Penetrating physical security: 
the anatomy of physical security systems 
and corresponding defenses

V. Pothamsetty

Critical Infrastructure Assurance Group, 
Cisco Systems, Austin, Texas, U.S.A.

Abstract

Physical security systems and the corresponding networks that the systems run on form a vital component of infrastructure security. The physical security world is embracing Ethernet at a tremendous pace, and yet there has not been much focus on the security aspects of different physical security systems.

In this paper, we analyze the security posture of physical security systems and present the corresponding defense mechanisms. First, we present the anatomy for the most common physical security systems: the building control and automation system (system which controls heating, ventilation, life-safety, etc in a building), the video surveillance system (video monitoring, video storage, video access etc) and the access control system (system which controls and monitors physical entry into or exit from a building such door entry, door alarm etc). We document various device components used in each of these systems, the protocols that they employ for communication, and finally layout their typical deployment network architecture.

We then move on to present a detailed analysis of the defense posture of these systems using well known attack techniques and corresponding tools. Finally, we will present our findings, analyze the underlying causes of the corresponding vulnerabilities and recommend best practice counter measures to mitigate the vulnerabilities found during the analysis.

1 Introduction

The physical security systems automate and/or control the access, monitoring and general operations of a physical location. Vulnerabilities in these security systems
can cause damage not only to property but also to human life. The core protocols which run the physical security systems and the corresponding control system networks are moving to IP; thanks to the cost advantage. But the same trend brings in all the IP based vulnerabilities to the physical security systems and therefore the need for securing these systems has never been more important.

1.1 Disclaimer

This paper is based solely on our own experiences. We, the authors and our employer, Cisco Systems Inc will not be liable for any indirect, special, or consequential damages arising from the use of, failure to use or improper use of any of the content, procedures or tools in this document.

2 Building automation systems

Building automation and control systems are systems which help to automate and control the building services across various facets of a site’s operation. From now on, we will frequently refer to the building automation and control Systems simply as BAS.

2.1 Components, terminology and trends

In this section we will layout different facets of a building automation system, introduce common terms used, and present the prevailing trends in the industry.

2.1.1 The facets and components of a building automation and control system

The facets of a typical BAS are HVAC (Heating, Ventilation and Air Conditioning), fire detection and alarming, lightening control (life safety) and physical security (access control and surveillance). The automation and control services include alarm and event management, trending, scheduling and data sharing.

2.1.2 Trends in BAS

We are observing increased awareness regarding protecting the building automation systems. Recent building automation conferences have included sessions that provide an introduction to network security and its application in this domain. Additionally, talks have been given that discuss the possibility of viruses being written that specifically target building systems [1]. Attempts are being made to bring firewall functionality into control system networks [2, 3].

2.2 Related work

Section 24 of the BACnet standard [4] is dedicated to network security. It is a limited and optional security architecture that provides peer entity, data origin, and operator authentication, as well as data confidentiality and integrity. We do not
know of any vendors who have implemented these features on their devices. The NIST paper [5] does a detailed threat assessment of the BACnet protocol. It categorizes the threats to the BACnet protocol into four types: snooping, application service attacks, network layer attacks, network layer denial of service attacks and application layer DoS attacks. The work has been done in support of the network security working group [6], which is trying to improve security of BACnet protocol through cryptographic measures.

### 2.3 Goals of the section and contribution

Though the earlier work does a pretty good job threat assessment, it does so in a theoretical way. There is no work on the assessment of BACnet devices in the real world and no assessment of “what it takes” to attack these devices and corresponding protocols.

Specifically, the goals of our exercise are:
1. To find out whether an attacker can remotely identify the BACnet devices.
2. To find out whether the BACnet devices are vulnerable to traditional network and stack attacks [look at [7] for a detailed description of stack attacks].
3. To implement and test the vulnerabilities that are described in earlier papers, especially [5].

### 2.4 The BAS protocols

The following are the prevalent protocols in the BAS world:
1. **BACnet**: BACnet [4] is one of the most prominent protocols in building automation networks. BACnet is developed by ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning Engineers). BACnet is a data communication tool for building automation and Control Networks. The BACnet protocol is based on “objects”. An object’s properties are viewed and manipulated through object access services such as readProperty and writeProperty.
2. **ARCnet**: ARCnet [8] is a token based layer 2 protocol, with 2.5 Mbps speed. Its performance characteristics are touted as being well suited for control because of its determinism.
3. **LonTalk**: LonTalk [9] is developed by Echelon corporation. It is a general purpose inter-operable, peer-to-peer control protocol. Most LonTalk devices contain a Neuron chip that is used to achieve inter-operability. The LonTalk protocol defines and implements all seven layers of OSI reference model.
4. **Modbus**: Modbus [10] is one of the early protocols that was implemented on programmable logic controllers. Originally developed my modicon, Modbus is essentially a remote memory access protocol. There are several Modbus derivatives including a version that works with TCP/IP.
2.5 The components of a building automation network

The following are the different components of a building automation network:

1. BACnet Routers (BR) or Gateways: A BACnet router routes packets between different layer two media (ARCnet, BACnet, LonTalk etc) to Ethernet or BACnet/IP and vice versa. The BACnet routers are typically stand-alone hardware devices running Linux or a proprietary OS. Some vendors all call these devices as Building Control Units (BCUs).

