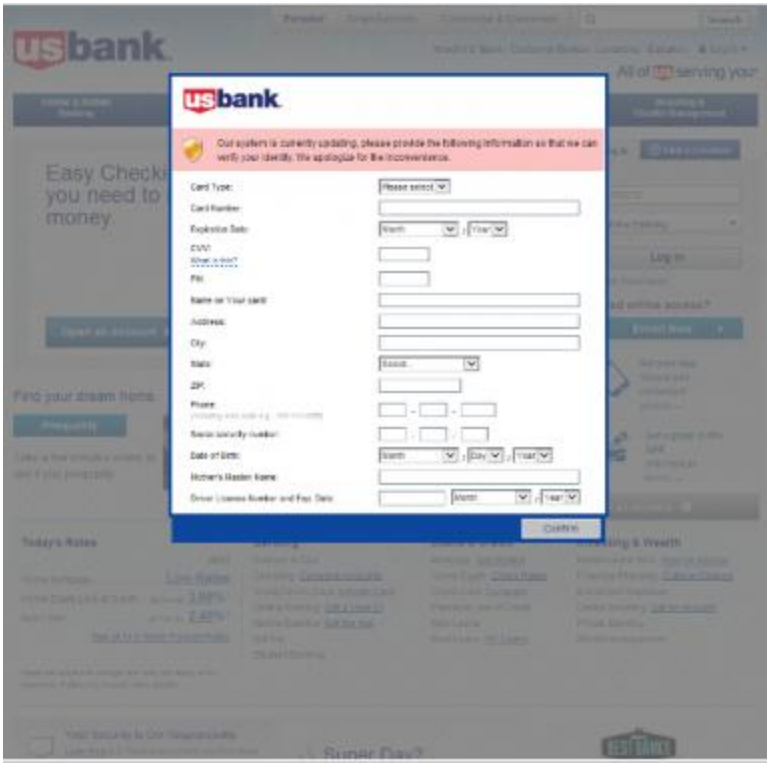
garage4hackers

garage4hackers.com/entry.php

CSIS Security Group A/S has uncovered a new trojan-banker family which we have named Tinba (Tiny Banker) alias Zusy.



Tinba is a small data stealing trojan-banker. It hooks into browsers and steals login data and sniffs on network traffic. As several sophisticated banker-trojan it also uses Man in The Browser (MiTB) tricks and webinjects in order to change the look and feel of certain webpages with the purpose of circumventing Two factor Authentification (2FA) or tricking the infected user to give away additional sensitive data such as credit card data or TANs.

Tinba is the smallest trojan-banker we have ever encountered and it belongs to a complete new family of malware which we expect to be battling in upcoming months.
source:https://www.csis.dk/en/csis/news/3566/

This time we are investigating the tiny banker's DGA. We are again only interested in the code that is dealing
with the domain name generation.
Execute the sample and let it terminate itself. It will inject its code into Explorer.exe process. Attach debugger to explorer.exe.

Now where should we start in vast pool of explorer.exe address space?
The answer is again the place where we can find teh simptoms of DGA. Let us set breakpoint at
"gethostbyname".
The code will break at gethostbyname, let it finish its work and you'l land in DGA code from its return address.
The following code belongs to the DGA :

