セキュリティログ同期解析プログラムの開発

03H005 伊藤 一

目次

1 はじめに 2

2 概要 3

3 ログ閲覧によるセキュリティ対策の現状 4
- 3.1 セキュリティ対策 . . . 4
- 3.2 ログの役目 . . . 4
- 3.3 問題点 . . . 5
- 3.4 解決方法 . . . 6

4 システムの説明 7
- 4.1 SLSAP とは . . . 7
- 4.2 SLSAP の機能と特徴 . . . 7
- 4.3 データベースの構造 . . . 8
- 4.4 SLSAP の概略 . . . 9
- 4.5 SLSAP の使用例 . . . 10
- 4.6 データベースへのアクセス . . . 11
- 4.7 選択したテーブル情報の表示と保存 . . . 12
- 4.8 結合されたテーブルの表示 . . . 14
- 4.9 結合されたテーブルの表示と保存 . . . 15
- 4.10 保存された CSV 形式のデータ例 . . . 17

5 開発環境 18

6 まとめ 19
- 6.1 今後の課題 . . . 19

付録 A 付録 1 20
- A.1 SLSAP のプログラムのソース . . . 20

1　はじめに

今日までに、コンピュータの技術は急速に発展してきた。特に、インターネットの規模は世界規模にまで発展し、今日では経済や社会の基盤となり、なくてはならないものになっている。

その発展とともに、コンピュータを用いた犯罪も増加してきた。代表的なのは、コンピュータシステムを破壊するなど、コンピュータに対して被害を与えるプログラム、俗にウイルスプログラムと呼ばれるものがある。また、サーバに対してアクセスを集中させることで、サーバを不安定な状態にさせる DOS 攻撃などもある。

このようにネットワークは様々な危険にさらされてきた。その度に、それらの対策のためのセキュリティソフトウェアが開発された。例えば、ウイルス対策用ソフトウェアや Fire Wall 用ソフトウェアなどが挙げられる。そのソフトウェアの数や種類は、世界中に数多く存在し、利用されている。

しかし、その結果、セキュリティ対策ソフトが乱立し、各々が優れた機能を持っているが、それらを統合的に管理することが難しくなっている。それは、高いセキュリティを維持するためには、セキュリティ対策の要であるログ情報を迅速に把握する必要があるからだ。

本システムは、これらの対策ソフトの持つログ情報をまとめることで、セキュリティ担当者たちが、トラブルなどの原因を突き止めやすくし、一刻も早くセキュリティ対策に取り組めるようにすることを目的とする。

2　概要

セキュリティにかかわるログの情報は一概に膨大な量になりがちである。それは、企業や個人が、セキュリティ対策のために、大抵そのログを最低数ヵ月は保存しておくためである。そのため、トラブル発生時などで保存しておいたログを調べる場合、その膨大な量のために、とても手間のかかる作業になる。しかも、ソフトウェアが独自にログを保持することが多い。それらのログに対して目を通すことは、セキュリティを向上させることにつながるが、情報量が多く、なかなか難しい。

そこで、ログの情報をデータベースに登録し、複数のログ情報を一度に参照することができれば、効率良くログに目を通すことが可能になる。また、このことを Web ブラウザを使用して行うことにより、複雑な操作や知識を必要とせず、簡単な操作でシステムを利用することができ、的確な作業を迅速に行うことが可能になる。

このシステムを活用することで、バラバラにあるログを収集することができ、迅速に膨大な情報を把握することができる。また、膨大な量のログを扱うシステム管理者や、技術者の負担を減らすことができる。

3　ログ閲覧によるセキュリティ対策の現状

3.1　セキュリティ対策

コンピュータを運用する上で最も重要なことは、セキュリティ対策である。例えば、OS のアップデートやセキュリティ対策ソフトウェアの導入などが挙げられる。

しかし、それらの対策だけでは安全とは言い切れない。なぜなら、これらの対策は新しいトラブルが発生した後に、メーカー等が対策を取ることで成り立つ対策であり、そのトラブルが発生した直後では、ほとんど役にも立たないからである。

このようなトラブルに迅速に対応するためには、セキュリティ対策ソフトウェアや OS が保持しているログを管理、保存することが懸命である。

3.2　ログの役目

ログとは、コンピュータの利用状況やデータ通信の記録のことである。コンピュータ上での操作やデータの送受信が行われた日時と、操作の内容や送受信されたデータの中身などがログに記録される。このログを見ることで、ログを取得しているコンピュータの状態からユーザの動作内容といったことまで把握することが出来る。

このようにログを見ることで、ログを取っている対象に対し、どんなことが行われているのかを把握することができ、何か問題が発生した場合の原因究明の手がかりとなり、証拠にもなる。

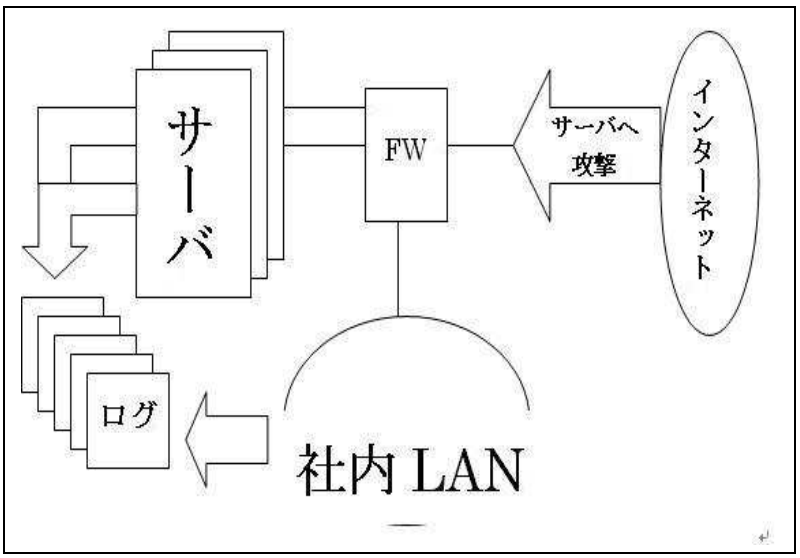


図 1　セキュリティにおけるログの出力状況

3.3　問題点

ログは、常にログの対象の状況などを記録している。例えば Fire Wall ソフトの場合、内部からや外部からのアクセスがあるたびに、そのときの時間や使用したポート番号などを記録している。

しかし、それらの記録をすべてログに残していくために、ログの量は多くなる。また、セキュリティ対策のために、そのログを最低数ヵ月は保存しておくことも、ログの量が膨大になりやすい要因となっている。

また、ログはソフトウェアごとに記録されており、それぞれが独立しているために、それらを比較したりすることは難しい。このことにより、ログに目を通す量が多くなり、多種多様なため複雑になり、適切な対策を取れるまでに時間がかかる。

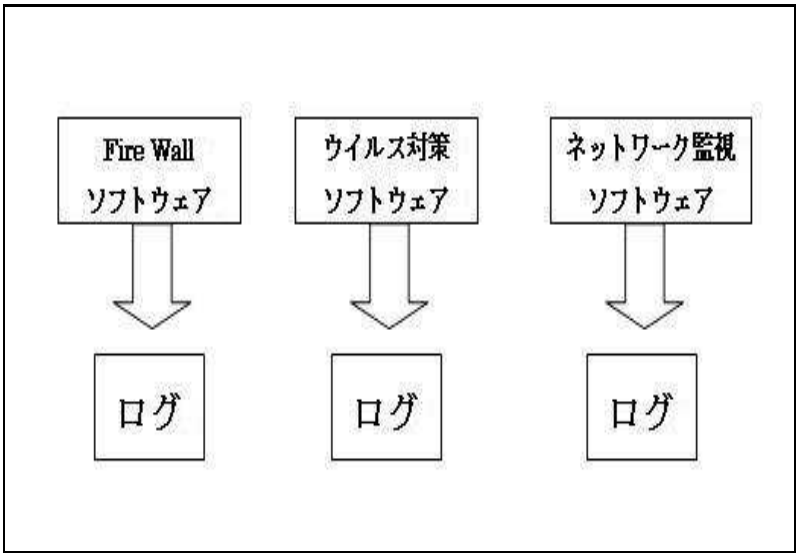


図 2　ログの出力例

3.4　解決方法

これらの問題は、ログの量と種類の多さが要因となっている。しかし、重要な情報を保存しているログの量を減らすことはできない。そこで、複数のログをまとめて表示することで、この負担を減らす方法を取ることにする。理由は、ログの種類の多さはあっても、そのログを出力する際には、共通の拡張子で出力されるからだ。

この問題を解決するためには、これらのログを一つにまとめることで、わかりやすく比較することで解決を目指す。ここでは、CSV(Comma Separated Value) 形式でまとめることで、複数のデータベースを扱うようにする。

