ICIMP 2014

The Ninth International Conference on Internet Monitoring and Protection

ISBN: 978-1-61208-362-9

July 20 - 24, 2014

Paris, France

ICIMP 2014 Editors

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain
Constantion Paleologu, University 'Politehnica' Bucharest, Romania

ICIMP 2014

Foreword

The Ninth International Conference on Internet Monitoring and Protection (ICIMP 2014), held between July 20-24, 2014, in Paris, France, continued a series of special events targeting security, performance, vulnerabilities in Internet, as well as disaster prevention and recovery.

We take here the opportunity to warmly thank all the members of the ICIMP 2014 Technical Program Committee, as well as all of the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to ICIMP 2014. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the ICIMP 2014 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that ICIMP 2014 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of Internet monitoring and protection.

We are convinced that the participants found the event useful and communications very open. We hope that Paris, France, provided a pleasant environment during the conference and everyone saved some time to enjoy the charm of the city.

ICIMP 2014 Chairs:

ICIMP Advisory Committee
Go Hasegawa, Osaka University, Japan
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Jaime Lloret Mauri, Polytechnic University of Valencia, Spain
Constantion Paleologu, University 'Politehnica' Bucharest, Romania
Michael Grottke, University of Erlangen-Nuremberg, Germany
Emmanoil Serelis, University of Piraeus, Greece
William Dougherty, Secern Consulting - Charlotte, USA

ICIMP Industry/Research Chairs
Matthew Dunlop, United Sates Army Cyber Command, USA
Mohamed Eltoweissy, Pacific Northwest National Laboratory, USA
Nicolas Fischbach, COLT Telecom, Germany
Emir Halepovic, AT&T Labs - Research, USA
Miroslav Velev, Aries Design Automation, USA
Wei Wang, SnT Centre, University of Luxembourg, Luxembourg
Wenjing Wang, Attila Technologies, USA
Steffen Wendzel, Fraunhofer FKIE, Germany
Artsiom Yautsiukhin, National Council of Research, Italy

ICIMP 2014

COMMITTEE

ICIMP Advisory Committee

Go Hasegawa, Osaka University, Japan
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Jaime Lloret Mauri, Polytechnic University of Valencia, Spain
Constantion Paleologu, University 'Politehnica' Bucharest, Romania
Michael Grottke, University of Erlangen-Nuremberg, Germany
Emmanoil Serelis, University of Piraeus, Greece
William Dougherty, Secern Consulting - Charlotte, USA

ICIMP Industry/Research Chairs

Matthew Dunlop, United Sates Army Cyber Command, USA
Mohamed Eltoweissy, Pacific Northwest National Laboratory, USA
Nicolas Fischbach, COLT Telecom, Germany
Emir Halepovic, AT&T Labs - Research, USA
Miroslav Velev, Aries Design Automation, USA
Wei Wang, SnT Centre, University of Luxembourg, Luxembourg
Wenjing Wang, Attila Technologies, USA
Steffen Wendzel, Fraunhofer FKIE, Germany
Artsiom Yautsiukhin, National Council of Research, Italy

ICIMP 2014 Technical Program Committee

Jemal Abawajy, Deakin University - Victoria, Australia
Mohd Taufik Abdullah, Universiti Putra Malaysia, Malaysia
Manos Antonakakis, Damballa Inc., USA
Javier Barria, Imperial College London, UK
Lasse Berntzen, Vestfold University College Norway
Jonathan Blackledge, Dublin Institute of Technology, Ireland
Matthias R. Brust, University of Central Florida, USA
Christian Callegari, University of Pisa, Italy
Eduardo Cerqueira, Federal university of Para, Brazil
Christopher Costanzo, U.S. Department of Commerce, USA
Jianguo Ding, University of Skövde, Sweden
Matthew Dunlop, United Sates Army Cyber Command, USA
Mohamed Eltoweissy, Pacific Northwest National Laboratory, USA
Nicolas Fischbach, COLT Telecom, Germany
Ulrich Flegel, SAP Research - Karlsruhe, Germany
Alex Galis, University College London, UK
João Gomes, University of Beira Interior, Portugal
Stefanos Gritzalis, University of the Aegean - Karlovassi/Samos, Greece
Michael Grottke, University of Erlangen-Nuremberg, Germany

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Classification of TLS Applications

Chris Richter, Michael Finsterbusch, Klaus Hänßgen
Faculty of Computer Science,
HTWK Leipzig, Germany
{richter|finster|haenssge}@imn.htwk-leipzig.de

Jean-Alexander Müller
Dept. of Communication and Computer Science,
Hochschule für Telekommunikation Leipzig, Germany
Jean-Alexander.Mueller@ieee.org

***Abstract*—Traffic monitoring, traffic engineering, quality of service applications, network intrusion detection systems, as well as network management systems require the basic knowledge of which traffic is transmitted over a network. The increasing number of applications which are using encryption techniques such as TLS lower the ability to determine the applications that are running within a network. In this paper, we propose a method to detect applications in TLS encrypted connections. Our method uses a hybrid approach which combines protocol decoding to identify TLS traffic and to gather reliable information about the application data. Furthermore, a machine learning algorithm is used to determine the application which is protected by TLS. We describe our approach and compare it with other related methods in theory and prove its advantages on network measurements. The results show a significant improvement on classification Recall and Precision.**

Keywords–application classification, TLS, Internet traffic, machine learning.

I. Introduction

An increasing number of network protocols and applications encrypt the payload to protect privacy and integrity of the data. One popular way of doing this is to use the Transport Layer Security (TLS) protocol [1], which is a further stage of the Secure Socket Layer (SSL) protocol standardised by the Internet Engineering Task Force (IETF). Thus, the acronyms SSL and TLS are are often used as a synonym. An Internet study [2] from 2013 revealed that 356 applications within enterprises networks used SSL in some way, while 85 did not use standard SSL ports.

In order to do their work properly network management systems and security related systems such as firewalls or Network Intrusion Detection Systems (NIDS) need to know the kind of application. Therefore, these systems have to know whether the traffic is encrypted and which kind of application is being transmitted. To solve this problem, our approach is to use a hybrid method. First, we identify the TLS traffic. Second, the TLS data is analysed to determine the application. Due to encryption, only statistical information can be used for the second step.

The remainder of this paper is structured as follows: in Section II the related work is outlined. This is followed by Section III, which describes our approach for better TLS application classification, and Section IV which demonstrates the benefit of this approach on measurement results, Section V concludes the paper.

II. Related Work

Most research on TLS application classification has been done merely with statistical analysis. In most cases different kinds of well known machine learning algorithms were used. In some papers the authors concentrate on a single statistical parameter and use a dedicated method to evaluate the results.

There are two kinds of detecting applications for TLS connections. The first is to detect whether the network traffic is TLS or not. The second is to classify different applications, e. g., Hypertext Transport Protocol Secure (HTTPS), Simple Mail Transport Protocol Secure (SMTPS), etc., which are using TLS encryption. The goal in [3] is to distinguish TLS from non-TLS traffic. The authors are using the machine learning algorithms AdaBoost, C4.5, RIPPER and NaiveBayes and the statistical parameters packet length, inter-arrival time, duration and packet count. The detection rate varies between 70% and 98% for the different algorithms and different data sets.

The most work is related to the second approach which tries to classify different applications on top of TLS. In [4] the machine learning algorithm Random Forest as well as the clustering algorithm K-Means were used to classify network traffic for an intrusion detection system. It was shown that the approach is feasible for network monitoring, but the authors do not give further information about the classification rates. The authors of [5] used only the statistical parameter packet size for application classification. Therefore, the packet size of a packet is ranged to one of 30 bytes bins. The packet size distribution for a packet flow is compared with the Chi-square test to the values of known applications. This approach has a low classification accuracy of 10% to 40% for most observed applications.

Two statistical parameters – inter-arrival time and packet length – were used in [6] in conjunction with one of the three clustering algorithms DBSCAN, K-means and EM. On a data set with the File Transport Protocol (FTP), Real-time Protocol (RTP) and the Remote Framebuffer protocol (RFB), they could reach an accuracy up to 99%. The same parameters were also used by [7], but they used feature vectors containing several sub-parameters of inter-arrival time and packet size such as minimum, maximum, mean value and standard deviation. To compare the vectors of the ongoing packets with the known data set, the Euclidean distance or the Hamming distance are used. With this approach the authors could classify 80% to 94% of the used network traffic.

[8] is a PhD thesis about the identification of applications in encrypted tunnels, with the focus rests on HTTPS tunnels. The packet size of network packets is ranged to one of 15

 ISBN: 978-1-61208-362-9

bins. Several machine learning algorithms (Naive Bayes, C4.5, Decision Tree, neural networks, Nearest Neighbour, OneR) were used to classify the applications. The results vary between 30% and 100%.

Another paper [9] uses a bayesian machine learning algorithm with some more statistical parameters: packet length (min, max, mean), inter-arrival (min, max, mean), duration and packet count. Therewith, TOR and HTTP traffic could be classified with 85% of Precision and Recall.

All the papers cited above use only machine learning algorithms. The following two papers describe hybrid methods with additional preprocessing. In [10], at first a pattern based TLS detection is used to filter all TLS traffic. Only the TLS traffic is observed with the Naive Bayes machine learning algorithm. With this method 93% to 96% of HTTP and TOR traffic can be classified. Later we refer this as 'method 1'. A more advanced TLS preprocessing is done in [11]. The authors also use a pattern based TLS detection, but they observe the TLS session and using only application traffic without TLS handshake messages. Furthermore, they pay attention to the offset added by the Keyed-Hash Message Authentication Code (HMAC) and encryption. A statical offset of 21 bytes is used in their per-packet approach. The classification rate is between 81% to 100% for the ten observed applications. We refer to this as 'method 2' in the following sections.

Our own related work was on payload-based methods for application classification [12], with particular focus on protocol decoding. The protocol decoding inspects the network traffic and tries to decode each packet. If the decoded values match to the protocol description and if it fulfils all constraints of the protocol, the protocol is detected. This method is reliable but can only be used for unencrypted network traffic.

Another related work [13] [14] was on machine learning algorithms and investigated which kind of statistical information is useful for application classification. Furthermore, we investigated 20 different machine learning algorithms to find out which algorithms are suitable for network traffic analysis.

Besides TLS, other encryption techniques exist. [15] and [16] investigated traffic characteristic changes caused by Internet Protocol Security (IPsec) and encrypted Point-to-Point Tunneling Protocol (PPTP). The authors used the Naive Bayes, Support Vector Machines and C4.5 decision tree as machine learning algorithm for classification, but used two strategies for preprocessing the feature set. Either they split the traffic into encrypted and unencrypted traffic, or they do a normalisation of the feature set from encrypted traffic. The first strategy is used to approximate the feature set of the unencrypted traffic carried by the encrypted tunnel, to use only one classification model for the whole traffic. The second strategy use two classification models for each type of traffic. Their results show significant improvements in classification.

III. Hybrid Analysis Method

To identify TLS data in network traffic and to classify its content, we are using a hybrid method. First, to identify the TLS data, protocol decoding is used. As described in [12] and related papers, protocol decoding is a very reliable method for detecting TLS traffic. Additionally, some further information from the decoded TLS record headers are extracted to provide more precise statistical values regarding statistics gathered from the Transport Control Protocol (TCP) flow. The statistical values are used in conjunction with a machine learning algorithm to classify the protocol or application transmitted within TLS.

The TLS protocol is divided into five sub-protocols: the TLS Record Protocol, three handshaking protocols and the Application Data Protocol [1]. Application data messages are carried by the record layer protocol and are compressed, fragmented and encrypted with the negotiated master secret. A TLS session starts with a handshake. The handshake consists of the negotiation of a cipher suite, the exchange of certificates and keying material (e. g., Diffie Hellman). The application messages are treated as transparent data to the record layer.

Depending on the client and server configuration (e. g., usage and size of certificates), the number of packets exchanged during connection establishment varies. Additionally, the contents (except keying material) of the handshake messages of client and server are identical, even if a TLS connection is used by different applications. Thus, all the handshaking messages should never be considered for application classification.

After the TLS handshake, application data exchange starts. The application data is processed by the TLS layer as outlined in Figure 1. The application data can be compressed, but this is optional. The integrity of the data is protected by a HMAC, which is added to the application data. Then, the data with HMAC is encrypted and a TLS record header is added, which contains the TLS version, content length and type of content (application data or handshaking protocols). Due to TLS record header and HMAC, the payload of TLS is smaller than it seems on TCP level. The TLS record header has a constant size of 5 bytes, the length of the HMAC is one of six values: 0, 8, 16, 20, 32 or 48 bytes [17]. The used HMAC length depends on the used cipher suite which is negotiated during the handshake and can be provided by the protocol decoding. We propose considering this offset when using statistical data of TLS traffic. Compression was not used in all investigated network traffic and is frequently deactivated in the most applications. This is due to a security issue called CRIME. It was first described by [18] and later published as proof of concept exploit [19]. Thus, the compression has no influence on our statistical calculation and the classification results.

With the above description of the related parts of TLS, we can define four different methods for TLS application classification. The first two methods were already described in short at Section II. Method 1 [10] simply takes statistical values on TCP level. The TLS handshake, which is in general the same for all applications, is also included in the statistical calculation as the data records. The offset of the TLS record layer and HMAC is not removed. Large application data which was fragmented into several TCP segments will be counted as single application message, but a collection of small data records will be counted as one application message.

Method 2, as described in [11], skips the TLS handshake and starts evaluating the statistical data from the TLS stream after it detects the first data record by using packet inspection. Beginning from this point, it expects that every TCP segment

 ISBN: 978-1-61208-362-9

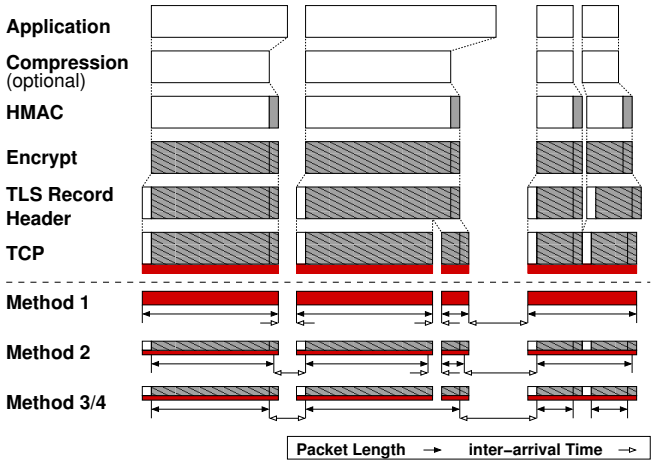

Figure 1: TLS fragmentation and network traffic statistics

transmits one TLS data record. The authors of [11] paid no attention to fragmentation. The analysis of the traffic used in Section IV showed that approximately 63% of TLS traffic is fragmented. In [11], a fixed offset of 21 bytes is removed from TCP data length because it is considered as TLS record header (5 bytes) and HMAC (16 bytes, e. g., for Message-Digest Algorithm 5 (MD5)).

We think it is necessary to extend the TLS traffic inspection to get more precise statistical data of the application messages. Therefore, we define a third and fourth method. Method 3 also skips the TLS handshake but inspects all succeeding TCP segments for TLS data records. Each data record is counted as one application message independent of the fragmentation. When a data record is split across several TCP segments, it is counted as one message. If a TCP segment contains different data records, each record is recognised as one application message. On method 3, a fixed offset of 21 bytes is used. The inter-arrival time between two data records within one TCP segment is considered as zero.

Method 4 works in the same manner as method 3, except it determines the concrete size of the used HMAC from the handshake. This results in more accurate statistical values but it increases the processing effort. This individual offset must be stored for each TLS connection. The four classification methods are outlined in Figure 1. It can be seen that method 1 and method 2 will capture values for packet length and inter-arrival time which do not match to the application data. The discrepancy between transmitted TCP segments and TLS data records can be large. The network traffic used in Section IV contains TLS data records which were split across up to ten TCP segments, but there were also TCP segments which contained up to ten data records. Only 37% of the TCP segments, which were captured in a campus network with many different clients and servers, contained one TLS data record. All other segments transmitted fragmented data. Methods 3 and 4 capture values which are very close to the application messages, while, method 4 provides the closest approximation.

We used the NaiveBayesUpdateable machine learning algorithm from WEKA Data Mining Software [20] and the statistical parameters described in [13] to process the statistical information from TLS data streams. For this work, we decided to use a packet based approach with supervised machine learning. As a result, the protocol decoding provides one data record for each TLS data record which is transmitted. These data sets are then used for learning and classification. Thus, the machine learning algorithm makes a classification decision for each TLS data record rather than for the whole flow.

We decided to use a bayesian classifier to get comparable results, because in [10] where method 1 is described, a bayesian classifier was used. Furthermore, bayesian classifiers are frequently used in the field of network traffic classification [21]. Nevertheless, the NaiveBayesUpdateable classifier can be exchanged with another machine learning algorithm and the approach will continue to work well.

IV. Experimental Results

This section discusses the classification results of the four applied methods to identify TLS encrypted traffic.

A. Metric

For evaluating and comparing the classification results, a metric is required. Various numbers of metrics, e. g., True Positive Rate, False Positive Rate, Recall and Precision, have been used in the past for evaluating traffic classification results. All of them are based on the following four metrics:

- true positive (t_p): objects belonging to protocol X and classified as protocol X
- true negative (t_n): objects not belonging to protocol X and not classified as protocol X
- false positive (f_p): objects not belonging to protocol X, but classified as protocol X
- false negative (f_n): objects belonging to protocol X, but not classified as protocol X

In this paper, the common used metrics Recall and Precision are applied to evaluate the performance of a classification method. The metric Recall defines the ratio of correct classified objects of a protocol to the total number of objects belonging to this protocol:

$$recall = \frac{t_p}{t_p + f_n} \quad (1)$$

Additionally, the accuracy of the classification is defined by the metric Precision, which defines the ratio of correct classified objects of a protocol to the number of all objects which were classified as this protocol:

$$precision = \frac{t_p}{t_p + f_p} \quad (2)$$

The primary goal of improving classification methods and an indicator for comparing the performance is to increase the Recall on the classification of protocols, and at the same time to increase the Precision.

 ISBN: 978-1-61208-362-9

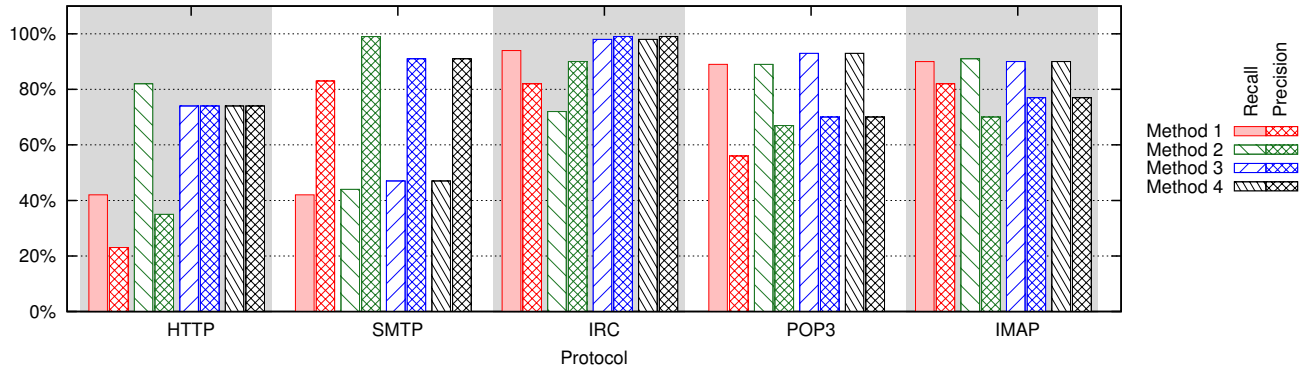

Figure 2: Classification results for all methods and protocols

TABLE I: Overview of the test data

Protocol	Method 1	Method 2	Method 3	Method 4
HTTP	21311	18516	20269	20269
SMTP	63880	60958	20456	20456
IRC	5735	5552	20800	20800
POP3	21287	17146	19682	19682
IMAP	15644	15420	20354	20354

B. Test data

TLS is used to protect a lot of applications and protocols. We decided to use the five protocols listed in Table I. These protocols was chosen because there are many publicly accessible servers to collect traffic with different server software and configurations. The HTTP traffic contains only ordinary HTML pages without Flash and video content. The e-mail protocol traces were captured at our university mail server as well as at our laboratory to capture conversations to publicly accessible server (e.g., Gmail). For the Internet Relay Chat (IRC) traces we also captured conversations to public servers. To get a realistic chat, the IRC client connected between five to ten minutes to a server which provides well-frequented IRC channels (e.g., Ubuntu support channel) without sending any chat message (only control messages). It received only chat messages of the connected channel, so the traces contain IRC talks between five to six hours for training and test, respectively.

Table I contains the data records determined from the traffic traces. We used a uniform distribution of data records (≈20.000) with respect to the data portions sent by the applications. The deviation from this values at Method 1 and 2 for SMTP or IRC results from ignoring the TLS fragmentation. Some TLS data records were split across up to ten TCP segments and some TCP segments contained up to ten data records. Table I shows only the test data. The training data for the machine learning algorithm contains the same amount of data records.

