

Towards A User-Centric Identity-Usage Monitoring System

Daisuke Mashima
College of Computing, Georgia Institute of Technology
mashima@cc.gatech.edu

Mustaque Ahamad
College of Computing, Georgia Institute of Technology
mustaq@cc.gatech.edu

Abstract

The misuse of identity-related information in cyberspace is one of the biggest concerns among all Internet users. So far several technologies have been proposed and implemented to prevent and detect identity theft and misuse, but none of them are completely successful in terms of privacy, user-centricity and generality. In this paper, we identify the requirements for a user-centric identity-usage monitoring system to solve such problems and propose a monitoring system that runs on a trusted third party. This system can transparently use context information of a request to detect anomalous use of online identity. Finally, we provide a prototypical implementation in an OpenID setting and evaluate it in terms of scalability, performance, user-centricity, and security.

1. Introduction

Today, theft and misuse of online identity credentials is one of the biggest concerns among Internet users. According to Federal Trade Commission's Consumer Fraud and Identity Theft Complaint Data released in February, 2007 [1], identity theft is ranked in the first place for the seventh year in a row with more than 240,000 complaints. Internet-related complaints account for more than 10% of total complaints. To combat the theft and misuse of identity credentials, almost all online service providers (SP), also called relying parties (RP) in the identity management area, rely on network security technologies. For instance, online banking systems use SSL/TLS to protect users' sensitive information by encryption. Recently new techniques such as a site authentication image, for example Bank of America's SiteKey [2], are introduced. However, Schechter [3] shows through experiments that they are not effective enough. In addition, physical theft of users' devices, in which identity information could be accumulated, is also a potential threat. Symantec [4] reports that theft or loss of computers or data storages accounted for 46% of identity theft cases in the first half of 2007. Malware and social engineering techniques are also possible threats. Since there are numerous ways to steal user's online identity, it is almost

impossible to counter all of them. Therefore, to minimize losses or damages caused by identity theft and resulting misuse, an identity-usage monitoring system is required. So far a number of mechanisms that aim to monitor and detect anomalous identity-related transactions have been proposed. Fraud detection systems [5][6][7] are typical examples. Though these schemes are successful to some extent, they have applicability only to specific domains and suffer from problems in terms of privacy, user-controllability, and generality.

In this paper, our primary focus is to identify requirements for a user-centric identity-usage monitoring system that can address problems that exist in current systems. We advocate deploying an identity-usage monitoring system that relies only on context of identity usage, such as timestamp, location, device characteristics, and so on. Such a system runs on a trusted third party that stores and releases user's identity credentials under user's intention, for example an OpenID provider [8]. Our system is best suited for architectures where the trusted third party is already involved in identity-related transactions. With the growing popularity of single sign-on systems and identity management systems, many users will manage their online credentials where this assumption holds. We also discuss how to capture and use context information for identity-usage monitoring.

We will first discuss related work in the next section. Then, in section 3, we will identify design principles for a user-centric identity-usage monitoring system. Section 4 focuses on context information that we are able to utilize in our monitoring system. We will show a prototype of our system in the OpenID setting and evaluate it in section 5. Finally, we will conclude this paper in section 6.

2. Related Work

2.1. Identity Management Systems

In general, the purpose of identity management systems is to manage and store identity securely and efficiently. Among them, *user-centric models*, such as Windows CardSpace [9] and OpenID [8], aim to achieve user control. For example, in CardSpace, users can choose an appropriate identity credential for each

transaction [10]. VeriSign PIP [11] enables users to select to whom and how much information is disclosed. As pointed out in [12], such user control is imperative for privacy and user-centricity. Although detection of identity theft and misuse is not the primary focus of these identity management systems, well-designed and user-centric identity management systems are expected to reduce such risks.

2.2. Fraud Detection Systems

Fraud detection systems have been designed with a similar objective as ours. For example, in e-commerce setting, fraudulent usage of credit card numbers is one obvious problem to be detected by our monitoring system. Several companies provide solutions that use a variety of information to detect fraudulent activities. Bharosa Tracker [13] works behind the scene by utilizing context information, such as IP address, geographic location, device characteristics, and workflow information. Also, some research has been done in the academic field [5][6][7]. But there remain problems in terms of privacy, generality, and user-centricity, which will be elaborated in section 3.

