Time and Time Again

Leveraging TCP Timestamps to Improve Remote Timing Attacks

Vik Vanderlinden, Tom Van Goethem, Mathy Vanhoef

KU LEUVEN DistriNet

Overview

- (Remote) Timing Attacks
- TCP Timestamps
- Attack Mechanisms
- Attack Performance
- Defenses & Future Work

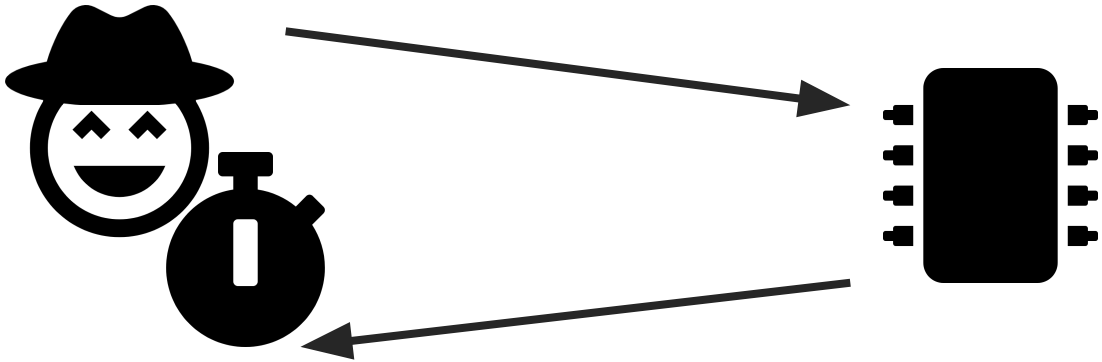
Overview

- **(Remote) Timing Attacks**
- TCP Timestamps
- Attack Mechanisms
- Attack Performance
- Defenses & Future Work

Timing Attacks

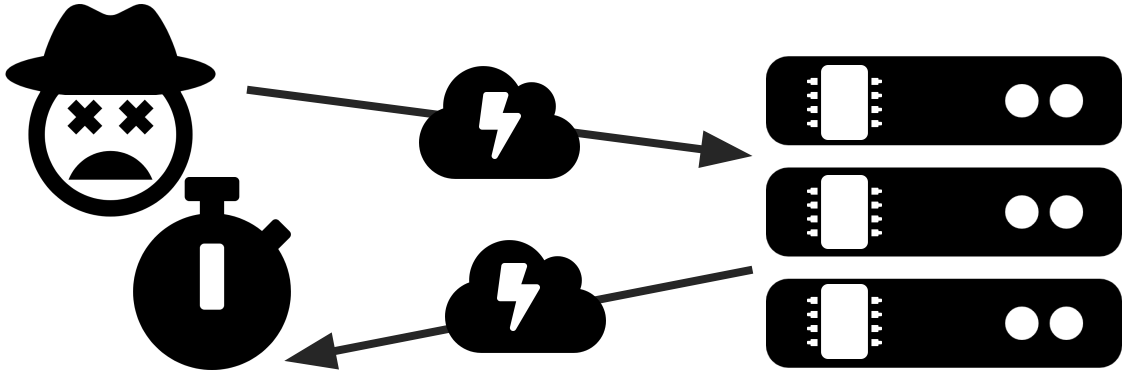
Bug-free execution

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



Remote Timing Attacks

- Timing Attack over the Internet
- Measure RTT
- There is noise (*jitter*)
- Noise = annoying



- Need for many more measurements
- Repeat + analyse[3, 4, 5] (95% success rate)

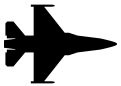
Improving Remote Timing Attacks

- Limitations due to network?

