

# Router and IoT Vulnerabilities:

Insecure by Design



manySECURED

# Introduction

Audience: IoT Device and Router Manufacturers, Internet and Communication Service Providers, Internet Browser Vendors, IoT Solution Vendors, Certificate Authorities, IoT End Users, Security Professionals, Policy and Standards Experts.

There is much talk of security issues arising in Internet of Things (IoT) systems [ref 1], and for good reason. IoT devices provide a whole array of new attacks that are, generally, poorly understood and so, of course, weakly protected against. Additionally, the cost pressures of producing low-cost devices combined with the complexity of creating true end-to-end security, running from hardware – to software – to cloud service, mean IoT devices are one of industry's biggest security concerns. These concerns are being recognised across the world and action taken e.g.: in the UK, the government are moving towards the introduction of new cyber security laws to protect smart devices [ref 2]; the standards organisation ETSI [ref 3] have released a European Standard, ETSI EN 303 645, Cyber Security for Consumer Internet of Things: Baseline Requirements [ref 4]; in the USA the National Institute of Standards and Technology (NIST) National Cybersecurity Center of Excellence (NCCoE) have set up the Trusted IoT Device Network-Layer Onboarding and Lifecycle Management Community of Interest [ref 5] and in Asia, the Cyber Security Agency of Singapore (CSA) have launched the Cybersecurity Labelling Scheme (CLS) for IoT Devices [ref 6].

This Whitepaper from the Internet of Things Security Foundation (IoTSF) [ref 7] ManySecured Special Interest Group (SIG) [ref 8] aims to: raise awareness to a fundamental design flaw that has received little attention to date; elicit feedback; and recruit organisations and individuals interested in being part of the solution. This design flaw affects many IoT devices and standard Internet routers. The design flaw is that the management interface typically provided for you to configure and manage your device, is insecure. Very specifically, what we are saying is:

**When you are directed to manage an IoT device or router using a browser, your password and all communications (everything you do) are typically passed over an unencrypted connection.**

This design flaw is a very serious problem; it is pervasive, affecting most domestic installations, and it represents a huge security exposure, leaking both passwords and activity to anyone who is listening. **This problem cannot be mitigated by implementing cybersecurity best practice; it is due to a fundamental design flaw.**



**many**SECURED

# Understanding the problem/design flaw in detail

Before we look at what we can do about it, let's try to understand the precise nature of the problem. Firstly, how big a problem is it, how many devices does it impact?

## A pervasive vulnerability

The harsh reality is the problem we are talking about impacts most domestic installations and a surprisingly high number of industrial ones.

Looking at router problems alone, the support pages for the routers provided by major European ISPs typically say something like the following two examples:

- Access the Hub Manager to manage your hub settings, change the hub's name or change passwords. Type 192.168.1.254 into a browser to view the Hub Manager.
- In the address bar enter 192.168.0.1/ and press return [Enter].

## What exactly are we seeing here?

What you are seeing is a set of fairly standard instructions for configuring your device, using a browser, which defaults to http and not https. You will find similar instructions for most routers and most IP enabled IoT devices. These instructions help you as a user solve a very practical problem: how do I discover and connect to a new device, possibly one I have never seen before? Your standard internet browser is a great way to help you solve this problem. The internet browser is a general-purpose application, that every user is very familiar with. As a manufacturer, all I need to do is help the user find the device (enter the IP address), and I have provided a simple, easy to use method for the user to configure and provision the end device.

