SIEMStack

By

Kyle Graham, Joseph Lazarte

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

Advisor: Bogdan Vykhovanyuk

Signature: Bogdan Vykhovanyuk Date: 4/20/2019

Team Member: Kyle Graham

Signature: Kyle Graham Date: 4/20/2019

Team Member: Joseph Lazarte

Signature: Joseph Lazarte Date: 4/20/2019

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2019

Table of Contents

Abstract:

SIEMStack is a full featured, ready to deploy, Open Source SIEM solution tailored towards small business, nonprofit, and educational users. SIEMStack comes with pre-packaged configurations and an installation experience that makes it a breeze for organizations to deploy SIEM. SIEMStack models its alerting on the MITRE ATT&CK framework, an industry standard set of TTP (Tactics, Techniques, and Procedures) used by cyber adversaries; both primitive and advanced. Our mission is to strengthen the security posture of our user's organizations and prevent the widespread damage caused by cyber breaches.

1. Problem Statement:

Small to mid-sized businesses, especially non-profits, are prime tar gets for hackers in today's age due to the increasing cost and complexity of securing infrastructure. In todays age small businesses account for sixty one percent of malware attackers on businesses. There is a need free, open source, and ready to deploy technology that can provide the administrators of these organizations actionable security intelligence within their environment

1.1. Problem Solution:

SIEMStack will create a fully functioning, ready to deploy SIEM utilizing the ELK (ElasticSearch, LogStash, Kibana) Stack for its backend. We will create premade dashboards that will provide actionable security intelligence from logs collected from devices. We will also create a wrapper around deploying forwarders to clients, as well as installing ELK to cloud providers. This solution will allow the IT Administrators of these organizations to better understand their security posture and take action against potential threats.

1.2 User Profile:

1.2.1 Potential Users*:* IT Administrators in the small to medium business sectors, non-profits, education, as well as home enthusiasts.

1.2.2 Required Experience:

SIEMStack aims to be very user friendly, although targeted towards those with previous IT experience. Users of this application should understand Microsoft Windows, Active Directory, as well as basic information security related concepts. Interpreting alerts from the system will require some independent research, we are hoping to add information to all alerts generated to explain why it was generated and how to remediate it.

1.2.3 Similar Applications:

- Splunk Enterprise Security
- Arcsight
- LogRhythm
- Radar

1.2.4 Task Experience:

- Navigating web applications
- Installing Windows Applications via Graphical User Interface
- Applying Group Policy Objects to Windows Domain

1.2.5 Frequency of Use:

SIEMStack is designed to be the IT Administrators daily security overview and as such will be frequently monitored by users. Users have the capability to search past data, so it can be used on as frequent of a base as required.

1.2.6 Key Interface Design Requirements that the Profile Suggests:

Ease of Use

Simplistic, Actionable Alerting

Easy Installation

Open Source Technologies

2. Figures and Diagrams:

2.1 Project Requirements

Figure 1:Project Requirements shows the requirements of our product. The information displayed is the requirement and a brief explanation on how the product addresses them. The last column is the priority of each requirements.

Requirement	Description	Priority
Ease of installation	Product will be able to be installed and configured without any previous knowledge	High
Actionable Intelligence	Product will generate low noise, actionable intelligence	High
Standard Configurations	Product will come with standard configurations for Windows and Linux	High
Scalable	Product will be scalable for organizations of small to medium size.	High
Customization	Product will allow for easy customization to increase/decrease noise of alerts	High

Figure 1 : Project Requirements

2.2 User Diagram

Figure 2: User Diagrams shows how each section of the project interacts with each other. Our project can be separated into three critical

components, the user interface, our application server, and our client agent. SIEMStacks goal is for the IT Administrator to interact solely with our user interface and avoid the trouble of command line installations and server maintenance.

