



WIRELESS SECURITY: A COMPARATIVE ANALYSIS FOR THE NEXT GENERATION NETWORKS

¹D.V. Chandra Shekar, ²V.V. Jayarama Krishnaiah, ³Suresh Babu Yalavarthi

¹Assistant Professor, Department of Computer Science, T.J.PS COLLEGE (P.G COURSES)
GUNTUR- 522006 , ANDHRA PRADESH – INDIA

²Assistant Prof , Department of Computer Science, A.S.N. COLLEGE , TENALI , GUNTUR DISTRICT

³Asst. Prof. In Department of Computer Science, JKC COLLEGE, GUNTUR- 522 006

E- mail : ¹Chand.info@gmail.com

ABSTRACT

New wireless technologies such as WiMAX, NFC and ZigBee are rapidly being adopted, along with existing wireless standards such as Bluetooth, Wi-Fi, GSM and other cellular technologies. Bluetooth and Wi-Fi have already become notorious for severe security shortcomings during their relatively brief existence. New vulnerabilities and exploits are reported and demonstrated every week on live public networks. The credibility of these wireless technologies has been damaged by security incidents, stemming from fundamental problems in requirement gathering, implementation quality and protocol design. Despite boasts of hardened security measures, security researchers and black-hat hackers keep humiliating vendors. On the other hand, GSM and various descendant technologies have been almost 100 percent free of security incidents. What can be done to avoid making the same mistakes all over again with new emerging wireless technologies such as WiMAX? What is the anatomy and evolution behind security problems, and why have some cellular technologies been almost problem-free? This paper draws from the past and current state of existing wireless technologies and reflects experiences with emerging technologies. It describes how robustness-testing techniques can be used to assess the security of the available implementations and give statistics about the current state of affairs of Bluetooth and Wi-Fi. Quality and reliability improvements in these implementations will lead directly to decreased development and deployment costs, as well as increased public acceptance and faster adoption.

Key words: *WiMax , Bluetooth , Wi-fi , GSM and ZIG BEE*

1. INTRODUCTION

When discussing the security of wireless technologies, there are several possible Perspectives. Different authentication, access control and encryption technologies all fall under the umbrella of security. Although relevant and important building blocks for overall security, these are not the focus of this paper. Instead, it will

explore the problems at the implementation level of the current wireless access technologies and their real world implications. The problems are explored through one attack category, namely fuzzing, and the remediation through systematic robustness testing. This is because most security attacks do not exploit features in wireless technologies, but they abuse various

implementation mistakes in the products. Robustness testing is one of the most effective black-box assessment technologies for security and reliability problems. All robustness tests presented in this paper have been conducted with the DEFENSICS product family, and can be repeated by third parties to verify the research results. This research is based on the work of various security personnel works The study has been conducted during the past 10 years, between 1996 and 2007. Still, the oldest individual test results date backs only to 2006, and therefore should indicate the current state of maturity in wireless products. To protect the reputation of the software companies, A disclose details of individual vulnerabilities, nor the names of any products tested in these assessments.

This paper is focusing on Bluetooth and Wi-Fi, with some preliminary information on WiMAX and other new wireless technologies. Each case is has an introduction, the test results and the threat evolution section, which includes an examination of the drivers behind the threats and security incidents. The threat evolution section for WiMAX is naturally forward-looking, drawing from the cellular, Wi-Fi and Bluetooth experiences, as little real-word data currently exists.

1.1. Security and Robustness

Today, security problems plague the software products used to access the vast Internet, operating systems, WWW browsers and e-mail programs. These all have had their share of reported problems. Significant portions of these vulnerabilities are robustness problems caused by careless or misguided programming. The Internet’s “underground community” searches for these flaws using non-systematic, ad-hoc methods, and then publishes their results for profit. The large number of reported problems from some software packages can be explained by the huge attention they have received, and also by the numerous flaws they contain. Security assessment of software by source code auditing is expensive and laborious. There are only a few methods for security analysis without access to the source code, and they are usually limited in scope. This may be one reason why many major software vendors have been stuck randomly fixing vulnerabilities that have been found and providing countless patches to their clients to keep the systems protected. This is also known as the “patch-and-penetrate race.”

