NAVIGATING IPTABLES & HOST FIREWALLS

MANAGEMENT & RULESET OPTIONS

❖ **iptables -F**	Flush all chains
❖ **iptables -X**	Delete all chains
❖ **iptables -Z**	Reset packet & byte counter in chains
❖ **iptables -L**	Show ruleset

❖ **iptables -A**	Append rule
❖ **iptables -D**	Delete rule
❖ **iptables -I**	Insert rule

KEY

{rX} = rule number

{iX} = interface name

{pX} = port number

{X.X.X.X} = IP address

ALLOW & DENY ANY-ANY

rules	***comment/description***
sudo iptables -A INPUT -j ACCEPT	allow any-any in
sudo iptables -A OUTPUT -j ACCEPT	allow any-any out
sudo iptables -A INPUT -j ACCEPT	deny any-any in
sudo iptables -A OUTPUT -j ACCEPT	deny any-any out

CHANGE THE CHAIN RULES

rules	***comment/description***
sudo iptables -P INPUT ACCEPT	change input chain to accept
sudo iptables -P INPUT DROP	change input chain to drop
sudo iptables -P FORWARD ACCEPT	change forward chain to accept
sudo iptables -P FORWARD DROP	change forward chain to drop
sudo iptables -P OUTPUT ACCEPT	change output chain to accept
sudo iptables -P OUTPUT DROP	change output chain to drop

LOOPBACK AND SAVE SESSIONS/EXISTING CONNECTIONS

rules	***comment/description***
sudo iptables -A INPUT -i lo -j ACCEPT	loopback in
sudo iptables -A OUTPUT -o lo -j ACCEPT	loopback out
sudo iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT	rule to keep established sessions coming in
sudo iptables -A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT	rule to keep established sessions going out

COMMON PORTS TO CONFIGURE/KNOW

rules	***comment/description***
sudo iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT	allow icmp echo in
sudo iptables -A OUTPUT -p icmp --icmp-type 0 -j ACCEPT	allow icmp echo reply out
sudo iptables -A INPUT -p tcp --dport 21 -j ACCEPT	allow ftp in
sudo iptables -A OUTPUT -p tcp --dport 21 -j ACCEPT	allow ftp out
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT	allow ssh in
sudo iptables -A OUTPUT -p tcp --dport 22 -j ACCEPT	allow ssh out
sudo iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT	allow http out
sudo iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT	allow https out
sudo iptables -A OUTPUT -p udp --dport 53 -j ACCEPT	allow DNS out over udp
sudo iptables -A OUTPUT -p tcp --dport 53 -j ACCEPT	allow DNS out over tcp
sudo iptables -A OUTPUT -p udp --sport 123 --dport 123 -j ACCEPT	allow NTP out

RULES FOR SPECIFIC IPs

rules	***comment/description***
sudo iptables -A INPUT -s {X.X.X.X} -j ACCEPT	allow traffic in from specific IP
sudo iptables -A OUTPUT -s {X.X.X.X} -j ACCEPT	allow traffic out to specific IP
sudo iptables -A INPUT -i {iX} -s {X.X.X.X} -j ACCEPT	allow traffic in from specific IP to named interface
sudo iptables -A OUTPUT -i {iX} -s {X.X.X.X} -j ACCEPT	allow traffic out to specific IP from named interface
sudo iptables -A INPUT -p tcp -s {X.X.X.X} --dport {pX} -m conntrack --ctstate NEW -j ACCEPT	allow traffic from specific IP over named port to be established

sudo iptables -A OUTPUT -p tcp -s {X.X.X.X} --sport {pX} -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT	paired rule for the one above, allows responses from our end

PORT REDIRECTION ON THE SAME LOCAL MACHINE

rules	***comment/description***
sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to 8080	Redirect any traffic coming in on port 80 to port 8080
sudo iptables -t nat --A PREROUTING -p tcp --dport 443 -j REDIRECT --to 8443	Redirect any traffic coming in on port 443 to port 8443
sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to 8022	Redirect any traffic coming in on port 22 to port 8022

