Cyber Security Protection Enters a New Era

By Jeff Melrose, Yokogawa

A software engineer is trying to complete a major block of code, but his boss cut out a large section including some open-source routines downloaded from the Internet. Replacing those routines will add days to the project. He runs to his boss' office and pleads: "I need to use that software in the system!"

"You can't use it. It's been compromised."

The engineer nods, having anticipated that reply. "Yes, it's open-source and came from the Web, but we've used it before. I also talked with the software engineers, and they will do a line-by-line review of the source and object code."

The boss looks up and glances at his award for years of service at an undisclosed location. "You can never be sure something isn't in there," he says.

Figure 1

That brief scene might sound like something from a suspense movie, but the situation could be very real given recent events in the cyber security community. Most think of software as something that does what it's supposed to most of the time and therefore sometimes neglect lurking danger.

Software engineers trying to write code for devices and industrial systems want to avoid re-inventing the wheel. If someone has already written code to do a certain job, and it works, they don't want to write it again. They'd rather save time by downloading freeware and open-source code off the Web. Or, they could pick up existing code from earlier products with a proven track record.

All of this gets cobbled together and loaded into a new device. As long as it does what it's supposed to, nobody needs to know or care where it came from.

This has been the working assumption for quite a while, but the landscape is changing. The cyber security world is becoming more confusing with nation-states, hacktivists, and cyber criminals making their presence known. Hackers and their efforts reflect a wide spectrum of skill levels. Some are clumsy and easy to spot. Others are more insidious and undetectable by all except the most sophisticated forensic cyber specialists.

While the engineer looking to streamline the project means well, his boss is correct: unsecure code can lurk within such software. Sometimes it can be found and removed, but a recent example of a cyber security breach proves that the threat can be well camouflaged.

The Password is "<<< %s(un='%s') = %u"

Those of us old enough can remember hearing the "Password" game show announcer whispering the key word for viewers at home. Nobody would guess this one, but it will become prevalent to the casual user because it is changing the threat landscape.

In December 2015, Ars Technica published a stunning report: "On Dec. 17 [2015], Juniper Networks issued an urgent security advisory about 'unauthorized code' found within the operating system (OS) used by some of the company's NetScreen firewalls and secure service gateway (SSG) appliances. A patch was issued to the affected device OS, and forensic investigation determined the unauthorized code acted as a backdoor into the device" (see Figure 1).

What makes this stunning is the way the password was hidden. Forensic investigations determined the administrator password used to evade normal authentication was "<<< %s(un='%s') = %u." Security researchers looking at this bit of gibberish might recognize that it was crafted to appear as debugging or test code within a software source code file. This suggests two conclusions:

The unauthorized backdoor was put there intentionally.

It was carefully designed to evade detection.

This is the beginning of a new era of cyber criminal threats. We are all used to the notion of attackers exploiting vulnerabilities caused by software flaws. It is a common tactic, and everyone is aware of it. Software patches are supposed to fix these flaws and address these vulnerabilities.

Now we seem to be moving into an era where vulnerabilities are built into software deliberately and then carefully hidden. Attackers aware of the hidden code's function can use such planted vulnerabilities when they like.

Naturally, some companies are taking this threat very seriously. Cisco, for example, undertook an effort to see if similar backdoors exist in its products and discovered that they do. Like Juniper, Cisco is developing patches to prevent breaches.

Fortinet has also acknowledged that a backdoor exists in a variety of its products. The hard-coded password has been characterized as a feature for remote management.

Other companies have not always been so quick to respond. Before Juniper, there was also RuggedCom, which included a backdoor in products with its Rugged Operating System. However, they did not inform purchasers of this. A user discovered it in 2011, but the company was reluctant to address the situation. This backdoor was also apparently installed deliberately.

Returning to the Juniper case, the purpose of the backdoor was apparently to gain access to the network device's configuration and its seed parameters for virtual private network (VPN) encryption routines. Juniper used a nonstandard set of parameters to initialize encryption, and the only way to obtain the encryption parameters was to gain administrator access. There has been much speculation as to who did this, but the "why" question is easy to answer. A backdoor's purpose is to create an entrance to a network.

A Door to the Network

Network device vendors are targeted in this manner because their products are entry points to networks. Access to a router or gateway provides entry to an industrial or enterprise system. Network device security thus often proves to be the soft underbelly of many organizations' defensive strategies. The value of such a backdoor secretly placed in a device, hidden with normal-looking code, is huge, and the larger implications are frightening.

Many organizations view their network devices simply as infrastructure; specifically, waypoints in their information distribution systems. The thought of information switches being accessed in an undetectable way is truly disturbing. The larger and more alarming message is that much of the last 20 years of network security best practices have now been rendered obsolete.

Best Practices Are No Longer Best

Why? Let's consider some examples of how this new network device threat will change security best practices:

Using network switches to implement virtual local area network (VLAN) separation between industrial control and business networks is no longer adequate. No organization can design networks with VLAN separation and expect them to be secure. If devices can be compromised at the administrative level, then any virtual separation cannot be guaranteed. It will be time to return to physical separation, creating huge communication problems.

Depending on VPN encryption as a magic bullet to protect confidentiality is no longer adequate. An organization will need to start looking at how deeply it depends on VPN techniques as their "go to" solution to move information on secured networks. A VPN tunnel is no longer safe across any network-particularly for long-distance communication within global organizations.

Assuming all is well with network device configuration isn't safe anymore. Many organizations follow a basic practice: if nobody touches a device, it has the same configuration it had before. That is no longer true. Companies will need to ramp up configuration control and auditing to account for the possibility of device configurations being changed by unauthorized means.

