Q RADAR TUTORIALS POINT pdf

1: QRadar supported DSMs

Radar and Radar with Markers show values relative to a center point. Radar with Markers shows with the markers for the individual points and Radar shows without the markers for the individual points. You can use the Radar and Radar with Marker charts when the categories are not directly comparable.

As all consultants say The elements I would factor in are: If not then Q-Radar may be a better fit. So if folks outside of security team will use the tool and subsequently help fund the endeavor this makes a strong case for Splunk. That being said, it does log management and analytics very well. So obviously if this is a pure play log management move, then Splunk becomes a strong choice here. Because the Splunk licensing model is based on the number of events being produced in your environment, then this is a factor that must be considered. Q-Radar on the other hand is one of most straight-forward SIEM installations, and shortest time to value out there. As such, they have often been associated with small to mid sized organizations. There are other factors out there to consider Q-Radar tends to focus its out-of-the-box reports on compliance reporting, as well as tracking behavior-based tracking that is arduous for the DIY script writer. Having used both, they are both great platforms that take quite a bit of training to fully understand and wring the most value. If you love scripting and going after known deviations, there are alot of Splunk consultants and expertise for hire. This makes Splunk slightly better for small organizations. If known deviations are "table stakes" and you focus is on exploring risks currently unknown to you For my business I was looking to build a shared environment that would service multiple customers so multi-tenancy, data security, roles based access controls and self-service-ability were key requirements. For the purposes of providing the SOC a single-pane of glass I needed a single configurable dash, and in a single-tenant environment both Splunk and QRadar could do it but in a multi-tenanted scenario on one could do it, at least without having to adding unnecessary systems. For me QRadar ticked all the boxes. Additionally the vast range of free apps which include user behavioral analytics are available which let you leverage its analytics engine. That said, Splunk is an effective analytics platform that has use cases outside of SecOps. You will need to have the depth of certified knowledge, expertise and deep pockets to make effective use. I am uncertain about Splunk reporting but I agree with Tim in that QRadar does offer extensive compliance-related reporting. I would also add that IBM has done a lot in the past few years to open up their API to partners to allow for a greater interoperability between multiple tools. So the integration between multiple tools is getting better. Unfortunately for us most of those tools are not used in our environment. It really depends on what are you expecting from the solution and how skilled and how ready for new approaches you are. Easy to use, super integrated with common devices, fantastic correlations and reports, everything expected from a top SIEM solution. Splunk, on the other hand, offers much more. Splunk is more a general purpose big-data collecting platform, used to search and find anything very quickly and correlate any data, without prior knowledge of the data structure. However, a newbie to Splunk might find the search-oriented GUI and more advanced approach than competitors rather confusing. Believe me, once you get used to the philosophy of Splunk and get familiar with its usage, there is no way back to any other tool.

Q RADAR TUTORIALS POINT pdf

2: QRadar SIEM (BYOL)

Python Pandas Tutorial PDF Version Quick Guide Resources Job Search Discussion Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

