



Poly VoIP Devices

SECURITY DISCLOSURE REPORT

ID: MZ-23-01

Version: 1.0

Date: 2023-12-29

Classification: Public

Versioning

Version	Date	Author	Comment
1.0	2023-12-29	modzero	Initial document

Credits

The work contained in this report was conducted over an extended period by the modzero team consisting of (alphabetical order):


- Christoph Wolff
- Pascal Zenker


Disclosure Timeline


Date	Comment
2023-08-23	modzero sends document draft as well as information about the disclosure policy to security@polycom.com .
2023-08-25	The Poly/HP Inc. security team acknowledges receiving the report and investigating the findings.
2023-08-29	Poly/HP Inc. triages the vulnerabilities and sends a list of questions and comments to modzero.
2023-09-01	Poly/HP Inc. thanks modzero for the report and additional explanations, clarifies the disclosure process and how the vulnerabilities will be published.
2023-11-15	Poly/HP Inc. asks for a delayed disclosure date. modzero offers a delayed release of the report by end of December, and release of Proof-of-Concept code in January 2024 after updates have been rolled out.
2023-12-11	Poly/HP Inc. sends an updated list of affected products.
2023-12-26	modzero sends final report to Poly/HP Inc.
2023-12-29	modzero publishes security advisory.


CVEs


The following CVEs have been assigned by MITRE:

Vulnerability	Administrator Session Prediction
CVSS Rating	 8.8 High CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H
CVE-ID	CVE-2023-4462


Vulnerability	Denial of Service Through HTTP Request
CVSS Rating	 7.5 High CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H
CVE-ID	CVE-2023-4463

Vulnerability	OS Command Injection in Diagnostics-Telnet
CVSS Rating	 7.2 High CVSS:3.0/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H
CVE-ID	CVE-2023-4464

Vulnerability	Configuration Import Allows Unverified Password Change
CVSS Rating	 2.7 Low CVSS:3.0/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:H/A:N
CVE-ID	CVE-2023-4465

Vulnerability	Missing Firmware Anti-Rollback Protection
CVSS Rating	 2.7 Low CVSS:3.0/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:L/A:N
CVE-ID	CVE-2023-4466

Vulnerability	Backdoor-Mode Allows Telnet Root Access
CVSS Rating	 6.2 Medium CVSS:3.0/AV:P/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H
CVE-ID	CVE-2023-4467

Vulnerability	Missing Authorization for Cloud Registration
CVSS Rating	 5.7 Medium CVSS:3.0/AV:P/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:H
CVE-ID	CVE-2023-4468



Contents

- 1 Summary..... 5**
- 2 Findings – Network Access..... 8**
 - 2.1 Administrator Session Prediction 8
 - 2.2 Denial of Service Through HTTP Request..... 10
 - 2.3 OS Command Injection in Diagnostics-Telnet 11
 - 2.4 Configuration Import Allows Unverified Password Change..... 13
 - 2.5 Missing Firmware Anti-Rollback Protection15
- 3 Findings – Physical Access..... 16**
 - 3.1 Backdoor-Mode Allows Telnet Root Access 16
 - 3.2 Missing Authorization for Cloud Registration Code 19



1 Summary

Poly Inc., formerly Polycom, is a corporation that develops video and voice communication technology. Their business desk and conference IP phones are popular and commonly used in enterprise business environments.

modzero identified several vulnerabilities in the *Poly CCX* series, a business media desk phone¹ and the *Poly Trio* series, which are smart conference phones². It is confirmed by the vendor that other devices are also vulnerable to some of the same attacks, as they share many software components. The discovered vulnerabilities can be combined to take over a device either through the local network or with physical access to it. An attacker could then employ the device to eavesdrop using the built-in microphones or reroute incoming and outgoing calls. It would also be possible to install malicious applications, attack the connected network or perform phishing attacks on users by prompting for their credentials.

The session tokens generated for the different Poly devices' web management interfaces are using weak randomness. Effectively all tokens generated in the span of a second have the same value and the tokens can be predicted by an attacker due to a deterministic algorithm based on the time in seconds. An attacker with network access can continuously generate valid session tokens for the web management interface, trying to authenticate with them, eventually stealing an administrator session once they log in. One of the discovered vulnerabilities allows an attacker to crash the devices with an unauthenticated HTTP request. They may thus provoke an administrator to log into the Poly device's web management interface thereby enabling the session takeover.