C. Classification results

Figure 2 shows the classification results of all applied methods. The protocols of the used traffic are placed on the x-axis, where the Recall and Precision results were displayed in percentage (y-axis) as bars. Each used method is represented by an own colour.

Starting with HTTP, method 1 classifies less then 50% of the HTTP traffic correctly with the Precision also lower than 25%, which implies that three out of four as HTTP classified packets are non-HTTP traffic. With method 2, Recall could be improved to 83% but nevertheless the Precision reaches only 36%. It is an improvement over method 1 but still two of three as HTTP classified packets are non-HTTP traffic. In general, it is not hard to implement a classification method with a high Recall, e.g., an algorithm that classifies each packet as HTTP reaches a Recall of 100%, but the Precision will be low according to the protocol distribution of the used traffic. In contrast, method 3 and method 4 gain a Recall of around 75%, which is less than method 2, but the Precision is improved to 74%. Thus, only one out of four as HTTP classified packets is non-HTTP traffic. This implies a higher reliability on the classification decision.

For SMTP, the Recall is continuously improved from 43% on method 1 up to 48% on method 2. Also, the Precision could be increased from 84% on method 1 up to 92% on method 2. The highest Precision could be realised with method 2 (99%). On all methods, the false negatives — SMTP traffic which was not classified as SMTP — were nearly entirely classified either as HTTP or Post Office Protocol version 3 (POP3).

With a Recall between 95% (method 1) and 99% (method 3 and method 4) the IRC protocol has the best classification results. Besides the high values for the Recall also the Precision with 82% (method 1) and 99% (method 3 and method 4) on a high level. Method 3 and method 4 achieve almost perfect classification results. Only method 2 decreases the classification accuracy; nearly all false negatives were classified as HTTP and Internet Mail Access Protocol (IMAP).

For POP3, the classification accuracy could be increased from method 1 to method 4. The recall could be enhanced from 89% to 94% and the Precision was enhanced from 57% to 71%. In contrast the Recall on IMAP was nearly constant at 91%, but the Precision was decreased from 82% on method 1 to 77% on method 4.

D. Future trend

The similarity and the missing enhancements on the classification accuracy between method 3 and method 4

 ISBN: 978-1-61208-362-9

are based on the applied cipher suites in the used traffic, respectively. The used traffic contains 11 different cipher suites, but only one cipher suite which is less than 1% of the whole traffic, uses a MD5 hash with a HMAC size of 16 bytes. All other cipher suites are using a Secure Hash Algorithm version 1 (SHA1) with a HMAC size of 20 bytes. Accordingly, nearly the entire traffic, there is only a fixed offset in the data record length between method 3 and method 4. This fixed offset causes no differences for the machine learning algorithm, and there is no improvement from method 3 to method 4 according to our data set. However, the Internet Assigned Numbers Authority (IANA) specified more than 300 cipher suites with the different HMAC sizes as described in section III. In consideration of the current lack of security, it can be supposed that stronger cipher suites will be used to secure the data. In this case, there will be a larger distribution of the used HMAC sizes and thereby the advantages of method 4 will be proved.

To determine if this assumption is right or not, we added some TLS traces from servers which support SHA256 for the HMAC to our test and training data set. Currently, only a small subset of all TLS servers support HMAC algorithms which are more secure than SHA1. Additionally, the TLS client makes a suggestion of the cipher suites to use, but only the newest versions support the stronger HMAC algorithms. Currently, only the latest web-browsers support TLS 1.2 with the new cipher suites [22]. Browsers take a pioneering role, while other applications do not support these cipher suites in the stable versions and providing support only within development versions (e.g., e-mail user agent *Mozilla Thunderbird* development version 30.0 beta 1 [23]). Furthermore, the web-browsers use their own TLS libraries, whereas other applications use the TLS libraries provided by the operating system or the used programming language (e.g., Java, C#). Only the latest versions of the operating systems and programming languages support TLS 1.2 [22] with the appropriate cipher suites. Thus, we concentrate on HTTP and IMAP. HTTP causes a significant amount of traffic in the Internet and our results of IMAP showed no improvements to the other methods.

To test the assumed enhancements of method 4 against those in method 3, we applied a set of HTTP and IMAP flows with cipher suites which are using SHA256 for calculating the HMAC with a length of 32 bytes. Due to the small set of these flows, the results can only give an indication of the behaviour for method 3 and method 4 on traffic with wider distribution of more secure cipher suites. The classification results on the test set with these new flows support our assumptions that method 4 leads to better classification results than method 3 when the investigated traffic includes different cipher suites with different HMAC sizes. Method 4 has achieved an enhancement between 2% and 3% on Recall and on Precision according to method 3. Nevertheless, further investigations with a well-balanced data set are required for a final confirmation of the enhancements of method 4 compared to method 3.

V. Conclusion

We compared four approaches for TLS application classification, each with different depth of TLS investigation. As a preparation for these methods, protocol decoding was used to filter TLS traffic from non-TLS traffic — to focus the analysis on dedicated applications — as part of our hybrid classification method. The results show an improvement of the classification accuracy according to Recall and Precision on the investigated protocols. For most applications, the reliability, which is based on Precision, could be increased from method 1 to method 4. The advantages of method 4 in contrast to method 3 will be shown on the deployment of other cipher suites on client site and server site. No significant differences could be determined between both methods on the underlying traffic. However, on an exemplary data set, an enhancement between 2% and 3% on Recall and on Precision could supported the assumption of method 4 as compared with method 3 on more secure cipher suites with larger HMACs.

As a result, method 3 and method 4 show a clear enhancement on the classification results according to Recall and Precision when compared to method 1 and method 2, which are well-known and commonly used methods for classifying TLS applications. Therefore, it is defenitly worth making the additional effort to processing the detailed statistic values for both methods. As other traffic classification methods have shown, it is expensive to improve an approach to gain the last remaining percentages which could achieve a perfect classification accuracy of nearly 100%.

In general, the used traffic is the critical fact in such evaluations, because the traffic covers only a limited part and is based on the underlying network. According to other evaluations, our classification results are in most cases not the best, but when repeating other approaches with our traffic, the results are partially quite different from the announced results. In conclusion, the stability of the statistical features strongly depends on the used traffic.

In future, the influence of the usage of compression for the classification accuracy has to be analysed, as well as the detection of further applications which are using TLS. Furthermore, the performance of other machine learning algorithms should be inspected for our presented methods.

Acknowledgements

We thank the reviewers for their valuable comments which helped to considerably improve the quality of the article. This work is supported by the European Regional Development Fund (ERDF) and the Free State of Saxony.

Gefördert aus Mitteln der Europäischen Union

Europa fördert Sachsen.
EFRE
Europäischer Fonds für regionale Entwicklung

References

[1] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246 (Proposed Standard), Internet Engineering Task Force, Aug. 2008.

[2] "The Application Usage and Threat Report – An Analysis of Application Usage and Related Threats within the Enterprise," Palo Alto Networks, Tech. Rep. 10, Jan. 2013.

 ISBN: 978-1-61208-362-9

[3] C. McCarthy and A. Zincir-Heywood, “An investigation on identifying SSL traffic,” in Computational Intelligence for Security and Defense Applications (CISDA), 2011 IEEE Symposium on, April 2011, pp. 115–122.

[4] K. Ethala, R. Shesadri, and N. G. Renganathan, “The use of random forest classification and k-means clustering algorithm for detecting time stamped signatures in the active networks,” Journal of Computer Science, vol. 9, no. 7, 2013, pp. 875–882.

[5] G. Mujtaba and D. Parish, “Detection of applications within encrypted tunnels using packet size distributions,” in Internet Technology and Secured Transactions, ICITST, Nov 2009, pp. 1–6.

[6] M.-D. Wu and S. D. Wolthusen, “Network Forensics of Partial SSL/TLS Encrypted Traffic Classification Using Clustering Algorithms.” in IT Incident Managegent & IT Forensics, ser. LNI, vol. 140. Gesellschaft fuer Informatik, 2008, pp. 157–172.

[7] H. Liu, Z. Wang, and Y. Wang, “Semi-supervised Encrypted Traffic Classification Using Composite Features Set,” Journal of Networks, vol. 7, no. 8, 2012, pp. 1195–1200.

[8] G. Mujtaba, “Identification of Networked Tunnelled Applications,” Ph.D. dissertation, Loughborough University, May 2011.

[9] G.-L. Sun, F. Lang, M. Yang, and J. Hua, “Application protocols identification using Non-parametric Estimation method,” in Strategic Technology (IFOST), 2011 6th International Forum on, vol. 2, Aug 2011, pp. 765–768.

[10] G.-L. Sun, Y. Xue, Y. Dong, D. Wang, and C. Li, “An Novel Hybrid Method for Effectively Classifying Encrypted Traffic,” in GLOBECOM, IEEE, Dec 2010, pp. 1–5.

[11] L. Bernaille and R. Teixeira, “Early Recognition of Encrypted Applications,” in Proceedings of the 8th International Conference on Passive and Active Network Measurement, ser. PAM’07, 2007, pp. 165–175.

[12] M. Finsterbusch, C. Richter, E. Rocha, J.-A. Müller, and K. Hänßgen, “A Survey of Payload-Based Traffic Classification Approaches,” Communications Surveys Tutorials, IEEE, vol. PP, no. 99, 2013, pp. 1–22.

[13] M. Finsterbusch, C. Richter, and J.-A. Müller, “Parameter Estimation for Heuristic Based Internet Traffic Classification,” in ICIMP 2012: The Seventh International Conference on Internet Monitoring and Protection, IARIA, Ed. Stuttgart, Germany: IARIA, 2012, pp. 13–22, ISBN: 978-1-61208-201-1 .

[14] C. Richter, M. Finsterbusch, K. Hänßgen, and J.-A. Müller, “Impact of Asymmetry of Internet Traffic for Heuristic Based Classification,” International Journal of Computer Networks (IJCN), vol. 4, no. 10, 2012, pp. 167–176.

[15] Y. Okada, S. Ata, N. Nakamura, Y. Nakahira, and I. Oka, “Application identification from encrypted traffic based on characteristic changes by encryption,” in Communications Quality and Reliability (CQR), 2011 IEEE International Workshop Technical Committee on, May 2011, pp. 1–6.

[16] ——, “Comparisons of machine learning algorithms for application identification of encrypted traffic,” in Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on, vol. 2, Dec 2011, pp. 358–361.

[17] E. Rescorla. Transport Layer Security (TLS) Parameters. Internet Assigned Numbers Authority. [retrieved: Jan., 2014]

[18] J. Kelsey, “Compression and Information Leakage of Plaintext,” in Revised Papers from the 9th International Workshop on Fast Software Encryption, ser. FSE ’02. London, UK, UK: Springer-Verlag, 2002, pp. 263–276.

[19] J. Rizzo and T. Duong, “CRIME exploit – crime.py,” 2014, URL: https://gist.github.com/stamparm/3698401 [accessed: 2014-05-11].

[20] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA Data Mining Software: An Update,” SIGKDD Explor. Newsl., vol. 11, no. 1, Nov. 2009, pp. 10–18.

[21] T. Nguyen and G. Armitage, “A survey of techniques for Internet traffic classification using machine learning,” Communications Surveys Tutorials, IEEE, vol. 10, no. 4, 2008, pp. 56–76.

[22] Java Platform Group, “JDK 8 will use TLS 1.2 as default ,” 2014, URL: https://blogs.oracle.com/java-platform-group/entry/java_8_will_use_tls [accessed: 2014-05-20].

[23] Mozilla Foundation, “Thunderbird,” 2014, URL: www.mozilla.org/thunderbird [accessed: 2014-05-20].

 ISBN: 978-1-61208-362-9

Trust and Risk Relationship Analysis on a Workflow Basis: A Use Case

Valentina Viduto, Karim Djemame, Paul Townend, Jie Xu, Sarah Fores, Lydia Lau, Vania Dimitrova
School of Computing, University of Leeds
Leeds, UK
{V.Viduto, K.Djemame, P.M.Townend, J.Xu, S.Fores, L.M.S.Lau, V.G.Dimitrova}@leeds.ac.uk

Martyn Fletcher[2], Stephen Hobson[2,]
Jim Austin[1,2], John McAvoy[2],
[1] Department of Computer Sciece, University of York,
[2] Cybula Ltd.
York, UK
{martyn, stephen, austin, mcavoy}@cybula.com

Charlie Dibsdale,
Rolls Royce PLC
Derby, UK
charlie.e.dibsdale@o-sys.com

***Abstract*— Trust and risk are often seen in proportion to each other; as such, high trust may induce low risk and vice versa. However, recent research argues that trust and risk relationship is implicit rather than proportional. Considering that trust and risk are implicit, this paper proposes for the first time a novel approach to view trust and risk on a basis of a W3C PROV provenance data model applied in a healthcare domain. We argue that high trust in healthcare domain can be placed in data despite of its high risk, and low trust data can have low risk depending on data quality attributes and its provenance. This is demonstrated by our trust and risk models applied to the BII case study data. The proposed theoretical approach first calculates risk values at each workflow step considering PROV concepts and second, aggregates the final risk score for the whole provenance chain. Different from risk model, trust of a workflow is derived by applying DS/AHP method. The results prove our assumption that trust and risk relationship is implicit.**

Keywords- trust; risk model; provenance; decision support; workflow; DS/AHP.

I. INTRODUCTION

In recent years, business critical decisions heavily rely on data collected and manipulated by many distributed sources and services. To make sure that crucial, high value decisions will not put business at risk, it becomes important to put trust in information and system data outputs. Trust is one of the concepts that is used to verify the usefulness and/or criticality of data, systems, personnel and whole workflow. However, it is quite challenging to define the term because it is being used with a variety of meanings and in many different contexts, sociology, psychology, and philosophy. The common notions of trust are associated with hope, faith, belief, confidence reliance on the integrity, dependence or character of a person or thing [10]. The variety of common terms shows that there is no precise definition of trust as it largely depends on author's viewpoint. Trust is also often situation specific; in one environment trust does not directly transfer to another environment and the notion of context is necessary [10].

Recent research inherently links trust to risk. There is no reason to trust if there is no risk involved. Thus, the cooperation or interaction with the system or human is less likely with higher risk unless the benefits from such interaction are worth the risk. The SECURE project [10] has made a good attempt in demonstrating that risk and trust are inexorably linked and must both be considered when making a decision about some ambiguity whose outcome depends on another entity's action. Also, considering observations made by Solhaug et al. [2] seeing that trust is generally neither proportional nor inverse proportional to risk under various constraints, in this paper, we put a first attempt to demonstrate how trust and risk relationship can enhance trustworthiness in systems and inform decisions. Inspired by the challenge of relating trust while considering consequences of risk, the trusted digital Spaces through Timely Reliable And Personalised Provenance (STRAPP) project [17] aims to provide an approach to enable users make informative decisions by considering three notions associated with the data: risk, provenance and trust. To demonstrate the STRAPP view of trust and risk relationship we use W3C PROV Data model [11] for provenance interchange. This data model describes entities, activities and people involved in the creation of data, its operation and decision making. It allows the decision maker to see the chain of activities, processes and data inputs as well as agents who performed certain actions with regard to data.

The aim of the paper is to address an assumption that trust in system can be placed knowing the data source and its quality, and risk associated with some processes may be high despite of good quality data used. We model risk and trust independently on a basis of a same workflow generated using BII (Brain Injury Index) case study data. Under STRAPP context, we define risk as a "*probability of some unwanted events at every workflow process which may result in unwanted consequences to this process*", whereas trust is assessed in the context of data quality of a particular data file, and defined as *"a degree of confidence placed in input data while considering data quality attributes: completeness, accuracy, relevance, of the data file."* Data file in the BII case study consists of several metadata input fields that are

 ISBN: 978-1-61208-362-9

assessed in terms of their quality and importance. The ranking of input files is performed by applying Dempster–Shafer Analytical Hierarchy Process (DS/AHP) [14].

The remainder of the paper is organised as follows: Section II gives an overview of the STRAPP project highlighting its aims and applicability to the BII case study. Section III provides the most relevant work in three research areas: trust, risk and provenance and tries to highlight how these fields can facilitate decision making process. Section IV discusses BII case study as well as presents risk and trust models on a workflow basis. Section V summarises the results, work accomplished and provides future research directions.

II. STRAPP OVERVIEW

The STRAPP project has been established, funded by Rolls-Royce, Cybula Ltd, and the UK Technology Strategy Board to facilitate the assessment of provenance-based, personalised trusted digital spaces where timely and critical decisions should be made. The objective of STRAPP is to enable users to place increased trust on data shown by, and decisions made by a system and by allowing them to view the provenance of that data or decision, presented in a personalised manner (for example, based on their role; managers may need to view the provenance and risk of a decision at a different level than software engineers, etc.) Furthermore, the project aims to provide visualization mechanisms to ensure users understand trust and the risks associated with data and decision-making. In the short term, these mechanisms are integrated to both the Equipment Health Management (EHM) system developed by OSys - a subsidiary company of Rolls-Royce PLC - that provides customers (primarily in the aerospace, marine and energy sectors) with the ability to diagnose and predict equipment faults, and to the Brain Injury Index (BII) system developed by Cybula Ltd that assists researchers and practitioners in the healthcare industry, with a focus on neuroscience. In the longer-term, it is hoped that many other decision-support systems in a wide range of sectors will be able to take advantage of the STRAPP system.

In this paper, we are primarily concerned with the trust and risk assessment components modelled using BII case study data. The purpose is to demonstrate the implicit relationship between trust and risk, as discussed by Solhaug et al. [2] and Cahill et al. [10] and visualise this relationship on a workflow basis.

III. RELATED WORK

Our research encompasses several research directions: trust assessment and modelling, risk analysis and its conceptual relation to trust, provenance modelling and its usability with regard to decision making process. Therefore, in this paper we will focus on trust and risk modelling on a basis of provenance data to make an attempt of demonstrating the implicit relationship between risk and trust as it was observed [2] [10], under specific use.

Trust is a widely explored topic within a variety of computer science domains. Trust is defined as a relationship between two entities, a trustor and a trustee where a trustor places some level of trust in a trustee under a specific set of contexts. Thus, trust, in literature, is used in a variety of meanings. A distinction between context-independent trust (reliability trust) and context-dependent trust (decision trust) can often be recognized among scientific community, although usually not explicitly expressed [4]. Reliability trust is interpreted as the reliability of something or somebody independent of the context. As such, according to Gambetta [1], *trust is a particular level of the subjective probability with which "an agent assesses that another agent or group of agents will perform particular action, both before he can monitor such action and in the context in which it affects his own action."* It is a crucial question then, whether or not to engage in cooperation with an agent. This cooperation depends on the extent to which the agent (trustor) believes that the trustee will behave in a certain way. Hence, the level of trust is determined subjectively based on evidences available to the trustor on trustree's behaviour and constraints by which this behaviour might be regulated.

Decision trust, when seen within a context, is defined as *the extent to which a given party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible* [4]. This definition implicitly covers contextual elements, such as possible outcomes, environmental factors (existing safety/security mechanisms) and risk attitude (taking, avoiding, and transferring). Josang and Presti in [5] draw a model of trust composed of a reliability trust as the probability of a transaction success and a decision trust derived from a decision surface. With such example, authors provide a first attempt to shape the relationship between risk and trust. The model first, calculates expected gain of a possible transaction and second, introduces a fraction of the capital the agent is willing to risk. Risk, as part of the model, is taken in order to derive a more complete definition of trust, the decision trust. Therefore, the approach of including risk into the model provides more meaningful notion of trust because it combines trust with risk attitudes.

Recently, trust is modelled by highlighting the presence and importance of provenance data. The semantic representation of trust and provenance data is modelled through the provenance ontology. As such, Emaldi et al. [6] present a trust model for the measurement of trust value in the context of smart cities. The trust value is calculated according to each factor independently. The factors calculated are defined as trust of authority, popularity, recommendation, provenance, timeliness and geographical distance. Another method for assessing trust based on provenance information is presented by Hartig and Zhao [7]. The authors proposed an assessment method which calculates trust values based on timeliness of data quality. Ceolin et al. [8] assessed the trust by first computing reputation-based trust values and second, trust values are computed based on provenance information, represented by

 ISBN: 978-1-61208-362-9

means of W3C standard PROV model. By merging trust values authors claim that it can be beneficial for reliability of the estimated trust value. In trust management domain, *reputation* is used to define trust between two agents. Reputation is what generally said or believed about a person's or thing's character or standing [4]. It influences trust in two ways: firstly, it positively affects the trustor's reliability trust in the trustee and secondly, it disciplines the trustee as it is known that bad behavior will be seen. The good example of difference between trust and reputation can be seen in the following statements: *(1) I trust because of its good reputation (2) I trust despite of its bad reputation.* Statement 1 states that trust is placed based on reputation, while statement 2 reflects that a relying party has some extra knowledge about a party to trust, e.g., through direct experience or relationship that can overrule any positive or negative reputation. A fuzzy model for calculating trust based on a workflow was proposed in [9]. Rajbhandari et al. argue that provenance provides a useful way to capture information and to be used to evaluate trust and fuzzy rules enable greater degree of flexibility in assessing provenance information.

