Editors: Marcus Sachs, marcus.sachs@verizon.com
David Ahmad, drma@mac.com

Wireless Malware Propagation

A Reality Check

Stefano Zanero
Politecnico di Milano

In recent years, several authors (including myself, I confess) began to work on the concept of security attacks against wireless communication protocols—in particular, the propagation of malware through them. The idea challenged and thrilled us, mainly because it brought back the concept of physical and geographic interaction in attack scenarios.

It was fun to design covert attack devices and evaluate the Bluetooth user population's exposure to them.[1] It was also fun, I bet, for other authors of articles on wireless router subversion[2] and malware propagation on Wi-Fi networks[3] to envision their attacks.

However, I believe that the latest "developments" on these threats are stepping progressively away from reality and into an abstract, academic world of their own—something that might be just as fun, but that should be brought back into perspective when assessing the actual risks related to these scenarios. Let's see why, with two distinct examples.

The Unlikely Router Contagion

A recent article that received some news coverage outside scientific circles describes the propagation of a virus over a set of overlapping wireless LAN networks in an urban area.[4] The work is theoretically and mathematically interesting, but, sadly, it describes a type of malware that's highly unlikely to appear in the real world, for several reasons.

In their introductory assessment of wireless worms' prevalence, the authors mix together very different threats. They actually refer to attacks described in other works,[5] which come from the Internet side of the router connection and need to interact with a client on the Wi-Fi network itself. Something similar happened in the wild with the Zlob Trojan (which, by the way, attacked wireless and wired routers indifferently, to stress that the wireless component was irrelevant; see http://en.wikipedia.org/wiki/Zlob_trojan). Zlob infected clients on the LAN and then tried to guess administration passwords using a built-in list of default username/password combinations. If successful, it would then alter the Domain Name System (DNS) records to perform a man-in-the-middle (MITM) attack. Similarly, the worm propagation described in Periklis Akritidis' work uses clients roaming from network to network to spread a contagion in a local metropolitan area (a so-called "wildfire" worm).[3] All of these are very realistic attack vectors.

On the other hand, a router-to-router attack, such as the one Hao Hu and colleagues envisioned,[4] is much more difficult to execute than the article maintains. First, domestic wireless routers (as opposed to core Internet routers) are a very diverse family of devices. In virology (and even in the biological world), homogeneity breeds danger, whereas heterogeneous devices are less susceptible to a digital contagion. This is evident, for instance, in the world of mobile phones, in which device heterogeneity is a key obstacle to virus propagation. We look at this further later in the article.

Hu and his colleagues theorize the creation of a universal bogus firmware, which isn't going to happen, as should be evident to any reader with experience in embedded networking devices. It's difficult to write reliable bogus firmware for similar, but not identical, platforms, let alone for completely different ones. Of course, attackers could target a popular and easy to customize model (such as the Linksys WRT54G, for instance), but this will, of course, reduce the number of possible targets to way below the numbers the authors use. To my knowledge, the only known example of firmware malware in the wild was the Bluepill botnet (http://dronebl.org/blog/8), which was deployed from the Internet and, once again, didn't deal with the wireless side of the router.

One reason attackers are so

shy about touching the wireless connectivity part is, of course, stealthiness. To use the router's radio apparatus for scanning and deploying malware on other networks, an attacker would need to disrupt wireless service to clients. Users would likely notice this, which would lead to disinfection or disconnection of the router—a big no-no for a malware author.

But setting this huge problem aside for a moment, we're left with the idea of uploading bogus firmware on open routers with a specific model. Earlier work has already explored this more sensibly, and the authors estimated that 34 percent of observed routers were the appropriate models.[2] Reading through this paper, it becomes obvious that this percentage is an overestimate, because it comes from a potentially biased projection of a very limited subset of identifiable routers. But we can still use it as an upper bound: less than one-third of observed routers could run a hypothetical attack firmware. Furthermore, in that article, only 16.7 percent of such routers are shown to have default settings, which the authors translate in an overestimated 10 percent of routers with no password or default passwords. On the other hand, Hu and his colleagues assume this to be a staggering 50 percent.[4] This brings the targets down from 50 percent to less than 5 percent—quite a show-stopper for any aggressor.