An attacker can then leverage the lack of password protection in the configuration import to override the currently set password and gain persistence on the device. From here, an attacker has multiple options to elevate their access:

On older devices such as the *Trio 8800*, they can enable a diagnostics Telnet shell and use a command injection vulnerability to gain full control with *root* privileges. If they attack a device where the command injection has been patched, for example the *Poly CCX 400*, they can use the management interface to roll back the firmware to an older, vulnerable version and exploit it the same way afterwards.

¹ <https://www.poly.com/us/en/products/phones/ccx>

² <https://www.poly.com/us/en/products/phones/trio>

An attacker with physical access but without administrative privileges can gain these on *Trio* devices with internet connection, by registering them with Poly's management cloud called *Lens*. This can be achieved by navigating to a menu which is not password-protected and using the displayed cloud registration code.

With administrative access on *Trio* devices an attacker can enable the "Test Automation" mode on the device by solving a challenge-response problem posed by the device. By reverse-engineering the algorithm behind the challenge, modzero was able to create a proof-of-concept tool for generating valid responses to these challenges. The only required information is the device's MAC address, which is printed on the bottom of the device. Once the mode is enabled, the devices start an *ADB* and *Telnet* daemon on boot. Both allow unauthenticated shell-level access to the device, to run arbitrary code.

Products that were tested by modzero:

Finding	CCX (8.1.3.1301)	Trio 8800 (7.2.6.0019)	Trio C60 (8.1.3.1300)
Administrator Session Prediction	Vulnerable	Vulnerable	Vulnerable
Denial of Service Through HTTP Request	Vulnerable	Vulnerable	Vulnerable
OS Command Injection in Diagnostics-Telnet	<8.0.2.3267	Vulnerable	<8.0.2.3266
Configuration Import Allows Unverified Password Change	Vulnerable	Vulnerable	Vulnerable
Missing Firmware Anti-Rollback Protection	Vulnerable	Vulnerable	Vulnerable
Backdoor-Mode Allows Telnet Root Access	Not Affected	Vulnerable	Not Affected
Missing Authorization for Cloud Registration Code	Not Affected	Vulnerable	Vulnerable

While not explicitly verified by modzero, the vendor noted that the following devices or product lines are affected at least in part:

- Trio 8300/8500/8800/C60
- CCX
- VVX
- Edge E

Further details about these will be published on the respective product pages³ or at the HP security bulletin site⁴.


³ <https://support.hp.com/us-en/poly>

⁴ <https://support.hp.com/us-en/security-bulletins>

2 Findings – Network Access

This chapter describes all identified findings, where the attacker needs network access.

2.1 Administrator Session Prediction

Class	CWE-330: Use of Insufficiently Random Values
CVSS Rating	 8.8 High CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H
Component	Session Management
CVE-ID	CVE-2023-4462

Summary

The session tokens generated for the different Poly devices' web configuration interfaces are using weak randomness. Effectively only one token per second is generated and the tokens can be predicted by an attacker due to a deterministic algorithm based on the time in seconds. This allows attackers to hijack an active administrator session.

Requirements

The attacker needs network access to the Poly device and the webserver needs to be enabled. An administrator needs to be logged in.

Details


Many Poly devices have a management interface and REST API that can be accessed with network access to the devices. Access to both components is secured by a login mechanism. The application supports the roles *Admin* as well as *User*.

The web configuration application uses session tokens to maintain the authenticated state of a session. The implemented algorithm uses predictable random numbers based on the Unix epoch in seconds. This results in exactly one new token being generated per second which makes it possible for an attacker to predict the given-out session tokens. For example, an attacker can generate past tokens and use them to assume the identity of users and take over their session.

This is especially critical due to the extensive permissions of the admin account. The admin account can fully configure the devices as well as remotely control them via the API. For example, calls can be started, or the screen of the device can be viewed. In addition, the management interface contains various sensitive information such as log files or the possibility, among other things, to send DNS / ICMP requests from the device to the connected network. It does not matter whether the functions are initially activated, because the admin account allows the reconfiguration of the device even during operation. An attacker thus has full access to the described functionalities when taking over an admin session. As further shown in this report, full remote code execution can be achieved once administrator privileges have been gained by an attacker.

