MultiNet Usable and Secure WiFi Device Association

Anthony Brown Richard Mortier Tom Rodden
School of Computer Science, University of Nottingham, UK
firstname.lastname@nottingham.ac.uk

ABSTRACT

This demo presents *MultiNet*, a novel method for joining devices to a domestic Wi-Fi network. MultiNet dynamically reconfigures the network to accept each device, rather than configuring each device to fit the network as is the norm. It does so by assuming that each device is pre-configured with a cryptographically generated WPA2 network SSID/passphrase pair, and then providing a lightweight interaction through which the user creates a new network for each device. This approach makes securely adding devices to a wireless network straightforward without compromising security or burdening the user, and maintaining backward compatibility with existing deployed standards and protocols.

The demo deploys a MultiNet Access Point (AP) and a number of Wi-Fi enabled consumer devices to allow viewers to dynamically construct and deconstruct the network via the MultiNet controller currently implemented as an app on an Android phone (Figure 1). The code for MultiNet is publicly available under open-source licenses.[1]

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

Keywords

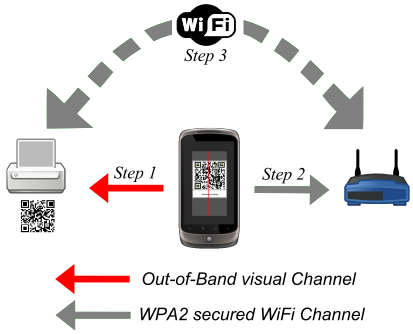
Usable security, domestic environments, 802.11

1. MULTINET OVERVIEW

Usability of domestic network security mechanisms remains a challenge, with the HCI community suggesting *interface veneers* alone cannot solve all the apparent issues [1]. MultiNet is an infrastructure intervention which changes the way in which devices are joined to the network, *configuring of the infrastructure to the device* rather than configuring each device to fit the infrastructure as is the norm. This approach has allowed us to redesign the joining interaction to be consistent across all devices while retaining backwards compatibility with legacy equipment. In our first prototype the interaction relies on a visual out-of-band channel created via the use of an intermediary device possessing a camera[2]. We assume devices are pre-configured, and we then modified *hostapd* to dynamically create and destroy WPA2 secured networks on demand. The result is that the router offers multiple networks, approximately one per device.

Credential exchange takes place via an intuitive "capture image and connect" interaction. The MultiNet controller connects to the router over a dedicated WPA2 secured *control network*, which forms a secure signalling channel over which the credentials can be passed. Devices are assumed to be preconfigured by the manufacture with an SSID and

Figure 1: Joining a printer using MultiNet.



Step One	Use the camera on the MultiNet controller to read the QR Code on the joining device.
Step Two	Request that the joining device is added to the network by sending the SSID/passkey pair for the virtual network to the AP.
Step Three	Establish the virtual network allowing the pre-configured joining device to connect.

Figure 2: MultiNet dataflow for adding a device.

[1]Available at `https://github.com/Toshbrown/`

DE2012, October 23-25, 2012, Aberdeen, UK.
ACM X-XXXXX-XX-X/XX/XX.

[2]intuitively named the "MultiNet Controller"

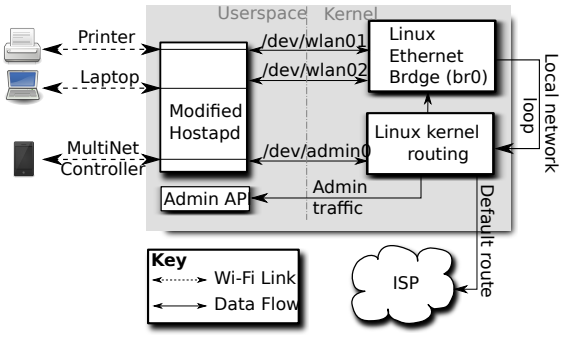


Figure 3: Overview of MultiNet implementation.

