eclypsium

DEFENDING THE FOUNDATION
OF THE ENTERPRISE

THE MISSING SECURITY PRIMER FOR BARE METAL CLOUD SERVICES

Introduction

Organizations are increasingly looking to move their IT infrastructure to the cloud. With the rise of bare-metal cloud offerings, organizations can easily scale up their operations in the cloud while retaining the confidence of having dedicated hardware.

However, while the cloud removes the need to buy and manage this physical hardware, it also takes away some of the control organizations have traditionally enjoyed over their security. It can even introduce new security issues. While physical servers are dedicated to one customer at a time, they don't stay that way forever. Servers are provisioned and reclaimed over time and naturally move from customer to customer. Vulnerabilities in a device's firmware and weaknesses in the reclamation process open the door for firmware implants and rootkits to be passed from one customer to the next. And this could lead to damage and disruption to critical applications and the theft of private data.

The result is a new and somewhat counterintuitive security challenge for any organization that is transitioning assets to the cloud. While the cloud allows organizations to abstract themselves from the underlying hardware, it is only the hardware that remains consistent from customer to customer. As such, it provides the avenue for threats to persist and leap from customer to customer. Much of the daily responsibility for addressing these security risks will fall to the service providers, but organizations will need to employ their own best practices and also be able to evaluate service providers based on their ability to protect the hardware and firmware layers.

In this blog, we will examine the security implications for bare-metal and general cloud services, provide recent research that demonstrates the real-world risks to organizations, and finally provide guidance and best practices that will help IT teams regain control over their security in the cloud, and better evaluate prospective cloud service providers.

We believe that these issues represent a fundamental gap in the security of cloud infrastructure. Our overriding goal with our research is to improve the state of security for our customers and the industry at large. As such, we are eager to collaborate with other organizations and researchers in order to address this and similar security issues. While our case study was based on IBM SoftLayer technology, this is not an issue limited to any one service provider. Firmware vulnerabilities and threats apply to all service providers, and we expect this area of research to remain very active based on the growing importance of cloud services. As always, responsible security disclosure is one of our highest priorities, and we look forward to working with organizations to help identify and fix any issues found in the course of our research.

A Brief Introduction to Cloud Infrastructure

Many modern IT organizations are looking to get out of the business of owning their own hardware and instead are looking to take advantage of the economies of scale provided by cloud infrastructure. With the advent of Infrastructure as a Service (IaaS), organizations can now purchase computing, storage, and network resources in an elastic, on-demand model.

However, most standard IaaS service options will have multiple customers share the resources of an underlying physical server. This may not be adequate for organizations that have high performance requirements for their applications or possess sensitive data that they don't want to have stored on a shared machine.

For these high-value applications, cloud service providers offer bare-metal cloud options in which customers buy access to dedicated, physical servers they can use in any way they see fit. There is no need to worry about buying and supporting hardware—they can grow on-demand as needed.

The Firmware Backdoor Into the Bare-Metal Cloud

While bare-metal cloud offerings can provide considerable benefits, they also introduce new risks and challenges. As with all cloud services, once a customer is done using a bare-metal server, the hardware is reclaimed by the service provider and repurposed for another customer.

This means that even though the hardware is dedicated to a single customer at a given point in time, they could easily be using second-, third-, or nth-hand hardware. Supply chain security has become a major concern for organizations over the past few years, and that is assuming the buyer is the only owner of the hardware. In a bare-metal cloud service offering, the underlying hardware could easily pass through dozens of "owners" with direct access to and control over that hardware.

Why does this matter? In short, it boils down to firmware vulnerabilities and implants. Vulnerabilities in UEFI and server baseboard management controller (BMC) firmware have become all too common. Eclypsium research has previously revealed **firmware vulnerabilities in Supermicro systems** that would allow malware to install backdoors and rootkits to steal information. We similarly found **weaknesses in methods for updating server BMC firmware** that would allow an attacker to install malicious BMC firmware, and subsequently demonstrated how such an issue could be used to **permanently "brick" a server**. As we will see, these vulnerabilities can allow an attacker to not only do damage, but also add other malicious implants that can persist and steal data.

