A system, method, and computer program product are provided for preventing access to data associated with a data access attempt. In use, a data access attempt associated with a remote data sharing session is identified. Further, access to the data is prevented.
FIGURE 1
Identifying a data access attempt associated with a remote data sharing session

Preventing access to the data

Figure 3
START

URL ACCESS REQUEST?

COMPARE URL TO KNOWN URLs ASSOCIATED WITH REMOTE DESKTOP SHARING

MATCH?

PREVENT ACCESS TO URL

MANUALLY ALLOW?

ALLOW ACCESS TO URL

FIGURE 4
START

DATA ACCESS REQUEST?

YES

DATA FINGERPRINTED?

YES

IDENTIFY APPLICATION THAT ISSUED REQUEST

ALLOW ACCESS

APPLICATION ALLOWED TO ACCESS THE DATA?

YES

NO

PREVENT ACCESS

FIGURE 5
START

REMOTE DATA SHARING ENABLED?

DATA ACCESS REQUEST?

IDENTIFY FINGERPRINT OF DATA

ALLOW ACCESS

DATA FINGERPRINT MATCHES KNOWN FINGERPRINT?

PREVENT ACCESS

FIGURE 6
SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR PREVENTING ACCESS TO DATA WITH RESPECT TO A DATA ACCESS ATTEMPT ASSOCIATED WITH A REMOTE DATA SHARING SESSION

FIELD OF THE INVENTION

[0001] The present invention relates to data loss prevention, and more particularly to preventing data loss by preventing access data.

BACKGROUND

[0002] In the past, security systems have been developed for preventing data loss. For example, such data loss has generally included the unauthorized or otherwise unwanted disclosure of data (e.g., confidential data, etc.). However, security systems have exhibited various limitations in preventing data loss. For example, security systems have conventionally been deficient in preventing data loss due to remote data sharing.

[0003] There is thus a need for addressing these and/or other issues associated with the prior art.

SUMMARY

[0004] A system, method, and computer program product are provided for preventing access to data associated with a data access attempt. In use, a data access attempt associated with a remote data sharing session is identified. Further, access to the data is prevented.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] FIG. 1 illustrates a network architecture, in accordance with one embodiment.

[0006] FIG. 2 shows a representative hardware environment that may be associated with the servers and/or clients of FIG. 1, in accordance with one embodiment.

[0007] FIG. 3 shows a method for preventing access to data associated with a data access attempt, in accordance with one embodiment.

[0008] FIG. 4 shows a method for preventing access to a uniform resource locator (URL) associated with remote desktop sharing, in accordance with another embodiment.

[0009] FIG. 5 shows a method for preventing access to data based on an application that initiated a data access request, in accordance with yet another embodiment.

[0010] FIG. 6 shows a method for preventing access to data based on a fingerprint of the data, in accordance with still yet another embodiment.

DETAILED DESCRIPTION

[0011] FIG. 1 illustrates a network architecture 100, in accordance with one embodiment. As shown, a plurality of networks 102 is provided. In the context of the present network architecture 100, the networks 102 may each take any form including, but not limited to a local area network (LAN), a wireless network, a wide area network (WAN) such as the Internet, peer-to-peer network, etc.

[0012] Coupled to the networks 102 are servers 104 which are capable of communicating over the networks 102. Also coupled to the networks 102 and the servers 104 is a plurality of clients 106. Such servers 104 and/or clients 106 may each include a desktop computer, lap-top computer, hand-held computer, mobile phone, personal digital assistant (PDA), peripheral (e.g., printer, etc.), any component of a computer, and/or any other type of logic. In order to facilitate communication among the networks 102, at least one gateway 108 is optionally coupled therebetween.

[0013] FIG. 2 shows a representative hardware environment that may be associated with the servers 104 and/or clients 106 of FIG. 1, in accordance with one embodiment. Such figure illustrates a typical hardware configuration of a workstation in accordance with one embodiment having a central processing unit 210, such as a microprocessor, and a number of other units interconnected via a system bus 212.

