SURVEY PAPER

A survey on security visualization techniques for web information systems

Tran Khanh Dang and Tran Tri Dang
Faculty of Computer Science and Engineering, University of Technology, Vietnam National University, Ho Chi Minh City, Vietnam

Abstract

Purpose – By reviewing different information visualization techniques for securing web information systems, this paper aims to provide a foundation for further studies of the same topic. Another purpose of the paper is to discover directions in which there is a lack of extensive research, thereby encouraging more investigations.

Design/methodology/approach – The related techniques are classified first by their locations in the web information systems architecture: client side, server side, and application side. Then the techniques in each category are further classified based on attributes specific to that category.

Findings – Although there is much research on information visualization for securing web browser user interface and server side systems, there are very few studies about the same techniques on web application side.

Originality/value – This paper is the first published paper reviewing extensively information visualization techniques for securing web information systems. The classification used here offers a framework for further studies as well as in-depth investigations.

Keywords Security visualization, Web browser security, Usable security, Human security, Secure user interface design, Web, Information systems

Paper type General review

1. Introduction

Today web information systems are used widely because of their platform-independence, convenience, and rich capability nature. With this widely use, there is also a need for protecting web information system components. These components includes web users, web browsers, web applications, and server-side systems. Although these components are highly related, they are usually studied separately because of the complexity involved when handling the whole systems and the significant differences between them. In particular, current researches tend to study either web browser security, web application security, or server-side system security individually.

The techniques used to secure web information systems range from manual human operated techniques to fully automatic techniques. Security visualization techniques fall between these two extreme approaches. Security visualization still requires human users in the processing chain, but it also provides them with powerful visual tools to reduce their efforts on manual tasks. This semi-automatic approach exploits the advantages of both the computer processing system and the human processing system.

International Journal of Web Information Systems
Vol. 9 No. 1, 2013
pp. 6-31
© Emerald Group Publishing Limited
1744-0084
DOI 10.1108/17440081311316361

The main reasons why security visualization techniques for web information systems are surveyed in this paper are:

- The wide range of application of security visualization techniques currently used in web information systems.
- The supports provided by security visualization techniques to human users. This is a valuable complement to automatic solutions.
- The emerging need of new approaches to deal with web information system security problems.

The rest of this paper is structured as follows: in Section 2, the general benefits of using information visualization for security problems and the challenges involved are identified; in Section 3, we propose a classification of security visualization techniques used on web information system security; the details of each branch of the classification is studied from Sections 4 to 6; and finally, in Section 7, we conclude the paper by pointing out the gaps in current researches and suggesting directions for future studies on this topic.

2. Security visualization – benefits and challenges

People is an integral part of the security of information systems, and it is often the weakest part in the chain (Schneier, 2000). Therefore, it is important to design systems that are not only mathematically secured but also practically secured for normal uses. One way to do that is by improving systems' user interface, making it clear for users to understand the meanings of possible choices and their appropriate consequences. For more complicated situations, for example, network monitoring, intrusion analysis, etc. novel techniques are needed to enable users to quickly grasp the entire situation picture while still be able to concentrate on interesting portions. These security problems lead to the study of information visualization for security purposes. Marty (2008) mentioned two specific advantages of using visualization for security: first is the ability to process and present huge amount of data; and second is the capability of interactive exploration of data. In a broader sense of visualization, we believe there are two more advantages: visual presentations are more intuitive to users; and it can get users' attention more easily.

According to Card *et al.* (1999), information visualization is the use of computer-supported, interactive visual representations of data to amplify cognition. That means the output of visualization process is targeted directly at human users. Therefore, to measure the efficiency and effectiveness of security visualization techniques, the characteristics of human users need to be taken into account. Researchers usually set up controlled environments in which test users are asked to do some specific tasks. The experiment results are then analyzed by statistical models to rate the differences between visualization techniques. The problems with this approach is that controlled environment is different from production environment; and different users have different levels of knowledge, skills, and interests, so the measurement results may be different from reality. Plaisant (2004) listed the challenges with current information visualization measurement methods and proposed possible steps to improve the evaluation and facilitate wide adoption of visualization tools by the public users.

3. A classification of security visualization techniques for web information systems

Web information systems consist at least the following components: client-side systems, server-side systems and web applications. Each type of component differs from each other in their roles, locations, and in particular their main users. The differences in users levels of knowledge, skills, and interests lead to the differences in goals, visual designs and interaction designs of visualization techniques applied on these components.

The users at client-side systems are ordinary people surfing the web for informational, entertainment or educational purposes. These users may also do more sensitive activities like shopping and banking with other institutions. The type of software communicates directly with these users is the web browser. Because security is not the main goal of these users, visualization techniques for this type of users focus on getting attention of users on suspected cases so that users can make informed decisions. The interface design here needs to be simple to make the learning process as easy as possible.

The users at server-side systems are administrators with more or less experience in networking, server configuration and security. These users are responsible for the normal operation of server-side systems, usually including web servers, database servers, other servers and some security tools (IDS/IPS, firewall, etc.). These users are more motivated to learn about their system security, so the visualization techniques and tools here tend to be more complex. The main goals of visualization techniques in this case are situational awareness, intrusion recognition, attack analysis, etc. and not just getting users' attention.

The users at web application side include web developers developing applications, administrators configuring and securing applications and end-users using applications. This mixed of user types makes developing security visualization techniques for web applications more complicated. In fact, our survey will point out that there is a lack of extensive research on security visualization techniques for web application component in the state-of-the-art literature of web information systems security visualization.

Based on the types of components described above, we first classify security visualization techniques according to where the techniques are used: techniques used on client-side systems, techniques used on server-side systems, and techniques used on web applications. Going down each branch, we will make further classification in its respective section. The first level of our classification is shown in Figure 1 with related references for each category.

4. Security visualization for client-side systems

At client-side, the most frequently used type of software is the graphical web browser. The functions of these web browsers are to get web page content and other information (e.g. SSL certificate data) from web servers and to render them onto the browsers' windows. Attacks that target web users mostly exploit vulnerabilities and design weaknesses in these browsers to trick users into executing malicious programs or providing sensitive information without aware of the consequences. One of the most popular types of attack on web users is phishing. These phishing attacks exploit weaknesses in browser interface design and/or human psychology nature to make users falsely recognizing malicious web sites as trusted ones and submitting sensitive

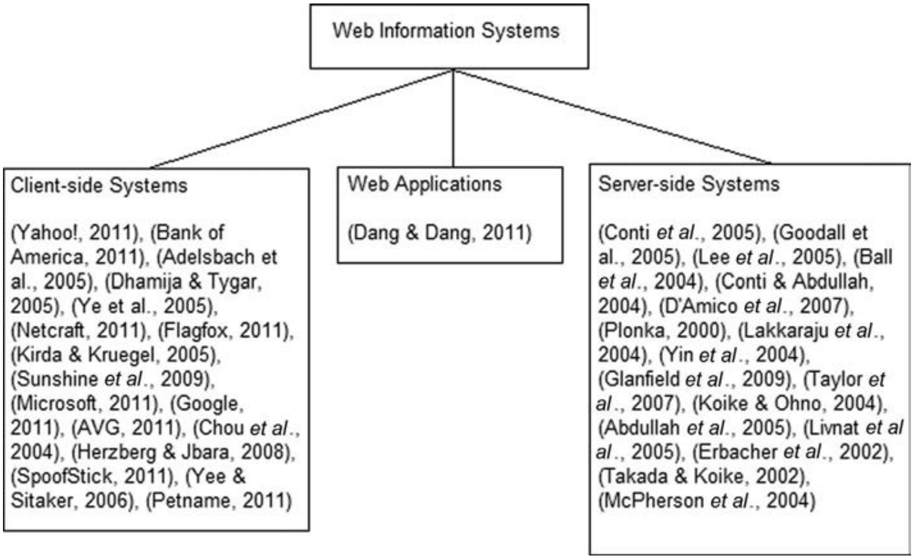

Web Information Systems
Client-side Systems
(Yahoo!, 2011), (Bank of America, 2011), (Adelsbach et al., 2005), (Dhamija & Tygar, 2005), (Ye et al., 2005), (Netcraft, 2011), (Flagfox, 2011), (Kirda & Kruegel, 2005), (Sunshine et al., 2009), (Microsoft, 2011), (Google, 2011), (AVG, 2011), (Chou et al., 2004), (Herzberg & Jbara, 2008), (SpoofStick, 2011), (Yee & Sitaker, 2006), (Petname, 2011)
Web Applications
(Dang & Dang, 2011)
Server-side Systems
(Conti et al., 2005), (Goodall et al., 2005), (Lee et al., 2005), (Ball et al., 2004), (Conti & Abdullah, 2004), (D'Amico et al., 2007), (Plonka, 2000), (Lakkaraju et al., 2004), (Yin et al., 2004), (Glanfield et al., 2009), (Taylor et al., 2007), (Koike & Ohno, 2004), (Abdullah et al., 2005), (Livnat et al al., 2005), (Erbacher et al., 2002), (Takada & Koike, 2002), (McPherson et al., 2004)

Figure 1.
First level classification of security visualization techniques on web information systems

information (e.g. password) to them. The reasons why phishing attacks can work successfully are described and summarized in Dhamija *et al.* (2006). These attack strategies may exploit users' lack of knowledge, deceit users' vision, or exploit users' lack of attention. Because knowing clearly about human behaviors is important for both phishing attackers as well as security system designers, there are related studies about the impact of human behaviors on phishing attacks, one example is Jakobsson (2007). Furthermore, the issue of human behaviors is more difficult to handle when special groups of users are considered, e.g. children (Iwata *et al.*, 2010).