3. Monitoring Approach

3.1. Problems of Fraud Detection Systems

First of all, most fraud detection systems are located in SP sites and managed by them, which means that there is almost no room for users to exercise control. Another problem is user's privacy. Some fraud detection mechanisms utilize not only contents of transaction but also context information, including user's geographic location, and user's personal information, such as address of the residence. Since these are collected outside of user's control, users need to worry about privacy and risk of information breach. Furthermore, although such information might not be essential to provide services, redundant information is captured without giving users any options, which is controversial in terms of "User Control and Consent" and "Minimum Disclosure" laws outlined for identity management systems [14]. The other problem is generality and scalability. Since fraud detection systems rely on the contents of transactions, the system needs to be truly service-specific. Therefore, different systems are required to support different services.

3.2. Design Implication for Our System

To enhance user control and mitigate privacy concern outlined in section 3.1, a monitoring system should be located outside of SP's domain so that the system is accessible to users. There are two possibilities: deploying the system on users' devices or on a trusted third party.

If a monitoring system is running on a user's device, it is completely under user's control, and there is no privacy concern. Also, it is easy to obtain detailed identity-usage information. However, it has one critical drawback. When the device itself is compromised, there remains nothing to detect identity misuse because the monitoring system is compromised as well. In addition, as a deployment issue, some kind of secure client software needs to be installed on users' device, which might limit the scalability.

As for the second approach in which the monitoring system runs on a trusted third party, even if a user device is compromised, anomalous identity usage can be detected. But acquisition of usage information becomes more difficult. In this case, the monitoring system needs to capture relevant information about the source of a request over the network. This implies that the amount and precision of the information could be limited. Since the monitoring system resides outside of SP's site and user's device, the information must be provided by at least one of them, preferably a SP because of user devices' susceptibility to physical theft and compromise. Moreover, the monitoring system should be actively contacted because otherwise a SP (or a user device) needs to keep a communication port open for it to be contacted by the monitoring system, which could add extra vulnerability. We should also consider ways to prevent SPs from omitting or refusing to provide the monitoring system with usage information and to encourage them to report the usage.

Based on these observations, we propose to implement the monitoring system on a user's "identity provisioning service" (IPS), which is not an authoritative identity provider but stores and releases users' identity credentials on behalf of users under users' intention, such as an OpenID provider, Homesite in Sxip [15], VeriSign PIP, and Identity Agent in GUIDE-ME [16]. These entities are trusted by users and work for users' sake. Thus, they can be considered in user's trust domain, outside of the SP's control. Also, since SPs need identity credentials provided by IPS, SPs can not avoid interacting with IPS anyway.

Although the monitoring system resides at users' IPSs, users should have control over which information about a request is captured by the monitoring system and whether a certain transaction is monitored or not. Of course, such choice must be made only by the authorized user with his/her explicit action to prevent the monitoring system from being disabled by an adversary as well as to avoid being inadvertently turned off by a user. One solution is to provide an interface accessible via a secondary authentication path by which users can easily access and check the status and change configuration of the monitoring system. In addition, our system, especially an anomaly detection feature, should be able to work correctly with any choice of context parameters since some users will agree on providing only some of them. Such flexibility minimizes

privacy concern and increase user control over the identity-usage monitoring.

Regarding generality problem, our approach relies solely on context information. By following widely-used identity management schemes and conducting coarse-grained monitoring that audits the context when identity credentials are requested and presented at the beginning of a user session with a service, we can focus only on the use of identity credentials. This offers more generality and privacy than most fraud detection systems which look into actual transaction parameters, and hence our system can be used in services ranging from blogs to e-commerce.

In [7], it is pointed out that utilizing contents of a transaction is mandatory for the sake of precise detection. Although we did not evaluate it, it is certainly possible that reducing the number of parameters would reduce the precision of anomaly detection. Thus, we need to compensate for this disadvantage. One way is to more closely engage users in anomaly detection process. We propose to push a summary of identity usage to a user periodically, say once a day, in addition to sending an alert when an anomaly is detected. By doing so, we can grab user's attention and give him/her an opportunity to deal with the problem. As for the anomaly detection algorithm, though it is generally required to minimize both false positive and false negative rates, we need to focus more on minimizing the false negative rate because false alarms, as long as they are not too many, help users pay attention to identity usage. In addition, if the system has an interface to receive feedback from users, which tells the monitoring system that the alarm is correct or not, the user profile in an anomaly detection model can be updated quickly.

