Cloud Computing based Forensic Analysis for Collaborative Network Security Management Systems

Zhen Chen, Fuye Han, Junwei Cao, and Shuo Chen
Research Institute of Information Technology
Department of Computer Science & Technologies
Department of Automation
Tsinghua National Laboratory for Information Science and Technology (TNList)
Tsinghua University, Beijing 100084, P. R. China

Abstract - Internet security problems are still a big challenge as there are many security events occurred, such as Internet worms, Spam and phishing attacks etc. Botnet, a well-organized distributed network attack, consists of a large volume of bots, which generates huge volumes of spam or launching Distributed Denial-of-Service (DDoS) attacks to victim hosts. This new emerging botnet attack makes Internet security status even worse. To address these problems, a practical Collaborative Network Security Management System is proposed with well deployed collaborative UTM (Unified Threat Management) and traffic probers. Such distributed security overlay network with a centralized Security Center leverage a Peer-to-Peer communication protocol used in UTM's collaborative module and virtually interconnect them to exchange network events and security rules. Also security functions for UTM are retrofitted to share security rules. In this paper, we propose a design and implementation of cloud based Security Center for network security forensic analysis. We propose to use cloud storage to keep collected traffic data and processing it with cloud computing platform to find the malicious attacks. A workable case, phishing attack forensic analysis is presented and the required computing and storage resources are evaluated based on real trace data. Cloud based Security Center can instruct each collaborative UTM and prober to collect events and raw traffic, sent them back for deep analysis and to generate new security rules. These new security rules are enforced by collaborative UTM and the feedback events of such security rules are also returned to Security Center. By this type of close-loop control, the Collaborative Network Security Management System can identify and address new distributed attacks more quickly and effectively.

Key word: Cloud Computing, Overlay Network, Collaborative Network Security System, Computer forensics, Anti-Botnet, Anti-Phishing, Hadoop File System, Eucalyptus, Amazon Web Service.

1. Introduction and Background

As Internet plays a more and more key role as information infrastructure, e-business and e-pay in Internet is booming due to its convenience and benefits for users. Internet security problems are still a big challenge as there are many security events occurred. The underground economics based on Internet Scam and Fraud is also booming. These attackers initiate more and

more E-crime attacks and abuse, such as Spams, Phishing attack, Internet worms etc. Firewalls, Intrusion Detection System (IDS) and Anti-Virus Gateway are now widely deployed in edge-network to protect end-systems from the attacks. When the malicious attacks have fixed patterns, they can be easily identified and matching these patterns[39-42]. However, sophisticated attacks are distributed over the Internet, and have fewer characteristics and evolved quickly. For example, the Distributed Denial of service (DDoS) contains very few, if any, signatures strings to identify.

Nowadays DDoS attacks are likely launched by a large volume of bots which forms a Botnet controlled by bot master. The bots are commanded to generate attack new victim machine and enlarge botnet. The bots also commanded to conduct other issues such as disseminating spam or launching Distributed Denial-of-Service (DDoS) attacks to victim hosts. To countermeasure botnet, secure overlay is proposed. To prevent such distributed attacks, collaboration is a way need to be taken. Collaborative intrusion detection system is reviewed by researches in [36]. By collaboration, the network security system could realize scalability, teamwork, and has a bigger picture of events in the whole network. With collaboration, an algorithm is presented to improve the alert event's accuracy by aggregate information from different sources in [37]. A similar alert correlation algorithm [38] is put forward which is based on Distributed Hash Tables (DHT).

The Collaborative Network Security Management System (CNSMS) [28] aims to develop a new collaboration system to integrated well deployed UTM such as NetSecu [27]. Such distributed security overlay network coordinated with a centralized Security Center leverage a Peer-to-Peer communication protocol used in UTM's collaborative module and virtually interconnect them to exchange network events and security rules. CNSMS also has a huge output from operation experience, e.g., traffic data collected by multiple sources in different vantage point, operating reports and security events generated from different collaborative UTMs etc. As such data is so huge and not easy to analyze in real-time mode, it need to be keep them archived for further forensic analysis.

In this paper, we evaluate cloud based solution in Security Center for traffic data forensic analysis. The main contribution of our paper is that we propose a practical solution to collect data trace and analyze these data in parallel in a Cloud Computing platform. We propose to use cloud storage to keep huge traffic data and processing it with cloud computing platform to find the malicious attacks. As we already operate Collaborative Network Security Management System which has big data output. A workable case, phishing attack forensic analysis is presented and the required computing and storage resource are investigated. We have concluded that this phishing filter functions can be effectively scale to analyze a large volume of trace data for phishing attack detection with Cloud computing. The results also show that this solution is economical for large scale forensic analysis for traffic data.

2. Collaborative Network Security Management System

2.1 System Design and Implementation

Collaborative Network Security Management System (CNSMS) [28] deployed in multisite is shown in Figure 1. Multisite deployment, includes Beijing Capital-Info network, IDC Century-Link, an enterprise network and a campus network, is to demonstrate the workability of our system. These four sites are all managed by Collaborative Network Security Management System in Security Center over Internet. In each site, there are several NetSecu nodes [27] which take charge in different network environment to adapt to different physical link respectively.

