Fingerprinting hardware devices using clock-skewing

Renaud Lifchitz
renaud.lifchitz@gmail.com

#HES2010
8,9,10 April 2010 – Paris, France

Presenter's bio

- French computer security engineer
- Main activities:
  - Penetration testing & security audits
  - Security trainings
    (EC-Council CEH, ECSA/LPT, CHFI, CEI certified)
  - Security research
- Main interests:
  - Security of protocols (authentication, cryptography, information leakage, zero-knowledge proofs...)
  - Number theory (integer factorization, primality tests, elliptic curves)

Intro

What is clock-skewing?

- Also known as “clock skew” or “timing skew”
- Drift compared to the actual exact time
- Negative or positive skew
- Why is there a drift?
  - Software implementation of clock
  - Material imperfections
    (e.g. quartz fabrication)
  - Differences in wire lengths
  - Differences in input capacitance
  - Intermediate components
  - ...

Basis (1/2)

- The idea is to build a fingerprint from this drift
- Local or remote fingerprinting!
- Most important: correct time reference
- Then, target clock deviation measurements and clustering

Basis (2/2)

"The more imprecise is your clock,
the more precise will be your fingerprint!"

How does it work?

How does a computer handle time?

- In fact, it has 2 different clocks:
  - An hardware clock called “RTC” (Real time clock), made of quartz, battery powered
  - A software clock (“system clock”) handled by the OS kernel with a counter and interrupts (ticks)
- Under Linux & Windows:
  - Kernel synchronizes its software clock with RTC at boot time
  - RTC is almost never read after (even synchronizations are rare)

Measurement mechanisms

- First of all, we need the more precise local time for target drift measurement
- How to measure a clock?

  Using a better clock!

- Better clocks:
  - Atomic clocks
  - GPS clocks (basically the same!)
  - Radio clocks (e.g. DCF77, TDF...)
- Typical atomic clock precision:

  1 s./3000 years

- Fortunately, atomic clocks can be queried using NTP protocol

Correct time reference

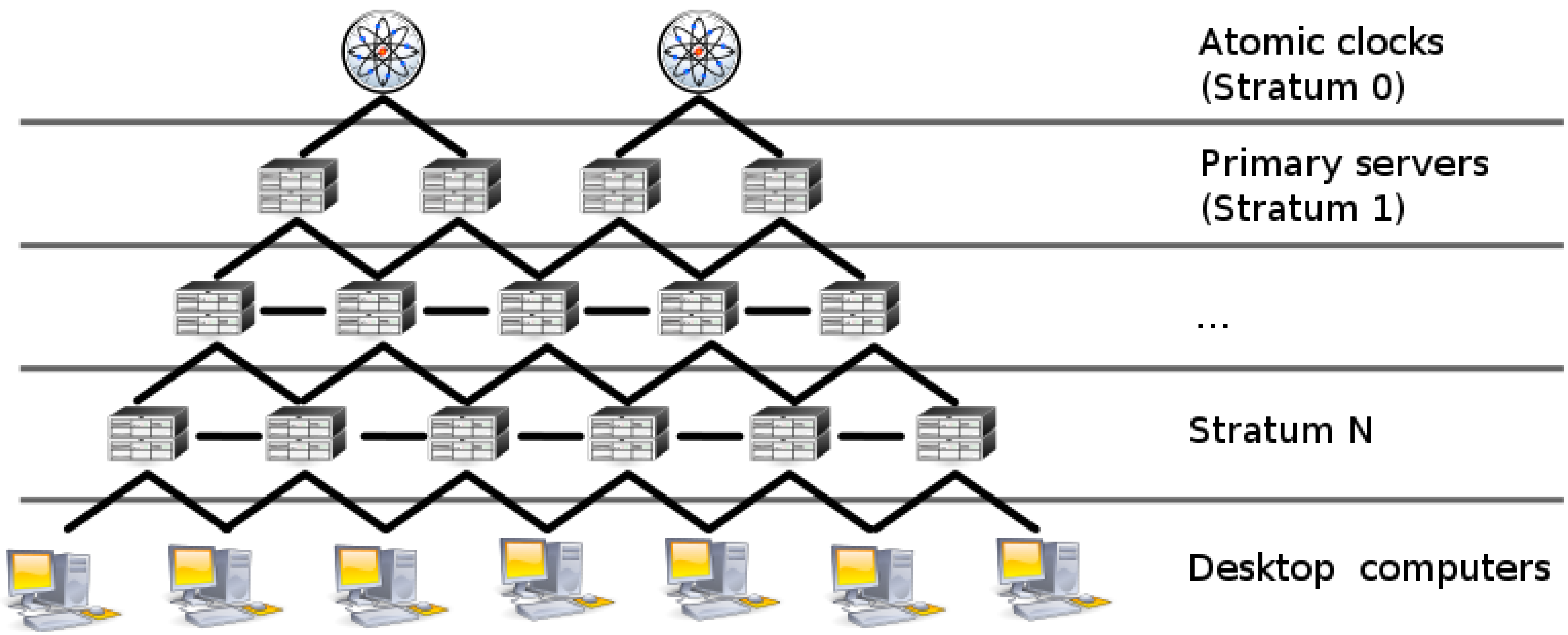
- Windows (S)NTP client can only guarantee 1-2 second precision
- We should better use Linux NTP client for measurement (10-30 ms precision!)

