Secure Usable Browser Connections for Intranet Scenarios

Problem Statement

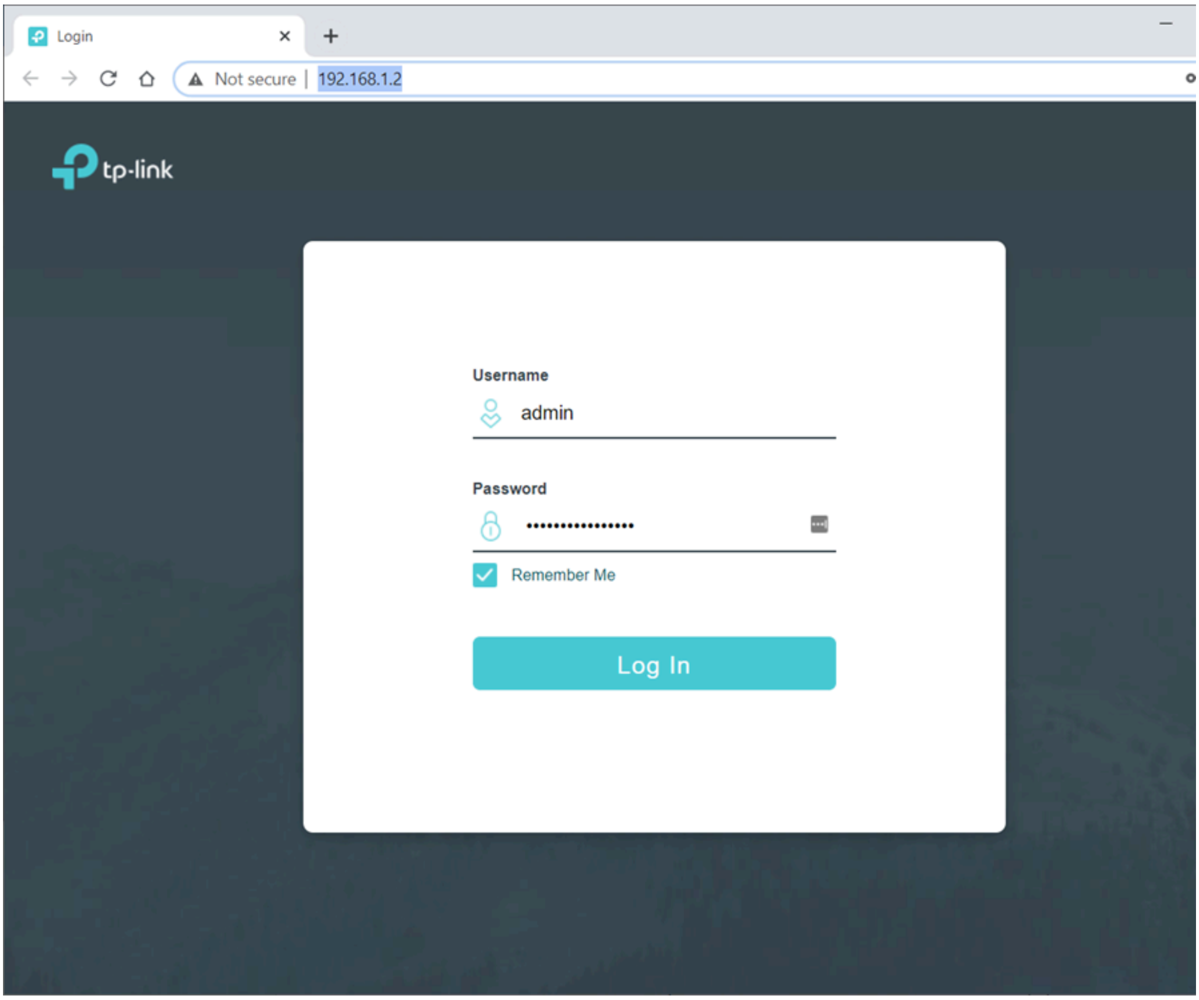
Almost all consumer networking devices and a majority of IOT devices support local HTTP/S connections for management. This browser based interface is the typical default mechanism for managing, configuring and provisioning the device. The instructions in the user guides will give the user instructions to connect to the devices over `http://192.168.0.1` or similar.

The user will of course have to enter their password into a web form on the website to get a connection. Here is where the problem lies. There are two scenarios to consider:

1. Where the manufacture as hosted the administration interface on HTTP
2. Where the manufacture as hosted the administration interface on HTTPS

HTTP - insecure management

If the management interface is hosted on HTTP, then all content will be transmitted in clear text. This includes the transmission of the administration password. Any device hosting their management interface on an HTTP connection, is therefore announcing the users passwords on the internal network.



