EXHIBIT C

IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF ILLINOIS
EASTERN DIVISION

MIKE HARRIS and JEFF DUNSTAN, individually and on behalf of a class of similarly situated individuals,	Case No. 1:11-5807
Plaintiffs,	
v.	Hon. James F. Holderman
COMSCORE, INC., a Delaware corporation,	
Defendant.	

Expert Witness Report of Don Waldhalm

Prepared Pursuant to Fed. R. Civ. P. 26(a)(2)(B)

I have prepared this Expert Witness Report ("Report") pursuant to Rule 26(a)(2)(B) of the Federal Rules of Civil Procedure for the purpose of summarizing my forthcoming expert opinion testimony to be offered in the above-captioned case.

A. Qualifications

I am currently the Director of Software Development for The Sylint Group, Inc., a company specializing in cyber security, digital forensics, and eDiscovery. I graduated in 2001 with an industrial engineering degree from Mississippi State University and have been employed as a professional computer programmer for over a decade. For the past seven (7) years I have worked extensively in the fields of digital forensics and source code analysis. A copy of my *Curriculum Vitae*, which further describes my educational and professional background, is attached as **Exhibit A**.

B. Statement of Opinions to be Expressed at Trial and the Bases or Reasons for Those Opinions.

The bases for my opinions summarized below, which I expect to provide further testimony about in this matter, are the result of an extensive, eight (8) month review of Defendant comScore, Inc.'s ("comScore") source code that was provided to me by Plaintiffs' counsel. In addition, I was provided with three (3) Exhibits that were produced by comScore during discovery in this lawsuit, two of which describe the types of data that its software was designed to collect from individual computers, and the other shows a variety of dialog boxes that may be displayed to a user during the software's installation process.

In January 2012, I was provided with two (2) sets of comScore's source code. The first relates to comScore's software designed for use with Microsoft Windows ("Windows") operating systems; the other is designed for use with computers running the Macintosh operating system ("MacOS"). Subsequently, in April 2012, I was provided with eight additional sets of source code related to various builds of the MacOS software. In addition to the below, I intend to testify that while certain differences exist amongst the MacOS and Windows versions of the source code, both are designed to function in a substantially similar fashion. Accordingly, this Report collectively refers to the source code comprising comScore's software, unless otherwise specified. Following are the six main topics that I intend to testify about.

1. RKVerify, the software component that was designed to validate user consent when installing comScore's software, confirms that certain textual elements contained in the dialog box displayed to users during installation are present, but does not verify the existence of a functioning hyperlink to comScore's full EULA.

comScore developed a software module titled "RKVerify," which is designed to confirm that certain information is displayed in a "dialog box" to users during its software's installation process. I intend to testify that RKVerify is designed to confirm that certain textual elements are contained in that dialog box, such that every user who installs the software views the same, or substantially similar Terms of Service language. In addition, based upon my review of its source code, I do not believe that RKVerify was designed to confirm that a functioning hyperlink to comScore's full End User License Agreement ("EULA") is contained within the dialog box.

2. *comScore's software captures the contents of webpages viewed by a user, as well as information that he or she inputs into a webpage.*

A significant portion of my testimony will involve explaining the manner in which comScore's software is designed to capture and transmit back to its servers data that a user inputs into a web browser, as well as the content displayed on websites viewed by the user. This data collection occurs during secure and unsecure web browsing sessions. Additionally, during HTTPS POST operations—where the user is most likely to input certain sensitive information (such as credit card numbers, social security numbers, or passwords) into his or her web browser—the software intentionally captures and hashes fields in that data and sends it to comScore's servers. A rudimentary explanation of the technology underlying the act of web browsing, as well as the collection and hashing method used by comScore's software are helpful in illustrating how this works.

The "language" used to facilitate communications between web browsers and websites is called Hypertext Transfer Protocol ("HTTP"). When a user requests information—through his or her web browser—from a website, the browser makes an HTTP "request," and the website's server sends back an HTTP "response." Traditional HTTP communications are unencrypted, and thus are susceptible to interception and exploitation. To avoid these dangers, a secure version of the HTTP service was created—HTTP Secure ("HTTPS"). Using the HTTPS protocol, communications sent and received from a user's web browser are encrypted. To inform users when a connection with a website is "secure," web browsers generally display a graphical image of a padlock in the upper left corner of the URL toolbar.