While we're at it, in Hu's work, the percentage of encrypted networks is extremely low (compared to many other studies) and actually doesn't even match the Web site cited as a source (which we report for comparison in Table 1).

On the other hand, using data in other work,[6] cracking a 104-2 bit WEP key takes approximately 1 minute of data dumping and 3 seconds of computation using a 1.7-GHz Pentium M processor and 3 Mbytes of RAM. WRT54G devices have various processors on board, but the best ones are 200-MHz MIPS32 processors with 16 Mbytes of RAM; it would seem likely that the authors actually overestimated the times for cracking a WEP-encrypted network. Bruteforcing administrative passwords might make sense in theory, but a million-password list is impossible to use in such a setting because it would easily exhaust the available device memory. Also, the fact that "wireless routers don't have bruteforcing protections" (as Hu and colleagues stated in that work) isn't stated anywhere else and should be demonstrated.

Dulcis in fundo, Hu assumed that each infected router in a given infection cycle will attack only "new" routers not previously attacked. This is impossible to ensure (given that it would require a level of distributed coordination of the malware, which is difficult to envision without a botnet-like command-and-control structure—and if you're going to make this a botnet, you might as well infect those devices from the Internet side). This assumption, which is seemingly of small importance owing to its confinement in the work's appendix, actually significantly biases the propagation data because it removes a reinfection term that, in proximity-based infections (such as this one), is overwhelming.

The net result is that although the mathematics in Hu's work is fun to read (if you're an engineer, and thus inclined to this type of fun), the type of malware described is highly unlikely to appear, to put it mildly. It should exploit weaknesses on a heterogeneous population, without relying on client infection. It would work on only a small fraction of routers, which are unlikely to form a connected cloud; it would disrupt regular network use, and thus be noticed. Also, the simulation parameters are far from real-world data. But the most critical question left unanswered here is why—why should an attacker run such a complex attack, if (as shown in the Bluepill work [http://dronebl.org/blog/8]) he or she can obtain significantly high penetration through routers' wired Internet in-

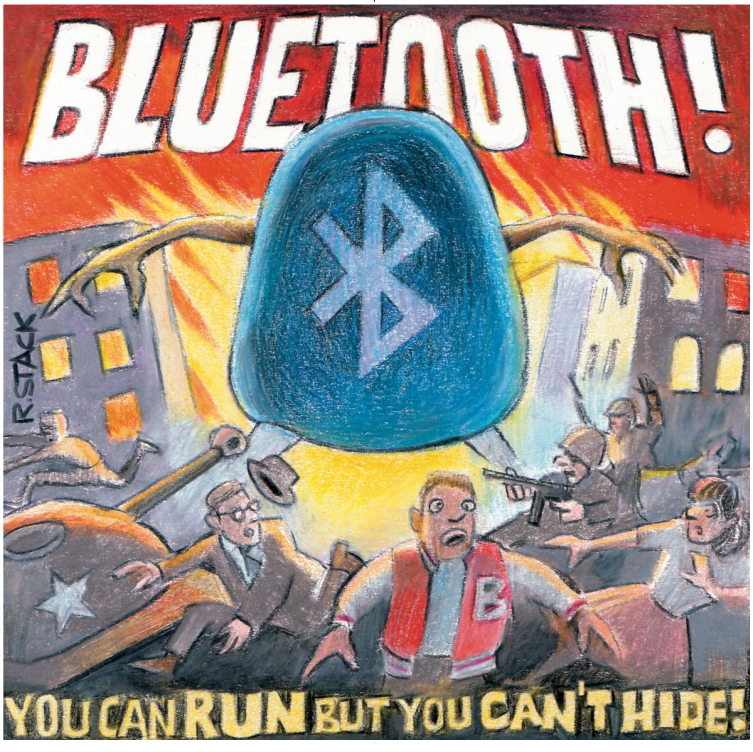

Table 1. Data on network protection from wigle.net.