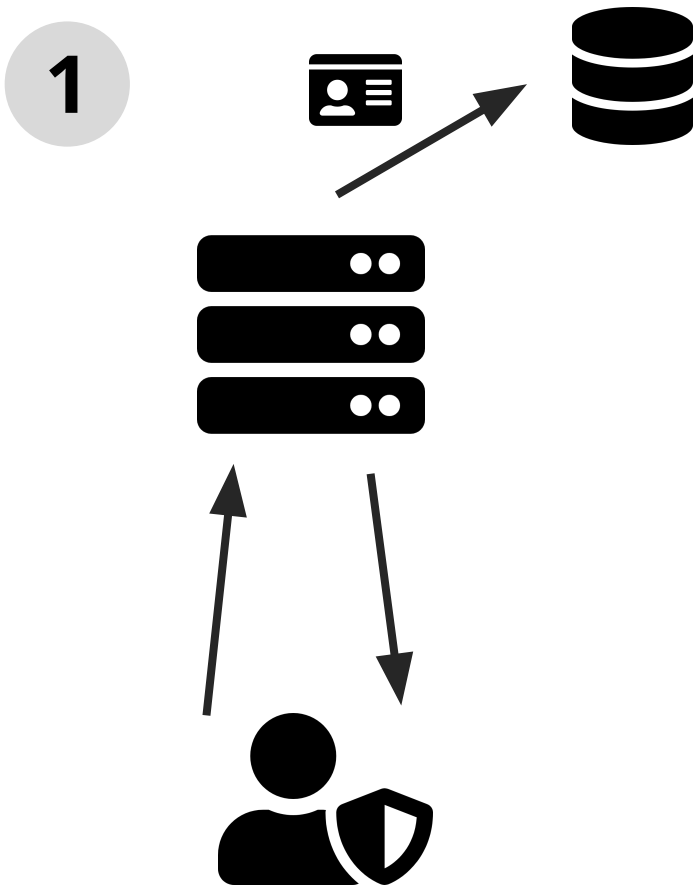
- Improvements available 

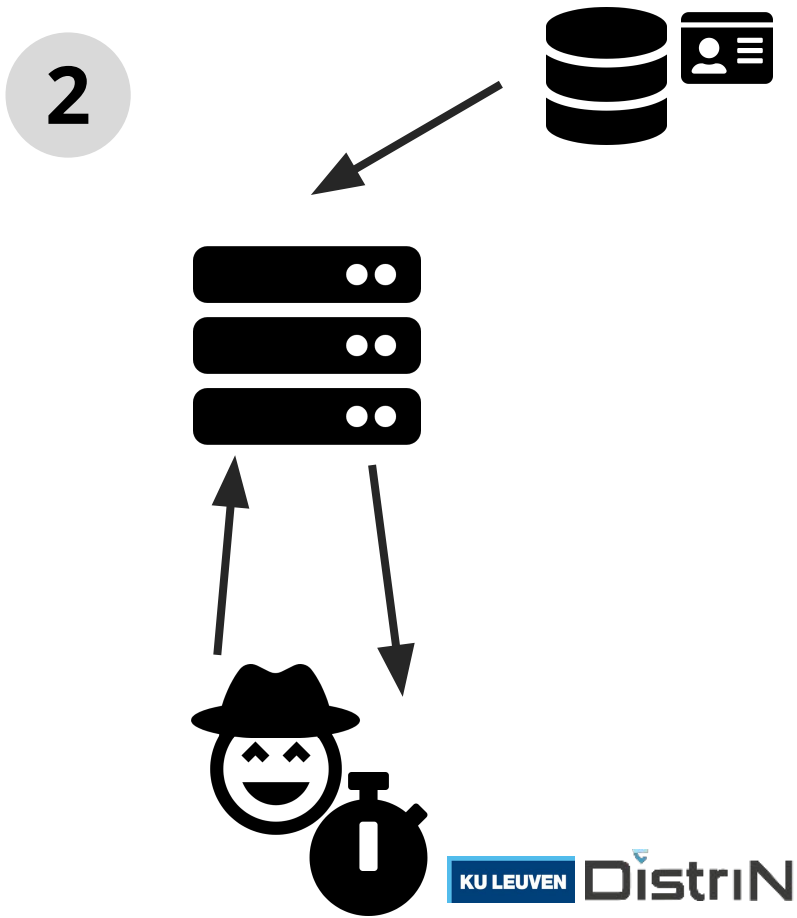
 Date Header (HTTP)

