Timothy E Robinson

5/19/2018

ICTN4040

Term Paper

X86 processor architecture vulnerabilities and Intrusion Detection and Prevention

Abstract

For quite some time now, I have been interested in the computer engineering. I particularly enjoyed processor design and implementation. Designing and building x86 processors is a long and complicated process; there are many variables that have to be taken into consideration. Furthermore, there are many different ways to design the x86, however in this paper I will mostly tend to security issues. It would be impossible to include every aspect of the architecture of the x86 within this paper. However, I will pay some attention to few different aspects of the architecture which I believe are relevant to attain a deeper understanding of some of the proposed security concerns. One of those aspects is referred to FFL, otherwise known as Fixed Function Logic. Moreover, as this paper is for a security course, I will dive into several other topics that will be listed below that pertain to security. I will discuss intrusion detection systems and machine learning, intrusion prevention, various vulnerabilities found in the x86, and various exploitations and patches. Also, I will discuss the difference between RISC and CISC. Moreover, there are also a few more main points that I would like tend to, first and foremost, I would like talk about, the buffer overflow vulnerability and the Solaris Vulnerability. I will also cover the subject of intrusion detection and intrusion prevention.

Furthermore, the first aspect of this subject I would like to delve a little deeper into is the basic architecture of the x86. There are roughly eight different main components to the x86. There is the bus

interface unit, which are two components merged into one. Then we have the instructive queue, the memory interface, the Arithmetic logic unit, the execution unit and the control unit. This then leads me into my next topic which is Fixed Function Logic. I would really like to look deeply at the subject I mentioned above, Fixed Function Logic. What is Fixed Function Logic? If you think of logic as it relates to computer science, there are several logic gates that are used in the design of x86 microprocessors. All of computations are done in the arithmetic logic unit. The logic gates are as follows: AND gate, OR gate, NAND gate, NOR gate, NOT gate, EOR gate, and the ENOR gate. According to the Glossary History of Corrections “Fixed Function logic refers to digital logic devices, such as a 7408 quad 2-input AND gate, whose operation cannot change” (Glossary 2001 Pg. 1). In other words, essentially fixed Function Logic means that the regardless of the inputs the output will always remain the same. So how does this all relate to security? Well, all of the logic computations are done in the arithmetic logic unit. The input queue and the memory unit contain the buffer. I mention the prior statement to allow the reader to understand a little better what is going on during a buffer overflow. This is an example of a exploit that I will go deeper into later.

Continuing on, now that we have discussed the main architecture of the x86. I would now like to talk about how to detect unwanted intruders. Why yes this is my next point, intrusion detection systems and machine learning. Within the broad realm of Information security, there exist many different subjects. One of which is intrusion detection. There are two basic different types of intrusion detection systems. The first is internal node sensors and external sensors. The internal node sensors can be placed anywhere within the network. An external sensor is placed as a software application on a host machine that has access to the internal network. As Tsai so eloquently states,” Security concerns are becoming increasingly important in modern computer systems. With the development of networking and interoperation on public networks, the number as well as the severity of security threats has significantly increased. In 2006, vulnerabilities were reported by Computer Emergency Response Team

(CERT) almost every hour (Fig, 1.1) [1]."(Tai 2010 Pg.1). As one can see, there is a need for security professionals within the information security field. Furthermore, what does it mean to be intruded on? Why do we need a system that can detect intrusions? What does intrusion mean in the context of the networking field? As Tsai so eloquently states "More systems deploy Intrusion Detection System (IDS) as another layer of security mechanism to protect the system. The term "intrusion" refers to attempts to compromise the confidentiality, integrity, availability of a resource, or to bypass the security mechanism of a computer or network system. An IDS tries to detect attempts to penetrate into a system by monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions (Page 1). So, yes we want a safe networking environment to work within. Furthermore, we want to be pre-emptive in placing the needed sensors within the network so as to catch the perpetrator early. Intrusion detection is the cutting edge of network security. There are pluses and minuses to host based detection and network based. As Howlett, so eloquently states, "Some of the advantages of a Host-Based Intrusion detection system are, fewer false positives, Activities rather than signatures are tracked, so you don't need constant signature updates. They are less prone to being tricked. They require less tending and tuning…Some of the disadvantages of a Host-Based Intrusion Detection System are; the network administrator has to load and manage software on every box. The alert comes after an attack has been successful; IDS sometimes provides earlier warning." (Howlett 2005). As you can see, there are some significant weaknesses to the host-based system. Many times, network intrusion detection systems are much more efficient and reliable.

