Off-Path Network Traffic Manipulation via Revitalized ICMP Redirect Attacks

Xuewei Feng[1)], Qi Li[1,4)], **Kun Sun**[2)], Zhiyun Qian[3)], Gang Zhao[1)], Xiaohui Kuang[5)], Chuanpu Fu[1)], and Ke Xu[1,4)]

1) *Tsinghua University & BNRist*
2) *George Mason University*
3) *UC Riverside*
4) *Zhongguancun Lab*
5) *Beijing University of Posts and Telecommunications*

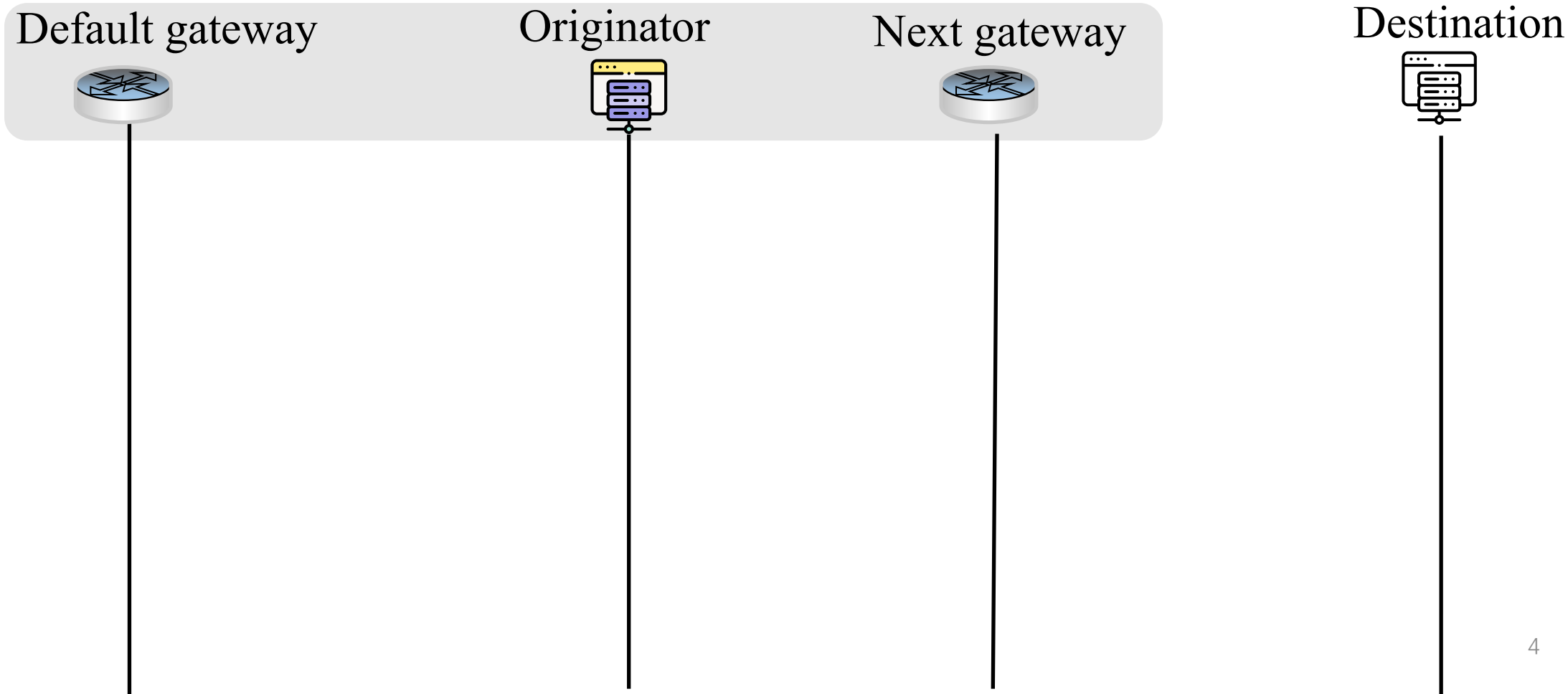
Overview

- The Mechanism of ICMP Redirect
- Legitimacy Check over ICMP Errors
- Stealthy Remote DoS Attacks
- Network Traffic Hijacking Attacks
- Countermeasures
- Conclusion

The Mechanism of ICMP Redirect (RFC 792)

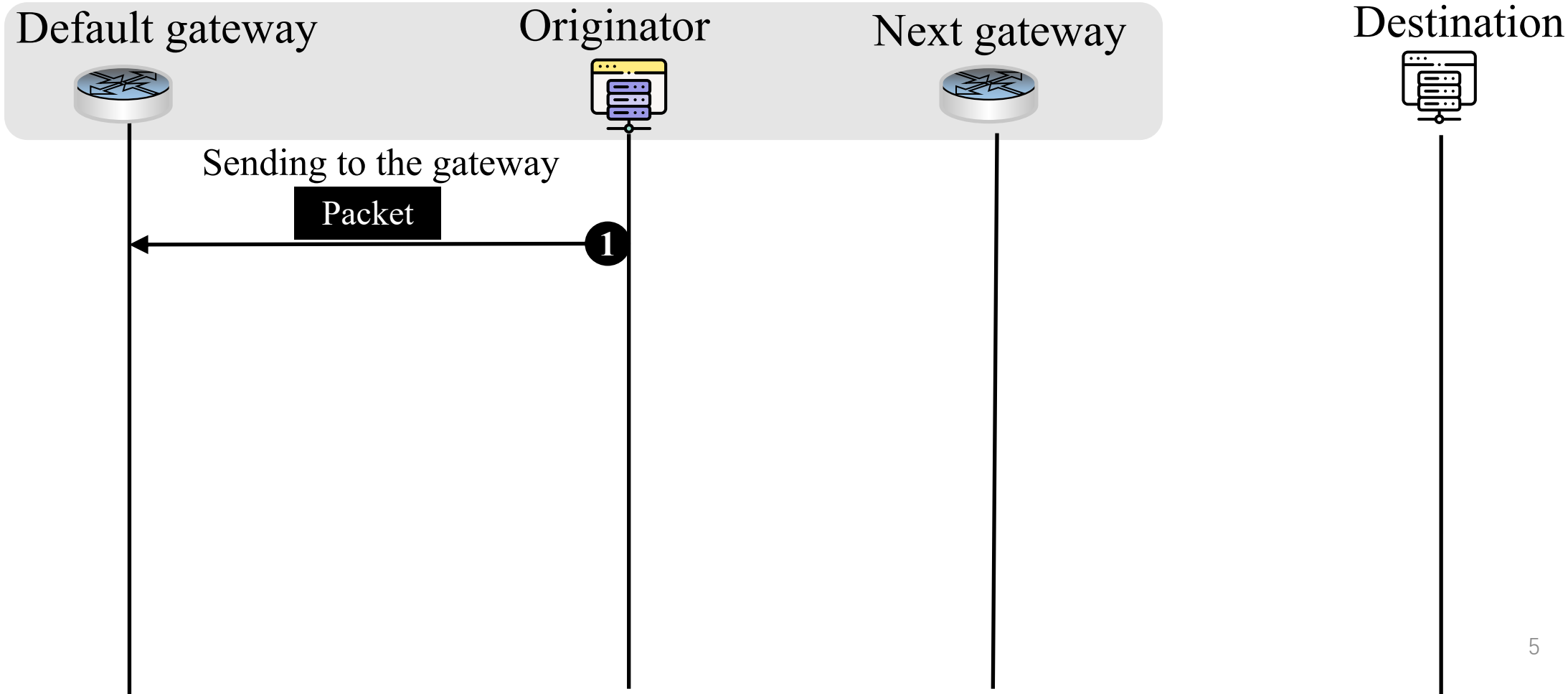
The Mechanism of ICMP Redirect

ICMP redirects allow the originator to dynamically update its routing, thus optimizing the forwarding path.



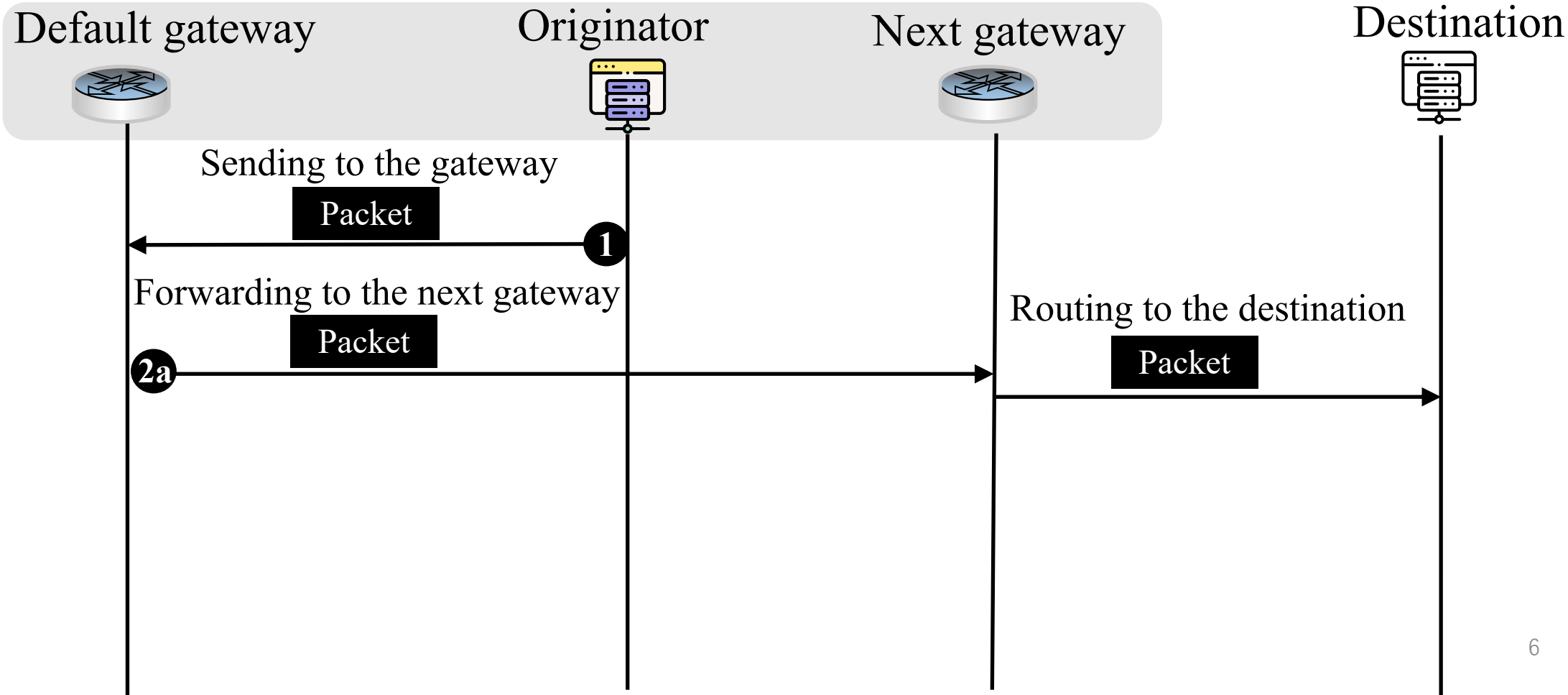
The Mechanism of ICMP Redirect

ICMP redirects allow the originator to dynamically update its routing, thus optimizing the forwarding path.



