講演番号: 12

DEP 回避の検知

Detection of DEP bypass

栗原 寛昇† 岡本 剛†

Hironori KURIHARA† Takeshi OKAMOTO†

† 神奈川工科大学情報学部情報ネットワーク・コミュニケーション工学科

† Deparment of Information Network and Communication,Kanagawa Institute of Technology

1. はじめに

近年，インターネットの普及とともに攻撃パターンが増加し，さらに攻撃コードを自動生成するツールの登場などにより，脆弱性攻撃による脅威は深刻なものとなっている[1]．脆弱性攻撃を防ぐ方法の 1 つに，Microsoft が実装した DEP(DataExecution Prevention)という機能がある．しかし，DEP を回避する脆弱性攻撃が登場し[2], Windows のセキュリティ強度の低下が問題になっている． 本稿では，DEP を回避する脆弱性攻撃を検知するプログラムを提案し，それを実装した．さらに実装したプログラムにより，DEP 回避を検知できることを確認した．

2. DEP の回避

DEP は, データ領域からの実行を防止する機能である．DEP 回避を行う攻撃は, Win32API が利用される. DEP回避に使用される API を，表 1 に示す.

表 1．DEP回避に使用される API[3]

VirtualAlloc 関数	NtSetInformationProcess 関数
HeapCreate 関数	WriteProcessMemory 関数
VirtualProtect 関数	SetProcessDEPPolicy 関数

DEP回避は, APIを呼び出して, 引数の値を設定することにより，あるデータ領域に働いている DEPを回避する. 例として，NtSetInformationProcess 関数を用いた DEP 回避について述べる．この関数は，別名 ZwSetInformationProcess 関数と呼ばれている，ネイティブ API である．この API を呼び出して，DEP を回避するためには, ZwSetInformationProces 関数の ProcessInformationclass という引数に 0x00000022（プロセスの DEP を有効にするか，無効にするかの設定を変更可能な状態にする．）の値を設定する．次に，ProcessInformation という引数に 0x00020410（現行プロセスにおいて DEP を無効にする．）の値を設定することにより DEP を回避する．この API 呼び出しを行うことにより，DEP を回避した上で，悪意のある攻撃を行う．

3. 検知手法

3.1 API フック API フックとは, プロセスが API を呼び出した際に，その呼び出しに割り込むことである．API フックにより, API の動作の監視が可能となる．フックの方法には，関数アドレスを管理しているモジュールのインポートセクションを利用したユーザーモードでの API フックと，デバイスドライバによってフックする，カーネルモードでの API フックを用いた[4].

3.2 引数の監視 API をフックするだけでは，Windows 上で動く安全なプロセスも検知されてしまう．DEP 回避を行う攻撃に使用される API は，安全なプロセスでも一般的に使用されている．そのために，誤検知が起こる．DEP 回避は, API を呼び出して, 引数の値を設定して，DEP 回避を行う．このことを利用して，API が DEP を回避する引数の値を監視し，API フックを行う．それにより，DEP 回避の誤検知を減らす.

3.3 ホワイトリストの利用 ホワイトリストとは，検知する必要のないプロセス名の一覧である．DEP 回避によって API が使用する引数の値も，安全なプロセスでも使用される場合がある．ホワイトリストを用いて検知する必要のないプロセスを除外して，誤検知をさらに減らす.

4. 評価

4.1Metasploit framework によるテスト 本稿では，Metasploit framework という攻撃コードを生成するツールを用いる．動作検証では，windows/smb/ms08_067_netapi[4]という脆弱性攻撃を利用した．この攻撃は，ZwSetInformationProcess 関数を利用した DEP 回避を行う攻撃である.

4.2 評価の結果 windows/smb/ms08_067_netapi の攻撃がカーネルモードで動作する API を使用しているため，カーネルモードで動作する API をフックした．デバイスドライバによって検知した内容は，DebugView を用いて確認した．それにより，DEP 回避が検知できることを確認した.

5. 今後の課題

DEP を回避する脆弱性攻撃の検知を確認した上で，攻撃の動作を阻止する処理を加えることが今後の課題である.

参考文献

[1] Jon Eerickson：HACKING：美しき策謀, O'REILLY, pp. 12-15, 2005.

[2] Windows のセキュリティ機能 DEP を回避する新手法が公開される:http://www.computerworld.jp/topics/vs/175929.html ,2010

[3] CorelanTeam, http://www.corelan.be:8800/index.php/2010/06/16/exploit-writing-tutorial-part-10-chaining-dep-with-rop-the-rubikstm-cube/

[4] Jeffrey Richter: Advanced Windows 第 5 版 下，日経 BP ソフトプレス，pp. 314-338, 2008.

[5] Microsoft:MicrosoftSecurityBulletinMS08-067:http://www.microsoft.com/technet/security/bulletin/MS08-067.mspx,2008