It is also possible to continuously generate tokens for the current time and test whether they are valid. As soon as an admin logs in, the valid session can be taken over.

2.2 Denial of Service Through HTTP Request

Class	CWE-20: Improper Input Validation
CVSS Rating	 7.5 High CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H
Component	HTTP Server
CVE-ID	CVE-2023-4463

Summary

An attacker can send a specific HTTP request to the device which crashes the device and results in a reboot.

Requirements

The attacker needs network access to the device and the webserver needs to be enabled.


Details

The webserver of the device expects the user to authenticate through a cookie. When the HTTP header "Cookie" is sent to the server and it contains the expected cookie name ("session") but does not include a corresponding value (refer to Listing 1), the device crashes, and it reboots. While the exact cause is unknown, the HTTP request is handled by a C++ application which seems to fail during the parsing process, potentially leading to a memory corruption. If the attacker keeps sending the request in a continuous loop, the device will remain unusable until they stop.

```
1 GET / HTTP/1.1
2 Host: __HOSTNAME__
3 Cookie: session
```

Listing 1 – HTTP request crashing the device

2.3 OS Command Injection in Diagnostics-Telnet

Class	CWE-78: Improper Neutralization of Special Elements used in an OS Command
CVSS Rating	 7.2 High CVSS:3.0/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H
Component	Command Validation
CVE-ID	CVE-2023-4464

Summary

Poly devices include a diagnostic Telnet mode, which allows an administrator to execute a restricted set of commands, once the mode has been enabled via configuration. Some of these commands are not properly validated and allow for OS command injection as the root user.

Requirements

The webserver needs to be enabled. The attacker needs administrative access to the device and enable the diagnostics Telnet through a configuration import.

Details

An attacker with administrator access, needs to import a configuration which enables the diagnostic Telnet as can be seen in Listing 2.

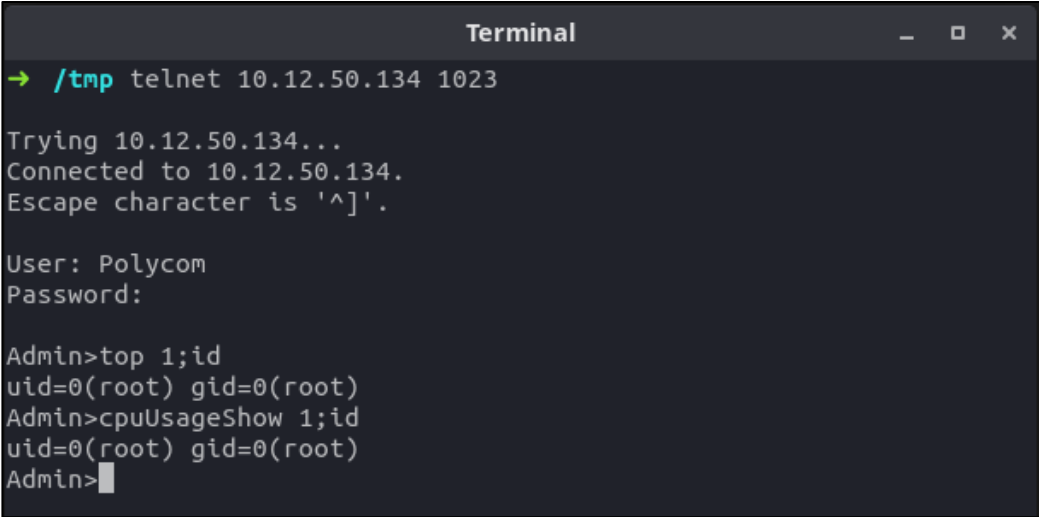
```

1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <PHONE_CONFIG>
3      <CONFIG_FILES
4          diags.telnetd.enabled="1"
5      />
6  </PHONE_CONFIG>
```

Listing 2 – Configuration that enables the diagnostic Telnet

Afterwards the diagnostic Telnet is available on port 1023 of the device. The user for the login is “Polycom” while the password is the administrator’s password. The commands of the diagnostics program are limited and do not allow executing arbitrary Android shell-commands.

At least two commands are not properly validated and allow to perform an OS command injection as can be seen in Figure 1.