To defense phishing attacks, human users must be tightly involved in the defensive process. Users may set up some kind of personalization or use their recognition ability to detect attacks proactively, or users may rely on some kind of warnings from browsers to guide their actions. In the first case, recognizing suspicious sites is mainly done by users. In the second case, browsers do their jobs of estimating how much suspicious a site may be and presenting the results to users. Of course, in both cases, the final decisions are made by users. But users are more proactive in the first case, looking for attack evidences themselves, and vice verse more reactive in the second case, waiting for hints from browsers before acting. Furthermore, with the later (the reactive) approach, browsers' warnings may be intrusive or non-intrusive. Although in general intrusive warnings are more effective at defending phishing attacks, they are also more annoyed to users because they disrupt users' normal operations. For that reason, non-intrusive warnings are still used, especially when the suspicious level falls below a threshold or when browsers need to present some positive events. Based on above observation, we shown in Figure 2 the classification of security visualization techniques used at client-side systems with related references.

4.1 Proactive approach

In this approach, users proactively recognize suspicious web sites themselves with the help from browsers. One effective technique used in this approach is personalization.

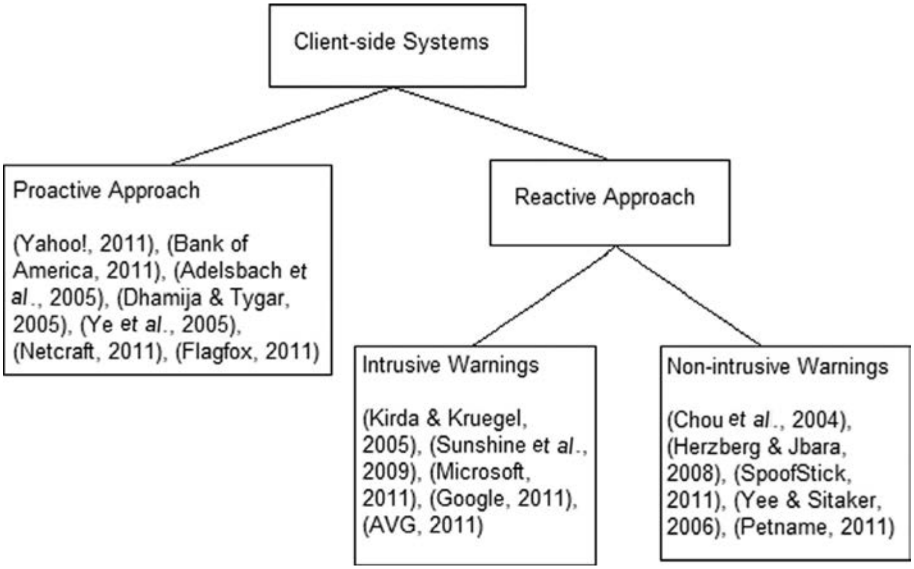

Client-side Systems
Proactive Approach
(Yahoo!, 2011), (Bank of America, 2011), (Adelsbach et al., 2005), (Dhamija & Tygar, 2005), (Ye et al., 2005), (Netcraft, 2011), (Flagfox, 2011)
Reactive Approach
Intrusive Warnings
(Kirda & Kruegel, 2005), (Sunshine et al., 2009), (Microsoft, 2011), (Google, 2011), (AVG, 2011)
Non-intrusive Warnings
(Chou et al., 2004), (Herzberg & Jbara, 2008), (SpoofStick, 2011), (Yee & Sitaker, 2006), (Petname, 2011)

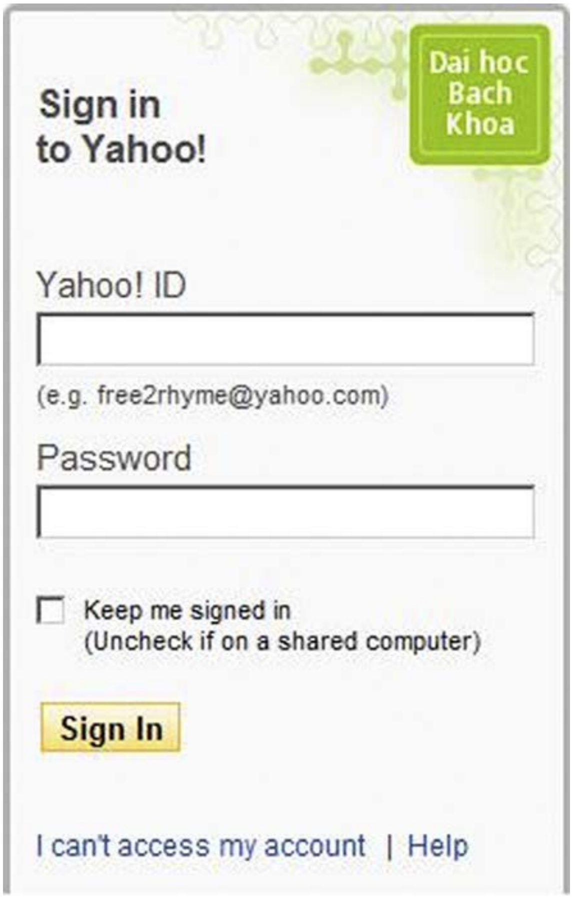
Figure 2.
Classification of security visualization techniques used at client-side systems

The use of personalization makes it difficult (or impossible) for attackers to fake legitimate content and visual presentation of authentic web sites. The personalization may be in the form of customized texts and/or customized images. The personalized items (texts and/or images) may be positioned in browsers' content area and/or at browsers' chrome (e.g. status bar, title bar, etc.). Finally, some personalization solutions are site-specific, while others are implemented at browser side. To prevent users from using default settings that are known to attackers, usually an initial random text or image is created, then users can choose another setting if they want.

Selected site-specific personalization solutions include Yahoo! Sign-in Seal (Yahoo!, 2011) for Yahoo! web site (Figure 3) and SiteKey for Bank of America (2011) web site. The login pages on these sites can be configured by users with customized texts and/or images. The customized texts and/or images are displayed directly in the content area, effectively captures users' attention. Yahoo! also supplies different color background patterns for users to choose from to make the customization more complex.

While site-specific personalization solutions use content area as the place for displaying personalized information, browser-based solutions often use browser chrome as the place for the customization. According to Adelsbach *et al.* (2005), browsers' secure connection indicators (one example is the padlock icon usually found on browser status bar indicating SSL protected web sites) is the only mean for users to get information about the security status of a connection, but these indicators can also be faked by visual spoofing attacks. The solution proposed in Adelsbach *et al.* (2005) is using personalization of browsers' chrome (Figure 4) to defend against visual spoofing attacks.

The Dynamic Security Skins proposed in Dhamija and Tygar (2005) is one example of browser-based personalization solutions. Dynamic Security Skins introduced "trusted window" which is the place where users can safely enter sensitive information (e.g. username and password) without worrying these information will be leaked to faked window. The trusted window is created by allowing users to set a customized



Dai hoc Bach Khoa

Sign in to Yahoo!

Yahoo! ID

(e.g. free2rhyme@yahoo.com)

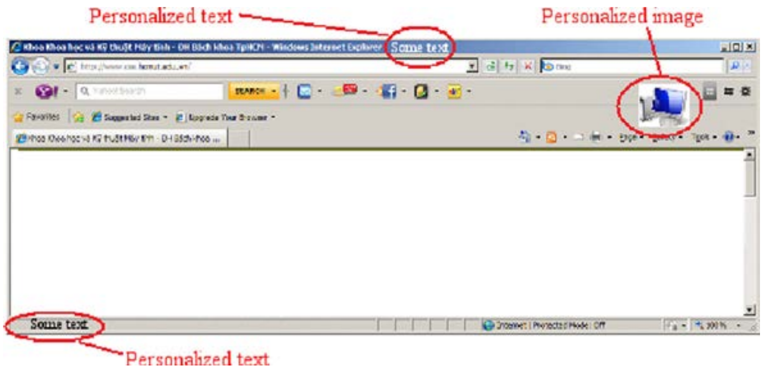
Password

□ Keep me signed in
(Uncheck if on a shared computer)

Sign In

I can't access my account | Help

Figure 3.
Yahoo! Sign-in Seal with a customized text on a user-selected background pattern



Note: The personalization can be in the form of text or image, and can be put at different locations such as title bar, status bar, tool bar, etc.

Figure 4.
Using personalization to defend visual spoofing attacks

background image (Figure 5). An initial background image is set randomly so that non-motivated users at least have some levels of protection. The second part of Dynamic Security Skins is the way it displays authenticated web sites to users. Instead of displaying a general padlock in the status bar as the way popular browsers do, Dynamic Security Skins displays a visual hash in the content area of authenticated web sites. This visual hash is compared by users with the pattern on the trusted window to see if they match (Figure 6). The visual hash displayed for authenticated

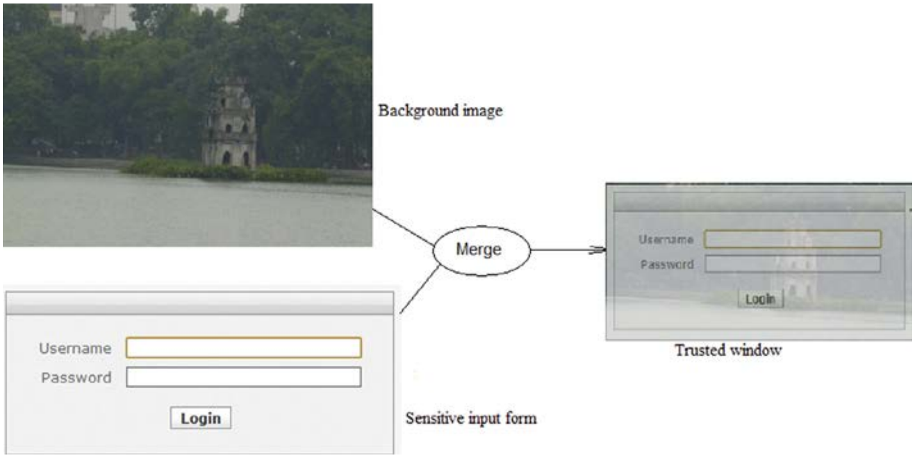
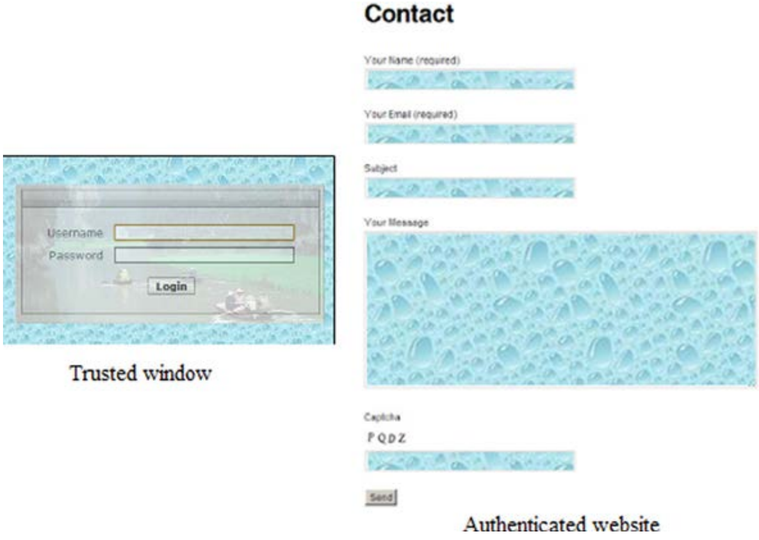


