Practical Data Mining and Analysis for System Administration

[Extended Abstract]

Tanner Lund
BYU Infrastructure Lab
842 E 280 S
Orem UT 84606
tanner.lund@byu.edu

Hayden Panike
BYU Infrastructure Lab
256 N 600 W
Provo UT 84601
hpanike@gmail.com

Samuel Moses*
BYU Infrastructure Lab
381 N 600 W
Provo UT 84601
smoses45@gmail.com

ABSTRACT

Modern networks are both complex and important, requiring excellent and vigilant system administration. System administrators employ many tools to aid them in their work, but still security vulnerabilities, misconfigurations, and unanticipated device failures occur regularly. The constant and repetitive work put into fixing these problems wastes money, time, and effort. We have developed a system to greatly reduce this waste. By implementing a practical data mining infrastructure, we are able to analyze device data and logs as part of general administrative tasks. This allows us to track security risks and identify configuration problems far more quickly and efficiently than conventional systems could by themselves. This approach gives system administrators much more knowledge about and power over their systems, saving them resources and time.

The system is practical because it is more straightforward and easier to deploy than traditional data mining architectures. Generally, data analysis infrastructure is large, expensive, and used for other purposes than system administration. This has often kept administrators from applying the technology to analysis of their networks. But with our system, this problem can be overcome. We propose a lightweight, easily configurable solution that can be set up and maintained by the system administrators themselves, saving work hours and resources in the long run.

One advantage to using data mining is that we can exploit behavioral analysis to help answer questions about points of failure, analyze an extremely large number of device logs, and identify device failures before they happen. Indexing the logs and parsing out the information enables system administrators to query and search for specific items, narrowing down points of failure to resolve them faster. Consequently, network and system downtime is decreased.

In summary, we have found in our tests that the system increases security response time significantly. We have also found that the system identifies configuration problems that had gone on unnoticed for months or even years; problems that could be causing many other issues within the network. This system's ability to identify struggling devices by early warning signs before they go down has proven invaluable. We feel that the benefits of this system are great enough to make it worth implementing in most any professional computer network.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Theory, Design

Keywords

Elasticsearch, Logstash, Kibana, System Administration, Syslogs

1. INTRODUCTION

System Administrators are essential resources for every organization. System Administrations are responsible for a broad range of systems and services. They are busy behind the scenes maintaining and monitoring the infrastructure that runs all the computer systems of a company. They have numerous and broad range of responsibilities. They have so many tasks to complete that it is important to make sure they have tools to help them keep track of all the different systems.

System Administrators have a busy job. They determine the needs in the network and computer systems for the organization before it is purchased and set up. Administrators are responsible for installing all network hardware and software and maintain them with the needed upgrades and repairs. They are in charge of system integration, making sure all the network and computer systems are operating correctly with one another. It is their duty to collect data in order to evaluate the networkâĂŹs or systemâĂŹs performance to improve upon the systems to make them better and faster.

*He's actually Batman

They control the domain and are the primary person adding and removing users to have access to the network and systems. System Administrators automate monitoring systems to alert them when issues appear. They are tasked with securing the network, servers and systems under their responsibility. In their job they often train users on the proper use of the hardware and software that they are involved with as well [Bureau of Labor Statistics]. Some System Administrators are tasked with the day-to-day help desk problems that users run into as well, like login issues or desktop computers breaking. It is difficult for System Administrators to keep up with to keep up with all the systems.

With all these different and demanding responsibilities on System Administrators it is important for them to keep track of and know as much as they can about their systems. Administrators can spend a lot of time implementing different tools to try and understand their infrastructure, but there is a better way. By implementing a practical data mining infrastructure, System Administrators can reduce wasted time, and leverage existing infrastructure data and logs to easy many tasks.

2. PRACTICALITY

2.1 Background

The average Systems Administrator has more tasks a day, then he can accomplish in his limited amount of time. These task are usually prioritized by business needs and rate of completion. The more complex a task gets, the more time that must be expended on it instead of on other tasks. These task do not include the times when problems arise. As such a Systems Administrators quickly becomes friends with the data and logs inherent in the systems he maintains. Most of the time this means that an Administrator is moving from box to box collecting the data he finds relevant. If the number of critical box is too large to manage one at a time, then the Administrator may employ some sort of tool such as nagios or zabbix to help do the monitoring for him. Setting up these tools takes time and maintenance, and it adds overhead to the Administrators daily tasks. The often overlooked tool is to mine the data that each machine creates by default. Much like the Administrator who goes from box to box looking at individual logs, data mining can provide the same correlation without the wasted time.

