Honeypots for Threat Intelligence in Building Automation Systems*

Johann Bauer, Johannes Goltz, Thomas Mundt, and Simeon Wiedenmann
University of Rostock, Institute of Computer Science, Information and Communication Services Working Group, 18051 Rostock, Germany
{johann.bauer,johannes.goltz,thomas.mundt,simeon.wiedenmann}@uni-rostock.de

Abstract. Direct Digital Controls (DDCs) are components of building automation systems that have a network interface. In this paper we describe how to create a honeypot for such a device. We have replicated the individual services available on the device with extensive logging mechanisms. To do this, we dealt with both protocol-specific as well as operating system-specific properties. We then conduct an experiment in which the honeypots are made publicly accessible on the Internet in order to check whether existing vulnerabilities are already being exploited in the examined devices. We then analysed the data and found no specific attacks for this device. Finally, we suggest possible improvements for future experiments.

Keywords: network security · building automation · honeypot · direct digital control

1 Introduction

In building automation systems, Direct Digital Controls (DDCs) are being used to manage the presets for actors and sensors in the system. Being in such a central position, the security of these devices should indubitably be of great importance.

In [2] one DDC, a Kieback&Peter DDC 4200 was examined and severe security vulnerabilities were found. A quick search using Shodan returns hundreds of these devices publicly available on the internet [13]. Using the vulnerabilities, it is possible to take over the device remotely and control the entire building infrastructure connected to it. Possible attacks range from turning off the lights to trapping people inside a room by using the AC to create a low-pressure atmosphere thus preventing to open a door [7].

* Johann Bauer’s work is co-funded by the Internal Security Fund of the European Union. Simeon Wiedenmann’s work is supported by the German Federal Ministry for Economic Affairs and Energy through a grant for the project Sindabus (Security components for industrial automation and bus systems.)
Fig. 1. Searching Shodan for the DDC device’s web interface yields plenty of results.

We will show how we created a honeypot that pretends to be a DDC4200 from a network perspective. We further make it available on the internet as an experiment to answer the following questions:

– Are attacks against such a device already carried out by malicious actors?
– How can targeted attacks be distinguished from those attacking arbitrary devices?
– How can one prevent honeypots from being recognised as such?

2 Background Knowledge

This section provides a brief overview of the use of DDC devices in building automation. In addition, the basic principles of honeypots are explained.

2.1 DDC devices

In building automation systems, the components are usually divided into one of three levels. The field level contains actuators and sensors. At the automation level, control specifications are implemented based on which sensors are monitored and actuators controlled. These specifications are usually made at the third layer, the management level.

Direct digital control devices are used in the automation level and thus connect the sensors and actuators with the management level. From an attacker’s point of view, these devices are particularly interesting:
Fig. 2. Building automation systems can be divided into three levels—

- they are centrally located within the building automation system
- connected sensors and actuators can usually be addressed without further authentication
- a failure of a single device may lead to extensive system malfunctions
- being embedded devices, they are usually not equipped with defensive software such as a firewall or on-device process monitoring
- being embedded devices, their software is in many cases not updated after the initial release even if security issues arise

2.2 Honeypots

A honeypot is a device or software system that is specifically crafted to attract attacks from the internet or internal networks. Reasons to use a honeypot include distracting attackers from real targets or gathering information about attacks. Honeypots are usually equipped with extensive logging mechanisms in order to study the specifics of incoming attacks. Using honeypots is one of the few ways to proactively defend systems against attacks from the internet.

When a honeypot is exposed to possible attackers, it should react as realistically as possible to deceive them. However, since it’s almost impossible to replicate a real system perfectly, there is a trade-off to be made between realism and effort to create the honeypot. In addition, certain functions of the
honeypot - such as network access - should be restricted in order not to endanger other devices [16]. When designing a honeypot, one should therefore first consider what data should be collected so that a meaningful degree of authenticity is achieved.

3 Experimental Setup

In this section we describe the experimental setup we used to imitate a DDC4200.