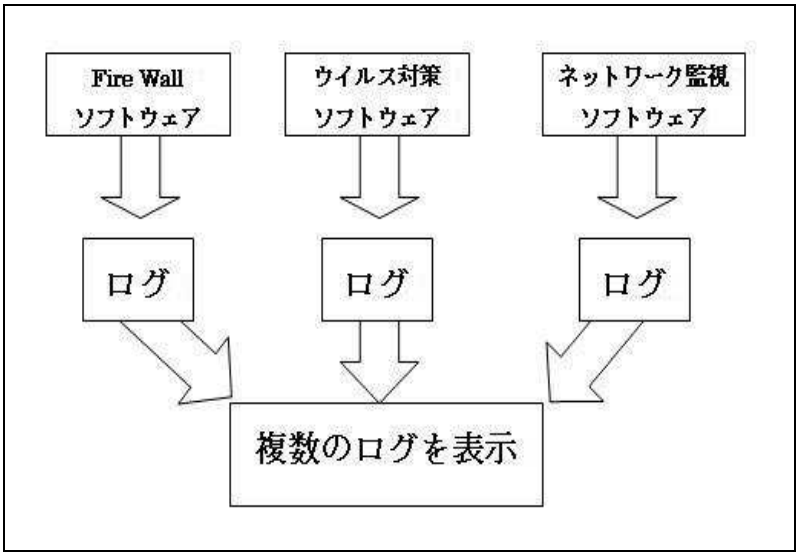


図 3　出力したログの収集イメージ

4　システムの説明

4.1　SLSAP とは

SLSAP とは、本研究の題でもある、セキュリティログ同期解析プログラム (Security Log Synchronous Analysis Program) の略称である。

このシステムは、複数のログをデータベースに登録し、それらを結合させた後、その結果を Web ブラウザ上に表示することを目的として作成した。このことにより、別々の場所にあるソフトウェアのログを調べなくても、Web ブラウザ上から操作を行うことができるので、簡単にまとめて調べることが出来る。

これ以降、このシステムのことを SLSAP と呼称する。

4.2　SLSAP の機能と特徴

SLSAP は、複数のログをデータベースに登録して、それらを結合させ、その結果を Web ブラウザ上に表示する機能を有している。SLSAP の特徴は、これらの機能を Web ブラウザで使用することが出来る点です。それぞれのツールでも使用することができますが、だれでも使えるように作成したいと考え、Web ブラウザを使用することにしました。さらに、簡易的な構造で作成することで、携帯電話のブラウザでも表示やコントロールできるようにした。

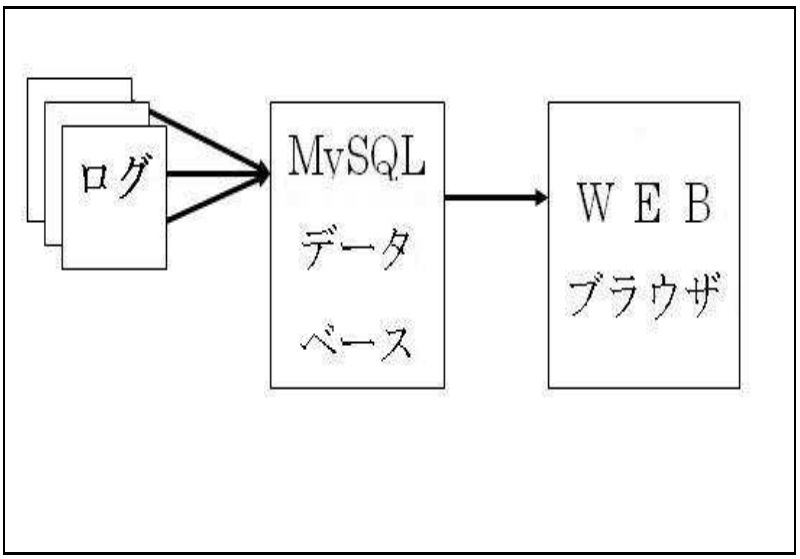


図 4　SLSAP の簡易イメージ図

4.3　データベースの構造

データベースとして MySQL を採用した。理由は、コンピュータに対する負荷が少ないことや、高い機動性と安定性があるからです。また、PHP との互換性もよく、CGI として稼働させることに優れている。

MySQL のデータベースを構築する際には、必ず主キーを入れなければならない。それは、データベースのテーブルを結合する際に、識別するのに必要になるからである。主キーは、オートインクリメントを使用する。オートインクリメントを使用することで、重複しない数値を主キーに登録することが出来る。このシステムで使用するデータベースのテーブル構成例を図 5 に示す。

```
mysql> SHOW COLUMNS FROM VBFW_T;
+-------+--------------+------+-----+---------+----------------+
| Field | Type         | Null | Key | Default | Extra          |
+-------+--------------+------+-----+---------+----------------+
| PK    | int(11)      | NO   | PRI | NULL    | auto_increment |
| 1ST   | varchar(100) | YES  |     | NULL    |                |
| 2ND   | varchar(50)  | YES  |     | NULL    |                |
| 3RD   | varchar(20)  | YES  |     | NULL    |                |
| 4TH   | varchar(50)  | YES  |     | NULL    |                |
| 5TH   | varchar(100) | YES  |     | NULL    |                |
| 6TH   | varchar(100) | YES  |     | NULL    |                |
| 7TH   | varchar(100) | YES  |     | NULL    |                |
| 8TH   | varchar(100) | YES  |     | NULL    |                |
| 9TH   | varchar(100) | YES  |     | NULL    |                |
| 10TH  | varchar(30)  | YES  |     | NULL    |                |
+-------+--------------+------+-----+---------+----------------+
11 rows in set (0.01 sec)
```

図 5　MySQL のテーブル構成例

4.4　SLSAP の概略

この SLSAP の概略図を図 6 に示す。SLSAP は slsap.html が TOP ページである。ここから一つのデータベースに格納されてあるテーブルを見るときは choiceload.php を、結合させたデータベースのテーブル結果を表示するときは choicejoin.php を選択する。詳細は次のページ以降で説明を行う。

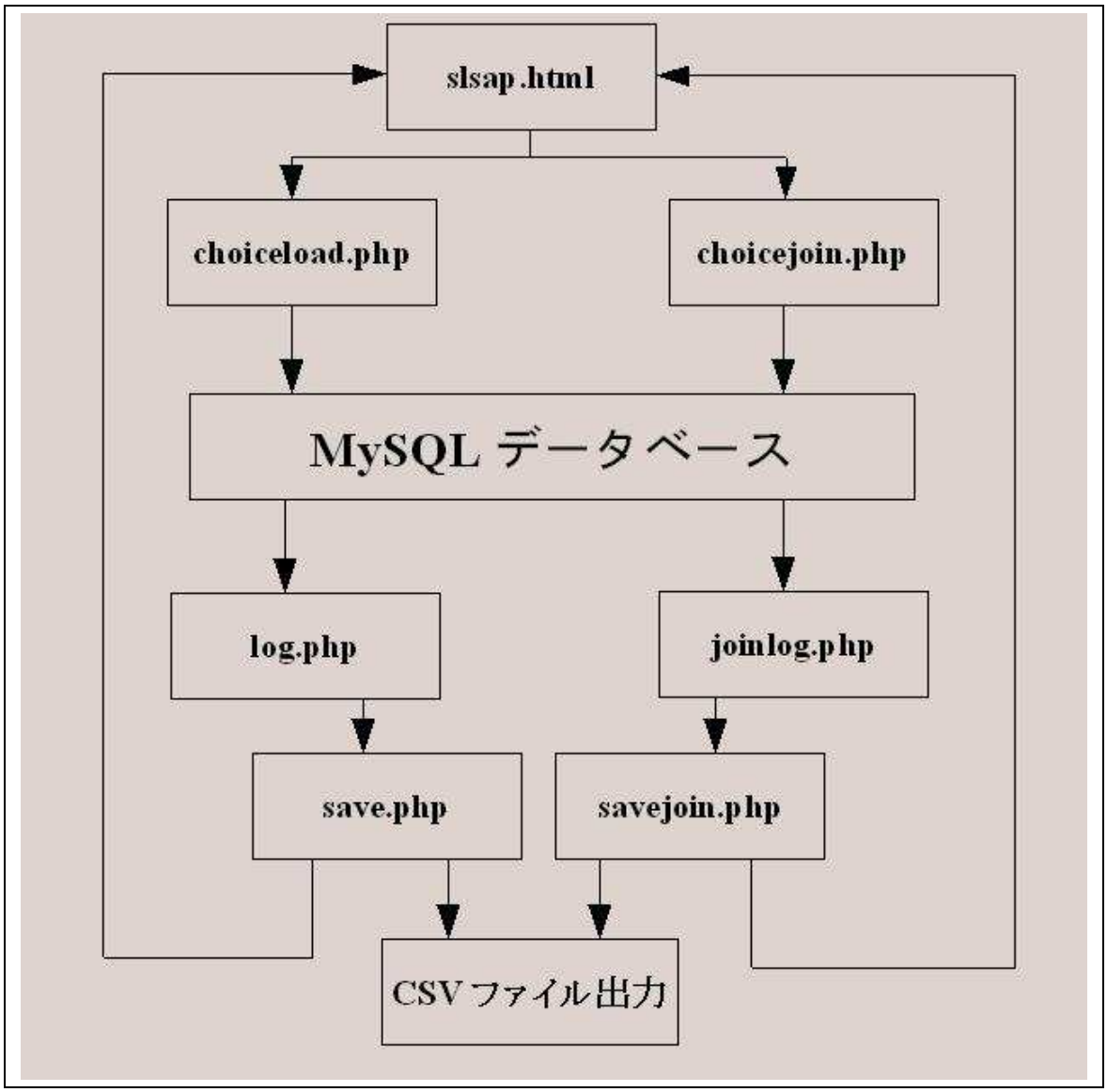


図 6　SLSAP の概略図

4.5　SLSAP の使用例

図 7 に表した通り、これが SLSAP のメインメニューとなる。携帯電話等のモバイル端末からでも操作可能にするために、シンプルに構成した結果、このようなメニューとなった。

