UDaaS: A Cloud-based URL-Deduplication-as-a-Service for Big Datasets

Shams Zawoad, Ragib Hasan, Gary Warner, and Anthony Skjellum*
{zawoad, ragib, gar}@cis.uab.edu, skjellum@auburn.edu
Department of Computer and Information Sciences
University of Alabama at Birmingham, AL 35294, USA
*Department of Computer Science and Software Engineering
Auburn University, AL 36849, USA

***Abstract*—Since the number of potential malicious URLs from diverse sources is large, URL deduplication is needed for the efficient identification of malicious websites. URL Deduplication-as-a-Service (*UDaaS*) was developed to help a URL analyst to deploy and manage a cloud-based distributed and parallel URL deduplication infrastructure easily; this can improve the performance of malicious websites detection while reducing duplication and quantity of local storage requirements.**

***Index Terms*—URL Deduplication, Cloud, Parallel Architecture**

I. Introduction

Duplication of URLs are one of the key challenges in identifying malicious websites from large datasets of suspect URLs. The Phishing Data Mining Lab of the University of Alabama at Birmingham extracts nearly 10^6 URLs daily from its spam email sources [1]. *bounce.io* [2] collects more than a million of URLs every hour from their spam email sources. However, a significant percentage of these URLs point to the same website. Therefore, URL deduplication can greatly reduce the cost of malicious website analysis in terms of time, storage, and cost. On the other hand, identifying unique websites from a big URL dataset by analyzing the contents is nearly an impossible tasks using limited, fixed infrastructures.

A cloud-based parallel architecture can make possible the deduplication URLs from a large number of URLs in a reasonably short period of time. Using the elastic nature of the cloud and pay-as-you-go service, with low cost, we can scale up such infrastructure based on the volume of URLs. Moreover, by changing the IP address of the cloud VMs, we can protect our URL fetcher machines from being reverse blacklisted [1]. In this demo, we present *UDaaS*, which can be used to deploy and manage such cloud-based distributed and parallel URL deduplication infrastructure easily.

II. The UDaaS System

A. Overview:

UDaaS was developed based on the architecture proposed in [2]. Whenever a new URL is issued to an Amazon SQS queue, an Amazon Elastic Computing Cloud (EC2) instance reads the URL from the queue and fetches the content at the URL. After fetching the content, the instance generates MD5 hashes of the fetched files and compares them with the previously identified websites using a Bloom filter-based matching algorithm [2]. We construct an Amazon Machine Image (AMI) including all the required software, configurations, and our Java-based application to fetch, analyze, and upload the content of unique websites. The end-user utilizes a Java-based desktop application to create instances using the AMI and manage the operation of the system. There are four major modules in this application: Queue Manager, Instance Manager, URL Source Manager, and Configuration Manager. Figure 1 presents an overview of *UDaaS* operation.

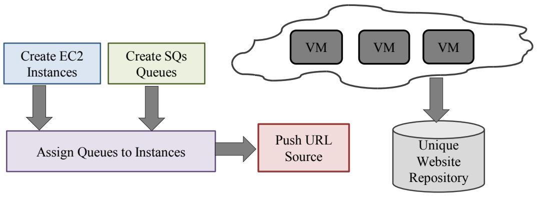


Fig. 1: Overview of *UDaaS* Operation

B. Queue Manager:

The queue manager panel is presented in Figure 2, which provides the following features:

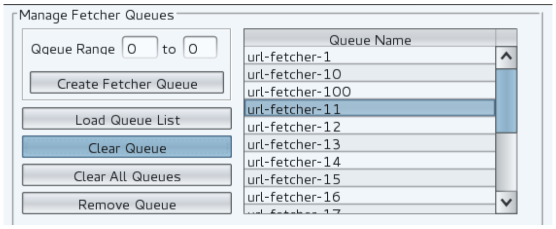
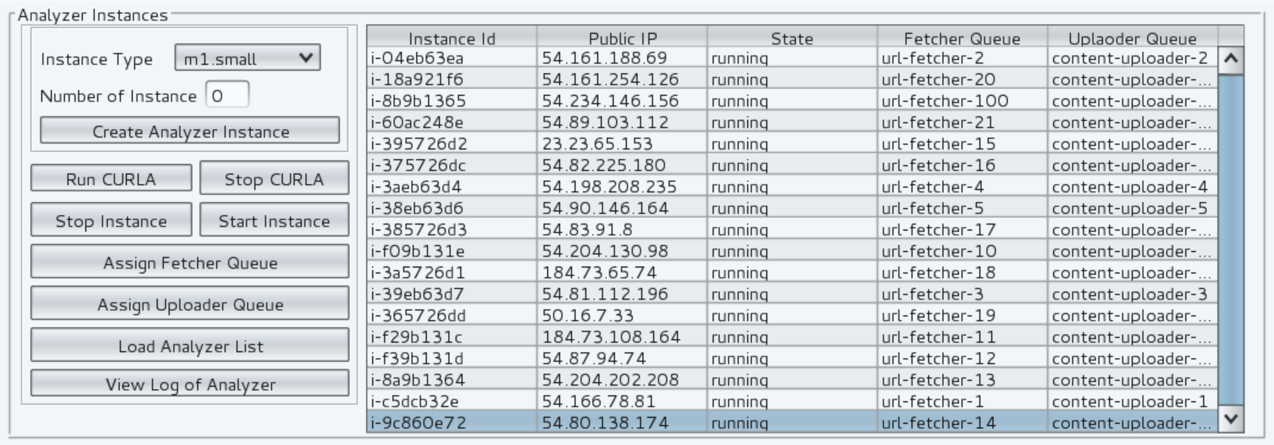


Fig. 2: Fetcher Queue Manager

- *Create Queues:* This feature is used at the system initialization step to create any number of fetcher or uploader queues depending on the required number of instances. This is also used later, when we need to add new instance(s) on the fly.
- *Clear Queues:* This feature is required to re-initiate one or more instances by clearing the messages stored in the associated fetcher and uploader queues.
- *Remove Queues:* To troubleshoot and to relaunch a new instance or the whole system, we may need to remove the existing queue(s). This feature supports this requirement.