A few words about NTP (1/3)

- NTP = Network Time Protocol
- Protocol for synchronizing the clock of computer systems
- One of the oldest internet protocols (September 1985)
- Works with UDP, port 123
- NTP only adjusts the system clock rate so that system clock match exact time
- Precision (at best):
  - 10 ms over Internet
  - 200 µs in LAN
- Common versions: NTP v3 (RFC 1305) & NTP v4

A few words about NTP (2/3)

- NTP uses a hierarchical, layered system of levels of clock sources:



A few words about NTP (3/3)

U.S. Naval Observatory in Colorado (Stratum 0 source)

Active measurement of the target (1/2)

- One can remotely query the time of a target using "ICMP Timestamp Requests" packets (ICMP Type 13 Code 0)
- Target replies with "ICMP Timestamp Replies" (ICMP Type 14 Code 0)
- Number of milliseconds since midnight (GMT Time)
- Generated from system clock

Active measurement of the target (2/2)

```
###[ ICMP ]###
     type= timestamp-reply
     code= 0
     chksum= 0x7012
     id= 0x0
     seq= 0x0
     ts_ori= 12:19:17.427
     ts_rx= 12:47:39.852
     ts_tx= 12:47:39.852
```

ICMP layer of an ICMP Timestamp Reply

Passive measurement of the target (1/2)

- Or semi-active!
- Using TCP timestamps
- Proportional to uptime
- Generated from tick counter only
- Seems more accurate than ICMP timestamps

Passive measurement of the target (2/2)

```
###[ TCP ]###
  sport= telnet
  dport= 56066
  seq= 2240595391L
  ack= 4265897507L
  dataofs= 8L
  reserved= 0L
  flags= PA
  window= 3032
  chksum= 0x7017
  urgptr= 0
  options= [('NOP', None), ('NOP', None), ('Timestamp', (2775749850L, 3584624))]
```

TCP layer of a “timestamp-enabled” TCP packet

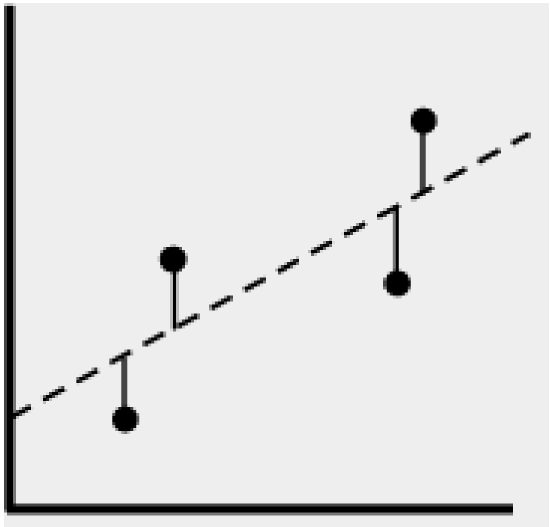
Precision & measurement resolution (1/3)

- We have to deal with 10ms of NTP precision and 30ms network latency
- According to Tadayoshi Kohno's study, average drift:
  - is stable on a given computer (+/- 1-2 ppm)
  - varies up to +/- 50 ppm

→ This gives 4-6 bits of information

Precision & measurement resolution (2/3)

