Cracking Analyzer の設計コンセプト

1T－10

藤田直行、高橋匡康、岩宮敏幸
航空宇宙技術研究所数理解析部

1　はじめに

インターネット利用の急速な普及に伴い、構内ネットワーク（以下、ＬＡＮと記す）においてもCracking への警戒が必要となっている。CrackingからＬＡＮ内の様々な資源を守るための一手法に緩衝帯ネットワークがある。ＬＡＮ～インターネット間のあらゆる通信を、緩衝帯ネットワークを介して行う事により、通信の安全性を高めるものである。航技研でも、緩衝帯ネットワークを構築しＬＡＮの安全性の確保に努めている[1]。しかし、現状では最小限のネットワークサービスしか利用できず、また、どの様な Cracking が行われようとしているのか十分に把握できていない。

そこで本研究ではネットワーク上の通信を監視することにより、ネットワークセキュリティ解析の基礎となる情報をアプリケーション層を含めた任意のプロトコル層で取り出し、解析する方法を検討している。本論文ではこれを実現したソフトウェア Cracking Analyzer （以下、ＣＡと記す） の基本設計について述べる。

2　ＣＡの必要条件

ネットワークセキュリティ対策用ソフトウェアは、監視対象ホストにインストールし Cracking を常時監視するもの[2,3]、定期的にプログラムを走らせチェックを行うもの[4,5,6]が既に存在し有効に利用されている。しかし、対象ホスト毎にインストールする、解析プロトコル毎にシステムを構築する等の基本設計のため、監視対象ホストやプロトコル数の増大に対応できないという問題点を抱えている。一方 Cracking は、パケットの盗聴・ＯＳやコマンド実装上の不備・プロトコル設計のセキュリティ上の問題点等あらゆる層を用いて行われ[1]、また、攻撃対象は無差別であることが多い。

本研究では、既存のネットワーク機器に一切変更を加える事無くかつ任意のデータ形式でCracking解析を行えるシステムの開発を行った。具体的には次の条件を満たすソフトウェアの開発を目標にした。

(1) 対象プロトコルを限定しない(汎用性)
(2) ＬＡＮ上の個々のネットワーク機器に一切手を加えない(スケーラビリティ)
(3) 人間が見て意味のあるデータ単位(判読性)
(4) パケットの取りこぼしが無い(確実性)
(5) リアルタイムに解析を行う(リアルタイム性)

Cracking Analyzer Design Concept
Naoyuki FUJITA(fujita@nal.go.jp)
Tadayasu TAKAHASHI(takahasi@nal.go.jp)
Toshiyuki IWAMIYA(iwamiya@nal.go.jp)
NAL Computational Sciences Div.

3　ＣＡ概要

ＣＡシステム全体を各ネットワーク層に対応したモジュールに分割し、各モジュールにはその層の情報のみを処理させる設計とした（図１）。各層は、ＣＡのコンフィグレーションファイル（以下、ca.conf と記す）で指定された条件を満たす通信内容を、Cracking 解析プログラム（以下、アナライザと記す）もしくは一つ上位の層を担当するモジュールに渡すだけで、実際の解析はアナライザが行う。

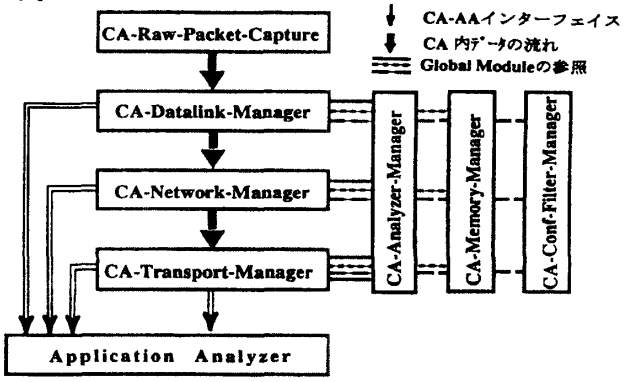

図１　ＣＡ概念図

アナライザが解析しやすいデータをネットワーク上のパケットから生成するため、アプリケーション層までのセッションの再構築機能を実装した。これにより、解析対象とする Cracking の記述が容易になり、簡便な ca.conf の記述が可能となる。ＣＡ～アナライザ間のデータの受け渡しは、図２に示すヘッダを持つひと固まりのデータの受け渡しにより行う。アナライザへ渡すデータの指定は ca.conf

4Byte	header length
4Byte	data length
4Byte	session start time
4Byte	session end time
4Byte	interface index
4Byte	physical layer　protocol ID
4Byte	physical layer　data offset
4Byte	datalink layer　protocol ID
4Byte	datalink layer　data offset
4Byte	network layer　protocol ID
4Byte	network layer　data offset
4Byte	transport layer　protocol ID
4Byte	transport layer　data offset
4Byte	session layer　protocol ID
4Byte	session layer　data offset
4Byte	presentation layer　protocol ID
4Byte	presentation layer　data offset
4Byte	application layer　protocol ID
4Byte	application layer　data offset

