(Still) Exploiting TCP Timestamps

Veit N. Hailperin[1]

[1] scip AG

Swiss Network Operators Group #28

Outline

1. Facts About TCP Timestamps

2. A History of Exploitation and Failed Remediation

3. More Fun with Timestamps

4. What now?

About Me

- Security Consultant & Researcher @ scip AG
- @fenceposterror
- Bug in the matrix

Disclaimer

I will use IP on the slides synonym to IP address for space reasons.
Timestamps allows refer to TCP timestamps if not otherwise noted.

Facts About TCP Timestamps

- Introduced in 1992
- Described in RFC1323
- Extension to provide PAWS and improved RTTM

A TCP Timestamp

```
Kind: 8
Length: 10 bytes

    +-------+-------+---------------------+---------------------+
    |Kind=8 |  10   |   TS Value (TSval)  |TS Echo Reply (TSecr)|
    +-------+-------+---------------------+---------------------+
        1       1              4                     4
```

Attack Vector - Timestamp

2005 - Host Identification

Attack Vector - Timestamp

2005: Host Identification

- TCP timestamp is unique to TCP Stack

Attack Vector - Timestamp

2005: Host Identification

- TCP timestamp is unique to TCP Stack
- Attack against servers

Attack Vector - Timestamp

2005: Host Identification

- TCP timestamp is unique to TCP Stack
- Attack against servers
- All services use the same TCP Stack

Attack Vector - Timestamp

2005: Host Identification

- TCP timestamp is unique to TCP Stack
- Attack against servers
- All services use the same TCP Stack

Attack Vector - Timestamp

2005: Host Identification

- TCP timestamp is unique to TCP Stack
- Attack against servers
- All services use the same TCP Stack ⇒ all services on a host with same timestamp

Attack Vector - Timestamp

2005: Host Identification

- TCP timestamp is unique to TCP Stack
- Attack against servers
- All services use the same TCP Stack ⇒ all services on a host with same timestamp
- Ticks in milliseconds range

Attack Vector - Timestamp

2005: Host Identification

- TCP timestamp is unique to TCP Stack
- Attack against servers
- All services use the same TCP Stack ⇒ all services on a host with same timestamp
- Ticks in milliseconds range

Attack Vector - Timestamp

2005: Host Identification

- TCP timestamp is unique to TCP Stack
- Attack against servers
- All services use the same TCP Stack ⇒ all services on a host with same timestamp
- Ticks in milliseconds range ⇒ if close they are likely the same host

Attack Vector - Timestamp

2005: Host Identification

- TCP timestamp is unique to TCP Stack
- Attack against servers
- All services use the same TCP Stack ⇒ all services on a host with same timestamp
- Ticks in milliseconds range ⇒ if close they are likely the same host
- Multiple IPs with same timestamp

Attack Vector - Timestamp

2005: Host Identification

- TCP timestamp is unique to TCP Stack
- Attack against servers
- All services use the same TCP Stack ⇒ all services on a host with same timestamp
- Ticks in milliseconds range ⇒ if close they are likely the same host
- Multiple IPs with same timestamp

Attack Vector - Timestamp

2005: Host Identification

- TCP timestamp is unique to TCP Stack
- Attack against servers
- All services use the same TCP Stack ⇒ all services on a host with same timestamp
- Ticks in milliseconds range ⇒ if close they are likely the same host
- Multiple IPs with same timestamp ⇒ hosted on the same machine

Attack Vector - Timestamp

2005: Host Identification

- TCP timestamp is unique to TCP Stack
- Attack against servers
- All services use the same TCP Stack ⇒ all services on a host with same timestamp
- Ticks in milliseconds range ⇒ if close they are likely the same host
- Multiple IPs with same timestamp ⇒ hosted on the same machine
- Attack same server through multiple vectors (IPs)

Attack Vector - Timestamp

2005: Host Identification - Remediation

- Disable timestamps (bad idea)

Attack Vector - Timestamp

2005: Host Identification - Remediation

- Disable timestamps (bad idea)
- Randomizing/Zeroing timestamps (loss of functionality)

Attack Vector - Timestamp

2005: Host Identification - Remediation

- Disable timestamps (bad idea)
- Randomizing/Zeroing timestamps (loss of functionality)
- Use a different counter for each connection and initialize with 0 (problem: PAWS)

Attack Vector - Timestamp

2005: Host Identification - Remediation

- Disable timestamps (bad idea)
- Randomizing/Zeroing timestamps (loss of functionality)
- Use a different counter for each connection and initialize with 0 (problem: PAWS)
- Like above but with randomized start (problem: PAWS)

Attack Vector - Timestamp

2015: Host Identification

- Still possible[1] . . .

