

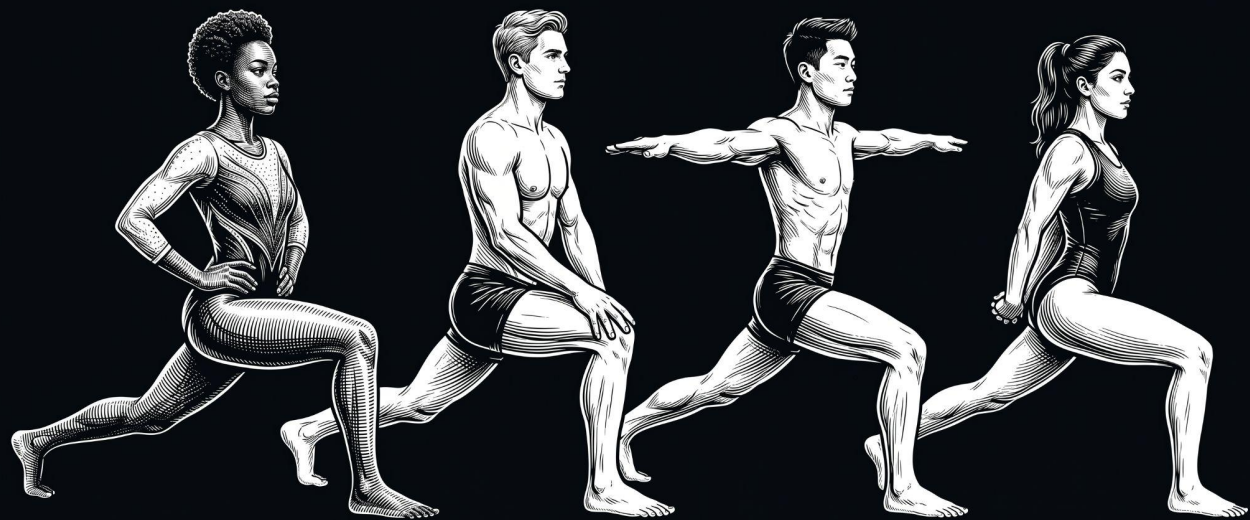
Time

for

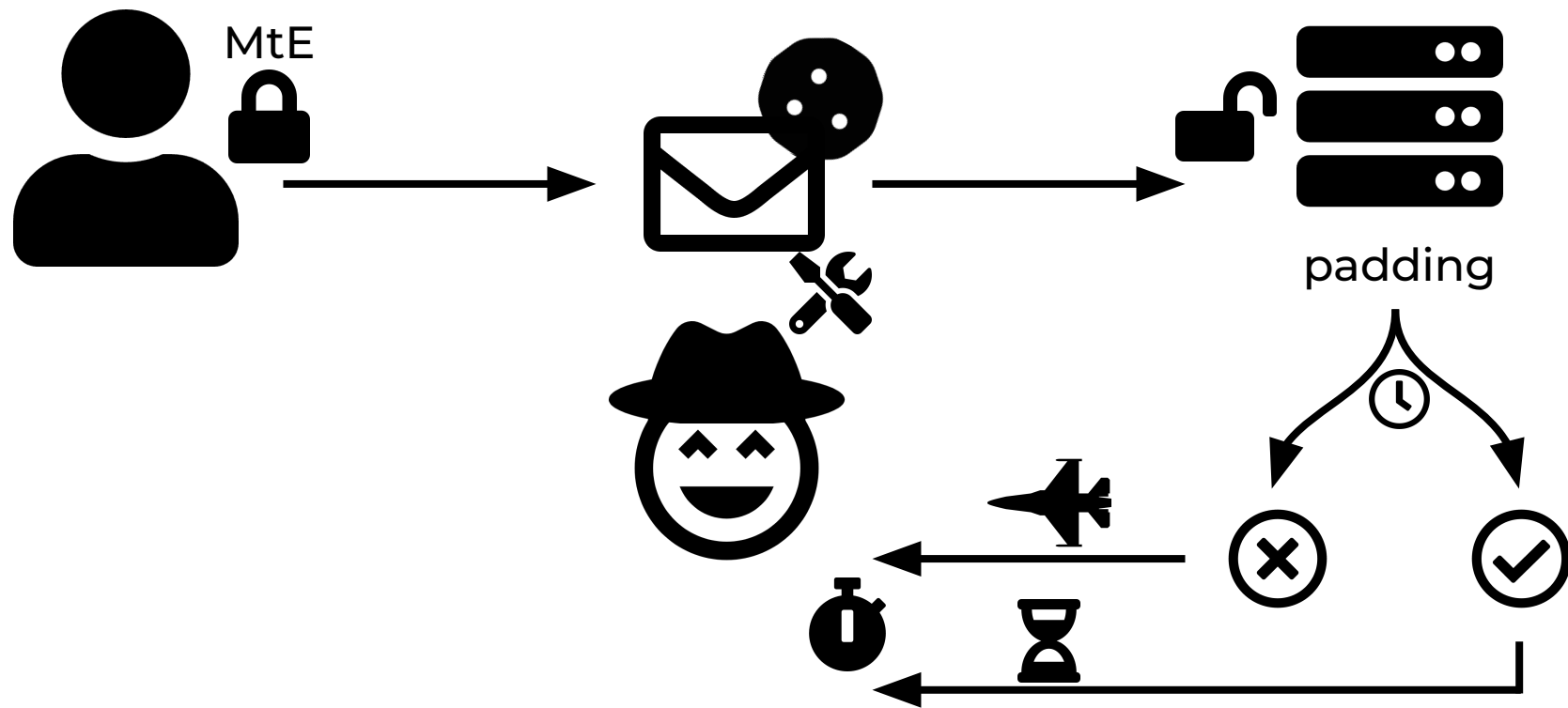
ACKrobotics

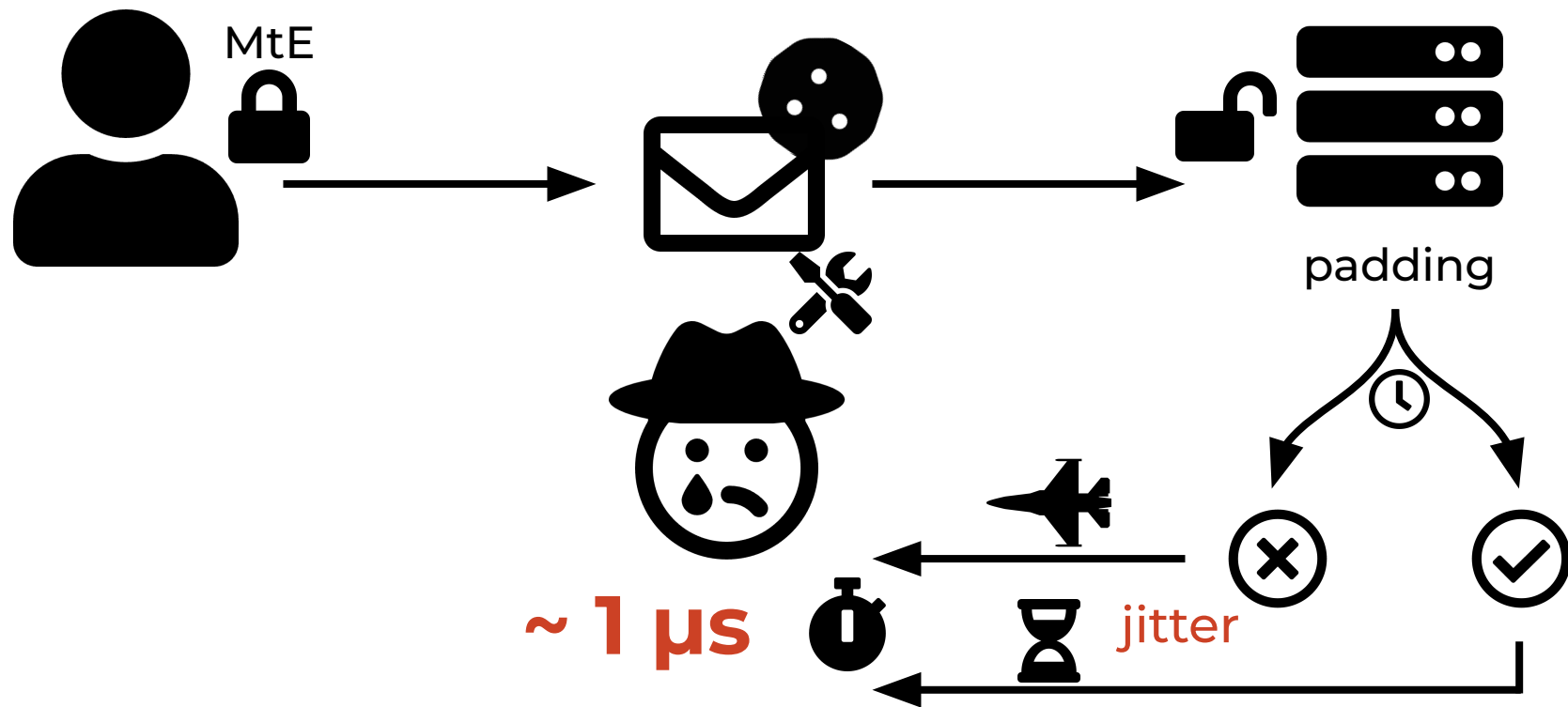


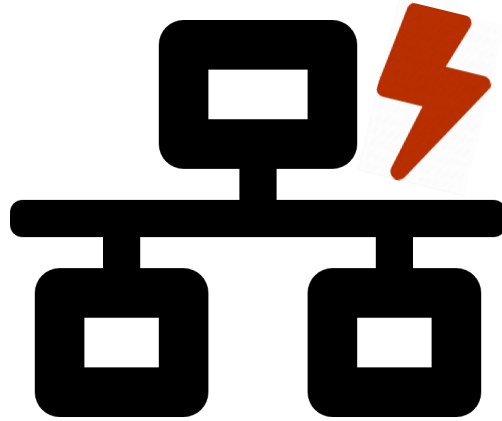
