CVSS: A Cloud Based Visual Surveillance System

Lei Zhou, Yun Shu, Jian Yu, Wei Qi Yan
Department of Computer Science
Auckland University of Technology, 1010 New Zealand

ABSTRACT

A large amount of surveillance videos and images need sufficient storage. In this paper, an architecture of cloud based surveillance system and its modules will be designed, the Cloud-based Visual Surveillance System (CVSS) will be implemented on private cloud using Virtual Machine (VM). The users are able to link their cameras to the CVSS system so that the goal of this design can be achieved. Our CVSS system is able to push notification messages of captured videos to receivers, our users could receive a surveillance video along with its events. The CVSS system fully makes use of the merits of cloud computing which make it more advanced as stated in the evaluation section of this paper. The contributions of this paper are to implement the CVSS system with: (1) video stream input, (2) intelligent visual surveillance, (3) real-time video transcoding and storage, (4) message push and media streaming output.

Keywords: Intelligent surveillance system; computer vision integrated system; cloud storage; push notification

1. INTRODUCTION

Visual surveillance is an application of security engineering, the functions of a surveillance system include camera deployment and calibration, video and image recording, surveillance data transmission, etc. Traditional visual surveillance refers to analog monitoring, its infrastructure includes a front-side camera, transmission cable, and visual monitoring platform (Hossain, et al., 2016). Because traditional surveillance systems are lack of intelligent computing, therefore it is "passive" or "not smart" (Zhao, Cui & Zhang, 2012). Cloud computing is a storage oriented system that is growing rapidly in recent years. Video and image storage shows its challenges with the substantial requirement of infrastructure because visual surveillance needs storage facilities like big data that may be costly to any users. Also, once the storage disks are full or damaged, the huge data will be in risk, and thus the backup is absolutely needed (Hossain, et al., 2012). With timely data backup, users can easily access the data in anytime without worrying about cloud facilities (Bogardi-Meszoly, et al., 2006). Visual Surveillance as a Service (VSaaS) (Kim, et al., 2009) follows the cloud computing paradigm SaaS which has outperformed than other services of traditional surveillance systems.

The VSaaS is primarily driven by numerous pivotal factors such as dynamic technology, cyber security, remote access, etc. A Cloud based Visual Surveillance System (CVSS) allows any users to benefit from upfront capital costs (Qi & Yu, 2006). The surveillance system therefore lessens the resources and human workload. The purposes of this paper are to design and implement the CVSS system that has better functions, satisfactory performance and friendly user experience than those existing ones (Jiang, et al., 2012). As visual surveillance requires sufficient space to store the big surveillance data, we think the first research problem of this project is how to dynamically allocate enough space to deposit these videos and images from surveillance sensors.

We believe pushing notifications is an important feature of a cloud based system. Hence, our CVSS system could deliver any latest events to the user's terminals in real time so as to achieve surveillance alarm making timely. In order to make the CVSS more intelligent, we need integrate all components into the system. We set up the system with the functions such as face recognition, motion detection, license plate recognition, etc. The rest of this paper is organized as follows. Section 2 will present our system design, we will describe the implementation in Section 3 while Section 4 illustrates our tests and analysis, we will conclude this paper and depict our future work in Section 5.

2. SYSTEM DESIGN

The requirements to develop the CVSS system are necessary, especially for those cloud devices and Apps. In the CVSS system, our users can watch and get videos from anywhere in the world (Wo, et al.,2012). By using Network Video Analytics (NVA) module, when a surveillance event is captured, the users could get the push notification that includes event entities such as "who", "where", "what", "why", "when", etc. (Zheng & S, 2009) The users are able to access the surveillance video footages associated with the event via Cloud Apps.

Our CVSS system consists of two parts. In the client side, our users are able to control the CVSS system. In the server side, Microsoft Hyper-V will create a Virtual Machine (VM). We deploy a program running environment in Hyper-V based on Windows Server 2012 R2. Finally, we export the Microsoft Windows server in a VM file to the private cloud (Bogardi-Meszoly, et al., 2006).