INSERTING, REMOVING, LISTING RULES

rules	***comment/description***
sudo iptables -I INPUT {rX} --dport 80 -j ACCEPT	insert rule at specified list number to allow all incoming HTTP traffic
sudo iptables -I OUTPUT {rX} -p udp --dport 53 -j ACCEPT	insert rule at list number to allow all outbound DNS traffic using the udp protocol
sudo iptables -D INPUT 1	remove input table first rule
sudo iptables -D OUTPUT 1	remove output table first rule
sudo iptables -D INPUT -p tcp --dport 22 -j ACCEPT	remove rule from input table by matching the entire rule configured *(accept ssh in shown)*
sudo iptables -F INPUT	flush entire input chain ruleset (chain behavior remains the same)
sudo iptables -F OUTPUT	flush entire output chain ruleset (chain behavior remains the same)
sudo iptables -L {table-name}	show rules listed in specified table *(filter table is default)* - tables: nat, mangle, raw, security
sudo iptables -L -nv --line-numbers	show all rules in the filter table by line number

LOGGING RULES

rules	***comment/description***
sudo iptables -A INPUT -m limit --limit 20/hr -j LOG --log-prefix “[netfilter] INPUT:DROP: ” --log-level 7	Any packet that is going to be dropped will be

sudo iptables -A OUTPUT -m limit --limit 20/hr -j LOG --log-prefix “[netfilter] OUTPUT:DROP: ” --log-level 7	added to the syslog with the specified prefix
sudo iptables -A OUTPUT -p tcp -m multiport --dports 53,80,443,8080,8443 -m limit --limit 20/hr -j LOG --log-prefix “[netfilter] OUTPUT:DROP: ” --log-level 7	Monitors for outbound traffic over ports commonly used by C2 servers

SCORING ENGINE THOUGHTS

rules	***comment/description***
sudo iptables -A INPUT -s {scoring-IP} -m conntrack --ctstate NEW -j ACCEPT	Basic rules for allowing scoring communications in and out **The OUTPUT rule here most likely isn’t even needed as the session tracking rule will handle the outbound traffic from the box
sudo iptables -A OUTPUT -d {scoring-IP} -m conntrack --ctstate NEW -j ACCEPT	
sudo iptables -A INPUT -p {tcp/udp} --dport {pX} -s {scoring-IP,VLAN-Subnet,LAN-Subnet} -m conntrack --ctstate NEW -j ACCEPT	Map the service you need to provide to the scoring engine and/or the internal subnet for TCP and UDP communications ***NEEDS RESEARCH
sudo iptables -A INPUT -p icmp --icmp-type 8 -s {scoring-IP,VLAN-subnet,LAN-subnet} -j ACCEPT	Ping *might* require an inbound and outbound rule, but need to look into that
sudo iptables -A OUTPUT -p icmp --icmp-type 0 -d {scoring-IP,VLAN-subnet,LAN-subnet} -j ACCEPT	

EXAMPLE RULESET [*AS A FLOW CHART*] FOR COMPETITIONS

rules	***comment/description***
CONFIGURE A SAFETY-NET ‘ALLOW ANY-ANY' & CHANGE CHAIN RULES	
sudo iptables -A INPUT -j ACCEPT	Create allow any-any rules while creating custom ruleset (prevents loss in scoring during this time)
sudo iptables -A OUTPUT -j ACCEPT	
sudo iptables -P INPUT DROP	Change the default chain rules to drop all packets if no rules match **Don’t deny FORWARD if on proxy server that actually routes
sudo iptables -P FORWARD DROP	
sudo iptables -P OUTPUT DROP	
BASELINE RULES: TRACK CURRENT SESSIONS & ALLOW LOOPBACK TESTING	
sudo iptables -A INPUT -i lo -j ACCEPT	Allow traffic over the loopback interface (test services locally)
sudo iptables -A OUTPUT -o lo -j ACCEPT	
sudo iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT	Uphold current connections with the system (prevents a lockout