In order to imitate the device from a network point of view, the following aspects must be considered:

1. The behaviour of the SSH server on port 22. An attacker has access to all aspects of the operating system.
2. The behaviour of the web server on port 80. Features that should be replicated include the web content, the HTTP headers, the HTTP version and the response delay.
3. The behaviour of the operating system when responding to specific TCP, UDP or ICMP packets. This also includes responses to requests for closed ports.

3.1 SSH Server

The DDC4200 provides a Dropbear SSH server. Information about the SSH server can be obtained even before an SSH session is established. This includes, among other things:

1. SSH version string (for example SSH-2.0-OpenSSH_5.1p1 Debian-5)
2. Supported algorithms for public-key authentication (e.g. Ed25519 and RSA), encryption (e.g. Chacha20 and AES) or compression (e.g. zlib and lz4)
3. SSH host keys
4. Supported authentication methods (e.g. public-key or keyboard-interactive)

To imitate these features, we used the powerful SSH Honeypot Cowrie [9]. By changing the configuration file it is possible to freely select these properties. An exception are the SSH host keys: The DDC4200 uses only DSA host keys. However, Cowrie does not offer the possibility to deactivate the use of RSA keys, these are always offered in addition to any other keys that might be configured. Therefore, it was necessary to change the program code to avoid this. The configuration file and the change to the program code are available in [10].

3.2 Web Server

The DDC4200 also has a web interface. This web interface allows you to access the user interface (see Figure 3 in the browser, which is also displayed on the touch panel of the unit. The actual user interface is not displayed directly via the Document Object Model of the browser. Instead, an image of the user interface
is generated on the device and displayed in the browser. When clicking on the image, the click coordinates are sent via JavaScript to the device, which then returns a new image.

This procedure means that no structured information about the screen content is available on the client side. A program that automatically operates the user interface (e.g. to perform a brute force attack) would have to perform image recognition to recognise the content and perform actions. Such an attack would therefore require more specialised knowledge than an attack that could rely on sending HTML forms.

We have written our own HTTP honeypot to replicate this behaviour, the source code is available at [1]. We have reconstructed the most important menu structures of the user interface and adopted the HTTP headers of the original software. To replicate the slow response behaviour of the device with weak hardware, the software waits a random period of time based on the usual response time of the original when sending a response.

3.3 IP Addresses

In order to deceive potential attackers, the IP addresses must match the specified device and should not indicate that they are being used as a honeypot. For
example, an IP address associated with a virtual server (for example in the network of Amazon Web Services or DigitalOcean) would be inappropriate because the DDC4200 is a physical device and thus not virtualised. For a placement that is as inconspicuous as possible, IP addresses in different networks should be assigned to the honeypots.

For the experiment, we have used five servers in the PlanetLab Europe server network and three private home internet connections to expose the honeypot to the internet.

3.4 Operating System Specific Characteristics

The network code of the operating system can also affect the created packets. Certain aspects may even differ between operating system versions. Nmap uses this circumstance to identify the operating system of a device over the network. A total of 16 packets are sent, including ICMP as well as TCP and UDP to known open and closed ports. The requests are made in such a way that unclear formulations in the standards allow differing answers.

Since all DDC4200s run on the same Linux version, this can be used to detect a honeypot if its operating system version differs from the expected version. However, it is not advisable to use the same operating system version to operate the honeypot, since this version (Linux 2.6.34) is almost 9 years old.

There are tools that simulate certain operating system versions, such as honeypot or IP Personality. However, these are not necessary for the experimental setup described, since the devices are only connected to the Internet via NAT and the operating system detection therefore fails anyway. This is because there is another device involved that responds to most of the probing packets, thus affecting the result.

4 Attack Evaluation

After running the experiment for ten weeks, we collected the data from the honeypots and analysed it. In this section we will evaluate the collected data.

Although some automated accesses to the web honeypots have been recorded, no attacks have been detected. We will therefore limit the following evaluation to the SSH honeypot.