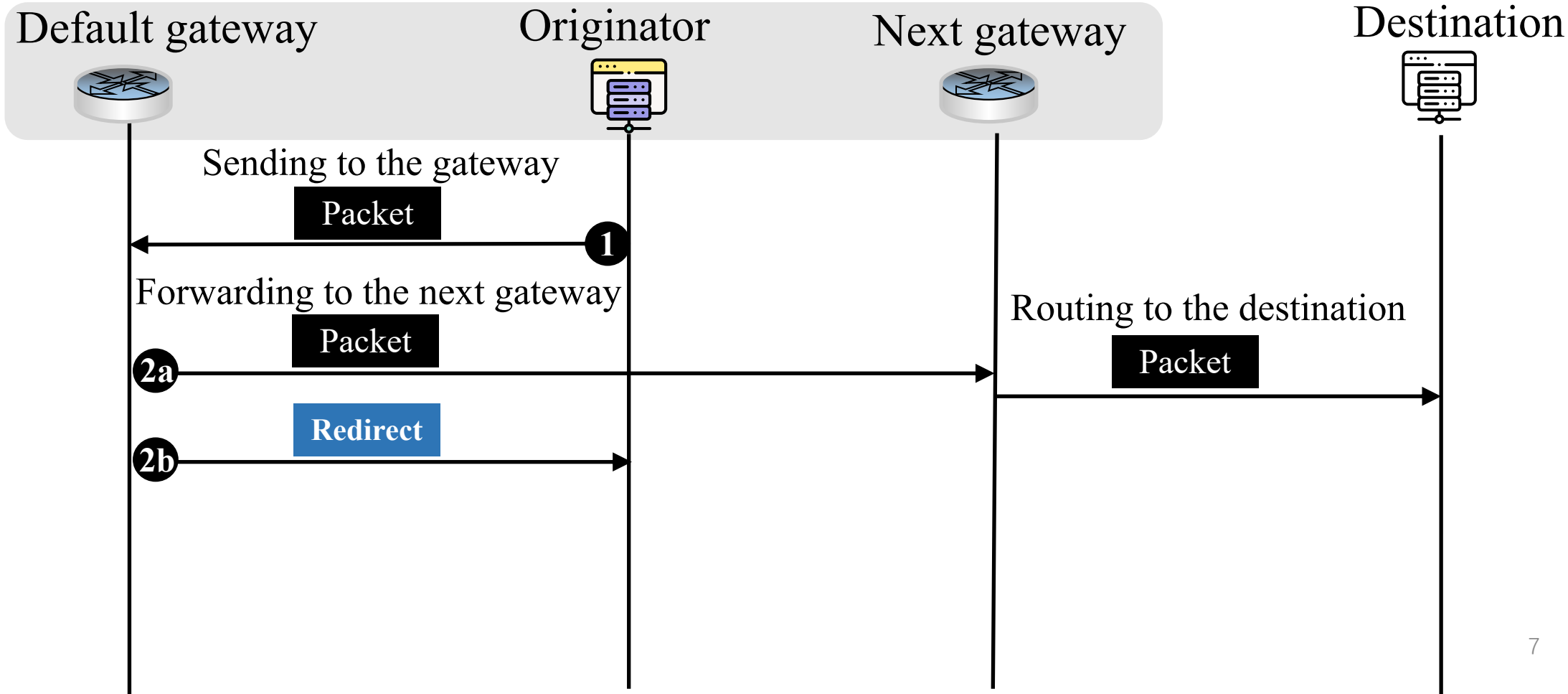
The Mechanism of ICMP Redirect

ICMP redirects allow the originator to dynamically update its routing, thus optimizing the forwarding path.



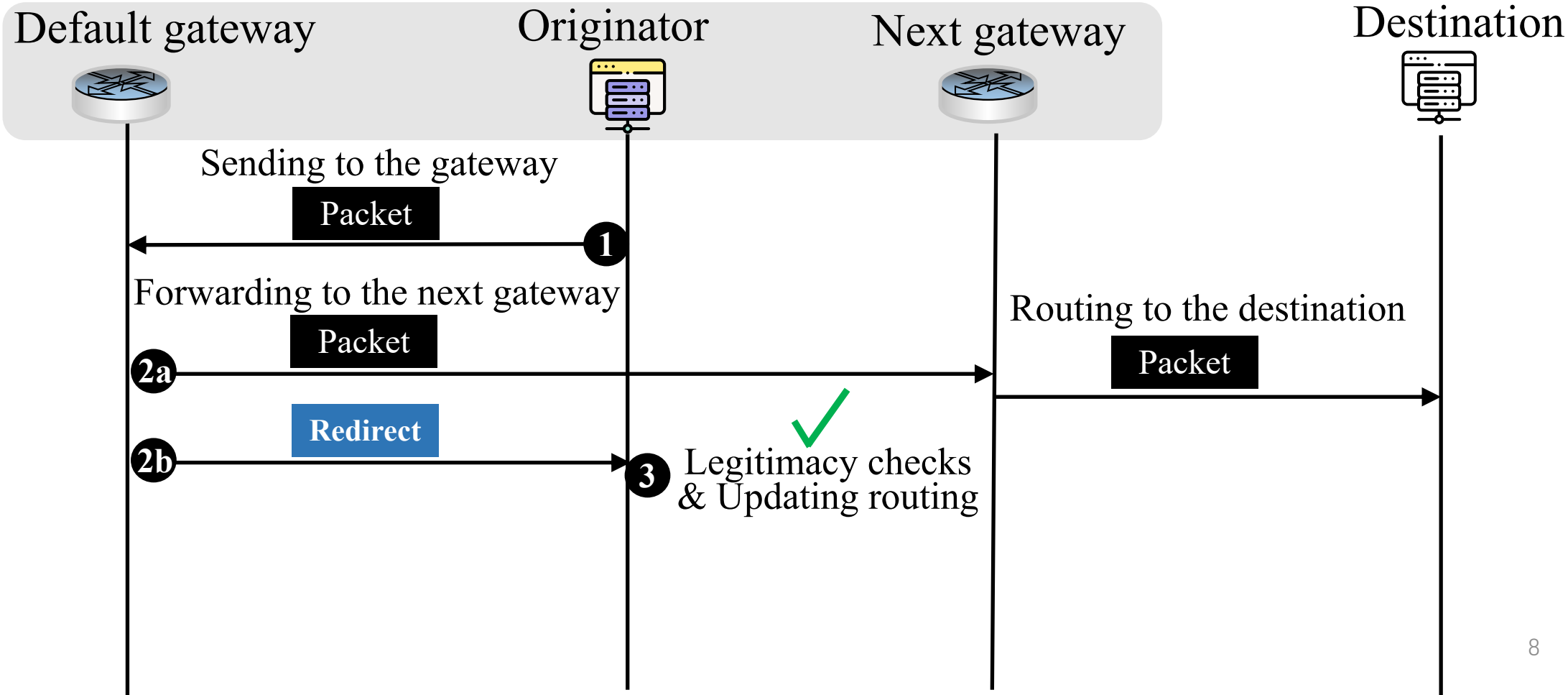
The Mechanism of ICMP Redirect

ICMP redirects allow the originator to dynamically update its routing, thus optimizing the forwarding path.



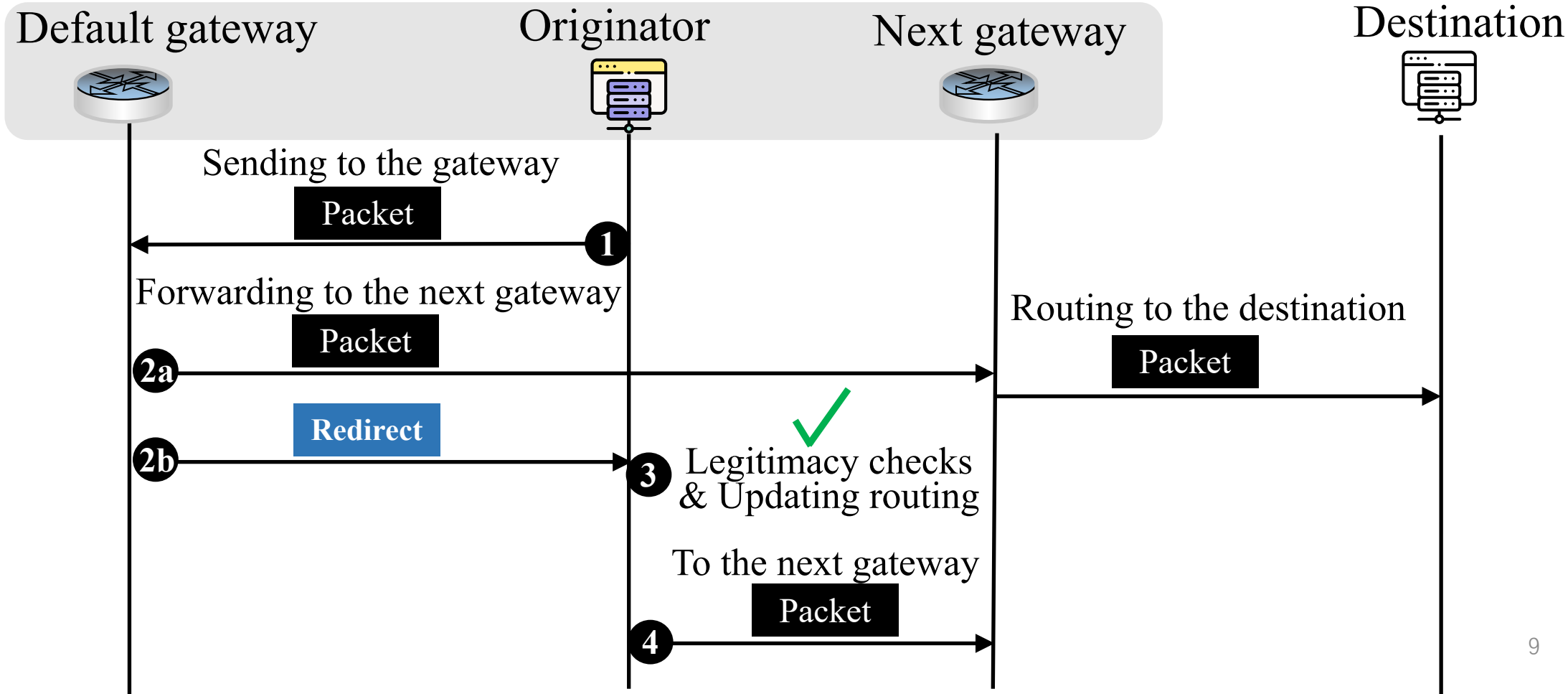
The Mechanism of ICMP Redirect

ICMP redirects allow the originator to dynamically update its routing, thus optimizing the forwarding path.