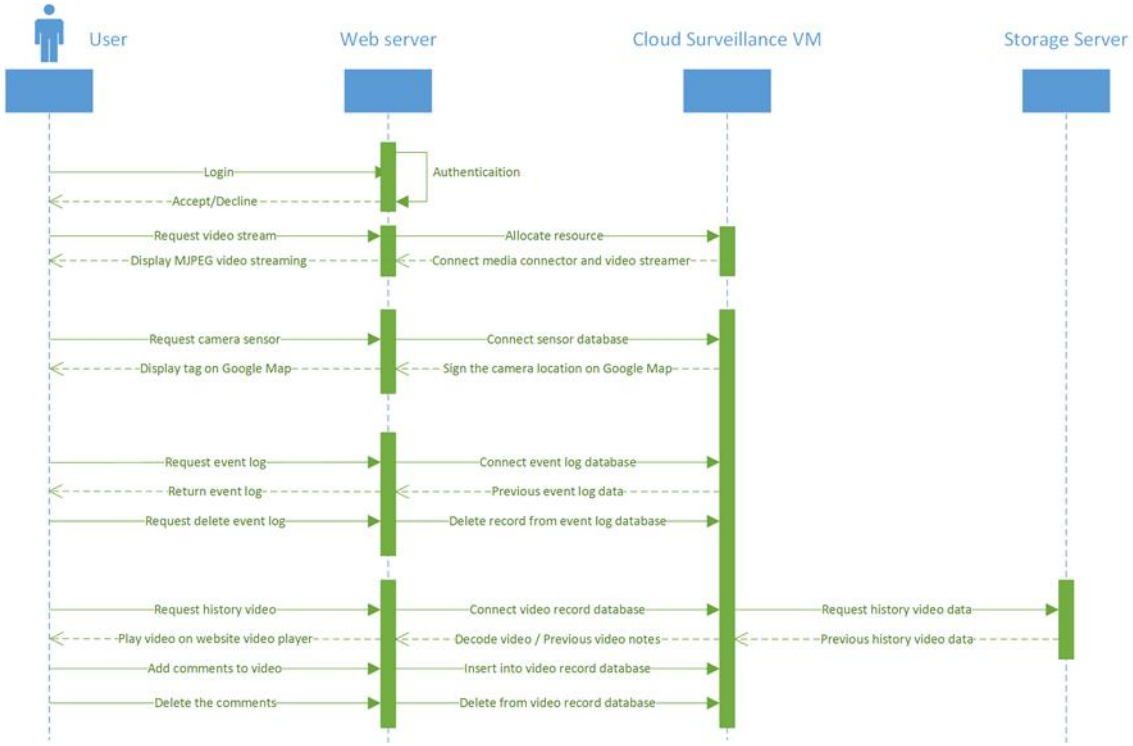

Figure 1 The flowchart of user operations

With the development of cloud computing, the traditional Client/Server (C/S) architecture is difficult to meet the requirements such as visual information sharing, notification push, etc. Brower / Server (B/S) structure is based on HTTP as the transmission protocol in use of WWW technology, however cloud users access the powerful features via the SaaS while reducing the pressure of traffic jam in the client side. Our proposed CVSS system adopts the cloud based architecture, the user access and system control are carried out by using the well-designed PaaS, SaaS, and IaaS (Tekeoglu & Tosun, 2015).

Our CVSS system utilizes ASP.NET to deploy the B/S model based on three tiers: client layer, web layer and data layer (Xiong, et al., 2014). The client layer takes use of the SaaS to access the site for the purpose of Human Computer Interaction (HCI) (Prati, et al., 2013). The web layer lies between the client layer and the data layer accrossing the web server and the application server. The data layer is the base tier of the CVSS system, which was designed especially for huge data storage. ASP.NET supports three types of authentication (Karimaa, 2011), namely, windows authentication, forms authentication, and passport authentication. Our CVSS system applies cookies as the authentication mechanism which is based on form authentication. A cookie is a small piece of binary information which responds to the user requests for the purpose of authentication. In the CVSS login page, once a user submits information package to the cloud server, ASP.NET will send a new cookie to confirm the effectiveness of an identity ticket. If a user passes the authentication, a new record will be inserted into the database. While a user is in the operating process, the recorder will update the database timely shown as Figure 1. In CVSSS, we have designed the features like,

- Real-time surveillance video stream. The camera SDK is based on standard: Open Network Video Interface Forum (ONVIF). Therefore, it supports connections between any ONVIF standard devices. The camera SDK provides the function of video streaming from a camera in real time. We utilize Real-Time Streaming Protocol (RTSP) to create a connection between a camera and the surveillance VM.
- Locating a surveillance camera. An IP camera has its location. When an authorized user requires viewing the location of an IP camera, the system will read the latitude and longitude of the IP camera from the sensor database. Subsequently, the GPS location of the IP camera will be set as a tag on Google Map.
- Surveillance event log. Our CVSS system is able to provide data with regarding to the events detected by the CVSS Computer Vision (CV) module, our users can search and view the corresponding video footages.
- Comments on the surveillance events. The CVSS system allows our users to enter their comments on surveillance events. The users are able to add multiple comments on each surveillance event which usually are thought as evidences from witnesses.

3. SYSTEM IMPLEMENTATION

The current visual surveillance system faces tremendous challenges. Transferring and storing High Definition (HD) surveillance videos have resulted in issues of resources shortages related to network bandwidth and storage capacity, etc. The proposed system CVSS of this paper is able to overcome these shortcomings. In this section, we will implement our proposed CVSS system and detail the procedure of implementation.