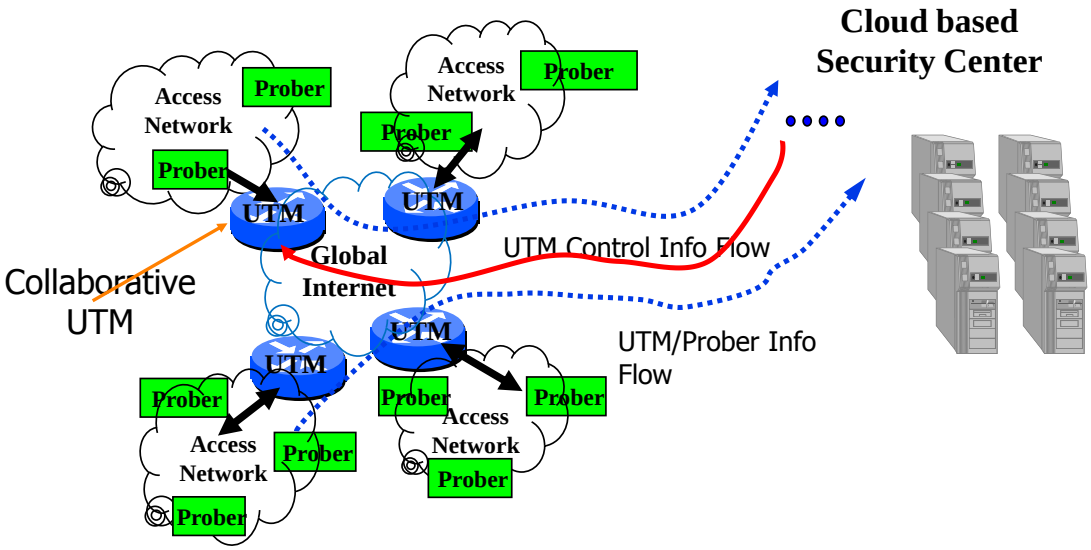

Figure 1. The deployment of Collaborative Network Security Management System in Multisite.

During the system's operating, the collaborative mechanism runs as we expected to share security events and rulesets, and new rulesets are enforced on demands as instructed by Security Center. Operating reports from each NetSecu node and Prober have been collected and send back to Security Center. Also there are a lot of network security events have been observed and recorded in the deployment, such as DDoS reflect attacks, Spam scatter and ad hoc P2P protocols etc.

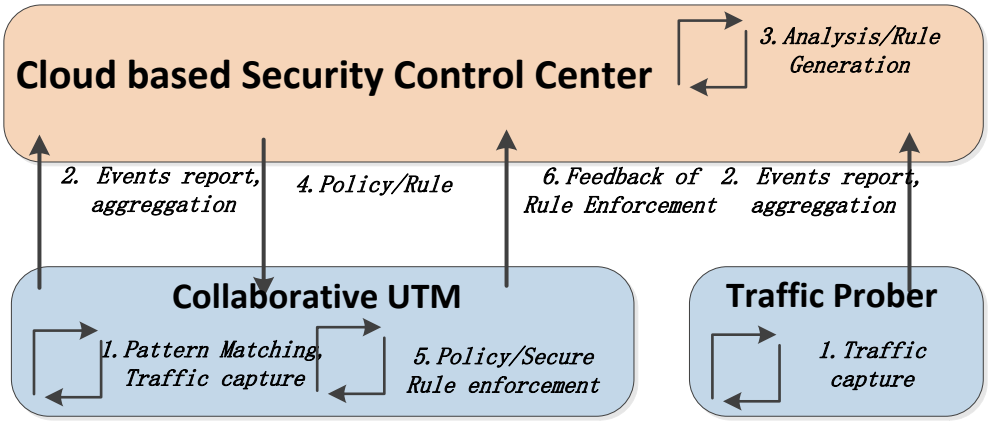

Figure 2. The work principle of Collaborative Network Security Management System with Cloud based Security Center.

Figure 2 illustrates the whole procedure of network security events processing. In general speaking, it is an information control cycle which divides several steps. Collaborative UTM and Prober acts as sensors and report the security events and traffic data to Security Center. The Security Center aggregates all the events and digs into the collected traffic data. After a detailed analysis and with the assistance of expertise manager, Security Center generates new policy or ruleset to disseminate to each collaborative UTM and Prober for enforcement, and receive the feedback information.

2.1.1 Traffic Prober

A traffic probe is the building block for recording the raw Internet traffic in connection level. Hyperion [29], Time Machine [30-31] and NProbe [32] are all well-known representative project in this function area. Traffic probe can be designed to focus on specified traffic incurred by certain security event when needed.

We enhance TimeMachine and deployed with TIFA [25-26] act as prober in separated device or Collaborative UTM and. The key strategy for efficiently recording the contents of a high volume network traffic stream comes from exploiting the heavy-tailed nature of network traffic: Most network connections are quite short, with a small number of large connections (the heavy tail) accounting for the bulk of total volume [31]. Thus, by recording only the first N bytes of each connection (the cutoff is 15 Kilobyte), we can record most connections in their entirety, while still greatly reducing the volume of data we must retain. For large connections, only the beginning of a connection is recorded as the beginning of such connection is the most interesting part (containing protocol handshakes, authentication dialogs, data items names, etc.).

2.1.2 Collaborative UTM

Acted as collaborative UTM, NetSecu is introduced in [27]. A NetSecu node consists of the following features:

1) Incrementally deployable security elements;

2) Dynamically enable/disable/upgrade security functions;

3) Policy-instructed collaboration over the Internet.