A terminal window titled "Terminal" with standard window controls. The terminal shows a telnet session initiated from a directory /tmp. The user connects to 10.12.50.134 on port 1023. The device prompts for a user (Polycom) and password. After logging in as Admin, the user enters two commands: 'top 1;id' and 'cpuUsageShow 1;id'. Both commands result in the output 'uid=0(root) gid=0(root)', indicating successful privilege escalation to root.

```
Terminal
→ /tmp telnet 10.12.50.134 1023

Trying 10.12.50.134...
Connected to 10.12.50.134.
Escape character is '^]'.

User: Polycom
Password:

Admin>top 1;id
uid=0(root) gid=0(root)
Admin>cpuUsageShow 1;id
uid=0(root) gid=0(root)
Admin>
```

Figure 1 – Command injection in diagnostic Telnet

2.4 Configuration Import Allows Unverified Password Change

Class	CWE-620: Unverified Password Change
CVSS Rating	■ 2.7 Low CVSS:3.0/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:H/A:N
Component	Configuration Import
CVE-ID	CVE-2023-4465

Summary

By importing a configuration file specifying a new administrator password, an attacker can bypass the requirement to enter the currently set password. This can be used to gain persistence when a session has been taken over.

Requirements

The webserver needs to be enabled. An attacker needs access to an active administrator-session, e. g., by exploiting the vulnerability outlined in Finding 2.1.

Details

When accessing the management interface of a device, the administrator's password is required. To change the password, the old password must be entered to confirm the change:

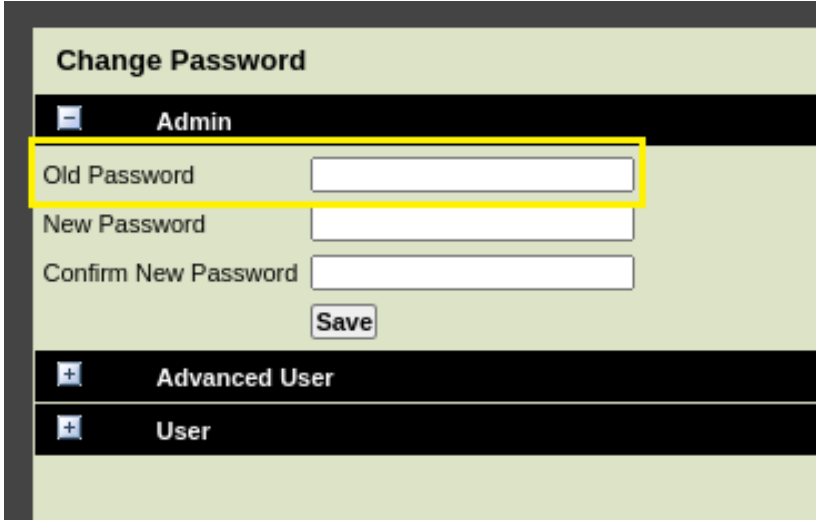


Figure 2 – "Change Password"-dialog in the web interface

By importing a configuration file specifying a new administrator password, this can be circumvented. An example configuration to achieve this can be seen in Listing 3.

