Design and Deployment of Security Sensitive, Networked Embedded Systems

Johan Dams
WRD Systems
Email: johan.dams@wrdsystems.co.uk
http://www.wrdsystems.co.uk

***Abstract*—With contemporary developments pushing towards an ever connected world, spearheaded by the push for the 'Internet of Things' coupled to all things 'cloud', security still remains an afterthought. Cost is often cited as the main reason, followed by the ever shorter deadlines and competition to bring devices to market as fast as possible. In this paper we will review some of the recent issues with a wide variety of connected devices and show how lacking security decisions ended up costing the end users as well as the implementers. Additionally, we present a possible solution to one of the core issues: trust in the communications network, or the lack thereof. In addition, we will report a project with a currently deployed, networked device which adheres to accepted security requirements while still being cost effective.**

I. Introduction

This paper consists of three parts. The first part reviews some of our the past research, observations, and conclusions. It shows the current trend with security issues specifically within embedded systems, and recent developments therein, and some managerial aspects we need to overcome to improve the situation.

The second part introduces an adaptation of the technology behind Bitcoin and others to provide a secure communication system for embedded devices. This is still very much work in progress, but we have been able to create a working prototype that solves the Byzantine Generals problem.

The final section reviews some of the recent work done by WRD Systems in deploying a large amount of GPS tracking devices, with focus on security. We revisit some of the development process and observations, analysis of existing devices, and report on the specific security focus points.

II. Review and Current State

In the past, we have demonstrated [1] that the security of many networked devices left much to be desired. The focus of that previous research was within the energy sector, and specifically in all aspects smart grid such as metering, SCADA, and security issues in substation automation. Since the publication of that article, several attacks against smart meters have been shown, such as the one demonstrated by Javier Vazquez Vidal and Alberto Garcia Illera [2] at Black Hat Europe in 2014.

Outside the energy sector, we have recently seen remote attacks against vehicles. A recent paper [3] documents the remote exploitation of an unaltered passenger vehicle which also received a lot of publicity outside of the security community.

What is usually not publicised are the security issues with thousands of devices that are being connected to the internet without any kind of though with regard to security. We think of the 'Internet of Things', or IoT, devices that are being developed at a rapid pace to ensure time to market is as short as possible and lowest cost because of the competition. It is not just the devices themselves however, it is the infrastructure around it as well. These issues are highlighted in a white paper [4] by Symantec.

A. The Cost Issue

One of the common issues with security in the IoT is that of cost. IoT is essentially a subset of Machine to Machine, or M2M. M2M is nothing new, but has until recently been the domain of industrial applications. With cheaper and pervasive networks and the rise of the 'app', or mobile phone application, we see more and more devices being used to communicate ever more personal, medical, environmental, financial and more data from one device to another. This data, and the applications and analysis of this data, is very valuable.

Because of the competition in the market, vendors have to get their devices and apps out as soon as possible, because the competition would otherwise form a threat. Time to market is essential. This means that aspects such as security keep being considered as a cost despite that facts show they lead to huge expenses later on in the case of lost data and security breaches. As long as shortest time to market and short term profits rule, things are not going to get better.

B. The Lack of Knowledge Issue

Most embedded developers are not security experts. This means that in order to develop proper security, a security expert is required. Since this leads back to the cost issue, managers tend not to see the need for a dedicated security team. This leads to security being implemented by people

who are not competent to do so. Several examples, such as the AnonaBox [5] incident, have shown this issue in the recent past.

There needs to be a change in security attitude whereby customer security gets taken more serious. This requires dedicated security people to design and review secure architectures for both devices and the infrastructure around them.

III. Solving Trust Within the Internet of Things

One of the most essential problems when providing security is the issue of trust. One of the major hurdles when trying to communicate over an untrustworthy link such as the internet is illustrated with the so called Byzantine Generals problem. The category of Byzantine failures is that in which a system fails in an arbitrary way. This means they do not just stop working, but could produce the wrong outputs, have inconsistent outputs, etc. A Byzantine fault tolerant system will be able to keep working correctly as long as not too many Byzantine faulty components are present.

One interesting recent example of a Byzantine fault tolerance solution in use is Bitcoin [6], a peer-to-peer digital currency system. The Bitcoin network works in parallel to generate a chain of hashcash style proof-of-work. The proof-of-work chain is the key to overcome Byzantine failures and to reach a coherent global view of the system state [7].

In a similar way, Bitmessage [8] is a protocol for the secure exchange of electronic messages. Bitmessage is intended to provide a system that is fully decentralized, encrypts all messages, masks the sender and receiver of messages, and guarantees that the sender of a message cannot be spoofed. The design of Bitmessage is heavily influenced by the Bitcoin protocol.

Bitmask [9] was proposed to address some of the shortcomings of Bitmessage. The work was presented in a white paper, but no implementation was made. One of the most important additions was the guaranteed forward secrecy.