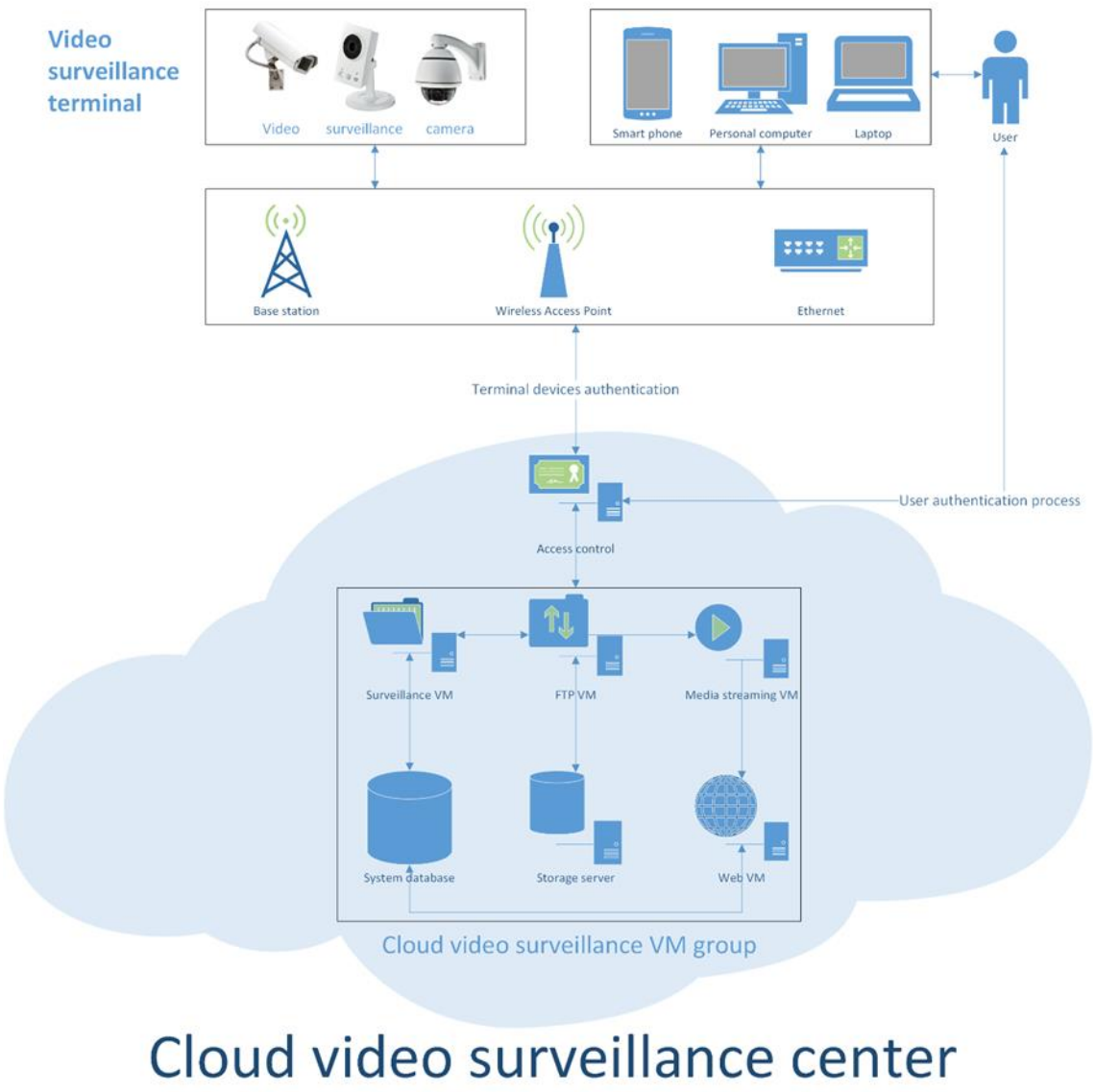

Figure 2 The flowchart of user operations

In Figure 2, the CVSS system includes video surveillance terminals, cloud server, and CVSS VM group. An IP camera as the surveillance sensor is used for capturing surveillance videos through networks (Base Station, Wireless AP or Ethernet). An access server has been used for authenticating the users and surveillance terminals or receivers, it is also used to record user actions and camera status. The CVSS VM group includes surveillance VM, FTP VM, Media streaming VM, Web VM, storage server and cloud databases which store user information, surveillance data and its metadata.

