Shadow IT Detection with IBM QRadar SIEM

Introduction

Shadow IT refers to the information technology solutions used inside an organization without the explicit approval of the organization. In recent years, the advent of cloud computing has made it easier for employees to circumvent IT department and use a variety of cloud applications without the knowledge or approval of the organization. Despite the high visibility of recent data breaches, most employees still choose to use cloud services to be able to do their job more efficiently. In a study conducted by IBM Security, it was found that 1 in every 3 Fortune 1000 employees regularly saves and shares company data to third-party cloud-based platforms that are not explicitly approved by their organization [1]. This figure is expected to increase as the workplace demographic starts to change and millennials who are greater users of cloud applications [2] make up more and more of the workforce.

According to Gartner, most organizations grossly underestimate the number of shadow IT applications already in use and a data breach resulting from these can result in very large financial liabilities due to a mix of costs that include notification penalties, auditing processes, loss of customer revenue, brand damage, security remediation and investment, and cyber insurance [3]. However simply blocking access to all third-party cloud services is not the right solution. Organizations need to strike a balance between the security risks and cost savings and efficiency benefits of using these services.

The first step to tackling the shadow IT problem is to identify what applications are being used, by whom and for what purpose. This discovery period will help the organization to determine the high risk applications that need to be eliminated and allow the adoption of safer alternatives and create safe usage policies. In this article, we focus on shadow cloud applications and illustrate how you can utilize IBM QRadar SIEM with IBM X-Force Intelligence feed to gain visibility into unapproved cloud application usage within your organization.

IBM Security X-Force Intelligence feed

The IBM Security X-Force Intelligence Feed delivers insight into entities on the Internet that is based on knowledge of more than 15 billion web pages. The feed adds dynamic Internet threat data to the analytical capabilities of the QRadar Security Intelligence Platform, enriching QRadar's threat analysis capabilities with up-to-the-minute data.

Shadow Cloud Application Discovery

In this article, we extract the URL information from the raw logs generated by enterprise web gateways and/or next generation firewall devices and have QRadar process it with the help of IBM Security X-Force Intelligence Feed to identify cloud applications that are in use without the approval of the enterprise.

