The BlueBag:

A Mobile, Covert Bluetooth Attack and Infection Device

Claudio Merloni - c.merloni@securenetwork.it
Luca Carettoni - l.carettoni@securenetwork.it

August 3, 2006

Agenda

- Where did all of this start from?
- (Very) short intro to the Bluetooth technology and its vulnerabilities
- Our 4 W: the Why, What, hoW and Wow of the BlueBag
- Surveying bluetooth devices
- Going distributed
- Going malicious
- Giving it a try...

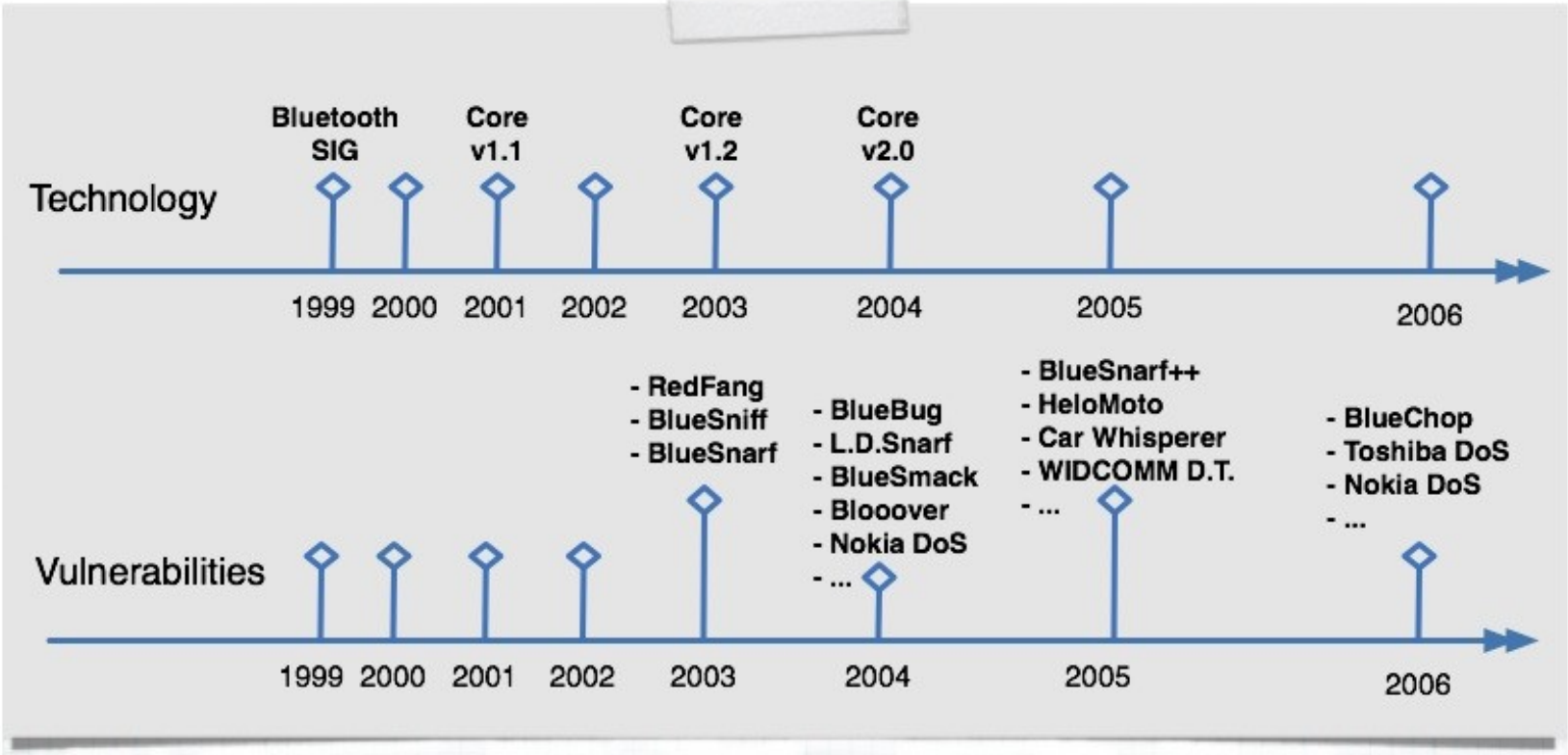
Where did all of this start from?

