BITSHIELD

CYBER SECURITY

CONFIDENTIALITY INTEGRITY AVAILABILITY

LINUX FIREWALL

POLICY/SCRIPT

Design and implementation of a packet filter firewall including policy configuration and packet filtering via bash shell scripting and iptables on a Linux O/S

WHITE PAPER

Document Prominence

Copyright in this document is vested in Terence Broadbent and is issued in confidence only for the purpose for which it is supplied. Information contained herein must not be reproduced in whole or in part nor communicated to any third party except with the prior written authority of an authorized representative of the business, subject to any conditions or limitations defined in the authority.

Unauthorised copying, replication or distribution is strictly prohibited in all countries of the World.

This document always remains the individual property of Terence Broadbent.

Document Handling

ACCESS	HANDLING	COMMUNICATION	DISPOSAL
Access to this file is restricted to authorized personnel only.	Electronic documents are stored in an encrypted state on electrostatic media.	Documents are to be communicated externally in accordance with business standards and Customer requirements.	Only dispose in accordance with document retention scheme.
The original electronic version is to be held at business HQ only.	Hardcopies of this document must be held securely and not left unattended at any time.	This document is to be classified as uncontrolled if printed.	All hard copies of this document will be shredded when the final version has been authorized.

Customer Reference

Customer Reference	Internal Document
Customer Contact Name	-
Customer Contact Number	-

Business Reference

Business Reference	WP-2018
Business Contact Name	Terence Broadbent – Business Manager
Cyber Contact Number	

Document Inspection

Document Creator	Terence Broadbent
Version Number	1.0
Approved By	Terence Broadbent – Document Creator
Checked By	Terence Broadbent – Document Author
Status	Issued – Approved

Version History

VERSION	NAME	DATE STAMP
1.0	Terence Broadbent – Document Author	25/11/2018

Business HQ

STAMP ADDRESS
HERE

Document Tracking Reference Number - 0000 000X

DOCUMENT CONTENTS

Design and implementation of a packet filter firewall including policy configuration and packet filtering via bash shell scripting and iptables on a Linux O/S

LIST OF ILLUSTRATIONS

Design and implementation of a packet filter firewall including policy configuration and packet filtering via bash shell scripting and iptables on a Linux O/S

LIST OF TABLES

Design and implementation of a packet filter firewall including policy configuration and packet filtering via bash shell scripting and iptables on a Linux O/S

SECURITY POLICY

Employment policies describe best-practices for employees and managers, likewise security policies describe best-practice for how the Company wants to protect its information assets. Security policies are high-level plans that describe the goals of the procedures - they are not guidelines or standards, nor are they procedures or controls. Policies describe security in general terms, not specifics, they provide the blueprint for an overall security program - (Scott Barman, 2001).

However, Dubrawsky and Noonan argue that the term security policy has several different meanings within the security industry. On one hand, it refers to the actual written policies that dictate how a Small to Medium Enterprise (SME) manages the security of their information assets. On the other hand, it refers to the actual configuration of the device in question, such as with an access control list or firewall rule set - (Ido Dubrawsky, Wes Noonan, 2006).

The National Institute of Standards and Technology (NIST) states that a firewall policy must dictate how the firewall should handle Network traffic for specific IP addresses and address ranges, protocols, applications and content types based on the SME's information security policies.

A firewall policy should be documented in the system security plan and maintained and updated frequently as classes of new attacks or vulnerabilities arise, or as the SME's needs regarding Network applications change. When a firewall policy is written, some form of risk analysis should be performed to develop a list of the types of traffic required by the SME's and categorise how they will be secured - (Karen Scarfone, Paul Hoffman, 2009).

FIREWALL POLICY

NIST goes on to state that a firewall policy should only allow essential IP protocols through - commonly used IP protocols include ICMP, TCP and UDP. These essential protocols should be restricted whenever possible to specific Hosts and Networks within the SME with a need to use them. By permitting only essential protocols, all unessential IP protocols are denied by default - (Karen Scarfone, Paul Hoffman, 2009).

Detailed below is the firewall policy for Staffordshire University simple firewall.

SIMPLE FIREWALL POLICY

Where electronic equipment is used to capture, process or store data identified by Staffordshire University as “Legal/Private/ Confidential/ Restricted” and the electronic equipment is accessible via a direct or indirect Internet connection, a network firewall appropriately installed, configured and maintained is required.

All installations and implementations of and modifications to a network firewall and its configuration and ruleset are the responsibility of the authorised Staffordshire University Information Technology (SUIT) firewall Administrator, with this exception: maintenance of a network firewall ruleset may be performed by other than SUIT personnel where permitted by a documented agreement between SUIT and the School/ Department/ Business-unit assuming the firewall Administrator’s responsibilities.

TECHNICAL NOTES

Staffordshire University has put the option of limiting rather than blocking in the below policy because the ICMP protocol exists for a reason and not all of that reason is ping. It is a meta protocol that is used to communicate control messages about the network itself - the same logic can also be applied to SYN packets.

Further, it was also prudent to include universal blocking of DoS attacks into the policy on 03/04/2018 as these types of attacks occur on a regular basis.

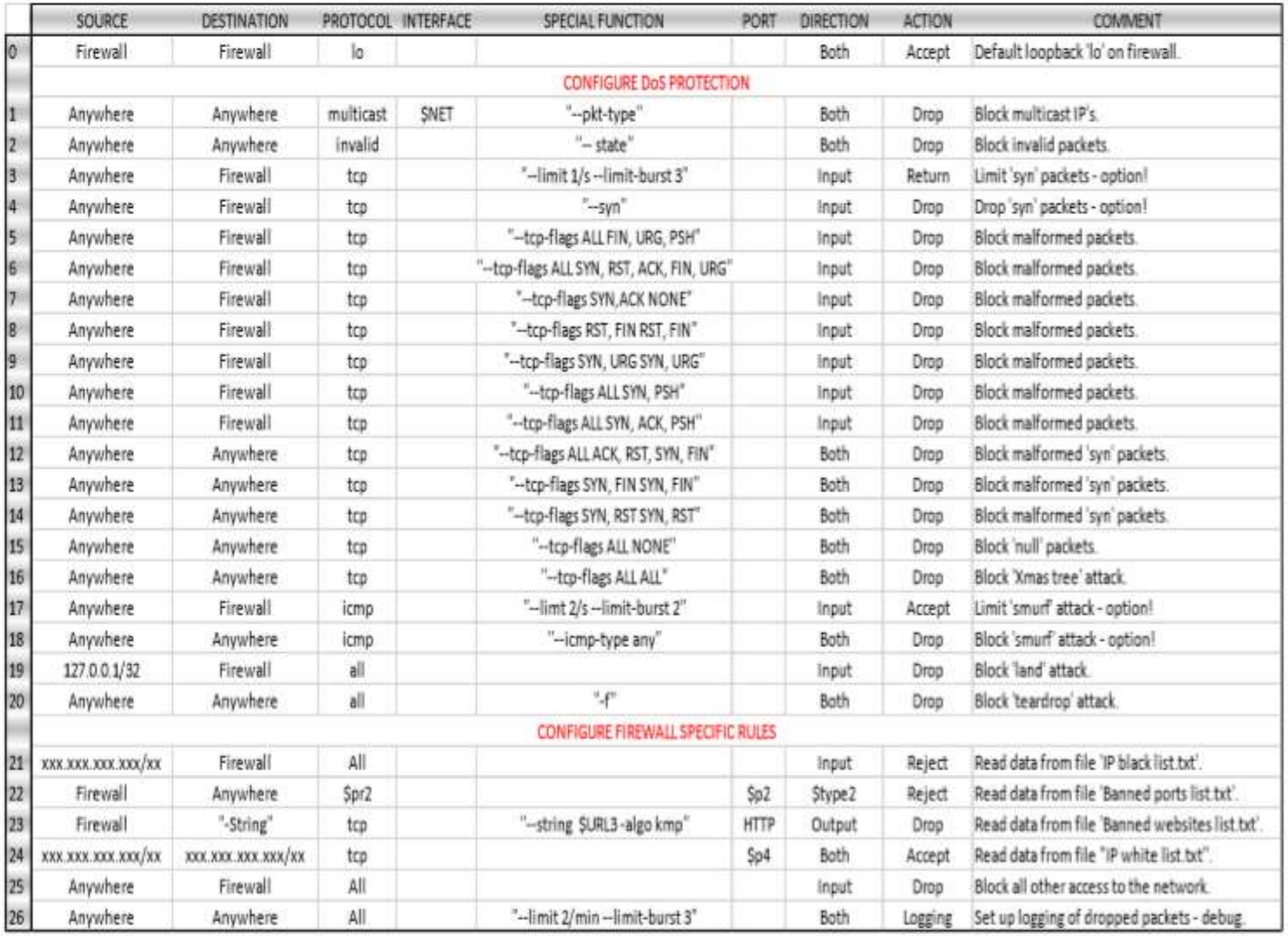
DOCUMENT APPROVAL

Signed:	Terence Broadbent - November 2018 ***Terence Broadbent BSc Cyber Security (First Class).***

VERSION CONTROL

VERSION	DATE	CHANGE
Version 1.0	25/11/2018	Initial draft for review.

CONFIGURATION TABLE



	SOURCE	DESTINATION	PROTOCOL	INTERFACE	SPECIAL FUNCTION	PORT	DIRECTION	ACTION	COMMENT
0	Firewall	Firewall	lo				Both	Accept	Default loopback 'lo' on firewall.
					CONFIGURE DoS PROTECTION				
1	Anywhere	Anywhere	multicast	$NET	"--pkt-type"		Both	Drop	Block multicast IP's.
2	Anywhere	Anywhere	invalid		"-- state"		Both	Drop	Block invalid packets.
3	Anywhere	Firewall	tcp		"--limit 1/s --limit-burst 3"		Input	Return	Limit 'syn' packets - option!
4	Anywhere	Firewall	tcp		"--syn"		Input	Drop	Drop 'syn' packets - option!
5	Anywhere	Firewall	tcp		"--tcp-flags ALL FIN, URG, PSH"		Input	Drop	Block malformed packets.
6	Anywhere	Firewall	tcp		"--tcp-flags ALL SYN, RST, ACK, FIN, URG"		Input	Drop	Block malformed packets.
7	Anywhere	Firewall	tcp		"--tcp-flags SYN,ACK NONE"		Input	Drop	Block malformed packets.
8	Anywhere	Firewall	tcp		"--tcp-flags RST, FIN RST, FIN"		Input	Drop	Block malformed packets.
9	Anywhere	Firewall	tcp		"--tcp-flags SYN, URG SYN, URG"		Input	Drop	Block malformed packets.
10	Anywhere	Firewall	tcp		"--tcp-flags ALL SYN, PSH"		Input	Drop	Block malformed packets.
11	Anywhere	Firewall	tcp		"--tcp-flags ALL SYN, ACK, PSH"		Input	Drop	Block malformed packets.
12	Anywhere	Anywhere	tcp		"--tcp-flags ALL ACK, RST, SYN, FIN"		Both	Drop	Block malformed 'syn' packets.
13	Anywhere	Anywhere	tcp		"--tcp-flags SYN, FIN SYN, FIN"		Both	Drop	Block malformed 'syn' packets.
14	Anywhere	Anywhere	tcp		"--tcp-flags SYN, RST SYN, RST"		Both	Drop	Block malformed 'syn' packets.
15	Anywhere	Anywhere	tcp		"--tcp-flags ALL NONE"		Both	Drop	Block 'null' packets.
16	Anywhere	Anywhere	tcp		"--tcp-flags ALL ALL"		Both	Drop	Block 'Xmas tree' attack.
17	Anywhere	Firewall	icmp		"--limt 2/s --limit-burst 2"		Input	Accept	Limit 'smurf' attack - option!
18	Anywhere	Anywhere	icmp		"--icmp-type any"		Both	Drop	Block 'smurf' attack - option!
19	127.0.0.1/32	Firewall	all				Input	Drop	Block 'land' attack.
20	Anywhere	Anywhere	all		"-f"		Both	Drop	Block 'teardrop' attack.
					CONFIGURE FIREWALL SPECIFIC RULES				
21	xxx.xxx.xxx.xxx/xx	Firewall	All				Input	Reject	Read data from file 'IP black list.txt'.
22	Firewall	Anywhere	$pr2			$p2	$type2	Reject	Read data from file 'Banned ports list.txt'.
23	Firewall	"-String"	tcp		"--string $URL3 -algo kmp"	HTTP	Output	Drop	Read data from file 'Banned websites list.txt'.
24	xxx.xxx.xxx.xxx/xx	xxx.xxx.xxx.xxx/xx	tcp			$p4	Both	Accept	Read data from file "IP white list.txt".
25	Anywhere	Firewall	All				Input	Drop	Block all other access to the network.
26	Anywhere	Anywhere	All		"--limit 2/min --limit-burst 3"		Both	Logging	Set up logging of dropped packets - debug.

