|  |
| --- |
| Title : Parsing Defects/Parsing Improvements: Where To Start When You Don’t Know |
| URL Name |
| **PART IV of VI: Parsing Improvement: Unprocessed Log Queue Growing**  This particular part of the MDI/MPE series is for dealing with ***parsing improvement cases***. KA Parsing Defect (2 of 6) discussed parsing defects and this one follows up with parsing improvements. There are three high-level and general types of parsing improvements: due to **unprocessed log queue,** due to **timeouts**, or parsing **field improvement**. Parts III A, B, and C address them all below.  **PART IV of VI: Rule Not Parsing Field Improvement**  Part III-A outlined one of the more common “help me I’m in trouble scenarios”. You will see that more often because it’s rather difficult to miss when your mediator is spooling. However, that is not the ONLY way that parsing can take a nosedive for the worse. Another common type of elevation is when a rule works but a particular field is left blank. This can also blend with the parsing defect example above where perhaps the field is over or under-parsing a particular field. These cases will usually come from astute customers who are very observant of their traffic and/or sales engineers here at LogRhythm doing a demonstration. In either case the point is that these types of cases you will usually have a knowledgeable customer who is able to explain the challenge in depth. Get on a call to diagnose.  Case 435794 submitted by Sales Engineer Brandon Pace here at LogRhythm is a fantastic example of this type of case. In this particular case there was no call needed because, to the point above, Brandon explained concisely in the ticket. See what Brandon wrote in the ticket below:  Case 435794: *I noticed during a demo today that "Flat File - Cisco Umbrella DNS" no longer parses out the URL/Domain info into the URL field (or any field). When was this change done and was there are reason for this, or was it a mistake? Looks like the "Group" field could use some attention as well:*  *Group example*  *Gregory Pallet,PSB-WIN-ECHO,Default Site*  Based on the above there are up to four steps to completing this case successfully.   1. Check that llx (or csv…but llx preferred) is attached to case. 2. Engage with customer for additional information (skipped here because all info was present) 3. Using the same methods as earlier sections you will want to test the llx against the appropriate policy as a whole. 4. Once the rule you need to test is identifed you will want to confirm/refute what the customer is claiming by testing against that rule specifically.   ***\* Should you test the log sample and receive different results than the customer you might want to check mediator configurations, LPS Detail (other log sources messing up), custom policies, or KB version to name a few things.***  Here are two log samples from the LLX presented in case 435794. Notice the URLs and Groups as pointed out by Brandon in the case summary.  "2018-04-02 17:23:15","Jake Reynolds","Jake Reynolds,PSB-WIN-MGMT,Default Site,PM Sandbox,Pub NAT","192.168.51.127","65.127.112.131","Allowed","1 (A)","NOERROR","s3.us-west-2.amazonaws.com.",""  "2018-04-02 15:07:13","Manish Bhatnagar","Manish Bhatnagar,PSB-STL-AS1,Default Site","192.168.135.221","54.183.86.198","Allowed","1 (A)","NOERROR","lse.ac.uk.","Educational Institutions"  **Screenshot 13**  Graphical user interface, application, table, Excel  Description automatically generated  **Screenshot 14**  Graphical user interface, application  Description automatically generated  Screenshot 13 is the embodiment of steps 1-3 above. We had the llx on the case and the case summary had enough details to understand the identified problem (steps 1-2). We then logged in as a Super User, opened MPE Rule Builder, and selected the Umbrella Flat File Policy with the “test-all” function. We loaded the llx and when we tested the logs where we can see that 100% of these logs identified as Umbrella DNS Requests.  Screenshot 14 models the step 4 above and follows a similar procedure except in this round we test against a single rule in Rule Builder where we can see how it parses. In this case Brandon told us it was DNS Requests so finding the rule is easy. The idea here is to confirm/refute what evidence the customer has sent. If it doesn’t work on your VM then you’re ready to elevate. ***If the results show success on your side then you have to do more digging to rule out whatever environmental variable exists on that customer’s deployment.***  In screenshot 14 you can see that the issue Brandon identified also appears on my VM which was using the most recent KB 7.1.631. This tells us that it is NOT the customer environment but rather the regex inside the rule that could do with some improvement.  We would again use the parsing improvement template which I have complete an example of here:  **Parsing Improvement**    **Existing customer or proof of concept? Proof of Concept.**    **Current Log Processing Policy Version: (Required)**  **LogRhythm Default**    **Use case or justification for the request (Required)**  **Sales was modelling this log parsing policy and one of the key points is to have URLs parsed out to (theoretically the object field…but parsed in general). It doesn't look good when we say look at this and then it doesn't work.**    **MPERuleRegexID to be improved? 1009127**    **The customer MUST be on the latest version before we can elevate to MDI.**  **7.1.631**    **Does the logging format adhere to our standards (e.g., the correct IIS fields are enabled, etc.)?**    **Yes**  **Actual parsing behavior: URL does not parse….Group category looks like it occasionally parses too much (secondary concern)**    **Desired parsing behavior: URL to be parsed into object and perhaps a clean up of how group is parsed.**     * Log samples are required - Included   HAVE YOU ENTERED THE LOGSOURCE TYPE INTO THE SUPPORT CASE? Yes  **You would then follow the SOP set at this link which would have you move it into the appropriate MDI queue.**  <https://confluence.logrhythm.com/display/GS/MDI+Case+Handling#MDICaseHandling-ParsingImprovementRequest> |
| Remediation Process |
| Root Cause |
| Additonal Resources  A really great introduction to how regex works…not mandatory but will for sure help you understand these concepts.  <https://coralogix.com/blog/regex-101/>  A great tool to test regex on logs that is fairly user-friendly.  <https://regex101.com/>  A guide to the top 20 Regex Strings  <https://regexland.com/most-common-regular-expressions/> |
| Tags |