Out-of-Band Remote Computer Recovery Concepts and Strategies

John Prestwich
Schweitzer Engineering Laboratories, Inc.

Presented at the
Power and Energy Automation Conference
Spokane, Washington
March 10–12, 2015

Out-of-Band Remote Computer Recovery Concepts and Strategies

John Prestwich, *Schweitzer Engineering Laboratories, Inc.*

***Abstract*—Out-of-band management refers to a group of technologies that allows the owner of remote computer assets to perform many maintenance and recovery tasks over the network. An especially useful feature of the technology is that it can perform many tasks—such as restoring an operating system—that once required an operator to be physically present at the device. This technology was developed for the information technology industry, so it may not be that well understood by those in the automation industry. This paper gives a general overview of this technology and how it can be applied to a supervisory control and data acquisition (SCADA) system.**

I. Introduction

Out-of-band management (OOBM) refers to a set of technologies that was originally designed for the information technology (IT) industry that allows remote maintenance of network and computer assets. A distinguishing feature of OOBM is its ability to respond to incidents that may render a system inoperative, such as the lockup of the operating system on a computer.

It should be noted that OOBM is used to help address the problem of "soft errors" in a system. A soft error can be loosely defined as an error that can be recovered from by rebooting or reconfiguring the malfunctioning device. Soft errors are introduced into a system by human error, noise in data, and cosmic rays.

The motivation behind OOBM is to reduce costs and downtime by limiting the need to be physically present to maintain a remote system and find the root cause of problems. The features available in an OOBM system are dependent on the components used in its design. These features can range from simple controls like the remote power cycling of an endpoint, to more advanced features like remote keyboard, video, and mouse (KVM) and media (hard drives, USB, and so on) redirection.

While there is a significant amount of information on OOBM technology, it is largely provided by manufacturers promoting a given solution. Because of this, it can be difficult to find literature presenting an independent overview of the design tradeoffs that must be considered. Also, most material is targeted at data centers rather than supervisory control and data acquisition (SCADA) environments. This paper is meant to be a starting point for those in the automation industry who want to learn more about OOBM technology. The following information is presented:

- The technology that makes OOBM possible.
- Design considerations for adding OOBM.
- Security considerations for adding OOBM.

II. OOBM Technology

OOBM uses a master-slave topology similar to many SCADA systems (see Fig. 1). The master is often referred to as a management client, configuration manager, or systems management package. The slave is either the endpoint device, a network node device, or management device (see Fig. 2). More details about each of these components are provided below in this section.

