
EBX	000A2037
ESP	00A8FEB4
EBP	00A8FEB4
ESI	095C0003
EDI	7C9115EF ntdll.7C9115EF
EIP	000B432B

**Stack [00A8FEB4]=00A8FEE4
EBP=00A8FEB4**

Address	Hex dump	ASCII
00A8FEB4	00A8FEE4	
00A8FEB8	000B7938	
00A8FEBC	4F4A0544	
00A8FEBD	B0B48F89	
00A8FEC0	000B72C8	
00A8FEC4	7C90D1FC	
00A8FEC8	7C9115EF	
00A8FED0	00000000	
00A8FED4	00A8FF00	
00A8FED8	000007D0	
00A8FEDC	095C0003	
00A8FEE0	00A8FFA4	
00A8FEE4	00A8FF00	
00A8FEE8	00000000	

Registers (3DNow!)

EAX	00000000
ECX	00A8FEC8
EDX	7C90E4F4 ntdll.KiFastSystemCallRet
EBX	000A2037
ESP	00A8FEB4
EBP	00A8FEB4
ESI	095C0003
EDI	7C9115EF ntdll.7C9115EF
EIP	000B432B

**Stack [00A8FEB4]=00A8FEE4
EBP=00A8FEB4**

Address	Hex dump	ASCII
00A8FEB4	00A8FEE4	
00A8FEB8	000B7938	
00A8FEBC	4F4A0544	
00A8FEBD	B0B48F89	
00A8FEC0	000B72C8	
00A8FEC4	7C90D1FC	
00A8FEC8	7C9115EF	
00A8FED0	00000000	
00A8FED4	00A8FF00	
00A8FED8	000007D0	
00A8FEDC	095C0003	
00A8FEE0	00A8FFA4	
00A8FEE4	00A8FF00	
00A8FEE8	00000000	

Registers (3DNow!)

EAX	00000000
ECX	00A8FEC8
EDX	7C90E4F4 ntdll.KiFastSystemCallRet
EBX	000A2037
ESP	00A8FEB4
EBP	00A8FEB4
ESI	095C0003
EDI	7C9115EF ntdll.7C9115EF
EIP	000B432B

**Stack [00A8FEB4]=00A8FEE4
EBP=00A8FEB4**

Address	Hex dump	ASCII
00A8FEB4	00A8FEE4	
00A8FEB8	000B7938	
00A8FEBC	4F4A0544	
00A8FEBD	B0B48F89	
00A8FEC0	000B72C8	
00A8FEC4	7C90D1FC	
00A8FEC8	7C9115EF	
00A8FED0	00000000	
00A8FED4	00A8FF00	
00A8FED8	000007D0	
00A8FEDC	095C0003	
00A8FEE0	00A8FFA4	
00A8FEE4	00A8FF00	
00A8FEE8	00000000	

Registers (3DNow!)

EAX	00000000
ECX	00A8FEC8
EDX	7C90E4F4 ntdll.KiFastSystemCallRet
EBX	000A2037
ESP	00A8FEB4
EBP	00A8FEB4
ESI	095C0003
EDI	7C9115EF ntdll.7C9115EF
EIP	000B432B

**Stack [00A8FEB4]=00A8FEE4
EBP=00A8FEB4**

Address	Hex dump	ASCII
00A8FEB4	00A8FEE4	
00A8FEB8	000B7938	
00A8FEBC	4F4A0544	
00A8FEBD	B0B48F89	
00A8FEC0	000B72C8	
00A8FEC4	7C90D1FC	
00A8FEC8	7C9115EF	
00A8FED0	00000000	
00A8FED4	00A8FF00	
00A8FED8	000007D0	
00A8FEDC	095C0003	
00A8FEE0	00A8FFA4	
00A8FEE4	00A8FF00	
00A8FEE8	00000000	

Registers (3DNow!)

EAX	00000000
ECX	00A8FEC8
EDX	7C90E4F4 ntdll.KiFastSystemCallRet
EBX	000A2037
ESP	00A8FEB4
EBP	00A8FEB4
ESI	095C0003
EDI	7C9115EF ntdll.7C9115EF
EIP	000B432B

**Stack [00A8FEB4]=00A8FEE4
EBP=00A8FEB4**

Address	Hex dump	ASCII
00A8FEB4	00A8FEE4	
00A8FEB8	000B7938	
00A8FEBC	4F4A0544	
00A8FEBD	B0B48F89	
00A8FEC0	000B72C8	
00A8FEC4	7C90D1FC	
00A8FEC8	7C9115EF	
00A8FED0	00000000	
00A8FED4	00A8FF00	
00A8FED8	000007D0	
00A8FEDC	095C0003	
00A8FEE0	00A8FFA4	

Patchwork

Example: Nymaim (Control Flow Obfuscation)

OllyDbg - svchost.exe

File View Debug Trace Plugins Options Windows Help

CPU - thread 2. (00000960)