By design, comScore's software captures both HTTP and HTTPS traffic. For practical purposes, this means that information inputted into a web browser by a user—including names, addresses, phone numbers, social security numbers, usernames, passwords, bank account numbers, and credit card numbers—is collected and transmitted to comScore's servers. The software hashes sensitive data before sending it to comScore's servers.

In this context, a hash function is a cryptographic method used to convert an arbitrary block of data into a fixed-size bit string. The particular hash method used by comScore's software is known as the MD5 hash. To illustrate how the method works, when a MD5 hash is applied to the password "12345," the algorithm would output the following hash value: 827ccb0eea8a706c4c34a16891f84e7b. Thus, with respect to comScore's software, when a user inputs the password "12345" into his or her web browser, the hash value, as well as the corresponding password field's name, are transmitted to comScore's servers.

Input boxes on websites are assigned "field names" in the webpage's code. For example, an input box that allows a user to provide a password may be titled "password_1" within that webpage's code. Depending on the type of sensitive information to be collected, comScore's software will identify such field names, hash the data inputted into that field, and then send both the field name and its now transformed contents back to comScore's servers. Thus, by way of example, if John Doe enters his password into a website, that password will be manipulated and transmitted to comScore's servers, along with the underlying name of the field (*i.e.*, "password_1").

In addition to the foregoing data collection practices, comScore's software also captures certain text contained in PDF files viewed by a user. To accomplish this, the software converts the text of PDF files into HTML format, and then transmits the converted data to its servers. Another function of the software is that it identifies certain predetermined keywords on a webpage and then captures "snippets"—a set number of characters—immediately proceeding and following that keyword.

With the above forming the bases of my opinion, I intend to testify that comScore takes deliberate steps to gather and transmit sensitive information that a user inputs into a web browser. I also intend to testify that comScore does not take steps to "filter" or refrain from collecting such data. Furthermore, I intend on testifying that comScore's methods for attempting to protect users' sensitive information (*i.e.*, hashing) do not meet commercially viable standards.

3. comScore's software scans every directory on a user's computer and collects information about certain files.

I also intend to testify about comScore's software's methods for searching through the "directory tree" on a user's computer and for gathering information about the existence of certain files. During such searches, comScore's software is designed to capture, among other information, file names and directory paths. comScore's software packages this information into a compressed file and transmits it back to comScore's servers.

In order to complete the installation process, comScore's software requires administrator privileges. Once installation is complete, the software retains such privileges. Thus, while operating on a user's computer, the software is able to access or manipulate any file, regardless of user security settings that may protect such files. When the software does access a file, comScore's software is designed to modify the file's "last accessed date" metadata field, such that it would appear that the file was not accessed by the software even though it has been.

Based on my review, I also understand that only certain types of files or file names will be sent back to comScore's servers, as defined by rules enumerated by comScore. comScore's software does appear to categorically collect and transmit full directory path names to comScore's servers.

4. comScore designed its MacOS software to scan and inventory all network drives.

As introduced above, both the MacOS and Windows versions of comScore's software are designed to function in substantially similar fashion. One difference I intend to testify about, however, is that certain MacOS software was designed to scan files and file directories (as described in Section 3) located not only on local hard drives but also on networked drives mapped to the user's computer.

5. comScore captures a user's web browsing history from the time period prior to installation on his or her computer.

My review of the source code also revealed, and I expect to testify, that comScore's software was designed to access the Internet Explorer web browser's cache—a temporary file containing the user's web viewing history—and collect the previous 25 websites visited by the user prior to the installation of comScore's software. The software then sends an HTTP request to those 25 websites, in order to initialize the top URLs per process metric in the comScore software—as if the software had been installed during the last 25 web requests.

6. comScore targets iPhone, iPad, iPod, and Blackberry backup files for data collection.