「Data Load」を選択すると、データベースへアクセスしてログの情報をブラウザで表示するシステムへ移行する。また、CSV(Comma Separated Value) 形式のデータのダウンロードも可能である。また、「Join Data Load」を選択すると、データベースへアクセスして、複数のログの情報を持つテーブルを左外部結合した状態で、ブラウザから閲覧、ダウンロードすることができる。

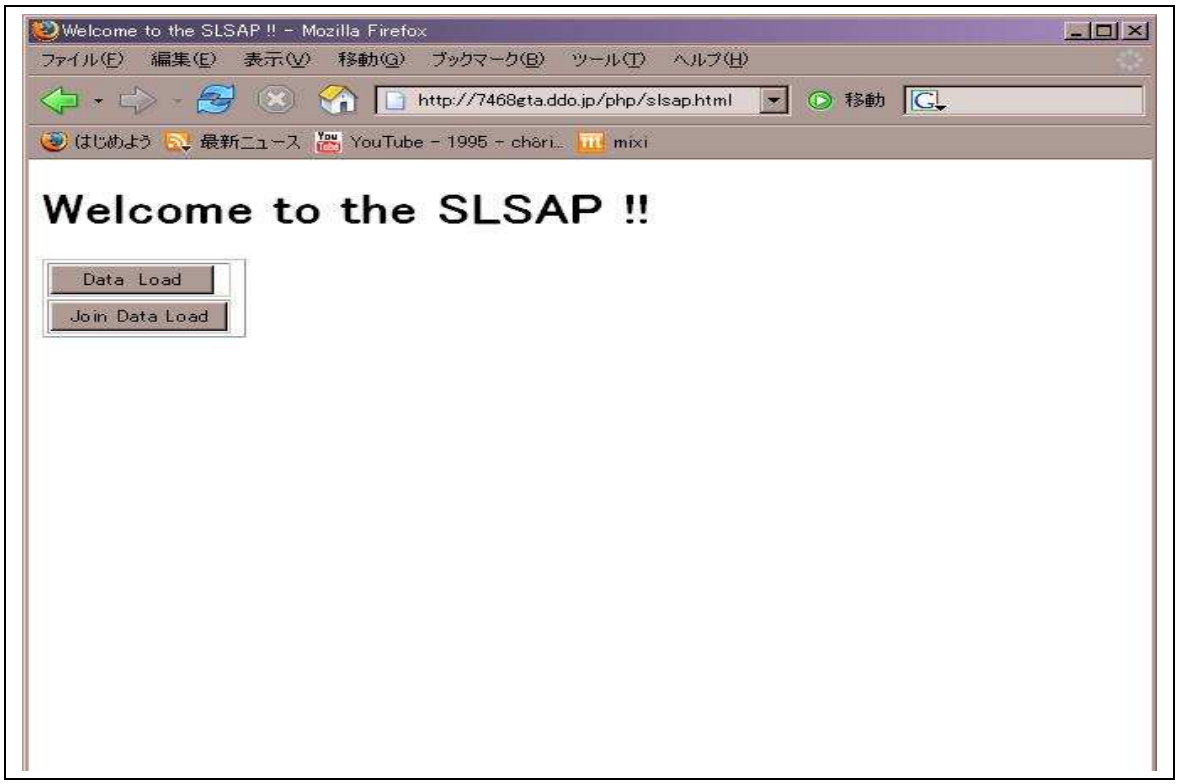


図 7　SLSAP の起動画面

4.6　データベースへのアクセス

まず、「Data Load」を選択する。このページには、入力フォームがいくつかある。これらは、ログを保存しているデータベースである MySQL にアクセスするために、必要な情報を入力をするためのものである。

まず、それぞれのフォームに MySQL の情報を入力する。データベース名から入力していき、テーブル名まで入力ができれば「次へ」のボタンを押す。入力したフォームの内容をまとめて消去するときは、「リセット」を押す。

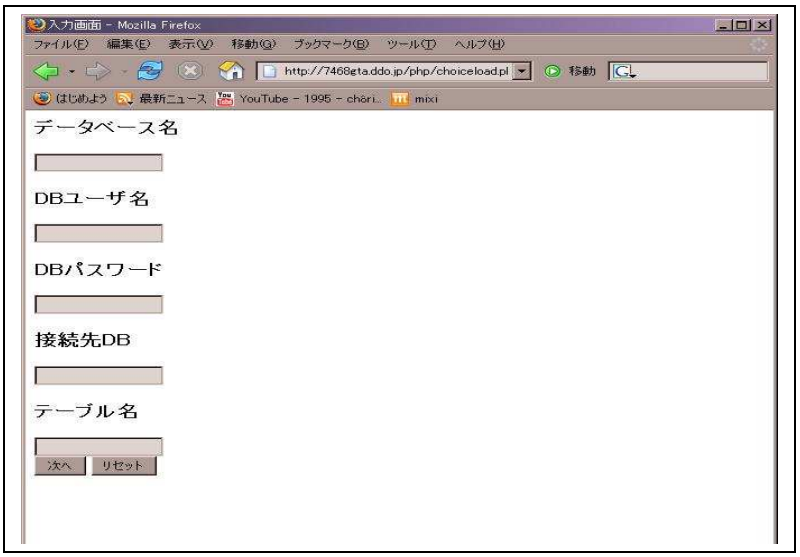


図 8　データベース情報入力画面

「Data Load」のボタンを選択すると、図 8 のような画面に移る。ここでは、指定された項目をその下のフォームに MySQL に関する情報を書き込む。入力時、大文字と小文字を識別するので気を付ける。図 9 は、入力の例である。