Sounds simple doesn't it? It is, it's very simple and very easy to use, but for one glaring problem, it's not secure:

**No certificate  
=  
No encryption**

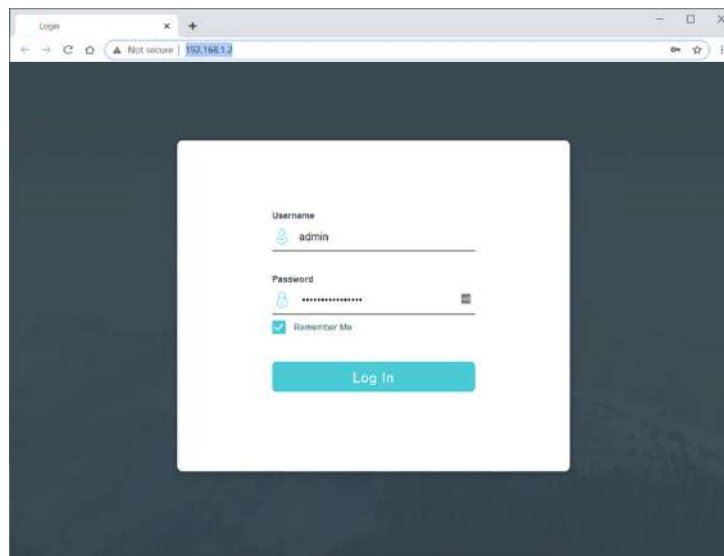


So what exactly is the problem? Look carefully at the image below.

When you browse to a local router or internet device, look in the top left corner. The internet community has been training users for ages to look for the padlock. It is now well understood that good internet hygiene requires that you check that your internet connection is secure. That's what the padlock tells you; the website you are navigating to is backed by a certificate and your connection is secure.

The problem is that most routers and IoT devices do not provide a certificate for their local device.

As you can see from this typical IoT/router login screen, you are being encouraged to enter your username and password on a website, when the browser is telling "Not secure" where there should be a comforting padlock.



## Insecure by Design

So why are these manufacturers being so careless with our security? It turns out it is not entirely their fault. Browser based security was not designed for connecting the local internet (by local internet we mean devices sitting on our local network, typically on the 192.\* or 10.\* subnet ranges).

The CA/Browser Forum [ref 9] is the industry group that, among other things, sets the security standards for web browsers e.g.:

The CA/Browser Forum in their baseline requirements 7.1.4.2.1 [ref 10]

CAs SHALL NOT issue certificates with a subjectAltName extension or subject:commonName field containing a Reserved IP Address or Internal Name.

What this is saying, very specifically, is that CAs (Certification Authorities – the organisations that issue certificates) are not allowed to issue a certificate for an internal name (local internet).

So it turns out that this major security weakness, which is impacting almost every consumer installation, is the result of an industry internet specifications: it's insecure by design.

Now there are some very good reasons why the specifications above state what they do. If we want the public internet to be secure, we need to set some very clear guidelines around the semantics and interpretation of issued certificates. But like a lot of things in the security space, the law of "unintended consequence" is there to bite us if we don't pay enough attention.

The problem is this: when we use a web browser to browse to a local device, we are using the technology in a way it was not intended.

## Why is it such an issue?

Is this another example of security professionals, exaggerating the threats of some esoteric attack, to increase their sense of self-importance?

We think not. This is a pretty serious issue. It's widespread; it affects many installations. And it's wide reaching; if you leak router passwords (or IoT passwords), you pretty much expose the entire network.

Very specifically, current IoT best practice, such as not using default passwords, is rendered almost useless if every time you enter the password you are doing so on an unencrypted connection.



## Zero Trust Architectures – How complex is the attack?


Technically how difficult is it to steal a password in this way?

If the router is securely configured and contains no vulnerabilities, the attacks listed below can only be initiated from within the internal network. However, if the router is poorly configured, or the router suffers from one or more systemic vulnerabilities, some of these attacks can be issued from the public internet, which increases the attack surface considerably.

The main attack methods are:	The principal attack vectors are:
<ul style="list-style-type: none"><li>• Through access to the internal network, e.g. the attacker has the WLAN credentials. This attack normally cannot be executed from outside the internal network if the network is well configured. However, the recently reported [ref 11], [ref 12] traffic fragmentation attacks are examples of how this can be done from outside the network, even when the network is well configured.</li><li>• Through UPnP and port forwarding, gateway devices could allow traffic originated from the internet.</li><li>• Through JavaScript attacks, the user may be served content that can attack their own infrastructure.</li><li>• The attacker could use a Wi-Fi module that can execute in either monitor mode [ref 13] or promiscuous mode [ref 14]</li></ul>	<ul style="list-style-type: none"><li>• a guest on your network to whom you have granted Wi-Fi credentials;</li><li>• a visitor in your home or organisation with access to the credentials printed on the hub, or who presses the WPS button;</li><li>• an external attacker who cracks your Wi-Fi [ref 15];</li><li>• a publisher of an "application" (PC or mobile), which is running on a device that supports the necessary modes, which has implicit access to the network;</li><li>• a manufacturer of a device to which you have granted WLAN access in the setup phase.</li></ul>