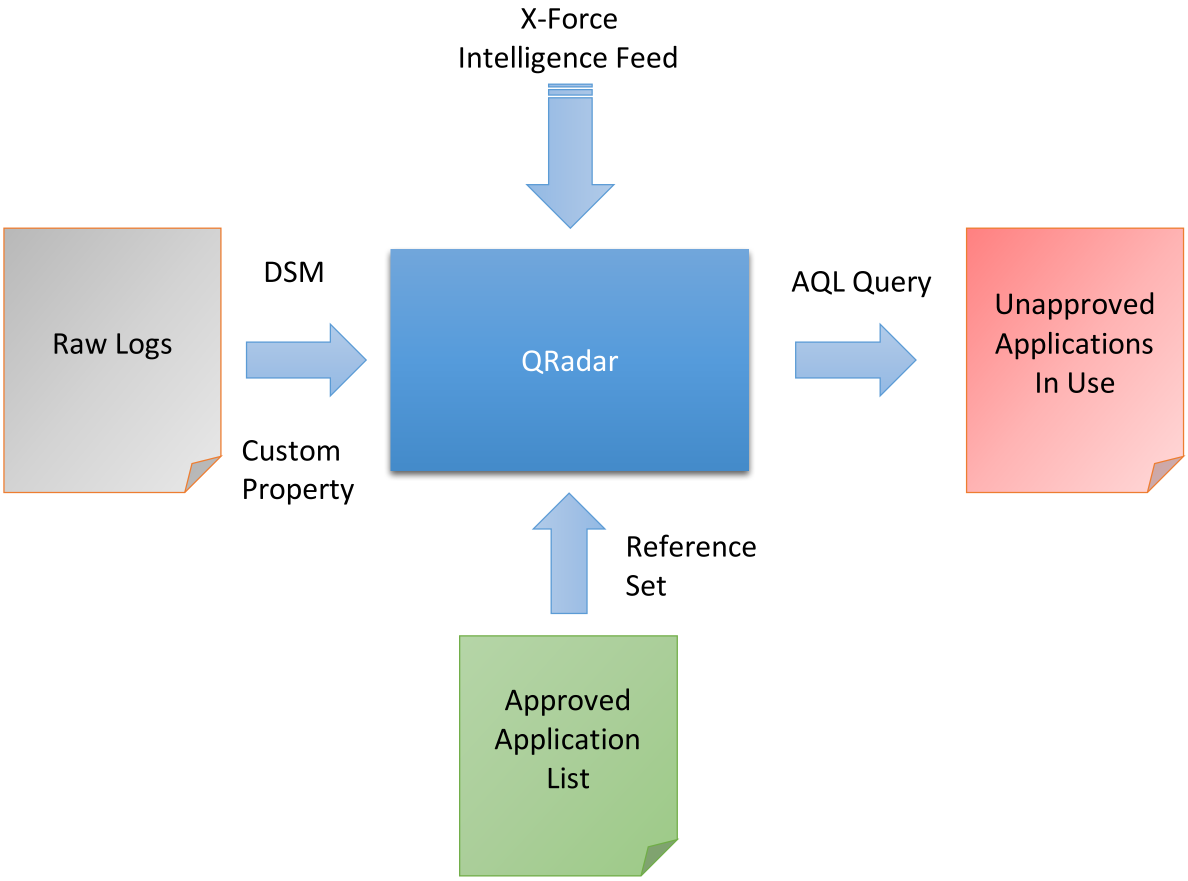


Figure 1. Generating a report of unapproved cloud application usage with QRadar

As outlined in figure 1 above, the process involves:

Setup

- Add QRadar custom properties to extract the relevant information from raw logs:
  QRadar custom properties provide a way to extract extra fields from logs that are not parsed by default via QRadar DSMs. We will need to extract URL fields to be able to identify shadow cloud applications in use.
- Define the list of approved cloud applications:
  QRadar reference sets provide a way to store extra information in QRadar tables that are not part of the logs fed into the QRadar but can be referenced by the AQL queries for log analysis. We will create a reference set to store the list of approved cloud applications so they can be excluded from the shadow IT analysis results.

Analysis

- Create the AQL queries to discover the shadow cloud applications in use:
  We will provide several AQL queries that can be used to detect the shadow cloud applications in use with the aid of IBM Security X-Force Intelligence feed. These queries are meant as examples that you can use as a starting point to define your own analysis queries. X-Force intelligence feed is enabled on QRadar if you have a valid X-Force feed license. Note: You will need at least QRadar version 7.2.7 to be able to utilize the X-Force AQL functions used in this article.

Setup: Add custom properties to extract URL from raw logs

First step is to extract the URL information from the raw logs using the Custom Event Property feature of QRadar. Custom event properties are regular expressions used to extract fields from raw logs that are not parsed out by default. In this article, we are using Bluecoat logs in bcreportermain_v1 format as example but the same concepts apply to any web application gateway or next generation firewall logs that contain information on the URLs being accessed by the enterprise users.

Bluecoat bcreportermain_v1 format is shown below (The fields we are interested in are shown in bold):

