Whitelist Malware Defense for Embedded Control System Devices

Josh Powers and Rhett Smith
Schweitzer Engineering Laboratories, Inc.

Published in
Sensible Cybersecurity for Power Systems: A Collection of Technical Papers Representing Modern Solutions, 2018

Previously presented at
Saudi Arabia Smart Grid 2015, December 2015

Originally presented at the
Power and Energy Automation Conference, March 2015

Whitelist Malware Defense for Embedded Control System Devices

Josh Powers and Rhett Smith, *Schweitzer Engineering Laboratories, Inc.*

***Abstract*—Malware protection is a necessity for any electric device in modern critical infrastructure. We must all protect our critical cyber assets with antivirus as North American Electric Reliability Corporation (NERC) CIP-007 R4 states, but more broadly, we must protect our assets from malicious code infection regardless of whether they are identified as critical assets or not. Embedded devices and traditional personnel computer devices should be protected. The Stuxnet worm demonstrated that air gaps and unplugged devices are not immune from infection. We must engineer devices and systems to protect against the impact of malware.**

Traditionally, this protection was accomplished by using blacklist technology, where the technology watched for known bad code and blocked it. This resulted in a race to update malware protection technology when new threats were discovered, before infection happened. With malware statistics topping 83 million pieces of code, based on the August 2014 McAfee Labs Threats Report, and growing every day, the administrative task is impossible to keep up with. This design also can put excessive burden on processors, slowing computations and communications.

New malware protection technology is designed using a whitelist architecture that only allows known good code to execute on the device. This simplifies administrative overhead because new updates are not needed when new malware is released. A control system environment is built with application-specific devices that are set to accomplish one or more tasks and left alone to continue accomplishing the same tasks for many years, setting a perfect stage for whitelist malware protection technology.

This paper investigates the benefits that whitelist malware protection provides at the application layer (similar to existing anti-malware technology) and explains why embedded devices need architecture-specific malware protection. The paper shows that correctly combining malware protection and embedded architecture improves the reliability and cost of ownership of the whole system. The paper also highlights the enhanced security that whitelist malware protection provides over traditional solutions and how these principles apply to computers and embedded devices. The paper shows how whitelist malware protection meets and exceeds the NERC CIP requirements in Versions 3 and 5.

I. Introduction

Malicious software, or malware, is a tool often used to compromise the integrity of software or hardware. It is primarily used due to the power of automating the reconnaissance, infection, and compromise of a wide selection of targets. Simply put, malware can automate the exploitation of a system and do it much faster than one person.

Strict laptop computer usage policies and constant malware protection updates are protection methods that already exist in the electric sector. Malware trends have moved from targeting code flaws to enticing people to click links in emails or visit infected websites. Corporate infrastructure protection plans and technology are mature and established for malware protection. So how can we bridge the gap between corporate infrastructure protection plans and malware protection for control systems that consist of embedded, application-specific devices, many of which run on real-time operating systems? This changes the game completely because we are now talking about an infrastructure that is built for machine-to-machine (M2M) communications that have to meet high-reliability and availability requirements with very little downtime tolerance in control systems where physical consequences to cyber exploitation exist. Malware protection solutions have to support safe and reliable operations and work with the attributes of the system they are applied to. This leads us to the conclusion that the solutions designed to protect corporate systems are not a good fit for the control system due to the vast differences in their attributes.

Whitelist malware protection provided at the application layer is a viable solution for control systems. This paper discusses the enhanced security provided by whitelist malware protection compared with traditional malware solutions. It further discusses how the combination of malware protection and embedded architecture can improve the reliability and ownership cost of an entire control system. The paper also discusses the implications of whitelist malware protection on North American Electric Reliability Corporation Critical Infrastructure Protection (NERC CIP) requirements in Versions 3 and 5.

II. Increasing Malware Risks in Control Systems

Increasing demands on power systems today are creating more opportunities for malware infections. Smart grid is a term that has many definitions, but all of those definitions can be boiled down to advancements in control, measurement, and operations to automate new functions or previously manual ones. These advances have increased the number of electronic devices and the amount of code in the devices that make up power systems. They have also increased the communications links between all of those devices. These factors have increased the attack surface and the potential spread of malware by providing more targets, entry points, propagation paths, and potential vulnerabilities.

