Insecurity at U of M

HKN Tech Talk

Jon Oberheide
CTO, Duo Security

Who am I?

Jon Oberheide

- BS, MS, and PhD from U of M
- Broken a bunch of stuff here and elsewhere

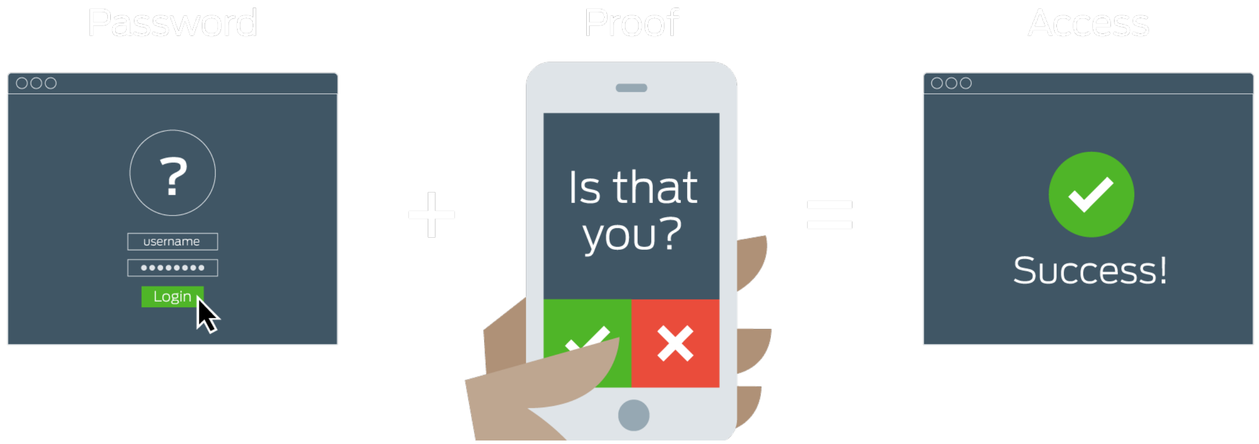
Duo Security

- Ann Arbor-based, founded in 2009, growing rapidly!
- Technology, customers, culture!
- RECRUITING, RECRUITING, RECRUITING!

What is Duo?

Two-factor authentication

- What you know + what you have
- Stops NSA, China, hoodlums



Why should we care about security?

Security at the University

EDUs are soft targets

- With lots of valuable personal information
- Including your personal information
- In your best interest to ensure (responsibly!) that this information is securely protected

