IBM Security

Penetration testing a building automation system

Is your "smart office" creating backdoors for hackers?

IBM X-Force® Research

IBM

Click here to start ►

Contents

Executive overview

There is much focus in the IT industry on securing web servers, applications and critical network infrastructure from cyber attack, but what about the new class of web-enabled devices that are connected to the systems that control heating, lights and locks in buildings?

If compromised, such devices may have a more profound impact on our physical surroundings than, for example, the defacing of a web server. Even in an ordinary office building, hackers could gain control of the devices that regulate data center temperatures, causing cooling fans to shut down and servers to overheat. Not only could these devices impact physical surroundings, but through shared connections with enterprise IT networks, they could also open a backdoor to company data.

To help companies understand risks associated with IT systems and how to protect them, the IBM X-Force Ethical Hacking team conducts simulated attacks on various software and systems with the purpose of identifying security flaws. The idea behind ethical hacking, also known as penetration testing, is to identify weak areas and help enterprises implement countermeasures that can strengthen the system.

Our Ethical Hacking team conducted an assessment of a building automation system (BAS) that controlled sensors and thermostats in a commercial office. Working with the system operator and building management, we tested and found several areas of concern in the BAS architecture that could allow a malicious attacker not only to take control of the individual building system, but also to then gain access to a central server, operated by the system operator, which could extend control to several other geographically dispersed buildings.

IBM X-Force then worked directly with all the identified vendors and with the building management to ensure that fixes and mitigations were addressed and patches are now available.

The aim of this paper is to help educate enterprises on how to test for the potential risks associated with building automation systems and how to implement the proper controls to protect those systems. The IBM X-Force Ethical Hacking team presents here the lessons gleaned from this simulation and describes best practices and configuration settings that should help companies develop secure architectures and protections for these building automation devices.

Contents

Project background

Working with the operators of the building automation system, the IBM X-Force Ethical Hacking team attempted to determine if it was possible to break into the main monitoring and control BAS server. This central BAS server controls building automation in several other locations; therefore, access to the central BAS server could allow attackers to control all the buildings in the system.

Each building location is defined as a "station" in the system, and each station does various things. For example, one of the stations, Station 1 in our diagram (see Figure 1), measures the temperature in a computer lab that contains virtual machine (VM) farms and web servers. Keeping this temperature constant is important to preventing the servers from overheating.

Each station is composed of a modem/router for Internet connectivity, a building automation controller, which connects to the central BAS server and reports status, and a series of interconnected sensors and thermostats responsible for controlling the room temperature.

The diagram in Figure 2 represents the architecture of the automation system we tested.

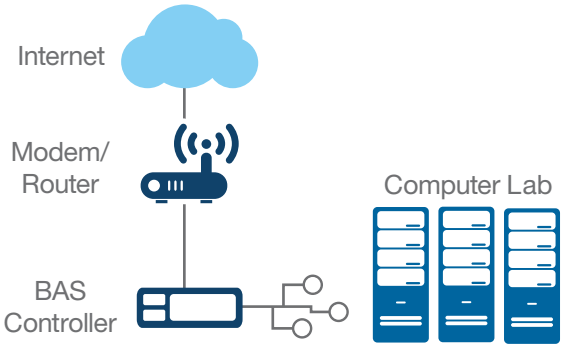


Figure 1. This diagram represents Station 1, which is a building location supported by the building automation system central server.

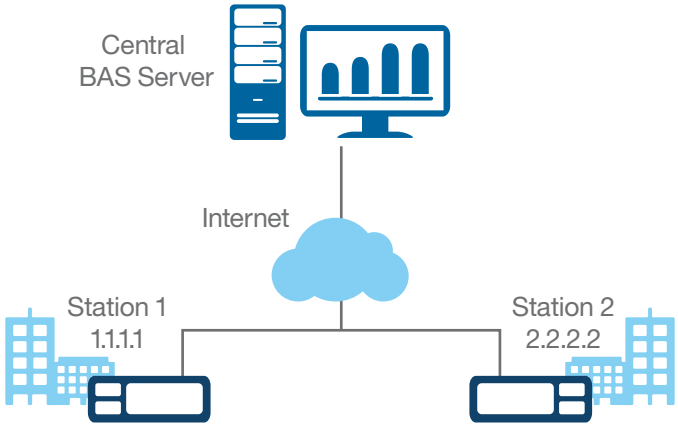


Figure 2. This diagram represents the architecture of the building automation system tested.

