Security-Issue-Classifier (secplugin-cls) Training a Machine Learner Guide V1.1

I. Introduction

SecurityPlugin-classifier is a command line tool to train a text classification model. This model can then be subsequently used to classify issues, stories, features, etc into security or non-security categories. To use a trained model in the JiraSecPlugin, it is important to run this tool on a locally labeled dataset for a project/organization. The benefit of doing this is reflected in the precision and recall values of the classifier when compared to a string pattern function.

A classifier can either be a machine learning model or a string pattern match function. We have validated the plugin model's algorithm using rigorous empirical analysis and comparing the performances of using different machine learning techniques and simple string pattern matches.
Our empirical results show that machine learners trained on project's vocabularies produce powerful and impressive classification models when compared to generic security keywords.

We created 4 categories of terms to form the features/attributes that can be used to train a classifier. Some security vocabularies may vary across projects and organizations. These are mostly reflected in the "Asset", "Control", and "Indirect" terms. Security terms are thus divided into these four categories:

1. Assets or Personally Identifiable Information (PII),
2. Direct (terms related to attacks and vulnerabilities),
3. Control (terms related to implemented security controls), and
4. Indirect (terms that are indirectly related to security and not in the above 3 categories).

As a foundation, we have extracted a set of terms from different sources such as the CWE, OWASP, CVE, RFC 4949, and industrial issue tracking databases. However, to improve your own classification model, you would need to augment these terms with your project specific vocabularies (assets, control, and indirect) terms.

II. Configuration

1. Set up your configuration file

The very first step is to set up a configuration file. This is not a hassle as a default config file comes handy with the zipped files. Training a classification model is simple once the parameters for the algorithms are setup. The parameters in the "config.txt" file are described as shown in the table below.

2. config.txt

-learner ml	choose whether to use machine learning classifier (ml) or only keyword/term search (ts) for classification
-method 2	Choose between method 1, 2 or 3. Method 1 uses LevenshteinDistance algorithm for term with length > term_min_len. This is the number of changes needed to change one String into another, where each change is a single character modification (deletion, insertion or substitution). #Method 2 uses JaroWinklerDistance algorithm for term with length > term_min_len. The Jaro measure is the weighted sum of percentage of matched characters from each file and transposed characters. Winkler increased this measure for matching initial characters. #Method 3 uses full string search for term with length <= term_min_len and substring search for term with length > term_min_len
-classifyonly no	Skip training a model and use already trained classifier (yes or no)
-termMinLen 4	#Assumption-the shorter the length of term the lower the probability of making typo or other errors
-threshLev 1 -threshJaro 0.95	#Set Threshold for determining accepted terms with LevenshteinDistance/jarowinkler algorithm: has the benefit of absorbing human typo errors in the commit desc/summary
-textIndex 1,2	# example "Log user login attempts; We should be able to log login attempts by user; YES"

	# text_index=1,2, class_index=3, and separator=;
-classIndex 3	# use -1 when data has no label and should be classified only.
-separator ;	#security issue labels – Values defined must match the labels in JIRA
-file /User/Test/derby170316.xlsx	#File containing your dataset (Text and labels)
-header true	# does file have a header? true or false
-numExpr 10	# number of times to train a classifier
-algorithm RF	# algorithm- choose any or all of RF,NB,SVM,J48 separated by comma
-trainSize 0.7	# size of training set (0.0 and 1.0)->(0% - 100%)
-testAll no	# use all the dataset for testing? (yes/no). If no, the remaining unseen data will be used for testing

3. Example of local dataset with classification

The example in the table below shows how we performed a security labeling of issues. We have done similar classification of over 7000 messages (issues, commits, and emails) spanning 4 different projects.
In the example below:
-textIndex 1
-classIndex 4