Let's explore a few vulnerabilities at UM

- Doesn't take l33t skillz
- Just a bit of curiosity and free time

Agenda

Cosign authentication bypass

Mcard forgery attack

Physical security of CSE

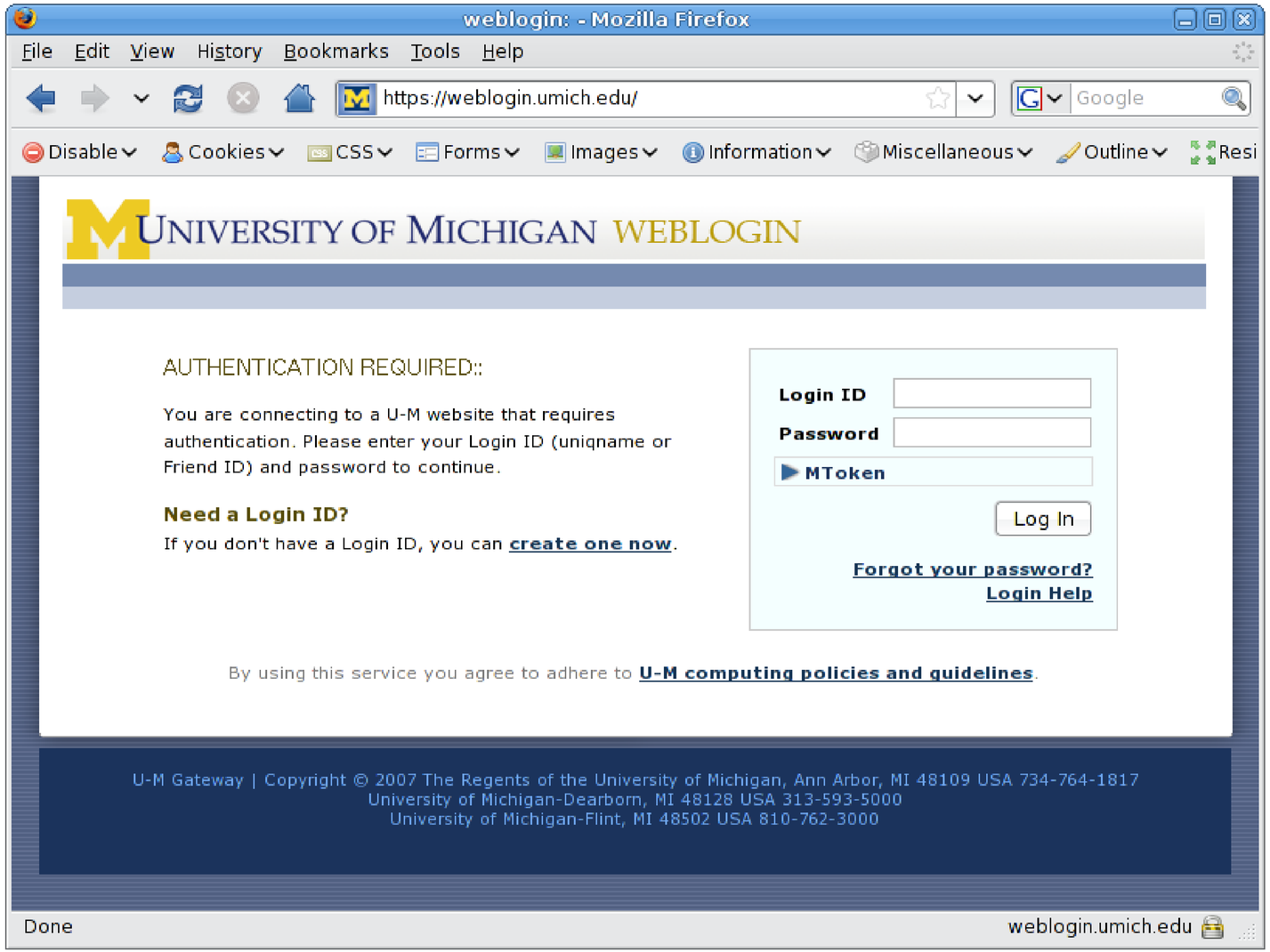
Other things...