There are many forms and variations of risk and trust analysis, depending on the application domain, such as health care, finance, reliability and safety, IT security. In finance, risk analysis is concerned with balancing potential gain against risk of investment loss. In this setting risk can be both positive and negative. Within reliability, safety and IT security risk analysis is concerned with protecting existing infrastructure and assets. This paper focuses on analysing risk and trust of a health care system under specific use case. We are aiming to demonstrate that risk and trust are not necessarily proportional [2], but have an impersonal relation [3] and fulfill each other. In safety critical and health care systems, it is often stated that trust is better understood in terms of cost/benefit analysis and calculated risks, as well as by knowing provenance information. Therefore, in a situation when users should make critical decisions they users should be aware of possible outcomes and their probabilities, risks to be taken and uncertainties involved in the analysis as well as provenance of information.

As it can be seen, the research on trust often highlights importance of provenance. Moreover, the way trust is modelled depends on perspective of the domain and trust definition. We base our research on the assumption that trust can be enhanced knowing the quality of data and its provenance. Also, we make an assumption that knowing data related risks and their scale can improve the knowledge of a system, its processes and most critical data-related activities. In overall, knowing how data was processed, derived, operated, agents involved as well as associated trust and risk values provided at each stage of data processing.

IV. BII CASE STUDY

A neuroscience researcher wants to choose a set of data files on which to validate a new analysis technique. They use the BII portal to select files for appropriate patients, but want to be able to choose a subset of these files which represent the data which is the most trustworthy. For any given file, the researcher wants to see a summary which helps them understand to what extent they can trust the data and what is the level of risk associated with this data.

All files on the BII portal have associated metadata. If the metadata is not present, the data should be deemed to be less trustworthy. However, it will not necessarily mean the data is more risky, as the risk is associated with other parameters, such as threats of agent's failure, wrong data export settings and/or various bugs in software agents.

A. Provenance-Based Risk Model of a Domain-Based Workflow

In order to assess risk associated with making critical, high-value health decisions based on evidence presented by a system, it is essential to know how the data was derived, processed and transformed. For this purpose, we build on a workflow generated and associated provenance meta-data which is unique for each system under observation and contains the linking between system personnel, processes and documents along with configuration management information as a connected directed graph. The provenance modeling builds upon the W3C's de-facto ontological representation of PROV named PROV-O [15], which is defined using the W3C's Web Ontology Language OWL2 [16]. The provenance data consists of a list of entities from the workflow graph as well as provenance specific meta-data: software version, training data for software systems, personnel associated with system processes. Within STRAPP, we apply a quantitative risk assessment approach to estimate the level of risk possessed by the provenance data recorded within the PROV data model. Therefore, an identification of the elements of risk within the provenance chain becomes important. It should be noted, that the nature of risks may differ thus, the quantitative risk estimation too.

In order for a risk model to be applied to the BII use case, STRAPP first is used to generate a provenance chain. Based on a provenance chain risk model can be applied and relevant queries are made. As such, STRAPP performs a number of queries to the target system, where risk data is stored and dynamically monitored. Table 1 shows risk attributes generated by the BII system and risk matching combinations. A Domain expert usually is responsible for estimating the probability of such combinations and their impact. These data are then passed to STRAPP, which performs necessary calculations and risk aggregation as well as presents risk output on a scale from 1 to 7, where 1 is low risk and 7 is considered as high. Risk is calculated based on an Activity_ID, Entity used by and Agent associated with this Activity_ID. Fig. 1 shows an output from STRAPP

 ISBN: 978-1-61208-362-9

system based on BII use case data. The workflow illustrates a chain of processes starting from its initial data source (Patient) and finishing by an Entity "Diagnosis" made to the patient.

Threats and vulnerabilities shown in Table I are specific to the activities, entities and agents involved in the chain. The list can change depending upon the domain. Risks in BII domain are clearly associated with data completeness, relevance, accuracy (e.g., V2, V3, V4, V5, etc).

TABLE I. RISK COMBINATIONS

Vulnerability (Vi)	Threat (Tj)	Matching Combinations
Poor signal quality (V1)	Electrical Interference (T1)	V1T1, V1T3, V1T4
Incomplete Data (V2)	Software Agent Failure (T2)	V2T2, V2T8
Inaccurate values (V3)	Incorrect Calibration (T3)	V3T3
Incorrect data exported (V4)	Poor Electrode Contact (T4)	V4T5, V4T6, V4T7
Malfunction in a training model (V5)	Software agent Export failure (T5)	V5T5
Incorrect data set (V6)	Incorrectly labelled units (T6)	V6T6
Data set conversion failure (V7)	Wrong Export Settings (T7)	V7T10
Undetected event (V8)	Human agent error (T8)	V8T12, V8T6, V8T12
Detection routine failure (V9)	Human agent malicious intent (T9)	V9T11
Incorrect parameters chosen (V10)	Bug in conversion software (T10)	V10T6,
	Bug in detection software (T11)	
	Unseen event type (T12)	

From Fig. 1, risk is calculated per block. The block is defined in terms of an entity, activity and associated agent:

$$R_{block} \in (R_{ent}, R_{act}, R_{Ag}); \quad (1)$$

where R_{ent}, R_{act}, R_{Ag} is risk of an entity, activity and agent respectively.

STRAPP is querying target system for an activity ID and string of risks with regard to this activity. The system should respond with a string of risks of an entity, activity and agent:

$$R_{ent}, R_{act} = \{R_1 \ldots R_n\}; \quad (2)$$

Risk for an agent is defined in terms of agents' years of experience and assigned a factor from a scale of 0 to 1, where 1 is very experienced (e.g., more than 10 years experience, and 0 – no experience at all). As such, risk for an agent can be scaled as follows:

$$R_{Ag} \in [0.33, 0.66, 0.99]; \quad (3)$$

Risk per block is aggregated as follows:

$$R_{agg_{act\_ID}} = 1 - (1 - R_1) * (1 - R_n) * R_{Ag}; \quad (4)$$

Overall aggregated risk of a chain under analysis is calculated as follows:

$$R_{total} = 1 - \left(1 - R_{agg_{act_{ID_1}}}\right) * \left(1 - R_{agg_{act_{ID_2}}}\right) \ldots (1 - R_{agg_{actID_n}}) \quad (5)$$

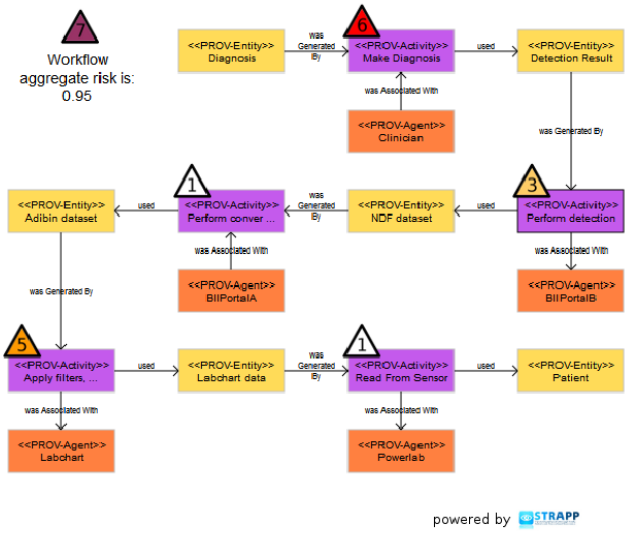

Figure 1. Risk output

Activity "*Make Diagnosis*" and agent "*Clinician*" has got high risk level. This is because agent's risk is defined in terms of its years of experience. Therefore, inexperienced clinician could make an incorrect diagnosis and result in a high aggregated workflow risk. More years of experience would dramatically reduce the overall risk of a final "*Diagnosis*".

B. Provenance-Based Trust of a Domain-Based Workflow

Our trust model is concerned with the ranking of decision alternatives over a number of attributes. Based on a case study data, some of the attributes can be incomplete. There are numerous methods to aid decision makers solve multi-attribute decision making (MADM) problems with incomplete information, amongst these methods the analytic hierarchy process (AHP) has been widely used, originally proposed by Saaty [13].

Our trust algorithm first identifies all possible focal elements from incomplete decision matrix, then it calculates the Basic Probability Assignment (BPA) of each focal element. Second, belief interval of each decision alternative is evaluated according Dempster-Shafer theory (DS). Third, by applying the ranking method decision, alternatives are determined by comparing their belief intervals. More details on DS/AHP and its application can be found in [14].

The following metadata fields shown in Table II contribute to the trust decision matrix:

 ISBN: 978-1-61208-362-9

TABLE II. TRUST METADATA

Field	Example value	Trust implications
Patient Identifier	KCH116	Conforms to expected format. These are a 3 letter centre ID concatenated to a three digit patient number. Trust is high or low based on presence/absence.
Centre	King's College Hospital	Reputation of centre. Should match with the patient identifier given above.
Sensor fitted by	John McAvoy	Experience/reputation of clinician. Initially based on the number of procedures carried out over the previous two years.
Data Administrator	Martyn Fletcher	Experience/training of data administrator. Each administrator is registered to upload data for a given centre- trust is reduced if data is uploaded for a different centre. Trust also based on number of files uploaded by the administrator (i.e. experience)
Data Channels	LPF0, LPF1,LPF2,HPF0,HPF1, HPF2,BP	Expected channels are present. Trust is reduced if channel names are not recognised as standard.
Recording frequency	200Hz	Is a standard recording frequency (200Hz and 400Hz are the current standards). Trust is reduced for other recording frequencies.
Start Date	21/01/2013	Date should be valid and in the past.
Recording Setup	Depth and strip probes through PowerLab	Trust is reduced in a less tried and tested setup. Where a large number of recordings have been made with a certain setup, the trust is increased.

Data on the BII portal contains provenance information about the services which were used to generate it, and the inputs to those services. This information is crucial in the determination of the level of trust which can be placed in the data. The following pieces of information shown in Table III are pertinent to the initial trust model, and will apply to all pieces of data/services in the provenance chain:

TABLE III. DATA PROVENANCE/SERVICE INFORMATION

Provenance Information	Example Value	Trust Implications
Service version	1.2	Should be the latest version of the service. Trust is reduced if an older service version was used. Additionally, some service versions may have known problems. Trust is greatly reduced where this is applicable.
Service creator	Stephen Hobson	Trust will be higher in service developers with more experience/better reputations
Number of service executions	1000	Trust will be higher in services which have been used a larger number of times
Data trustworthiness	0.9	Trust in this piece of data is partially defined by the trustworthiness of the data which was used to create it. This will have been defined using the trust model.

Some data, after analysis, will have some results associated with it, such as event detections. As part of this analysis, some measures may be available which would help determine the trustworthiness of the data. Initially these are limited, but could be increased in future. Table IV shows an example of such data:

TABLE IV. DATA

Data quality measure	Sample Value	Trust Implications
Channel Uptime	96.4%	This is a measure of the percentage of time that the channels in the file were providing "good quality data". Trust is higher where this value is higher.

Fig. 2 shows the trust levels derived by applying DS/AHP to input data shown in Tables II, III, IV. For every PROV element, trust level is estimated taking as an input a set of files with relevant data entries and applying DS/AHP algorithm the ranking is performed. As such, we have applied DS/AHP to rank the trust level at the source: Entity "Patient".

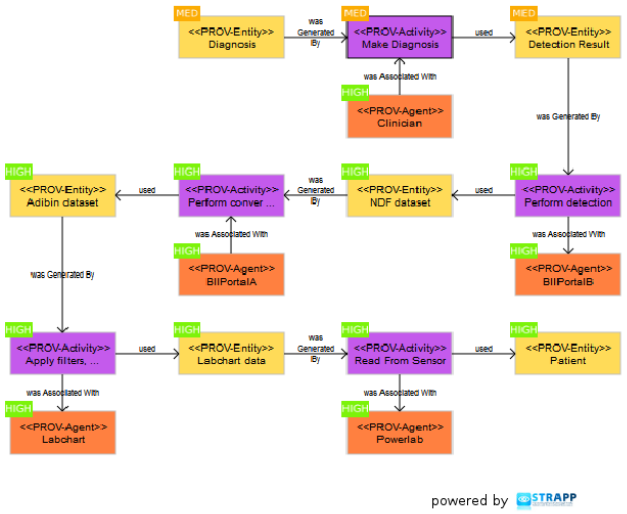

Figure 2 Trust output

The input to DS/AHP consists of 10 files, each with 8 data fields. As it can be seen in Table V, some of these fields are missing. Data fields such as patient_ID, center, sensor fitted by, administrator, data channels, recording frequency, and recording setup are treated equally, without emphasizing on importance. After running DS/AHP, it was derived that some of the files have low trust, e.g., *"sample.ps"*. This is because most of the data fields are empty, missing or incomplete. Medium trust files have several empty fields. Similarly, the set of data files relevant to activities within a workflow can be analysed and ranked according to DS/AHP as shown in Table V. The user of a system can then see at what stage data might get lost, corrupted or tempered with. Therefore, somebody knowing such situation would be interested in knowing possible consequences or risks associated with the decision trust.

Risk and trust can be seen implicitly. As such, we have demonstrated risk view on a basis of a workflow taking as an input risks relevant to data completeness, accuracy, relevance. It was seen that high risk activities may also result in high trust, if the data is of a high quality. As such, we can compare risk and trust of an activity *"Apply Filters"* from Fig. 1 and Fig. 2. In terms of risk – *"Apply Filters"* risk level is 5 (out of 7) and trust is high. Risk was calculated knowing that a number of threats and vulnerabilities are present and may harm the data quality of a data set. However, trust algorithm when applied on this activity has shown high trust in data set, as most of the data fields were complete. Therefore, we have made an assumption, that knowing that trust level in data is high does not necessarily mean it has low risk. Risk in our context is more associated with external factors which are not considered by the trust algorithm, e.g., software bug, software agent export failure.

 ISBN: 978-1-61208-362-9

TABLE V. TRUST DECISION MATRIX RANKING RESULTS

Filename	Patient Identifier (2.0)	Center (2.0)	Sensor Fitted By (5.0)	Data Administrator (5.0)	Data Channels (2.0)	Recording Frequency (2.0)	Start Date (2.0)	Recording Setup (2.0)	Trust / Distance Metric
KCH101.txt	KCH101 (2.0)	KCH (2.0)	John Smith (2.0)	John Smith (2.0)	BP (2.0)	200Hz (2.0)	2013-11-29T14:04:34 (2.0)	Standard (2.0)	HIGH 0.1039
KCH116 Section.ndf	KCH116 (2.0)	KCH (2.0)	(0.0)	Stephen Hobson (2.0)	LPF 0, LPF 1, LPF 2, LPF 3, LPF 4, LPF 5, HPF 0, HPF 1, HPF 2, HPF 3, HPF 4, HPF 5, BP, ICP (2.0)	200Hz (2.0)	2012-01-08T19:23:29 (2.0)	(0.0)	HIGH 0.16
KCH116 Section.mat	KCH116 (2.0)	KCH (2.0)	(0.0)	Stephen Hobson (2.0)	LPF 0, LPF 1, LPF 2, LPF 3, LPF 4, LPF 5, HPF 0, HPF 1, HPF 2, HPF 3, HPF 4, HPF 5, BP, ICP (2.0)	200Hz (2.0)	2012-01-08T19:23:29 (2.0)	(0.0)	HIGH 0.16
KCH011.ndf	KCH011 (2.0)	KCH (2.0)	(0.0)	Stephen Hobson (2.0)	BP (2.0)	200Hz (2.0)	2004-02-01T18:37:18 (2.0)	(0.0)	HIGH 0.16
KCH011.mat	KCH011 (2.0)	KCH (2.0)	(0.0)	Stephen Hobson (2.0)	BP (2.0)	200Hz (2.0)	2004-02-01T18:37:18 (2.0)	(0.0)	HIGH 0.16
KCH11522.ndf	KCH115 (2.0)	KCH (2.0)	(0.0)	Stephen Hobson (2.0)	BP (2.0)	(0.0)	2011-12-21T17:40:49 (2.0)	(0.0)	HIGH 0.187
KCH11522.mat	KCH115 (2.0)	KCH (2.0)	(0.0)	Stephen Hobson (2.0)	BP (2.0)	(0.0)	2011-12-21T17:40:49 (2.0)	(0.0)	HIGH 0.187
PG 29Apr_2ver1.ndf	PG29 (2.0)	PG (0.0)	(0.0)	(0.0)	(0.0)	200Hz (2.0)	2009-04-29T19:52:41 (2.0)	(0.0)	MED 0.2887
PG 29Apr_2ver1.mat	PG29 (2.0)	PG (0.0)	(0.0)	(0.0)	(0.0)	200Hz (2.0)	2009-04-29T19:52:41 (2.0)	(0.0)	MED 0.2887
sample.ps	Ps (2.0)	Ps (0.0)	(0.0)	(0.0)	(0.0)	(0.0)	Ps (0.0)	(0.0)	LOW 0.5299

V. CONCLUSION AND FUTURE WORK

Considering that trust and risk are implicit, this paper proposes for the first time a novel approach to view trust and risk on a basis of a W3C PROV provenance data model applied in the healthcare domain. We have made an assumption that high trust in data does not necessarily mean low risk, as these factors fulfill each other rather than being seen independently. This is demonstrated by our trust and risk models applied to the Brain Injury Index (BII) case study data. We first, present the risk model, which first starts by calculating risk values at each workflow step considering PROV concepts and second, aggregates the final risk score for the whole provenance chain. Different from risk model, trust of a workflow is derived by applying DS/AHP method. In situation when user should make a critical decision, users should be aware of possible outcomes and their probabilities, risks to be taken and uncertainties involved in the analysis as well as provenance of information. The system is trustworthy when these aspects are open to the system user. The evaluation of such system will be performed under the STRAPP context in the medical domain. We make a hypothesis that if user is aware of risks and trust levels involved in the PROV chain the trustworthiness in a system can be improved. Therefore, more analysis needs to be done in the area of risk and trust. Nevertheless, our first attempt of visualizing risk and trust concepts on a workflow basis and making a relational comparison of derived results proved our assumption that risk and trust are implicit, not proportional.

ACKNOWLEDGMENT

The STRAPP project (Trusted Digital Spaces through Timely Reliable and Personalised Provenance) is funded by the UK Technology Strategy Board (Grant reference 1926-19253), Rolls-Royce plc, OSyS Ltd, Cybula Ltd, and the UK Engineering and Physical Sciences Research Council Knowledge Secondment Scheme. Their support is gratefully acknowledged.

REFERENCES

[1] D. Gambetta, "Can We Trust Trust?", in D. Gambetta (Ed.) Trust: Making and Breaking Cooperative Relations, Electronic edition, Department of Sociology, University of Oxford, chapter 13, pp. 213-237, 2000

[2] B. Solhaug, D. Elgesem, and K. Stølen. "Why trust is not proportional to risk", 2nd International Conference on Availability, Reliability and Security (AReS'07), pp. 11-18, IEEE Computer Society, 2007.

[3] O.E Williamson, "Calculativeness, Trust, and Economic Organization", Journal of Law and Economics, University of Chicago Press, vol. 36(1), pp. 453-86, April 1993

[4] A.Jøsang, C. Keser, and T. Dimitrakos, "Can We Manage Trust? ", in Trust Management, P. Herrmann, V. Issarny, and S. Shiu (Eds.), Springer, Berlin Heidelberg. pp. 93-107, 2005

[5] A. Josang and S. L. Presti, "Analysing the Relationship Between Risk and Trust", 2nd International Conference on Trust Management (iTrust'2004), Oxford, UK, Springer, pp. 135-145, April 2004.

[6] M. Emaldi et al., "To trust, or not to trust: Highlighting the need for data provenance in mobile apps for smart cities", Computer Vol. 15, pp. 26-32, 2013

[7] O. Hartig and J. Zhao, "Using web data provenance for quality assessment", In: Proc. of the Workshop on Semantic Web and Provenance Management at ISWC, 2009

[8] D. Ceolin, P. Groth, W. R. van Hage, A. Nottamkandath, and W. J. Fokkink, "Trust evaluation through user reputation and provenance analysis", 8th Workshop on Uncertainty Reasoning for the Semantic Web (URSW'2012), Boston, Massachussetts, pp. 15-26, Nov. 2012

[9] S. Rajbhandari, O. F. Rana, and I. Wootten. "A fuzzy model for calculating workflow trust using provenance data", 15th ACM Mardi Gras conference, New York, NY, USA, pp. 1–8, 2008. ACM.