Background: BMC, IPMI, and Out-of-Band Server Management

BMCs have become standard components for most servers and provide management capabilities via the Intelligent Platform Management Interface (IPMI). The BMC is a highly privileged component designed to enable out-of-band management of the server. This could include initial provisioning or an operating system reinstall from a remote management console without the need to physically attach a monitor, keyboard, and installation media to the server.

In addition to external-facing LAN and serial channels, IPMI defines what is known as the "system interfaces," which are communication channels within the server platform itself to allow software running on the host processor to talk to the BMC. This includes KCS (keyboard controller style), SMIC (system management interface chip), BT (block transfer), and SSIF (SMBus system interface). Additionally, IPMB (intelligent platform management bus/bridge) channels can allow multiple BMCs to communicate when more than one BMC is present.

Beyond Bare-Metal: Firmware Threats to Virtualization and Cloud Services

These system interfaces and IPMB channels open the door for threats to move from Internet-facing services to the underlying firmware of the host device. This is because, unlike LAN/serial channels, they are session-less. Session-less channels, such as the system interface/IPMB channels, do not provide a method for authentication.

As a result, malware can potentially send malicious IPMI commands over system interfaces from the host without the commands being authenticated. Since there is no authentication performed when using system interfaces, the only barrier to running arbitrary code within the BMC is whether the BMC itself performs cryptographically secure signature verification of the firmware update image before applying the update. Unfortunately, not all BMCs perform this check, and even when they do, malware can exploit vulnerabilities in the BMC firmware to bypass it.

This has an interesting implication for cloud services in general, even beyond bare-metal services. Any untrusted code, either from a malicious user or a remote attacker, could mount an attack against the device's underlying firmware. The attacker would need to escape the virtual environment, which would be more complex than simply modifying firmware directly on bare metal. But, once successful, one cloud service customer could compromise the underlying firmware and spread to other customers on the same physical hardware. Given the enormous scope of hosted cloud services such as Amazon's AWS and Microsoft Azure, it is an important vector to monitor going forward.

IBM SoftLayer Case Study

So, there are plenty of vulnerabilities (both known and unknown) for attackers to choose from. But it is also worth pointing out that bare-metal cloud services can introduce a new vector by providing attackers with direct access to the hardware itself. An attacker could spend a nominal sum of money for access to a server, implant malicious firmware at the UEFI, BMC, or even component level, such as in drives or network adapters. Then the attacker could release the hardware back to the service provider, which could put it back into use with another customer.

We tested this scenario against IBM's SoftLayer cloud services. IBM acquired SoftLayer Technologies, a managed hosting and cloud computing provider, in 2013; it was subsequently integrated into what is now known as IBM Cloud. SoftLayer offers bare-metal instances in most of its 35 data centers around the globe.

The vulnerability—which, from here on in this report, we will refer to as Cloudborne—we tested for in the experiment is common to many cloud providers and should not be considered limited to IBM SoftLayer. We originally chose SoftLayer for our testing environment because of its simplified logistics and access to hardware but noticed SoftLayer was using Supermicro server hardware that, **based on our previous research**, we knew to be vulnerable. It should be noted that SoftLayer uses other hardware vendors in addition to Supermicro, and Supermicro devices are used by many other service providers.

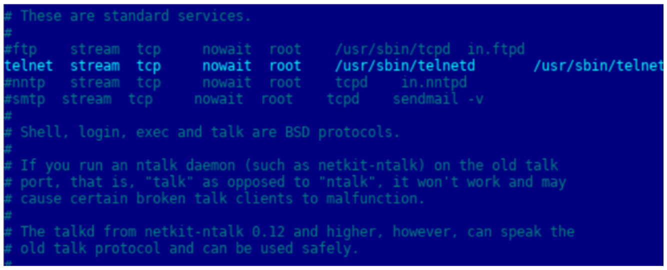
Our goal was to acquire access to a device, make a small change, release it back to IBM for reclamation, and then reacquire the same device from a different user account to see if Cloudborne survived the reclamation process. In our initial investigation, we identified a particular SoftLayer data center that seemed to have a small supply of a particular type of hardware. This small pool meant it would be much easier to reacquire the same device. This trait made the test easier, but this type of vulnerability is not unique to IBM SoftLayer.