Address	Hex dump	Assembly	Comment
000B42D2	8B07	MOV EAX, DWORD PTR DS:[EDI]	
000B42D4	81FD 203C4D39	CMP EBP, 394D3C20	
000B42DA	8B40 14	MOV ECX, DWORD PTR SS:[EBP+14]	
000B42DD	81FD 203D4D09	CMP EBP, 94D3D20	
000B42E3	81F9 6EFA0C00	CMP ECX, 0CFA6E	
000B42E9	0F84 3015FEFF	JE 0009581F	
000B42EF	81F9 4FFC0C00	CMP ECX, 0CFC4F	
000B42F5	0F84 4544FFFF	JG 000A8740	
000B42FB	8995 B4FCFFFF	MOV DWORD PTR SS:[EBP-34C], EDX	
000B4301	8985 ACFCFFFF	MOV DWORD PTR SS:[EBP-354], EAX	
000B4307	C785 B8FCFFFF	MOV DWORD PTR SS:[EBP-348], 0	
000B4311	E9 3544FFFF	JMP 000A874B	
000B4316	89E5	MOV EBP, ESP	
000B4318	50	PUSH EAX	
000B4319	8B45 04	MOV EAX, DWORD PTR SS:[EBP+4]	
000B431C	8945 10	MOV DWORD PTR SS:[EBP+10], EAX	
000B431F	8B45 08	MOV EAX, DWORD PTR SS:[EBP+8]	
000B4322	E9 2B8DFE FF	JMP 0009D052	
000B4327	0145 04	ADD DWORD PTR SS:[EBP+4], EAX	
000B432A	58	POP EAX	
000B432B	C9	LEAVE	
000B432C	C2 0800	RETN 8	
000B432F	0145 04	ADD DWORD PTR SS:[EBP+4], EAX	
000B4332	58	POP EAX	
000B4333	C9	LEAVE	
000B4334	C2 0800	RETN 8	
000B4337	6A 03	PUSH 3	
000B4339	E9 8973FFFF	JMP 000AB6C7	
000B433E	55	PUSH EBP	
000B433F	89E5	MOV EBP, ESP	
000B4341	83EC 24	SUB ESP, 24	
000B4344	8955 E4	MOV DWORD PTR SS:[EBP-1C], EDX	
000B4347	8145 EC 00F877	ADD DWORD PTR SS:[EBP-14], 77F800	
000B434E	56	PUSH ESI	
000B434F	57	PUSH EDI	
000B4350	8175 E8 003524	XOR DWORD PTR SS:[EBP-18], 00243500	
000B4357	8955 E4	MOV DWORD PTR SS:[EBP-1C], EDX	

Imm=0008
Top of stack [00A8FE8]=000B0798

Address	Hex dump	ASCII
01004000	72 55 74 75 72 6E 20 74 6F 20 20 20 20 20 20 20 20	return to
01004010	64 55 74 6F 75 72 65 64 20 61 64 64 72 65 73 73	dettoured address
01004020	20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
01004030	20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004060	00 00 00 00 40 BB 00 00 00 00 00 00 00 00 00 00	
01004070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01004090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010040B0	BF 44 FF FF 00 00 00 00 00 00 00 00 00 00 00 00	

Registers (32Now!)

EAX 00000000
ECX 00A8FEC8
EDX 7C90E4F4 ntdll.KiFastSystemCallRet
EBX 000A2037
ESP 00A8FE88
EBP 00A8FEE4
ESI 095C0003
EDI 7C9115EF ntdll.7C9115EF
EIP 000B432C

C 1 ES 0023 32bit 0(FFFFFFFF)
P 0 CS 001B 32bit 0(FFFFFFFF)
A 1 SS 0023 32bit 0(FFFFFFFF)
Z 0 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FFDD000(FFF)
T 0 GS 0000 NULL
D 0
O 0
0 0 LastErr 00000000 ERROR_SUCCESS
EFL 00000213 (NO, B, NE, BE, NS, PO, GE, G)

MM0 5.995635e+36 5.981526e+36
MM1 8.399271e-40 5.989963e+36
MM2 5.981526e+36 5.989963e+36
MM3 6.011055e+36 9.934225e-40
MM4 5.989963e+36 6.011055e+36
MM5 8.399131e-40 1.404938e-38
MM6 6.011055e+36 5.647233e-43
MM7 1.404938e-38 5.995635e+36

XMM0 7C903400 7C90B5FD 00000193 00000017
XMM1 000B3000 0038FD44 7C9056F4 7C900000
XMM2 00001000 0038FDBC 7C906469 7C90B5FD
XMM3 7C90CF50 00000040 0098FDC4 00000000
XMM4 7C90B5FD 00000193 00000150 0098F444
XMM5 0098FC60 7C9056F4 7C900000 7C903400
XMM6 00000193 00092560 7C903400 00092556
XMM7 7C9056F4 7C900000 7C903400 7C90B5FD

PU 0 2 0 I
MXCSR 00001F80 FZ 0 DZ 0 Err 0 0 0 0 0
Rnd NEAR Mask 1 1 1 1 1 1

00A8FE8 000B0798 0-3
00A8FEB8 4F4AD544 0-FJ0
00A8FEC0 B0B48F89 0-AH3
00A8FEC4 000B02CB 0-000
00A8FEC8 7C90D1FC 0-000
00A8FED0 7C9115EF 0-000
00A8FED4 00A8FF00 0-000
00A8FED8 000007D0 0-000
00A8FEDC 095C0003 0-000
00A8FEE0 00A8FFA4 0-000
00A8FEE4 00A8FF00 0-000
00A8FEE8 000AD0E2 0-000
00A8FEFC 7C900150 0-000