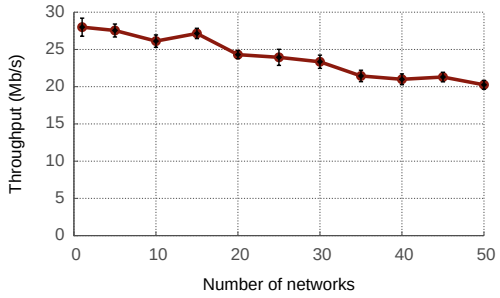


Figure 4: Device to AP throughput (Mb/s) as the number of configured networks on the AP increases.

a passphrase; in our prototype these are encoded visually as a QR Code, which the MultiNet controller acquires through its built-in camera. These credentials are then securely conveyed to the router. On receipt the router creates a virtual AP with the required SSID/passphrase pair and the joining device joins this newly created network. The overall protocol flow is depicted in Figure 2. This approach effectively configures the AP to accept the device and creates a "one network per device" infrastructure.

The AP uses the *hostapd* user-space daemon to provide wireless access point with authentication via IEEE 802.11i WPA2. We modified *hostapd* to enable dynamic creation and destruction of WPA2 secured networks, and to change the configuration re-load code to only de-authenticate stations when the SSID or passphrase of a network is changed or removed from the configuration file. After creation, these virtual network interfaces are connected to the standard Linux layer 2 bridge (*br0*), enabling communication between them. An overview of the design is depicted in Figure 3. The AP is controlled through a set of RESTful web services allowing the MultiNet controller to add and remove networks by configuring *hostapd*. These services are only accessible over HTTPS using the trusted controller network. The admin network is completely isolated from the device specific networks, protecting the admin functions from all non-trusted devices connected to the network. In our prototype the MultiNet controller is implemented as a simple Android application; other implementations are possible.

2. PROTOTYPE EVALUATION

Performance. We configured the MultiNet AP to offer the required number of networks and to act as an Iperf traffic sink. Once configured a device was added to act as a Iperf traffic source and measurements of throughput, latency and jitter were made. The AP to device throughput shows approximately linear reduction as the number of networks and associated overheads increases, as expected. At 20 networks there is a 13% reduction in maximum throughput, and by 50 networks this figure has risen to 27%, this is shown in Figure 4. Results for per-packet latency show a generally linear upward trend from from 8 ms to 15 ms for up to 20 networks. Jitter also rises slowly from 3 ms to 5 ms in the 1 to 20 network region. While MultiNet has some impact on throughput, latency and jitter it is acceptably small for up to 20 devices (a recent survey [2] put the UK household average at 4.3).

Usability. We performed an initial usability evaluation comparing MultiNet to Wi-Fi Protected Setup (WPS) with 16 participants. The trial consisted of one task building a small home network with three consumer devices under two conditions: (*C1*) using WPS, (*C2*) using MultiNet. All participants experienced both conditions in a lab environment. Several metrics were used to asses the overall usability of both systems. The effectiveness and efficiency were measured using task completion time, while user satisfaction was measured using SUS scales and post trial interviews. In all cases MultiNet showed a statistically significant improvement over WPS.

3. SECURITY CONSIDERATIONS

Recently discovered brute force attacks on the In-band WPS configuration highlight the issues of trading usability for security [3] and the danger of hiding security related features from the user: MultiNet exposes security related features making them more easily perceivable. Use of a visual out-of-band channel reduces the opportunity for undetected attacks, as reading the credentials from the QR codes requires reasonable physical proximity and line of sight to acquire a decodable image. By removing the choice of SSID and passphrase from the users we are also able to use securely generated values which are much longer and use a larger alphabet than could be entered by a human. Finally, MultiNet effectively maps network security to the physical security of devices and MultiNet controller which should help users better manage the risks associated with their actions as they manage the physical security of objects everyday.

4. ACKNOWLEDGEMENTS

This work was supported by the Doctoral Training Centre in Ubiquitous Computing, RCUK grant EP/G037574/1 and by Horizon Digital Economy Research, RCUK grant EP/G065802/1.

5. REFERENCES

[1] K. Edwards, R. Grinter, R. Mahajan, and D. Wetherall. Advancing the state of home networking. *Communications of the ACM*, 54(6):62–71, June 2010.

[2] TP-Link. 2011 connectivity report. `http://thenextweb.com/uk/2011/12/09/uk-households-have-an-average-of-4-6-devices-connected-to-wifi`, 2011.

[3] S. Viehböck. Us-cert vulnerability note vu#723755 - wifi protected setup (wps) pin brute force vulnerability, 2011.