SOURCE

Allow HOST traffic to loopback lo interface.
Allow USER traffic to smtp and pop3 services.
Allow NETWORK traffic to port 7001 from IP address 192.168.50.21.

Match specified packets with source addresses.
Match specified packets with source port numbers.

DESTINATION

Allow NETWORK traffic to loopback lo interface.
Allow USER traffic to smtp and pop3 services.
Allow USER traffic to HTTP(S) services.

Match specified packets with destination addresses.
Match specified packets with destination port numbers.

PROTOCOL

Telnet port 23 – NOT STIPULATED.
SSH port 22 - BLOCK.
FTP port 21 – NOT STIPULATED.
SMTP port 25 – ALLOW.
POP port 110 – ALLOW.
IMAP port 143 – ALLOW.
HTTP port 80 - BLOCK.
SSL port 445 – BLOCK.

MAIL

Allow USER traffic to smtp and pop3 services.

ACTIONS

PERMIT traffic through loopback lo on host interface.
PERMIT smtp and pop3 services on host.
PERMIT services on port 3306 on host.
PERMIT host access on port 7001 from client 192.168.50.21
DROP multicast IP's.
DROP invalid TCP packets from client.
DROP null packets from client.
REJECT host access to port 3333.
REJECT host access from client 169.254.0.0/16.
REJECT or limit a SYN flood attack from a client.
BLOCK client access to HTTP services on host.
BLOCK or limit client ICMP flooding.
BLOCK host from visiting specified web sites.
BLOCK host access to SSH.
BLOCK client DoS attacks.
BLOCK all other client traffic by default.
LOG all rejected packets.

BLACK LIST

USE "IP black list.txt" with identified IP addresses/ranges.

WHITE LIST

USE "IP white list.txt" with identified IP addresses/ranges and port number.

BANNED PORTS

USE "Banned ports list.txt" with identified INPUT/OUTPUT, tcp/udp and port number.

BANNED WEB SITES
USE “Banned website list.txt” with identified URL of the banned website.

Figure 1 - Simple Firewall Policy

FIREWALL RULES

Detailed below are the actual implemented Iptables rules based on the above firewall policy.

CHAIN INPUT(POLICY DROP)				
No	TARGET	PROT OPT SOURCE	DESTINATION	INFORMATION
1	ufw-before-logging-input all	-- anywhere	anywhere	
2	ufw-before-input all	-- anywhere	anywhere	
3	ufw-after-input all	-- anywhere	Anywhere	
4	ufw-after-logging-input all	-- anywhere	anywhere	
5	ufw-reject-input all	-- anywhere	anywhere	
6	ufw-track-input all	-- anywhere	anywhere	
7	ACCEPT	all -- anywhere	anywhere	
8	DROP	all -- anywhere	anywhere	PKTTYPE = multicast
9	DROP	all -- anywhere	anywhere	state INVALID
10	RETURN	tcp -- anywhere	anywhere	tcp flags:FIN,SYN,RST,ACK/SYN limit: avg 1/sec burst 3
11	DROP	tcp -- anywhere	anywhere	tcp flags:FIN,SYN,RST,PSH,ACK,URG/FIN,PSH,URG
12	DROP	tcp -- anywhere	anywhere	tcp flags:FIN,SYN,RST,PSH,ACK,URG/FIN,SYN,RST,ACK,URG

13	DROP	tcp -- anywhere	anywhere	tcp flags:SYN,ACK/NONE
14	DROP	tcp -- anywhere	anywhere	tcp flags:FIN,RST/FIN,RST
15	DROP	tcp -- anywhere	anywhere	tcp flags:SYN,URG/SYN,URG
16	DROP	tcp -- anywhere	anywhere	tcp flags:FIN,SYN,RST,PSH,ACK,URG/SYN,PSH
17	DROP	tcp -- anywhere	anywhere	tcp flags:FIN,SYN,RST,PSH,ACK,URG/SYN,PSH,ACK
18	DROP	tcp -- anywhere	anywhere	tcp flags:FIN,SYN,RST,PSH,ACK,URG/FIN,SYN,RST,ACK
19	DROP	tcp -- anywhere	anywhere	tcp flags:FIN,SYN/FIN,SYN
20	DROP	tcp -- anywhere	anywhere	tcp flags:SYN,RST/SYN,RST
21	DROP	tcp -- anywhere	anywhere	tcp flags:FIN,SYN,RST,PSH,ACK,URG/NONE
22	DROP	tcp -- anywhere	anywhere	tcp flags:FIN,SYN,RST,PSH,ACK,URG/FIN,SYN,RST,PSH,ACK,URG
23	ACCEPT	icmp -- anywhere	anywhere	limit: avg 2/sec burst 2
24	DROP	all -- localhost	anywhere	
25	DROP	all -f anywhere	anywhere	
26	REJECT	all -- 0.0.0.0/8		anywhere reject-with icmp-port-unreachable
27	REJECT	all -- 10.0.0.0/8	anywhere	reject-with icmp-port-unreachable
28	REJECT	all -- 100.64.0.0/10	anywhere	reject-with icmp-port-unreachable
29	REJECT	all -- 127.0.0.0/8	anywhere	reject-with icmp-port-unreachable
30	REJECT	all -- linklocal/16	anywhere	reject-with icmp-port-unreachable
31	REJECT	all -- 172.16.0.0/12	anywhere	reject-with icmp-port-unreachable
32	REJECT	all -- 192.0.2.0/24	anywhere	reject-with icmp-port-unreachable
33	REJECT	all -- 192.168.0.0/16	anywhere	reject-with icmp-port-unreachable
34	REJECT	all -- 198.18.0.0/15	anywhere	reject-with icmp-port-unreachable
35	REJECT	all -- 198.51.100.0/24	anywhere	reject-with icmp-port-unreachable
36	REJECT	all -- 203.0.113.0/24	anywhere	reject-with icmp-port-unreachable
37	REJECT	all -- baseaddress.mcast.net/3	anywhere	reject-with icmp-port-unreachable
38	REJECT	tcp -- anywhere	anywhere	tcp dpt:0 reject-with icmp-portunreachable
39	REJECT	udp -- anywhere	anywhere	udp dpt:0 reject-with icmp-portunreachable
40	REJECT	tcp -- anywhere	anywhere	tcp dpt:http reject-with icmp-portunreachable
41	REJECT	udp -- anywhere	anywhere	udp dpt:http reject-with icmp-portunreachable
42	REJECT	tcp -- anywhere	anywhere	tcp dpt:loc-srv reject-with icmp-portunreachable
43	REJECT	udp -- anywhere	43 anywhere	udp dpt:loc-srv reject-with icmp-portunreachable
44	REJECT	tcp -- anywhere	anywhere	tcp dpt:136 reject-with icmp-portunreachable

45	REJECT	udp -- anywhere	anywhere	udp dpt:136 reject-with icmp-portunreachable
46	REJECT	tcp -- anywhere		anywhere tcp dpt:netbios-ns reject-with icmp-portunreachable
47	REJECT	udp -- anywhere	anywhere	udp dpt:netbios-ns reject-with icmp-portunreachable
48	REJECT	tcp -- anywhere	anywhere	tcp dpt:netbios-dgm reject-with icmpport-unreachable
49	REJECT	udp -- anywhere	anywhere	udp dpt:netbios-dgm reject-with icmpport-unreachable
50	REJECT	tcp -- anywhere	anywhere	tcp dpt:netbios-ssn reject-with icmp-portunreachable
51	REJECT	udp -- anywhere	anywhere	udp dpt:netbios-ssn reject-with icmpport-unreachable
52	REJECT	tcp -- anywhere	anywhere	tcp dpt:microsoft-ds reject-with icmpport-unreachable
53	REJECT	udp -- anywhere	anywhere	udp dpt:microsoft-ds reject-with icmpport-unreachable
54	REJECT	tcp -- anywhere	anywhere	tcp dpt:socks reject-with icmp-portunreachable
55	REJECT	udp -- anywhere	anywhere	udp dpt:socks reject-with icmp-portunreachable
56	REJECT	tcp -- anywhere	anywhere	tcp dpt:3333 reject-with icmp-portunreachable
57	REJECT	udp -- anywhere	anywhere	udp dpt:3333 reject-with icmp-portunreachable
58	ACCEPT	tcp -- anywhere	anywhere	tcp dpt:smtp

CHAIN OUTPUT(POLICY ACCEPT)

No	TARGET	PROT OPT SOURCE	DESTINATION	INFORMATION
1	ufw-before-loggingoutput all	-- anywhere	anywhere	
2	ufw-before-output all	-- anywhere	anywhere	
3	ufw-after-output all	-- anywhere	anywhere	
4	ufw-after-loggingoutput all	-- anywhere	anywhere	
5	ufw-reject-output all	-- anywhere	anywhere	
6	ufw-track-output all	-- anywhere	anywhere	
7	ACCEPT	all -- anywhere	anywhere	
8	DROP	all -- anywhere	anywhere	PKTTYPE = multicast
9	DROP	all -- anywhere	anywhere	state INVALID
10	DROP	icmp -- anywhere	anywhere	
11	DROP	tcp -- anywhere	anywhere	tcp flags:FIN,SYN,RST,PSH,ACK,URG/FIN,S

				YN,RST,PSH,ACK,URG
12	DROP	tcp -- anywhere	anywhere	tcp flags:FIN,SYN,RST,PSH,ACK,URG/NONE
13	DROP	tcp -- anywhere	anywhere	tcp flags:FIN,SYN,RST,PSH,ACK,URG/FIN,SYN,RST,ACK
14	DROP	tcp -- anywhere	anywhere	tcp flags:FIN,SYN/FIN,SYN
15	DROP	tcp -- anywhere	anywhere	tcp flags:SYN,RST/SYN,RST
16	DROP	all -f anywhere	anywhere	
17	REJECT	tcp -- anywhere	anywhere	tcp dpt:ssh reject-with icmp-portunreachable
18	REJECT	udp -- anywhere	anywhere	udp dpt:ssh reject-with icmp-portunreachable
19	DROP	tcp -- anywhere	anywhere	STRING match "facebook.co.uk" ALGO name kmp TO 65535
20	DROP	tcp -- anywhere	anywhere	STRING match "facebook.com" ALGO name kmp TO 65535
21	DROP	tcp -- anywhere	anywhere	STRING match "twitter.co.uk" ALGO name kmp TO 65535
22	DROP	tcp -- anywhere	anywhere	STRING match "twitter.com" ALGO name kmp TO 65535
23	DROP	tcp -- anywhere	anywhere	STRING match "myspace.co.uk" ALGO name kmp TO 65535
24	DROP	tcp -- anywhere	anywhere	STRING match "myspace.com" ALGO name kmp TO 65535
25	DROP	tcp -- anywhere	anywhere	STRING match "linkedin.co.uk" ALGO name kmp TO 65535
26	DROP	tcp -- anywhere	anywhere	STRING match "linkedin.com" ALGO name kmp TO 65535
27	DROP	tcp -- anywhere	anywhere	STRING match "instagram.co.uk" ALGO name kmp TO 65535
28	DROP	tcp -- anywhere	anywhere	STRING match "instagram.com" ALGO name kmp TO 65535
29	ACCEPT	tcp -- anywhere	anywhere	tcp dpt:smtp
30	ACCEPT	tcp -- anywhere	anywhere	tcp dpt:pop3
31	ACCEPT	tcp -- anywhere	anywhere	tcp dpt:imap2
32	ACCEPT	tcp -- anywhere	anywhere	tcp dpt:urd
33	ACCEPT	tcp -- anywhere	anywhere	tcp dpt:imaps
34	ACCEPT	tcp -- anywhere	anywhere	tcp dpt:pop3s
35	ACCEPT	tcp -- anywhere	anywhere	tcp dpt:mysql
36	ACCEPT	tcp -- anywhere	192.168.50.21	tcp dpt:afs3-callback
37	LOGGING	all -- anywhere	anywhere	

Table 1 - IptableS Rules

Technical note – the above is founded on limiting rather than blocking ICMP/SYN packets as per the Cyber security officer’s recommendation.