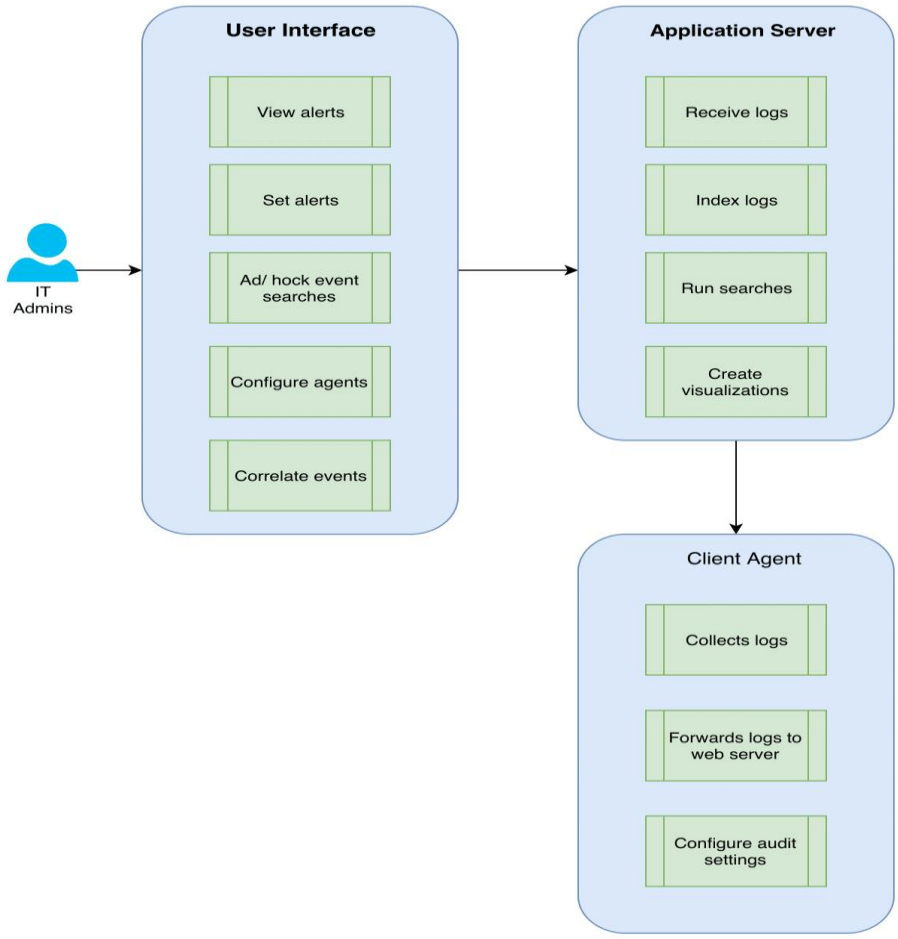


Figure 2 : User Diagram

2.3 WBS

Figure 3: WBS shows the timeline on how much time we want to spend at each point of the project. WBS is pending to change and acts as a rough representation of the work.

WBS	Start Date	Finish Date	Durations (in Work days)
Team Contract	9/20/2018	9/24/2018	5
Team Contract Revised with better WBS and Gantt Chart	10/10/2018	10/14/2018	5
Abstract	10/13/2018	10/15/2018	2
Finalize Project Direction	10/15/2018	10/20/2018	5
1st Team Meeting	10/20/2018	10/21/2018	2
Determine Relevant TTP (Tactics Techniques and Procedures)	10/22/2018	10/26/2018	5
Determine log sources required	10/29/2018	11/2/2018	5
Implement ELK STACK Test infrastructure	11/5/2018	11/9/2018	5
Configure log sources in log infrastructure	11/12/2018	11/16/2018	5
2nd Team Meeting	11/17/2018	11/17/2018	2
Implement TTPs to ELK STACK	11/19/2018	12/3/2018	15
Testing (UX/UI Testing and Fuctionability) after TTP deployment	12/3/2018	12/7/2018	5
Automate Server Install	12/10/2018	12/14/2018	5
Testing (UX/UI Testing and Fuctionability) after Server Install deployment	12/17/2018	12/21/2018	5
Automate Client Install	1/7/2019	1/11/2019	5
Testing (UX/UI Testing and Fuctionability) after Client Install deployment	1/14/2019	1/18/2019	5
UI/UX Tweaking and Development	1/21/2019	1/25/2019	5
3rd Team Meeting	1/28/2019	1/29/2019	2
Polish deliverables and start working on presentation	2/4/2019	2/8/2019	5
Presentation Prep for IT Expo	2/11/2019	3/1/2019	15

Figure 3: WBS

2.4 Gantt Chart

Figure 4: Gantt Chart shows the WBS in a more visual presentation. It shows how long each section is expected to last.