Figure 5.
Dynamic Security Skins trusted window is implemented by setting a customized background image behind a sensitive input form



Note: It is impossible for attackers to know the visual hashes to spoof authenticated web sites

Figure 6.
The visual hash on the trusted window matches with the visual hash displayed for an authenticated web site

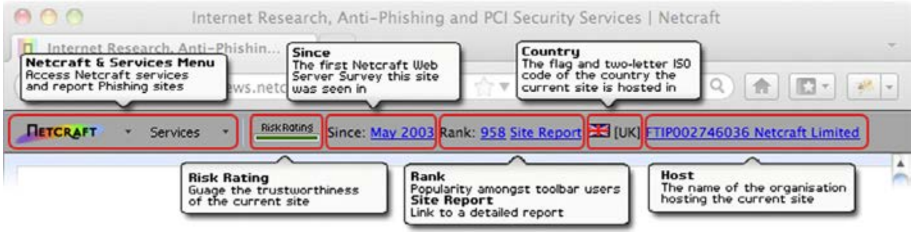
web sites cannot be spoofed because attackers do not know the customized patterns displayed on the trusted windows and because the trusted windows cannot be faked.

A work similar to Dynamic Security Skins is the synchronized random dynamic (SRD) boundary approach (Ye *et al.*, 2005). The SRD boundary approach's method of defending visual spoofing attacks is by establishing a trusted path between web browsers and users. Web browsers' jobs include not only displaying web site content but also showing security status information of the channels used to obtain content. The problem with current browser interface design is that there is no way for users to guarantee that the shown status information is actually from the browsers, e.g. malicious attackers can craft web pages that look like the whole browser windows with fake status information embedded in them to deceit users' vision. The SRD approach builds trust between browsers and users by adding a random dynamic element into its interface design. Because this element is random, it is not known by the servers and hence cannot be spoofed. Moreover, the SRD used randomness instead of customization to minimize users' works. Although users do not need to do much customization, they still need to compare the patterns displayed on different windows themselves to see if they match. For this reason, we put SRD in this proactive approach section.

Simpler proactive tools include the Netcraft toolbar (Netcraft, 2011) (Figure 7) and the Flagfox Firefox extension (Flagfox, 2011). These tools display the geographic locations of the hosting web servers, domains' whois information, web site traffic ranking, etc. so that users can check the authenticity of these web sites using the provided information themselves. Although these tools are simple and easy to use, they can become victims of visual spoofing attacks mentioned in Oppliger and Gajek (2005) and Adelsbach *et al.* (2005).

4.2 Reactive approach

In the reactive approach, the suspicious web sites are determined first by browsers with some probability values, then the results are presented to users in order for them to make final decisions. Certainly, the ways these results are presented are important. Ideally, these results should be displayed in a non-intrusive way, but still be able to grasp users' attention and communicate their meanings clearly to users. Unfortunately, these requirements are not always possible together, so usually there are trade-offs among these requirements. Depending on what are the main goals of displaying these results, appropriate methods are chosen. When security is considered as the main goal, an intrusive warning display proves to be more effective. But in reality, security is hardly the main goal of users (Whitten and Tygar, 1999). This reason leads to non-intrusive warning display approach, which does not disrupt users' normal operations. In term of interface design, intrusive warnings


Internet Research, Anti-Phishing and PCI Security Services | Netcraft
Internet Research, Anti-Phishin...
Netcraft & Services Menu
Access Netcraft services and report Phishing sites
Since
The first Netcraft Web Server Survey this site was seen in
Country
The flag and two-letter ISO code of the country the current site is hosted in
NETCRAFT
Services
Risk Rating
Since: May 2003
Rank: 958 Site Report
[UK]
FTIP002746036 Netcraft Limited
Risk Rating
Guage the trustworthiness of the current site
Rank
Popularity amongst toolbar users
Site Report
Link to a detailed report
Host
The name of the organisation hosting the current site

Source: Netcraft (2011)

Figure 7.
Netcraft anti-phishing toolbar shows information about web sites to help users decide the authenticity of the visited web sites themselves

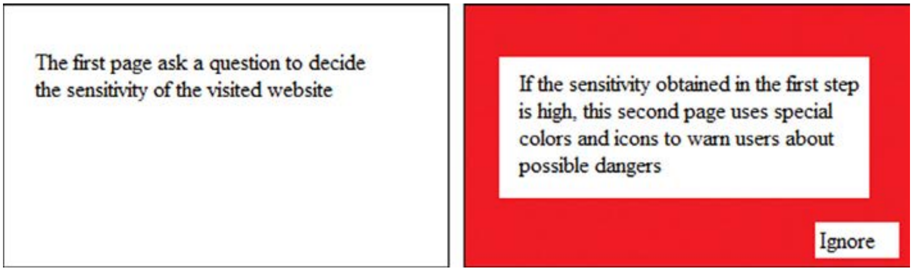
are usually displayed directly in the browsers' content area in large (or full) area, while non-intrusive warnings are usually displayed in the browsers' chrome area in small area. Some systems use both kinds of warning display and according to the level of estimated risk, an appropriate displaying method is selected by the browsers.

4.2.1 Intrusive warning. Intrusive warning design can be as simple as displaying just an informational modal message box as done in AntiPhish (Kirda and Kruegel, 2005). However, most of the tools are more complicated in their interface design with the purposes that do not only grasp users' attention (by deliberately disrupting users' current activities) but also explain clearly the possible risks involved with available decisions that can be made by users. Security tools achieve these goals by:

- presenting warning messages in the whole browsers' content area;
- combining colors and icons with descriptive text to communicate meanings quickly and effectively; and
- displaying different choices (usually in different font sizes, colors, and locations) for users to choose from.

The effectiveness of warning displays is studied in Sunshine *et al.* (2009) and the results of that work suggest the following three essential properties warning windows should possess to be effective: explain clearly the potential danger facing users; make it difficult for users to ignore; and ask a question users can answer. Based on this result, the paper proposed a warning design for web browsers. The design consists of multi-pages (Figure 8). The first page asks a question that users need to answer so that the browsers can decide the sensitivity of the visited web sites. The second page displays a severe warning message in case the sensitivity obtained in the previous step is high and users decide to continue. The severe warning page is designed with special colors and icons to make users scare enough to read the descriptive text carefully. If users are still determined to continue, they can click on a small "Ignore" link at the bottom right of the severe warning page. This "Ignore" link is put far away from the warning message center to prevent users from clicking on it by mistake.

The warning design principle proposed in Sunshine *et al.* (2009) is actually implemented in several popular applications, including the SmartScreen Filter in Internet Explorer 9 (Microsoft, 2011; Figure 9), Google Safe Browsing API (Google, 2011), and AVG security toolbar (AVG, 2011; Figure 10) with slightly differences among them.

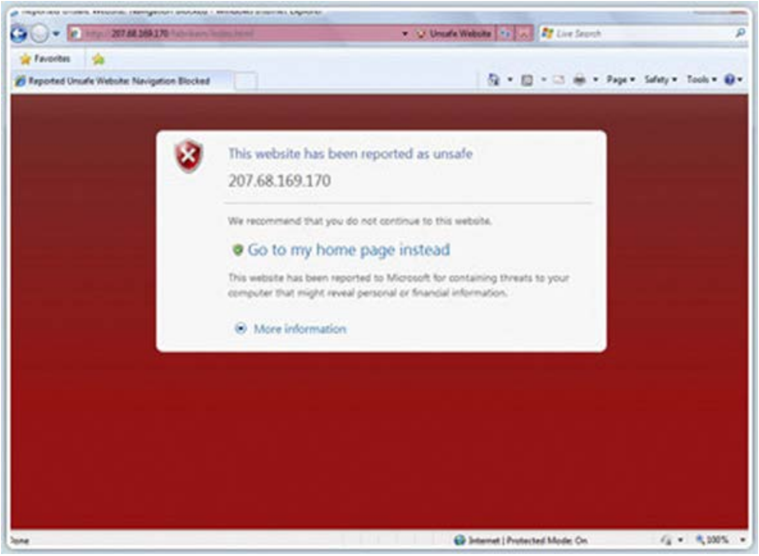


The first page ask a question to decide the sensitivity of the visited website

If the sensitivity obtained in the first step is high, this second page uses special colors and icons to warn users about possible dangers

Figure 8.
Multi-page warning design

Notes: The first page is used to obtain the sensitivity of the visited web site; the second page uses special warning colors and icons to get users' attention; users can still continue their risky action but with more effort in locating the small "Ignore" link far from the center



This website has been reported as unsafe

207.68.169.170

We recommend that you do not continue to this website.