IPTABLES

A ‘Defence in Depth’ approach (also known as the Castle Approach) is an Information Assurance (IA) concept in which multiple layers of security controls (defences) are placed throughout an Information Technology (IT) system.

When applying such concepts to the Information Security of a Linux Network System, Iptables are the perfect companion tool for a Cyber-security Professional to utilise to secure layer 3 and 4 of the system via a Firewall.

A Firewall is a Network Security System that monitors, and controls incoming and outgoing layer 3 and 4 traffic based on predetermined security rules. Layer 3 in the OSI model is the Network Layer where IP works and Layer 4 is the Transport Layer where TCP and UDP function.

INSTALLATION SCRIPT Cyber-security professional often have to set up Iptables on various networks and computers and as a result often use installation scripts to accomplish this, the following is a simple text-based script written in bash to accomplish such a task based on the above firewall policy.

Technical Note: Entering hundreds of IP and port addresses can be a very laborious and error incurring task, hence this script has been designed and coded to read-in data from smaller feeder files that can be easily maintained and updated more effectively than writing out each command by hand.

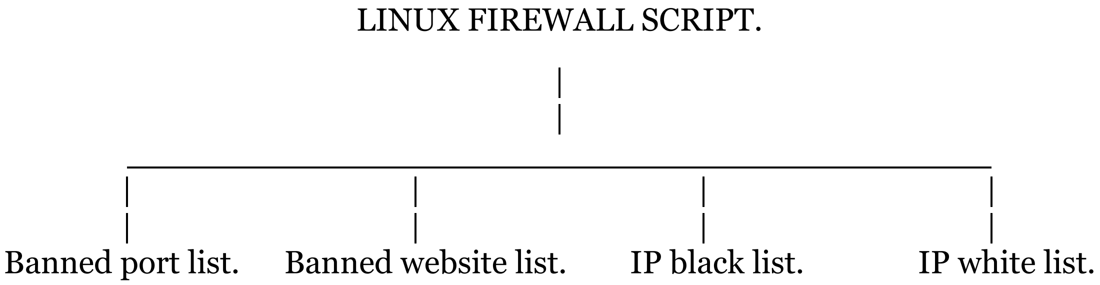


Figure 2 - Firewall Script Hierarchy

This modular-design approach allows for speculative expansion of file lists that may become pertinent in the future – such as a list to identify suspicious internal/external IPs and track and log them etc.

LINUX INSTALLATION SCRIPT

```
#!/bin/sh
#***********************************************************************#
#                                                                       #
#            Firewall implementation script for firewall policy         #
#                                                                       #
#        Release version 1.0 by Terence Broadbent BSc Cyber Security    #
#                                                                       #
# ********************************************************************* #

# ********************************************************************* #
# AUTHOR  : Terence Broadbent                                           #
# CONTRACT: Stafford University                                         #
# Version : 1.0                                                         #
# Details : Define any global variables that will be used throughout this bash script.#
# Modified: N/A                                                         #
# ********************************************************************* #

NET="ens33"              # IMPORTANT!! CHANGE THIS TO MATCH YOUR NETWORK
IFS=","                  # Enables this script to read data from text files separated by commas.
LOGFILE="./Log1.txt" # The default log filename.

# ********************************************************************* #
# AUTHOR  : Terence Broadbent                                           #
# CONTRACT: Stafford University                                         #
# Version : 1.0                                                         #
# Details : Create a logging system & check that this bash script has root privileges.#
# Modified: N/A                                                         #
# ********************************************************************* #

echolog() ( echo $1 echo $1 >> $LOGFILE )

if [ $USER != "root" ]
   then echolog "Please run this bash script as root..."
   exit 0
else
   echolog "\n\tLINUX FIREWALL INSTALLATION LOG - VERSION 1.0\n"
fi

# ********************************************************************* #
# AUTHOR  : Terence Broadbent                                           #
# CONTRACT: Stafford University                                         #
# Version : 1.0                                                         #
# Details : Ensure the current firewall configuration is backed up then wiped clean.#
# Modified: N/A                                                         #
# ********************************************************************* #

echolog "[1]. Starting the firewall installation...\n"
iptables-save -c > "./Iptables-old.txt" 2>&1 | tee -a $LOGFILE
echolog "\t - Your current settings have been saved to ./Iptables-old.txt\n"
echolog "[2]. Cleaning up any existing firewall protocols...\n"
echolog "\t + Stopping iptables services." ufw disable 2>&1 | tee -a $LOGFILE
echolog "\t + Cleaning iptables." iptables -F 2>&1 | tee -a $LOGFILE
iptables -t nat -F 2>&1 | tee -a $LOGFILE
iptables -t mangle -F 2>&1 | tee -a $LOGFILE
iptables -X 2>&1 | tee -a $LOGFILE
echolog "\t + Iptables cleaned and wiped."
echolog "\t + Restarting services." ufw enable 2>&1 | tee -a $LOGFILE
echolog "\t - Cleaning of iptables completed.\n"

# **************************************************************************** #
# AUTHOR  : Terence Broadbent                                                  #
# CONTRACT: Stafford University                                                #
# Version : 1.0                                                                #
# Details : Accept traffic through loopback 'lo' interface on the network.     #
# Modified: N/A                                                                #
# **************************************************************************** #

echolog "[3]. Setting up a loopback on the firewall...\n"
echolog "\t + Allowed: Loopback services."
iptables -A INPUT -i lo -j ACCEPT 2>&1 | tee -a $LOGFILE
iptables -A OUTPUT -o lo -j ACCEPT 2>&1 | tee -a $LOGFILE
echolog "\t - Provision of loopback completed.\n"

# **************************************************************************** #
# AUTHOR  : Terence Broadbent                                                  #
# CONTRACT: Stafford University                                                #
# Version : 1.0                                                                #
# Details : Protect the network from denial of service and pesky hackers.      #
# Modified: N/A                                                                #
# **************************************************************************** #

echolog "[4]. Protecting the network from threat actors (Hackers!)...\n"
echolog "\t + Blocking: Multicast IPs."
iptables -A INPUT -m pkttype --pkt-type multicast -i $NET -j DROP 2>&1 | tee -a $LOGFILE
iptables -A OUTPUT -m pkttype --pkt-type multicast -o $NET -j DROP 2>&1 | tee -a $LOGFILE

# ifconfig $NET -multicast 2>&1 | tee -a $LOGFILE --comment alternative option.

echolog "\t + Blocking: Invalid packets."
iptables -A INPUT -m state --state INVALID -j DROP 2>&1 | tee -a $LOGFILE
echolog "\t - Setting up syn attack configuration..."
echolog "\t   [1] Limit SYN packets (recommended)?"
echolog "\t   [2] Block all SYN packets?"
while true;
     do read -p "Option:" CONT
     if [ "$CONT" = "1" ];
         then echo Option:$CONT >> $LOGFILE
         echolog  "\t + Limiting: SYN flooding attack."
         iptables -A INPUT -p tcp --syn -m limit --limit 1/s --limit-burst 3 -j RETURN
         break
    elif
        [ "$CONT" = "2" ];
        then echo Option:$CONT >> $LOGFILE
        echolog  "\t +    Blocking: SYN flooding attack."
        iptables -A INPUT -p tcp --syn -j DROP 2>&1 | tee -a $LOGFILE
        break
   else
       printf "Error please re-select "
   fi
done
echolog "\t + Blocking: Malformed packets."
iptables -A INPUT -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP 2>&1 | tee -a $LOGFILE
iptables -A INPUT -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP 2>&1 | tee -a $LOGFILE
iptables -A INPUT -p tcp --tcp-flags SYN,ACK NONE -j DROP 2>&1 | tee -a $LOGFILE
iptables -A INPUT -p tcp --tcp-flags RST,FIN RST,FIN -j DROP 2>&1 | tee -a $LOGFILE
iptables -A INPUT -p tcp --tcp-flags SYN,URG SYN,URG -j DROP 2>&1 | tee -a $LOGFILE
iptables -A INPUT -p tcp --tcp-flags ALL SYN,PSH -j DROP 2>&1 | tee -a $LOGFILE
iptables -A INPUT -p tcp --tcp-flags ALL SYN,ACK,PSH -j DROP 2>&1 | tee -a $LOGFILE
echolog "\t + Blocking: Malformed syn packets."
iptables -A INPUT -p tcp --tcp-flags ALL ACK,RST,SYN,FIN -j DROP 2>&1 | tee -a $LOGFILE
iptables -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP 2>&1 | tee -a $LOGFILE
iptables -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j DROP 2>&1 | tee -a $LOGFILE
echolog "\t + Blocking: Null packets."
iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP 2>&1 | tee -a $LOGFILE
echolog  "\t + Blocking: Xmas tree attack."
iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP 2>&1 | tee -a $LOGFILE
echolog "\t - Setting up smurf attack configuration..."
echolog "\t   [1] Limit ICMP packets (recommended)?"
echolog "\t   [2] Block all ICMP packets?"
while true;
   do read -p "Option:" CONT
    if [ "$CONT" = "1" ];
        then echo Option:$CONT >> $LOGFILE
        echolog  "\t + Limiting: Smurf attack."
        iptables -A INPUT -p icmp -m limit --limit 2/second --limit-burst 2 -j ACCEPT 2>&1 | tee -a
$LOGFILE
       break
   elif
       [ "$CONT" = "2" ];
       then echo Option:$CONT >> $LOGFILE
       echolog  "\t + Blocking: Smurf attack."
       iptables -A INPUT -p icmp --icmp-type any -j DROP 2>&1 | tee -a $LOGFILE
       break
   else
       printf "Error please re-select "
  fi
done
echolog  "\t + Blocking: Land attack." iptables -A INPUT -s 127.0.0.1/32 -j DROP 2>&1 | tee -a
$LOGFILE
echolog  "\t + Blocking: Teardrop attack."
iptables -A INPUT -f -j DROP 2>&1 | tee -a $LOGFILE
echolog  "\t + Blocking: Invalid packets from leaving the network."
iptables -A OUTPUT -m state --state INVALID -j DROP 2>&1 | tee -a $LOGFILE
iptables -A OUTPUT -p icmp -j DROP 2>&1 | tee -a $LOGFILE
iptables -A OUTPUT -p tcp --tcp-flags ALL ALL -j DROP 2>&1 | tee -a $LOGFILE
iptables -A OUTPUT -p tcp --tcp-flags ALL NONE -j DROP 2>&1 | tee -a $LOGFILE
iptables -A OUTPUT -p tcp --tcp-flags ALL ACK,RST,SYN,FIN -j DROP 2>&1 | tee -a $LOGFILE
iptables -A OUTPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP 2>&1 | tee -a $LOGFILE
iptables -A OUTPUT -p tcp --tcp-flags SYN,RST SYN,RST -j DROP 2>&1 | tee -a $LOGFILE
iptables -A OUTPUT -f -j DROP 2>&1 | tee -a $LOGFILE
echolog "\t - Protection of the network completed.\n"

# ******************************************************************************** #
# AUTHOR  : Terence Broadbent                                                      #
# CONTRACT: Stafford University                                                    #
# Version : 1.0                                                                    #
# Details : Open a IP black list file and reject them from the network.            #
# TechNote: Best practice is to DROP however - specification states REJECT.        #
# Modified: N/A                                                                    #
# ******************************************************************************** #
```

```
echolog "[5]. Loading the IP black list into the firewall...\n"
echolog "\t + Checking list exists."
test -e "IP black list.txt" 2>&1 | tee -a $LOGFILE
ReturnValue=$?
    if [ $ReturnValue = "1" ]
    then   echolog "\t + Warning! - the required file 'IP black list.txt' is missing...\n"
    exit 1
else
    echolog "\t + List found,all good."
fi
while read ip1 do
    echolog "\t + Rejecting: $ip1"
    iptables -A INPUT -s $ip1 -j REJECT 2>&1 | tee -a $LOGFILE
done < "IP black list.txt"
echolog "\t - Blacklisting IP addresses completed.\n"

# ******************************************************************************** #
# AUTHOR  : Terence Broadbent                                                       #
# CONTRACT: Stafford University                                                     #
# Version : 1.0                                                                     #
# Details : Open a simple list file of protocols and ports to reject from the network. #
# TechNote: Best practice is to DROP however -specification states REJECT.          #
# Modified: N/A                                                                     #
# ******************************************************************************** #

echolog "[6]. Loading the list of ports to block into the firewall...\n"
echolog "\t + Checking list exists." test -e "Banned ports list.txt" 2>&1 | tee -a $LOGFILE
ReturnValue=$?
     if [ $ReturnValue = "1" ]
     then echolog "\t + Warning! the required file 'Blocked ports list.txt' is missing...\n"
     exit 1
else
     echolog "\t + List found,all good."
fi
while read type2 pr2 p2 do
     echolog "\t + Rejecting: $type2 on port $p2 [$pr2]"
     iptables -A $type2 -p $pr2 --destination-port $p2 -j REJECT 2>&1 | tee -a $LOGFILE
done < "Banned ports list.txt"
echolog "\t - Port blocking completed.\n"

# ******************************************************************************** #
# AUTHOR  : Terence Broadbent                                                       #
# CONTRACT: Stafford University                                                     #
# Version : 1.0                                                                     #
# Details : Open a simple list of banned websites to block from the network.        #
# TechNote:  Upgrade to transparent HTTP proxy utilising squid in the future!!      #
# Modified: N/A                                                                     #
# ******************************************************************************** #

echolog "[7]. Loading the list of banned websites into the firewall...\n"
echolog "\t + Checking list exists." test -e "Banned websites list.txt" 2>&1 | tee -a $LOGFILE
ReturnValue=$?
    if [ $ReturnValue = "1" ]
    then   echolog "\t - Warning! the required file 'Banned websites list.txt' is missing..."
    exit 1
else
    echolog "\t + List found,all good."
```

```
fi
while read url3 do
    echolog "\t + Blocking: $url3"
    iptables -A OUTPUT -p tcp -m string --string $url3 --algo kmp -j DROP 2>&1 | tee -a $LOGFILE
done < "Banned websites list.txt"
echolog "\t - Banning websites completed.\n"

# ***************************************************************************** #
# AUTHOR  : Terence Broadbent                                                   #
# CONTRACT: Stafford University                                                 #
# Version : 1.0                                                                 #
# Details : Open a IP white list file and allow them on the network.            #
# Modified: N/A                                                                 #
# ***************************************************************************** #

echolog "[8]. Loading the IP white list into the firewall.\n"
echolog "\t + Checking list exists."
test -e "IP white list.txt" 2>&1 | tee -a $LOGFILE
ReturnValue=$?
    if [ $ReturnValue = "1" ]
    then   echolog "\t - Warning! the required file 'IP white list.txt' is missing..."
    exit 1
else
    echolog "\t + List found,all good."
fi
while read ip4 p4 do
    echolog "\t + Allowing: $ip4 on port $p4"
    iptables -A INPUT -p tcp -s $ip4 --dport $p4 -j ACCEPT 2>&1 | tee -a $LOGFILE
    iptables -A OUTPUT -p tcp -d $ip4 --dport $p4 -j ACCEPT 2>&1 | tee -a $LOGFILE
done < "IP white list.txt"
echolog "\t - White listing of IP addresses completed.\n"

# ***************************************************************************** #
# AUTHOR  : Terence Broadbent                                                   #
# CONTRACT: Stafford University                                                 #
# Version : 1.0                                                                 #
# Details : Finally block access to everyone else.                              #
# Modified: N/A                                                                 #
 # ***************************************************************************** #

echolog "[9]. Blocking all other access to the network...\n"
iptables -A INPUT -j DROP 2>&1 | tee -a $LOGFILE
echolog "\t + Blocking: All other access."
echolog "\t - Blocking of all other access completed.\n"

# ***************************************************************************** #
# AUTHOR  : Terence Broadbent                                                   #
# CONTRACT: Stafford University                                                 #
# Version : 1.0                                                                 #
# Details : Creating a logging chain for all dropped packets.                   #
# Modified: N/A                                                                 #
# ***************************************************************************** #

echolog "[10]. Finally - creating a logging chain for dropped packets...\n"
iptables -N LOGGING 2>&1 | tee -a $LOGFILE
iptables -A INPUT -j LOGGING 2>&1 | tee -a $LOGFILE
iptables -A OUTPUT -j LOGGING 2>&1 | tee -a $LOGFILE
iptables -A LOGGING -m limit --limit 2/min --limit-burst 3 -j LOG --log-prefix "IPTables-Dropped: " -
```

```
-log-level debug 2>&1 | tee -a $LOGFILE
iptables -A LOGGING -j DROP 2>&1 | tee -a $LOGFILE
echolog "\t + Logging chain IPTables-Dropped created."
echolog "\t - logging all dropped packets completed.\n"

#***************************************************************************#
# AUTHOR  : Terence Broadbent                                               #
# CONTRACT: Stafford University                                             #
# Version : 1.0                                                             #
# Details : Save the current configuration & display the final settings to the screen. #
# Modified: N/A                                                             #
# ************************************************************************** #

echolog "[11]. Program completed sucessfully...\n"
echolog "\t - Setting up save and exit configuration."
echolog "\t   [1] Save and display this new configuration (recommended)?"
echolog "\t   [2] Save but do not display this new configuration?"
echolog "\t   [3] Exit without saving?"
while true; do
   read -p "Option:" CONT
   if [ "$CONT" = "1" ];
      then echo Option:$CONT >> $LOGFILE
      iptables-save >> /dev/null
      echolog "\t - Configuration saved.\n"
      iptables -L INPUT --line-numbers 2>&1 | tee -a $LOGFILE
      iptables -L OUTPUT --line-numbers 2>&1 | tee -a $LOGFILE
      break
  elif
      [ "$CONT" = "2" ];
      then echo Option:$CONT >> $LOGFILE
      iptables-save > /dev/null   echolog "\t - Configuration saved.\n"
      break
  elif
      [ "$CONT" = "3" ];
      then echo Option:$CONT >> $LOGFILE
      echolog "\t - Configuration not saved.\n"
      break
  else
      printf "Error please re-select "
  fi
done
echolog "\nFor any additional manual commands - all rules are kept in \etc\sysconfig\iptables.\n"
#eof
```