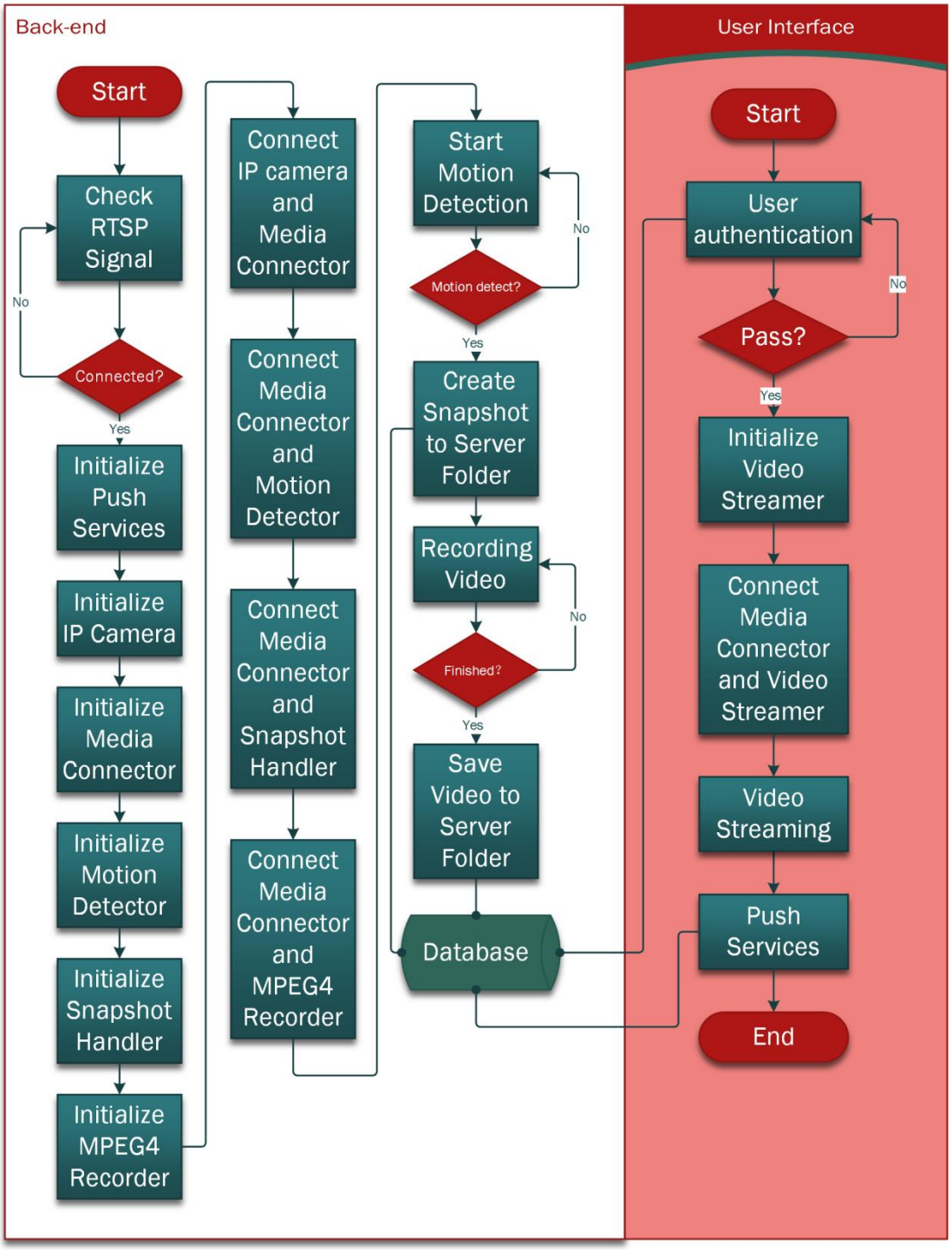

Figure 3 The diagram of our CVSS surveillance system

The surveillance VM is used to hold NVA module and CV module, etc. When the CV module is activated, the video recorder will automatically start taking and saving the surveillance videos. Finally, the videos through FTP protocol will be transferred to FTP VM and stored them in the cloud based storage server. In CVSS, the VM transfers recorded data through FTP, the receiver

is a cloud based storage server. On the FTP server, the first iSCSI service using iSCSI initiator is to discover and connect the target disk iSCSI LUN from the storage server, and the physical path of root directory of the FTP site is the target disk. The storage server is responsible for supplying data storage space. The storage pool is divided into two parts which include data volume and iSCSI LUN. We create a routine for backing up iSCSI storage per week. The media streaming server provides surveillance videos for the web server. The purpose of this media streaming server is to reduce the web VM resource consumption and improve response time (Luo, et al., 2011). The CVSS database is deployed on surveillance VM, it is mainly used to store surveillance videos, the user authentication database is deployed in Microsoft Azure. The reason why the CVSS database was deployed in a surveillance VM is that it reduces the delay for data transmissions between the databases and those surveillance cameras.

The CVSS surveillance system is mainly used to deal with the incoming video streams from the cameras. Figure 3 is a flowchart to show how the CVSS surveillance system works. It has two parts, namely, foreground and back-end. The foreground of the surveillance system is to provide a user interface, meanwhile the back-end is used to coordinate the operations between various modules of the CVSS.

In Figure 3, we see the first column reveals initial procedure of each module. Before the CVSS is initialized, the camera must have passed a series of verifications. It means the system is able to receive video signals from an IP camera.

The second column shows the connections between various CVSS modules. In Figure 4, we see the media connector plays a significant role which provides the interface for linking a number of modules together. After the connection module of an IP camera is attached to the media connector, the media connector will pop up the video interface for the NVA, snapshot handler, video recorder and video streamer in real time.

The last column shows how the NVA module interacts with the core module of CVSS which includes snapshot handler, video recording and push notification service. The NVA is based on events and controls the core module so as to achieve the full functionalities of the CVSS system.

The CVSS storage system has three components: iSCSI LUN (storage), FTP Virtual Machine (VM) and storage management system (Zhang, Lin & Liu, 2014). The iSCSI is based on Internet Protocol (IP) and SCSI-3 protocol which adopts TCP as the transport protocol, the port number is 3260. The hard disk is allocated at the storage server QNAP TS-253 PRO.

