Dartmouth Internet Security Testbed (DIST): building a campus-wide wireless testbed

Sergey Bratus · David Kotz · Keren Tan · William Taylor · Anna Shubina · Bennet Vance
Dartmouth College, Hanover, New Hampshire

Michael E. Locasto
George Mason University, Fairfax, Virginia

July 15, 2009

Abstract

We describe our experiences in deploying a campus-wide wireless security testbed. The testbed gives us the capability to monitor security-related aspects of the 802.11 MAC layer in over 200 diverse campus locations. We describe both the technical and the social challenges of designing, building, and deploying such a system, which, to the best of our knowledge, is the largest such testbed in academia (with the UCSD's Jigsaw infrastructure a close competitor). In this paper we focus on the *testbed setup*, rather than on the experimental data and results.

1 Introduction

The Dartmouth Internet Security Testbed (DIST)[1] wireless infrastructure is a collection of 802.11 MAC layer monitors, servers, and software, which operates with support from Dartmouth's Peter Kiewit Computing Services (PKCS), the College's central IT organization. DIST wireless software and operational procedures ensure that the data collection experiments are well-documented with detailed audit traces, and the gathered data remains confidential.

Unlike simulated network testbeds, DIST wireless infrastructure is distributed across diverse locations throughout the production wireless network. Thus it also allows researchers to assist PKCS in diagnosing network conditions.

Since DIST is a production infrastructure, the design and use policies of DIST wireless were affected by many social, technical, and legal considerations and concerns. In this paper, we share our experiences of designing, building, and deploying the DIST wireless infrastructure.

2 Why production networks?

Access to real-world network traffic is crucial for many security and privacy research problems.

Network Intrusion Detection. Anomaly-based intrusion detection requires establishing the concept of the network's "normal" traffic, or even validating whether such a concept can be meaningfully defined (cf. [6, 7]). Even though this concept is not central to rule-based systems, in practice their accuracy[2] can only be evaluated on realistic test data.

Traffic variability & network performance. Understaning variations of "normal" network traffic goes beyond intrusion detection. Depending on the network's topology and components, certain hard-to-detect traffic patterns may cause significant degradation in the quality of service while consuming much smaller bandwidth than the network is designed to handle. Such patterns may even be artificially crafted (e.g., [12, 8] for 802.11, TCP). Not surprisingly, describing real traffic variability has attracted practitioners' and researchers' attention [2, 13].

Research Data Sanitization & User Privacy. Whereas real data is highly desirable for researchers, its use comes at the (very high) price of ensuring that

[1] This paper results from a research program in the Institute for Security, Technology, and Society (ISTS), supported by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

[2] More precisely, the level of costly false alarms, which can make a theoretically attractive system entirely unusable in production environments.

it is purged of associations with the actual persons using the network. We must make it difficult for a malicious entity to discover their identities or to cause them harm. Previous miscalculations, such as the unfortunate example of the "AOL data set", and the increasing public consciousness of the capabilities of adversarial data mining create an atmosphere of legal uncertainty about researcher access to real data [15].

Unfortunately, sanitization of real network data is in itself a hard research problem [11, for example]. A testbed that gives researchers access to real network traffic is a powerful means of studying this problem.

3 Architecture & operation

DIST wireless, in development since January 2006, consists of 210 Wi-Fi access points in 10 large buildings across campus, of the same brand and model that Dartmouth uses to provide its production Wi-Fi network service on campus. We leveraged our existing scalable network-monitoring and intrusion-detection software base (from our MAP project [14]), in which we re-flashed the Aruba AP70 access points with OpenWRT Linux and ran our own software for sniffing on the Wi-Fi network interface. This software uses the `pcap` library to capture Wi-Fi frames and packs multiple frames into a custom format for transmission to our central server for real-time analysis and (optionally) storage. We added in-line trace sanitization, including anonymization, and never store un-sanitized data.

The DIST wireless infrastructure consists of sniffers we call *Air Monitors* (AMs) distributed across the campus and of *DIST servers*, which process frames captured by the AMs and may store sanitized frames to disk. The AMs send data to, and receive configuration files and command scripts from, the servers.