To understand what we mean by correlation let us look at potential sources of data. All infrastructure environments include servers, workstations, and other network connected devices. Each type of device produces data, either in the form of logs, debug code, or simple authentication request. The data in these example are usually presented as a simple text message with the relevant information contained with. A server will produce events logs from individual programs, and from the the system itself. This data is kept locally on the machine, but it can be forwarded to alternate locations for storage or further processing. Once this data has been generated and moved it can be processed for help turn information into knowledge.

Using text based search, parsing, and retrieval an Administrator can process his data into the units that provide understanding. Since data is being processed in near real time the Administrator is able to compare differentiating source of incoming data all at the same time. He does not have to go individually to each machine to find information. Better yet the Administrator is able to gain a better understanding of how the action of one machine effect the entire ecosystem of his infrastructure. Equipped with a better understanding of his infrastructure as a whole an Administrator can spend more time preventing problems, rather than fixing problems.

3. ADVANTAGES OF DATA ANALYSIS

3.1 Analyze Large Amounts of Data

Data aggregation systems are a part of this model precisely because they can handle much larger data sets. Servers, workstations, and network devices generate orders of magnitude more data than system administrators generally see about them, even with IDS or monitoring tools. These data aggregation tools allow us to ingest and analyze much more than just SNMP traps. By collecting the log data of devices on the network and sending it to be processed and analyzed, we can exponentially increase what is knowing about the network.

Log data provides insights into almost anything a system administrator could care to know about. Every login, connection to a network, program error, overheating warning, and flapping port is reported in one way or another. The system we propose not only aggregates this data, but turns it into useful information and provides it to the system administrators in a useful format. It goes without saying that this approach can be extended beyond device logs to services and programs themselves, enabling application-layer analysis.

3.2 Easily Identify Device Failures

One great advantage of having so much information to pull from is that it gives administrators the power to do predictive analysis. An excellent example is that of device failures, as devices often show signs of problems well before they die. Take this example log message:

```
Error Message %ASA-3-210007:  LU allocate xlate failed
for type [ static | dynamic ]-[ NAT | PAT ] secondary(optio
translation from ingress interface name :  Real IP Ad-
dress / real port ( Mapped IP Address / Mapped Port
) toegress interface name :  Real IP Address / Real
Port ( Mapped IP Address / Mapped Port )
```

This message indicates a memory allocation failure, due to all the memory being allocated or due to memory corruption. Either way, that is a problem that should be investigated. There are hundreds of log messages like this one. They report memory errors, power failure errors, network and connection issues (like port flapping, or lots of dropped connections), and many other unfavorable changes. We have found through first-hand experience that without a proper log analysis engine, these logs often go unread and these issues unnoticed for months, sometimes even years. It would, of course, be impossible for administrators to manually read every log from every system on a daily basis. Log analysis can help catch these problems early.

3.3 Explicit Behavioral Analysis

This concept of behavioral analysis can be taken even further with respect to cybersecurity incidents. Certain log patterns are left behind by most types of cyber attacks, be they attempted root logins, a high amount of network traffic, or periodic mass-disconnects from access points. Some attacks are explicitly mentioned in the logs generated by network devices, and it is also possible to ship logs from an IDS into the system. However, even attacks that donâĂŹt leave such an obvious trail can be detected by those who understand the patterns they leave behind.

We have identified the patterns left behind by a number of different attacks. As we have done so, we have been able to modify our data and alert models in order to watch for these patterns to repeat. Frankly, there is enough information there for another paper, but we shall summarize our findings here.

Man-in-the-middle attacks can be identified by Xirrus access points. However, they also leave a suspicious trail of disconnections. A certain volume of simultaneous disconnections indicates either a problem with the device in question or due to a malicious forced disconnect. Either is a cause for concern. <more to come>

There is great potential for even more sophisticated behavioral analysis in this field. Machine learning algorithms could be implemented and tied in to configuration management systems like Puppet or SaltStack, which essentially is a self-managing network in embryo. The potential is exciting.

4. SYSTEM EXPLANATION

The core of this methodology is search or data mining technologies. There are many tools that provide the ability to analyze large amounts of data and produce usable information. Tools currently available include, Splunk, Apache Solar, and Elasticsearch FoundationsâĂŹ ELK stack (Elasticsearch, Logstash, and Kibana). Each of these tools has the ability to take textual data and parse it into desired fields and information. For the purpose of this paper we will be using the ELK stack as our software of choice.

