MicroDog 技術研討

校稿 ： 正新電腦技術部 2014/12/25

前言

盜版技術日新月異，早期的保護策略已不符合時代的變遷，因此定期更新軟體、制定新保護策略，更是不能被忽視的首要任務，本通報會將說明如何提升高強度保護。

您是否將 Dog Key 變成單純 Memory Key 進行讀寫驗證，這樣僅發揮此產品一半不到的功能且安全強度相對較低，下面說明內容，將一步步探討各種策略的安全性，快來檢驗您的方案安全性吧。

第一節 讀取 Memory 驗證策略（安全等級：✪）

保護策略：此種方式為檢查 Key 和內部資訊重覆驗證，著重在加入許多內碼讓程式運行時有大量的”垃圾”訊息傳遞，讓惡意使用者無法輕易判斷哪些訊息為真、哪些為假。

```
RetCode = DogCheck();
... //    加入其他運算邏輯、資料
if ( RetCode != 0 )
{
...
}
RetCode = ReadDog();
If ( RetCode != 0 )
{
...
}
... // 加入其他運算邏輯、資料
If ( RetCode != 0 )
{
...
}
```

技術說明：現在電腦運算能力越來越強，已經可以快速分析大量資訊以及測錄工具不斷升級的的情況下安全強度相對較弱，建議一定要增強保護策略。

第二節 將返回值和傳遞訊息複雜化（安全等級：✪✪）

保護策略：Memory 驗證策略的進階用法，產生多個數值讓那些企圖追蹤程式碼的人無法輕易取得資訊，只要任何返回值錯誤就退出程式或回報異常;另一方面也可以在 Key 中放入硬體常量像是檔案名稱或是 DOG 檔案的路徑、和一些常用的訊息。

```
//將 RetCode 返回值打散
RetCode = ReadDog();
for ( i = 0; i < 100; i ++ )
{
a [i] = RetCode ;
}
for ( i = 0; i < 100; i ++ )
{
b [i] = a [i] ;
}
//於 Dog 內部藏硬體常量資訊或是資料來做驗證
DogBytes = 5;
DogAddr = 10;
RetCode = ReadDog();
Strcpy ( Path, “C:\” );
Strcat ( Path, DogData);
Strcat ( Path, “EXEC.BAT” );
...
DogBytes = 1;
DogAddr = 15;
RetCode = ReadDog ();
for ( i = 0; i < 100; i= i + *DogData)
{
...
}
```

//隨機讀取值驗證

DogAddr = Random(200);

DogBytes = Random(200 –DogAddr);

RetCode = ReadDog();

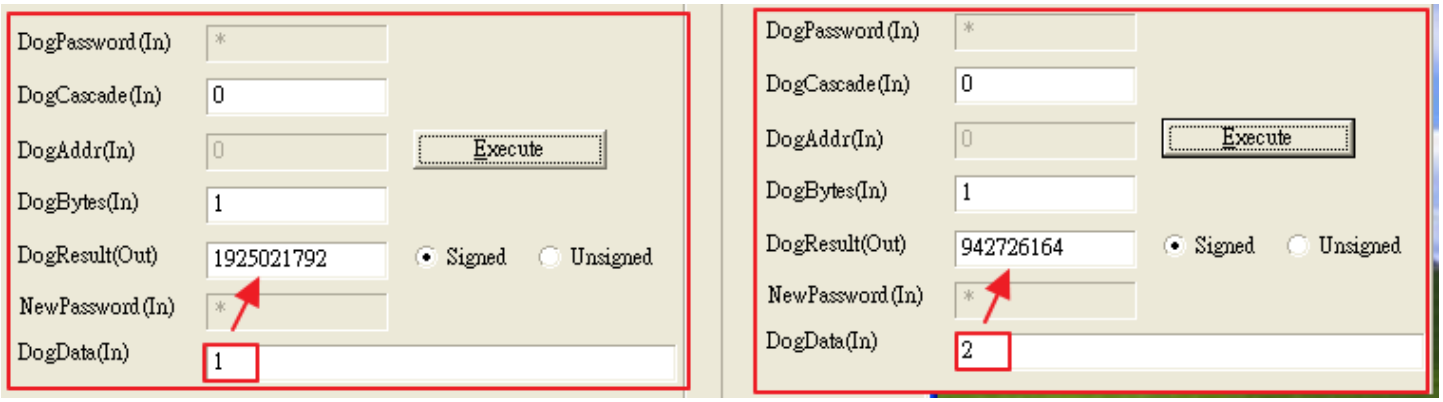
...