Code:

```
02252739   55              PUSH EBP
0225273A   89E5            MOV EBP,ESP
0225273C   81EC 0C020000    SUB ESP,20C
02252742   57              PUSH EDI
02252743   6A 20            PUSH 20
02252745   8D93 605E4000    LEA EDX,DWORD PTR DS:[EBX+405E60]
0225274B   52              PUSH EDX
0225274C   8D95 F8FEFFFF    LEA EDX,DWORD PTR SS:[EBP-108]
02252752   52              PUSH EDX
02252753   E8 040B0000      CALL 0225325C
02252758   0FB783 C25E4000  MOVZX EAX,WORD PTR DS:[EBX+405EC2]
0225275F   8945 FC          MOV DWORD PTR SS:[EBP-4],EAX
02252762   EB 6C            JMP SHORT 022527D0
02252764   837D FC 00       CMP DWORD PTR SS:[EBP-4],0
02252768   75 21            JNZ SHORT 0225278B
0225276A   6A 20            PUSH 20
0225276C   8D95 F8FEFFFF    LEA EDX,DWORD PTR SS:[EBP-108]
02252772   52              PUSH EDX
02252773   8D93 605E4000    LEA EDX,DWORD PTR DS:[EBX+405E60]
02252779   52              PUSH EDX
0225277A   E8 DD0A0000      CALL 0225325C
0225277F   0FB783 C25E4000  MOVZX EAX,WORD PTR DS:[EBX+405EC2]
02252786   8945 FC          MOV DWORD PTR SS:[EBP-4],EAX
02252789   EB 45            JMP SHORT 022527D0
0225278B   FF4D FC          DEC DWORD PTR SS:[EBP-4]
0225278E   8DB3 A05E4000    LEA ESI,DWORD PTR DS:[EBX+405EA0] ; "oGkS3w3sGGOGG7oc"
02252794   B9 10000000      MOV ECX,10
02252799   31C0            XOR EAX,EAX
0225279B   31D2            XOR EDX,EDX
0225279D   AC              LODS BYTE PTR DS:[ESI]
0225279E   01C2            ADD EDX,EAX
022527A0  ^E2 FB           LOOPD SHORT 0225279D
022527A2   8DBB 605E4000    LEA EDI,DWORD PTR DS:[EBX+405E60] ; "ssrgwnrmgrxe.com"
022527A8   B9 0C000000      MOV ECX,0C
022527AD   0207            ADD AL,BYTE PTR DS:[EDI]
022527AF   30D0            XOR AL,DL
022527B1   0247 01          ADD AL,BYTE PTR DS:[EDI+1]
022527B4   3C 61            CMP AL,61
022527B6   76 04            JBE SHORT 022527BC
022527B8   3C 7A            CMP AL,7A
022527BA   72 04            JB SHORT 022527C0
022527BC   FEC2            INC DL
022527BE  ^EB ED            JMP SHORT 022527AD
022527C0   AA              STOS BYTE PTR ES:[EDI]
022527C1  ^E2 EA            LOOPD SHORT 022527AD
022527C3   B0 2E            MOV AL,2E ; "."
022527C5   AA              STOS BYTE PTR ES:[EDI]
022527C6   8B83 B85E4000    MOV EAX,DWORD PTR DS:[EBX+405EB8] ; "com"
022527CC   AB              STOS DWORD PTR ES:[EDI]
022527CD   31C0            XOR EAX,EAX
022527CF   AA              STOS BYTE PTR ES:[EDI]
022527D0   8D93 605E4000    LEA EDX,DWORD PTR DS:[EBX+405E60]
022527D6   52              PUSH EDX
022527D7   FF93 4D354000    CALL DWORD PTR DS:[EBX+40354D]
022527DD   85C0            TEST EAX,EAX
```

```
022527DF   79 1C           JNS SHORT 022527FD
022527E1   8D93 605E4000   LEA EDX,DWORD PTR DS:[EBX+405E60]
022527E7   52              PUSH EDX
022527E8   FF93 5F354000   CALL DWORD PTR DS:[EBX+40355F]  ; call to gethostbyname
022527EE   85C0            TEST EAX,EAX
022527F0  ^0F84 6EFFFFFF   JE 02252764
022527F6   8B40 0C         MOV EAX,DWORD PTR DS:[EAX+C]
022527F9   8B00            MOV EAX,DWORD PTR DS:[EAX]
022527FB   8B00            MOV EAX,DWORD PTR DS:[EAX]
022527FD   8B55 08         MOV EDX,DWORD PTR SS:[EBP+8]
02252800   8942 04         MOV DWORD PTR DS:[EDX+4],EAX