Figure 3 - Firewall Installation Script

The feeder files ready by this script are in Appendix A.

PENETRATION TEST SCRIPT

Once the installation has completed, a Cyber-security Professional will need to ensure that the firewall is configured and functioning as anticipated. This can again be easily achieved with the aid of a specifically crafted penetration test script.

LINUX TEST SCRIPT

```
#!/bin/sh
# ****************************************************************************** #
#                                                                                #
#                 Firewall test script based on prescribed network layout        #
#                                                                                #
#             Release version 1.0 by Terence Broadbent BSc Cyber Security        #
#                                                                                #
# ****************************************************************************** #

# ****************************************************************************** #
# AUTHOR: Terence Broadbent                                                      #
# CONTRACT: Stafford University                                                  #
# Version: 1.0                                                                   #
# Details: Define any global variables used throughout the bash script.          #
# Modified: N/A                                                                  #
# ****************************************************************************** #

NET="ens33"            # IMPORTANT! CHANGE THIS TO MATCH YOUR NETWORK.
DOMAIN="1"             # IMPORTANT! SET 1 FOR EXTERNAL AND 2 INTERNAL TEST.
IP="xxx.xxx.x.xx"      # IMPORTANT! CHANGE THIS TO MATCH THE FIREWALL IP.
LOGFILE="./Log2.txt" # The default log filename.
SEC="10"               # Increase this value to prolong the DoS flooding period.

# ****************************************************************************** #
# AUTHOR: Terence Broadbent                                                      #
# CONTRACT: Stafford University                                                  #
# Version: 1.0                                                                   #
# Details: Create a logging system & check that this bash script has root privileges. #
# Modified: N/A                                                                  #
# ****************************************************************************** #

echolog() ( echo $1 echo $1 >> $LOGFILE
if [ $USER != "root" ]
  then echo "Please run this bash script as root..."
   exit 0
else
   echolog "\n\tLINUX FIREWALL EXT/INT PENETRATION TEST LOG - VERSION 1.0\n"
fi

# ****************************************************************************** #
# AUTHOR: Terence Broadbent                                                      #
# CONTRACT: Stafford University                                                  #
# Version: 1.0                                                                   #
# Details: Comprehensive penetration test for newly installed firewall.          #
# Modified: N/A                                                                  #
# ****************************************************************************** #

echolog "[1]. Starting the Penetration test...\n"

#EXT
if [ $DOMAIN="1" ];
   then  telnet $IP 80 > ./Report1.txt 2>&1 | tee -a $LOGFILE
   echolog "\t- Penetration test one completed...\n"
fi

#EXT
if [ $DOMAIN="1" ];
  then  hping3 -V -c $SEC -S --faster --rand-source -1 $IP > ./Report2.txt 2>&1 | tee -a $LOGFILE
  echolog "\t- Penetration test two completed...\n"
fi

#INT
if [ $DOMAIN="2" ];
   then  curl -Is http://www.facebook.com | head -1 > ./Report3.txt 2>&1 | tee -a $LOGFILE
   echolog "\t- Penetration test three completed...\n"
fi

#INT
if [ $DOMAIN="2" ];
  then  ping -v -c 10 127.0.0.1 > ./Report4.txt 2>&1 | tee -a $LOGFILE
  echolog "\t- Penetration test four completed...\n"
fi

#INT
if [ $DOMAIN="2" ];
  then  ssh -v 127.0.0.1 22 > ./Report5.txt 2>&1 | tee -a $LOGFILE
  echolog "\t- Penetration test five completed...\n"
fi

#EXT
if [ $DOMAIN="1" ];
   then  nc -zv $IP 3333 > ./Report6.txt 2>&1 | tee -a $LOGFILE
   echolog "\t- Penetration test six completed...\n"
fi

#EXT
if [ $DOMAIN="1" ];
   then  nmap -v -e $NET -S 169.254.0.0 $IP -Pn > ./Report7.txt 2>&1 | tee -a $LOGFILE
   echolog "\t- Penetration test seven completed...\n"
fi

#EXT
if [ $DOMAIN="1" ];
  then  hping3 -V -c $SEC -1 -C 17 $IP > ./Report8.txt 2>&1 | tee -a $LOGFILE
  echolog "\t- Penetration test eight completed...\n"
fi

#EXT
if [ $DOMAIN="1" ];
  then  hping3 -V -c $SEC -p 80 -s 5050 -A $IP > ./Report9.txt 2>&1 | tee -a $LOGFILE
  echolog "\t- Penetration test nine completed...\n"
fi
```

```
#EXT
if [ $DOMAIN="1" ];
  then  hping3 -V -c $SEC -p 80 -s 5050 -Y $IP > ./Report10.txt 2>&1 | tee -a $LOGFILE
  echolog "\t- Penetration test ten completed...\n"
fi

#EXT
if [ $DOMAIN="1" ];
   then  hping3 -V -c 10 -S --faster --rand-source $IP > ./Report11.txt 2>&1 | tee -a $LOGFILE
   echolog "\t- Penetration test eleven completed...\n"
fi

#EXT
if [ $DOMAIN="1" ];
   then  nmap -v $IP -p 25 -Pn > ./Report12.txt 2>&1 | tee -a $LOGFILE
   nmap -v $IP -p 110 -Pn >> ./Report12.txt 2>&1 | tee -a $LOGFILE
   nmap -v $IP -p 143 -Pn >> ./Report12.txt 2>&1 | tee -a $LOGFILE
   nmap -v $IP -p 465 -Pn >> ./Report12.txt 2>&1 | tee -a $LOGFILE
   nmap -v $IP -p 993 -Pn >> ./Report12.txt 2>&1 | tee -a $LOGFILE
   nmap -v $IP -p 995 -Pn >> ./Report12.txt 2>&1 | tee -a $LOGFILE
  echolog "\t- Penetration test twelve completed...\n"
fi

#INT
if [ $DOMAIN="2" ];
   then  nmap -v lo -p 3306 -Pn > ./Report13.txt 2>&1 | tee -a $LOGFILE
   echolog "\t- Penetration test thirteen completed...\n"
fi

#EXT
if [ $DOMAIN="1" ];
   then  nmap -v -e $NET -p 7001 -S 192.168.50.21 $IP -Pn > ./Report14.txt 2>&1 | tee -a $LOGFILE
   echolog "\t- Penetration test fourteen completed...\n"
fi

#INT
if [ $DOMAIN="2" ];
  then  grep -i "UFW" /var/log/syslog > ./Report15.txt 2>&1 | tee -a $LOGFILE
  echolog "\t- Penetration test fifthteen completed...\n"
fi

#EXT
if [ $DOMAIN="1" ];
   then  mz -v -Q $NET -A rand -B $IP -t dns "q=pentesting.blog" -c $SEC > ./Report16.txt 2>&1 | tee
-a $LOGFILE
   echolog "\t- Penetration test sixteen completed...\n"
fi

#EXT
if [ $DOMAIN="1" ];
   then  python hulk.py -site $IP:80 & sleep 10 >> ./Report17.txt 2>&1 | tee -a $LOGFILE
   echolog "\t- Penetration test seventeen completed...\n"
fi

#EXT
if [ $DOMAIN="1" ];
  then  hping3 -V -c $SEC --icmp $IP > ./Report18.txt 2>&1 | tee -a $LOGFILE
```

```
   echolog "\t- Penetration test eighteen completed...\n"
fi

#EXT
if [ $DOMAIN="1" ];
  then  nmap -v -e $NET -sX $IP -Pn > ./Report19.txt 2>&1 | tee -a $LOGFILE
  echolog "\t- Penetration test nineteen completed...\n"
fi

#EXT
if [ $DOMAIN="1" ];
  then  sh -c "ping -c $SEC -s 1000 $IP > ./Report20.txt 2>&1 | tee -a $LOGFILE"
  fragroute $IP & sleep $SEC > ./Report20.txt 2>&1 | tee -a $LOGFILE
 echolog "\t- Penetration test twenty completed...\n"
fi

#EXT
if [ $DOMAIN="1" ];
   then  hping3 -V -d 120 -c $SEC -S -w 64 -p 445 -s 455 --faster --rand-source $IP >> ./Report21.txt
2>&1 | tee -a $LOGFILE
   echolog "\t- Penetration test twentyone completed...\n"
fi

echolog "\nYou have successfully performed the penetration test....\n"
#eof
```