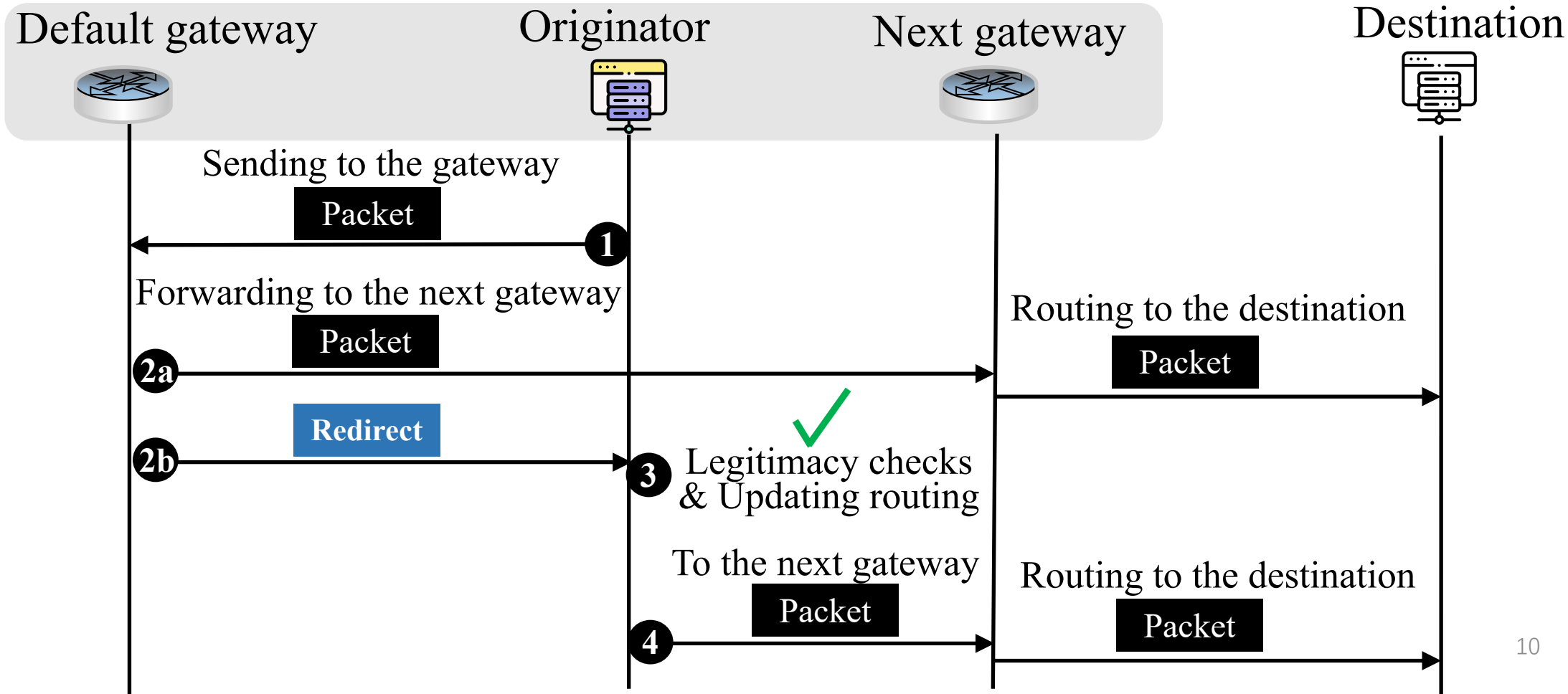
The Mechanism of ICMP Redirect

ICMP redirects allow the originator to dynamically update its routing, thus optimizing the forwarding path.



The Mechanism of ICMP Redirect

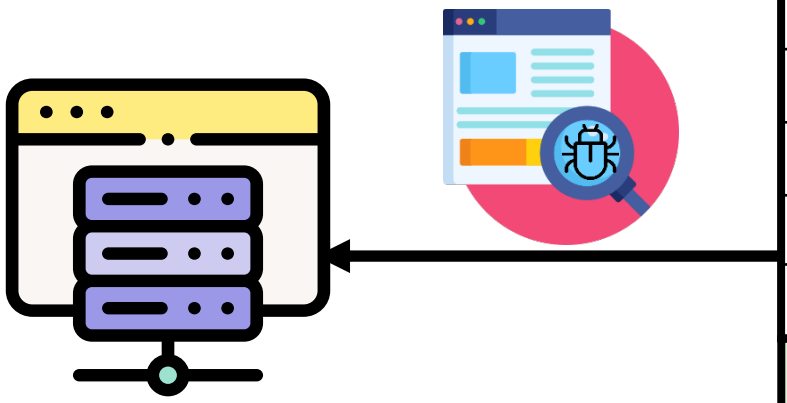
ICMP redirects allow the originator to dynamically update its routing, thus optimizing the forwarding path.



Legitimacy Checks over ICMP Errors

Legitimacy Checks over ICMP Errors

The originator will perform **two checks** over the received ICMP redirects.



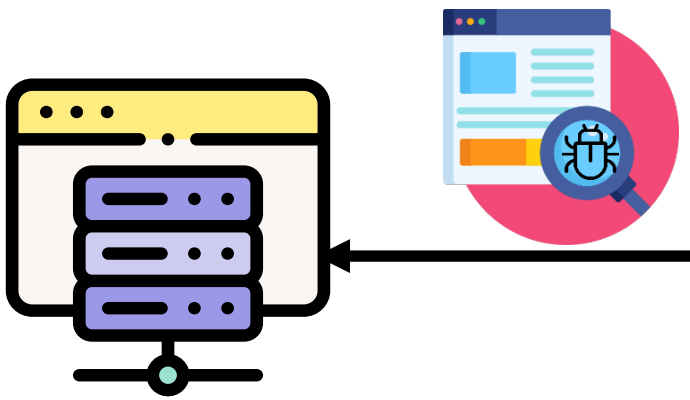
V4	IHL = 20	TOS	Total Length
IPID		X|DF|MF	Frag Offset
TTL	Protocol = ICMP		IP Header Checksum
Source address = *Gateway*			
Destination address = *Originator*			
Type = 5	Code = 0/1/2/3		ICMP Checksum
Gateway Internet Address			
V4	IHL = 20	TOS	Total Length
IPID		X|DF|MF	Frag Offset
TTL	Protocol = UDP		IP Header Checksum
Source address = *Originator*			
Destination address = *Destination*			
Source port			Destination port
Length			Checksum

28 octets

Legitimacy Checks over ICMP Errors

The originator will perform **two checks** over the received ICMP redirects.

(1) Whether the message was sent by its default gateway.



V4	IHL = 20	TOS	Total Length
IPID		X|DF|MF	Frag Offset
TTL	Protocol = ICMP		IP Header Checksum
Source address = ***Gateway***			
Destination address = *Originator*			
Type = 5	Code = 0/1/2/3		ICMP Checksum
Gateway Internet Address			
V4	IHL = 20	TOS	Total Length
IPID		X|DF|MF	Frag Offset
TTL	Protocol = UDP		IP Header Checksum
Source address = *Originator*			
Destination address = *Destination*			
Source port			Destination port
Length			Checksum

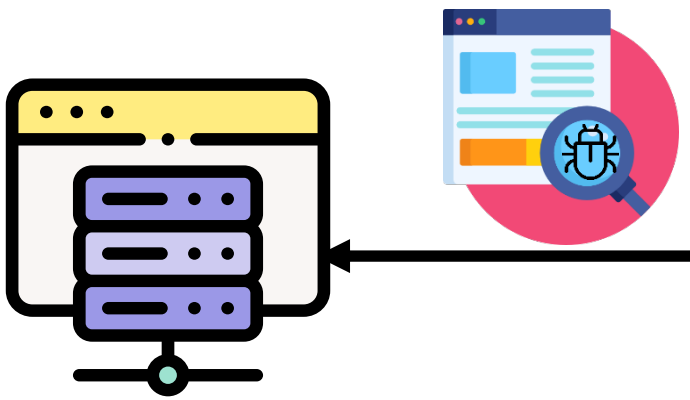
28 octets

Legitimacy Checks over ICMP Errors

The originator will perform **two checks** over the received ICMP redirects.

(1) Whether the message was sent by its default gateway.

IP spoofing



V4	IHL = 20	TOS	Total Length
IPID		X|DF|MF	Frag Offset
TTL	Protocol = ICMP		IP Header Checksum
Source address = *Gateway*			
Destination address = *Originator*			
Type = 5	Code = 0/1/2/3		ICMP Checksum
Gateway Internet Address			
V4	IHL = 20	TOS	Total Length
IPID		X|DF|MF	Frag Offset
TTL	Protocol = UDP		IP Header Checksum
Source address = *Originator*			
Destination address = *Destination*			
Source port			Destination port
Length			Checksum

28 octets

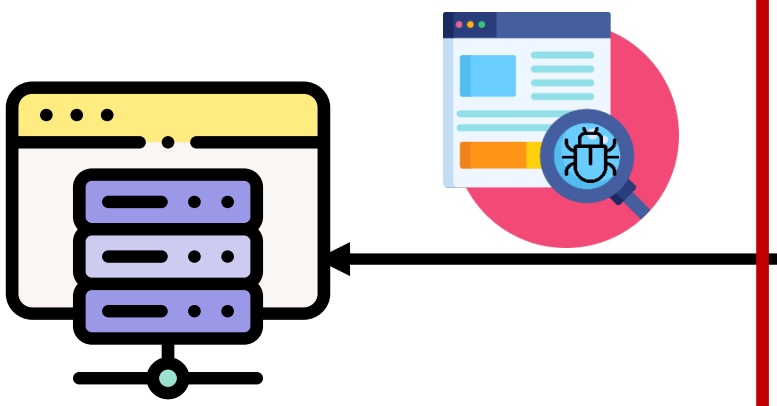
Legitimacy Checks over ICMP Errors

The originator will perform **two checks** over the received ICMP redirects.

(1) Whether the message was sent by its default gateway.

IP spoofing

(2) Checking at least 28 octets of the original packet that triggered the message.



V4	IHL = 20	TOS	Total Length
IPID		X|DF|MF	Frag Offset
TTL	Protocol = ICMP		IP Header Checksum
Source address = *Gateway*			
Destination address = *Originator*			
Type = 5	Code = 0/1/2/3		ICMP Checksum
Gateway Internet Address			
V4	IHL = 20	TOS	Total Length
IPID		X|DF|MF	Frag Offset
TTL	Protocol = UDP		IP Header Checksum
Source address = *Originator*			
Destination address = *Destination*			
Source port			Destination port
Length			Checksum

28 octets

Legitimacy Checks over ICMP Errors

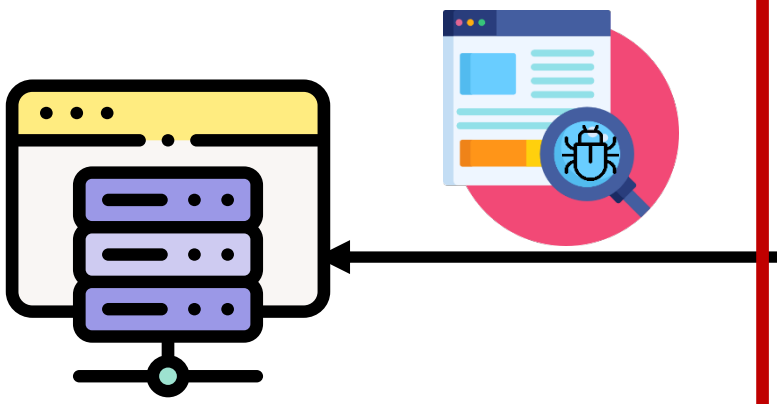
The originator will perform **two checks** over the received ICMP redirects.

(1) Whether the message was sent by its default gateway.

IP spoofing

(2) Checking at least 28 octets of the original packet that triggered the ICMP message.