It took about 45 minutes to provision the server, almost evenly split between OS provisioning (DEPLOY) and configuration (DEPLOY2) stages. Once the instance was provisioned, we verified that it had the latest BMC firmware available, according to the **Supermicro site**.

```
Device ID                 : 32
Device Revision           : 1
Firmware Revision         : 3.72
IPMI Version              : 2.0
Manufacturer ID           : 10876
Manufacturer Name         : Supermicro
Product ID                : 2114 (0x0842)
Product Name              : Unknown (0x842)
```

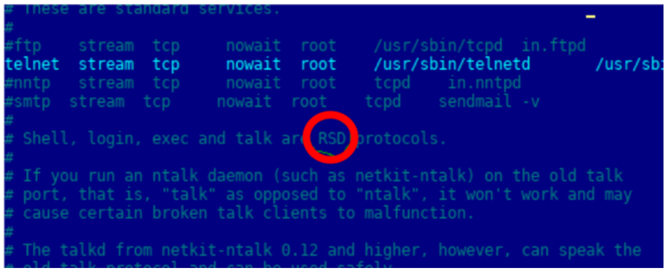
We also recorded the chassis and product serial numbers by running ipmitool fru, so we could identify this system later.

Now that we knew how to recognize the server we were using, we next wanted to make a benign change to firmware. It is important to note that any customer could make this modification without the need for hacking skills. The BMC image was backed up and an image with a single bitflip inside a text file comment was prepared. This bitflip would allow us to recognize if our updated image survived the reclamation process.



```
# These are standard services.
#
#ftp    stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd
telnet  stream  tcp     nowait  root    /usr/sbin/telnetd       /usr/sbin/telnet
#nntp   stream  tcp     nowait  root    tcpd    in.nntpd
#smtp  stream  tcp     nowait  root    tcpd    sendmail -v
#
# Shell, login, exec and talk are BSD protocols.
#
# If you run an ntalk daemon (such as netkit-ntalk) on the old talk
# port, that is, "talk" as opposed to "ntalk", it won't work and may
# cause certain broken talk clients to malfunction.
#
# The talkd from netkit-ntalk 0.12 and higher, however, can speak the
# old talk protocol and can be used safely.
```

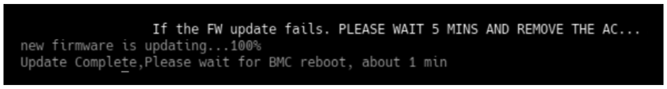
/etc/inetd.conf before the bitflip



```
# These are standard services.
#
#ftp    stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd
telnet  stream  tcp     nowait  root    /usr/sbin/telnetd       /usr/sb
#nntp   stream  tcp     nowait  root    tcpd    in.nntpd
#smtp  stream  tcp     nowait  root    tcpd    sendmail -v
#
# Shell, login, exec and talk are RSD protocols.
#
# If you run an ntalk daemon (such as netkit-ntalk) on the old talk
# port, that is, "talk" as opposed to "ntalk", it won't work and may
# cause certain broken talk clients to malfunction.
#
# The talkd from netkit-ntalk 0.12 and higher, however, can speak the
# old talk protocol and can be used safely.
```

/etc/inetd.conf after the bitflip

Next, we updated the BMC firmware using the AlUpdate tool.

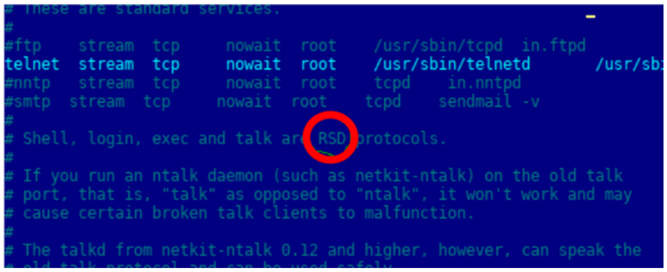


```
                  If the FW update fails. PLEASE WAIT 5 MINS AND REMOVE THE AC...
new firmware is updating...100%
Update Complete,Please wait for BMC reboot, about 1 min
```

We also created an additional IPMI user and gave it administrative access to the BMC channels. The system was then released to IBM, which kicked-off the reclamation process.