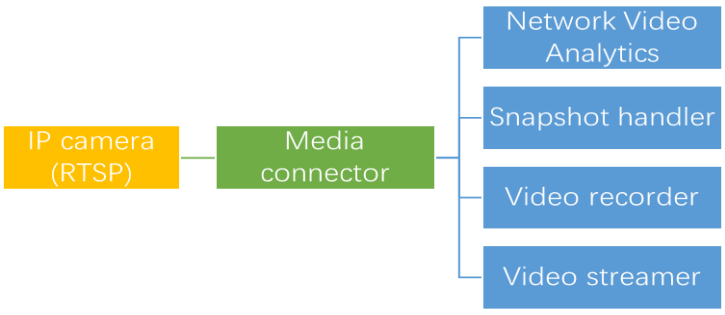

Figure 4 The connection bridge of our CVSS system

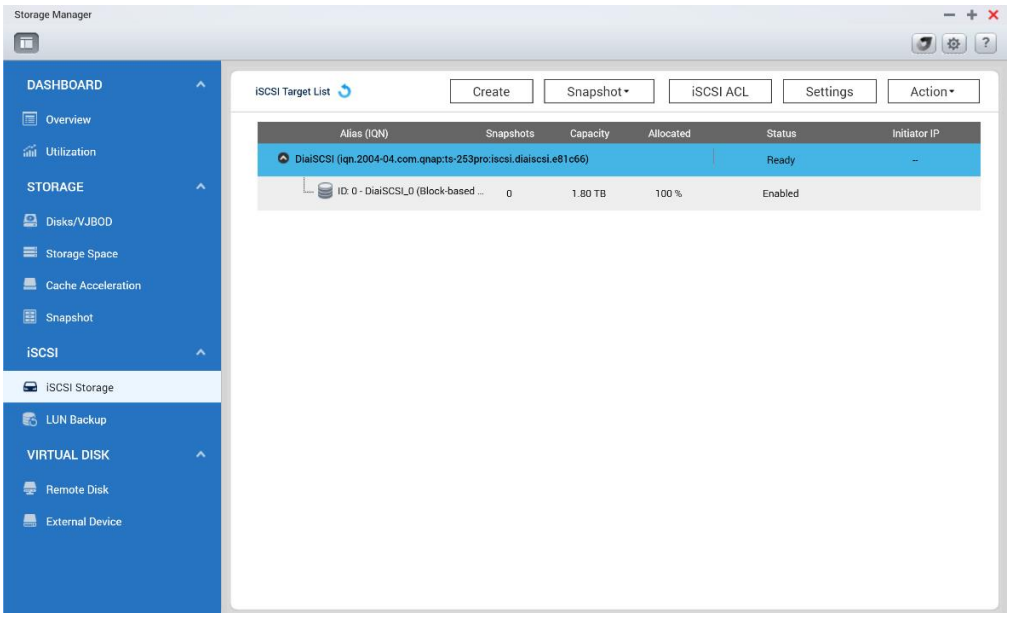

Figure 5 The iSCSI LUN of our CVSS system

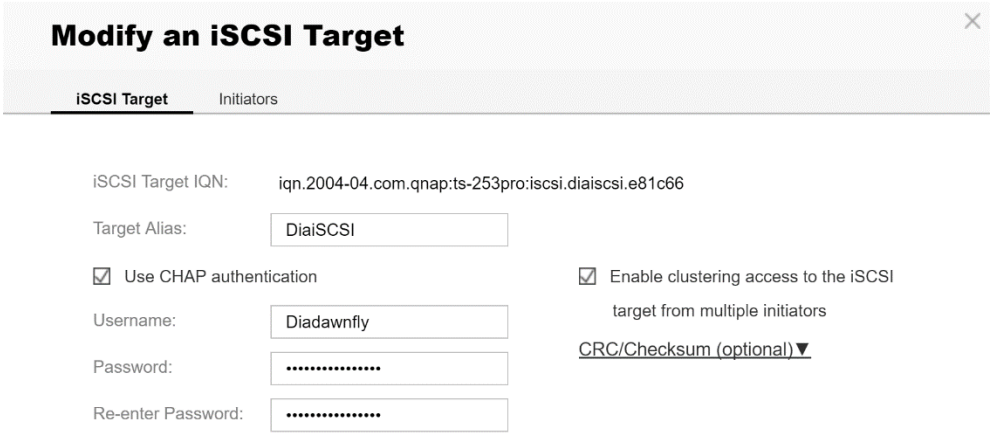

Figure 6 The iSCSI target CHAP authentication

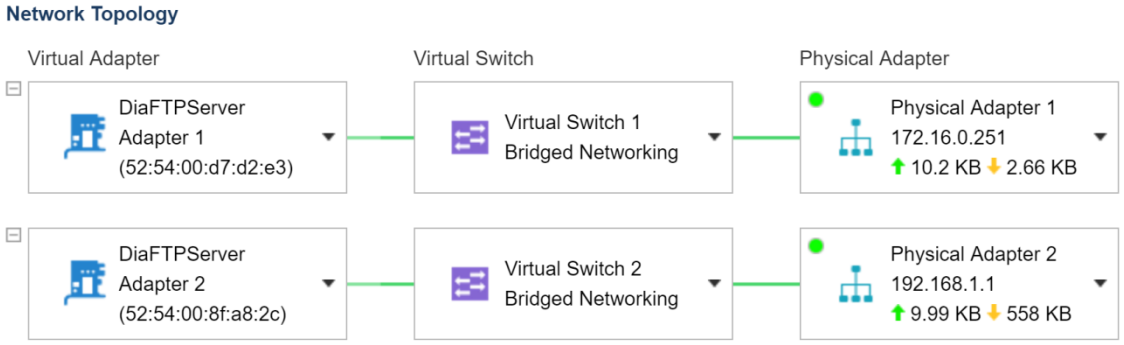

Figure 7 The network topology of FTP VM

In Figure 5, we see that the capacity of iSCSI storage has 1.8TB according to the data capacity, the iSCSI has expansion capacity. As shown in Figure 6, the iSCSI target using CHAP authentication includes username and password, clustering access from multiple initiators is able

to be accepted. The user's IQN must be added to initiator connection list so as to ensure the data security. Otherwise, the connection for iSCSI target will be declined. The backup plan for iSCSI LUN has been employed once a week, the backup files will be stored in both storage server and surveillance server. The FTP VM is running on the storage server. In Figure 7, there are two network adapters applied to FTP VM, VM connects public network (Internet) via Switch 1. Switch 2 is connected to a private network that means there is no Internet access, only four devices connect with this network including storage server, FTP VM, Camera 1 and Camera 2. CVSS uses FTP protocol through FTP VM to transfer surveillance data and store them in CVSS iSCSI LUN (Qi & Yu, 2006). Our users manage the surveillance data via WebDAV, FTP or the storage management system (Zhao, Cui & Zhang, 2012).