TYPE OF PROTECTION	NUMBER OF NETWORKS	%
With WEP	8,405,111	48.0
Without WEP	6,078,663	34.7
WEP unknown	3,008,994	17.2

terfaces, in an easier and stealthier way? Even then, a compromised router is of little use as a bot (because its data storage capabilities are extremely limited). Interception of user data is an interesting and worrying perspective, but then again, what's the specific advantage to creating a wireless-propagating malware as opposed to an Internet-based worm or drive-by attack to do that?

This is what happens when mathematics comes first and actual risk assessment comes later, if ever. When I shared these remarks with the editorial board of the journal in which this article appeared, I obtained an answer that actually provoked the line of thought in this article: "an argument about whether a given model does or doesn't 'reflect the real world' (especially in [an] as rapidly evolving field as wireless) is not a good use of [our] letters pages. I don't think there is much danger that the model published will be taken as literally exact." In other words, this is just maths, folks, nothing's really happening here, move along please.

Bluetooth Epidemics on Paper

Bluetooth is a short range, short-wave radio communication protocol that was designed as an alternative to traditional infrared communication (such as the IrDA standard) to create small-range personal area networks of mobile devices. An important improvement over IrDA is that Bluetooth doesn't require a line of sight among devices, which incidentally also makes it useful as a malware propagation vector or attack target, because it allows for "casual" or unwanted interaction.

Even if Bluetooth is theoretically quite robust, since late 2003, several security issues in various specific implementations of the standard stack have surfaced. Such attacks are well described online at www.trifinite.org, and they allow different degrees of data access (from the agenda to any file on a vulnerable device) and communication interception, up to and including taking full control of the phone, something that can be effectively used to transform a telephone into a spyphone (www.secuobs.com/news/05022006-bluetooth5.shtml). To further stress that implementation glitches lurk below the surface, in June 2008, Microsoft released an interesting security bulletin that reported a vulnerability in the Windows Bluetooth stack that could allow remote code execution with system privileges (www.microsoft.com/technet/security/Bulletin/MS08-030.mspx). Most of these attacks can be run from cell phones or portable devices, or from a distance using long-range antennas and modified Bluetooth dongles (up to ranges on the order of one mile).

These flaws demonstrate how, in many cases, it's possible to steal information from mobile devices, control them from a distance, make calls, send messages, or even connect to the Internet. In computer systems, this type of problem is traditionally handled with patch release and application. However, this approach doesn't extend to cell phones because, in most cases, a firmware update can be performed only at service points and shops, not by the customers themselves; so, many vulnerable phones and firmwares keep circulating long after a vulnerability is discovered.

Viruses for mobile devices propagating over Bluetooth also reportedly exist. The propagation of a Bluetooth virus can occur in several different ways. The most common is through simple social engineering. The worm sends messages with copies of itself to any device that comes into range through an OBEX push connection (OBEX is the protocol used for exchanging binary objects via Bluetooth). Different profiles for this service exist, and "push" is the profile generally used for phone-to-phone occasional transfers without authentication (such as for exchanging electronic business cards). Much like in the case of email worms and Trojans, the receiver, upon finding an "attractive" message on the cellular phone with the invitation to download and install an unknown program, often has no clue that this can pose a danger. For instance, Cabir—one of the first cell phone worms and the first case of malware able to replicate itself only through Bluetooth—used this technique (www.symantec.com/security_response/writeup.jsp?docid=2004-061419-4412-99). Using some vulnerabilities, seemingly innocent files such as images could be used as viral propagation vectors (www.zerodayinitiative.com/advisories/ZDI-08-033). Bluetooth attacks, such as the ones described earlier, could also be employed, but because they're quite platform-specific, they're a difficult and unreliable means of propagation when compared to social engineering's simplicity.

