Passive Data Link Layer 802.11 Wireless Device Driver Fingerprinting

Presenter: Tyler Sidell
April 2, 2008

1

Authors

- Jason Franklin, Carnegie Mellon
- Damon McCoy, University of Colorado
- Paria Tabriz, University of Illinois
- Vicentiu Neagoe, University of California, Davis
- Jamie Van Randwyk, Sandia National Laboratories
- Douglas Sicker, University of Colorado

2

Agenda

- Overview
- Fingerprinting Technique
- Evaluation of Technique
- Preventing Figerprinting
- Conclusions

3

OVERVIEW

4

Problems

- Device Drivers
  - Primary source of security holes in modern operating systems
  - Execute in kernel space
    - Exploits a vulnerable driver that can lead to a compromise of the entire OS

5

Device Drivers

- Experience error rates of 3 to 7 times higher than other kernel code
  - Poorest quality code in most kernels
- Frequently modified to support new hardware features
- Developed by programmers who lack intimate knowledge of the operating system kernel

6

Device Drivers

- Interaction requires physical access to a system
  - Security Holes difficult to exploit remotely
- More vulnerable
  - Wireless cards
  - Ethernet cards
  - Modems
- User must be in transmission range of wireless device

7

Main Contributions

- Develop a passive fingerprinting technique that identifies the wireless device driver running on an IEEE 802.11 compliant device
  - Valuable to attacker looking to conduct reconnaissance against a potential target so that he may launch a driver-specific exploit
- Evaluation of Fingerprinting Technique
- Prevention of Fingerprinting
  - Improve security of wireless communication for devices that employ 802.11 networking

8

Fingerprinting

9

Fingerprinting

- Background
  - Today, most common and widespread devices are those conforming to the IEEE 802.11 standards
  - Fingerprint the Device Driver in order to evaluate ability of an attack to launch a driver-specific exploit
- Definition
  - Process by which a device or the software it is running is identified by its externally observable characteristics

10

Fingerprinting

- Accurately and efficiently identifies the wireless driver without modification to or cooperation from a wireless device
- Design, Implement, Evaluate
  - Technique for fingerprinting IEEE 802.11 a/b/g wireless networks
- Statistical Analysis
  - Rate at which common 802.11 data link layer frames are transmitted by a wireless device
- Transmission Range
  - Attacker can successfully gain information about his victim without fear of detection

11

Fingerprinting

Not a New Concept

- Ethereal
  - Use the wireless device's MAC address to identify the card manufacturer and model number
- IEEE Standards Association assigns each NIC manufacturer a special three-byte code
  - Identifies the model and manufacturer of the NIC

12

Fingerprinting Approach

13

Active Scanning

- Fingerprinting Technique relies solely on the active scan function in wireless clients
- Clients use probe request frames to scan an area for a wireless access point
  - Provides data rates

14

Active Scanning

- If an access point is compatible with the client's data rates
  - Send a probe response frame to acknowledge the request
  - The client can attempt to join the network by issuing an association request
  - Access point will respond to the client with an association ID for future communications
  - All communication between a client and another machine is routed through and controlled by the access point

15

Active Scanning

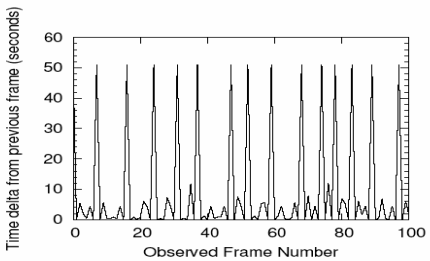
- For each channel, the client broadcasts a probe request and starts a timer.
  - If timer reaches MinChannelTime and the channel is idle, the client scans the next channel
  - Otherwise, the client waits until the timer reaches MaxChannelTime, processes the received probe response frames and then scans the next channel.