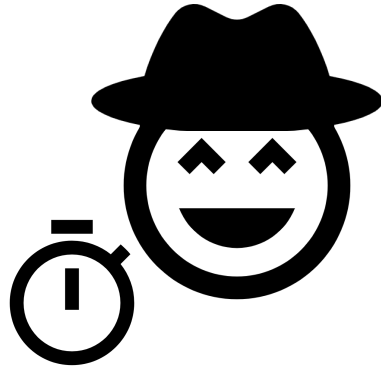
Leveraging TCP Timestamps to Improve Remote Timing Attacks



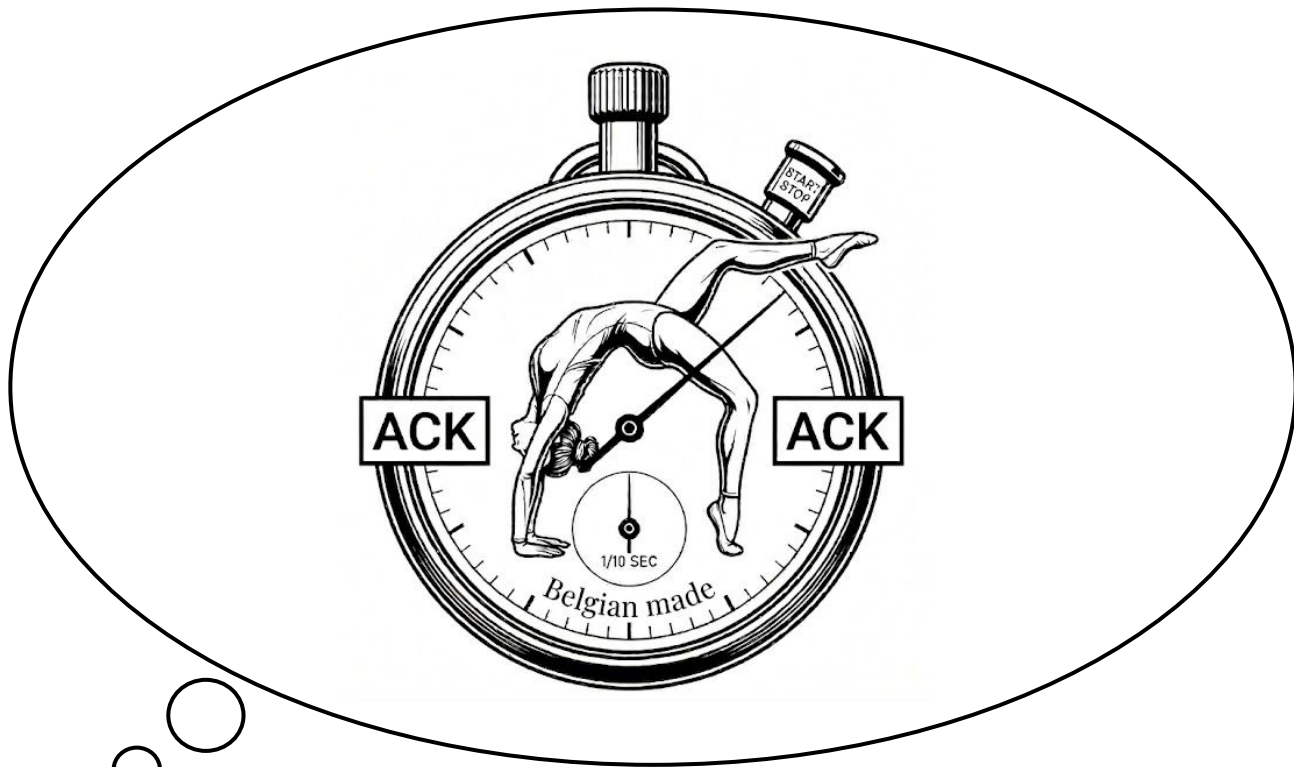
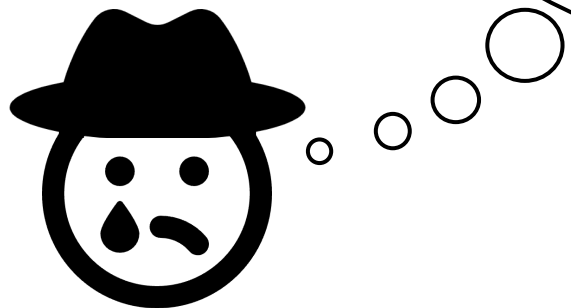
Lucky 13