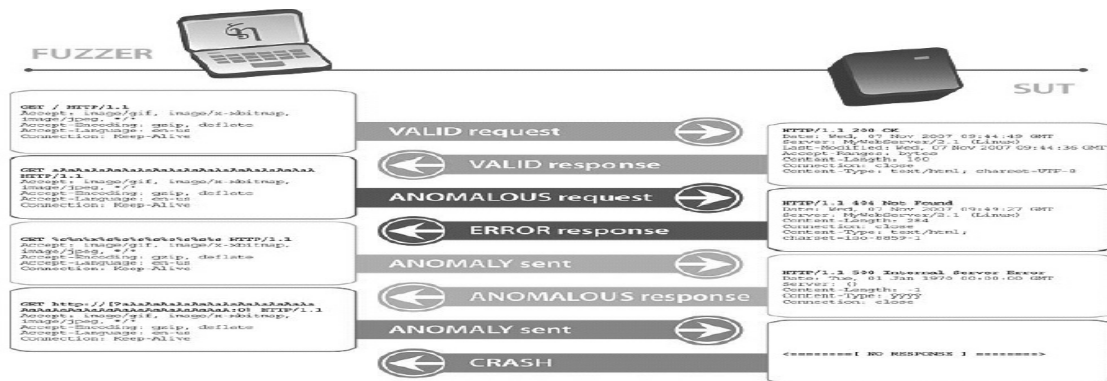


Figure 1: Example test cases sent to the system under test (SUT) with related responses



Negative testing of communication protocols is called robustness testing or fuzzing. The robustness testing is based on the systematic creation of a very large number of protocol messages, from thousands to several million test cases, containing exceptional elements simulating malicious attacks. The method provides a low-cost pre-emptive way of assessing software robustness. Robustness indicates the extent to which software can tolerate exceptional input and stressful environmental conditions. A piece of software that is not robust fails when facing such circumstances. A malicious intruder can easily take advantage of robustness shortcomings and compromise the system running the piece of

software. A large portion of information security vulnerabilities reported to the public is caused by robustness weaknesses. All robustness problems can be exploited by causing denial-of-service conditions by feeding the vulnerable component with maliciously formatted input. Figure 1 shows some examples from robustness testing, with both clean and anomalous messages sent to the system under test and example responses from the target of the tests. There are no false positives in robustness testing. A found failure is always a critical failure, such as a crash or a memory leak. Some of the mistakes are exploitable – an attacker can execute malicious code on the target system. An example is a buffer overflow type of robustness flaw that can almost always be exploited to run externally supplied code in the vulnerable component.

1.2. Wireless Vulnerabilities and Incidents

Wireless technology sometimes allows threats, attacks and vulnerabilities to enter the wireless

space. Malicious people will exploit any known weaknesses through service attacks, worms, spam, malware and man-in-the-middle attacks.

Wireless networks have three additional aspects that make the security of wireless networks even more challenging than the security of fixed networks:

- Wireless networks are always open
- Attackers can connect into the network from anywhere and from any distance
- Attackers are always anonymous

Wireless networks are always open – Physical media does not protect them. Any device that implements the same radio interface can access a wireless network. One common assumption is that wireless technologies are secure when authentication and encryption are properly deployed. Looking closely at the operation of related protocols, there are many message sequences that take place before the authentication. These message sequences can always be attacked regardless of the deployed security measures. Attacks are not limited by location or distance. The distance from where the attacker can reach the wireless network is only limited by the power of the transmitter. For example, Bluetooth attack tools are known to have several-mile radiuses, although valid usage

scenarios would never attempt such range of coverage for Bluetooth. Attackers are always anonymous. Although a valid user can be pinpointed with good accuracy, an attacker can use directed antennas that will only target a selected victim. It is impossible to guarantee detection of malicious users in wireless networks. As stated above, an attacker can also always attack the



message sequences that happen before the authentication of the device and thus avoid identification. After the analysis above, it is interesting to observe that the threats have realized for these technologies with certain characteristics:

- Easy, low-cost access to technology
- Programmable environment, such as stack access in PC

- Motivation to hack
- Complex technology

Both Bluetooth and Wi-Fi have had plenty of security incidents fall into these categories, whereas almost-security-incident-free cellular technologies have not.