[10] V. Cahill et al. ,"Using trust for secure collaboration in uncertain environments", Pervasive Computing, IEEE , Vol. 2, No.3, pp.52-61, July 2003

[11] L. Moreau and P. Missier, "PROV-dm: The prov data model". W3C Recommendation, April 2013

[12] D. Ceolin, P. Groth, and W.R.van Hage, "Calculating the Trust of Event Descriptions using Provenance", Second International Workshop on the role of Semantic Web in Provenance Management (SWPM'10)

[13] T.L.Saaty, "The Analytic Hierarchy Process", New York:McGraw-Hill, 1980

[14] Z. Hua, B. Gong, and X. Xu, "A DS–AHP approach for multi-attribute decision making problem with incomplete information", Expert Systems with Applications, Vol. 34, iss. 3, pp. 2221-2227, ISSN 0957-4174, April 2008

[15] T. Lebo, S. Sahoo, and D. McGuinness, "PROV-O: The PROV Ontology", W3C Recommendation, April 2013

[16] W3C OWL Working Group, "OWL2 Web Ontology Language", W3C Recommendation, December 2012

[17] STRAPP: https://www.engineering.leeds.ac.uk/strapp

 ISBN: 978-1-61208-362-9

A Security Policy for Cloud Providers

The Software-as-a-Service Model

Dimitra Georgiou
Secure Systems Laboratory
Department of Digital Systems
School of Information & Communication Technologies
University of Piraeus, Piraeus, Greece
dimitrageorgiou@ssl-unipi.gr

Costas Lambrinoudakis
Secure Systems Laboratory
Department of Digital Systems
School of Information & Communication Technologies
University of Piraeus, Piraeus, Greece
clam@unipi.gr

***Abstract*—Cloud Computing is a new computing paradigm originating and combining characteristics from grid computing, distributed computing, parallel computing, virtualization and other computer technologies. Trust and security in Cloud Computing are more complex than in traditional IT systems. Conventional security policies designed for other technologies do not map well to the cloud environment, which, on top of that, may exhibit additional security requirements. In an attempt to assist cloud providers to secure their environment, and specifically for the Software-as-a-Service Model (SaaS), this paper starts with the presentation of the already reported threats. Because of these security threats, there are specific requirements that we claim must be clearly addressed in the Security Policy for the Cloud Environment. Our work focuses on the required structure and contents of such a security policy. In this respect, this paper proposes a model to describe the relationship between threats, measures, and security policies applicable to the SaaS model. It is worth stressing that in the SaaS service model, the client depends on the provider for the proper security measures.**

Keywords-Cloud Computing Security; Security Policies; Security Requirements; Software-as-a-Service (SaaS)

I. INTRODUCTION

Nowadays, in an interconnected world, every corporation needs a very well thought security policy. The rapid growth of the information age has significantly changed the nature of computing, and gives rise to a new set of security concerns and issues. According to the National Institute of Standards and Technology (NIST), the Security Policy is defined as an "Aggregate of directives, regulations, rules, and practices that prescribes how an organization manages, protects and distributes information"[1].

For the new era of Cloud Computing**,** the purpose of a security policy is to protect people and information, set rules for expected behavior by users, minimize risks and help to track compliances with regulation[2].Considering the fact that in recent times anyone with an interest in information technology has come across the term Cloud Computing [3], it is really important to seriously consider the security issues in Cloud Computing: Are there any Security threats in Cloud Computing, that do not appear in non- Cloud Systems? Is the Cloud secure and safe for the users? As Cloud Computing is achieving popularity, we attempt to demystify the security and privacy risks that are introduced, because of its transformational nature [4].The success of a Cloud Policy really depends on the way the security contents are addressed in the policy document and how the content is communicated to users [5]. But, before we analyze all these risks, we need to have a clear understanding of what "Cloud Computing" is.

Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [6].Cloud is a recent trend in Information Technology that moves computing and data away from desktop and portable PCs into large data centers. It refers to applications delivered as services over the Internet, as well as to the actual cloud infrastructure, namely, the hardware and systems software in data centers that provide these services [7] (see Figure 1).

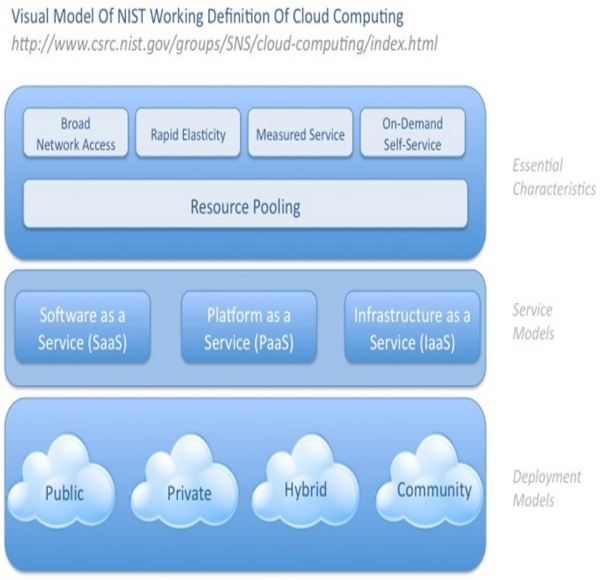

Figure 1. Visual Model of NIST Working Definition of Cloud Computing [7]

The advantages of Cloud Computing and specifically its ability to scale rapidly (through subcontractors), store data remotely (in unknown places) and share services in a dynamic environment, can become a major flow in maintaining a level of privacy assurance sufficient to sustain confidence in potential customers. Cloud has exacerbated the strain on traditional frameworks for privacy that globalization has already started. To understand the importance of Cloud

 ISBN: 978-1-61208-362-9

Computing and its adoption, we must understand its principal characteristics, its delivery and deployment models, how customers use these services, and how to safeguard them.

There are three service models of Cloud Computing: Software-as-a-Service **(SaaS)**, Platform-as-a-Service **(PaaS),** Infrastructure-as-a-Service **(IaaS)** and three main deployment models which are: Private cloud, Public cloud and Hybric cloud [8][9][10][11][12][13] (see Figure 2). These service models also place a different level of security requirements in the cloud environment. IaaS is the foundation of all cloud services, with PaaS built up on it and SaaS in turn built upon it. Just as capabilities are inherited, so are the information security issues and risks.

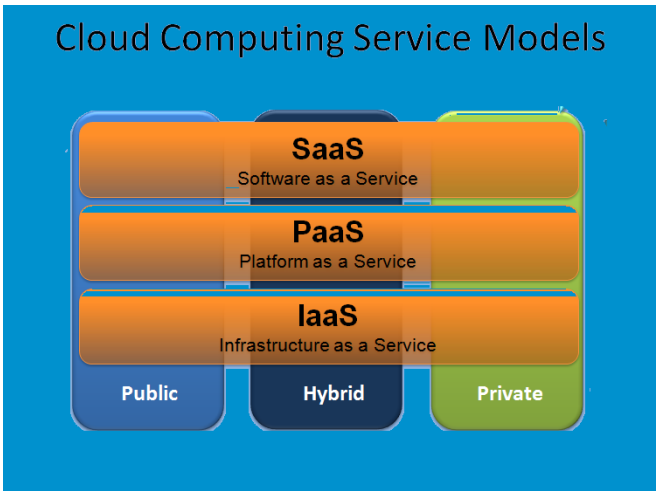

Figure 2. Cloud Computing Service Models

This paper proposes a methodology that may be adopted for the development of a Cloud Security Policy. It assesses how security, trust and privacy issues can be addressed in the context of a Cloud Computing Policy and is organized as follows: Section II presents an overview of related work on security issues and security Policies for Cloud Computing. Then, Section III, analyses the policy issues related to Cloud Computing, while Section IV depicts the proposed methodology for a Cloud Security Policy, for Cloud Providers in the SaaS service model. Section V presents the linking of threats, security measures and security policy rules for Threat 5 (Introduction of damaging or disruptive software) and finally, Section VI concludes the paper and provides some pointers for future work.

II. RELATED WORK

Cloud Computing is a new computing model originating from grid computing, distributed computing, parallel computing, virtualization technology, utility computing and other computer technologies. It exhibits many advantages such as large scale computation and data storage, virtualization, high expandability, high reliability and low price service. Trust and security in Cloud Computing are more complex than in a traditional IT systems. But, what exactly is the problem?

In order to have a secure Cloud Computing deployment, it is necessary to consider the following areas: the Cloud Computing architecture, governance, portability and interoperability, traditional security, business continuity and disaster recovery, data center operations, incident response, notification and remediation, application security, encryption and key management, identity and access management [14][15][16]. Many of the security issues arising from the aforementioned areas, have been already addressed in other systems. However, the specific characteristics of cloud environments result into new security concerns; Cloud architecture is fundamentally different from other systems, the cloud environment is by nature multitenant with shared resources, and the location of the data and the local privacy requirements will not be controlled by the user. Another major problem is the lack of standardization. Since no proper standards for Cloud Computing exist, it becomes extremely difficult for a company to secure the services that it offers or uses through a cloud.

Cloud Computing security challenges and issues have been addressed by various researchers. The National Institute of Standards and Technology contends that security, interoperability, and portability are the major barriers to a broader cloud adoption [17]. Data confidentiality and service availability in Cloud Computing are also key security issues. A single security method cannot solve the Cloud Computing security problem and many conventional and new technologies and strategies must be employed together for protecting the entire Cloud environment.

Robert Gellman's report at the World Privacy Forum [18] focuses on privacy issues and legal compliance of sharing data in the cloud. He mentions various legal issues such as the possibility of the cloud being in more than one legal location at the same time with different legal consequences and such uncertainty making it very difficult to assess the privacy protection level offered to the users [19]. Also, ENISA investigated the different security risks related to adopting Cloud Computing along with the affected assets, the risks likelihood, impacts, and vulnerabilities in Cloud Computing that may lead to such risks [20].

According to Al Morsy et al. [21] the Cloud Computing model has different stakeholders involved, namely: cloud provider, service provider and service consumer. Each stakeholder has its own security management systems/processes and each one has its own expectations (requirements) from the other stakeholders.

Cloud environments exhibit different architectures based on the services they provide, thus making it even harder to find 'global' security measures .Louay Karadsheh [22] examines the risks encountered by implementing the Infrastructure-as-a-Service (IaaS) model and discusses the role of security policies, Service Level Agreement (SLA) and compliance for enhancing the security of the IaaS service model. Subashini and Kavitha [13] describe the various security issues of Cloud Computing in relation to its service delivery model and they list some of the existing solutions that partly address the security challenges posed by the cloud. Cheng and Lai [23] introduce the characteristics of the newly developed Cloud Computing technology first, and then they highlight the reasons for emphasizing the issue of information privacy in relation to new cloud applications. Vaquero et al. [24] analyze the security risks that multitenancy induces the Infrastructure-as-a-Service clouds and present the most relevant threats and relevant state of the art of solutions. Also, in the same paper they continue analyzing the open security issues and challenges that should be addressed. Even though the majority of the research work published focuses on security issues, legal and jurisdictional

 ISBN: 978-1-61208-362-9

risks [25][26][27], none addresses the need for a Cloud Security Policy. For instance, Karadsheh [22] discusses the role of security policies, SLA and compliance for enhancing the security of the IaaS service model, by presenting several applicable policies. Furthermore, this paper discusses the possibilities of applying different types of security policies to enhance security of IaaS to acceptable level, but they do not propose a security policy. Similar is the approach by Subashini and Kavitha [13], who describe the common security issues posed by the cloud service delivery models and the security threats posed by the IaaS delivery model, but they do not provide a comprehensive analysis of the specific threats to be addressed by cloud providers.

In an attempt to assist cloud providers to secure the environment that they offer, and specifically for the Software-as-a-Service Model (SaaS), this paper presents the already reported threats to ease their comprehension. Because of these security threats, there are specific requirements that we claim must be clearly addressed in the Security Policy for the Cloud Environment. Our work focuses on the required structure and contents of such a security policy.

III. AN ANALYSIS OF THE SECURITY POLICY ISSUES RELATED TO CLOUD COMPUTING

The Cloud Computing model involves different stakeholders: the Cloud Provider (an entity that offers the cloud infrastructure or /and services to the cloud consumers), the Service Provider (an entity that utilizes the cloud infrastructure to deliver applications/services to the end users) and the Service Consumer (End user; an entity that uses services hosted on the cloud infrastructure). Each stakeholder has its own expectations (requirements) and security management systems/processes [21]. For instance, if we consider user expectations they would expect that the cloud provides: reliability and liability, security, privacy, anonymity, access and usage restrictions [28].

The decision of whether the Cloud Customer or the Cloud Provider (Service Provider) is responsible for a given control and for security and privacy depends on three factors:

a) The cloud model (SaaS, IaaS, or PaaS) chosen.
b) The extent to which the Cloud Customer is allowed to configure the CP's controls.
c) Legislations, which may dictate the assignment of responsibilities and thereby overrides the previous two factors.

Next, we highlight the possible threats for a Cloud Provider who adopts the Software-as-a-Service (SaaS) model:

Threat 1: **Masquerading of user identity by insiders:** The threat of masquerading of user identity by insiders covers attempts by authorized users to gain access to information to which they have not been granted access. These users may attempt to gain access to that information by using another user's account.

Threat 2: **Masquerading of user identity by contracted service Providers:** The threat of masquerading of a user identity by contracted service providers covers attempts by people working for a contracted service provider to obtain unauthorized access to information by using an authorized person.

Threat 3: **Masquerading of user identity by outsiders:** The threat of masquerading of a user identity by outsiders covers attempts by outsiders to obtain unauthorized access to information by posing as an authorized user.

Threat 4: **Unauthorized use of an application:** It covers various cases of unauthorized use of an application.

Threat 5: **Introduction of damaging or disruptive software:** This threat covers Viruses, Worms, Trojan Horses, logic bombs, any other form of malicious software.

Threat 6: **Misuse of system resources:** Identifies factors that increase the threat of misuse of system resources; covers People playing games on business systems, People using business systems for personal work, People downloading non-work related information from the internet, People setting up databases or other packages for non-work related matters.

Threat 7: **Communications infiltration:** This threat covers the following types of event: Hacking into a system using, for example, buffer overflow attacks, Masquerading as a server, Masquerading as an existing user of an e-commerce application, Masquerading as a new user of an e-commerce application, Denial of service (deliberate), Flaming attacks, and Spamming.

Threat 8: **Communications interception:** This threat covers Passive interception and Traffic monitoring. The ease of interception is determined by two basic-factors: The medium of transmission and the type of protocols being used. Interception of some types of traffic on the internet is relatively easy. It can be achieved by attackers sending messages to target systems instructing them to send traffic via specific (hostile) machines.

Threat 9: **Communications manipulation:** Active interception, Insertion of false messages, Deliberate delivery out of sequence, Deliberate delay of delivery, Deliberate misrouting. If an attacker can force a message to be sent via a hostile host, the attacker may be in a position to intercept, alter and the forward the message.

Threat 10: **Repudiation:** This threat addresses cases of people denying that they sent a message (repudiation of origin), or that they received a message (repudiation of receipt).

 ISBN: 978-1-61208-362-9

Threat 11: **Communications failure:** Unavailability of Service Provider, Failure of data link, Non – delivery of message, Accidental delivery out of sequence, Accidental delay in delivery, Accidental denial of service. The Internet does not provide a service level agreement. There are no guarantees on how long it will take for a message to get to a recipient, or even that it will get there, eventually.

Threat 12: **Embedding of malicious code:** Includes email viruses and hostile mobile code (for example hostile Active X applets). Once on a network, they can quickly infect many machines causing significant disruption. Java and Active X raise a range of new security concerns. Users are now running code written by people from outside of the organization, sometimes from unknown sources. This code has often not been tested by the organization. There are concerns that hostile code written using these types of techniques could inflict damage on systems and networks.

Threat 13: **Accidental misrouting:** The threat of accidental misrouting covers the possibility that information might be delivered to an incorrect address when it is being sent over a network.

Threat 14: **Technical failure of host:** This threat covers failures of the CPU or other hardware items.

Threat 15: **Technical failure of storage facility**: This threat covers disk crashes and disk failures.

Threat16: **Technical failure of Print facility:** This questionnaire identifies the factors that increase the threat for a technical failure of the print facility.

Threat 17: **Technical failure of network Distribution Component:** This threat addresses cases of network distribution components, such as bridges and routers, failure.

Threat 18: **Technical failure of Network Management or Operational Host:** This questionnaire identifies the factors that increase the threat of technical failure of a network management or operation host.

Threat 19 : **Technical Failure of Network Interface:** Here, the factors that increase the threat of failure of the network interface are identified.

Threat 20: **Technical failure of Network service:** Here, the factors that increase the threat of failure of the network service are identified.

Threat 21: **Power failure:** This threat covers the possibility that the power supply to the building may fail. The types o power failure covered include: spikes, surges, brown outs, black outs.

Threat 22: **Air conditioning failure:** This threat covers the possibility that operation may have to be suspended because temperatures in the location fall outside of acceptable parameters.

These threats are being used for illustrating where the dangerous points lurk at every level of the typical SaaS model in a Cloud Provider's environment.

In all three cloud models, the Cloud Provider manages and controls the infrastructure, which comprises the servers, networks, electricity, human resources, and site services. As such, the Cloud Provider is responsible to implement and operate suitable infrastructure controls such as employee training, physical site security, network firewalls, and others. Infrastructure controls are of fundamental importance. It is evident, from the complexity of Cloud Computing and the threats that the cloud is facing, that the development and adoption of a Security Policy is necessary. Understanding the threats relevant to the SaaS service model will assist in formulating a well-established security policy.

Although much research into cloud services security engineering has been undertaken and almost everybody accepts that there are a lot of security and privacy issues for Cloud Computing, no one has raised the need for a Security Policy for Cloud Computing.

IV. A SECURITY POLICY STRUCTURE FOR CLOUD COMPUTING

Existing research analysis methodologies are not appropriate for Cloud Computing since threats in Cloud are different. The appropriate Security policies designed for conventional architectures do not map well to the cloud environment. Cloud architectures must have well-defined security policies and procedures in place. As companies move to Cloud Computing, the traditional methods of securing data are being challenged. For instance, it may be difficult for the cloud customer to effectively control the data processing that the cloud provider carries out and thus to be sure that the data is handled in a lawful way. Failure to comply with data protection law may lead to administrative, civil and also criminal sanctions, which vary from country to country, for the data controller. It is therefore important all security requirements, including the ones that are only applicable to the cloud environments, to be covered by a security policy. Therefore in this paper we indeed provide a new methodology for assessing the threats/risks in Cloud, in order to identify new rules that must be incorporated in the Cloud Security Policy. The work, in this paper, does not result in a Cloud Security Policy. Instead, it proposes a methodology that may be used for the development of the appropriate Cloud Security Policy.

The proposed methodology for the development of a cloud Security Policy exhibits three distinct levels:

 ISBN: 978-1-61208-362-9

1) The Cloud Provider level,
2) The Service Provider level and
3) The User level.

Even though there are parts of the security policy that are common to all levels, each level will also exhibit dedicated security policy parts/rules. This three-layered classification of security requirements of cloud systems and the common parts of the Policy (colored) is illustrated in Figure 3. As already mentioned earlier, the focus will be on SaaS (Software-as-a-Service) models.

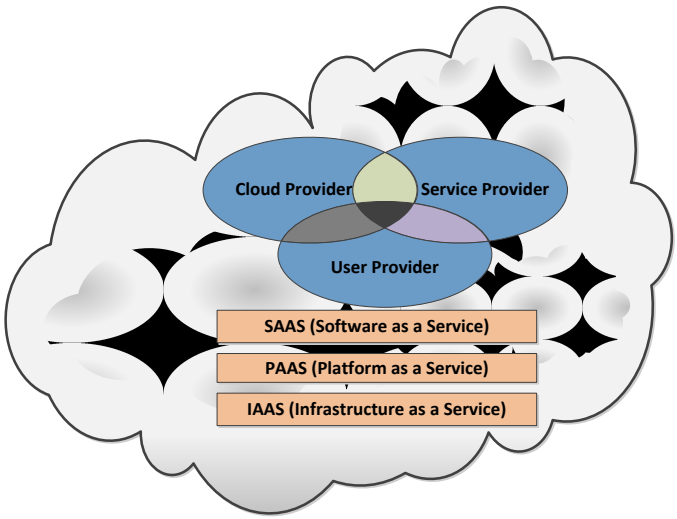

Figure 3. Security Policy Structure for Cloud Providers

The threats that we referred to in the previous section can be employed for deducing the security requirements that must be satisfied by the cloud provider.

To demonstrate this, a specific threat (Threat 5 - Introduction of damaging or disruptive software) has been chosen to depict the correlation between Threat -Requirement – Security Measures – Policy for a Cloud Provider (see Section V and Figure 4).

More specifically, in Figure 4, each security measure that can be employed for eliminating Threat 5 is associated with the necessary set of rules that make up the security policy of the cloud provider. The same information is provided in more detail with more analysis in Section V below. Doing this type of analysis for each Threat that the SaaS service model is facing will help in formulating a well-established security policy.

V. LINKING THREATS , SECURITY MEASURES AND SECURITY POLICY RULES

A. Threats

Next, Threat 5: *Introduction of damaging or disruptive software*, will be analyzed as an example. In parallel the security measures and policy rules linked to that threat will also be examined.

