TCP的厄运，网络协议侧信道分析及利用

- Zhiyun Qian, University of California, Riverside

Research Interest

- Vulnerability discovery and exploitation techniques
  - CVE-2016-5696, CVE-2015-8950, CVE-2016-8756, CVE-2016-8757, CVE-2015-8950, CVE-2016-8758, CVE-2016-3360, CVE-2016-3683, …

- Side channels analysis (system/network)
  - Live demo competition @ GeekPwn 2016 and 2017

Research Interest

- Vulnerability discovery and exploitation techniques
  - CVE-2016-5696, CVE-2015-8950, CVE-2016-8756, CVE-2016-8757, CVE-2015-8950, CVE-2016-8758, CVE-2016-3360, CVE-2016-3683, …
- Side channels analysis (system/network)
  - Live demo competition @ GeekPwn 2016 and 2017

Research Interest

- Vulnerability discovery and exploitation techniques
  - CVE-2016-5696, CVE-2015-8950, CVE-2016-8756, CVE-2016-8757, CVE-2015-8950, CVE-2016-8758, CVE-2016-3360, CVE-2016-3683, …
- Side channels analysis (system/network)
  - Live demo competition @ GeekPwn 2016 and 2017
- Measurement/characterization
  - Internet-wide scan
  - One-click root app

Real world side channel attacks – mafia game

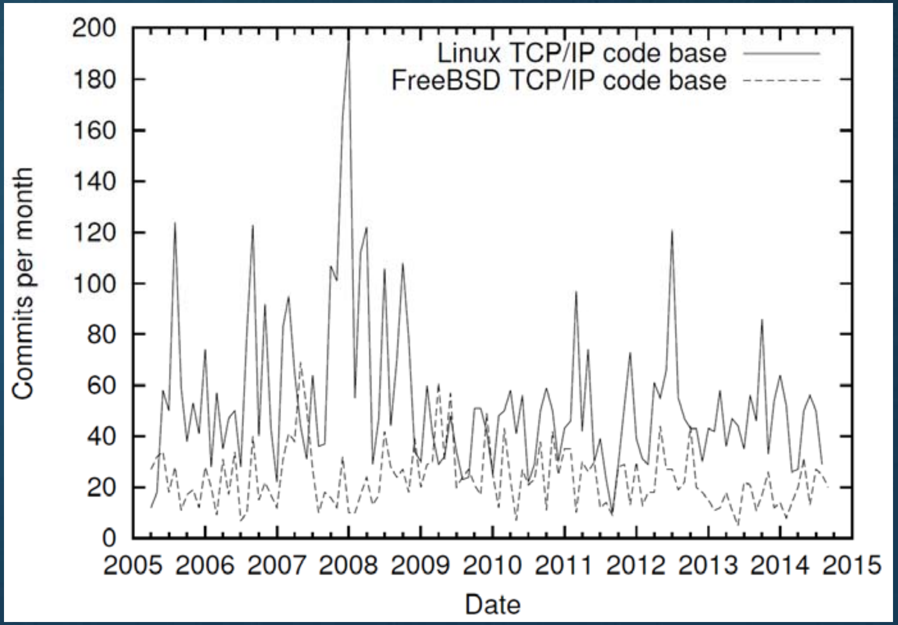
Another example

Anyone at home?

???

Why TCP?

- Extremely widely used, any security issues could have huge impact



- But decades old
- Yet still under heavy development
  - TCP fast open
  - Security improvements
  - Maintenance

Outline

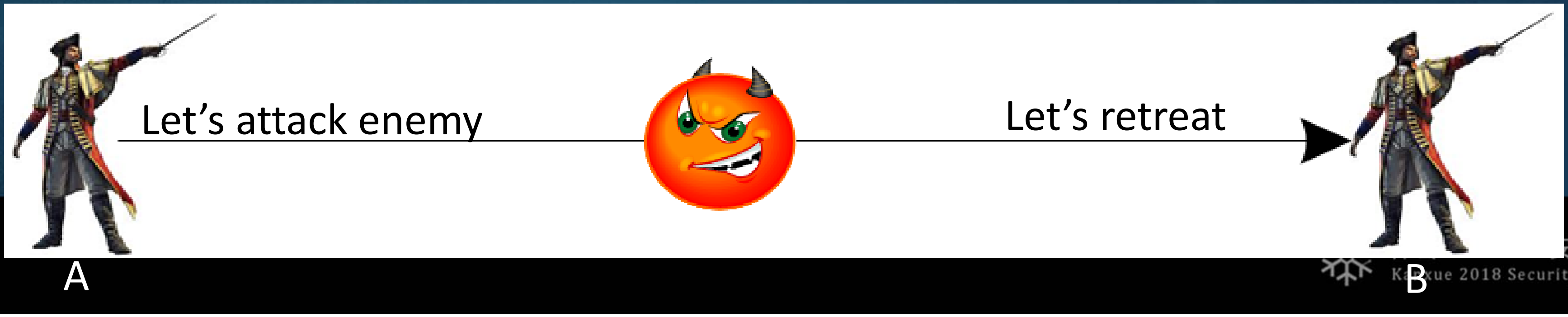
- Background on Off-Path TCP Exploits
- Off-Path TCP Exploits
  - Malware-assisted [Oakland 12, CCS 12]
  - Pure Off-Path [USENIX Security 16]
  - Unfixable WiFi timing [USENIX Security 18]

Outline

- Background on Off-Path TCP Exploits
- Off-Path TCP Exploits
  - Malware-assisted
  - Pure Off-Path
  - Unfixable WiFi timing

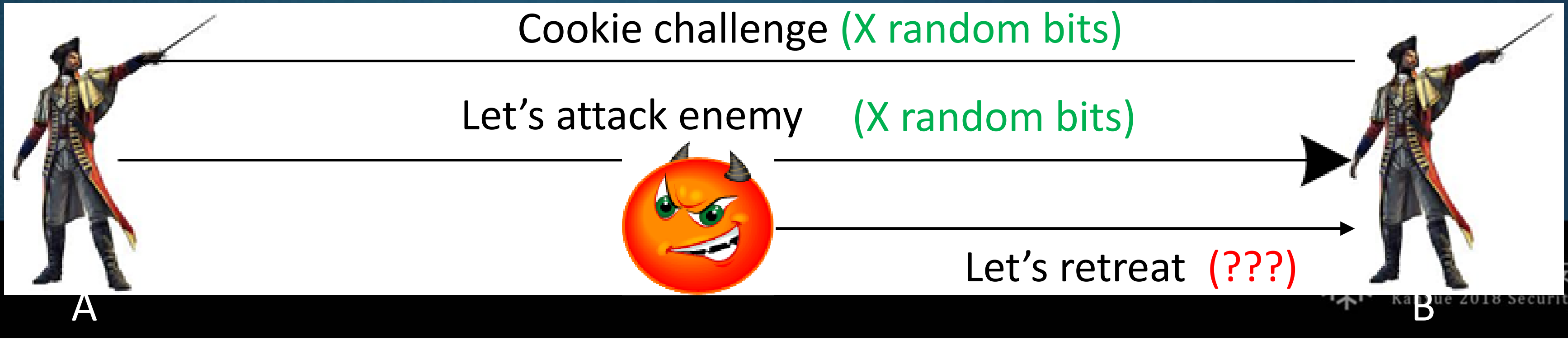
Man-in-the-middle vs. Off-path attacks

- Man-in-the-middle attacks
  - On the communication path
    - Harder but possible: open wifi, route hijack, etc.
  - Prevention with crypto: PKI, complexity, overhead
    - Caching (HTTP-only) in mobile networks
    - Why bother, <45% of Alexa top 1M traffic encrypted in 2016 [wikipedia]



Man-in-the-middle vs. Off-path attacks

- Off-path attacks
  - Off the communication path
    - Cannot intercept/modify/block traffic
  - Prevention with challenge-response (e.g., cookie)
    - Subject to prediction or side channel attacks!