The ACKrobats



Vik Vanderlinden

PhD @ KU Leuven



Tom Van Goethem

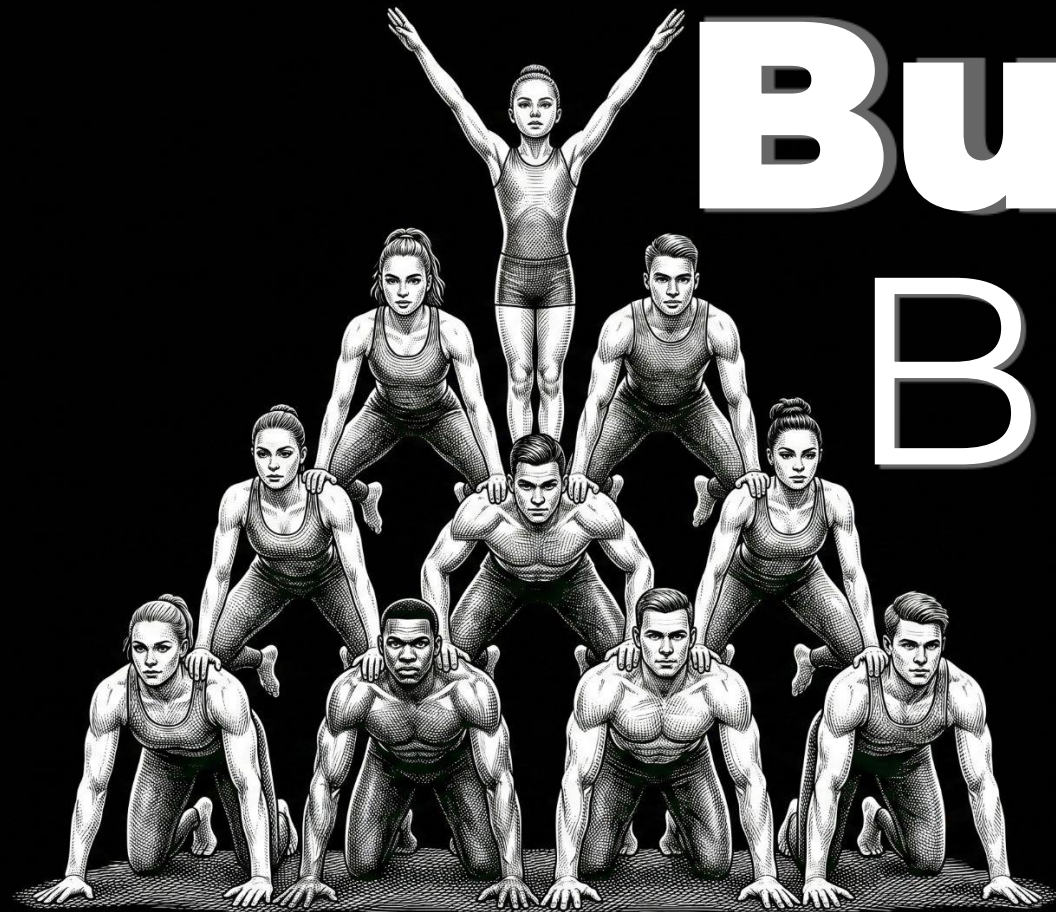
SWE @ Google
PhD @ KU Leuven



Mathy Vanhoef

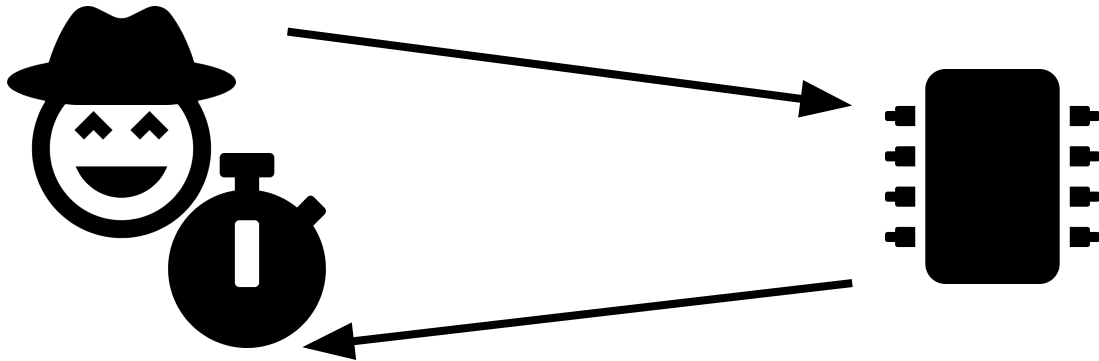
Prof. @ KU Leuven

Building Blocks



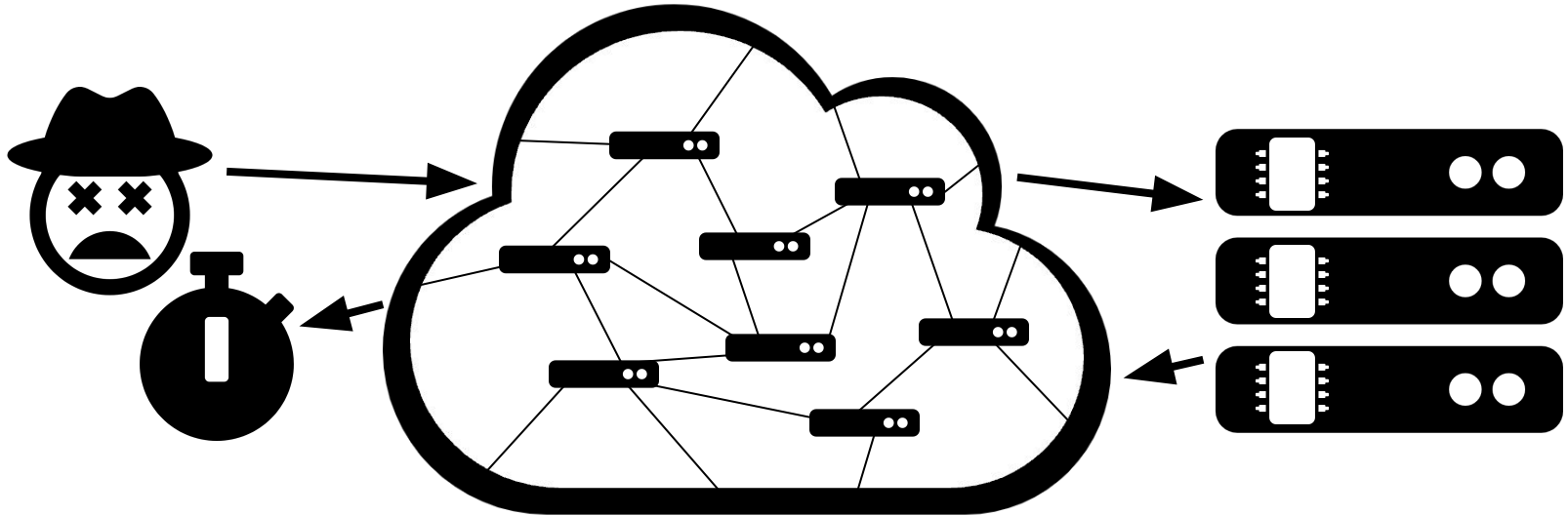
Timing Attacks

1. Measure execution time
2. Infer secret based on time

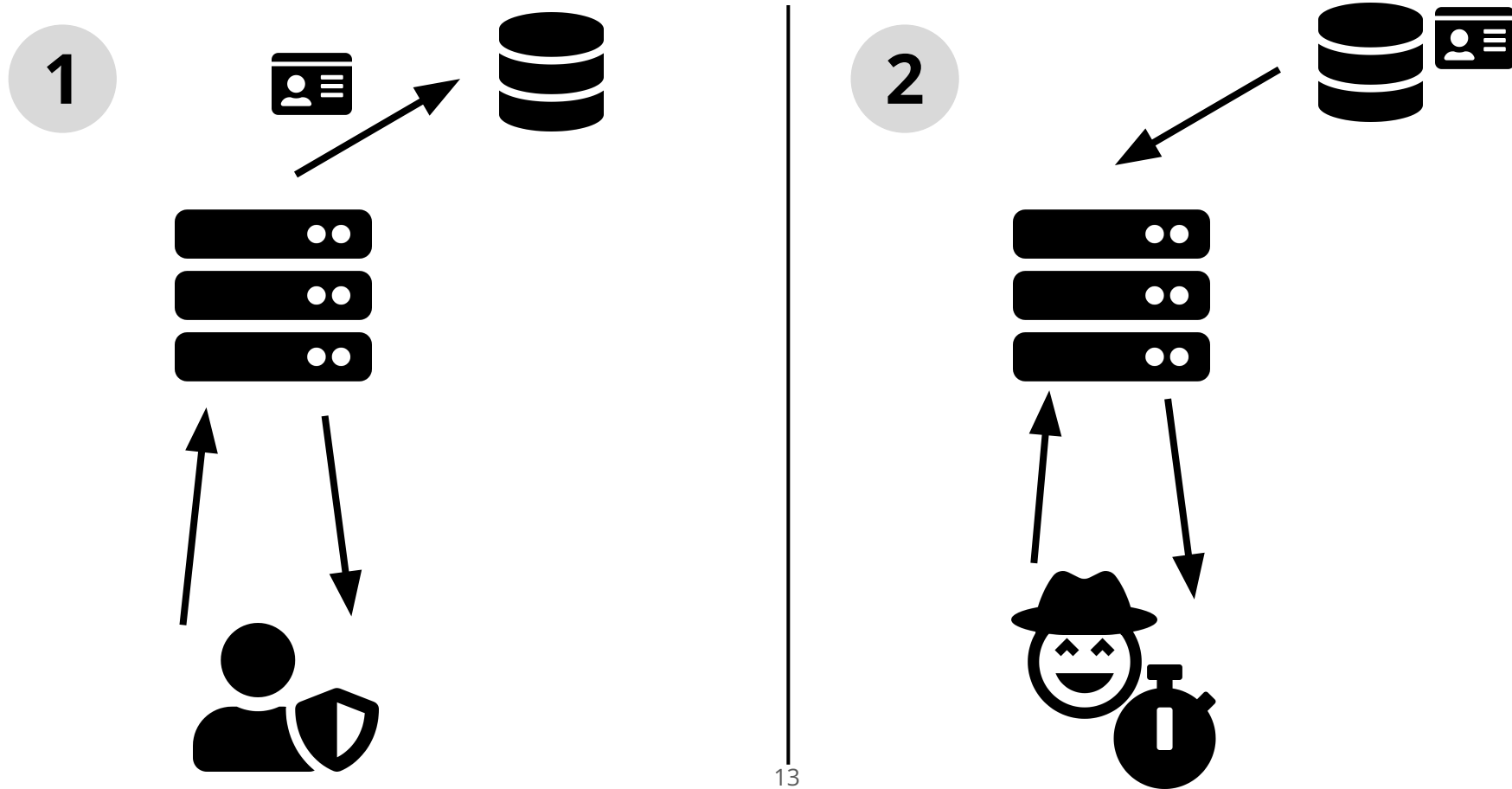


Remote Timing Attacks

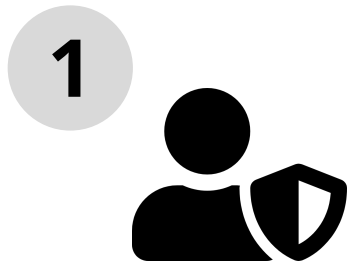
1. Send request
2. Cry because of jitter 🥹
3. Measure response time



Example Remote Timing Attacks^[2]



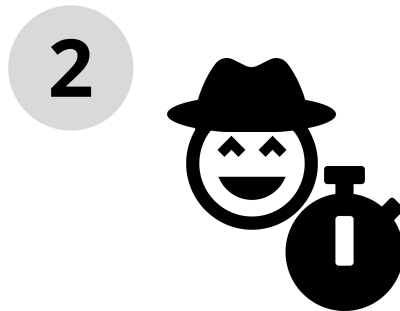
Example Remote Timing Attacks



Joins private group

Reports security vulnerability

...