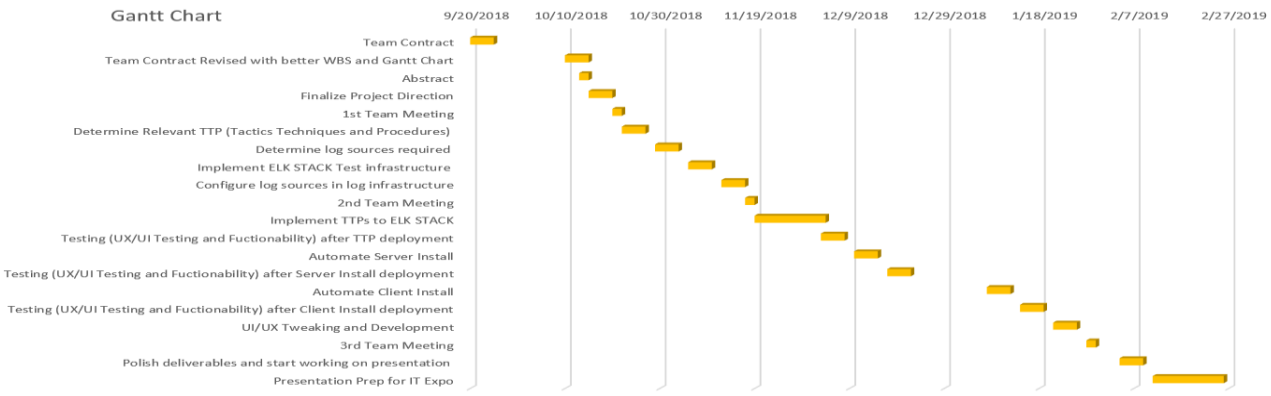


Figure 4: Gantt Chart

2.5 Figure 5: Budget

Figure 5 breaks down the cost of the project. Our budget is solely comprised of labor, all of the technology SIEMStack will use is fully open sourced at no cost.

Table 3: Project Budget

ITEM	UNITS	HOURS	COST	TOTAL
Labor	2	580	$20 per hour	$23,200
TOTAL				$23,200

3. Technical Design:

3.1 Hardware/Network:

SIEMStack is designed to ran off of a centralized, scaleable server environment which is a well documented standard within the ELK Community. We anticipate most organizations will running this application from a central server for all functions of the application and so we will use that model for our development and testing. We would also like to support deployments to AWS and other cloud providers if time permits.

3.2. Application:

We are utilizing open source technologies to power our SIEM, mainly the Elastic Stack (Elasticsearch, Logstash, Kibana, and Beats). ElasticSearch indexes our datasources and make them searchable, LogStash is for sending and receiving logs between the server and clients, and Kibana existing to provide visualizations of data. Using the Elastic Stack provides us with a powerful engine for searching through

vast amounts of logs that is battle tested and proven. To make it easier for System Administrators to setup the Elastic Stack we have packaged each component into a separate Docker Container so that it may reach desired state configuration on any platform.

The success of our project is also largely based upon the sources of data we utilize to find indicators of compromise. Windows Event Logs provide much of this ability including logon activity, object access auditing, and a plethora of other events. We are also utilizing Sysinternals Sysmon which is a device driver for Windows that creates rich endpoint logs of things like process creation, network connections, driver loads, file creation etc.

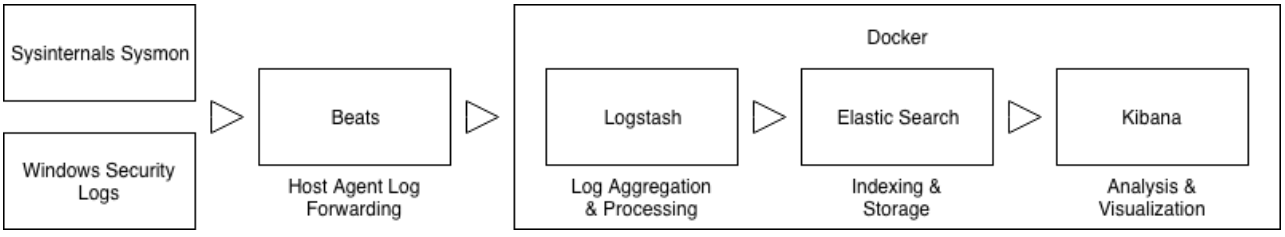
To make sure our project meets its goals of increasing the security posture of small businesses we used the MITRE ATT&CK Framework to analyze the current threat landscape and determine what techniques attackers were using. We identified techniques that would be detectable

Figure 5: Mitre Att&ck Techniques via Windows Event Logs and our Sysmon agent.