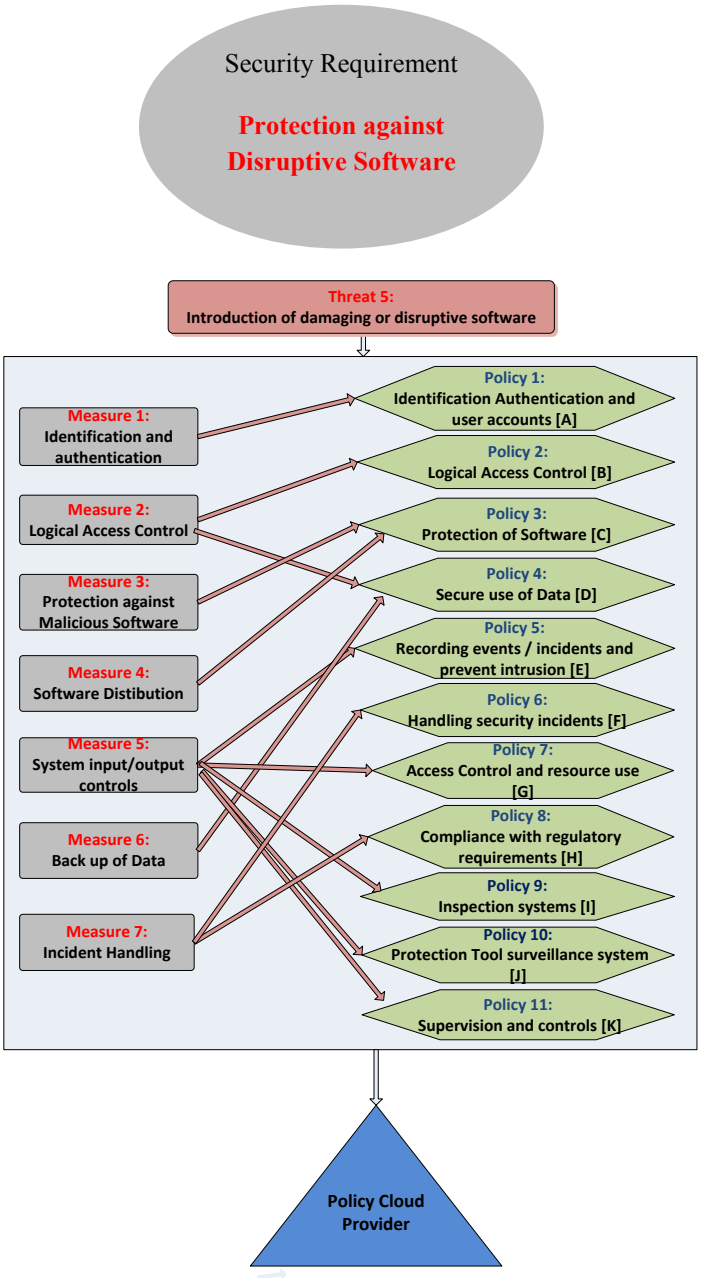

Figure 4. Security Policy rules covering Threat 5

B. Measures

The security measures associated with the aforementioned threat follow.

- Identification and authentication (Security Policy Rules A)
- Logical access control (Security Policy Rules B & D)
- Protection against malicious Software (Security Policy Rules C)
- Software Distribution (Security Policy Rules C)
- System input /output controls (Security Policy Rules E & G & I & J & K)
- Back-up of Data (Security Policy Rules D)
- Incident Handling (Security Policy Rules F & H)

 ISBN: 978-1-61208-362-9

C. Security Policy Rules

The security policy rules associated with the aforementioned threat and security measures follow.

1) Identification and authentication

Users are identified uniquely ensuring that any action can be attributed to a specific user. This rule applies to the operating system level and to the application level, while the following minimum requirements should be satisfied.

- Each user has a unique identity (user ID).
- A list of users and their unique identities is maintained.
- Each authentication identifier is assigned to a user and is used by a single user.
- The system administrators have identities that correspond to accounts with elevated privileges.

2) Logical access control

There shall be a formal user registration and de-registration procedure in place for granting and revoking access to all information systems and services. Specifically:

- Registered user accounts shall be reviewed for applicability at specific periods.
- Privileges shall be defined for specific business purposes.
- The allocation and use of privileges shall be restricted and controlled.
- Privileges and privilege allocation shall be reviewed for applicability at specified periods.
- The allocation and establishment of user passwords shall be controlled through a formal management process.
- Management shall review user rights at regular intervals using a formal process.
- Users shall be required to follow good security practices in the selection use of passwords.

3) Protection software

Special care should be taken to control the development and maintenance of software applications. Specifically:

- Application development should be conducted with specific, scientifically accepted methodologies.
- Each new application must be accompanied by sufficient documentation in accordance with international standards.
- The risk analysis must fit into the requirements analysis.
- Systems utilized for the development and testing of software must be separate from the operational systems.

Software changes should be authorized prior to their implementation:

- Application software changes require approval by their respective makers.
- Any proposed change should be examined whether it affects the security of the information system.

Changes that affect - directly or indirectly - security requirements must be approved by the Security Officer. Specifically:

- The amendments must be made in the development/testing environment and should be tested prior to their application to the operational system.
- All changes must be characterized by a unique serial number.
- At each change request it is necessary to record the corresponding date and the name of the applicant.
- All software changes must be accompanied by documentation updates.

In case where urgent changes are required, it is necessary to ensure the following:

- Keep to a minimum the changes that will be performed.
- The modified files must be monitored.
- The Security Officer must be informed.
- Irrespectively of how urgent are the modifications, they must be tested before they are incorporated in the live system.

After any kind of modifications on the live system it is necessary to re-test system security. To this end the security officer must monitor the effectiveness of the security mechanisms after the modification took place.

4) Secure Data Management

Data should be categorized according to the protection they need, as derived from the risk analysis or assessment of the head of Information System. The following categories have been identified:

- Top secret: information and critical data of the Information System that any disclosure or unauthorized modification will have direct impacts on the operation.
- Confidential: information and data that is important for seamless operation and should be subject to strict controls and protected.
- Sensitive: information and data that is subject to legislation on protection of personal data. Disclosure of this data requires specific permission / license.
- Reportable: information and data that can be disclosed.

The requirements of information security and the way data is processed vary according to the category of information. It is necessary to specify the authorized data recipients, according to the above classification. Data processing must ensure procedural and technical resources that can be attributed to a specific individual. Therefore, all critical operations will be accessed in a strictly personalized way.

5) Recording actions / events and intrusion prevention

Incidents of failure or non-routine functions of hardware or/and software, should be recorded and evaluated in relation to the operation that they support. Critical application systems should exhibit real time alarm systems. If there is a risk of invasion by external systems, intrusion detection and prevention systems should be in place. Systems will record the suspicious actions for the invasion and react automatically if this is dangerous for the security of the Cloud Provider. Proven invasions activate alarm system in real-time. The log files should be protected from loss or intentional corruption. The logs will be inspected by authorized personnel from time to time to highlight events / actions that endangered the Service Provider.

 ISBN: 978-1-61208-362-9

6) Handling security incidents

A procedure for reporting faults and general security incidents is mandatory. There should be documented procedures that will ensure the timely and effective response to the occurrence of a security incident. This framework should include:

- The roles and responsibilities to be undertaken.
- Recorded evidence of what happened.
- Rescuing electronic material proving the breach (e.g., unchanged medium).
- The process of identifying the cause of the break up.
- The process of recovery.

7) Access control and resource use

A strict registration process should be in place. As a minimum it should support the following:

- The access rights are determined through a rigorous registration process.
- The new system users are required to submit an application in order to obtain an account.
- The application contains the elements of the applicant's position and the department to which she belongs.
- The application is signed by the user and her supervisor and is forwarded to the IT director.
- The rights granted are always appropriate for the purpose that they serve.
- Inspections must be conducted by the Security Officer.
- If a user changes responsibilities and requires a new set of usage rights, she should request it through a new application.
- When a user is given a new set of usage rights, old rights he should be removed.
- Users should take care of the safe use of their accounts.
- The idle time of a workstation should be limited. After some time of inactivity, workstations should lock (e.g., password protected screen saver).

Regarding the use of system resources it is necessary to keep a list of all IT resources (hardware, software and documentation) and to record the classification level of each resource.

Furthermore an Access Control Policy is necessary for controlling access to the resources of the Information System. The access control policy should exhibit the following:

- The access policy setting takes into account the principle of « need to know» (need-to-know).
- Users can use only the applications and the resources needed to perform the tasks associated with their position.
- The use rights assigned to each user category are inspected at least once every six months, with the responsibility of IS Security Officer to ensure that it is not given more rights than necessary.
- A copy of the password of the system administrator account must be kept in a safe place. The access to stored passwords should be controlled.
- System administrators should use different passwords for administrative accounts and the accounts they use as ordinary users.
- The exercise of rights of access users will be monitored and controlled in order to avoid the abuse of rights.

8) Compliance with regulatory requirements

It is necessary to comply with existing legal and regulatory framework. Specifically:

- Monitor all legal and regulatory requirements and examine how they can be satisfied.
- Notification of the Data Protection Authority for keeping personal data.
- If records of sensitive data are kept, permission from the Data Protection Authority is necessary.
- Description of procedures to ensure the fulfillment of legal obligations for use hardware / software, ie the necessary licenses.
- Employ the necessary measures for protecting critical data from loss, destruction and unauthorized amendment in accordance with legislative requirements.
- Employ the necessary measures to ensure data protection and privacy as required by laws and regulations.
- Monitor and comply with all existing technical standards.

9) Inspection systems

Determine all audit requirements in accordance to the existing legal and regulatory framework, as well as the procedures for controlled access to inspection tools in order to avoid damage, loss or misuse.

10) Protection of surveillance system

Access to the tools of IS surveillance shall be controlled. Specifically:

- Access to the monitoring tools should be restricted to authorized persons.
- Ensure that maintenance contractors will not have access to surveillance tools. If they need some data they should be provided by the system administrators according to the need-to-know principle.
- Restrict the access rights of the administrators in order to ensure that they will not be able to remove or change registration details of their own actions.
- In order to facilitate correct monitoring, the clocks of different systems must be synchronized.

11) Supervision and control

Audit trails and event logs must be recorded in order to support the identification of violations or attempted violations and scrutinizing every suspicious incident. To this end the following are necessary:

- To maintain monitoring data for all systems supporting multi-user access.
- To use special software for managing these files.
- To record the use of privileged functions.
- To record system startup.

 ISBN: 978-1-61208-362-9

- To record failed attempts.
- To record binding energy (log-on).
- To record disconnect actions (log-off).
- To record changes in access rights and use.
- To record the basic data for each suspected case.
- To record the user identifiers (User IDs).
- To record the time and the time of the event.
- To record the type of the event.
- To record the files accessed.
- To record the identity of the station.
- To record the state of the data before and after the changes.
- A copy of the audit data files must be kept in back up media (back-up).
- Data must be kept at least for a period of three months. In systems that manage classified information, data must be retained for the period specified by the national safety regulations.
- Copies are kept in a safe place, so to prevent any theft or sabotage.
- Access to log files is prohibited in those that do not have privileges (administrative rights).
- Log files should be protected from potential disaster.
- There should be integrity checks in place.
- Log files should be tested at least once a year.
- If the space available for log files reaches 75% of its storage capacity, an alarm must be produced.
- Inform users which of their activities are recorded by the system.
- Analyze logs of actions and events.
- Monitor the creation of accounts with elevated permissions.
- Identify deviations from normal use of system resources (e.g. unusually large number of prints from a user).
- The system automatically notifies the Security Officer when it detects certain suspicious events.

VI. CONCLUSION

Cloud Computing is a very promising technology that helps companies reduce operating costs while increasing efficiency. Even though Cloud Computing has been deployed and used in production environments, security in Cloud Computing is still in its infancy and needs more research attention. This paper reviews the potential threats for the Software-as-a-Service Model (SaaS), in an attempt to assist cloud providers identifying the security requirements and securing the environment that they offer.

We claim that by linking each threat to the security measures that can be utilized for eliminating it, and in turn, with the security rules that are necessary for the implementation of the security measures, a Security Policy for cloud providers that clearly addresses the specific threats can be defined. The aforementioned correlation / linking is provided indicatively, for only one of the identified threats.

REFERENCES

[1] National Institute of Standards and Technology, systems, “Guide for developing security plans for federal information systems”, vol. 800-18, February 2006, [Online]. Available from:http://csrc.nist.gov/publications/nistpubs/800-18-Rev1/sp800-18-Rev1-final.pdf/, [accessed December 2013].

[2] Divers S. - SANS Institute, "Information Security Policy A development Guide for large and small companies", November 2007, pp. 43-44.

[3] Svantesson D. and Clarke R., “Privacy and consumer risks in Cloud Computing", Computer Law and Security Review, vol. 26, 2010, pp. 391-397.

[4] Kshetri, N., “Privacy and security issues in Cloud Computing: The role of institutions and institutional evolution”. 2012, Bryan School of Business and Economics, The Univ. of North Carolina at Greensboro, NC27402-6165,USA.

[5] Hone K., Eloff J. H., “Information security policy: what do international information security standards say?”, Proc. of the 8th European Conference on Information Warfare and Security, Computers and Security, vol. 21, Issue 5, 2002, pp. 402- 409.

[6] Mellet P.and Grance T., “The NIST Definition of Cloud Computing”, NIST, 2011, Special Publications 800-145.

[7] Dikaiakos, Katsaros M.D., Mehra D., Pallis P. and Vakali G, “Cloud Computing Distributed Internet Computing for IT and Scientific Research”, IEEE Press 2009, vol. 13, Issue: 5, pp. 10-13.

[8] European Network and Information Security Agency (Enisa), “Cloud Computing Benefits, risks and recommendations for information security”, November 2009.

[9] Arnold S., “Cloud Computing and the issues of privacy”, July 2009, KM World, pp.14-22

[10] Whitepaper, A, “Enterprise Cloud Computing: Transforming IT”, Platform Computing, viewed 13 March 2010, pp.6.

[11] Global Netoptex Incorporated, “Demystifying the cloud. Important opportunities, crucial choices”. 13 December, 2009, pp. 4-14., [Online]. Available from: http://www.gni.

[12] Kuyoro S.O., "Cloud Computing Security Issues and Challenges", Proc. International Journal of Computer Networks (IJCN), 2011, vol. 3, Issue: 5.

[13] KavithaV. and Subashini S.,“A survey on security issues in service delivery models of cloud”, International Journal of Network and Computer Applications, January 2011, vol. 34 Issue 1, pp.1-11

[14] Brodkin J., “Gartner: Seven cloud- computing security risks”, NetworkWorld, April 2013, [Online]. Available from: http://www.idi.ntnu.no/emner/tdt60/papers/Cloud_Computing_Security_Risk.pdf

[15] Okuhara M. et al- FUJITSU, “Security Architecture for Cloud Computing”, vol. 46, no 4, October 2010, Sci.Tecch.J., pp.397- 402].

[16] Min Y., Shin H., Bang Y., “Cloud Computing Security Issues and Access Control Solutions”, Journal of Security Engineering, February 2012, vol. 9, no2.

[17] National Institute of Standards and Technology, “Cloud Computing Synopsis and Recommendations”, May 2012, Special Publication 800-146.

[18] Gellman R., “Privacy in the Clouds: Risks to Privacy and Confidentiality from Cloud Computing”, World Privacy Forum February 2009, [Online]. Available from: http://www.scribd.com/doc/12805751/ Privacy-in-Cloud-Computing-World-Privacy-Council-Feb-2009, [accessed November 2013].

[19] Chadwick W.D. and Fatema K., "A privacy preserving authorisation system for the cloud", November 2012.

[20] European Network and Information Security Agency, “Cloud Computing benefits, risks and recommendations for information security”, 2009.

 ISBN: 978-1-61208-362-9

[21] Morsy M. Al., Grundy J. and Müller I., "An Analysis of the Cloud Computing Security Problem", Proc. APSEC 2010 Cloud Workshop, Sydney, Australia, 2010.

[22] Karadsheh L. "Applying security policies and service level agreement to IaaS service model to enhance security and transition "Computers & Security, vol. 31, Issue 3, May 2012, pp. 315-326.

[23] Cheng F., and Lai W., "The impact of Cloud Computing Technology on Legal Infrastructure within Internet-Focusing on the Protection of Information Privacy", Proc International Workshop on Information and Electronics Engineering, Elsevier Ltd Press 2012, vol.29, pp.241-251, doi: 10.1016/j.proeng.2011.12.701

[24] Vaquero M., Rodero-Merino L., and Moran D., " Locking *the* Sky*:* A Survey on IaaS Cloud Security Computing", Springer, Press, January 2011, vol. 91*,* Number 1*,* pp. 93-118*.* doi: 10.1007/s00607-010-0140-x

[25] European Commission. "Official Journal of the European Union On Data protection guidelines for the Early Warning and Response System", 9 February 2012 L 36/31.

[26] Pearson S. and Charlesworth A., "Accountability as a Way Forward for Privacy Protection in the Cloud", HP Laboratories HPL-2009-178, Proc. 1st CloudCom, Beijing, Springer LNCS Press, December 2009.

[27] European Commission, "Safequarding Privacy in a Connected World: A European Data Protection Framework for the 21th Century" COM (2012), 25 January 2012, article 9 final Brussels.

[28] Jaeger P.T., Lin J. and Grimes J.M., "Cloud Computing and Information Policy: Computing in a Policy Cloud?", Forthcoming in the Journal of Information Technology and Politics (ITI 2008), vol. 5, no. 3, pp. 269-283.

 ISBN: 978-1-61208-362-9

A Survey on Tor and I2P

Bernd Conrad and Fatemeh Shirazi

Department of Computer Science, TU Darmstadt
Darmstadt, Germany
Email: {bconrad,fshirazi}@cdc.informatik.tu-darmstadt.de

***Abstract*—This paper gives a short introduction and a comparison on two low-latency anonymous communication networks. The main part features a review of the low latency anonymous communication networks, namely, The Onion Routing (Tor) and the Invisible Internet Project (I2P). An introduction to their overall structure is given, followed by a detailed description of the core parts of both networks. Furthermore, a comparison of both will feature important aspects like node selection, performance and scalability. The detailed description and comparison of the two systems show that determining which system to use highly depends on the field of application, since each system has its strength and weaknesses in specific areas.**

Keywords–Tor; I2P; low latency anonymous communication networks.

I. INTRODUCTION

When communicating over the Internet, IP addresses are used to provide a unique identifier to address each party. Even if a message is encrypted to protect the data content, source and destination address are contained in clear in the corresponding IP datagram headers, otherwise messages could not be routed to their destination. Thus, communication over the Internet is not anonymous. An adversary monitoring the network traffic could easily identify two parties communicating with each other. Anonymous Communication Networks (ACNs) are an essential building block for protecting privacy online, as they enable users to communicate anonymously over the Internet. Using the ACN, users can conceal the destination of their communications towards local adversaries, e.g., their ISP, as well as protect their identity towards the destination itself, e.g., a website. Typically, an ACN is an overlay network composed of a set of routers (also-called relays, or *nodes*), in which packets are relayed using multiple routers to achieve anonymity. In general, anonymous communication networks can be divided into two main categories; *high latency anonymous communication networks* in which it takes a relatively longer time for the message to travel through the network and reach it's destination, usually ranging from a few hours to several days [1]. This delay is tolerable when using those systems for non-interactive applications like email, however today most applications on the Internet are real-time, interactive applications that require a low latency, e.g., web browsing. Systems designed to provide anonymity and low latencies when using real-time, interactive applications are called *low-latency anonymous communication networks*. In this paper, we review and compare Tor [2][3] and *Invisible Internet Project* (I2P) [4], which are currently among the most commonly used low latency anonymous communication networks. Other examples of practical anonymous communication networks are Freenet [5][6], JAP [7][8], and GNUnet [9][8]. Regardless of several similarities, Tor and I2P have noticeable differences, which makes them preferable for specific usages. In order to be able to decide on which one of them to use, one has to fully understand these differences. In this paper, we review some of the main differences between Tor and I2P.

The remaining paper is organized as follows: A short introduction to the anonymous communication network Tor is given in Section 2. Section 3 describes I2P and its core elements. Finally, a comparison of both systems will be presented in Section 4. Section 5 concludes.

II. ONION ROUTING AND TOR

One approach to achieve low latencies and at the same time protect against a strong adversary is the arguably most prevalent onion routing design [1], a distributed overlay network designed to anonymize TCP-based applications [3]. According to Danzis and Diaz [10] "the objective of onion routing is to make traffic analysis harder for an adversary, as well aims first at protecting the unlinkability of two participants who know each other from third parties, and secondly, at protecting the identities of the two communicating parties from each other".