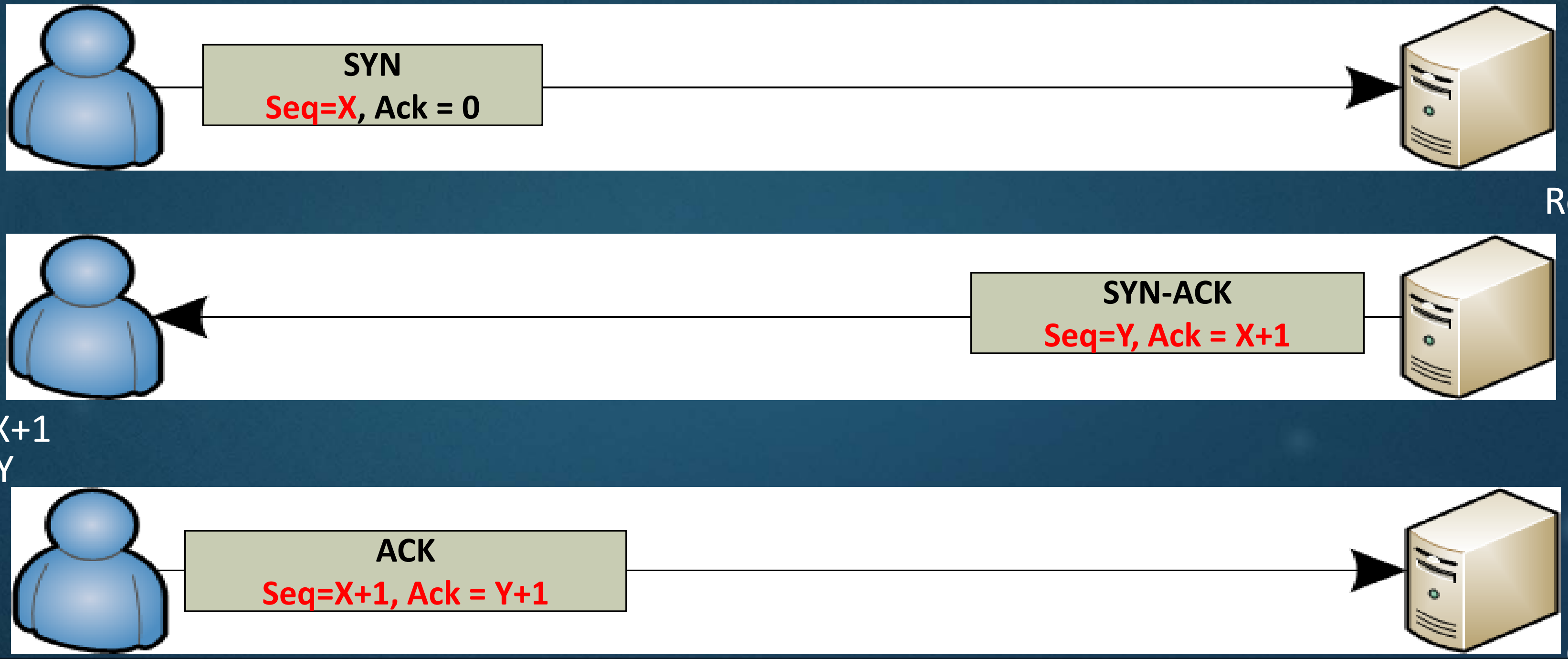
Cookie challenge (X random bits)

Let’s attack enemy (X random bits)

Let’s retreat (???)

Initial TCP sequence number as challenge/response

□Three-way handshake



SYN
Seq=X, Ack = 0

Remembers X

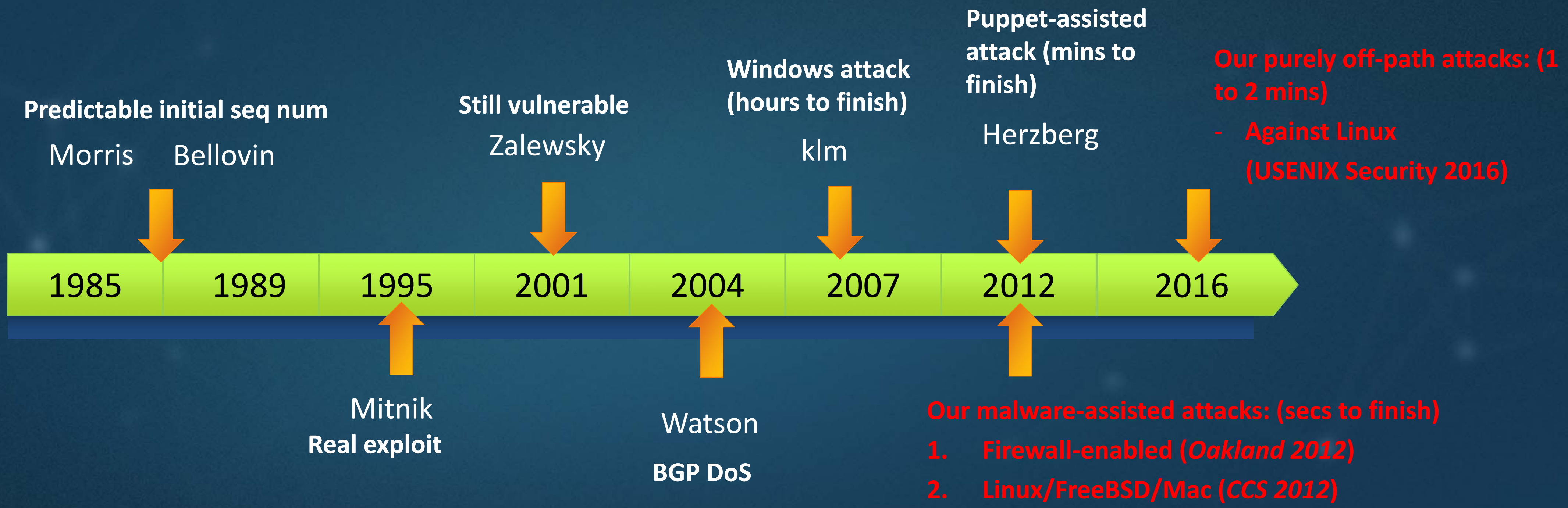
SYN-ACK
Seq=Y, Ack = X+1

Checks Ack=X+1
Remembers Y

ACK
Seq=X+1, Ack = Y+1

Checks Ack=Y+1

TCP sequence number war timeline



Predictable initial seq num

Morris Bellovin

Still vulnerable

Zalewsky

Windows attack (hours to finish)

klm

Puppet-assisted attack (mins to finish)

Herzberg

Our purely off-path attacks: (1 to 2 mins)

- **Against Linux (USENIX Security 2016)**

Mitnik

Real exploit

Watson

BGP DoS

Our malware-assisted attacks: (secs to finish)

1. **Firewall-enabled (*Oakland 2012*)**
2. **Linux/FreeBSD/Mac (*CCS 2012*)**

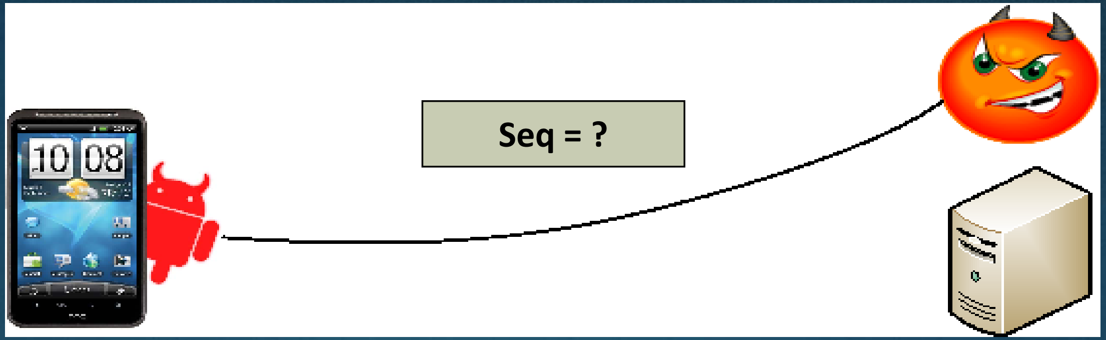
Research contributions

- Uncover a new class of side channel attacks against TCP
- Real-world security vulnerabilities caused by
  - Firewall middleboxes
  - OS implementation
  - TCP specifications
- Develop program analysis tools to automatically identify such class of vulnerabilities

Outline

- Background on Off-Path TCP Exploits
- Off-Path TCP Exploits
  - Malware-assisted [Oakland 12, CCS 12]
  - Pure Off-Path
  - Unfixable WiFi timing

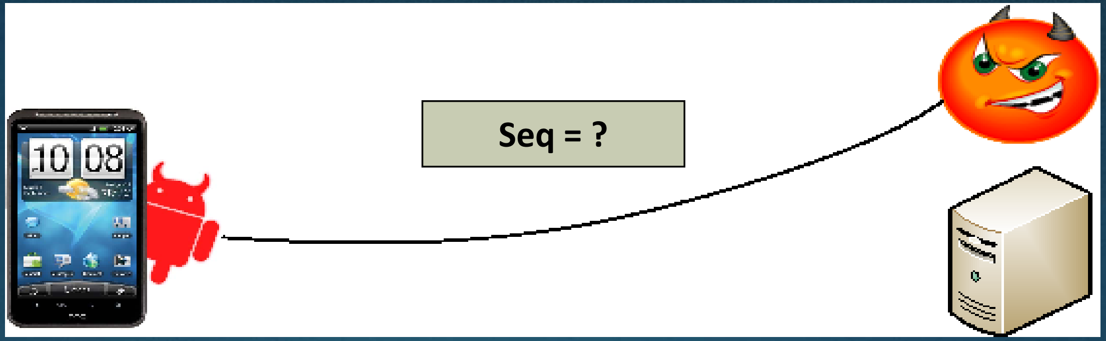
Threat model and problem formulation



- Unprivileged malware + off-path attackers
- Attack goal
  - Write into a target connection owned by a different app