Cosign SSO Vuln

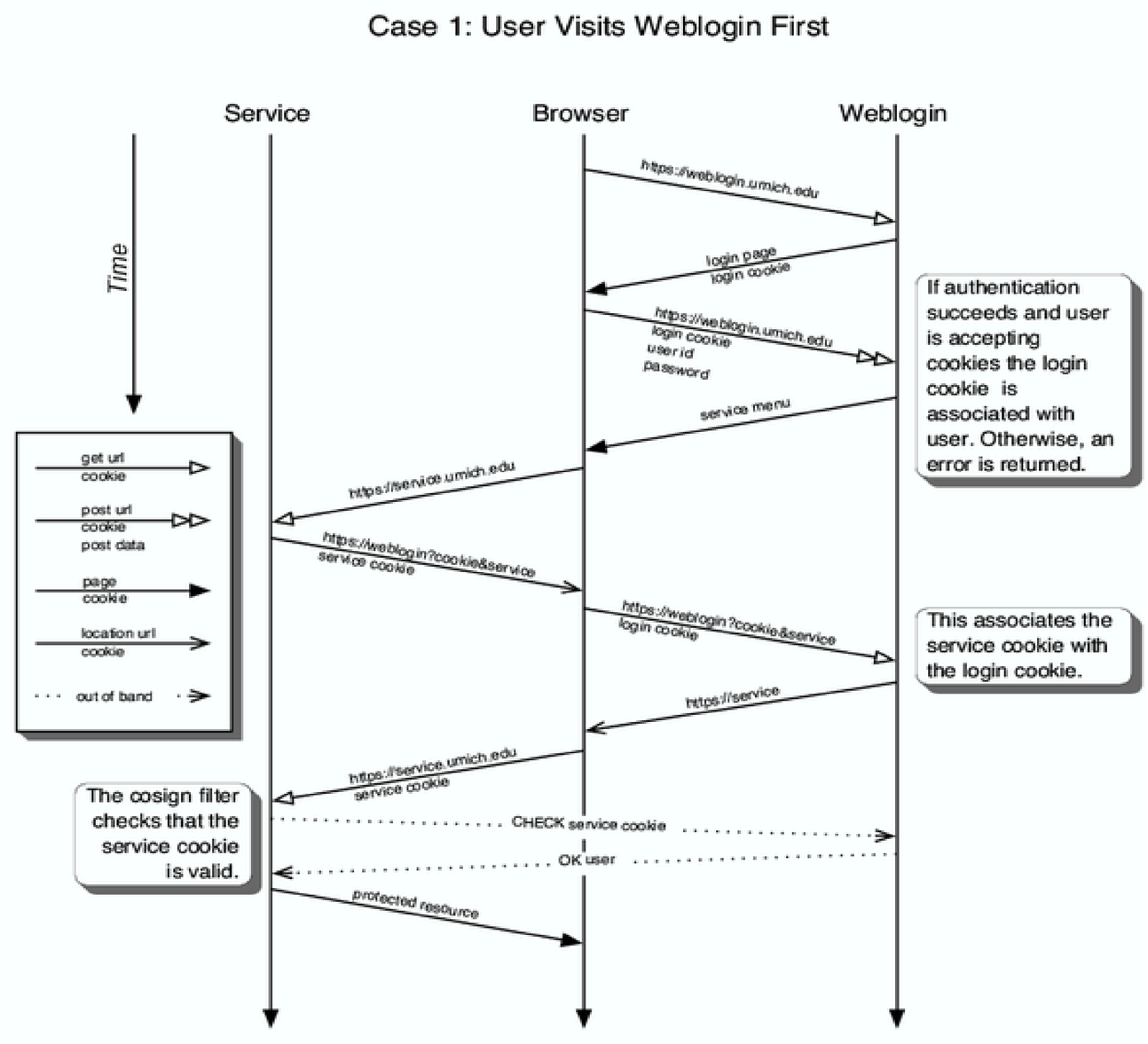
Cosign Single Sign-On Authentication

- Deployed extensively at UofM and many other educational institutions and orgs around the world
- Protects web mail, wolverine access, mfile, mpathways, and umm...everything

This is Cosign



Cosign Architecture



Cosign Architecture

Cosign-enabled Webservers (eg. web mail)

CGI Web Frontend (weblogin.umich.edu)

Backend Daemon (cosignd)

Hugely simplified operation:

- Webserver redirects user to CGI web frontend for auth
- CGI authenticates user, then communicates with daemon
- Webserver checks with daemon to verify authentication

Cosign Protocol

Protocol between CGI and daemon:

- Plaintext-based protocol, SMTP-ish
- Commands terminated '\n', '\r', or '\r\n'

Example commands:

- CHECK: check whether a given cookie is valid in daemon's backend db (eg, if a user is already auth'ed)
- LOGIN: tell daemon a user has auth'ed
- REGISTER: associate service cookie with user's global cosign cookie

Typical Login Session

Exchange between CGI and daemon:

- CHECK cosign=X
- LOGIN cosign=X 1.2.3.4 username
- REGISTER cosign=X 1.2.3.4 cosign-servicename=Y

where X and Y are randomly generated base64 strings of length 128, username is the principal that successfully authenticated, and 1.2.3.4 is the IP associated with the user