A set of servers called Onion Routers (OR) are used to relay messages. Each OR maintains a private and public key pair, while the public part should be known to all clients wishing to participate in the network. Clients choose an ordered sequence of ORs they want to use to relay their data and establish a so-called *circuit*, a bidirectional tunnel. This method is called onion encryption and will be described more precisely later on. Each layer contains a symmetric key, a label and addressing information about the next OR. Messages sent through circuits are also onion encrypted, this time using the symmetric key of each OR [10]. Each OR is only able to remove the corresponding layer of encryption and forwards the message to the next OR in the circuit. The last OR in the circuit is able to forward the message to its destination. The potential response of the receiver is sent to the last OR in the circuit and is relayed back to sender through the exact same circuit. This time, each OR adds a layer of encryption to the message. Hence, another onion encrypted message is constructed that only the sender is able to decrypt and therefore recover the response. An important fact in regards to anonymity and security is that only the first OR in a circuit knows the IP address of the client, and only the last OR of a circuit knows the receiver of a message. All intermediate ORs only know its predecessor and its successor, without even knowing which other ORs are participating in the circuit [3]. A circuit may be used to relay multiple messages from a single application [1], but each TCP stream needs its own circuit [3]. The ORs implement a very close to first-in first-out mixing strategy to provide low

 ISBN: 978-1-61208-362-9

latency. This makes onion routing susceptible to a number of attacks. Due to missing cover traffic, an adversary may use traffic analysis and timing attacks to monitor a traffic pattern, follow the message stream and identify communicating parties [10][11]. Nonetheless, onion routing is a promising design to provide a low latency anonymous communication network and many currently used systems are built upon this design.

A. Tor

Tor is a distributed-trust, circuit-based low latency anonymous communication network. It builds upon the onion routing design, but makes many modifications and improvements in regards to security, efficiency, and deployability [1]. The Tor network is an overlay network that uses a set of volunteer servers, called *Onion Routers* (OR), to build *circuits* and relay messages [11]. Each user runs a software called *Onion Proxy* (OP) that manages all Tor related processes, e.g., establishing circuits or handling connections from user applications [3]. To build a circuit, the OP select an ordered set of usually three ORs out of the set of all known ORs. The first OR in the set is called entry guard, the last is called exit router and all others are called intermediate routers [12][13]. The process of selecting ORs for a circuit is called *node selection* and will be described more precisely later. To obtain a list of all known ORs, a set of *directory authority servers* are used. After selecting a set of ORs, the OP contacts the entry guard and builds a circuit with it. This newly created circuit is used to contact the next OR to extend the circuit. This procedure is iteratively repeated until all ORs of the set are part of the circuit. The established circuit can now be used to anonymously relay messages. Messages are *onion encrypted* and only the exit router is able to access and forward a message to its destination.

Onion Router: Onion Routers (OR) are the core part of the network since they are necessary to build circuits. All ORs are connected with each other using Transport Layer Security (TLS) connections. This prevents an attacker from modifying data or impersonating an OR [3]. Each OR also maintains two keys: a long-term identity key, used to sign TLS certificates, *router descriptors*, and directories; and a short-term onion key, used to decrypt user requests to build a circuit and negotiate short-lived symmetric keys [3][11]. Router descriptors uniquely identify each OR and contain all relevant data to contact and list an OR: public keys, IP address, bandwidth, *exit policies*, and more [3]. Exit policies describe which hosts and ports the OR is willing to connect to, this is particularly important for the later described node selection process.

Directory Server: To be able to retrieve a list of all available ORs, authoritative directory servers distributing signed directories are used [12]. These servers need to be well-known, which means that the IP addresses of this servers are commonly known and/or published on specific websites, and able to track changes in network topology. The directory contains the router descriptor of each listed OR and a network status document. The network status document contains measured bandwidths of ORs. Only ORs that are verified via their identity key are listed in the directory, otherwise they are ignored. There are multiple directory servers to protect against active attacks against directory servers [3], e.g., potentially by Denial-of-Service (DoS) attacks, which prevents having a single point of failure. All directory servers also merge their known topology of the network with each other and release a common signed directory of the whole network. Directories are automatically fetched by the OP. The client software also contains a default list of directory servers [3].

Node selection: To guarantee a good performance and to prevent choosing a corrupted OR as entry guard, the Tor client uses a path selection algorithm to select the ORs used to build circuits [14]. All known ORs are categorized into three tiers:

- *Entry guard router*: Stable, fast and well-known ORs.
- *Intermediate router*: All known ORs.
- *Exit router*: ORs with matching exit policies.

The network status document and all router descriptors maintained by the directory servers are fetched by the OP. Both contain router bandwidth information. The router descriptor contains a self-advertised bandwidth and the network status document contains a value measured by the directory servers. As long as the measured value is available, it will be preferred due to the fact that self-advertised information are considered not trustworthy. The bandwidth information is used to select the intermediate and exit routers in a weighted probabilistic manner [14]. This means a router with a higher bandwidth is more likely to be chosen.

The OP maintains a list of three potential entry guards, chosen from a list of all ORs with a long uptime and known to be fast and stable [14]. The entry guard is then randomly chosen from this list of three entry guards, and used for all circuits. After normally 30 days, the list of three entry guards is rebuilt and a new entry guard is chosen. ORs serving as exit routers can also be considered as entry guards and intermediate routers, but "only if the available total bandwidth of exit nodes is at least one third of the overall available bandwidth of all routers", also, to provide load balancing, "their probability of being chosen is lowered in a weighted way" [14].

Cell: Tor uses a special format, called cells, for all messages that are sent through the network. Cells have a fixed size of 512 bytes and consist of a header and a payload. Tor uses two kinds of cells, *control cells* and *relay cells*. Control cells are used to set up, maintain and destroy circuits. Relay cells are used to relay messages along the circuit. Relay cells also contain an additional header in front of the payload used to distinguish between different streams and to perform end-to-end integrity checking [3]. The additional header also allows the network to detect congestion or flooding, and therefore reduce outgoing traffic until the congestion subsides [3].

Circuit: A circuit is a bidirectional virtual connection set up between the OP and an ordered set of ORs. In contrast to onion routing, a single circuit can be used by multiple TCP streams at the same time. To prevent an adversary from linking streams together, the default circuit lifetime is 10 minutes. After this time, a circuit is destroyed and a newly built circuit is used. Building new circuits is done beforehand in the background, therefore no additional latency is generated [3].

Onion Encryption: After establishing a circuit, the OP can start sending data messages within relay cells. Similar to the onion routing design the header and payload of a cell is iteratively encrypted using the symmetric key of each OR participating in the circuit [3]. Starting with the key of the exit router, traversing the circuit using the key of each intermediate

 ISBN: 978-1-61208-362-9

OR until the entry guard is reached. The following example shows the encryption process for three ORs, whereas k_1 is the key of the entry guard, k_2 the key of the intermediate router and k_3 the key of the exit router. $E_{k_1}(E_{k_2}(E_{k_3}(cell)))$ This procedure is called *onion encryption*. As the cell moves along the circuit, one layer per OR is removed. Only the exit router is able to extract the destination address and the payload, which may be the actual plaintext or an end-to-end-encrypted message, and forwards the payload to its destination. The reply can only be sent back along the same circuit. Each OR in the circuit adds his layer of encryption, using his negotiated symmetric key, before relaying the cell to its predecessor. Only the OP is able to fully decrypt the onion encrypted reply, since he is the only one that knows all negotiated symmetric keys.

III. I2P

I2P is a message-oriented, peer-to-peer-based low latency anonymous communication network. The network was mainly designed to enable fully anonymous communication between two parties inside the network [15][16]. I2P was first proposed in 2003, having its roots in the Invisible Internet Project (IIP) [17][18]. A wide range of applications inside the I2P network are available, e.g., anonymous web-hosting, web browsing, file-sharing, email and many more. Using external services, meaning services that are not hosted within the I2P network, requires the use of an out-proxy [16]. At the time of writing, the I2P network consists of 23738 routers with an average count of 25687 routers [19].

I2P is an overlay network allowing users to anonymously interact within the network. Technically, I2P is a multi-application Java framework designed to provide anonymous P2P networking [20]. Each user is running a so-called I2P router, the core part of the I2P software. All messages are relayed through *tunnels* built by each I2P router using other I2P peers. Tunnels can only be used in one direction; therefore, tunnels for outgoing and incoming traffic need to be built, so-called *inbound* and *outbound* tunnels. The selection of peers is done via a *tier-based peer selection* algorithm running on each I2P router. After establishing inbound and outbound tunnels clients may publish their contact information in a global database, called *netDB*. The netDB contains contact information for each I2P peer and each publicly running service inside the I2P network. Messages sent through the I2P network are end-to-end encrypted using *garlic encryption*. Garlic encryption is very similar to onion encryption, with the difference that multiple data messages may be contained in a single *garlic message*. Therefore, a single garlic message may contain multiple messages for different recipients.

I2P Router: The I2P network is formed by peers (also-called clients, nodes or router) running the I2P software, allowing applications to communicate through the I2P network [16]. The core part of this software is the I2P router. The I2P router is responsible for maintaining peer statistics, which are required for the peer selection described later, performing cryptographic operations, building tunnels, providing services and relaying messages. Applications heavily rely on the tunnels built by the I2P router to remain anonymous [20].

NetDB, RouterInfo and LeaseSet: Super-peers, called *floodfill peers*, are used to build and manage a network database, called *netDB*. The netDB is based on a distributed hash table and contains all known information about the I2P network, therefore all I2P peers and services. Each floodfill peer is only responsible for information of a specific part of the network. The Kademlia XOR distance metric [21] is used to determine which part of the network a floodfill peer is responsible for, based on the peers ID [20]. Peers with sufficient bandwidth may get promoted to floodfill peers if the amount of available floodfill peers drops below a certain threshold [22]. The netDB stores two types of data, a *routerInfo* structure that describes an I2P peer and a *leaseSet* for each known service [23]. All I2P peers are identified by a data structure called *routerInfo*, containing all important information about the peer (IP address, port, peer ID, I2P stable version number, network version, transport capabilities and some statistical data [23]), his public key and a 256 bit hash-identifier. To retrieve an initial list of available I2P peers, a list of routerInfos can be downloaded from a non-anonymous, well-known web server. Retrieving the initial list of routerInfos is called *reseeding* [20][23].

A *leaseSet* is used to store information about how to contact an internal I2P service, called *destination*. The leaseSet specifies a set of entry points, called leases. A lease identifies a peer that serves as an inbound gateway to an inbound tunnel of the corresponding service [20]. Both, routerInfos and leaseSets, can easily be stored and retrieved by contacting the nearest floodfill peer. In case of storing, the floodfill peer will distribute the received routerInfo or leaseSet to the seven nearest floodfill peers. In case of retrieving, the two closest floodfill peers are contacted. If the requested information is not available, the floodfill peer replies with a list of other near floodfill peers. The peer keeps contacting floodfill peers until the needed information is retrieved or all floodfill peers have been contacted [20].

Destination: All destinations in the I2P network are identified by a 516 byte crypto key that consists of a 256-byte public key, a 128-byte signing key and a (currently unused) null certificate. A destination in I2P refers to an internal service provided by an I2P router. To map destination names to their crypto key, three local host files are used, similar to traditional DNS. To merge external and local host files, I2P provides an address book application [18][22]. This way of addressing each individual destination further increases the anonymity since it also decouples the service from the I2P router its provided by [16].

Tunnel: All messages in the I2P network are transmitted through so-called *tunnels*. A tunnel is a unidirectional encrypted virtual connection using typically 2 to 3 I2P peers [23][18]. Unlike Tor the I2P router seeking to establish a tunnel is also part of the tunnel. At startup each I2P router builds up some tunnels for incoming traffic, called inbound tunnel, and outgoing traffic, called outbound tunnel. The first I2P peer of a tunnel is called tunnel gateway, the last I2P peer of tunnel is called tunnel endpoint. For outbound tunnels, the I2P router that established the tunnel is always the gateway. For inbound tunnels, the I2P router that established the tunnel is always the endpoint. The default amount and length of tunnels can be specified by the user in the I2P settings. The length of a tunnel is a trade-off between performance and anonymity [20]. Longer tunnels increase the anonymity, while they decrease the performance and the other way round. An application is not bound to a specific tunnel and may use different tunnels to relay messages. There are two kinds of tunnels, *exploratory*

 ISBN: 978-1-61208-362-9

and *client* tunnels [17]. Exploratory tunnels are low bandwidth tunnels and not used for privacy-sensitive operations. A router uses this tunnel to contact floodfill peers and retrieve the netDB. Exploratory tunnels are also used to build, manage and destroy other tunnels [23]. Client tunnels are used to relay application messages and retrieve leaseSets; therefore, are high bandwidth tunnels. Tunnels have a maximum lifetime of 10 minutes. After this period of time the tunnel is destroyed and a new one is used. Constantly rebuilding tunnels seeks to prevent traffic analysis attacks [16].

Tunnel Establishment: Building a new tunnel is done by first selecting an ordered set of I2P peers. This selection of peers is done with *tier-based peer selection* and *peer profiling* to categorize peers into tiers. An exploratory tunnel is used to send a single, multiple times encrypted *tunnel construction request* to the first I2P router. Every layer contains necessary information for each single I2P peer, e.g., symmetric key and successor address. Like in the original onion routing design the message is forwarded until it reaches the last I2P peer of the tunnel. The response is then routed back to the originator while each I2P peer adds a layer of encryption [24]. The receiving I2P peers are free to decide if they want to decline the request or accept to participate in the tunnel. An already established tunnel can still fail at any time if, e.g., the I2P peer is not able to handle the traffic or leaves the network (goes offline) [20].

Tier-based Peer Selection and Peer Profiling: Tier-based peer selection is the process of selecting peers used to build a tunnel based on tiers they are assigned to. *Peer profiling* is used to categorize peers into those tiers. Peers sharing a tier share certain performance characteristics [23]. Peer profiling is done by the I2P router, he keeps track of various performance statistics of other peers and maintains a database containing this statistics, called *profiles*. However, no active bandwidth probing or other actions that may generate non-data traffic are used. Every 30 seconds all profiles are sorted into three tiers based on various metric like speed and capacity [23][20]:

- *Not-failing*: All known peers. Typically 300-500 peers.
- *Well-integrated*: Peers that claim to know many other peers.
- *High-capacity*: Peers that are known to most likely accept tunnel build request. Typically 10-30 peers.
- *Fast*: Peers from the high-capacity tier with a high bandwidth. Typically 8-15 peers.

Note that all fast tier peers are always also high-capacity tier peers [23]. When constructing a client tunnel, peers from the fast tier are used. If no sufficient amount of fast tier peers is available, high-capacity tier peers are selected. High-capacity tier peers are used when constructing an exploratory tunnel. Both, the well-integrated and not-failing tier peers are fallback options, if no high-capacity and fast tier peers are available. However this is unlikely to happen in practice [20]. The actual selection of peers for exploratory tunnels is done using a weighted random function [23]. Also peers sharing the same /16 subnet will not be used together within the same tunnel [23].

Garlic Routing, Garlic Message and Garlic Encryption: When at least one outbound and one inbound tunnel is constructed, the I2P router is able to send and receive messages through the I2P network. To communicate with an I2P service, the router first needs to retrieve the destination of this service from a floodfill peer [17]. The destination specifies a set of inbound tunnel gateways of the corresponding service. I2P uses so-called *Garlic routing*, a variation of the onion routing design described in Section II. Garlic routing uses garlic messages that can contain multiple so-called cloves. Cloves are data messages with additional routing instructions like delays. This means a garlic message may contain multiple application messages. The actual data messages are end-to-end encrypted with the receiver's public key. The garlic message itself is encrypted multiple times using the symmetric keys negotiated with the tunnel peers [22][20]. When traversing the tunnel, each I2P peer removes one layer of encryption until the garlic message reaches the outbound tunnel endpoint. The outbound endpoint forwards each message to its destination's inbound tunnel gateway. The inbound gateway will forward the garlic message to the actual recipient while each peer participating in the tunnel adds a layer of encryption (using the negotiated symmetric keys). Only the recipient is able to remove all encryption layers of the garlic message as well as the end-to-end encryption of the data-messages [20][17]. As mentioned before, if a service outside the I2P network is addressed, an outproxy has to be used [18], although according to Schimmer et al. [23] "only one HTTP outproxy is publicly advertised and accessible". When using an outproxy, end-to-end encryption is, similar to Tor, not guaranteed, since it depends on the transport layer protocol that is used.

IV. TOR VS I2P

There are a few obvious difference between both networks. While Tor is relying on servers provided by volunteers to build circuits, I2P uses peers with sufficient performance characteristics participating in the network to build tunnels. Also, Tor is optimized and designed for exit traffic with a large number of exit routers, whereas I2P is designed to provide services inside the network and only features a small set of outproxies [25]. Nonetheless, both seek to provide strong anonymity with low latency when using real-time, interactive services. A comparison of a few important aspects of anonymous communication networks is presented as follows.

SOCKS vs I2P API: While this seems like a rather technical aspect, it greatly changes the effort and ability to build applications that use either the I2P or Tor network to anonymously communicate over the Internet. Tor uses the *Socket Secure* (SOCKS) interface and therefore SOCKS-aware applications may be easily pointed at the Tor software, which then handles everything else. Tor, in this case, acts as a proxy server. This means, applications able to use SOCKS can be used without any changes [3]. I2P, on the other hand, is a middleware providing APIs that applications can use to communicate through the network, meaning applications either need to be costly adjusted, if at all possible, or implemented from scratch. The use of SOCKS by Tor has two downsides:

1) The SOCKS interface is only able to transmit messages over TCP while I2P has the choice between UDP and TCP [25]. This may enable I2P to deliver better performance when using certain applications.
2) Messages sent by applications may still contain information that could identify the sender. To prevent this, application-level proxies with filtering features, e.g., Privoxy, need to be used [3].

 ISBN: 978-1-61208-362-9

Available Applications: Both, I2P and Tor feature a wide range of applications, whereas most I2P applications are exclusively made to access services inside the I2P network, with some exceptions, e.g., Susimail/2IpMail is able to send and receive mails from the public Internet [18]. Tor on the other hand, due to the fact it is using the SOCKS interface as mentioned before, is able to be used with any application able to be configured using a SOCKS proxy, e.g., nearly every commonly used web browser.

Message Security and Anonymity: Both networks feature various layers of encryption, starting with transport layer encryption provided by the TLS connection maintained by the ORs or respectively I2P peers. I2P also features an additional tunnel encryption. Messages sent through the networks are either onion or Garlic encrypted. This means the connection from the user to the tunnel or circuit is always encrypted. As long as interacting inside the network, messages in I2P are also end-to-end encrypted. In the case of Tor, end-to-end encryption can not be guaranteed since it depends on the transport layer protocol that is used. Therefore, insecure protocols should not be used, as a corrupted exit node may record messages sent in plaintext and recover usernames and/or passwords [12]. In Tor only the first OR of a circuit knows the IP address of the actual user, all subsequent ORs only know its predecessor and successor. Also only the last OR in the circuit knows the actual receiver. Nonetheless, this is a potential risk since corrupted ORs may be able to link communicating parties together. Therefore, the user's anonymity highly depends on Tors node selection algorithm selecting trustworthy entry guards. In the case of I2P even the first peer does not know if it is forwarding the message for another peer that is part of the tunnel or the actual sender. Therefore, entry guards like in Tor are not necessary.

Performance: In 2011, Ehlert analyzed and compared the latency and bandwidth when accessing the public Internet with either I2P or Tor [17]. The latency when issuing simple HTTP-GET-Requests and the average latency when accessing whole web pages were recorded and evaluated, as well as the download speed when receiving files from a fixed location. While I2P was able to achieve better results when issuing simple HTTP-GET-Requests, Tor provides clearly better results in terms of accessing whole web pages and downloading files. In 50% of all cases Tor was able to retrieve a whole web page in less than 16.99 seconds, while 50% of the I2P request took up to 103.19 seconds. In case of download speed, Tor was able to deliver an average speed of 51.62 kB/s compared to the 12.91 kB/s of I2P. The author also seeks to explain why I2P is scoring better results than Tor when issuing simple HTTP-GET-Requests. He states, that the discrepancy may be explained with the good distribution of I2P peers in Europe and therefore good latencies when issuing simple HTTP-GET-Request. For further information see [17].

Scalability: Increasing the number of clients participating in the anonymous network directly influences both Tor and I2P. Although the anonymity set becomes larger and therefore stronger anonymity may be present, the network traffic increases and may cause problems like congestion. In case of Tor this means, the amount of routers used to build circuits may very likely need to be increased. This problem may get even worse due to the fact that only a small subset of all ORs is used as entry guards and exit routers. This may, depending on the amount of new clients joining the network, sooner or later lead to congestion problems and therefore increase the latency and decrease the available bandwidth. Congestion and high latencies will directly affect the user experience and network usability. Increasing the amount of ORs also serves another problem, the growing directory. On the one hand additional bandwidth is used to receive directories and on the other hand the effort to keep track of the whole network increases. As mentioned before, Tor also uses active bandwidth probing which additionally increases the traffic depending on the amount of new ORs joining the network. Also the assumption that every OR in the network is able maintain a direct connection to each other OR seems rather unlikely as the number of ORs increases [1][3]. In case of I2P, new peers joining the network may also be peers that can be used to build tunnels, assuming they provide enough capacity and bandwidth. Therefore, congestion is not likely to appear, however, if a sufficient number of clients seek to access services outside the I2P network, more outproxies may have to be provided. Apart from that, more clients joining the network provides a lot of benefits:

1) The amount of potential fast tier peers will most likely increase and therefore tunnels with less latency and more bandwidth may be the consequence.
2) The amount of cover traffic in the network will most likely increase and therefore provide a stronger anonymity.
3) With more clients using the network, it is very likely a greater range of services will be provided.

