# Abstract

Cyberattacks ranging from amateur hackers to nation state level adversaries on organizations both small and large have been consistently on the rise in recent years, and thus, has become a topic of major importance throughout many industries. Many businesses, both large and small, are not well-equipped to deal with such problems often restricted by budget set by management. The purpose of this paper is to explore free or low cost open-source solutions, such that businesses no matter the size can have a base network defense framework composed of tools and programs that they can modify to fit their goals and organization. By building upon research that was already done on this subject along with doing additional in-depth research done by myself, I have constructed a network defense framework that can be implemented in almost any business setting. I have tested my framework with myself acting as the sole defender of an organization. The framework itself proved itself to be of considerable value despite my lack of expertise as a network defender. Better organizations with more trained defenders will find my framework useful.

# Introduction

Most cybersecurity teams would like to dream of a world where they can have any tool of their choosing, as many people as they need, and management that grants them whatever resources they need to get their job donel. But due to budget, one or often more than one of these things is hampered. For large organizations with billions of dollars of capital, this is not a problem. However, budget is a major concern for most organizations no matter the industry, and is of especially high concern for those in the Information Security field. Thus, many individuals are charged with the dauntingly difficult task of having to monitor and defend their organization’s networks, often without the necessary tools that are critical for doing their job due to budget constraints. What I wish to do with the research that I have undertaken is to create a software stack that have free and open-source options such that they can implement this stack with budget **not** being a concern to upper management.

# Background Literature

## SYSMON

**What is Sysmon?**

John Strand of Black Hills Infosec once described sysmon as the “…heroin of information security tools (Implementing Sysmon)…” Microsoft System Monitor (Sysmon) is a windows service that is part of the Windows Sysinternals suite. Its purpose, once installed, is to monitor the host computer that it resides in. Sysmon keeps track of information such as process creation, network connections, and file creation/deletion through the use of ID’s (Markruss). Due to Sysmon’s innate ability to track these events that take place on a system, it will be the primary method of this software stack to collect host data from a system which will then be forwarded to the SIEM of choice, the ELK stack.

## ELK STACK

**What is the ELK stack?**

The ELK stack is made up of three open-source tools that synergize together.

### Elastisearch

*Elasticsearch is a distributed, open source search and analytics engine for all types of data, including textual, numerical, geospatial, structured, and unstructured*. Elastisearch indexes the data it is given from Logstash so that later on, defenders can query Elastisearch when hunting for adversaries(What Is Elasticsearch?).

### Logstash

LogStash is the heart and core of the ELK stack. Logstash is the part of the ELK stack that takes in the logs from various sources on a network, examines the data, parses it, identifies field names, and normalizes it into a common format such that Elastisearch can easily query such data(Logstash: Collect, Parse, Transform Logs).

### Kibana

Kibanacan be described as the “eyes” of the ELK stack. Kibana essentially provides data visualization for defenders, giving them a more aesthetic look at what is going on inside their network instead of just logs upon logs (What is Kibana?).

## SIGMA ANALYTICS

“Sigma is for log files what Snort is for network traffic and YARA is for files (Roth).” Sigma will be the “language” of choice that will be used in this framework. Sigma is a rule format designed from the ground up to be flexible so that can be used to create rules that can be easily converted into other rule formats for specific tools like our ELK stack. Sigma is useful due to the fact that if an organization wishes to swap one of these tools out, for example, ELK for Splunk, the Sigma rule can still be used provided that Sigma is compatible with the new tool.

## Atomic Red Team

Atomic Red Team will be the primary means of adversary emulation in this software stack. Atomic Red Team was created by the company Red Canary as a means for analysts to tests machines on their network. Atomic Red Team is a database of tests for Windows, Linux, and macOS machines (Redcanaryco). For this project, Windows will be the only operating system that is tested.