We designate one of the DIST servers as the *launchpad*. The launchpad is the only host that may launch experiments using the AMs. This policy is enforced by the AMs' firewall scripts and allows us to tightly limit the allowed communications between the DIST hosts.

We configure and run each experiment through the *MAPmaker* scripting system on the launchpad. Configuration files describe the experiment (in particular, the kind of 802.11 frames to capture, which AMs and servers to use, and which frame-processing programs to run). MAPmaker initiates processes on the DIST servers and AMs and defines communication channels among these processes.

As a part of our self-imposed discipline, we require that all uses of the DIST system are thoroughly documented and leave an audit trail. Consequently, MAPmaker logs an extensive amount of information about an experiment, including: (a) source control version stamps of MAPmaker, (b) source control stamps and checksums of executables launched by MAPmaker, (c) raw and expanded MAPmaker parameter values for each MAPmaker process, (d) copies of process configuration files, (e) `ssh` commands issued to DIST servers or AMs, (f) output (stderr and stdout) and exit codes of all `ssh` commands, and (g) output of all MAPmaker processes.

4 Harsh realities

Installing, connecting, and managing over 200 AMs in 10 large campus buildings proved to be a significant challenge in many ways. We hope that our enumeration will be helpful to others who wish to deploy such a system.

IT services permission. This step was easy, because we had developed the concept in collaboration with our Network Services group. We are fortunate to have a group of talented professionals who are also enthusiastic collaborators with researchers. We have repeatedly heard from colleagues, however, that this hurdle is very difficult in their organizations.

IRB approval. All universities with US research funding are required to operate an Institutional Review Board (IRB), so that research involving human subjects can be evaluated to ensure that risks are acceptable, and that subjects provide informed consent where possible. We obtained formal permission from the Dartmouth's IRB, the Committee for Protection of Human Subjects (CPHS).

IRB interaction can be a challenge for computer security researchers, since IRBs primarily operate in terms of medical research and often lack the conceptual structure for evaluating computer and network-specific risks [5]. Our approach in earlier CPHS applications has been to thoroughly document the terms and concerns of computer security and user privacy to provide such a structure. In the case of DIST, our prior Wi-Fi network tracing effort was approved by CPHS several years ago, and since our proposed effort was a subset of what we had done earlier, a simple renewal was sufficient.

Aesthetics. We then had to address the aesthetic concerns of the building owners (department heads, building managers, and staff with responsibilities for building upkeep) regarding the appearance of the AMs, necessary wiring, and the signs explaining the AMs' nature and purpose. These concerns included wires or wire-mold running to the AMs along walls, as well as drilling holes to place the AMs. These concerns proved to be non-trivial to address and necessitated compromises in the placement of AMs.

Our deployment spans the main library complex, the school of engineering, the school of business, a gymnasium, a student center, several dormitories, and several academic buildings.

In some cases, we chose sites where renovation was underway, and our monitors (and their wiring) could be easily installed during the construction process, requiring less cost and no inconvenience to the building residents. In all cases, we met personally with the lead staff of each department, describing what we planned to do. We walked through their buildings, sometimes repeatedly, discussing in detail the placement of AMs and their wiring.

Each building required several months of planning to obtain permission, choose sites, confirm the sites with department staff, obtain quotes from electricians, install the wiring, and install the AMs. In several cases, proposed locations had to be changed, due to availability of power (e.g., PoE-capable network ports being already occupied by other devices), office occupants' objections, or projected costs of electrical work. A major factor of success was access to detailed building floor plans, to which we added DIST-related annotations, and which we kept under source control.

Signage. As a part of our approval process, we committed to providing explanatory signs for the buildings where we deployed our AMs. The signs explained the function of the AMs and provided a URL for further information. The text, placement, and appearance of these signs turned out to be an important concern for the building owner, and required multiple personal consultations.