 Server-Timing Header (HTTP)

 TTA[1] (TCP)

Attack Models[2]: Direct Attack





Direct Attack: Examples

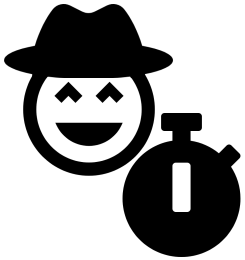
1

Joins private group

Creates private albums

…

2



Infers group membership

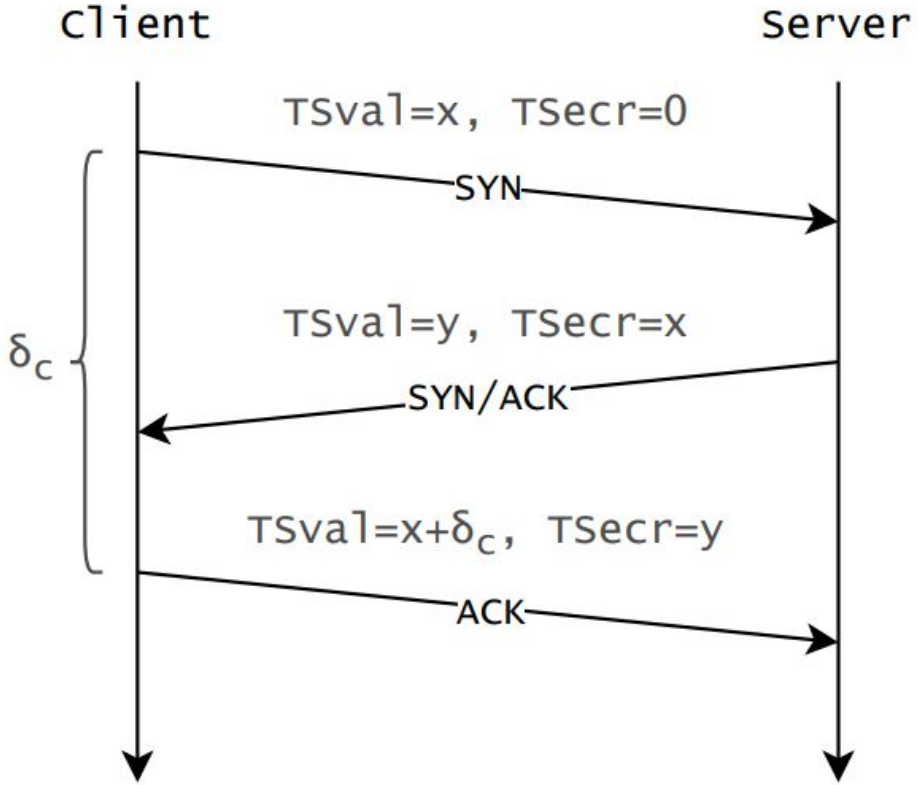
Counts number of albums[2]

…

Overview

- (Remote) Timing Attacks
- **TCP Timestamps**
- Attack Mechanisms
- Attack Performance
- Defenses & Future Work

TCP Timestamps: What?



TCP Timestamps: Why?

- Improve RTTM [1] [12]
  - Congestion Control
- PAWS [2] [12]
  - High-bandwidth applications (DC)
- Reducing Time-Wait state [2] [13]
- LEDBAT [1] [14]
- …

Performance Improvements

[1]requires time-based
[2]monotonic non-decreasing

TCP Timestamps

- Exploited in the past
  - Measuring uptime (0 at boot) [15]
  - Counting number of hosts behind a NAT/load balancer [16]
  - Traffic flow identification [17]
  - Covert messaging [18]
- Randomized initial value → solve
- Various proposals for adding µsec support[8, 9]

