IOActive Security Advisory

Title	Authentication Bypass In Tranax Remote Management Software
Severity	Critical
Date Reported	04/05/10
Discovered by	Barnaby Jack

Affected Products

MB1700/MB1700W. Firmware version: `V020102W.bin`

Additional models that utilize the same RMS code base are presumed to be affected.

Description

The Tranax Remote Management Software (RMS) allows for the administration of common Automated Teller Machine (ATM) tasks from a remote location.

To successfully authenticate to a remote ATM, both the serial number and the RMS password are required. Due to an implementation flaw when verifying these credentials, it is possible to craft a request that bypasses all authentication measures, which means that remote management tasks can be performed with invalid credentials.

The RMS interface is enabled by default on a typical ATM installation.

Technical Details

All RMS communication between the host and the ATM is encrypted with a custom encryption wrapper that implements an XOR lookup table.

An RMS encrypted packet uses the following format; values are represented in hexadecimal:

```
Length    Content  Name       Description
1 byte    02       LeadByte   Signifies start of RMS data
2 bytes   XX XX    CryptLen   Length of encrypted data
1 byte    XX       Seed       Encryption seed
variable           Data       Encrypted Data
1 byte    03       TailByte   Signifies end of RMS data
1 byte    XX       LRC        Longitudinal Redundancy Check
```

The following decrypted packet contains a legitimate ATM initialization request:

```
I5555555555666666.

Length    Content         Name      Description
1 byte    'I'             Request   Request type (Initialization)
10 bytes  '5555555555'    Serial    ATM serial number
6 bytes   '666666'        RMSpass   RMS password
1 byte    NULL            NULL      NULL terminator
```

When the ATM receives this request, both the serial number and RMS password are verified, and the request type is executed.

The code that parses ATM credentials expects RMS requests to adhere to the above standard; it does not take into account the possibility of malformed serial numbers or passwords. By appending an additional byte to the authentication string, the credential comparison code parses the serial and password incorrectly, and allows the request to go through.

The following request, once encrypted correctly, will force an ATM initialization with obviously incorrect credentials.

```
Length    Content         Name      Description
1 byte    'I'             Request   Request type (Initialization)
10 bytes  'EEEEEEEEEE'    Serial    ATM serial number
7 bytes   'BBBBBBB'       RMSpass   RMS password
2 bytes   00 00           NULL      NULL terminator
```

Although the initialization request is used as an example, all valid request types are accepted.

Proof of Concept

A proof-of-concept tool and accompanying source code that demonstrate this vulnerability are available on request.