

Warszawa, 22 marca 2011

Mariusz.Burdach@prevenity.com

Wstęp

Pod koniec lutego 2011 roku klienci kilku polskich banków zostali zaatakowani nową wersją złośliwego oprogramowania Zeus (ZBOT) – ZITMO (Zeus In The MObile). Poniżej znajduje się krótka analiza wersji malware, która przeznaczona jest na urządzenia mobilne z systemem operacyjnym Symbian S60.

Główna funkcja złośliwego oprogramowania, która pozwala intruzom na wykonanie nieautoryzowanych przelewów, związana jest z przechwytywaniem wiadomości SMS z jednorazowymi kodami TAN. Aby atak się powiódł muszą być spełnione następujące warunki:

- Warunek 1: Intruz musi znać dane (login i/lub hasło) pozwalające na logowanie się do części transakcyjnej;
- Warunek 2: Intruz musi przechwytywać wiadomości SMS z kodami TAN, które przesyła bank.

Warunek 1 jest możliwy do spełnienia, gdyż najpierw zainfekowana jest stacja robocza klienta banku. Jedną z poprzednich wersji malware na system operacyjny Microsoft Windows została opisana w naszych materiałach na warsztaty dla funkcjonariuszy Policji i Prokuratorów http://forensic.seccure.net/pdf/warsztat_banking_malware.pdf. Próba nakłonienia do instalacji ZITMO na urządzeniu mobilnym odbywa się właśnie po zalogowaniu się ofiary (klienta banku) do systemu bankowości internetowej.

Przykładowy scenariusz ataku może wyglądać następująco.

1. Intruz infekuje stację roboczą klienta banku;
2. Przy logowaniu klienta banku do systemu bankowości internetowej, intruz:
 - a. Przechwytuje login oraz hasło. Hasło statycznie dodatkowo pozwala intruzowi na zalogowanie się do systemu bankowości w dowolnym momencie bez wiedzy klienta.
 - b. Podszywa się pod bank i „nakłania” klienta banku do zainstalowania na urządzeniu mobilnym aplikacji „zwiększającej” bezpieczeństwo transakcji. Jest to jeden z wielu scenariuszy. Innym przykładem może być zainfekowanie urządzenia mobilnego w momencie, gdy telefon podłączony jest do komputera stacjonarnego (np. gdy użytkownik zgrywa zdjęcia lub wykonuje backup kontaktów) – ten przykład dokładniej omówiony jest na naszej prezentacji: <http://www.prevenity.com/materialy/konferencja-IBS-2010.html>.
3. **Kradzież środków z konta ofiary może się odbywać online** – podczas sesji nawiązanej przez użytkownika (tak działa „stara” wersja Zeus) **lub w dowolnym, dogodnym dla intruza momencie** (gdy hasło do systemu przesyłane jest SMS-em, intruz również je dostanie, lub gdy zarejestruje je keylogger zainstalowany na komputerze stacjonarnym).

Instalacja

Pakiet instalacyjny cert.sis zawiera między innymi 3 pliki wykonywalne:

- C:\sys\bin\SmsControl.exe (główna aplikacja)
- C:\sys\bin\UniPass.exe (uruchamiany przy odinstalowywaniu)
- C:\system\apps\u.dat (zamieniany później na pakiet u.sisx)

cert.sis podpisany jest certyfikatem wydanym przez Symbian CA I dla ANUJ MOBILITY SA INDIA LIMITED. Certyfikat ważny jest od 26 stycznia 2011 do 26 stycznia 2021.



Szczegóły certyfikatu



Szczegóły procesu SmsControl.exe

SmsControl.exe (AppID i SecureID dla procesu wynosi 20039e30). Proces posiada następujące "uprawnienia" (ang. capabilities) w systemie: ReadDeviceData, WriteDeviceData, TrustedUI, SwEvent, NetworkServices, ReadUserData, WriteUserData.

Pliki konfiguracyjne znajdują się w katalogu procesu SmsControl.exe - C:\Private\20039e30\. Proces za pomocą funkcji MkdirAll() tworzy ten katalog.

```
.text:797F1D80 MOVS R0, R7
.text:797F1D82 BLX _ZN3RFs8MkdirAllERK7TDesC16 ; RFs::MkdirAll(TDesC16 const&)
.text:797F1D86 MOVS R2, #0
```

Uwaga: Gdybyśmy chcieli uzyskać dostęp do plików konfiguracyjnych ZITMO to wymagany jest AllFiles capability. Dzięki temu możemy uzyskać dostęp do katalogu 20039e30 i plików tam się znajdujących.