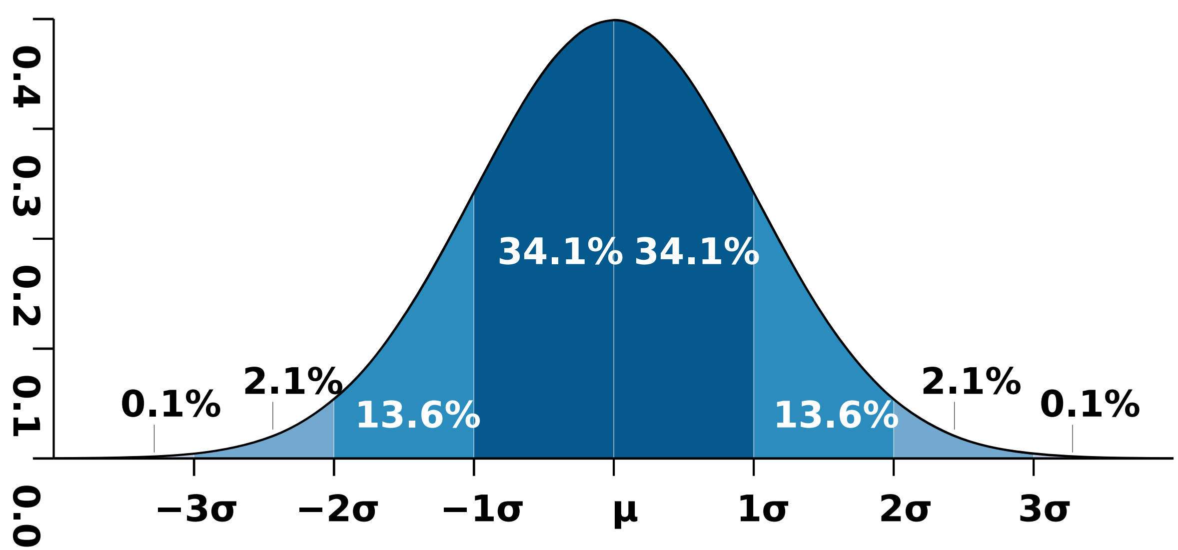
- Least square fitting on the set of measurement points:
  {(local host time, target time difference)}



- Obviously, longer measurement = better precision

Precision & measurement resolution (3/3)

- Enhancement: we can add an additional measurement dimension to fingerprint target clock precision: standard deviation around average slope (if network latency is nearly constant)
→ adds 1-3 bits of information



Distinguishing devices

- Using those 1 or 2-dimension measurements, we can easily define a distance measure between any 2 points
- Then, use any known multidimensional clustering algorithm:
  - Hierarchical algorithms
  - Partitional algorithms (e.g. k-means)
  - Density-based algorithms
- Ability to distinguish between about 2^(6+3)=512 different computers on Internet
- Can be combined with other fingerprinting techniques for better efficiency (OS TCP/IP fingerprinting, IP IDs, banners...)

Weaknesses

- Not so high resolution on Internet (need for longer measurement or additional characteristics)
- Sensitivity:
  - Temperature:
    +/- 1 ppm in typical computer temperature
  - Altitude
  - High computer activity:
    see known attacks on Tor anonymity network (ref. [1])

Applications

Identification of stolen devices

- Compute the fingerprint of your computer in case you loose it
- You are now able to find it remotely among hundreds of similar computer (a lot easier on a LAN)...
- … even if IP address / MAC address / hard drive was changed! (OS type shouldn't have been changed...)

Detection of remote virtual machines

- If guest VMs are time-synchronized with host (option in most virtualization solutions), they will share a very similar fingerprint
- Otherwise, same guest OSes on the same host will have similar fingerprints

Computer forensics

- These kinds of fingerprints can be computed offline
- Fingerprints computed from a short PCAP network capture done on a well-synchronized computer
- Ability to fingerprint an attacker computer even if entire attack isn't completely recorded
- Compare attack fingerprint with suspected computer fingerprints

Countermeasures

Countermeasures

- Frequent NTP synchronizations
- Disable:
  - TCP timestamps
  - ICMP or ICMP timestamp requests/replies
  - Any service delivering time (or just the time fonctionality, not the service!): e.g. Apache "Date" HTTP header
- Regularly change:
  - Your temperature
  - Your altitude
  - Your computer activity
  - Your processor & motherboard!