NetSecu node contains Traffic Prober, Traffic Controller, Collaborator Element, and Reporting Element to fulfill the above design goals.

A collaborator element in NetSecu manages other security elements based on Security Center's command. It unites individual NetSecu platforms into a Secure Overlay Network. The communication command between NetSecu nodes and the security center is transmitted in a SSL channel to ensure security. A collaborator can start or stop a security element at runtime. Collaborators can respond to security event such as limiting the DDoS traffic on demand.

NetSecu integrates security functions such as firewall, Intrusion Detection System (IPS) and antivirus (AV). These functions can be loaded in NetSecu nodes at runtime, and can be dynamically enabled, disabled and upgraded. NetSecu is based on commodity hardware and commonly used Java with Linux. With the multi-core technology matured, NetSecu has a comparable MLFFR 1 (Maximum Loss-Free Forwarding Rate) with bare Linux forwarding performance and most of security functions can run in multi-thread model to accelerate the flow processing and pattern matching needed for UTM.

NetSecu is also equipped with Bypass and self-protection capability to resist DoS attack in case of fault happening and malicious attacks for high availability and survivability.

2.1.3 Security Center

Collaborative Network Security Management System (CNSMS) is proposed in [28] and operated in Security Center. As NetSecu nodes could manage security problems in a subdomain and provide P2P communication interfaces, CNSMS orchestrates the communication between these NetSecu nodes. More specifically, CNSMS will achieve the following objectives:

[1] MLFFR is the highest forwarding rate with zero packet loss

1. Security policy collaborative dissemination and enforcement;
2. Security event collaborative notification;
3. Security ruleset dissemination, enforcement and update;
4. Trust infrastructure;
5. Scalability.

Another key function in Security Center is the forensic analysis of the collected traffic and network security events. We use cloud computing in Security Center to store large volume of traffic data origin from different and conduct data analysis to generate new security ruleset as shown the step 6 in Figure 2.

For further instruct the UTM to defeat new attacks, such as botnet, we must investigate the traffic in depth and acquire the communication graph of botnet, and generate security rules for enforcement in UTM to suppress the communication in-between bots and bot master.

Also this is workable to resist the DDoS attack launched by Botnet. As we equip the NetSecu node with open source application protocol identification and bandwidth management technology, the Security Center can instruct the system to be a collaborative distributed traffic management system, which detects and manages the traffic collaboratively after the analysis of collected traffic in Security Center. It could effectively improve the identification ratio of unknown botnet protocols and throttle the DDoS traffic.

2.2 System Application-Botnet Suppression

A typical distributed attack is Botnet, which is extremely versatile and are used in many attacks, for example, sending huge volumes of spam or launching Distributed Denial-of-Service (DDoS) attacks. The work principle of botnet is shown in Figure 1. Suppressing botnets become more and more difficult. There are many reasons, firstly, the Botmaster will keep their own botnets as small as possible not only to hide themselves but also to rent the botnets in an easy way, secondly, bots can automatically change their command and control server (C&C) in order to hide and rescue themselves.

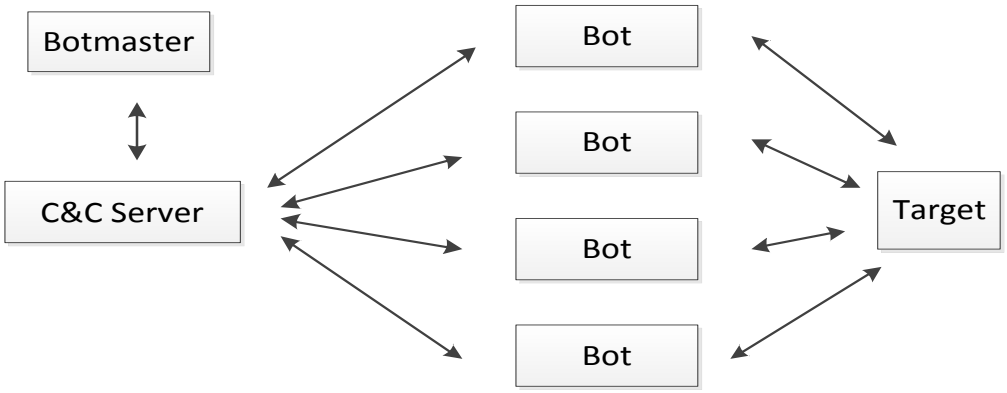

Figure 3. Botnet structure.

Based on overlay network, Collaborative Network Security System can be used for distributed botnets suppressing system. This system can automatically collect network traffic from every collaborative UTM in a distributed mode, and then process these collected data in Security Center. The detection algorithm proposed by [33-34] is based on behavior feature of botnet, the system will generate and distribute rules when botnets are detected in processing. The most important feature of this system is its close loop control characteristics, i.e., gather the feedback events resulted from the deployed rules, process and analyze in control node, remove invalid rules to make system more efficient and reliable.

3. Cloud based Forensic Analysis in Security Center

3.1 Cloud Storage and Computing platform

We focus on the traffic data storage and forensic analysis. The underground cloud storage and computing platform is based on Hadoop and Eucalyptus Cloud Computing. We also give some analysis the use of Cloud Computing platform based on Eucalyptus and Amazon EC2 respectively.