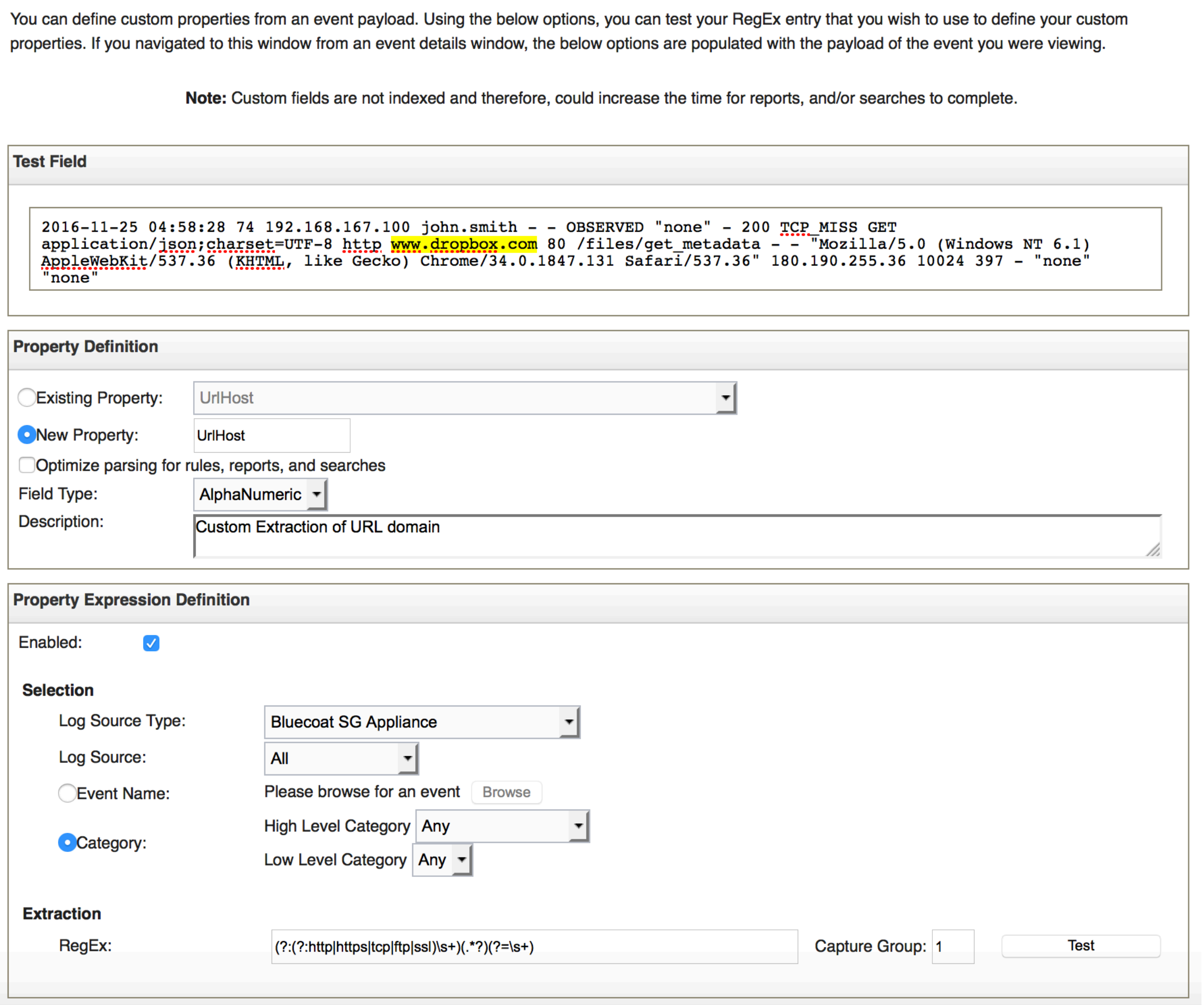
```
date time time-taken c-ip cs-username cs-auth-group x-exception-id sc-filter-result cs-categories cs(Referer) sc-status s-action cs-method rs(Content-Type) cs-uri-scheme cs-host cs-uri-port cs-uri-path cs-uri-query cs-uri-extension cs(User-Agent) s-ip sc-bytes cs-bytes x-virus-id
```

Note: Bluecoat supports many different logging formats, including user configured custom formats. If you are using a different log format than what's described here, you will need to adjust your custom property regular expressions accordingly.

The **cs-host** field contains the hostname of the URL accessed by the user. The **cs-uri-path** contains the URL path accessed by the user. To be able to identify the application correctly, we will need to extract both of these fields. For example, the cs-host field can be www.google.com and cs-path field can be /photos. To be able to identify the application as Google Photos, we need to extract both fields.

To extract the cs-host field, we created a QRadar custom event property named 'UrlHost' (Figure 2) and used the following regular expression to extract it from the raw log:

```
(?:(?:http|https|tcp|ftp|ssl)\s+)(.*?)(?=\s+)
```



You can define custom properties from an event payload. Using the below options, you can test your RegEx entry that you wish to use to define your custom properties. If you navigated to this window from an event details window, the below options are populated with the payload of the event you were viewing.

Note: Custom fields are not indexed and therefore, could increase the time for reports, and/or searches to complete.