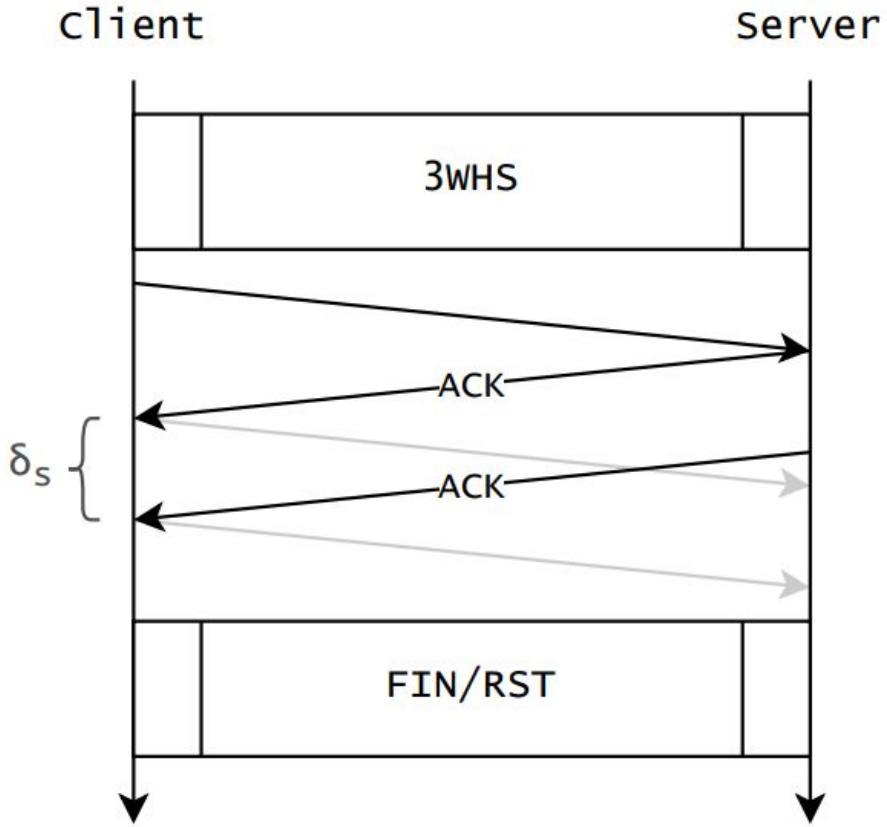
In newer linux kernels[10] (e.g. included in ubuntu >24.10), enable using ip route option[10]

Overview

- (Remote) Timing Attacks
- TCP Timestamps
- **Attack Mechanisms**
- Attack Performance
- Defenses & Future Work

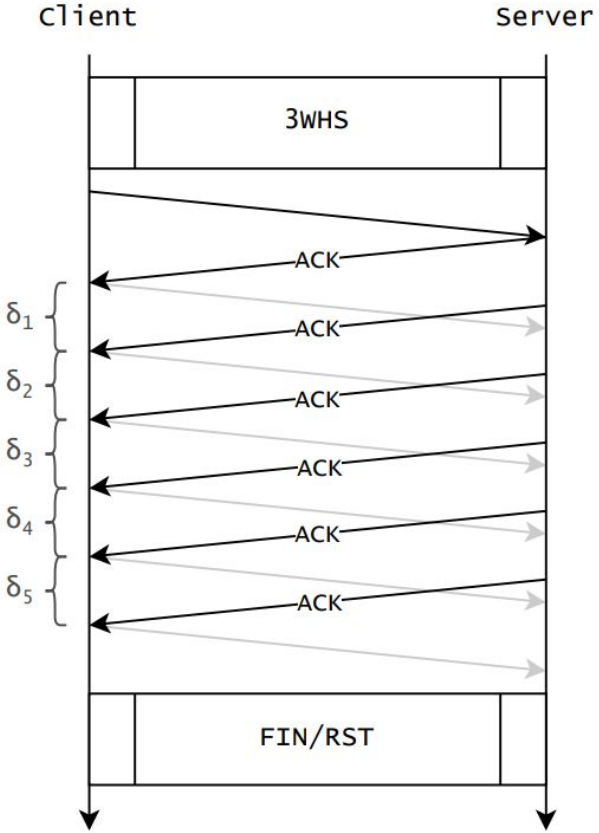
Timing Attacks leveraging TCP Timestamps