3.1.1 Cloud Storage with Hadoop

The Hadoop file system with version 1.0.1 is used for Cloud storage system of collected traffic. The master node is acted as namenode, secondarynamenode, jobtraker, Hmaster, and other node is working as datanode, tasktracker, regionserver.

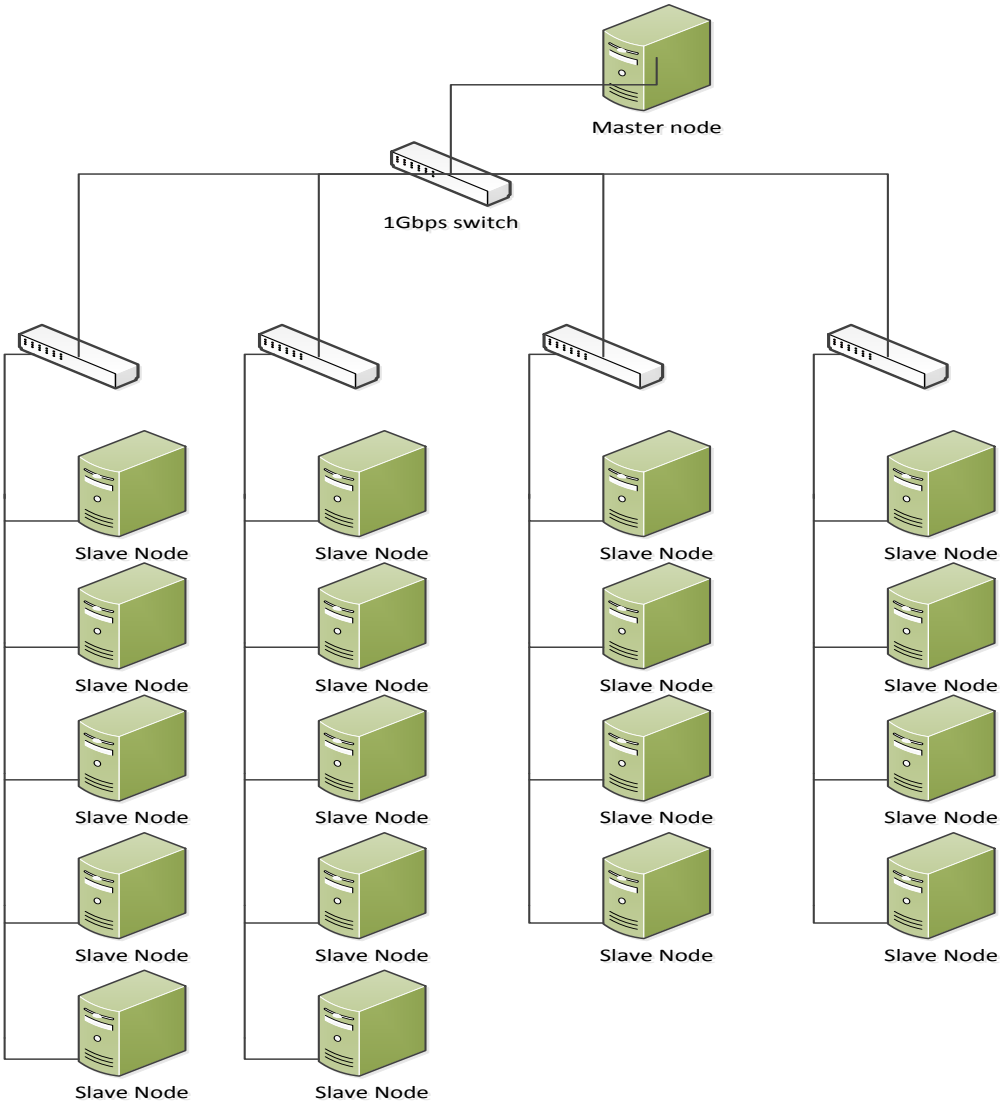

Figure 4. Cloud Storage for traffic collected with collaborative UTM.

There are totally 4 racks of machines with 5,5,4,4 in each rack. There are 18 slave nodes in total. The topology is shown in Figure 4.

As the Hadoop system is used for traffic analysis. The traffic collected in individual collaborative UTM is aggregated, and uploaded to this cloud platform. Each node has an Intel four cores CPU with 800MHz, and Memory size is 4GB, and with a 250G HardDisk.

We test the writing throughput for our Hadoop system with Hadoop's TestDFSIO utility2. We also test two scenarios where we write 18 files with each size 300MB and 36 files with each file size 100MB. The final results are shown in Table 4.

[2] Hadoop TestDFSIO command
hadoop jar hadoop-test-1.0.1.jar TestDFSIO -write -nrFiles 18 -fileSize 300
hadoop jar hadoop-test-1.0.1.jar TestDFSIO -write -nrFiles 36 -fileSize 100

Table 1. The average writing throughput of Hadoop files system in cloud platform.

Throughput(MBps) per node	File Size=100MB	File Size=300MB
Writing 18 files in total	176.4 MBps	202.5 MBps
Writing 36 files in total	151.2 MBps	90.0 MBps

3.1.2 Cloud Computing IaaS Platform

3.1.2.1 Cloud Computing based on Eucalyptus

In this section, we introduce our Cloud Computing platform based on Eucalyptus, an open-source platform by NASA and Ubuntu Enterprise cloud.