Table 2 - Firewall Penetration Test

The report files generated by this script are in appendix B.

Technical note: You will need the following extra packages to run the script: -

- apt-get install mz
- https://github.com/grafov/hulk
- https://www.monkey.org/~dugsong/fragroute/

The script will still run without them, but you will receive error messages when the command lines tries to run the program listed above.

Run the script on the firewall platform O/S and penetration platform O/S to achieve maximum results.

EVALUATION

The trick to any evaluation is to take a snapshot of what is current, undertake the prescribed changes and then compare if the changes are for the better or worse.

VMWARE CONFIGURATION

Machine A (Mint):	xxx.xxx.x.xx - Iptables firewall on a Linux Mint system.
Machine B (Kali):	yyy.yyy.y.yy - External threat actor on a Linux Kali system.
VM Setting:	Bridged network with replicate network connection states.

Table 3 - Evaluation Settings

PRE-TEST PENETRATION SCAN

```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-03-29 15:17 BST Pre-scan script
results: | broadcast-avahi-dos: |   Discovered hosts: |     224.0.0.251 |   After NULL
UDP avahi packet DoS (CVE-2011-1002). |_  Hosts are all up (not vulnerable). Nmap
scan report for Linux-mint.home (xxx.xxx.x.xx) Host is up (0.00026s latency). All
1000 scanned ports on Linux-mint.home (xxx.xxx.x.xx) are closed MAC Address:
00:00:00:00:00:00(VMware)



Nmap done: 1 IP address (1 host up) scanned in 35.72 seconds
```

Figure 4 - Pre-test Penetration Scan

Detailed below is a list of post evaluation tasks that was identified as required to be completed in order to ensure compliance with Staffordshire's University simple firewall policy.

On the left-hand side is the task id and task description, followed by a snapshot of the Network system at the time. On the right-hand side is the post installation penetration test commands required to ensure compliance and the current (post installation) status – this table acts as a check list of achieved/outstanding tasks.

PRE/POST PEN-TEST RESULTS

ID	TASK	PRE-TEST	TEST COMMAND	POST-TEST
1.	Prevent B from accessing http (80) service	telnet: Unable to connect to remote host: Connection reused	telnet xxx.xxx.x.xx 80	telnet: Unable to connect to remote host: Connection refused
2.	Prevent B from sending ICMP flooding to A	N/A	hping -V -e 10 -S -faster-rand-source -1 xxx.xxx.x.xx	10 packets transmitted, 0 packets received, 100% packer loss
3.	Prevent A from visiting an external website.	HTTP/1.1 200n OK	Curl -Is http://www.facebook.com | head-1	HTTP/ 1.1 200 OK
4.	Accept traffic through loopback lo interface on A.	10 packets transmitted, 10 received, 0% loss	ping -v -c 10 127.0.0.1	10 packets transmitted, 10 received, 0% packet loss
5.	Prevent access to SSH from any source on A	ssh: connect to host 127.0.0.1 port 22: Connection refused	ssh -v 127.0.0.1 22	ssh: connect to host 127.0.0.1 port 22: Connection refused
6.	Reject access on port 3333 on A	Nc: connect to xxx.xxx.x.xx port 3333 (tcp) failed: Connection refused	nc -zx xxx.xxx.x.xx 3333	nc: connect to xxx.xxx.x.xx port 3333 (tcp) failed: Connection refused
7.	Reject any access coming from 169.254.0.0/16	Nmap scan report for xxx.xxx.x.xx [host down]	nmap -v -e eth0 -S 169.254.0.0 xxx.xxx.x.xx -Pn	Nmap scan report for xxx.xxx.x.xx [host down]
8.	Drop multicast IP’s	10 packets transmitted, 0 packets	hping -V -c 1- -1 -C 17 xxx.xxx.x.xx	10 packets transmitted, 0 packets

		received, 100% packet loss		received. 100% packet loss
9.	Drop incoming packets to A based on invalid combination of TCP flags	10 packets transmitted, 0 packets received, 100% packet loss	hping -V -e 10 -p 80 -s 5050 -A xxx.xxx.x.xx	10 packets transmitted, 0 packets received, 100% packet loss
10,	Drop null packets from A	10 packets transmitted, 0 packets received, 100% packet loss	hping -V -e 10 -p 80 -s 5050 -Y xxx.xxx.x.xx	10 packets transmitted, 0 packets received, 100% packet loss
11.	Prevent SYN flood from A	N/A	Hping -V -c 10 -S -faster-rand-source xxx.xxx.x.xx	10 packets transmitted, 0 packets received, 100% packet loss
12.	Allow SMTP and POP3 from A	25/tcp closed smtp 110/tcp closed pop3 143/tcp closed imap 465/tcp closed Smtps 993/tcp closed imaps 995/tcp closed pop3s	nmap -v lo 25 -Pn nmap -v lo -p 110 -Pn nmap -v lo -p 143 -Pn nmap -v lo -p 993 -Pn nmap -v lo -p 995 -Pn	25/tcp filtered smtp 110/tcp filtered pop3 143/tcp filtered imap 465/tcp filtered smtps 993/tcp filtered imaps 995/tcp filtered pop3s
13.	Allow service on port 3306 from A	3306.tcp closed mysql	nmap -v lo -p 3306 -Pn	3306/tcp filtered mysql
14.	Allow IP 192.168.50.21 to access port 7001 on A	7001/tcp closed afs3-callback	nmap -v -p 7001 xxx.xxx.x.xx	7001/tcp filtered afs3-callback

15.	Create logging chain of rejected packets from A	N/A	grep -i “UFW” /var/log/syslog	UFW entries found – see appendix C.
16.	Simulated DNS flood attack from A	N/A	mz -v -Q eth0 -A rabd -B xxx.xxx.x.xx -t dns “q=pentesting.blog” -c 10	Mausezahn will send 10 frames…
17.	Simulation of HTTP flood attack on A	N/A	python hulk.py -site xxx.xxx.x.xx:80	- HULK Attack Started…
18.	Simulation of Smurf attack on A	N/A	hping -V -e 10 –icmp xxx.xxx.x.xx	10 packets transmitted, 0 packets received, 100% packet loss
19.	Simulation of Xmas tree attack on A	N/A	nmap -v eth0 -sX xxx.xxx.x.xx	nmap scan report for xxx.xxx.x.xx [host down]
20.	Simulation of fragmented packets	N/A	fragroute xxx.xxx.x.xx & sleep 10	fragroute: tcp_seq -> ip_frag -> ip_chaff -> order -> print
21.	Simulation of LAN attack on A	N/A	hping3 -V -d 120 -c 1- -S -w 64 -p 445 -s 455 –faster --rand-source	10 packets transmitted, 0 packets received, 100% packet loss

🟩	Test undertaken, and expected results achieved.
🟧	Test no undertaken or not required.
🟥	Test failed to meet the expected results.

Table 4 - Pre/Post-test Penetration Results

Finally, the post vulnerability test.

POST-TEST PENETRATION SCAN

```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-03-29 18:31 BST Pre-scan script
results: | broadcast-avahi-dos: |   Discovered hosts: |     224.0.0.251 |   After NULL
UDP avahi packet DoS (CVE-2011-1002). |_ Hosts are all up (not vulnerable). Nmap
scan report for Linux-mint.home (xxx.xxx.x.xx) Host is up. All 1000 scanned ports
on Linux-mint.home (xxx.xxx.x.xx) are filtered


Nmap done: 1 IP address (1 host up) scanned in 238.07 seconds
```

Figure 5 - Post-test Penetration Scan

The one failed task, task 3, should be discussed here – this task failed because Iptables are primary designed for IP addresses rather than a string comparison analysis. Although you could rectify this error by looking up the website in a 'whois' lookup url and block a range of IPs with the Iptables command such as: -

```
iptables –A FORWARD –m iprange ––src–range 10.0.5.25–10.0.7.33 –j REJECT –– reject–with
icmp–host–prohibited

iptables –A INPUT –m iprange ––src–range 10.0.5.25–10.0.7.33 –j REJECT –– reject–with icmp–
host–prohibited Figure 4 - Range Blocking IP's
```

Figure 6 - Range Blocking IP's

Technical note: However, this would be sort lived as webservers have multiple IP addresses. A much smarter and more practical solution to this problem would be to list the websites in the etc/hosts file as shown below:-

```
0.0.0.0   www.example.com

0.0.0.0   example.com
```

This will block the user from accessing the specified sites permanently.