BlueBat: Bluetooth Epidemics in the Real World

Researchers have proposed several models for Bluetooth worm propagation, almost invariably showing

great propagation potential.[1,7–9] Antivirus vendors also claim every year to be the "year of mobile malware," but such predictions constantly fail to materialize.[10]

To assess the effective prevalence of such targets, we're building a set of Bluetooth honeypots named BlueBat.[11] The name is a joke on the broader research project, WOMBAT (Worldwide Observatory of Malicious Behaviors and Attack Threats) of which Bluebat is a part. WOMBAT is a European research project that aims to provide new means to understand both existing and emerging threats that target the Internet economy and Net citizens (www.wombat-project.eu).

Several authors have already proposed the concept of "Bluetooth honeypots," but without code release or results (http://trifinite.org/Downloads/syscan2005_slides.pdf). In particular, no extensive deployment has ever been attempted (the only such experience being briefly reported in a post on F-Secure's blog at www.f-secure.com/weblog/archives/00000836.html, without actually publishing results and in a set up more similar to our former experiments[1]). Our aim, on the contrary, was to develop a practical approach in which inexpensive sensors could be realistically deployed to gather information on a wide and diverse population.

Our preliminary results are threefold. First, we positively confirmed what we already demonstrated in previous work—a lot of Bluetooth devices are kept enabled and in visible mode, paving the way for potential attacks.[1,11] Also, a relatively high percentage of users (up to 8 percent) will carelessly accept files via Bluetooth from unknown sources. This should spell trouble. However, during our initial tests, we received a limited number of files transmitted to the honeypots, only one of which was potentially harmful but incorrectly transmitted. So, even if there are anecdotal tales of Bluetooth infection, the threat seems to be of limited diffusion as of today.

The explanations that come to mind are the difficulty of writing effective code that works across different mobile platforms and uses Bluetooth, even in the case of a benign application. This also creates a nonuniform population in which it's difficult to envision a common bug paving the way for automated worm transmission. Second, as our tests show, "casual" transmission of a file is quite difficult: a simple scan for devices takes seconds or even minutes, and then transmission happens, for each device, over several tens of seconds. During this time lapse, shifting positions might well place the target device out of range. This was actually predicted in other work,[12] which went against the common perception that mobility helped spread such worms.[8] Also, because Bluetooth is transmitted on a 2.4GHz band, which is absorbed by water, the human body itself acts as a shield and can easily interrupt transmission. So, in the wild, worm transmission can effectively happen only in a semistatic scenario. These preliminary results cast many doubts on Bluetooth's viability as a worm propagation mechanism, and on the effectiveness of former Bluetooth spreading models, including our own.[1] This conclusion creates a need for updated models of viral propagation, and for reevaluating the infection likelihood, even in closed environments, which we're currently working on.

Wireless and mobile security, and, in particular, worm propagation over wireless networks, is an interesting and novel concept. It challenges and thrills us, creating appealing newspaper headlines along the way. However, we must be sure to check our models against reality (even if, for some scientific communities, this isn't particularly important), and after predicting threats that failed to materialize, we must be able to understand where we went wrong.

We're very likely to see an increasing number of wireless attacks in the future. Targeted penetration of wireless networks, or reflected attacks brought through roaming clients, will surely occur, and the "wildfire" worm scenario could well materialize. On the other hand, router-to-router attacks aren't going to happen anytime soon, no matter how appealing they look on paper (especially if the wrong parameters are chosen for the simulations). Not caring about "whether a given model does or doesn't reflect the real world" (as I was answered by the editor when I tried to comment on Hu's work) is a serious issue for anybody involved in security choices, and engineers should therefore approach with caution any result that comes out of models grounded on thin air. Otherwise, we might end up deploying antivirus software on wireless routers, as opposed to doing something more sensible.

Speaking of antivirus software deployed to respond to unlikely threats, Bluetooth worms aren't yet here (not in a raging fury, at least), and in spite of all our models, it's not likely they'll emerge at all. Bluetooth is just too unreliable to give birth to a real pandemic, unless something major changes in range, stability of communication, and most important, unless the mobile world evolves and provides a way to reliably write portable applications. On the other hand, targeted attacks on high-profile devices are probably happening and will keep happening below our radar—because antivirus software isn't designed to

deal with them but only with self-propagating malcode that hasn't even left the zoo.