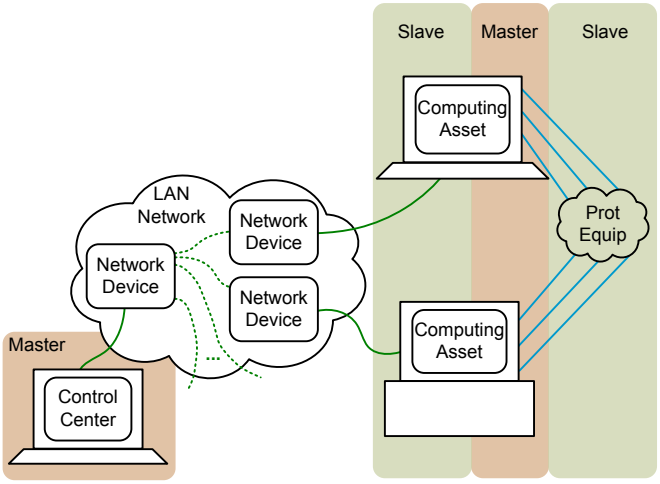

Fig. 1. Basic SCADA Network Layout

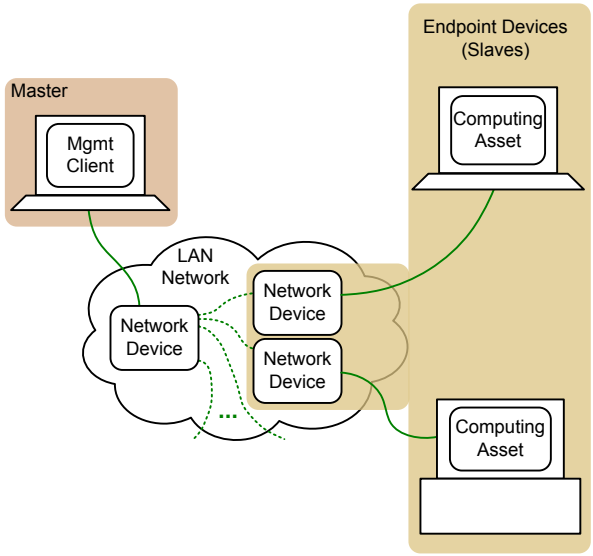

Fig. 2. OOBM Client and Endpoints

A. Management Client

There does not appear to be a universally accepted term for an application that provides remote control of an OOBM system, but to aid in the presentation of this material, this paper uses the term "management client" to generically represent all applications of this type. There are a variety of applications that can serve as management clients, ranging anywhere from a simple serial terminal to an asset management system targeted at IT infrastructure.

OOBM, like SCADA, has had many different solutions over time, each solution using its own set of protocols and technologies. Because of this, one design challenge is ensuring that the management client can communicate with all of the endpoint and management devices. Management clients designed around the needs of IT administrators probably offer the best chances for an integrated solution that incorporates all desired features into one set of applications. In general, IT management solutions are designed around features like asset tracking and update management, which may be useful in a SCADA system as well. OOBM is generally offered as a secondary feature. Often it is provided as a plugin, which allows the management solution provider to accommodate out-of-band (OOB) solutions from different manufacturers. This leads to two design considerations when evaluating management clients:

1. Does the client offer OOBM integration?
2. Which endpoints and management devices does the management client support?

The harder of these two questions to answer is generally the second. Even though the management client may support a given manufacturer's solution it may not support every version of their solution. Different solutions may vary significantly over time in regard to features offered and protocols used.

B. Endpoint Devices

Endpoint devices are the equipment of concern in an OOBM design. These are the computers, network switches, and other devices that require high availability. Although any device that is critical can be deemed an endpoint device, this paper focuses primarily on computers. Network switches and gateway devices are also discussed, as needed, to provide complete coverage of the subject.

OOBM is made possible in many of these devices through the inclusion of a baseboard management controller (BMC) [1]. The BMC is a low-power microcontroller with connections into various system buses, which allow it to control the features of interest. It is connected to voltage rails in the device that are always active so that it remains powered even when the rest of the device is not.

Early BMCs were often add-on options that were installed into the computer through an external bus, such as a peripheral component interconnect (PCI). Most of the computer manufacturers who provide enterprise servers have some form of BMC either built into the mainboard of their computers or as an add-on feature. Many of the more recent middle- to high-performance processors now have BMCs built into their chipsets, making OOBM a standard feature of many business-grade computers.

The external interface to the BMC is called a management port, and in most modern computers this is an Ethernet interface (see Fig. 3). A dedicated management port only provides OOBM functionality and is not directly accessible by the operating system on the computer. A shared port shares network traffic between the BMC and main processor, so when booted, the operating system can use the port in conjunction with the BMC. Management ports have evolved over time and now offer many advanced features that are useful in the recovery and maintenance of an endpoint computer. A few of the more prominent features are described as follows:

- Power management. This is the ability to remotely reset the main processors and peripherals. Both hard power cycling and soft resets are possible.
- Serial console redirection. Many operating systems provide terminal services or low-level kernel access via serial ports. Accessing these serial consoles can often be useful in diagnosing and fixing setup problems. Serial console redirection creates a virtual serial device, which appears to the local operating system like a physical serial port, but its output is transmitted over a Transmission Control Protocol/Internet Protocol (TCP/IP) connection.
- Media redirection. This makes it possible to share media (CD-ROM and USB drives and virtual hard and floppy disks) that are on the management client computer over the network with the endpoint device. These media are available during a reboot, making it possible to boot diagnostic or installation media remotely.
- KVM redirection. This technology gives the user the ability to remotely interact with the video console, keyboard, and mouse. Often access is available even if the operating system is not functioning, allowing access to boot time services like Basic Input/Output System (BIOS) setup.

