LAB 8. Phân tích gói tin WireShark nâng cao và tạo rule Firewall

I. Yêu cầu:

Wireshark là 1 công cụ phân tích mạng mạnh, có lợi thế trong việc bắt các lưu lượng từ xa để tạo các rules firewall dựa trên thông tin các gói tin bắt được.

Chuẩn bị

- Bài LAB sử dụng 1 máy Windows 10
- Bảo đảm đường truyền internet đã thông.
- Đổi password của Administrator máy Windows 10 là abc@123
- Cài đặt sẵn phần mềm Wireshark

II. Triển khai

1. **Phân giải tên mạng**: có thể hiển thị tên domain thay cho địa chỉ ip khi phân tích
   Edit > Preferences > Name Resolution > Enable Network Name Resolution
   Wireshark -i nicname -k
2. **Bắt lưu lượng từ các máy tính ở xa**
   Để bắt gói từ một router, server hay một máy tính khác trong mạng, cần khởi tạo WinPcap trên hệ thống ở xa. Nếu đã cài đặt wireshark, WinPcap đã được bao gồm,
   Gõ lệnh: services.msc
   Start dịch vụ Remote Packet Capture Protocol
   Trong cửa sổ Wireshark Capture Options, Interface: chọn Remote, Host: nhập địa chỉ hệ thống ở xa. Port: nhập 2002, chọn null authentication. Sau khi kết nối, có thể chọn card mạng khác trên máy ở xa trong hộp danh sách xổ xuống. Click Start để bắt đầu bắt gói.
3. **Lệnh Tshark:** là tiện ích phân tích giao thức mạng, sử dụng trong terminal. Có thể bắt các gói dữ liệu từ mạng đang online hay đọc các gói đã bắt lưu trong tập tin (định dạng pcapng)

Cú pháp:

tShark: bắt lưu lượng từ card mạng đầu tiên và hiển thị thông tin tóm tắt cho mỗi gói, tương đương tcpdump

tShark –D: liệt kê danh sách các network interface
tShark –i <capture interface #>: bắt đầu tiến trình tự bắt gói ở card mạng # (ctrl+C để stop)
tShark –i <capture interface #> -w xuanlam: ghi thông tin bắt gói từ card mạng # ra tập tin có tên xuanlam
tShark –r <xuanlam>: đọc gói từ file có tên xuanlam
tShark –c <n>: dừng sau n gói
tShark –d <layer type>==<selector>, <decode as protocol>: xác định gói được decode như protocol

Các tùy chọn kết hợp

-V: liệt kê thông tin chi tiết từng gói, tất cả các field của tất cả protocol trong gói, gồm <frame number>

-O: chỉ show chi tiết đầy đủ của protocol được xác định và show tóm tắt thông tin các protocol khác.

-P: hiển thị thông tin tóm tắt của gói

- **Lọc gói trong tShark và Wireshark**

tShark -f <capture filter>: lọc gói theo cú pháp lọc trong thư viện pcap

tshark [other options] [-Y "display filter expression" | --display-filter "display filter expression"]

Các toán tử lọc

eq, ==	So sánh bằng
ne, !=	So sánh khác
gt, >	Lớn hơn
lt, <	Nhỏ hơn
ge, >=	Lớn hơn hoặc bằng
le, <=	Nhỏ hơn hoặc bằng

eq, any_eq, ==	Các field bất kỳ phải bằng
ne, all_ne, !=	Tất cả field phải không bằng
all_eq, ===	Tất cả field phải bằng
any_ne, !==	Các field bất kỳ phải không bằng

WIRESHARK DISPLAY FILTERS · PART 1 packetlife.net

Ethernet		
eth.addr	eth.len	eth.src
eth.dst	eth.lg	eth.trailer
eth.ig	eth.multicast	eth.type

