Automatic Generation of Compact Alphanumeric Shellcodes for x86

Aditya Basu, Anish Mathuria, Nagendra Chowdary

DA-IICT

Alphanumeric Shellcodes. Why ?

- ✦ Defensive filters strip all the non-alphanumeric characters from input.
- ✦ This ruins many injection attacks.

Attacker's Goal

- ✦ Generate code that only consists of:
  - ▪ A-Z
  - ▪ a-z
  - ▪ 0-9

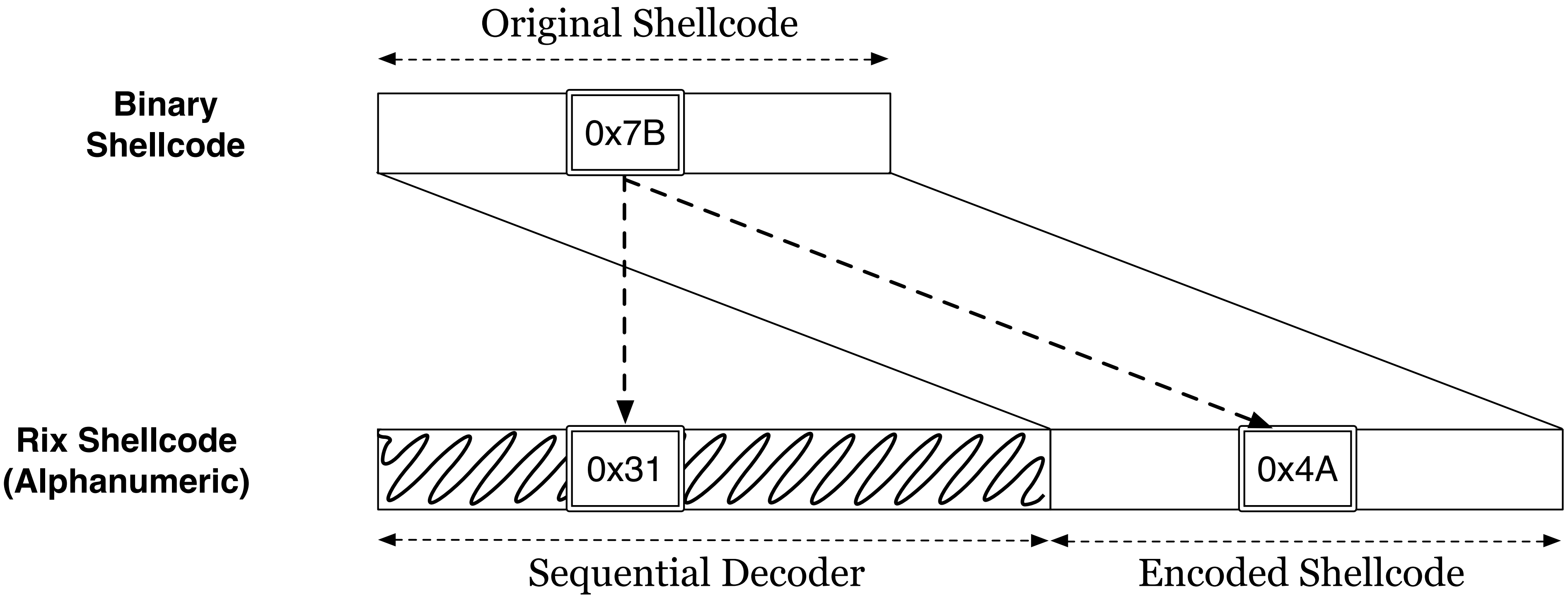
Compact. Why ?

- ✦ Transforming the shellcode to alphanumeric range *significantly* increases the shellcode size.
- ✦ Size restrictions on input.