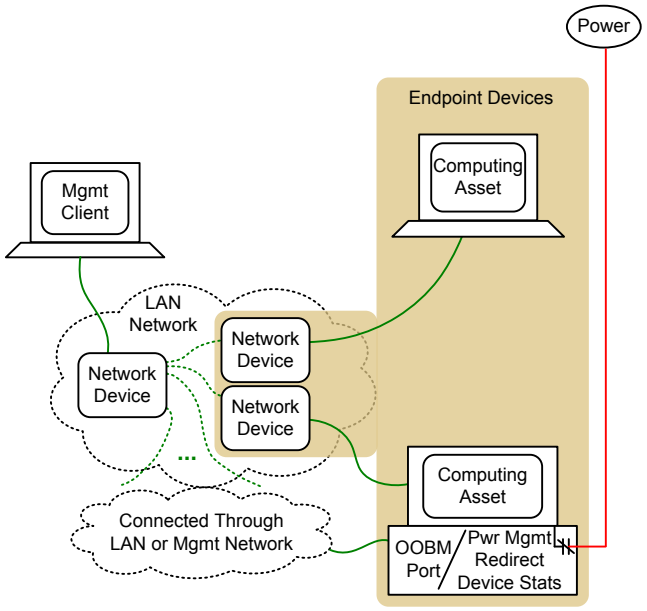

Fig. 3. Endpoint Devices

The protocols used by management ports vary from manufacturer to manufacturer. Because an understanding of these protocols is not critical to administering an OOBM system, an attempt to cover them is not made in this paper. For those who are interested in this subject, the Distributed Management Task Force (DMTF)—a standards organization for system management—website is a good starting point [2]. Their Web Services Management (WS-MAN) protocol is the base on which many management solutions are built.

C. *Management Devices*

The term "management device" is used in this paper to refer to a device that can aid in the management of endpoint devices. Management devices can provide OOB functionality to endpoints that do not have management functionality. Many of these devices are made up of components that are familiar to those designing SCADA systems, such as serial terminal servers (also called port servers) and automation controllers (see Fig. 4).

It is common for IT devices to incorporate several of these functions into one unit. A common mix is an OOBM gateway with a serial port server and automation controller. For clarity, this paper discusses each function individually. The precise packaging of these features is a design consideration that is made in the context of the OOBM requirements of the system. The most common features are discussed below; please note the parallel to the management port features:

- Serial terminal server. The serial terminal server is used to access the management ports on switches, gateways, and so on, as well as serial ports on computers. Many server-class computers and operating systems offer BIOS and kernel-level management via a serial port console. This is the same functionality as serial console redirection in the management port.
- Power control (automation controller). This is an automation controller that provides power supply control for endpoints and network equipment. While it does not allow the degree of control a management port does (such as hard and soft resets), it does provide a basic method for recovering an unresponsive device.
- Internet Protocol (IP) KVM. This is similar to the KVM redirection of the management port in an endpoint device. IP KVMs generally connect to a computer in the same fashion as the standard keyboard, monitor, and mouse do (e.g., VGA and USB ports). Many of these devices allow multiple computers to be connected to them so that multiple endpoints can be accessed from one KVM.
- OOB gateway. OOB gateway devices are the same as standard network gateway devices but are used exclusively for OOB activities. Often gateway devices targeted for this functionality have alternative communications paths built into them in addition to the local-area network (LAN), such as dial-up modem, wireless, and so on (see Fig. 5).

