;login:

THE MAGAZINE OF USENIX & SAGE

November 2000 • volume 25 • number 7

THEME ISSUE: SECURITY

edited by Rik Farrow

inside:

SECURITY DEVICES THAT MIGHT NOT BE

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

security devices that might not be

And How to Approach Them as a Consumer

by Mudge

Mudge is Vice President of Research amd Development for @stake Inc.

<mudge@atstake.com>

Many times we, as consumers of products for the online world, make assumptions about those products' security stance. Everyone would love to assume that any commercial piece of software that they purchase is "secure." After all, it says so on the box. This is a common problem. What about the devices that have an implied security connotation when in fact they might not? Conversely, what about devices that appear to have no bearing on security but upon closer inspection are critical to an infrastructure?

While engaged in some network-design work in the @stake labs, my team and I came across crypto-accelerator appliances. The one in particular that we examined at the time was a self-contained unit. It would boot and run from a memory card and take the burden of encryption off of the end node. In other words, it would act as an invisible device (like a hub) and take HTTPS streams in from the outside world and output HTTP streams on the inside. From the inside nets to the external networks the device would take the HTTP streams and output HTTPS for the appropriate session. Thus the device was required to keep state and session information locally.

Here is an example of a device that contains a public key and a private key, presents a credential as if it were the final end node, and is conducting cryptographic transforms on data passing through it. Instantly one is led to the conclusion that this is a security device. However, closer examination will show that this is not the case and might even present liabilities.

A crypto-accelerator of this type is designed to offload computational work that is processor-expensive for systems. Oftentimes this is done through dedicated hardware on the appliance in custom ASICs. This reduces the load on the end system general-purpose processor so it can go back to serving content, accepting credit cards, and kicking out instructions to other systems as to where to send the goods. Yes, it is in fact a load balancer or coprocessor in nature, much like older systems where you could opt to have a math coprocessor. Few people would think of a math coprocessor as a security device; instead most would consider it a load balancer of some ilk where it is taking the expensive operations and handling them for the main CPU. In reality, though, it could very well be performing the math portions of cryptographic transforms. Here, the device is removing the security blanket to speed the processing on the data within.

Simply having the words cryptography, crypto, crypto-accelerator, certificates, SSL, HTTPS, etc. in a product name or description gives the consumer the impression that what is being used is a security device that is putting security into the mix – not removing it. This is not necessarily the case.

The appliance here is not intended to protect the end systems. It is not even claiming to protect itself. In fact, one can argue that it is now more important to secure the back-end network, as the traffic is not actually encrypted all the way to a final destination, and thus the potential for monitoring and compromise of confidentiality is exaggerated.

If you see a device in your network that is designed to be appliance-like and offer security, be very suspicious.

Does this present a problem? Only if it caught the consumer off guard. A little analysis up front can go a long way.

- The device is used to remove a security layer.
- The device is designed to be largely "plug and play."
- The device is an embedded system with no moving parts.
- The vendor offers remote support.
- The owner can remotely manage the device.
- The owner can locally manage the device.

If we abstract the above to more generalized security devices, or nonsecurity devices that have an implied security component, we can take the first four items above and elaborate a bit.

THE DEVICE IS USED TO REMOVE A SECURITY LAYER

In the real world this unfortunately often translates to a lax security stance in the design stage. The goal in the above example is to strip the HTTPS coming in on one end and spit out raw HTTP on the other. A relatively simple goal, if that is all one is thinking about. If one were working in the other direction, of introducing security in an embedded system, one would (hopefully) think about how to harden the system itself. The notion of not caring about the identity of the end node connecting, just that the session is encrypted but not necessarily authenticated, lends itself to this poor stance. This is an important area to analyze before deployment. Was the vendor lackadaisical and not treating the device as security relevant?

THE DEVICE IS DESIGNED TO BE LARGELY "PLUG AND PLAY"

This should almost always raise a large, red warning flag when seen in conjunction with "security devices." If there were a silver bullet, one-size-fits-all solution, then there would be no need for all of the different products and vendors. There would be one operating system. No need for public markets, etc., etc.

To be honest, Microsoft even gets a somewhat unfair rap on this count for security. One of their main goals is to sell an operating system that is ubiquitous. To do so their product must need minimal – or more appropriately no – custom configuration in order to work in all environments. The same build-and-stock configuration must exist in academic, military, corporate, medical, and personal environments. A custom build for each area and the associated support costs would be prohibitive. We wonder why there are so many security ramifications? Because we, the consumer, have demanded that it be largely "plug and play" for all environments. If you see a device in your network that is designed to be appliance-like and offer security, be very suspicious.

THE DEVICE IS AN EMBEDDED SYSTEM WITH NO MOVING PARTS

So what if the component in question is a more or less dedicated system? Chalk one up toward a step in the right direction. In many cases it is much easier to batten down the hatches on a product or system that is designed to do one thing in one particular environment, and that alone. There very well might not be all of the problems associated with a generic one-size-fits-all system. Then again, there is also the strong possibility that the embedded system was chosen simply for cost and in reality is just a generic system on the inside. Even if it is not a generic OS, did the vendor really take security seriously, or are there tell-tale signs that point to less than master-craftsman type work?

Here are a few of the things we have seen in "embedded" appliance devices:

- entire generic OS running on flash memory cards – not secured in the least
- poorly crafted and tested TCP/IP stacks on ASICs
- proprietary chips without tamper-resistant epoxy on them
- serial EEPROMs with programming leads exposed
- tamper-evident tape placed on the inside of the appliance where it is not visible

THE VENDOR OFFERS REMOTE SUPPORT

If you are lucky, the vendor knows one of the passwords of an account that you set up for him. More often, the vendor is aware of a hidden account that you were not told existed. While this is arguable, even if it is done for truly nonsecurity-related devices (what are those?), it should be a career-limiting move for the marketing or sales person that originally decided this was required to sell a security device. Does this still happen? Unfortunately so – the crypto-accelerator mentioned above contained a couple. We have also found them in printers, hubs, and plenty of software servers and clients. Of course, the remote support might be something more obvious such as a modem and analog line, or perhaps it was given away when customers asked for yet more holes to be placed in the firewall to allow them to get in for troubleshooting and diagnostic purposes.

Does this happen on your network? How strong is the stack on that VPN box? Let us rephrase – how strong is the stack on that VPN box that you deployed parallel to the firewall? Are the infrastructure components such as switches and load-balancers managed in-band or out-of-band? How many addressable devices are on your network and how many of them were able to be dropped on the network right out of the box and they basically configured themselves? Does that NTP server offer more than just the correct time? Are your hubs and switches addressable? Why?

Hopefully this article has caused some to think about their current environment and others to take a different look at the items they are about to deploy.

Sleep well.

[Editor's note: Peter Guttman's paper, An Open-Source Cryptographic Coprocessor, <http://www.usenix.org/publications/library/proceedings/sec2000/gutmann.html>, *makes an excellent companion to this article, with very concrete examples.]*