16

Probing Behavior

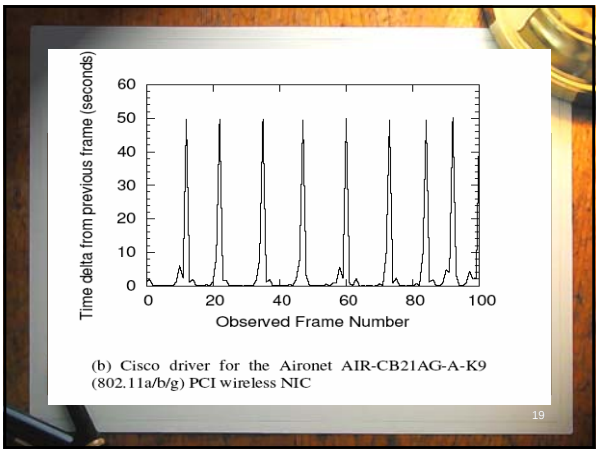
- Dependent on
  - Unobservable internal factors
    - Timers
  - Uncontrollable external factors
    - Background Traffic

17



(a) D-Link driver for the D-Link DWL-G520 (802.11b/g) PCI wireless NIC

18



(b) Cisco driver for the Aironet AIR-CB21AG-A-K9 (802.11a/b/g) PCI wireless NIC

19

Device Driver Fingerprinting

- Two Stages
  - Trace Capture
    - A fingerprinter within wireless transmission range of fingerprintee captures 802.11 traffic
  - Fingerprint Generation
    - Captured trace is analyzed using a Bayesian approach to generate a robust device driver fingerprint

20

Trace Capture

21

Trace Capture

- Assume a one-to-one mapping of MAC addresses to wireless devices
- Each NIC is assigned an unique MAC address by its manufacturer
  - Reassigning MAC address independently only cause for duplicate MAC address
- Still there are 2^{48} acceptable MAC addresses, probability of choosing an existing MAC on the network is slim.

22

Trace Capture

- Use any device that is capable of eavesdropping on the wireless frames transmitted by the fingerprintee.
- Assume fingerprinter is using a single, high-gain, COTS wireless card
  - Monitor Mode
    - Allows the wireless card to capture frames promiscuously
- Must prevent their card from associating with an access point or sending its own probe request frames
  - Keeps collection completely passive.
- Network Protocol Analyzer (i.e. Ethereal) to record the eavesdropped frames and filter out all irrelevant data

23

Fingerprint Generation

24

Fingerprint Generation

- Binning approach
  - Characterizes the time deltas between probe requests because of the inherently noisy data due to frame loss.
- Bayesian approach
  - Relates the conditional and marginal probabilities of two random events

25

Binning Approach

- Translates an interval of continuous data points into discrete bins
- Bin
  - Internal value used in place of the true value of an attribute.
- Advantage
  - Smooth probabilities
    - Places them into groups
- Disadvantage
  - Loss of information for continuous data
  - Some noise is averaged out because each bin probability is an estimate for that interval

26

Binning Approach

- Isolated two attributes from the probing rate that were essential to the wireless driver
  - Bin Frequency
  - Average of the probe request frames placed in that bin.

27

Device Signature

- Create a signature for each individual wireless driver that embodies these attributes
- Calculate Bin Probabilities
  - Rounded the actual delta arrival time value to the closest discrete bin value
  - Researchers used an optimal bin width size of 0.8 secs.

28

Digital Signature

Bin	Percentage	Mean
0	0.676	0.16
1.2	0.228	1.72
50	0.096	49.80

Table 1: Sample signature for the Cisco Aironet 802.11 a/b/g PCI driver

- These values compromise the signature for a wireless driver which we add to a master signature database containing all the tagged signatures that are created
- Signatures can be inserted, modified, deleted without affecting other signatures

29

Closeness

- Compute how “close” an untagged signature from the probe request trace is to the signatures in the master signature database

30

Closeness

- Legend:
  - p_n - percentage of probe request frames in the nth bin of T
  - m_n - mean of all probe requests frames in the nth bin
  - S - set of all signatures in the master signature database
  - s - single signature within the set S
  - v_n - percentage of probe request frames in the nth bin of s
  - w_n - mean of all probe request frames in the nth bin of s

$$C = \min(\forall s \in S \sum_{0}^{n} (|p_n - v_n| + v_n|m_n - w_n|))$$

31

Evaluation of Fingerprinting Technique

32

Fingerprint Technique

- Tested with a total of 17 different wireless interface drivers in their default configurations
- Characterized wireless device drivers for
  - Linux 2.6 Kernel
  - Windows XP Service Pack 1 and 2
  - Mac OS X 10.3.5
- 2.4 GHZ Pentium 4 desktop with a Cisco Aironet a/b/g PCI wireless card, running the MadWifi wireless NIC driver