Contents

For testing purposes, we were given three public IP addresses by the building system management operator; we will represent those using the fake addresses in the following table:

First Building Site – Station 1	1.1.1.1
Second Building Site – Station 2	2.2.2.2
Central BAS Server	3.3.3.3

Security issues found

What follows is a complete list of security issues and incorrect configurations identified in the building automation system.

1. Exposed administration ports

A scan of the 1.1.1.1 address (Station 1), performed using Nmap, an open source utility for network discovery and security auditing, identified several available ports, detailed in the following table.

443	tcp	open	https	
25	tcp	closed	Smtp	
80	tcp	open	http	
135	tcp	closed	Msrpc	
139	tcp	closed	netbios-ssn	
445	tcp	closed	microsoft-ds	
1234 (edited)	tcp	open	http	Building Automation Controller Administration Port
8080	tcp	open	http	D-Link

Navigating to port http://1.1.1.1:8080, the team was presented with the login screen of a D-Link router.

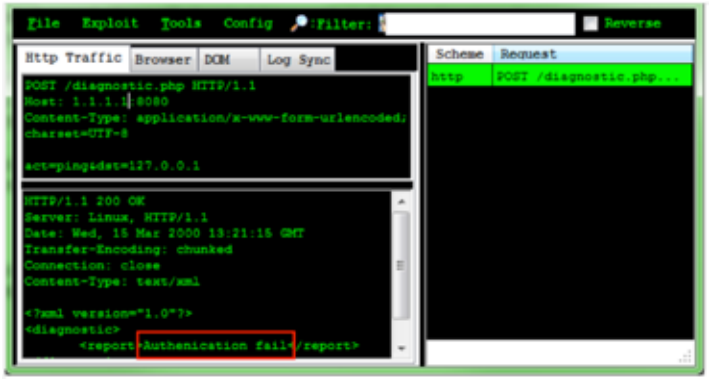
Contents

The login was available because the building system management operator had enabled remote management on the router in order to resolve networking problems remotely.

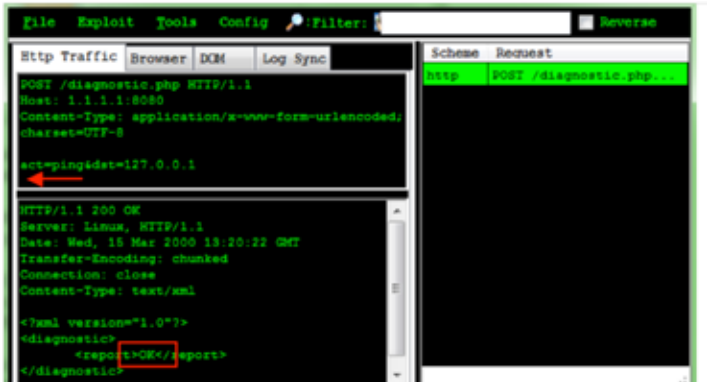
On port 1234(Edited) there was a different, unfamiliar web server showing a platform authentication dialog. The team investigated and determined that this was the building controller running an embedded device framework software instance, which is used to report environmental sensor information to the central server, and that the D-Link router was port-forwarding to it.

2. Bypassing the router login screen

The team attempted to access the router's diagnostic page and received an authentication failure message. This was expected as the router was password protected.



However, the team determined that by adding an extra carriage return after the page request, it was possible to bypass the router's authentication. You can notice in the next screenshot that instead of displaying an "Authentication fail" error the router returns the message OK. This means that the router was manipulated to send ping requests without authentication.