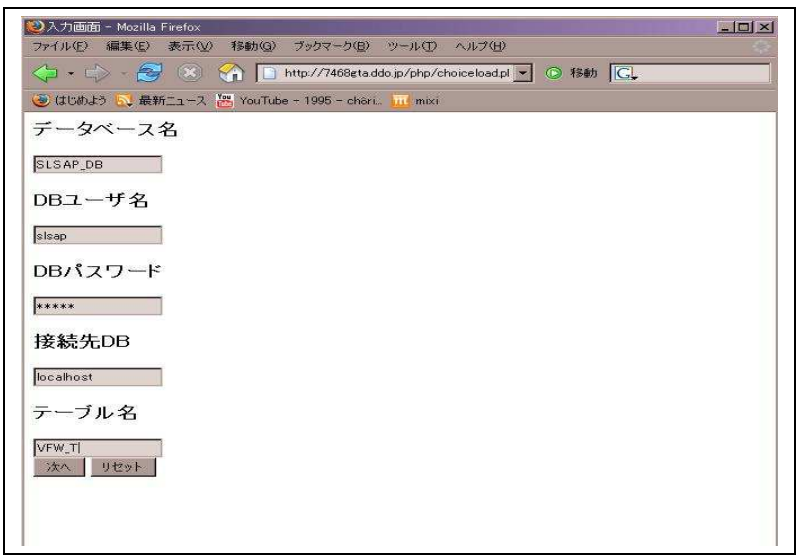
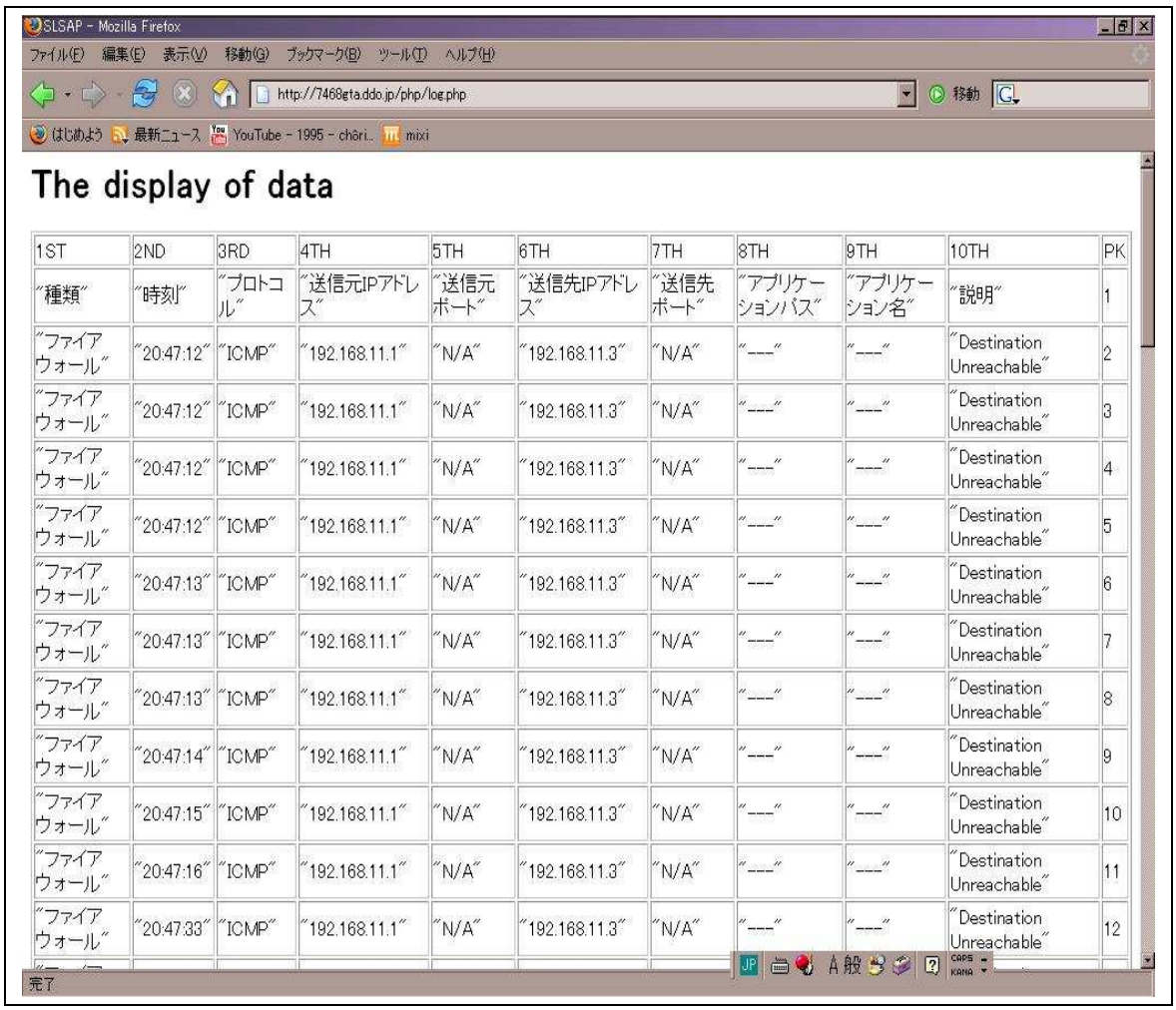


図 9　データベース情報の入力例

4.7　選択したテーブル情報の表示と保存

前画面での入力が正しく行われた場合、ブラウザ上に MySQL に格納されたデータが図 10 のように表示される。



1ST	2ND	3RD	4TH	5TH	6TH	7TH	8TH	9TH	10TH	PK
″種類″	″時刻″	″プロトコル″	″送信元IPアドレス″	″送信元ポート″	″送信先IPアドレス″	″送信先ポート″	″アプリケーションパス″	″アプリケーション名″	″説明″	1
″ファイアウォール″	″20:47:12″	″ICMP″	″192.168.11.1″	″N/A″	″192.168.11.3″	″N/A″	″---″	″---″	″Destination Unreachable″	2
″ファイアウォール″	″20:47:12″	″ICMP″	″192.168.11.1″	″N/A″	″192.168.11.3″	″N/A″	″---″	″---″	″Destination Unreachable″	3
″ファイアウォール″	″20:47:12″	″ICMP″	″192.168.11.1″	″N/A″	″192.168.11.3″	″N/A″	″---″	″---″	″Destination Unreachable″	4
″ファイアウォール″	″20:47:12″	″ICMP″	″192.168.11.1″	″N/A″	″192.168.11.3″	″N/A″	″---″	″---″	″Destination Unreachable″	5
″ファイアウォール″	″20:47:13″	″ICMP″	″192.168.11.1″	″N/A″	″192.168.11.3″	″N/A″	″---″	″---″	″Destination Unreachable″	6
″ファイアウォール″	″20:47:13″	″ICMP″	″192.168.11.1″	″N/A″	″192.168.11.3″	″N/A″	″---″	″---″	″Destination Unreachable″	7
″ファイアウォール″	″20:47:13″	″ICMP″	″192.168.11.1″	″N/A″	″192.168.11.3″	″N/A″	″---″	″---″	″Destination Unreachable″	8
″ファイアウォール″	″20:47:14″	″ICMP″	″192.168.11.1″	″N/A″	″192.168.11.3″	″N/A″	″---″	″---″	″Destination Unreachable″	9
″ファイアウォール″	″20:47:15″	″ICMP″	″192.168.11.1″	″N/A″	″192.168.11.3″	″N/A″	″---″	″---″	″Destination Unreachable″	10
″ファイアウォール″	″20:47:16″	″ICMP″	″192.168.11.1″	″N/A″	″192.168.11.3″	″N/A″	″---″	″---″	″Destination Unreachable″	11
″ファイアウォール″	″20:47:33″	″ICMP″	″192.168.11.1″	″N/A″	″192.168.11.3″	″N/A″	″---″	″---″	″Destination Unreachable″	12

図 10　選択されたテーブルの表示例

また、ログ情報の一番下に、図 11 のように「save」のボタンと「TOP」のリンクがある。「TOP」を選択すると、最初のページ (slsap.html) に戻る。「save」ボタンを選択すると、図 12 のような画面が出力される。この画面では、今表示しているデータを CSV 形式で保存することが出来る。CSV 形式で保存する際、デフォルトで保存する日時を出力する CSV データの名前にしている。これは、保存した日を分かりやすくしているためだ。