Example

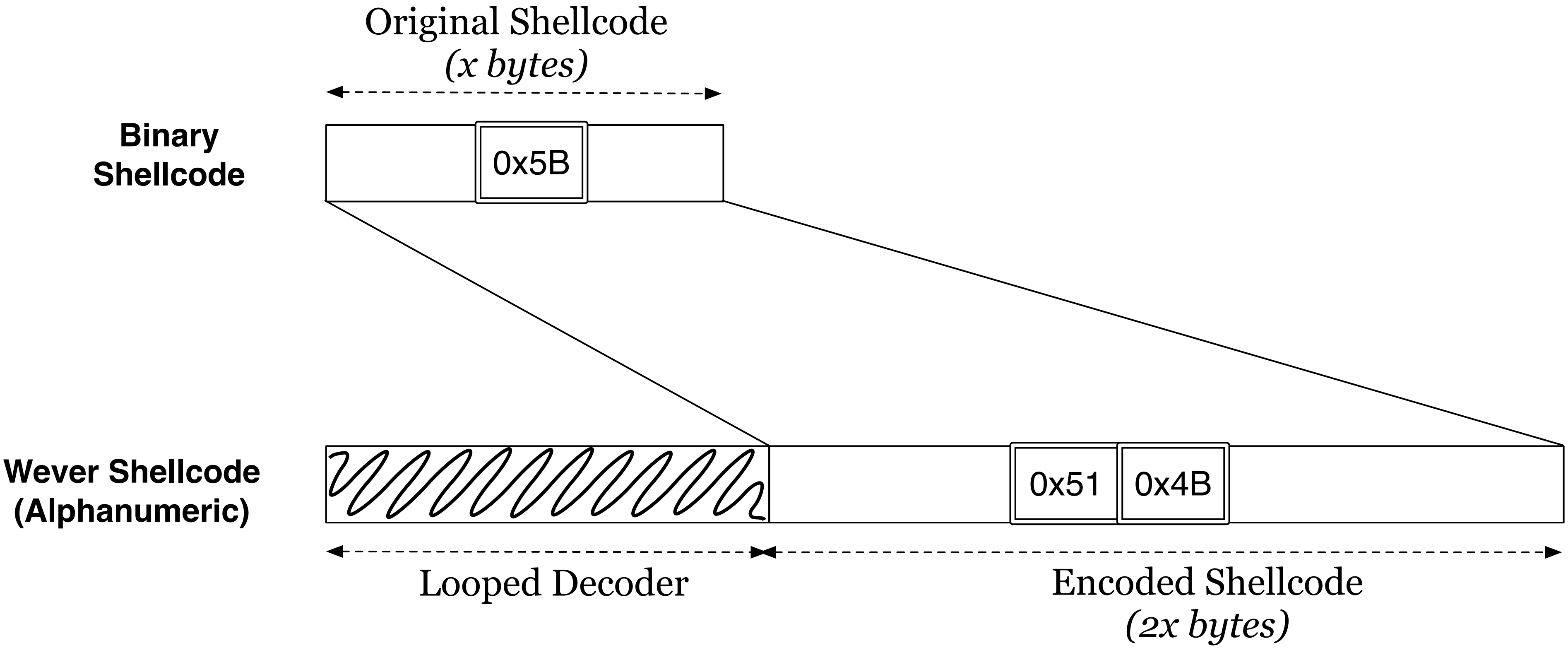
- ✦ Buffer overflow exploits are limited by the buffer size.