# Additional Tools/Configurations

## SwiftOnSecurity

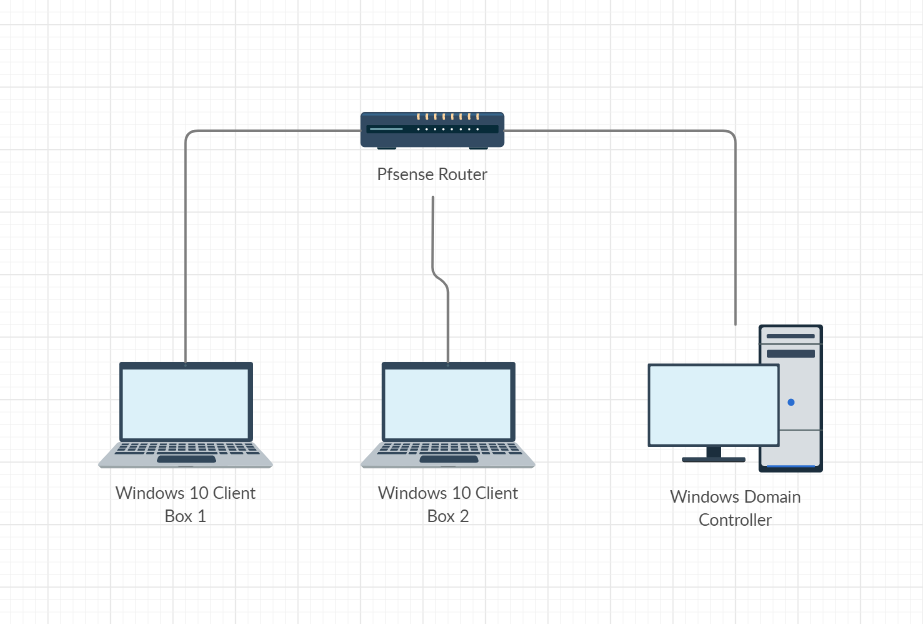
For Sysmon, a good general starting point for all businesses would be to use the SwiftOnSecurity configuration. Sysmon, by default, can and will generate many logs, most of which are unimportant to the organization. Through this configuration, a lot of the “noise” generated by Sysmon will not be logged (SwiftOnSecurity - Overview). Because this config is an all-around “vanilla” type of framework that can be used by different organizations throughout many industries as a starting point, it will be the configuration that is used in testing. If necessary, organizations can further enhance their defenses through the numerous on-going open-source Sysmon projects.  
WinLogBeat  
WinLogBeat is the chain between Sysmon and the ELK stack. It acts as a liaison that will port Sysmon event logs to Elastisearch or Logstash (Winlogbeat).

# Description of Research Process

The research for this project was fairly simple. Some of the tools that I researched are commonly known. When I found the tools, I looked through available Github repositories in order to find basic data about each of the tools. I then searched YouTube to find talks done by experts in order to find more information about each of the tools.

## The Network Setup

The network I have designed to imitate small businesses will be relatively simple. A pfsense machine will act as the router for the network which will act as the DHCP and DNS server. The machines that will be attacked through Atomic Red Team will all be Windows 10 machines, 2 machines being normal Windows 10 user machines while the third acts as Domain Controller. All of these machines will have the appropriate software installed and running during testing. DSU’s IAlab will be used as the means of creating this simulation network. The following is a diagram of what the network will look like.



## The Testing

For testing the efficacy of the software stack, I designed a test of one round of four of Atomic Red Team’s Threats. In order to prevent bias, I did not choose any threats. I asked one classmate to sift through all of Atomic Red Team’s repository of tests and choose four. The classmate then went onto the vApp, executed all the threats, and I will have to find all four. I will be given two hours to find all the threats and take screenshots to prove that I have found the threat. These screenshots will most often be in the form of logs. There are some limitations to what I can do, however. I am not allowed to remove Atomic Red Team itself. I am not allowed to use any tools other than the ones that are installed by default on Windows or in the software stack of this project. I can also create “queries” using only Sigma rules.

# Discussing the Study

## The Study’s Results

The results of this study were underwhelming. I was able to find only two of the four threats that I was assigned to hunt down. These threats were an adversary clearing system event logs and a malicious regsvr command trying to call out and download a malicious file from a URL. Screenshots of my finding these threats were submitted with this paper. There were two other threats that I did not find: a malicious scheduled task and the guest account being promoted to administrator and allowing remote sign-in. Even with all the tools at my disposal, I was not able to locate half of the threats. This test has led me to believe that the manner in which I had conducted this project was not done in a particularly scientific manner.