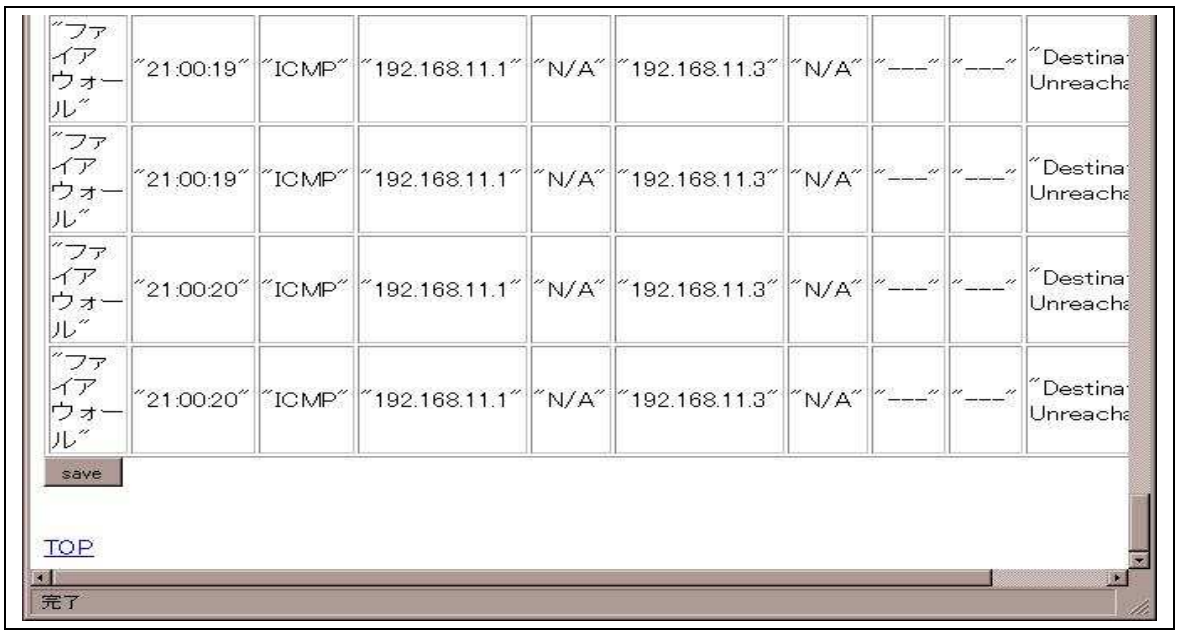


図 11　データベース情報の一番下の項目

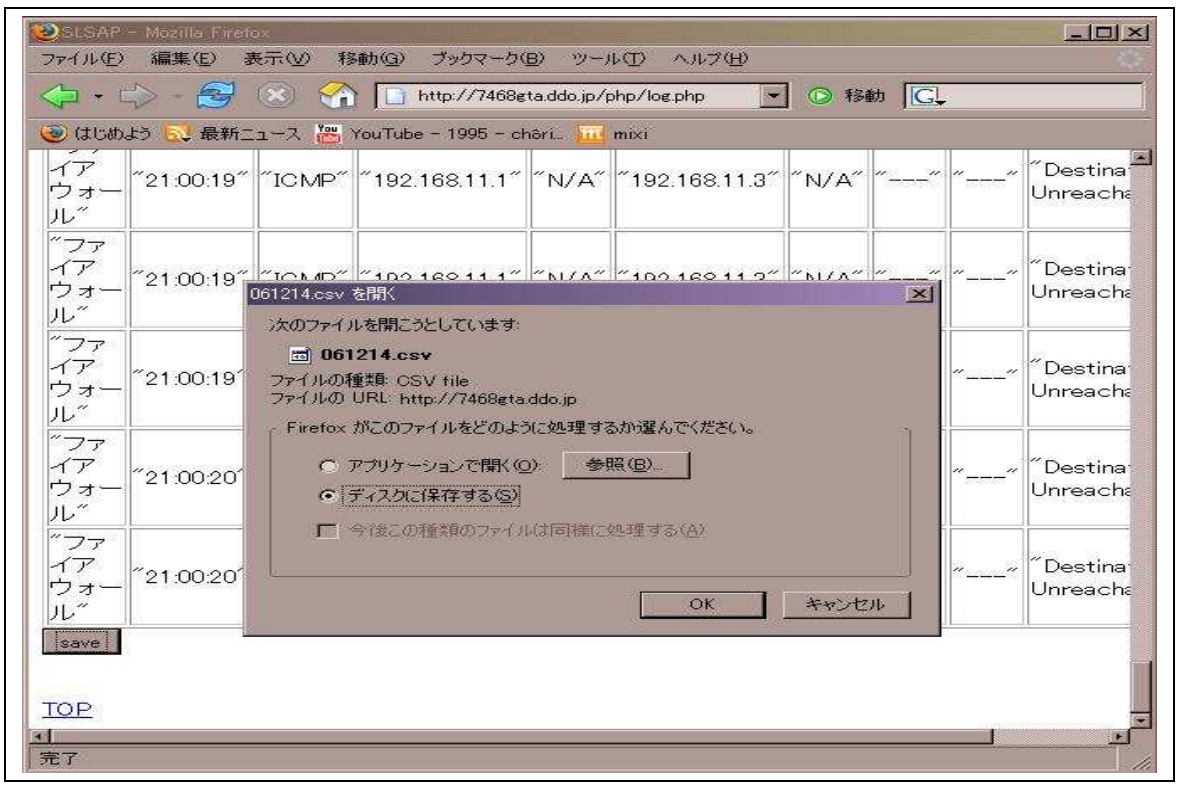


図 12　保存する方法の選択画面

4.8　結合されたテーブルの表示

先ほどは、一つのデータベースの一つのテーブルを扱うだけであった。ここでは、一つのデータベース上にある二つのテーブルを扱う。最初のページ (slsap.html) で「Join Data Load」を選択する。すると、図 13 のようなフォーム入力画面へ移行する。先ほどの入力画面と異なる点は、ブラウザに出力するデータベースのテーブル名と、それに結合させたいデータベースのテーブル名の両方を入力する点である。図 14 は、このページの入力例である。テーブル名は、Fire Wall のログデータ、結合させるテーブル名は、ウイルスログのデータを選択した。

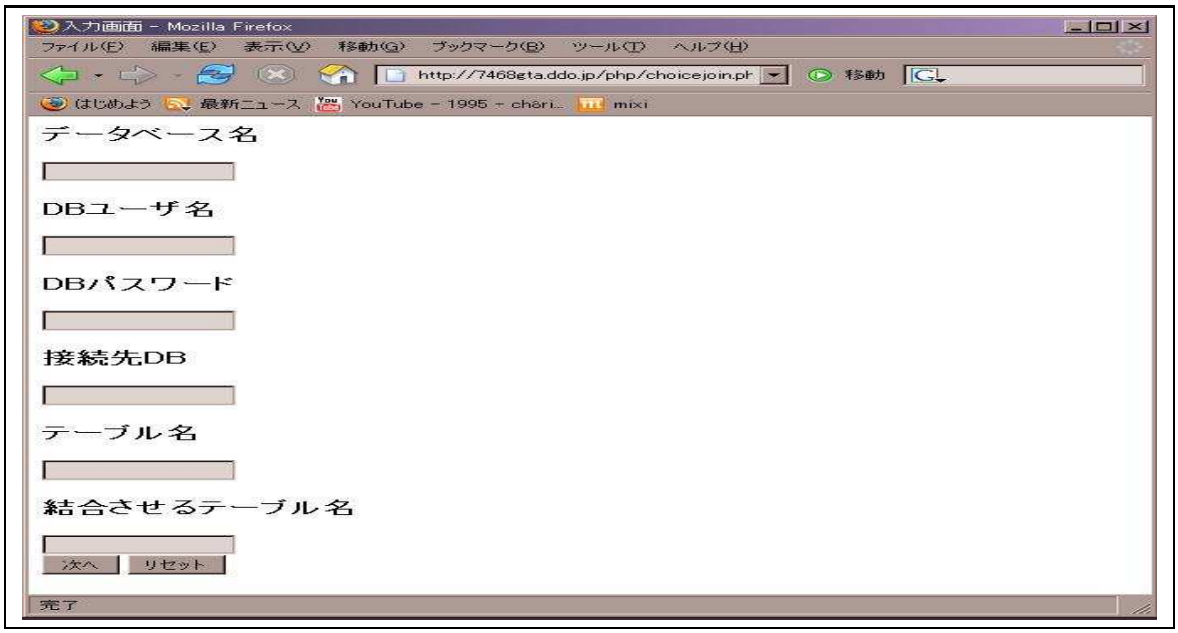


図 13　データベース情報の入力画面

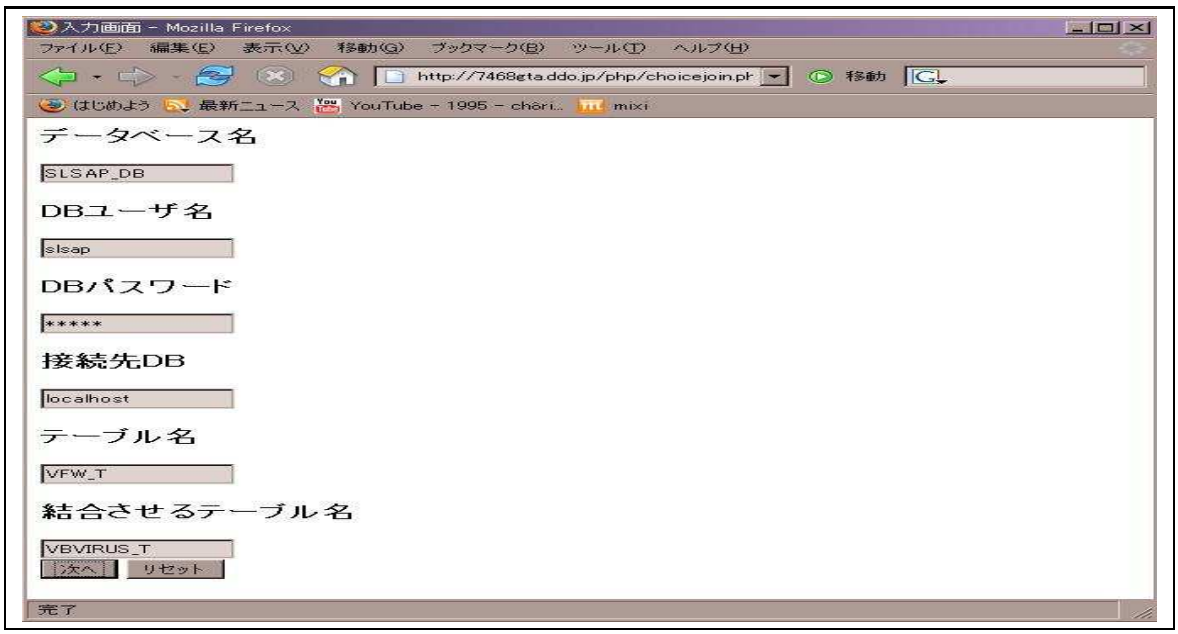


図 14　データベース情報の入力例

4.9　結合されたテーブルの表示と保存

Fire Wall のログデータとウイルスログのデータを結合した状態で、ブラウザに出力した。PK は主キーのことであり、これにより二つのテーブルが結合されているのが図 15 より分かる。また、たとえ主キー同士が一致していなくても、片側だけでも表示する左外部結合により、ログ情報が欠落することがないようにしている。