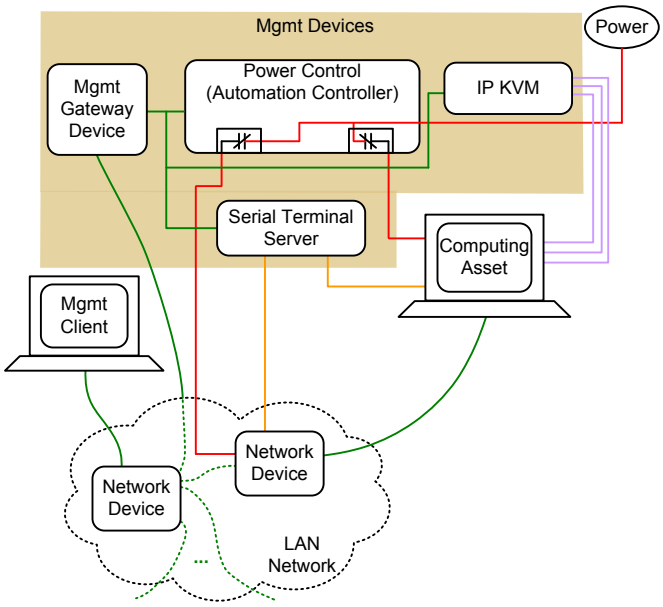

Fig. 4. Management Devices

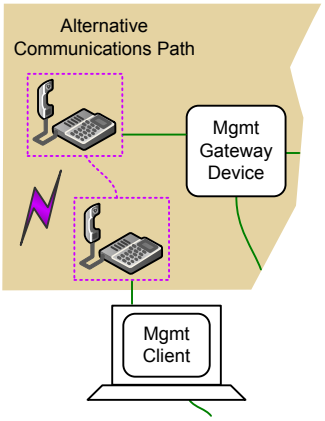

Fig. 5. Alternative Communications Path

III. Design Considerations

A. Initial Planning

The first step when starting an OOBM design is to identify what assets to include in it. Adding OOBM to a system needs to be weighed from both a business and an engineering standpoint. With this in mind, the following questions should be answered:

- What devices are critical to the system, and what are the consequences (in time, cost, security, safety, and so on) of each device being down?
- Which of the computers and network devices are most likely to be disabled by a soft error?

These should be straightforward questions to answer given past maintenance history. Once the questions are answered, the following questions should also be considered:

- Should the systems be retrofitted to accommodate the existing endpoint equipment, replace endpoint equipment with OOBM-capable endpoints, or a combination of both?
  The more homogeneous the components in a system are the easier they will be to maintain and use. Given the nature of the computer market though, it is unlikely that the components of a system will remain the same over time. Upfront planning of how to phase in a design and manage change over time can make for smoother transitions.
- What level of OOB functionality is required?
  Depending on the technology used, functionality can range from remote power cycling and serial console access to being able to completely reimage the hard drive of an endpoint device remotely.
- Is integration with existing business and control systems desired and/or needed?
  Different management client packages and management devices offer varying levels of integration with existing IT and SCADA infrastructure. For instance, with the right choice of a serial terminal server and automation controller, an OOBM system could be integrated into an existing SCADA human-machine interface (HMI).
- What are the security risks of adding OOB access to the systems under evaluation?
  OOB access by its very nature grants administrative-level access to critical equipment, so securing it is of paramount importance. More about security concerns will be covered in Section IV of this paper.

B. Management Network Design

Depending on the complexity of a network, there may be multiple points where a misconfiguration or lockup can disrupt communications. If OOBM features are only reachable through the primary network, the loss of a switch, router, or gateway on that network may prevent remote recovery of the system. A potential answer to this problem is the creation of a dedicated management network. Fig. 6 and Fig. 7 show the differences between using the primary network and a management network for OOBM. A management network should have the following features:

- Solely used for maintenance and recovery activities.
- Accessible by multiple methods, such as LAN, modem, wireless, or cellular.
- Isolated from the primary network, as much as possible.
- Should avoid using complex network components (such as managed switches) where possible.

These features are calculated to keep the management network easy to maintain, reliable, and immune to misconfiguration.