- Bluetooth is a geek technology
- More people than you would expect already rely on this technology. Businesses too...
- Up to some months ago, neither real data nor estimates about technology and devices spreading
- Interested in the evaluation of the exposure to worms and human aggressors

BT technology overview

- Developed as a technology replacement for low range wireless standards (eg. IrDA)
- Targeted to personal devices information exchange and networking (eg. vCard, PAN)
- Core specs v2.0 from Bluetooth SIG:
  - Hardware based radio system + Software stack
  - 2.4GHz ISM
  - Frequency Hopping Spread Spectrum (1600 hops/s on 79 channels)
  - Low power consumption, short range (up to 100m)
  - Data rates: 2 and 3 Mbps (Enhanced Data Rate)

Technology and flaws timeline



Playing in a real scenario

- The Trifinite guys @ trifinite.org showed us quite a lot of interesting things... ...please keep up the good work! ;)
- We focused on how an attacker could pose complex threats using existing knowledge and technologies
- Vulnerability != Risk

BlueBag: Why?

- Wide area survey
- 8+ hours power autonomy
- Covertness
- Easy carrying
- No human interaction
- Perfect for long sessions

Long long…

…long long sessions

BlueBag: What? - 1

- VIA EPIA Mini-ITX motherboard
- iPod 1.8in HD
- #8 Class 1 BT dongles
- #1 modded Linksys BT dongle
- #1 omnidirectional 5dBi antenna
- PicoPSU power supply connected to a 12V-26Ah lead acid battery

  = 40W power consumption (max)

BlueBag: What? - 2

- GNU/Linux Gentoo OS
- v2.6 kernel + BlueZ subsystem
- Custom python software

BlueBag: hoW?

- Making it reliable
- Firing it up
- Remote controlling
- Monitoring
- Data storage
- Data gathering in crowded places and related issues

BlueBag: Wow! ;-)

Testing on the road - 1

- Focus on identifying active and visible BT devices
- Gathered info that can help pinpointing device types and models
- Different contexts and different users (eg. shopping mall, train station, airport)
- 1405 unique devices in less 24 hours

Testing on the road

- 93% mobile phones, 3% PCs,
  ~2% PDAs, ~1% GPS, ~1% other
  - 60% Nokia (12% 6680, 8% 6310i, 7.4% 6230i, 7.1% 6600)
  - 14% SonyEricsson/Ericsson
  - 7% Samsung
  - 1.8% Motorola
- “Visibility time”: shopping mall – 12.3s, university campus – 10.1s, airport – 23.1s, bank HQ - 14.4s

Answers to /. readers

- Q: It's the same old stuff, isn't it?
  A: No. More data, long sessions, etc.
- Q: There's no security risk
  A: Hello, McFly?
- Q: [Data theft] That scenario strikes me as relatively pointless ...
  A: ... and that's why the original survey idea evolved to our current project

Looking for more data

- Getting a quantitative measure of the spreading power of Bluetooth worms
- Needed to implement mathematical spreading models and simulations
- Average number of “victims” reachable by a single wandering device
- Success rate of social engineering techniques

Going distributed - 1

- The BlueBag, as any other surveying tool, has an intrinsic limit: m-to-n inquiry
- To get real data about worm propagation effectiveness we need to implement a distributed surveying framework
- Agents spread by the BlueBag, that propagate, do the inquiry and return results back

Going distributed - 2

- Designing the agent:
  - Envelope:
    - Piece of software able to scan for Bluetooth devices and to propagate to found devices
    - It has a list of targets to propagate to, and a set of payloads that it can “deploy” on the targets
  - Payload:
    - To do the distributed survey this is just something that collects and logs data and sends the logs back to the BlueBag via Bluetooth