External audit. Due to the scale of the DIST effort and the sensitive issues related to the privacy of network users, we met with several leadership groups on campus to explain our plans, answer their questions, obtain their feedback, and ultimately seek official approval from the College to proceed with trace collection. In particular, we met with the high-level faculty committee responsible for sponsored research and the provost-level council that includes all campus deans. In both cases we obtained valuable feedback.

The College hired an outside expert, a researcher with several years of network-tracing experience in academic settings, to visit campus, interview the research teams, and to study our trace-collection infrastructure in detail. This visit served as a tremendous help to us, providing a critical eye to help us recognize where our plans could be improved or become more specific.

In response to the auditor's feedback, we added additional layers of security to ensure that the infrastructure itself cannot be compromised by attackers and to preserve confidentiality of stored data, such as hardening the DIST hosts to a very small number of network services, restrictive firewall configurations, no crypto keys in persistent storage, periodic AM restarts and firmware checks, and frequent defensive port-scans.

We also developed an extensive 20-page internal document on our design principles and procedures, which included guidelines for future changes to the infrastructure.

5 Surviving network changes

To an extensive testbed that relies on and instruments a production network environment, substantial configuration changes in this production network are something of a "net-quake" in which the underlying environment is "shifting" under it. Yet, such changes are almost certain to happen within a few years, as production networks respond to the changes in both the organization's mission and the Internet itself. These changes are even more likely to happen in campus networks than in industry networks, because college students and faculty are likely to be early adopters of new network protocols and applications, some of which are unfortunate distractions, whereas others quickly become indispensable.

A testbed overlayed on top of a production network must be designed with enough flexibility to recover from "net-quakes", both planned and unplanned. In particular, we realized (too late) the importance of maintaining a fallback control channel to the AMs during an unannounced subnet migration in one of our DIST-covered buildings.

Thus, even though we configure AMs' firmware with static IP addresses and maintain those in source-controlled tables on the launchpad server (together with other configuration information), we are also considering a fallback configuration method, based on AMs obtaining a DHCP address and reporting it to a logging port on the launchpad via a UDP message, so that a management `ssh` connection could be made to it from the launchpad.

We believe that *network recovery*, despite its practical importance for administrators and architects, has not been given proper researcher attention, and direct the reader to the discussion in [10].

6 Data protection and sanitization

We discard all but the MAC layer from each frame, then encrypt each packet of captured frames before sending them to the server; at the server they are decrypted and immediately anonymized before being used for inline analysis or storage for offline analysis. The anonymization map is generated anew for each experiment, using a random seed, which is discarded after use. Thus very little sensitive information is captured, and the most sensitive components (MAC addresses and SSIDs) are thoroughly anonymized.

The external auditor expressed concern about the DIST's initially proposed fast but home-grown MAC sanitization and suggested using some well-understood standard cryptographic solution instead.

The computation power of our AP70 Air Monitors being already stretched by other tasks, we decided on performing sanitization on DIST servers. In order to preserve the confidentiality of data captured by the AMs in transit to servers, these data must be encrypted by a cipher.

6.1 In search of a better cipher

Based on the mode of operation, ciphers are subdivided into two kinds: block ciphers and stream ciphers. A block cipher operates on data blocks, usually of fixed size, and a stream cipher operates on a continuous stream of data. We chose a stream cipher over a block cipher because of two considerations: speed and security. With regards to speed, a stream cipher generally could be much faster than a block cipher. As to security, when using the same encryption key, there is a strict one-to-one mapping between the plaintext and the ciphertext for a block cipher, whereas there is no such one-to-one mapping for a stream cipher [9]. For DIST, this property of block ciphers could be a potential security flaw, because all possible values in many fields of Radiotap header and IEEE 802.11 header can be easily enumerated, and thus a block cipher may facilitate attacks by providing a much smaller search space than a stream cipher.