Infers group membership

Leaks contents of report

...

Improving Remote Timing Attacks



Date Header^[5] (HTTP)



Server-Timing Header^[4] (HTTP)

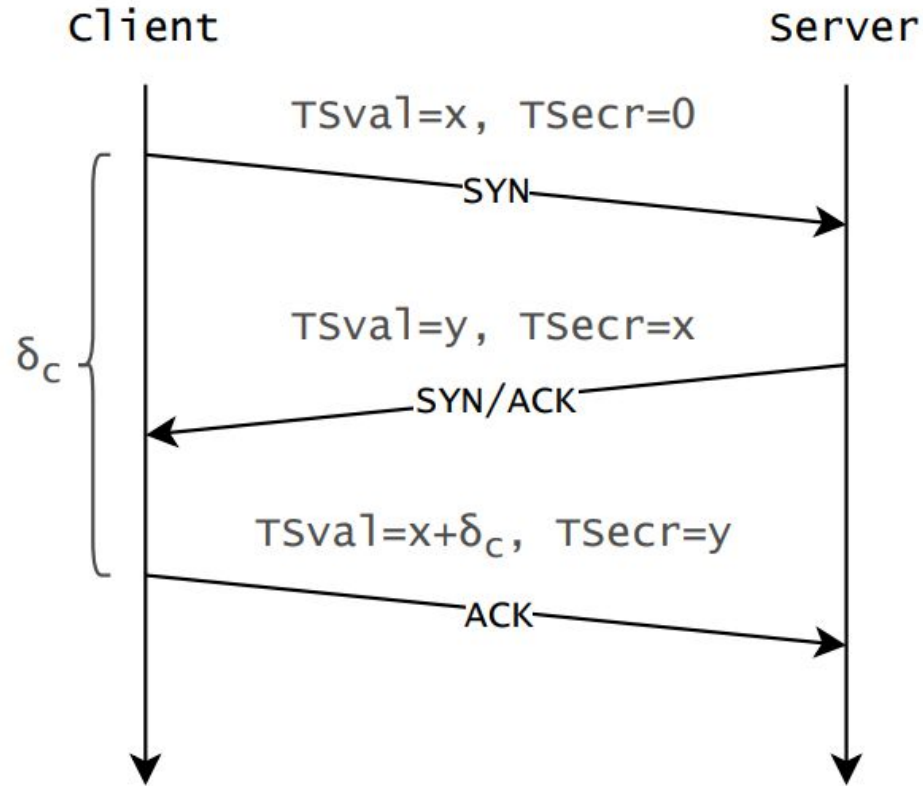


Timeless Timing Attack^[1] (TCP)

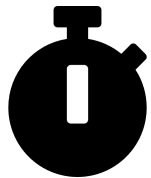
Enter: TCP Timestamps



What are TCP Timestamps



Why TCP Timestamps Exist



Improve RTTM^[12]

Congestion Control



PAWS^[12]

High-bandwidth applications (DC)



LEDBAT^[14]



**Reducing
Time-Wait state**^[13]

Evolution of TCP Timestamps

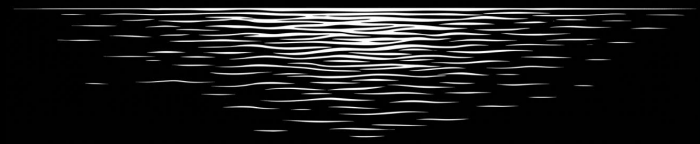
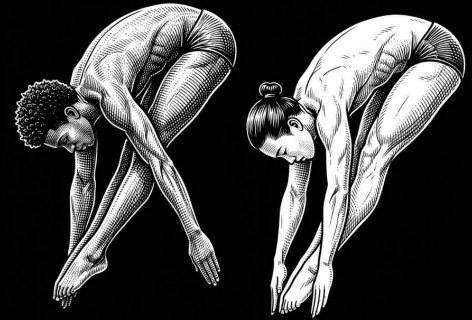
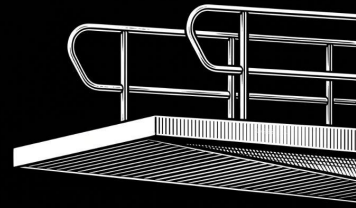
2021, 2020 - μ s timestamp accuracy proposals^[8, 9]

2023 - μ s timestamps implemented in kernel^[10]

2023 - μ s timestamps can be enabled using ip route option^[10]

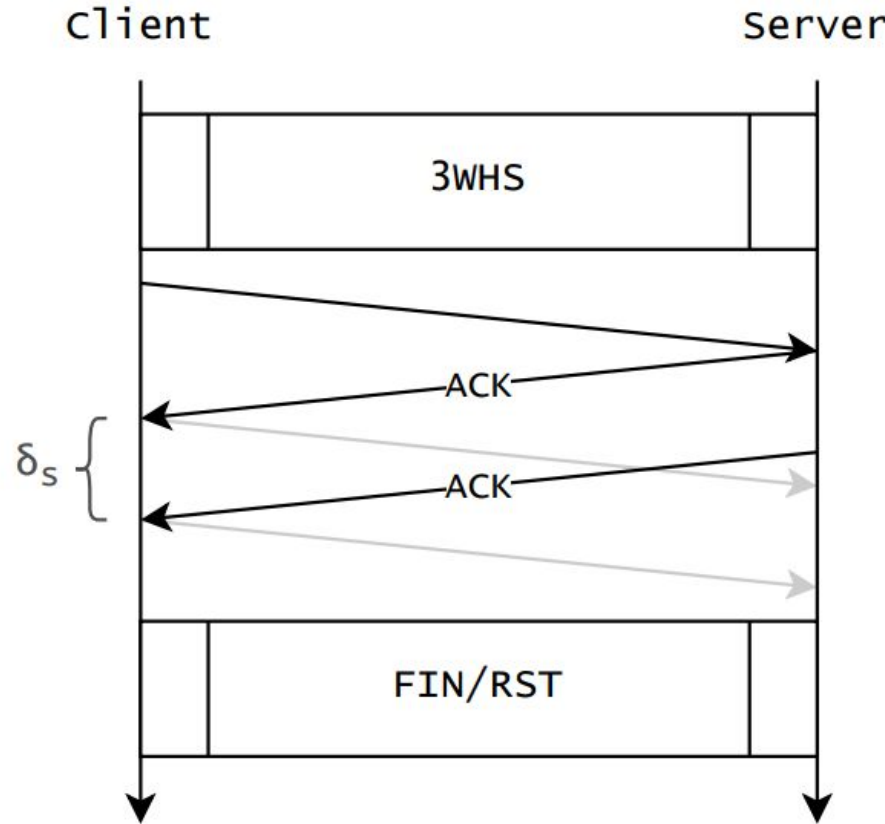


Attack Detail s



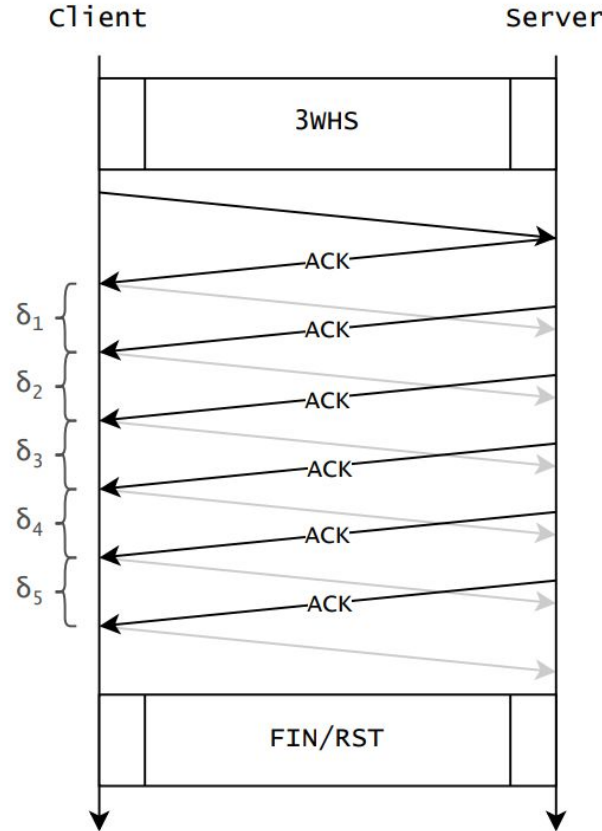
Timing Attacks **Leveraging** TCP Timestamps