[0014] The workstation shown in FIG. 2 includes a Random Access Memory (RAM) 214, Read Only Memory (ROM) 216, an I/O adapter 218 for connecting peripheral devices such as disk storage units 220 to the bus 212, a user interface adapter 222 for connecting a keyboard 224, a mouse 226, a speaker 228, a microphone 232, and/or other user interface devices such as a touch screen (not shown) to the bus 212, communication adapter 234 for connecting the workstation to a communication network 235 (e.g., a data processing network) and a display adapter 236 for connecting the bus 212 to a display device 238.

[0015] The workstation may have resident thereon any desired operating system. It will be appreciated that an embodiment may also be implemented on platforms and operating systems other than those mentioned. One embodiment may be written using JAVA, C, and/or C++ language, or other programming languages, along with an object oriented programming methodology. Object oriented programming (OOP) has become increasingly used to develop complex applications.

[0016] Of course, the various embodiments set forth herein may be implemented utilizing hardware, software, or any desired combination thereof. For that matter, any type of logic may be utilized which is capable of implementing the various functionality set forth herein.

[0017] FIG. 3 shows a method 300 for preventing access to data associated with a data access attempt, in accordance with one embodiment. As an option, the method 300 may be carried out in the context of the architecture and environment of FIGS. 1 and/or 2. Of course, however, the method 300 may be carried out in any desired environment.

[0018] As shown in operation 302, a data access attempt associated with a remote data sharing session is identified. In the context of the present description, the data may include information, code, and/or anything else capable of being associated with a remote data session. In various embodiments, the data may include any number of documents, electronic mail (email) messages, programs, uniform resource locators (URLs), etc. Additionally, the data may be stored on a client, a server, and/or any other device (e.g., such as any of the devices described above with respect to FIGS. 1 and/or 2, etc.).

[0019] To this end, the data access attempt may include any attempt associated with a remote data sharing session to access data. For example, the data access attempt may include a request to access the data. In other examples, the data access attempt may include an attempt to open the data, read the data, write to the data, copy the data, attach the data to other data (e.g., an email), display the data utilizing a liquid crystal display (LCD) projector, etc.

[0020] In the context of the present description, the remote data sharing session may include any session in which the
data may be shared remotely, where the term remotely indicates the involvement of any device separate from the device on which the data is stored, etc. For example, the remote data sharing session may, in one embodiment, include a time period in which remote data sharing is enabled. As an option, the data may be shared remotely by viewing the data remotely, interacting with the data remotely, etc. In one embodiment, such remote data sharing may include any displaying, presenting, etc. of data located at a first location to a remote second location. Just by way of example, the remote data sharing may include sharing a desktop display with a remote computer, sharing the data with a projector (e.g. LCD projector, etc.) which projects the data, etc.

Moreover, the remote data sharing session may be associated with (e.g. facilitated by, etc.) a remote data sharing application. For example, the remote data sharing application may include a remote desktop application (e.g. Microsoft® Office Live Meeting, Citrix® GoToAssist®, etc.). Thus, the remote data sharing application may optionally be capable of sharing data remotely from a first device with a second device. As an option, the data access attempt may be associated with the remote data sharing session by being initiated via the remote data sharing session (e.g. via a command executed during the remote data sharing session). As another option, the data access attempt may include an attempt to access the remote data sharing session, the remote data sharing application associated with such session and/or any other aspect associated with the remote data sharing session.

To this end, the data access attempt may be initiated manually (e.g. by a user), in one embodiment. In another embodiment, the data access attempt may be initiated automatically (e.g. via an application, etc.). As described above, the data access attempt may also be initiated via the remote data sharing session.

Further, the data access attempt may be identified in any desired manner. In one embodiment, the data access attempt may be identified utilizing a client (e.g. on which the data is stored, etc.). In this way, the client may identify data access attempts initiated at the client. For example, the data access attempt may be identified utilizing an agent installed on the client, which monitors data access attempts.

As another example, the data access attempt may be identified utilizing a plug-in, add-in, etc. to an application (e.g. web browser, word processing application, data sharing application, etc.) associated with, installed on, etc. the client. As an option, such application may be the source of the data access attempt, an application utilized in accessing the data, an application utilized for sharing the data remotely, etc. Thus, each of a plurality of applications associated with the client may be associated with a separate plug-in, etc. As another option, the plug-in, etc. may be continuously active when the application is running (e.g. being executed).

