Telecommunications and Security Bundles at JMA Wireless: An Internship

CS4991 Capstone Report, 2023

Bronte Sundstrom
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
Bws8r@virginia.edu

ABSTRACT

JMA Wireless, a telecommunications company headquartered in Syracuse, NY, engages with every aspect of the cellular network industry needed a security bundle for their 5G software. I not only learned how 4G and 5G work on a software level and why virtualization is important to industry, but also received hands-on experience adapting the CIS Linux Benchmark into Salt Audits and Remediations for their security bundle. Visual Studio Code was used to SSH into the virtual Linux OS that hosted the Python and Bash files and Salt was used to run them all. The intended result of the project was a comprehensive bundle to keep the nodes updated to CIS standards. While unfinished by the end of the summer, this bundle project has since been completed and sent to QA for testing and will be used until the next Linux Benchmark is released.

1. INTRODUCTION

Living in a world of the internet and technology, virtual security is becoming increasingly important. Consequently, businesses need to put more resources in keeping their data secure. Companies that produce software need to make sure their products are secure against possible forms of attack. This can be accomplished by making sure their software is up to date according to agreed upon standards. This is no exception for JMA Wireless.

In order for their 4G and 5G nodes to be secure enough to be released to the public, they need to have their configurations set according to the Center for Internet Security's Linux Benchmark. These setting changes can be done individually, but when dealing with countless nodes, automating the changes is the most effective way of accomplishing this task.

Salt is an automated configuration management system that can be used as a template to insert, alter, or remove settings or text from files of the minion nodes or servers. This allows for quick changes to system configurations. The finished security bundle that we created consisted of a collection of all the Salt audits and remediations to align the nodes with the CIS Linux Benchmark.

2. RELATED WORK

As Verma (2022) notes, virtualization is just as susceptible to security breaches as physical computers, even though they provide an increased sense of security. Therefore, security for virtual computers is something companies still have to deal with and work towards. Verma continues to discuss vulnerabilities of virtualization, how those vulnerabilities can lead to attacks, and what architectures can be implemented in order to keep the computer secure. This article demonstrates the importance of security as well as suggesting security models, one of which, Terra, being similar to security bundles.

The US Department of Defense (n.d.) gives a short overview of how 5G works, but more importantly, emphasizes the need for security within the 5G nodes. They identify four attack attempts made by attackers and presents guidance on how to combat each of these. Within the subsection "Securely Configure Networking within the 5G Cloud," there is an in-text citation that references using the CIS Benchmark for Securing Kubernetes. This is very similar to the security bundle developed over the summer as we used the CIS Benchmarks as our basis. While the bundle I developed was not directly for Kubernetes, JMA does use Kubernetes for other parts of their 5G infrastructure.

3. PROJECT DESIGN

This project was a multi-step process that required learning about what we were doing and why, creating the audits and remediations that were not already created, and bundling all of the files together for distributed use by the company and for further testing by QA.

3.1 Background

Years before Jordan and I started working on this bundle, there was another bundle that was developed with an older CIS Benchmark. While there have been many audits and remediations that have been changed or added, there was a portion that was the same.

3.2 Developing Audits and Remediations

The first step was to find the audits that were already completed. To do this, we searched for the name or description of the audit with the "grep" command in Linux. If the audit number was correct, we would simply check if off, otherwise we would correct the number and move to the next one.

Once all of the old audits were accounted for, we then knew which ones we would have to write. Referring to the CIS Linux Benchmark, we adapted the given commands or bash files into Salt using modules. As many of the audits were simply checking for a specific string in a specific file, the "file.grep" module often sufficed.

After going through all the audits that had to be written, we once again started at the top and wrote all of the remediations that were unaccounted for. Looking at the CIS Benchmark, we first had to determine what configuration we would have to add, change, or remove. From there, we used Salt modules to alter the necessary files and get the configurations set correctly. A commonly used module for the remediations was "file.replace."

While some of the audits and remediations were fairly straightforward, some of them required writing bash scripts and using a Salt module to run the scripts while others required multiple different strings to be changed in several files.

3.3 Collaboration

Throughout this process we also had to add and commit all of our edits to a git repository so that both of us had access to all of the changes made by either of us. This made working on the project with a partner much more manageable.

3.4 Creating the Bundle

After all of the audits and remediations were complete, the final step was to bundle all the files together. With the help of our mentor, Vishal, and the use of a program available to JMA we completed this process fairly quickly without much hassle.

4. RESULTS

Testing was a large part of making sure each of the audits and remediations worked. And while each was tested as the development went along, the purpose of a bundle for automation is that it can run audits correctly and fix everything with one command at the

very beginning. Therefore, the last step of development was testing.

This multi-step process started by creating new minion node VMs with the default configurations. We then ran all of the audits on our master server and noted the number of tests that failed. The next step was to run all of the remediations to change the configurations of the node. The last step was to ensure that the remediations did their job by once again running the audits, noting the new number of failed tests, and comparing that value to the original one.

When the initial audits were run, there were 179 passes, 148 fails, and 1 skipped test, indicating a 54.6% pass rate. This shows that the default nodes are not very secure according to the CIS Benchmarks. However, after the remediations were run there were consistently 288 passes and 47 fails, indicating a 86.0% pass rate. The reason for a different number of total tests was that some would only run if others passed.

These results were very encouraging as this proved that the nodes were much more secure after our remediations. At one point, the remediations did pass more than 90% of the audits. However, when this occurred, the node became nearly useless as it was so secure that it was unable to function as a node. While there was no target pass rate going into the project, 86.0% was satisfactory.

5. CONCLUSION

This project is very important to JMA as they use off-the-shelf platforms that can open backdoors into their systems. This security bundle will make sure any unused backdoors are closed to mitigate security concerns.

This bundle was a project that JMA had wanted done for a while but had not found the time or labor to complete it. Our development project allowed Vishal to work on other tickets that needed to be completed that Jordan and I were not qualified to complete. Therefore, we estimate that we saved Vishal approximately 1 month of work time.

I was fortunate enough to intern with the Richmond office over the summer of 2022. I had the opportunity to learn about how the company was impacting the wireless market space, how 4G and 5G work on a software level, and why virtualization is important to industry.

My future in this line of work will include working for them full-time starting in the summer of 2023. While I will most likely not be working on the security bundle again, I will have plenty of experience about what the company does and how they operate, and I will be working with many of the same people.

6. FUTURE WORK

The next step for our security bundle will be to go through Quality Assurance at their Boulder office. At this stage, the bundle will be more rigorously tested and adjusted to make sure it meets the standard that is expected and required. Once it passes all of their testing, it will be added to the next version of the system that JMA uses. The estimated date of launch for that is currently unknown.

From this point on, the bundle will only have to be updated when newer versions of the CIS Linux Benchmark are released. Yet even then, it should be more of an update as opposed to another project. It would mostly likely be a similar process with renumbering and writing new tests, but if the company stays updated with it, each update would be much smaller.

REFERENCES

Verma, R., Rane, D., Jha, R. S., & Ibrahim, W. (2022). Next-generation optimization models and algorithms in cloud and fog computing virtualization security: The Present State and Future. *Scientific Programming*, *2022*, 1–10. https://doi.org/10.1155/2022/2419291

US Department of Defense. (n.d.) *Security guidance for 5G Cloud Infrastructures - U.S. Department of Defense*. (n.d.). Retrieved on Feb. 22, 2023 from https://media.defense.gov/2021/Oct/28/2002881720/-1/-1/0/SECURITY_GUIDANCE_FOR_5G_CLOUD_INFRASTRUCTURES_PART_I_20211028.PDF