Existing Schemes: Rix



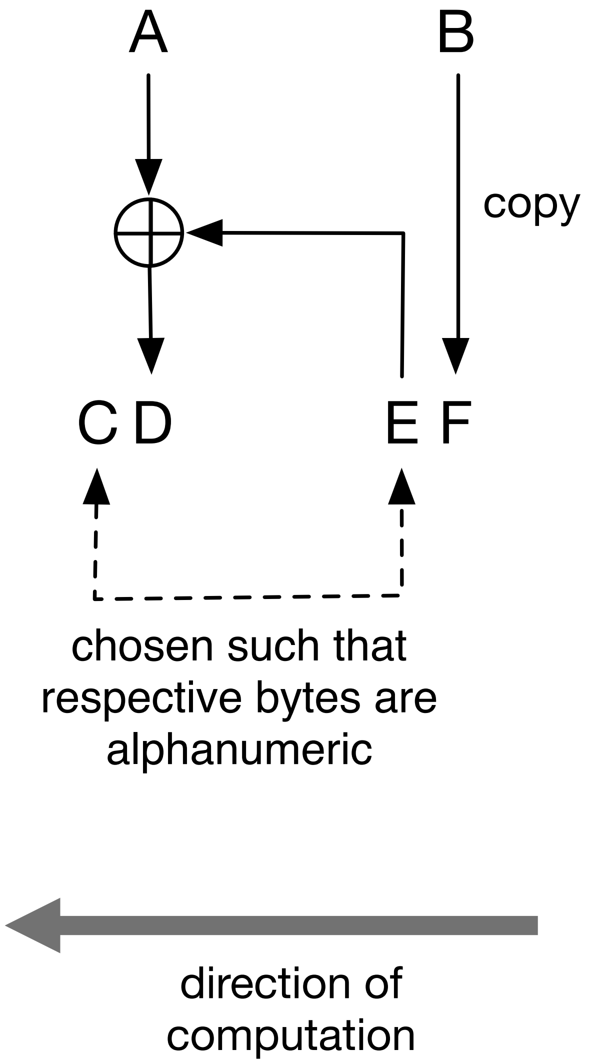
$$0x31 \oplus 0x4A = 0x7B$$

Existing Schemes: Jan Wever



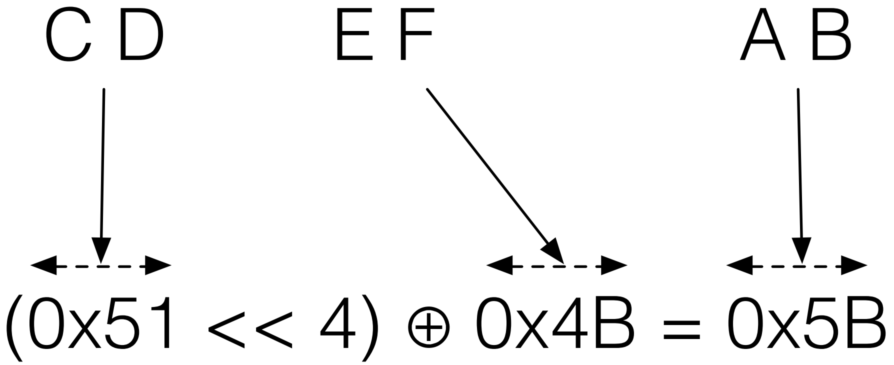
$$(0x51 << 4) \oplus 0x4B = 0x5B$$

Wever’s Encoding Scheme



chosen such that respective bytes are alphanumeric

direction of computation



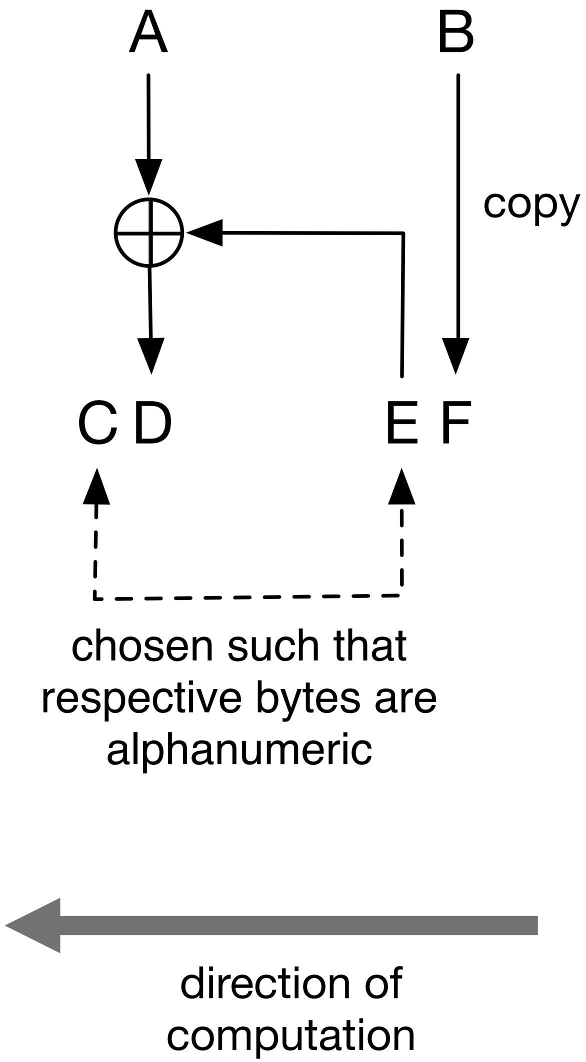
$(0x51 << 4) \oplus 0x4B = 0x5B$