Podczas pracy SmsControl.exe odczytuje pliki z tego katalogu w następującej kolejności:

1. Settings2.dat (plik z konfiguracją złośliwego oprogramowania)
2. NumbersDB.db (lokalna baza danych)

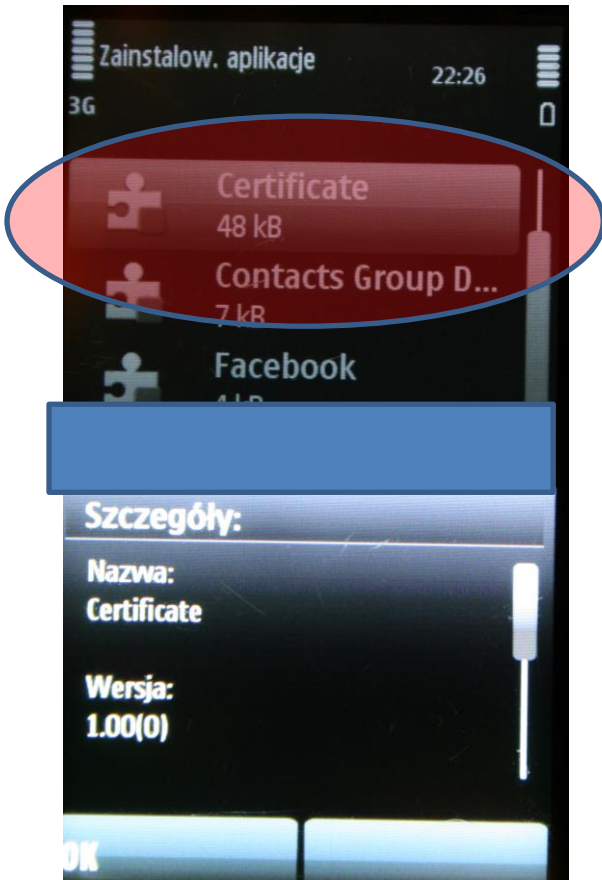
Settings2.dat zawiera następujące informacje:

- Aktualny status (ON – ZITMO jest aktywny, OFF – ZITMO jest wyłączony)
- Tryb monitorowania (monitorowane są numery telefonów znajdujące się w lokalnej bazie danych, monitorowane są wszystkie numery za pomocą komendy ADD SENDER ALL)
- Tryb blokowania (BLOCK ON – blokowane są połączenia przychodzące, BLOCK OFF – blokowanie jest wyłączone)
- Numer telefonu aktualnego administratora (ustawiany za pomocą SET ADMIN)

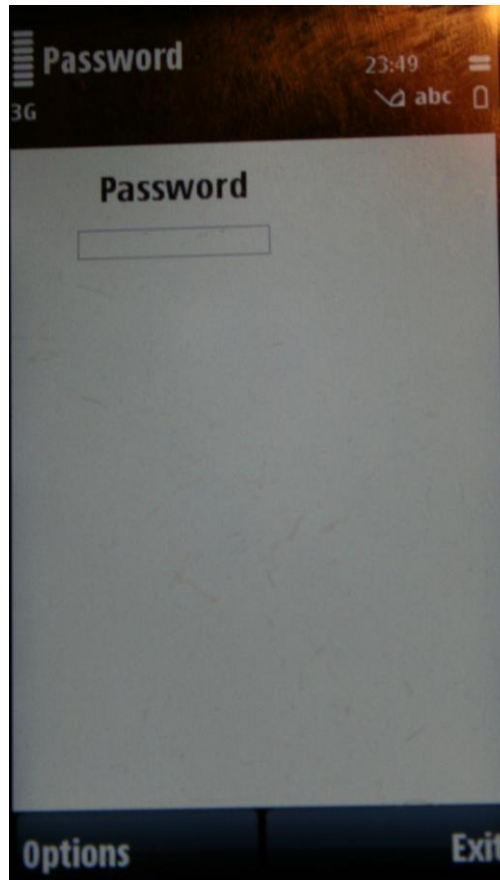
Poniżej fragment funkcji odczytującej numer telefonu administratora z pliku konfiguracyjnego Settings2.dat. Rejestr R5 zawiera numer telefonu administratora (numer ustawiony za pomocą funkcji SET ADMIN).

Identyfikacja i deinstalacja

W systemie proces SmsControl.exe widoczny jest jako Nokia Update. Na liście zainstalowanych aplikacji widnieje aplikacja Certificate.



Aplikacja Certificate – to ZITMO



Hasło: 45930

Istnieje możliwość zdalnego odinstalowania aplikacji. Wysyłamy SMS o treści UNINSTALL 45930.