Crafting 28 octets data

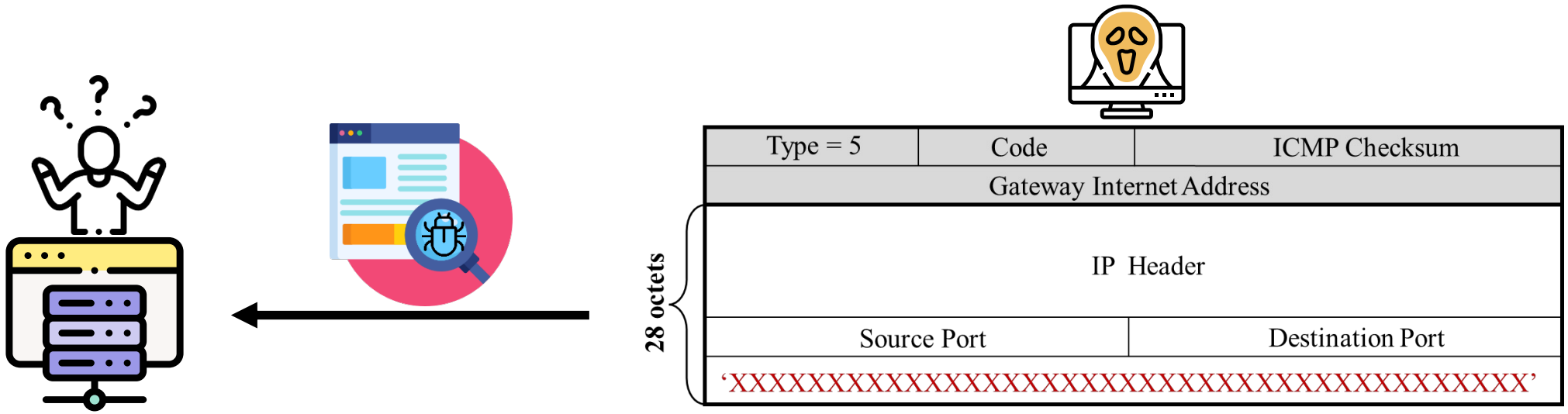


V4	IHL = 20	TOS	Total Length
IPID		X|DF|MF	Frag Offset
TTL	Protocol = ICMP		IP Header Checksum
Source address = *Gateway*			
Destination address = *Originator*			
Type = 5	Code = 0/1/2/3		ICMP Checksum
Gateway Internet Address			
V4	IHL = 20	TOS	Total Length
IPID		X|DF|MF	Frag Offset
TTL	Protocol = UDP		IP Header Checksum
Source address = *Originator*			
Destination address = *Destination*			
Source port			Destination port
Length			Checksum

28 octets

Legitimacy Checks over ICMP Errors

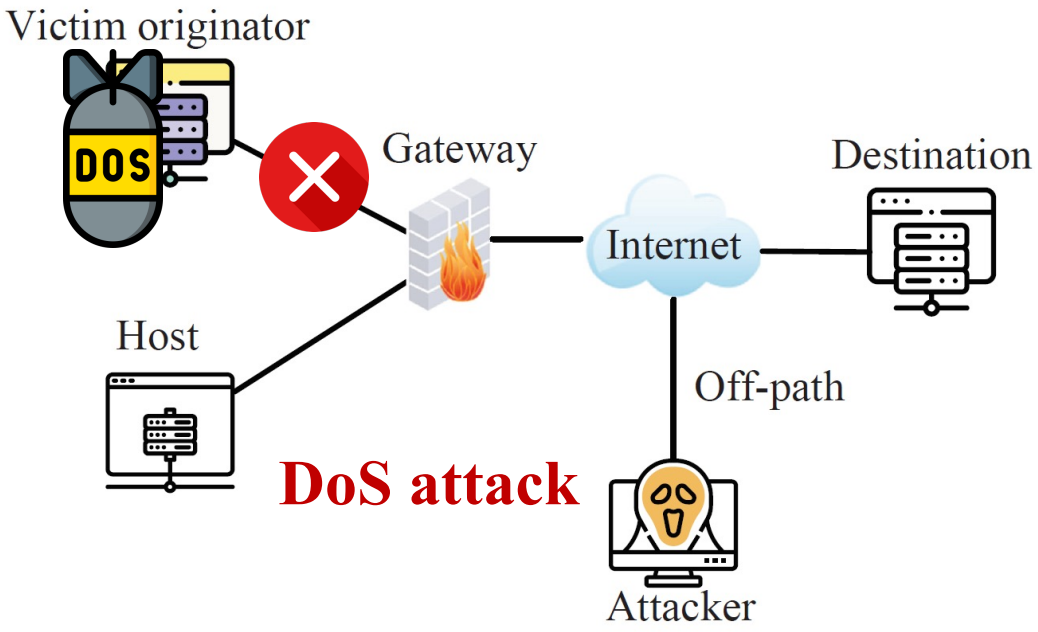
- Stateless protocols (**e.g., UDP**) cannot remember the data that has been sent earlier.
- Attackers can craft ICMP redirects **embedded with stateless protocol data** to evade the checks (including the existence of the corresponding UDP socket).

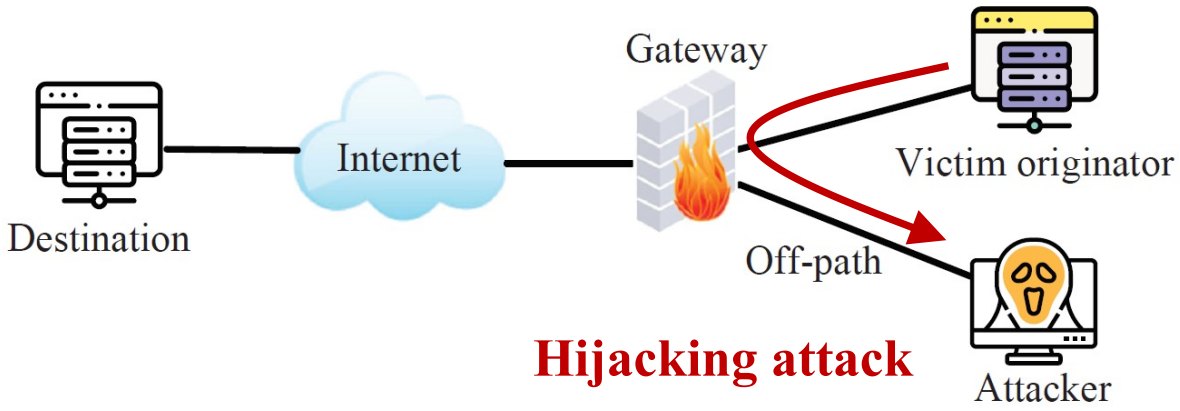


New Attacks

By crafting an evasive ICMP redirect message, attackers can construct **two types** of attacks:

- Stealthy remote **DoS attacks**
- **Network Traffic Hijacking attacks** in NAT networks



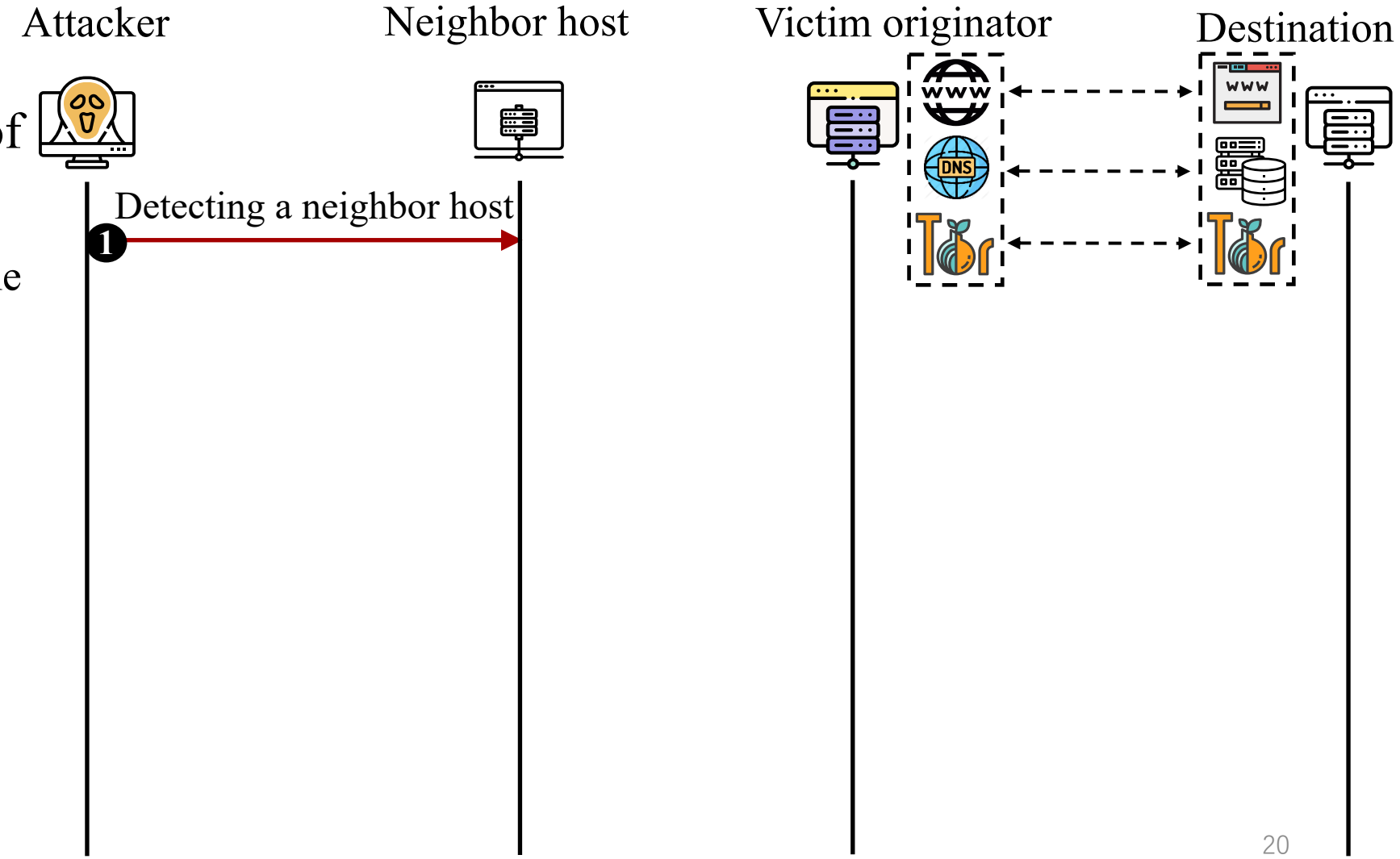


Stealthy Remote DoS Attacks

Stealthy Remote DoS Attacks

Our DoS attack consists of the following steps:

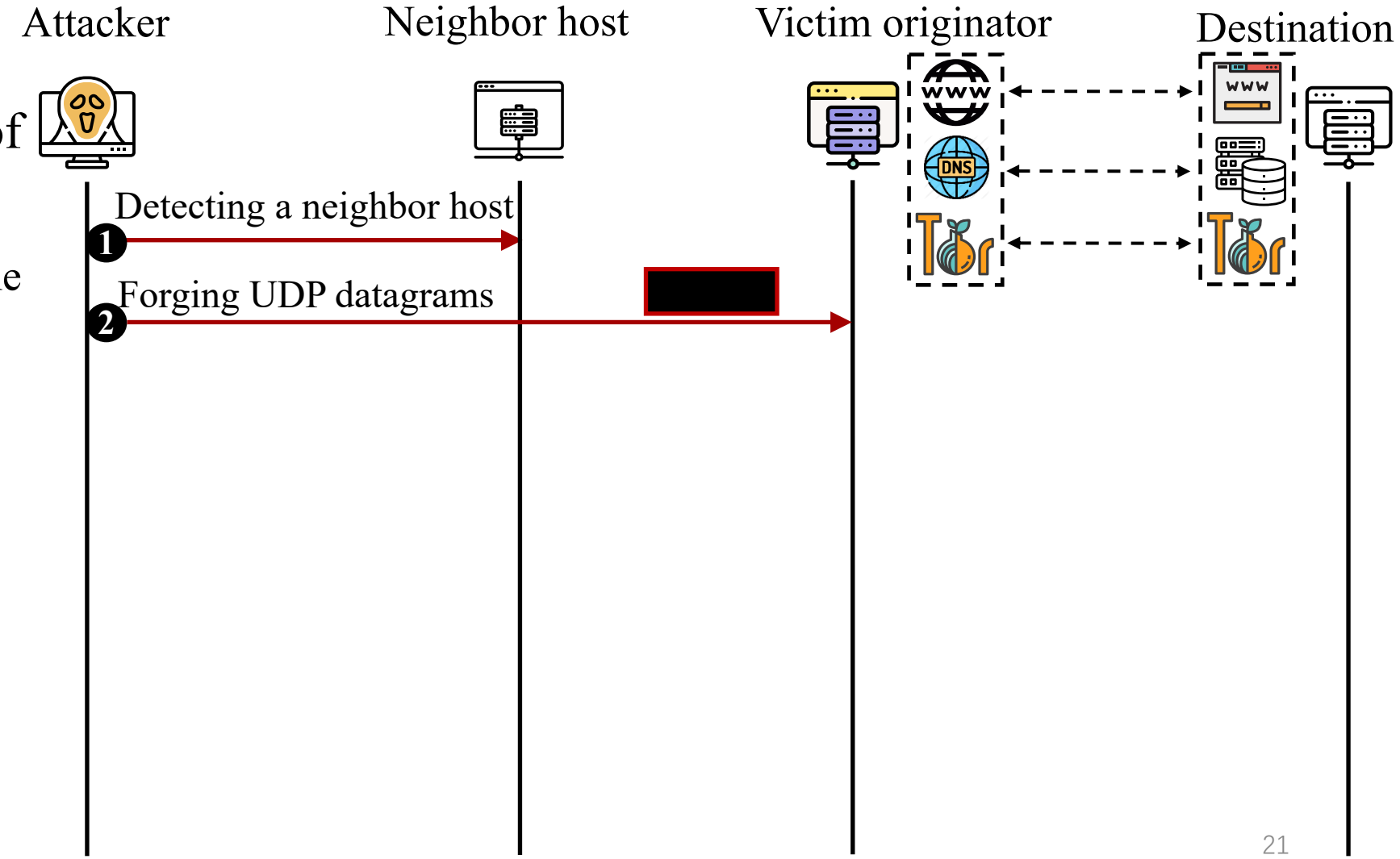
- Detecting neighbor hosts of the victim originator



Stealthy Remote DoS Attacks

Our DoS attack consists of the following steps:

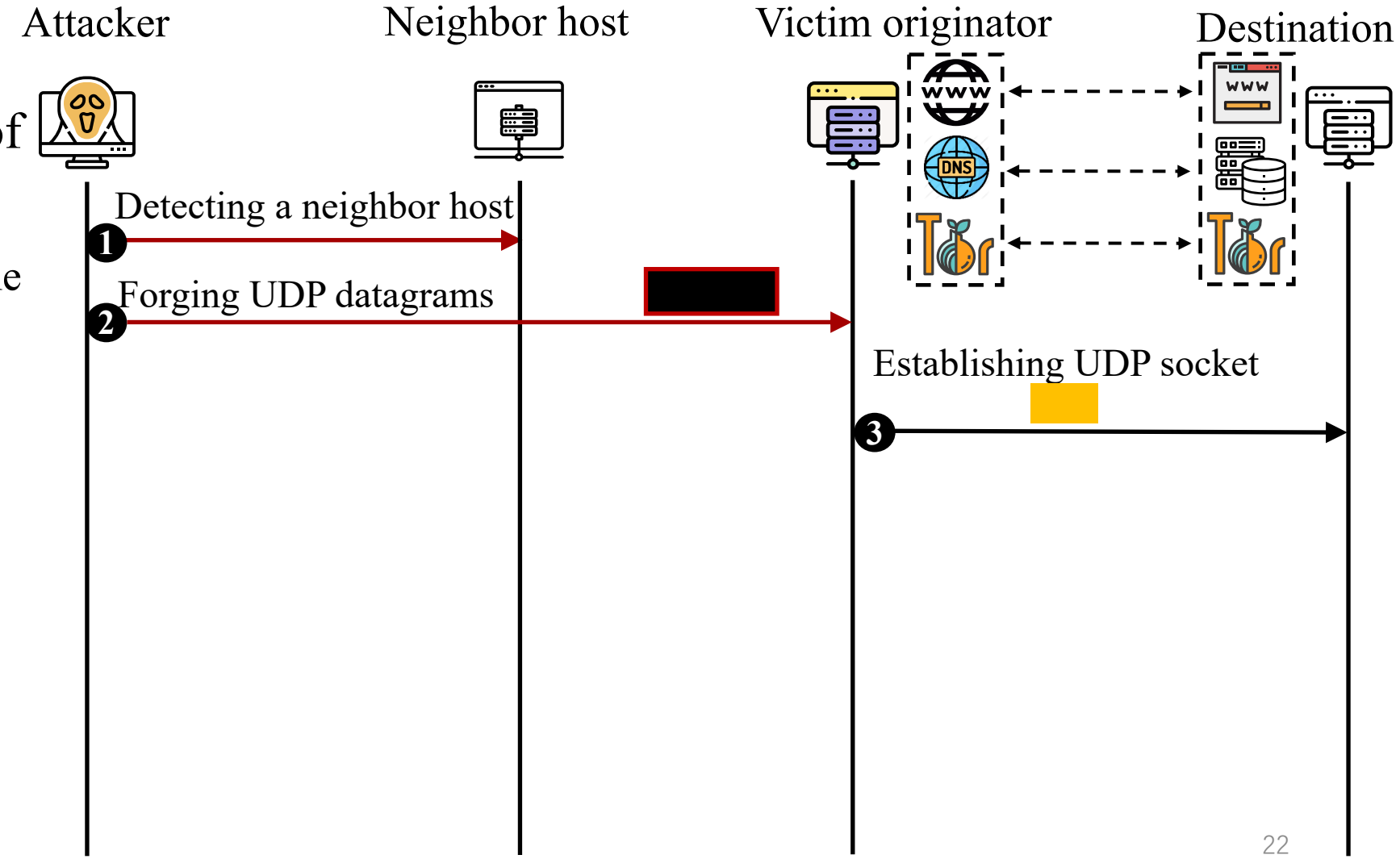
- Detecting neighbor hosts of the victim originator
- Forging UDP datagrams



Stealthy Remote DoS Attacks

Our DoS attack consists of the following steps:

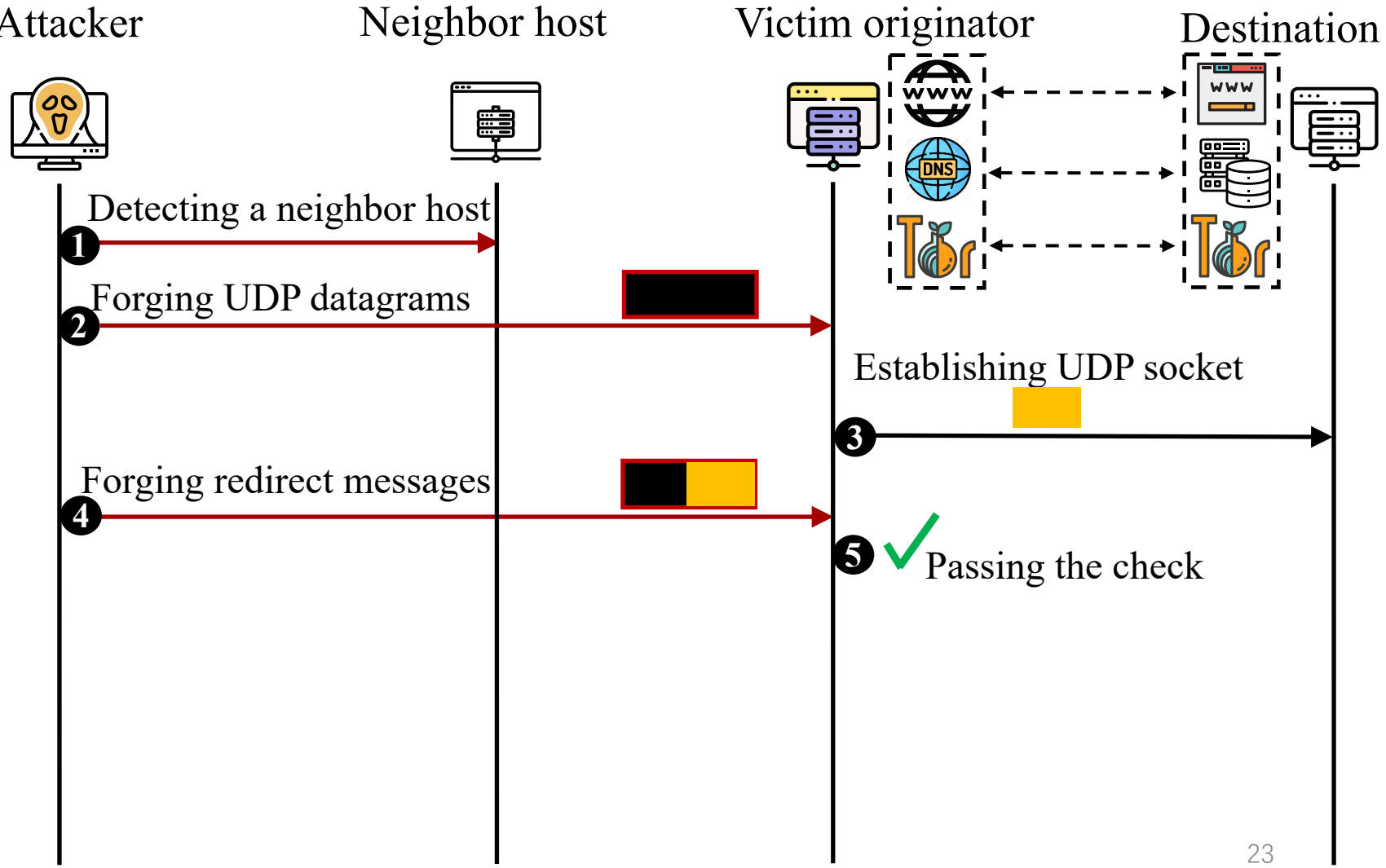
- Detecting neighbor hosts of the victim originator
- Forging UDP datagrams
- Establishing UDP sockets



Stealthy Remote DoS Attacks

Our DoS attack consists of the following steps:

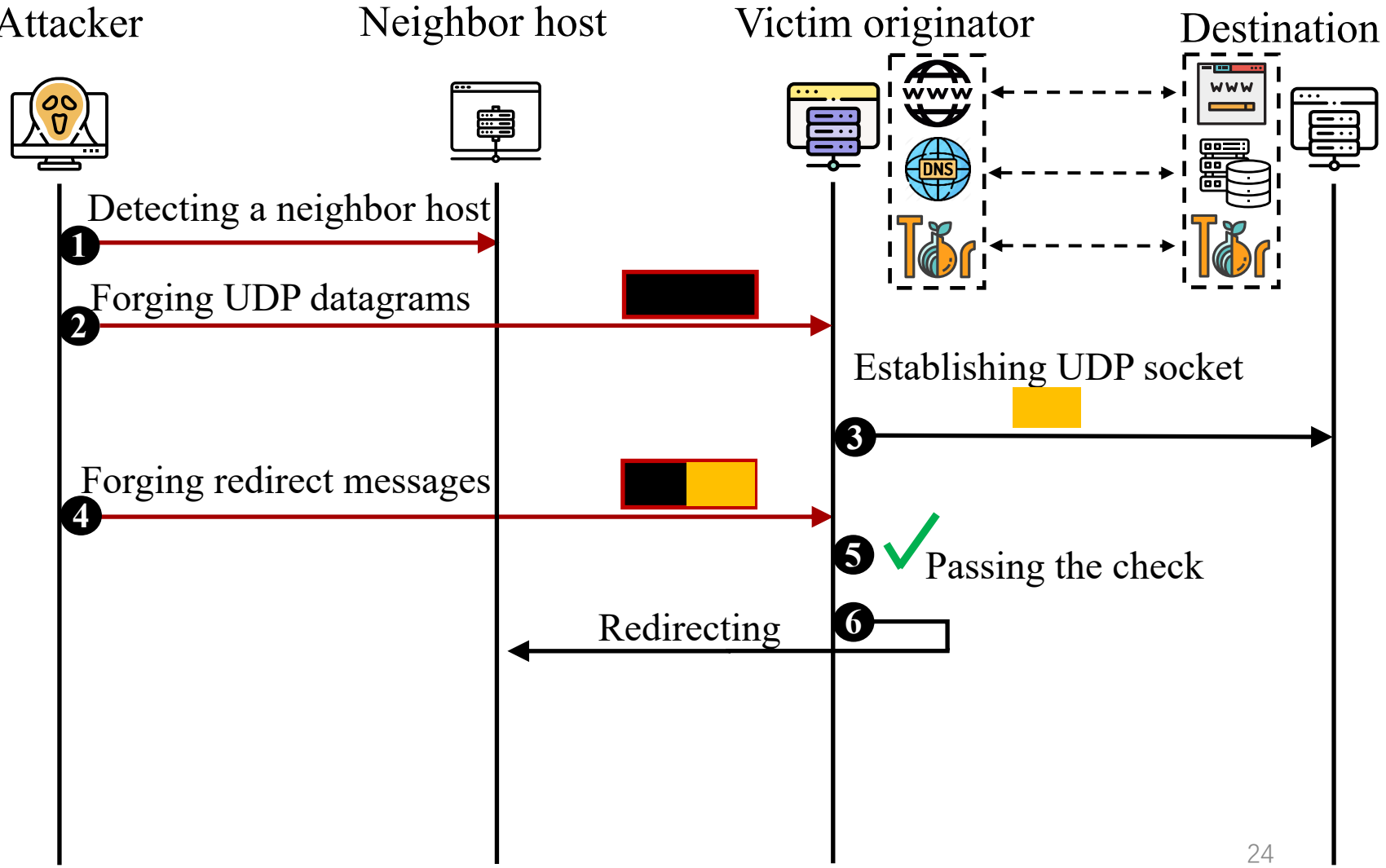
- Detecting neighbor hosts of the victim originator
- Forging UDP datagrams
- Establishing UDP sockets
- Forging redirect messages which will passing the victim's check



Stealthy Remote DoS Attacks

Our DoS attack consists of the following steps:

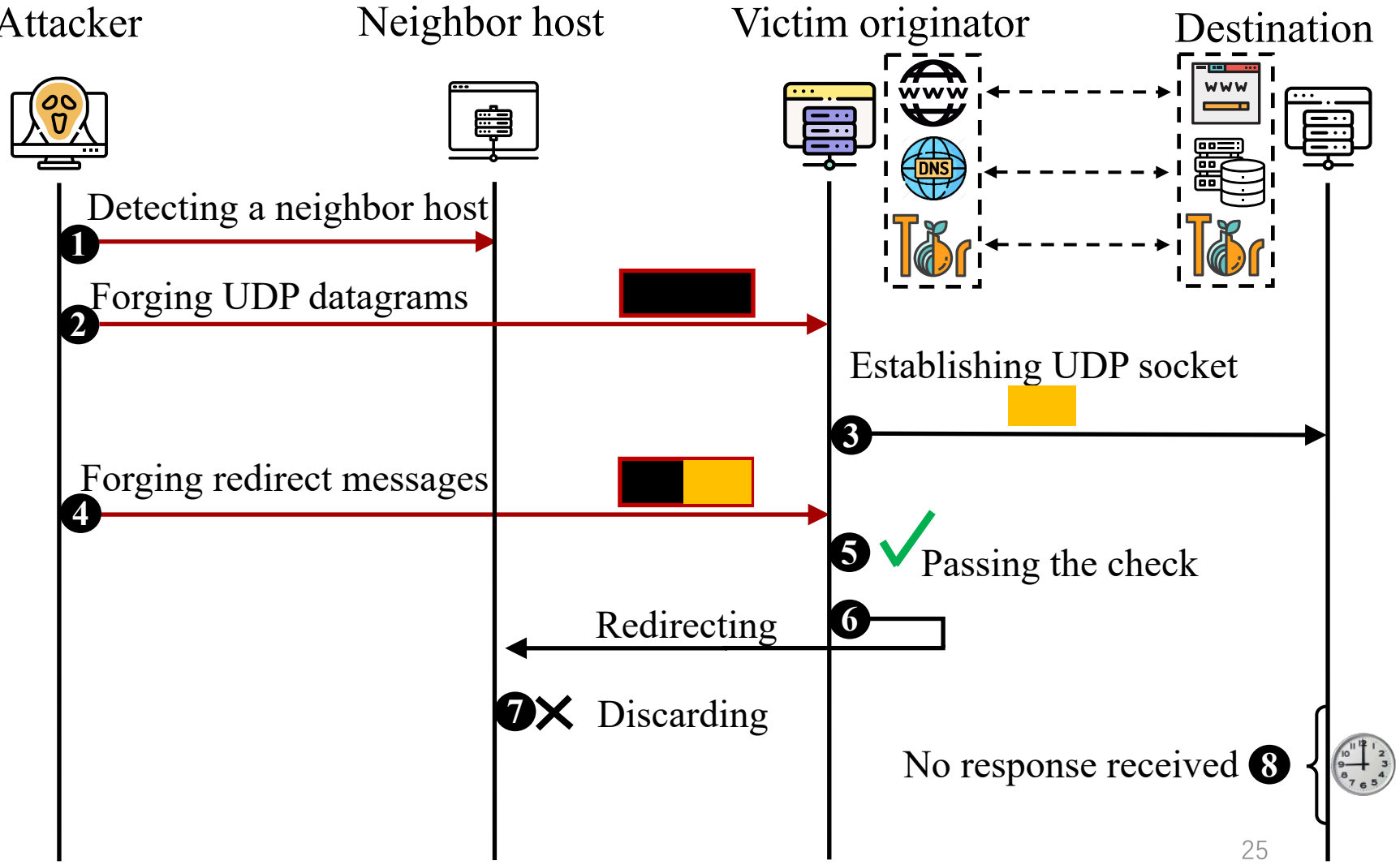
- Detecting neighbor hosts of the victim originator
- Forging UDP datagrams
- Establishing UDP sockets
- Forging redirect messages which will passing the victim's check
- Redirecting network traffic to the neighbor host



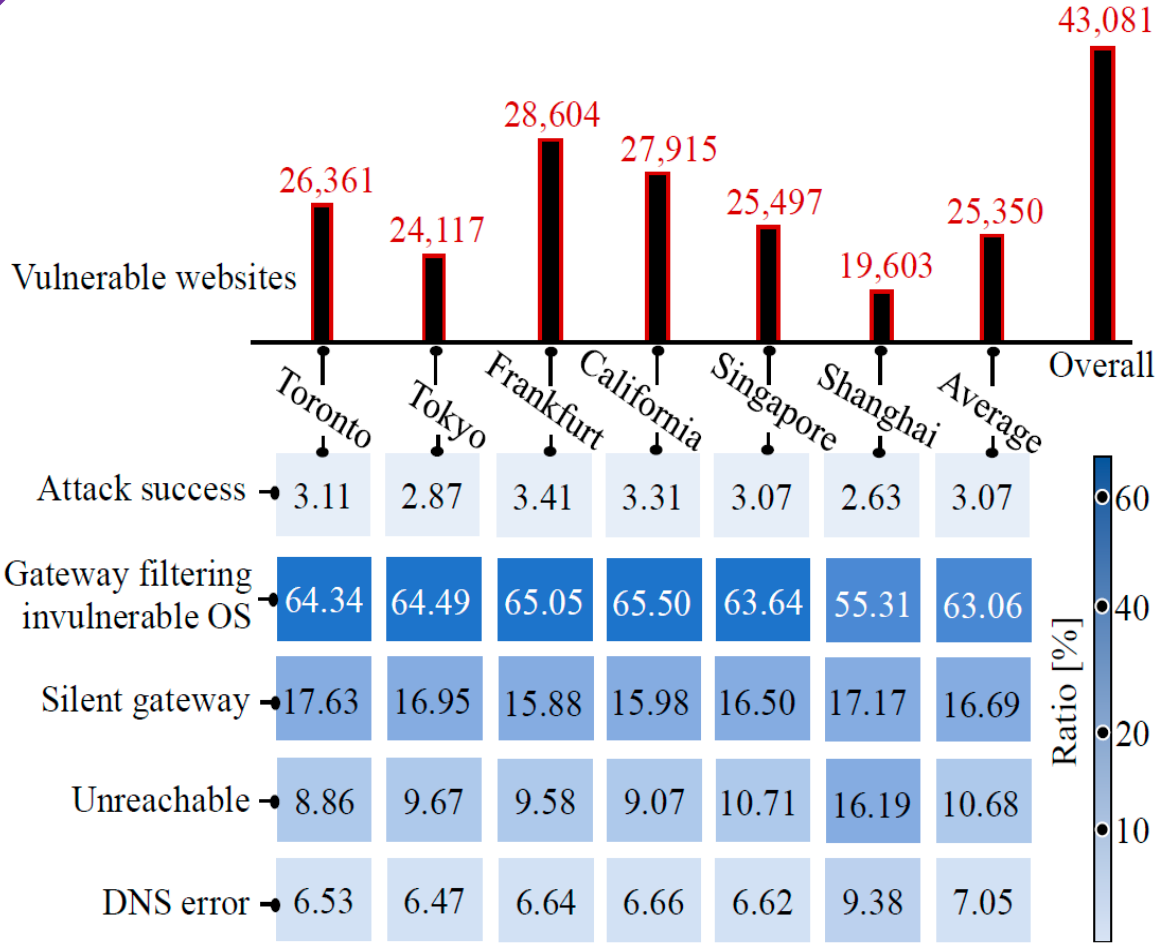
Stealthy Remote DoS Attacks

Our DoS attack consists of the following steps:

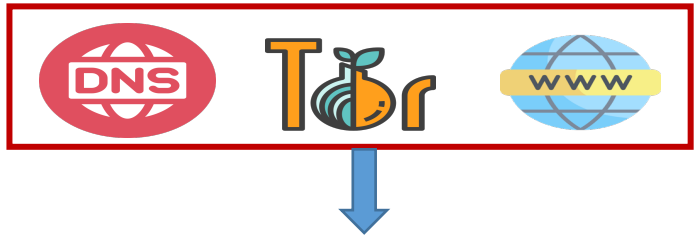
- Detecting neighbor hosts of the victim originator
- Forging UDP datagrams
- Establishing UDP sockets
- Forging redirect messages which will passing the victim's check
- Redirecting network traffic to the neighbor host
- The redirected network traffic is discarded, and the destination cannot receive any responses