[1] It's a tiny bit more tricky for a small group of systems

Attack Vector - Timestamp

2007/2015 - Network Layout Information Gathering

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- TCP timestamp is unique to TCP Stack

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- TCP timestamp is unique to TCP Stack
- Attack against anonymization through NAT

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- TCP timestamp is unique to TCP Stack
- Attack against anonymization through NAT
- All services use the same TCP stack

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- TCP timestamp is unique to TCP Stack
- Attack against anonymization through NAT
- All services use the same TCP stack

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- TCP timestamp is unique to TCP Stack
- Attack against anonymization through NAT
- All services use the same TCP stack ⇒ all services on a host with same timestamp

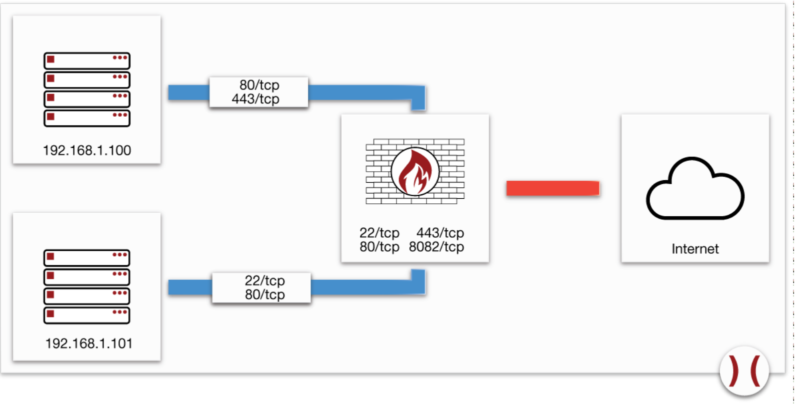
Attack Vector - Timestamp

2007: Network Layout Information Gathering

- TCP timestamp is unique to TCP Stack
- Attack against anonymization through NAT
- All services use the same TCP stack ⇒ all services on a host with same timestamp ⇒ possible to map which ports belong to which system
- Attack known since at least 2007

Attack Vector - Timestamp

2007: Network Layout Information Gathering



Attack Vector - Timestamp

2007: Network Layout Information Gathering

- Count IPs behind a NAT (if you are the receiving end of connections) (2007)

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- Count IPs behind a NAT (if you are the receiving end of connections) (2007)
- Count IPs behind a NAT (if you have multiple ports open) (2015)

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- Count IPs behind a NAT (if you are the receiving end of connections) (2007)
- Count IPs behind a NAT (if you have multiple ports open) (2015)
- TCP timestamp is the same

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- Count IPs behind a NAT (if you are the receiving end of connections) (2007)
- Count IPs behind a NAT (if you have multiple ports open) (2015)
- TCP timestamp is the same

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- Count IPs behind a NAT (if you are the receiving end of connections) (2007)
- Count IPs behind a NAT (if you have multiple ports open) (2015)
- TCP timestamp is the same ⇒ services on same host

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- Count IPs behind a NAT (if you are the receiving end of connections) (2007)
- Count IPs behind a NAT (if you have multiple ports open) (2015)
- TCP timestamp is the same ⇒ services on same host
- TCP timestamp is different

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- Count IPs behind a NAT (if you are the receiving end of connections) (2007)
- Count IPs behind a NAT (if you have multiple ports open) (2015)
- TCP timestamp is the same ⇒ services on same host
- TCP timestamp is different

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- Count IPs behind a NAT (if you are the receiving end of connections) (2007)
- Count IPs behind a NAT (if you have multiple ports open) (2015)
- TCP timestamp is the same ⇒ services on same host
- TCP timestamp is different ⇒ services on different hosts

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- Count IPs behind a NAT (if you are the receiving end of connections) (2007)
- Count IPs behind a NAT (if you have multiple ports open) (2015)
- TCP timestamp is the same ⇒ services on same host
- TCP timestamp is different ⇒ services on different hosts
- Some ports answer with no timestamp