Going distributed - 3

Envelope

Main

```
If ( inTarget() ){
  P.run();
}else{
  while( true ){
    scanDevices();
    propagate();
  }
}
```

Payload

run(){ ... }

scanDevices()

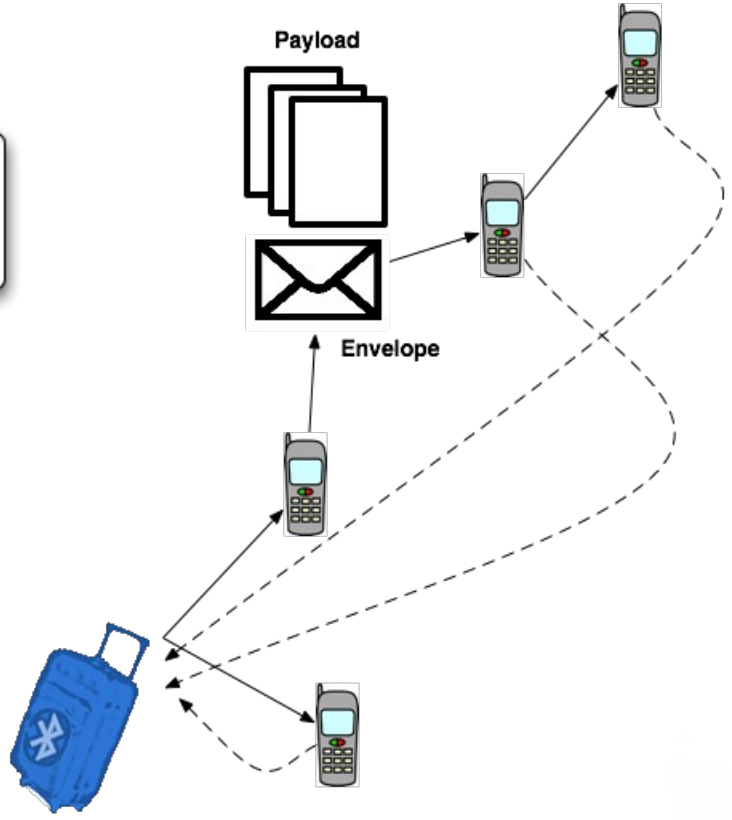
- Inquire for neighbours

propagate()

- Obex PUSH or Attacks Lib

targetsList[]

- Array of {bt_addr, payload, payload_parameters}



How can agents propagate?

- Worm techniques:
  - 2004, Cabir.A, Symbian Series 60: OBEX Transfer to the first found BT device. The victim must accept the transfer
  - 2005, Lasco.A, Symbian Series 60: same type of BT propagation, but infect SIS files too
  - 2005, Commwarrior.A, Symbian Series 60: same type of BT propagation, but use also MMS
- At present they don't exploit any vulnerabilities

Now we have tools that...

- Can do quite massive BT scanning
- Can try to deploy agents to remote devices
- Can propagate like worms but could also use more effective techniques
- Can carry payloads to be launched on the target and return results back

Going evil ;)

- We could then:
  - Give our agents a specific target
  - Tell them to use different payloads on different victims doing evil things:
    - Keylogger
    - Sniffer
    - Audio recorder
  - Tell them to give us data back using any victim device capability
  - Maybe without ever getting into the victims device Bluetooth TX range

Propagation model

- Models from epidemiology have been applied to computer viruses
- Kermack and McKendrick mathematical models:
  - Homogeneous environment (E.g. Internet)
  - No locality
- These hypotheses doesn't apply in our context ...
- ... then we go down the simulation path!

