Peer to Peer Secure Update for Heterogeneous Edge Devices

Emily Band, Herry Herry,* Colin Perkins, and Jeremy Singer
School of Computing Science, University of Glasgow

November 2017

Deployment of smart campuses and smart cities use low-cost, low-power edge compute devices to build sensor and control systems and provide smart edge compute nodes. The systems often start, with a few tens of nodes, but can rapidly scale to many thousands of devices. These devices can be in inaccessible locations, be mobile, or be in private residential locations, so remote administration is essential to ensure that security updates are deployed and to install new applications.

The conventional approach for managing compute resources uses tools such as Chef or Puppet. These apply updates using either direct `ssh` access to the hosts being managed, or by pulling them from a well-connected central server. Accordingly, they work well for managing large clusters in data centre environments with reliable network infrastructure, but are not suitable for more distributed, mobile/ad-hoc, and less well-connected environments.

To handle such situations, we designed and built a prototype decentralised management framework that distributes system updates and management scripts via a peer to peer overlay network. This allows us to manage devices that do not have direct network access, for example nodes behind network address translators (NATs) or firewalls.

Our system combines several techniques: 1) STUN-based UDP hole punching to discover and open NAT bindings; 2) an overlay protocol to deliver short messages, similar in scope to HashiCorp Serf, to distribute update notifications; and 3) BitTorrent to securely distribute the software updates. The key novelty is that our system assumes partial network connectivity, and works in the presence of NATs are firewalls.

We securely send the metadata needed to trigger BitTorrent downloads to target devices that are indirectly connected to the administrator, ensuring they are not vulnerable to man in the middle attacks. We also enriched this metadata with: 1) a resource-unique identifier, which distinguishes updates for two different resources, and 2) a version, which avoids the device using an outdated update which might still exist in the network. Last but not least, we employ UDP hole punching technique to enable devices that run behind different NATs, to directly communicate with each other for distributing the torrent-file.

Future work will integrate this peer-to-peer secure update framework into FRµIT testbed[1] to enhance scalability and allow us to manage hosts in challenging network environments.

*Corresponding author; e-mail: herry.herry@glasgow.ac.uk

[1] `http://fruit-testbed.org`