Basic Attack

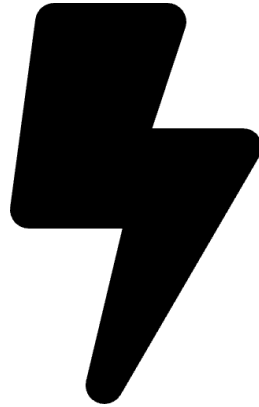


Timing Attacks **Leveraging** TCP Timestamps

Runtime
Multiplication
Enhancement

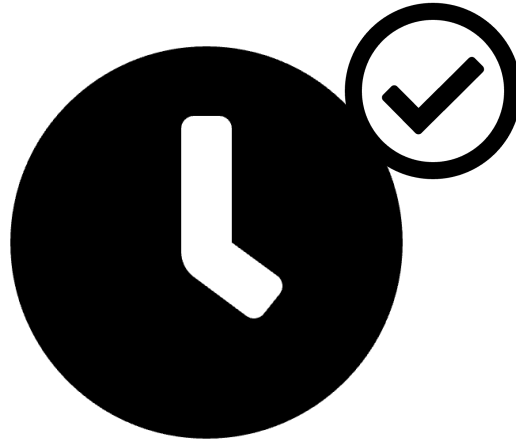


Attack Preconditions



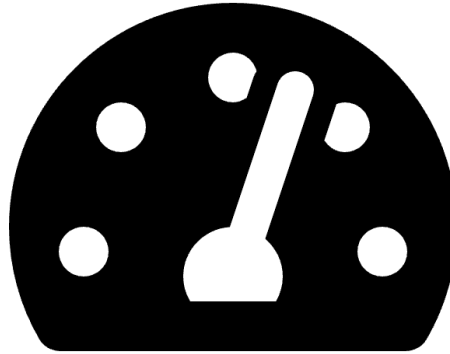
>880k Servers tested

Attack Preconditions



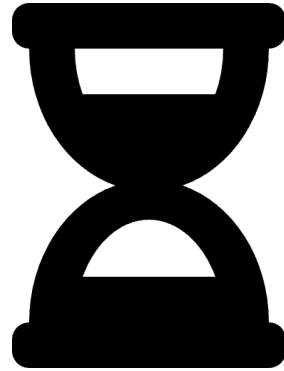
>88% TCP Timestamps enabled

Attack Preconditions



>99% Immediate ACKnowledgement

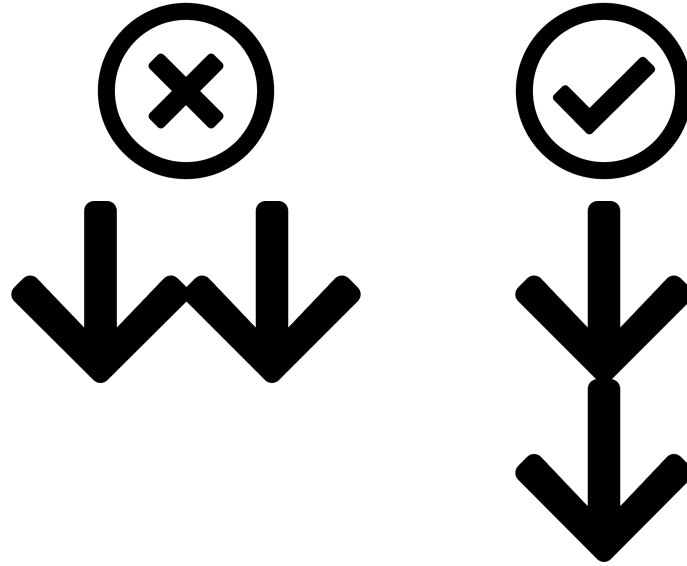
Attack Preconditions



>95% Persistent connections

HTTP Specifics

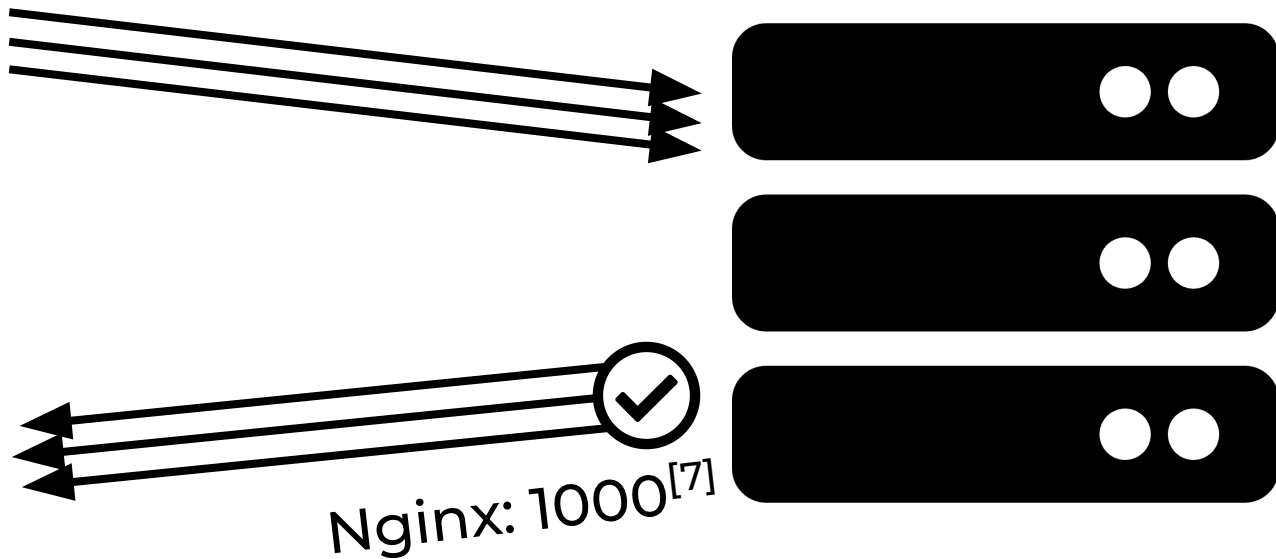
Attack Preconditions



Support for non-concurrency (e.g. HTTP/1.1)

Coalescing Practicalities

or: how to get all those requests to the server?



Coalescing Practicalities

or: how to get all those requests to the server?

TCP segment



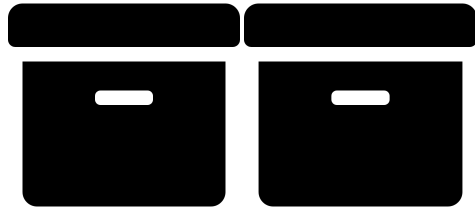


Coalescing Practicalities

or: how to get all those requests to the server?

TCP segment size

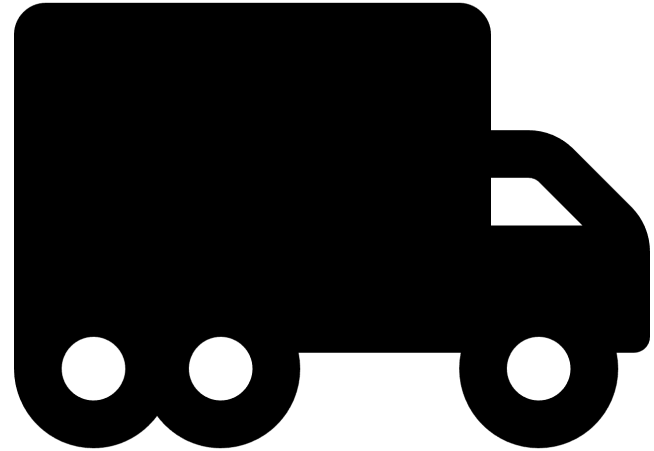
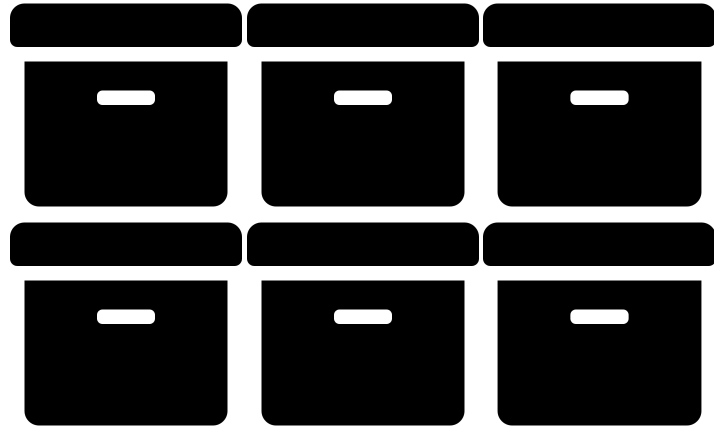
(1,5KB; MSS)



Coalescing Practicalities

or: how to get all those requests to the server?