4. Context Information of Identity Usage

Available context information can be divided into three categories: "From Who and to Whom," "Where," and "When." In this section, we will elaborate each of them assuming that our monitoring system is implemented on a user's IPS. We eliminate the option to use dedicated client-side software to send context information because such subjective information may not be reliable in case a user device is compromised.

4.1. Who and to Whom?

Probably the most important context is who attempts to use the identity credential. Unfortunately, the monitoring system can not simply rely on users' online identifier because someone else could be trying to impersonate the legitimate user. Thus, practically it is very difficult to know the person who claims to present an identity credential. Instead, we utilize characteristics of devices that users are using since users are assumed to use a limited number of devices to access online services. Therefore, relying on

device characteristics that are tied to a user is a viable and reasonable solution.

Assuming that a trusted computing module (TPM) sends device information to the system [17] is a possible solution, but unfortunately TPM has not yet been widely deployed. More generally, the operating system running on user's device can be identified by OS fingerprinting. Basically, OS fingerprinting exploits distinctive characteristics in TCP/IP protocol stack of each operating system. They are divided into two categories: active and passive ones. Active schemes, such as Nmap, identify OS by sending specially crafted packets to a target machine and see how the machine replies to them. But they are hindered by firewall or some other reasons. On the other hand, passive ones, such as p0f [18], just sniff packets transmitted by a target computer, which is suited for use on IPS. In web settings, the system can also use browser type as a part of device information. The common technique is to utilize User-Agent header field in HTTP [19].

Information about a SP to whom an identity credential is disclosed is given as a name or some type of identifier of a targeted relying party. It is typically provided by either users or SP themselves. Since an IPS, on which our system is running, is involved in each identity-related transaction, we can assume that an identifier of SP is provided via IPS.

4.2. Where?

GPS and RFID tags are becoming popular in location-based services. However, requiring users to always carry special devices is not desirable in terms of scalability and usability. In addition, RFID is supposed to be used in relatively short range, so it can be used only in limited places.

Alternatively, IP addresses of sources of requests can be used to determine approximate geographic location of a user. There are several ways to get the geographic location from an IP address. Some methodologies using DNS [20] and WHOIS are proposed. However, the availability and precision of those schemes are very limited.

Another way to identify geographic location from IP addresses is to use IP geo-location systems. There are various techniques available today. MaxMind [21] provides an API and database that enable us to look up location information locally. Delay-based schemes [20][22] are also available. Octant [23], which combines geographical and demographical constraints with delay-based approach, is recently proposed. These techniques outperform DNS-based and WHOIS-based ones. According to Wong [23], the median of error distance of Octant is only 22 miles. Thus, we suggest using IP geo-location.

Though it is not precise enough to identify the location, WHOIS is still useful to know through which internet service provider a user is accessing or to which organization a user belongs, which can be considered as another context.

4.3. When?

Time information can basically be obtained through timestamps of each identity credential usage. In this sense, time information is available in most settings. Through timestamps, we can derive a lot of useful information to profile users, such as frequency of usage within a certain time period, usage interval, day of week, week of month, and so forth. Usually user's access time tends to fluctuate, so it is not good to build user's normal profile based on raw timestamps. This problem can be avoided by categorizing timestamps meaningfully, say MORNING, AFTERNOON and EVENING [5], or by building more flexible categories by using Calendar Schema [24].

4.4. Context-based Anomaly Detection

In this section, we will briefly see how we can utilize context information to detect anomalous usage pattern. Detailed design of the algorithm is included in future work.

In general, the identity thief is expected to use the stolen credential, such as the pair of user ID and password, as much as possible before being detected. This is because the stolen information is no longer useful once the legitimate owner notices the illegal usage and invalidates the credential by revoking it. Thus, keeping track of the frequency of usage is an effective way to detect misuse.

Even if the adversary is cleverer, he would try not to use the credential too much to evade detection. In this case, frequency-based method could fail to detect the problem. However, if he lives in different region from the actual owner, location information should indicate anomaly. Therefore, anomaly detection by focusing on unobserved context in past usage or geographic distance based scheme could be used. Likewise, anomaly detection in terms of time categories and device characteristics is also promising. Such detection schemes might be too strict, but, as mentioned earlier, acceptable amount of false alarms will not only make users attentive but also assure users that the identity usage is actively monitored.