Figure 2: Examples of attacks and vulnerabilities in wireless technologies, drawn from public media such as Slashdot, US-CERT, SecurityFocus, News.com and InfoWorld

2. CASE STUDIES

2.1. Bluetooth Robustness Testing

Since its inception in 2002, Bluetooth has seen an increasingly accelerating adoption rate, reaching over 2,700 certified end products by the end of 2007. Bluetooth has had a variety of security incidents in the past, varying from mildly annoying events to severe information leaks, some

employing techniques for stealing the contact information from the mobile phone. As Bluetooth technology is added to more sophisticated and critical devices, security and robustness are becoming even more important. A robustness test has found a number of critical flaws in Bluetooth implementations. Examples include cases where mobile phones have been rendered completely useless as a result of negative tests, with the only remediation being reflashing the device. Anyone

could conduct attacks similar to these tests and, in the case of modern smart phones, they could result in a loss of important data. As Bluetooth finds its way to new application areas like medical devices (with Bluetooth Medical Device Profile, MDP), it is important to harden the implementations against malicious attacks and unintentional bad input. In the following sections, we will give results of real-world testing and insights into the Bluetooth threat evolution. The Bluetooth robustness tester currently covers all of the major profiles and stack layers, starting from L2CAP to individual profiles, such as HFP and HSP. The Bluetooth robustness tester supports 21 profiles out of the 30 or so

profiles specified in Bluetooth, and the number of supported profiles constantly increases as new profiles are introduced to the specifications.

2.1.1. Test Results

Bluetooth consists of various profiles that implement the different features used in the various devices. These profiles can be implemented on different layers on the protocol stack. Figure 3 shows one view of Bluetooth components by enumerating the interfaces supported by the Bluetooth robustness testing tool.

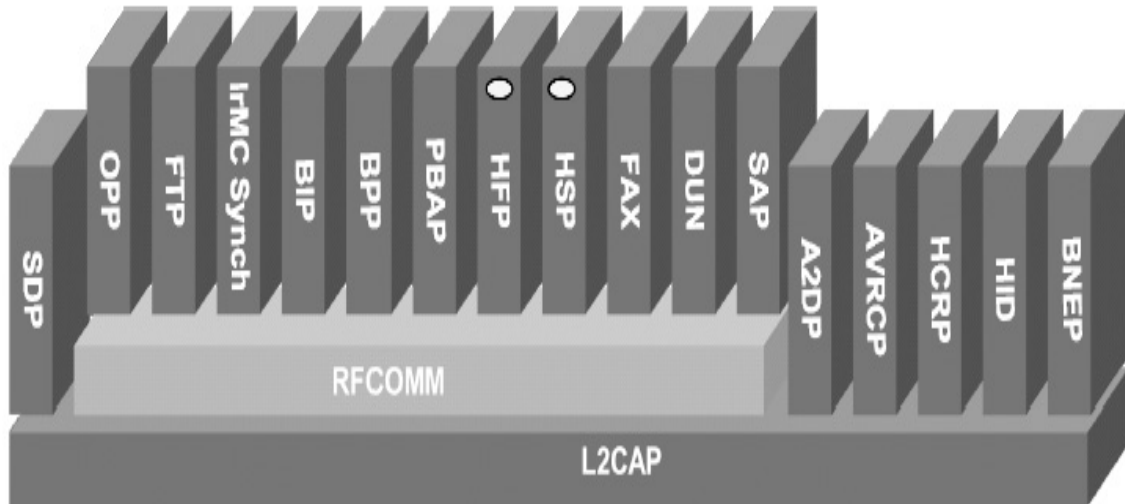


Figure 3: Selected Bluetooth interfaces and profiles support (2007)

Table 1 shows the results of testing the robustness of 31 different Bluetooth implementations. Only three implementations survived all tests that were executed, and all other products had problems with at least one profile. Failure modes varied. Most of

the Bluetooth-enabled embedded products simply crashed when tested with any level of robustness testing. Sometimes, the result from the testing was that the device ended up totally corrupt, requiring re-programming of its corrupt flash memory to become operable.