Go to my home page instead

This website has been reported to Microsoft for containing threats to your computer that might reveal personal or financial information.

More information

Unsafe Website
Reported Unsafe Website: Navigation Blocked
Internet | Protected Mode: On

Source: Microsoft (2011)

Figure 9.
SmartScreen filter in Internet Explorer 9

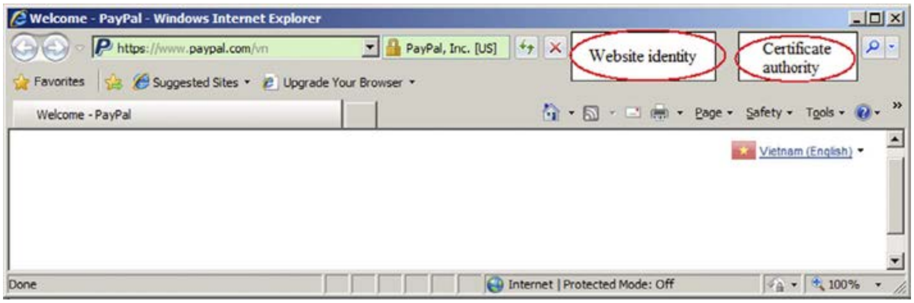


Figure 10.
TrustBar displays a web site identity and its respected certificate authority information in text or image form at the top of web browsers

4.2.2 Non-intrusive warning. Non-intrusive warnings are warnings that do not break normal operations of users. As a result, these warnings are often displayed in small area at some places out of the content area. The word "warning" here is used not only for negative meaning (suggesting there are some risks involved) but also for positive meaning (suggesting there are some trusts involved), as opposed to intrusive warning which is only used for negative meaning. This is natural, because intrusive warning displays are costly in term of users' productivity and hence should only be used in high severity cases.

The study in Egelman *et al.* (2008) pointed out that there is no significant difference between (negative) non-intrusive warnings and no warning at all. Because of this result, our survey only focuses on positive non-intrusive warning tools (from now on, we will just write positive non-intrusive warning as non-intrusive warning to make the text simple and short).

The most common type of displaying non-intrusive warnings is by changing some elements on the browser chrome with new graphical styles. For example, popular browsers like Internet Explorer, Mozilla Firefox, and Google Chrome display a padlock icon on the status bar and change the address bar background color when users visit an SSL protected web site. A similar approach is implemented in SpoofGuard toolbar (Chou *et al.*, 2004). It uses familiar traffic colors (green, yellow, red) to denote the different suspicious levels of visited web sites. TrustBar (Herzberg and Jbara, 2008) is another browser toolbar that employs the same technique but uses highly visible text and image to display the identity of the visited web sites and their respected certificate authority information (Figure 11). For sites that are not protected by SSL/TLS, TrustBar also displays a warning message about the situation so that users are aware of it.

Similar to TrustBar is SpoofStick (2011) browser extension for Mozilla Firefox and Internet Explorer (Figure 12). SpoofStick displays the most relevant domain information of the visited web sites in clear form so that users can spot suspected issues quickly. One advantage of SpoofStick is that it does not require web sites to be protected by SSL. Furthermore, users can customize the size and color of SpoofStick display to defend spoofing of SpoofStick itself.

In addition to having user interface elements that appear differently between SSL and non-SSL sites as popular browsers do, the Passpet (Yee and Sitaker, 2006) and Petname (2011) tools introduced a new concept called "petname" (or "label"). Simply talk, petnames are local names users assign to visited web sites so that they are easier for them to remember. Petnames describe the relations from users to web sites and are displayed when users visit these assigned web sites in the future. The petnames are only visible to users and their browsers, so it is difficult for attackers to know and spoof them. While the petnames are configured and recognized by users, SSL and

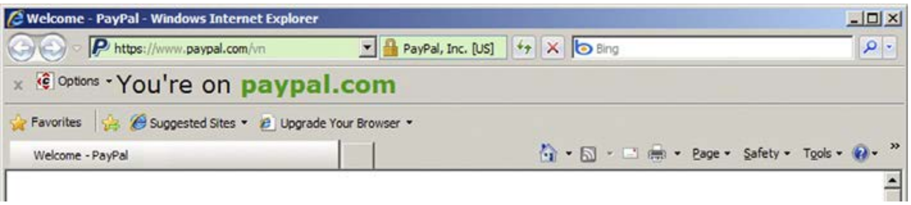
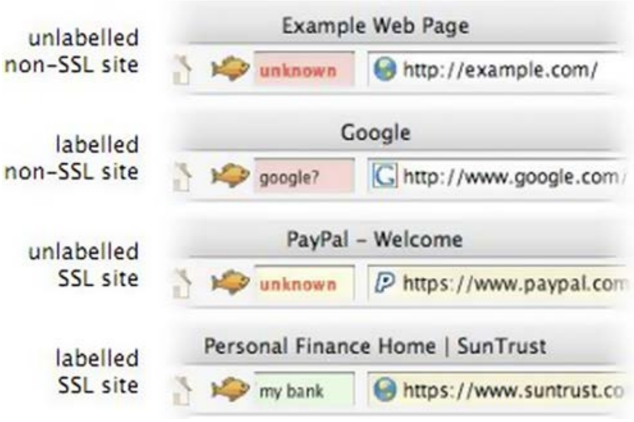


Figure 11.
SpoofStick browser extension on Internet Explorer



Source: Yee and Sitaker (2006)

Figure 12.
Passpet tool with unlabelled vs labeled, and non-SSL vs SSL sites

non-SSL statuses are recognized and displayed by web browsers, so these tools use both proactive approach and reactive approach in their design. A screenshot of the Passpet tool is in Figure 13.

5. Security visualization for server-side systems

While security visualization techniques used at client-side tend to be simple and easy to use, even for ordinary people, security visualization techniques used at server-side are more complex, powerful, and require at least some level of technical knowledge from their users. This is not unusual, because the primary users of server-side tools are administrators who are more motivated to learn about their tools for their jobs. The classification of security visualization techniques/tools used at server-side systems may base on:

- The main goals of the tools: management, monitoring, analysis, intrusion detection, etc.
- The visualization algorithms used: pixel drawing, chart, graph, 3D, etc.
- The data source types used for the visualizations: packet, flow, logging records, etc.

Each of the classification has its own advantages. In this survey, we use data source type to classify visualization techniques because of the following reasons:

- Each server-side system has only a limited data source types available. Knowing what techniques are suitable to what data source types will save administrators' time looking for possible solutions.
- When administrators want to do something but it is difficult (or impossible) to do so with available data source types, they will know what data source types to collect for the required tasks.
- The abundance of different data source types that may come in the future promises interesting challenges and opportunities for security visualization researchers to come up with novel techniques and applications.

Instead of using individual data source type (there are many of them) for the classification, we group the data source types with a common abstraction level together

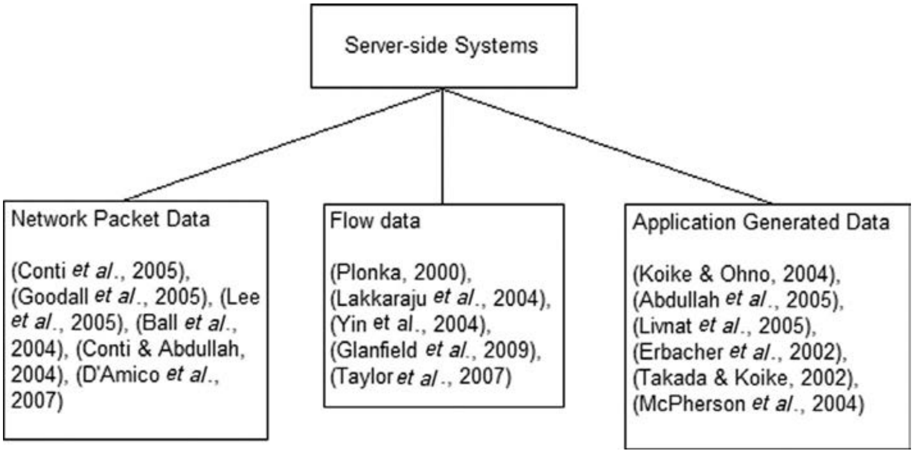

Server-side Systems
Network Packet Data
(Conti *et al*., 2005), (Goodall *et al*., 2005), (Lee *et al*., 2005), (Ball *et al*., 2004), (Conti & Abdullah, 2004), (D'Amico *et al*., 2007)
Flow data
(Plonka, 2000), (Lakkaraju *et al*., 2004), (Yin et al., 2004), (Glanfield *et al*., 2009), (Taylor *et al* ., 2007)
Application Generated Data
(Koike & Ohno, 2004), (Abdullah *et al*., 2005), (Livnat *et al*., 2005), (Erbacher *et al*., 2002), (Takada & Koike, 2002), (McPherson *et al*., 2004)

Figure 13.
Classification of security visualization techniques used at server-side systems

to keep the classification clear and short. This grouping does not lose the main theme of the classification because data types at the same level of abstraction usually contain the same informational structure. The data source levels we identify include (from lowest to highest level of abstraction): network packet, network flow, and application generated data. Hence, the classification of security visualization techniques used at server-side systems and related references are shown in Figure 14.

5.1 Network packet visualization

Visualization techniques that use network packet as data source can zoom into the most detailed level of individual packets. But examining individual packets is rarely useful, not to mention the overwhelming number of them. So, tools often present an overview of data, show the relationships between them and allow users to zoom into interesting sections to see more detailed data. This interaction process is summarized as the information visualization mantra (Shneiderman, 1996): "Overview first, zoom and filter, then details on demand." One particular example tool using this design principle is Rumint (Conti *et al.*, 2005). It uses binary rainfall visualization to display the details of packets captured (Figure 15). This visualization helps administrators to see an overview of thousands of packets at a time (overview). To support the data analysis task of administrators, Rumint provides different modes of display such as 24 bits/pixel, 8 bits/pixel, ASCII, HEX, etc. The textual display modes (e.g. ASCII, HEX) use more spaces when compared to graphical display modes, but they are more useful when administrators want to view bit and byte values of individual network packets (details on demand).