Flume Kafka When used in the right way and for the right use case, Kafka has unique attributes that make it a highly attractive option for data integration. Apache Kafka is creating a lot of buzz these days. While LinkedIn, where Kafka was founded, is the most well known user, there are many companies successfully using this technology. So now that the word is out, it seems the world wants to know: What does it do? Why does everyone want to use it? How is it better than existing solutions? Do the benefits justify replacing existing systems and infrastructure? Kafka is one of those systems that is very simple to describe at a high level, but has an incredible depth of technical detail when you dig deeper. The Kafka documentation does an excellent job of explaining the many design and implementation subtleties in the system, so we will not attempt to explain them all here. In summary, Kafka is a distributed publish-subscribe messaging system that is designed to be fast, scalable, and durable. Like many publish-subscribe messaging systems, Kafka maintains feeds of messages in topics. Producers write data to topics and consumers read from topics. Since Kafka is a distributed system, topics are partitioned and replicated across multiple nodes. Messages are simply byte arrays and the developers can use them to store any object in any format â€" with String, JSON, and Avro the most common. It is possible to attach a key to each message, in which case the producer guarantees that all messages with the same key will arrive to the same partition. When consuming from a topic, it is possible to configure a consumer group with multiple consumers. Each consumer in a consumer group will read messages from a unique subset of partitions in each topic they subscribe to, so each message is delivered to one consumer in the group, and all messages with the same key arrive at the same consumer. What makes Kafka unique is that Kafka treats each topic partition as a log an ordered set of messages. Each message in a partition is assigned a unique offset. Kafka does not attempt to track which messages were read by each consumer and only retain unread messages; rather, Kafka retains all messages for a set amount of time, and consumers are responsible to track their location in each log. Consequently, Kafka can support a large number of consumers and retain large amounts of data with very little overhead. Kafka at Work Suppose we are developing a massive multiplayer online game. In these games, players cooperate and compete with each other in a virtual world. Trades will be flagged if the trade amount is significantly larger than normal for the player and if the IP the player is logged in with is different than the IP used for the last 20 games. In addition to flagging trades in real-time, we also want to load the data to Apache Hadoop, where our data scientists can use it to train and test new algorithms. For the real-time event flagging, it will be best if we can reach the decision quickly based on data that is cached on the game server memory, at least for our most active players. Our system has multiple game servers and the data set that includes the last 20 logins and last 20 trades for each player can fit in the memory we have, if we partition it between our game servers. Our game servers have to perform two distinct roles: The first is to accept and propagate user actions and the second to process trade information in real time and flag suspicious events. To perform the second role effectively, we want the whole history of trade events for each user to reside in memory of a single server. This means we have to pass messages between the servers, since the server that accepts the user action may not have his trade history. Kafka has several features that make it a good fit for our requirements: We have configured Kafka with a single topic for logins and trades. The reason we need a single topic is to make sure that trades arrive to our system after we already have information about the login so we can make sure the gamer logged in from his usual IP. Kafka maintains order within a topic, but not between topics. When a user logs in or makes a trade, the accepting server immediately sends the event into Kafka. We send messages with the user id as the key, and the event as the value. This guarantees that all trades and logins from the same user arrive to the same Kafka partition. Each event processing server runs a Kafka consumer, each of which is configured to be part of the same groupâ€"this way, each server reads data from few Kafka partitions, and all the data about a particular user arrives to the

same event processing server which can be different from the accepting server. Keep in mind that Kafka was mostly tested with fewer than 10, partitions for all the topics in the cluster in total, and therefore we do not attempt to create a partition per user. This may sound like a circuitous way to handle an event: Send it from the game server to Kafka, read it from another game server and only then process it. However, this design decouples the two roles and allows us to manage capacity for each role as required. In addition, the approach does not add significantly to the timeline as Kafka is designed for high throughput and low latency; even a small three-node cluster can process close to a million events per second with an average latency of 3ms. When the server flags an event as suspicious, it sends the flagged event into a new Kafka topicâ€"for example, Alertsâ€"where alert servers and dashboards pick it up. Meanwhile, a separate process reads data from the Events and Alerts topics and writes them to Hadoop for further analysis. Because Kafka does not track acknowledgements and messages per consumer it can handle many thousands of consumers with very little performance impact. Kafka even handles batch consumersâ€"processes that wake up once an hour to consume all new messages from a queueâ€"without affecting system throughput or latency. Additional Use Cases As this simple example demonstrates, Kafka works well as a traditional message broker as well as a method of ingesting events into Hadoop. Here are some other common uses for Kafka: The web application sends events such as page views and searches Kafka, where they become available for real-time processing, dashboards and offline analytics in Hadoop Operational metrics: Alerting and reporting on operational metrics. One particularly fun example is having Kafka producers and consumers occasionally publish their message counts to a special Kafka topic; a service can be used to compare counts and alert if data loss occurs. Kafka can be used across an organization to collect logs from multiple services and make them available in standard format to multiple consumers, including Hadoop and Apache Solr. A framework such as Spark Streaming reads data from a topic, processes it and writes processed data to a new topic where it becomes available for users and applications. Other systems serve many of those use cases, but none of them do them all. First, it is interesting to note that Kafka started out as a way to make data ingest to Hadoop easier. When there are multiple data sources and destinations involved, writing a separate data pipeline for each source and destination pairing quickly evolves to an unmaintainable mess. Kafka helped LinkedIn standardize the data pipelines and allowed getting data out of each system once and into each system once, significantly reducing the pipeline complexity and cost of operation. My own involvement in this started around after we had shipped our key-value store. My next project was to try to get a working Hadoop setup going, and move some of our recommendation processes there. Having little experience in this area, we naturally budgeted a few weeks for getting data in and out, and the rest of our time for implementing fancy prediction algorithms. So began a long slog. Diffs versus Flume There is significant overlap in the functions of Flume and Kafka. Here are some considerations when evaluating the two systems. Kafka is very much a general-purpose system. You can have many producers and many consumers sharing multiple topics. As a result, Cloudera recommends using Kafka if the data will be consumed by multiple applications, and Flume if the data is designated for Hadoop. Those of you familiar with Flume know that Flume has many built-in sources and sinks. Kafka, however, has a significantly smaller producer and consumer ecosystem, and it is not well supported by the Kafka community. Hopefully this situation will improve in the future, but for now: Use Kafka if you are prepared to code your own producers and consumers. Use Flume if the existing Flume sources and sinks match your requirements and you prefer a system that can be set up without any development. Flume can process data in-flight using interceptors. These can be very useful for data masking or filtering. Kafka requires an external stream processing system for that. Both Kafka and Flume are reliable systems that with proper configuration can guarantee zero data loss. However, Flume does not replicate events. As a result, even when using the reliable file channel, if a node with Flume agent crashes, you will lose access to the events in the channel until you recover the disks. Use Kafka if you need an ingest pipeline with very high availability. Flume and Kafka can work quite well together. If your design requires streaming data from Kafka to Hadoop, using a Flume agent with Kafka source to read the data makes sense: Conclusion As you can see, Kafka has a unique design that makes it very useful for solving a wide range of architectural challenges. It is important to make sure you use the right approach for your use case and use it correctly to ensure high throughput, low latency, high