Performing risk assessment is still in many ways an art, rather than a science. But even the most skilled gypsy will have trouble reading the future in a stained crystal sphere: it's high time to review the mathematical models we use and ensure that they reflect reality, after all. □

Acknowledgments

I gratefully acknowledge the help of Kostas Anagnostakis (Institute for Infocomm Research, Singapore), and of Politecnico di Milano colleagues Carlo Piccardi, Renato Casagrande, and Guido Salvaneschi. BlueBat was partially supported by the European Commissions through project IST-216026-WOMBAT, funded by the Seventh Framework program. The opinions expressed in this article are exclusively my own.

References

1. L. Carettoni, C. Merloni, and S. Zanero, "Studying Bluetooth Malware Propagation: The Bluebag Project," *IEEE Security & Privacy*, vol. 5, no. 2, 2007, pp. 17–25.
2. A. Tsow et al., "Warkitting: The Drive-by Subversion of Wireless Home Routers," *J. Digital Forensic Practice*, vol. 1, no. 3, 2006, pp. 179–192.
3. P. Akritidis et al., "Proximity Breeds Danger: Emerging Threats in Metro-Area Wireless Networks," *Proc. 16th Usenix Security Symposium*, Usenix Assoc., 2007, pp. 1–16.
4. H. Hu et al., "WiFi Networks and Malware Epidemiology," *Proc. National Academy of Sciences*, vol. 106, no. 5, 2009, pp. 1318–1323.
5. S. Stamm, Z. Ramzan, and M. Jakobsson, "Drive-By Pharming," *Information and Communication Security*, LNCS 4861, Springer-Verlag, 2008, pp. 495–506.
6. E. Tews, R.P. Weinmann, and A. Pyshkin, "Breaking 104-bit WEP in Less than 60 Seconds," *Information Security Applications*, LNCS 4867, Springer-Verlag, 2008, pp. 188–202.
7. J. Su et al., "A Preliminary Investigation of Worm Infections in a Bluetooth Environment," *Proc. 4th ACM Workshop on Recurring Malcode* (WORM 06), ACM Press, 2006, pp. 9–16.
8. J.W. Mickens and B.D. Noble, "Modeling Epidemic Spreading in Mobile Environments," *Proc. 4th ACM Workshop on Wireless Security* (WiSe 05), 2005, ACM Press, pp. 77–86.
9. G. Yan and S. Eidenbenz, "Modeling Propagation Dynamics of Bluetooth Worms," *IEEE Trans. Mobile Computing*, vol. 8, no. 3, 2009, pp. 353–368.
10. G. Lawton, "Is It Finally Time to Worry about Mobile Malware?" *Computer*, vol. 41, no. 5, 2008, pp. 12–14.
11. A. Galante, A. Kokos, and S. Zanero, "Bluebat: Towards Practical Bluetooth Honeypots," *Proc. 2009 IEEE Int'l Conf. Communications*, IEEE Press, 2009, pp. 1–6.
12. G. Yan and S. Eidenbenz, "Bluetooth Worms: Models, Dynamics, and Defense Implications," *Proc. 22nd Ann. Computer Security Applications Conf.* (ACSAC 06), IEEE CS Press, 2006, pp. 245–256.

Stefano Zanero *is an assistant professor at the Politecnico of Milano and a partner and CTO of Secure Network, a firm specializing in information security training and consulting, based in Milan. His research interests include the development of intrusion detection systems based on unsupervised learning algorithms, security of Web applications, and computer virology. Zanero has a PhD in computer engineering from the Politecnico of Milano. He's a member of the IEEE and the IEEE Computer Society, the ACM, and serves as a member of the International Board of Directors of the Information Systems Security Association, and as a member of the board of the* Journal in Computer Virology. *In a past life, he was a columnist for* Computer World Italy.