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- Count IPs behind a NAT (if you are the receiving end of connections) (2007)
- Count IPs behind a NAT (if you have multiple ports open) (2015)
- TCP timestamp is the same ⇒ services on same host
- TCP timestamp is different ⇒ services on different hosts
- Some ports answer with no timestamp

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- Count IPs behind a NAT (if you are the receiving end of connections) (2007)
- Count IPs behind a NAT (if you have multiple ports open) (2015)
- TCP timestamp is the same ⇒ services on same host
- TCP timestamp is different ⇒ services on different hosts
- Some ports answer with no timestamp ⇒ Can't tell

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- Count IPs behind a NAT (if you are the receiving end of connections) (2007)
- Count IPs behind a NAT (if you have multiple ports open) (2015)
- TCP timestamp is the same ⇒ services on same host
- TCP timestamp is different ⇒ services on different hosts
- Some ports answer with no timestamp ⇒ Can't tell

Attack Vector - Timestamp

2007: Network Layout Information Gathering

- No tool that exploits this knowledge

Attack Vector - Timestamp

2007: Network Layout Information Gathering

DEMO

Attack Vector - Timestamp

2007: Network Layout Information Gathering - Remediation

- Increment randomly (defeats RTTM)

Attack Vector - Timestamp

2007: Network Layout Information Gathering - Remediation

- Increment randomly (defeats RTTM)
- Rewrite timestamp on NAT device

Attack Vector - Timestamp

2015: Network Layout Information Gathering

- Still possible . . .

Attack Vector - Timestamp

2001 - Uptime Calculation

Attack Vector - Timestamp

2001: Uptime Calculation

- Timestamp != Uptime

Attack Vector - Timestamp

2001: Uptime Calculation

- Timestamp != Uptime
- Multiple Timestamps

Attack Vector - Timestamp

2001: Uptime Calculation

- Timestamp != Uptime
- Multiple Timestamps

Attack Vector - Timestamp

2001: Uptime Calculation

- Timestamp != Uptime
- Multiple Timestamps ⇒ frequency of host

Attack Vector - Timestamp

2001: Uptime Calculation

- Timestamp != Uptime
- Multiple Timestamps ⇒ frequency of host ⇒ timestamp & frequency

Attack Vector - Timestamp

2001: Uptime Calculation

- Timestamp != Uptime
- Multiple Timestamps ⇒ frequency of host ⇒ timestamp & frequency ⇒ uptime
- Uptime related to patch level :)

Attack Vector - Timestamp

2001: Uptime Calculation - Remediation

- Randomize timestamps at boot

Attack Vector - Timestamp

2001: Uptime Calculation - Remediation

- Randomize timestamps at boot

Attack Vector - Timestamp

2001: Uptime Calculation - Remediation

- Randomize timestamps at boot (problems: lack of entropy, determination of initial value easy)

Attack Vector - Timestamp

2001: Uptime Calculation - Remediation

- Randomize timestamps at boot (problems: lack of entropy, determination of initial value easy)
- Start each new TCP Connection with 0 (problem: still PAWS)

Attack Vector - Timestamp

2001: Uptime Calculation - Remediation

- Randomize timestamps at boot (problems: lack of entropy, determination of initial value easy)
- Start each new TCP Connection with 0 (problem: still PAWS)
- Timestamp per IP/port pair (problem: only a question of time)

Attack Vector - Timestamp

2001: Uptime Calculation - Remediation

- Randomize timestamps at boot (problems: lack of entropy, determination of initial value easy)
- Start each new TCP Connection with 0 (problem: still PAWS)
- Timestamp per IP/port pair (problem: only a question of time)
- More problems: Might break syn flood protection under linux

Attack Vector - Timestamp

2001: Uptime Calculation - Remediation

- Randomize timestamps at boot (problems: lack of entropy, determination of initial value easy)
- Start each new TCP Connection with 0 (problem: still PAWS)
- Timestamp per IP/port pair (problem: only a question of time)
- More problems: Might break syn flood protection under linux
- Timestamp counter for each IP

Attack Vector - Timestamp

2015: Uptime Calculation

- Still possible[2] . . .
- To cut it short: timestamps observed over a longer period also lets us know their habits, e.g. when shutting down, when booting, . . .

