#!/bin/bash

#storyline: Extract IPs from emergingthreats.net and create a firewall ruleset

#!/bin/bash

# switches to include the block menu later in the lab

while getopts 'icwmod:b' OPTION; do

case "$OPTION" in

i) iptables=1 ;;

c) cisco=1 ;;

w) windows=1 ;;

m) mac=1 ;;

o) output=${OPTARG} ;;

d) domain\_filter=1 ;;

b) block\_menu=1 ;;

\*)

echo "Invalid Value"

echo "use -i, -c, -w, -m, -o <output>, -d, -b"

exit 1

esac

done

# function to display the block menu for #2

function block\_menu() {

echo "Block list menu"

PS3='Please enter your choice: '

options=("Mac blocklist generator" "Windows blocklist generator" "Cisco blocklist generator" "Quit")

select opt in "${options[@]}"

do

case $opt in

"Mac blocklist generator")

echo "Generating Mac blocklist"

./parse-threat.bash -m -o pf

;;

"Windows blocklist generator")

echo "Generating Windows blocklist"

./parse-threat.bash -w -o badips

;;

"Cisco blocklist generator")

echo "Generating Cisco blocklist"

./parse-threat.bash -c -o badips

;;

"Quit")

break

;;

\*) echo "Invalid option $REPLY";;

esac

done

}

# originall had $0 but wouldnt work unless ./parse-threat.bash was put in

# if -b is set, then the menu will come up.

if [[ ${block\_menu} ]]; then

block\_menu

exit 0

fi

File="/tmp/emerging-drop.suricata.rules"

# if statement to check if the file is downloaded on the different block menu options.

if [[ -f "${File}" ]]; then

echo -n "The file exists. Want to redownload it? Y|N "

else

echo -n "The file does not exist. Download it? Y|N "

fi

read download

if [[ "${download}" == "N" || "${download}" == "n" ]]; then

echo "Exiting"

exit 0

elif [[ "${download}" == "Y" || "${download}" == "y" ]]; then

wget https://rules.emergingthreats.net/blockrules/emerging-drop.suricata.rules -O ${File}

echo "Downloaded!!!"

fi

# Create a firewall ruleset

egrep -o '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' "${File}" | sort -u | tee badIPs.txt

# Generate rules

if [[ ${iptables} ]]; then

echo "Generating IPtables"

for eachIP in $(cat badIPs.txt); do

echo "iptables -A INPUT -s ${eachIP} -j DROP" | tee -a ${output:-badIPs}.iptables

done

fi

if [[ ${mac} ]]; then

echo "Generating Mac file"

mFile=${output:-"pf"}.conf

if [[ ! -f "${mFile}" ]]; then

echo -e 'scrub-anchor "com.apple/\*"\nnat-anchor "com.apple/\*"\nrdr-anchor "com.apple/\*"\ndummynet-anchor "com.apple/\*"\nanchor "com.apple/\*"\nload anchor "com.apple" from "/etc/pf.anchors/com.apple"' | tee ${mFile}

fi

for eachIP in $(cat badIPs.txt); do

echo "block in from ${eachIP} to any" | tee -a ${mFile}

done

fi

if [[ ${cisco} ]]; then

echo "Generating Cisco file"

for eachIP in $(cat badIPs.txt); do

echo "access-list 1 deny ip ${eachIP} any" | tee -a ${output:-"cisco"}.conf

done

fi

if [[ ${domain\_filter} ]]; then

echo "Generating Cisco URL filter ruleset..."

echo "class-map match-any BAD\_URLS" > ${output:-cisco\_url\_ruleset}.conf

wget -q -O - https://raw.githubusercontent.com/botherder/targetedthreats/master/targetedthreats.csv | \

awk -F ',' '/domain/ { print "match protocol http host \"" $2 "\"" }' >> ${output:-cisco\_url\_ruleset}.conf

cat ${output:-cisco\_url\_ruleset}.conf

echo "Cisco URL filter ruleset generated!"

fi

if [[ ${windows} ]]; then

echo "Generating Windows file"

wFile=${output:-"badips"}.netsh

for eachIP in $(cat badIPs.txt); do

# this appends windows firewall rules to block IP addresses.

echo "netsh advfirewall firewall add rule name=\"BLOCK IP ADDRESS - ${eachIP}\" dir=in action=block remoteip=${eachIP}" | tee -a ${wFile}

done

fi