33

Evaluation Characteristics

- Five primary characteristics to be evaluated against
  1. Identification granularity between drivers for different NICs, different drivers that support identical NICs, and different versions of the same driver
  2. Consistency of technique
     - Measure how successful technique is in a variety of scenarios and over multiple network sessions
  3. Robustness of technique
  4. Efficiency of technique with respect to both data and time
  5. Resistance of technique to varying configuration settings of a driver and evaluate the potential ways one might evade their fingerprinting technique

34

Evaluation

- Identified two properties of a device and driver that altered their signatures
  - Whether the wireless device was unassociated or associated to an access point
    - All wireless drivers transmit probe request frames when disassociated from an access point
  - Only applicable to Windows drivers
    - How the driver is managed
    - Slight differences in behavior of probing depending on which option a user chooses to manage their device

35

Master Signatures

- Collected trace data and constructed individual signatures across all 17 wireless drivers
  - Every configuration known to affect the signature and supported by the wireless driver

36

Drivers Used

- Apple
- Cisco
- D-Link
- Intel
- Linksys
- MadWifi
- Netgear
- Proxim
- SMC
- Majority were for Windows

37

Four Configurations

- Unassociated and controlled by Windows
- Unassociated and controlled by a standalone program
- Associated and controlled by Windows
- Associated and controlled by a standalone program

38

Master Signature Conclusions

- 57 signatures were compiled in the master signature database
- Collected four signatures at a time and each signature trace contained a minimum of 12 hours worth of data points
- Changing configurations for some drivers had little impact on the probe request frame transmission rate and consequently the generated signatures were indistinguishable from one another
- After pruning the database of all duplicate signatures, there remained 31 unique signatures
  - Tagged with the corresponding driver's name and configurations

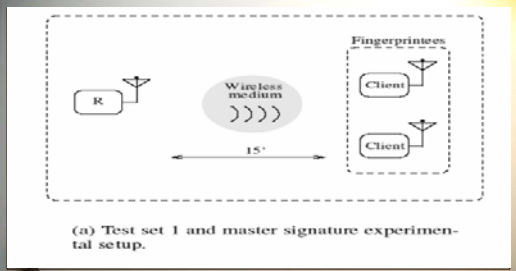
39

Collecting Test Data

- Test Set 1
  - Used unused 30 min trace from each of the 57 raw signature traces
  - Captures the probing behavior of the driver and the signatures can identify their associated drivers with a limited amount of traffic
- Repeated the 57 half hour experiments in two different physical locations
  - Test sets 2 and 3

40

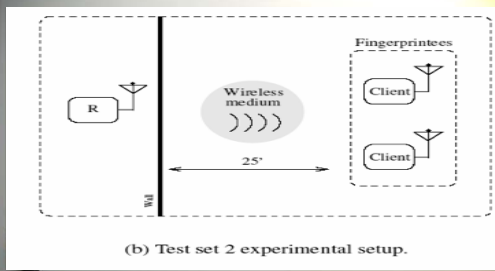
Test Set 1



(a) Test set 1 and master signature experimental setup.

41

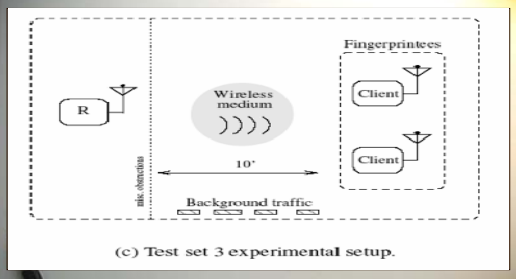
Test Set 2



(b) Test set 2 experimental setup.

42

Test Set 3



(c) Test set 3 experimental setup.