Our practical data mining system includes data sources, a parser, storage, and an analyzer. Each piece serves a crucial part of an integral system. A piece of data enters the system from the log source where it is parsed and placed into storage. The analyzer then accesses the storage through API calls. This same basic concept is used by all data mining implementations.

Let us begin with our data sources. Data sources can be anything you want that produces text based events. Common log sources include servers, workstations, printers, and access points. In reality we produce data event in everything we do. Each time we browse a web page, purchase an item, or a click on an ad we produce an event. Things that we would normally not consider as a source of data often produces vast amounts data. HVAC units produce vast amount of useful data that can be leveraged to better control the climate of a server room. Once again anything that produces textual data can be potential source. Once we have identified our potential data sources we must transfer the data from that source into our parser so that we can glean useful information, and as such it is the parse we look next.

The parser in concept takes provided data and breaks it apart according the rules specified by the user. Different software accomplishes this in different manners. The ELK stacks uses the apache lucene language as its base for text based parsing and retrieval. The actual parsing of the text logs is accomplished in the Logstash portion of the stack. Logstash use grok to match patterns in text. Grok is open source natural language processing library that uses regular expressions saved in the form of variables. By using certain grok patterns we map them to the known format of data sources. Many formats have been pre written into the grok library, including general syslog, specific syslog formats, and many cisco formats. If the format has not yet been included in the grok library, you can custom write them using your own regular expressions. The parsed data then is passed into the storage database in the form of JSON. This object notation allows for easy searching of the broken out fields, and all the original data is retained with the JSON.

The storage portion of the ELK stack is provided by Elasticsearch. Here we see some of the real magic of this open source software. Elasticsearch uses indices in which the parsed JSON resides. The indices can grow large, so it is common to have a new index created everyday to control the their size. Once an index grows extremely large it because difficult to search it. For this reason we can use outside programs such as elasticsearch curator to control the number of indices that are opened and searchable. As with any storage system there will always arise a need to expand. To accomplish this elasticsearch breaks apart the storage into the rolls of master, backup master, data node, and balancer. The indices shared across multiple data nodes to provide a more fault tolerant platform. Data nodes can be added or removed with easy allowing for simple horizontal expansion. With the ability to quickly expand and adapt to changing needs, the elasticsearch storage provides great stability.

With a stable storage platform, we need something to access the data residing inside it. In the ELK stack Kibana provides the data analytics and correlation. Kibana is a web front end similar to Splunk. Kibana uses API calls to the elasticsearch storage to sift through parsed data using user created queries. Kibana queries utilize boolean logic to help string together complex searches. By stringing together multiple ideas over multiple data sources we can correlate event that may have escaped an Administrator's notice.

5. RESULTS AND FINDINGS

We have found, through our analysis, that system administration becomes much more efficient when a data mining infrastructure is implemented, even when networks have entire teams of engineers dedicated to their maintenance.

6. INCREASED SECURITY RESPONSE TIME

A number of attacks, including illegal VPNs and attempted internal hacks were identified and shut down quickly. Some of the security incidents detected would not have been caught at all without this system, despite a well-trained cybersecurity team. In fact, this team has been very happy with the log analysis system and has put it to good use.

6.1 Identified Configuration Problems

We identified memory errors, PoE failures, port flapping, ntp errors, over-taxed access points, unused access points, and more.

6.2 Identified Struggling Devices

Yes, we did.

6.3 Future Work

In the future, we hope to improve our data models and pattern-identifying methodology. We also hope to implement machine learning algorithms and configuration management systems to create a self-managing network, allowing system administrators to do higher level of work, thus taking network management to a higher level of abstraction.

7. CONCLUSIONS

Practicality, pros, and data prove that implementing it will help a SA in any professional computer network

We have concluded that there is more work to do before we are ready to submit this to a conference. This is really cool stuff though, which is why we're happy to geek-out about it instead of writing individual SysAdmin papers for this assignment (with Russel Havens' blessing).

8. ACKNOWLEDGMENTS

We'd like to thank Dr. Joseph Ekstrom for providing us the opportunity to work on this problem and get paid to do something we love. We'd also like to thank Professors Dale Rowe and Russel Havens for consulting with us on these ideas and for being secondary authors (we'll add you in when we figure out how to do that). Lastly, we are appreciative of BYU OIT and Citrix for funding the BYU Infrastructure Lab and by extension our research.

APPENDIX

.1 References

To be added in a later draft