## Limitations of this study

While I did find some success with hunting down threats in my faux network, I now question the validity and application of how my trials were actually conducted? Is my network representative of the target demographic business? Was my familiarity of the tools comparable to even a junior level analyst? Is the scenario I have created even realistic? These are some of the questions I have contemplated after doing this study.

One of the more salient observations readers can make is that the network shown in the vApp is nowhere close to that of many medium or even small businesses. I did not do a thorough enough job in making a decision on *which* businesses specifically I wanted. Many small businesses will have more hardware with everything ranging from more PC’s to switches and routers. Many small businesses will even have their own custom apps that aren’t compatible with the latest operating environments. The Wannacry ransomware attack in 2017 targeted windows xp boxes as well, an operating system that was not in this testing ground (Barrett). *Who* is this research really for? I intended to help analysts in information security whose business was stringent on resources, but that is hardly narrowing it down. These are the reasons why I believe that the realism of my network may influence the outcome of how applicable my research is.

Going along with the realism of my network, I now ask whether or not the scenario I have set up is even realistic. Adversarial emulation is an important part of an organization’s preparation for a real incident. In this study, the participants were aware that an adversary would be in our network. We were ready. While analysts should always be ready to defend their network, adversaries, as long as they do not come in the form of a pentest, would almost always have an element of surprise. Real-life adversaries with years of experience would most likely be more skilled than an emulator like Atomic Red Team. Also, the classmate chose Atomic Red Team tests nearly randomly. I did not do my due diligence on what businesses I specifically wanted to emulate, and this spawned the problem of not knowing which threats I wanted to emulate. This is why I question whether or not the scenario that I have set up in this research paper is even realistic.

I also have to look at myself in assessing the real-life application of my research. While I have done my best to study and analyze the tools that were used in my scenario, I have not been *trained* to be a security analyst or a sysadmin defending a network. The fact that I might have missed some of the indications on the threats that I was testing may not even have to do with realism of the scenario or the tools themselves. It might just have to do with the fact that I am still very inexperienced with the tools, and that I do not have the level of training, knowledge, and experience, necessary to utilize these tools to their full extent. Hopefully, through the acknowledgment of these shortcomings, I will be able to continue and refine my research.

# Conclusion

This has been a very enlightening research topic that I do wish to develop more in the future. Through this research, I have been able to examine and investigate a potential software stack that can help many small businesses. The results have shown that while there are some threats that I or the analysts did not catch, there may be factors outside the tools themselves that may be part of if not the reason I failed to catch them. Even though I now see that my study was very, very flawed in the approach I took to explore and research this topic, I do wish to further refine my research. This project was a surface level venture into this topic. I would have changed many things about my experiment including everything from what businesses I should replicate, the actual attack, scenario, and perhaps even the tools themselves. I encourage those who read this paper to take into consideration the mistakes I have made, learn from them, and continue this line of study.

# Sources

Markruss. *Sysmon - Windows Sysinternals*. docs.microsoft.com/en-us/sysinternals/downloads/sysmon.

“Implementing Sysmon and Applocker.” *YouTube*, uploaded by Black Hills Information Security, 21 August 2019, <https://www.youtube.com/watch?v=9qsP5h033Qk>

“What Is Elasticsearch?” *Elastic*, www.elastic.co/what-is/elasticsearch.

<https://www.elastic.co/logstash>

“Logstash: Collect, Parse, Transform Logs.” *Elastic*, www.elastic.co/logstash.

“What Is Kibana?” *Elastic*, www.elastic.co/what-is/kibana.

“Winlogbeat: Analyze Windows Event Logs.” *Elastic*, www.elastic.co/beats/winlogbeat.

Roth, Florian. “Neo23x0/Sigma.” *GitHub*, github.com/Neo23x0/sigma.

Redcanaryco. “Redcanaryco/Atomic-Red-Team.” *GitHub*, github.com/redcanaryco/atomic-red-team.

“SwiftOnSecurity - Overview.” *GitHub*, github.com/SwiftOnSecurity.

Barrett, Brian. “If You Still Use Windows XP, Prepare For the Worst.” *Wired*, Conde Nast, 2 June 2017, www.wired.com/2017/05/still-use-windows-xp-prepare-worst/.