availability, and no loss of data. Jeff Holoman is a Systems Engineer at Cloudera.

3: Apache Kafka for Beginners - Cloudera Engineering Blog

IBMÂ® Security QRadar SIEM is a tech platform developed by IBM to provide a degree overview of an organization's security system. The platform can detect security offenses report them. QRadar normalizes events that come from a security system's log sources and correlates them according to.

Tutorials Introduction Quadrature signals are based on the notion of complex numbers and perhaps no other topic causes more heartache for newcomers to DSP than these numbers and their strange terminology of j operator, complex, imaginary, real, and orthogonal. Quadrature signal processing is used in many fields of science and engineering, and quadrature signals are necessary to describe the processing and implementation that takes place in modern digital communications systems. Next we examine the notion of negative frequency as it relates to quadrature signal algebraic notation, and learn to speak the language of quadrature processing. This tutorial concludes with a brief look at how a quadrature signal can be generated by means of quadrature-sampling. Why Care About Quadrature Signals? Quadrature signal formats, also called complex signals, are used in many digital signal processing applications such as: These applications fall in the general category known as quadrature processing, and they provide additional processing power through the coherent measurement of the phase of sinusoidal signals. A quadrature signal is a two-dimensional signal whose value at some instant in time can be specified by a single complex number having two parts; what we call the real part and the imaginary part. The words real and imaginary, although traditional, are unfortunate because of their meanings in our every day speech. Communications engineers use the terms in-phase and quadrature phase. More on that later. The Development and Notation of Complex Numbers To establish our terminology, we define a real number to be those numbers we use in every day life, like a voltage, a temperature on the Fahrenheit scale, or the balance of your checking account. These one-dimensional numbers can be either positive or negative as depicted in Figure 1 a. In that figure we show a one-dimensional axis and say that a single real number can be represented by a point on that axis. An graphical interpretation of a real number and a complex number. However, complex numbers are not restricted to lie on a one-dimensional line, but can reside anywhere on a two-dimensional plane. That plane is called the complex plane some mathematicians like to call it an Argand diagram , and it enables us to represent complex numbers having both real and imaginary parts. Think of those real and imaginary axes exactly as you think of the East-West and North-South directions on a road map. Taking a look at Figure 2, we can use the trigonometry of right triangles to define several different ways of representing the complex number c. Our complex number c is represented in a number of different ways in the literature, such as: The magnitude of c, sometimes called the modulus of c, is [Trivia question: In what movie, considered by many to be the greatest movie ever made, did a main character attempt to quote Eq. The suspicious reader should now be asking, "Why is it valid to represent a complex number using that strange expression of the base of the natural logarithms, e, raised to an imaginary power? That substitution is shown on the second line. Next we evaluate the higher orders of j to arrive at the series in the third line in the figure. Those of you with elevated math skills like Euler or those who check some math reference book will recognize that the alternating terms in the third line are the series expansion definitions of the cosine and sine functions. Figure 3 verifies Eq. The polar form of Eqs. Stated in words, we say that j represents a number when multiplied by itself results in a negative one. Well, this definition causes difficulty for the beginner because we all know that any number multiplied by itself always results in a positive number. Unfortunately DSP textbooks often define the symbol j and then, with justified haste, swiftly carry on with all the ways that the j operator can be used to analyze sinusoidal signals. Readers soon forget about the question: Euler, going beyond the province of real numbers, showed that complex numbers had a clean consistent relationship to the well-known real trigonometric functions of sines and cosines. As Einstein showed the equivalence of mass and energy, Euler showed the equivalence of real sines and cosines to complex numbers. For our purposes, the j operator means rotate a complex number by 90o counterclockwise. For you good folk in the UK, counterclockwise means anti-clockwise. What happens to the real number 8 when you start multiplying it by j. Multiplying any number on the real axis by j results in an imaginary product that lies on

