NFB

Near-Field Beers

RFID everywhere!





Mifare Classic everywhere!

seckey A	es key ^F B ncou Yanu Francou	res
sec key A r	es key Bincou -Ydani Francoi	res
	. Ydann Francou	
ooo fffffffffff	1rev@ffffffffffffb.ne	t, Beho
001 fffffffffff	l fffffffffff	1
002 fffffffffff	1 ffffffffffff	1
003 fffffffffff	1 ffffffffffff	1
004 ffffffffff	1 ^D	1
005 fffffffffff	ıD raffffffffffff	1
006 fffffffffff	1D raffffffffffffff	1
007 fffffffffff	lobm fffffffffffff	1
008 fffffffffff	1_ _fffffffffff	1
009 fffffffffff	1 ffffffffffff	1
O10 fffffffffff	1 ffffffffffff	1
Oll fffffffffff	l ffffffffffff	1
012 fffffffffff	l ffffffffffff	1
013 fffffffffff	ı İ fffffffffffff İ	1
014 fffffffffff	l ffffffffffff	1
015 fffffffffff	l 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>0
ISO14443 Type A UIDs in one go"},
  {"sim", CmdHF14ASim, 0, "<UID> -- Simulate ISO
14443a taq"},
+ {"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, O, NULL}
```

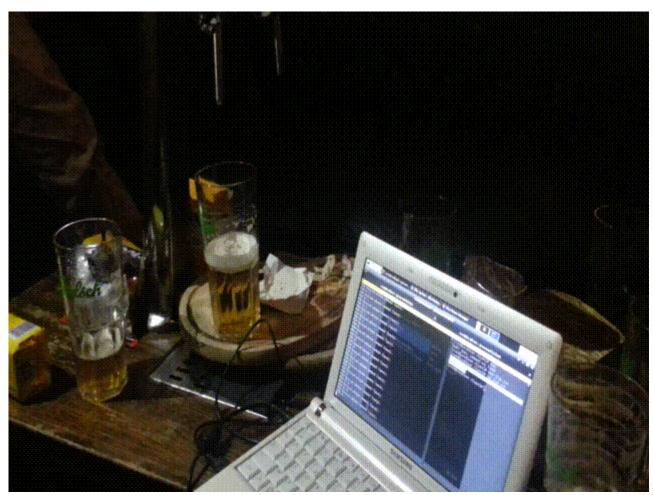
Proxmark FTW

```
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





The end

 No beers were harmed during the making of the demonstration