Alphanumeric ASCII Range

C, E = {3, 4, 5, 6, 7}

0 – 9	0x30 – 0x39
A – Z	0x41 – 0x5A
a – z	0x61 – 0x7A

Observation



Sufficient Subset

C, E = {4, 5}

K – Z

0x4B – 0x5A

AF: Alpha Freedom

Idea

Scan for bytes in the range **K-Z** and

Decode!

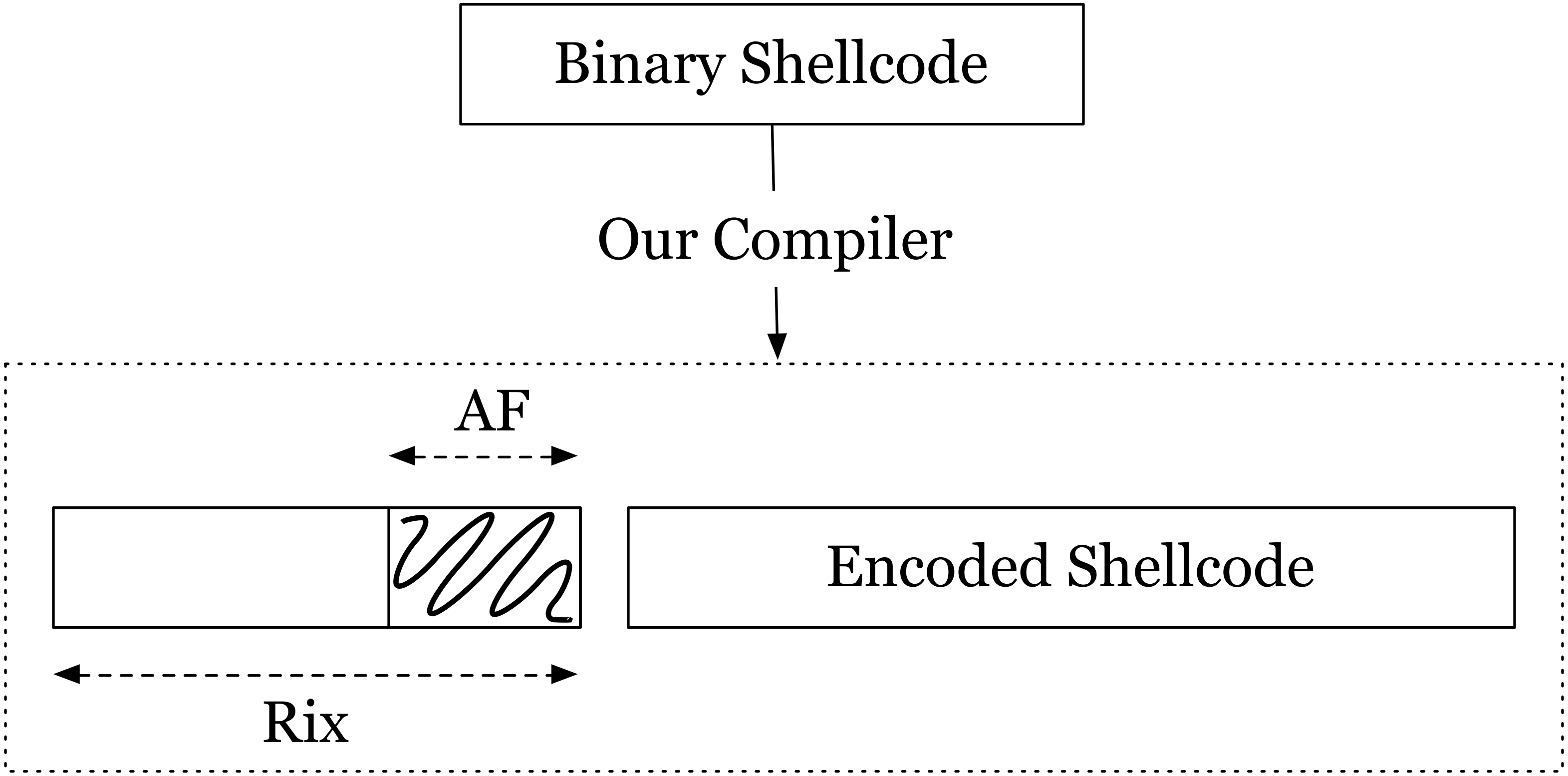
Drawback

All alpha values of the original shellcode in K-Z, also need to be encoded, although they are alphanumeric.