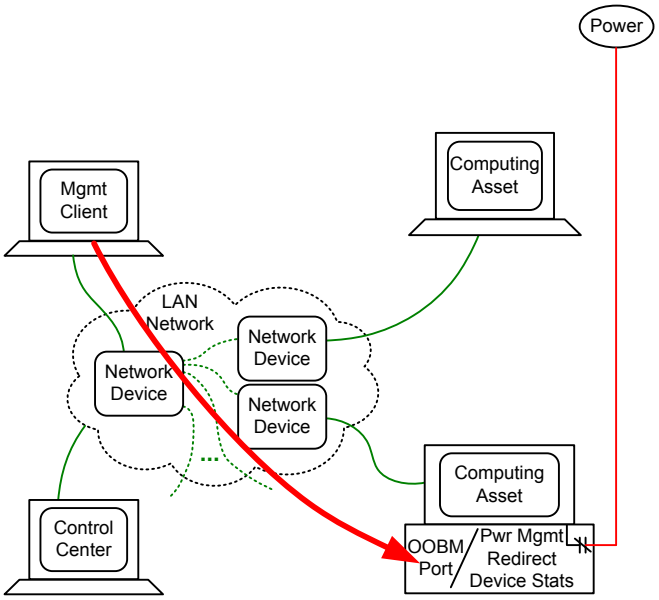

Fig. 6. OOBM Access Through Primary Network

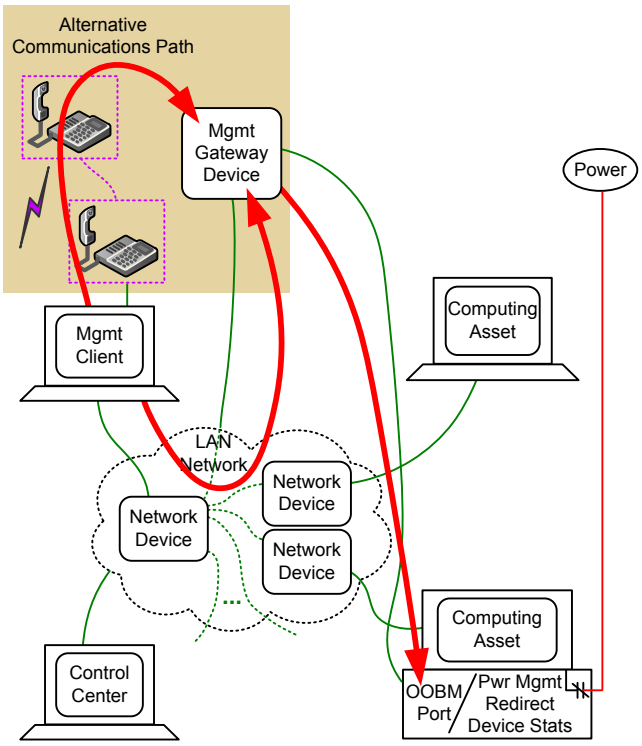

Fig. 7. Use of a Management Network to Provide OOBM

A separate management network may not always be warranted, and whether or not to create one must be evaluated by considering the following:

- The likelihood of a set of endpoints being isolated due to a network device (such as a managed switch) being misconfigured or malfunctioning. Networks that have network devices that are often being reconfigured are good candidates for a separate management network. This is because the likelihood of a misconfiguration increases in proportion to how often the settings of the device are changed. A review of historical maintenance data should help identify systems having these characteristics.
- The additional cost of adding extra network equipment.
- The increased security risk due to additional points of access to the equipment.

C. Recovery Planning

Knowing how the OOB system will be used to recover malfunctioning endpoints is crucial to its design. Remotely recovering a computer is somewhat analogous to doing robotic surgery. It is possible to do all of the procedures that are performed in a standard surgery, but the logistics of how they are performed is different and requires some refining of process to perfect. In the special case of recovering a computer from an unresponsive operating system, this is especially true. Some of the logistics of remotely recovering an operating system are discussed below.

1) Remote Booting

It may be necessary to boot an endpoint computer into an operating environment other than the one installed on its primary boot device. The reasons for doing this may range from the primary operating system not booting to a suspected malware infection. With physical access, this is generally accomplished using a CD-ROM or USB drive. How these steps are accomplished when doing this remotely depends largely on the capabilities of the management and endpoint devices. The two most common alternatives are presented in the subsections below.

a) Media Redirection

As discussed in Section II, Subsection B, media redirection allows local media (CD-ROM, USB drives, virtual disks, and so on) to be mapped to an endpoint device over the network. This mapping is done in such a way that it appears like a physical drive to the endpoint. This service is provided by the BMC through a management port. It should be noted that media redirection is not a standard feature on all BMCs.

While media redirection is a very useful tool, there may be some limitations, depending on the implementation. As of the writing of this paper, the two most notable limitations seen are slow transfer rates and media size restrictions. Some implementations of media redirection intentionally limit the transfer rate to ensure that large file transfers do not choke off other management port communications. This restriction is not noticeable when booting a small operating system but may become troublesome when transferring a new disk image for installation.

The media size restricts the amount of data that can be shared with the remote endpoint. While this is not a critical factor in the boot stage, it can affect the reimaging of an endpoint machine where a large backup file needs to be stored on the media. To work around both of these restrictions, techniques have been devised that use media redirection to boot but transfer image data via a shared network connection.

b) Preboot Execution Environment (PXE) Booting

The PXE (pronounced pixie) specification has been in existence for some time [3], and almost every motherboard has at least one Ethernet interface with this functionality built into it. This makes PXE a viable option for remote recovery. A number of auto-deploy packages (both proprietary and open source) use PXE as the underlying technology for their image deployment. There is also a lot of information available for those who would like to develop their own deployment system.