Figure 5 shows the Eucalyptus Cloud Computing platform we used. As shown in Figure 1, Eucalyptus Compute consists of seven main components, with the cloud controller component representing the global state and interacting with all other components. An API Server acts as the web services front end for the cloud controller. The compute controller provides compute server resources, and the Object Store component provides storage services. An auth manager provides authentication and authorization services. A volume controller provides fast and permanent block-level storage for the compute servers. A network controller provides virtual networks to enable compute servers to interact with each other and with the public network. A scheduler selects the most suitable compute controller to host an instance.

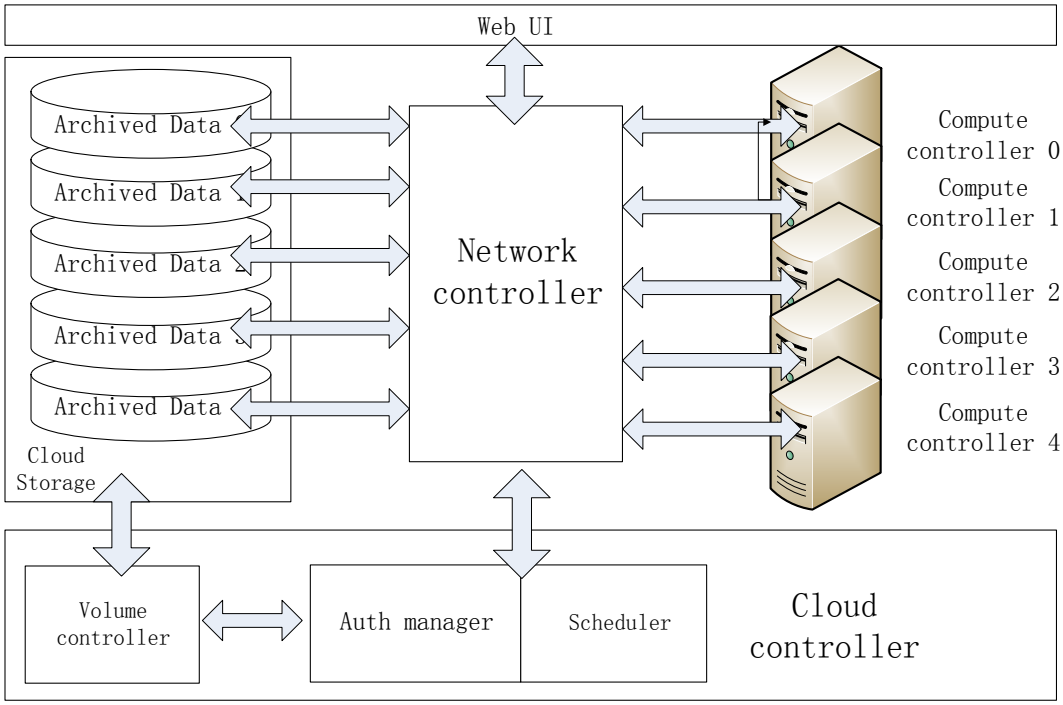

Figure 5. The Cloud Computing Platform based on Eucalyptus.

Our computer cluster consists of four-six heterogonous servers. Each server is with the following hardware parameters:

1. Intel Core 2 Quad Processor with 1.333 GHz FSB and 2MB cache, double channel 4GB DDR3 with 1.066GHz, Intel G41 + ICH7R Chipset and Intel 82574L Network Chipset;
2. Dual Intel Xeon5X00 series Processors with Intel 5000P+ESB2 chipset, E5330 + 8GB;
3. Intel Xeon 5X00 series with FSB - 4.8/5.86/6.4 GT/s QPI Speed with Intel 5520 + ICH10R chipset, 24GB.

In Eucalyptus's term, there is one cloud controller, and the others are compute nodes. Cloud controller acts as the computing portal, task assigner and result aggregation. There is computing instance affiliated with each compute node. In our usage scenario, we run 4 VM instances in each compute node, hence there about 24 running instances simultaneously. Each computing instance runs the pipeline divided into the following phases: data fetcher, data processing, and posting computing results. By this method, we can achieve best working efficiency of hardware and software resource's usage.

3.1.2.2 Cloud Computing based on Amazon

Amazon EC2 and S3 are used for comparative analysis. The main purpose to use Amazon service is with comparing purpose to our home-brewed Eucalyptus system. As the consideration of user privacy and legal issues, we conduct anonymization processing the data and upload the amazon S3 service.

3.2 Forensic Analysis of Phishing Attack

Phishing is an intriguing practical problem due to the sensitive information stolen (e.g. monetary user account name and password) and estimated about billion loss in accumulation annually. Not only the users but also the backing financial institutions such as e-banks and e-pay systems have been impaired by phishing attacks.

There is already much research works [7-9] to countermeasure phishing attacks. To protect web browser user from phishing attacks, plugins to compare visited URL with blacklist URL are already provided by main-stream web browsers. Google also provide safe Browser API [12] for check a URL in Google collected phishing database.

Some research on the LiveCycle of phishing web site is also given in [11], and the results show that the phishing URL is quite ephemeral, and make the collection of forensics [1-6] is difficult. It even makes it worse because of the un-awareness of this phishing attack for most of innocent Internet users.

Gregor Maier et al. [22] propose a traffic archiving technology for post-attack analysis in Bro IDS. Using Timemachine, the network trace data is archived and can be feed back to the IDS with current knowledge of modern attacks to find the forensics of attacks was undiscovered in that time. K. Thomas et al. proposed Monarch system [24] for real-time URL spam filtering for tweets and spam mails stream. Compared with Monarch, we put emphasis on phishing forensics analysis of large volume of offline trace with Cloud Computing platform.