AF: Decoding Algorithm

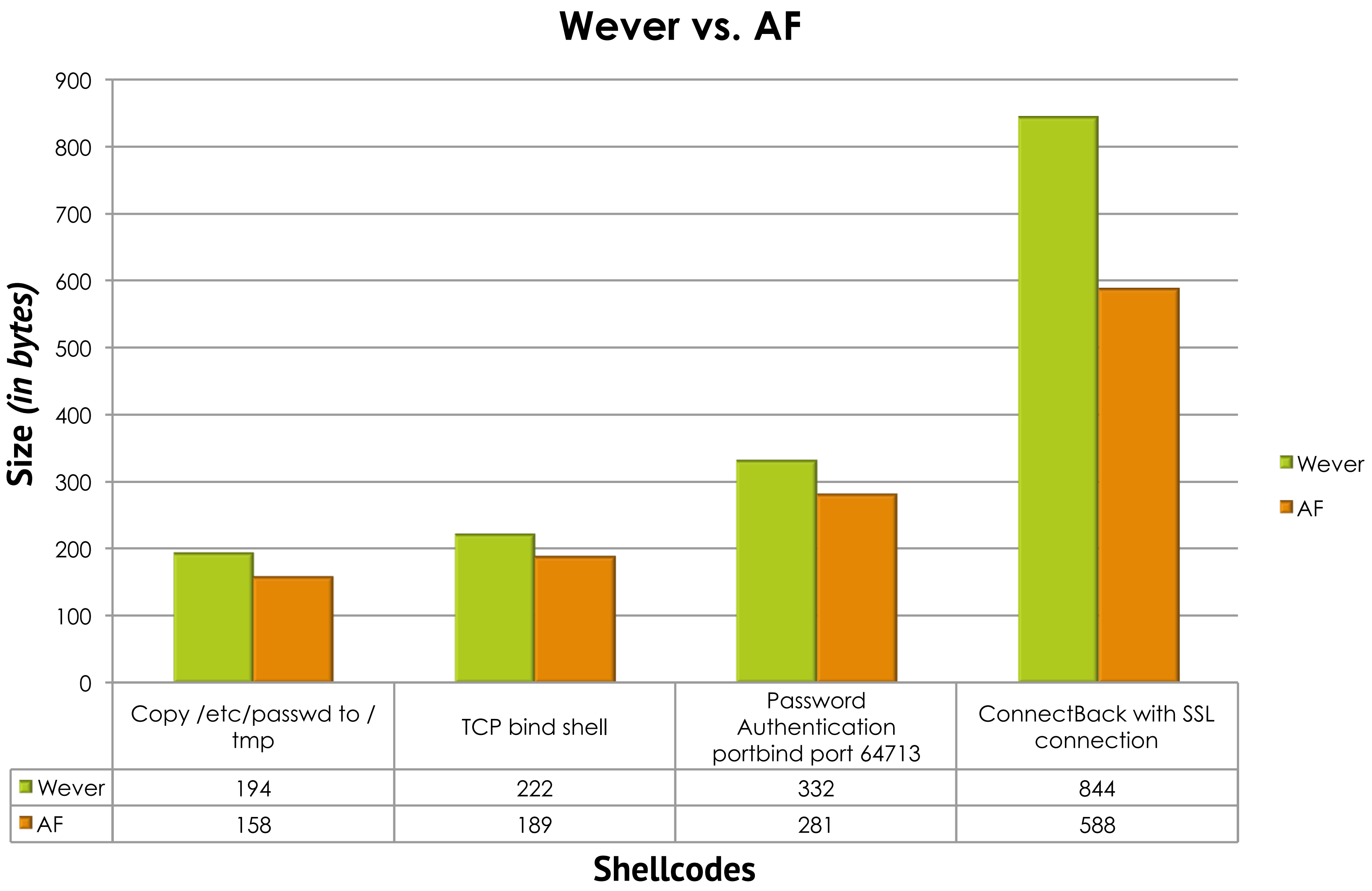
```
For every byte,
    if ‘z’,
        then end decoding loop
    else if in range ‘K’ to ‘Z’,
        then decode the current byte & the next byte
        and replace both bytes with the decoded byte
    else,
        skip byte
```

