1 - Imagine um servidor qualquer dotado de um *firewall netfilter* onde as regras para INPUT, OUTPUT e FORWARD apareçam listadas da seguinte maneira:

Chain INPUT (policy DROP 0 packets, 0 bytes)

pkts	bytes	target	prot	opt	in	out	source	destination	
3	252	ACCEPT	icmp	--	*	*	0.0.0.0/0	0.0.0.0/0	limit: avg 1/sec burst 2
0	0	LOG	icmp	--	*	*	0.0.0.0/0	0.0.0.0/0	limit: avg 1/sec burst 2 LOG flags 0 level 4 prefix "PING-NO-DROP:"
35	1400	DROP	all	--	*	*	0.0.0.0/0	0.0.0.0/0	state INVALID
0	0	DROP	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	state NEW tcpflags: 0x3F/0x3F
0	0	DROP	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	state NEW tcpflags: 0x3F/0x00
46156	32M	ACCEPT	all	--	lo	*	0.0.0.0/0	0.0.0.0/0	
71433	34M	ACCEPT	all	--	*	*	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
991	59460	ACCEPT	tcp	--	eth0	*	0.0.0.0/0	0.0.0.0/0	state NEW multiport dports 80,443,465,25,587,995,22,21,123,8080,3128,3000
0	0	ACCEPT	tcp	--	eth1	*	0.0.0.0/0	0.0.0.0/0	state NEW multiport sports 80,443,465,25,587,995,22,21,123
0	0	ACCEPT	tcp	--	eth0	*	10.100.0.20	0.0.0.0/0	state NEW tcp spt:389
111	36456	ACCEPT	udp	--	eth0	*	0.0.0.0/0	0.0.0.0/0	state NEW udp dpts:50:68
198	33400	REJECTLOG	all	--	*	*	0.0.0.0/0	0.0.0.0/0	

Chain FORWARD (policy DROP 0 packets, 0 bytes)

pkts	bytes	target	prot	opt	in	out	source	destination	
0	0	DROPLOG	icmp	-f	*	*	0.0.0.0/0	0.0.0.0/0	
0	0	DROP	all	--	*	*	0.0.0.0/0	0.0.0.0/0	state INVALID
0	0	REJECTLOG	all	--	*	*	0.0.0.0/0	0.0.0.0/0	

Chain OUTPUT (policy DROP 0 packets, 0 bytes)

pkts	bytes	target	prot	opt	in	out	source	destination	
3	252	ACCEPT	icmp	--	*	*	0.0.0.0/0	0.0.0.0/0	
0	0	DROPLOG	icmp	-f	*	*	0.0.0.0/0	0.0.0.0/0	
0	0	ACCEPT	icmp	--	*	*	0.0.0.0/0	0.0.0.0/0	state ESTABLISHED
0	0	ACCEPT	icmp	--	*	*	0.0.0.0/0	0.0.0.0/0	state RELATED
0	0	ACCEPT	icmp	--	*	*	0.0.0.0/0	0.0.0.0/0	icmptype 8
6	300	DROP	all	--	*	*	0.0.0.0/0	0.0.0.0/0	state INVALID
46176	32M	ACCEPT	all	--	*	lo	0.0.0.0/0	0.0.0.0/0	
61431	39M	ACCEPT	all	--	*	*	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
0	0	ACCEPT	udp	--	*	eth0	0.0.0.0/0	0.0.0.0/0	state NEW udp spt:53
0	0	ACCEPT	tcp	--	*	eth0	0.0.0.0/0	0.0.0.0/0	state NEW multiport sports 80,465,25,22,587,995,21,123,8080,3128,389,3000
1	60	ACCEPT	tcp	--	*	eth0	0.0.0.0/0	10.100.0.20	state NEW tcp dpt:389
776	46560	ACCEPT	tcp	--	*	eth1	0.0.0.0/0	0.0.0.0/0	state NEW multiport dports 80,443,53
227	16357	ACCEPT	udp	--	*	eth1	0.0.0.0/0	0.0.0.0/0	state NEW multiport dports 53,123
0	0	ACCEPT	udp	--	*	eth0	0.0.0.0/0	0.0.0.0/0	state NEW udp spts:50:68
9	540	REJECTLOG	all	--	*	*	0.0.0.0/0	0.0.0.0/0	

Assuma também que as cadeias personalizadas “REJECTLOG” e “DROPLOG” vão descartar todos os pacotes que forem direcionados a elas. Tendo isso em conta, responda os items de a) até d)

a) Com base nas regras anteriores, é possível presumir se existe algum serviço de DNS nesse sistema? Se sim, em qual interface o serviço DNS estaria disponível?

b) Caso este servidor venha a realizar o encaminhamento de pacotes, o que precisa ser feito para que os todos pacotes TCP e UDP passem de uma rede para outra? Mostre o que precisa ser alterado no iptables para permitir esse funcionamento adequado.

c) Qual a quantidade máxima de *icmp “echo request”* a qual o servidor pode responder efetivamente em um período de ?

d) Suponha que este servidor deverá prover um serviço novo de proxy/cache. Este novo serviço deverá utilizar a porta 3128. Sendo assim, informe como criar uma regra que intercepta todo tráfego HTTP e HTTPS e o redireciona para o proxy/cache de forma transparente (squid porta 3128).