Procedura zdalnej deinstalacji

1. Wysłać SMS o treści SET ADMIN na zainfekowany telefon
2. Wysłać SMS o treści UNINSTALL 45930 na zainfekowany telefon

Hasło do deinstalacji zaszyte jest w kodzie aplikacji SmsControl.exe

```

loc_A072
MOVS R0, R5
MOV  R1, R10
MOVS R2, #9
BLX  _ZNK7TDesC164LeftEi ; TDesC16::Left(int)
LDR  R1, =aUninstall ; "UNINSTALL"
MOVS R0, R4
BLX  _ZN7TPtrC16C1EPKT ; TPtrC16::TPtrC16(ushort const*)
MOVS R0, R5
MOVS R1, R4
BLX  _ZNK7TDesC167CompareERKS_ ; TDesC16::Compare(TDesC16 const&)
CMP  R0, #0
BEQ  loc_A092

loc_A092
MOVS R6, 0x470
MOVS R5, #0x83
ADD  R6, SP
LSLS R5, R5, #3
MOVS R0, R6
MOVS R2, #0xA
MOVS R3, #5
MOV  R1, R10
ADD  R5, SP
BLX  _ZNK7TDesC163MidEii ; TDesC16::Mid(int,int)
MOVS R1, R6
MOVS R0, R5
BL  sub_8D14
LDR  R1, =a45930 ; "45930"
MOVS R0, R6
BLX  _ZN7TPtrC16C1EPKT ; TPtrC16::TPtrC16(ushort const*)
MOVS R1, R6
MOVS R0, R5
BLX  _ZNK7TDesC167CompareERKS_ ; TDesC16::Compare(TDesC16 const&)

```

Funkcje ZITMO

Funkcja	SMS
Zmiana administratora	SET ADMIN
Aktywacja ZITMO	ON
Wyłączenie ZITMO	OFF
Przechwytywane wszystkich SMS-ów	ADD SENDER ALL
Wyłączenie powyższej funkcji	REM SENDER ALL
Blokowanie przychodzących połączeń	BLOCK ON
Wyłączenie powyższej funkcji	BLOCK OFF
Dodanie pojedynczego numeru do lokalnej bazy	ADD SENDER numer telefonu
Usuwa pojedynczy numer z bazy	REM SENDER numer telefonu
Filtruje pojedynczego numeru	SET SENDER numer telefonu
Zdalne odinstalowanie	UNINSTALL 45930

Przechwytywanie wiadomości SMS

Główną funkcją analizowanej aplikacji jest przechwytywanie wiadomości SMS i przesyłanie ich bez wiedzy właściciela telefonu do osoby trzeciej. Dzięki temu możliwe jest między innymi wykonanie nieautoryzowanego przelewu bez wiedzy klienta banku.

Poniżej znajduje się funkcja odpowiedzialna za przechwytywanie wiadomości SMS.

```

sms:
PUSH {R4-R7,LR}
SUB SP, SP, #0x44
ADD R5, SP, #0x59+var_28
MOVS R7, R0
MOVS R1, #2
MOVS R0, R5
BLX _ZN9TBufBase@C2Ei; TBufBase@::TBufBase@ (int)
LDR R0, =dword_7992F020
BL nullsub_5
MOVS R3, R0
MOVS R1, R3
MOVS R0, R5
MOVS R6, R7
BLX _ZN17DesB@4CopyERK7TDesC16; TDesB@::Copy (TDesC16 const&)
ADDS R6, #0x2B; '('
MOVS R1, R7
MOVS R3, #2
ADDS R1, #0x24; '$'
MOVS R0, R6
MOVS R2, #0x10
STR R3, [SP, #0x59+var_5B]
BLX _ZN7RSocket@4OpenER11RSocketServjjj; RSocket@::Open (RSocketServ &, uint, uint, uint)
CMP R0, #0
BEQ loc_7992F0BC

```

```

ADD R4, SP, #0x59+var_50
MOVS R0, R4
BLX _ZN17TSmsAddr@C1Ev; TSmsAddr@::TSmsAddr@ (void)
MOVS R0, R4
MOVS R1, #4
BLX _ZN17TSmsAddr@16SetSmsAddrFamilyE14TSmsAddrFamily; TSmsAddr@::SetSmsAddrFamily (TSmsAddrFamily)
MOVS R0, R4
MOVS R1, R5
BLX _ZN17TSmsAddr@12SetTextMatchERK6TDesC9; TSmsAddr@::SetTextMatch (TDesC9 const&)
MOVS R0, R6
MOVS R1, R4
BLX _ZN7RSocket@4BindER9TSockAddr; RSocket@::Bind (TSockAddr &)
CMP R0, #0
BNE loc_7992F0BB

```

```

ADD SP, SP, #0x44
POP {R4-R7,PC}

```

```

MOVS R4, R7
ADDS R4, #0x3B; 'B'
MOVS R0, R4
BL sub_7992FF10
MOVS R5, #1
STR R5, [R0]
ADDS R2, R7, #4
MOVS R0, R6
MOVS R1, #1
MOVS R3, R4
STR R5, [SP, #0x59+var_5B]
BLX _ZN7RSocket@15IoctlEjR14TRequestStatusP5TDesBj; RSocket@::Ioctl (uint, TRequestStatus &, TDesB *, uint)
STR R5, [R7, #0x34]
MOVS R0, R7
BLX _ZN7CActive@9SetActiveEv; CActive@::SetActive@ (void)
B loc_7992F0BB