IEEE 802.1Q		
vlan.cfi	vlan.id	vlan.priority
vlan.etype	vlan.len	vlan.trailer

IPv4	
ip.addr	ip.fragment.overlap.conflict
ip.checksum	ip.fragment.toolongfragment
ip.checksum_bad	ip.fragments
ip.checksum_good	ip.hdr_len
ip.dsfield	ip.host
ip.dsfield.ce	ip.id
ip.dsfield.dscp	ip.len
ip.dsfield.ect	ip.proto
ip.dst	ip.reassembled_in
ip.dst_host	ip.src
ip.flags	ip.src_host
ip.flags.df	ip.tos
ip.flags.mf	ip.tos.cost
ip.flags.rb	ip.tos.delay
ip.frag_offset	ip.tos.precedence
ip.fragment	ip.tos.reliability
ip.fragment.error	ip.tos.throughput
ip.fragment.multipletails	ip.ttl

ARP	
arp.dst.hw_mac	arp.proto.size
arp.dst.proto_ipv4	arp.proto.type
arp.hw.size	arp.src.hw_mac
arp.hw.type	arp.src.proto_ipv4
arp.opcode	

TCP	
tcp.ack	tcp.options.qs
tcp.checksum	tcp.options.sack
tcp.checksum_bad	tcp.options.sack_le
tcp.checksum_good	tcp.options.sack_perm
tcp.continuation_to	tcp.options.sack_re
tcp.dstport	tcp.options.time_stamp
tcp.flags	tcp.options.wscale
tcp.flags.ack	tcp.options.wscale_val
tcp.flags.cwr	tcp.pdu.last_frame
tcp.flags.ecn	tcp.pdu.size
tcp.flags.fin	tcp.pdu.time
tcp.flags.push	tcp.port
tcp.flags.reset	tcp.reassembled_in
tcp.flags.syn	tcp.segment
tcp.flags.urg	tcp.segment.error
tcp.hdr_len	tcp.segment.multipletails
tcp.len	tcp.segment.overlap
tcp.nxtseq	tcp.segment.overlap.conflict
tcp.options	tcp.segment.toolongfragment

ip.fragment.overlap	ip.version

IPv6	
ipv6.addr	ipv6.hop_opt
ipv6.class	ipv6.host
ipv6.dst	ipv6.mipv6_home_address
ipv6.dst_host	ipv6.mipv6_length
ipv6.dst_opt	ipv6.mipv6_type
ipv6.flow	ipv6.nxt
ipv6.fragment	ipv6.opt.pad1
ipv6.fragment.error	ipv6.opt.padn
ipv6.fragment.more	ipv6.plen
ipv6.fragment.multipletails	ipv6.reassembled_in
ipv6.fragment.offset	ipv6.routing_hdr
ipv6.fragment.overlap	ipv6.routing_hdr.addr
ipv6.fragment.overlap.conflict	ipv6.routing_hdr.left
ipv6.fragment.toolongfragment	ipv6.routing_hdr.type
ipv6.fragments	ipv6.src
ipv6.fragment.id	ipv6.src_host
ipv6.hlim	ipv6.version

tcp.options.cc	tcp.segments
tcp.options.ccecho	tcp.seq
tcp.options.ccnew	tcp.srcport
tcp.options.echo	tcp.time_delta
tcp.options.echo_reply	tcp.time_relative
tcp.options.md5	tcp.urgent_pointer
tcp.options.mss	tcp.window_size
tcp.options.mss_val	

UDP		
udp.checksum	udp.dstport	udp.srcport
udp.checksum_bad	udp.length	
udp.checksum_good	udp.port	

Operators
eq or ==
ne or !=
gt or >
lt or <
ge or >=
le or <=

Logic	
and or &&	Logical AND
or or ||	Logical OR
xor or ^^	Logical XOR
not or !	Logical NOT
[n] […]	Substring operator

WIRESHARK DISPLAY FILTERS · PART 2

packetlife.net

Frame Relay	
fr.becn	fr.de
fr.chdlctype	fr.dlci
fr.control	fr.dlcore_control
fr.control.f	fr.ea
fr.control.ftype	fr.fecn
fr.control.n_r	fr.lower_dlci
fr.control.n_s	fr.nlpid
fr.control.p	fr.second_dlci
fr.control.s_ftype	fr.snap.oui
fr.control.u_modifier_cmd	fr.snap.pid
fr.control.u_modifier_resp	fr.snaptype
fr.cr	fr.third_dlci
fr.dc	fr.upper_dlci