(e.g., facebook connection)

TCP sequence number inference attack



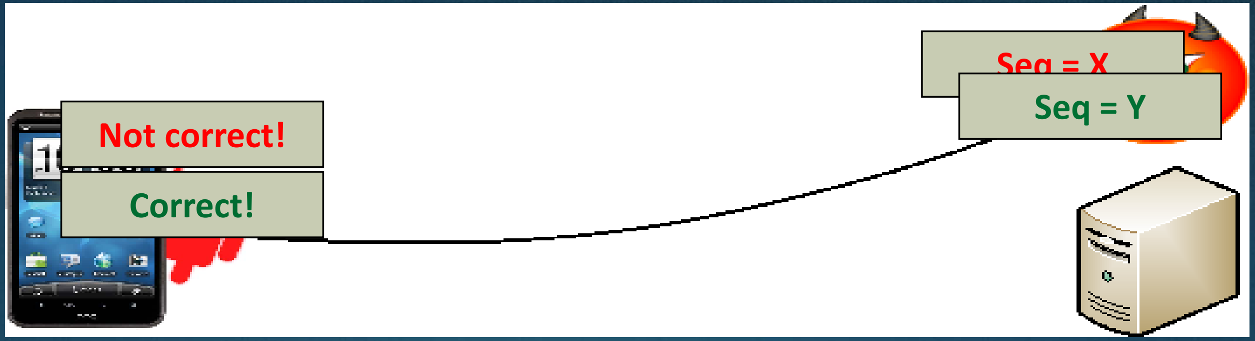
- Required information
  - Target four tuples: (source/dest IP, source/dest port)
  - Sequence number
    - How? Unprivileged malware is isolated from other apps

Req 1 – Obtaining target four tuples

- On-site unprivileged malware
  - netstat (no root required)

```
netstat -nn
Active Internet connections
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4      37      0  192.168.1.102.50469    199.47.219.159.443     CLOSE_WAIT
tcp4      37      0  192.168.1.102.50468    174.129.195.86.443     CLOSE_WAIT
tcp4      37      0  192.168.1.102.50467    199.47.219.159.443     CLOSE_WAIT
tcp4       0      0  192.168.1.102.50460    199.47.219.159.443     LAST_ACK
tcp4       0      0  192.168.1.102.50457    199.47.219.159.443     LAST_ACK
tcp4       0      0  192.168.1.102.50445    199.47.219.159.443     LAST_ACK
tcp4       0      0  192.168.1.102.50441    199.47.219.159.443     LAST_ACK
tcp4       0      0  127.0.0.1.26164        127.0.0.1.50422        ESTABLISHED
```

Req 2 – Feedback through side channels

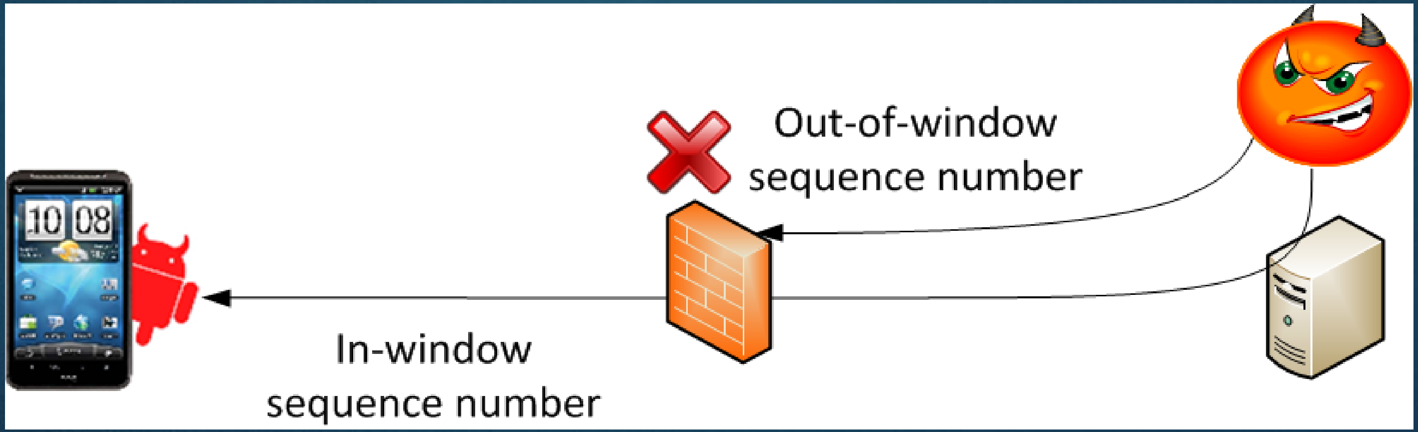


Expecting seq Y

- Intuition: actively guess sequence numbers and observe feedbacks through side channels

Firewall-enabled side channels

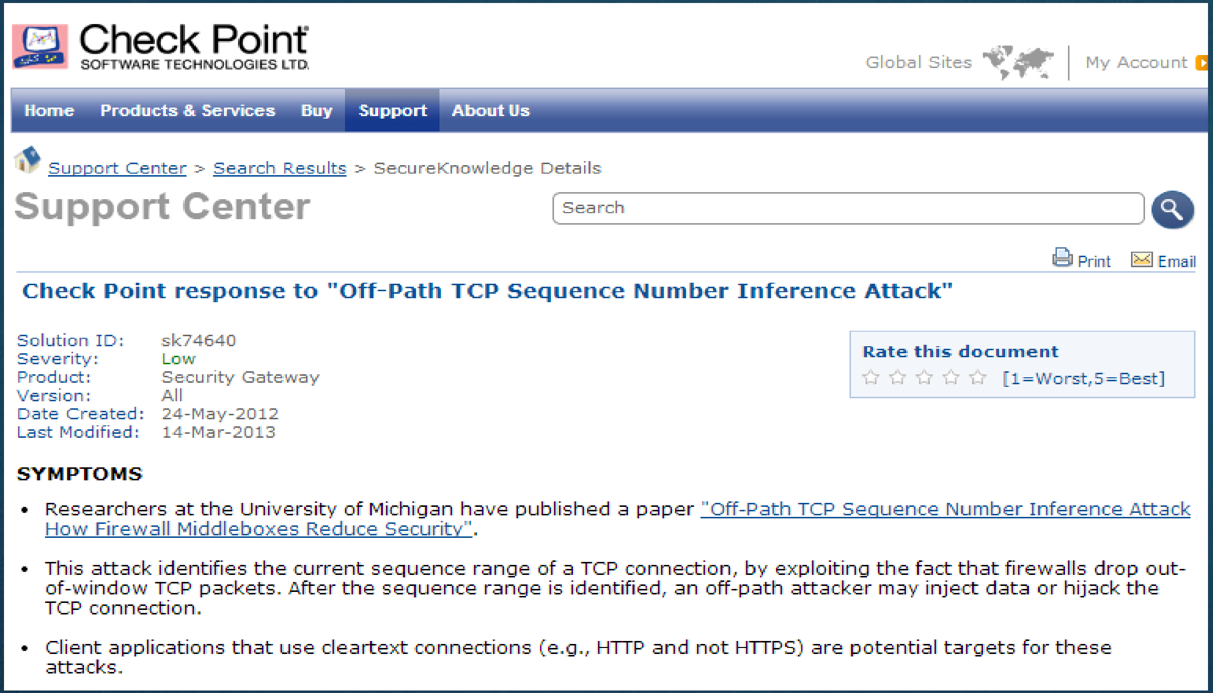
- Sequence-number-checking firewalls
  - Drop out-of-window (likely random or malicious packets)
  - Cut down resource waste and “**supposedly”** improve security
- However, we turn it into a side channel attack!