Centralization: In Tor, the network is not fully distributed as it is in I2P. The information about the relay nodes and the hidden services are provided by (currently 9) authorized directories which are placed in US and Europe. These authorized directories keep track of changes in the network and distribute this information, therefore if all of them collude the anonymity is endangered. However, in I2P such centralization doesn't exist. Each participating relay locally maintains a list of all known relays.

Routing and Node Selection: Both Tor and I2P run specific node selection algorithms to improve performance and protect against adversaries. While Tor distinguishes between entry, exit and intermediate nodes, I2P has none such. In case of I2P, each peer selected for a specific tunnel may either be the first, the last or an intermediate peer. To be able to select intermediate and exit routers, Tor's directory servers use active bandwidth probing to measure and record the bandwidth each OR is able to provide. This generates non-data message traffic. Also Tor has to rely on self-advertised bandwidth values if no probing data is available for this specific OR. This may lead to misclassification or may potentially be used by an adversary to classify his OR as an entry guard. With the exception of the entry guard, which is chosen from a small set of well-known ORs with long uptime, all other ORs in Tor are chosen with a probability proportional to its bandwidth [3]. This means only bandwidth and capacity are considered while other attributes like the actual location of the ORs are ignored. This may lead to high latencies when ORs are chosen that are, for example, located on different continents. In case of bandwidth, this way of selecting nodes may be optimal, but when taking into consideration, that the latency is an important point when browsing websites, this may not seem to be the

 ISBN: 978-1-61208-362-9

optimal way. The current load of the network is also not considered; therefore, existing resources may not be optimally used [3].

I2P clients on the other hand solely rely on previously monitored performance values and the current state of the network. No active bandwidth probing is used. The I2P node selection algorithm is also able to react very fast to failing peers and other changes in the network topology. This behavior of quickly reacting to failing nodes also holds a security problem. As described by Herrmann and Grothoff [20], a selective DoS attack targeting the current fast tier peers may give an adversary the possibility to inject his own corrupted peers into the fast tier. Due to the short lifetime of tunnels, some I2P users will most likely use one or more corrupted peers to build their tunnels. This of course may lower the grade of anonymity provided for this particular I2P users. The location of I2P peers is also not considered when categorizing them into tiers, therefore, similar to Tor, high latencies may be the result. Last but not least, newly joined I2P peers may have insufficient or outdated peer statistic and network informations to select optimal tunnel peers.

Avoiding Congestion: Tor uses circuit switching, whereas I2P uses packet switching, hence, Tor has often to cope with high congestion leading to high latency [26]. Whereas in I2P, the packet switching leads to some implicit load balancing and helps to avoid congestion and service interruptions. This is specifically important for large file transfers and therefore I2P is more suitable for such purposes.

Usage: I2P offers several applications and is rather designed for communication within the I2P network, in particular because it has few out-proxies. Whereas Tor is rather designed for routing traffic outside the network and has in comparison to I2P more exit nodes. In addition as mentioned in earlier in this section Tor's performance is better for visiting web pages than I2P, which makes Tor a better choice for surfing. However, for downloading I2P shows better results hence for applications such as file sharing I2P is more suitable.

Attacks: There are essentially two main classes of attacks that target the Tor network: *Traffic analysis attacks* [27], [28] and *DoS attacks*. Attacks on Tor have been commonly reviewed in the literature [29][1][30]. Grahn et al. give a review on general anonymous communications, which includes Tor and I2P [31]. Zantout et al. describe I2P and the known attacks on I2P where the main attacks are classified as DoS attacks, Partitioning attacks, and Intersection attacks [32]. Here, we review some new attacks for both Tor and I2P. For our review the main goal of the adversary is to identify peers (in the case of I2P) or de-anonymizing users (in the case of Tor).

Recently, two DoS attacks have been proposed on Tor: Johnson et al. propose the Sniper attack, which exploits the reliable data transport in Tor by consuming a large amount of memory [33], and Barbera et al. propose the CellFlood attack [34], which exploits the circuit construction process in Tor by flooding the router with circuit construction requests. Both attacks exploit technical vulnerabilities of Tor. Another known DoS attack on Tor, proposed in 2007 by Borisov et al., is the selective DoS attack [35], which rather describes the method for selecting nodes for the DoS attack and is not proposing technical measures for performing the attack. The main goal of DoS attacks against Tor is to either force users to choose malicious routers which in turn reduces the smaller user base and weakens anonymity [36]. More recently, website fingerprinting attacks on Tor have been proposed by Wang et al. which seek to deanonymize users by matching packet quantities and sizes of received packets [37].

Herrmann et al. proposed an attack on I2P uses a sort of selective DoS attack and exploits the node selection bias towards nodes with good performance in order to de-anonymize peers that are hosting Eepsites [20]. Crenshaw investigated de-anonymization attacks on I2P connections by analyzing the data that is leaked by applications that are run using I2P [38]. More recently, Egger et al. proposed some practical attacks on I2P, where the attacker tries to break the anonymity of users by using DoS and Sybil attacks [39] as part of her attack scenario [40].

V. Conclusion

In this paper, two state of the art Onion Routing based low latency anonymous communication systems were presented and compared. While Tor is the at the moment most popular and most used system, I2P is a fast growing competitor. Both systems are constantly being updated to improve performance and provide better anonymity while protecting against adversaries. Tor, due to the fact of it's greater awareness in the academic community, was already able to solve problems that I2P will sooner or later have to face. Tor also benefits from a large number of formal studies of its anonymity, resistance to attacks, and performance. As pointed out before, the key difference of both networks is the way they set up and use their virtual connections, in terms of node selection and client's node participation. Another important difference is that while Tor was designed for exit traffic, I2P seeks to provide services inside the network to provide stronger anonymity for service provider and users.

Overall, this comparison shows that it highly depends on the field of application to determine which system delivers better results in terms of performance and anonymity. When browsing the public web, Tor undoubtedly delivers better performance, while I2P is almost unusable. On the other hand, I2P provides a stronger anonymity and better performance compared to Tor when interacting with services or users inside the network. In the end, it is always a trade-off between performance and anonymity, no matter which system is used.

References

[1] M. Edman and B. Yener, "On anonymity in an electronic society: A survey of anonymous communication systems," ACM Computing Surveys (CSUR), vol. 42, no. 1, 2009, pp. 5:1–5:35.

[2] "The Tor project," https://www.torproject.org/, accessed: 11/07/2014.

[3] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," DTIC Document, Tech. Rep., 2004.

[4] "The Invisible Internet Project project," https://geti2p.net/en/, accessed: 11/07/2014.

[5] "The FreeNET project," https://freenetproject.org/, accessed: 11/07/2014.

[6] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," in International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability, 2001, pp. 46–66.

[7] "Project: AN.ON- anonymity.online," http://anon.inf.tu-dresden.de/index_en.html, accessed: 11/07/2014.

 ISBN: 978-1-61208-362-9

[8] O. Berthold, H. Federrath, and S. Köpsell, "Web MIXes: A system for anonymous and unobservable Internet access," in Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, H. Federrath, Ed. Springer-Verlag, LNCS 2009, July 2000, pp. 115–129.

[9] "The GNUnet project," https://gnunet.org/, accessed: 11/07/2014.

[10] G. Danezis and C. Diaz, "A survey of anonymous communication channels," Computer Communications, vol. 33, 2008.

[11] J. Ren and J. Wu, "Survey on anonymous communications in computer networks," Computer Communications, vol. 33, no. 4, 2010, pp. 420–431.

[12] D. McCoy, K. Bauer, D. Grunwald, T. Kohno, and D. Sicker, "Shining light in dark places: Understanding the Tor network," in Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008), N. Borisov and I. Goldberg, Eds. Leuven, Belgium: Springer, July 2008, pp. 63–76.

[13] R. Snader and N. Borisov, "A tune-up for Tor: Improving security and performance in the Tor network," in Proceedings of the Network and Distributed Security Symposium - NDSS '08. Internet Society, February 2008.

[14] A. Panchenko, F. Lanze, and T. Engel, "Improving performance and anonymity in the Tor network," in Proceedings of the 31st IEEE International Performance Computing and Communications Conference (IPCCC 2012), December 2012, pp. 1–10.

[15] J. Timpanaro, I. Chrisment, and O. Festor, "I2P's usage characterization," Traffic Monitoring and Analysis, 2012, pp. 48–51.

[16] J. Timpanaro, C. Isabelle, F. Olivier et al., "Monitoring the I2P network," 2011.

[17] M. Ehlert, "I2p usability vs. Tor usability a bandwidth and latency comparison," Seminar Report, Humboldt University of Berlin, November 2011.

[18] "I2p...the *other* anonymous network," http://sempersecurus.blogspot.de/2011/06/i2pthe-other-anonymous-network_18.html, accessed: 08/07/2014.

[19] "stats.i2p - the home for NetDB statistics," http://stats.i2p.to/, accessed: 08/01/2013.

[20] M. Herrmann and C. Grothoff, "Privacy-implications of performance-based peer selection by onion-routers: A real-world case study using i2p," in Privacy Enhancing Technologies. Springer, 2011, pp. 155–174.

[21] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in Peer-to-Peer Systems. Springer, 2002, pp. 53–65.

[22] T. E. Y. Iwan Hoogendoorn and J. Soeurt, "Further reducing the anonymity set of web servers hidden within the i2p network," 2011.

[23] zzz (Pseudonym) and L. Schimmer, "Peer profiling and selection in the i2p anonymous network," in Proceedings of PET-CON 2009.1, March 2009, pp. 59–70.

[24] "Tunnel implementation," http://www.i2p2.de/tunnel-alt.html, accessed: 08/07/2014.

[25] "I2P COMPARED TO TOR AND FREENET," http://www.i2p2.de/how_networkcomparisons.html, accessed: 08/07/2014.

[26] R. Dingledine and S. J. Murdoch, "Performance improvements on Tor or, why Tor is slow and what we're going to do about it," The Tor Project, Tech. Rep. 2009-11-001, November 2009.

[27] S. J. Murdoch and G. Danezis, "Low-cost traffic analysis of Tor," in Security and Privacy, 2005 IEEE Symposium on. IEEE, 2005, pp. 183–195.

[28] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker, "Low-resource routing attacks against Tor," in Proceedings of the 2007 ACM workshop on Privacy in electronic society. ACM, 2007, pp. 11–20.

[29] N. Danner, S. Defabbia-Kane, D. Krizanc, and M. Liberatore, "Effectiveness and detection of denial-of-service attacks in Tor," ACM Trans. Inf. Syst. Secur., vol. 15, no. 3, Nov. 2012, pp. 11:1–11:25.

[30] T. G. Abbott, K. J. Lai, M. R. Lieberman, and E. C. Price, "Browser-based attacks on Tor," in Proceedings of the 7th International Conference on Privacy Enhancing Technologies, ser. PET'07, 2007, pp. 184–199.

[31] K. J. Grahn, T. Forss, and G. Pulkkis, "Anonymous communication on the internet," in Proceedings of Informing Science & IT Education Conference (InSITE) 2014, December 2014, pp. 103–120.

[32] B. Zantout and R. Haraty, "I2p data communication system," in Proceedings of ICN 2011, The Tenth International Conference on Networks, January 2011, pp. 401–409.

[33] R. Jansen, F. Tschorsch, A. Johnson, and B. Scheuermann, "The sniper attack: Anonymously deanonymizing and disabling the Tor network," in To appear in Proceedings of the 21st Annual Network & Distributed System Security Symposium (NDSS '14). Internet Society, 2014.

[34] M. V. Barbera, V. P. Kemerlis, V. Pappas, and A. Keromytis, "CellFlood: Attacking Tor onion routers on the cheap," in Proceedings of ESORICS 2013, September 2013, pp. 664–681.

[35] N. Borisov, G. Danezis, P. Mittal, and P. Tabriz, "Denial of service or denial of security? How attacks on reliability can compromise anonymity," in Proceedings of CCS 2007, October 2007, pp. 92–102.

[36] R. Dingledine and N. Mathewson, "Anonymity loves company: Usability and the network effect," in Proceedings of the Fifth Workshop on the Economics of Information Security (WEIS 2006), R. Anderson, Ed., June 2006.

[37] T. Wang and I. Goldberg, "Improved website fingerprinting on Tor," in Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society. ACM, 2013, pp. 201–212.

[38] A. Crenshaw, "Darknets and hidden servers: Identifying the true ip/network identity of i2p service hosts," in Proceedings of Black Hat 2011, January 2011.

[39] J. Douceur, "The Sybil Attack," in Proceedings of the 1st International Peer To Peer Systems Workshop (IPTPS 2002), March 2002, pp. 251–260.

[40] C. Egger, J. Schlumberger, C. Kruegel, and G. Vigna, "Practical attacks against the I2P network," in Proceedings of the 16th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2013), October 2013, pp. 432–451.

 ISBN: 978-1-61208-362-9

Autonomous Alternative Complex with Remote Data Collection

Alexey Lagunov, Anton Belugin, Ksenya Semkiv
Department of Computer Science and Electronic Devices
Northern (Arctic) Federal University named after M.V. Lomonosov
Arkhangelsk, the Russian Federation
emails: a.lagunov@narfu.ru, d.belugin@narfu.ru, k.semkiv@narfu.ru

***Abstract* — The northern areas have a strategic importance for the Russian Federation. Enormous energy resources are concentrated in the circumpolar region. The successful development of the Arctic region requires high quality telecommunications systems. The power supply for these communication devices is an acute issue that needs to be solved in the remote areas of the Arctic. Satellite communication is available up to a latitude of 80° North. As a rule, the supply of satellite equipment needs diesel generators. In this paper, we propose the use of hybrid power energy sources, such as solar panels and wind turbines. Remote monitoring of such system makes it possible to understand the running processes, while generating electricity with the help of alternative sources of power. On the grounds of the analysis of available battery charge controllers from solar panels and wind turbines, the remote acquisition system was developed and tested in the Arctic. The system is based on a single-board computer RaspberryPI.**

Keywords-Alternative Energy; Remote Data Collection; Wind Turbine; Solar Panel.

I. INTRODUCTION

The northern territory plays a significant role in the economy of the Russian Federation. Currently, the Russian Federation has as priority the development of information and telecommunication systems. The Arctic has enormous reserves of important mineral resources, especially oil and gas, which are strategically necessary for the dynamic development of the Russian economy, the provision of the country safety in mineral resources and to protect geopolitical interests.

Within the mainland of the Arctic zone, there are a number of major oil and gas provinces (PNC) and deposits. Considering the initial recoverable resources in the Yamal-Nenets Autonomous District, the largest oil districts (which have totally over 100 million tons) are: the Timan-Pechersk one, with initial total recoverable hydrocarbon resources of 6 billion tons in fuel equivalent (the fourth place in Russia), Russian province, Novo Portovsoe, Sutorminskoye, North Komsomolskoe Tarasovskoe, Kharampurskoye oil deposits. More than 90% of the gas deposits in the district are unique and large - Urengoiskoye, Yamburgskoye, Bovanenkovskoye, Zapolyarnoe, Kharasaveyskoye, South Tambey with their reserves from 1 to 10.6 trillion m [1] (Figure 1).

The initial aggregate hydrocarbon resources of the Arctic continental shelf make about 100 billion tons of fuel equivalent, 80% of which is gas. The main hydrocarbon resources (approximately 70%) are concentrated in the Barents, Pechora and Kara Seas. The unique and large Shtokman Prirazlomnoe, Leningrad, Rusakovskaja hydrocarbon deposits are located here. The commercial development of fuel and energy of the Arctic shelf is going to stabilize the dynamics of oil and gas production since it compensates possible recession of production activity in the continental deposits in the years 2015 - 2030. But this is possible only if material, scientific and technical foundation for the offshore oil and gas deposits development is provided.

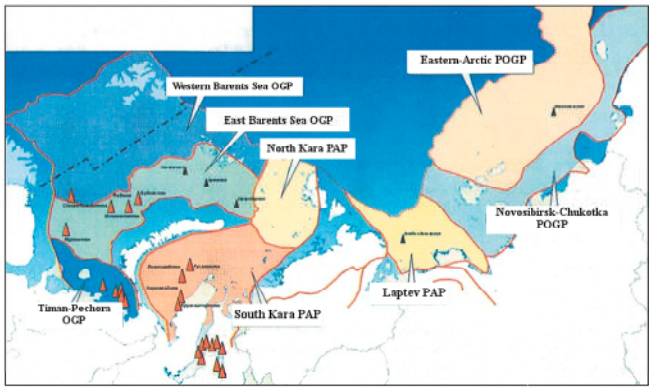

Figure 1. Oil and gas provinces in the Arctic region of the Russian Federation.

The marine transportation system, especially the Northern Sea Route, takes a special place in the transport sector of the Arctic region. The shortest routes between markets of northwestern Europe and Asian countries pass through the Arctic zone. The increased transport activity in the global economy for the development of transcontinental transportation, the increase of oil and gas production on the continental shelf of the Arctic, the improvement of the internal and external transport needs, have all led to the growth of the Northern Sea Route role and importance. When using the Northern Sea Route instead of operating the routes through the Suez and Panama Canals, the route from Rotterdam to the port of Yokohama is reduced by 34%, to the port of Shanghai by 23%, and to the port of Vancouver by 22% [2].

The vast distances and high-latitude location of the Arctic make it difficult to build communication systems in this area. The organization of a telecommunication infrastructure in the Arctic region meets both technical and organizational difficulties. Having analyzed foreign

 ISBN: 978-1-61208-362-9

publications about projects focused on providing satellite communications in the Arctic region, it can be concluded that many countries are looking for the best way to implement systems for satellite communications management in the Arctic (Figure 2).

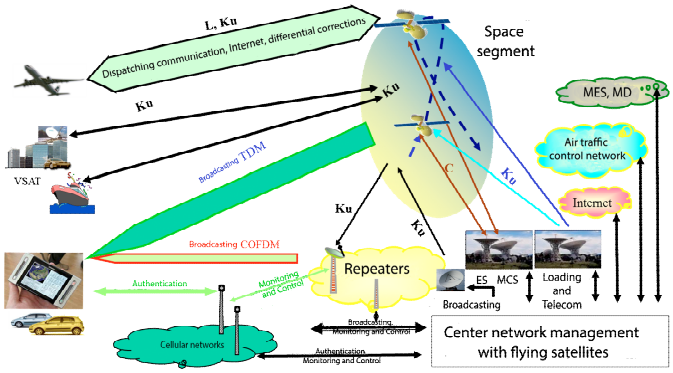

Figure 2. Architecture of the Arctic Communications System.

The number of scientific and technical papers and articles on this subject in international journals and conferences has increased by several times over recent years. It shows this topic became urgent. However, a single technical solution does not exist. The majority of projects relate to the creation of the space segment based on satellite in elliptical orbits, but, in some publications, the authors rightly point out strong influence of the Earth's radiation belts, which can cause the reduction of the satellites viability [3]. The "Tundra" orbit type is offered as an alternative [3]. We should note that, today, the circumpolar region has only one Iridium system working adequately, but, for personal communications, this system has many limitations in bandwidth [4].

The other systems, such as “Globalstar”, and especially Inmarsat, are available only up to a latitude of about 70° North [4]. Any other geostationary systems have the same limitation, although there are some examples of their usage up to 80th latitude (Table. I presents estimates of fading in the radio link). The low-orbit systems, like “Orbcomm” [5] or the Russian system "Gonets" [6], which is planned to be deployed, actually are not communication systems, but data transmission systems working like "email".

TABLE I. FADING OF THE RADIO LINK FOR THE CLIMATE ZONE IN THE WESTERN PART OF THE BARENTS SEA

Frequency, GHz	Elevation angle of 2-3°, dB		Elevation angle of 5°, dB	
	Precipitation	Without precipitation	Precipitation	Without precipitation
30	26	22	15	12
14	8	6,4	4,7	3,4
6	2	2	1	1

The Russian market of satellite communications is based on the resources of the orbital groups named FSUE (Federal State Unitary Enterprise) "Satellite Communications" and Public corporation "Gazprom Space Systems". The first group is represented by 11 communication satellites; the second one is represented by the two satellites "Yamal". The stable connection area with geostationary satellites (elevation angle of 5 degrees) is shown in Figure 3. The main types of traffic are: spreading television and broadcast programs according to broadcast zones; telephone lines and data transmission; data exchange in enterprise and dedicated networks; direct television and audio broadcasting; mobile and fixed government bond (totally about 300 transponders).