3. Remote command execution on the router

Because the diagnostic page executed a system command (ping), the team needed to assess whether the diagnostic page could be manipulated to execute commands other than ping. This could allow attackers to take control of the router. This type of attack is called command injection and is conducted by replacing the arguments of the request with system commands.

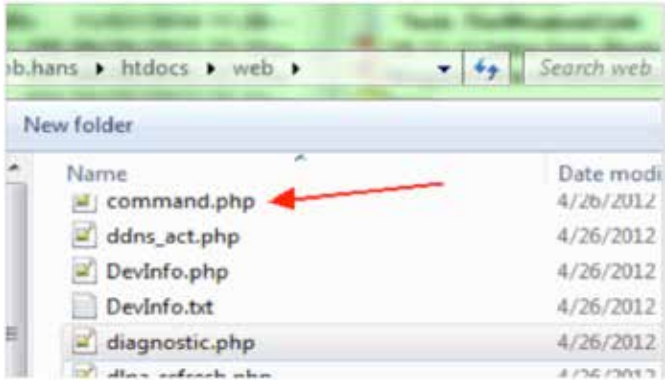
Contents

It is important for testers to recognize command injection vulnerabilities because they often allow attackers to completely compromise a system.

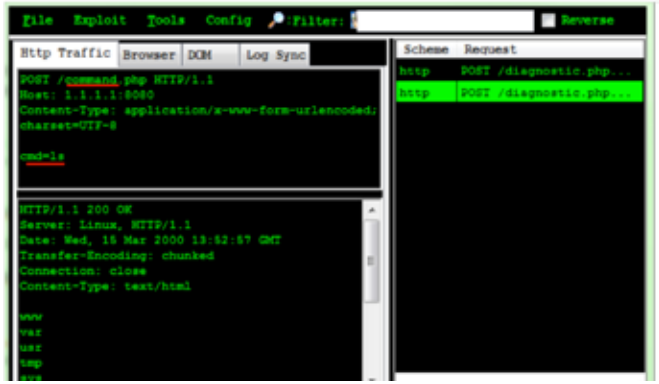
The team tried a black box approach to identify potential ways to exploit the diagnostic page. As sending several commands and combinations of characters to the page was not successful, the team looked for other alternatives.

Since the D-Link router firmware was open source, the team downloaded the entire source code to look at the code of the **diagnostic.php** page and potentially understand more about whether or not it was vulnerable to exploit. This is commonly known in security testing as white box analysis or security code review.

As it turned out, the white box analysis of **diagnostic.php** was not necessary because in the source code, right next to the diagnostic page, there was a page named command.php, which seemed to do exactly what was needed: execute commands.



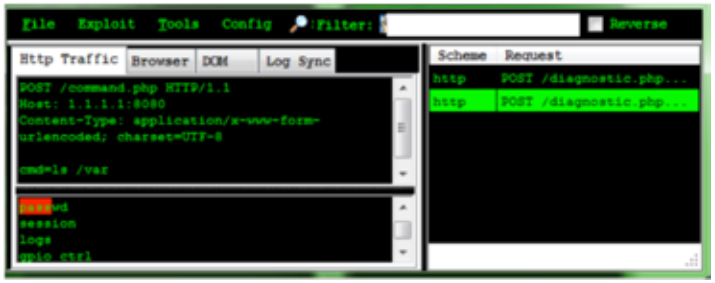
The screenshot that follows shows how an attacker could easily use this vulnerability to control the router like they would control a computer terminal.