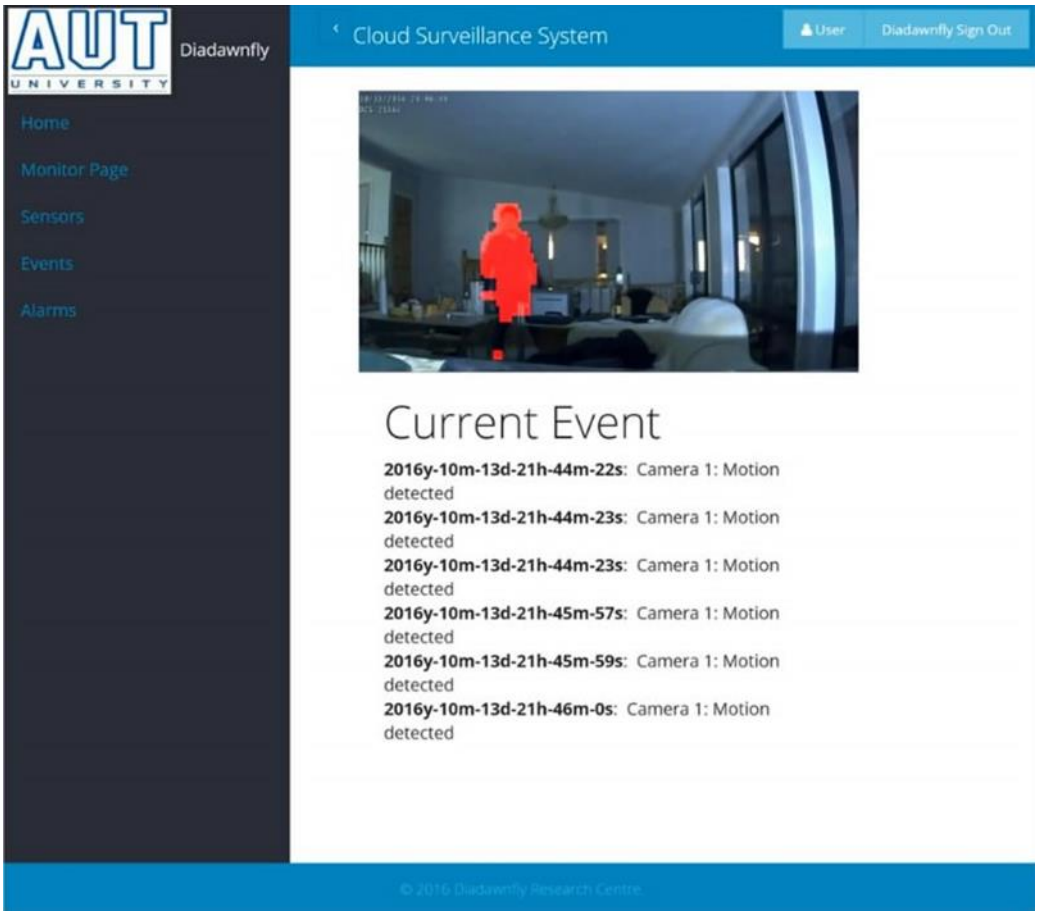

Figure 8 The CV module with motion detection

4. TESTS AND ANALYSIS

In this section, our CVSS running-time environment includes two parts i.e. hardware and software. The types of hardware are grouped into three categories: IP camera, network infrastructure and cloud server. The CVSS system was developed using Microsoft Visual Studio 2015, the operating environment needs Microsoft IIS, MySQL and .Net Framework. In following

sections, we will explain the main aspects of our CVSS system. In this section, we will also test the CVSS system. The testing is to ensure that the system can efficiently and correctly respond to our user's requests. Moreover, the GUI snapshot of each module is timely provided to demonstrate availability of the whole CVSS system. For the tests, we are going to verify the system performance and use Apache JMeter as the benchmark (Tekeoglu & Tosun, 2015).

The cloud for CVSS is divided into internal network and external network in public which is mainly used for explaining components of the private cloud and providing Internet services (Goyal, 2014). The router has been mapped onto the system ports, it is possible to access our proposed CVSS system.

4.1 Demonstrations

In this section, the demonstration of our CVSS system will be shown, each module will be tested to affirm whether it is properly being run as well. The CVSS system is accompanied with the models like user authentication, sensor deployment, surveillance monitoring, event viewing and alarm making.

When a user need access the CVSS, (s)he must sign in the system. The user authentication database is allocated in Microsoft Azure (a public cloud). When a user leaves the CVSS system, (s)he may click the button at the upper right corner and log out.

If a user passes the authentication, the overview page will be automatically popped up. The user will receive all push messages from this SignalR hub in real time, no matter which camera detects an event. If the user hopes to view the monitoring, (s)he can click the top-left button to hide the navigation bar.

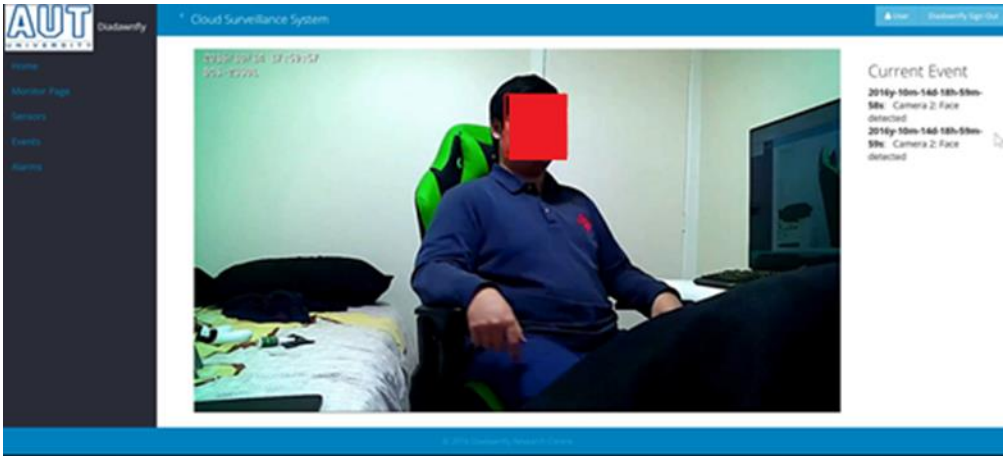

Figure 9 The face detection module of our CVSS system

When a user clicks the button on the right side, the user will enter the specified state of camera monitoring, the user can see the interface of real-time visual monitoring. Figure 8 shows the Camera 1 with motion detection module and Figure 9 shows Camera 2 with face detection module, respectively. When a user needs to view the camera location, the user can click the

sensor button on the navigation bar to enter the sensor interface. If the user selects a camera ID, the Google Map will display location of the sensor (Zhang, et al., 2010).