These are obvious security threats, and more will emerge as the full effect of this situation is realized. With the Pandora's Box of suspect code in networking devices now open, no one really knows how far the trail goes into rethinking cyber security. With this new reality in mind, there are some tips that end users, integrators, and device manufacturers, respectively, should follow.

What to Do: End Users

Review device patch status - Obviously, the first thing any organization should do is start on an organizationwide review of vulnerable and potentially vulnerable network devices. This should be done not only for Juniper

hardware, but also for all other network gear vendors. Begin by assuming devices from all vendors are similarly compromised.

Patch or attempt to patch all network devices - After the review, patches should be applied to all current devices, even in advance of approved patches for non-Juniper equipment. The reason for this is two-fold: to identify which devices are in particularly critical areas (such as industrial control systems) and to find those too old to be updated.

Create a risk matrix - the results of the first two steps will generate information, which can help define the attack surface (see Figure 2). This matrix should have two axes: The first is capability of patching, running from impossible due to age to easy thanks to cooperation from the vendor. The second is operational importance, running from high for critical 24/7 industrial networks to low for a small branch office switch. An unpatchable device in a critical operation should be replaced. Following this analysis will help your organization stay one step ahead of the inevitable disclosures of other network gear being compromised by these types of hacks.

Create a plan to change your attack surface vulnerabilities - The matrix should guide any efforts to lay out a patching plan. With all this information, security personnel can provide a burn-down list with percentage-based metrics showing the risks posed by new network gear vulnerabilities as they emerge. The matrix also provides a good plan of action if the worst happens: a new and exploitable network vulnerability is discovered with zero-day malware.

Increase network and configuration monitoring - If an organization is using Snort, Fox-IT already has IDS signatures to help detect this attack. An organizationwide effort should also be implemented to bring all network gear under configuration control. Periodic security audits should not only verify the configuration of network gear, but also assess the actual live network configuration by testing traffic patterns.

What to Do: Integrators Supplying Network Gear

Review lab device patch status and implementation guides - Integrators should patch any lab systems and then update their implementation guides to reflect the change in network gear configurations. For example, Juniper now has a code-signing step for firmware updates. Implementation personnel should be prepared as other network gear vendors are discovered to have similar issues.

Patch or attempt to patch all network devices - Integrators should patch all devices with their clients and end users as soon as possible. There also should be frank discussions explaining to clients how this is a new generation of threat, emphasizing the importance of preparing for more patching and monitoring over the long term. Characterizing this as another Stuxnet-type event is not an exaggeration.

Integrators can offer solutions and services to increase device monitoring - System integrators should lead the effort to inform customers of this new threat and craft solutions and services to ramp up monitoring of security and network devices. There is good reason to push customers to consider services aimed at determining network configuration control and implementation levels of network patches.

What to Do: Device Manufacturers (Including Industrial Control Devices)

Review development and implementation lab device patch status - A device manufacturer should patch any development and lab systems immediately. Security policies and procedures should be updated to reflect the change in network gear configurations. Controls should be tightened on devices and software migrated into development environments.

Re-examine development lab and development office architectures - Device manufacturers need to be more paranoid about how development networks are connected to other networks. Juniper will likely spend many resources finding out how the backdoor appeared in its device software. You can also be sure Juniper will be investing in more development configuration control and rethinking its development network security to include greater auditing and monitoring.

Also, users should re-examine software development policies and procedures, along with personnel vetting. Here are some basic guidelines to follow:

- Configuration control of all source code under development.
- Configuration control of software, including all personnel touching code.
- Segregated air-gapped networks for software development.

- Two-person integrity on updates to source code files.
- Critical files, such as encryption routines, authorization routines, and so on, should be under increased configuration control with auditing.
- Updated security policies-explicitly stating adherence to code quality, security, and integrity-should be mandatory for software engineers, testers, and quality personnel.
- Background investigations on software engineers, testers, and quality personnel.
- Implement source code classification according to critical security services, with critical software modules undergoing more security checks.
- Code reviews and testing for conformance to coding guides.
- Code reviews to verify implementation of separation of duties on code considered critical.
- Coding guides specifically spelling out how formats of certain functions are to be done to prevent obfuscation by bad actors.
- Software supply-chain reviews including regular checks on vendors via on-site inspections, audits, and conformance to the vendor's own security policies.
- Implement thorough review of all software moved into development networks, including all commercially packaged software.

Assessing Future Risks

Following this incident, device manufacturers will need to review software development risks far more closely. This is a turning point, similar to when Stuxnet showed the industrial world how control systems could be systematically attacked and compromised. Every device manufacturer should consider itself a target and perhaps already a victim of such an attack. Risk management is key. All outside software code sources must be re-evaluated top to bottom in this threat environment.

All manufacturers will suffer if these types of fundamental flaws are found in any supplier's code-especially if the company is unwilling to fix it or offer ways to mitigate the problem. Loss of trust related to such an incident is difficult to rebuild.

We should not try to minimize the implications of this incident. This breach is so clever and audacious it boggles the mind. All of us in the industrial automation business must view it as a cautionary tale and do everything possible to prevent new incidents, to ferret out problems yet undiscovered, and to proactively protect our customers from problems in devices already deployed. Everyone—end users, integrators, and device manufacturers alike—needs to rethink the fundamentals of how they approach software development. It's truly a new age for device cyber security.

YOKOGAWA

Co-innovating tomorrow®

Yokogawa Corporation of America
12530 W. Airport Blvd.,
Sugar Land, TX 77478

yokogawa.com/us

Yokogawa Canada, Inc.
Bay 4, 11133 40th Street SE,
Calgary, AB T2C 2Z4

yokogawa.com/ca

Yokogawa de Mexico, SA de CV
Urbina No. 18
Parque Industrial Naucalpan
Naucalpan de Juarez, Estado de México
C.P. 53370

yokogawa.com/mx