Based on the research results of McAfee Labs in their August 2014 quarterly threats report, malware is increasing on average 100 percent per year, and that trend is accelerating [1]. Malware developers have a lucrative market and are able

to sell their malware for Bitcoin or other currency. In comparison to the average $60,000 starting salary of a software engineer in the United States, and considering that there have been very few successful convictions on malware charges in the court system, illegal software engineering activities do not have strong enough deterrents to stop malware from being created. It is easy to see that ethics are the only thing stopping more people from making a career out of malware development. The writing of malware has gone from the curious and smart just wanting to see what they can do to organized criminals and nation-state actors with financial and political agendas offering advanced training and recruitment programs. Protecting power systems from these motivated and advanced sources is challenging, but we have the advantage when we design and engineer systems with protection capabilities that leverage the core attributes of the power system.

III. Power System Attributes Perfect for Cybersecurity

Power system networks are not like corporate information technology networks because they have a unique set of attributes that make information technology (IT) cybersecurity technology an imperfect fit. Building a cybersecurity program around these unique attributes provides the long-term stability and core foundations that we can use to advance cybersecurity to new levels. The control systems operating power systems are engineered with a specific purpose and are built to the highest levels of reliability. Each piece of technology, communications session, and data set is implemented for a reason. Every device used on the power system is carefully engineered and has a specific task. Each task is carefully programmed and then, in most cases, left alone to run for many years. This provides a baseline behavior for the device (how long it takes to respond, the amount of data served, what other devices talk to it, or what other devices it responds to).

Because the control system is built with M2M applications, baseline behaviors will not change unless the owner changes the services or devices on the system. These changes are rare in comparison to corporate IT infrastructures, so they can be managed with good change control policies and planned for in order to accept a new baseline. This level of understanding is the cybersecurity advantage. The best defense is to know the system, establish methods and means to monitor what is on the system, and react to undesired events. Instead of watching for bad code on the device, operators monitor and confirm that only approved devices and data are on the system. This provides the platform to protect against known and unknown malware. It also provides measurable success criteria for system uptime, reliability, and service provided, giving purpose to the engineers that operate the power systems on a daily basis. Baselining such as this results in metrics for asset management that inform operators what devices are approved on the system and that those devices are operating correctly. Communications outages are captured and unauthorized devices are logged. When systems are understood and monitored to this level, it is extremely difficult for attackers to hide their actions.

Power systems consist of many control and monitoring devices that are application-specific technology or embedded devices. These embedded devices have a variety of microprocessor architectures and operating systems. With a whitelist malware protection approach, we have the device operations and communications that can be monitored to confirm the system is doing only what is desired by the asset owners.

Even better, the operational and administrative management (OAM) costs are very low when technology applies safeguards in a whitelist architecture because it is locked in by the manufacturer. The only time the footprint changes is when a firmware upgrade is performed. There are no requirements for signature or patch updates as new malware is released. The devices are purpose-built, so the running of specific tasks and the communications are consistent. Specific protocols are enabled and turned on for a task and allowed to run continually for that task, enabling a communications baseline to be established. Control system devices on the power system measure the power at various distributed geographic locations, and any change in measurement will be seen by the operators or the automation schemes, triggering an event response action.

Based on the native attributes, the power system is perfect for some of the most advanced cybersecurity ever seen. Two simple protection methods contribute to this level of cybersecurity: whitelisting and deny-by-default. A whitelist approach is the method used to ensure that only desired devices, communications, and data are present on the system. The keys to its success are knowing what is on the system and knowing what each device is doing (this is the baseline). The deny-by-default method requires each device and communication to be off unless explicitly turned on for a purpose. In power systems, the advantage goes to engineers and operators when they know their system, establish a known good baseline, and have methods to ensure that this baseline is preserved.