Test Field

```
2016-11-25 04:58:28 74 192.168.167.100 john.smith - - OBSERVED "none" - 200 TCP_MISS GET
application/json;charset=UTF-8 http www.dropbox.com 80 /files/get_metadata - - "Mozilla/5.0 (Windows NT 6.1)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/34.0.1847.131 Safari/537.36" 180.190.255.36 10024 397 - "none"
"none"
```

Property Definition

○ Existing Property: UrlHost

● New Property: UrlHost

□ Optimize parsing for rules, reports, and searches

Field Type: AlphaNumeric

Description: Custom Extraction of URL domain

Property Expression Definition

Enabled: ☑

Selection

Log Source Type: Bluecoat SG Appliance

Log Source: All

○ Event Name: Please browse for an event

● Category: High Level Category Any

Low Level Category Any

Extraction

RegEx: (?:(?:http|https|tcp|ftp|ssl)\s+)(.*?)(?=\s+)

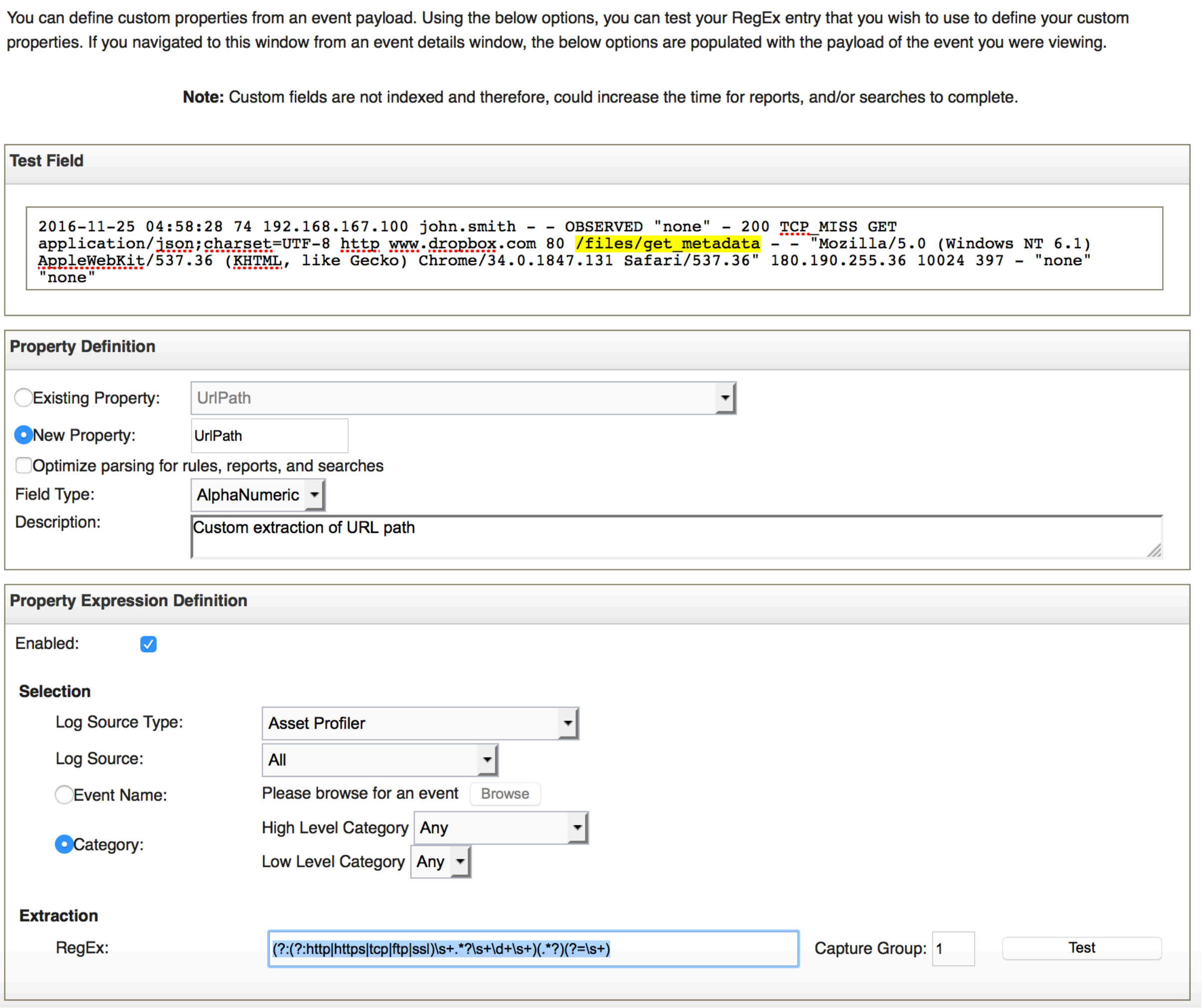
Capture Group: 1

Figure 2. Creating custom event property ‘UrlHost’