Off-Path TCP Sequence Number Inference Attack How Firewall Middleboxes Reduce Security, IEEE Security & Privacy (Oakland) 2012

Popularity of sequence-number-checking firewalls

- 33% of the 179 tested carriers deploy such firewalls
  - Vendors: Checkpoint, Cisco, Juniper
  - Could be used in other networks as well


Check Point SOFTWARE TECHNOLOGIES LTD.
Global Sites
My Account
Home
Products & Services
Buy
Support
About Us
Support Center > Search Results > SecureKnowledge Details
Support Center
Search
Print
Email

Check Point response to "Off-Path TCP Sequence Number Inference Attack"

Solution ID:	sk74640
Severity:	Low
Product:	Security Gateway
Version:	All
Date Created:	24-May-2012
Last Modified:	14-Mar-2013

Rate this document
☆ ☆ ☆ ☆ ☆ [1=Worst,5=Best]

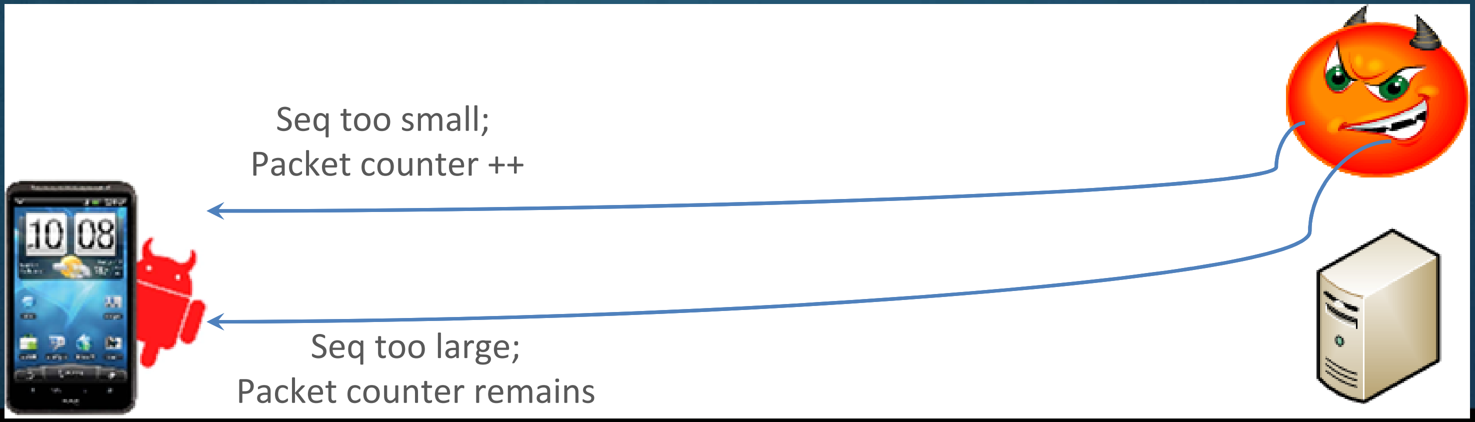
SYMPTOMS

- Researchers at the University of Michigan have published a paper "Off-Path TCP Sequence Number Inference Attack How Firewall Middleboxes Reduce Security".
- This attack identifies the current sequence range of a TCP connection, by exploiting the fact that firewalls drop out-of-window TCP packets. After the sequence range is identified, an off-path attacker may inject data or hijack the TCP connection.
- Client applications that use cleartext connections (e.g., HTTP and not HTTPS) are potential targets for these attacks.

Host packet counter side channels

- What if no firewall is deployed?

```
Tcp:
   157921111 segments received
   125446192 segments send out
   39673 segments retransmited
   489 bad segments received
   679561 resets sent
TcpExt:
   25508 ICMP packets dropped because they were out-of-window
   9491 TCP sockets finished time wait in fast timer
   Quick ack mode was activated 160830 times
```

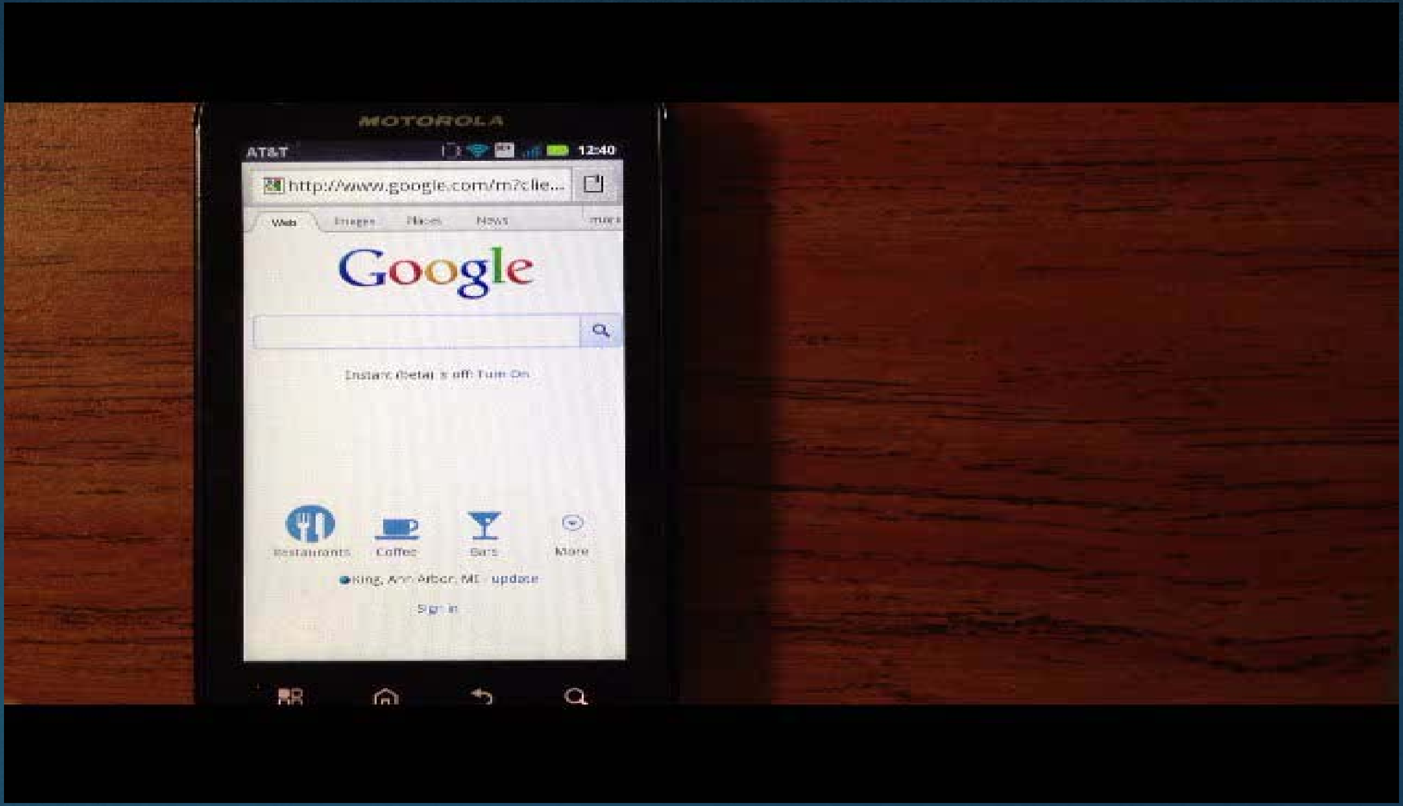


Seq too small;
Packet counter ++

Seq too large;
Packet counter remains

Collaborative TCP Sequence Number Inference Attack — How to Crack Sequence Number Under A Second, CCS 2012

Results



- Total inference time: ~1s
- Success rate: 62.5 – 97.5%

Impact

- Contacted security team in Linux, FreeBSD, Apple
  - Vulnerability partially patched in Linux
  - FreeBSD acknowledged this issue but cannot easily patch
- Backward compatibility is a challenge
- Side channels not always easy to patch

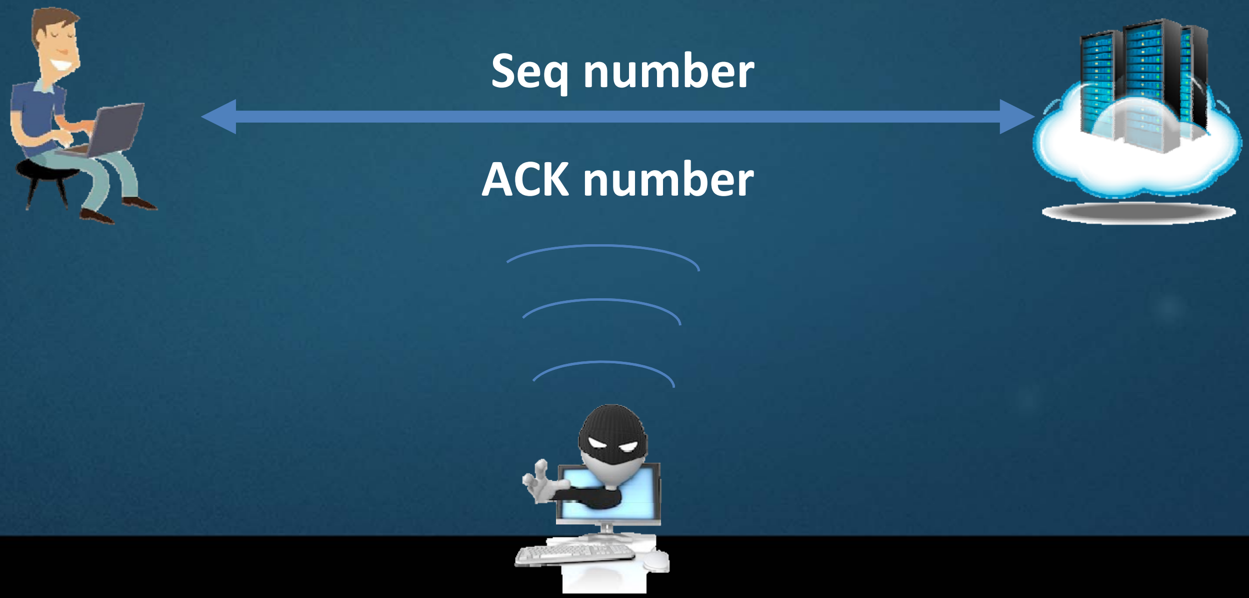
Outline

- Background on Off-Path TCP Exploits
- Off-Path TCP Exploits
  - Malware-assisted
  - Pure Off-Path [USENIX Security 16]
  - Unfixable WiFi timing

Even worse problems?