Text	Project	Source	label	SecurityProperty	SecurityClass
Github user PiotrKlimczak closed the pull request at: https://github.com/apache/camel/pull/416 --- If your project is set up for it you can reply to this email and have your reply appear on GitHub as well. If your project does not have this feature enabled and wishes so or if the feature is enabled but not working please contact infrastructure at infrastructure@apache.org or file a JIRA ticket with INFRA. ---	camel	email	0		
[https://issues.apache.org/activemq/browse/CAMEL-1930?page=com.atlassian.jira.plugin.system.issuetabpanels:all-tabpanel] Claus Ibsen updated CAMEL-1930: ---------- Fix Version/s: 2.1.0 > Synchronized access to XPathExpression resulting in contention for multiple consumers > ---------- > > Key: CAMEL-1930 > URL: https://issues.apache.org/activemq/browse/CAMEL-1930 > Project: Apache Camel > Issue Type: Bug > Components: camel-core > Affects Versions: 2.0-M3 > Environment: Java 1.6 Spring 2.5.6 > Reporter: Fabrice Delaporte > Fix For: 2.1.0 > > > Hi > I'm using Camel to do some JMS message routing. Messages are XML so xpath is a natural choice. > However when using a choice with an xpath expression the XPathBuilder creates one XPathExpression object. According to the specification these objects are not thread safe so synchronizing looks natural. But then using multiple jms consumers is totally useless since no concurrent evaluations can be made. > XPathExpression objects would rather need to be stored in a ThreadLocal to avoid synchronization and contention. > Cheers > Fabrice --	camel	email	1	integrity	implicit
Hadrian Zbarcea commented on CAMEL-3099: ---------- @Willem Encryption has nothing to do with this the issue is not not display a password in clear in logs jmx consoles etc. The patch hardcodes password and passphrase to be considered as secrets. Always. Which may or may not be the case. If you saw my comment in the message Lorrin sent to the users@ list I was thinking about the same issue and a solution I am working on now is to annotate with @Secret fields that are considered well secrets and must never be displayed in clear. I think that is a more general solution. We will then need to document how to best provide secrets to camel like properties files with 400 permissions not use them as arguments in command lines etc. Obviously the credit still goes to Lorrin for reporting this :). > passwords and other private data contained in URIs should not be logged in plaintext > ---------- > > Key: CAMEL-3099 > URL: https://issues.apache.org/activemq/browse/CAMEL-3099 > Project: Apache Camel > Issue Type: Improvement > Components: camel-core > Reporter: Lorrin Nelson > Assignee: Hadrian Zbarcea > Priority: Minor > Attachments: 0001-Reduce-risk-of-showing-passwords-in-URIs-by-adding-c.patch > > > URIs with sensitive data are common and that URIs are frequently logged. I bumped into this myself most recently with an FTP consumer. I ended up with log messages like this: > RemoteFileProducer 2010-08-31 16:21:45 459 -- INFO -- Connected and logged in to: Endpoint[sftp://myusername@my.host.name/var/my/path?fileName=myFile.txt&password=yikesMyPassword] > I propose a sane-defaults patch of modifying DefaultEndoint.java's toString to sanitize the URI by looking for URI params containing the tokens "password" or "passphrase" and rendering their value as "********" instead of the actual value. Obviously this isn't always the right thing to do in every situation but it seems appropriate for many endpoints. Any for which it is not appropriate could override toString	camel	email	1	confidentiality	explicit
[https://issues.apache.org/activemq/browse/CAMEL-477?page=com.atlassian.jira.plugin.system.issuetabpanels:all-tabpanel] Willem Jiang reassigned CAMEL-477: ---------- Assignee: Willem Jiang > allow Spring Integration channels/messages/endpoints to be used as native Camel endpoints > ---------- > > Key: CAMEL-477 > URL: https://issues.apache.org/activemq/browse/CAMEL-477 > Project: Apache Camel > Issue Type: New Feature > Components: camel-spring-integration > Reporter: James Strachan > Assignee: Willem Jiang > Fix For: 1.4.0 > > --	camel	email	0		
Support to specify the operation and operation and operation namespace from camel-cxf endpoint URI for camel-cxf producer ---------- Key: CAMEL-2780 URL: https://issues.apache.org/activemq/browse/CAMEL-2780 Project: Apache Camel Issue Type: Improvement Components: camel-cxf Reporter: Willem Jiang Assignee: Willem Jiang Fix For: 2.4.0 In this way user don't need to set the message	camel	email	1	api	implicit