[2] It's a tiny bit more tricky for a small group of systems

Attack Vector - Clock Skew

- Let's assume we did fix all the aforementioned issues, are we done?

Attack Vector - Clock Skew

- Let's assume we did fix all the aforementioned issues, are we done?
- no :(

Attack Vector - Clock Skew

- Let's assume we did fix all the aforementioned issues, are we done?
- no :(
- (Mainly) due to physical properties (heat, fabrication, . . .) clock isn't exact

Attack Vector - Clock Skew

- Let's assume we did fix all the aforementioned issues, are we done?
- no :(
- (Mainly) due to physical properties (heat, fabrication, . . .) clock isn't exact
- This slight imperfection of clock can be used as identifier (clock skew)

Attack Vector - Clock Skew

2005 - Host Identification

Attack Vector - Clock Skew

2005: Host Identification

- Possible even if host/port tuple TCP timestamp solution got implemented

Attack Vector - Clock Skew

2005: Host Identification

- Possible even if host/port tuple TCP timestamp solution got implemented
- Multiple IPs virtually hosted not possible with timestamp (because per OS)

Attack Vector - Clock Skew

2005: Host Identification

- Possible even if host/port tuple TCP timestamp solution got implemented
- Multiple IPs virtually hosted not possible with timestamp (because per OS)
- With clock skew not a problem, because they share hardware

Attack Vector - Clock Skew

2005: Host Identification

- Possible even if host/port tuple TCP timestamp solution got implemented
- Multiple IPs virtually hosted not possible with timestamp (because per OS)
- With clock skew not a problem, because they share hardware
- Interesting to track users

Attack Vector - Clock Skew

2005: Host Identification - Remediation

- Reduce device's clock skew (difficult!)

Attack Vector - Clock Skew

2005: Host Identification - Remediation

- Reduce device's clock skew (difficult!)
- Mask clock skew by multiplying timestamp with random value (breaks RFC)

Attack Vector - Clock Skew

2005: Host Identification - Remediation

- Reduce device's clock skew (difficult!)
- Mask clock skew by multiplying timestamp with random value (breaks RFC)
- mod_skewmask: Mask clock skew with constant

Attack Vector - Clock Skew

2005: Host Identification - Remediation

- Reduce device's clock skew (difficult!)
- Mask clock skew by multiplying timestamp with random value (breaks RFC)
- mod_skewmask: Mask clock skew with constant
- Encrypt timestamps (breaks RFC)

Attack Vector - Clock Skew

2005: Host Identification - Remediation

- Reduce device's clock skew (difficult!)
- Mask clock skew by multiplying timestamp with random value (breaks RFC)
- mod_skewmask: Mask clock skew with constant
- Encrypt timestamps (breaks RFC)
- Table mapping between random 32-bit values and internal representation of real timestamps (breaks RFC)

Attack Vector - Clock Skew

2015: Host Identification

Still possible[3]

[3] Some honeypots try to avoid it

Attack Vector - Clock Skew

2005 - Network Layout Information Gathering

Attack Vector - Clock Skew

2005: Network Layout Information Gathering

- If solution would be used than it wouldn't work
- But since it usually isn't ⇒ still possible . . .

Attack Vector - Clock Skew

2005: Network Layout Information Gathering - Remediation

Timestamp solution would work

Attack Vector - Clock Skew

2015: Network Layout Information Gathering

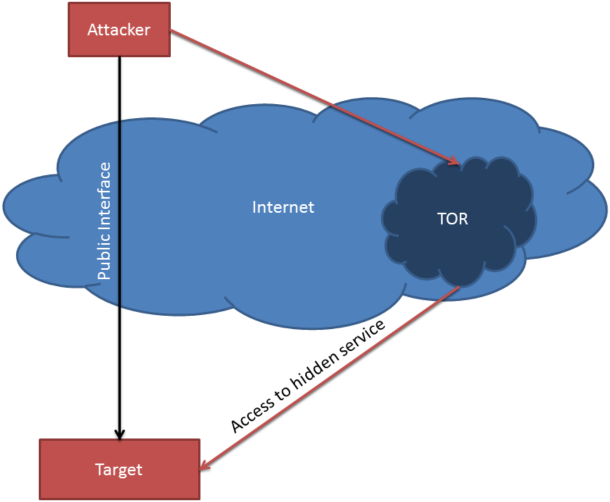
Still possible

Attack Vector - Clock Skew

2006 - Reveal Hidden Services

Attack Vector - Clock Skew

2006: Reveal Hidden Services



Attack Vector - Clock Skew

2006: Reveal Hidden Services - Remediation

- Dummy Traffic

Attack Vector - Clock Skew

2006: Reveal Hidden Services - Remediation

- Dummy Traffic
- No anonymous stream should affect another ⇒ fixed QoS for all connections (problem: potential DoS if connections idle)