RETURN from 0009D009 to 000BA2CB
RETURN to ntdll.ZwDelayExecution+0C
RETURN from 000908D7 to 000AD0E2
ntdll.ZwDelayExecution

Patchwork

Example: Nymaim (Deobfuscation)

■ Select:

50		push	eax
68	89 8F B4 B0	push	0B0B48F89h
68	44 D5 4A 4F	push	4F4AD544h
E8	3E 2D FE FF	call	sub_9D009
89	D8	mov	eax, ebx

```
push_push_call_regex = (  
  r"\x68(?P<operand_1>[\S\s]{4})"  
  r"\x68(?P<operand_2>[\S\s]{4})"  
  r"\xE8"  
)
```

■ Emulate:

Until first ret / retn instruction

Patchwork

Example: Nymaim (Deobfuscation)

■ Validate:

```
ppc_validators = {  
    "call_detour": [  
        'push dword',  
        'push dword',  
        'push ebp',  
        'mov ebp,esp',  
        'push eax',  
        'mov eax,[ebp+0x4]',  
        'mov [ebp+0x10],eax',  
        'mov eax,[ebp+0xc]',  
        ", # contains the operand -> add, sub, xor",  
        'add [ebp+0x4],eax',  
        'pop eax',  
        'leave'],  
}
```

Patchwork

Example: Nymaim (Deobfuscation)

■ Transform:

```
def _deobfuscate_call_detour(self, validation):  
    obf_start_addr = validation.selection.selectionOffset  
    call_offset = validation.emulation.cbResult - (obf_start_addr + 10 + 5)  
    deobf_call = "\x90" * 10 + "\xE8" + struct.pack("I", (call_offset) & 0xffffffff)  
    ida_lib.patch_bytes(obf_start_addr, deobf_call)  
    self.updateCallXref(obf_start_addr + 10, validation.emulation.cbResult)
```

000BA2BB	push	eax
000BA2BC	push	0B0B48F89h
000BA2C1	push	4F4AD544h
000BA2C6	call	sub_9D009
000BA2CB	mov	eax, ebx



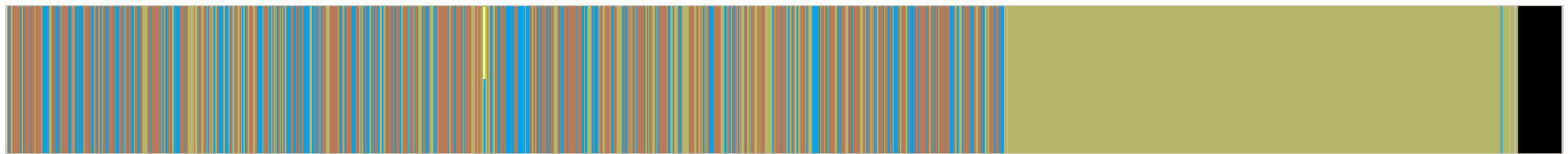
000BA2C3	nop	
000BA2C4	nop	
000BA2C5	nop	
000BA2C6	call	sub_B0798
000BA2CB	mov	eax, ebx

Patchwork

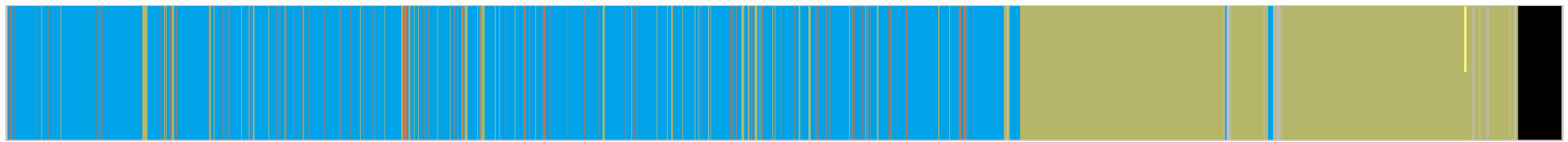
Example: Nymaim (Deobfuscation)

- Applying all deobfuscations:
 - ~2 min run time
 - 4443 transformations
 - Functions recognition: 463 -> 920

Before:



After:



Patchwork

Future plans

- Looking at more use cases
 - Memory usage analysis (deobfuscate Nymaim's blown up stack)
 - KINS BaseConfig (VM-based) decryption
 - Import reconstruction
- Extend / patch PyEmu
 - Change disassembly engine to IDA / capstone
 - Increase coverage of opcodes

Patchwork

Conclusion

- Give it a try :)
 - Repository at <http://patchwork.pnx.tf>
 - (points to: https://bitbucket.org/daniel_plohmann/idapatchwork)
- Send **feedback** or **ideas** for improvement!
 - patchwork@pnx.tf / plohmann@cs.uni-bonn.de