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <PHONE_CONFIG>
3   <CONFIG_FILES
4     device.auth.localAdminPassword.set="1"
5     device.auth.localAdminPassword="abctest"
6     device.passwordConfigured="1"
7     device.set="1" />
8 </PHONE_CONFIG>
```

Listing 3 – Configuration example to change an administrator's password.

When an existing session is taken over by an attacker, the lack of a password check allows them to gain persistence beyond the session's lifetime.

2.5 Missing Firmware Anti-Rollback Protection

Class	CWE-693: Protection Mechanism Failure
CVSS Rating	■ 2.7 Low CVSS:3.0/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:L/A:N
Component	Firmware Update Process
CVE-ID	CVE-2023-4466

Summary

The analyzed devices did not include firmware anti-rollback protection. This allows an attacker with administrative access to roll-back or downgrade the firmware to older versions which might allow the exploitation of previously patched security vulnerabilities.

Requirements

The attacker needs administrative access to the web-interface or the Poly cloud management to perform a firmware downgrade.

Details

The management interface of Poly devices (on-device and cloud) allows checking for software updates, as well as installing them. The update interface not only allows uploading newer firmware versions but offers a list of older versions as well:

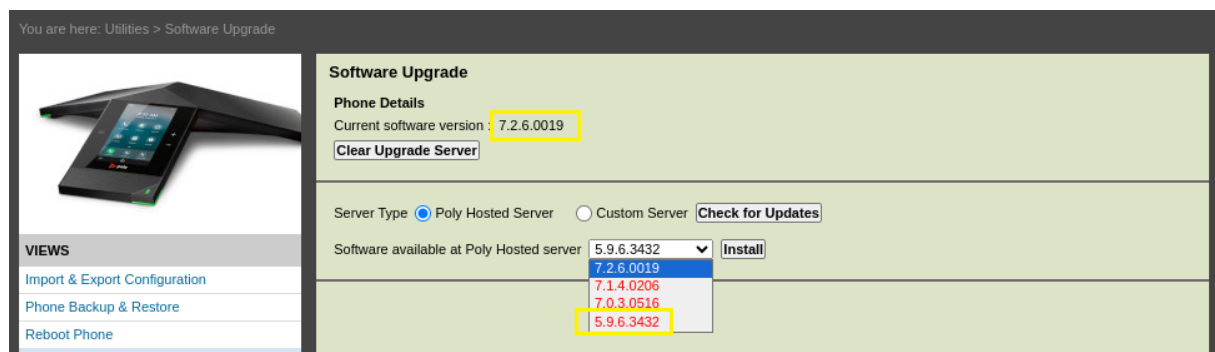



Figure 3 – Interface for managing firmware updates, providing the option to downgrade to version 5.9.6.3432

This functionality can be used by an attacker to install older software versions with known vulnerabilities. It is for example possible to downgrade the Poly CCX 400 on the most recent firmware (8.1.3.1301) to version 8.0.2.3267 or below, which are vulnerable to code execution via the diagnostic Telnet shell (refer to finding 2.3).

3 Findings – Physical Access

This chapter describes all identified findings, where the attacker needs physical access to the devices.

3.1 Backdoor-Mode Allows Telnet Root Access

Class	CWE-912: Hidden Functionality
CVSS Rating	 6.2 Medium CVSS:3.0/AV:P/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H
Component	Firmware
CVE-ID	CVE-2023-4467

Summary

If the hidden functionality called “Test Automation”-mode is enabled, Poly devices start an *ADB*⁵ and Telnet daemon on boot. These allow unauthenticated access with *root* privileges. The functionality can be activated by an attacker with physical access and the administrator password or with shell access.

Requirements

An attacker needs either physical access to the device and knowledge of the administrator password or shell access to the device.

Details

On some Poly devices an attacker can unlock the option “Test Automation” (TA) in the administrative settings interface on the device if the following configuration is imported:

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <PHONE_CONFIG>
3   <CONFIG_FILES
4     httpd.ta.enabled="1"
5   />
6 </PHONE_CONFIG>
```

Listing 4 – Configuration to enable the TA mode

⁵ Android Debug Bridge, <https://developer.android.com/tools/adb>

The mode can then be activated by supplying a response to a challenge offered by the device (refer to Figure 5). Once the mode is activated and the device has been rebooted, it will start an *ADB* daemon on port 5555 as well as a Telnet listener on port 23. These can be accessed without authentication and allow access with *root* permissions. The challenge-response procedure is insecure and predictable as it solely relies on the serial number of the device as well as a hardcoded key. The serial number is equal to the MAC address and is printed on the device and thus not considered secret (refer to Figure 4).



Figure 4 – Photo of the bottom side of a Trio 8800 device, half of the serial number/mac address is hidden