To extract the cs-uri-path field, we created another QRadar custom event property named ‘UrlPath’ (Figure 3) and used the following regular expression to extract it from the raw log:

```
(?:(?:http|https|tcp|ftp|ssl)\s+.*?\s+\d+\s+)(.*?)(?=\s+)
```

We don’t need to add a custom event property for cs-username field as QRadar extracts this field by default and assigns to the ‘username’ property.



You can define custom properties from an event payload. Using the below options, you can test your RegEx entry that you wish to use to define your custom properties. If you navigated to this window from an event details window, the below options are populated with the payload of the event you were viewing.

Note: Custom fields are not indexed and therefore, could increase the time for reports, and/or searches to complete.

Test Field

```
2016-11-25 04:58:28 74 192.168.167.100 john.smith - - OBSERVED "none" - 200 TCP_MISS GET
application/json;charset=UTF-8 http www.dropbox.com 80 /files/get_metadata - - "Mozilla/5.0 (Windows NT 6.1)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/34.0.1847.131 Safari/537.36" 180.190.255.36 10024 397 - "none"
"none"
```

Property Definition

○ Existing Property: UrlPath

● New Property: UrlPath

□ Optimize parsing for rules, reports, and searches

Field Type: AlphaNumeric

Description: Custom extraction of URL path

Property Expression Definition

Enabled: ☑

Selection

Log Source Type: Asset Profiler

Log Source: All

○ Event Name: Please browse for an event

● Category: High Level Category Any

Low Level Category Any

Extraction

RegEx: (?:(?:http|https|tcp|ftp|ssl)\s+.*?\s+\d+\s+)(.*?)(?=\s+)

Capture Group: 1

Figure 3. Creating custom event property ‘UrlPath’

Analysis: Obtaining the list of applications used

Now that we have the custom event properties created, we are ready to run the AQL query to generate the list of applications used:

```
select CONCAT(UrlHost + ‘/’ + UrlPath) as FullUrl,
XFORCE_WAC_APPLICATION(FullUrl) as ApplicationName from events group
by ApplicationName last 7 days
```

In this query, we are first concatenating UrlHost and UrlPath to construct the FullUrl and then invoking the XFORCE_WAC_APPLICATION() AQL function to get the application name for the FullUrl. The XFORCE_WAC_APPLICATION() function uses the X-Force Intelligence Feed to identify the application from the provided URL. The query results contain all applications used within the last 7 days.