Zooming in and out back and forth between different zooming levels is needed for complicated analysis tasks, but doing so can make administrators losing the overall goal of their works. For administrators to be able to concentrate on detailed tasks, but at the same time to have glues of the whole works, some tools use a

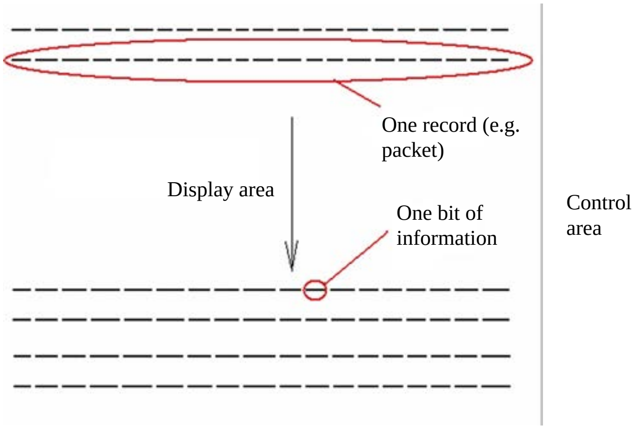
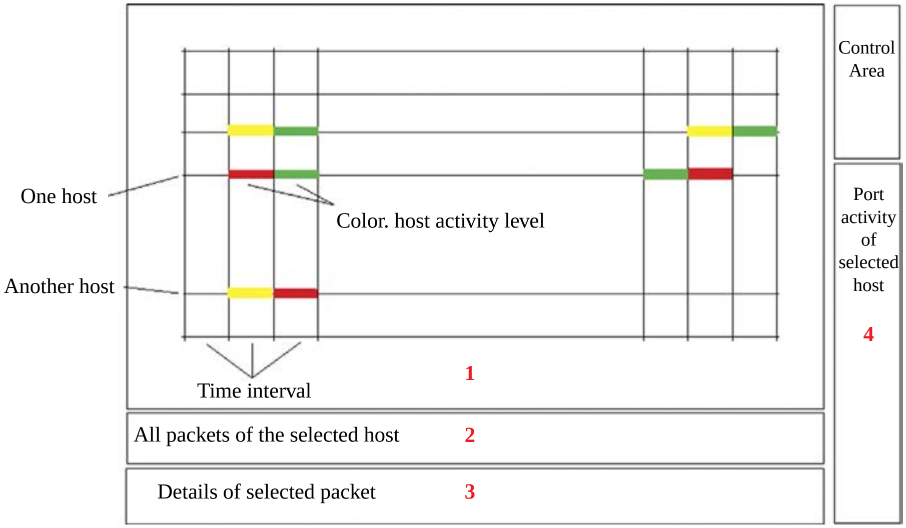


Figure 14.
Rumint tool with binary rainfall visualization of network packets

Notes: Each packet is plotted in a row, and each individual bit (or byte, or word, etc. depends on the resolution) of it is plotted at particular column; the control area contains buttons to zoom, change display mode, etc.


Control Area
Port activity of selected host
4
One host
Color. host activity level
Another host
Time interval
1
All packets of the selected host
2
Details of selected packet
3

Note: In this design, area 1 provides the context and areas 2-4 provide the focuses

Figure 15. Visualization design of TNV: area 1 is the main display, areas 2-4 are detailed displays with different zoom levels and different view perspectives

"Focus + Context" approach. In this approach, the main working area is clearly marked by its large, clear, and central nature (usually is the focus); and is surrounded by supporting areas providing complimentary information (usually is the context). The advantages of this approach are there are less clutter and more useful information displayed, plus the cognitive processing of users is reduced by minimizing the switching between different screens. TNV (Goodall *et al.*, 2005) and Visual Firewall (Lee *et al.*, 2005) are two such tools that use the "Focus + Context" approach.

TVN is mainly a traffic and attack analysis tool. The central and largest area of TNV displays a matrix where time intervals are mapped to columns and hosts are mapped to rows (Figure 16). Each row represents one host and has a unique IP address. The hosts are sorted by their IP addresses with local hosts are emphasized because of their higher importance. In this matrix, an intersection between a column and a row represents the activity of a host in a particular time interval. The amount of activity is denoted by colors. The surrounding areas enable filtering (by protocol, port, etc.), viewing activity details (ports used, statistics, packet details, etc.) without the need of switching to other screens.

The "Focus + Context" approach is implemented in Visual Firewall by using four different views: real time traffic view, visual signature view, statistics view, and IDS alarm view (except the IDS alarm view, which uses Snort alerts for the visualization, other views use network packets as the data source). Administrators can select any one of the four views to put it into focus, the other remaining three views are then located in the sidebar at the right, supporting additional perspectives. Figure 17 shows Visual Firewall design. Unlike TNV, Visual Firewall uses the largest area for focus and the other areas for context.

The arrangement and positioning of elements in visualization techniques play an important role in deciding the scalability of techniques. With binary rainfall, Rumint (Conti *et al.*, 2005) can display thousands of packets in one screen. The matrix method

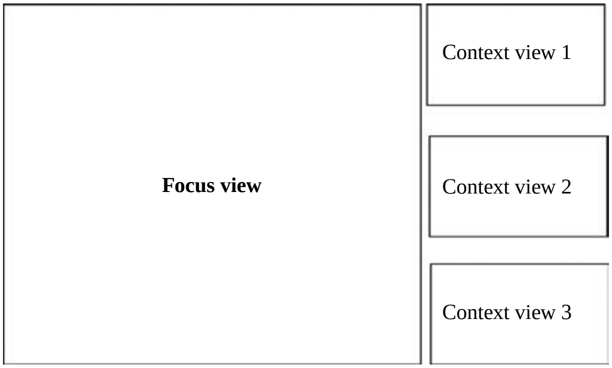


Figure 16.
Visual Firewall provide four different views

Notes: At a particular time, one view is selected by users as the focused view (largest area) and the other three as context views; unlike TNV, Visual Firewall provides only one focused view and many context views

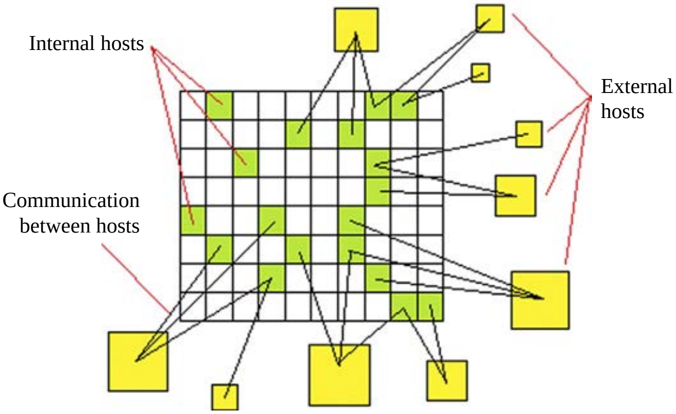


Figure 17.
Arrangement and positioning of internal and external hosts in VISUAL: internal hosts are represented as cells in a matrix while external hosts surround them

Notes: Straight lines connecting internal and external hosts denote communication between them; sizes of external hosts are in proportion to their communication levels

used in TNV (Goodall *et al.*, 2005) can display activities of hundreds of hosts in tens of time intervals. VISUAL (Ball *et al.*, 2004) also uses matrix method for presenting internal hosts (while external hosts are located around the matrix; Figure 18). The communication between internal hosts and external ones are depicted by straight lines connecting them. Sizes of external hosts are proportional to their activity levels. Finally, to show the correlation between many variables, tools often use parallel coordinate visualization, for example: visual fingerprinting of attack tools (Conti and Abdullah, 2004), Visual Firewall (Lee *et al.*, 2005), and VIAssist (D'Amico *et al.*, 2007).

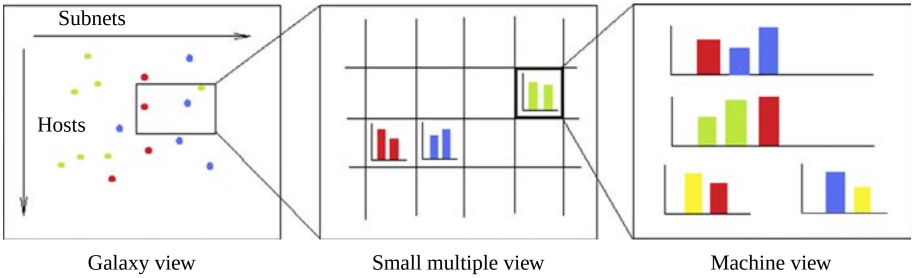
5.2 Flow visualization

Flow records are aggregates of related packets in a unidirectional transaction between two machines. Because of this nature, flow allows administrators doing security and traffic analysis at a higher level than they can do so with network packet. The NetFlow format (Cisco Systems, 2004) includes many communication attributes, among which are source port/IP address, destination port/IP address, protocol, volume of traffic, etc. Visualization techniques using flow as their data source can provide summarized information of transactions between hosts easily. For example, FlowScan (Plonka, 2000) uses NetFlow traffic data for its visual reporting function. It can display traffic reports of network activities happened in short periods as well as rather long periods.

Flow visualization techniques are usually applied on large networks, so the information visualization mantra (Shneiderman, 1996) is often implemented in these techniques. NVisionIP (Lakkaraju *et al.*, 2004) offers three levels of zoom to support viewing from overview to details, namely: galaxy view, small multiple view, and machine view (Figure 19). In galaxy view, a whole class B network is shown. The machines in the network are positioned into a subnet – host coordinate. When a small area of galaxy view is selected, that area is displayed in small multiple view which is more detailed (additional charts are added). The most detailed level is machine view, which is opened when a particular host is selected in small multiple view. The information displayed in machine view includes: bytes and flows for all protocols, for all ports, or for particular protocols like TCP, UDP, etc.