PPP	
ppp.address	ppp.direction
ppp.control	ppp.protocol

ICMPv6	
icmpv6.all_comp	icmpv6.option.name_type.fqdn
icmpv6.checksum	icmpv6.option.name_x501
icmpv6.checksum_bad	icmpv6.option.rsa.key_hash
icmpv6.code	icmpv6.option.type
icmpv6.comp	icmpv6.ra.cur_hop_limit
icmpv6.haad.ha_addrs	icmpv6.ra.reachable_time
icmpv6.identifier	icmpv6.ra.retrans_timer
icmpv6.option	icmpv6.ra.router_lifetime
icmpv6.option.cga	icmpv6.recursive_dns_serv
icmpv6.option.length	icmpv6.type
icmpv6.option.name_type	

RIP		
rip.auth.passwd	rip.ip	rip.route_tag
rip.auth.type	rip.metric	rip.routing_domain
rip.command	rip.netmask	rip.version
rip.family	rip.next_hop	

MPLS	
mpls.bottom	mpls.oam.defect_location
mpls.cw.control	mpls.oam.defect_type
mpls.cw.res	mpls.oam.frequency
mpls.exp	mpls.oam.function_type
mpls.label	mpls.oam.ttsi
mpls.oam.bip16	mpls.ttl

ICMP		
icmp.checksum	icmp.ident	icmp.seq
icmp.checksum_bad	icmp.mtu	icmp.type
icmp.code	icmp.redir_gw	

DTP		
dtp.neighbor	dtp.tlv_type	vtp.neighbor
dtp.tlv_len	dtp.version	

VTP	
vtp.code	vtp.vlan_info.802_10_index
vtp.conf_rev_num	vtp.vlan_info.isl_vlan_id
vtp.followers	vtp.vlan_info.len
vtp.md	vtp.vlan_info.mtu_size
vtp.md5_digest	vtp.vlan_info.status.vlan_susp
vtp.md_len	vtp.vlan_info.tlv_len
vtp.seq_num	vtp.vlan_info.tlv_type
vtp.start_value	vtp.vlan_info.vlan_name
vtp.upd_id	vtp.vlan_info.vlan_name_len
vtp.upd_ts	vtp.vlan_info.vlan_type
vtp.version	

BGP	
bgp.aggregator_as	bgp.mp_reach_nlri_ipv4_prefix
bgp.aggregator_origin	bgp.mp_unreach_nlri_ipv4_prefix
bgp.as_path	bgp.multi_exit_disc
bgp.cluster_identifier	bgp.next_hop
bgp.cluster_list	bgp.nlri_prefix
bgp.community_as	bgp.origin
bgp.community_value	bgp.originator_id
bgp.local_pref	bgp.type
bgp.mp_nlri_tnl_id	bgp.withdrawn_prefix

HTTP	
http.accept	http.proxy_authorization
http.accept_encoding	http.proxy_connect_host
http.accept_language	http.proxy_connect_port
http.authbasic	http.referer
http.authorization	http.request
http.cache_control	http.request.method
http.connection	http.request.uri
http.content_encoding	http.request.version
http.content_length	http.response
http.content_type	http.response.code
http.cookie	http.server
http.date	http.set_cookie
http.host	http.transfer_encoding
http.last_modified	http.user_agent
http.location	http.www_authenticate
http.notification	http.x_forwarded_for
http.proxy_authenticate	

4. **Tạo các rules ACL Firewall**

   Từ thông tin các gói được bắt, có thể block một số lưu lượng đáng nghi.

   Wireshark firewall acl rules sẽ phát sinh các lệnh cần để tạo các rule trong firewall

   Click chọn gói cần phát sinh rule firewall

   Tools > Firewall ACL rules

   Product: chọn loại firewall

   - netfilter (iptables)
   - ipfilter
   - ip firewall (ipfw)
   - windows firewall (netsh)
   - packet filter (pf)
   - cisco ios (standard, extended)

Filter: tạo rule dựa trên địa chỉ MAC, địa chỉ IP hay Port

76.73.3.245 + TCP port 6969

Mặc định tool sẽ tạo 1 rule cấm inbound traffic, có thể sửa đổi tác dụng của rule bằng cách bỏ check hộp kiểm inbound hay hộp kiểm deny. Sử dụng nút copy để copy nó để chạy trên firewall tương ứng.