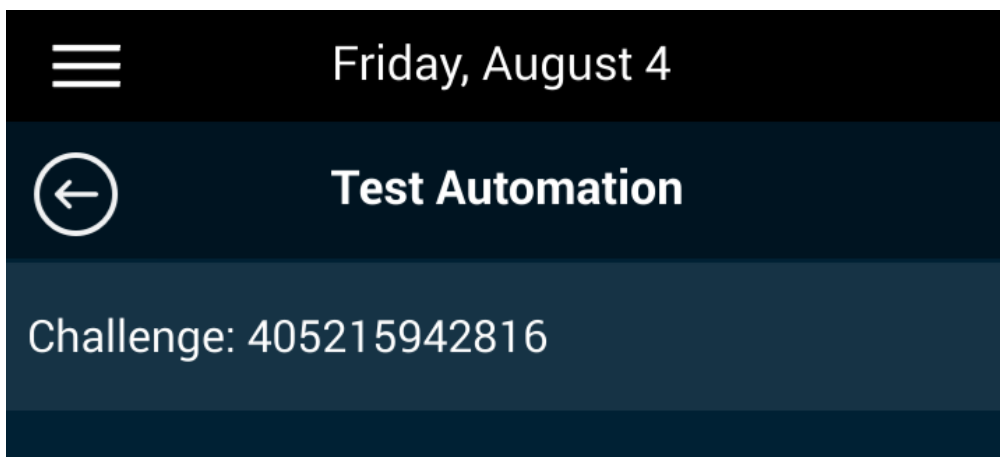


Figure 5 - Test Automation mode challenge

An attacker can simply craft the response code themselves, gaining *root* access to the device in question.


While the attacker needs administrative access to the device, the default credentials are “456”. Additionally, finding 2.1 and 3.2 can be used to gain administrator access without valid credentials.

The TA mode can also be activated remotely with shell access (refer to Listing 5) and during testing was persistent through firmware up- or downgrades. This allows an attacker to downgrade the firmware to a vulnerable version, enable the TA mode and upgrade the firmware to the most recent version while keeping their access.

```
1 busybox printf '\xfe\xca' | dd of=/dev/block/platform/sdhci.1/by-name/plcm_cfg bs=1 seek=131088 conv=notrunc
```

Listing 5 – Enabling TA mode via shell access

3.2 Missing Authorization for Cloud Registration Code

Class	CWE-862: Missing Authorization
CVSS Rating	 5.7 Medium CVSS:3.0/AV:P/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:H
Component	Cloud Registration
CVE-ID	CVE-2023-4468

Summary

To register a Poly device in the Poly *Lens* management cloud, a short alphanumeric code must be entered. This code is displayed on the device, without having to enter the administrator's PIN. Registering the device allows for administrative access via the *Lens* interface, providing an elevation of privilege to a local attacker.

Requirements

An attacker must have physical access to the device and has to be able to navigate to the settings app. The device must be able to connect to Polycom servers for the cloud registration code to be displayed. Already enrolled devices are not vulnerable to this attack, as they do not display their registration code.

Details

Poly devices offer a web interface for remote administration via network access. Apart from this interface, Poly also offers a cloud-based management interface, which can be accessed at <https://lens.poly.com>. To use the *Lens* interface, users first must create an account and then add their devices. To do this, users must enter a 6-character alphanumeric code that is displayed either on the device's screen or its web interface. After registering their devices, most or all functions of the device's management interface are available in *Lens*, including the ability to configure proxies, SIP servers and initiate calls.

The device's registration code can be found in the device's setting, navigating to "Status" and then "Cloud Status":

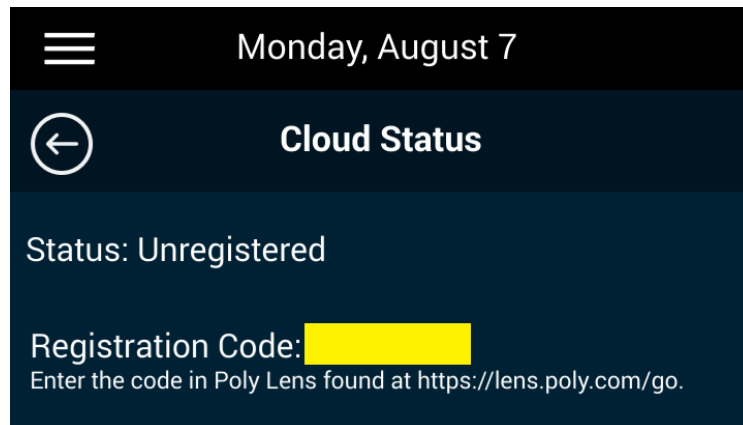


Figure 6 – “Cloud Status” view in settings app

The code is only displayed if the device can connect to Poly’s servers.

To view the device’s registration code on the device’s display, no administrator password or other authentication is needed. Thus, an unauthenticated local attacker can use this feature to register the device with their own *Lens* account and gain remote administrative access to it. This would enable them to eavesdrop on the device’s users by initiating calls or configuring a (SIP) proxy server.