My review also shows that comScore's software is designed to access data contained in a user's iPhone, iPad, iPod, and BlackBerry backup files.

Examples of information that the source code is designed to collect from the user's smartphone's backup files are the device's unique ID, web browsing history, and information about mobile applications used on the device.

C. Exhibits

In my testimony, I may use as exhibits certain illustrative diagrams to help explain how comScore's software works. I may also use relevant portions of comScore's source code and certain other documents produced by comScore during discovery in this lawsuit as exhibits to assist in that explanation.

D. Compensation

I am compensated at a rate of $165.00 per hour for my work on this case, including for any appearances at deposition or trial.

E. Testimony

I have not testified at any trial or during a deposition within the past four (4) years.

Dated: September 17, 2012

DON WALDHALM

EXHIBIT A

The Sylint Group, Inc.

240 N Washington Blvd
Suite 600
Sarasota FL, 34202

Don Waldhalm

Software Development Work Experience

Director of Software Development

Sylint Group

February 2005-Present

Projects

Source code analysis for international intellectual property infringement case:

- Organized the review process for neutral third party arbitrator
  - Negotiated neutral third-party selection process
  - Reviewed neutral third-party qualifications and applications
  - Determined neutral third-party preference
- Outlined source code comparison goals and techniques
  - Reviewed source code to validate third-party opinion
  - Identified relevant portions of source code to be compared
  - Identified relevant properties to be sought
  - Identified terms and conditions to be considered infringing
- Relevant Technologies:
  - C++
  - Delphi
  - Windows API
  - IE add-on

E-discovery development for internal Sylint Use:

- Designed and developed the "DRIL ®" application (Document Review and Inspection for Litigation).
- DRIL provides the mechanism by which Sylint accelerates communication between technical analysts and the litigation teams that they support.
- Provides an intuitive interface which is accessible from anywhere, so our clients and their clients can collaborate in their e-discovery efforts with ease.
- Software is used extensively by several national law firms.

Forensic development for internal Sylint Use:

- Designed and developed "Disk Mapper ®" – which helps Sylint detect and describe intentional spoliation of electronic evidence.

- Automatically identifies discrepancies between actual data and the MFT on an evidence drive.
- Has passed Daubert challenge in Federal Court

Cyber security development and support for multiple fortune 1000 companies

- Developed “Sylint Stalker”-- a series of applications and windows services that help Sylint detect and defeat state-of-the-art viruses, malware, and cyber warfare attacks.
- Scalable to very large networks (over 35,000 computers).
- Prevents command and control communication without alerting virus handlers.

Line-of-business migration for a Large National Nonprofit Organization:

- Replaced an underperforming and high-maintenance business application.
- Led a team of developers comprised of both internal and client staff.
- Architected an entire enterprise solution for streamlining the line-of-business operations and performance monitoring.
- Designed and built a SQL database to replace three different Access databases and oversaw the data-migration effort.
- Designed and built all the middleware business classes.
- Oversaw and participated in the building of the Windows UI.
- Architected the integration of existing ASP web pages with new ASPX pages.
- Designed ASP.net web services for partners and vendors to use for automatic integration with the new application.
- Integrated the existing website with new business classes.
- Technologies used:
    - SQL server 2005
    - C# and VB.net (Dot Net framework 2.0 and 1.1)
    - ASP.net
    - ASP classic
    - CSS style sheets
    - Authorize.net shopping card
    - SOAP web services
    - CSLA business object framework
    - DevExpress UI controls
    - Code Smith code generator

Line-of-business application for a Multi-Lawyer Immigration Law Firm:

- Built a custom business application
    - Tracks the progress of cases and the legal status of both clients and potential clients.
    - Provides an interface for daily operations, as well as client retention and generation activities.
    - Serves as a digital archive for a large collection of historical records.