Similar to NVisionIP interaction design, VisFlowConnect (Yin *et al.*, 2004) provides different levels of view for flow visualization: global view (showing interactions between local machines and external domains), domain view (showing interactions between local machines and individual machines from a particular domain), internal view (showing interactions between local machines with each other), and host statistics view (detailed information about the flow traffic for a selected local machine). VisFlowConnect uses parallel coordinate visualization to show the interactions between communication components (hosts and domains).

OverFlow (Glanfield *et al.*, 2009), on the other hand, concentrates on displaying overview level of zoom only, while allows adding of plug-in tools developed by third parties to its framework to provide additional detailed views. This extensible design makes it flexible for administrators in choosing appropriate details viewing tools for their jobs. The main view of OverFlow displays communication between network



Note: Users can go from one level of zoom to the next level by selecting an appropriate area/machine

Figure 18.
Three levels of zoom, from the most brief to the most detailed, supported by NVisionIP: galaxy view, small multiple view, and machine view

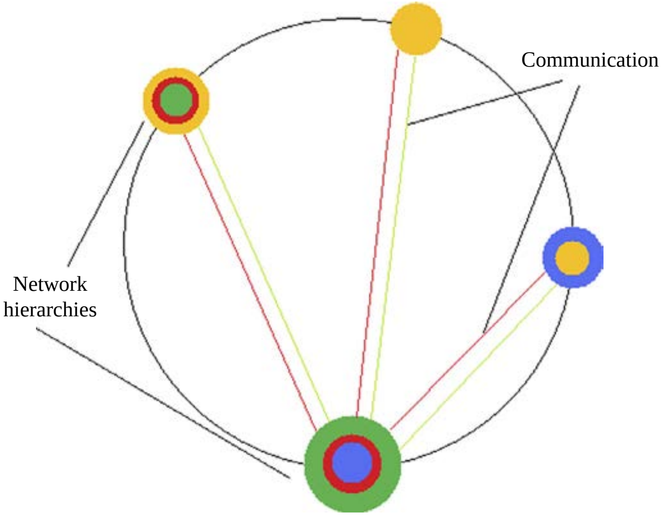


Figure 19.
OverFlow main view: concentric circles present network hierarchies and lines present communication between them

Note: Detailed views are added to the visualization framework by third party developers

hierarchies (Figure 20). The internal structure of hierarchies is presented using Treemap (Johnson and Shneiderman, 1991).

3D visualization is rarely used in security because it makes comparing between items difficult, and some elements may be covered by others. NetBytes (Taylor *et al.*, 2007) is one rare tool that provides 3D visualization of NetFlow data (Figure 21).

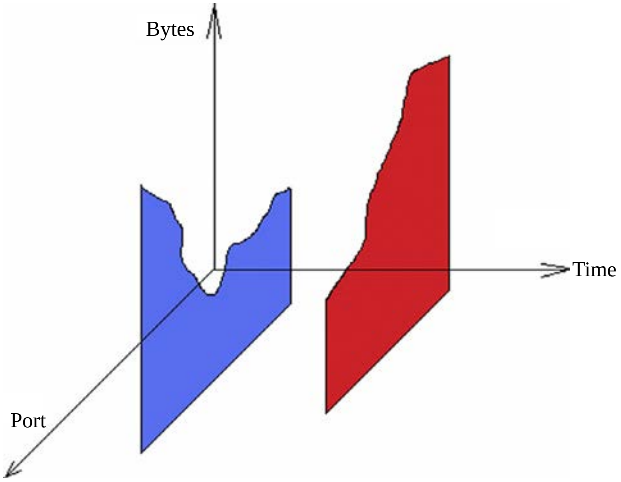
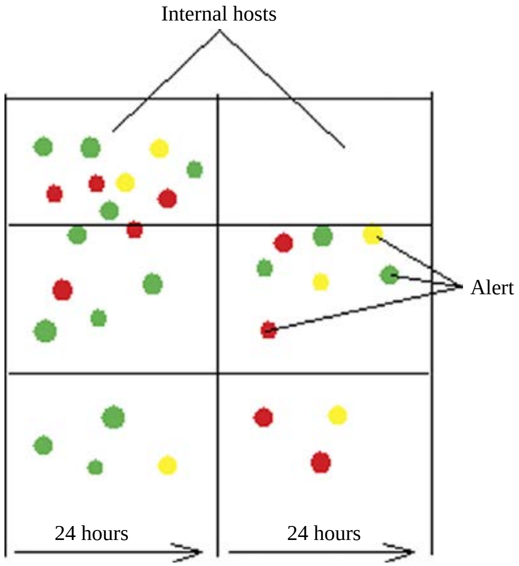


Figure 20.
NetBytes visualizes NetFlow data in a 3D coordinate with time, port, and bytes axes

Note: To avoid concealment, users interaction mechanisms such as rotation, projection, etc. are provided



Note: The severity of each alert is encoded by its color

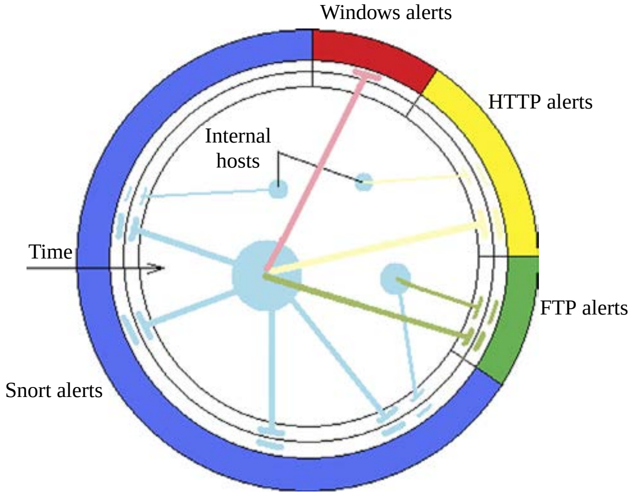
Figure 21. IDS RainStorm visualization design: internal hosts are divided into vertical rectangles within which the horizontal direction presents 24-hour period and alerts are plotted as pixels

To compensate above mentioned problems with 3D visualization, NetBytes provides extensive interaction functions: rotation, projection, etc.

5.3 Application generated data visualization

Application generated data is the data source type with the highest level of abstraction. These generated data have specific meanings in their respective domains: web server logs record HTTP access events, IDS alert messages describe potential intrusions, operating system logs help us to know who accessed to the system and when these actions happened, and so on. Visualization tools using application generated data as the data source have the specific advantage of presenting relevant information clearly. However, to go deeper to see more detailed information, other data sources like network packets or flows are needed.

The most popular type of data sources used at this level is IDS alert. By visualizing IDS messages, administrators can see an overview of the security situation of their networks. Another benefit of IDS alerts visualization is that administrators can quickly check the correlations between these alerts, partly deduce the validity of these large amount of generated alerts, from which many are false positive. SnortView (Koike and Ohno, 2004) uses matrix visualization to display real time alerts for the whole network. Based on the visualization and some heuristics, SnortView can help administrators to detect false alerts quickly. IDS RainStorm (Abdullah *et al.*, 2005) divides the range of internal IP addresses into vertical rectangles. Within each rectangle, the horizontal lines represent the 24-hour time period. IDS RainStorm uses alerts collected by StealthWatch IDS (Lancope, 2011) as the data source and plots them as pixels on the computer screen (Figure 22). IDS RainStorm has ability to zoom in at a particular position (IP range, time period).



Notes: The last time period is mapped to the inner most ring; internal hosts are put inside the rings; connecting lines between hosts and sectors are used to present the types of alert generated for internal hosts; many alerts of the same type appear on a host are grouped together into a wider line

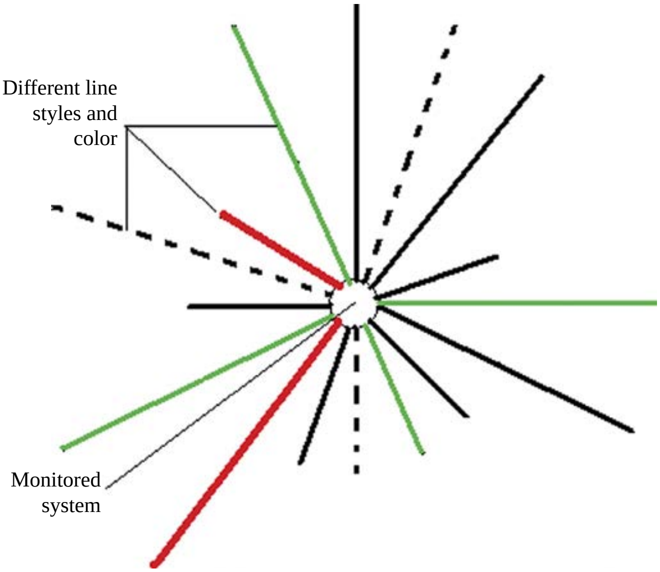
Figure 22.
VisAlert maps different IDS types into separated sectors of concentric rings

The visualization of VisAlert (Livnat *et al.*, 2005) uses not only IDS alert messages but also log records generated by other systems: web server, operating system, file server, etc. as its data source. These additional data sources can greatly improve the correlation analysis task of administrators. Concentric circles are used to present consecutive time intervals. Alerts are presented by straight lines connecting alert generating sources and destination hosts (Figure 23). A host with many connecting lines is possibly under some real attacks and is needed to be reviewed carefully by administrators.

A visualization design similar to VisAlert is described in Erbacher *et al.* (2002). It displays collection of connections to a monitored system located at the center of the visualization screen. Each connection type has a different style and high critical connection types (e.g. port scan) have more attentive styles. The locations of connection lines' endpoints are determined by how far is estimated between the monitored system and the other connecting hosts, based on their IP addresses (Figure 24).