02252803   66:8B83 7E5E4000 MOV AX,WORD PTR DS:[EBX+405E7E]
0225280A   66:8942 02      MOV WORD PTR DS:[EDX+2],AX
0225280E   66:C702 0200    MOV WORD PTR DS:[EDX],2
02252813   C642 08 00      MOV BYTE PTR DS:[EDX+8],0
02252817   C642 0C 00      MOV BYTE PTR DS:[EDX+C],0
0225281B   8DBD F4FDFFFF   LEA EDI,DWORD PTR SS:[EBP-20C]
02252821   6A 20           PUSH 20
02252823   8D93 605E4000   LEA EDX,DWORD PTR DS:[EBX+405E60]
02252829   52              PUSH EDX
0225282A   57              PUSH EDI
0225282B   E8 2C0A0000     CALL 0225325C
02252830   83C7 20         ADD EDI,20
02252833   B9 18000000     MOV ECX,18
02252838   60              PUSHAD
02252839   6A 01           PUSH 1
0225283B   FF93 93104000   CALL DWORD PTR DS:[EBX+401093]
02252841   61              POPAD
02252842   0F31            RDTSC
02252844   AB              STOS DWORD PTR ES:[EDI]
02252845   49              DEC ECX
02252846  ^75 F0           JNZ SHORT 02252838
02252848   8B7D 0C         MOV EDI,DWORD PTR SS:[EBP+C]
0225284B   31C0            XOR EAX,EAX
0225284D   AA              STOS BYTE PTR ES:[EDI]
0225284E   B8 80000000     MOV EAX,80
02252853   AB              STOS DWORD PTR ES:[EDI]
02252854   68 80000000     PUSH 80
02252859   8D95 F4FDFFFF   LEA EDX,DWORD PTR SS:[EBP-20C]
0225285F   52              PUSH EDX
02252860   57              PUSH EDI
02252861   E8 F6090000     CALL 0225325C
02252866   83EF 04         SUB EDI,4
02252869   57              PUSH EDI
0225286A   E8 57F9FFFF     CALL 022521C6
0225286F   4F              DEC EDI
02252870   68 85000000     PUSH 85
02252875   57              PUSH EDI
02252876   FF75 08         PUSH DWORD PTR SS:[EBP+8]
02252879   E8 8F010000     CALL 02252A0D
0225287E   89C6            MOV ESI,EAX
02252880   29F8            SUB EAX,EDI
02252882   6A 04           PUSH 4
02252884   E8 05000000     CALL 0225288E
02252889   0D 0A0D0A00     OR EAX,0A0D0A
```

```
0225288E   50              PUSH EAX
0225288F   57              PUSH EDI
02252890   E8 F30A0000     CALL 02253388
02252895   85C0            TEST EAX,EAX
02252897  ^0F84 C7FEFFFF   JE 02252764
0225289D   8D78 04         LEA EDI,DWORD PTR DS:[EAX+4]
022528A0   39F7            CMP EDI,ESI
022528A2  ^0F84 BCFEFFFF   JE 02252764
022528A8   87FE            XCHG ESI,EDI
022528AA   AD              LODS DWORD PTR DS:[ESI]
022528AB   3B83 573D4000   CMP EAX,DWORD PTR DS:[EBX+403D57]
022528B1  ^0F85 ADFEFFFF   JNZ 02252764
022528B7   AC              LODS BYTE PTR DS:[ESI]
022528B8   84C0            TEST AL,AL
022528BA  ^0F85 A4FEFFFF   JNZ 02252764
022528C0   AD              LODS DWORD PTR DS:[ESI]
022528C1   3D 80000000     CMP EAX,80
022528C6   74 12           JE SHORT 022528DA
022528C8   3D 00010000     CMP EAX,100
022528CD   74 0B           JE SHORT 022528DA
022528CF   3D 00020000     CMP EAX,200
022528D4  ^0F85 8AFEFFFF   JNZ 02252764
022528DA   50              PUSH EAX
022528DB   56              PUSH ESI
022528DC   8D95 F4FDFFFF   LEA EDX,DWORD PTR SS:[EBP-20C]
022528E2   52              PUSH EDX
022528E3   E8 0D000000     CALL 022528F5
022528E8   85C0            TEST EAX,EAX
022528EA  ^0F84 74FEFFFF   JE 02252764
022528F0   5F              POP EDI
022528F1   C9              LEAVE
022528F2   C2 0800         RETN 8
```