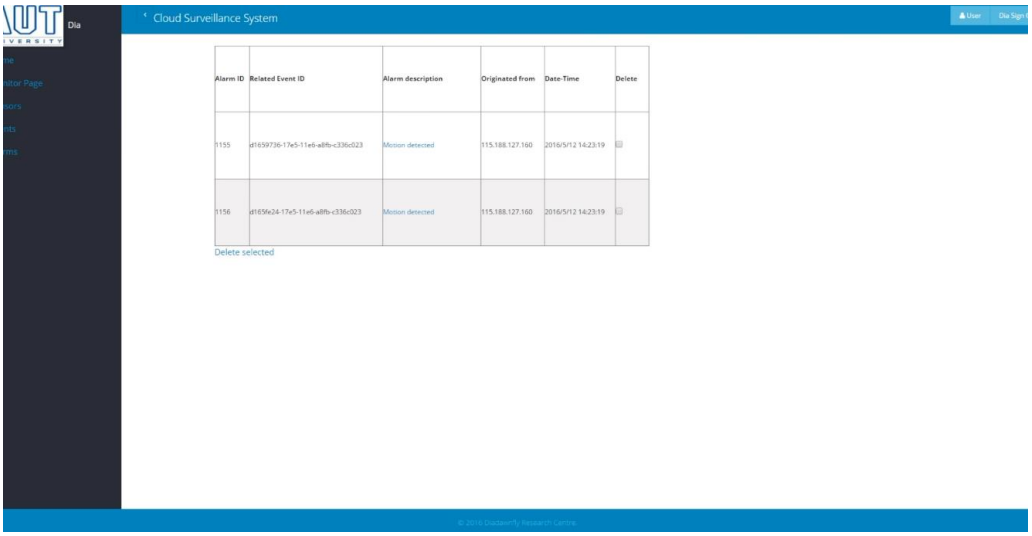

Figure 10 The alarming module of CVSS system

A user can access the alarm making module through clicking the alarm button on the left to view all the list of alarms shown in Figure 10. The user can select the alarm records and delete the selected entries in batch. Meanwhile the event database will be updated correspondingly.

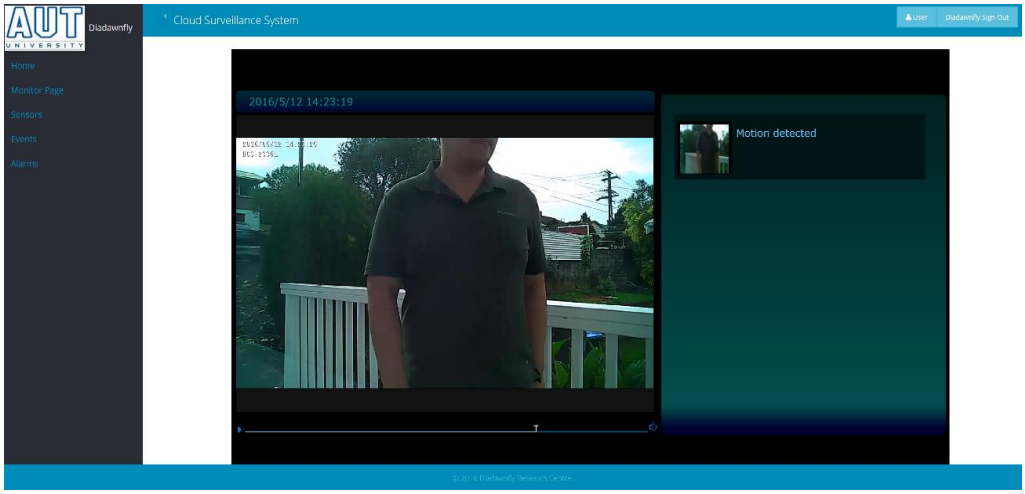

Figure 11 The CVSS alarm-marking module

When a user clicks the corresponding alarm description button which links to the alarm making module, the user can view the corresponding events from the CVSS event module. In Figure 11, the surveillance videos associated with the event will be displayed.

4.2 Evaluations

In our tests, we use Table 1 to evaluate system response to each action. The parameter *Min* stands for the minimum response time of the CVSS system whilst the *Max* refers to the maximum one. The *Average* represents the average response time of the means in the tests. Total *error* means the times in total the CVSS system could not respond to a request when an error is occurred in the connections. The *Throughput* expresses the total time required, the unit *KB/s* is to describe the throughput.

Our first test was conducted on the development machine which is called Dia Razer Profession Computer, CPU Intel i7-5930K 3.5GHz 6 Core 12 Thread, 32GB DDR4 2400MHz RAM, ASUS PCE-AC68 network card, NVMe SSD 512GB, NVIDIA GTX TITAN X 2xSLI (12GB GDDR5) graphic cards. As a top-of-the-line computer, the results are perfect. In Table 1, we see that the average access time is 1.0 second. The highest throughput rate is up to 19231 *KB/s* (153848 *Kbps*). In Table 2, the client also is Dia Razer Professional PC, the cloud server we used is private Cloud QNAP TVS 682. Owning to the optimized configuration of the cloud server and advanced network equipment, the results of the tests are acceptable.

Table 1 CVSS benchmark in local PC

Experimental number		1	2	3	4	5	6	7	8
Parameters	Thread (user) number	5	10	20	40	80	200	500	1000
	Ramp-UP Period (s)	10	10	10	10	10	10	10	10
	Loop Count	5	5	5	5	5	5	5	5
	Element	6	6	6	6	6	6	6	6
Results (Total)	Sample	150	300	600	1200	2400	6000	15000	30000
	Minimum(ms)	1	1	1	0	0	0	0	0
	Maximum(ms)	10	4	6	20	17	26	33	194
	Average(ms)	1	1	1	1	1	1	1	1
	Total Error	0	0	0	0	0	0	0	0
	Error Rate	0%	0%	0%	0%	0%	0%	0%	0%
	Throughput(s)	18.6	33.1	62.8	122.3	241.7	597.1	1476.8	2878.8
	KB/s	124.5	221.2	419.3	817.1	1614.4	3986.6	9865.4	19231

4.3 Analysis

In this paper, we have two groups of tests. In the first, we ran the CVSS system based on our local PC. In the second, the CVSS system was being executed on our private cloud (Goyal, 2014), the testing PC is our local computer.