Based on the above presented work, we implemented a system that allows embedded devices to securely send and receive messages, and listen to broadcast messages, using a decentralized, trust-less, peer-to-peer protocol over IP. Devices only have to exchange an address to ensure secure communication. This address provides a unique identity, and prevents spoofing of message sender. The protocol is designed to hide data such as the sender and receiver of messages from those not involved in the communication in order to prevent eavesdropping.

A. Implementation

The entire proposal depends on the feasibility of implementing the proof of work (POW) scheme such as implemented today in the Bitcoin and Bitmessage network on embedded devices while having acceptable storage, bandwidth and battery considerations. Older types of proof of work systems such as Hal Finney's reusable proof of work relied on trusted secure hardware (like a TPM chip) instead of trusting a data structure created by a peer to peer network. While this might seem like a good solution for embedded devices at first, if the chip is compromised by an attacker, the entire network with embedded devices becomes instantly insecure. Instead, a typical proof of work creation is used such as implemented in Bitmessage and illustrated in Fig. 1. Similarly, a proof of work verification can be implemented as illustrated in Fig. 2 below.

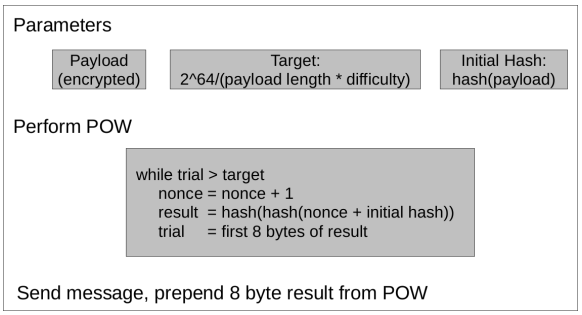

Fig. 1. Proof of Work Generation.

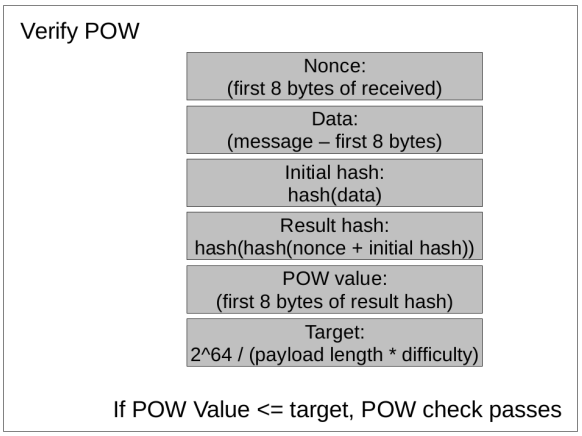

Fig. 2. Proof of Work Verification.

The main obstacle to provide an implementation of such a scheme on low power embedded systems is the need for a significant amount of hashing operations during the proof of work generation phase. To make this feasible, we borrow the idea of using a dedicated security chip, with its sole responsibility to generate these hashes, and do so at very low power requirements. While this ASIC could potentially be expensive to design and build, we reused existing ASIC chip designs and implementations as used in Bitcoin mining. This means that even older generation Bitcoin mining ASIC designs can be used that have been extensively tested, but are not useful to mining Bitcoin any more since the Bitcoin

difficulty has risen significantly since. Our proof of work implementation is thus be based on the same hash algorithm as Bitcoin, namely SHA256.

The 'difficulty' parameter in the POW generation and verification is a parameter that can be changed dynamically based on the amount of nodes in the network, or requirements on speed for message transactions. This parameter can be set on a per node basis. Due to the peer to peer nature of the network, and the way the protocol works, each node on the network receives each message sent within the network. This could potentially lead to excessive storage needs for embedded devices. To resolve this, we implemented a similar scheme as used by Bitmessage and Bitmask whereby the network will self-segregate into clusters. Each cluster is identified by a number which is encoded into each device address. When a message needs to be sent, the device first connects to the cluster as encoded in the destination address. The entire parent-child cluster network can be traversed if no peers are known, until it arrives at the destination cluster.

The communicating devices exchange a hash of a public key that also serves as the device's address. The public key can be retrieved by the underlying protocol, and so it can easily be hashed to verify that it belongs to the intended recipient. A message claiming to be from a specific address can simply be checked by decoding a special field in the data packet with the public key that represents the address. If the decryption succeeds, the message is from the address it claims to be.

Based on the type of network and available hardware, a wide range of public key cryptographic methods can be employed. The current implementation makes use of elliptic curve based systems, but these can be implemented in a modular way so other systems can be swapped in. There are several ECC implementations, such as the one presented during our previous research in [10], that have been specifically optimised for use in resource constrained embedded systems without additional hardware.

IV. Deployment of Secure GPS Tracking Devices

In the past we have reviewed the security of several GPS tracking devices in operation today. While some of the more expensive trackers provided encrypted data, the majority of them did not. In 2014 WRD Systems was contracted by a large City Council in the U.K. to provide GPS tracking devices for a long term bike rental scheme. As part of our study, several bike trackers were analysed for both functionality and security. From this analysis we found that:

- None of the bike specific trackers encrypted data sent to the server.
- None of the bike specific trackers used authentication for incoming commands.
- None of the server applications receiving the data did any authentication of incoming data.
- None of the mobile applications interacting with the data used encryption.
- All the tracking devices could be made to send data to other destinations by sending SMS commands to the tracker.