- Malware assisted
  - With and w/o firewall

- What now? Is there still a vulnerability even without help of malware?



Off-Path TCP Exploits: Global Rate Limit Considered Dangerous

- GeekPwn 2016 most creative idea award
- Facebook Internet Defense Prize Finalist

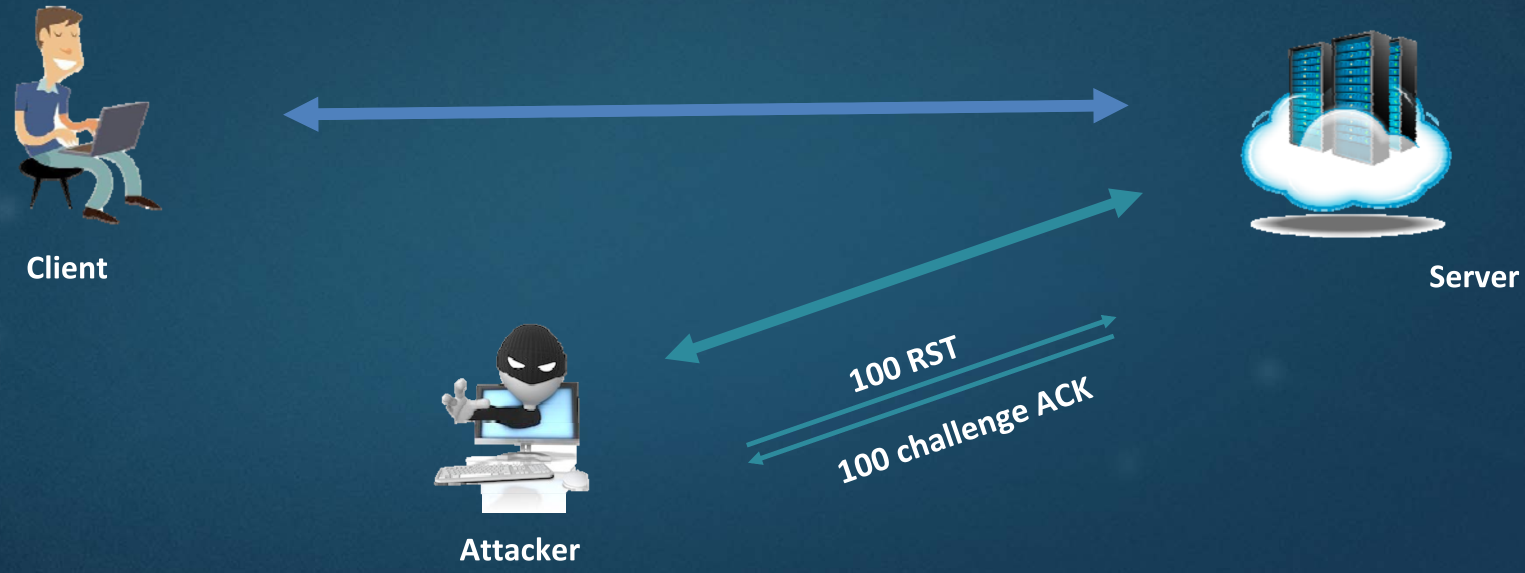
Yet another side channel

- Discovered a subtle TCP side channel vulnerability in Linux 3.6+
  - **CVE-2016-5696**

- Can be used towards:
  - Privacy breach (which service you are visiting)
  - TCP connection termination attack
  - Malicious data injection attack

Global rate limit (from RFC 5961)

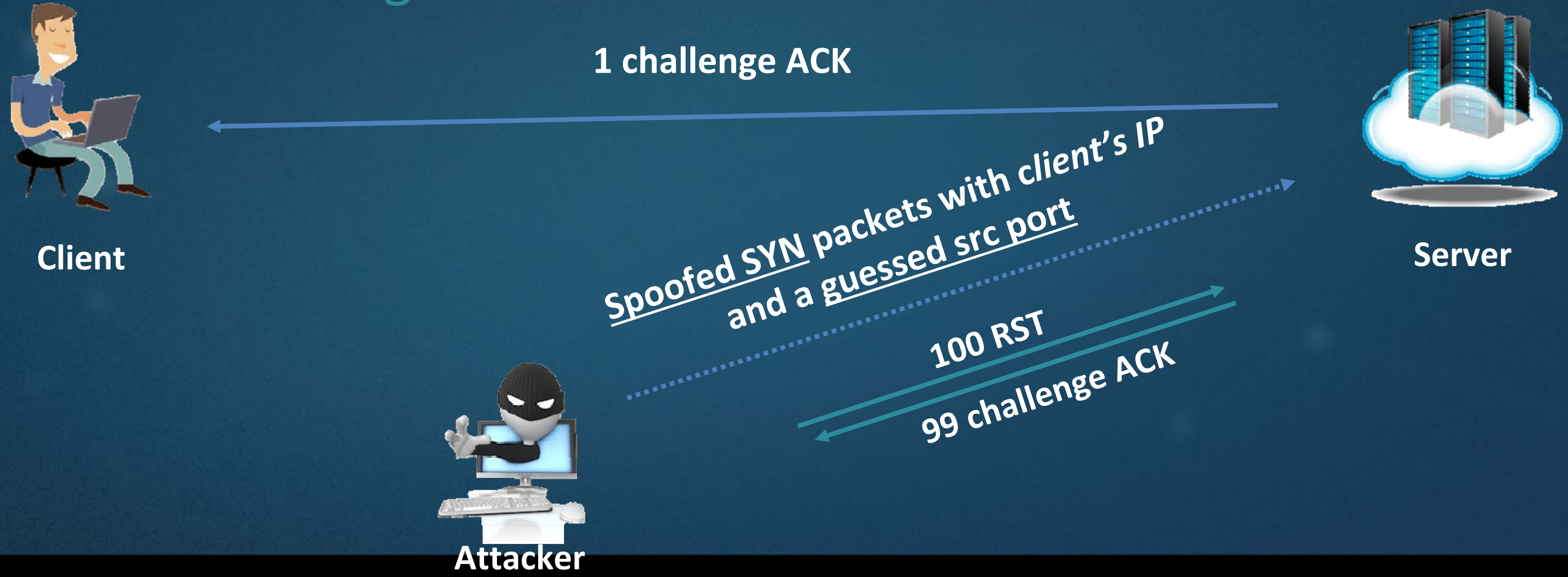
- sysctl_tcp_challenge_ack_limit: implemented in Linux 3.6+
  - *Global* limit of all challenge ACK per sec, ***shared across all connections***
  - Default value: 100 (**reset** per second)



Exploit the vulnerability

– Example: to guess correct client-port number

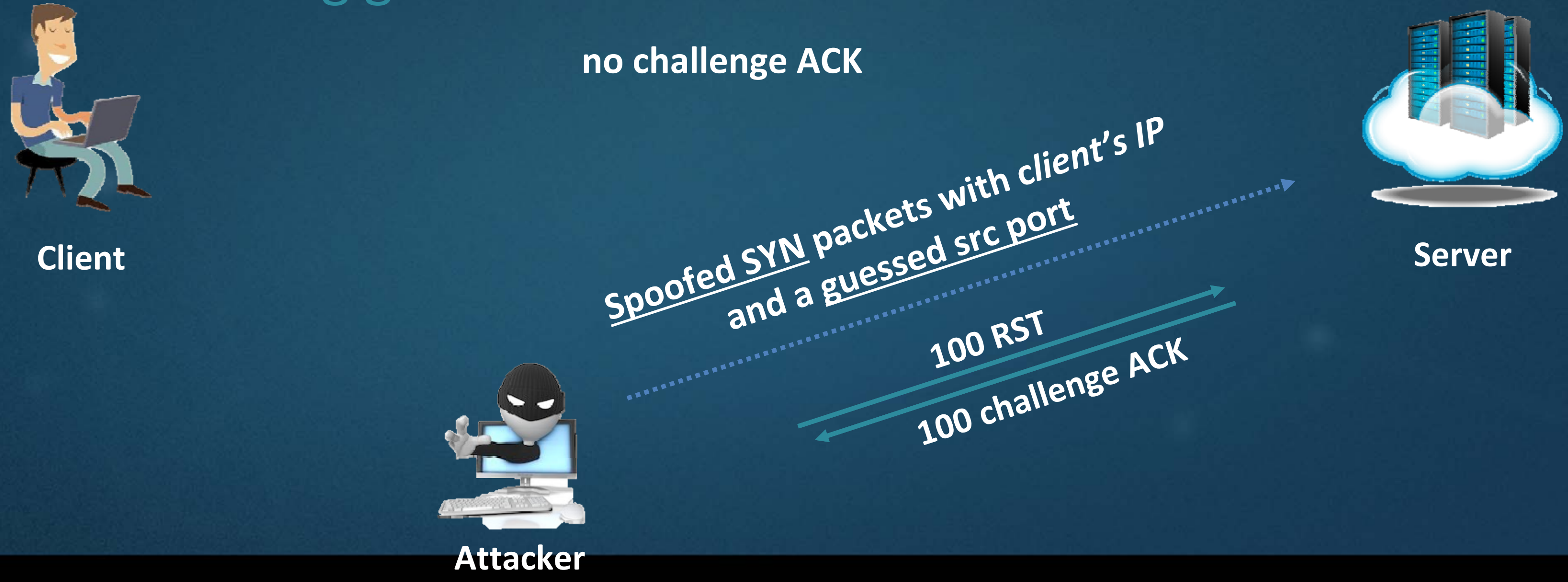
- If it's a correct guess:



Exploit the vulnerability

– Example: to guess correct client-port number

- If it’s a wrong guess:



no challenge ACK

Client

Server

Spoofed SYN packets with client’s IP and a guessed src port

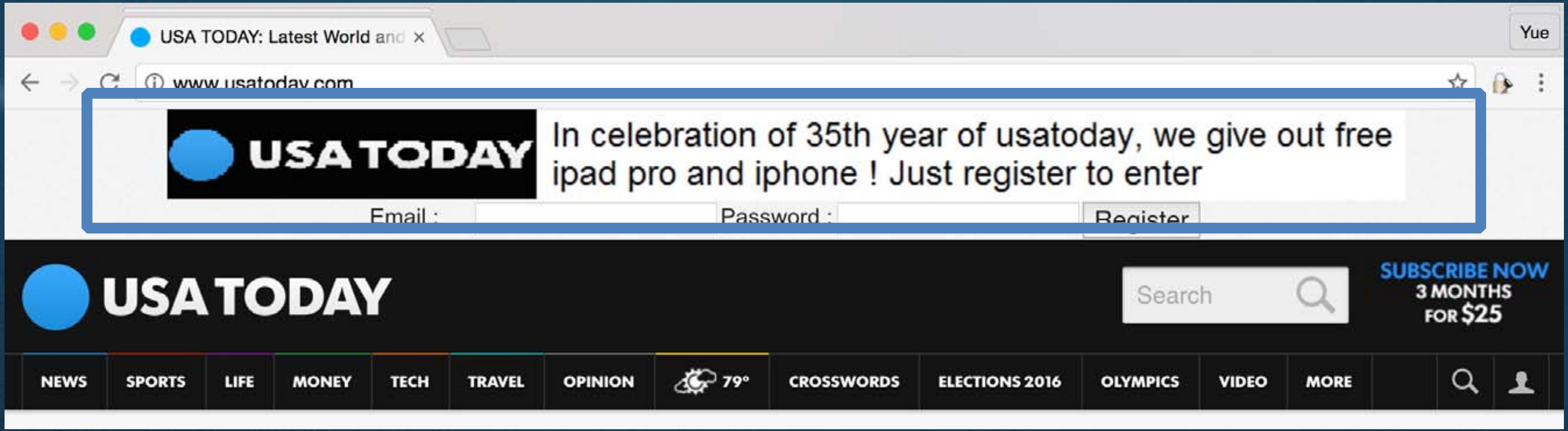
100 RST

100 challenge ACK

Attacker

Evaluation

- Existence of connection: <10 seconds
- Sequence number: 30 seconds
- ACK number: <10 seconds



USA TODAY In celebration of 35th year of usatoday, we give out free ipad pro and iphone ! Just register to enter

Email : Password : Register

Defense

- Our proposed defenses:
  - Add random noise to the channel (global challenge ACK rate limit)

    Adopted in Linux kernel 4.7 in July 2016 (within days)

  - Eliminate the side channel

    Adopted in Linux kernel subsequently

TCP specification (RFC 5961) amended

https://tools.ietf.org/html/draft-lvelvindron-ack-throttling-02

What now?

- So far …
- All side channels are software vulnerabilities
  - Relatively simple fixes
- Anything more fundamental?

Outline

- Background on Off-Path TCP Exploits
- Off-Path TCP Exploits
  - Malware-assisted
  - Pure Off-Path [USENIX Security 16]
  - Unfixable WiFi timing [USENIX Security 18]

Off-Path TCP Exploit: How Wireless Routers Can Jeopardize Your Secret

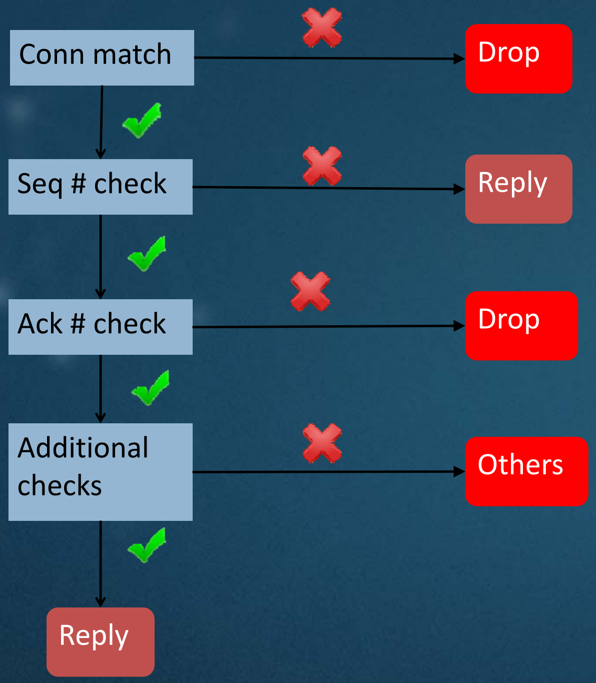
GeekPwn 2017 award-winning pwn

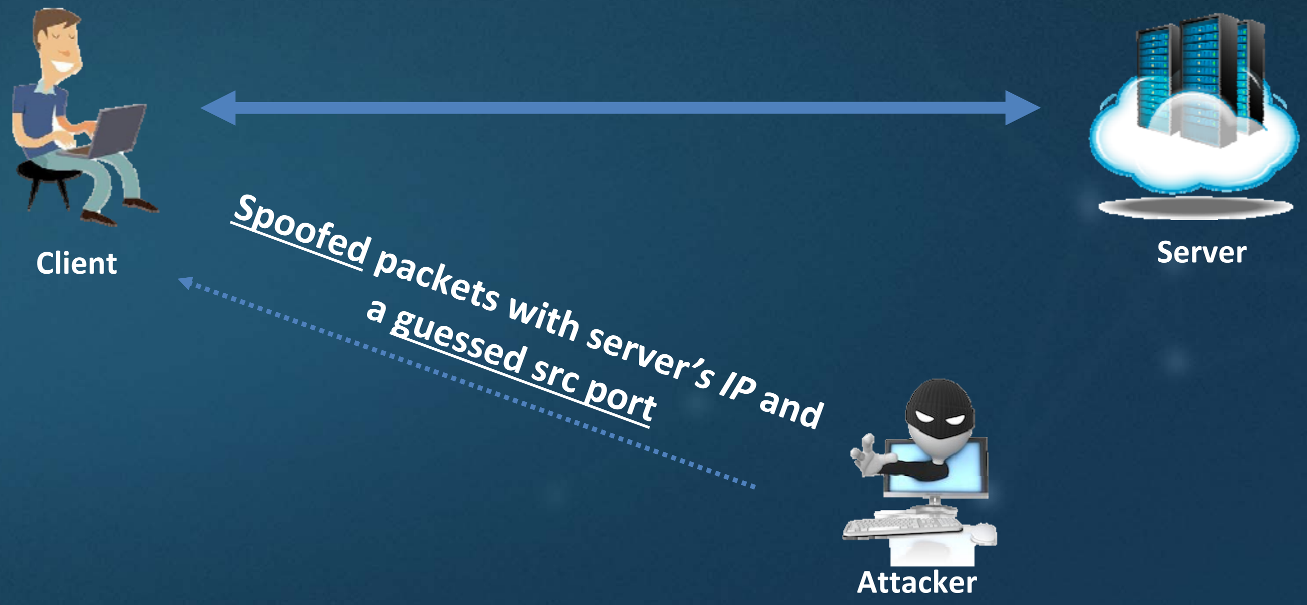
USENIX Security 2018

General research interest

GeekPwn 2017
GEEKPWN CYBERSECURITY CONTEST U.S. LAUNCH
GEEKPWN 2017 CYBERSECURITY GEEKS CONTEST
DATE: 2017.11.13
PAY TO THE ORDER OF Chen Weiteng, Qian Zhiyun
$ 15,000.00
Fifteen Thousand ++++++ DOLLARS
Attacker
Prepare To Get Wrecked
RESERVED