In another embodiment, the data access attempt may be identified utilizing a gateway. For example, the gateway may identify the data access attempt based on network traffic received over a network (e.g. such as any of the networks described above with respect to FIG. 1). As an option, such gateway may similarly utilize an agent, plug-in, etc. for identifying the data access attempt.

As shown, access to the data is prevented. Note operation 304. In the context of the present description, the access of operation 304 may include any access associated with (e.g. requested in conjunction with, etc.) the data access attempt. In various embodiments, the access may be prevented by blocking the access, disallowing the access, denying a request associated with the data access attempt, disallowing network traffic associated with the data access attempt, etc. Of course, however, the access to the data may be prevented in any desired manner.

In one embodiment, the access may be prevented, if it is determined that the data matches predetermined data. Such predetermined data may include known confidential data (e.g. data predetermined to be confidential, etc.). In another embodiment, the access may be prevented, if it is determined that a fingerprint (e.g. hash, etc.) of the data matches a predetermined fingerprint, such as a fingerprint of known confidential data, for example.

In yet another embodiment, the access may be prevented, if it is determined that a remote data sharing application associated with the remote data sharing session is predetermined to be disallowed from accessing the data. For example, a user may configure (e.g. predefine, etc.) remote data sharing applications allowed to and/or disallowed from accessing data. As an option, such remote data sharing applications may be predetermined with respect to each of a plurality of instances of different data, with respect to locations of data capable of being accessed, with respect to categories of data capable of being accessed (e.g. file types, etc.), and/or with respect to any data capable of being accessed.

In still yet another embodiment, the access may be prevented based on a determination of whether the remote data sharing session is enabled. For example, if the remote data sharing session is enabled, access to the data may be prevented. Of course, however, preventing access to the data may be based on any desired criteria.

To this end, such access to data may be prevented in any desired manner. In one embodiment, such access prevention may eliminate unwanted loss, disclosure, etc. of the data via the remote data sharing session. For example, preventing access to the data may prevent the data from being presented, displayed, etc. to a remote device utilizing remote data sharing techniques associated with the remote data sharing session. Accordingly, in addition to optionally educating users on potential data leakage via remote data sharing sessions, such data leakage may also be limited by preventing access to data when a data access attempt is associated with a remote data sharing session.

More illustrative information will now be set forth regarding various optional architectures and features with which the foregoing technique may or may not be implemented, per the desires of the user. It should be strongly noted that the following information is set forth for illustrative purposes and should not be construed as limiting in any manner. Any of the following features may be optionally incorporated with or without the exclusion of other features described.

FIG. 4 shows method 400 for preventing access to a uniform resource locator (URL) associated with remote desktop sharing, in accordance with another embodiment. As an option, the method 400 may be carried out in the context of the architecture and environment of FIGS. 1-3. Of course, however, the method 400 may be carried out in any desired environment. It should also be noted that the aforementioned definitions may apply during the present description.

As shown in operation 402, it is determined whether a URL access request has been issued. In the context of the present embodiment, the URL access request may include a request to access content (e.g. web content, etc.) associated
with a URL. In one embodiment, the URL access request may be
issued via a web browser. For example, the URL access request may be issued based on a user selection of a web link on a web page displayed via the web browser, a user entry of the URL into the web browser, etc.

[0034] Further, the URL access request may be identified utilizing an agent installed on a client via which the URL access request is issued. In another embodiment, the URL access request may be identified utilizing a plug-in, add-in, etc. associated with the web browser via which the URL access request is issued. In yet another embodiment, the URL access request may be identified utilizing a plug-in, add-in, etc. associated with an application enabled for remotely sharing data. In still yet another embodiment, the URL access request may be identified utilizing an agent, plug-in, etc. installed on a gateway (e.g. via which the URL access request is communicated over a network, etc.).