PXE works by using the Dynamic Host Control Protocol (DHCP) [4] and Trivial File Transfer Protocol (TFTP) [5] to first locate a server and then download a boot image. The reason DHCP and TFTP are used is because they are easy to implement in the space-limited firmware of the Ethernet interface. The simple nature of these protocols does raise some security concerns because they are easy to spoof. Techniques for mitigating these issues are discussed in [6].

Note that PXE booting will not work over a dedicated management port. For a computer to boot via PXE, the network interface must be accessible by the host system.

2) Remote Diagnostics and Repair

This paper recommends that a minimum OOB diagnostic suite consist of the following:

- Boot system analysis and repair tools.
- File system analysis and repair tools.
- Malware detection and removal tools.

Most operating systems have recovery tools to aid in the event of boot failure; third-party repair tools also exist. Many of these tools require some preparation ahead of a failure to realize their full potential. Also, when deploying these tools remotely, they may need to be modified to be compatible with the chosen deployment method. The same is true for file system tools. Often, file system tools and boot recovery tools are packaged together.

The boot process may change from one version of an operating system to the next, so recovery tools from one version may not always be compatible with another. Ensure that recovery tools are prepared for all versions of every operating system that needs to be supported.

Most antivirus solution providers have a bootable recovery tool that allows scanning of the file system and malware removal without having to boot the main operating system of the computer. Even if an antivirus solution is not being used as part of a SCADA system, (for example, when using a whitelisting antivirus instead) it is still recommended that a self-booting antivirus scanning tool be a part of an OOBM tool kit. While the goal is always to prevent malware from reaching a computing asset, it is always best to be prepared to respond if malware does manage to circumvent the existing safety precautions. Also, self-booting antivirus tools are necessary in the removal of rootkit and self-replicating malware. Again, how to deploy the tool remotely is one of the concerns that must be address. Another consideration is having a process to ensure that the virus definition database is up-to-date when it is needed.

3) Remote Reimaging

Many times it is faster to reimage (reinstall the operating system) a computer than to diagnose and repair the problem. To do this as efficiently as possible, some upfront planning is necessary. Some things to consider are:

- Software relicensing.
- Backup image or installation media storage space.
- Scalability.
- Time required to restore the image.

Depending on how a computer is reimaged, the software licenses may or may not be restored. Most software licensing algorithms use one or more physical characteristics of the computer (such as processor or hard drive IDs) to help determine if the software is still installed on the original device. For this reason, backup tools that make exact copies of a hard drive tend to have the highest success rates, provided that the backup image is restored to the same computer from which it was made.

This raises the logistical issue of whether it is practical to store a backup of every computer in the system. With the low cost of network-based storage, it is worth evaluating if this method is practical. The solution needs to not only meet current needs but must also be able to scale to handle future growth. Of all the solutions presented in this paper, this method promises the quickest restore times and is the least likely to involve relicensing issues.

If maintaining full backups of each computer is not practical, an alternative is to have a separate backup image for each of the roles that computers may fulfill in the system. An example of one of these roles would be a substation data concentrator. To help speed a recovery effort, backups of the necessary configuration files for each computer in the system could be kept so that full functionality can be restored quickly. This solution is the best compromise between scalability and time to restore.

Reinstalling from the original installation media is the least desirable method because it is slower than the other two methods and more labor intensive. Despite this, it is occasionally necessary to reimage this way. Ensuring that all installation media are easily obtainable and in a form compatible with the chosen network deployment strategy will speed up the process considerably.

Because it is inevitable that at some point software relicensing will have to be dealt with, it is a best practice to document the relicensing process for each software package that will need to be restored. Some things to consider in this plan are:

- Licensing information storage. Store necessary licensing information in a way that is easy to retrieve and organized. This should include documentation of the restoration process, license key numbers, and backup copies of license files.
- Registration methods. Many software packages use Internet registration as the primary licensing method; if this is not available for the computer, document the alternatives for restoring the license.
- Site and dongle licensing. For entities of sufficient size, site licensing may provide an easier way to manage relicensing issues because most software companies provide more flexible licensing options for their volume clients. USB license dongles may also make relicensing less complicated.