TLS frame size (16KB)

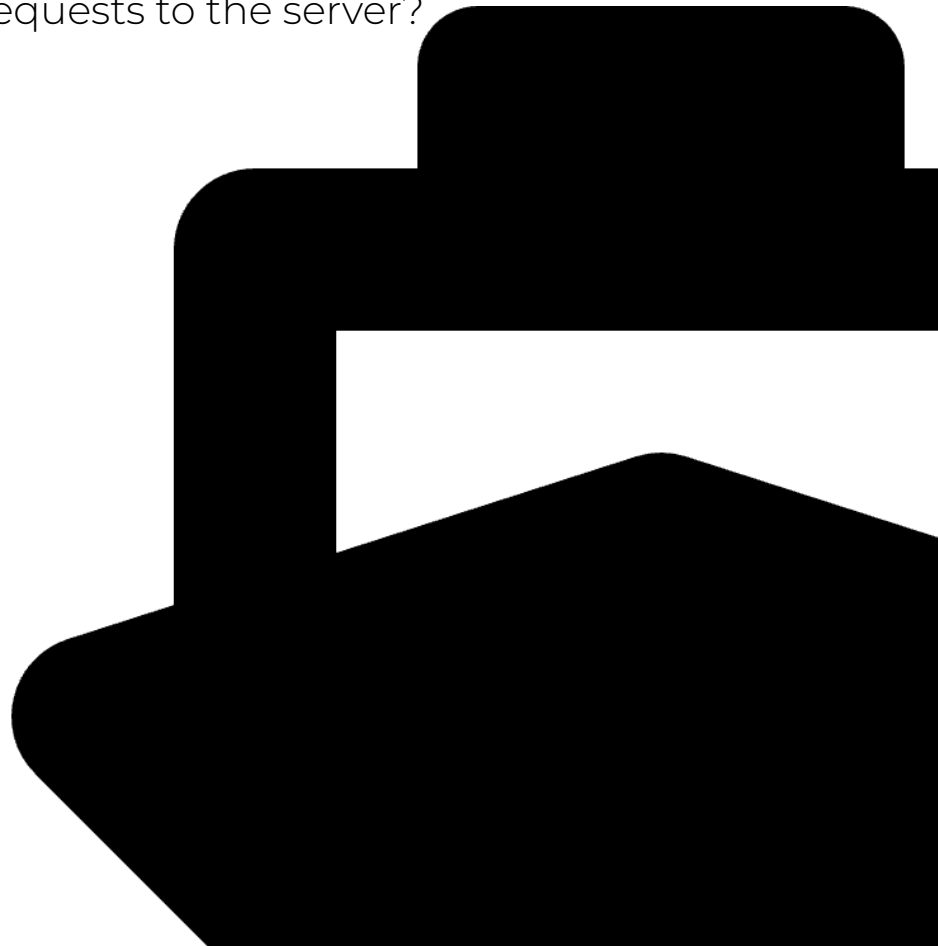
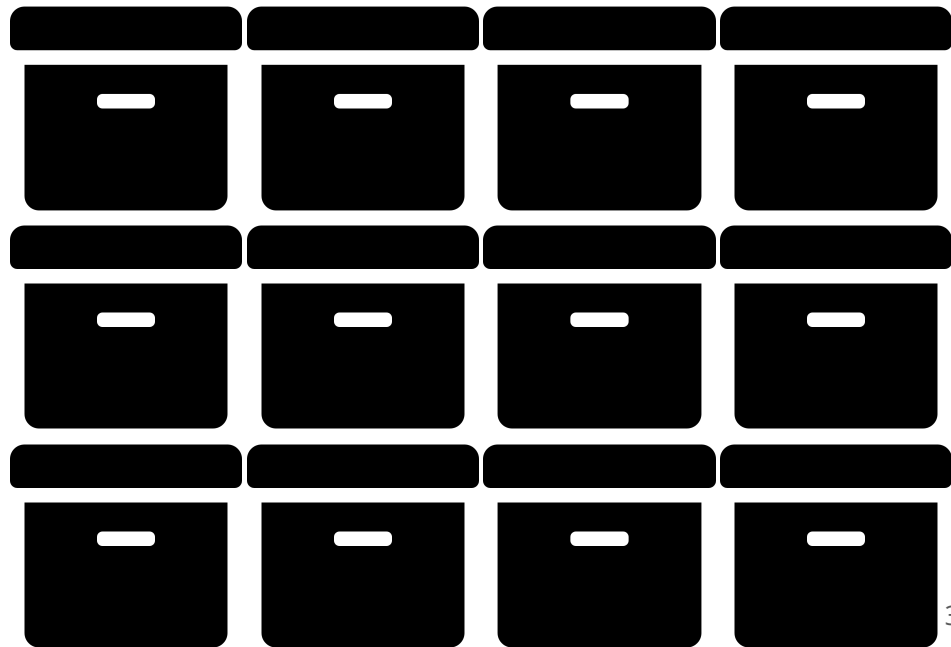


Coalescing Practicalities

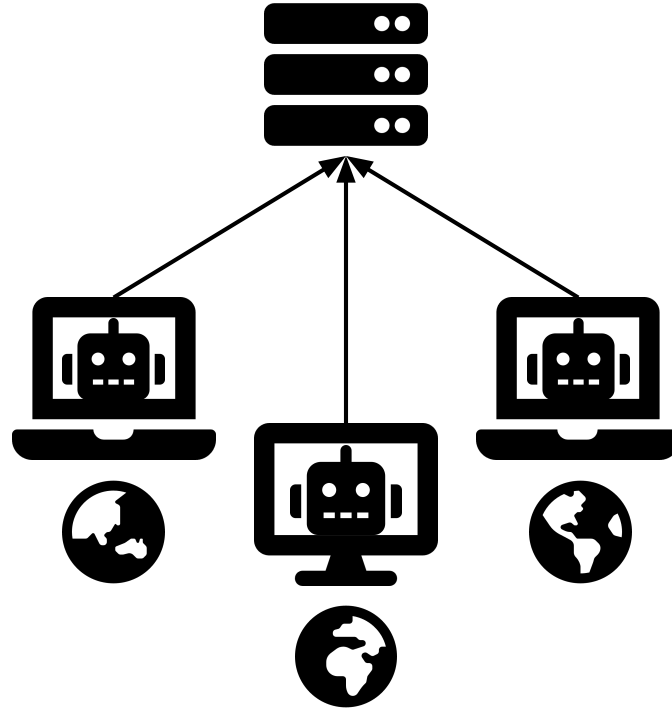
or: how to get all those requests to the server?

Out-of-order TCP segments

(6 MiB default @ AWS ubuntu)