Instead of using IDS logs, Tudumi (Takada and Koike, 2002) uses different audit logs to present security information visually. These logs include access log, log-in log, and user substitution log. Before visualization step, a summarize step is applied on the data. This step is used to reduce the huge amount of generated log messages.

While most of the tools mentioned above require a significant amount of data for their effective visualization works, PortVis (McPherson *et al.*, 2004) needs only aggregate data for a particular port with a particular protocol and in a specific of time. The aggregate information includes session counts, unique source addresses, unique destination addresses, and unique source countries. There are some cases when these



Note: Distances between monitored system and other hosts are estimated using their IP addresses

Figure 23.
Visualization of connections to a monitored system: different line styles and colors present different characteristics and types of connections

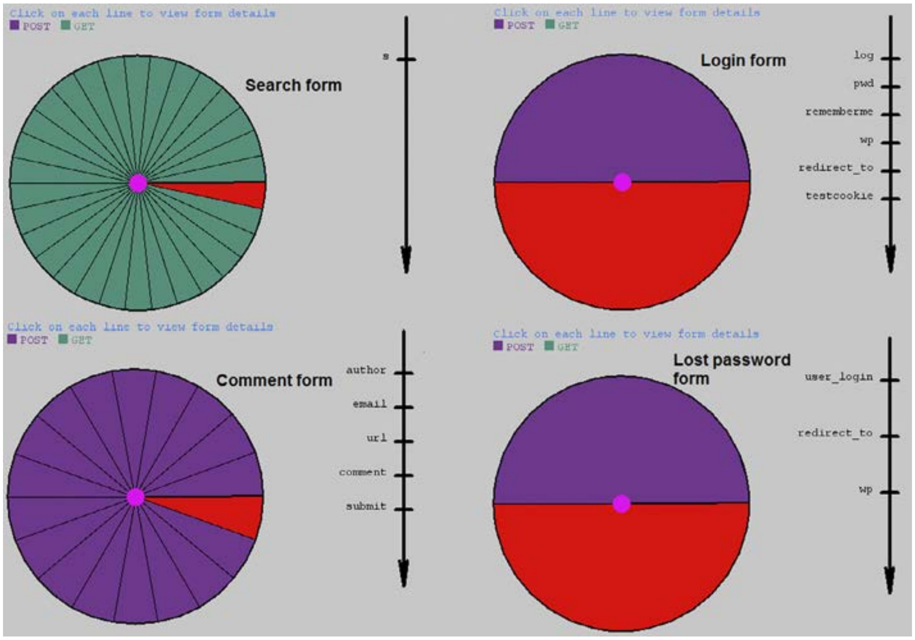


Figure 24.
Visualization of HTML form structure: each form has a target and a source (the web page the form is in), contains several required and optional input controls, and has explicit and implicit constraints imposed on these controls' possible values

restrictions are valid, for example, when there is a need to keep the detailed information secret, or when the privacy of individual accesses need to be protected. Nevertheless, PortVis can help administrators to recognize many type of malicious activities.

6. Security visualization for web applications

As mentioned in the classification of Section 3, there is currently a lack of extensive research about security visualization specifically for web applications. The web application side is also a special one because there are at least three main user types of it. The needs and skills of these types of user are different from each other:

(1) Web developers need to see critical points in their source code, dependencies between different modules, etc.

(2) System administrators need to know usage information of installed web applications, consequences of different configuration values, etc.

(3) Web application end-users need to see where their submitted data is sent to, what browsers' components or operating systems' components are executed when they access some applications, etc.

One of the rare security visualization solution designed specifically for web applications is the monitoring tool described in Dang and Dang (2011). The visualization design of it consists of three levels of abstraction on user submitted data and HTML web form elements. The highest level displays a collection of HTML forms and user accesses through them; the second level displays structural information of selected forms and structural information of user accesses; the lowest level displays individual selected controls, their constraints and their respective users input values. This visualization tool is designed with the goal of helping administrators to quickly recognize possible attacks and drill down for more detailed information so that they can check the actual issues. A screenshot of the tool is in Figure 24.

Selected researches similar to security visualization for web applications that may provide additional related information include web application security (Scott and Sharp, 2002; Huang *et al.*, 2003; Kruegel and Vigna, 2003; Kals *et al.*, 2006) and web information visualization (Munzner and Burchard, 1995; Weinreich and Lamersdorf, 2000; Dachselt and Ebert, 2001; Chung *et al.*, 2005; Ortega and Aguillo, 2008).

7. Conclusion and future research directions

This survey is written to inform readers about the state-of-the-art research about web information systems security based on information visualization techniques. Because visualization outputs are used directly by human users, users study is an essential part of this research area. The complexity involved in user study is caused by the differences of types of users with different levels of knowledge, skills, age, interests, etc. The users at client-side systems need simple and intuitive solutions because security is not their main goals when surfing the web. At the server-side, security administrators need powerful tools that can provide management, monitoring, recognition, and analysis capability for them. The huge amount of data generated at the server-side requires novel methods of presenting them effectively and efficiently on a limited 2D screen. The mixing of user types at web application side make it the special one. Although web application is an essential part of any web information systems, there is not much researches about security visualization methods for this component. When this gap is filled by future researches, the whole picture of security visualization for web information systems will be more complete.

The requirements of new visualization techniques for different kinds of tasks poses another challenging problem. On one hand, there is a need for generic tools that are suitable for a varieties of tasks, such as management, monitoring, analysis, simulation, etc. But on the other hand, there is also a need for specific tools for particular tasks. Until there is a standard of tasks defined and the specific requirements for these tasks are set up, the integrating of these tools is very difficult.

The measurement of security visualization solutions is another hard problem. As pointed out in Plaisant (2004), security visualization, or information visualization in general, needs to have different measurement techniques to make the adoption of major users. Making these visualization measurements meaningful for both academic community and ordinary users will require new experiment settings and evaluation techniques.

References

Abdullah, K., Lee, C., Conti, G., Copeland, J.A. and Stasko, J. (2005), "IDS rainStorm: visualizing IDS alarms", *IEEE Workshops on Visualization for Computer Security, Minneapolis, MN, October*, pp. 1-10.

Adelsbach, A., Gajek, S. and Schwenk, J. (2005), "Visual spoofing of SSL protected web sites and effective countermeasures", in Deng, R., Bao, F., Pang, H. and Zhou, J. (Eds), *Information Security Practice and Experience*, Springer, Berlin, pp. 204-17.

AVG (2011), "AVG security toolbar help", available at: www.avg.com/ww-en/help-avg-toolbar (accessed 26 July 2011).

Ball, R., Fink, G.A. and North, C. (2004), "Home-centric visualization of network traffic for security administration", *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, ACM, New York, NY, pp. 55-64.

Bank of America (2011), "SiteKey", available at: www.bankofamerica.com/privacy/index.cfm?template=sitekey (accessed 26 July 2011).

Card, S.K., Mackinlay, J.D. and Shneiderman, B. (1999), *Readings in Information Visualization: Using Vision to Think*, Morgan Kaufmann, San Francisco, CA.

Chou, N., Ledesma, R., Teraguchi, Y. and Mitchell, J. (2004), "Client-side defense against web-based identity theft", *Proceedings of the 11th Annual Network and Distributed Systems Security Symposium, San Diego, CA, February*, The Internet Society 2004.

Chung, W., Chen, H. and Nunamaker, J. Jr (2005), "A visual framework for knowledge discovery on the web: an empirical study of business intelligence exploration", *Journal of Management Information Systems*, Vol. 21 No. 4, pp. 57-84.

Cisco Systems (2004), "Cisco Systems NetFlow services export version 9", available at: www.ietf.org/rfc/rfc3954.txt (accessed 26 July 2011).

Conti, G. and Abdullah, K. (2004), "Passive visual fingerprinting of network attack tools", *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, ACM, New York, NY, pp. 45-54.

Conti, G., Grizzard, J., Ahamad, M. and Owen, H. (2005), "Visual exploration of malicious network objects using semantic zoom, interactive encoding and dynamic queries", *IEEE Workshops on Visualization for Computer Security, Minneapolis, MN, October*, IEEE Computer Society Press, Washington, DC, pp. 83-90.

Dachselt, R. and Ebert, J. (2001), "Collapsible cylindrical trees: a fast hierarchical navigation technique", *Proceedings of the IEEE Symposium on Information Visualization 2001*, IEEE Computer Society Press, San Francisco, CA, pp. 79-86.

D'Amico, A.D., Goodall, J.R., Tesone, D.R. and Kopylec, J.K. (2007), "Visual discovery in computer network defense", *IEEE Computer Graphics and Applications*, Vol. 27 No. 5, pp. 20-7.

Dang, T.T. and Dang, T.K. (2011), "A visual model for web applications security monitoring", *Proceedings of the 2011 IEEE International Conference on Information Security and Intelligence Control, Jilin, August*, IEEE Computer Society Press, Washington, DC, pp. 158-62.

Dhamija, R. and Tygar, J.D. (2005), "The battle against phishing: Dynamic Security Skins", *Proceedings of the 2005 Symposium on Usable Privacy and Security*, ACM, New York, NY, pp. 77-88.

Dhamija, R., Tygar, J.D. and Hearst, M. (2006), "Why phishing works", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems 2006*, ACM, New York, NY, pp. 581-90.

Egelman, S., Cranor, L.F. and Hong, J. (2008), "You've been warned: an empirical study of the effectiveness of web browser phishing warnings", *Proceeding of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, pp. 1065-74.

Erbacher, R.F., Walker, K.L. and Frincke, D.A. (2002), "Intrusion and misuse detection in large-scale systems", *IEEE Computer Graphics and Applications*, Vol. 22 No. 1, pp. 38-48.

Flagfox (2011), "Flagfox 4.1.4", available at: https://addons.mozilla.org/en-US/firefox/addon/flagfox/ (accessed 26 July 2011).

