IPMI: Express Train to Hell, v2.0

dan farmer/zen@trouble.org/7-28-2013

IPMI is a protocol that enables remote management of servers. Designed by Intel and other server vendors it's nearly universal and is widely used for emergency maintenance as well as the provisioning and rollout of applications and operating systems, installation of software, etc.

An embedded server called the BMC implements IPMI and lives on server motherboards; it typically run Linux and has its own little CPU, memory, and storage. The BMC also provides remote web access along with email capabilities, LDAP support, emulation of remote CDs and other media, and a host of other capabilities. The BMC is powerful, and operates and controls the server at a very low-level. Designed to operate when the bits hit the fan it runs even when the server is powered off. Anyone who has control of either the BMC or IPMI (they're closely related) enjoys complete control of the server.

There are three classes of serious security problems with IPMI: the Intel specification, the vendor's implementation of the protocol and the BMC, and how it's all used in the wild by the end users. While none of them are individually showstoppers when combined they create a monumental problem about as large as the Grand Canyon.

The IPMI specification has flaws that allow intruders to access passwords remotely as well as grant system level access without any passwords at all. It also mandates that passwords must be stored unencrypted, a very unusual and unsafe practice which means that anyone who beraks into a server or has physical access to one – say, from purchasing it on eBay, can uncover passwords used to secure your inner corporate sanctum. The same flexibility and power that make IPMI so useful, combined with a lack of cross-vendor security tools, research, and knowledge, make insecure implementations de rigueur.

Vendors frequently rebrand IPMI – Dell has iDRAC, Hewlett Packard iLO, IBM calls theirs IMM2, etc. – but they universally add lots of features and mystery to their implementations. Almost all vendors have the most insecure features of IPMI enabled as a default. Fixing or patching BMC security problems isn't possible because the vendors only allow it to run their own proprietary software. Users have no visibility to any of the activity on the BMC, and no forensics or audit tools exist. Even backing up the firmware is disallowed, so you can't even restore a BMC to a known good state. Most vendors put semi-secret backdoors in their implementations so that their field and support specialists may gain access and control that you cannot. As a final straw: attackers compromising a BMC may make it impossible to dislodge them short of some unknown vendor trick or physically damaging the BMC.

Users tend to manage servers in very large collections that all share the same IPMI password; Groups of 100,000 servers or more that share a common password is not unusual with large organizations. While IPMI doesn't force users to do this both the protocol and vendors almost collude in making it very difficult to do anything else. And because of the difficulty managing IPMI the passwords tend to remain unchanged a very long time, often measured in years, which also makes any password compromise especially serious. **Any** server that is compromised in a group may compromise **all** of the other servers.

In sum, you may not know it, but your goose may already be cooked and you're simply asking for the orange sauce. There is no easy fix, but I'd suggest a dialogue between customers, vendors, and the security community for starters. Vendors must open up these black boxes for change and review and allow customers and third parties to examine and protect their servers. IPMI awareness, security best practices, FAQs, and tools are needed. The de-provisioning of old servers must be handled even more carefully, since eBay attacks and the like are a real threat. Processes and risk management might well need to be changed. And perhaps the IPMI standard itself should be revised.

In any case, good luck. We may all need it.