the imaginary axis. Multiplying -8 by j results in a further 90o rotation giving the -j8 lying on the negative imaginary axis. Whenever any number represented by a dot is multiplied by j, the result is a counterclockwise rotation of 90o. Conversely, multiplication by -j results in a clockwise rotation of o on the complex plane. Granted, not only is the mathematics of complex numbers a bit strange at first, but the terminology is almost bizarre. While the term imaginary is an unfortunate one to use, the term complex is downright weird. This is regrettable because the concept of complex numbers is not really all that complicated. Just know that the purpose of the above mathematical rigmarole was to validate Eqs. Consider a number whose magnitude is one, and whose phase angle increases with time. Figure 5 a shows the number, represented by the black dot, frozen at some arbitrary instant in time. A snapshot, in time, of two complex numbers whose exponents change with time. They each have quadrature real and imaginary parts, and they are both functions of time. Return to Figure 5 b and ask yourself: Implementations of modern-day digital communications systems are based on this property! A cosine represented by the sum of two rotating complex phasors. We could have derived this identity by solving Eqs. Similarly, we could go through that same algebra exercise and show that a real sine wave is also the sum of two complex exponentials as Look at Eqs. They are the standard expressions for a cosine wave and a sine wave, using complex notation, seen throughout the literature of quadrature communications systems. Those two equations, along with Eqs. We can now easily translate, back and forth, between real sinusoids and complex exponentials. That way none of the phase relationships of our quadrature signals will be hidden from view. Figure 8 tells us the rules for representing complex exponentials in the frequency domain. Interpretation of complex exponentials. With all that said, take a look at Figure 9. Complex frequency domain representation of a cosine wave and sine wave. See how a real cosine wave and a real sine wave are depicted in our complex frequency domain representation on the right side of Figure 9. The directions in which the spectral impulses are pointing merely indicate the relative phases of the spectral components. Figure 10 is a straightforward example of how we use the complex frequency domain. There we begin with a real sine wave, apply the j operator to it, and then add the result to a real cosine wave of the same frequency. This figure shows the big payoff: If you understand the notation and operations in Figure 10, pat yourself on the back because you know a great deal about the nature and mathematics of quadrature signals. Bandpass Quadrature Signals In the Frequency Domain In quadrature processing, by convention, the real part of the spectrum is called the in-phase component and the imaginary part of the spectrum is called the quadrature component. The signals whose complex spectra are in Figure 11 a , b , and c are real, and in the time domain they can be represented by amplitude values that have nonzero real parts and zero-valued imaginary parts. Real signals always have positive and negative frequency spectral components. For any real signal, the positive and negative frequency components of its in-phase real spectrum always have even symmetry around the zero-frequency point. Conversely, the positive and negative frequency components of its quadrature imaginary spectrum are always negatives of each other. This means that the phase angle of any given positive quadrature frequency component is the negative of the phase angle of the corresponding quadrature negative frequency component as shown by the thin solid arrows in Figure 11 a. Quadrature representation of signals: The directions in which the impulses are pointing show the relative phases of the spectral components. As for the positive-frequency only spectrum in Figure 11 d , this is the spectrum of a complex-valued analog time-domain bandpass signal. And that signal does not exhibit spectral symmetry centered around zero Hz, as do real-valued time-domain signals, because it has no negative-frequency spectral energy. Quadrature mixing of a signal: Quadrature-sampling is the process of digitizing a continuous analog bandpass signal and translating its spectrum to be centered at zero Hz. That arrangement of two sinusoidal oscillators, with their relative 90o phase difference, is often called a quadrature-oscillator. The Figure shows how we get the filtered continuous in-phase portion of our desired complex quadrature signal. Quadrature-sampling block diagram and spectra within the in-phase upper signal path. Spectra within the quadrature phase lower signal path of the block diagram. I f â€”jQ f is the spectrum of a complex replica of our original bandpass signal xbp t. We show the addition of those two spectra in Figure This â€”jQ f is then added to I f. The complex spectrum at the bottom Figure 18 shows what we wanted, a digitized version of the complex bandpass signal centered about zero Hz. The continuous complex signal i t â€”q t is digitized to