Moreover, although it is true that we desperately need intrusion detections systems in our personal networking systems, our public and industrial networking structure. There is another subject that I would like to bring to your attention. The next subject is intrusion prevention, while some might think that this is the same thing as intrusion detection, they are wrong. Let's take a closer look at the difference between intrusion detection and intrusion prevention systems. Intrusion detection systems

can either be host based as in external or they can be network based as in internal. Intrusion prevention systems take the necessary action after the intrusion detection systems detect the intrusion. In other words, the intrusion prevention system helps to implement the proper networking tools to prevent the intrusion from happening in the first place. For means of definition IPA stands for Intrusion Prevention System. As Barkers so eloquently states" One negatives about IPS is that because it is inline, if the sensor fails and you do not have an alternate path in your network, the entire network could fail as a result"(Page 374). This is also a big difference between intrusion detection systems and intrusion prevention systems. Intrusion detection systems are not necessarily network dependent. So your next question may be what means does the network administrator tell whether the IPS or IDS detected a legitimate or an illegitimate threat? As Barker states, "A False negative is when there malicious traffic on the network and for whatever reason the IPS/IDS did not trigger an alert."(Barker 2013 Pg. 377). Preventing something like this is why we have human network administrators to watch for such events in their production network. As, Barker so eloquently states, "A true positive means that there was malicious traffic and that the sensor saw it and reported on it;" (Barker 2013 Pg. 377). This is the best case scenario. There is yet another option, As Barker so eloquently states, "A false positive is when the sensor generates an alert about traffic and that traffic is not malicious or important as related to the safety of the network "(Pg. 377). This case isn't that bad due to the fact the intruder did not actually get into to the system, however it causes more debugging manually by the network administrator.

Furthermore, given the importance of intrusion prevention systems, I would like to mention the Solaris vulnerability and how it affected the x86. According to the hacking context article, "Argus Systems (/www.argussystems.com) awarded a team of four hackers $48,000 for accessing a server protected with its PitBull intrusion-prevention software…Argus says hackers were able to create a hole in the Solaris/x86 OS kernel. No patch was is required for PitBull because the hackers found no vulnerability within its software, according to the company" (Hacking Article 2001 Pg. 1). Here is a real

world example of why it is so important to have an intrusion prevention system in your network. I concluded that the hackers were able to find a way around PitBull. So while there was not anything inherently wrong with PitBull, it did not completely protect the system the way it should have. The article was clear on how exactly the hackers were able to work around PitBull.

Moreover, as the Solaris Vulnerability proved the increasing need for intrusion prevention and detection systems within our networking world, I would like to mention the Buffer Overflow Vulnerability. As Padmanabhuni so eloquently states, “To mitigate buffer overflows, developers typically perform size checks and input validation. We propose static code attributes characterizing buffer usage and defense mechanisms implemented in the code for preventing buffer overflows”(pg. 450). Performing the given procedures above will help ensure that the targeted computer will not be compromised. There are probably several questions going through your head right now. One would be what a size check is, or what is input validation? And what does all of this have to do with computer security and keeping the hacker out of your system? Size validation has to do with validating the correct size of the input variable. Sometimes hackers will add code in a hidden spot where there was not supposed to be any data. Input validation is simply, are you who you say you are? Essentially the x86 performs regular checks on the input and size of variables so that the hacker can’t get in.

Furthermore, RISC stands for Reduced Instruction Set Computer. With this system the instruction set allows for far fewer cycles per instruction set and therefore a more efficient system. On the other hand, CISC stands for Complex Instruction Set Computer. With this system you are allowed to operate several low level operations, some of these may include, arithmetic operations or a memory store etc.

In conclusion, I have covered the following topics in this paper. I spoke about the x86 in terms of architecture and how Fixed Function Logic relates to the x86. I listed every single logic gate that is used

for the design of the x86. Furthermore, I spoke quite a bit about intrusion detection and prevention. Within the subject of intrusion detection I spoke about internal and external sensors and the differences between the two. I gave several real world examples of vulnerabilities and exploits, such as the Solaris Vulnerability. I explained what the vulnerability was and if there was a patch what that did as well. I also spoke RISC and CISC and how they differ from each other. I also spoke of the buffer overflow vulnerability and the details of what happened as a result of an exploitation of that vulnerability.

BIBLIOGRAPHY

ALTERA CORPORATION; patent issued for programmable device using fixed and configurable logic to implement floating-point rounding (USPTO 9348795). (2016,). Journal of Engineering

Seiler, L., Cavin, R., Espasa, R., Grochowski, E., Juan, T., Hanrahan, P., . . . Sugerman, J. (2008). Larrabee: A many-core x86 architecture for visual computing. ACM Transactions on Graphics, 27(3), 1.

"Glossary." History of Corrections, dev.studyguide.pls.pearsoncmg.com/pls/products/coco/digital_electronics_ap/1269148060/presentations/el_de_03_03_01.

Tsai, J. J. P., & Yu, Z. (2010). Intrusion detection : a machine learning approach. Retrieved from https://ebookcentral.proquest.com

Barker, Keith, and Scott Morris. CCNA Security 640-554 Official Cert Guide. Cisco Press, 2013.

Howlett, T. (2005). Open source security tools: Practical applications for security;. Upper Saddle River, NJ: Prentice Hall PTR.

Hacking contest shows solaris 7 x86 OS vulnerability. (2001). ServerWorld, 15(6), 17. Retrieved from http://search.proquest.com.jproxy.lib.ecu.edu/docview/224853023?accountid=10639

Liu, G., Wu, G., Zheng, T., Shuai, J., & Tang, Z. (2008). Vulnerability analysis for X86 executables using genetic algorithm and fuzzing. Paper presented at the , 2 491-497. doi:10.1109/ICCIT.2008.9

Padmanabhuni, B. M., & Tan, H. B. K. (2015). Buffer overflow vulnerability prediction from x86 executables using static analysis and machine learning. Paper presented at the , 2 450-459. doi:10.1109/COMPSAC.2015.78