We compare two groups of results shown in Table 1 and Table 2, we found that the average response time of the CVSS system is longer. We also find that the testing results from local server are much better than those from the private cloud. We see that the maximum throughput rate is close to 150Mbps, that means the wireless network protocol of the CVSS system should select 802.11n (>300Mbps) or 802.11ac, the 0 error rate shows the stability of the CVSS system in the running time. Throughout the results, we see that the high-performance hardware, the support of running the CVSS system, provides an effective and efficient running environment.

5. CONCLUSION

Our CVSS system is able to integrate various surveillance modules into a cloud based running-time environment which associates visual monitoring and remote control. First of all, the system pushes notifications, the CVSS users access the events through computers, tablets, mobiles and other network-enabled devices timely. Moreover, the storage system adopts iSCSI LUN + ownCloud solution that allows users to view the archived event logs with networked devices. Meanwhile, this solution also guarantees data security on the storage server because our users are able to directly access the CVSS system only after authorization and authentication. On the other hand, the solution reduces the traffic jam of the surveillance server. Our users are able to obtain surveillance data provided by ownCloud or the client (including PC and mobile phones). If PC client software provided by ownCloud is utilized, video surveillance data is able to be backed up remotely.

In future, we will also take our visual surveillance system on hybrid cloud into consideration and develop a Hybrid CVSS system (H-CVSS). We will use the Hybrid Cloud as the solution to create a bigger cloud storage

Table 2 CVSS benchmark in private cloud

Experimental number		1	2	3	4	5	6	7	8
Parameters	Thread (user) number	5	10	20	40	80	200	500	1000
	Ramp-UP Period (s)	10	10	10	10	10	10	10	10
	Loop Count	5	5	5	5	5	5	5	5
	Element	6	6	6	6	6	6	6	6
Results (Total)	Sample	150	300	600	1200	2400	6000	15000	30000
	Minimum(ms)	5	5	5	4	4	4	4	5
	Maximum(ms)	14	16	22	28	20	33	55	251
	Average(ms)	6	6	6	6	6	7	7	15
	Total Error	0	0	0	0	0	0	0	0
	Error Rate	0%	0%	0%	0%	0%	0%	0%	0%
	Throughput(s)	18.3	32.6	61.8	120.7	238	586.7	1456	2827.7
	KB/s	122.2	217.6	413.2	806.5	1590.5	3920.7	9727.4	18854

REFERENCES

Bogardi-Meszoly, A., Levendovszky, T., & Charaf, H. (2006). Performance factors in ASP. NET web applications with limited queue models. In International Conference on Intelligent Engineering Systems (pp. 253-257). IEEE.

Goyal, S. (2014). Public vs private vs hybrid vs community-cloud computing: A critical review. International Journal of Computer Network and Information Security, 6(3), 20.

Hossain, M. A., & Song, B. (2016). Efficient Resource Management for Cloud-enabled Video Surveillance over Next Generation Network. Mobile Networks and Applications, 21(5), 806-821.

Hossain, M. S., Hassan, M. M., Al Qurishi, M., & Alghamdi, A. (2012). Resource allocation for service composition in cloud-based video surveillance platform. In IEEE International Conference on Multimedia and Expo Workshops (pp. 408-412). IEEE.

Jiang, J., Sekar, V., & Zhang, H. (2012). Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. In Proceedings of International Conference on Emerging Networking Experiments and Technologies (pp. 97-108). ACM.

Karimaa, A. (2011). Video surveillance in the cloud: Dependability analysis. In Proceedings of the fourth international conference on dependability, DEPEND (pp. 92-95).

Kim, S., Nam, Y., Kim, J., & Cho, W. D. (2009). ISS: intelligent surveillance system using autonomous multiple cameras. In International Conference on Ubiquitous Information Technologies & Applications (pp. 1-6). IEEE.

Luo, J. Z., Jin, J. H., Song, A. B., & Dong, F. (2011). Cloud computing: architecture and key technologies. Journal of China Institute of Communications, 32(7), 3-21.

Prati, A., Vezzani, R., Fornaciari, M., & Cucchiara, R. (2013). Intelligent video surveillance as a service. In Intelligent Multimedia Surveillance (pp. 1-16). Springer Berlin Heidelberg.

Qi, W., & Yu, Z. (2006). Design and Implementation of Digital Video Surveillance System Based on B/S Structure. Computer Engineering, 19.

Tekeoglu, A., & Tosun, A. S. (2015). Investigating Security and Privacy of a Cloud-Based Wireless IP Camera: NetCam. In International Conference on Computer Communication and Networks (pp. 1-6). IEEE.

Wo, T., Hu, C., Li, J., & Huai, J. (2012). Netros: A virtual computing environment towards instant service of network software. In International Conference on Semantics, Knowledge and Grids (SKG) (pp. 24-31). IEEE.

Xiong, Y. H., Wan, S. Y., He, Y., & Su, D. (2014). Design and implementation of a prototype cloud video surveillance system. Journal of Advanced Computational Intelligence and Intelligent Informatics, 18(1), 40-47.

Zhang, H., Li, M., Chen, Z., Bao, Z., Huang, Q., & Cai, D. (2010). Land use information release system based on Google Maps API and XML. In Geoinformatics, International Conference on (pp. 1-4). IEEE.

Zhang, S., Lin, Y., & Liu, Q. (2014). Secure and Efficient Video Surveillance in Cloud Computing. In Mobile Ad Hoc and Sensor Systems (MASS), International Conference on (pp. 222-226). IEEE.

Zhao, Z. F., Cui, X. J., & Zhang, H. Q. (2012). Cloud storage technology in video surveillance. In Advanced Materials Research (Vol. 532, pp. 1334-1338). Trans Tech Publications.

Zheng, S. B. (2009). Intelligent video surveillance technology and its application. TV Engineering, 1, 94-96.