実践セキュリティ特論 6/10 課題

28G23027
川原尚己

1. 「リターンアドレスの書き換え」の原理を以下の単語を用いて説明せよ。(600 文字以内)：EIP，バッファオーバーフロー，リターンアドレス

   情報を一時的に格納するメモリでは「スタック」と呼ばれるデータ構造が採用されている．スタックでは最初に入れられたデータほどアドレスが大きく，後になるほど小さくなる．また，データを利用するときには最新のデータから順に取り出されるという特徴がある．スタックのアドレスは，ソースコードを実行するときには EIP と呼ばれる変数によって管理されている．処理が進むと，その内容によって EIP が前後に移動する．C 言語において scanf 関数で標準入力を受け取る場合には，初めにどれほどの部分までユーザが利用するのか宣言する必要がある．もし確保したデータ部分よりもさらにデータを書き込んでしまうと，標準入力のために用意している部分を超過し，別の処理のために確保されている領域まで侵入してしまうことになる（バッファオーバーフローという）．このとき，EIP は標準入力で用意されている領域のアドレスではなく，オーバーフローした部分のアドレスを持っていることになる．ここでもし，本来は意図されていない処理の開始地点を EIP が指していたら，その処理を実行させることができる．この，「処理の開始地点」のことをリターンアドレスという．このように，攻撃によって無理やり処理を実行させようとさせると，目的の処理のリターンアドレスがどこであるのかを突き止める必要がある．

2. セキュリティ機能である「カナリア」およびその回避手法について説明せよ。(600 文字以内)

   1 では，脆弱性を突き，スタックオーバーフローを起こすことでプログラム中の予期しない処理を動作させることが可能であることを示した．このような攻撃が可能となるような原因として，異なる処理が連続するアドレスで格納されていることが挙げられる．これを防ぐ技術として「カナリア」がある．これはスタック中にカナリアという，スタックオーバーフローを検知するための領域を確保しておくことである．もし，処理を実行するときにカナリアに格納されているデータが書き換わっていたら，バッファオーバーフローが起こったとみなし，処理を強制的に終了する．

   カナリアは，関数の呼び出し時にスタックに格納され，関数の終了時に書き換えの判定がなされる．すなわち，その間にバッファオーバーフローを発生させ，目的の処理を実行させることができれば，カナリアを回避することができる．あるいは，fork によって作成される子プロセスは親プロセスのプロセス空間をコピーするこ

とから，カナリアの値は親プロセスと子プロセスで一致することを利用し，ブルートフォース攻撃によりカナリアの値を特定する方法もある．

参考文献：

・Master Canary Forging: 新しいスタックカナリア回避手法の提案 by 小池 悠生 - CODE BLUE 2015
https://www.slideshare.net/codeblue_jp/master-canary-forging-by-code-blue-2015