Cosign Audit

Had some free time one weekend

C-based CGI component

- It's C-based parsing of HTTP, it _has_ to be buggy
- Yet, surpisingly well-coded

String-based communications protocol

- Typically not as fruitful as binary proto, but eh
- Maybe some unsafe string handling in daemon

Audit Findings

Lots of NULL derefs triggerable in CGI

- No big deal, CGI spawned off on each request

Buffer overflow in daemon

- Unfortunately, not enough for stack smashing, only can overflow into subsequent buffer on stack
- Overflows into krb_ticket var which is unlink()'ed
- May be exploitable with other archs/stack layouts

No exploitable common C-based vulns,
anything application specific???

Initial CHECK Command

When you hit weblogin.umich.edu

- CGI takes your HTTP cookie and sends to daemon
- **`send_daemon("CHECK %s\r\n", cookie);`**
- Needs to check whether presented cookie is valid

Remember our protocol line terminators???

- '\n', '\r', '\r\n'

Your security-sense should be tingling now...

Embedded Terminators

Can't embed '\n' in HTTP header fields (cookie)

But carriage returns '\r' are completely legal

- Set HTTP cookie
  - cosign=blah\rred\rgreen\rblue
- CGI sends to daemon:
  - CHECK cosign=blah\rred\rgreen\rblue\r\n
- Daemon interprets as four separate commands due to '\r'
  - ”CHECK cosign=blah”, ”red”, ”green”, ”blue”

Uh-oh...

Just achieved arbitrary Cosign command execution!

- From a completely unauthenticated web user

How to exploit?

- Need to replicate the standard login procedure
- Inject LOGIN, REGISTER command sequence

Exploitation

Example malicious cookie:

- cosign=X\r
  LOGIN cosign=X 1.2.3.4 username\r
  REGISTER cosign=X 1.2.3.4 cosign-servicename=Y
- Replace username with the uniqname you want to authenticate as ('marysuec', 'grue', ...)

Success!

- Steal personal information, change your grades, read email (and that's the tame stuff...)
- MPathways is where the fun is

Agenda

Cosign authentication bypass

Mcard forgery attack

Physical security of CSE

Other things...

