組込みLinuxのセキュリティ向上技術の開発

組込み機器が高機能化し、インターネットに接続される製品が増加しています。このような組込み機器のセキュリティを向上させる技術を研究開発しました。その結果、既存のソフトウェアに変更を加えずに適用可能な技術を開発できました。

組込み機器のネットワーク化とセキュリティ

近年、インターネットに接続される組込み機器が増加しています。例えば、多くのデジタルテレビには、電子メールを読み書きしたり、番組表をインターネットから取得して録画予約をしたりする機能が付加されています。このような組込み機器は、パソコンと同じようにコンピュータウィルスや不正侵入などの被害を受ける可能性があります。

本研究では、組込み機器のネットワークセキュリティを向上させる技術について研究開発を行いました。組込み機器はパソコンに比べて、セキュリティ向上に利用できるハードウェア機能が少なく、ハードウェアの多様性が高いという特徴があります。このため、本研究ではソフトウェアによるセキュリティ向上技術を開発し、これらの課題に対応しました。また、LinuxをOSとして用いる組込み機器を対象として開発を行いました。Linuxは今日、高機能な組込み製品で多く用いられています。

組込み機器のセキュリティ技術

ウィルスなど大部分の攻撃手法は、以下の手順で攻撃を行います。①不正な振る舞いをするプログラムを機器に送り込む。②このプログラムを実行させる。このどちらかの手順を防ぐことができれば、セキュリティを大きく向上させることができます。

①は、本質的に防ぐのが難しい手順です。組込み機器にとって、プログラムとデータはともに0と1の列に過ぎないため、簡単には見分けられません。そして、データは読み書きの両方ができなければならないため、書き換えが可能な状態にしなければなりません。このため、本来データが置かれるべきところに不正なプログラムを書き込む攻撃が可能です。パソコンで用いられるウィルス対策ソフトウェアでは、ウィルスが持つビットパターンなどの特徴を、いわばプログラムの指紋のように扱い、不正なプログラムが送り込まれるのを水際で防ぎます。このような既知のパターンを多量に蓄積しなければならないような技術は、メモリ容量の少ない組込み機器への適用が困難です。

一部のパソコンでは、不正なプログラムの実行を困難にする機能がハードウェアによって実現されています。しかし、安価な組込み機器では、このような機能を持つハードウェアを使用することができません。このように、組込み機器では主にハードウェアの制限から、パソコンで用いられるセキュリティ技術の適用が困難です。本研究では、②の手法として最もよく用いられるバッファオーバフローを利用した攻撃を困難にするソフトウェアを開発しました。

バッファオーバフロー攻撃

バッファオーバフロー攻撃は、ソフトウェアの不具合を利用して、上述の攻撃手順①、②を同時に行う攻撃手法です。これは最も頻繁に利用される攻撃手法の一つです。バッファオーバフロー攻撃が利用するソフトウェアの不具合は、C言語などで作成されたプログラムにおいて見られるものです。

C言語のプログラムは、論理的には変数と関数から構成されます。変数はプログラムの扱う

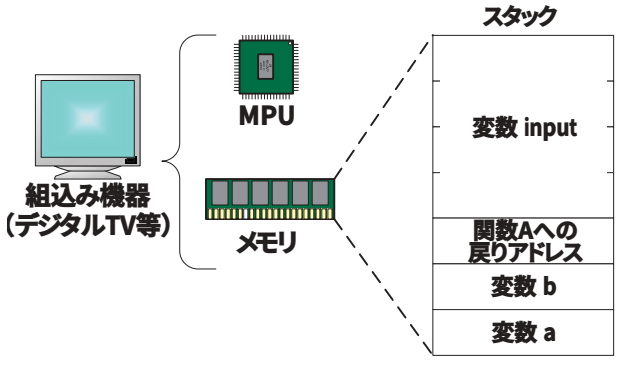

図1　スタック領域
メモリ内にはスタックと呼ばれる領域があり、変数と制御情報の両方が格納されます

データを抽象化し、関数はプログラムの処理手順を抽象化する概念です。多くのC言語処理系では、変数と関数の処理手順を制御するための情報が、スタックと呼ばれるメモリの領域に共存します。図1は、関数Aがその処理の途中で関数Bを利用する際のスタックの使われ方の例です。関数Aはa、b二つの変数を使用し、関数Bは16文字以内の入力を外部から受け取ってこれを変数inputに格納するとします。図で変数inputの直下には、関数Bの処理を終えた後に関数Aの実行を再開すべき位置を示す戻りアドレスと呼ばれる制御情報が格納されています。