[0035] In response to a determination that the URL access request has been issued, the URL is compared to known URLs associated with remote desktop sharing. Note operation 404. Such known URLs may include any URLs predetermined to be associated with remote desktop sharing. For example, the known URLs may include a location on a network of a remote desktop sharing application capable of being utilized for remotely sharing a desktop. Optionally, such known URLs may be predetermined based on a user configuration, based on an automatic configuration (e.g. web crawler, etc.).

[0036] In one embodiment, the known URLs may be stored in a library of known URLs. In another embodiment, the known URLs may be stored on the client via which the URL access request is initiated. In yet another embodiment, the known URLs may be stored at a central location (e.g. central server, etc.) capable of being accessed by the client and/or gateway. Optionally, the URL may be compared to the known URLs by comparing any portion or entirety of the URL with any respective portion or entirety of the known URLs.

[0037] It is further determined whether the URL matches any of the known URLs, as shown in decision 406. To this end, such determination may be based on the comparison of the URL with the known URLs. If is it determined that the URL does not match any of the known URLs, access to the URL is allowed. Note operation 412. Such access may include the access requested by the URL access request. In one embodiment, content associated with the URL, such as a web page, may be allowed to be presented. In another embodiment, the URL access request may be allowed to be sent to a destination (e.g. web server, etc.) associated with the request.

[0038] If however, it is determined that the URL matches one of the known URLs, access to the URL is prevented. Note operation 408. In one embodiment, content associated with the URL may be prevented from being presented. In another embodiment, the URL access request, such as network traffic associated with such URL access request, may be prevented from being communicated to the destination associated with the request. As an option, access to the URL may be prevented utilizing the agent, plug-in, etc. used for identifying the URL access request (as described above in operation 402).

[0039] Moreover, it is determined whether access to the URL is manually allowed, as shown in operation 410. In one embodiment, manually allowing access to the URL may include a user selecting (e.g. via a user interface) to allow the access. The user may include any user authorized to manually allow such access. For example, in response to preventing access to the URL (operation 408), a notification may be communicated to the user. Additionally, such notification may include an option capable of being selected by the user for manually allowing access to the URL.

[0040] In another embodiment, access to the URL may be manually allowed based on a predefined list of URLs to which access is allowed. For example, a user may configure a list of URLs associated with remote desktop sharing to which access is allowed. Thus, if the URL matches a URL in the predefined list of URLs to which access is allowed, access to the URL may be manually allowed.

[0041] In response to a determination that access to the URL is manually allowed, access to the URL is allowed, as shown in operation 412. To this end, access to a URL may be allowed automatically if the URL does not match known URLs associated with remote desktop sharing or manually as desired by a user. Still yet, it may be continuously determined whether access to the URL is manually allowed (e.g. for a predefined time period, etc.). In this way, access to the URL may optionally be allowed at any time after access to the URL is prevented.

[0042] FIG. 5 shows a method 500 for preventing access to data based on an application that initiated a data access request, in accordance with yet another embodiment. As an option, the method 500 may be carried out in the context of the architecture and environment of FIGS. 1-4. Of course, however, the method 500 may be carried out in any desired environment. Again, it should also be noted that the aforementioned definitions may apply during the present description.

[0043] In decision 502, it is determined whether a data access request has been issued. In one embodiment, the data access request may include a request to access a document. Just by way of example, the data access request may include a request to open the document. As another example, the data access request may include a request to attach the data to an email, a document, etc.

[0044] In another embodiment, the data access request may be issued via an application program interface (API). In yet another embodiment, the data access request may be issued manually by a user, for example, by selecting to open the data. In still yet another embodiment, the data access request may be issued automatically (e.g. via an application requesting to access the data, etc.).

[0045] Further, the data access request may be identified utilizing an agent installed on a client via which the data access request is issued. In another embodiment, the data access request may be identified utilizing an agent installed on a gateway (e.g. via which the data access request is communicated over a network, etc.). Of course, however, the data access request may be identified in any manner.