Then, a number of other bare metal provisioning requests were made and we were able to reacquire the same piece of hardware that was released earlier. We validated it by matching the chassis serial number with the original as displayed on the asset page and by the ipmitool.

We did notice that the additional IPMI user was removed during the reclamation process; however, the BMC firmware containing the flipped bit was still present. This indicated that the servers' BMC firmware was not re-flashed during the server reclamation process. The combination of using vulnerable hardware and not re-flashing the firmware makes it possible to implant malicious code into the server's BMC firmware and inflict damage or steal data from IBM clients that use that server in the future.



```
# These are standard services.
#
#ftp    stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd
telnet  stream  tcp     nowait  root    /usr/sbin/telnetd       /usr/sb
#nntp   stream  tcp     nowait  root    tcpd    in.nntpd
#smtp  stream  tcp     nowait  root    tcpd    sendmail -v
#
# Shell, login, exec and talk are RSD protocols.
#
# If you run an ntalk daemon (such as netkit-ntalk) on the old talk
# port, that is, "talk" as opposed to "ntalk", it won't work and may
# cause certain broken talk clients to malfunction.
#
# The talkd from netkit-ntalk 0.12 and higher, however, can speak the
# old talk protocol and can be used safely.
```

We also noticed that BMC logs were retained across provisioning, and that the BMC root password remained the same across provisioning. By not deleting the logs, a new customer could gain insight into the actions and behaviors of the previous owner of the device. Meanwhile, knowledge of the BMC root password would enable an attacker to more easily gain control over the machine in the future.

Impacts of Firmware Attacks Against Bare-Metal and Cloud Services

Having seen how an attacker could deliver an implant in a real-world cloud environment, it is important to consider the ways such an attack could be used in the real world. Since firmware underlies even the host operating system and the virtualization layers of a server, any implants would naturally be able to subvert the controls and security measures running at these higher layers. Likewise, given a BMC's ability to control the server, any compromises to the BMC firmware can be particularly powerful for an attacker. Given the nature of the applications and data hosted on bare-metal offerings, this opens up the possibility for high-impact attack scenarios.

Application Disruption: As previously demonstrated by our research, a **malicious implant at the BMC level could permanently disable a server**. We refer to this as a permanent denial-of-service attack (PDoS) or simply "bricking" the server.

Data Theft: With control over the BMC firmware, **attackers can gain access to the data stored on the physical host**. Additionally, attacks against the firmware on drives and network adapters themselves can provide attackers with another very low-level way of stealing or intercepting data. Similarly, with low-level control over the server and network adapters the attacker would have a variety of options for exfiltrating data out of the cloud environment.

Ransomware Attacks: With the ability to disable applications and damage data, attackers would naturally have the ability to perform high-value ransomware attacks. As shown in previous research, attacks against BMC and system firmware can be used to **disable servers entirely**, which also provides an avenue for ransomware attacks.

Best Practices for Cloud Service Customers and Providers

Given the potential impacts of malicious firmware, it's important that both customers and service providers take steps to address the risk.

FOR CLOUD SERVICE CUSTOMERS

For customers, firmware security can be broken into four high-level phases:

1. **Evaluate service providers for vulnerabilities before deploying**: Before making a large time and resource investment in a service provider, prospective customers should evaluate a test system for firmware vulnerabilities.

2. **Validate that new servers are free of implants and backdoors**: Since server firmware could have been modified by the previous tenant's malware infection or intentionally modified by a malicious customer, organizations should evaluate all system and component firmware for malicious implants.

3. **Consider reflashing the firmware of newly acquired hosts**: Customers can validate that the latest firmware is in use and can re-flash the firmware to better ensure they are running a valid image.

4. **Monitor for any firmware changes during server use**: While the server is in use, organizations should regularly check for any newly discovered firmware vulnerabilities as well as any unexpected modifications to the firmware that could result from an intrusion into the system.