If gethostbyname returns empty hand, it then loops through the code to generate another domain :

Code:

```
022527E8   FF93 5F354000   CALL DWORD PTR DS:[EBX+40355F]  ; call to gethostbyname
022527EE   85C0            TEST EAX,EAX
022527F0  ^0F84 6EFFFFFF   JE 02252764
```

Above code snippet will loop through the code starting at 0x02252764. So this address is where the DGA starts,
We should check the stack and all register's conditions when we'll land on address 0x02252764.
Remember these conditions are needed as arguments while reversing any subroutine :

Code:

```
02252764   837D FC 00       CMP DWORD PTR SS:[EBP-4],0 ; Starts with 0x03E8 ; 1000
Number of domains
02252768   75 21            JNZ SHORT 0225278B
0225276A   6A 20            PUSH 20
0225276C   8D95 F8FEFFFF    LEA EDX,DWORD PTR SS:[EBP-108]
02252772   52               PUSH EDX
02252773   8D93 605E4000    LEA EDX,DWORD PTR DS:[EBX+405E60]
02252779   52               PUSH EDX
0225277A   E8 DD0A0000      CALL 0225325C
0225277F   0FB783 C25E4000  MOVZX EAX,WORD PTR DS:[EBX+405EC2]
02252786   8945 FC          MOV DWORD PTR SS:[EBP-4],EAX
02252789   EB 45            JMP SHORT 022527D0

0225278B   FF4D FC          DEC DWORD PTR SS:[EBP-4]
0225278E   8DB3 A05E4000    LEA ESI,DWORD PTR DS:[EBX+405EA0] ; "oGkS3w3sGGOGG7oc"
02252794   B9 10000000      MOV ECX,10
02252799   31C0            XOR EAX,EAX
0225279B   31D2            XOR EDX,EDX
0225279D   AC              LODS BYTE PTR DS:[ESI]
0225279E   01C2            ADD EDX,EAX
022527A0  ^E2 FB           LOOPD SHORT 0225279D
022527A2   8DBB 605E4000    LEA EDI,DWORD PTR DS:[EBX+405E60] ; "ssrgwnrmgrxe.com"
022527A8   B9 0C000000      MOV ECX,0C
022527AD   0207            ADD AL,BYTE PTR DS:[EDI]
022527AF   30D0            XOR AL,DL
022527B1   0247 01         ADD AL,BYTE PTR DS:[EDI+1]
022527B4   3C 61           CMP AL,61
022527B6   76 04           JBE SHORT 022527BC
022527B8   3C 7A           CMP AL,7A
022527BA   72 04           JB SHORT 022527C0
022527BC   FEC2            INC DL
022527BE  ^EB ED           JMP SHORT 022527AD
022527C0   AA              STOS BYTE PTR ES:[EDI]
022527C1  ^E2 EA           LOOPD SHORT 022527AD
022527C3   B0 2E           MOV AL,2E ; "."
022527C5   AA              STOS BYTE PTR ES:[EDI]
022527C6   8B83 B85E4000    MOV EAX,DWORD PTR DS:[EBX+405EB8] ; "com"
022527CC   AB              STOS DWORD PTR ES:[EDI]
022527CD   31C0            XOR EAX,EAX
022527CF   AA              STOS BYTE PTR ES:[EDI]
022527D0   8D93 605E4000    LEA EDX,DWORD PTR DS:[EBX+405E60]
022527D6   52              PUSH EDX
```

A decision is made at 0x02252764 for comparison of value at address [EBP-4] which is 0x03E8 in teh begining that is 0x3E8 and then decreses along the looping through the successive loops.
This is the maximum number of domains to be generated. If this value is greater than 0 then a jump is made to
address 0x0225278B. Let us check what is at this address:

Code:

```
0225278B   FF4D FC          DEC DWORD PTR SS:[EBP-4]
0225278E   8DB3 A05E4000    LEA ESI,DWORD PTR DS:[EBX+405EA0] ; "oGkS3w3sGGOGG7oc"
02252794   B9 10000000      MOV ECX,10
02252799   31C0             XOR EAX,EAX
0225279B   31D2             XOR EDX,EDX
0225279D   AC               LODS BYTE PTR DS:[ESI]
0225279E   01C2             ADD EDX,EAX
022527A0  ^E2 FB            LOOPD SHORT 0225279D
022527A2   8DBB 605E4000    LEA EDI,DWORD PTR DS:[EBX+405E60] ; "ssrgwnrmgrxe.com"
022527A8   B9 0C000000      MOV ECX,0C
022527AD   0207             ADD AL,BYTE PTR DS:[EDI]
022527AF   30D0             XOR AL,DL
022527B1   0247 01          ADD AL,BYTE PTR DS:[EDI+1]
022527B4   3C 61            CMP AL,61
022527B6   76 04            JBE SHORT 022527BC
022527B8   3C 7A            CMP AL,7A
022527BA   72 04            JB SHORT 022527C0
022527BC   FEC2             INC DL
022527BE  ^EB ED            JMP SHORT 022527AD
022527C0   AA               STOS BYTE PTR ES:[EDI]
022527C1  ^E2 EA            LOOPD SHORT 022527AD
022527C3   B0 2E            MOV AL,2E ; "."
022527C5   AA               STOS BYTE PTR ES:[EDI]
022527C6   8B83 B85E4000    MOV EAX,DWORD PTR DS:[EBX+405EB8] ; "com"
022527CC   AB               STOS DWORD PTR ES:[EDI]
022527CD   31C0             XOR EAX,EAX
022527CF   AA               STOS BYTE PTR ES:[EDI]
022527D0   8D93 605E4000    LEA EDX,DWORD PTR DS:[EBX+405E60]
022527D6   52               PUSH EDX
```

The above code snippet is the DGA itself. It utilizes one hardcoded domain name as seed and another string as salt.
Every new generated domain name gets consumed in generating next domain name. The difference in this DGA is
that it it doesnt use date/time for domain generation, rather uses the domain name as seed for generating next domain.

Code:

```
'''
   Filename  : TinBaDGA.py
   Developer : Garage4Hackers
   Greets    : b0nd, FB1H2S, "vinnu", l0rdDeathStorm, nightrover and all g4h team

'''

import os, time

utility = "TinBaDGA"

def tinbaDGA(idomain, seed):
    print "[+] "+utility+" : Initiated"
    suffix = ".com"
    domains = []

    count = 0x03E8
    eax = 0
    edx = 0
    for i in range(count) :
        buf = ''
        esi = seed
        ecx = 0x10
        eax = 0
        edx = 0
        for s in range(len(seed)) :
            eax = ord(seed[s])
            edx += eax
        edi = idomain
        ecx = 0x0C
        d = 0
        while ( ecx > 0 ):
            al = eax & 0xFF
            dl = edx & 0xFF
            #print "0 eax : %x edx : %x ecx : %x" % (eax, edx, ecx)
            #print "0 al : %x dl : %x" % (al, dl)
            al = al + ord(idomain[d])
            al = al ^ dl
            #print "1 al : %x dl : %x" % (al, dl)
            al += ord(idomain[d+1])
            al = al & 0xFF
            #print "2 al : %x dl : %x" % (al, dl)
            eax = (eax & 0xFFFFFF00)+al
            edx = (edx & 0xFFFFFF00)+dl
            if al > 0x61 :
                if al < 0x7A :
                    #al = ord(idomain[d])
                    eax = (eax & 0xFFFFFF00) +al
                    buf += chr(al)
                    d += 1
                    ecx -= 1
                    #print "\tal : %x ecx : %x" % (al, ecx)
                    continue
            #time.sleep(4)
            dl += 1
```

```
            dl = dl & 0xFF
            edx = (edx & 0xFFFFFF00)+dl

        domain = buf+suffix
        print "[%d] %s" %(i, domain)
        domains.append(domain)
        idomain = domain
    return domains


def init():
    harddomain = "ssrgwnrmgrxe.com"
    seed = "oGkS3w3sGGOGG7oc"
    domains = tinbaDGA(harddomain, seed)
    index = 0
    fp = open(utility+".log", "wb")
    for domain in domains :
        index += 1
        line = "[%d] %s" % (index, domain)
        fp.write(line+'\n')
        print line
    fp.close()
init()
```

SHA256 hash of Sample under investigation :
856e486f338cbd8daed51932698f9cdc9c60f4558d22d963f5 6da7240490e465