III.Bài tập

1. Decode các gói tcp trên port 8888 như http
   tshark -d tcp.port==8888,http
2. Decode các gói tcp, trên các port 8888,8889,8890 như http
   tshark -d tcp.port==8888:3,http
   tshark -d tcp.port==8888-8890,http
3. Thực hiện lọc các gói tin thỏa các điều kiện sau

```
all tcp.port > 1024
any ip.addr != 1.1.1.1
http contains https://www.wireshark.org
frame.len > 10
eth.dst eq ff:ff:ff:ff:ff:ff
ip.src == 192.168.1.1
ip.dst eq huflit.edu.vn
ip.addr == 129.111.0.0/16
http.request.method == "POST"
http.request.method == "\x48EAD"
smb.path contains "\\\\SERVER\\SHARE"
http.content_type[0:4] == "text"
tcp.port in {80,443,8080}
tcp.port == 80 or tcp.port == 443 or tcp.port == 8080
http.request.method in {"HEAD", "GET"}
tcp.port in {443, 4430..4434}
ip.addr in {10.0.0.5 .. 10.0.0.9, 192.168.1.1..192.168.1.9}
frame.time_delta in {10 .. 10.5}
tcp.payload contains "GET"
tcp.payload contains 47.45.54
udp.dstport >= udp.srcport + 1
tcp.dstport >= 4 * {tcp.srcport + 3}
tcp.port == 80 and ip.src == 192.168.2.1
```

4. Tìm các gói tin syn ack và ack tương ứng của một gói syn xác định
5. Truy xuất trang web huflit.edu.vn. Tìm lọc các gói tin liên quan.
6. Truy xuất trang web https://www.wireshark.org. Tìm lọc các gói tin liên quan.
7. Phát sinh lệnh netsh tạo rule để chặn truy xuất trang https://www.wireshark.org từ các gói tin lọc được ở câu 6.

Wireshark Cheat Sheet

comparitech

Default columns in a packet capture output

No.	Frame number from the beginning of the packet capture
Time	Seconds from the first frame
Source (src)	Source address, commonly an IPv4, IPv6 or Ethernet address
Destination (dst)	Destination address
Protocol	Protocol used in the Ethernet frame, IP packet, or TCP segment
Length	Length of the frame in bytes

Logical Operators

Operator	Description	Example
and or &&	Logical AND	All the conditions should match
or or ||	Logical OR	Either all or one of the condition should match
xor or ^^	Logical XOR	exclusive alternation - Only one of the two conditions should match not both
not or !	NOT(Negation)	Not equal to
[n] [-:]	Substring operator	Filter a specific word or text

Filtering packets (Display Filters)

Operator	Description	Example
eq or ==	Equal	ip.dest == 192.168.1.1
ne or !=	Not Equal	ip.dest != 192.168.1.1
gt or >	Greater than	frame.len > 10
lt or <	Less than	frame.len <10
ge or >=	Greater than or Equal	frame.len >= 10
le or <=	Less than or Equal	frame.len<=10

Filter Types

Capture Filter	Filter packets during capture
Display Filter	Hide Packets from a capture display

Wireshark Capturing Modes

Promiscuous mode	Sets interface to capture all packets on a network segment to which it is associated to
Monitor mode	Setup the Wireless interface to capture all traffic it can receive (Unix/Linux only)

Miscellaneous

Slice Operator	[...] - Range of values
Membership Operator	{} - In
CTRL+E	Start/Stop Capturing

Capture Filter Syntax

Syntax	protocol	direction	hosts	value	Logical operator	Expressions
Example	tcp	src	192.168.1.1	80	and	tcp dst 192.168.30.1

Display Filter Syntax

Syntax	protocol	String 1	String 2	Comparison Operator	value	logical operator	Expressions
Example	http	dest	ip	==	192.168.1.1	and	tcp port