Magnetic Card Security

Magnetic Cards

- Trivial to clone given physical access

House key analogy

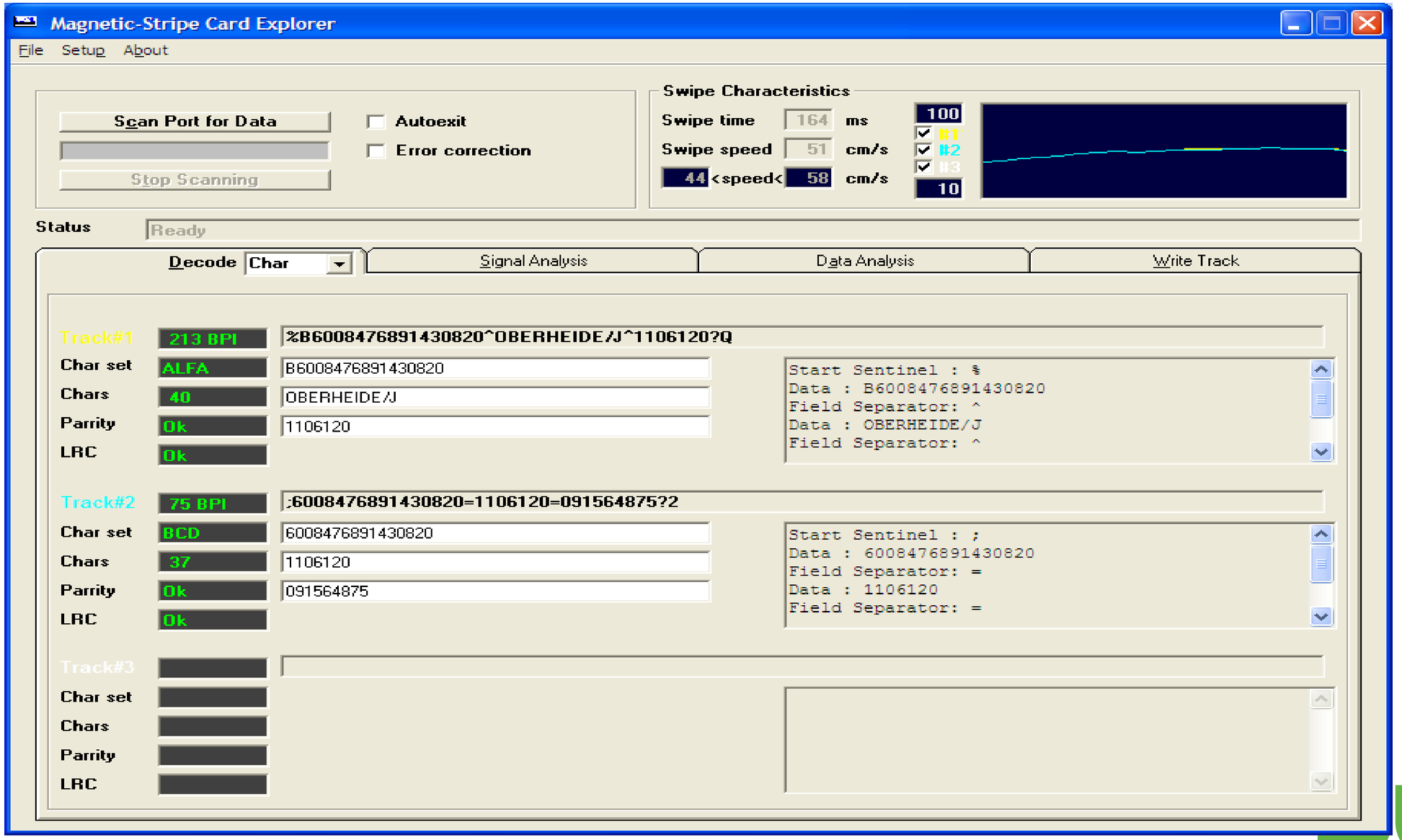
- Copies made at hardware stores
- If attacker obtains your physical key, he can make a copy and break into your house
- Obvious!