We evaluated all stream ciphers from the eSTREAM project [4] and the SNOW2 cipher [3]. Our top 2 final winners are Rabbit and SNOW2. It is worth mentioning that the implementation of Rabbit cipher evaluated here is in assembly language and optimized for MIPS 4Kc processor, whereas the implementation of SNOW2 is written in C language and not specifically optimized for this processor. Since our goal was to transmit the protected data most efficiently, we tried the ciphers both without compression and in combination with compression. We observed the following:

1. For stream ciphers, Rabbit emerges as a winner on AP70s. superseding SNOW2. For executing 5000 loops, Rabbit takes 5.33–5.55 seconds, whereas SNOW2 takes 7.42–7.73 seconds.[3]

[3] The simpler Rabbit cipher is implemented in assembly language and optimized for MIPS 4Kc processor, whereas the implementation of SNOW2 was in C.

2. A more interesting observation is that, if encryption and UDP forwarding are included, adding in an efficient compression not only greatly saves network bandwidth, but also decreases the net processing time.

Securely transmitting 5000 14K jumbo frames (each jumbo frame may contain tens to hundreds of captured Radiotap and IEEE 802.11 headers) to a DIST server took 6.2–6.4 seconds, which encompassed two operations: encryption and UDP forwarding. The Ethernet bandwidth used is 14K bytes per frame. However, if we compress these jumbo frames first, handling them takes *less* time, namely 5.3–5.4 seconds for *three* operations: compression + encryption + UDP forwarding. The required Ethernet bandwidth is also reduced by nearly 80% (from 14K per frame to just 3K per frame). This observation illustrates that an efficient compression can not only save the bandwidth, but also saves a lot of time for the operations of encryption and UDP forwarding.

6.2 MAC address and SSID sanitization

Depending on the particular clients' setup, MAC addresses of 802.11 wireless client stations may enable personal identification of the stations' owners. Although MAC addresses by themselves do not constitute personally identifiable information, they can be correlated by third parties with other data, such as DHCP dynamic DNS updates or *Netreg* logs. Although the degree of efficiency with which such data can be exploited to reveal the wireless station owners' identities is still being investigated, we take the position that logging 802.11 wireless clients' MAC addresses *may* be a privacy concern.

To address this concern, we mandate that the 802.11 client MAC addresses *must* be sanitized during capture in DIST experiments. We conduct MAC anonymization on the servers, as explained above, and protect the MAC addresses in transit from AMs with a separate encryption step. Processes running on DIST servers are provided with a simple frame-sanitization interface, which consists of a small set of functions to sanitize MAC addresses of 802.11 wireless stations.

A randomly-generated sanitization key, akin to an encryption key, seeds a stream of pseudo-random numbers for the algorithm. This key is not reported to users, is not written to stable storage, and only resides in a single process for the minimal time necessary. The stream of pseudo-random numbers is used to construct a set of in-memory tables, which are consulted as part of the sanitization task. The tables are never written to stable storage.

Each set of tables is identified by a unique ID, by which they can be located in RAM. This design makes it possible for several concurrently running experiments to use different and unrelated sanitization tables. In particular, no MAC-related data collected by one experiment can then be meaningfully correlated or de-anonymized with the data collected by another. As a matter of policy, MAC anonymization tables are derived from a per-experiment key; the key is never stored on disk, and is destroyed after the tables are generated; the tables are discarded after the data-collection experiment is complete; the tables are never written to disk.

A well-understoodchannel of potential private information disclosure is the 802.11 *Probe Request* frame information element that contains the name of the desired *ESSID*. Some 802.11 clients still attempt to probe and then associate with the last known network by ESSID, although this behavior has long been revealed as a privacy risk. The probed ESSID often contains private information, such as the home wireless network owner's name. Transmitted in the clear, the ESSID may reveal the identity of the client station's owner. The DIST server anonymization process includes a special case for the probed ESSIDs. If an ESSID is not a member of Dartmouth's *whitelist*, the corresponding information element is anonymized by hashing its ASCII string value.

7 Related Work

Several projects at other academic institutions targeted large-scale wireless network monitoring. Whereas DIST remains, to the best of our knowledge, the largest such infrastructure, we direct the reader to the experiences of Yeo et al. [16], the UCSD's Jigsaw [1] and EPFL's DOMINO [12].