2. Software Interfaces: The operator work stations run human software interfaces for BRs and BCUs. The software usually supports multiple operating systems, such as Windows, Solaris and Linux.

3. Communication Bridges: The communication bridges are specialized devices which bridge control protocols such as Modbus and BACnet. These types of devices are also called field servers by some vendors.

4. Programmable I/O Devices: These devices are the end devices in a BACnet network, which speak BACnet natively.

2.6 Communications architecture and our experimental network

We constructed a lab for our testing purposes, with BACnet routers, BCUs and communication software from multiple vendors. The IP enabled BACnet devices communicate with the control software. The control software communicates with the BR/BCU and sends commands to the devices attached to the BR/BCU, either directly or through a communications bridge. A typical BAS architecture is depicted in Figure 1.

2.7 Security analysis of the BAS devices

We analyzed each device for security vulnerabilities. We used traditional security tools such as NMap [11] and IT security methodologies [12, 7]. The following are our goals of the exercise:

- To discover about the open ports and services running on the devices.
- To analyze the security features, if any on the devices.
- To determine whether an attacker can identify the devices remotely.
- To analyze the security vulnerabilities in the applications and services running on the devices.

We summarize the results in the next two sections.

2.7.1 BACnet Routers and BCUs

The following are our findings with regards to BACnet routers:

1. The routers have management ports such as Telnet (23), HTTP (TCP port 80), FTP (TCP port 21) and SSH (TCP port 22) open. The ports are used for management and firmware upgrades.

2. The routers typically have other network related applications running such as time-server (TCP port 37).
3. The routers usually have BACnet software listening on a high port (typically UDP 47808, can be any other port which is known to other BACnet devices).
4. The management ports often do not have any authentication procedures by default. Some BACnet routers have “login” functionality while some do not.
5. UDP scans are unreliable against the BACnet routers because many of the embedded operating systems that the BACnet routers run do not send ICMP port unreacheables for UDP scans.
6. Device fingerprinting is possible because of high open ports.

2.7.2 Control software
The control software usually connects to the BACnet devices for command and control of devices. The following are our observations with regards to control software:

1. The control software runs on various operating systems so it is not possible to fingerprint the control machines based on the operating system.
2. The software runs on a high port, so it is possible to fingerprint the machine based on the high port.
3. The operating system is not typically hardened, for there are unnecessary default Windows services running on the machines.

### 2.7.2.1 Device analysis - conclusions

An attacker will be able to determine that a given device is a BACnet device, either routers or control software. The operating system and applications on both routers and control software machines are not hardened.

### 2.8 Security analysis of the protocols

BACnet does not have security built into it. The BACnet implementations accept packets from other BACnet implementation without any standard security techniques such as authentication, authorization, encryption nor integrity.

#### 2.8.1 Finding BACnet Devices Remotely

An attacker’s first task would be to find out if there is any reliable way to identify BACnet devices on the network. The UDP scans are not reliable enough, because the corresponding stacks often do not reply with ICMP unreachable for any of its UDP ports, whether the port is open or closed. So we tried to push the scanning to the application level, to find out whether a sophisticated attacker can identify BACnet devices on a network. By sending the initial BACnet packet to the BACnet port, we learned that the BACnet devices will reliably reply back with a unicast packet for a BACnet packet with NPDU header and malformed APDU data.

As a proof of concept, we wrote a scanning tool, called *bacnetFinder.py*, which sends the initial packet and finds the BACnet devices on a network based on the replies it receives back.

Here is a sample run of the tool:

```
# python bacnetFinder.py -h A.B.C.D -p port
Got a response from IP A.B.C.D
```

Some devices do not respond to malformed APDUs. In that case the only way to find those devices is to broadcast a “who-is” message. The attacker has to be on the local subnet of such devices.

#### 2.8.2 Gathering data from the BACnet devices found

After finding the devices, the attacker would send a who-is message. BACnet devices that hear it will respond with a *iam [device_id]* message. If the device is not on the local subnet, the attacker can get the device using brute force methods. Since there are 22 bits worth of space for a unique *device_id*, the attacker can try to iterate through them all each time sending a readproperty request. After the attacker determines the *object id* of the device, he can get a listing of most of its "point's".