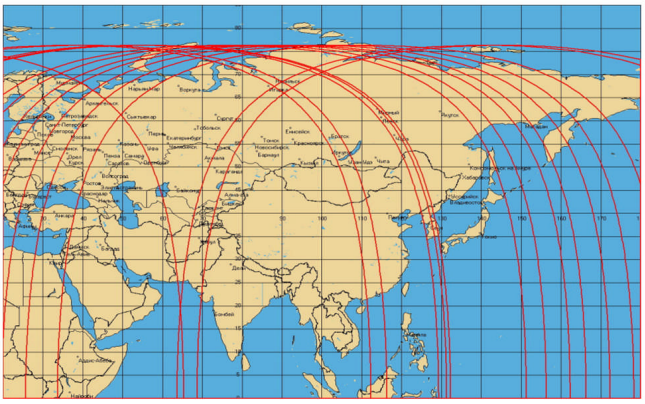

Figure 3. Stable connection area with the geostationary spacecraft (elevation angle of 5 degrees.)

Satellite communication devices are low-power devices. According to research, the maximum possible power consumption is limited by 900 VA. Currently, for the power supply of satellite communication devices in the Arctic, diesel generators are commonly used; they consume quite a lot of fuel and pollute the environment.

It was decided that the installation of the current power supply of satellite communications be established at Cape Desire of Novaya Zemlya archipelago (76°51' North latitude, 68°33' East longitude). There was no possibility to leave a researcher at the installation place, so a remote acquisition system was needed.

Data acquisition systems are essential to estimate the potential of renewable energy sources. For instance, a large quantity of data from different years is necessary to estimate scenarios using renewable energy sources. These aspects have an importance, mainly for developing countries, where decentralized power plants based on renewable sources are in some cases the best option for supplying electricity to rural areas. Nevertheless, the cost of commercial data acquisition systems is still a barrier for a greater dissemination of such systems in developing countries [3]. A local web server (on board) is constrained by lower memory limitations for storage of large amounts of data. In addition, satellite channels in the Arctic have very low bandwidth, so usage of this method is not possible.

Hence, remote operators of renewable energy plants connected to that local web server can view only limited data [7]. The applied data is usually organized in text files, which is inefficient. Hence, the development of an automated

 ISBN: 978-1-61208-362-9

database is indispensable [7]. The monitoring system consists of a microcontroller-based unit to acquire interest signals, while the collected data is transmitted to a database server by a Ground Station Module (GSM) modem. The GSM standard extends the effectiveness of the system independently wherever the plants are placed, even far from the electrical distribution network and from the traditional and wired telecommunication systems. Due to the low cost and diffusion of the GSM devices, the transmission system is fairly cheap and it is expected to become even cheaper [8]. The collected data is further processed, stored on the disk and displayed on the web page using the PHP language (Hypertext Preprocessor). This method has the advantage of a rapid data acquisition system development and provides an easy-to-use graphical environment that permits system operators to process the collected data easily. The maintenance operator presence in a decentralized renewable energy plant should be as low as possible, considering the moderate value of the energy produced [8]. This way, the purpose of the present paper is to allow such plants to be remotely monitored and controlled by a remote operator.

II. ARCTIC EXPEDITION

In recent years, the study of the northern and Arctic areas has been growing very fast. The number of weather stations, scientific bases, oil and other extractive companies building settlements around the deposits is increasing. Any living infrastructure requires energy, and the increased consumption of petroleum products is not economically feasible. The wind and solar energy application technologies have already been considered to be used successfully for a long time in the southern areas. The northern areas impose serious constrains on the use of such systems, especially in the Arctic region.

Our team began to study the possibility of introducing alternative energy systems on scientific, weather, oil and other stations in the Arctic region. During the laboratory tests for the project, various "green" energy systems were installed in Novaya Zemlya archipelago and in the Zhizhgin Island, which is situated in the White Sea. As these systems have low efficiency and are not stable enough to work, the construction of full-power stations for livelihoods is a rather expensive and unprofitable task. Therefore, in our studies, all the systems were created to provide people with a permanent connection to the mainland. Due to the temporary need of communication with the mainland and because the amount of energy consumed by the communication systems does not exceed 200 watts at peak load, the use of such systems is considered appropriate.

The project targets is to:

- determine the most appropriate hardware configuration for the conditions of the Far North;
- provide a stable working condition of the communication system and configure the system so that it has maximum autonomy.

At this stage, this is considered a research project. Therefore, one of the priorities is to obtain the maximum possible amount of information about how the constructed systems work. Thus, it is necessary to design a hardware and software system for the collection, processing and presentation of data on the alternative energy system functioning in the Far North.

III. DESCRIPTION OF THE GENERAL SCHEME

Our team has been conducting studies on this subject for two years. The initial step was to install the first test alternative energy complex in Novaya Zemlya archipelago. During the first laboratory tests, we decided to use the interfaces provided by the equipment suppliers without any modification. The system was installed near the research station that had a satellite communication channel. The network had a direct connection to the controller of the solar panels and the data was obtained through the server inquiry. Not only solar panels, but also a wind turbine of horizontal type was delivered to the archipelago. Unfortunately, we were unable to read the data from the wind turbine controller. According to the results of the first laboratory tests, the following conclusions can be drawn:

- it was determined that the use of horizontal type wind turbines is impractical in the Far North due to the gusty wind, quickly changing the direction of movement;
- the data collection should be performed by the system and not by the server, since, in the first case, the service traffic required for communications protocol demands many expenses for the satellite channel;
- the data obtained using standard controllers was found to be insufficient; it requires additional sensors installation.

Based on the above conclusions, we assembled the second installation. In the complex, we decided to use wind turbines of vertical type. However, this controller did not have the ability to be connected directly to the network. It was necessary to use the digital output RS232 to receive telemetry data from the controller. Based on the previous experience, we decided that the data will be collected not by the server only, but by the complex itself. This method helped to solve several problems at once. Firstly, we accessed the data from several devices. Secondly, there was a possibility to obtain and collect data even in the absence of the Internet connection with the system. Thirdly, there were additional opportunities to collect and send data, as well as to control the whole system.

SBC Raspberry Pi [9] was chosen as a device for data collection. The main advantages for us were, firstly, its low energy consumption, which helped to save the system efficiency in the conditions of very strong battery discharge, and, secondly, it has a complete operating system at a sufficiently low cost and small size of the device. All these factors helped to reduce the development time and improve the whole system reliability.

After the second installation was tested, the new technical requirements to both the system and the software were determined. In the Arctic region, the quality of satellite communication is extremely unstable because of various reasons. Therefore, we faced the problem of data safety during the process of transmission. While operating, the data was revealed to be obtained after a long delay and was

 ISBN: 978-1-61208-362-9

incomplete. For further studies, it was necessary to increase the reliability of the data transmission through the unstable Internet connection.

According to the results of the second laboratory test, the remote acquisition system was completed and implemented into the third complex of "green" energy. We finalized the technological infrastructure of the complex. As the equipment from other manufacturers was used, the number of devices requiring data collection was increased. The final system consists of the two wind turbines [11] and the four solar panels [12] that must be operated by the two hybrid charge controllers [13]. Each controller is connected to one wind generator and two solar panels. The system also includes four batteries [14] and an inverter [15] converting voltage from 12 to 220 V and managing energy flows in the system. Thus, the operation of the complex can be divided into three stages:

- Conversion of solar and wind energy into electrical energy;
- Buffering the electrical energy;
- Consumption of the accumulated energy.

The individual devices are responsible for each stage of the system. Therefore, all devices that compose the complex can be divided into three groups.

The first group of the devices, which are responsible for the conversion of “green” energy into electrical energy, includes charge controllers. The devices of this group provide data on direct work of the energy sources and display battery charge during their charging. In terms of the data obtained from these devices, we can, on one hand, receive information about the performance of energy conversion devices as a complex, and, on the other hand, watch each of them separately. The second group is represented by the storage batteries of various power that are able to accumulate electric charge. The third group consists of the inverters and provides information about energy consumption and the state of the battery during the process of their discharge. Thus, we have so far covered the stages of energy generation and energy consumption. The stage of energy buffering is the only one left uncovered. A device to control batteries was developed based on the microcontroller ATMega8 [16]. Using this, we were able to control the voltage of each battery in the system. Monitoring of the current flowing through each section of the circuit became possible by connecting current sensors to each battery. Thus, we could follow the status of each battery separately to react to possible failures in time. The circuit system described above is shown in Figure 4.

While designing the monitoring system, we had to solve some problems. In places where it was planned to install such systems to communicate with the "main land", the satellite channel is mostly used. The GSM channel is available in exceptional situations. However, both of them are quite expensive, so we needed to minimize the amount of data traffic, but not at the expense of the volume of data transmitted. In addition, it should be noted that the satellite channel is sufficiently sensitive to weather conditions and so disconnections can take place. Therefore, it was essential to provide a guarantee of either data delivery or delayed delivery.

We chose the JSON document format [17] as an internal data format for data storage and its transmitting to the mainland. This data format is very suitable for computer processing and, unlike XML, is compact enough. The binary data formats were rejected as the JSON documents appeared to be more compact than the binary ones [17].

As mentioned above, all the data from the system of the alternative energy is read using the single-board computer RaspberryPI. It takes readings from all the devices connected to it twice a minute, forming a "snapshot" of the entire system at any given moment in time. The collected data is added to the daily log, which is stored in a separate file. Originally, we considered the asynchronous data collection from each device with a separate entry in the log because each parameter has its own validity interval. The weak point of this method is the difficulty to carry out correlation of the readings, since we still have to register all the data at one time.

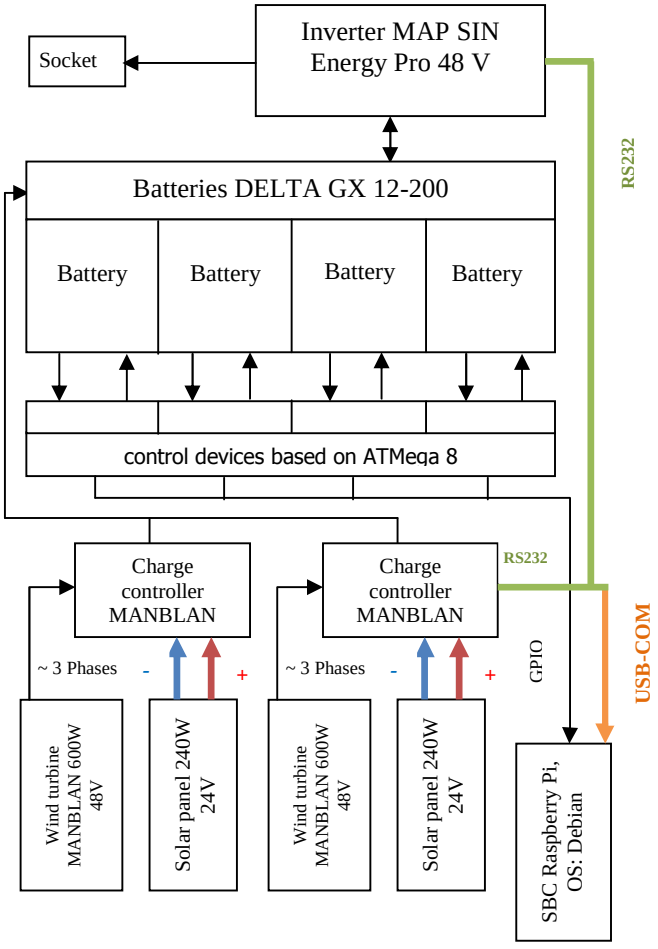

Figure 4. System circuit.

The problem of the unstable channel was solved by using the synchronization system Rsync [18] through the SSH connection to the server. The embedded Rsync data

 ISBN: 978-1-61208-362-9

compression algorithms significantly reduced the amount of the data traffic. The streaming compression of SSH connection did not give such results. Rsync also allowed us to solve the problem of disconnection as it has the algorithms of the file difference transmission.

Once the data is delivered to the server, the documents are sorted out and the sorted data is carried over into the single database. PostgreSQL [19] was decided to be the database due to its ability to use multiple programming languages in store procedures [19]. Based on the incoming data, the database automatically counts and updates daily, hourly and monthly reports for each installation point. This data transmission is presented in Figure 5.

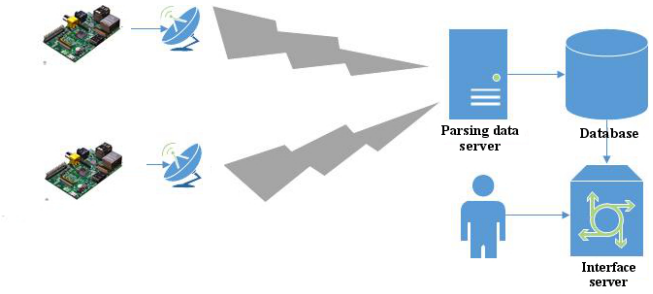

Figure 5. Data traffic pattern.

IV. RESULTS

This project resulted in designing the hardware and software complex that allows remote monitoring of the alternative energy systems through the data collection from the maximum number of sources. Such sources can include not only ready devices such as charge controllers but also a variety of additional current and voltage sensors. The configurability of such systems explains the integration of the scaling feature of several devices and sensors that need monitoring. If necessary, the system supports connection to some additional devices, for example, a portable weather station. Our laboratory tests were run in the vicinity of the meteorological stations, so we did not need such data. However, if these systems are required to be installed in remote areas, the weather information is vital to control the entire system.

Data transmission takes a special place in our system. We took its design seriously. The foreground task was to ensure a reliable transmission of the telemetry data and its safety. According to the research results, this task was accomplished.

We generated a web-interface to control the system operation. This interface allows the available data to be viewed in a convenient form from each system designed by our team. Currently, we can view the archival data obtained from the solar panels located at Novaya Zemlya as well as the data from the island Zhizhgin. The data received from our recent tests is preparing to be published. The web-based interface was originally built on the PHP framework Yii [20]. However, for technical reasons, it was moved to a python framework Django [21]. The main reason for changing the framework was rather small amount of code if case of using Django. One more reason was the number of programmers in our team who know syntax of python so we could support this product better and faster. The main function of data-viewing was realized. We plan to increase the functionality of the application to query the historical data and to compare it in different variants. The Yii framework screenshot is presented in Figure 6.

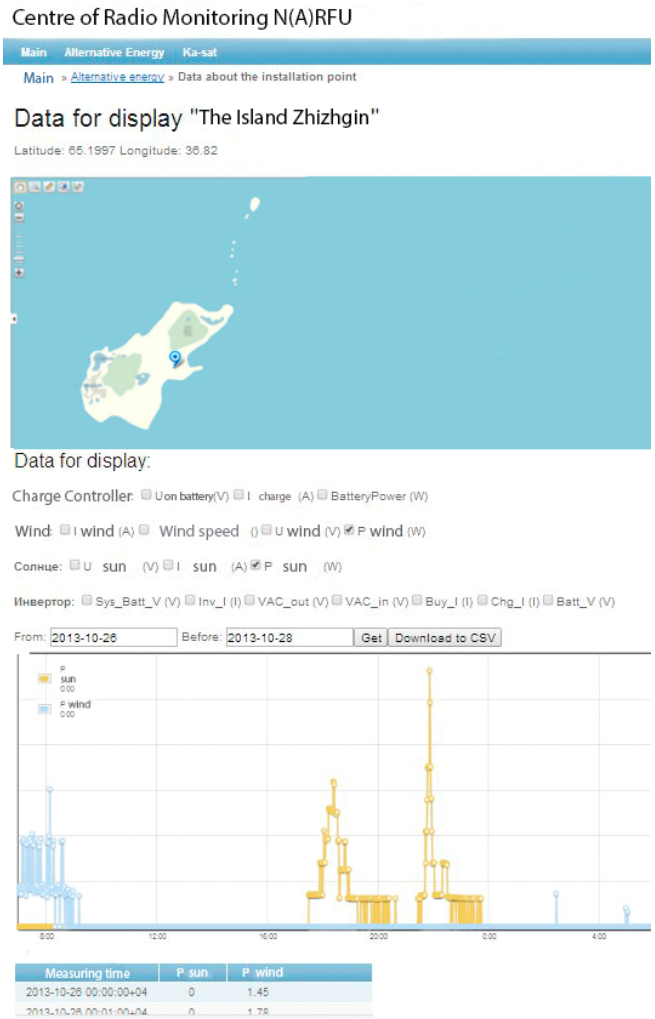

Figure 6. Web-interface, created on the Yii Framework.

In general, on the grounds of the research results, it may be concluded that the use of alternative energy systems in the Arctic region is highly promising. However, the use of such systems in the northern regions requires constant monitoring and studying the effect of the weather conditions. These scientific studies target long term testing. Presently, such systems are rarely used in the North. We need to determine how long they are able to work to be successfully and efficiently used.

 ISBN: 978-1-61208-362-9

V. CONCLUSION AND FUTURE WORK

After tests, we can conclude that, in the conditions of the northern latitudes, the use of horizontal wind turbines is not suitable because of their instability to gusty winds. The vertical wind generators show a better performance, but still require some modifications to strengthen their structure. During the summer, the wind turbines produce power stably. However, their power depends strongly on the wind direction and strength variability.

In the future, we plan to use horizontal wind turbines along with vertical ones for a more stable work. Our last research in data acquisition and representation shows better results in case of using Zabbix as monitoring system [22]. While using this system, we had interesting results. But we also experienced some problems in its performance and user interface. In the future, we are planning to solve these problems and use the system with our own interfaces.

REFERENCES

[1] D.A. Dodin, A.N. Evdokimov, V.D. Kaminsky, “Mineral resources Russian Arctic (condition, prospects, research areas)”, St. Petersburg.: Science, 2007.

[2] V.I. Pavlenko, ‘Arctic zone of the Russian Federation in the system of national interests” // Arctic: ecology and economy, № 4 (12), 2013, p.16-25.

[3] Aleksandre Akimov, Vitaly Poleshchuk, Denis Shevchuk, “Spacecraft service area modeling for constellations based on “Tundra””, [Online]. Available from: http://www.tssonline.ru/articles2/sputnik/modelirovanie-rabochey-zony-sputnikovoy-gruppirovki-sformirovannoy-na-orbite-tundra-spacecraft-service-area-modeling-for-constellations-based-on-tundra 2014.01.28.

[4] TEleSat Systems, “How to choose a satellite phone - Globalstar, Thuraya, Iridium, Inmarsat”, [Online]. Available from: http://www.tecckom.ru/satellite_choose.php 2014.02.21.

[5] ORBOCOMM, “ORBCOMM”, [Online]. Available from: http://www.orbcomm.com/ 2014.03.18.

[6] Gonets, “Gonets leostat system”, [Online]. Available from: http://english.gonets.ru/Page206.html 2014.03.18.

[7] S.C.S. Jucá, P.C.M. Carvalho and F.T. Brito, “A low cost concept for data acquisition systems applied to decentralized renewable energy plants”, Sensors, 2011, vol.11, pp. 743-756.

[8] M. Benghanem, “A low cost wireless data acquisition system for weather station monitoring”, Renewable Energy, 2010, vol. 35, pp.862–872.

[9] Raspberry Pi Foundation, “Raspberry Pi”, [Online]. Available from: http://www.raspberrypi.org/ 2014.03.17.

[10] S. Rosiek and F. Batlles, “A microcontroller-based dataacquisition system for meteorological station monitoring” Energy Conversion and Management, 2008, vol. 49, pp. 3746-3754.

[11] Alpek Co LTD, “MBWP-400W User manual”,. [Online], Available from: http://manblan.ru/files/manual_MBWP_400W-EN.pdf 2013.12.27

[12] Alpek Co LTD, “Qsolar specification”, [Online]. Available from: http://manblan.ru/files/solarmodule.pdf 2014.01.15.

[13] Alpek Co LTD, “User manual MBWS-B1”, [Online]. Available from: http://manblan.ru/files/MBWS-B1%20manual%20eng.pdf 2014.01.17.

[14] Delta battery, “DELTA Batteries for communication systems”, [Online]. Available from: http://www.delta-batt.com/upload/iblock/548/Delta%20GL12-200_eng.pdf 2014.01.17

[15] “MAC "ENERGIA" SINE inverter”, [Online]. Available from: http://macenergia.com/ 2014.01.23.

[16] Atmel Corporation, “ATMega 8 datasheet”, [Online]. Available from: http://www.atmel.com/images/atmel-2486-8-bit-avr-microcontroller-atmega8_l_datasheet.pdf 2014.01.18.

[17] “JSON”, [Online]. Available from: http://json.org/ 2014.03.23.

[18] W. Davison, “rsync.samba.org,”. [Online]. Available from: http://rsync.samba.org/examples.html 2014.02.15.

[19] The PostgreSQL Global Development Group, “PostgreSQL: Documentation: 8.0: PL/pgSQL - SQL Procedural Language”, [Online]. Available from: http://www.postgresql.org/docs/8.0/static/plpgsql.html 2014.02.21.

[20] Yii Software LLC, “Yii PHP Framework: Best for Web 2.0 Development”, [Online]. Available from: http://www.yiiframework.com/ 2014.02.12.

[21] Django Software Foundation, “The Web framework for perfectionists with deadlines | Django”, [Online]. Available from: https://www.djangoproject.com/ 2014.04.03.

[22] Zabbix SIA, An Enterprise-Class Open Source Distributed Monitoring Solution, [Online]. Available from: http://www.zabbix.com/ 2014.05.20.

 ISBN: 978-1-61208-362-9