

Stack Smasher - Useful Gadget Encodings

A. Encodings of `movq` instructions

`movq S, D`

Source <i>S</i>	Destination <i>D</i>							
	<code>%rax</code>	<code>%rcx</code>	<code>%rdx</code>	<code>%rbx</code>	<code>%rsp</code>	<code>%rbp</code>	<code>%rsi</code>	<code>%rdi</code>
<code>%rax</code>	48 89 c0	48 89 c1	48 89 c2	48 89 c3	48 89 c4	48 89 c5	48 89 c6	48 89 c7
<code>%rcx</code>	48 89 c8	48 89 c9	48 89 ca	48 89 cb	48 89 cc	48 89 cd	48 89 ce	48 89 cf
<code>%rdx</code>	48 89 d0	48 89 d1	48 89 d2	48 89 d3	48 89 d4	48 89 d5	48 89 d6	48 89 d7
<code>%rbx</code>	48 89 d8	48 89 d9	48 89 da	48 89 db	48 89 dc	48 89 dd	48 89 de	48 89 df
<code>%rsp</code>	48 89 e0	48 89 e1	48 89 e2	48 89 e3	48 89 e4	48 89 e5	48 89 e6	48 89 e7
<code>%rbp</code>	48 89 e8	48 89 e9	48 89 ea	48 89 eb	48 89 ec	48 89 ed	48 89 ee	48 89 ef
<code>%rsi</code>	48 89 f0	48 89 f1	48 89 f2	48 89 f3	48 89 f4	48 89 f5	48 89 f6	48 89 f7
<code>%rdi</code>	48 89 f8	48 89 f9	48 89 fa	48 89 fb	48 89 fc	48 89 fd	48 89 fe	48 89 ff

B. Encodings of `popq` instructions

Operation	Register <i>R</i>							
	<code>%rax</code>	<code>%rcx</code>	<code>%rdx</code>	<code>%rbx</code>	<code>%rsp</code>	<code>%rbp</code>	<code>%rsi</code>	<code>%rdi</code>
<code>popq R</code>	58	59	5a	5b	5c	5d	5e	5f

C. Encodings of `movl` instructions

`movl S, D`

Source <i>S</i>	Destination <i>D</i>							
	<code>%eax</code>	<code>%ecx</code>	<code>%edx</code>	<code>%ebx</code>	<code>%esp</code>	<code>%ebp</code>	<code>%esi</code>	<code>%edi</code>
<code>%eax</code>	89 c0	89 c1	89 c2	89 c3	89 c4	89 c5	89 c6	89 c7
<code>%ecx</code>	89 c8	89 c9	89 ca	89 cb	89 cc	89 cd	89 ce	89 cf
<code>%edx</code>	89 d0	89 d1	89 d2	89 d3	89 d4	89 d5	89 d6	89 d7
<code>%ebx</code>	89 d8	89 d9	89 da	89 db	89 dc	89 dd	89 de	89 df
<code>%esp</code>	89 e0	89 e1	89 e2	89 e3	89 e4	89 e5	89 e6	89 e7
<code>%ebp</code>	89 e8	89 e9	89 ea	89 eb	89 ec	89 ed	89 ee	89 ef
<code>%esi</code>	89 f0	89 f1	89 f2	89 f3	89 f4	89 f5	89 f6	89 f7
<code>%edi</code>	89 f8	89 f9	89 fa	89 fb	89 fc	89 fd	89 fe	89 ff

D. Encodings of 2-byte functional `nop` instructions

Operation		Register <i>R</i>			
		<code>%al</code>	<code>%cl</code>	<code>%dl</code>	<code>%bl</code>
<code>andb</code>	<i>R, R</i>	20 c0	20 c9	20 d2	20 db
<code>orb</code>	<i>R, R</i>	08 c0	08 c9	08 d2	08 db
<code>cmpb</code>	<i>R, R</i>	38 c0	38 c9	38 d2	38 db
<code>testb</code>	<i>R, R</i>	84 c0	84 c9	84 d2	84 db