```

Poniżej fragment kodu źródłowego. Opis wszystkich API pozwalających na obsługę SMS znajduje się tutaj http://wiki.forum.nokia.com/index.php/SMS_Utilities_API.

```

void SMSRead ()
{
    TBuf8<2> matchTag;
    LIT8 (KTag1, "");
    matchTag.Copy (KTag);
    iReadServer.Connect ();
    TInt err = iReadSocket.Open (iReadServer, KSMSAddrFamily,
        KSockDatagram, KSMSDatagramProtocol);
    if (!err)
    {
        TSmsAddr smsAddr;
        smsAddr.SetSmsAddrFamily (ESmsAddrMatchText);
        smsAddr.SetTextMatch (KTag1);
        TInt bindErr = iReadSocket.Bind (smsAddr);
        if (!bindErr)
        {
            sbuf () = KSockSelectRead;
            iReadSocket.Ioctl ( KIOctlSelect, iStatus, &sbuf, KSOLSocket);
            iRead=ETrue;
            SetActive ();
        }
    }
}

```

Fragment kodu odpowiedzialny za odczytanie każdej wiadomości SMS.


```

Open_or_create_DB+A0 BLX _ZN4User12LeaveIfErrorEi ; User::LeaveIfError(int)
Open_or_create_DB+A4 B loc_791AA9A4

```

Aplikacja w celu zapisania lub odczytania danych z lokalnej bazy SQL buduje zapytanie. Poniżej przykład zapytania SQL do tablicy tbl_phone_number.

```

SELECT * from tbl_phone_number WHERE phone_number LIKE 'numer'

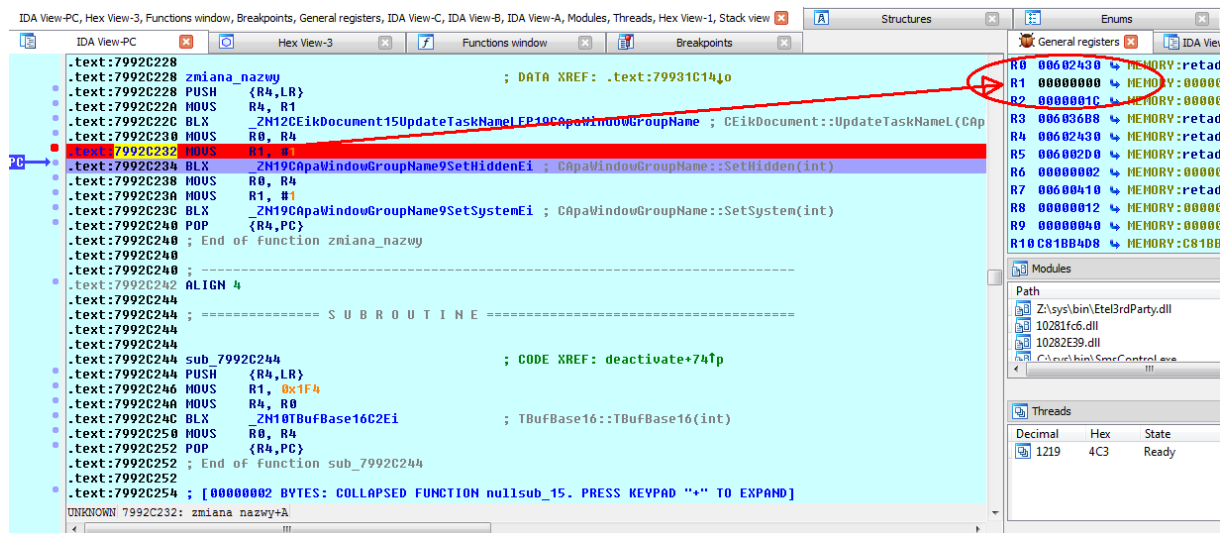
```

Monitor z konfiguracją

W kodzie aplikacji znajduje się ukryte okno (Debug), które wyświetla aktualną konfigurację. Okno to można czasami zauważyć na zainfekowanym telefonie – jednak jest to bardzo krótkie mignięcie.

Włączenie na stałe okna realizujemy w następujący sposób.

1. CAppWindowGroupName::SetHidden(EFalse) – rejestr R1 ustawiamy na 0



2. Następnie modyfikujemy RWindowTreeNode::SetOrdinalPosition (rejestr R2) ustawiamy pozycję z-order.
3. Na ekranie urządzenia mobilnego pojawi się okno debug z aktualną konfiguracją.

