1D-8

アプリケーション操作ログの取得方式に関する一考察

荒木　信行†　　赤迫　貴行†　　楯　武士†

ＮＴＴコムウェア株式会社　研究開発部‡

１．はじめに

近年の情報漏洩事件の頻発や個人情報保護法の施行などにより、社会における情報セキュリティの重要性が増している。セキュリティを向上するための技術の分類として、1) 暗号化などの情報そのものを保護するアクティブセーフティ技術と 2) 内部の情報の流れを監視、あるいは情報が漏洩したときの経路の追跡などのパッシブセーフティ技術がある。本稿では、パッシブセーフティ技術の代表的な技術である情報の流通経路を監視する技術について新たな方式を提案し、そのシステム例と考察について述べる。

２．従来の技術とその課題

保護されるべき情報の一般的な流通経路について考え。保護されるべき情報はデータベースやファイルシステムのファイル上にデータとして蓄積されており、利用者はこのデータを専用あるいは汎用アプリケーションソフトウェア(以下、アプリケーション)を通して情報を利用する。そして情報漏洩事件のほとんどは、悪意ある利用者がこのアプリケーションを通して得た情報をネットワークあるいは外部記録装置を介して外部に持ち出すことによって発生している。

従来の情報流通の監視はネットワーク上を流れる情報を監視する、あるいは端末への外部記録装置の接続を監視することによって行われている。つまり、従来技術は端末の外部に情報が流出することを検知する技術であると言うことができる。

しかし、この技術では次の問題点があった。

1. ネットワーク上を流通する情報量が膨大であるため、個人情報など特定の情報の流通のみを監視・解析することが困難である。
2. ネットワークや外部記録装置を解さない手段によって情報がファイルに保存された場合、例えばディスプレイ画面に映し出された情報がデジタルカメラやデジタルビデオ、あるいはスクリーンコピーを利用して画像・映像ファイルとして保存された場合には、保存された画像・映像ファイルが外部へ流出しても情報の流出経路を追跡することは困難である。

３．提案方式

本稿では課題を解決するために端末からの外部への情報の流れを監視するのではなく、アプリケーションからの情報の流れに着目し、その監視を行う方式を提案する。具体的にはアプリケーションから OS(Operating System) あるいはミドルウェアへのシステムコールや API(Application Program Interface) の呼び出しとそのパラメータを監視、記録することで、いつ、誰が、どの情報に対してどのような操作を行ったかを把握する。（図１）

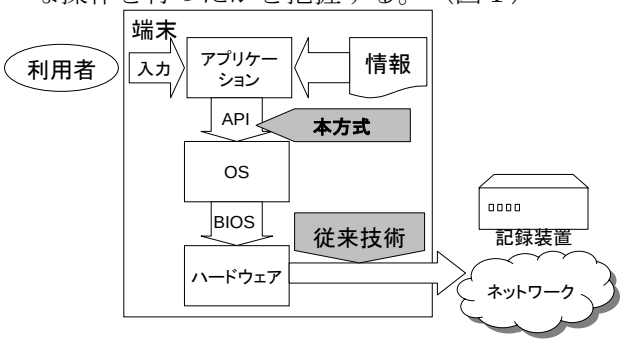

図１　提案方式

例として、ファイルの編集が行われた場合について考える。あるアプリケーションからあるファイルの読み出しが行われた場合、ファイル読み出しを行う API がファイル名をパラメータとして呼び出される。これによってアプリケーションはファイルの中のデータにアクセスすることができる。この後、利用者がデータを編集し、ファイル保存を行う際に、ファイル書き込みを行う API が同じファイル

A study of operation log system for arbitrary application software.

† Nobuyuki ARAKI, Takayuki AKASAKO, Takeshi TATE

‡ Research and Development Department, NTT COMWARE CORPORATION

名で共に呼び出される。このファイル操作を行う API の呼び出しをユーザ名、日時などと合わせて取得することで、いつ、誰が、どのデータ(ファイル)に対して、どのような操作を行ったかを取得することができる。また、ファイル操作以外についても他の API コールを監視対象とすることで、より多様な操作情報を取得することが可能となる。

４．実装例

前項の方式を元にネットワーク内の端末操作に関する情報を取得、解析するシステムについて、監視対象 API の登録から監視対象 API の解析までを以下に示す。（図２）

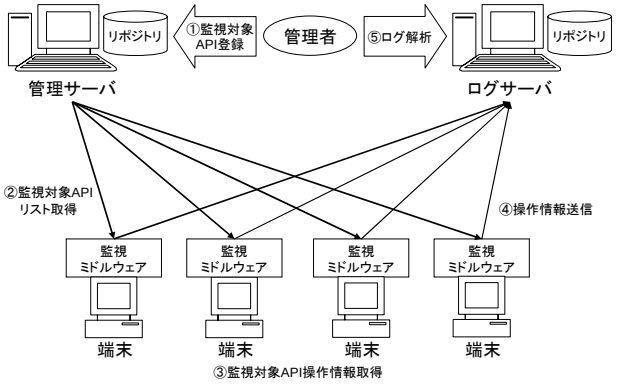

図２　実装例

① 管理者は監視対象 API を管理する管理サーバを用意し、監視対象となるアプリケーション名および API を登録し、さらに必要であれば監視対象のユーザ名やファイル名等を管理サーバのリポジトリへ登録する。

② API 監視のためのミドルウェアをネットワーク内の各端末に配備する。端末内の監視対象のアプリケーション起動時に管理サーバから監視対象の API リストを取得する。

③ 監視の条件下で監視対象の API が利用された場合、API 名とパラメータ、ユーザ名、端末名、日時などを取得する。

④ 取得した端末操作に関する情報をログサーバに送信し、ログサーバ内のリポジトリで一元管理する。

⑤ 管理者は収集されたログのリポジトリを解析することで、ネットワーク内でいつ誰がどの端末でどの情報に対してどのような操作を行ったかを解析する。

なお、各端末内の監視ミドルウェアは、システムコールや API の呼び出しをフックすることで実現可能である。

５．考察

本方式により次のことが確認された。

- 従来のネットワーク内や記録装置を監視する方式に比べて、監視対象を実際の保護すべきデータとすることで、より具体的かつ現実に則した情報流通の監視が可能となる。例えば特定のデータに対してのみ監視を行うといったことも新たに可能となった。
- データにどのような操作が行われたかを知ることが可能となったため、情報漏洩の際の追跡も容易になる。例えば、コピー・ペーストの API を監視することでそのファイル内の情報がどのように伝搬したかも知ることができた。
- ディスプレイ画面に映し出された情報がデジタルカメラやデジタルビデオを利用して画像・映像ファイルとして保存された場合には、ファイルの作成日時とログサーバに記録されたファイルオープン・クローズの日時とを照らし合わせることで、情報流出経路を絞り込むことができた。
- データへの操作を知ることでデータの漏洩だけでなく、改ざんについても検知することができた。

また、課題としては次の点がある。

- 一般に API の呼び出しは回数が多いため、重要な情報が埋没する可能性がある。例えば、高度なアプリケーションの場合、利用者の一つのアクションが数十、数百の API の呼び出しを伴うものがある。この中から重要な情報を取得するのは困難となる恐れがある。この課題を解決するために、ログのリポジトリ解析の際、クエリとその結果の表示の際に対処する必要がある。

６．おわりに

本稿では情報の流れに着目して、アプリケーションと OS の API の呼び出しを監視する新たな情報流通監視の方式について提案し、その実装例及び考察を示した。

今後は、各アプリケーションについて、どの API の呼び出しに対してどのようなパラメータを監視すればよいか精査を行っていく予定である。