Tool & demo

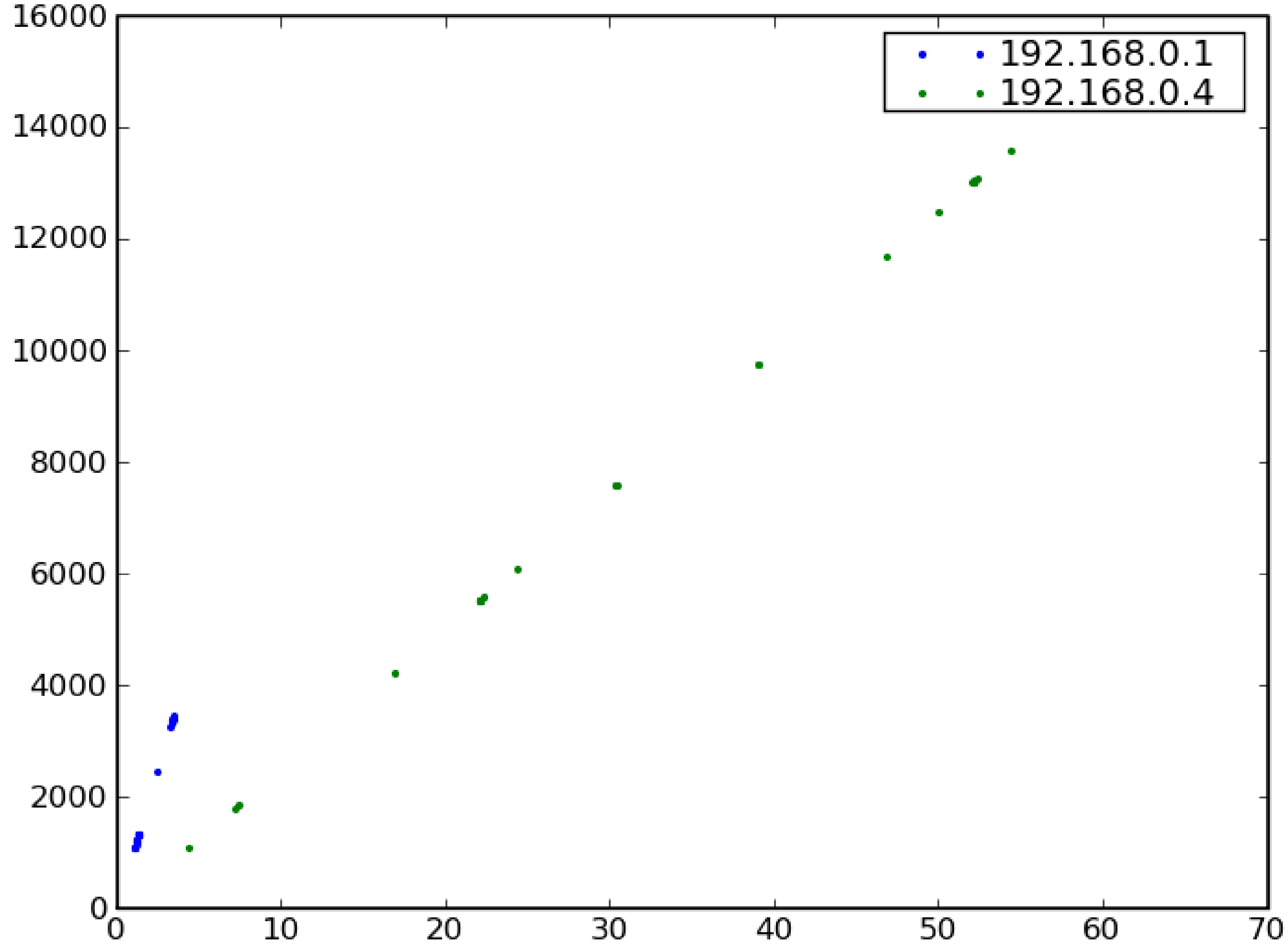
Tool

- No tool seems to exist!
- Open source tool using Python & Scapy
- Very basic & naive tool for the moment
- “Quick and dirty” coded
- Tool will be published on Google Code just after the event
- Feel free to contribute & improve the tool!

Live demo (1/2)

- Requirements:
  - Computers on a wired network (latency is too important on wireless networks): either TCP or ICMP-enabled
  - Some NTP servers for suitable time synchronization
  - Python & Scapy installed

Live demo (2/2)



192.168.0.1
192.168.0.4
16000
14000
12000
10000
8000
6000
4000
2000
0
0
10
20
30
40
50
60
70

Outro

References

- [1] Tadayoshi Kohno, Andre Broido, and K.C. Claffy, “Remote physical fingerprinting”, IEEE Transactions on Dependable and Secure Computing, 2(2):93-108, 2005.
- [2] Talk “Fingerprinting hosts through clock skew”, Steven Murdoch, EuroBSDCon, 2007
- [3] “NTP, une simple histoire de temps”, GNU/Linux Magazine France, Diamond Editions, April 2010

Thanks!

Questions?