IV. Power System Attributes Challenging for Cybersecurity

The foundations of the control system architecture enable the industry to advance cybersecurity to greater levels than corporate networks, but there are specific challenges we must address to get there. Power systems are built with many embedded control and measurement devices. Embedded devices are not open computer platforms that allow the end user to install new software. The software running on these devices is produced by the manufacturer, and steps are taken to ensure that no new software can be installed. This is good and bad. The good part is that malware is software trying to install itself on these devices, so the architecture is already safeguarding against this. The bad part is that the end user has limited visibility of what software is running on the device. This is important for patch management procedures. The end user now has to establish monitoring processes for the

manufacturers they have purchased products from and rely on these manufacturers to not only alert them when security vulnerabilities are discovered but release the mitigations in a timely manner.

Another challenge is the availability requirement for control systems. There is very little tolerance for downtime. Any reboot or decommissioning to take a product out of service for updates costs the company money and increases safety hazards. These updates need to be planned and tested well in advance, which will result in a slower deployment time between when security vulnerabilities are fixed and when they are deployed on the control system. The best mitigation to this is to select devices that accomplish the job they are intended to do with as small a code footprint as possible. These devices all work as a larger system, so many times when taking one device out of service for maintenance, the overall system suffers.

It is good that these systems have many channels for monitoring, and that operators watching the system understand event response plans. The challenge comes with change control. When changes are made, alarms and logs generated by these changes need to be expected or operators will waste time investigating them, or worse, will get comfortable seeing alarms and not respond. Most importantly, these systems are built for reliability and use redundancy to meet extreme reliability requirements. The contingencies to any change must be planned and well understood before the change is applied. This mandates that lab testing and system validation testing be performed and that engineering standard documents inform work instructions to prevent any undesired operations.

V. Malware Protection Architectures

There are four common means of protecting a system from malware that we look at in this paper: blacklisting, whitelisting, mandatory access control (MAC), and rootkit prevention.

A. Blacklisting

Blacklisting is the traditional approach used in corporate environments to protect computing resources. In this environment, systems change frequently to support corporate requirements. Blacklisting works well in these environments because updates are easily managed and automated. When new malware is detected, a signature is created and all of the clients receive the new signature. Blacklisting has a long history and has been shown to work reasonably well in many cases. The signatures are stored in large proprietary databases that are updated regularly with the newly detected signatures. Because new signatures are created regularly, a system with a recently updated signature database must scan all files and processes on the device in order to check for possible infections it did not previously know about.

B. Whitelisting

Whitelist anti-malware creates a signature for all of the allowed software on a system and assumes that the system will rarely change. This means that programs cannot be installed or modified without updates to the whitelist. Whitelist protection is fairly unexplored because it is difficult to manage in corporate environments. There are also two kinds of whitelist anti-malware. In some systems, the whitelist can be modified in the field by entering a password. This type of system is used in some corporate environments. The other way to use whitelist anti-malware is to cryptographically sign the files with a public and private key pair and keep the private key elsewhere. This type of system is more difficult to update because any updates have to be signed before they can be brought out into the field. It is more secure in the field because the secret protecting the whitelist security is not kept on the device being secured.

C. Mandatory Access Control

MAC has a lot of support in the open source community and has a strong backer in the National Security Agency (NSA). MAC works by segregating applications into separate domains of execution with very specific permissions granted to those domains. This is in contrast to discretionary access control (DAC), which is the default system used by most operating systems. Fig. 1 shows that with DAC systems, permission levels lower on the list have access to anything above them. Kernel can access root and root can access anything user specific. While the kernel still has access to anything in user space with MAC, each user space application is segregated into separate domains that all have limited, specifically granted permissions to each other. This narrows the scope of an exploit, limiting its reach to only the permissions the original domain had. Before, if the root layer was compromised, the entire user space was compromised.

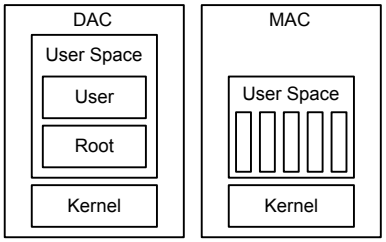

Fig. 1. DAC and MAC Protection Architectures

D. Rootkit Prevention

Rootkit prevention is the newest and most unexplored area of malware protection. It works by attempting to ensure that drivers and kernel modules come from a trusted source and, if not, preventing their use entirely. Drivers and kernel modules can both circumvent many other security measures because of their access to the kernel or operating system. Some rootkit prevention systems also attempt to verify that system calls have not been modified or interfered with. Adding a hook, a piece of code that runs when another function is called, to a system call is a common means of getting a rootkit into the kernel, and it is exceptionally difficult to detect.