header with operation name and operation namespace every time. Here is the [mail thread|http://old.nabble.com/How-Do-We-Specify-Operation-To-Choose-In-Camel-CXf-ts28745738.html] which discusses about it. --					
Hi I'm running camel inside karaf 2.2.5 bootstrapping the camel context with Spring DM. I also could not reproduce it with a basic unit test like the ones in the camel-core source tree. However I can reproduce it with the camel-maven-plugin (mvn camel:run): ********** Error occurred while running main from: org.apache.camel.spring.Main java.lang.reflect.InvocationTargetException at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:597) at org.apache.camel.maven.RunMojo$1.run(RunMojo.java:416) at java.lang.Thread.run(Thread.java:680) Caused by: java.lang.StackOverflowError at org.apache.camel.model.RouteDefinition.toString(RouteDefinition.java:116) at java.lang.String.valueOf(String.java:2826) at java.lang.StringBuilder.append(StringBuilder.java:115) ... etc ... Thanks for looking into this. Jerry -- View this message in context: http://camel.465427.n5.nabble.com/Camel-2-9-0-StackOverflowError-initializing-route-with-otherwise-clause-tp5110599p5113396.html Sent from the Camel Development mailing list archive at Nabble.com.	camel	email	1	integrity	implicit
[https://issues.apache.org/activemq/browse/CAMEL-2371?page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel&focusedCommentId=58048#action_58048] Ashwin Karpe commented on CAMEL-2371: ---------- Hi Claus Thanks for the feedback. I will work on the changes you suggested and submit the next iteration for your review and comment. I will try to get it done by this weekend. I am on a customer gig all day tomorrow and proceed on vacation all of next week and may not be able to get a chance to work on it. In the event I cannot get it done this weekend please bear with me. I will try and submit it early the week following. As for the annotation ***@ChannelPipelineCoverage("all")*** it simply means that the same handler may be applied against multiple channel pipeline instances. The only options available are "one" and "all". The choice of "one" will force creation of individual handler instances per pipeline. The exceptionCaught() debug statement stating that the channel is closed but the fact that it is not is a glitch. I need to rework the debug statement (possibly even remove the method and fall back on default handling). Channel closure is not essential since there are some cases where exceptions caught are not fatal in nature. I will make the countdownLatch local and check out ServicePoolAware. That is a good catch. I did not pick up on this implication during development. Thanks. Cheers Ashwin... > Netty component > ---------- > > Key: CAMEL-2371 > URL: https://issues.apache.org/activemq/browse/CAMEL-2371 > Project: Apache Camel > Issue Type: New Feature > Reporter: Claus Ibsen > Assignee: Ashwin Karpe > Fix For: Future > > Attachments: camel-netty-patch.diff camel-netty20100304.zip > > > Consider creating a new JBoss [Netty|http://www.jboss.org/netty/] component as a supplement to the MINA component. > It starts to become a _joke_ with the Mina 2.0 release which has take 2+ years and still not released. > And one of the primary drivers behind MINA joined JBoss and created Netty instead. It appears as a good alternative. > Netty is also Apache licensed. --	camel	email	0		
Github user chenapan closed the pull request at: https://github.com/apache/camel/pull/1288 --- If your project is set up for it you can reply to this email and have your reply appear on GitHub as well. If your project does not have this feature enabled and wishes so or if the feature is enabled but not working please contact infrastructure at infrastructure@apache.org or file a JIRA ticket with INFRA. ---	camel	email	0		
[https://issues.apache.org/activemq/browse/CAMEL-1012?page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel&focusedCommentId=46629#action_46629] Claus Ibsen commented on CAMEL-1012: ---------- Currently RedeliveryPolicyType will be generated into 7 sub elements. Why not use attributes for the 7 simple types? > redeliveryPolicyType should use attributes for simple values > ---------- > > Key: CAMEL-1012 > URL: https://issues.apache.org/activemq/browse/CAMEL-1012 > Project: Apache Camel > Issue Type: Sub-task > Reporter: Claus Ibsen > Assignee: Jonathan Anstey > --	camel	email	0		
Hi Is there any specific reason why we should wait upgrading to Lucene 4.5.1 https://github.com/apache/camel/blob/master/parent/pom.xml#L255 The matching SMX bundle version 4.5.1_1 would be in central as well. Other than that I'm currently not able to back port commits to the 2.12.x branch any other committer having a similar issue? ~/dev/workspace/camel-2.12.x>git cherry-pick 1f868778e error: could not apply 1f86877... Upgraded elasticsearch to the version 0.90.5. hint: after resolving the conflicts mark the corrected paths hint: with 'git add <paths>' or 'git rm <paths>' hint: and commit the result with 'git commit' Babak -- View this message in context: http://camel.465427.n5.nabble.com/Upgrading-to-Lucene-4-5-1-tp5742937.html Sent from the Camel Development mailing list archive at Nabble.com.	camel	email	1	quality	implicit
[https://issues.apache.org/activemq/browse/CAMEL-2692?page=com.atlassian.jira.plugin.system.issuetabpanels:all-tabpanel] Roland Knight reopened CAMEL-2692: ---------- Claus the synchronize you added didn't fix the problem. Seems that importNode requires the Document to be synchronized (yuck). Changing your fix in XmlConverter.toDOMDocument to: Document doc = createDocument(); // import node must no occur concurrent on the same node // so we need to synchronize on it synchronized (node.getOwnerDocument()) { doc.appendChild(doc.importNode(node true)); } fixed the problem. I agree about the JDK XML API. It is horrible. I wrote a converter for DOM4J and always convert the body to a DOM4J Document before any DOM manipulation. > Multithreading bug: getBody sporadically returns null > ---------- > > Key: CAMEL-2692 > URL: https://issues.apache.org/activemq/browse/CAMEL-2692 > Project: Apache Camel > Issue Type: Bug > Components: camel-core > Affects Versions: 1.6.2 2.3.0 > Environment: Windows 7 64 bit JDK 1.6.0_20 > Reporter: Roland Knight > Assignee: Claus Ibsen > Fix For: 1.6.3 2.3.0 > > Attachments: news_20100502000001.zip > > > Note that the only workaround for this bug is to remove the parallelProcessing() call in the builder. > I have a simple route that processes a file by splitting on a tag and processing the DOM of each split message. The problem is that getBody is randomly returning null but ONLY when using the parallelProcessing feature of split. For some runs of the same XML file the error does not occur at all (the file is about 2MB of data) others it will happen once or twice. I am currently using the latest 2.3-SNAPSHOT. > Also note that after detecting the null I tried calling getBody(String.class) - this also may return null. Sometimes it does return the proper XML. > Route configuration that reproduces	camel	email	1	integrity	implicit