Stealthy Remote DoS Attacks



Measurement results on Alexa Top 1M



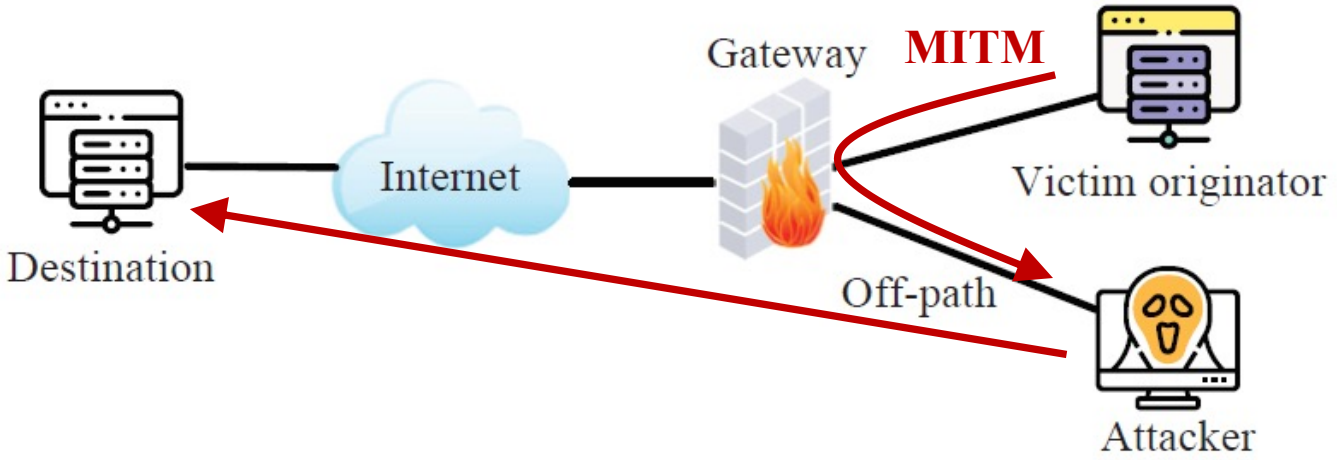
Target	Quantity	Inaccessible	Silent gateway	Invulnerable OS or filtering	Qty of Vuls.
DNS resolver	1,951,381	39.69%	15.74%	41.78%	54,470 (4.63%)
Tor relay node	6,518	18.52%	26.22%	52.41%	186 (3.50%)
Website	Alexa top 1 million	17.73%	16.69%	63.06%	25,350 (3.07%)

Comparisons of the DoS attack under different network scenarios (DNS and Tor)

Network Traffic Hijacking Attacks

Network Traffic Hijacking Attacks

- The attacker and the victim originator reside in the same network.
- The attacker can act as the next hop of the victim to hijack the victim's traffic for the destination.

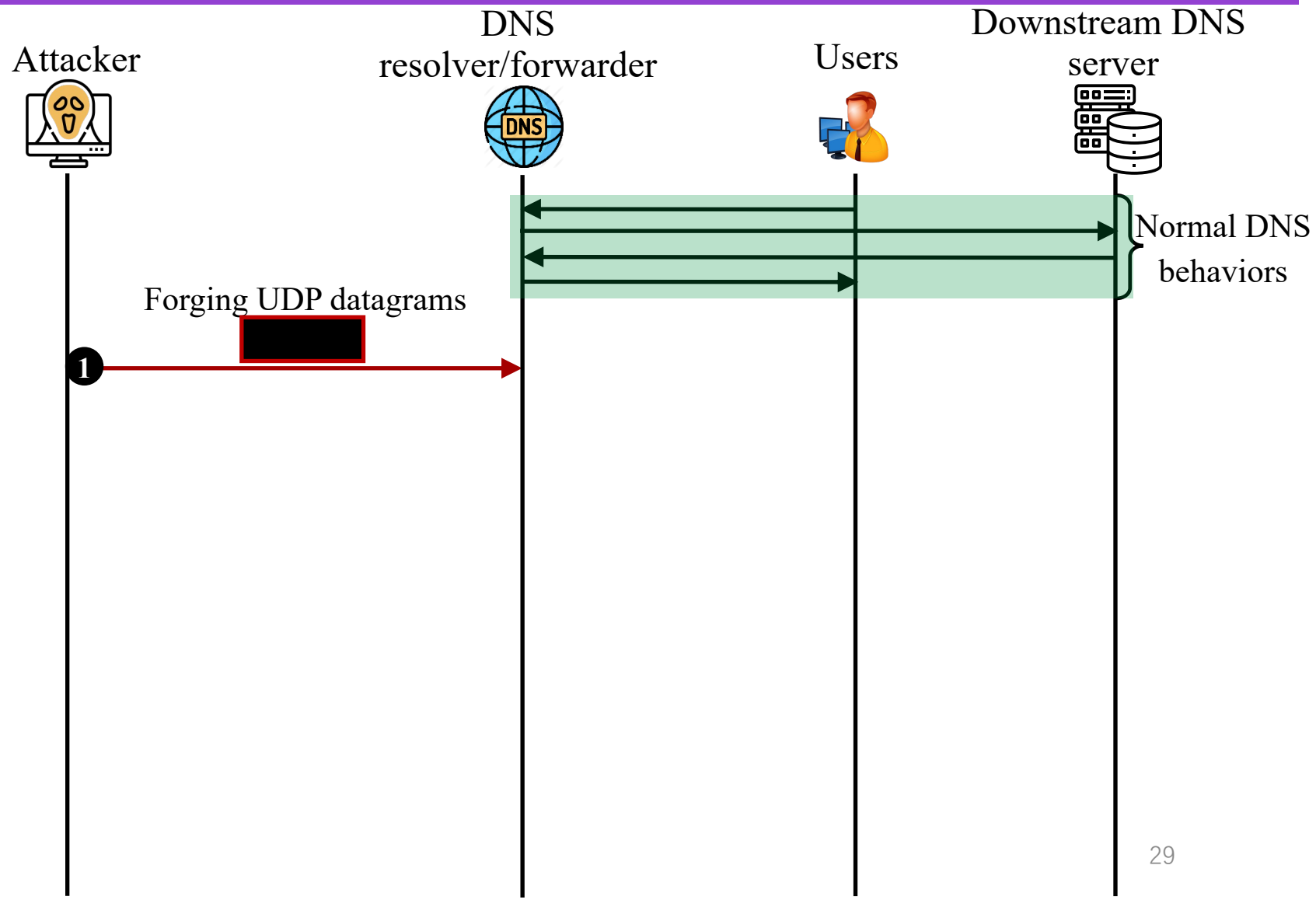


- Our attack can be conducted under various scenarios to compromise the network.
  - e.g., hijack DNS queries from a local DNS forwarder and then poison the local DNS cache of the NAT network.