We wrote a tool called *bacXplor* to perform the above tasks. Here is a sample run of the tool on a BACnet device:

```
bacXplor> known devices
```
DEVICE: 02000001 NAME: BCU-1?
bacXplor>show tree
DEVICE: 02000001 NAME: BCU-1?
  POINT: (ai:1) 00000001 NAME: CHW Supply Temperature?
  POINT: (ai:2) 00000002 NAME: CHW Return Temperature?
  POINT: (ai:3) 00000003 NAME: Ahu-1 Supply Air Temperature?

2.9 BAS - key recommendations

The following are key recommendations with respect to BAS networks:
1. BACnet devices should be placed on a VLAN separate from all other corporate devices.
2. On IP networking devices supporting a BACnet VLAN, filter not only BACnet ports (47808, or any port that the software listens on) but also other miscellaneous ports present on the BACnet devices (22/21/37 etc).
3. Deploy HIDS (non-patch, Host based Intrusion Detection Systems, such as [13]) on PC based BACnet devices.
4. The BACnet device vendors need to harden the operating system and should have mechanisms for IP based access control (IP filters).

3 Access control network

The primary components of an access control network are the door, the card reader interface, the hardware controller, the network connectivity device and the control software. The access network is infrastructure which controls the physical access into a facility and rooms in a facility.

3.1 Component and network architecture details

The door’s card reader is connected to a hardware controller, via a RS-232 link. The controller controls the door with commands such as “door open” or “door close”. The controller makes decisions based on the user database present on it. One or more card reader interfaces can be connected to a single hardware controller through a RS-232 or RS-485 multi-drop. The hardware controller is nothing more than an I/O port logic controller, to which various sensors can be connected such as sensors which can detect window breaks.

The hardware controller communicates with the IP world through a network connectivity device, which acts as a gateway to the IP network. The main user interface to an access network is the access control software, typically running on a Windows machine called access control server. The software has functionalities such as user configuration, and forced door commands. The user credentials (such as biometric templates) are stored in a database on the access control machine itself and are downloaded onto the hardware controller periodically. The access control software can be configured to communicate with the HR database servers for its own database update.
To summarize, the components are

1. **Door Card Reader Interface**: The device against which the user verifies his credentials. It acts based on the commands sent by the controller.

2. **Hardware I/O controller**: The hardware controller sends commands to the interface based on which the interface either opens or closes the door.

3. **Network connectivity device or serial ethernet gateway**: The device which is responsible for connecting the controller to the IP network.

4. **Access control software**: The software running the user interface to the access network, also used to log the events of an access network.

![Diagram of access network components]

**Figure 2**: Typical access network diagram.

### 3.2 Communication details

Whenever a user swipes his card against the card reader, the credentials are sent to the controller. The controller verifies the user credentials and makes a decision on whether or not to allow the user. The controller then sends “door open” or “door
closed” commands back to the door. All this communication happens over a serial link.

On the IP side, the controller communicates the logs and alarms generated to the control software. The hardware controller also accepts various control commands (such as forced door opens or user additions) from the access control software and relays them back to the card reader interface. A typical access network is depicted in Figure 2.

To summarize, the communication links are
1. The link between the card reader interface and the hardware controller; the link is a RS-232 serial or a RS-485 multi-drop.
2. The link between the hardware controller and the serial-Ethernet gateway; the link is a RS-232 serial, with no encryption or integrity.
3. The link between the serial-Ethernet gateway and the access control software; the link is a IP link over the corporate network, the link has no encryption or integrity built into it.

3.3 Operating system details

The controllers run various proprietary embedded operating systems and the access control software typically runs on Windows.

3.4 Threat scenarios and evaluation

The following are the threat scenarios and the discussion of evaluation of the corresponding scenarios:

3.4.1 Remotely send commands to the door
The communication and decision process (for example, the decision of whether or not to open the door) is made in the serial communication link between the card reader interface and the controller. However, the same commands can be sent directly over IP, from the access control software to the card reader interface through the controller.

The protocol used between the access software and the controller does not have any encryption or integrity built into it. Therefore, it is possible for an attacker to send various commands to the card reader interface and to the door remotely. In the actual evaluation, we were able to replay the captured “door open” command and were able to successfully open the door.

3.4.2 Prevent the controllers from opening doors
The controller is not directly connected to IP; the network connectivity device would act as the gateway. Therefore, it is impossible (through packet floods etc) to remotely disrupt the communication happening between the door and the controller.
3.4.3 Make the controller not send alarms
An attacker can flood the gateway on its IP side and would be able to disrupt the communication between the gateway and access control server. However, the gateway typically stores the alarms locally and if it cannot access the server, it waits until the time it can talk to the server. Therefore, unless the attacker can pose as an access server, he would not be able to prevent alarms reaching the server at a later time.