Another practice that can ease the labor required in restoring a computer to working order is the separation of application software and data. This is accomplished by using separate disks or, if only one disk is available, separate logical partitions. The idea is to put the operating system and application software in a physically separate location from the data that it is processing and/or archiving. If only the software or the data becomes corrupt, then only the corrupted portion has to be restored.

D. Conclusions on Design Considerations

The tools needed for an effective OOBM system are the same that are needed to effectively manage any computer

asset. The difference is a need to operate remotely. Remote computer recovery tools can present several logistical challenges, but these challenges can be effectively managed if planned for in advance.

IV. Security

UNIX®/Linux® administrators have a saying that "physical access equals root access." Root is the administrative user built into most UNIX operating systems, so the phrase means that whoever has physical access has administrative access to the computer. This saying holds true no matter the operating system. Enabling OOBM on a system grants anyone able to access it the same privileges as someone with physical access to the asset. This means that OOBM needs to be considered a critical asset in the security architecture of an organization.

A security architecture is the policies, standards, and guidelines by which the security infrastructure (technology, policy compliance, and so on) is measured [7]. A security policy is a high-level document used to express the security concerns of an organization; it does not discuss implementation. The standard provides the criteria by which compliance is measured for a given subsystem of the policy. Guidelines are the implementation details. They provide the technical documentation (product selection, setup, and so on) and processes that are used to enforce the policy [7].

Creating a security architecture is beyond the scope of this paper, but some details of how this is accomplished need to be discussed so that they can be understood in the context of OOBM. For a more complete overview of creating a security architecture and the relationship of OOB equipment to it, please see [7] and [8]. A security architecture addresses the following:

- Data confidentiality and integrity. Confidentiality is the protection of system data from access by unauthorized persons. Integrity is the protection of data from corruption or manipulation by unauthorized persons or systems.
- Availability. Availability is the assurance that a system is capable of performing its function when needed.
- Nonrepudiation. Nonrepudiation is the assurance that all transactions on a system are credited to the correct transactor.
- Auditing and monitoring. Auditing and monitoring is the tracking and analyzing of events on a system.

A list of the vulnerabilities that OOBM introduces to a system must be made with the risks they may introduce to the system. Vulnerabilities are documented in standards documents. An example of how this can be accomplished is given by Kolaks in [8] and is as follows:

- Vulnerability. Some management ports have no authentication systems in place for access.
- Risks. This vulnerability has potential risks in all four of the areas discussed: data confidentiality and integrity, availability, nonrepudiation, and auditing and monitoring.
- Standard. Any management port that does not have authentication control must have an authentication system or device installed. This system can be shared between multiple devices if the access control can be managed in a granular enough method.

Ultimately, a guideline will be created detailing exactly how the standards will be met. Because guidelines are concerned with implementation, it is worth noting that OOBM systems use the same underlying technology that is used in enterprise systems. The same is true when addressing security concerns; the technology and processes that work with enterprise infrastructure are used in OOBM as well. As with any set of assets, there is no panacea when it comes to securing an OOBM system. No one technology or practice makes a system completely immune to attack. This has led to the methodology called "defense in depth."

Defense in depth is a best practice strategy of defending a system by instituting multiple layers of protection so that if one mechanism fails, another is already in place as a backup [9]. A defense-in-depth strategy incorporates the following:

- People. The training of personnel. Teach employees to use strong passwords and to not run unauthorized software.
- Technology. The devices and software used to protect assets and data.
- Processes. The policies and practices used in the defense of a system.

Defense in depth is the same regardless of whether it is applied to an entire enterprise or just the OOB system. Adequate coverage of training and processes are beyond the scope of this paper, so it is recommended to refer to [9] and [10] for more information about these topics. The technology used to provide defense can be categorized as perimeter enforcement, authentication enforcement, encryption, and auditing and monitoring. The following are some insights related to their use on an OOBM system:

- Perimeter enforcement. This includes devices like routers and firewalls that control the flow and type of network traffic to and from a given network segment. Recommendations for OOBM networks are to limit the network communications to just those required for recovery and maintenance work.
- Authentication enforcement. Whenever a device can enforce authentication access in an OOBM network, it is recommended that this feature be enabled. Two-factor authentication is recommended for public-facing perimeters, such as between the Internet and an intranet.