Interface/profile	Number of implementations tested with a fuzzer	Number of implementations that failed in the test	Percentage of failed products
L2CAP	31	26	84%
SDP	31	24	77%
RFCOMM	31	28	90%
A2DP	2	2	100%
AVRCP	3	3	100%
HCRP	1	1	100%
HID	1	1	100%
OPP	15	12	80%
FTP	5	5	100%
IRMC Synch	1	1	100%
BIP	1	1	100%
BPP	1	1	100%
HFP	5	2	40%
HSP	5	2	40%
FAX	2	0	0%
DUN	5	2	40%
SAP	4	4	100%

Table 1: Test results from Bluetooth fuzzing with Other DEFENSICS

2.1.2. Threat Evolution

The Bluetooth started gaining popularity. One of the earliest and well-known Bluetooth exploits was BlueSnarf, discovered in 2003. BlueSnarf allowed a remote attacker to steal phonebook or calendar entries from mobile devices with vulnerable implementation of an OPP profile. With Bluetooth, the cheap, easy access to technology and programmable environment were available early on. Moreover, devices to play around with and hack were within reach of even the most resource-constrained hackers. Furthermore, the Bluetooth stack as a whole is very complex, giving a lot of room for implementation-level errors. Based on experience from testing the security and robustness of the Bluetooth stacks, it should be noted that the new vendors have repeated the same mistakes that more established vendors experienced in the past,

and that the new profiles are often plagued by security related implementation errors.

It can be speculated that these factors together have driven a fast evolution and the high quantity of security incidents Bluetooth has already experienced.

2.2. Wi-Fi Robustness Testing

Wireless IP carrier protocols are just like any other packet-based protocols. They can be attacked on various different layers, starting from Layer 2 and up, including IPv4, IPv6 and all application protocols that the wireless infrastructure depends on. The example in

Figure 4 shows a test case drawn from a DEFENSICS robustness tester that systematically breaks the packets in 802.11 frames for testing purposes. During the testing and development of the DEFENSICS robustness test

suite for Wi-Fi, several weaknesses were found from both Wi-Fi access points and from Wi-Fi client devices. Some of the found issues were kernel-level problems on well-known operating systems. Typically, the results were not as dramatic as with Bluetooth, but still serious denial of service attacks and exploits could have potentially derived from these weaknesses. From the industry

perspective, the Wi-Fi hacks are more serious than the Bluetooth hacks since the technology is typically used in a more critical environment. For example, the hacks allowing kernel-level remote code execution on a laptop can lead to significant damages for the owner of the laptop and brand damages for the vendor.

```

+ Frame 2 (54 bytes on wire (54 bytes captured)
  IEEE 802.11
    Type/subtype: Probe Request (4)
    Frame Control: 0x0040 (Normal)
    Duration: 0
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Source address: GemtekTe_cf:4b:5e (00:90:4b:cf:4b:5e)
    BSS Id: Broadcast (ff:ff:ff:ff:ff:ff)
    Fragment number: 0
    Sequence number: 394
  IEEE 802.11 wireless LAN management frame
    Tagged parameters (30 bytes)
      SSID parameter set: "chickenshack"
        Tag Number: 0 (SSID parameter set)
        Tag length: 255 ← INVALID LENGTH IN TAGGED PARAMETER
        Tag interpretation: chickenshack
      Supported Rates: 1,0 2,0 5,5 11,0
        Tag Number: 1 (Supported Rates)
        Tag length: 4
        Tag interpretation: Supported rates: 1,0 2,0 5,5 11,0 [Mbit/sec]
      Extended Supported Rates: 6,0 9,0 12,0 18,0 24,0 36,0 48,0 54,0
        Tag Number: 50 (Extended Supported Rates)
        Tag length: 8
        Tag interpretation: Supported rates: 6,0 9,0 12,0 18,0 24,0 36,0 48,0
  
```

Figure 4: Fuzzing 802.11 frames with the fuzzer

The DEFENSICS robustness tester tortures 802.11 from various perspectives. 802.11 features and resulting messages and elements that are tested for robustness include:

- . Management frames
- . Open authentication
- . QoS parameters
- . WEP
- . WPA1
- . WPA2

2.2.1. Test Results

Table 2 summarizes results from the testing of seven wireless access points with fuzzing against different protocols. WLAN refers to the testing of the MAC layer features, whereas the other protocols are on higher-level interfaces such as transport and application layers.