Glanfield, J., Brooks, S., Taylor, T., Paterson, D., Smith, C., Gates, C. and McHugh, J. (2009), "OverFlow: an overview visualization for network analysis", *Proceedings of the 6th International Workshop on Visualization for Cyber Security, Atlantic City, NJ*, IEEE Computer Society Press, Washington, DC, pp. 11-19.

Goodall, J.R., Lutters, W.G., Rheingans, P. and Komlodi, A. (2005), "Preserving the big picture: visual network traffic analysis with TNV", *IEEE Workshops on Visualization for Computer Security, Minneapolis, MN, October*, IEEE Computer Society Press, Washington, DC, pp. 47-54.

Google (2011), "Google safe browsing API", available at: http://code.google.com/apis/safebrowsing/ (accessed 26 July 2011).

Herzberg, A. and Jbara, A. (2008), "Security and identification indicators for browsers against spoofing and phishing attacks", *ACM Transactions on Internet Technology*, Vol. 8 No. 4, 36 pp.

Huang, Y.-W., Huang, S.-K., Lin, T.-P. and Tsai, C.-H. (2003), "Web application security assessment by fault injection and behavior monitoring", *Proceedings of the 12th International Conference on World Wide Web*, ACM, New York, NY, pp. 148-59.

Iwata, M., Arase, Y., Hara, T. and Nishio, S. (2010), "Web browser for children using bubble metaphor", *International Journal of Web Information Systems*, Vol. 6 No. 1, pp. 55-73.

Jakobsson, M. (2007), "The human factor in phishing", *Privacy & Security of Consumer Information 2007*, available at: www.informatics.indiana.edu/markus/papers/aci.pdf (accessed 26 July 2011).

Johnson, B. and Shneiderman, B. (1991), "Tree-maps: a space-filling approach to the visualization of hierarchical information structures", *Proceedings of the 2nd Conference on Visualization, Los Alamitos, CA*, IEEE Computer Society Press, Washington, DC, pp. 284-91.

Kals, S., Kirda, E., Kruegel, C. and Jovanovic, N. (2006), "SecuBat: a web vulnerability scanner", *Proceedings of the 15th International Conference on World Wide Web*, ACM, New York, NY, pp. 247-56.

Kirda, E. and Kruegel, C. (2005), "Protecting users against phishing attacks with AntiPhish", *29th Annual International Computer Software and Applications Conference 2005*, IEEE Computer Society Press, Washington, DC, pp. 517-24.

Koike, H. and Ohno, K. (2004), "SnortView: visualization system of Snort logs", *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, ACM, New York, NY, pp. 143-7.

Kruegel, C. and Vigna, G. (2003), "Anomaly detection of web-based attacks", *Proceedings of the 10th ACM Conference on Computer and Communications Security*, ACM, New York, NY, pp. 251-61.

Lakkaraju, K., Yurcik, W. and Lee, A.J. (2004), "NVisionIP: NetFlow visualizations of system state for security situational awareness", *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, ACM, New York, NY, pp. 65-72.

Lancope (2011), "StealthWatch products", available at: www.lancope.com/products/ (accessed 26 July 2011).

Lee, C.P., Trost, J., Gibbs, N., Beyah, R. and Copeland, J.A. (2005), "Visual Firewall: real-time network security monitor", *IEEE Workshops on Visualization for Computer Security, Minneapolis, MN, October*, IEEE Computer Society Press, Washington, DC, pp. 129-36.

Livnat, Y., Agutter, J., Moon, S., Erbacher, R.F. and Foresti, S. (2005), "A visualization paradigm for network intrusion detection", *Proceedings of the Sixth Annual IEEE SMC Information Assurance Workshop, June*, IEEE Computer Society Press, Washington, DC, pp. 92-9.

McPherson, J., Ma, K.-L., Krystosk, P., Bartoletti, T. and Christensen, M. (2004), "PortVis: a tool for port-based detection of security events", *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, ACM, New York, NY, pp. 73-81.

Marty, R. (2008), *Applied Security Visualization*, Addison-Wesley Professional, Boston, MA.

Microsoft (2011), "SmartScreen filter", available at: http://windows.microsoft.com/en-US/internet-explorer/products/ie-9/features/smartscreen-filter (accessed 26 July 2011).

Munzner, T. and Burchard, P. (1995), "Visualizing the structure of the world wide web in 3D hyperbolic space", *Proceedings of the 1st Symposium on Virtual Reality Modeling Language*, ACM, New York, NY, pp. 33-8.

Netcraft (2011), "Netcraft anti-phishing toolbar", available at: http://toolbar.netcraft.com/ (accessed 26 July 2011).

Oppliger, R. and Gajek, S. (2005), "Effective protection against phishing and web spoofing", in Dittmann, J., Katzenbeisser, S. and Uhl, A. (Eds), *Communications and Multimedia Security*, Vol. 3677, Springer, Berlin, pp. 32-41.

Ortega, J.L. and Aguillo, I.F. (2008), "Visualization of the Nordic academic web: link analysis using social network tools", *Information Processing & Management*, Vol. 44 No. 4, pp. 1624-33.

Petname (2011), "Petname tool 1.7", available at: http://petname.mozdev.org/ (accessed 26 July 2011).

Plaisant, C. (2004), "The challenge of information visualization evaluation", *Proceedings of the Working Conference on Advanced Visual Interfaces in Italy*, ACM, New York, NY, pp. 109-16.

Plonka, D. (2000), "FlowScan: a network traffic flow reporting and visualization tool", *Proceedings of the 14th USENIX Conference on System Administration, New Orleans, LA*, USENIX Association, Berkeley, CA, pp. 305-18.

Schneier, B. (2000), *Secrets and Lies: Digital Security in a Networked World*, Wiley, Indianapolis, IN.

Scott, D. and Sharp, R. (2002), "Abstracting application-level web security", *Proceedings of the 11th International Conference on World Wide Web*, ACM, New York, NY, pp. 396-407.

Shneiderman, B. (1996), "The eyes have it: a task by data type taxonomy for information visualizations", *Proceedings of the IEEE Symposium on Visual Languages 1996*, IEEE Computer Society Press, Washington, DC, pp. 336-43.

SpoofStick (2011), "SpoofStick", available at: www.spoofstick.com/ (accessed 26 July 2011).

Sunshine, J., Egelman, S., Almuhimedi, H., Atri, N. and Cranor, L.F. (2009), "Crying wolf: an empirical study of SSL warning effectiveness", *Proceedings of the 18th Conference on USENIX Security Symposium*, USENIX Association, Berkeley, CA, pp. 399-416.

Takada, T. and Koike, H. (2002), "Tudumi: information visualization system for monitoring and auditing computer logs", *Proceedings of the Sixth International Conference on Information Visualization*, IEEE Computer Society Press, Washington, DC, pp. 570-6.

Taylor, T., Brooks, S. and Mchugh, J. (2007), "NetBytes viewer: an entity-based NetFlow visualization utility for identifying intrusive behavior", *Proceedings of the 2007 Workshop on Visualization for Computer Security, Sacramento, CA*, Springer, Berlin, pp. 101-14.

Weinreich, H. and Lamersdorf, W. (2000), "Concepts for improved visualization of web link attributes", *Computer Networks*, Vol. 33 Nos 1-6, pp. 403-16.

Whitten, A. and Tygar, J.D. (1999), "Why Johnny can't encrypt: a usability evaluation of PGP 5.0", *Proceedings of the 8th USENIX Security Symposium, Washington, DC*, USENIX Association, Berkeley, CA, pp. 169-84.

Yahoo! (2011), "Yahoo! Sign-in Seal", available at: http://security.yahoo.com/article.html?aid=2006102507 (accessed 26 July 2011).

Ye, Z., Smith, S. and Anthony, D. (2005), "Trusted paths for browsers", *ACM Transactions on Information and System Security*, Vol. 8 No. 2, pp. 153-86.

Yee, K.-P. and Sitaker, K. (2006), "Passpet: convenient password management and phishing protection", *Proceedings of the Second Symposium on Usable Privacy and Security*, ACM, New York, NY, pp. 32-43.

Yin, X., Yurcik, W., Treaster, M., Li, Y. and Lakkaraju, K. (2004), "VisFlowConnect: NetFlow visualizations of link relationships for security situational awareness", *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, ACM, New York, NY, pp. 26-34.

About the authors

Tran Khanh Dang received his BEng degree from the Faculty of Computer Science & Engineering in HCMC University of Technology-HCMUT (Vietnam) in 1998. He achieved the medal awarded for the best graduation student. From 1998-2000, he worked as a lecturer and researcher in the same faculty. Then, he got a PhD scholarship of the Austrian Exchange Service (OeAD) from 2000-2003, and finished his PhD degree (Drtechn.) in May 2003 at FAW Institute, University of Linz (Austria). Afterwards, he worked as a lecturer and researcher at the School of Computing Science, Middlesex University in London (UK) from August 2003. In October 2005, he returned home and has continued working for the Faculty of Computer Science & Engineering in HCMUT. In 2010, he was appointed as an Associate Professor of Computer Science in Vietnam. Dr Dang's research interests include database and information security, biometric-based authentication and cryptography, privacy protection in location-based services, information retrieval, and distributed data management. He has published more than 75 scientific papers in international/national journals and conferences. Dr Dang has also participated in and managed many research as well as commercial projects. Tran Khanh Dang is the corresponding author and can be contacted at: khanh@cse.hcmut.edu.vn

Tran Tri Dang received his Bachelor degree in Mechanical Engineering and Master degree in Computer Science from the University of Technology, Vietnam National University Ho Chi Minh City (HCMUT) in 2005 and 2009, respectively. Before joining the Department of Information Systems, Faculty of Computer Science and Engineering, HCMUT in 2009, he had worked as a trainer, programmer, and technical project manager for some international companies from 2004. Mr Tri has taught undergraduate courses in information security, database management systems, management information systems, and electronic commerce. His research interests include: information visualization, computer security, World Wide Web, and mobile platform. He is currently a PhD candidate at HCMUT.