[0046] In response to a determination that the data access request has been issued, it is determined whether the data is fingerprinted. Note decision 504. For example, a plurality of predetermined fingerprints may be stored, in a database. Further, the database may store additional information with respect to the predetermined fingerprints. For example, the database may store identifiers of applications allowed to be utilized for accessing data associated with each of the predetermined fingerprints, disallowed for use in accessing such data, etc. As an option, the predetermined fingerprints and associated allowed/disallowed applications may be configured by a user.
Table 1 illustrates one example of a database capable of being utilized for storing predetermined fingerprints of data and identifiers of associated applications allowed to be utilized for accessing such data. In this way, the database may be utilized for associating each fingerprint with an application, it should be noted that the database is set forth for illustrative purposes only, and thus should not be construed as limiting in any manner.

<table>
<thead>
<tr>
<th>FINGERPRINT</th>
<th>ALLOWED APPLICATION IDENTIFIER</th>
</tr>
</thead>
<tbody>
<tr>
<td>FINGERPRINT_01</td>
<td>APPLICATION_01, APPLICATION_02</td>
</tr>
<tr>
<td>FINGERPRINT_02</td>
<td>APPLICATION_02</td>
</tr>
<tr>
<td>FINGERPRINT_03</td>
<td>APPLICATION_01</td>
</tr>
</tbody>
</table>

In another embodiment, predetermined applications may be determined to be applications allowed to access any data. For example, such dedicated applications may be predetermined based on a user configuration. As an option, the dedicated applications may include the only applications allowed to access fingerprinted data.

In yet another embodiment, predetermined applications may be disallowed from being utilized during a remote data sharing session. For example, if it is determined that one of the predetermined applications is running, a remote data sharing session may be prevented from being enabled. As another example, if it is determined that a remote data sharing session is enabled, one of the predetermined applications may be prevented from being initiated.

If it is determined that the application that issued the data access request is allowed to access the data, access to the data is allowed. Note operation 510. Such access may include the access requested by the data access request. In one embodiment, the data may be allowed to be presented, displayed, attached, etc. In another embodiment, the data access request may be allowed to be sent to a destination (e.g., server, etc.) associated with the request.

If, however, it is determined that the application that issued the data access request is not allowed to access the data, access to the data may be prevented. Note operation 512. In one embodiment, the data may be prevented from being presented. In another embodiment, the data access request, such as network traffic associated with such data access request, may be prevented from being communicated to the destination associated with the request. As an option, access to the data may be prevented utilizing the agent used for identifying the data access request (as described above in operation 502). Just by way of example, in one embodiment, the data access request may include a request to display the data utilizing a projector, such that data loss may be prevented with respect to a public sharing session associated with an LCD projector, etc.

In this way, for each of a plurality of different fingerprints of various data, applications may be indicated as being allowed to access the data and/or disallowed from accessing the data. Thus, particular data may only be accessible via predefined applications, as desired. In one embodiment, such predefined applications may allow a single agent installed on a client gateway, etc. to determine whether any of a plurality of different applications may be utilized for accessing data associated with a data access request.

FIG. 6 shows a method 600 for preventing access to data based on a fingerprint of the data, in accordance with still yet another embodiment. As an option, the method 600 may be carried out in the context of the architecture and environment of FIGS. 1-4. Of course, however, the method 600 may be carried out in any desired environment. Again, it should also be noted that the aforementioned definitions may apply during the present description.

As shown in decision 602, it is determined whether remote data sharing is enabled. In one embodiment, it may be determined whether the remote data sharing is enabled based on a determination of whether a remote data sharing application, or any associated processes, are executing. For example, an agent installed on a client may determine whether a remote data sharing application is executing on the client.

In response to a determination that the remote data sharing is enabled, it is determined whether a data access request has been issued, as shown in decision 604. In one
embodiment, the data access request may be identified utilizing an agent installed on the client via which the data access request is issued. In another embodiment, the data access request may be identified utilizing a plug-in, add-in, etc. associated with an application via which the data access request is issued. In yet another embodiment, the data access request may be identified utilizing a plug-in, add-in, etc. associated with a remote data sharing application.

If a data access request has been issued, a fingerprint of the data is identified, as shown in operation 608. The fingerprint of the data may be identified by hashing the data, in one embodiment. In another embodiment, the fingerprint of the data may be identified by calculating a value of the data utilizing a predetermined algorithm.