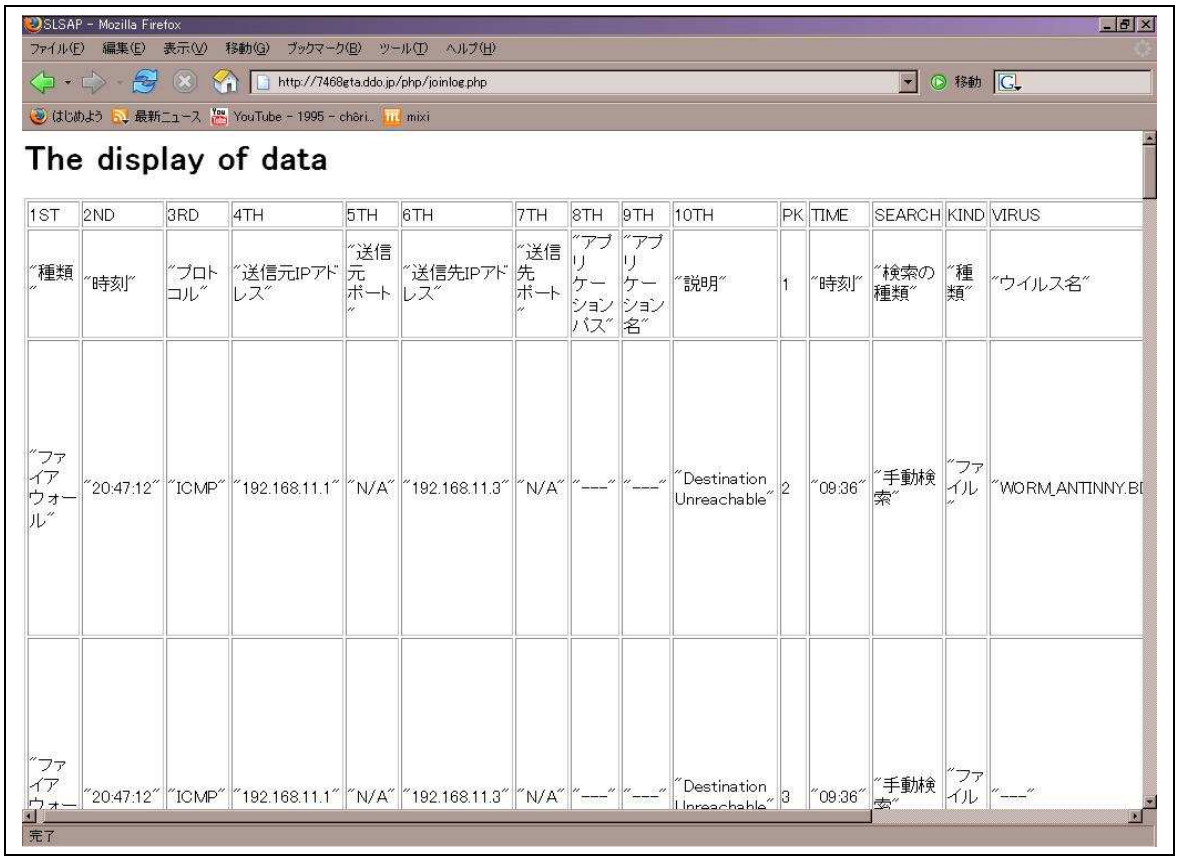


図 15　結合されたテーブル情報の表示

また、図 11 と同じく、ログ情報の一番下に、図 16 のように「save」のボタンと「TOP」のリンクがある。「TOP」を選択すると、最初のページ (slsap.html) に戻る。「save」ボタンを選択すると、図 17 のような画面が出力される。この画面では、今表示している結合データを CSV 形式で保存することが出来る。CSV 形式で保存する際、デフォルトで保存する日時と、「join」を出力する CSV データの名前にしている。これも、保存した日を分かりやすくしているためと、結合したテーブルのデータと区別するためである。

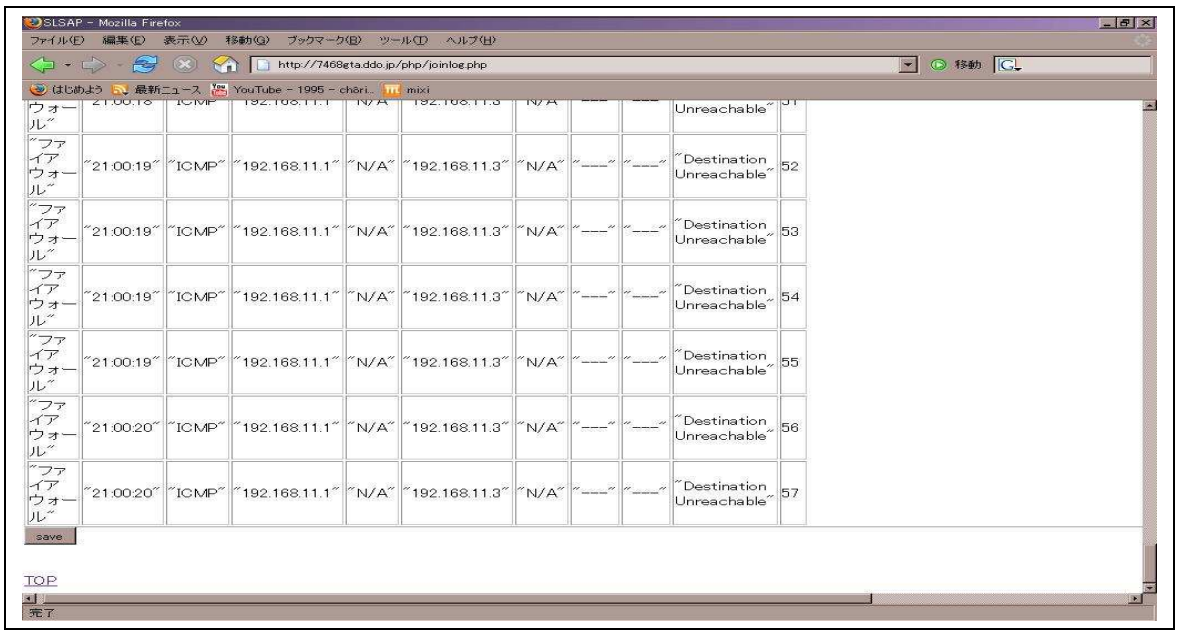


図 16　結合されたテーブル情報の表示

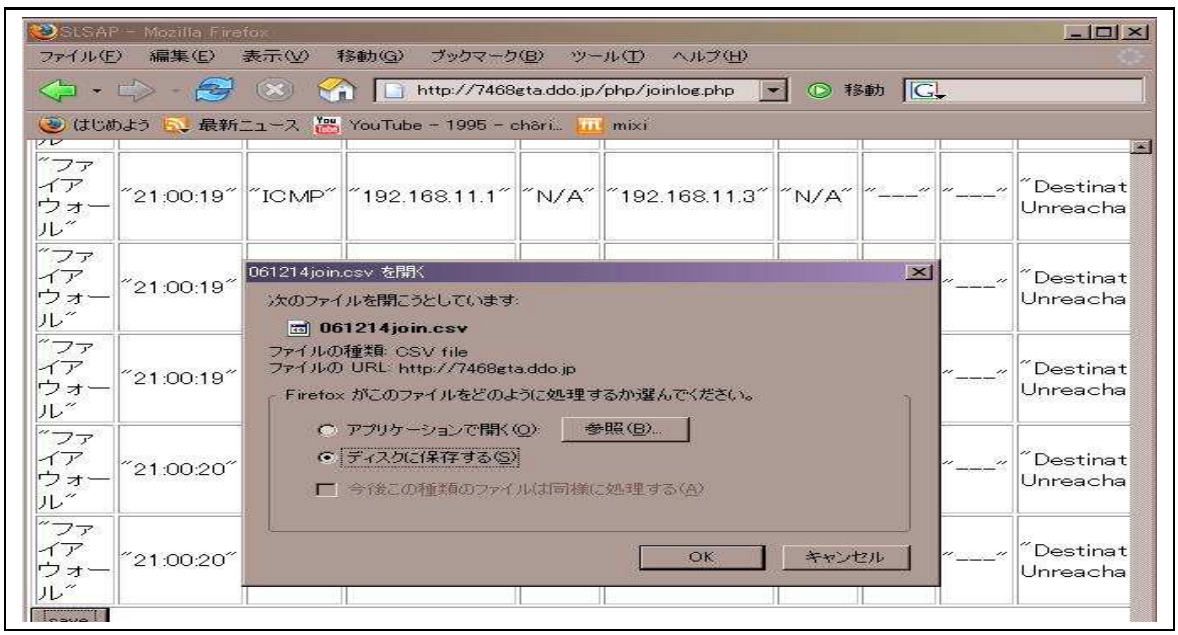
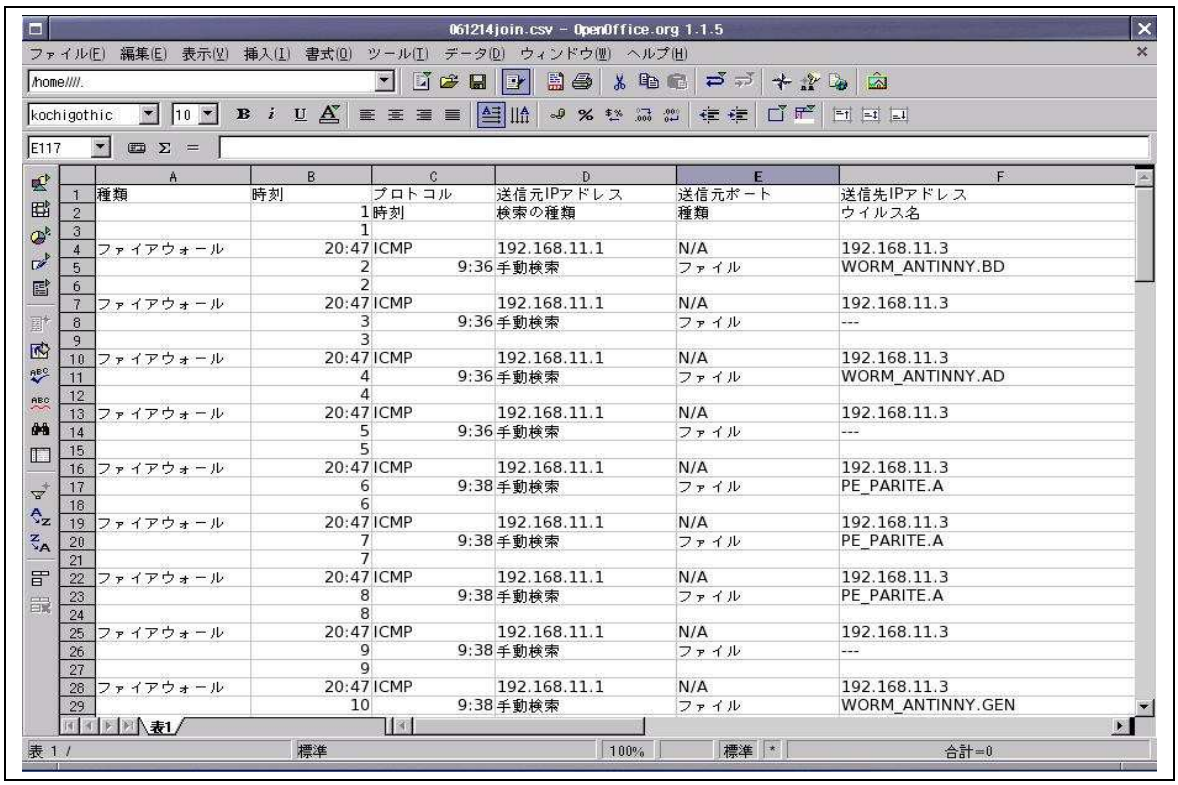