Initial Access	Execution	Persistence	Privilege Escalation
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features
Hardware Additions	Command-Line Interface	Account Manipulation	AppCert DLLs
Replication Through Removable Media	Compiled HTML File	AppCert DLLs	AppInit DLLs
Spearphishing Attachment	Control Panel Items	AppInit DLLs	Application Shimming
Spearphishing Link	Dynamic Data Exchange	Application Shimming	Bypass User Account Control
Spearphishing via Service	Execution through API	Authentication Package	DLL Search Order Hijacking
Supply Chain Compromise	Execution through Module Load	BITS Jobs	Dylib Hijacking
Trusted Relationship	Exploitation for Client Execution	Bootkit	Exploitation for Privilege Escalation
Valid Accounts	Graphical User Interface	Browser Extensions	Extra Window Memory Injection
	InstallUtil	Change Default File Association	File System Permissions Weakness
	Launchctl	Component Firmware	Hooking
	Local Job Scheduling	Component Object Model Hijacking	Image File Execution Options Injection
	LSASS Driver	Create Account	Launch Daemon
	Mshta	DLL Search Order Hijacking	New Service
	PowerShell	Dylib Hijacking	Path Interception
	Regsvcs/Regasm	External Remote Services	Plist Modification
	Regsvr32	File System Permissions Weakness	Port Monitors
	Rundll32	Hidden Files and Directories	Process Injection
	Scheduled Task	Hooking	Scheduled Task
	Scripting	Hypervisor	Service Registry Permissions Weakness
	Service Execution	Image File Execution Options Injection	Setuid and Setgid
	Signed Binary Proxy Execution	Kernel Modules and Extensions	SID-History Injection
	Signed Script Proxy Execution	Launch Agent	Startup Items
	Source	Launch Daemon	Sudo
	Space after Filename	Launchctl	Sudo Caching
	Third-party Software	LC_LOAD_DYLIB Addition	Valid Accounts
	Trap	Local Job Scheduling	Web Shell
	Trusted Developer Utilities	Login Item	
	User Execution	Logon Scripts	
	Windows Management Instrumentation	LSASS Driver	
	Windows Remote Management	Modify Existing Service	
	XSL Script Processing	Netsh Helper DLL	
		New Service	
		Office Application Startup	
		Path Interception	
		Plist Modification	
		Port Knocking	
		Port Monitors	
		Rc.common	
		Re-opened Applications	

3.3. <u>Architecture Diagram:</u>

Figure 6: Application Architecture Diagram



4. Testing Plan

4.1. Scope:

The scope of our testing is limited to SIEMStack running on Ubuntu 18.04 LTS and client-side testing will be conducted on a fully patched Windows 10 System.

4.2. Objective:

The objective of our testing is to ensure that malicious activity is properly identified with a low false negative rate, as well as testing to ensure high availability and fault tolerance considering the importance of the information we are handling.

4.3. Criteria:

Entry Criteria:

- Individual Dashboard complete
- Self-testing complete
- Changes merged to test server

Exit Criteria

- All tests are run

- All bugs/defects are documented and reported

4.4. Testing Procedures:

1. Functionality Test – All individual dashboards will be tested in phases to ensure that queries work as intended by performing test malicious actions on systems monitored by SIEMStack
2. Fault Tolerance Test - SIEMStack will be hard reset and tested to ensure that all services come back online in a clean state
3. Compatibility Test- This test will ensure that SIEMStack can run in a Docker container on supported architectures (64 Bit Windows, Linux, and MacOS)

4.5. Pass/Fail Conditions:

Functionality Test - All dashboards will have at maximum a false negative rate of 1/10

Fault Tolerance Test- SIEMStack must restart cleanly and ingest data within 5 minutes of successful boot.

Figure 7: Functionality Test

Trials	1	2	3	4	5	6	7	8	9	10
Pass		✓	✓	✓	✓	✓	✓	✓	✓	✓
Fail	X									

Figure 8: Fault Tolerance Test

Trials	1	2	3	4	5	6	7	8	9	10
Pass (Less than 5 minutes)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Fail (More than 5 minutes)										

5. Conclusion

Developing SIEMStack has been a challenge, but the ability to help secure small businesses is a reward up to par with the challenge. At the end of the project, managed to identify more attacker techniques. We also placed controls around and have implemented many of them. We implemented more visualizations techniques and controls, We ended up

increasing performance so it may run on all hardware, we however, did not end up adding authentication to the service. We realized that this solution is just out of our scope. While we would like to make this a complete solution there are some things we cannot do without limited scope and focus on ease of use and setup. We are proud of the product that we produced during these past two semesters and we certainly learned many things going into the future.

5. References

- "2017 State of Cybersecurity in Small & Medium-Sized Businesses." *Keeper Blog*, keepersecurity.com/2017-State-Cybersecurity-Small-Medium-Businesses-SMB.html.
- "MITRE ATT&CK™." *MITRE ATT&CK*, attack.mitre.org/.
- "ELK." *Elastic Blog*, www.elastic.co/elk-stack.
- Markruss. "Sysmon - Windows Sysinternals." *Microsoft Docs*, docs.microsoft.com/en-us/sysinternals/downloads/sysmon.