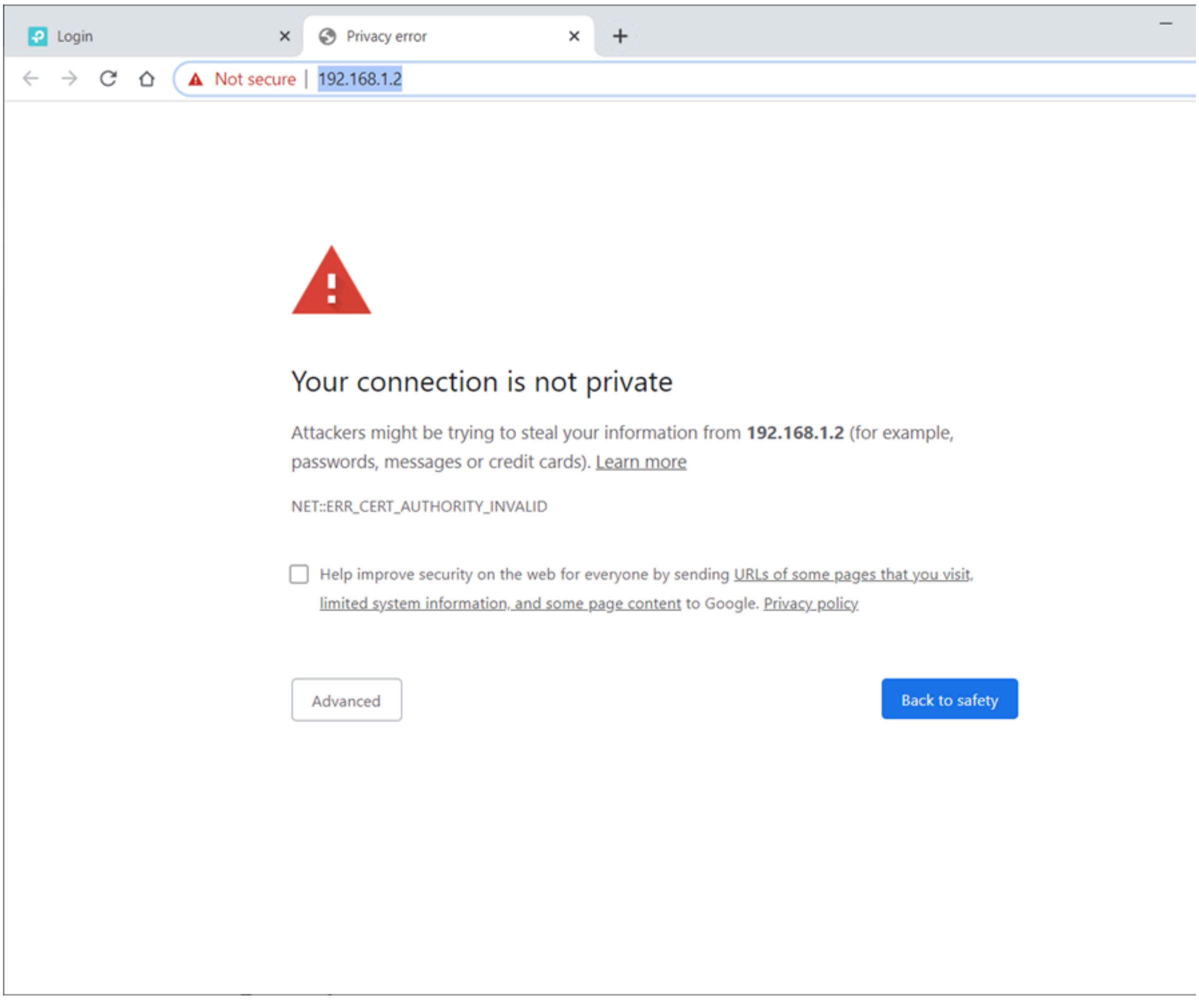
Username
admin

Password
••••••••••••••••

Note above the “insecure” icon by the URI

HTTPS - unusable management

The alternative is to host the management interface on a HTTPS connection. This option provides the assurances of encryption (the password is not passed in the clear), but the solution is unusable for most consumers because of the warnings generated



Your connection is not private

Attackers might be trying to steal your information from **192.168.1.2** (for example, passwords, messages or credit cards). Learn more

NET::ERR_CERT_AUTHORITY_INVALID

□ Help improve security on the web for everyone by sending URLs of some pages that you visit, limited system information, and some page content to Google. Privacy policy

The above screen shot show what the use if faced with if they attempt to access the same management interface, on the same device but this time through the HTTPS interface.