VI. Evaluating Security of Malware Technology

Each of the previously described methods that are used to protect a system from malware has advantages and disadvantages.

A. Blacklisting Benefits and Drawbacks

Blacklisting is a reactive approach. It suffers from zero-day vulnerabilities because of this, but it does have the advantage of experience. It also benefits from the fact that specific attacks can be countered once a new signature is created. The biggest disadvantage, however, comes from the requirement to update the blacklist antivirus signature database on a regular basis in order to maintain its effectiveness. This is a poor design in an embedded system where updates are costly and infrequent. As mentioned previously, when a signature database is updated, a scan of the system must be performed to ensure that an infection that was previously undetectable is now detectable. The problem, though, is that embedded systems generally perform a specific task, often with real-time constraints. They are generally not engineered with occasional central processing unit (CPU) spikes and I/O-intensive disk scans in mind. This means that simply updating the signature database could degrade the ability of an embedded system to perform its main function for a period after the install. These limitations make blacklist anti-malware unsuitable for use in embedded systems.

B. Whitelisting Benefits and Drawbacks

Whitelist anti-malware maximizes safeguards while minimizing the administrative overhead in purpose-built M2M infrastructures. Whitelist malware protection is optimized for control systems instead of corporate information systems because of the reduced change management requirements on control systems. When there are frequent changes in what each devices does, whitelist malware protections become administratively burdensome. Each time an update is made to the system, an update must be made to the whitelist signature database as well. This is cost prohibitive in corporate environments, where updates to software are frequent. However, in an embedded system, updates to software are infrequent and it is not infeasible to include updates to the whitelist signature database when updates are made. In fact, it is often the case that updates to software in an embedded system are done via firmware image updates that can include the whitelist signature updates as well. The strong suit of whitelist anti-malware is that it does not require periodic updates to keep up with recent malware activity and does not suffer from zero-day exploits, except those against the whitelist anti-malware software itself.

Another weakness of whitelist anti-malware is that it cannot perform checks on running software. Once a piece of software has been loaded into memory, whitelist anti-malware can no longer say anything meaningful about its integrity. This means that a whitelist anti-malware solution cannot protect against malware that exploits things like buffer overflows, except to contain the exploit to only the running process that was exploited. Whitelist anti-malware does prevent an infection that has been persisted to disk from running.

C. MAC Benefits and Drawbacks

MAC takes a different approach to security than that of either whitelist or blacklist anti-malware. Instead of attempting to block the execution of a program, it attempts to constrain the reach of running software. All resources on a system are placed in a predefined domain and domains are then given specific access to other domains.

There are many MAC implementations, but the most common MAC systems are AppArmor and Security-Enhanced Linux (SELinux), both of which provide a similar result when properly configured. Individual executables are limited to the minimum set of permissions they need to do their job. This means that an exploited process will have limited reach and will be less likely to corrupt a system or prevent it from performing its primary function.

The downside to MAC is that it is very difficult to configure correctly, and mistakes in the configuration may not be detectable without a significant design effort. Fortunately, the effort of setting up MAC for an embedded system falls to the company creating the firmware, and they generally have the information required to correctly configure MAC. In the corporate environment, MAC is much more difficult to configure because small changes in the system can be difficult to adapt to in the MAC policy. Another problem with MAC is that it makes no attempt to verify the integrity of the process being placed into a domain. This means that an exploit that can modify the file system can persist its infection and perhaps spread by infecting other executables on the system, increasing its reach over time.

D. Rootkit Prevention Benefits and Drawbacks

Rootkit prevention provides yet another approach to securing a system. Generally speaking, operating systems are divided into two segments: the user space and the kernel space. User space tools are kept secure largely by software running in kernel space. Requests for access to all resources on a device go through the kernel, so it is the logical place to provide security. However, the kernel can be compromised, so a layer of security to attempt to detect these types of attacks, called rootkits, is needed.