With similar idea, we proposed an offline phishing forensic collections and analysis system. This system targeted to solve the following challenging problems:

(1) How to collect the original data to search the phishing attack forensics wherein;

(2) How to handle the huge volume data in a reasonably short time.

Cloud computing platform[15-17] is used for offline phishing attack forensic analysis. Firstly, our CNSMS collect the network trace data and report to Security Center. Secondly, we have both constructed an IaaS cloud platform [21] and use the existing cloud platform such as Amazon EC2 and S3 [18-20] for comparabale reason. All phishing filtering operation is based on Cloud Computing platform and running in parallel with “divide and conquer scheme”.

2.2.1 Data trace collection

Our trace data is an un-interruptible collection about half year with multiple vantage points with UTM’s deployment. The total size of traffic passed through our vantage points is about 20 TB. The total data is about 20TB and divided into 512MB data blocks. Typically, a typical 512M data block consists of about 40K URLs. An explored URL’s distribution is shown as shown in Figure 6.

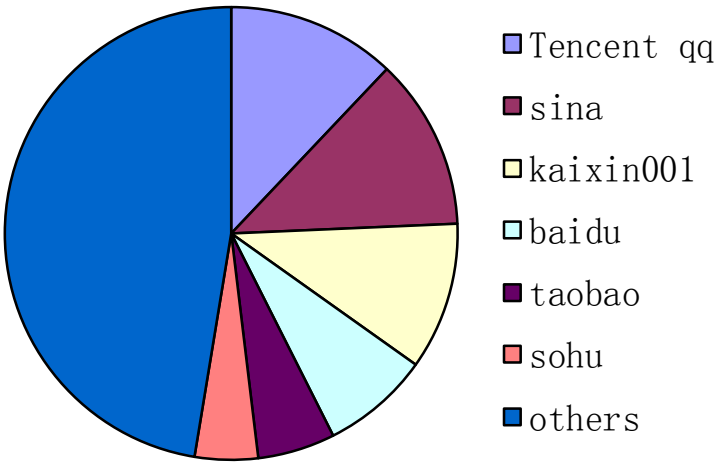

Figure 6. URLs distribution in a typical 512M trace data.

The experimental data is about 1TB when collected in a cut-off mode in a collaborative UTM. The data trace is still growing in the size during our experiments.

2.2.2 Data anonymization

To protect user’s privacy and avoid legal issues in the research, the trace data is anonymized to replace IP and other user information before the data processing in Amazon EC2.

2.2.3 Data processing

The data processing procedure are divided in different phases which are shown as follows:

(1) File splitting:

Each packet capture file created by Time Machine is 512 MB, and is further divided into smaller parts for processing by using tcpdump [23]. This is due to the amount of memory used during the extraction of data from TCP streams will exceed the maximum physical memory.

(2) TCP stream reassembly:

This stage is to restore the TCP streams in the captured pcap files using tcptrace [23].

(3) URL extraction:

After extracting data from TCP streams, grep is used to find all URLs contained in the data by searching for lines starting with “Referer: http://”.

(4) URL check:

URLs found are stored in a file to be checked for phishing by using Google Safe Browsing API [12]. In order to check URLs for phishing sites, we use phishing site data provided by Google. Google provides the first 32 bits of phishing sites' SHA256 values for users to use. If a match is found between a URL's SHA356 value is found, the full 256 bits hash value is sent to Google to check the site. More details on data provided by Google can be found in Google Safe Browsing API's documentation [12].
During the process of comparing URLs' hash values, a prefix tree is used for matching because the data provided by Google is only 32 bits long and a prefix tree can do the matching of a URL's SHA256 value with Google's data in O(1) time.

(5) Result reporter

This stage collects the final results in different machine, and aggregate the final report.

3.3 Experiments results

We conduct our evaluation experiment both on Eucalyptus and Amazon AWS for the comparison purpose.

3.1 Eucalyptus

We also run the phishing data block processing task in home-brewed Eucalyptus platform with Intel Core 2 Quad Processor with 1.333 GHz FSB and 2MB cache, double channel 4GB DDR3 with 1.066GHz, Intel G41 + ICH7R Chipset and Intel 82574L Network Chipset.

Time spending in different process stages in Eucalyptus platform are measured and concluded as shown in Table 2.

Table 2. Time spending in different stage in Eucalyptus.

stage	TCP stream reassembly	URL extraction	URL check
Time (seconds)	15~20	16~20	~5

It seems prefixTree comparison's speed is quite fast and this time spending can be almost ignored. But before URL check, it need take some time to download the Google Safe Browsing signature libraries, this time spending is quite undetermined due to network status and Google servers' response latencies.

It is also needed to point out that the m1.small instance in EC2 is memory constrained without swap partition support. It will cause problems when consuming a large volume of memory (exceeding the memory usage limit) during trace data analysis.

3.2 Amazon AWS

Trace file processing is written in Python and executes on an EC2 small instance running Ubuntu Linux 10.04. As Linux's command shows, the host CPU is Intel(R) Xeon(R) CPU E5430 @ 2.66GHz with cache size 6MB, and 1.7GB memory (with HighTotal: 982MB, LowTotal:734MB).