  Ensuring that default passwords are not used is a must. It is also recommended that passwords are not shared between different perimeter and endpoint devices and that they are changed regularly. The use of centralized management and proxy access tools can help ease the burden of managing and updating passwords.
- Encryption. Encryption can apply to both stored and transmitted data. Of particular interest to OOBM systems is the encryption of transmitted data.

Encryption is not a default feature of most OOBM protocols, so it must be enabled or provided by encrypted transport protocols (such as Secure Shell [SSH] tunneling or Internet Protocol Security [IPsec]). Modern management ports may use multiple protocols, so be sure that all protocols are transported over encrypted communications paths. Encryption is essential if management data need to be transmitted over a shared network (such as a network that is not exclusively used for OOBM).

- Auditing and monitoring. This can range from simply logging system activity to intrusion detection systems (IDSs). At a minimum, it is a good practice to log the accessing of assets and the changing of system settings. Many modern management ports provide the logging of connections, connection attempts, and other management activities. Also, many asset management systems allow the aggregation of these logs into a centralized location.

IDSs are either software or devices that actively monitor network communications or host-based events for signs of malicious activity. The most common form of IDS is antivirus software. More useful to an OOBM network are devices that monitor network activity because the endpoint connection is generally a BMC, serial console, or automation controller and not the actual host itself.

When deciding on the security requirements of an OOBM network, the right balance of protection and ease of access needs to be reached. OOBM assets need to be well protected due to their potential for misuse, but if security measures make their use oppressive, then their value is diminished.

V. Conclusion

OOBM is being used in the management of modern data centers to help reduce costs and improve availability. Modern SCADA systems, similar to data centers, have many remote, unmanned computer assets. This attribute is what has driven the addition of OOBM in the data center environment, and likewise, the technology may also prove useful for the SCADA environment. Adding OOBM to a system presents many design challenges, so proper planning is advised to help maximize its usefulness. While there are real security concerns with allowing OOB access to a given system, the concerns are not any different than those of any network system. If properly put into context within the overall security architecture, these security concerns can be effectively mitigated.

VI. References

[1] *–IPMI–Intelligent Platform Management Interface Specification Second Generation*, Rev. 1.1. Intel Hewlett-Packard NEC Dell, October 2013, pp. 32–36.

[2] Distributed Management Task Force, Inc. Available: http://www.dmtf.org.

[3] *Preboot Execution Environment (PXE) Specification*, Version 2.1. Intel Corporation, September 1999.

[4] M. Johnston and S. Venaas, ed., "Dynamic Host Configuration Protocol (DHCP) Options for the Intel Preboot eXecution Environment (PXE)," Network Working Group RFC 4578, November 2006.

[5] K. Sollins, "The TFTP Protocol (Revision 2)," Network Working Group RFC 1350, July 1992.

[6] IBM Corporation, "PXE Architecture and Security Considerations," June 2010. Available: http://www-01.ibm.com/support/docview.wss?uid=swg21247020.

[7] N. Arconati, "One Approach to Enterprise Security Architecture," SANS Institute, 2002.

[8] M. S. Kolaks, "Securing Out-of-Band Device Management," SANS Institute, 2003.

[9] T. McGuiness, "Defense in Depth," SANS Institute, 2001.

[10] National Security Agency, "Defense in Depth: A Practical Strategy for Achieving Information Assurance in Today's Highly Networked Environments." Available: https://www.nsa.gov/ia/_files/support/defenseindepth.pdf.

VII. Biography

John Prestwich received his BS in Electronic Engineering from Utah State University in 1994 and an MS in Computer Science from Boise State University in 2009. He has a diverse background with experience in embedded system design, computer design, computer applications, and information technology. Upon graduating, he worked for Bently Nevada providing custom eddy current proximity sensor designs and also worked with their SCADA collection systems. After Bently Nevada, he worked in the automotive and integrated circuit industries designing and testing embedded computing systems. In 2009, John joined Schweitzer Engineering Laboratories, Inc. in the computing systems group. He now serves this group as a lead application engineer. His interests include robust computer system design, the leveraging of open source software for the control industry, and computer security. John is a current member of the IEEE.

20150122 • TP6679-01