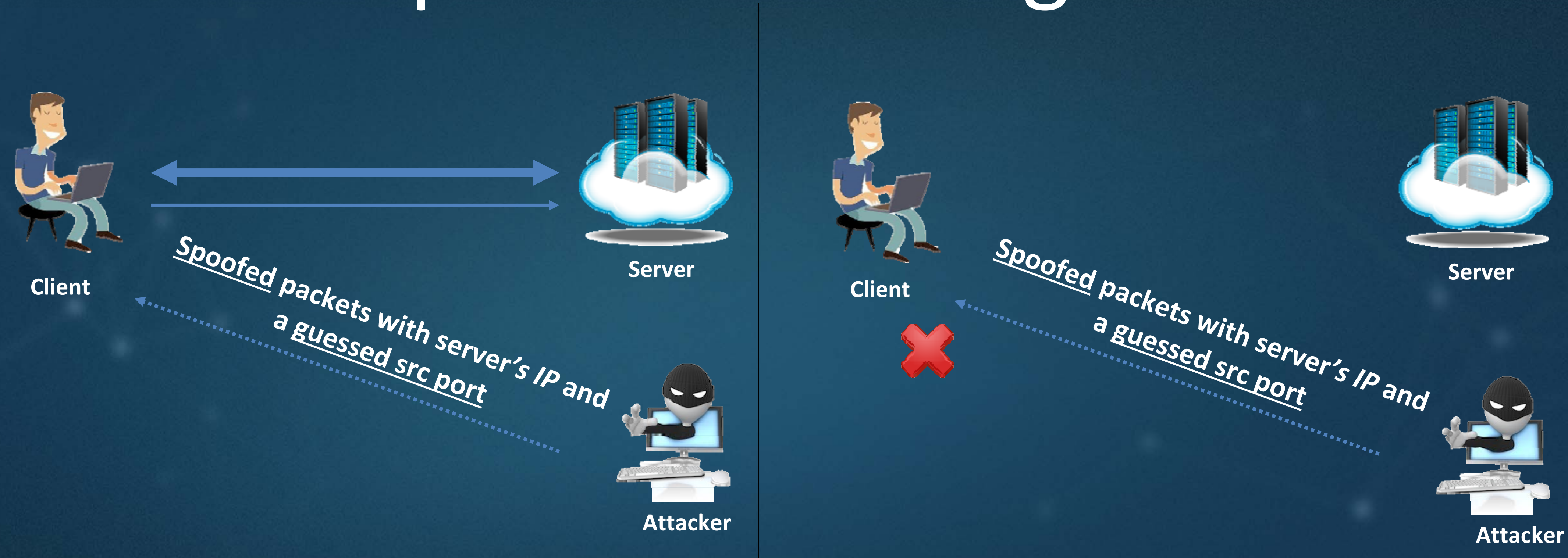
TCP packet receiving basics





Spoofed packets with server's IP and a guessed src port

TCP packet receiving basics


Has connection
No connection
Client
Server
Spoofed packets with server's IP and a guessed src port
Attacker
Client
Server
Spoofed packets with server's IP and a guessed src port
Attacker

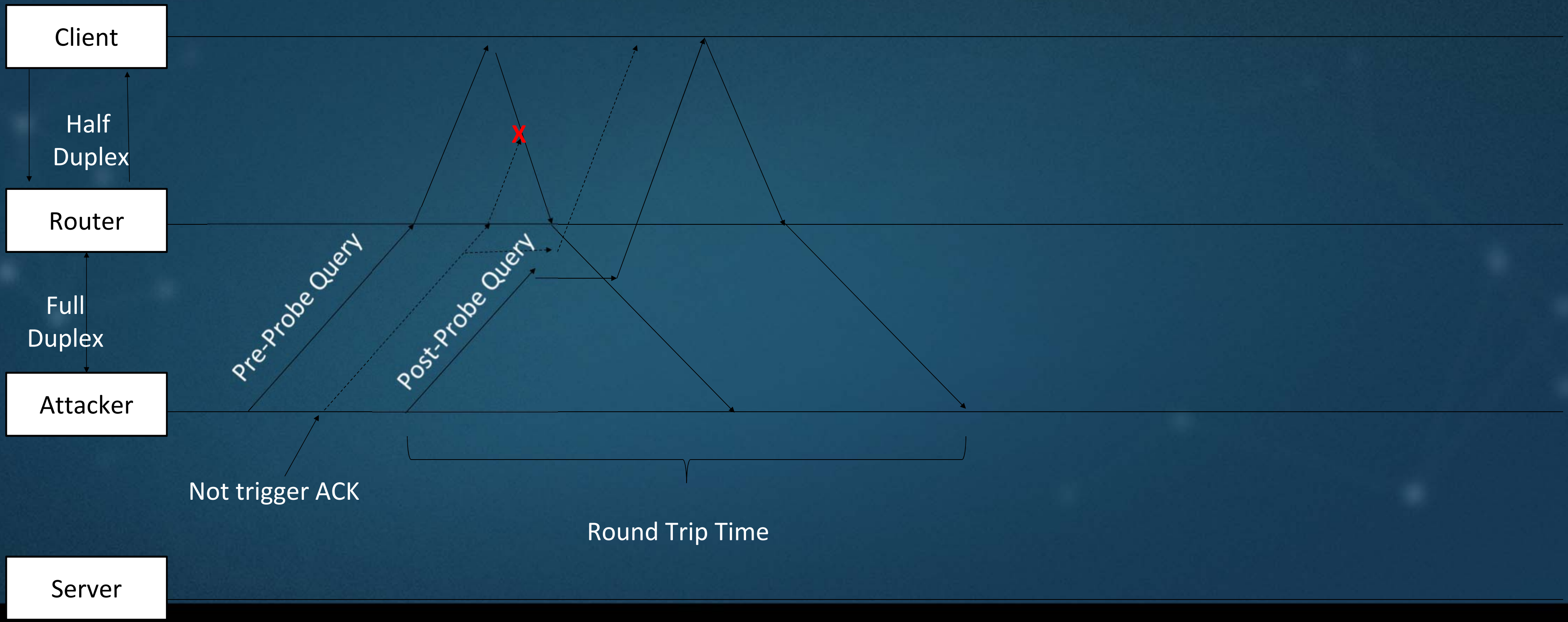
How can the attacker see the difference?

Timing channel

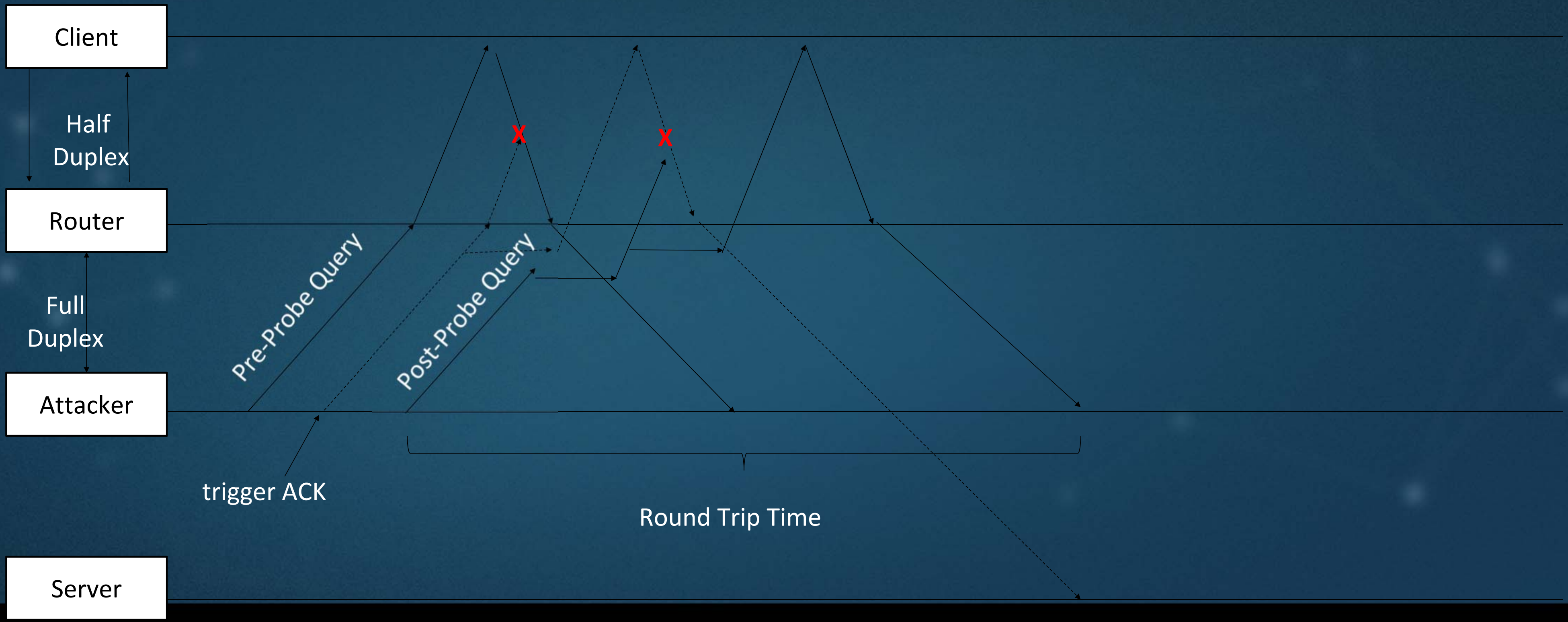
- Leaks information about whether the client has replied or not
  - Challenge: tiny amount of delay (on the order of us)