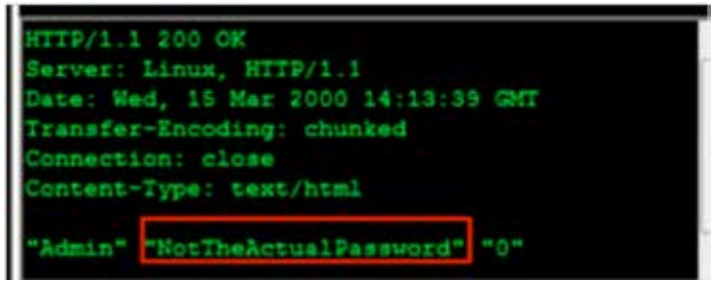
Contents

4. Router password in clear text

The team also looked to see if sensitive data, such as passwords, were available. They found the **passwd** file in the **var** directory.



The contents of this file are shown in the following screenshot. The actual password was in clear text, but it has been replaced with a dummy password in the image.



Storing passwords in clear text is never advisable because if attackers can access the file where the password is stored, they can access the clear text password. This is why it is advised that software use techniques like cryptography to protect stored passwords.

Note that with this password the team was able to log in to the router's administration interface and used the password multiple times during this assessment.

Protect passwords by using techniques such as encryption rather than storing them in clear text.

Contents

5. Using the same password for both router and building controller

Because the task was to determine if the central BAS server for the building system could be breached, the team proceeded to check whether it was possible to take over the station's building automation controller. Going back to the Nmap scan we notice that port 1234(Edited) was open.

...	...	...	...	...
1234 (edited)	tcp	open	http	Building Automation Controller Administration Port
8080	tcp	open	http	D-Link (edited)

The team determined that the building controller device had the same password as the router. The team was then able to log in to its administration interface.

Important security note: *Had the router password been encrypted or if a different password had been used, it would have been much harder for to the team to get access to the building automation controller.*

6. Remote command execution on the building controller

The team had established that it was possible to obtain full control of the individual station. To provide a full assessment of the risk profile for the entire BAS system, the team needed to determine if they could reach that central BAS server.

The same password did not work on the central BAS server located at 3.3.3.3. The team had to establish whether it was possible to obtain the password to the central BAS server from the file system of the building automation controller running on port 1234.

After doing a Google search, the team discovered that the embedded device software running on the building controller provided diagnostic pages that allowed executing commands. For example, the ifconfig command can be executed by navigating to a URL similar to the one below:

http://1.1.1.1:1234/diagnostics/ifconfig!-a (not the actual URL)

By tampering with the URL information and substituting different commands, remote execution on the controller was possible.

Contents

Note: *IBM has notified the embedded device software manufacturer of these issues, and it has taken measures to prevent access to the diagnostics page from a browser. The manufacturer informed IBM that the command set accessible from this interface is limited and that this interface is not intended to be exposed to the outside world. Therefore, to prevent access to this page, we advise that a firewall blocking external connections to the device should be implemented.*

By exploiting this misconfiguration, the team was able to take control of the controller device including accessing configuration files. By extracting one of those files, the team identified the IP address of the central BAS server and admin credentials to it.

7. Ineffective encryption of central BAS server password

The password to the central BAS server was encrypted, which should make it harder to obtain than if it were stored in clear text. However, in this case, the team could decrypt the password by downloading the available Java libraries from the controller device itself and writing a simple Java program to invoke the decrypt method.

To prevent such an attack, keys should be dynamically generated based on device characteristics. This way an attacker would need complete access to the system to be able to decrypt the password. In this case such a countermeasure would have proven effective since the team only had file system access.

8. Using a wireless router for Internet access

The building system management operator had enabled whitelisting of the IP address as a security defense on the central BAS server. Therefore, logging in directly with the decrypted password was not possible.

However, the D-Link router IP was whitelisted and the router had wireless capabilities. Since the team had the password of the router (see issue 4) and all the details of the wireless connection, they verified that it was possible to connect from the parking lot of the building and bypass the IP address restriction. However, this did require being close enough physically to access the router's Wi-Fi network.

Once on the Wi-Fi network, the team was able to log in to the central server's administration interface. From there someone could control sensors and thermostats for several other buildings all across the company.