Keyboard Shortcuts - main display window

Accelerator	Description	Accelerator	Description
Tab or Shift+Tab	Move between screen elements, e.g. from the toolbars to the packet list to the packet detail.	Alt+→ or Option+→	Move to the next packet in the selection history.
↓	Move to the next packet or detail item.	→	In the packet detail, opens the selected tree item.
↑	Move to the previous packet or detail item.	Shift+→	In the packet detail, opens the selected tree item and all of its subtrees.
Ctrl+ ↓ or F8	Move to the next packet, even if the packet list isn't focused.	Ctrl+→	In the packet detail, opens all tree items.
Ctrl+ ↑ or F7	Move to the previous packet, even if the packet list isn't focused.	Ctrl+←	In the packet detail, closes all tree items.
Ctrl+.	Move to the next packet of the conversation (TCP, UDP or IP).	Backspace	In the packet detail, jumps to the parent node.
Ctrl+,	Move to the previous packet of the conversation (TCP, UDP or IP).	Return or Enter	In the packet detail, toggles the selected tree item.

Protocols - Values

ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp and udp

Common Filtering commands

Usage	Filter syntax	Usage	Filter syntax
Wireshark Filter by IP	ip.addr == 10.10.50.1	Filter by URL	http.host == "host name"
Filter by Destination IP	ip.dst == 10.10.50.1	Filter by time stamp	frame.time >= "June 02, 2019 18:04:00"
Filter by Source IP	ip.src == 10.10.50.1	Filter SYN flag	tcp.flags.syn == 1
Filter by IP range	ip.addr >= 10.10.50.1 and ip.addr <= 10.10.50.100		tcp.flags.syn == 1 and tcp.flags.ack == 0
Filter by Multiple Ips	ip.addr == 10.10.50.1 and ip.addr == 10.10.50.100	Wireshark Beacon Filter	wlan.fc.type_subtype = 0x08
Filter out IP address	!(ip.addr == 10.10.50.1)	Wireshark broadcast filter	eth.dst == ff:ff:ff:ff:ff:ff
Filter subnet	ip.addr == 10.10.50.1/24	Wireshark multicast filter	[eth.dst[0] & 1)
Filter by port	tcp.port == 25	Host name filter	ip.host = hostname
Filter by destination port	tcp.dstport == 23	MAC address filter	eth.addr == 00:70:f4:23:18:c4
Filter by ip address and port	ip.addr == 10.10.50.1 and tcp.port == 25	RST flag filter	tcp.flags.reset == 1

Main toolbar items

Toolbar Icon	Toolbar Item	Menu Item	Description	Toolbar Icon	Toolbar Item	Menu Item	Description
	Start	Capture → Start	Uses the same packet capturing options as the previous session, or uses defaults if no options were set		Go Forward	Go → Go Forward	Jump forward in the packet history
	Stop	Capture → Stop	Stops currently active capture		Go to Packet...	Go → Go to Packet...	Go to specific packet
	Restart	Capture → Restart	Restarts active capture session		Go To First Packet	Go → First Packet	Jump to first packet of the capture file
	Options...	Capture → Options...	Opens "Capture Options" dialog box		Go To Last Packet	Go → Last Packet	Jump to last packet of the capture file
	Open...	File → Open...	Opens "File open" dialog box to load a capture for viewing		Auto Scroll in Live Capture	View → Auto Scroll in Live Capture	Auto scroll packet list during live capture
	Save As...	File → Save As...	Save current capture file		Colorize	View → Colorize	Colorize the packet list (or not)
	Close	File → Close	Close current capture file		Zoom In	View → Zoom In	Zoom into the packet data (increase the font size)
	Reload	View → Reload	Reloads current capture file		Zoom Out	View → Zoom Out	Zoom out of the packet data (decrease the font size)
	Find Packet...	Edit → Find Packet...	Find packet based on different criteria		Normal Size	View → Normal Size	Set zoom level back to 100%
	Go Back	Go → Go Back	Jump back in the packet history		Resize Columns	View → Resize Columns	Resize columns, so the content fits to the width