the problem (my input XML is about 2MB with about 500 article tags): > public void configure() throws Exception { > from("file:D:/inbox") > .split(new XPathBuilder("//article")) > .parallelProcessing() // remove this line getBody below never returns null > .process(new Processor() { > public void process(Exchange exchange) throws Exception { > Message inMessage = exchange.getIn(); > org.w3c.dom.Document domDocument = inMessage.getBody(org.w3c.dom.Document.class); > if (domDocument == null) { > log("Null body"); > } else { > // process DOM here > } > } > }) > .end() > } > }); --					
[https://issues.apache.org/activemq/browse/CAMEL-2995?page=com.atlassian.jira.plugin.system.issuetabpanels:all-tabpanel] Willem Jiang reassigned CAMEL-2995: ---------- Assignee: Willem Jiang > charset parser should cater for quotes both single and double quotes > ---------- > > Key: CAMEL-2995 > URL: https://issues.apache.org/activemq/browse/CAMEL-2995 > Project: Apache Camel > Issue Type: Bug > Components: camel-http > Affects Versions: 2.3.0 > Reporter: Claus Ibsen > Assignee: Willem Jiang > Fix For: 2.5.0 > > > See nabble > http://camel.465427.n5.nabble.com/issue-with-encoding-when-using-HTTP-component-td2227887.html#a2227887 > I bet many systems may report charset in different ways such as > {code} > Content-Type:text/xml;charset="utf-8" > Content-Type:text/xml;charset='utf-8' > Content-Type:text/xml;charset=utf-8 > {code} > We should ensure that we support all ways of setting this. And there may also be spaces between so we should trim and whatnot. > The code in 2.4 may have been improved. Just creating a ticket to be sure. --	camel	email	0		
[https://issues.apache.org/jira/browse/CAMEL-3551?page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel&focusedCommentId=13144730#comment-13144730] Christian MÃ¼ller commented on CAMEL-3551: ---------- For org.apache.avalon.framework:avalon-framework-api:4.3.1 org.apache.avalon.framework:avalon-framework-impl:4.3.1 SMX already provides OSGI bundles... > camel-fop component > ---------- > > Key: CAMEL-3551 > URL: https://issues.apache.org/jira/browse/CAMEL-3551 > Project: Camel > Issue Type: New Feature > Reporter: Jean-Baptiste OnofrÃ© > Assignee: Jean-Baptiste OnofrÃ© > Fix For: Future > > Attachments: camel-fop.diff > > > A new Camel FOP component could be helpful to turn Camel into a kind of printout and report generation system. > A typical use case could be something like: > from("amq:my.document.queue") > .to("xslt:mystylesheet.xsl") > .to("fop:pdf?some.extra.options.here") > .to("file:outputdirectory") > .to("printer:some.printer"); -- This message is automatically generated by JIRA. If you think it was sent incorrectly please contact your JIRA administrators: https://issues.apache.org/jira/secure/ContactAdministrators!default.jspa For more information on JIRA see: http://www.atlassian.com/software/jira	camel	email	1	api	implicit
[https://issues.apache.org/activemq/browse/CAMEL-1098?page=com.atlassian.jira.plugin.system.issuetabpanels:all-tabpanel] tim mcnamara updated CAMEL-1098: ---------- Attachment: multicast.patch Patch for Multicast and Splitter processors > multicast and file consumer using managed threads? > ---------- > > Key: CAMEL-1098 > URL: https://issues.apache.org/activemq/browse/CAMEL-1098 > Project: Apache Camel > Issue Type: Improvement > Components: camel-core > Affects Versions: 1.4.0 1.5.0 > Reporter: tim mcnamara > Fix For: 2.0.0 > > Attachments: multicast.patch > > > Running in a server managed environment it is preferable to use managed threads when ever possible. Is it possible to have these components (and others that spawn threads) modified to use Spring's TaskExecutor abstraction (a la JmsComponent). If this is the case we could configure the components to use the server's WorkManager API. --	camel	email	1	integrity	implicit