43

Fingerprinting Accuracy

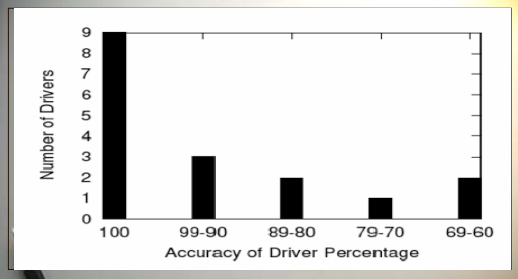
44

Accuracy of Fingerprinting Technique by Scenario

Test Set	Successful	Total	Accuracy
1	55	57	96%
2	48	57	84%
3	44	57	77%

45

Accuracy of Driver Percentage



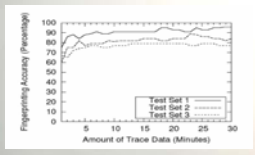
46

Rate of Fingerprinting

- Ideally, fingerprinter would be able to identify a wireless driver in real time after only a small traffic trace.
- Tested with one minute of collected data
  - Increased the amount in one min increments until the full thirty minute trace from each setting was used

47

Rate of Fingerprinting



- Average number of probes detected during one minute of observation: 10.79 across all testing scenarios
- At least 60% accuracy of the three test cases after only one minute of traffic
- Method successfully converges relatively fast on the correct wireless driver and needs only a small amount of communication traffic to do so

48

Limitations

49

Driver Versions

- Can our technique distinguish between two different versions of the same wireless driver.
- Tested wireless drivers with multiple driver versions
  - Technique failed in distinguishing between the different versions of the same driver
- New Versions of a driver might patch previous security vulnerabilities in the driver

50

Hardware Abstraction Layer

- Occurred when testing the MadWifi driver for Linux
- Driver works with most wireless cards containing the Atheros chipset because of the inclusion of a Hardware Abstraction Layer (HAL)
- Lack of driver diversity reduces appeal of fingerpringting wireless drivers
- Magnifies any security vulnerability identified

51

Fingerprinting Prevention

52

Configurable Probing

- Option to disable or enable probe request frames
- Disabling probe requests could be essential in discovering access point.
- Another option
  - Passively listen for beacons and only send probe requests for available networks when manually triggered by the user

53

Standarization

- Difficult to implement
- Effective
- Specify the rate at which probe request frames are transmitted in a future IEEE standard for the 802.11 MAC
- Develop a standard that all driver manufactures can agree upon
- Doubtful standardization can be agree upon due to manufactures disagreeing about expending the power or bandwidth necessary to transmit probe requests at a standard rate.

54

Automated Noise

- Generate Noise in the form of cover probe request frames
- Disguises a driver by masking the driver's true rate of probe request transmission
- Sufficiently random
- Transmit enough cover to confuse the technique
- Also difficult

55

Driver Code Modification

- Change the transmission rate of probe request frames
- Would fool our fingerprinting technique
- Must be open source drivers
- Requires a skilled programmer to alter the driver code
- Many Windows drivers excluded
- since most do not provide source code

56

MAC Address Masquerading

- Earlier Assumption
  - One-to-one mapping of MAC addresses to wireless devices
- Change the device's MAC address to match the MAC address of another device within transmission range
- Tricks the fingerprinting technique into believing probe requests from two different wireless drivers are originating from the same wireless driver.
- Must make sure that the other device is transmitting enough probe request frames to mask its signature

57

Driver Patching

- Not a full solution
- Improves overall security of device drivers as new driver exploits are found
- Community should create more robust patching methods, and improve the overall level of driver security

58

Conclusions

- Designed, implemented and evaluated a technique for passive wireless device fingerprinting
- Capable of accurately identifying the wireless driver used by 802.11 wireless devices without specialized equipment and in realistic network conditions
- Method is effective in fingerprinting a wide variety of wireless drivers currently on the market
- Takes advantage of the implementation-dependent differences between probing algoirthms in order to accurately fingerprint a driver.
- Accuracy ranges from 77-96%
- Can withstand realistic network conditions
- Identifies the wireless driver without modification to or cooperation from a wireless device.

59

Questions?

60