Technically, it is a moderately complex attack, but nothing beyond the abilities of an appropriately incentivised criminal/disgruntled neighbour/activist, or for that matter, a mildly bored, technically competent teenager.



The most disturbing aspect of this vulnerability, however, is just how widespread it is. Based on surveys undertaken by this working group (SIG), most router and IoT device vendors offer a locally hosted web server to manage the device. And in most instances the device user manual explicitly directs the users to an HTTP (unencrypted) local address to enter their password (e.g. <http://191.168.1.254>).

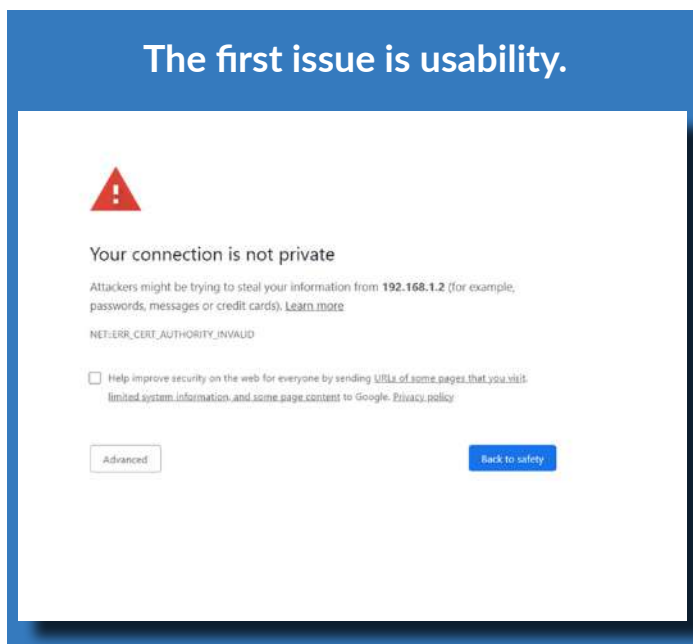
Zero Trust architectures, based on the principle that you should not automatically trust anything inside your network, are becoming more important in the industry, as illustrated in the US President Executive Order on Improving the Nation's Cybersecurity [ref 16]: "The Federal Government must adopt security best practices; advance toward Zero Trust Architecture". The current implementation of browser-based configuration of IoT and router devices is the antithesis of this. Any solution relying on Wi-Fi based access for security alone is bound to fail.

# Fixing the problem: current approaches

So what can we do about it? Let's start by looking at how industry is solving the problem currently.

## Solution 1: Use a certificate anyway

It's perfectly possible to just use a non-CA issued certificate; indeed many manufacturers do. But what are the problems?



The screen shot above shows you what a Google Chrome browser, for example, does when it comes across one of these non-CA certificates. A non-CA certificate breaches the CA browser guidelines. It is incumbent on the browser to notify the user. That is what the above message does.

It's not a particularly useful message; it basically just frightens people. It is also hard to use. You can click through this message, but many people will not attempt this or work out how.

The result is that although a router or IoT device can support an HTTPS configuration page backed by a non-CA issued certificate, and indeed many do, users are rarely directed to these sites because they generate a lot of concern and inevitably calls to the support desk of the manufacturer.

With this solution, the traffic is encrypted, but it's hard to use, and in some instances still vulnerable.