Mcard Hacking

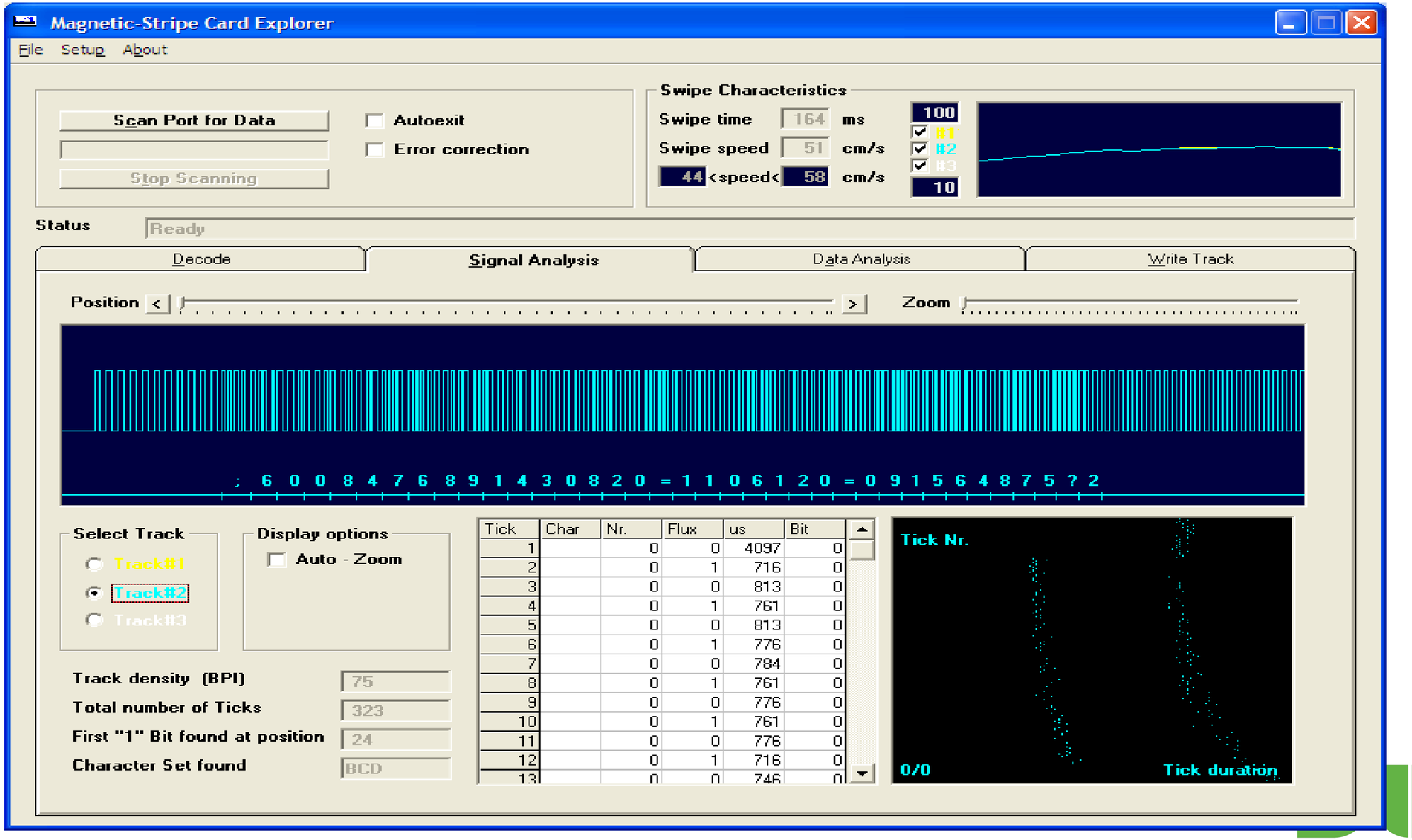
So, can we forge a Mcard without a physical copy/clone of it?

First, lets take a look at what the Mcard magnetic stripe contains...

Mcard Format



Mcard Format



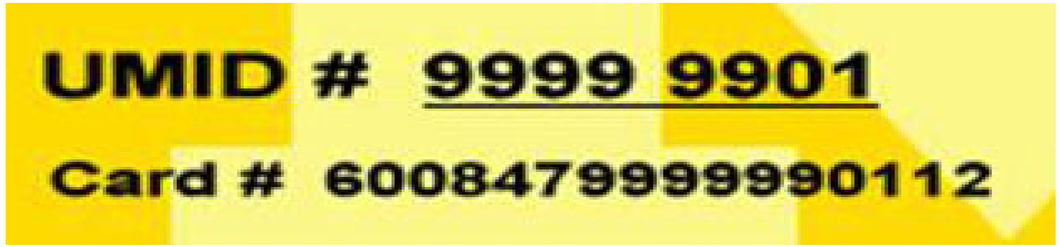
Mcard Format

What do the Mcard readers care about?

- Only data from track 2 is read
- Only the account number portion is verified
- Lots of trial and error...and candy!