4. Training a model

To train a model, follow these steps:

1. Download the securityplugin-classifier.zip.
2. Unzip the file
3. Edit the config.txt file in the folder “securityplugin-classifier”
4. cd securityplugin-classifier
5. run the command: java -jar securityplugin-cls.jar -config config.txt
6. When the experiment is finished, 2 files are created (best model)
    a. model.cls
    b. modeleval.cls
7. If you are satisfied with the model’s performance, copy (**THIS IS VERY IMPORTANT!!!**)
    a. model.cls and modeleval.cls into your JiraSecPlugin directory.
    b. controlterms.prop, directterms.prop, indirectterms.prop, and piiterms.prop into JiraSecPlugin directory
8. Disable and Enable the plugin from Jira from the “Add-on -> Manage add-ons” to activate your new changes.

To see help message,
run the command: java -jar securityplugin-cls.jar -help

III. Classifying Unlabeled data

The classifier can also be used to perform experiments on downloaded issues from repositories. A trained model can be used to label (classify) new dataset. The necessary parameters to change in the "config.txt" are:

-classifyonly [change this to yes]
-file [specify the file you want model to classify (only xlsx and csv are allowed)]
-textIndex [index or indices for the text messages]
-classIndex [enter -1]
-header [true/false]
-separator [specify the separator here]

When done,
run the command: java -jar securityplugin-cls.jar -config config.txt

When the classification is finished, a new file with the classification labels for each text (ends with "_classified.csv) will be generated.