5. Prototype Implementation in OpenID

In this section, we will see one proof of our proposed concept in the OpenID architecture [8]. As major companies start providing OpenID services, such as VeriSign's PIP [11], OpenID is getting popular and will be increasingly used. In addition, OpenID Attribute Exchange 1.0 [25] has been finalized in 2007, which enables an OpenID provider to also work as an identity attribute provider. However, the security of OpenID is not fully established [15]. For example, an adversary could steal users' password for OpenID providers by phishing. Once users' passwords are stolen, current OpenID providers do

not have effective ways to prevent identity misuse. The same thing can happen to other IPSs listed in section 3.2. Note that such interaction between users (or users' devices) and IPSs is directly tied to identity-usage in these architectures.

5.1. Overview of Implementation

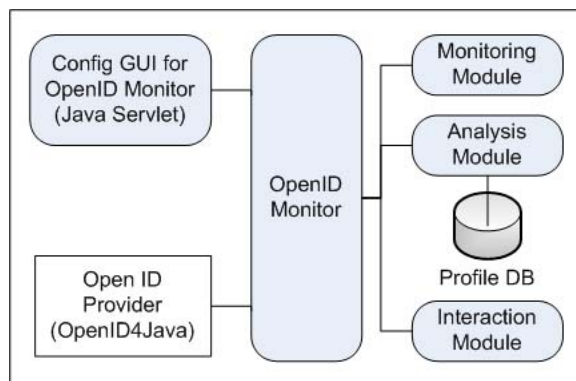


Fig1: Overview of OpenID Monitor

The overview of our prototype implementation is shown in Fig. 1. In this prototype, we used OpenID4Java library and implemented our system in Java. The rounded rectangles are the modules we added to a normal OpenID provider, *OpenID Provider* in the figure. As can be seen, our system consists of 5 parts. *OpenID Monitor* is the main module which controls the other sub modules, *Monitoring*, *Analysis*, and *Interaction*. These are modularized based on functionality. *Monitoring* captures context information, *Analysis* stores user profiles and detects anomalies by using observed information, and *Interaction* performs an appropriate action to notify a user based on the detection result. *Monitoring* uses p0f [18] and MaxMind GeoIP [21]. Also, it communicates with a WHOIS service. Currently, *Analysis* does simple anomaly detection which raises an alarm when unseen context information is observed. To detect an anomaly in time profile, we utilize time categories defined by Calendar Schema [24] ($w=1, \dots, 5$ (week of month), $d=1, \dots, 7$ (day of week), and $h=1, \dots, 8$ (3-hour category)). A triple unseen in the past is suspected as an anomaly. As for location, we use city name and country name since more than half of fraud cases are detected only by country-level geo-location, according to MaxMind [26]. *Interaction* sends periodic usage reports and real-time alert Emails, which contain context information of identity-usage. *Config GUI* is accessed by a user, through separated and independent password-based authentication over SSL, to control *OpenID Monitor*. Through the *GUI*, each user can turn on or off *OpenID Monitor*. A user also can choose which context information is used by *OpenID Monitor*. *Config GUI* also works as an interface to obtain a user's feedback about the correctness of an alarm.

5.2. Evaluation

5.2.1. Extensibility and Deployability. Because of the modular architecture, the functionalities of *Monitoring*, *Analysis*, and *Interaction* can be modified without significantly impacting other parts. For example, how to obtain context information largely depends on underlying system architecture and the way of deployment, so *Monitoring* needs to be tailored to each architecture. The anomaly detection algorithm used in *Analysis* can be replaced with a more sophisticated one easily. In addition, since the only interface with the external system, *OpenID Provider* here, is the main module, integration into different architecture is expected to be feasible. No change is required at the user side, so it is possible to integrate the monitoring system even into Email systems, aiming to make EBIA [27], in which the ability to receive Emails is considered as an identity credential, more secure. The other point to be noted here is that unlike fraud detection systems, our system can be used with any type of SPs that rely on OpenID.

5.2.2. Performance. The increased functionality is expected to increase the response time for processing each request. Therefore, we measured the additional overhead with our prototype. The setting of our experiments is shown below. To generate dummy authentication requests, we used Microsoft Web Application Stress Tool (WAS) on the user's PC. We prepared two *OpenID Providers*: one of them includes *OpenID Monitor* invocation (OP2), and the other without it (OP1). In this experiment, for simplicity, OP1 and OP2 do not interact with human users at all.