Attack Vector - Clock Skew

2006: Reveal Hidden Services - Remediation

- Dummy Traffic
- No anonymous stream should affect another ⇒ fixed QoS for all connections (problem: potential DoS if connections idle)
- Oven Controlled Crystal Oscillators (OCXO)

Attack Vector - Clock Skew

2006: Reveal Hidden Services - Remediation

- Dummy Traffic
- No anonymous stream should affect another ⇒ fixed QoS for all connections (problem: potential DoS if connections idle)
- Oven Controlled Crystal Oscillators (OCXO)
- Always run at maximum CPU load

Attack Vector - Clock Skew

2015: Reveal Hidden Services

Still possible

Possible Targets

- Users
- Servers

Conclusion

Basically everyone is vulnerable

More Fun with Timestamps

2015 - Reveal Active-Active Loadbalancing

What Else Can Be Done?

2015 Load-Balanced Check!

- Multiple timestamps per port

What Else Can Be Done?

2015 Load-Balanced Check!

- Multiple timestamps per port

What Else Can Be Done?

2015 Load-Balanced Check!

- Multiple timestamps per port ⇒ Active-Active Loadbalancing

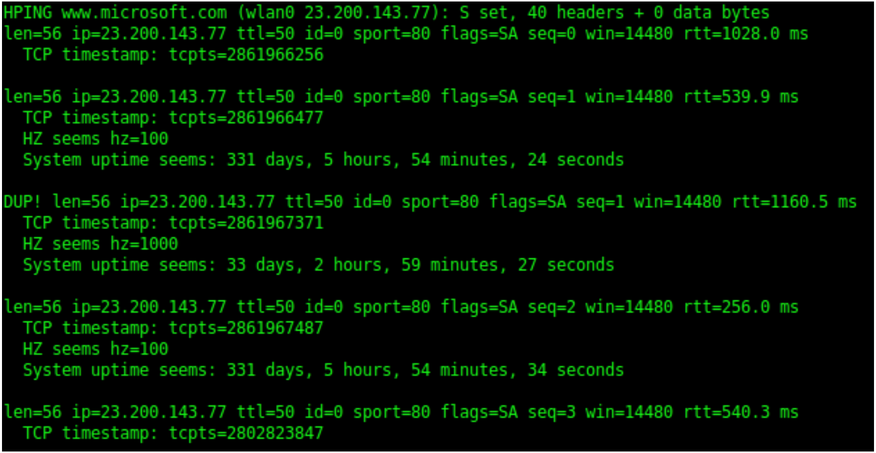
What Else Can Be Done?

2015 Load-Balanced Check!

- Multiple timestamps per port ⇒ Active-Active Loadbalancing
- Currently needs look at network dump

What Else Can Be Done?

2015 Load-Balanced Check!



```
HPING www.microsoft.com (wlan0 23.200.143.77): S set, 40 headers + 0 data bytes
len=56 ip=23.200.143.77 ttl=50 id=0 sport=80 flags=SA seq=0 win=14480 rtt=1028.0 ms
  TCP timestamp: tcpts=2861966256

len=56 ip=23.200.143.77 ttl=50 id=0 sport=80 flags=SA seq=1 win=14480 rtt=539.9 ms
  TCP timestamp: tcpts=2861966477
  HZ seems hz=100
  System uptime seems: 331 days, 5 hours, 54 minutes, 24 seconds

DUP! len=56 ip=23.200.143.77 ttl=50 id=0 sport=80 flags=SA seq=1 win=14480 rtt=1160.5 ms
  TCP timestamp: tcpts=2861967371
  HZ seems hz=1000
  System uptime seems: 33 days, 2 hours, 59 minutes, 27 seconds

len=56 ip=23.200.143.77 ttl=50 id=0 sport=80 flags=SA seq=2 win=14480 rtt=256.0 ms
  TCP timestamp: tcpts=2861967487
  HZ seems hz=100
  System uptime seems: 331 days, 5 hours, 54 minutes, 34 seconds

len=56 ip=23.200.143.77 ttl=50 id=0 sport=80 flags=SA seq=3 win=14480 rtt=540.3 ms
  TCP timestamp: tcpts=2802823847
```

More Fun with Timestamps

2015 - Improve OS Fingerprints of NAT-ed Devices

What Else Can Be Done?