	AP1	AP2	AP3	AP4	AP5	AP6	AP7	fail-rate
WLAN	INC	FAIL	INC	FAIL	N/A	INC	INC	33%
IPv4	FAIL	PASS	FAIL	PASS	N/A	FAIL	INC	50%
ARP	PASS	PASS	PASS	N/A	FAIL	PASS	PASS	16%
TCP	N/A	N/A	FAIL	N/A	FAIL	PASS	N/A	66%
HTTP	N/A	PASS	FAIL	PASS	INC	FAIL	FAIL	50%
DHCP	FAIL	FAIL	INC	N/A	FAIL	FAIL	N/A	80%
fail-rate	50%	40%	50%	33%	75%	50%	25%	

Table 2: Failure rates of various Wi-Fi access points when tested with DEFENSICS

Only the total crashes repeatable with a certain test case or test case sequence are diagnosed as failures. Inconclusive (INC) indicates a crash-level problem, but one that could not be easily repeated, and therefore was not included in the shown failure ratings. As the results indicate, all the access points failed with some of the DEFENSICS protocol tests, but more alarmingly there were access points that failed with almost everything that was run against them. Another interesting observation is that IPv4 caused problems for many implementations. A common assumption is that the more established and stable the protocol is, the less implementation-level faults exist. These test results clearly contradict that.

2.2.2. Threat Evolution

It took a relatively long time from the emergence of Wi-Fi as a commonplace technology to the first public security incidents to appear. Wi-Fi as a technology matured in late 90s. Some initial frame injection drivers were introduced between 2002 and 2004, but they were not maintained and did not evolve into general-purpose frame injection and hacking framework. Recalling the characteristics described in section 1.2, it can be noted that the easy, low-cost access to technology and a programmable environment was not easily obtained early on. The implementation-level

vulnerabilities started appearing around mid-2006. Lorcon was the first general framework that did abstract the HW intricacies and made frame injection relatively easy. Around the same time, Month of Kernel Bugs sought to disclose a kernel-level vulnerability from the common operating systems for each day of the month. Speculatively speaking, Wi-Fi might also have been a less interesting target compared to Bluetooth, which enabled the creation of “cool” hacks.

2.3. Emerging Wireless Technologies

Based on the experiences with the cellular Bluetooth and Wi-Fi technologies, educated guesses can be made on emerging technologies, such as WiMAX, NFC and others. Some preliminary tests were run against the WiMAX base laboratories in order to compare the security with the more well-known technologies. WiMAX is an emerging wireless technology that was originally intended for solving the “last-mile” problem in the areas where wired connections were not desirable. This was the so-called “Fixed WiMAX” (802.16-2004), or 802.16d. With the emergence of “Mobile WiMAX” (802-16e-2005) or 802.16e, carriers are looking at WiMAX as one of the technologies for delivering broadband content for mobile users. At the same time,



WiMAX security is getting attention in public discussions. Currently, there are two schools of thought: one is expecting to see security issues similar to Wi-Fi and Bluetooth

and the other believes that the threats are not severe, as security is built into WiMAX. It is important to note that most of the time when someone claims WiMAX is secure, they are referring to encryption and authentication, which were initially weak with Wi-Fi. Potential for the fuzzing type of attack, however, has received little attention.

2.3.1. Preliminary Test Results

At the time of testing, the over-the-air attack vector was unavailable, so the testing was conducted against the interfaces that were visible from the fixed network side. The following suspicious problems were found with the WiMAX equipment:

- Reboot of IPv4 stack in system under test that repeatedly caused state of denial of service when receiving large amount of abnormal IPv4 packets
- Crash and reboot of system under test when receiving SunRPC request packet with length anomalies
- Crash and reboot of system under test when receiving RPC request packet with overflow anomalies

These defects are such that they could manifest themselves without being explicitly exploited. This means the resulting defects are receiving badly formatted network messages as part of normal operation. Furthermore, it appears that anyone able to access the device would be able to trigger some of the found issues. As a mitigating factor, it is noted that found issues are likely to be exploitable

only if the user has the direct IP address to the base station's management interface or IP address. The testing conclusively proved that the software stack employed in WiMAX base stations is not free of the implementation-level errors anymore than those of the other wireless technologies.