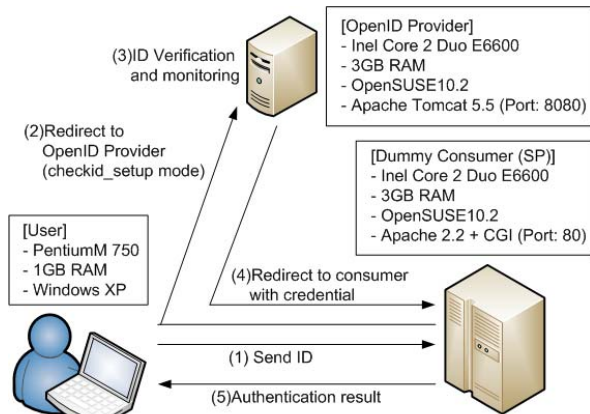


Fig2: Setting of Experiment

The results are shown in Table 1. Each value in the table is an average of 5 trials. Values in “Threads” column mean the maximum number of concurrent threads sending requests. When the number of threads is 1, a request is sent after the response to the previous request is received. Thus, we can estimate the response time as an inverse of the value in requests-per-second column. In addition, LAN means that the client PC is connected to the same local area

network as the server while CATV means that the client is connected to an external cable TV internet. In LAN, the network delay can be considered negligible, so the results can be used to evaluate the pure delay caused by *OpenID Monitor*. According to the table, the delay is approximately 0.1 second. In CATV, which is closer to realistic environment, the response time does not exceed 1 second and is much shorter than the 4-second threshold [28]. It exhibits good throughput even in multi-user setting. The results of 5-threads experiments can be regarded as the upper bound of the throughput for both of OP1 and OP2 because experiments more than 5 threads did not show any major difference.

Table 1: Results of Performance Evaluation

| Network | Threads | Type | Req. / Sec | Time / Req. |
|---------|---------|------|------------|-------------|
| LAN | 1 | OP1 | 2.254 | 0.443 |
| | | OP2 | 1.782 | 0.566 |
| CATV | 1 | OP1 | 1.614 | 0.612 |
| | | OP2 | 1.404 | 0.712 |
| | 5 | OP1 | 4.508 | - |
| | | OP2 | 3.708 | - |

5.2.3. Usability and User-friendliness. Concerning system-user interaction, our system is better than other existing OpenID-based services. For example, even though VeriSign PIP logs transaction, it does not actively notify users, which requires users to access PIP to see if something wrong has happened. By Emailing periodically or in real-time in case a suspicious request is detected, our system succeeds in reducing users’ burden and provides users an opportunity to notice problems sooner. Additionally, using context information makes the report more user-friendly. It is obviously much more understandable to write “Japan” than to write the raw IP address. This makes a window of vulnerability shorter.

5.2.4. Security Analysis. As for the threat of phishing pointed at the beginning of section 5, our system makes identity misuse more difficult. An adversary additionally needs to care about access time, location and device he uses to succeed in impersonating a user without being detected. If an adversary could imitate completely a user’s access pattern, he would avoid the detection. But his access is logged anyway and included in the periodic report to a user, which helps a user notice the problem. Likewise, if an adversary attempts to replay a sniffed request from a user, our monitoring system can detect it. Just like common phishing schemes, an adversary could forge the SP site, instead of an OpenID provider site, to retrieve user’s identity credential from an OpenID provider by cheating a legitimate user. In this case, even though none of timestamp, geographic location, and device information might look suspicious, *OpenID Monitor* can detect anomaly based on information about SPs. Monitoring those parameters is also effective to prevent OpenID Realm Spoofing attack [29]. Owing to the fact that *Config GUI* can

control the functionalities of *OpenID Monitor*, the attack against it could be a concern. By using separate and stronger authentication credential from one that is used for OpenID authentication, such risk is reduced because the compromise of user identity credentials does not imply the compromise of the monitoring system, and vice versa. It is crucial that *OpenID Monitor* shares no data with *OpenID Provider*. Finally, we are assuming that IPSs, an OpenID provider in this setting, are protected securely. We believe this assumption is realistic since trusted third parties manage their own security well for the sake of their reputation and credibility.

6. Conclusion and Future Work

In this paper, we presented the requirements for a user-centric monitoring and detection system for online identity usage. Also, by implementing a prototype in the OpenID setting, we showed that our idea is viable as well as effective. Though this implementation is just one example, we believe that design implications presented should work in other scenarios. As Maliki summarizes [15], most Identity 2.0 systems do not have enough protection against identity attacks like phishing. We expect that our work is one way for addressing this shortcoming.

As future work, we will explore other environments where our system can be integrated, such as Email systems. Also, more sophisticated context-based anomaly detection algorithm will be explored.