- Designed and built a SQL database and Windows client for digital archiving.
- Built a custom scanning and PDF-generating module for archiving cases as they are processed.
- Integrated government forms so they are filled out, processed, and archived in one step instead of three.
- Implemented various performance-monitoring reports.
- Technologies used:
    - SOAP web services
    - VB.NET (Dot net Framework 1.1)
    - SQL server 2000
    - VS TWAIN
    - PDF
    - Crystal Reports

Line-of-business application for a Genetic Forensic Analysis Company:

- Designed and built a line-of-business application for the company which had recently made several corporate acquisitions.
- Used SQL 2005 and C#.net to build a Windows application which is used in all processing information and accounting interfaces.
- Provided interfaces for existing shopping carts from several existing web sites—each using its own form of online payment.
- Integrated the new system with an existing oracle-based application that drives some of the DNA analysis machines.
- Re-engineered existing Java application for statistical analysis of DNA data.
- Designed and installed a processing system utilizing barcode scanners and label printers that helped to streamline the interactions with clients as well as internal operations.
- Technologies used:
    - C#.net
    - SQL server 2005
    - Oracle
    - SQL-CE
    - Java
    - Pay-Pal integration

Source code and development process analysis for a large power utility company:

- Analyzed an existing application, development team and development process. Reported to top management with technical and staffing recommendations.
- Identified technical weaknesses and strengths.
- Identified business process weaknesses and strengths.

- Identified staff members and management personnel who were in positions for which they are not qualified, and who were underperforming.
- Identified staff members who were in positions beneath their qualifications and skill level.
- Produced a project development plan to:
  - Reduced development staff from 43 to less than 10
  - Establish and meet reliability and performance goals
  - Properly implement n-tier distributed application design and SQL server best practices (especially security related)
  - Improve application scalability

Application development for Internal Sylint Use:

- Created a console login application that retains admin credentials for a list of severs and automatically opens a remote console session to the selected server.
- Created an event log file gatherer that sweeps the network and gathers previously identified log files from client PCs.
- Integrated phone systems with existing and new databases.
- Designed and set-up a printing system for voice-mail messages for an Altigen phone system.
- Built a utility to detect indicators of potential forensic tampering.
- Technologies used:
  - C++
  - Windows API
  - Regular Expressions
  - Extensive use of the Dot Net framework

Director of Logistics and Support

Glenroe Technologies (mid-sized orthodontic manufacturing company)

March 2002-February 2005

- Reported directly to the president
- Upper-management position with the following employees reporting to me:
  - Shipping department manager
  - Customer service department manager
  - Marketing Manager
  - IT Manager
  - Quality Control department manager
  - Warehouse Manager
  - Two Industrial Engineers (I served as the IE department manager)

Projects

Warehouse Partner:

VB.net, SQL 2000, RF barcode scanners, Zebra label printers.

Production modeling with Extend:

Custom API

ISO 9000 content management:

MS Access

Website Management:

Front page and Dreamweaver

Great Plains Dynamics:

Daily monitoring and SQL backups

Application configuration

Custom processing application production departments:

VB 6, MS Access, SQL 2000, MS Excel

Streamline solutions for Shipping, Sales, and Customer service:

MS Word, mail merge, MS Excel, MS Access, SQL 2000, VB6, VBA

IT support for 30 users:

MS Exchange server, MS CRM, Great Plains Dynamics (now also MS)

Ladder logic controllers for custom machines:

Heaters, robot arms, scales, printers, lights, bells, etc…

Sylint Software Development Methodology

Sylint develops software for its clients using the experience gained by John Jorgensen, President of Sylint, as both a Program Manager for the National Security Agency and Director of Program Management for Loral Data Systems Division in their execution of government contracts. Government software development standards and methodology are used by Sylint to ensure: 1) comprehensive development documentation; 2) testability and verification; 3) reliability; 4) non-problematic upgradability; 5) ease of implementation; and 6) longevity of base software.

Although we are always willing to adapt to existing frameworks whenever it is advantageous to the client or project at hand (for example, MS Project , AxoSoft On Time Bug Tracker, Microsoft Source

Control, Mercurial, CVS, Java Netbeans) we currently prefer to work with Microsoft's suite of tools and languages. Specifically, we use Microsoft Team Foundation to manage software development projects and the related source control. Also, we build most of our applications on top of the Dot Net framework. C# is our preferred language.

Education

Bachelor of Science

Industrial Engineering
Mississippi State University, 2001