REFERENCES

Ido Dubrawsky, Wes Noonan. (2006). *Firewall Fundamentals. United States of America: Ido Dubrawsky, Wes Noonan.*

Karen Scarfone, Paul Hoffman. (2009). *Guidelines on Firewalls and Firewall Policy. United States of America: NIST.*

Scott Barman. (2001). *Writing Information Security Policies. United States of America: Sams .*

APPENDIX A

BANNED WEBSITE LIST.TXT

facebook.co.uk

facebook.com

twitter.co.uk

twitter.com

myspace.co.uk

myspace.com

linkedin.co.uk

linkedin.com

instagram.co.uk

instagram.com

IP BLACK LIST.TXT

0.0.0.0/8

10.0.0.0/8

100.64.0.0/10

127.0.0.0/8

169.254.0.0/16

172.16.0.0/12

192.0.2.0/24

192.168.0.0/16

198.18.0.0/15

198.51.100.0/24

```
203.0.113.0/24

224.0.0.0/3
```

BANNED PORT LIST.TXT

```
INPUT,tcp,0

INPUT,udp,0

OUTPUT,tcp,22

OUTPUT,udp,22

INPUT,tcp,80

INPUT,udp,80

INPUT,tcp,135

INPUT,udp,135

INPUT,tcp,136

INPUT,udp,136

INPUT,tcp,137

INPUT,udp,137

INPUT,tcp,138

INPUT,udp,138

INPUT,tcp,139

INPUT,udp,139

INPUT,tcp,445

INPUT,udp,445

INPUT,tcp,1080

INPUT,udp,1080

INPUT,tcp,3333

INPUT,udp,3333
```

IP WHITE LIST.TXT

```
0.0.0.0/0,25

0.0.0.0/0,110

0.0.0.0/0,143

0.0.0.0/0,465

0.0.0.0/0,993

0.0.0.0/0,995

0.0.0.0/0,3306

192.168.50.21,7001
```

APPENDIX B

REPORT1.TXT

```
telnet: Unable to connect to remote host: Connection refused

Trying xxx.xxx.x.xx...
```

REPORT2.TXT

```
--- xxx.xxx.x.xx hping statistic ---

10 packets transmitted, 0 packets received, 100% packet loss

round-trip min/avg/max = 0.0/0.0/0.0 ms

using ens33, addr: xxx.xxx.x.xx, MTU: 1500

HPING xxx.xxx.x.xx (ens33 xxx.xxx.x.xx): icmp mode set, 28 headers + 0 data bytes
```

REPORT3.TXT

```
HTTP/1.1 200 OK
```

REPORT4.TXT

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.

64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.022 ms

64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.026 ms

64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.026 ms

64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.028 ms

64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.030 ms

64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.030 ms

64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.028 ms

64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.028 ms
```

```
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.027 ms
64 bytes from 127.0.0.1: icmp_seq=10 ttl=64 time=0.029 ms
--- 127.0.0.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8997ms
rtt min/avg/max/mdev = 0.022/0.027/0.030/0.005 ms
```

REPORT5.TXT

```
OpenSSH_7.2p2 Ubuntu-4ubuntu2.4, OpenSSL 1.0.2g  1 Mar 2016
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 19: Applying options for *
debug1: Connecting to 127.0.0.1 [127.0.0.1] port 22.
debug1: connect to address 127.0.0.1 port 22: Connection refused
ssh: connect to host 127.0.0.1 port 22: Connection refused
```

REPORT6.TXT

```
nc: connect to xxx.xxx.x.xx port 3333 (tcp) failed: Connection refused
```

REPORT7.TXT

```
Starting Nmap 7.01 ( https://nmap.org ) at 2018-04-02 03:56 BST
Initiating ARP Ping Scan at 03:56
Scanning xxx.xxx.x.xx [1 port]
Completed ARP Ping Scan at 03:56, 0.44s elapsed (1 total hosts)
Nmap scan report for xxx.xxx.x.xx [host down]
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (0 hosts up) scanned in 0.49 seconds
           Raw packets sent: 2 (56B) | Rcvd: 0 (0B)
```

REPORT8.TXT

```
--- xxx.xxx.x.xx hping statistic ---

10 packets transmitted, 0 packets received, 100% packet loss

round-trip min/avg/max = 0.0/0.0/0.0 ms

using ens33, addr: xxx.xxx.x.xx, MTU: 1500

HPING xxx.xxx.x.xx (ens33 xxx.xxx.x.xx): icmp mode set, 28 headers + 0 data bytes
```

REPORT9.TXT

```
--- xxx.xxx.x.xx hping statistic ---

10 packets transmitted, 0 packets received, 100% packet loss

round-trip min/avg/max = 0.0/0.0/0.0 ms

using ens33, addr: xxx.xxx.x.xx, MTU: 1500

HPING xxx.xxx.x.xx (ens33 xxx.xxx.x.xx): A set, 40 headers + 0 data bytes
```

REPORT10.TXT

```
--- xxx.xxx.x.xx hping statistic ---

10 packets transmitted, 0 packets received, 100% packet loss

round-trip min/avg/max = 0.0/0.0/0.0 ms

using ens33, addr: xxx.xxx.x.xx, MTU: 1500

HPING xxx.xxx.x.xx (ens33 xxx.xxx.x.xx): Y set, 40 headers + 0 data bytes
```

REPORT11.TXT

```
10 packets transmitted, 0 packets received, 100% packet loss

round-trip min/avg/max = 0.0/0.0/0.0 ms

using ens33, addr: xxx.xxx.x.xx, MTU: 1500

HPING xxx.xxx.x.xx (ens33 xxx.xxx.x.xx): S set, 40 headers + 0 data bytes
```

REPORT12.TXT

```
Starting Nmap 7.01 ( https://nmap.org ) at 2018-04-02 03:57 BST

Initiating Parallel DNS resolution of 1 host. at 03:57

Completed Parallel DNS resolution of 1 host. at 03:57, 0.04s elapsed

Initiating SYN Stealth Scan at 03:57

Scanning lo (92.242.132.15) [1 port]

Completed SYN Stealth Scan at 03:57, 2.04s elapsed (1 total ports)

Nmap scan report for lo (92.242.132.15)

Host is up.

rDNS record for 92.242.132.15: unallocated.barefruit.co.uk

PORT   STATE    SERVICE

25/tcp filtered smtp

Read data files from: /usr/bin/../share/nmap

Nmap done: 1 IP address (1 host up) scanned in 2.17 seconds

           Raw packets sent: 2 (88B) | Rcvd: 0 (0B)

Starting Nmap 7.01 ( https://nmap.org ) at 2018-04-02 03:57 BST

Initiating Parallel DNS resolution of 1 host. at 03:57

Completed Parallel DNS resolution of 1 host. at 03:57, 0.03s elapsed

Initiating SYN Stealth Scan at 03:57

Scanning lo (92.242.132.15) [1 port]

Completed SYN Stealth Scan at 03:57, 2.04s elapsed (1 total ports)

Nmap scan report for lo (92.242.132.15)

Host is up.

rDNS record for 92.242.132.15: unallocated.barefruit.co.uk

PORT    STATE    SERVICE

110/tcp filtered pop3
```

```
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.17 seconds
           Raw packets sent: 2 (88B) | Rcvd: 0 (0B)
Starting Nmap 7.01 ( https://nmap.org ) at 2018-04-02 03:57 BST
Initiating Parallel DNS resolution of 1 host. at 03:57
Completed Parallel DNS resolution of 1 host. at 03:57, 0.05s elapsed
Initiating SYN Stealth Scan at 03:57
Scanning lo (92.242.132.15) [1 port]
Completed SYN Stealth Scan at 03:57, 2.03s elapsed (1 total ports)
Nmap scan report for lo (92.242.132.15)
Host is up.
rDNS record for 92.242.132.15: unallocated.barefruit.co.uk
PORT    STATE    SERVICE
143/tcp filtered imap
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.19 seconds
           Raw packets sent: 2 (88B) | Rcvd: 0 (0B)
Starting Nmap 7.01 ( https://nmap.org ) at 2018-04-02 11:30 BST
Initiating Parallel DNS resolution of 1 host. at 11:30
Completed Parallel DNS resolution of 1 host. at 11:30, 0.04s elapsed
Initiating SYN Stealth Scan at 11:30
Scanning lo (92.242.132.15) [1 port]
Completed SYN Stealth Scan at 11:30, 2.04s elapsed (1 total ports)
Nmap scan report for lo (92.242.132.15)
Host is up.
rDNS record for 92.242.132.15: unallocated.barefruit.co.uk
```

```
PORT    STATE    SERVICE
465/tcp filtered smtps
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.18 seconds
           Raw packets sent: 2 (88B) | Rcvd: 0 (0B)
Starting Nmap 7.01 ( https://nmap.org ) at 2018-04-02 03:57 BST
Initiating Parallel DNS resolution of 1 host. at 03:57
Completed Parallel DNS resolution of 1 host. at 03:57, 0.04s elapsed
Initiating SYN Stealth Scan at 03:57
Scanning lo (92.242.132.15) [1 port]
Completed SYN Stealth Scan at 03:57, 2.04s elapsed (1 total ports)
Nmap scan report for lo (92.242.132.15)
Host is up.
rDNS record for 92.242.132.15: unallocated.barefruit.co.uk
PORT    STATE    SERVICE
993/tcp filtered imaps
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.17 seconds
Starting Nmap 7.01 ( https://nmap.org ) at 2018-04-02 03:57 BST
Initiating Parallel DNS resolution of 1 host. at 03:57
Completed Parallel DNS resolution of 1 host. at 03:57, 0.05s elapsed
Initiating SYN Stealth Scan at 03:57
Scanning lo (92.242.132.15) [1 port]
Completed SYN Stealth Scan at 03:57, 2.04s elapsed (1 total ports)
Nmap scan report for lo (92.242.132.15)
Host is up.
```

```
rDNS record for 92.242.132.15: unallocated.barefruit.co.uk
PORT    STATE    SERVICE
995/tcp filtered pop3s
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.19 seconds
          Raw packets sent: 2 (88B) | Rcvd: 0 (0B)
```

REPORT13.TXT

```
Starting Nmap 7.01 ( https://nmap.org ) at 2018-04-02 03:57 BST
Initiating Parallel DNS resolution of 1 host. at 03:57
Completed Parallel DNS resolution of 1 host. at 03:57, 0.04s elapsed
Initiating SYN Stealth Scan at 03:57
Scanning lo (92.242.132.15) [1 port]
Completed SYN Stealth Scan at 03:57, 2.03s elapsed (1 total ports)
Nmap scan report for lo (92.242.132.15)
Host is up.
rDNS record for 92.242.132.15: unallocated.barefruit.co.uk
PORT     STATE    SERVICE
3306/tcp filtered mysql
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.25 seconds
          Raw packets sent: 2 (88B) | Rcvd: 0 (0B)
```

REPORT14.TXT

```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-04-02 10:26 BST
Initiating ARP Ping Scan at 10:26
```

```
Scanning xxx.xxx.x.xx [1 port]
Completed ARP Ping Scan at 10:26, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:26
Completed Parallel DNS resolution of 1 host. at 10:26, 0.38s elapsed
Initiating SYN Stealth Scan at 10:26
Scanning Linux-mint.home (xxx.xxx.x.xx) [1 port]
Completed SYN Stealth Scan at 10:26, 0.26s elapsed (1 total ports)
Nmap scan report for Linux-mint.home (xxx.xxx.x.xx)
Host is up (0.00014s latency).
PORT     STATE    SERVICE
7001/tcp filtered afs3-callback
MAC Address: 00:00:00:00:00:00(VMware)
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.80 seconds
           Raw packets sent: 3 (116B) | Rcvd: 1 (28B)
```

REPORT15.TXT

See appendix C.

REPORT16.TXT

[SAMPLE SNIPPET AS LONG FILE].