図２　ＣＡヘッダ

で行う（図３）。 ca.conf にはこの他にバッファサイズや欠損パケットを含むセッションの取り扱い等ＣＡの動作自体を定義するステートメントもある。

```
#-------------------------------------------------------------#
# Filter for login Analyzer                                   #
#-------------------------------------------------------------#
filterset {
        physical  = ( ether: * <-> aa:bb:cc:xx:yy:zz )
        network   = ( ip: * <-> 202.26.95.68 )
        transport = ( tcp: * <-> 23 )
        analyzer  = ( login-A login.conf -log login.log -dump login.dump)
}
```

図３　ＣＡのコンフィグレーションファイルの例

ＣＡはＬＡＮ～インターネット間の Cracking を解析する事を目的としている。このため、監視対象のネットワーク機器個々にソフトウェアのインストールを要しないシステムの構築が可能となる。図４に示す様に、緩衝帯ネットワークに Cracking 測定用ネットワークを追加する。

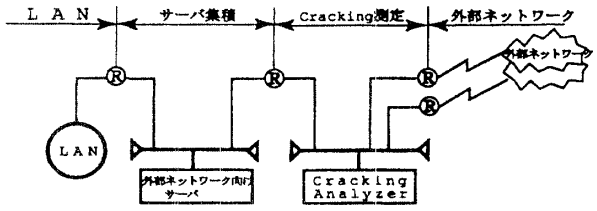

図４　Cracking 測定用緩衝帯ネットワーク構成

ＣＡ～アナライザ間のインターフェースはアナライザが、ＣＡアナライザインターフェース関数を用いて、図５の様なコーディングを行う。現在、smtp, nntp, ftp, login, 帯域監視等のアナライザが動作している。図６にアナライザの出力の一例を示す。

```
void main(int argc, char** argv){

        struct ca_packet* cap;

        ca_open();                      /* ＣＡとのインターフェース  OPEN    */

        while (TRUE) {
                if (ca_next(CA_W_NORMAL) < 0) {/* １データ読み込み      */
                        ca_close();     /* インターフェース close */
                        exit(1);
                }
                cap = ca_read(CA_PRN_APPLICATION);
                /*  読み込んだデータの注目している層のポインタを取得      */

                func(cap->...)                  /*  ポインタ  cap で      */
                                                /*  任意にデータを        */
                                                /*  参照し解析する        */
        }
}
```

図５　アナライザとのインターフェース関数群

４　まとめ

以下の特徴を持つＣＡを開発した。●データ収集部（ＣＡ本体）と解析部（アナライザ）を分離しさらにデータ収集部を層毎にモジュール化し担当層のみの情報を処理する設計とした事により全てのプロトコルに対応できるシステムが構築できた●アプリケーション層までのセッション再構築機能により任意の通信を任意の形式で取得できよって、Crakcing 解析に有用なセッション単位でのデータを取り扱えた●緩衝帯ネットワークとの組み合わせにより既存設備への変更を一切要さなかった●smtp, nntp, ftp, login, 帯域監視等のアナライザが動作している。また、今後より一層の研究が必要と思われる点は、●巨大データおよび不完全セッションの取り扱い●アナライザへのデータ受け渡しの高速化●ＣＡ内でのデータコピーの高速化●リアルタイム性のより一層の追求●forwarding rule の動的変更等である。

```
-------- Cracking Data contents ------------------------
Date                    : Dec 26 13:28:00 1996
Server Address          : 202.26.95.68:23
Client Address          : 202.26.95.69:1024
Defhost Name            : host1
Login Times             : Threshold User = 0   Threshold
System = 3
Easy Password Used   : NO

-------- Command Details -------------------------------
S :
S : Welcome to mail.nal.go.jp (ttyp4)
S :
S : login:
C : root
S : Password:
C : rootS
S : Login incorrect
S : login:
C : root
S : Password:
C : root
```

図６　アナライザの出力例

なお、本研究の一部は、科学技術振興調整費の省際ネットワークの研究により行われた。

参考文献

[1]藤田直行、岩宮敏幸、石塚只夫、インターネット接続時におけるＬＡＮのセキュリティ確保、第33回情報科学技術研究集会予稿集、日本科学技術情報センター、1996

[2]Wietse Venema, TCP WRAPPER:Network monitoring, access control and booby traps, In proceedings of the Third Usenix UNIX Security Symposium, pp. 85-92, 1992

[3]Gene Kim and Eugene H. Spafford, The design and implementation of Tripwire:A file system integrity checker, Technical Report CSD-TR-93-071, Purdue University, 1993

[4]Dan Farmer and Wietse Venema, Improving the security of your site by breaking into it, http://www.fish.com/satan

[5]Dan Farmer and Eugene H. Spafford, The COPS security checker system, In USENIX Conference Proceedings, pp. 165-170, Anaheim, CA, Summer 1990

[6]Alec D. E. Muffet, Crack:A Sensible Password Checker for Unix, ftp://ftp.cert.org/pub/tools/crack