2.3.2. Potential Threat Evolution

A comparison between Wi-Fi and Bluetooth has indicated that the complexity of the software stack gives indications about the amount of vulnerabilities. Even the very cursory testing of the well-established, higher-layer protocols of WiMAX has uncovered weaknesses. The MAC layer of 802.16e is fairly complex, settling somewhere between Wi-Fi and Bluetooth, but there is no reason to assume it will be free of the implementation-level vulnerabilities. As such, it will be susceptible to the fuzzing type of attacks. Assuming that this is a true assessment, as we believe it to be, the amount of security incidents will depend on couple of variables:

- What kind of access will be publicly available to MAC layer?
- What is the motivation to hack WiMAX?

It can be predicted that the MAC access will not be as readily available as it was for Bluetooth, but WiMAX is expected to make its way into laptops as well as mobile handsets. As Wi-Fi has shown, it will be only a matter of time before the open SDKs will be available for the general public. Looking at hacking from the motivation perspective, the mobile station side is likely to be as attractive a target as any consumer device. Base station hacking might not be as commonplace, but the stakes are much higher. A breach in security of the



base station could enable service disruptions or the exploits in a core network opening behind the base station. As a reminder, the base station can be a portal for an anonymous long-distance attack.

3. CONCLUSIONS

To understand the threats posed by malicious fuzzing and how systematic robustness testing can pre-emptively eliminate the threats, we have collected results from three different use cases, namely Bluetooth, Wi-Fi and WiMAX. We have looked at the different characteristics of these technologies and how they affect the threat evolution. We have also briefly mentioned cellular technologies, which enjoyed immunity in the past, as well as some emerging new wireless technologies. We can continue to evaluate the security of the present and future wireless technologies to improve the ability to identify the vulnerable ones. This paper only gives basics for assessing whether the wireless technology an organization is about to deploy would be susceptible to a malicious fuzzing attack. Robustness testing has been found to be extremely cost-effective and a fast security assessment tool. On average, testing found problems in 90 percent of the devices tested. In light of the test results from WiMAX tests, there is no reason to believe that other WiMAX interfaces, including those used on the physical layer, would be any less free of bugs than the tested, classical IP-based interfaces tested in the WiMAX study. Using any available tools to verify and guarantee the robustness of 802.16d, 802.16e and the MAC layer is highly recommended when robustness testers will be available that are similar to what we currently have for Bluetooth and Wi-Fi.

REFERENCES

- [1]. Miller, Barton; Fredriksen, Lars; and So, Bryan: An Empirical Study of the Reliability of UNIX Utilities. In: Communications of the ACM 33, 12. 1990.
- [2]. Lämsä, Jarkko; Kaksonen, Rauli; and Kortti, Heikki: Codenomicon Robustness Testing - handling the infinite input space while breaking software. Codenomicon whitepaper. <http://www.codenomicon.com/resources/whitepapers/code-whitepaper-2006-06.pdf>
- [3]. Kaksonen, Rauli; Laakso, Marko; and Takanen, Ari: Vulnerability Analysis of Software through Syntax Testing. University of Oulu, whitepaper. 2000. <http://www.ee.oulu.fi/research/ouspg/projects/analysis/WP2000-robustness/>
- [4]. Bluetooth SIG. <http://www.bluetooth.org>
- [5]. Marcel Holtmann: Blue Snarf. http://trifinite.org/trifinite_stuff/bluesnarf.html
- [6]. H D Moore: Apple Airport 802.11 Probe Response Kernel Memory Corruption. <http://projects.info-pull.com/mokb/MOKB-01-11-2006.html>
- [7]. Wright, Josh; Kershaw, Mike: Extensible 802.11 Packet Flinging. Shmoocon 2007.
- [8]. LORCON (Loss Of Radio CONnectivity). <http://802.11ninja.net/lorcon>
- [9]. Barbeau, Michael: WiMax/802.16 Threat Analysis. <http://www.scs.carleton.ca/~barbeau/Publications/2005/iq2-barbeau.pdf>