```
Eth: DA = ff:ff:ff:ff:ff:ff, SA = 00:0c:29:e6:0e:b5
 802.1Q VLAN-TAG = ens33
 IP:  ver=4, len=61, tos=0, id=0, frag=0, ttl=255, proto=17, sum=0, SA=106.46.50.0,
DA=xxx.xxx.x.xx,
     payload=[see next layer]
 UDP: sp=42000, dp=53, len=41, sum=0,

payload=42:42:05:00:00:01:00:00:00:00:00:00:0a:70:65:6e:74:65:73:74:69:6e:67:
04:62:6c:6f:67:00:00:01:00:01

 Eth: DA = ff:ff:ff:ff:ff:ff, SA = 00:0c:29:e6:0e:b5

 802.1Q VLAN-TAG = ens33

 IP:  ver=4, len=61, tos=0, id=0, frag=0, ttl=255, proto=17, sum=0,
SA=55.237.80.128, DA=xxx.xxx.x.xx,

      payload=[see next layer]

 UDP: sp=42000, dp=53, len=41, sum=0,

payload=42:42:05:00:00:01:00:00:00:00:00:00:0a:70:65:6e:74:65:73:74:69:6e:67:
04:62:6c:6f:67:00:00:01:00:01

0.36 seconds (27442 packets per second)
```

REPORT17.TXT

No output!

REPORT18.TXT

```
--- xxx.xxx.x.xx hping statistic ---

10 packets transmitted, 0 packets received, 100% packet loss

round-trip min/avg/max = 0.0/0.0/0.0 ms

using ens33, addr: xxx.xxx.x.xx, MTU: 1500

HPING xxx.xxx.x.xx (ens33 xxx.xxx.x.xx): icmp mode set, 28 headers + 0 data bytes
```

REPORT19.TXT

```
Starting Nmap 7.01 ( https://nmap.org ) at 2018-04-02 03:57 BST

Initiating ARP Ping Scan at 03:57

Scanning xxx.xxx.x.xx [1 port]
```

```
Completed ARP Ping Scan at 03:57, 0.44s elapsed (1 total hosts)

Nmap scan report for xxx.xxx.x.xx [host down]

Read data files from: /usr/bin/../share/nmap

Nmap done: 1 IP address (0 hosts up) scanned in 0.49 seconds

        Raw packets sent: 2 (56B) | Rcvd: 0 (0B)
```

REPORT20.TXT

```
yyy.yyy.y.yy > xxx.xxx.x.xx: icmp: type 8 code 0 (frag 46550:1480@0+)

yyy.yyy.y.yy > xxx.xxx.x.xx: (frag 46550:1480@1480+)

yyy.yyy.y.yy > xxx.xxx.x.xx: (frag 46550:1480@2960+)

yyy.yyy.y.yy > xxx.xxx.x.xx: (frag 46550:1480@4440+)

yyy.yyy.y.yy > xxx.xxx.x.xx: (frag 46550:1480@5920+)

yyy.yyy.y.yy > xxx.xxx.x.xx: (frag 46550:1480@7400+)

yyy.yyy.y.yy > xxx.xxx.x.xx: (frag 46550:1128@8880)

yyy.yyy.y.yy > xxx.xxx.x.xx: icmp: type 8 code 0 (frag 46602:1480@0+)
```

REPORT21.TXT

```
--- xxx.xxx.x.xx hping statistic ---

10 packets transmitted, 0 packets received, 100% packet loss

round-trip min/avg/max = 0.0/0.0/0.0 ms

using ens33, addr: xxx.xxx.x.xx, MTU: 1500 HPING xxx.xxx.x.xx (ens33
xxx.xxx.x.xx): S set, 40 headers + 120 data bytes
```

APPENDIX C

Linux system generated debug logs are shown below.

UFW GENERATED LOGS

UFW PENETRATION TEST LOGS

```
Apr 1 19:37:52 Linux-mint kernel: [ 413.341949] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.1 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr 1 19:39:57 Linux-mint kernel: [
538.376283] [UFW BLOCK] IN=ens33 OUT= MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00
SRC=192.168.1.254 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr
1 19:40:17 Linux-mint kernel: [ 558.785437] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:fb:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.251 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr 1 19:42:02 Linux-mint kernel: [
663.226238] [UFW BLOCK] IN=ens33 OUT= MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00
SRC=192.168.1.254 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr
1 19:44:07 Linux-mint kernel: [ 788.446501] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.1 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr 1 19:46:12 Linux-mint kernel: [
913.481654] [UFW BLOCK] IN=ens33 OUT= MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00
SRC=192.168.1.254 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr
1 19:48:17 Linux-mint kernel: [ 1038.297335] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.1 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr 1 19:49:14 Linux-mint kernel: [
1095.780495] [UFW BLOCK] IN=ens33 OUT= MAC=01:00:5e:00:00:fb:24:20:c7:62:7d:20:08:00
SRC=192.168.1.254 DST=224.0.0.251 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2
Apr 1 19:49:17 Linux-mint kernel: [ 1098.426668] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:fb:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.251 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr 1 19:50:22 Linux-mint kernel: [
1163.548935] [UFW BLOCK] IN=ens33 OUT= MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00
SRC=192.168.1.254 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr
1 19:52:27 Linux-mint kernel: [ 1288.275194] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.1 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr 1 19:54:32 Linux-mint kernel: [
1413.323302] [UFW BLOCK] IN=ens33 OUT= MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00
SRC=192.168.1.254 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr
1 19:56:37 Linux-mint kernel: [ 1538.346135] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.1 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr 1 19:58:42 Linux-mint kernel: [
1663.383410] [UFW BLOCK] IN=ens33 OUT= MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00
SRC=192.168.1.254 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr
1 20:00:47 Linux-mint kernel: [ 1788.412281] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.1 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr 1 20:02:52 Linux-mint kernel: [
1913.447657] [UFW BLOCK] IN=ens33 OUT= MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00
SRC=192.168.1.254 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr
```

```
1 20:04:57 Linux-mint kernel: [ 2038.480911] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.1 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2
Apr 1 20:07:02 Linux-mint kernel: [ 2163.515221] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.1 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr 1 20:09:07 Linux-mint kernel: [
2288.552250] [UFW BLOCK] IN=ens33 OUT= MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00
SRC=192.168.1.254 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr
1 20:11:12 Linux-mint kernel: [ 2413.597109] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.1 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr 1 20:13:01 Linux-mint kernel: [
2522.751650] [UFW BLOCK] IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:24:20:c7:62:7d:20:08:00
SRC=201.14.8.41 DST=xxx.xxx.x.xx LEN=56 TOS=0x00 PREC=0x00 TTL=126 ID=53042
PROTO=ICMP TYPE=3 CODE=3 [SRC=xxx.xxx.x.xx DST=201.14.8.41 LEN=28 TOS=0x00
PREC=0x00 TTL=63 ID=54228 PROTO=ICMP TYPE=0 CODE=0 ID=36120 SEQ=51037 ] Apr 1
20:13:01 Linux-mint kernel: [ 2522.764273] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:24:20:c7:62:7d:20:08:00 SRC=5.230.43.174 DST=xxx.xxx.x.xx LEN=56
TOS=0x00 PREC=0x00 TTL=126 ID=53127 PROTO=ICMP TYPE=3 CODE=3 [SRC=xxx.xxx.x.xx
DST=5.230.43.174 LEN=28 TOS=0x00 PREC=0x00 TTL=63 ID=53087 PROTO=ICMP TYPE=0
CODE=0 ID=36120 SEQ=44639 ] Apr 1 20:13:01 Linux-mint kernel: [ 2522.793993] [UFW BLOCK]
IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:24:20:c7:62:7d:20:08:00 SRC=192.72.153.58
DST=xxx.xxx.x.xx LEN=56 TOS=0x00 PREC=0x00 TTL=126 ID=53180 PROTO=ICMP TYPE=3
CODE=3 [SRC=xxx.xxx.x.xx DST=192.72.153.58 LEN=28 TOS=0x00 PREC=0x00 TTL=63 ID=1944
PROTO=ICMP TYPE=0 CODE=0 ID=36120 SEQ=28000 ] Apr 1 20:13:01 Linux-mint kernel: [
2522.908561] [UFW BLOCK] IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:24:20:c7:62:7d:20:08:00
SRC=196.200.9.128 DST=xxx.xxx.x.xx LEN=56 TOS=0x00 PREC=0x00 TTL=126 ID=53346
PROTO=ICMP TYPE=3 CODE=3 [SRC=xxx.xxx.x.xx DST=196.200.9.128 LEN=28 TOS=0x00
PREC=0x00 TTL=63 ID=60503 PROTO=ICMP TYPE=0 CODE=0 ID=36120 SEQ=4962 ] Apr 1
20:13:02 Linux-mint kernel: [ 2523.040525] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:24:20:c7:62:7d:20:08:00 SRC=186.221.34.26 DST=xxx.xxx.x.xx LEN=56
TOS=0x00 PREC=0x00 TTL=126 ID=53492 PROTO=ICMP TYPE=3 CODE=3 [SRC=xxx.xxx.x.xx
DST=186.221.34.26 LEN=28 TOS=0x00 PREC=0x00 TTL=63 ID=19243 PROTO=ICMP TYPE=0
CODE=0 ID=36120 SEQ=6501 ] Apr 1 20:13:02 Linux-mint kernel: [ 2523.103619] [UFW BLOCK]
IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:24:20:c7:62:7d:20:08:00 SRC=123.113.5.159
DST=xxx.xxx.x.xx LEN=56 TOS=0x00 PREC=0x00 TTL=126 ID=53538 PROTO=ICMP TYPE=3
CODE=3 [SRC=xxx.xxx.x.xx DST=123.113.5.159 LEN=28 TOS=0x00 PREC=0x00 TTL=63 ID=12648
PROTO=ICMP TYPE=0 CODE=0 ID=36120 SEQ=7526 ] Apr 1 20:13:02 Linux-mint kernel: [
2523.164394] [UFW BLOCK] IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:24:20:c7:62:7d:20:08:00
SRC=60.110.69.129 DST=xxx.xxx.x.xx LEN=56 TOS=0x00 PREC=0x00 TTL=126 ID=53642
PROTO=ICMP TYPE=3 CODE=3 [SRC=xxx.xxx.x.xx DST=60.110.69.129 LEN=28 TOS=0x00
PREC=0x00 TTL=63 ID=65086 PROTO=ICMP TYPE=0 CODE=0 ID=36120 SEQ=38502 ] Apr 1
20:13:02 Linux-mint kernel: [ 2523.164421] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:24:20:c7:62:7d:20:08:00 SRC=5.32.133.221 DST=xxx.xxx.x.xx LEN=56
TOS=0x00 PREC=0x00 TTL=126 ID=53647 PROTO=ICMP TYPE=3 CODE=3 [SRC=xxx.xxx.x.xx
DST=5.32.133.221 LEN=28 TOS=0x00 PREC=0x00 TTL=63 ID=20700 PROTO=ICMP TYPE=0
CODE=0 ID=36120 SEQ=40038 ] Apr 1 20:13:02 Linux-mint kernel: [ 2523.167742] [UFW BLOCK]
IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:24:20:c7:62:7d:20:08:00 SRC=176.205.156.174
DST=xxx.xxx.x.xx LEN=56 TOS=0x00 PREC=0x00 TTL=126 ID=53655 PROTO=ICMP TYPE=3
CODE=3 [SRC=xxx.xxx.x.xx DST=176.205.156.174 LEN=28 TOS=0x00 PREC=0x00 TTL=63
ID=42275 PROTO=ICMP TYPE=0 CODE=0 ID=36120 SEQ=37991 ] Apr 1 20:13:02 Linux-mint
kernel: [ 2523.236907] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:24:20:c7:62:7d:20:08:00 SRC=103.58.88.8 DST=xxx.xxx.x.xx LEN=56
TOS=0x00 PREC=0x00 TTL=126 ID=53695 PROTO=ICMP TYPE=3 CODE=3 [SRC=xxx.xxx.x.xx
DST=103.58.88.8 LEN=28 TOS=0x00 PREC=0x00 TTL=63 ID=37844 PROTO=ICMP TYPE=0
CODE=0 ID=36120 SEQ=49511 ] Apr 1 20:13:17 Linux-mint kernel: [ 2538.621614] [UFW BLOCK]
IN=ens33 OUT= MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254
DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Apr 1 20:13:30 Linux-
mint kernel: [ 2551.024055] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:24:20:c7:62:7d:20:08:00 SRC=4.112.232.196 DST=xxx.xxx.x.xx LEN=56
TOS=0x00 PREC=0x00 TTL=126 ID=6417 PROTO=ICMP TYPE=3 CODE=3 [SRC=xxx.xxx.x.xx
DST=4.112.232.196 LEN=28 TOS=0x00 PREC=0x00 TTL=63 ID=3753 PROTO=ICMP TYPE=0
CODE=0 ID=36120 SEQ=30704 ] Apr 1 20:14:40 Linux-mint kernel: [ 2621.065382] [UFW BLOCK]
IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:24:20:c7:62:7d:20:08:00 SRC=182.245.120.221
DST=xxx.xxx.x.xx LEN=56 TOS=0x00 PREC=0x00 TTL=126 ID=18755 PROTO=ICMP TYPE=3
CODE=3 [SRC=xxx.xxx.x.xx DST=182.245.120.221 LEN=28 TOS=0x00 PREC=0x00 TTL=63
ID=21915 PROTO=ICMP TYPE=0 CODE=0 ID=36120 SEQ=43822 ] Apr 1 20:14:40 Linux-mint
kernel: [ 2621.065388] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:24:20:c7:62:7d:20:08:00 SRC=84.120.206.132 DST=xxx.xxx.x.xx LEN=56
TOS=0x00 PREC=0x00 TTL=126 ID=18756 PROTO=ICMP TYPE=3 CODE=3 [SRC=xxx.xxx.x.xx
DST=84.120.206.132 LEN=28 TOS=0x00 PREC=0x00 TTL=63 ID=57775 PROTO=ICMP TYPE=0
CODE=0 ID=36120 SEQ=44078 ] Apr 1 20:14:40 Linux-mint kernel: [ 2621.065392] [UFW BLOCK]
IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:24:20:c7:62:7d:20:08:00 SRC=131.81.123.40
DST=xxx.xxx.x.xx LEN=56 TOS=0x00 PREC=0x00 TTL=126 ID=18757 PROTO=ICMP TYPE=3
CODE=3 [SRC=xxx.xxx.x.xx DST=131.81.123.40 LEN=28 TOS=0x00 PREC=0x00 TTL=63 ID=7009
PROTO=ICMP TYPE=0 CODE=0 ID=36120 SEQ=44334 ] Apr 1 20:14:41 Linux-mint kernel: [
2622.808095] [UFW BLOCK] IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:24:20:c7:62:7d:20:08:00
SRC=14.103.151.40 DST=xxx.xxx.x.xx LEN=56 TOS=0x00 PREC=0x00 TTL=126 ID=19322
PROTO=ICMP TYPE=3 CODE=3 [SRC=xxx.xxx.x.xx DST=14.103.151.40 LEN=28 TOS=0x00
PREC=0x00 TTL=63 ID=28077 PROTO=ICMP TYPE=0 CODE=0 ID=36120 SEQ=58649 ] Apr 1
20:15:22 Linux-mint kernel: [ 2663.346398] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.1 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Binary file /var/log/syslog matches