This situation shows that, while the tracking devices and software are available, the lack of security means they should not be used for anything serious. The fact that a GPS tracker sends sensitive data on whereabouts and behaviour should indicate proper security has to be implemented.

We believe that this issue is inherently linked to the lack of attention to detail and security know-how when it comes to engineering such devices. As part of the contract with the City Council, we developed and deployed a large scale GPS tracking project well within budget and time frame which consisted of a highly secure, customised GPS tracking device and surrounding infrastructure. Some of the security issues we encountered with the devices and applications we investigated were addressed as follows:

- All data sent by the tracker is encrypted.
- Every interaction with the device or server is authenticated as well as encrypted, with a minimum key length of 256 bit or equivalent.
- No incoming commands to the tracker. The firmware was locked down specifically for the requirements of the project.
- Firewall at server side drops all traffic outside the UK.
- All data at rest on the server is encrypted.

We believe that a lot of the issues with other trackers and their platforms is the increasing demand for features which are integrated for every project at the entire stack. Instead, we opted to only enable (physically compile in or deploy) features required on a per project basis. This way, it becomes much easier to review code as many features which could cause issues are just not present. We follow this methodology throughout the software stack.

In addition, we did not rely on third party 'cloud' providers due to the sensitive nature of the data in question. We deployed our own physical servers in a government approved data center where we physically separate each project on different servers. We believe that doing this is imperative to properly guarantee the data is secure at all stages.

While we initially thought we would come out more expensive than the competition by doing the above, this did not turn out to be the case. Since we do everything in-house from hardware to software, we did not have to calculate third party profit margins (hardware development, software, cloud, etc.) in account. We also noticed that even by producing the hardware itself in Europe (something WRD Systems does out

of principle), the project still came in cheaper than any offer from other third parties.

V. Conclusion

We believe that security should be a priority for network connected devices, especially those that deal with sensitive data related to health, personal details, and those that can be used to identify behavioural characteristics of the user. Revisiting our previous work has shown that a lot of the issues we reported on in recent years still exist today.

To help improve the situation, we have shown an approach to send messages securely across an untrusted network with the the inherent nature of the design that solves the Byzantine Generals problem and the key management issue within the protocol. While a prototype running on embedded hardware is functional, more work is needed to turn this into a readily available solution, and funding will have to be sourced to continue this work.

In addition, we have reported on our work of deploying a large number of GPS trackers with a focus on data security. The analysis of existing platforms in this field has shown that many leave much to be desired when it comes to security, especially in the low cost bracket.

References

[1] Johan Dams, *Securing where smart grid meets SCADA*, embedded.com, 2013, http://www.embedded.com/design/real-world-applications/4413576/Securing-the-smart-grid-and-SCADA, Last accessed 01/09/2015

[2] Javier Vazquez Vidal and Alberto Garcia Illera, *Lights Off! Te Darkness of the Smart Grid*, Black Hat Europe 2014, http://www.darkreading.com/perimeter/smart-meter-hack-shuts-off-the-lights/d/d-id/1316242, Last accessed 01/09/2015

[3] Charlie Miller and Chris Valasek *Remote Exploitation of an Unaltered Passenger Vehicle*, Black Hat USA 2015, http://illmatics.com/Remote Car Hacking.pdf, Last accessed 01/09/2015

[4] Mario Ballano Barcena and Candid Wueest, *Insecurity in the Internet of Things*, Symantec White Paper, 2015, https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/insecurity-in-the-internet-of-things.pdf, Last accessed 01/09/2015

[5] Andy Greenberg, *Anonabox Recalls 350 Privacy Routers for Security Flaws*, Wired, 2015, http://www.wired.com/2015/04/anonabox-recall/, Last accessed 01/09/2015

[6] Satoshi Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008, http://bitcoin.org/bitcoin.pdf, Last accessed 01/09/2015

[7] bitcointalk.org *The Byzantine Generals' Problem*, https://bitcointalk.org/oldSiteFiles/byzantine.html, Last accessed 01/09/2015

[8] Jonathan Warren, *Bitmessage: A PeertoPeer Message Authentication and Delivery System*, 2012, https://bitmessage.org/bitmessage.pdf, Last accessed 01/09/2015

[9] Tommy Anderson, *Bitmask: A Perfectly Anonymous, Naturally Scalable Peer-to-Peer Messaging System*, 2013, https://bitmessage.org/forum/index.php?action=dlattach;topic=3259.0;attach=124, Last accessed 01/09/2015

[10] Johan Dams, *Portable Elliptic Curve Cryptography For Medium-Sized Embedded Systems*, University of Vaasa, Faculty of Technology Department of Computer Science, Vaasa, 2008.