Different processing stage incurs different time consumption and is measured in Table 3.

Table 3 Time spending in different micro-stage in processing in Amazon EC2.

stage	TCP stream reassembly	URL extraction	URL check
Time (seconds)	~287	~47	1~2

Compared with Amazon case, it seems that the CPU used in in Amazon instance has better performance than QX9400 quad core CPU in our physical server.

3.3 Estimated the number of instances

Assume the time spending in a compute instance to handle a *k* bytes data block in stage (2), stage (3), and stage (4) are t_1, t_2, t_3 (in seconds) respectively. Assume there are *m* collaborative UTM or prober to collect traffic data, and the average traffic throughput is *f* bytes/s during the last 24 hours, and the traffic cut-off factor is *h*.

The number of total instances *L* in parallel needs to handle all last 24 hours traffic is calculated as follows:

$$T = t_1 + t_2 + t_3 \quad \text{(Eq. 1)}$$

$$L = (m*f*T*h)/k \quad \text{(Eq. 2)}$$

L is also affected by several factors such as the percentages of HTTP stream in the traffic, number of URLs in HTTP streams, user's behavior in exploring web sites etc.

In the Eucalyptus's case, we only run one instance in each physical server. Assume *m*=4, f = 100MByte/s (800Mbps) in 1 Gbps link, *h* =0.2 (means 20% traffic is captured), each block is 200M Bytes, T= 40 s, then the number of physical servers (or instances) in parallel is calculated as follows:

$$L = (m*f*T*h)/k = 4*100*40*0.2/200 = 16$$

In the Amazon EC2 case, T = 330s, and the number of needed EC2 m1.small instances in parallel is calculated as follows:

$$L = (m*f*T*h)/k = 4*100*330*0.2/200 = 132$$

4. Conclusion

The Collaborative Network Security Management System is very useful to countermeasure distributed network attacks. Its operation resulted in big data outputs, such as network traffics, security events, etc. In this paper, we propose to use cloud computing systems to explore the large volume of collected data from CNSMS to track the attacking events. Traffic archiving is implemented in collaborative UTM to collect all the network trace data and the cloud computing technology is leveraged to analyze the experimental data in parallel. An IaaS cloud platform is constructed with Eucalyptus and the existing cloud platform such as Amazon EC2 and S3 is also used for comparison purpose. Phishing attack forensic analysis as a workable case is presented and the required computing and storage resource are also evaluated by using real trace data. All phishing filtering operation is cloud-based and operated in parallel, and the processing procedure is also evaluated. The results show that the proposed scheme is practical and can be generalized to forensic analysis of other network attacks in the future.

ACKNOWLEDGMENT

This work is supported by Ministry of Science and Technology of China under National 973 Basic Research Program(grants No.2011CB302805, No. 2011CB302505, No.2012CB315801, and No. 2013CB228206), and National Natural Science Foundation of China (grant No. 61233016).

This work is also support with Intel Research Council's UPO program with the title of Security Vulnerability Analysis based on Cloud Platform with Intel IA Architecture.

References

[1] W.H. Allen, Computer Forensics, IEEE Security & Privacy, Volume: 3, Issue: 4, Page(s): 59 – 62, 2005.

[2] Michael A Caloyannides, Nasir Memon, Wietse Venema, Digital Forensics, IEEE Security & Privacy, Volume: 7, Issue: 2, Page(s): 16 – 17, 2009.

[3] F. Raynal, Y. Berthier, P. Biondi, D. Kaminsky, Honeypot forensics part I: analyzing the network, IEEE Security & Privacy, Volume: 2, Issue: 4, Page(s): 72 – 78, 2004.

[4] F. Raynal, Y. Berthier, P. Biondi, D. Kaminsky, Honeypot forensics part II: analyzing the compromised host, IEEE Security & Privacy, Volume: 2, Issue: 5Page(s): 77 - 80, 2004.

[5] N. Sklavos, N. Modovyan, V. Grorodetsky, O. Koufopavlou, Computer network security: report from MMM-ACNS, IEEE Security & Privacy, Volume: 2, Issue: 1, Page(s): 49 – 52, 2004.

[6] B.D. Carrier, Digital Forensics Works, IEEE Security & Privacy, Volume: 7, Issue: 2, Page(s): 26 – 29, 2009.

[7] B. Wardman, G. Shukla, G. Warner, Identifying vulnerable websites by analysis of common strings in phishing URLs, IEEE eCrime Researchers Summit, 2009.

[8] Shujun Li, R. Schmitz, A novel anti-phishing framework based on honeypots, IEEE eCrime Researchers Summit, 2009.

[9] R. Layton, P. Watters, R. Dazeley, Automatically determining phishing campaigns using the USCAP methodology, eCrime Researchers Summit (eCrime), 2010.

[10] P. Knickerbocker, Dongting Yu, Jun Li, Humboldt: A distributed phishing disruption system, eCrime Researchers Summit (eCRIME'09.), 2009.

[11] An Empirical Analysis of Phishing Blacklists, CEAS 2009 Sixth Conference on Email and AntiSpam, July 16-17, 2009, Mountain View, California USA.

[12] Google Safe Browsing v2 API
http://code.google.com/apis/safebrowsing/
Google Safe Browsing v2 API documentation:
http://code.google.com/apis/safebrowsing/developers_guide_v2.html

[13] APWG, http://www.apwg.org/ or http://www.antiphishing.org/crimeware.html

[14] StopBadware, http://stopbadware.org/

[15] Web search for a planet: the google cluster architecture, IEEE Micro March–April 2003.