Network Traffic Hijacking Attacks

DNS requests hijacking in NAT networks

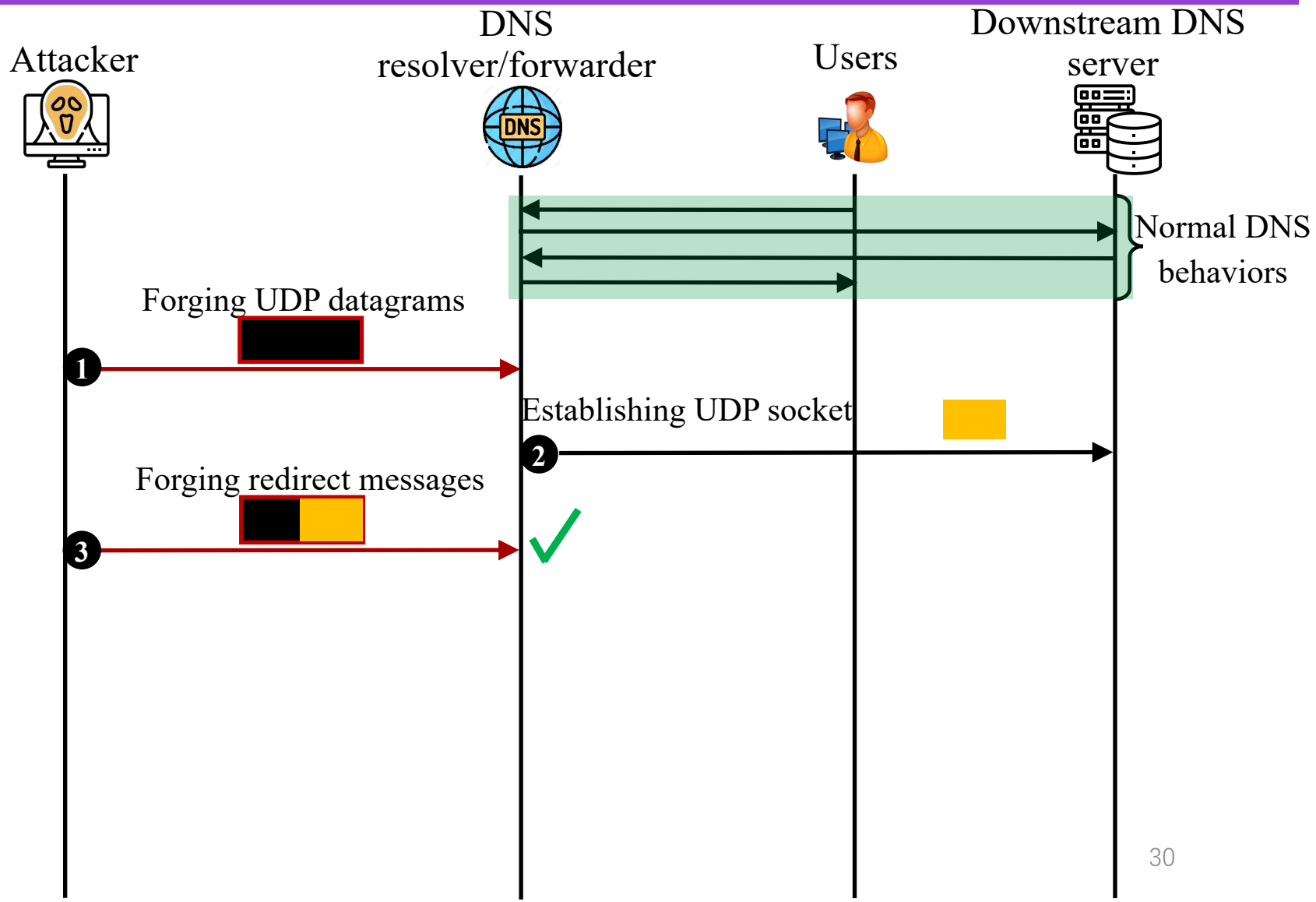
- Forging UDP datagrams



Network Traffic Hijacking Attacks

DNS requests hijacking in NAT networks

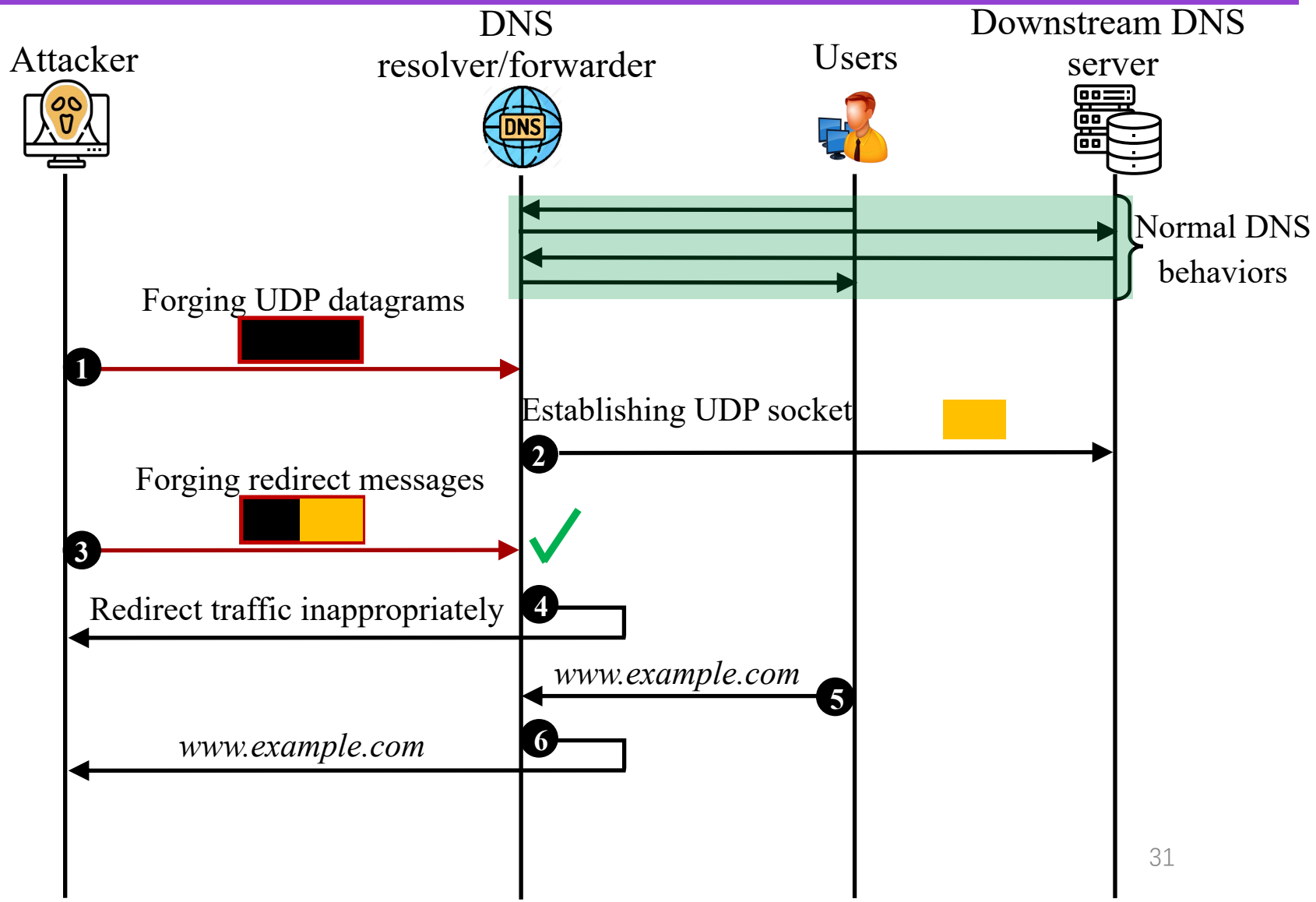
- Forging UDP datagrams
- Establishing UDP socket
- Forging ICMP redirects



Network Traffic Hijacking Attacks

DNS requests hijacking in NAT networks

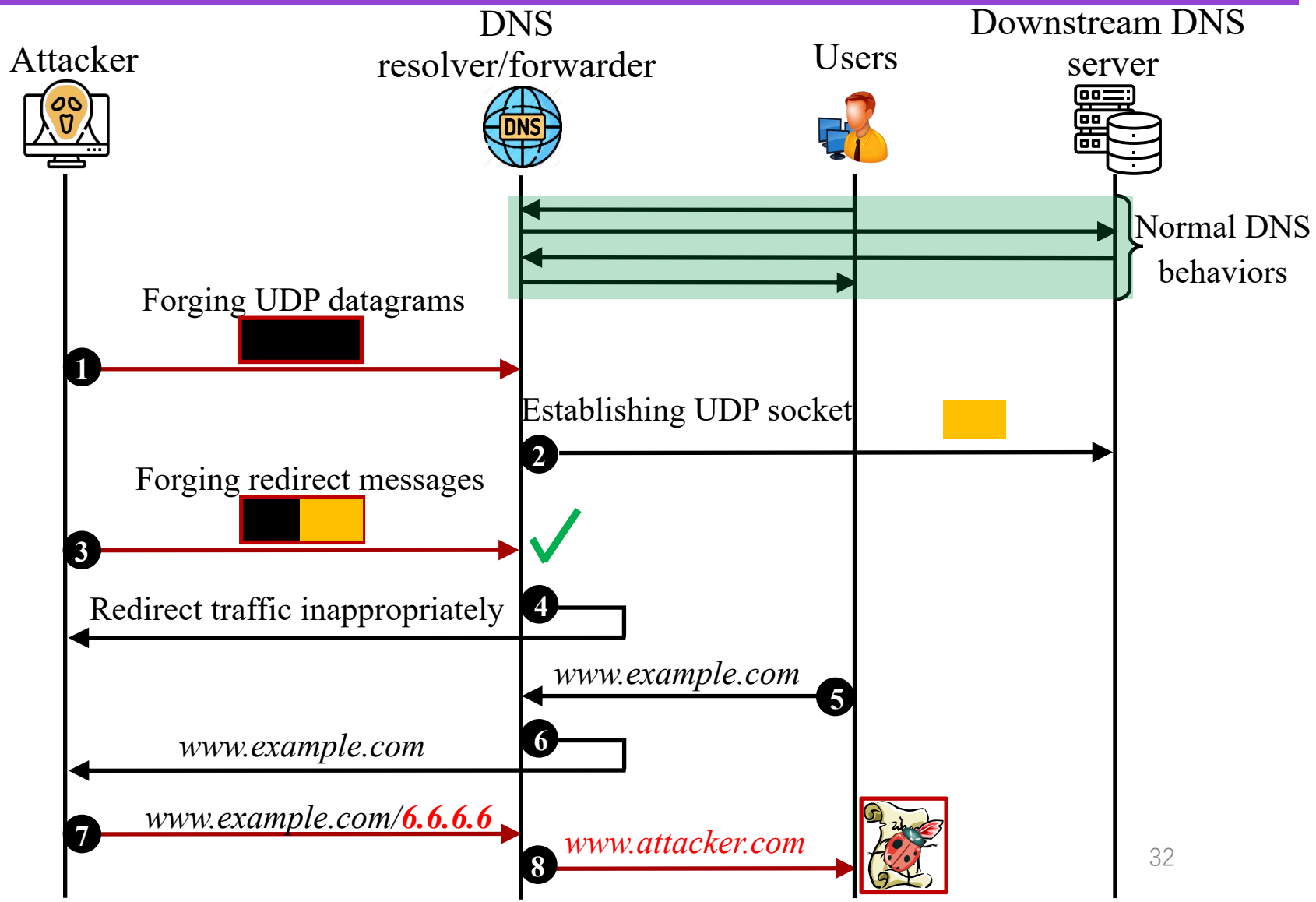
- Forging UDP datagrams
- Establishing UDP socket
- Forging ICMP redirects
- Redirecting network traffic
- DNS queries from users
- Intercepted by the attacker



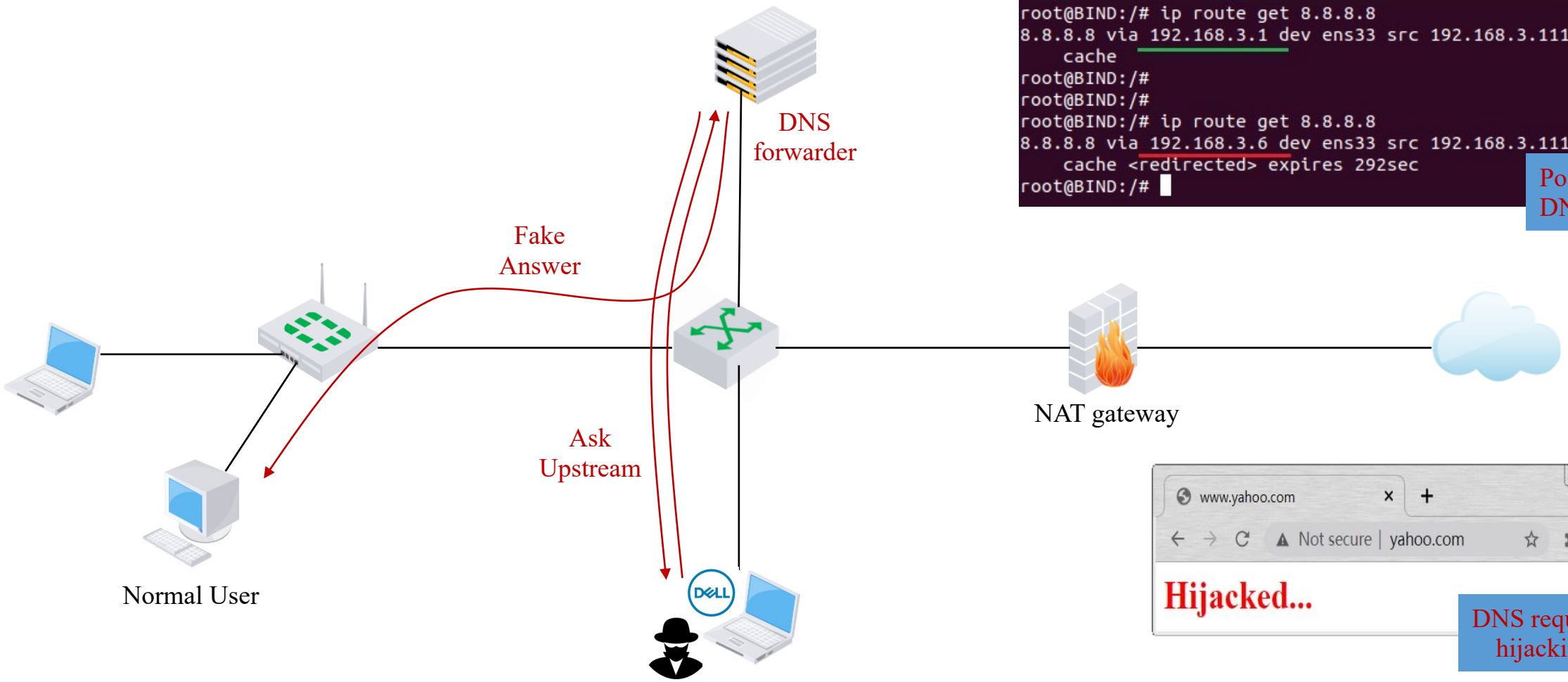
Network Traffic Hijacking Attacks

DNS requests hijacking in NAT networks

- Forging UDP datagrams
- Establishing UDP socket
- Forging ICMP redirects
- Redirecting network traffic
- DNS queries from users
- Intercepted by the attacker
- DNS cache poisoning
- Malicious replies to the user



Network Traffic Hijacking Attacks




```
root@BIND: /
root@BIND:/# ip route get 8.8.8.8
8.8.8.8 via 192.168.3.1 dev ens33 src 192.168.3.111 uid 0
    cache
root@BIND:/#
root@BIND:/#
root@BIND:/# ip route get 8.8.8.8
8.8.8.8 via 192.168.3.6 dev ens33 src 192.168.3.111 uid 0
    cache <redirected> expires 292sec
root@BIND:/#
```

Poisoned DNS cache

DNS requests hijacking

Countermeasures

Countermeasures

- Network Changes

  Block crafted ICMP redirects.

- Protocols Changes

  Improve legitimacy check mechanism of ICMP errors.

- Host Changes

  Disable the ICMP redirect mechanism for stateless protocols.

Conclusion

- We uncover a vulnerability in the legitimacy check mechanism of ICMP errors, which affects a wide range of major OSes.
- We show that ICMP redirect attacks can be revitalized to cause serious damages in the real world.
- We propose countermeasures via network changes, protocol changes, and/or host changes.

Thank you!