7. Acknowledgement

This work is supported in part by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001, under the auspices of the Institute for Information Infrastructure Protection (I3P) research program. The I3P is managed by Dartmouth College. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security, the I3P, or Dartmouth College.

8. References

- [1]FTC, "Consumer Fraud and Identity Theft Complaint Data," <http://www.consumer.gov/sentinel/pubs/Top10Fraud2006.pdf>
- [2]"SiteKey at Bank of America," <http://www.bankofamerica.com/privacy/sitekey>
- [3]Stuart E. Schechter et al., "The Emperor's New Security Indicators An evaluation of website authentication and the effect of role playing on usability studies," in The IEEE Symp. on Security and Privacy, 2007
- [4]"Symantec Internet Security Threat Report Volume XII," <http://www.symantec.com/business, Sep. 2007>.
- [5]Fawcett, T. and Provost, F., "Adaptive fraud detection" in Data Mining and Knowledge Discovery, 1(3), 1997
- [6]R. J. Bolton and D. J. Hand., "Statistical fraud detection: A review," in Statistical Science, 17(3), 2002.
- [7]Rosset, S., Murad, U., Neumann, E., Idan, Y., and Pinkas, G., "Discovery of fraud rules for telecommunications-challenges and solutions," in Proc. of KDD-99, 1999
- [8]David Recordon and Drummond Reed, "OpenID 2.0: A Platform for User-Centric Identity Management," in Proceedings of the 2nd ACM workshop on DIM, 2006
- [9]David Chappel, "Introducing Windows CardSpace." <http://msdn.microsoft.com/library/enus/dnlong/html/IntroInfoCard.asp>, 2006.
- [10]A. Bhargav-Spantzel, et al., "User centricity: A taxonomy and open issues," in Proc. of the 2nd ACM Workshop on DIM, 2006.
- [11]"Personal Identity Provider (PIP)." <https://pip.verisignlabs.com/learnmore.do>
- [12]Craig W. Thompson and Dale R. Thompson, "Identity Management" in Internet Computing, IEEE, 2007
- [13]"Bharosa Tracker," <http://www.bharosa.com>
- [14]Cameron, K., "The Laws of identity," <http://msdn2.microsoft.com/en-us/library/ms996456.aspx>
- [15]Tewfiq El Maliki et al., "A Survey of User-centric Identity Management Technologies," in SECUREWARE 2007, 2007.
- [16]"GUIDE-ME: Georgia Tech User-Centric Identity Management Environment," isis.poly.edu/disw07/presentations/Ahamad%20Poly%20Talk.ppt
- [17]Jaeger T. et al., "PRIMA: Policy Reduced Integrity Measurement Architecture", in The 11th ACM Symp. on Access Control Models and Technologies, ACM, 2006.
- [18]M. Zalewski, "p0f: Passive OS Fingerprinting tool." <http://lcamtuf.coredump.cx/p0f.shtml>.
- [19]Shreeraj Shah, "Browser Identification for Web Applications," http://net-square.com/whitepapers/browser_ident.pdf
- [20]Bamba Gueye et al., "Constraint-based geolocation of Internet hosts," in Proc. of the ACM/SIGCOMM Internet Measurement Conference 2004, 2004.
- [21]MaxMind, "GeoIP," <http://www.maxmind.com>
- [22]Ethan Kats-Bassett et al., "Towards IP Geolocation Using Delay and Topology Measurements," in Proc. of the 6th ACM SIGCOMM on Internet measurement 2006.
- [23]Bernard Wong, et al., "Geolocalization on the Internet through Constraint Satisfaction," in Proc of WORLDS, 2006.
- [24]Y. Li, N. Wu, S. Jajodia, and X. S. Wang, "Enhancing Profiles for Anomaly Detection Using Time Granularities," in Journal of Computer Security, 10(2), 2002.
- [25]D. Hardt et al, "OpenID Attribute Exchange 1.0 – Final," http://openid.net/specs/openid-attribute-exchange-1_0.html
- [26]"MaxMind Fraud Detection Whitepaper," <http://www.maxmind.com>
- [27]Simson L. Garfinkel, "Email-Based Identification and Authentication: An Alternative to PKI?" in IEEE Security & Privacy, 1(6), 2003.
- [28]Akamai, "Retail Web Site Performance," <http://www.akamai.com/4seconds, 2006>
- [29]"OpenID Phishing Brainstorm", http://wiki.openid.net/OpenID_Phishing_Brainstorm