[16] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, The Google File System, USENIX SOSP'03, October 19–22, 2003, Bolton Landing, New York, USA.

[17] Jeffrey Dean and Sanjay Ghemawat, MapReduce: Simplified Data Processing on Large

Clusters, USENIX OSDI 2004.
[18] Simson L. Garfinkel, An Evaluation of Amazon's Grid Computing Services: EC2, S3 and SQS, Technical Report TR-08-07, Harvard University, 2007.
[19] Amazon web services, Amazon elastic compute cloud (amazon ec2), March 18 2011. http://aws.amazon.com/ec2
[20] Amazon web services, Amazon simple storage service (amazon s3), March 18 2011. http://aws.amazon.com/s3
[21] Eucalyptus, open source Cloud Computing platform
Eucalyptus-nova; https://launchpad.net/nova
Eucalyptus-swfit; https://launchpad.net/swift
Eucalyptus-glance; https://launchpad.net/glance
[22] Gregor Maier, Robin Sommer, Holger Dreger, Vern Paxson, Enriching network security analysis with time travel, Sigcomm 2008.
[23] TCPtrace and TCPDUMP, http://www.tcptrace.org/ and http://www.tcpdump.org/.
[24] K. Thomas, C. Grier, J. Ma, V. Paxson and D. Song, Monarch: Providing Real-Time URL Spam Filtering as a Service, to be appeared in Proc. IEEE Symposium on Security and Privacy, May 2011.
[25] Jun Li, Shuai Ding, Ming Xu, Fuye Han, Xin Guan, Zhen Chen. TIFA: Enabling Real-Time Querying and Storage of Massive Stream Data, 1st International Conference on Networking and Distributed Computing (ICNDC), 2011.
[26] Zhen Chen, Xi Shi, Ling-Yun Ruan, Feng Xie and Jun Li, High Speed Traffic Archiving System for Flow Granularity Storage and Querying, ICCCN 2012 workshop on PMECT.
[27] Xinming Chen, Beipeng Mu, Zhen Chen, NetSecu: A Collaborative Network Security Platform for in-network Security. Proc. of the 3rd International Conference on Communications and Mobile Computing (CMC), 2011.
[28] Beipeng Mu, Xinming Chen, Zhen Chen, A Collaborative Network Security Management System in Metropolitan Area Network. Proc. of the 3rd International Conference on Communications and Mobile Computing (CMC), 2011.
[29] Peter Desnoyers and Prashant Shenoy, Hyperion: High Volume Stream Archival for Retrospective Querying, USENIX Annual Technical Conference 2007.
[30] Stefan Kornexl, Vern Paxson, Holger Dreger, Anja Feldmann, Robin Sommer, Building a Time Machine for Efficient Recording and Retrieval of High-Volume Network Traffic, IMC 2005.
[31] G. Maier, R. Sommer, H. Dreger, A. Feldmann, V. Paxson, and F. Schneider, Enriching Network Security Analysis with Time Travel. In Proc. ACM SIGCOMM, Seattle, WA, Aug. 2008.
[32] L. Deri, V. Lorenzetti, and S. Mortimer, Collection and exploration of large data monitoring sets using bitmap databases, Trac Monitoring and Analysis, Jan 2010.
[33] Fuye Han, Zhen Chen, Hongfeng Xu and Yong Liang, A Collaborative Botnets Suppression System Based on Overlay Network, the special issue of the International Journal of Security and Networks, Vo. 7, No. 4, 2012.
[34] Fuye Han, Zhen Chen, Hongfeng Xu and Yong Liang, Garlic: A Distributed Botnets Suppression System. Proc. of the IEEE ICDCS, the First International Workshop on Network Forensics, Security and Privacy (NFSP), 2012.
Phishing attack
[35] Tianyang Li, Fuye Han, Shuai Ding, Zhen Chen, LARX: Large-scale Anti-phishing by Retrospective Data-Exploring Based on a Cloud Computing Platform. ICCCN GridPeer workshop,

2011.
[36] R. Bye, S. A. Camtepe, and S. Albayrak, Collaborative intrusion detection framework: Characteristics, adversarial opportunities and countermeasures, in Proceedings of USENIX Symposium on Networked Systems Design and Implementation, April 2007.
[37] F. Cuppens and A. Mige, Alert correlation in a cooperative intrusion detection framework, IEEE Symposium on Security and Privacy, 2002.
[38] A. Hofmann, I. Dedinski, B. Sick, and H. de Meer, A novelty driven approach to intrusion alert correlation based on distributed hash tables, IEEE ICC's 2007.
[39] Donghua Ruan and Zhen Chen et al., Handling High Speed Traffic Measurement Using Network Processors, ICCT 2006.
[40] Jia Ni, Zhen Chen et al., A Fast Multi-pattern Matching Algorithm for Deep Packet Inspection on a Network Processor, ICPP 2007.
[41] Zhen Chen et al., AntiWorm NPU-based Parallel Bloom filters in Giga-Ethernet LAN, IEEE ICC'2006.
[42] Zhen Chen et al., AntiWorm NPU-based Parallel Bloom filters for TCP-IP Content Processing in Giga-Ethernet LAN, IEEE LCN WoNS2005.