## And how does the certificate get on the device

Even if we decide to go down this route, there is a world of implementation complexity to address. What type of certificate is it? Is it self-signed, or signed by a root untrusted by the browser? When is the certificate provisioned? How well is the certificate protected on the end device? What are the implications of the private key of the certificate (which must be on the end router or IoT device) being leaked?

The current situation is that different manufacturers take slightly different approaches, each with its own drawbacks. But regardless, the approach is rarely promoted because it's so hard to use.

## Solution 2: Install a new root certificate

Another solution to the problem is to install a new root certificate into the user's web browser (or device from which the browser is launched). In the way browsers currently work, there is a finite set of root certificates installed on each browser. This determines which originators each browser trusts to issue certificates (without generating warnings).

It is possible to install new certificates. Indeed, large enterprises often deploy custom certificates across the organisation's entire device inventory. However, this is not really a practical solution for consumer devices. Not only is it complex to manage from both a logistical and end user perspective, but encouraging end users to add "new certificates" to browsers or operating systems introduces a new attack vector of enormous proportions.

Any solution that habituates a user to the process of installing new security root certificates should be strongly discouraged.

## Solution 3: Use an app

The third and probably the most common solution is to use an application. In this scenario the manufacturer can work around the "insecure connection" problem by encouraging the end user to install a custom application to manage the IoT device or the router.

This does work, essentially because the application can carry its own set of root certificates. It solves both the additional root provisioning problem (by provisioning it with the application), and also the usability problem, as the application is in full control of the user interface.

But this solution is not without its drawbacks:

- 1 It's not scalable:** every new IoT device type, or device ecosystem, needs its own management application.
- 2 Burden on end user:** that's many applications that the user might have to install. If I have 10 different IoT device manufacturers on my internal network, that means 10 different management applications.
- 3 Usability:** it means we have added a new precondition to each IoT device/router installation – that is the need for yet another application.
- 4 Cost to manufacture:** it adds cost and complexity for the manufacturer. A new application needs not only to be developed, but maintained indefinitely.
- 5 Obsolescence and support costs:** applications need to be maintained indefinitely; operating systems updates increasingly render applications obsolete, meaning new versions have to be created, putting burden on user and manufacturer alike.
- 6 Increases the attack surface:** although this solution solves the bootstrap encryption problem, we have increased the attack surfaces for the local IoT ecosystem, by adding an arbitrary number of end applications, where each application-device pairing is free to choose its own mechanism of negotiation for a secure administration channel.
- 7 Control point:** by delegating management to an application, we are entirely dependent on the near-duopoly of application store providers (Google and Apple) to act as distributors of these applications. Given that both Apple and Google have clear commercial intent and ambition to be major players in the IoT space, there is a clear potential commercial conflict of interest here.
- 8 Vender lock-in and dependency:** openness, interoperability and security are fundamentals that should be core to new internet innovations. IoT should be no different.



# Fixing the problem - Secure Usable Intranet Browser (SUIB)

Within the IoTSE we have setup the Secure Usable Intranet Browser working group to address this challenge.

The group is so named because it is scoped to address those conflicting requirements of security and usability, especially as they relate to the internal network browsing problem.

It is an ambitious remit. We are looking to address a critical weakness in current browser behaviours, making them fit for purpose as regards browsing internal internet resources, and in particular enabling them to fulfil a role in the configuration and management of IoT devices.

We believe this is fundamental and important work. Each week new vulnerabilities and attacks are reported. There is a lot of positive industry activity relating to IoT security to address these problems. But unless we solve this foundational problem, many of these initiatives will be for naught.

The ethos of this group is highly practical. With extensive prototyping and testing we are moving forward with a number of complementary approaches to address the issue.

We will be publicising our first draft outline technical requirements and solution documents shortly:

- SUIB Technical requirements document: that need to be satisfied in order to address the problem of connecting and configuring an end point IoT device securely, in a usable fashion over an internet browser.
- Solutions document: outlining the problem dimensions and potential solutions to satisfy the SUIB technical requirements. It is anticipated that some solutions will require working together with standardisation bodies.