技術說明：與第一節相較之下，程式運行時所返回的訊息複雜許多，提高了安全性，但嚴格來說只能增加破解所需花費的時間，此安全策略必須搭配定期更新軟體和驗證機制，將驗證資料和替換訊息的邏輯式修改，避免風險。

第三節 使用 DogConvert()函數驗證（安全等級：✪✪✪ ）

保護策略：透過硬體 Key 晶片進行加密演算法，例如將加密演算後的答案紀錄 1000 組存 Table 裡，程式中隨機產生一組值進行加密，在驗證 table 中記錄的值是否符合。並且由於開發商編號不同，Key 換算出來的值也一定不相同，這部份於購買時已經客製化。

//可透過 DogEdit 執行 DogConvert()函數得出 Key 換算出的答案



//例如事先算出"MicroDog"加密後回得到"12345678"，並加入程式碼中驗證

char ConvertBuff[10]={"MicroDog"};

……

DogData= ConvertBuff;

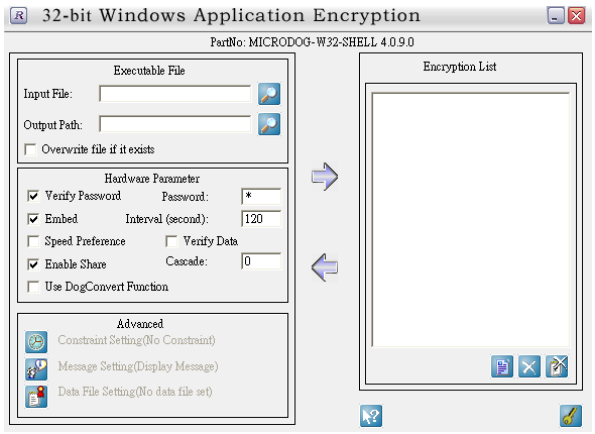
DogBytes = strlen(ConvertBuff);

if(DogConvert())

exit(2);

```
else
if(DogResult ！= 12345678)
exit(2);
```

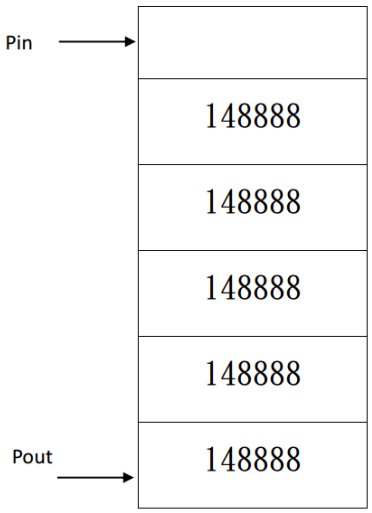
//內建 Shell 工具也會引用到 DogConvert()驗證



技術說明：此函數會動用內部晶片轉換，同時硬體本身有自己的安全規格，可將安全強度大大的提升，駭客必須耗費相當大的成本去破解軟體和硬體，建議一定要採取此保護策略。

第四節 透過佇列(Queue)管理返回值（安全等級：✪✪✪✪）

保護策略：可以建造一個佇列以管理返回值和返回結果，管理的方法可以複雜一些，而程序的執行是倚賴於這個佇列的。當然不用佇列而使用其他的數據架構也是一樣的，甚至效果更好。一個簡單的使用佇列進行返回值管理的例子如下圖所示：



在上圖中，有一個初始化的佇列，共有 6 個單元，初始數據為 148888，當發生的返回值不為 時，將 205429 賦給 Pin，並使隊列移動一個單元。在 Pout 處檢測被推出的數據，當為 148888 時不做動

作，當為 205429 時即認為非法。

技術說明：制定專屬的保護策略、尋求技術支援即使可透過上述範例增強您的保護，仍然是要加入自己本身的策略而非照本宣科，唯有將程式本身和 Key 的連結做到越緊密才能達到高強度保護。

第五節 制定專屬的保護策略、尋求技術支援

即使透過上述範例增強您的保護外，不外乎需加入自己本身的策略而非照本宣科，唯有將程式本身和 Key 的連結做到越緊密才能達到高強度保護。

相關連結

1. MicroDog 4.0 SDK https://copy.com/llJNryC6GJ4H9M6F
2. Microdog 中文快速入門 https://copy.com/1CYCPx7ttwkKcuAe
3. 驅動程式 http://www.pronew.com.tw/download/microdog/MicroDog_Driver_4.0.16.4.zip