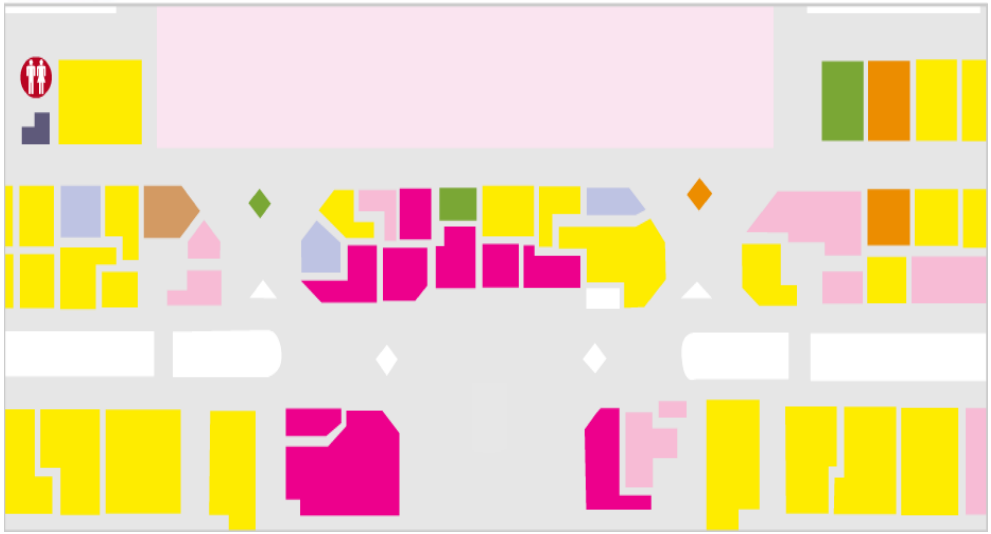
Putting it all together

- We must choose a propagation scenario and fix the parameters
- Data collected:
  - during the first survey
  - looking for “stupid” people
- What we need now is a way to estimate how effective would be that kind of targeted self-propagating malware...

Device traces and simulation

- To build a realistic scenario we need to describe how devices would displace in a physical area:
  - "A Community based Mobility Model for Ad Hoc Network Research" by Mirco Musolesi and Cecilia Mascolo
- We built a simulator that receive the traces as input and mimic the behaviour of an advanced bluetooth worm
- We are developing an evolution based on NS-2

Simulation context



- Shopping mall
- 250m x 100m surface
- 78 shops

- Number of devices: 184
- “Vulnerable” individuals: 7.5%
- Bluetooth range: 15m
- Link bandwidth: 0.3Mbps
- Payload size: 42Kb

Simulation results - 1

- Setting that tries to mimic the behaviour of people walking in and out of shops
- After 30 minutes the average percentage of (vulnerable) infected devices is 82.4%
- After 60 minutes the average percentage of (vulnerable) infected devices is 100%
- Every vulnerable device is infected after an average time of 35 min

Simulation results - 2

- Setting that tries to mimic the behaviour of people inside lunch areas
- After 30 minutes the average percentage of (vulnerable) infected devices is 100%
- Every vulnerable device is infected after an average time of 12 min

Summing up - 1

- Bluetooth technology is not only for geeks
- People aren't conscious of potential threats: visible mode, easy pairing, etc.
- Different spreading techniques can be combined to propagate more efficiently to specific devices

Summing up - 2

- A complex attack scenario, combining distributed and targeted propagation, exploiting known Bluetooth flaws and social engineering seems to be more than an idea
- The collected data, the BlueBag, our tools and what we've shown today can help to understand that the risk is definitely real
- How many ways to return back data?
- We're working on improving worm auto-execution and process hiding

References

- Bluetooth SIG technical reference: https://www.bluetooth.org/
- Linux kernel official implementation: http://www.bluez.org/
- Bluetooth security: http://trifinite.org/trifinite_org.html
- OBEX opensource implementation: http://openobex.triq.net/
- Mobility model for ad-hoc networks: http://www.cs.ucl.ac.uk/staff/m.musolesi/mobilitymodels
- NS - Network Simulator: http://www.isi.edu/nsnam/ns/

Thank you!

Any question?

We would greatly
appreciate your feedback.

Claudio Merloni - c.merloni@securenetwork.it

Luca Carettoni - l.carettoni@securenetwork.it