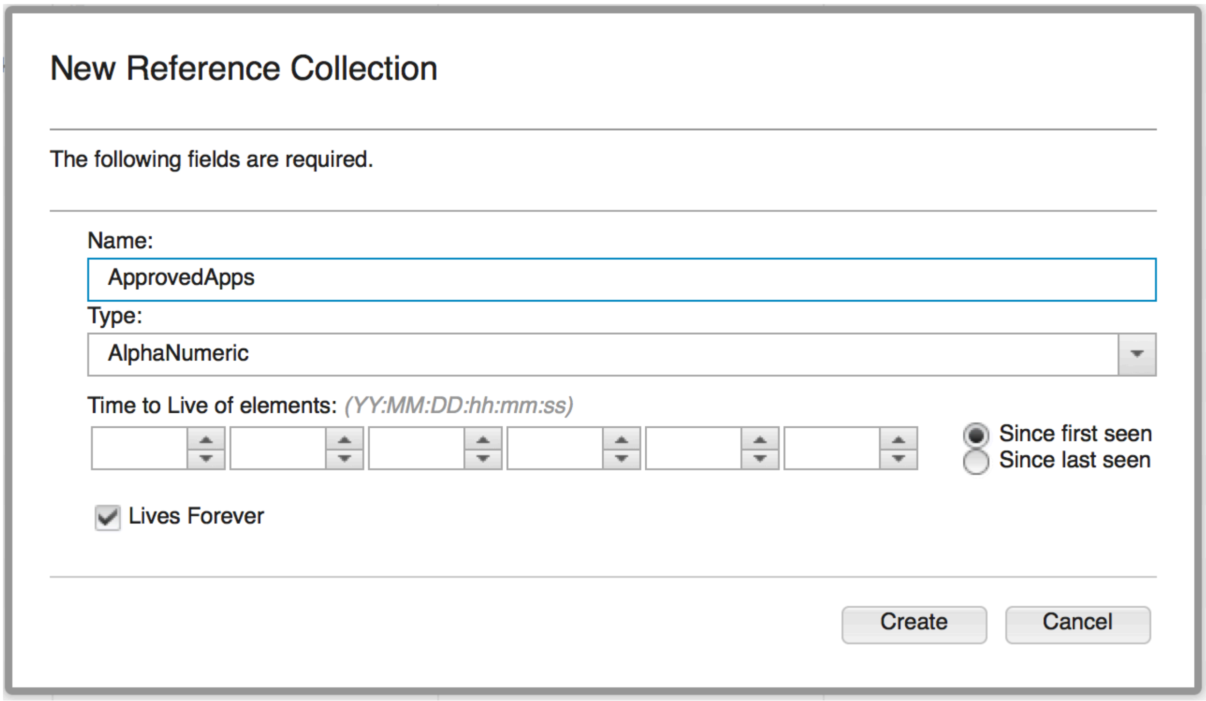
When we executed this query via QRadar ariel REST API, we got the following JSON data:

```
{
  "events": [
    {
      "FullUrl":
"www.box.com/files/apps/service/windows/support/service/service/windows/a.h
tml",
      "ApplicationName": "Box"
    },
    {
      "FullUrl": "www.successfactors.com/b/private.do",
      "ApplicationName": "SuccessFactors"
    },
    {
      "FullUrl": "www.webex.com/b/b/service/c/apps/files.mp4",
      "ApplicationName": "WebEx Cisco"
    },
    {
      "FullUrl": "github.com/b/a/service.do",
      "ApplicationName": "GitHub"
    },
    {
      "FullUrl":
"www.servicenow.com/a/support/c/b/a/support/purchase.html",
      "ApplicationName": "ServiceNow"
    },
    {
      "FullUrl":
"www.workday.com/files/private/a/html/windows/support/html/windows/c/privat
e.html",
      "ApplicationName": "Workday"
    },
    {
      "FullUrl": "www.dropbox.com/files/c/a/apps/windows/private.txt",
      "ApplicationName": "Dropbox"
    },
    {
      "FullUrl": "www.linkedin.com/files/downloads/service.html",
      "ApplicationName": "LinkedIn"
    },
    {
      "FullUrl": "www.echosign.adobe.com/support.mp4",
      "ApplicationName": "Adobe Document Cloud"
    },
    {
      "FullUrl": " mozy.co.uk/product/online-backup",
      "ApplicationName": "mozy"
    },
    ...
  ]
}
```

We need to further refine this query to only return the applications that are not approved.

Setup: Defining the list of approved applications

The next step is to let QRadar know the list of approved applications so it can exclude those from the list. For this, we created a QRadar Reference Set named ‘ApprovedApps’ using the ReferenceSetManagement option in the QRadar admin tab. The approved app entries should not expire so we checked the ‘Lives Forever’ option. (Figure 4).



New Reference Collection

The following fields are required.