- **But it becomes visible in wireless!**
  - Root cause: Half-duplex
    - In all generations of WiFi and 802.11 technology

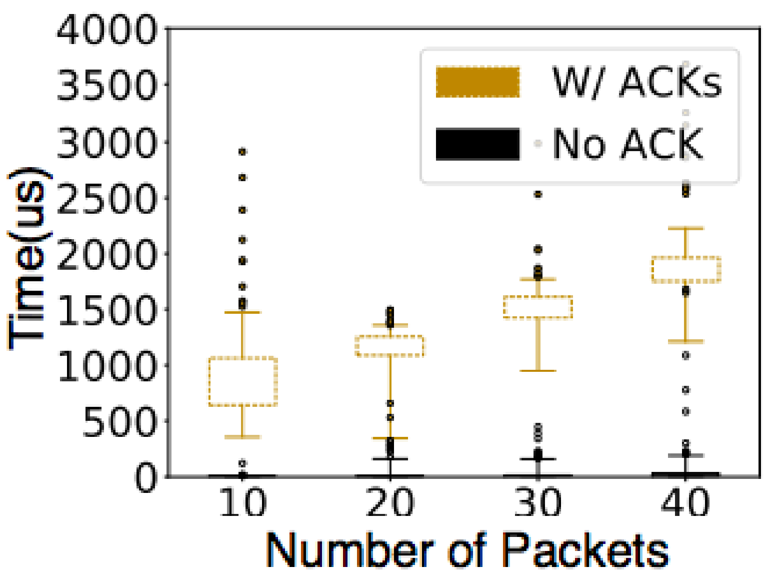
Timing difference – not trigger reply



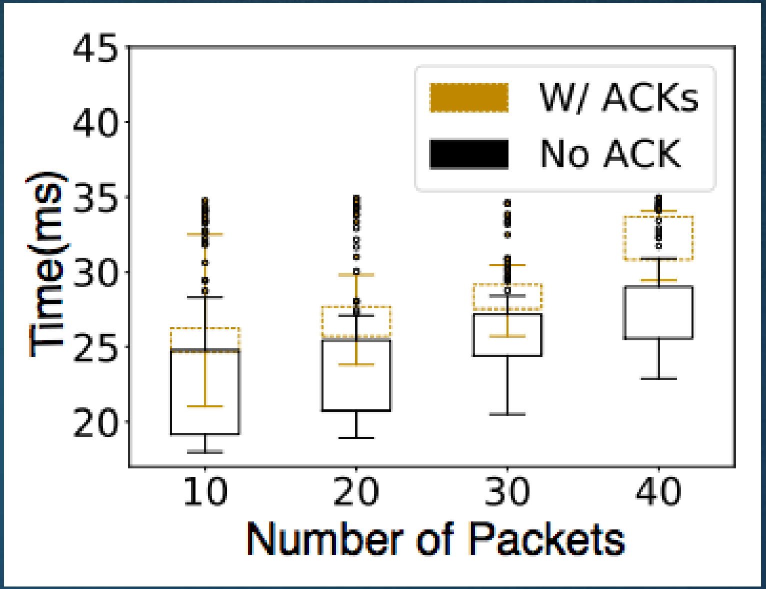
Timing difference – trigger reply



Timing difference – evaluation

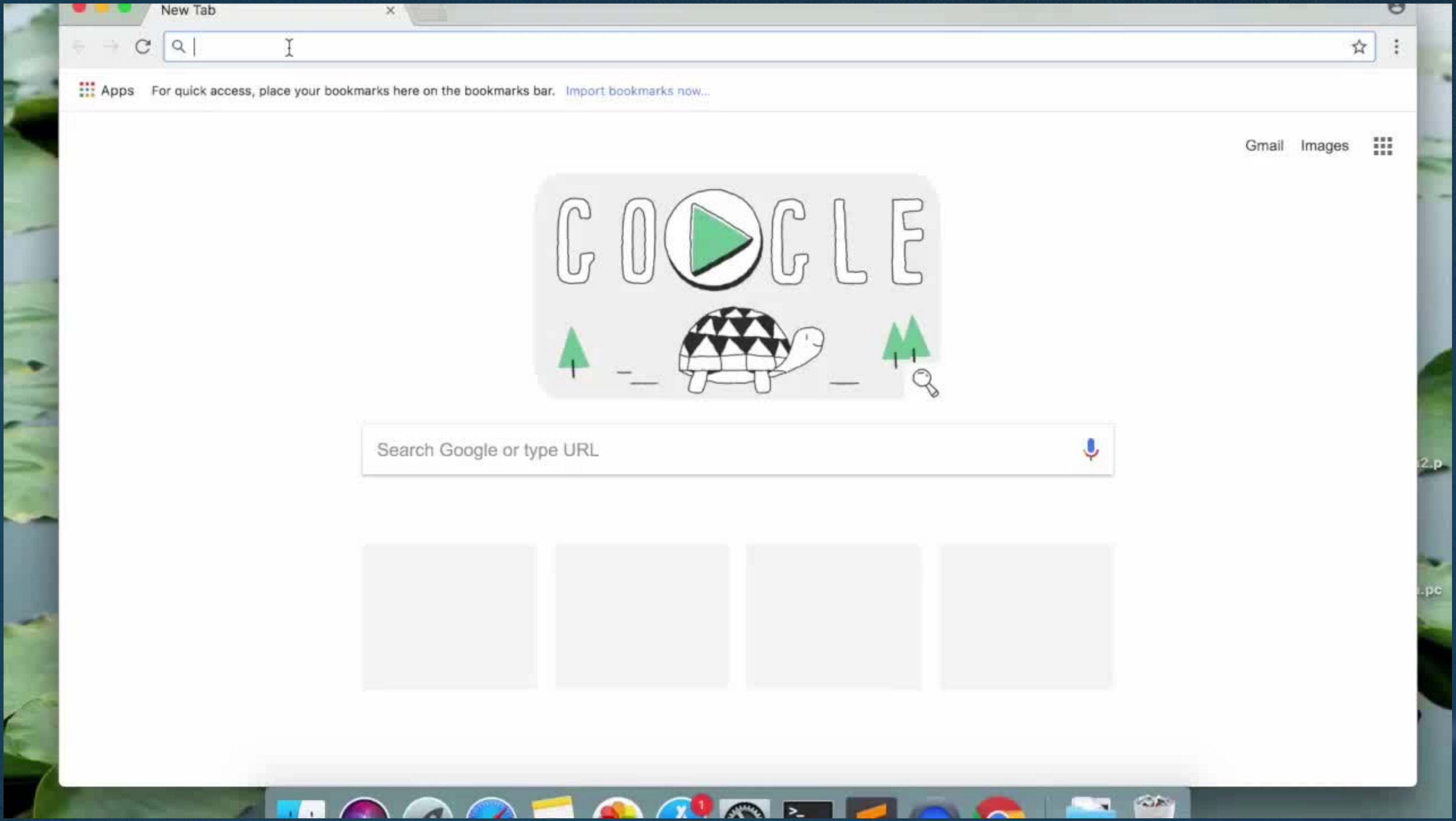


(a) Local experiment



(b) Remote experiment (RTT = ~ 20ms)

Web cache poisoning


New Tab
Apps
For quick access, place your bookmarks here on the bookmarks bar.
Import bookmarks now...
Gmail
Images
Search Google or type URL

Web cache poisoning

- Works against all major OSes + browsers:
  - Windows, macOS, Linux
  - Chrome, Firefox
- Success rate: 90%
- Time-to-succeed: 25s – 600s

Conclusion

- TCP side channel problems are real!
  - Huge impact on network security
  - Variety of side channels
  - Variety of exploitation scenarios and techniques
  - Difficult to fix at times

Thank you!
Q & A

Zhiyun Qian
zhiyunq@cs.ucr.edu

钱志云

United States Riverside City