sudo iptables -A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT	from ssh when removing the allow any-any)
CREATE ALLOW RULES FOR NECESSARY PORTS	
sudo iptables -A INPUT -p tcp --dport 22 -s {VPN-subnet} -m conntrack --ctstate NEW -j ACCEPT	Allows all ssh connections into the system (can specify IPs) **See ALTERNATE CONFIGS section for more info
sudo iptables -A OUTPUT -p udp --dport 53 -s 1.1.1.1 -j ACCEPT	Allows outbound DNS requests to be sent—necessary for curls, apt installs, accessing websites **See ALTERNATE CONFIGS section for more info
sudo iptables -A OUTPUT -p tcp --dport 53 -s 1.1.1.1 -j ACCEPT	
sudo iptables -A OUTPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW -j ACCEPT	Allow HTTP/HTTPS connections to be created outbound
WHITELIST THE DAMN SCORING-ENGINE	
sudo iptables -A INPUT -s {scoring-IP} -m conntrack --ctstate NEW -j ACCEPT	Basic rules for allowing scoring communications in and out **See ALTERNATE CONFIGS section for more info
sudo iptables -A OUTPUT -d {scoring-IP} -m conntrack --ctstate NEW -j ACCEPT	
ADD LOGGING FOR DROPPED PACKETS	
sudo iptables -A INPUT -m limit --limit 20/hour --limit-burst 20 -j LOG --log-prefix "[netfilter] INPUT:DROP: " --log-level 7	Adding logging into the ruleset that checks for any packages about to be blocked
sudo iptables -A INPUT -m limit --limit 20/hour --limit-burst 20 -j LOG --log-prefix "[netfilter] INPUT:DROP: " --log-level 7	
DELETE SAFETY RULES & THOSE THAT ARE TOO PERMISSIVE	
sudo iptables -D INPUT 1	Remove allow any-any rules when ruleset allows scoring
sudo iptables -D OUTPUT 1	
sudo iptables -D OUTPUT -p udp --dport 53 -j ACCEPT	Remove DNS rules after lunch to defend against the intensified red team attacks
sudo iptables -D OUTPUT -p tcp --dport 53 -j ACCEPT	
sudo iptables -D OUTPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW -j ACCEPT	Remove HTTP/S rule(s) to prevent communications to C2 servers
ALTERNATE/ADDITIONAL OPTIONS FOR MORE GRANULARITY	
sudo iptables -A INPUT -p {tcp/udp} --dport {pX} -s {scoring-IP} -m conntrack --ctstate NEW -j ACCEPT	Allow scoring and the internal LAN to reach specific ports on the box
sudo iptables -A OUTPUT -p udp --dport 53 -d $(cat /etc/resolv.conf | grep -m 1 nameserver | awk ‘{print $2}’) -j ACCEPT	One-liner command that takes the first listed nameserver from the resolv.conf file and only allows outbound DNS requests to it

sudo iptables -A OUTPUT -p tcp --dport 53 -d $(cat /etc/resolv.conf | grep -m 1 nameserver | awk ‘{print $2}’) -j ACCEPT	(prevents communication to C2 servers over DNS) **_**Change DNS server in resolv.conf to the active directory DN server or a public one first_**
sudo iptables -A INPUT -p tcp -m multiport --dports 20,21 -m conntrack --ctstate NEW -j ACCEPT	Configure an ftp rule in and outbound for vsftpd file transfers **_**Tying port rules to known IPs is best for security, but adds overhead and complexity_**
sudo iptables -A OUTPUT -p tcp -m multiport --dports 20,21 -m conntrack --ctstate NEW -j ACCEPT	
sudo iptables -A INPUT -p tcp --dport 22 -s {X.X.X.X} -j ACCEPT	For ssh incoming you can specify the IP of your device, a jump-point, or even a subnet itself
sudo iptables -A INPUT -p {tcp/udp} --dport {pX} -s {scoring-IP} -m conntrack --ctstate NEW -j ACCEPT	If you know the ports and services running on your device (you should, but adding them isn’t always conducive to a 5-min plan), you can always make the scoring-engine communication rules more granular
sudo iptables -A OUTPUT -p {tcp/udp} --dport {pX} -d {scoring-IP} -m conntrack --ctstate NEW -j ACCEPT	