/…

Mar 29 15:53:21 Linux-mint kernel: [ 106.918285] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.1 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2 Mar 29 15:55:26 Linux-mint kernel: [
37.414818] [UFW BLOCK] IN=ens33 OUT= MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00
SRC=192.168.1.254 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2
Mar 29 15:56:32 Linux-mint kernel: [ 103.113266] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44
TOS=0x00 PREC=0x00 TTL=55 ID=55953 PROTO=TCP SPT=41531 DPT=8080 WINDOW=1024
RES=0x00 SYN URGP=0 Mar 29 15:56:32 Linux-mint kernel: [ 103.113305] [UFW BLOCK]
IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0
DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=49 ID=22666 PROTO=TCP SPT=41531
DPT=995 WINDOW=1024 RES=0x00 SYN URGP=0 Mar 29 15:56:32 Linux-mint kernel: [
103.113316] [UFW BLOCK] IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00
SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=47 ID=19244
PROTO=TCP SPT=41531 DPT=113 WINDOW=1024 RES=0x00 SYN URGP=0 Mar 29 15:56:32
Linux-mint kernel: [ 103.113326] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44
TOS=0x00 PREC=0x00 TTL=49 ID=29800 PROTO=TCP SPT=41531 DPT=8888 WINDOW=1024
RES=0x00 SYN URGP=0 Mar 29 15:56:32 Linux-mint kernel: [ 103.113404] [UFW BLOCK]
IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0
DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=42 ID=13231 PROTO=TCP SPT=41531
DPT=80 WINDOW=1024 RES=0x00 SYN URGP=0 Mar 29 15:56:32 Linux-mint kernel: [
103.113461] [UFW BLOCK] IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00
SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=49 ID=59214
PROTO=TCP SPT=41531 DPT=3306 WINDOW=1024 RES=0x00 SYN URGP=0 Mar 29 15:56:32
Linux-mint kernel: [ 103.113510] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44
TOS=0x00 PREC=0x00 TTL=44 ID=48230 PROTO=TCP SPT=41531 DPT=554 WINDOW=1024
RES=0x00 SYN URGP=0 Mar 29 15:56:32 Linux-mint kernel: [ 103.113558] [UFW BLOCK]
IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0
DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=51 ID=26267 PROTO=TCP SPT=41531
```

```
DPT=3389 WINDOW=1024 RES=0x00 SYN URGP=0  Mar 29 15:56:32 Linux-mint kernel: [
103.113594] [UFW BLOCK] IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00
SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=51 ID=36301
PROTO=TCP SPT=41531 DPT=1723 WINDOW=1024 RES=0x00 SYN URGP=0  Mar 29 15:56:32
Linux-mint kernel: [  103.116993] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44
TOS=0x00 PREC=0x00 TTL=58 ID=30335 PROTO=TCP SPT=41531 DPT=587 WINDOW=1024
RES=0x00 SYN URGP=0  Mar 29 15:56:32 Linux-mint kernel: [  103.117008] nf_conntrack:
automatic helper assignment is deprecated and it will be removed soon. Use the iptables CT target to
attach helpers instead. Mar 29 15:57:09 Linux-mint kernel: [  139.910522] [UFW BLOCK] IN=ens33
OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=192.168.50.21 DST=xxx.xxx.x.xx
LEN=44 TOS=0x00 PREC=0x00 TTL=45 ID=8140 PROTO=TCP SPT=49729 DPT=7001
WINDOW=1024 RES=0x00 SYN
URGP=0  Mar 29 15:57:31 Linux-mint kernel: [  162.419142] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.1 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2  Mar 29 15:59:36 Linux-mint kernel: [
287.423591] [UFW BLOCK] IN=ens33 OUT= MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00
SRC=192.168.1.254 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2

/…

Mar 29 15:53:21 Linux-mint kernel: [  106.918285] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.1 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2  Mar 29 15:55:26 Linux-mint kernel: [
37.414818] [UFW BLOCK] IN=ens33 OUT= MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00
SRC=192.168.1.254 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2
Mar 29 15:56:32 Linux-mint kernel: [  103.113266] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44
TOS=0x00 PREC=0x00 TTL=55 ID=55953 PROTO=TCP SPT=41531 DPT=8080 WINDOW=1024
RES=0x00 SYN URGP=0  Mar 29 15:56:32 Linux-mint kernel: [  103.113305] [UFW BLOCK]
IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0
DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=49 ID=22666 PROTO=TCP SPT=41531
DPT=995 WINDOW=1024 RES=0x00 SYN URGP=0  Mar 29 15:56:32 Linux-mint kernel: [
103.113316] [UFW BLOCK] IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00
SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=47 ID=19244
PROTO=TCP SPT=41531 DPT=113 WINDOW=1024 RES=0x00 SYN URGP=0  Mar 29 15:56:32
Linux-mint kernel: [  103.113326] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44
TOS=0x00 PREC=0x00 TTL=49 ID=29800 PROTO=TCP SPT=41531 DPT=8888 WINDOW=1024
RES=0x00 SYN URGP=0  Mar 29 15:56:32 Linux-mint kernel: [  103.113404] [UFW BLOCK]
IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0
DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=42 ID=13231 PROTO=TCP SPT=41531
DPT=80 WINDOW=1024 RES=0x00 SYN URGP=0  Mar 29 15:56:32 Linux-mint kernel: [
103.113461] [UFW BLOCK] IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00
SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=49 ID=59214
PROTO=TCP SPT=41531 DPT=3306 WINDOW=1024 RES=0x00 SYN URGP=0  Mar 29 15:56:32
Linux-mint kernel: [  103.113510] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44
TOS=0x00 PREC=0x00 TTL=44 ID=48230 PROTO=TCP SPT=41531 DPT=554 WINDOW=1024
RES=0x00 SYN URGP=0  Mar 29 15:56:32 Linux-mint kernel: [  103.113558] [UFW BLOCK]
IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0
DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=51 ID=26267 PROTO=TCP SPT=41531
DPT=3389 WINDOW=1024 RES=0x00 SYN URGP=0  Mar 29 15:56:32 Linux-mint kernel: [
103.113594] [UFW BLOCK] IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00
SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=51 ID=36301
PROTO=TCP SPT=41531 DPT=1723 WINDOW=1024 RES=0x00 SYN URGP=0  Mar 29 15:56:32
Linux-mint kernel: [  103.116993] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44
```

```
TOS=0x00 PREC=0x00 TTL=58 ID=30335 PROTO=TCP SPT=41531 DPT=587 WINDOW=1024
RES=0x00 SYN URGP=0  Mar 29 15:57:09 Linux-mint kernel: [ 139.910522] [UFW BLOCK]
IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=192.168.50.21
DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=45 ID=8140 PROTO=TCP SPT=49729
DPT=7001 WINDOW=1024 RES=0x00 SYN URGP=0  Mar 29 15:57:31 Linux-mint kernel: [
162.419142] [UFW BLOCK] IN=ens33 OUT= MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00
SRC=192.168.1.254 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2
Mar 29 15:59:36 Linux-mint kernel: [ 287.423591] [UFW BLOCK] IN=ens33 OUT=
MAC=01:00:5e:00:00:01:24:20:c7:62:7d:20:08:00 SRC=192.168.1.254 DST=224.0.0.1 LEN=36
TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2

/...

Mar 29 15:56:32 Linux-mint kernel: [ 103.113266] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44
TOS=0x00 PREC=0x00 TTL=55 ID=55953 PROTO=TCP SPT=41531 DPT=8080 WINDOW=1024
RES=0x00 SYN URGP=0
Mar 29 15:56:32 Linux-mint kernel: [ 103.113305] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44
TOS=0x00 PREC=0x00 TTL=49 ID=22666 PROTO=TCP SPT=41531 DPT=995 WINDOW=1024
RES=0x00 SYN URGP=0  Mar 29 15:56:32 Linux-mint kernel: [ 103.113316] [UFW BLOCK]
IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0
DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=47 ID=19244 PROTO=TCP SPT=41531
DPT=113 WINDOW=1024 RES=0x00 SYN URGP=0  Mar 29 15:56:32 Linux-mint kernel: [
103.113326] [UFW BLOCK] IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00
SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=49 ID=29800
PROTO=TCP SPT=41531 DPT=8888 WINDOW=1024 RES=0x00 SYN URGP=0  Mar 29 15:56:32
Linux-mint kernel: [ 103.113404] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44
TOS=0x00 PREC=0x00 TTL=42 ID=13231 PROTO=TCP SPT=41531 DPT=80 WINDOW=1024
RES=0x00 SYN URGP=0  Mar 29 15:56:32 Linux-mint kernel: [ 103.113461] [UFW BLOCK]
IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0
DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=49 ID=59214 PROTO=TCP SPT=41531
DPT=3306 WINDOW=1024 RES=0x00 SYN URGP=0  Mar 29 15:56:32 Linux-mint kernel: [
103.113510] [UFW BLOCK] IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00
SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=44 ID=48230
PROTO=TCP SPT=41531 DPT=554 WINDOW=1024 RES=0x00 SYN URGP=0  Mar 29 15:56:32
Linux-mint kernel: [ 103.113558] [UFW BLOCK] IN=ens33 OUT=
MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44
TOS=0x00 PREC=0x00 TTL=51 ID=26267 PROTO=TCP SPT=41531 DPT=3389 WINDOW=1024
RES=0x00 SYN URGP=0  Mar 29 15:56:32 Linux-mint kernel: [ 103.113594] [UFW BLOCK]
IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00 SRC=169.254.0.0
DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=51 ID=36301 PROTO=TCP SPT=41531
DPT=1723 WINDOW=1024 RES=0x00 SYN URGP=0  Mar 29 15:56:32 Linux-mint kernel: [
103.116993] [UFW BLOCK] IN=ens33 OUT= MAC=00:0c:29:e6:0e:b5:00:0c:29:60:ae:89:08:00
SRC=169.254.0.0 DST=xxx.xxx.x.xx LEN=44 TOS=0x00 PREC=0x00 TTL=58 ID=30335
PROTO=TCP SPT=41531 DPT=587 WINDOW=1024 RES=0x00 SYN URGP=0
```