Furthermore, as shown in decision 608, it is determined whether the identified fingerprint matches a known fingerprint, in the context of the present embodiment, the known fingerprint may include any predetermined fingerprint of data. For example, a database may store a plurality of predetermined fingerprints of data. Optionally, such database may be stored locally (e.g. on a client on which the data access request was issued), but of course may also be stored remotely (e.g. at a location central to a plurality of clients on a network). Moreover, the predetermined fingerprints may be of known confidential data.

To this end, determining whether the identified fingerprint matches a known fingerprint may include comparing the identified fingerprint to a plurality of known fingerprints. If it is determined that the fingerprint of the data does not match a known fingerprint (e.g. based on the comparison, etc.), access to the data may be allowed. Note operation 610. For example, the access may include the access requested by the issued data access request (in operation 604). If, however, it is determined that the fingerprint of the data matches a known fingerprint (e.g. based on the comparison, etc.), access to the data may be prevented. Note operation 612.

To this end, data may be prevented from being accessed based on a fingerprint of the data when a remote data sharing session is enabled. In another optional embodiment, if it is determined that the data is already opened prior to enabling of a remote data sharing session, such data may be closed in response to a request to initiate the remote data sharing session. Thus, data loss may be prevented based on various access requests, including, for example, a public sharing session where the data is displayed on an LCD projector, etc.

While various embodiments have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of a preferred embodiment should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

1. A method, comprising:
   identifying, by a first computer, a data access attempt by a remote device, the data access attempt associated with a remote data sharing session wherein the remote data sharing session comprises sharing a desktop display of the first computer with the remote device; and
   automatically preventing access to data associated with the identified data access attempt.

2. The method of claim 1, wherein the data includes a uniform resource locator.

3. The method of claim 1, wherein the data includes a document.

4. (canceled)

5. The method of claim 1, wherein the act of identifying a data access attempt includes utilizing a plug-in to an application that is associated with the data access attempt.

6. The method of claim 1, wherein the act of identifying a data access attempt includes utilizing a plug-in to an application that is used for remote data sharing.

7. The method of claim 1, wherein the remote data sharing session is associated with a remote data sharing application.

8. The method of claim 7, wherein the remote data sharing application is predetermined to be disallowed from accessing the data.

9. The method of claim 8, wherein the remote data sharing application is predetermined to be disallowed from accessing the data, based on a user configuration.

10. The method of claim 1, wherein the act of identifying a data access attempt includes utilizing a client.

11. The method of claim 10, wherein the act of identifying a data access attempt includes utilizing a plug-in to an application installed on the client.

12. The method of claim 1, wherein the act of identifying a data access attempt includes utilizing a gateway.

13. The method of claim 1, further comprising identifying a fingerprint of the data.

14. The method of claim 13, further comprising comparing the fingerprint of the data to a plurality of predetermined fingerprints.

15. The method of claim 14, wherein the plurality of predetermined fingerprints include fingerprints of known confidential data.

16. The method of claim 14, wherein the plurality of predetermined fingerprints are each associated with an application.

17. The method of claim 14, wherein the act of automatically preventing access to the data includes preventing access to the data based on the comparison.

18. The method of claim 1, wherein the act of automatically preventing access to the data includes preventing access to the data if it is determined that a fingerprint of the data matches a predetermined fingerprint.

19. The method of claim 1, wherein the act of automatically preventing access to the data includes preventing access to the data if it is determined that the data matches predetermined data.

20. A computer program product embodied on a non-transitory computer readable medium, comprising:
   computer code for identifying a data access attempt by a remote device at a first computer, the data access attempt associated with a remote data sharing session wherein the remote data sharing session comprises sharing a desktop display of the first computer with the remote device; and
   computer code for automatically preventing access to data associated with the identified data access attempt.

21. A system, comprising:
   a memory; and
   a processor operatively coupled to the memory, the processor adapted to execute program code stored in the memory to:
   identify a data access attempt by a remote device at a first computer, the data access attempt associated with a remote data sharing session, wherein the remote data
sharing session comprises sharing a desktop display of the first computer with the remote device, and automatically prevent access to data associated with the identified data access attempt.

22-24. (canceled)