3.5 Conclusion and recommendations
Our key observation of the access control network is that various devices can be spoofed. The vendors need to incorporate encryption and integrity mechanisms on the devices. Network administrators should take care that the access network is not accessible from the corporate network.

4 Video surveillance network
We refer to the network used to capture, store and view the video for security and monitoring purposes as the “video network.”

4.1 Component and network architecture details
The cameras are the end devices of a video network. The cameras are either Ethernet enabled or analog cameras. The Ethernet based cameras are directly connected to corporate network while the analog cameras are connected to the corporate network through a analog to digital converter. Both the two types of cameras typically send their feed to the network video recorder. The network recorder typically has a disk array connected to it where the video is stored. Video can be viewed by connecting to the network video recorder.

In the real world, the video is viewed by connecting to a network video recorder. The network video recorder has provisions for graphical viewing, such as viewing video from multiple IP cameras at a time and the ability to associate live video and recorded video with simple user interface based dialog boxes. A typical video network is depicted in Figure 3.

4.2 Trends in video network communication
On the vendor side, many video monitoring equipment vendors are moving to IP, and consequently, IP based video monitoring systems are growing at a tremendous pace. A lot of IP based cameras have no authentication by default, and such cameras can found by a simple web search [14]. A lot of research has been done on fast and efficient multimedia encryption [15, 16, 17], but we have yet to see vendors embracing the production of video monitoring equipment with cryptographic mechanisms.
4.3 Communication details

The network recorder typically gets the video through HTTP get commands. The video server talks to the video recorder via various MS RPC based protocols.

To summarize, the communication links are:
1. The link between the cameras and the network video recorder; the link is HTTP over TCP, usually over corporate network, and has no encryption or integrity built into it.
2. The link between the network video recorder and the video client is UDP over the corporate network.
3. The link between the network video recorder and the video server; the link uses Microsoft based TCP protocols.

![Diagram of typical video network](image)

Figure 3: Typical video network diagram.

4.4 Threat scenarios and findings

The following are the threat scenarios analyzed:
4.4.1 Sniff video
The video is typically pulled down from the cameras by the network video recorder and is typically not encrypted. An attacker having access to the path of communication between the network video recorder and the cameras would be able to sniff the traffic and therefore capture the video.

4.4.2 Prevent video from getting recorded
The attacker can flood the video cameras and prevent the video getting recorded, but we typically saw a ‘time break’ in the video that was getting recorded. In this case, the recorder is still recording but since it is not receiving any video, it is recording blank images.

4.4.3 Send remote commands to the camera
The camera typically supports an API and the camera typically do not have any integrity built into it. So the attacker would be able to send commands directly to the camera and perform various functions such as pan and zoom. The API’s also support functions to reset/reboot the camera, so a miscreant would be able to do the same.

4.5 Caveats
Most of the commercial solutions bundle the access and video solutions into a single server, which may complicate things.

4.6 Conclusions and best practice recommendations for video networks
The primary vulnerabilities in a video network have to do with un-authenticated and un-encrypted access to video feeds from cameras and through communication links respectively. While the vendors should employ cryptographic mechanisms into the video monitoring equipment, the vulnerabilities can be mitigated through best practice recommended below.

5 Best practice counter measures for physical security networks

5.1 Protecting the communication links
The communication links between various components of physical networks should be protected. The network administrators should be careful when deploying the physical security equipment on the corporate network. In case the physical security links need to be accessible from the corporate network, proper encryption methods (such as segmenting the physical network with a VPN device) and access methods (such as filtering traffic with a Firewall) should be employed. The devices also need to be placed on a separate VLAN, for most of the physical security systems use broadcasting as a primary communication mechanism.
5.2 Protecting the embedded devices

The stack embedded devices of a physical security system needs to be protected by employing proper access-control mechanisms. If the embedded software do not have access control mechanisms, the devices need to be protected behind a firewall and ports should be opened accordingly.

5.3 Protecting the operating system

The operating system on which the routing and control software needs to be hardened. The Windows operating system on which the control software runs need to be updated with latest patches and it is a good practice to run any host intrusion detection (HIDS) software.

6 Conclusion

Physical security systems are inherently vulnerable to traditional network based attacks. The primary reasons for the corresponding vulnerabilities are the lack of security mechanisms in the core protocols used by physical security networks, lack of cryptographic and access-control mechanisms in the devices, and in-secure configuration of devices on the network. Many of these vulnerabilities, however can be mitigated by following some of the best practice counter measures which are presented in this paper.

Acknowledgements

We would like to acknowledge the contributions of our colleagues, Matthew Franz, David Leimbrock, Robert Sayle and Mark Kolar, without which the work would not have been possible.

References