FOR CLOUD SERVICE PROVIDERS

1. **Checking for modified firmware during the reclamation process**: The standard server reclamation process should extend to the firmware level to ensure that no changes were made either intentionally or unintentionally to the system or device level firmware. Service providers could choose to analyze firmware continuously, but it should be checked during reclamation at a bare minimum.

2. **Reflashing firmware on hosts after reclamation:** Processes should include updating UEFI firmware via the BMC, and updating the BMC firmware manually.

3. **Ensure data and passwords can't pass from one customer to the next**: In our research we noted that root BMC passwords remained consistent after reclamation, and that BMC logs were retained as well. These are two examples, but service providers should ensure that data of any kind and passwords do not pass from customer to customer. Service providers should ensure all logs are deleted during the reclamation process.

4. **Checking for vulnerabilities and applying firmware updates**: Devices should be continually assessed for new vulnerabilities and all relevant firmware updates should be applied. Devices should be analyzed to ensure all device protections are enabled, and providers may want to consider developing their own custom hardware protections.

5. **Ensure new physical hardware was not tampered with in the supply chain**: Supply chain security has become a major concern for high-value environments, and service providers must ensure that their hardware has not been tampered with prior to delivery.

Conclusion

Bare-metal cloud offerings are typically used for some of the most high-value and sensitive applications in the cloud. However, the cost and management benefits of moving to the cloud can also come with new security challenges that customers and cloud service providers need to be aware of. While it is easy to think of the cloud as a purely virtual environment, vulnerabilities and implants at the firmware level provide an often underappreciated way for threats to persist in the transition from one customer to the next. And while these issues have a heightened importance for bare-metal services, they also apply to all services hosted in public and private clouds. In order to properly secure their applications, organizations must be able to assess for and manage these issues—or run the risk of endangering their most critical assets.

Disclosure Timeline

Sep 6, 2018: Initial advisory sent to IBM at secvm@us.ibm.com

Sep 10, 2018: IBM Vulnerability Management acknowledged receipt of the advisory.

Sep 17, 2018: Eclypsium researchers contacted IBM to follow up and offer more information or answer any questions.

Oct 5, 2018: Eclypsium researchers contacted IBM and asked if they had any updates on their side regarding the Supermicro systems in their infrastructure.

Oct 16, 2018: Eclypsium researchers contacted IBM to inquire about any updates and if they were planning to fix the issue with the Supermicro servers. Offered recommendations and advice.

Oct 18, 2018: IBM Vulnerability Management responded and requested that Eclypsium contact IBM Cloud via the contact forms (**https://www.ibm.com/cloud-computing/bluemix/contact-us**) and (**https://www.ibm.com/security/secure-engineering/report.html**).

Oct 18-19, 2018: Eclypsium submitted the advisory through the supplied links.

Nov 16, 2018: Eclypsium contacted IBM to forward the necessary details to IBM Cloud and notified IBM that Eclypsium planned to publicly disclose the research in the week of December 3.

No response from IBM.

Jan 17, 2019: Eclypsium notified CERT and ICASI.

Feb 20, 2019: Eclypsium provided final draft to IBM.

Feb 25, 2019: **IBM PSIRT published a related blog**. Eclypsium would like to clarify the following concerns related to their response:

- Until this publication, Eclypsium had no indication that IBM had made changes based upon this work. As recently as Feb 16, we had not observed these remediations. We are relieved to learn that IBM appears to be mitigating the issue.
- Eclypsium does not agree with the characterization of this as a "Low Severity" issue. Using CVSS 3.0, we would classify it as 9.3 (Critical) Severity with the following details: **CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H**
- While the hardware specifications of BMC hardware are low as compared with the host server, the capability for security-critical impact is high. By design, the BMC is intended for managing the host system, and as such, it is more privileged than the host. The BMC has continual access to files, memory (using DMA), keyboard/video, and firmware of the host (which is required because it needs the ability to reinstall/reconfigure it). Furthermore, the BMC is able to send data to an external network, even potentially reconfiguring the host network interface. This provides an attacker with all the tools necessary for complete and stealthy control of a victim system. The potential impact includes access/modification of any/all user data as well as permanent denial of service ("bricking") of the equipment as we **previously demonstrated**.

Feb 26, 2019: Research published by Eclypsium.