Compiler



Download from: www.adityabasu.me

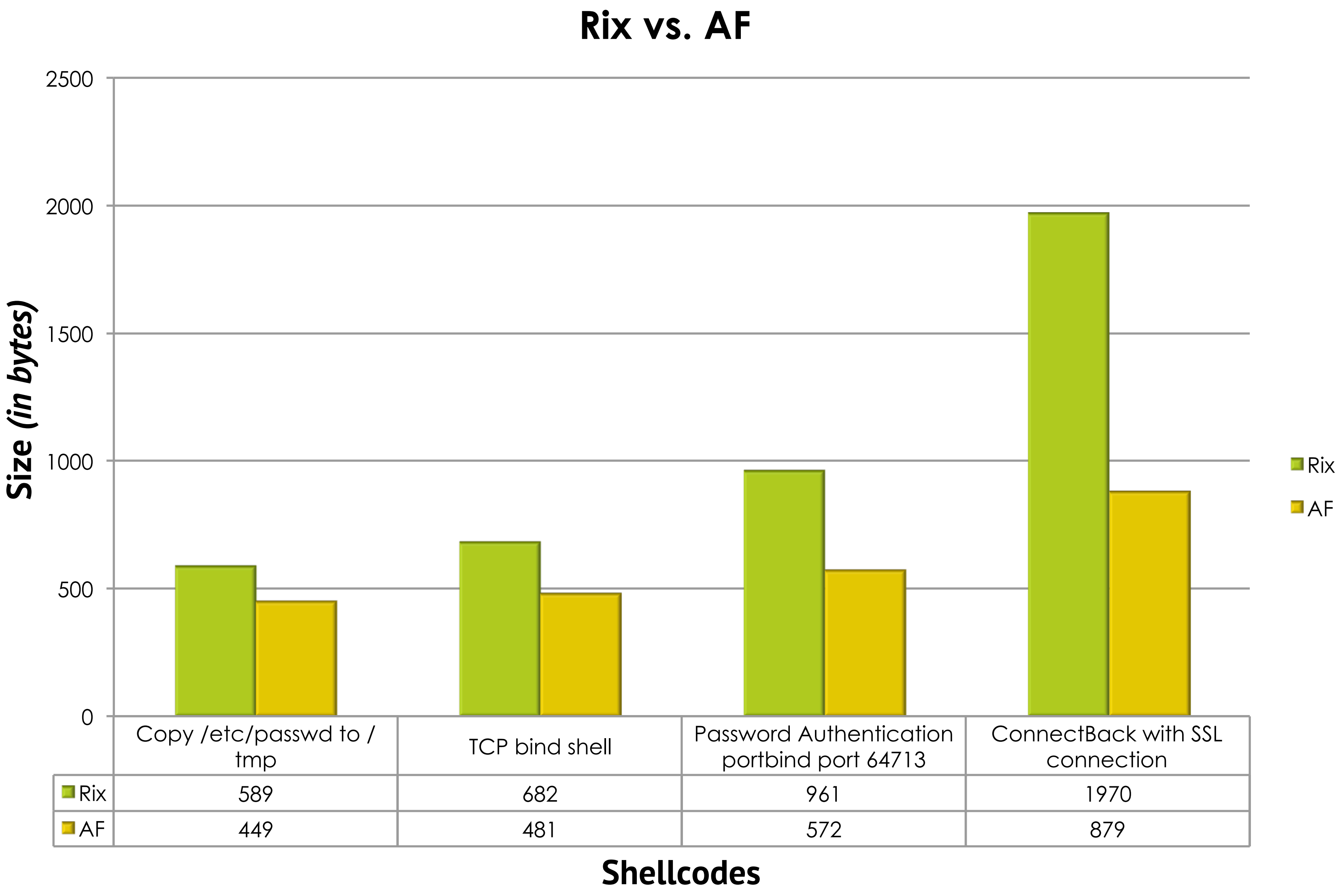
Compaction Performance



Wever vs. AF

	Copy /etc/passwd to / tmp	TCP bind shell	Password Authentication portbind port 64713	ConnectBack with SSL connection
🟩 Wever	194	222	332	844
🟧 AF	158	189	281	588

