Chapter 3: Creating A Repeatable, Reusable, and Reliable Security Data Analysis Workflow and Toolkit

While the majority of the content of this book focuses on how to analyze and visualize “security” data, there is a rhinoceros in the room1 that must be dealt with. If you aspire to become a security data scientist, it is not enough to just take snippets of code, apply them to your own data samples, produce a graph (or two) and declare victory. Just as the comic book hero Captain Marvel2 took on the powers of six mythical figures to save the day, you will need to be infused with the skills and abilities of a librarian, mechanic, systems administrator, programmer, project manager, and forensic pathologist to gain the most personal and organizational value from the concepts and techniques presented in these pages. Unfortunately, it will take a little more work than uttering “*Shazam!*”, but this chapter should help you down the path of acquiring those fundamentals skill with the added bonus of no lightning strikes.

You’ve Got Data!…Or, Do You?

It would be difficult to say that no data exists for us to process for the purposes of security-oriented analysis. Even the most nascent security analyst should be able to rattle off a list that would look similar to Table #. But, do *we* really *have* data? Network administrators are usually the owners and caretakers of their device configurations and logs. The same is also true for Windows and UNIX/Linux administrators. The security team may own firewalls, but if you’re a large organization, that may be a completely different team than the one that will perform analytics on the data.  Unless you own the data from generation to deletion, you will be relying on others to either provide it or provide access to it.

Table #.# Potential "Security" Data Sources

|  |
| --- |
| Windows Event Logs  Linux/UNIX syslogs  Mainframe Logs  Network Device Logs  Proxy Server Logs  Firewall Logs  Anti-malware Management Event Databases/Alerts  Vulnerability Management Databases  Patch Management Databases  System Configuration Logs  Identity & Access Management/RBAC Records  NetFlow Data  PCAP Data  HR Data Feed  Application Logs  Web Application Firewall Logs  Business Transactional Data Logs  Database Audit Logs  Asset Management Databases  Physical Security Event Logs  IDS/IDP Alerts  Indicators of Compromise  Help Desk/Non-security Incident Tickets  Risk Assessments  Penetration Testing Results  Application Security Scans  Firewall Port Requests |

Given the plethora and diversity of sources, your first and foremost task is to create and maintain a comprehensive catalog of them, *even if you’re not going to use them for analysis right now*. While some metadata will be unique, there are basic/common elements to record for each component:

**What is the generator of the data?** This could be as broad-based as an organization-wide proxy server farm or as specific as a line of business web application. Identifying the data source generator is important, though, especially as products and applications are retired or upgraded.

**What is the actual or potential security purpose of the data source?** The main thrust of this question is to determine how the data can or will be used for security-oriented analyses. While there is a definite school of thought that security practitioners should “log all the things”, this is truly not practical even in the age of “big data” and cheap storage. We’ve used the word “potential” on purpose since there is a huge difference in having a catalog of all data sources and actually using them all. Think if it in terms of an inter-library book loan. The book is in the catalog and you know you can get access to the resource when you need it, even if it isn’t at your local library at the moment. Knowing what and where the data is can save a great deal of time later on, especially if you’re in the middle of an incident.

**Who is the owner/custodian/controller of the data source?** Here, you should be recording the contact information of at least two people entrusted with care and feeding of the data. This is usually the application/service owner and it’s a good idea to go refresh your catalog on at least an annual basis to ensure you have up-to-date contact information. These records will come in handy when you have access or processing issues (and, you will).

**What steps need to be taken to gain access to the data source?** Most data sources require special permissions to gain access to them and many have special access methods. In some cases, this will involve pulling data via sftp, FTP, http[s], rsync, nfs or CIFS, either in real-time or at timed-intervals.  You may also need to make direct SQL calls to a myriad of databases or REST3ful/SOAP4-based queries to retrieve data from more modern sources or proprietary appliances. Finally, it’s almost a guarantee you’ll be either a Windows event log consumer or syslog consumer. Capturing solid details on the access method will be invaluable when it comes time to debugging why data mysteriously stopped flowing into your analytics engines.

**What format are the data elements encoded in?** In a way, your security analytics ingestion “hub” will be a digital Rosetta Stone5, knowing how to read and translate multiple representation formats into ones your engines can process. You can expect to be required to parse comma- and tab-separated (CSV/TSV) records, JavaScript Object Notation (JSON) objects, XML, Common Log Format (CLF) and a myriad of custom log formats. Recording this attribute will help you know when you need to add a new translator to your repertoire.

**What mechanisms are in place to validate the integrity of the data source and transmission processes?** We’ve hinted that your data acquisition setup will be less than perfect and the answer to this question should help provide an early warning system when gremlins decided to creep into your processes. If a source should be generating a mean of “n” events per second and that suddenly drops down a few standard deviations or—even worse—to “0”, you can fairly confidently assume that this is something you should investigate, especially if it’s a more vital data source (say, firewall logs or IDS alerts). Similarly, you may be expecting to consume a data source that has eight fields per event that now has more or fewer fields as a side effect of a vendor “upgrade”. Having a process in place to validate and notify of such integrity issues can help prevent lost time and visibility down the road.

**What are the record retention policies for the data source?** If you’re not in a large or highly regulated organization, issues such as data retention tend to come up only when storage space becomes scarce. However, in many companies there are strict policies on how long you must maintain access to certain types of data. You should check with both the data source owner and your legal/compliance department to determine what your responsibilities are as a consumer and processor of the data. Many times, a data source owner will believe they have transferred responsibility to the security department without explicitly stating so up front. Getting this confirmation can stave off future headaches and potential legal issues.

**What is the expected volume (*how much*) and velocity (*rate of consumption*) of the data?** You may say you want real-time access to your organization’s firewall logs but that access will do little good if your processing engines are not capable of digesting that fire hose of information. Thankfully, it’s fairly straightforward to measure and estimate the volume/velocity of the data flow and design an appropriate intake configuration. Skipping this step may end up with you only retrieving only one out of every four events and missing potentially critical data elements.

Notes

1. Joseph F. McDonald, “Russell, Wittgenstein, and the Problem of the Rhinoceros”, *The Southern Journal of Philosophy* (Volume 31, Issue 4, Winter 1993; pp.409-424)

2. http://en.wikipedia.org/wiki/Captain\_Marvel\_(DC\_Comics)

3. http://www.ibm.com/developerworks/webservices/library/ws-restful/

4. <http://www.ibm.com/developerworks/webservices/tutorials/ws-understand-web-services1/section2.html>

5. http://en.wikipedia.org/wiki/Rosetta\_Stone