4.1 SSH Honeypot

During the experiment, a total of 62,611 login attempts were recorded on the SSH honeypots. However, none of these attempts used the correct combination of username and password, so that the actions of the attackers could not be recorded. Apparently, the combination of the user name kesroot and an easy to guess password is more secure than expected. Table 1 shows an overview of the most frequently used user names and passwords. It seems like generic credentials were used that are likely to be successful in many cases.
Nevertheless, we have evaluated the data in order to obtain further information. The smallest interval between two login attempts with the most commonly used combination (admin:admin) is only about 17 minutes. Obviously, it is not very likely that the corresponding password was changed within this time span. This may indicate that the login attempts were not centrally coordinated, so it is possible that the attacks may come from different actors.

A representation of the temporal distribution of the attacks in Figure 4 shows how the attacks were distributed throughout the day. There is an increase in the number of login attempts in the time between 18:00-6:00 UTC (which is 20:00-8:00 local time). However, the relative difference between this time period and the remaining 12 hours of the day is small (40% and 60% of attacks respectively), so this property may not be significant. As an explanation, however, it is conceivable that there is malware that prefers to attack servers that are located in a region where it’s night to prevent the attack from being noticed immediately by attentive system administrators. This is currently only a weak hypothesis that would need further investigation to see if it will hold true.

In total, login attempts were based on 2506 different IP addresses, of which 1702 IP addresses carried out only a single login attempt. The locations of the associated geolocation data show that most attacks originate from IP addresses in Russia, followed by Egypt, Thailand, Colombia and South Korea.

Because the preconfigured password for the DDC4200 was not used, it is likely that no tool that was specifically designed to attack a DDC4200 connected to the honeypot.

### Table 1. Top 10 login attempts on the SSH Honeypot

<table>
<thead>
<tr>
<th>Username</th>
<th>Attempts</th>
<th>Password</th>
<th>Attempts</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>14607</td>
<td>&quot;</td>
<td>2350</td>
</tr>
<tr>
<td>admin</td>
<td>12385</td>
<td>admin</td>
<td>1694</td>
</tr>
<tr>
<td>user</td>
<td>2729</td>
<td>password</td>
<td>1338</td>
</tr>
<tr>
<td>ubnt</td>
<td>2153</td>
<td>132456</td>
<td>1260</td>
</tr>
<tr>
<td>test</td>
<td>1581</td>
<td>12345</td>
<td>1059</td>
</tr>
<tr>
<td>ftp</td>
<td>1534</td>
<td>1234</td>
<td>1008</td>
</tr>
<tr>
<td>oracle</td>
<td>1515</td>
<td>root</td>
<td>808</td>
</tr>
<tr>
<td>mysql</td>
<td>1066</td>
<td>ubnt</td>
<td>695</td>
</tr>
<tr>
<td>support</td>
<td>1063</td>
<td>test</td>
<td>683</td>
</tr>
<tr>
<td>pi</td>
<td>808</td>
<td>qwerty</td>
<td>543</td>
</tr>
</tbody>
</table>

5 Future Work

Since no attacks were detected in the course of this experiment, we would make the following change in a future experiment:

In order to investigate whether attackers could see through the camouflage of the honeypot, all incoming and outgoing TCP packets should be logged. This
Fig. 4. Distribution of login attempts based on time of day over the entire measurement period (time in UTC)

would make it possible to trace all interactions that were performed with the honeypot system at the network level. If any advanced methods were used, they would be recorded in the log.

A disadvantage of this method, however, is that this data is difficult to evaluate unless you know exactly what you are looking for. This would probably involve a lot of manual labour.

6 Conclusion

During a ten weeks experiment period with eight DDC4200 honeypots set up, a total of 62,611 login attempts took place. Our investigations found that these attacks came from various different countries around the globe and took place mainly during the evening and night hours at the target position.

In our analysis, we could not manage to correlate any of the observed and analysed attack behaviours as an attack dedicated to the specific target device DDC4200 under investigation.

Further improvements for future experiments have been suggested.
References