Mcard account number

- 16 digits, listed on front of card
- Static 6-digit prefix: 600847
- Then 8-digit UMID: 99999901
- Then card revision number digit
- Finally, Luhn checksum digit



Mcard Forgery Attack

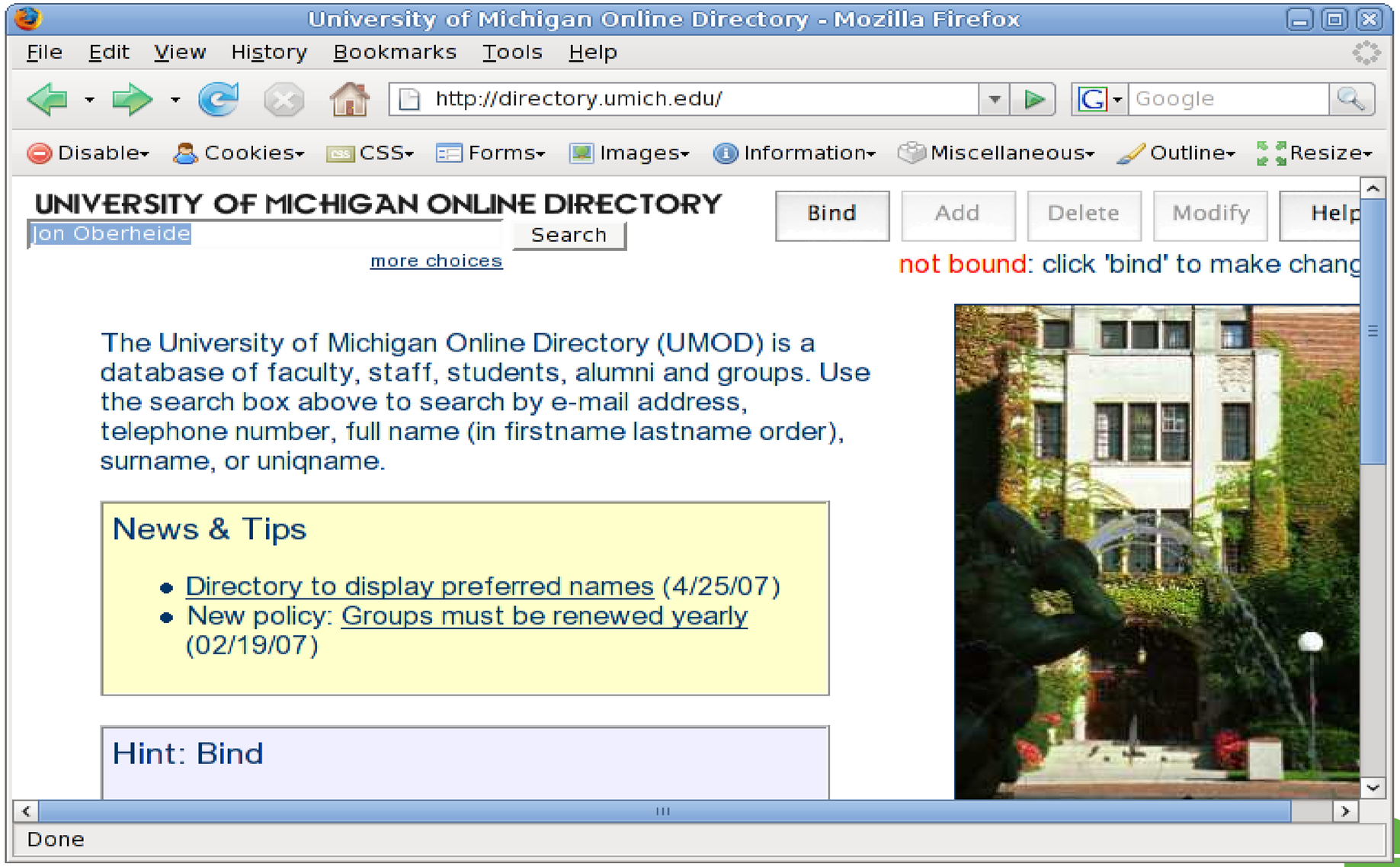
Implications?

The account number is completely predictable.
We can forge arbitrary Mcards!