## Get Involved

The IoTSE ManySecured SIG are looking for organisations, interested parties and people from across the IoT ecosystem (e.g. manufacturers, Internet and communication service providers, standards bodies, government agencies, browser/software/solution companies, CAs, IoT industry end users and consultants) to join in being part of the solution. If these problems are of concern, you would like to find out more and are interested in joining the SIG, please get in touch: <https://manysecured.net/contact/>

## References

1. Security and privacy in the internet of things: Maple, C., 2017. Journal of Cyber Policy, 2(2), pp.155-184.
2. UK Government, Department for Digital, Culture, Media & Sport and Matt Warman MP, Press Release, "New cyber security laws to protect smart devices amid pandemic sales surge", 21 April 2021: <https://www.gov.uk/government/news/new-cyber-security-laws-to-protect-smart-devices-amid-pandemic-sales-surge>
3. ETSI standardisation organisation: <https://www.etsi.org/>
4. ETSI, European standard, ETSI EN 303 645, "Cyber Security for Consumer Internet of Things: Baseline Requirements": [https://portal.etsi.org/webapp/workprogram/Report\\_WorkItem.asp?WKI\\_ID=57991](https://portal.etsi.org/webapp/workprogram/Report_WorkItem.asp?WKI_ID=57991)
5. National Institute of Standards and Technology (NIST), National Cybersecurity Center of Excellence (NCCoE), "Trusted IoT Device Network-Layer Onboarding and Lifecycle Management Community of Interest": <https://www.nccoe.nist.gov/projects/building-blocks/iot-network-layer-onboarding>
6. Cyber Security Agency of Singapore (CSA), "Cybersecurity Labelling Scheme (CLS)":
7. Internet of Things Security Foundation: <https://www.iotsecurityfoundation.org>
8. ManySecured Special Interest Group: <https://manysecured.net/manysecured-special-interest-group/>
9. CA/Browser Forum: <https://cabforum.org/>
10. CA/Browser Forum, "Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates", 19 October, 2020: <https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.7.3.pdf>
11. Traffic fragmentation attacks: Vanhoef, M., 2021. Fragment and Forge: Breaking Wi-Fi Through Frame Aggregation and Fragmentation. In 30th USENIX Security Symposium (USENIX Security 21): <https://papers.mathyvanhoef.com/usenix2021.pdf>
12. Bleeping Computer, "All Wi-Fi devices impacted by new FragAttacks vulnerabilities", 12 May, 2021: <https://www.bleepingcomputer.com/news/security/all-wi-fi-devices-impacted-by-new-fragattacks-vulnerabilities/>
13. Monitor mode: [https://en.wikipedia.org/wiki/Monitor\\_mode](https://en.wikipedia.org/wiki/Monitor_mode)
14. Promiscuous mode: [https://en.wikipedia.org/wiki/Promiscuous\\_mode](https://en.wikipedia.org/wiki/Promiscuous_mode)
15. Spacehop, "How to Hack Wifi Passwords in 2021" 2 April, 2021: <https://spacehop.com/how-to-hack-wifi-passwords/>
16. US President, The White House, "Executive Order on Improving the Nation's Cybersecurity", 12 May 2021: <https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/>

## Acknowledgements

We wish to acknowledge significant contributions from IoTSF members to this Whitepaper:

Christian Amsüss

Jan Geertsma, Signify

Nick Allott, nquiringminds

Michael Richardson, Sandelman Software

## Peer Reviewers:

Andrew Laughlin, Which?

Carsten Maple, University of Warwick

Duncan Purves, IoT Security Foundation

Hannes Tschofenig, Arm

Ian Poyner, IoT Security Foundation

Ken Munro, Pen Test Partners

Mike Faulks, loetec

Patrick MacGloin, Chartered Institute of Information Security

Paul Kearney, Birmingham City University

Peter Shearman, Cisco

Sam Boswell, Limejump

Steve Babbage, Vodafone

Steve Clark, Wisekey

## Copyright, Trade Marks and Licensing

All product names are trademarks, registered trademarks, or service marks of their respective owners.

Copyright © 2021, IoTSF. All rights reserved

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