図 17　結合されたテーブル情報の保存の表示

4.10　保存された CSV 形式のデータ例

「save」を選択した場合、CSV 形式のデータをダウンロードできる。ダウンロード完了後、図 18 のようにテキストエディタ等で表示、編集することが出来る。



	A	B	C	D	E	F
1	種類	時刻	プロトコル	送信元IPアドレス	送信元ポート	送信先IPアドレス
2		1	時刻	検索の種類	種類	ウイルス名
3		1				
4	ファイアウォール	20:47	ICMP	192.168.11.1	N/A	192.168.11.3
5		2	9:36	手動検索	ファイル	WORM_ANTINNY.BD
6		2				
7	ファイアウォール	20:47	ICMP	192.168.11.1	N/A	192.168.11.3
8		3	9:36	手動検索	ファイル	---
9		3				
10	ファイアウォール	20:47	ICMP	192.168.11.1	N/A	192.168.11.3
11		4	9:36	手動検索	ファイル	WORM_ANTINNY.AD
12		4				
13	ファイアウォール	20:47	ICMP	192.168.11.1	N/A	192.168.11.3
14		5	9:36	手動検索	ファイル	---
15		5				
16	ファイアウォール	20:47	ICMP	192.168.11.1	N/A	192.168.11.3
17		6	9:38	手動検索	ファイル	PE_PARITE.A
18		6				
19	ファイアウォール	20:47	ICMP	192.168.11.1	N/A	192.168.11.3
20		7	9:38	手動検索	ファイル	PE_PARITE.A
21		7				
22	ファイアウォール	20:47	ICMP	192.168.11.1	N/A	192.168.11.3
23		8	9:38	手動検索	ファイル	PE_PARITE.A
24		8				
25	ファイアウォール	20:47	ICMP	192.168.11.1	N/A	192.168.11.3
26		9	9:38	手動検索	ファイル	---
27		9				
28	ファイアウォール	20:47	ICMP	192.168.11.1	N/A	192.168.11.3
29		10	9:38	手動検索	ファイル	WORM_ANTINNY.GEN

図 18　テキストエディタによる CSV ファイルの表示

5　開発環境

このシステムの開発のために使用したプログラミング言語や環境は以下の通りである。

- HTML(Hyper Text Making Language)
  - HTML は WWW(World Wide Web) 上に、情報を発信するための Web ページを記述するためのマークアップ言語である。テキストファイルの間に他のテキストファイルや画像・音声ファイルなどを示すポインタを埋め込む手法が採用されている。
- Apache
  - Apache は、NCSA httpd 1.3 をベースに開発された Web サーバである。バグを修正したり、新しい機能を追加するためのパッチの寄せ集めとした開発されたためためこの名がついた。フリーソフトウェアとして配布され、世界で最も使用されている Web サーバである。
- PHP(Hypertext Preprocesser)
  - PHP は、HTML 中にプログラムを記述して、サーバ側で実行させるスクリプト言語で、動的な Web ページの作成に非常に適している。また、PHP は Apache のモジュールとして動作させることができる。このことにより、マシンの負担を小さくすることができ、起動を速くすることができるので、サーバを運用する上では最適である。これは、CGI と実行速度を比べると明らかである。また、データベースとの連携も優れており、今回使用した MySQL とも、相性がよい。これは PHP が、データベースをとの連携を想定して作られているからであり、それに基づき様々な関数が用意されているためである。
- MySQL
  - MySQL は、フィンランドの Michael Widenius によって 1995 年に開発された。現在 MySQL の開発・サポートは専門の企業 MySQL AB で行われている。特徴は検索の高速性に重点を絞った設計になっていることで、商用 RDBMS に比べ、処理を遅くする機構を大幅に省いた反面、大量のデータを高速に検索することができる。機能をある程度排除し、高速性を重視しているため、サーバ用途などに使用されている。オープンソースであるが、高速性と堅牢性に定評がある。

6　まとめ

本開発では、ログの情報をまとめることができるシステムの開発目的とした。開発システムとしては、PHP と Apache、MySQL を使用することで、Web ブラウザを操作するだけで、ログをまとめることを可能にした。Web ブラウザの対応については、パソコンで一般的に使用されている Web ブラウザや一部の携帯電話の Web ブラウザでも利用可能であることが確認された。

セキュリティに関しては、ローカルネットワークで動作することを前提としたため、ほとんど対策は施してない。しかし、MySQL のデータベースにアクセスする際には、データベース名からテーブル名、ユーザパスワードまで入力をしなければならないので、データベースに関しては、ある程度のセキュリティを保つことができている。

6.1　今後の課題

今後の課題としては、システムの拡張性とセキュリティを高めることが重要になります。今回は、軽快に動くことを目的としたため、余計な機能を排除してシンプルなシステムとなる。このシステムは Web ブラウザで利用可能なので、インターネットを経由しても利用可能ですが、セキュリティ対策があまり施されていないので、セキュリティ対策が必要だと考えらる。

謝辞

本研究を進めていく上で、御指導及び御協力頂いた大垣斉講師、藤井信夫教授、ならびに研究に協力して下さった友人たちに、深く感謝いたします。

参考文献

[1] 堀江美彦 MySQL5 構築ガイド オープンソース高速データベースと Apache、PHP、Perl、Java を使った Web+DB プログラミング環境の構築株式会社シナノ 2006 年 ISBN4-7980-1275-0

[2] 田中ナルミ 阿部忠光 MySQL コマンドブックソフトバンクパブリッシング株式会社 2005 年 ISBN4-7973-3082-1

[3] 最新 完全無敵のリナックス KNOPPIX で Linux はじめる Linux 入門株式会社秀和システム 2005 年 ISBN4-7980-1139-8

[4] インターネットサービス KOMONET (http://www.komonet.ne.jp/) 12 月 18 日現在

付録 A　付録 1

A.1　SLSAP のプログラムのソース

slsap.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Welcome to the SLSAP !!</title>
  </head>

  <body>
    <p>
    <h1>
      Welcome to the SLSAP !!
    </h1>
    </p>
    <table border="1">

<!-- データベースから読み出し -->
      <form action="/php/choiceload.php" method="post">
<tr>
  <td><input type="submit" value="   Data  Load   "></td>
  <td></td>
</tr>
      </form>

<!-- データベースから結合したものを読み出し -->
      <form action="/php/choicejoin.php" method="post">
<tr>
  <td><input type="submit" value="Join Data Load"></td>
  <td>
</td>