[1] https://cis.uab.edu/labs/computer-forensics-research/uab-spam-data-mine/
[2] https://bounce.io/



Instance Type: m1.small

Number of Instance: 0

Instance Id	Public IP	State	Fetcher Queue	Uplaoder Queue
i-04eb63ea	54.161.188.69	running	url-fetcher-2	content-uploader-2
i-18a921f6	54.161.254.126	running	url-fetcher-20	content-uploader-...
i-8b9b1365	54.234.146.156	running	url-fetcher-100	content-uploader-...
i-60ac248e	54.89.103.112	running	url-fetcher-21	content-uploader-...
i-395726d2	23.23.65.153	running	url-fetcher-15	content-uploader-...
i-375726dc	54.82.225.180	running	url-fetcher-16	content-uploader-...
i-3aeb63d4	54.198.208.235	running	url-fetcher-4	content-uploader-4
i-38eb63d6	54.90.146.164	running	url-fetcher-5	content-uploader-5
i-385726d3	54.83.91.8	running	url-fetcher-17	content-uploader-...
i-f09b131e	54.204.130.98	running	url-fetcher-10	content-uploader-...
i-3a5726d1	184.73.65.74	running	url-fetcher-18	content-uploader-...
i-39eb63d7	54.81.112.196	running	url-fetcher-3	content-uploader-3
i-365726dd	50.16.7.33	running	url-fetcher-19	content-uploader-...
i-f29b131c	184.73.108.164	running	url-fetcher-11	content-uploader-...
i-f39b131d	54.87.94.74	running	url-fetcher-12	content-uploader-...
i-8a9b1364	54.204.202.208	running	url-fetcher-13	content-uploader-...
i-c5dcb32e	54.166.78.81	running	url-fetcher-1	content-uploader-1
i-9c860e72	54.80.138.174	running	url-fetcher-14	content-uploader-...

Fig. 3: Analyzer Instance Manager

C. EC2 Instance Manager:

The control panel of this module is illustrated in Figure 3, which supports the following notable features:

• *Create instances:* This features provides support for creating any required number of instances using our pre-build AMI and chosen instance type (e.g., m1.small, m3.large, etc.).

• *Start/Stop Instance:* User can start and stop selected instances from a list of already created instances.

• *Assign Fetcher and Uploader Queue:* functionality, we can attach a previously created fetcher and uploader queue to an analyzer instance.

• *Start/Stop Analyzing Task:* We can start/stop the URL fetching and analyzing program in EC2 instances. After execution, the fetcher and uploader components listen to their associated fetcher and uploader queues that are selected in the previous step.

• *Log Viewer:* This features provides an interface to the end-users to view the logs of the program running in EC2 instances.

D. URL Source Manager:

End-users can push URLs to the system in two ways: by uploading a file that contains the URLs, or by using an Amazon S3 object that contains the URLs. For Amazon S3-based URL source, we prepared another AMI, which downloads contents from S3 and push the URLs to the fetcher queues.

E. Configuration Manager

This module is used to configure the database server that preserves the unique websites and also to configure the FTP server that receives unique website's content from the EC2 instances. Currently, we support PostgreSQL and MySQL database management system. All the EC2 instances will use these configurations while executing their operations.

III. Application Area

Detection of Phishing Website: Content-based phishing website detection schemes [3], [4], [5] require download of the content of the possible phishing websites and running various phishing detection techniques on the downloaded content. Duplicate URLs slow down the whole process by consuming unwanted computing resources while introducing massive amounts of storage wastage. The URL deduplication service provided by *UDaaS* can be applied to any phishing detection scheme, which can improve the rate of phishing website detection by focusing only on unique websites.

Detection of Fraudulent Goods' and Services' Website: A significant portion of websites advertised through spam emails are the counterfeit websites for goods/services. These websites sell branded goods, which they are not authorized to sell or send illegal spam campaign of fake insurance. The organization that owns the original brand needs to know which counterfeit websites sell their products, so that they can take appropriate legal steps against those websites.

IV. Conclusion

UDaaS can be used as a powerful tool in academia and in industry to deploy a highly scalable and distributed cloud-based infrastructure to deduplicate URLs from a big URL dataset. Utilizing *UDaaS* can thus help to improve the rate of detection of phishing and other types of counterfeit websites by providing unique websites to the analyst.

Acknowledgments

This research was supported by the National Science Foundation Career Award CNS-1351038, a Google Faculty Research Award, and the Department of Homeland Security Grant FA8750- 12-2-025.

References

[1] E. Ferguson, J. Weber, and R. Hasan, "Cloud based content fetching: Using cloud infrastructure to obfuscate phishing scam analysis," in *IEEE SERVICES,2012*, pp. 255–261.

[2] S. Zawoad, R. Hasan, M. M. Haque, and G. Warner, "Curla: Cloud-based spam url analyzer for very large datasets," in *IEEE Cloud*, 2014.

[3] B. Wardman, T. Stallings, G. Warner, and A. Skjellum, "High-performance content-based phishing attack detection," in *eCrime Researchers Summit, 2011*. IEEE, 2011, pp. 1–9.

[4] C. Whittaker, B. Ryner, and M. Nazif, "Large-scale automatic classification of phishing pages." in *NDSS*, 2010.

[5] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: a content-based approach to detecting phishing web sites," in *WWW 2007*. ACM, 2007, pp. 639–648.