Total Size of Output



	Copy /etc/passwd to / tmp	TCP bind shell	Password Authentication portbind port 64713	ConnectBack with SSL connection
🟩 Rix	589	682	961	1970
🟨 AF	449	481	572	879

NAT: Non-Alpha Touch

Idea

Scan for byte ‘y’ and

Decode!

Encode(‘\x5b’) = y3k (0x79 0x33 0x6b)

Drawback

All ‘y’ and ‘z’ values in the original shellcode, also need to be encoded.

NAT: Why ?

- There may be some shellcodes which mostly use alphanumeric characters in ***K – Z*** range.
- In this case, we avoid of lot of spurious encoding of bytes.

Future Work

- Optimize the size of the our decoder loop.
- Port the implementation to *x86_64* to compare against *alpha3*.

References

- Rix, “Writing IA32 Alphanumeric shellcodes,” *Phrack*, vol. 57, 2001.
- B. J. Wever, “Alphanumeric shellcode Decoder Loop,” *Skypher*, 2004.
- “Shellforge,” 04 2014. [Online]. Available: http://www.secdev.org/ projects/shellforge
- “Shell-Storm,” 04 2014. [Online]. Available: http://shell-storm.org/

Questions ?

ASCII Table

Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex	
0	00	NUL	16	10	DLE	32	20		48	30	0	64	40	@	**80**	**50**	**P**	96	60	`	112	70	p
1	01	SOH	17	11	DC1	33	21	!	49	31	1	65	41	A	**81**	**51**	**Q**	97	61	a	113	71	q
2	02	STX	18	12	DC2	34	22	"	50	32	2	66	42	B	**82**	**52**	**R**	98	62	b	114	72	r
3	03	ETX	19	13	DC3	35	23	#	51	33	3	67	43	C	**83**	**53**	**S**	99	63	c	115	73	s
4	04	EOT	20	14	DC4	36	24	$	52	34	4	68	44	D	**84**	**54**	**T**	100	64	d	116	74	t
5	05	ENQ	21	15	NAK	37	25	%	53	35	5	69	45	E	**85**	**55**	**U**	101	65	e	117	75	u
6	06	ACK	22	16	SYN	38	26	&	54	36	6	70	46	F	**86**	**56**	**V**	102	66	f	118	76	v
7	07	BEL	23	17	ETB	39	27	'	55	37	7	71	47	G	**87**	**57**	**W**	103	67	g	119	77	w
8	08	BS	24	18	CAN	40	28	(	56	38	8	72	48	H	**88**	**58**	**X**	104	68	h	120	78	x
9	09	HT	25	19	EM	41	29	)	57	39	9	73	49	I	**89**	**59**	**Y**	105	69	i	121	79	y
10	0A	LF	26	1A	SUB	42	2A	*	58	3A	:	74	4A	J	**90**	**5A**	**Z**	106	6A	j	122	7A	z
11	0B	VT	27	1B	ESC	43	2B	+	59	3B	;	**75**	**4B**	**K**	91	5B	[	107	6B	k	123	7B	{
12	0C	FF	28	1C	FS	44	2C	,	60	3C	<	**76**	**4C**	**L**	92	5C	\	108	6C	l	124	7C	|
13	0D	CR	29	1D	GS	45	2D	-	61	3D	=	**77**	**4D**	**M**	93	5D	]	109	6D	m	125	7D	}
14	0E	SO	30	1E	RS	46	2E	.	62	3E	>	**78**	**4E**	**N**	94	5E	^	110	6E	n	126	7E	~
15	0F	SI	31	1F	US	47	2F	/	63	3F	?	**79**	**4F**	**O**	95	5F	_	111	6F	o	127	7F	DEL