</tr>
      </form>
    </table>
  </body>
</html>
```

choiceload.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>入力画面</title>
  </head>

  <body>
    <h3>データベース名</h3>
    <FORM method="POST" action="log.php">
    <INPUT size"80" type="text" name="DBName"><BR>

    <h3>DB ユーザ名</h3>
    <FORM method="POST" action="log.php">
    <INPUT size"80" type="text" name="UserName"><BR>

    <h3>DB パスワード</h3>
    <FORM method="POST" action="log.php">
    <INPUT size"80" type="password" name="UserPass"><BR>

    <h3>接続先 DB</h3>
    <FORM method="POST" action="log.php">
    <INPUT size"80" type="text" name="ToDB"><BR>

   <h3>テーブル名</h3>
    <FORM method="POST" action="log.php">
    <INPUT size"80" type="text" name="UserTable"><BR>


    <INPUT type="submit" name="submit" value="次へ">
    <INPUT type="reset" name="reset" value="リセット">

    </FORM>
  </body>
</html>
```

log.php

```
<html>
<head>
<title>SLSAP</title>

</head>
```

```
<?php

/* 前のページのフォームから得た情報を変数に格納 */
/* MySQL DB DBNAME USER PASS etc..*/

$DB_NAME = $DBName;
$DB_USER = $UserName;
$DB_PASS = $UserPass;
$DB_TO = $ToDB;
$DB_TABLE = $UserTable;

/* MYSQL CONNECT */
$link = mysql_connect($DB_TO, $DB_USER, $DB_PASS)
or die("MYSQL disconnect");

/* データベースを選択 */
mysql_select_db($DB_NAME)
or die("DB disconnect");

/* SQL 文を変数に格納する */
$query="SELECT * FROM $DB_TABLE";

/* SQL 文を実行 */
$rs = mysql_db_query($DB_NAME, $query);

/* 列数を計算して取得 */
$num = mysql_num_fields($rs);

?>
<h1>
The display of data
</h1>
<table border="1">

<p>
<p>
<p>
<p>

<?php

/* 列数だけループしてデータを表示 */
```

```
for ($i = 0;$i < $num; $i++){
print("<td>".mysql_field_name($rs,$i)."</td>");
}

/* Roop for display of data */
while ($row = mysql_fetch_array($rs)){
print("<tr>");
for($j = 0;$j < $num;$j++){
print("<td>".$row[$j]."</td>");
}
print("</tr>");
}

print("</table>");


/* MySQL dissconnect */
mysql_close($link);

/* 保存用ページへ変数を送る */
  print "<form action = save.php method = post>";
    print "<input type = hidden name = DB_NAME value = \"$DB_NAME\">";
    print "<input type = hidden name = DB_USER value = \"$DB_USER\">";
    print "<input type = hidden name = DB_PASS value = \"$DB_PASS\">";
    print "<input type = hidden name = DB_TO value = \"$DB_TO\">";
    print "<input type = hidden name = DB_TABLE value = \"$DB_TABLE\">";
    print "<input type = submit name = submit value = save>";
  print "</form>";
?>

<BR>
<a href="slsap.html">TOP</a>

</body>
</html>

save.php

<?php

header("Content-Type: application/octet-stream");

/* Save File Name*/
```

```
header("Content-Disposition: attachment; filename=".date(ymd).".csv");

/* 前のページのフォームから得た情報を変数に格納 */
/* MySQL DB DBNAME USER PASS */

$DB_NAME2 = $DB_NAME;
$DB_USER2 = $DB_USER;
$DB_PASS2 = $DB_PASS;
$DB_TO2 = $DB_TO;
$DB_TABLE2 = $DB_TABLE;

/* DB connect */
$link = mysql_connect($DB_TO2, $DB_USER2, $DB_PASS2)
or die("Mysql disconnect");

/* SQL 文を変数に格納する */
$query2="SELECT * FROM $DB_TABLE2";

/* SQL 文を実行 */
$rs = mysql_db_query($DB_NAME2, $query2);

/* 行数と列数を計算して取得 */

$fields = mysql_num_fields($rs);
$rows = mysql_num_rows($rs);

/* 計算して取得した行数、列数だけループ */
/* ループの間データの表示を続ける */

for($i = 0;$i < $rows;$i++){
   for($j = 0;$j < $fields;$j++){
print(mysql_result($rs,$i,$j));
/* Insert canma */
if($j < $fields -1)
print(",");
}

print("\n");
}

/* MySQL dissconnect */
mysql_close($link);
```

```
?>

choicejoin.php

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>入力画面</title>
  </head>

  <body>
    <h3>データベース名</h3>
    <FORM method="POST" action="joinlog.php">
    <INPUT size"80" type="text" name="DBName"><BR>

    <h3>DB ユーザ名</h3>
    <FORM method="POST" action="log.php">
    <INPUT size"80" type="text" name="UserName"><BR>

    <h3>DB パスワード</h3>
    <FORM method="POST" action="log.php">
    <INPUT size"80" type="password" name="UserPass"><BR>

    <h3>接続先 DB</h3>
    <FORM method="POST" action="log.php">
    <INPUT size"80" type="text" name="ToDB"><BR>

   <h3>テーブル名</h3>
    <FORM method="POST" action="log.php">
    <INPUT size"80" type="text" name="UserTable"><BR>

   <h3>結合させるテーブル名</h3>
    <FORM method="POST" action="log.php">
    <INPUT size"80" type="text" name="UserTable2"><BR>

    <INPUT type="submit" name="submit" value="次へ">
    <INPUT type="reset" name="reset" value="リセット">

    </FORM>
  </body>
</html>
```

joinlog.php

```
<html>
<head>


<title>SLSAP</title>

</head>

<?php
/* 前のページのフォームから得た情報を変数に格納 */
/* MySQL DB DBNAME USER PASS etc..*/

$DB_NAME = $DBName;
$DB_USER = $UserName;
$DB_PASS = $UserPass;
$DB_TO = $ToDB;
$DB_TABLE = $UserTable;
$DB_TABLE2 = $UserTable2;



/* MYSQL CONNECT */
$link = mysql_connect($ToDB, $DB_USER, $DB_PASS)
or die("MYSQL disconnect  orz");

/* データベースを選択 */
mysql_select_db($DB_NAME)
or die("DB disconnect  orz");

/* SQL 文を変数に格納する */
$query="SELECT * FROM $DB_TABLE LEFT OUTER JOIN $DB_TABLE2 ON $DB_TABLE.PK = $DB_TABLE2.PK";

/* SQL 文を実行 */
$rs = mysql_db_query($DB_NAME, $query);

/* 列数を計算して取得 */
$num = mysql_num_fields($rs);



?>
<h1>
The display of data
```

```
</h1>
<table border="1">

<p>
<p>
<p>
<p>

<?php

/* 列数だけループしてデータを表示 */
for ($i = 0;$i < $num; $i++){
print("<td>".mysql_field_name($rs,$i)."</td>");
}

while ($row = mysql_fetch_array($rs)){
print("<tr>");
for($j = 0;$j < $num;$j++){
print("<td>".$row[$j]."</td>");
}
print("</tr>");
}

print("</table>");


/* MySQL dissconnect */
mysql_close($link);

/* 保存用ページへ変数を送る */
  print "<form action = joinsave.php method = post>";
    print "<input type = hidden name = DB_NAME value = \"$DB_NAME\">";
    print "<input type = hidden name = DB_USER value = \"$DB_USER\">";
    print "<input type = hidden name = DB_PASS value = \"$DB_PASS\">";
    print "<input type = hidden name = DB_TO value = \"$DB_TO\">";
    print "<input type = hidden name = DB_TABLE value = \"$DB_TABLE\">";
    print "<input type = hidden name = DB_TABLE2 value = \"$DB_TABLE2\">";
    print "<input type = submit name = submit value = save>";
  print "</form>";


?>
```

```
<BR>
<!-- TOP ページへ戻る -->
<a href="slsap.html">TOP</a>

</body>
</html>
```

joinsave.php

```
<?php

header("Content-Type: application/octet-stream");

/* Decide Save File Name*/
header("Content-Disposition: attachment; filename=".date(ymd)."join.csv");

/* 前のページのフォームから得た情報を変数に格納 */
/* MySQL DB DBNAME USER PASS */

$DB_NAMEj2 = $DB_NAME;
$DB_USERj2 = $DB_USER;
$DB_PASSj2 = $DB_PASS;
$DB_TOj2 = $DB_TO;
$DB_TABLEj = $DB_TABLE;
$DB_TABLEj2 = $DB_TABLE2;

/* DB connect */
$link = mysql_connect($DB_TOj2, $DB_USERj2, $DB_PASSj2)
or die("Mysql disconnect");

/* SQL 文を変数に格納する */

$query2="SELECT * FROM $DB_TABLEj LEFT OUTER JOIN $DB_TABLEj2 ON $DB_TABLEj.PK = $DB_TABLEj2.PK";

/* SQL 文を実行 */
$rs = mysql_db_query($DB_NAMEj2, $query2);

/* 行数と列数を計算して取得 */
$fields = mysql_num_fields($rs);
$rows = mysql_num_rows($rs);

/* 計算して取得した行数、列数だけループ */
/* ループの間データの表示を続ける */
```

```
for($i = 0;$i < $rows;$i++){
   for($j = 0;$j < $fields;$j++){
print(mysql_result($rs,$i,$j));
/* Insert canma */
if($j < $fields -1)
print(",");
}

print("\n");
}

/* MySQL dissconnect */
mysql_close($link);

?>
```