ここで、攻撃者は、不正なプログラムを20文字相当のデータとして入力します。C言語では、想定を超えた長い入力がなされた場合の対応は、プログラムの作成者に任されています。そのため、この対応が適切に行われていないプログラムは、不具合を持つといえます。この例では、関数Bで想定されている最大長16文字を超える20文字相当のデータが入力された結果、変数inputにデータが収まりきらず、その直下の戻りアドレスを書き換えられてしまう可能性があります。戻りアドレスが書き換えられると、関数Bの処理が終了した後に、関数Aではなく、別の処理が行われることになります。攻撃者は、これを利用して、戻りアドレスが不正なプログラムを指すように入力データを作ることで攻撃を実現します。

開発手法

開発した手法では、プログラムに以下の二つのセキュリティ機能を追加します。

- 関数の実行前に戻りアドレスをスタックとは異なる別の領域にコピーする。
- 関数が処理を終了した際に、スタック内の戻りアドレスと前項のコピーを比較することで、不正な書き換えの有無を検知する。

本研究では、Linuxでプログラムが実行される仕組みを改造し、プログラムが実行される直前にセキュリティ機能を自動的に付加できるようにしました。

図2は、Linuxにおいて通常のプログラムが実行される様子を示しています。ユーザがプログラムの実行を指示すると、まず動的リンカと呼ばれる特別なプログラムが実行され、動的リンカが指示されたプログラムを読み込んで実行します。この動的リンカを改造し、図3のようなプログラム書き換え機能を持たせました。改造された動的リンカは、プログラムをメモリに読み込んだ後、これを実行する前に、プログラム中から関数呼び出しに用いられるコール命令を検索して、この命令の呼び出し先を自動的に生成した呼出コード片に書き換えます。同様に、関数からの復帰に用いられるリターン命令を検索し、これを自動生成された復帰コード片への無条件ジャンプ命令に書き換えます。呼出コード片は上述した戻りアドレスのコピー処理をおこない、復帰コード片はスタックの内容とコピーの比較をおこないます。

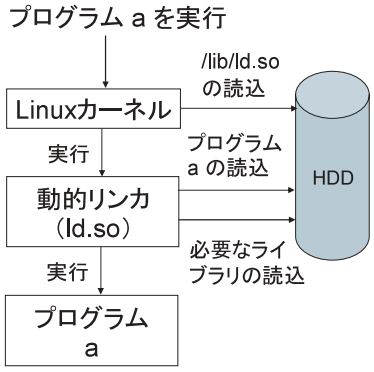

図2　Linuxのプログラム実行フロー
プログラムは動的リンカによって読み込まれます

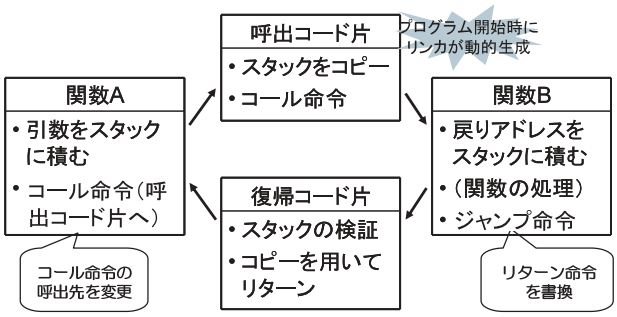

図3　本研究の提案手法
動的リンカが、呼出および復帰コード片を自動的に生成し、関数内のコール命令、リターン命令を書き換えて、これらのコード片を経由するようにします

提案手法は全てをソフトウェアによって実現するため、多様な組込み機器に適用することが可能です。また、プログラムを実行直前に書き換える方式のため、既存のソフトウェア資産を再コンパイルすることなくセキュリティを向上させることができます。

ITグループでは、OSを用いない組込み機器や、LinuxやμITRONなどを用いた組込み機器の開発、セキュリティ向上などについてご相談をお受けしています。お気軽にご相談ください。

研究開発部第一部　ITグループ　＜西が丘本部＞
大原衛　TEL 03-3909-2151 内線491
E-mail：ohara.mamoru@iri-tokyo.jp