Contents

Lessons learned

This simulation contained all aspects of penetration testing—even a bit of "wardriving" at the end. The vulnerabilities and misconfigurations that the team found could have allowed someone to gain access to the target system, the main monitoring and control BAS server.

IBM worked with the equipment vendors to address these previously unknown security issues, and they all have released patches. We have also worked with the building automation operator to correct the misconfigurations in any systems they managed.

Based on our experience, we offer the following best practices that organizations should consider when designing and implementing network security architectures for their building automation system environments:

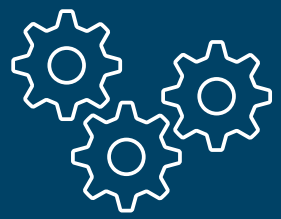
- Ensure that all your device software is up to date as new security issues are found every day.
- Employ IP address restrictions or firewall rules to prevent connections to vulnerable ports.
- Employ a whitelisting approach when designing a BAS network infrastructure, but be aware that firewall rules on their own are not always enough to prevent attacks. Additional controls should be used when needed.
- If there is no business justification for remote access, disable remote administration features on the devices of the building automation system. While remote administration pages can be an easy way to manage the devices, they expose the system to serious security attacks. If remote access is required for business reasons, strengthen the controls around logins. For example, enable two-factor authentication, restrict logins to certain known IP addresses, and increase the alert threshold on failed logins.
- Never re-use or share passwords between devices, and avoid making these passwords predictable. Also, never store passwords in clear text.
- Employ secure engineering and coding practices for authentication control, execution of shell commands and password encryption. Tools such as IBM AppScan® can be used as part of the development process to help identify vulnerabilities in applications and remediate them before the application is put into a production environment.
- Security Incident and Event Management systems (SIEMs) can be used to scan network activity between the router, the BAS system and embedded devices to identify suspicious activity on the network.

Contents

Changing mindsets, policies and technologies to create secure “connected” buildings will take time, effort and investment, but it can be done. In the meantime, companies must start paying attention to the potential cybersecurity risks within their physical spaces, in order to protect their building, employees and data. Continued research in this field will be critical to raise awareness about a potentially serious problem.

About IBM Security

IBM Security offers one of the most advanced and integrated portfolios of enterprise security products and services. The portfolio, supported by world-renowned IBM X-Force research and development, provides security intelligence to help organizations holistically protect their people, infrastructures, data and applications, offering solutions for identity and access management, database security, application development, risk management, endpoint management, network security and more. IBM operates one of the world’s broadest security research, development and delivery organizations, monitors billions of security events per day in more than 130 countries, and holds more than 3,000 security patents.



Building automation systems can offer hackers a way in to corporate networks and data.

Contents

IBM Security

About the authors

Paul Ionescu, IBM X-Force Ethical Hacking Team Lead

Jonathan Fitz-Gerald, IBM X-Force Ethical Hacker

John Zuccato, IBM X-Force Ethical Hacker

Warren Moynihan, IBM X-Force Ethical Hacker

Brennan Brazeau, IBM X-Force Ethical Hacker (Former)

The IBM X-Force Ethical Hacking Team conducts penetration testing services designed to identify systems vulnerabilities, validate existing controls and provide a roadmap for remediation. Drawing on the combined expertise of skilled consultants and industry-leading security assessment tools, the team helps simplify the process of identifying and prioritizing weaknesses while reducing risks and downtime by providing specific guidance and recommendations designed to reduce exposures. With this, clients can improve return on investment by helping strengthen protection of their critical IT assets.

For more information

To learn more about the IBM Security portfolio, please contact your IBM representative or IBM Business Partner, or visit:
ibm.com/security

For more information on IBM X-Force security intelligence, visit:
ibm.com/security/xforce

Follow @IBMSecurity on Twitter or visit the IBM Security Intelligence blog

IBM

Contents

IBM Security

IBM Corporation
IBM Security
Route 100
Somers, NY 10589

Produced in the United States of America
February 2016

IBM

WGL03110-USEN-00