Name: ApprovedApps

Type: AlphaNumeric

Time to Live of elements: *(YY:MM:DD:hh:mm:ss)*

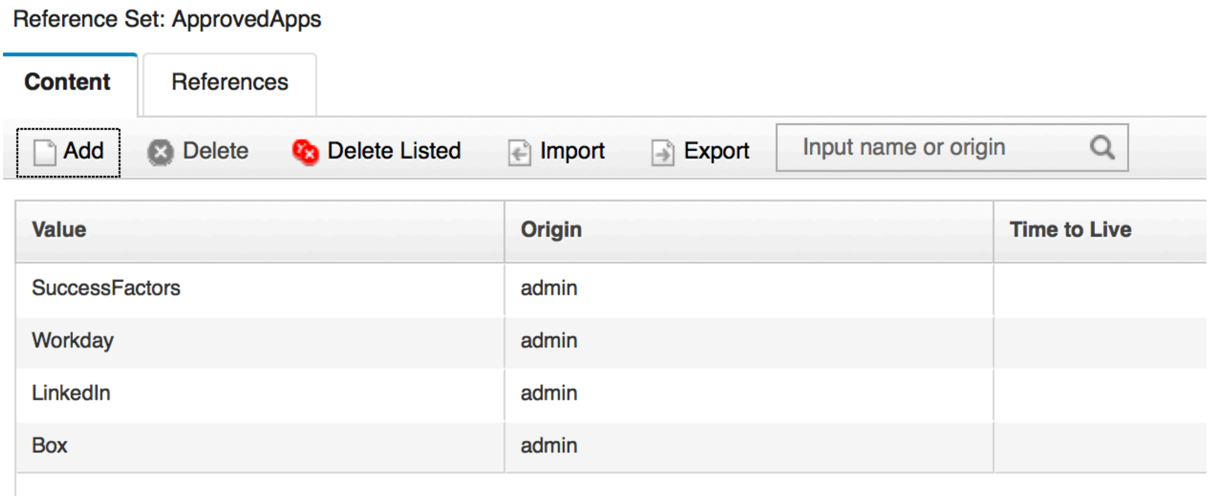
● Since first seen
○ Since last seen

☑ Lives Forever

Figure 4. Creating the ApprovedApps Reference Set

You can populate the ApprovedApps Reference Set by manually entering the names of the approved applications or importing a text file that contains the list of your approved applications. The text file should contain a single application name per line. In this example, we imported the following text file into our ApprovedApps reference set:

SuccessFactors
Workday
LinkedIn
Box



Reference Set: ApprovedApps

Value	Origin	Time to Live
SuccessFactors	admin	
Workday	admin	
LinkedIn	admin	
Box	admin	

Figure 5. Contents of the ApprovedApps reference set

Note: If you have a subscription for IBM X-Force Exchange, you can obtain the list of all recognized application names by using the X-Force Exchange REST API 'GET /app/' (See the API documentation for more information)

Analysis: Obtaining the list of unapproved applications used

Now that we configured QRadar with the list of all approved applications, we can refine our query to only return the unapproved applications:

```
select CONCAT(UrlHost + ‘/’ + UrlPath) as FullUrl,
XFORCE_WAC_APPLICATION(FullUrl) as ApplicationName from events where
REFERENCESETCONTAINS(‘ApprovedApps’, ApplicationName) = ‘false’
group by ApplicationName last 7 days
```

Since we are only interested in unapproved cloud applications used, we can further refine the query to only return the remote applications accessed by local users by including only the events that are in local-to-remote (L2R) direction:

```
select CONCAT(UrlHost + ‘/’ + UrlPath) as FullUrl,
XFORCE_WAC_APPLICATION(FullUrl) as ApplicationName,
XFORCE_URL_CATEGORY(FullUrl) as Category from events where
REFERENCESETCONTAINS(‘ApprovedApps’, ApplicationName) = ‘false’ and
eventdirection = ‘L2R’ group by ApplicationName last 7 days
```

When we executed this query via QRadar ariel REST API, we got the following JSON data:

```
{
  "events": [
    {
      "FullUrl": "www.webex.com/b/b/service/c/apps/files.mp4",
      "ApplicationName": "WebEx Cisco"
    },
    {
      "FullUrl": "github.com/b/a/service.do",
      "ApplicationName": "GitHub"
    },
    {
      "FullUrl":
"www.servicenow.com/a/support/c/b/a/support/purchase.html",
      "ApplicationName": "ServiceNow"
    },
    {
      "FullUrl": "www.dropbox.com/files/c/a/apps/windows/private.txt",
      "ApplicationName": "Dropbox"
    },
    {
      "FullUrl": "www.echosign.adobe.com/support.mp4",
      "ApplicationName": "Adobe Document Cloud"
    },
    {
      "FullUrl": " mozy.co.uk/product/online-backup",
      "ApplicationName": "mozy"
    },
    ...
  ]
}
```