Although some manufactures support this interface for obvious security reasons, they do not want the end users panic because of the warning messsage that cause confusion and raise support questions from end users.

Why use browsers for administration

This raised the question why use a browser to administer and IOT device in the first place?

The reason is simple, and it is the same reason the web has been so successful: a browser provides a universal user interface, for general purpose applications. Through a single application (a browser), a manufacture can implement almost any complex management and administrative interface they like, without creating any installation burden on the end user. Everybody has a browser.

If you cannot use a browser for administration purposes, a manufacture would have to create “new administration” applications at least for each manufacturer, and possibly each product.

The disadvantages of this approach are

- it is costly for the manufacture
- it creates a burden for the end user (installation overhead)
- each new management application, increases the attack surface

A browser by contrast, is typically well proven with a well known and well managed attack surface.

The Design problem

The problem described is by design. It is the unintended consequence of using an “internet tool” the browser, to browse “intranet resources”. It is a knock on effect of the following requirement from the CA/Browser forum

> The CA/Browser Forum in their baseline requirements 7.1.4.2.1

> https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.7.3.pdf
>
> *CAs SHALL NOT issue certificates with a subjectAltName extension or subject:commonName field containing a Reserved IP Address or Internal Name.*

Therefor CA's are forbidden to sign and therefor trust devices based on IP or local hostnames that are not globally unique.

Attack vector

Most networks use a NAT to centralize the access to the internet and conveniently block traffic originating from the internet. But there are ways for attackers to execute an attack.

- Access to the internal network: the attacker must have access to the network (e.g. have the WLAN credentials). This attack cannot be executed from outside the internal network, if it is well configured.
- Through UPNP and portforwarding, gateway devices could allow traffic originated from the internet.
- Through Javascript attacks, the user maybe served content that can attack their own infrastructure.
- The attacker could us a WIFI module that can execute in either monitor mode https://en.wikipedia.org/wiki/Monitor_mode or promiscuous mode https://en.wikipedia.org/wiki/Promiscuous_mode

The principle attack vectors are

- a guest on your network to whom you have granted WIFI credentials
- a guest in your home with access to the credentials printed on the hub, or who presses the WPS button
- an external attacker who cracks your wifi https://spacehop.com/how-to-hack-wifi-passwords/
- 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
- a manufacture of a device, to which you have granted WLAN access in the setup phase

Impact analysis

The potential impact of the current vulnerability of using cleartext management interfaces, cannot be understated.

- It is almost universal, it exists on a large number of IOT devices. Moreover, this vulnerability is present on the lion's share of currently deployed network devices, everything from routers to modems, to wireless access points
- It constitutes a "vulnerability by design". It is baked into existing specifications. A vulnerability discloser could be raised against almost every manufacture and every device currently in market
- It totally bypasses any password policy. It doesn't matter how complex or secure your password is, if its passed in the clear on the local network, the attacker can see and control everything
- It is relatively easy to execute.

Examples

Add in some concrete examples to named devices and the "user guides", which promote this interface

| Device | Link | | — — | — | | | |

Secure Usable Intranet Browser (SUIB) WG - Scope

The Secure Usable Browser working group will be setup to address this challenge.

This group will be chartered to identify workable mitigations to the above challenge, and to progress these mitigations within the industry to reduce the risk.

This group may produce any of the following outputs

- Whitepapers and publish problem statements and risk assessments (published by SUIB)
- Best practice and recommendation documents (published by SUIB)
- Technical documents/ protocols definitions (published by SUIB)
- Liaison statements - clearer recommendations and advisories to third party SDOs and trade bodies

The SUIB group should consider how this problem and potential mitigations address the creation of new devices and the impact on legacy devices already in market.

SUIB operating procedures

The SUIB group, will start initially as a new working group, within IOTSFs existing governance processes.

However, we anticipate moving it rapidly to a new governance process, more fit for purpose for the anticipated outputs.

Draft documents will be created in Github, as markdown documents. Github issues shall be used to track and resolve editing issues.

Candidate solutions

The full scope of candidate solution will be discussed within the SUIB WG itself, but candidate areas are

- Create a Home PKI, where a users can enroll their own devices. (FIDO Alliance)
- Create new X.509 Certificate Key Usage for Home Devices (IETF)
- Request CA/Browser Forum to amend 7.1.4.2.1 to allow signing of internal names (mdns .local, or private IPv4 address or IPv6 local scoped addresses (CA/Browser Forum)
- Create a ‘weak’ trust certificate, where vendors or labels/labs can sign certificates that only show warnings in the address bar. (CA/Browser Forum)
- Promote segregation of home network segregation. (?)

By aligning between standardisation bodies and industries we aim to make current best practices the norm.