Rootkit prevention is difficult at best because there is no other layer managing and monitoring the kernel, so the kernel must attempt to monitor itself. There are two common kinds of rootkit preventions. The first one attempts to verify that syscalls from user space to the kernel are not tampered with and the other attempts to verify that drivers and kernel modules that are loaded into the system are not malicious. The second kind is a form of whitelist anti-malware for drivers and kernel modules. Something similar is already implemented in Microsoft® Windows® systems, but not in Linux® systems, by default. The first type, however, is more difficult and is the subject of current research.

VII. Layering Malware Prevention Technologies

All of the malware prevention technologies we analyzed have strengths and weaknesses. A solution we identified to prevent the weaknesses from being exploited is to layer malware prevention technologies. As mentioned, a whitelist antivirus system cannot prevent runtime exploitation of things like buffer overflows. Layering on MAC to limit the scope of access that an exploited running application has is a good solution. MAC has no concept of integrity when placing a particular binary into a domain and granting it the permissions associated with that domain, so whitelist anti-malware should be layered on to prevent modified binaries from loading. Neither whitelisting nor MAC can detect a compromised kernel, so rootkit prevention technology should be added as another layer to mitigate these vulnerabilities.

By layering these technologies, we found that a secure system that is compatible with an embedded environment can be provided. This solution provides the best level of integrity of the software running on the embedded system, and the reach of attacks can be minimized. Also, the layered solution provides ample warning of attempted infections so additional measures can be taken outside of the embedded system. If an attack is detected against an embedded system, the network firewall can be hardened to stop that attack in particular and then forensic data can be gathered on the attack and the affected systems can be patched and updated.

VIII. Long-Term Administration

Long-term administration of an embedded system running blacklist anti-malware requires frequent signature updates. Regular updates must be pushed to the embedded system and regular scans must be made to ensure infections do not already exist that were not previously known about. Device burden is also a large problem. As previously mentioned, regular system scans must be performed. These scans create a large, irregular burden to disk I/O and to the CPU. Recent measurements show that up to 95 percent of the CPU processing power can be consumed during a scan.

Long-term administration of a system using whitelist antivirus depends on the environment. In a corporate environment, where software is updated very regularly, the administration of whitelist anti-malware would be time- and cost-prohibitive. However, in an embedded system, administration is minimized and only needs to be done when the embedded system itself is updated, which is generally not often. Additionally, the maintenance of the whitelist generally would fall to the firmware provider, therefore decreasing the required maintenance further. An embedded system running whitelist anti-malware should require no intervention between firmware updates for a device owner.

Another consideration in long-term administration is burden to the system. The whitelist anti-malware system we tested saw a 15 percent increase to system boot time but only had a 0.5 percent increase in the time to complete a task during runtime. The reason for a large impact to boot time but a smaller impact to general running time is that the whitelist anti-malware we tested uses cryptographic signatures to verify integrity but also caches integrity lookups. This means that at first boot, the system must run cryptographic analysis on every executable, but after an initial check has been done, the CPU burden decreases. In our tests, SELinux showed only a 0.5 percent increased burden overall. SELinux has no cryptographic security, so it adds little burden. The rootkit prevention software we tested, a variation of the program described in [2], showed an overall 5 percent increased runtime burden. All told, this provided a system with about 20 percent increased boot time and 6 percent increased runtime burden. Because all of these times are constant, they are easy to account for in an embedded system in contrast to blacklist anti-malware, which has inconsistent burden on a system.

IX. Configuration Management

Another interesting benefit of whitelist anti-malware is that it can be used to protect system configuration. Any embedded system will have configuration files that are not modified in the field. Things such as boot order and the disk to be mounted are set up in the firmware and never modified by the end user. By modifying the system binaries that use those configuration files and having them request integrity scans of their configuration files, the configuration integrity can be guaranteed. We call this voluntary scanning. Because the individual executables have had their integrity verified, it can be guaranteed that they will voluntarily scan their configuration. Then, all that must be done is to create signatures for the various unchanging configuration files on the system, and the same whitelist anti-malware that protects the system executables and libraries can be extended to protect configuration and scripts.