Analysis: Obtaining the list of unapproved applications used by each user

To obtain a list of users who are accessing the unapproved cloud applications, we can run the following query:

```
select username, CONCAT(UrlHost + ‘/’ + UrlPath) as FullUrl,
XFORCE_WAC_APPLICATION(FullUrl) as ApplicationName,
XFORCE_URL_CATEGORY(FullUrl) as Category from events where
REFERENCESETCONTAINS(‘ApprovedApps’, ApplicationName) = ‘false’ and
eventdirection = ‘L2R’ group by username,ApplicationName last 7 days
```

When we executed this query via QRadar ariel REST API, we got the following JSON data that shows the list of users and the unapproved cloud applications they have accessed:

```
{
  "events": [
    {
      "FullUrl": "www.webex.com/b/b/service/c/apps/files.mp4",
      "ApplicationName": "WebEx Cisco",
      "username": "bdenize@mycompany.com",

    },
    {
      "FullUrl": "www.webex.com/upload/files",
      "ApplicationName": "WebEx Cisco",
      "username": "rdanial@mycompany.com",

    },
    {
      "FullUrl": "www.dropbox.com/files/c/a/apps/windows/private.txt",
      "ApplicationName": "Dropbox",
      "username": "rdanial@mycompany.com",
    },
    {
      "FullUrl": "www.dropbox.com/downloads",
      "ApplicationName": "Dropbox",
      "username": "esmith@mycompany.com",
    },
    {
      "FullUrl": "www.dropbox.com/folders/customerdata",
      "ApplicationName": "Dropbox",
      "username": "jdrem@mycompany.com",
    },
    ...
  ]
}
```

Analysis: Obtaining the list of unapproved cloud applications by category

By using the XFORCE_URL_CATEGORY() AQL function, we can get category information for the applications and use this information to gain more insight into the type of shadow cloud usage within the enterprise. XFORCE_URL_CATEGORY() function utilizes X-Force Intelligence Feed to identify URL category.

To gain visibility into unapproved cloud storage usage, we can run the following query:

```
select CONCAT(UrlHost + ‘/’ + UrlPath) as FullUrl,
XFORCE_WAC_APPLICATION(FullUrl) as ApplicationName,
XFORCE_URL_CATEGORY(FullUrl) as Category from events where
REFERENCESETCONTAINS(‘ApprovedApps’, ApplicationName) = ‘false’ and
eventdirection = ‘L2R’ and Category = ‘Web Storage’ group by
ApplicationName,Category last 7 days
```

When we executed this query via QRadar ariel REST API, we got the following JSON data that shows the list of unapproved cloud storage applications in use:

```
{
  "events": [
    {
      "FullUrl": "www.dropbox.com/files/c/a/apps/windows/private.txt",
      "ApplicationName": "Dropbox",
      "Category": "Web Storage"
    },
    {
      "FullUrl": " mozy.co.uk/product/online-backup",
      "ApplicationName": "mozy",
      "Category": "Web Storage"
    }
  ]
}
```

We can also use this function to identify risky application usage by filtering our results on high risk categories such as Malware, Phishing and BotNet:

```
select CONCAT(UrlHost + ‘/’ + UrlPath) as FullUrl,
XFORCE_WAC_APPLICATION(FullUrl) as ApplicationName,
XFORCE_URL_CATEGORY(FullUrl) as Category from events where
REFERENCESETCONTAINS(‘ApprovedApps’, ApplicationName) = ‘false’ and
eventdirection = ‘L2R’ and ( Category = ‘Malware’ or Category =
‘BotNet Command and Control Server’ or Category = ‘Phishing URLs’ by
ApplicationName,Category last 7 days
```

Note: See the X-Force Exchange FAQ for the list of category names that can be returned by XFORCE_URL_CATEGORY AQL function.

References

[1] https://securityintelligence.com/to-the-cloud-whether-its-allowed-or-not/

[2] http://www.gallup.com/poll/183074/millennials-trusting-safety-personal-information.aspx%C2%A02

[3] http://www.gartner.com/smarterwithgartner/dont-let-shadow-it-put-your-business-at-risk/