

PROJECT REPORT

SQA INTERNSHIP @ NETSCOUT SYSTEMS, INC.

Shaun Munshi | SQA Intern | June 6th– August 9th (2019)

# Overview

Cybersecurity, in lieu of Modern terminology, has evolved into a field rather than a subdivision of programming. How our data is sent, ensured, protected, etc. has become of the utmost importance for not only highly classified government documents, but also for standard communication. It does not take a pessimist to see that the world of technology is constantly infected with malicious hackers. However, with the right mindset, in conjunction with the right information, threats can be reduced to a minimal state. In Cybersecurity, our way of ensuring security over data channels into 3 main topics, known as the CIA triad.

* C- Confidentiality: Luckily, this first word is perhaps the easiest to understand of the three. Confidentiality is simply the privacy of one’s data. For example, imagine a Hospital. After a child is born, the hospital must send the birth certificate to the government. The only people who should be able to see such a sensitive/confidential document are the parents, hospital, and government. It would be rather unwise if another person could tap the communications line and steal the confidential data. So, to protect data from being stolen by outside entities, encryption layers are often set up, protecting the data and thus making it *Confidential*.
* I- Integrity: To most, Integrity seems perplexing at first. However, after further in-depth reading, Integrity becomes exponentially easier to comprehend. Integrity is simply the component of verifying that the data that is sent over to the other communicator is unharmed in any way. For example, if a doctor from the hospital mentioned in the Confidentiality section needed to look up what ailment his next patient had, he would need the information to be right. If the medical information in the server says, “Heart Attack”, but the doctor gets the message as “Heat Exhaust”, he/she will administer the wrong treatment, potentially jeopardizing the patient’s life. Integrity takes care of this by verifying the message’s Hash: that is, the electronic signature of the packets, to make sure no communication is dropped or altered in any way.
* A: Availability: Finally, the CIA triad closes with availability. Perhaps the most important aspect of the three, this parameter plays a major role in daily cyber-messaging protocols and communication. In simple terms, Availability is the measure of how well one can access data: if the data is non-accessible when it should be viewed, the Availability field scores a giant zero. Let us revisit the hypothetical hospital again. This time take the example of a surgeon operating on a patient. If the patient has a sudden medical emergency during the surgery, and requires a precise treatment as soon as possible, the surgeon must access the medical database to retrieve any pertinent information. Now, if the surgeon can not access the database due to any reason, the patient’s life may end up being the payment. To prevent this from happening, cybersecurity experts place this category in the CIA triad to raise awareness.

Imagine yourself after a long day at work. All your tasks have been completed, and you check your email as a relaxing way to end the day. You see an email titled: “Your insurance policy is changing: Policy N0: XXXXXXX”. Because you see that the policy number is accurate, you click the link provided. The linked website reads, “Sorry, not available at this time”, so you give it no thought and close your computer. Little do you know that malware has just been deployed on your laptop.

Bad guys in cybersecurity are ever so prominent. Just with a simple click on one of their links in a phishing email, they can download malware on your computer and perform a variety of malicious intents such as password theft, webcam spying, and network hopping. What happens after this malware download is the center of my internship at Netscout.

After malware is downloaded, it will force the computer to make a connection with a malicious server tied to the original hacker. This server is run by that hacker, and is listening for any possible connections from various malware deployment. This results in a compromise in an organization’s communication. Consequently, evidence of such compromises are known as IOC’s, or Indicators of Compromise.

IOC detection is the main objection of my team at Netscout. One of Netscout’s core products, Arbor Threat Analytics(ATA), is the main device that analyzes how protected/malware-free a network of communications is. ATA sniffs any connection between the network and external entities, in hopes of detecting an IOC between the computer and a malicious entity. ATA’s IOC detection is split off into two additional categories: day zero vs known threats. My work in Netscout ATA regards known threats; specifically, the IOC simulation of a hacker connecting your computer to a malicious server.

ATA’s known threats rely on a certain parameter to help identify IOC connections: the Atlas Intelligence Field, more commonly known as AIF. The AIF comes in a .csv file, with over 49,000 malicious IP addresses and their respective ports to connect to. For my specific task, the collection of the AIF addresses was not nearly as important in comparison to the implementation of the AIF knowledge.

My specific task was to set up a simulation of the IOC communication, and test whether ATA would be able to detect this malicious communication. To do so, I set up a sample server and client machine. Both the server and client had access to the AIF.csv file containing all the malicious server and port configurations. Because the server in this case specifically was the bad guy, it was the one creating a socket on the IP address listed in the AIF sheet. The client, being the good guy who was injected with malware, connects to the same IP address as the bad guy, since both have the same exact aif.csv file. However, this AIF file was over 49,000 IP addresses long. So, both programs needed to know that after communication was established, they needed to move on to the next IP address in the AIF file.

The runtime of the program relied on connectivity and communication between the server and the client. To start the simulation, the server would create a socket on the first address listed on the AIF sheet. It would then listen for anyone who wanted to connect to the designated port on the AIF sheet. The client program would then start, connecting to the IP address on the first line of the AIF sheet, and sending a request on the same designated port as the server had. The server accepts this response, and sends back a response, stating that communication had been successfully established. The client then sends one last message, indicating that it received the server’s response. Then, the server unbinds from the first IP address and rebinds to the second IP Address and port. The client program takes the same steps as before, but this time with the second IP address instead of the first. This process keeps going for all 49,000+ addresses and ports.

Each step of this runtime bears significance in the real world. In my project, when the server binds to different IP addresses each time, it simulates how many servers can trigger an IOC warning because of their malicious intent. The client IP address does not change, since it is a single machine in a network. However, it keeps connecting to different IP addresses for server connection, because in the real world, the malware may cause many external malicious connections. Nevertheless, there is no AIF sheet in the real world, because the hacker will know exactly what IP address and port your computer, the client, should connect to.