Basic Attack



Timing Attacks leveraging TCP Timestamps

Runtime Multiplication Enhancement



Attack preconditions

- TCP Timestamps enabled [>88%]
- Immediate ACKnowledgement [>99%]
- Persistent connections [>95%]
- HTTP/1.1 support (non-concurrency) [infra-dependent]
  - CDN vs origin server?
  - Backward compatibility
- Request coalescing
  - Need low-level network access
  - (MitM position to read timestamps)

Practicalities

- How many to coalesce?
  - Nginx: 1000 (previously 100) before connection close[7]
  - TCP segment size (1,5KB; MSS)
  - TLS frame size (16KB)
  - Out-of-order TCP segments (6 MiB default @ AWS ubuntu)
- Fully distributable attack
- Discrete values (ms accuracy) more difficult to analyze

Overview

- (Remote) Timing Attacks
- TCP Timestamps
- Attack Mechanisms
- **Attack Performance**
- Defenses & Future Work

Attack Performance

ms timestamps

- Decrease in exploitable timing difference by 5 times (25 µs → 5 µs)

Attack	Analysis Method	Coalescing	1 µs	5 µs	10 µs	25 µs	50 µs	75 µs	100 µs	250 µs	500 µs
TCPTS	χ^2 / Threshold	900	–	27 900	900	900	900	900	900	900	900
		100	–	26 100	500	200	100	100	100	100	100
		50	–	32 550	600	200	50	50	50	50	50
		10	–	–	640	230	50	40	20	10	10
		5	–	–	2 320	225	55	35	25	5	5
		2	–	–	2 434	272	48	28	26	12	4
		1	–	–	5 009	305	47	28	25	13	6
RTT	box test	/	–	–	–	10 337	2 944	848	1 089	66	32

TABLE II

RESULTS OF THE ATTACK USING TCP TIMESTAMPS AND A ROUND-TRIP TIME ATTACK. EACH VALUE INDICATES THE NUMBER OF REQUESTS THAT ARE REQUIRED FOR A SUCCESSFUL ATTACK (95% SUCCESS RATE). VALUES MARKED BY A '–' WERE UNSUCCESSFUL WITH UP TO 100 000 REQUESTS.

Attack Performance

ms timestamps

- Decrease in number of requests required by 5 to 50 times (25 µs: >10k requests → 200 requests)

Attack	Analysis Method	Coalescing	1 µs	5 µs	10 µs	25 µs	50 µs	75 µs	100 µs	250 µs	500 µs
TCPTS	χ^2 / Threshold	900	–	27 900	900	900	900	900	900	900	900
		100	–	26 100	500	200	100	100	100	100	100
		50	–	32 550	600	200	50	50	50	50	50
		10	–	–	640	230	50	40	20	10	10
		5	–	–	2 320	225	55	35	25	5	5
		2	–	–	2 434	272	48	28	26	12	4
		1	–	–	5 009	305	47	28	25	13	6
RTT	box test	/	–	–	–	10 337	2 944	848	1 089	66	32

TABLE II

RESULTS OF THE ATTACK USING TCP TIMESTAMPS AND A ROUND-TRIP TIME ATTACK. EACH VALUE INDICATES THE NUMBER OF REQUESTS THAT ARE REQUIRED FOR A SUCCESSFUL ATTACK (95% SUCCESS RATE). VALUES MARKED BY A '–' WERE UNSUCCESSFUL WITH UP TO 100 000 REQUESTS.