Distributed Attack



Attack Performance

ms timestamps



25 μ s $\rightarrow$ 5 μ s



25 μ s: >10k requests $\rightarrow$ 200 requests

μ s timestamps



25 μ s $\rightarrow$ 750 ns

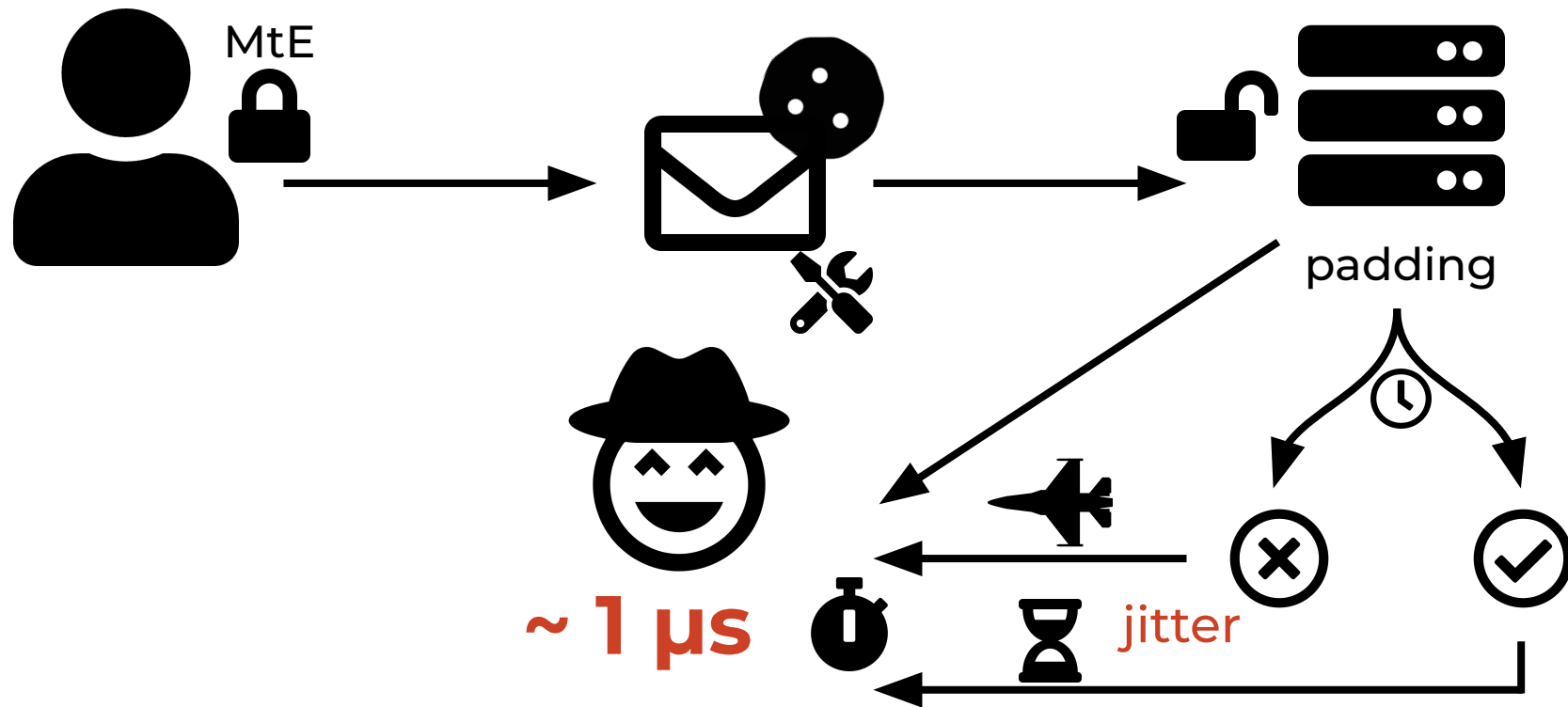


Case Studies





Case Studies: **Lucky 13**



Case Studies: **Lucky 13**

Client: **UK**

Server: **USA**

Timestamps: **μ s**

Target: **Embedded Linux Library**

CVE-2025-32998 assigned

transatlantic



Case Studies: **Lucky 13**

Distinguish 0x00 and 0x01 byte

150ns with 50k requests

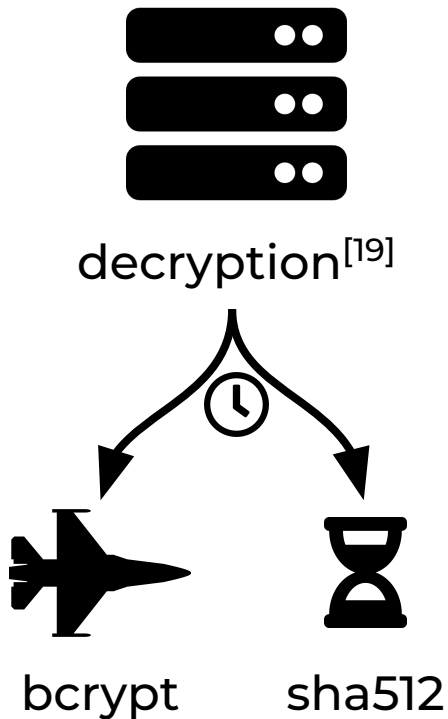
Responsibly disclosed

no response received



Case Studies: OpenSSH

User enumeration
OpenSSHd pre 7.3

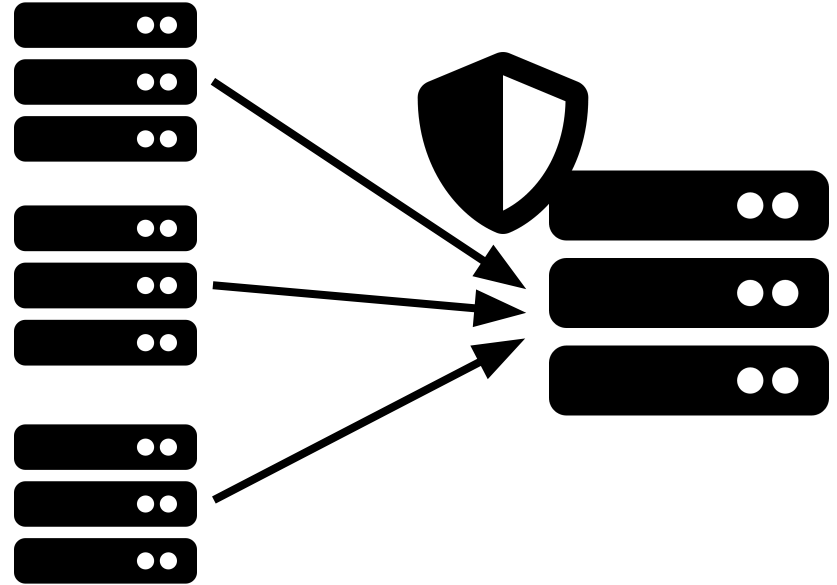


Case Studies: OpenSSH

Multiple Clients

Timestamps: **ms**

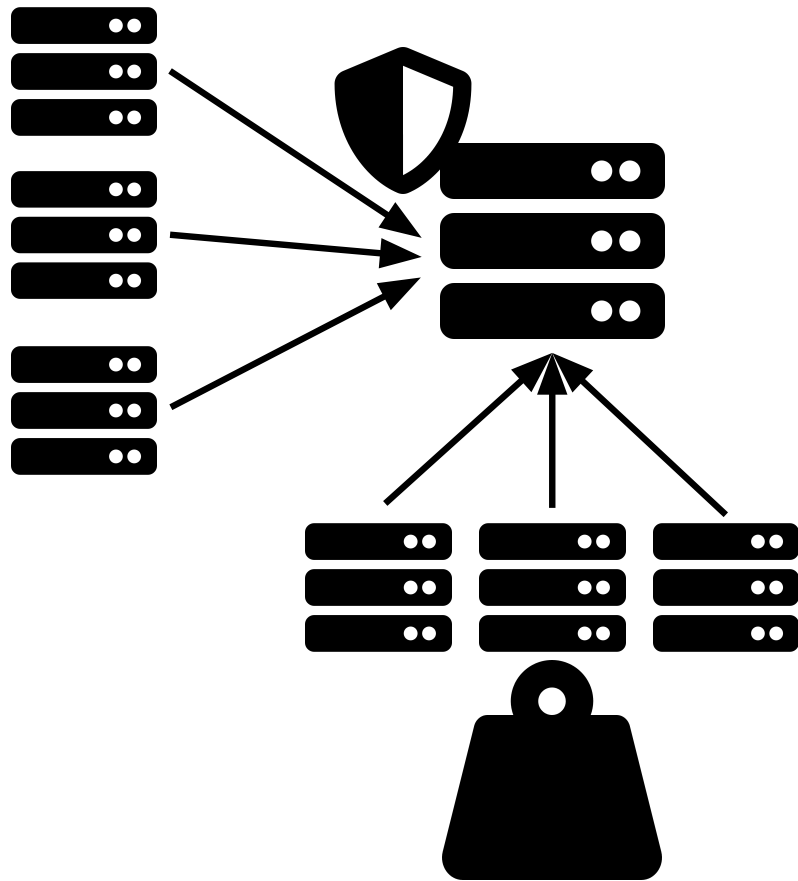
Distributed evaluation



Case Studies: OpenSSH

Artificial load: **>900 req/s**

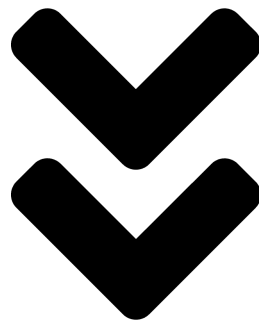
Results: **Unchanged**



Defense s



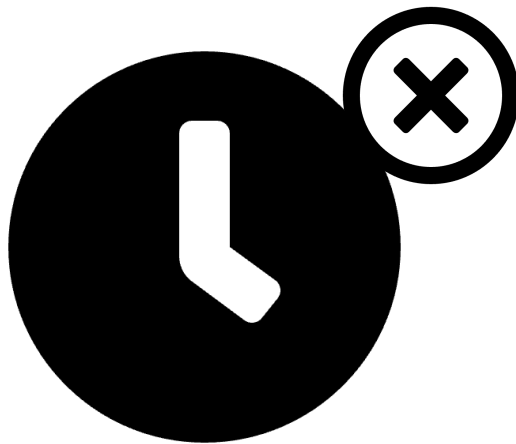
Defenses



Send TCP Timestamps **less often** (e.g. [11])