2015 Improve Fingerprints!

- Repeat: What is OS Fingerprint?

What Else Can Be Done?

2015 Improve Fingerprints!

- Repeat: What is OS Fingerprint?
- Nmap doesn't assume forementioned scenario, but direct fingerprinting

What Else Can Be Done?

2015 Improve Fingerprints!

- Repeat: What is OS Fingerprint?
- Nmap doesn't assume forementioned scenario, but direct fingerprinting
- Use knowledge which ports belong together

What Else Can Be Done?

2015 Improve Fingerprints!

- Repeat: What is OS Fingerprint?
- Nmap doesn't assume forementioned scenario, but direct fingerprinting
- Use knowledge which ports belong together
- Don't use closed ports

What Else Can Be Done?

2015 Improve Fingerprints!

DEMO

Proposed Solutions

- Terminate TCP connection at firewall

Why haven't we fixed this?

Quote: Kohno et al.

[. . .] it is possible to extract security-relevant signals from data canonically considered to be noise.

Why haven't we fixed this?

Quote: Kohno et al.

> [. . .] it is possible to extract security-relevant signals from data canonically considered to be noise.

- "There are other ways to gather the same intel"-excuse

Why haven't we fixed this?

Quote: Kohno et al.

[. . .] it is possible to extract security-relevant signals from data canonically considered to be noise.

- "There are other ways to gather the same intel"-excuse
- Not considered important

Why haven't we fixed this?

Quote: Kohno et al.

> [. . .] it is possible to extract security-relevant signals from data canonically considered to be noise.

- "There are other ways to gather the same intel"-excuse
- Not considered important
- Not many good solutions so far

More Timestamps

- ICMP Timestamp (CVE-1999-0524)
- TLS Timestamp (Tor Bug #7277)
- HTTP Timestamp (Murdoch, 2013)
- . . .

Summary of (presented) Attacks

- TCP Timestamps
  - 2001 - Uptime Calculation
  - 2005 - Host Identification
  - 2007, 2015 - Network Layout Information Gathering
  - 2015 - Reveal Active-Active Loadbalancing
  - 2015 - Improve OS Fingerprints of NAT-ed Devices
- Clock Skew
  - 2005 - Host Identification / User Tracking
  - 2005 - Network Layout Information Gathering
  - 2006 - Reveal Hidden Services

What now?

Good solutions/suggestions welcome!

For Further Reading

B. Ransford and E. Rosensweig.
SkewMask: Frustrating ClockSkew Fingerprinting Attempts.
December, 2007

T. Kohno, A. Broid and K. Claffy.
Remote physical device fingerprinting
IEEE Transactions on Dependable and Secure Computing, vol. 2, no. 2, pp. 93–108, May 2005.

S. Sharma, A. Hussain and H. Saran.
Experience with heterogenous clock-skew based device fingerprinting
Proceeding LASER '12 Proceedings of the 2012 Workshop on Learning from Authoritative Security Experiment Results, Pages 9-18.

B. McDanel.
TCP Timestamping - Obtaining System Uptime Remotely
http://www.securiteam.com/securitynews/5NP0C153PI.html, March 14, 2001

For Further Reading 2

V. Jacobson, R. Braden and D. Borman.
TCP Extensions for High Performance.
Network Working Group, Request for Comments: 1323, May 1992

S. Bellovin.
Defending Against Sequence Number Attacks.
Network Working Group, Request for Comments: 1948, May 1996

M. Silbersack.
Improving TCP/IP security through randomization without sacrificing interoperability.
University of Wisconsin – Milwaukee, 2005

S. Murdoch.
Hot or not: revealing hidden services by their clock skew.
Proceeding CCS '06 Proceedings of the 13th ACM conference on Computer and communications security, Pages 27 - 36

So Long and Thanks For All The Fish

Me: @fenceposterror

Thanks to people who inspired or helped:
Krzysztof Kotowicz, Stefan Friedli, Max Hailperin