Attack Performance

µs timestamps

Decrease in number of requests required by 33 times
(25 µs → 750 ns)

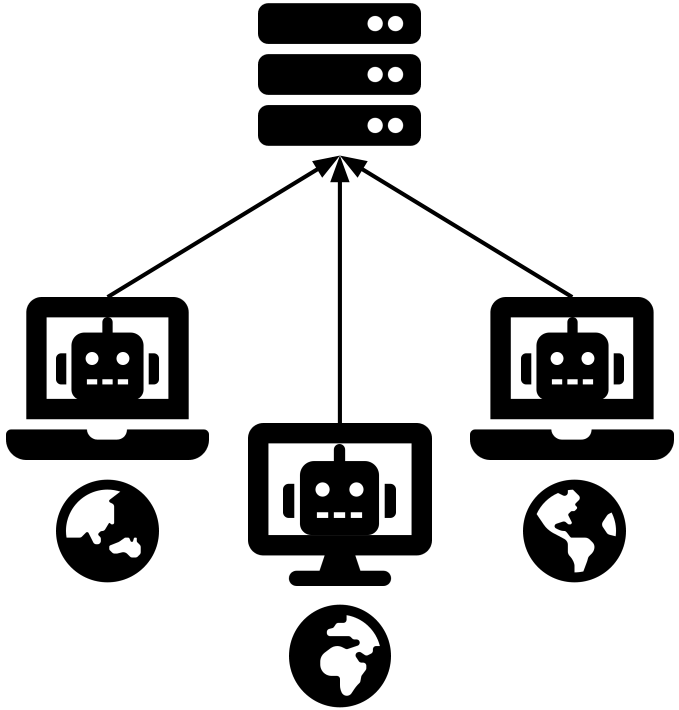
Attack Analysis Method	Coalescing	500 ns	750 ns	1 µs	5 µs
µs TCPTS box test	900	–	–	–	16 200
	100	–	90 100	45 700	8 500
	50	–	83 000	44 650	7 850
	10	–	80 410	38 370	6 890
	5	–	90 885	40 985	6 245
	2	–	–	45 490	6 654
	1	–	91 349	53 101	5 781

TABLE III
RESULTS OF THE MICROSECOND TIMESTAMPS ATTACK USING THE BOX TEST. EACH VALUE INDICATES THE NUMBER OF REQUESTS THAT ARE REQUIRED FOR A SUCCESSFUL ATTACK (95% SUCCESS RATE). VALUES MARKED BY A '–' WERE UNSUCCESSFUL WITH UP TO 100 000 REQUESTS.

Attack Performance

Distributed attack

- Collection from three geographically distant clients
- Datasets combined
- Results (50 µs) identical to non-distributed attack



Case Studies

- **TLS**
  - First transatlantic exploit of Lucky 13
  - Using µs-accurate timestamps
  - CVE-2025-32998 - responsibly disclosed
- Reproduced user enumeration against **SSH**
- Reproduced user enumeration against **FTP**
  - With 900 requests/s load to show robustness

Overview

- (Remote) Timing Attacks
- TCP Timestamps
- Attack Mechanisms
- Attack Performance
- **Defenses & Future Work**

Defenses?

- Send timestamps less often (e.g. [11])
  - Only hinders the attack slightly

- Disable TCP Timestamps
  - RTTM and PAWS stop working

- Obfuscated Timestamps
  - Requires kernel support
  - Impact on middleboxes (IDS/IPS might use Timestamps)?
  - Limited overhead: 20 entries → 95% of connections

Future Work

- Additional TCP-based protocols
- Non-TCP-based protocols with timestamping
  - E.g. QUIC
- More advanced analysis methods

Time and Time Again

Leveraging TCP Timestamps to Improve Remote Timing Attacks

Vik Vanderlinden, Tom Van Goethem, Mathy Vanhoef

KU LEUVEN DistriNet

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

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]

[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.

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

Timing Attacks: Example

```
Calculate   0 * 47 ..
Calculate  46 * 47 ..
```

Timing Attacks: Example

```
Calculate   0 * 47 .. = 0
Calculate  46 * 47 ..................... = 2 162
```

Attack Models[2]: Cross-site Attack