Only hinders the attack slightly

Defenses



Disable TCP Timestamps

RTTM and PAWS stop working

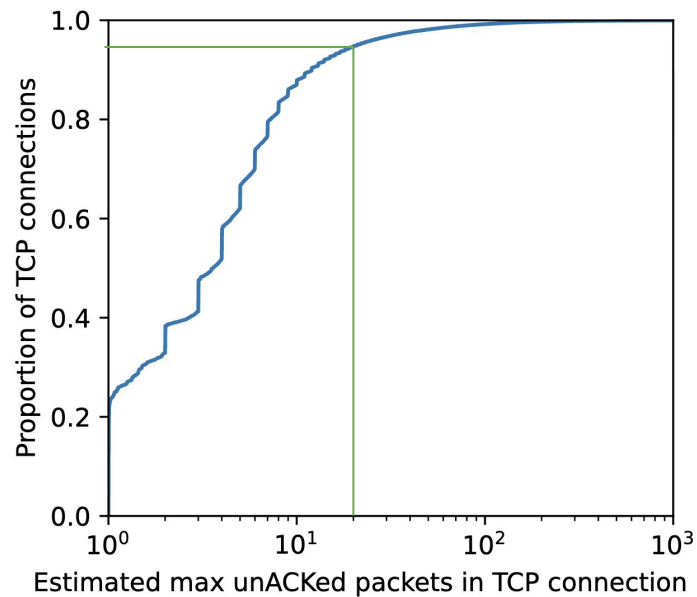
Defenses



Obfuscate TCP Timestamps

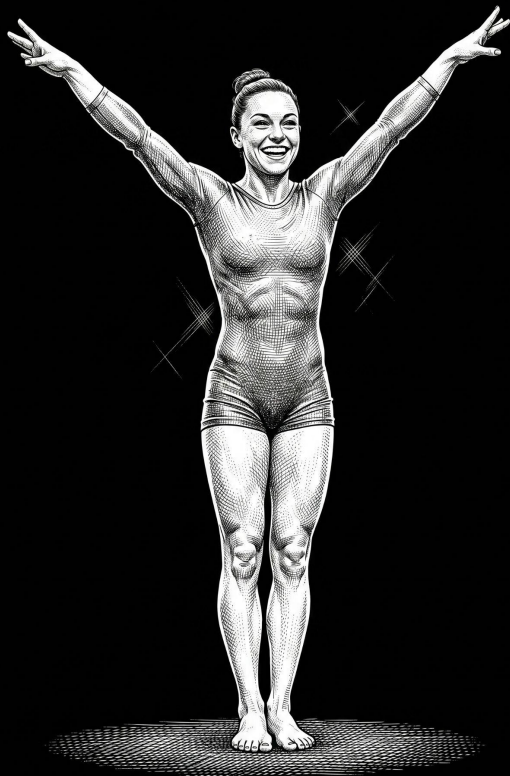
Requires kernel support & Potential middlebox impact

Defenses



Limited overhead: 20 entries $\rightarrow$ 95% of connections

Conclusion



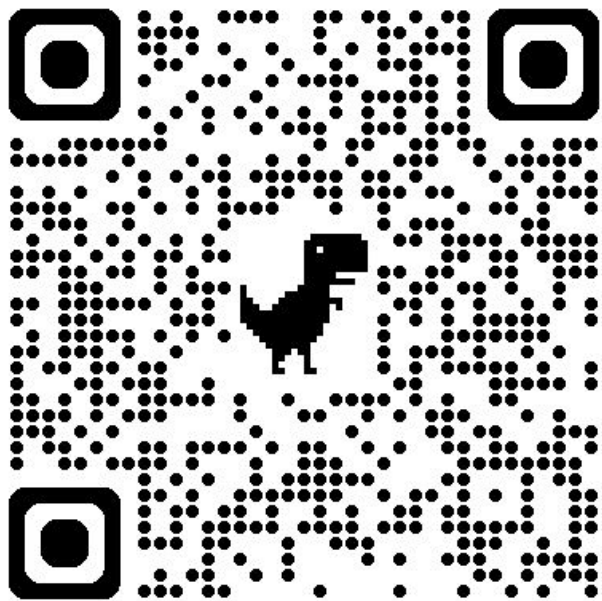
Takeaway

Timing Attacks are getting more performant

This attack can be performed fully distributed

Be careful when exposing time(-related) info

High time to check your
servers: use our scripts



Time

for



ACKrobotics

Leveraging TCP Timestamps to Improve Remote Timing Attacks

Refs

- [1] T. Van Goethem, C. Popper, W. Joosen, and M. Vanhoef, “Timeless – timing attacks: Exploiting concurrency to leak secrets over remote connections,” in 29th USENIX Security Symposium (USENIX Security 20), 2020, pp. 1985–2002.
- [2] A. Bortz and D. Boneh, “Exposing private information by timing web applications,” in Proceedings of the 16th international conference on World Wide Web, 2007, pp. 621–628.
- [3] B. B. Brumley and N. Tuveri, “Remote timing attacks are still practical,” in European Symposium on Research in Computer Security. Springer, 2011, pp. 355–371.
- [4] V. Vanderlinden, W. Joosen, and M. Vanhoef, “Can you tell me the time? security implications of the server-timing header,” in Proceedings 2023 Workshop on Measurements, Attacks, and Defenses for the Web. No. March, Internet Society, 2023
- [5] V. Vanderlinden, T. Van Goethem, and M. Vanhoef, “Time will tell: Exploiting timing leaks using http response headers,” in Computer Security – ESORICS 2023, G. Tsudik, M. Conti, K. Liang, and G. Smaragdakis, Eds. Cham: Springer Nature Switzerland, 2024, pp. 3–22.
- [7] nginx contributors, “nginx module ngx http core module directives,” accessed: 06 sept 2024. [Online]. Available: https://nginx.org/en/docs/http/ngx_http_core_module.html#keepalive_requests
- [8] W. Wang, N. Cardwell, Y. Cheng, and E. Dumazet, “TCP Low Latency Option,” Internet Engineering Task Force, Internet-Draft draft-wang-tcpm-low-latency-opt-00, Jun. 2017, work in Progress. [Online]. Available: <https://datatracker.ietf.org/doc/draft-wang-tcpm-low-latency-opt/00/>
- [9] K. Y. Yang, N. Cardwell, Y. Cheng, and E. Dumazet, “TCP ETS: Extensible Timestamp Options,” Internet Engineering Task Force, Internet-Draft draft-yang-tcpm-ets-00, Nov. 2020, work in Progress. [Online]. Available: <https://datatracker.ietf.org/doc/draft-yang-tcpm-ets/00/>
- [10] <https://github.com/torvalds/linux/commit/93be6ce0e91b6> - <https://github.com/torvalds/linux/commit/614e8316aa4ca> - <https://github.com/iproute2/iproute2/commit/a043bea750026>

Refs 2

- [11] Y. Nishida, “Disabling PAWS When Other Protections Are Available,” Internet Engineering Task Force, Internet-Draft draft-nishida-tcpmdisabling-paws-00, Jun. 2018, work in Progress. [Online]. Available: <https://datatracker.ietf.org/doc/draft-nishida-tcpm-disabling-paws/00/>
- [12] D. Borman, R. T. Braden, V. Jacobson, and R. Scheffenegger, “TCP Extensions for High Performance,” RFC 7323, Sep. 2014. [Online]. Available: <https://www.rfc-editor.org/info/rfc7323>
- [13] F. Gont, “Reducing the TIME-WAIT State Using TCP Timestamps,” RFC 6191, Apr. 2011. [Online]. Available: <https://www.rfc-editor.org/info/rfc6191>
- [14] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuhlewind, “Low Extra Delay Background Transport (LEDBAT),” RFC 6817, Dec. 2012. [Online]. Available: <https://www.rfc-editor.org/info/rfc6817>
- [15] B. McDanel, “TCP Timestamping and Remotely gathering uptime information,” Mar. 2001, accessed: 06 sept 2024. [Online]. Available: <https://seclists.org/bugtraq/2001/Mar/182>
- [16] E. Bursztein, “TCP Timestamp to count hosts behind NAT,” Jan. 2005, accessed: 06 sept 2024. [Online]. Available: [http://phrack.org/issues/63/3.html#:~:text=\[%20TCP%20Timestamp%20to%20count%20Hosts%20behind%20NAT%20\]](http://phrack.org/issues/63/3.html#:~:text=[%20TCP%20Timestamp%20to%20count%20Hosts%20behind%20NAT%20])
- [17] G. Wicherski, F. Weingarten, and U. Meyer, “Ip agnostic real-time traffic filtering and host identification using tcp timestamps,” in 38th Annual IEEE Conference on Local Computer Networks, Oct 2013, pp. 647–654
- [18] J. Giffin, R. Greenstadt, P. Litwack, and R. Tibbetts, “Covert messaging through tcp timestamps,” in Privacy Enhancing Technologies, R. Dingledine and P. Syverson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 194–208.
- [19] <https://github.com/openssh/openssh-portable/commit/9286875a73b2de7736b5e50692739d314cd8d9dc>

Icons used on slides: FontAwesome, <https://fontawesome.com/>