8 Conclusion

Pervasive monitoring of a network infrastructure presents a number of challenges that illustrate the stark differences between clean, simple academic research models and the real-life compromises needed to set up and maintain such a monitoring infrastructure. Building DIST as a testbed that monitors and relies on Dartmouth's main production wireless infrastructure presented us with many foreseen and unforeseen social and technical challenges. We hope that our experience will motivate others to build similar testbeds and help them do so more efficiently.

References

[1] Y.-C. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage. Jigsaw: solving the puzzle of enterprise 802.11 analysis. *SIGCOMM Comput. Commun. Rev.*, 36(4):39–50, 2006. DOI: 10.1145/1151659.1159920.

[2] A. B. Downey. Evidence for long-tailed distributions in the internet. In *IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 229–241. ACM, 2001. DOI: 10.1145/505202.505230.

[3] P. Ekdahl and T. Johansson. A new version of the stream cipher snow. In *SAC '02: Revised Papers from the 9th Annual International Workshop on Selected Areas in Cryptography*, pages 47–61, London, UK, 2003. Springer-Verlag.

[4] ECRYPT Stream Cipher Project (eSTREAM). Online at http://www.ecrypt.eu.org/stream/.

[5] S. L. Garfinkel. IRBs and security research: myths, facts and mission creep. In *UPSEC'08: Proceedings of the 1st Conference on Usability, Psychology, and Security*, pages 1–5. USENIX Association, 2008.

[6] C. Gates and C. Taylor. Challenging the anomaly detection paradigm: a provocative discussion. In *NSPW '06: Proceedings of the 2006 workshop on New security paradigms*, pages 21–29. ACM, 2007. DOI: 10.1145/1278940.1278945.

[7] C. Gates, C. Taylor, and M. Bishop. Dependable security: testing network intrusion detection systems. In *HotDep'07: Proceedings of the 3rd workshop on on Hot Topics in System Dependability*, page 12. USENIX Association, 2007.

[8] M. Guirguis, A. Bestavros, and I. Matta. Exploiting the Transients of Adaptation for RoQ Attacks on Internet Resources. In *ICNP '04: Proceedings of the 12th IEEE International Conference on Network Protocols*, pages 184–195, Washington, DC, USA, 2004. IEEE Computer Society.

[9] J. Katz and Y. Lindell. *Introduction to Modern Cryptography.* Chapman & Hall/CRC, 2007.

[10] M. E. Locasto and A. Stavrou. The Hidden Difficulties of Watching and Rebuilding Networks. *IEEE Security and Privacy*, 6(2):79–82, 2008. DOI: 10.1109/MSP.2008.48.

[11] P. Porras and V. Shmatikov. Large-scale collection and sanitization of network security data: risks and challenges. In *NSPW '06: Proceedings of the 2006 workshop on New security paradigms*, pages 57–64. ACM, 2007. DOI: 10.1145/1278940.1278949.

[12] M. Raya, J.-P. Hubaux, and I. Aad. DOMINO: a system to detect greedy behavior in IEEE 802.11 hotspots. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 84–97. ACM, 2004. DOI: 10.1145/990064.990077.

[13] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang. Experience in measuring backbone traffic variability: models, metrics, measurements and meaning. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 91–92. ACM, 2002. DOI: 10.1145/637201.637213.

[14] Y. Sheng, G. Chen, H. Yin, K. Tan, U. Deshpande, B. Vance, D. Kotz, A. Campbell, C. Mcdonald, T. Henderson, and J. Wright. MAP: A scalable monitoring system for dependable 802.11 wireless networks. *IEEE Wireless Communications*, 15(5):10–18, October 2008. DOI: 10.1109/MWC.2008.4653127.

[15] D. C. Sicker, P. Ohm, and D. Grunwald. Legal issues surrounding monitoring during network research. In *Proceedings of the Internet Measurement Conference (IMC)*, pages 141–148. ACM, 2007. DOI: 10.1145/1298306.1298307.

[16] J. Yeo, M. Youssef, and A. Agrawala. A framework for wireless LAN monitoring and its applications. In *WiSe '04: Proceedings of the 3rd ACM workshop on Wireless security*, pages 70–79. ACM, 2004. DOI: 10.1145/1023646.1023660.