Fig. 2 shows the general flow of whitelist integrity verification and voluntary scanning. An executable is loaded into memory from disk at the request of another process or user. The whitelist anti-malware automatically scans the executable to verify its integrity before it is allowed to be placed into executable memory. Once the application has loaded, it then attempts to load its configuration files. Because it has been modified to include voluntary scanning by the whitelist anti-malware system, it first requests that the whitelist system scan the configuration file to verify its integrity. If the configuration file has integrity, the executable is notified and continues to load its configuration.

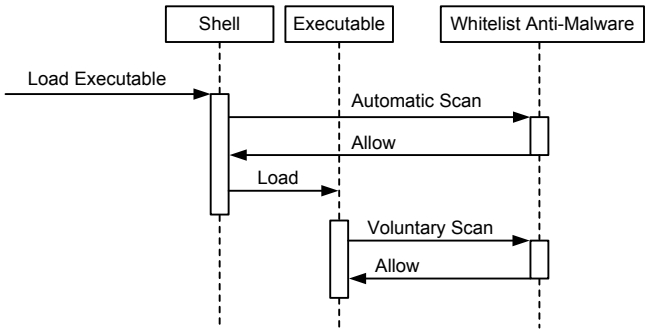

Fig. 2. Whitelist Integrity Verification and Voluntary Scanning

X. Compliance Considerations for NERC CIP Versions 3 and 5

NERC CIP from its inception recognized that devices need to have malware protections in place. The specific technology is not mandated, but the need for it is and the policies and procedures to keep the technology updated are mandated. NERC CIP Version 3 is specific to the device and states that every critical cyber asset (CCA) needs malware protection, and if the device cannot provide this, a technical feasibility exception (TFE) must be submitted. This was a huge generator for TFEs because many of these devices were embedded, so the end user could not install malware protection software.

NERC CIP Version 5 also requires malware protection and the procedures to keep it updated, but applies it to the system instead of the individual devices. This allows more freedom in the type of technology to select, and network-based technology can cover clients that do not have the capabilities to run malware protection.

For embedded devices, it is up to the manufacturer to provide the solution, either in the device or the system solution recommendations. When the whitelist malware protections discussed in this paper are implemented, the compliance to NERC CIP is accomplished. A small number of update procedures are required, keeping the operational costs low.

XI. Conclusion

Control systems are a very important area of focus for cybersecurity, and current malware protection technologies are not ideal in that environment. However, with the proper application of various anti-malware techniques such as whitelisting and deny-by-default, a control system can be reliably secured. This paper shows that control systems are an ideal candidate for cybersecurity.

Future research is still needed into rootkit protection to find ways to secure the kernel further and constrain exploits to the smallest area possible.

Control system owners reduce the total cost of ownership and improve cybersecurity by selecting technology from manufacturers investing in anti-malware solutions with whitelist architectures.

XII. References

[1] McAfee Labs, "McAfee Labs Threats Report," August 2014. Available: http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q2-2014.pdf.

[2] J. Reeves, A. Ramaswamy, M. Locasto, S. Bratus, and S. Smith, "Intrusion Detection for Resource-Constrained Embedded Control Systems in the Power Grid," *International Journal of Critical Infrastructure Protection,* Volume 5, Issue 2, July 2012, pp. 74–83.

XIII. Biographies

Josh Powers received his B.A. from Washington State University in 2008. He joined Schweitzer Engineering Laboratories, Inc. in 2010 as a software engineer and has focused on power system network security research. He has broad experience in the field of software engineering, but has focused largely on security and networking. He spent most of his time during college and shortly after working in public sector information technology, focusing on network security. Josh is currently working on his M.S. in computer science.

Rhett Smith is the development manager for the security solutions group at Schweitzer Engineering Laboratories, Inc. (SEL). In 2000, he received his B.S. degree in electronics engineering technology, graduating with honors. Before joining SEL, he was an application engineer with AKM Semiconductor. Rhett is a Certified Information Systems Security Professional (CISSP).