obtain the discrete i n â€”jq n. Some advantages of this quadrature-sampling scheme are: Returning to the Figure 14 block diagram reminds us of an important characteristic of quadrature signals. We can send an analog quadrature signal to a remote location. To do so we use two coax cables on which the two real i t and q t signals travel. Reiteration of how quadrature signals comprise two real parts. We can generate xc t in our laboratory and transmit it to the lab down the hall. All we need is two sinusoidal signal generators, set to the same frequency fo. However, somehow we have to synchronize those two hardware generators so that their relative phase shift is fixed at 90o. Now for a two-question pop quiz. In the other lab, what would we see on the screen of an oscilloscope if the continuous i t and q t signals were connected to the horizontal and vertical input channels, respectively, of the scope?

4: Excel Charts Radar Chart

A file with the links to all my recent videos can be found here: www.enganchecubano.com

5: Python Pandas Tutorial

This site uses cookies for analytics, personalized content and ads. By continuing to browse this site, you agree to this use. Learn more.

6: IBM QRadar vs. Splunk Comparison - UPDATED | IT Central Station

The IBMÂ® Security editors have pulled together many of the video tutorials about QRadarÂ® Security Information and Event Management (SIEM) and its related products so you can get a thorough view of all of its capabilities and, more importantly, so you can get ideas about how to do SIEM right in your environment.

7: QRadar - Documentation - Microsoft Graph

IBM Qradar Tutorials Introduction to QRadar: "Leader" in the Gartner SIEM Magic Quadrant for ; Only the SIEM to achieve perfect 5/5 score in behavior profiling.

8: Check Point and IBM: A Collaborative Approach to Information Security

Many radar sets are introduced briefly as examples with some technical data. Knowledge of basic technical mathematics is required to follow the examples provided. The information provided is intended for radar operators and maintenance personnel.

9: IBM Qradar Training | Qrader Corporate Tarining - Global Online Trainings

Learn more about what IBM Security QRadar unique in this short guided tour of the platform. You'll hear about how this integration solution can replace dozens of point products helping you collect.

Q RADAR TUTORIALS POINT pdf

History of Richard Cromwell and the restoration of Charles II The Guide to Cooking Schools 2005 Connecting students to a changing world Refractive index, absorption, wave length, rotatory power The application of goal setting in sport Kieran M. Kingston and Kylie M. Wilson One Hundred Brachos Counting Your Blessings 100 Times A Day Behavior cycle as a framework for dynamic psychotherapy Uh huh! Alia Starkweather Liberty, community, and justice Combat Conditioning The little book of angels The Stanford health exercise handbook Carbon compound chemistry A guide to hardware managing maintaining A history of western music 9th The challenges of Palestinian filmmaking (1990-2003 In praise of frivolity : on the cinema of Elia Suleim What is vocabulary development Introduction to distributed data processing Myths and unexplored areas State of illinois child support worksheet 2017 BEST PRACTICES 217 Practical guide to power distribution for information technology equipment Earth science by holt mcdougal ch 7 Ib chemistry textbook 2016 Finding emma steena holmes Death and the kings horseman full text Discovering old board games Meaning of production in economics Folk art fish decoys with values Heart of the hydra Gay Marshall The castle experience. Active risk management financial models and statistical methods Seeing the Pattern Commonsense Guide to Grammar and Usage 4e Modern communications electronics The Gambling Times guide to harness racing Reel 17. Barnstable, Berkshire, Bristol Counties Pedro pietri selected poetry The 2007-2012 World Outlook for Mens and Boys Belts Excluding Leather Belts Inheritance cycle book 1 Mankind not the Devils Works (1 Jn. 3.8)