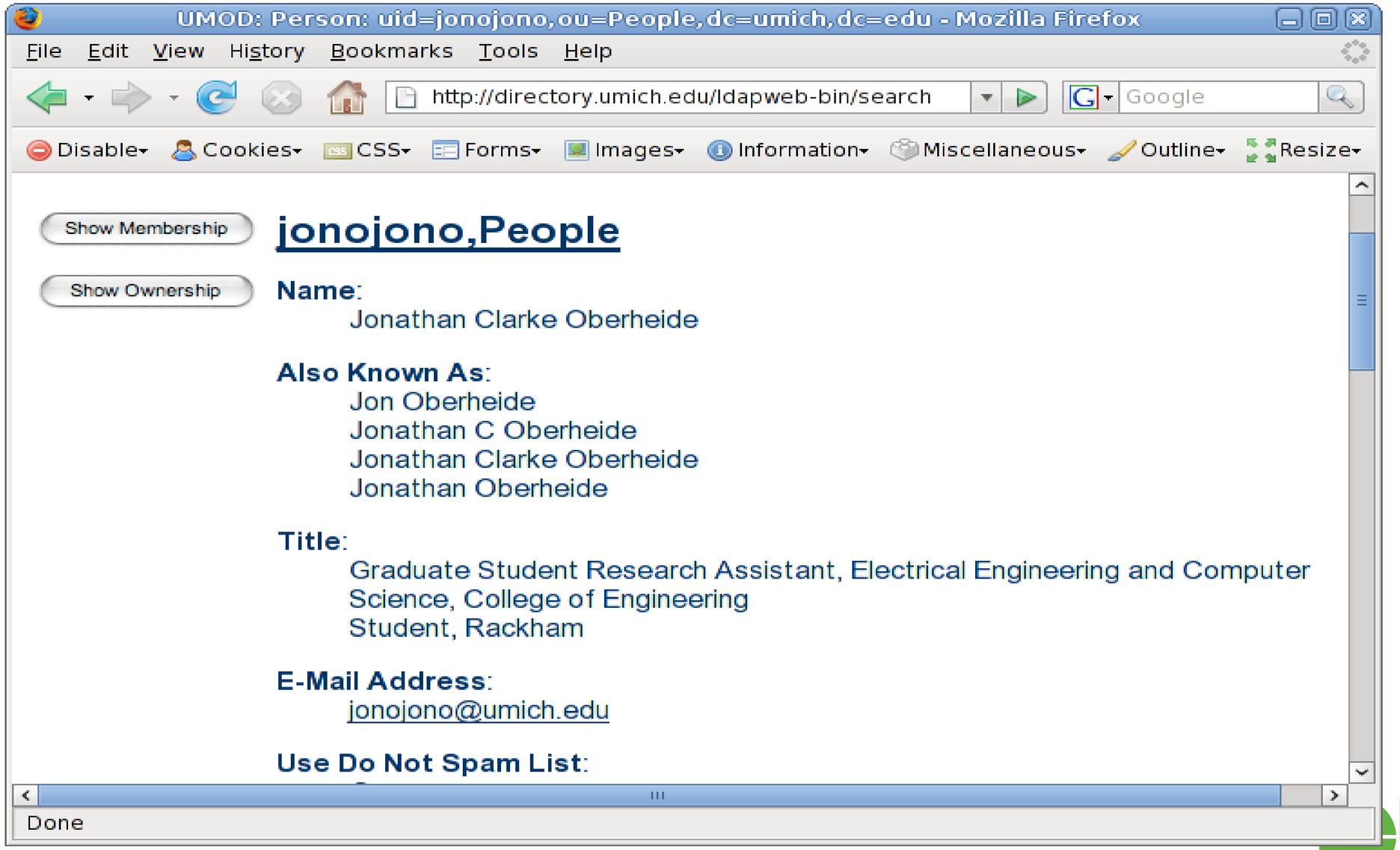
All we need:

- Victim's UMID
  - Public info, lookup via web or uns service
- Revision number
  - Usually 1 or 2, worst case ten guesses
- Luhn checksum
  - Standard algorithm, trivially calculated from other 15 digits

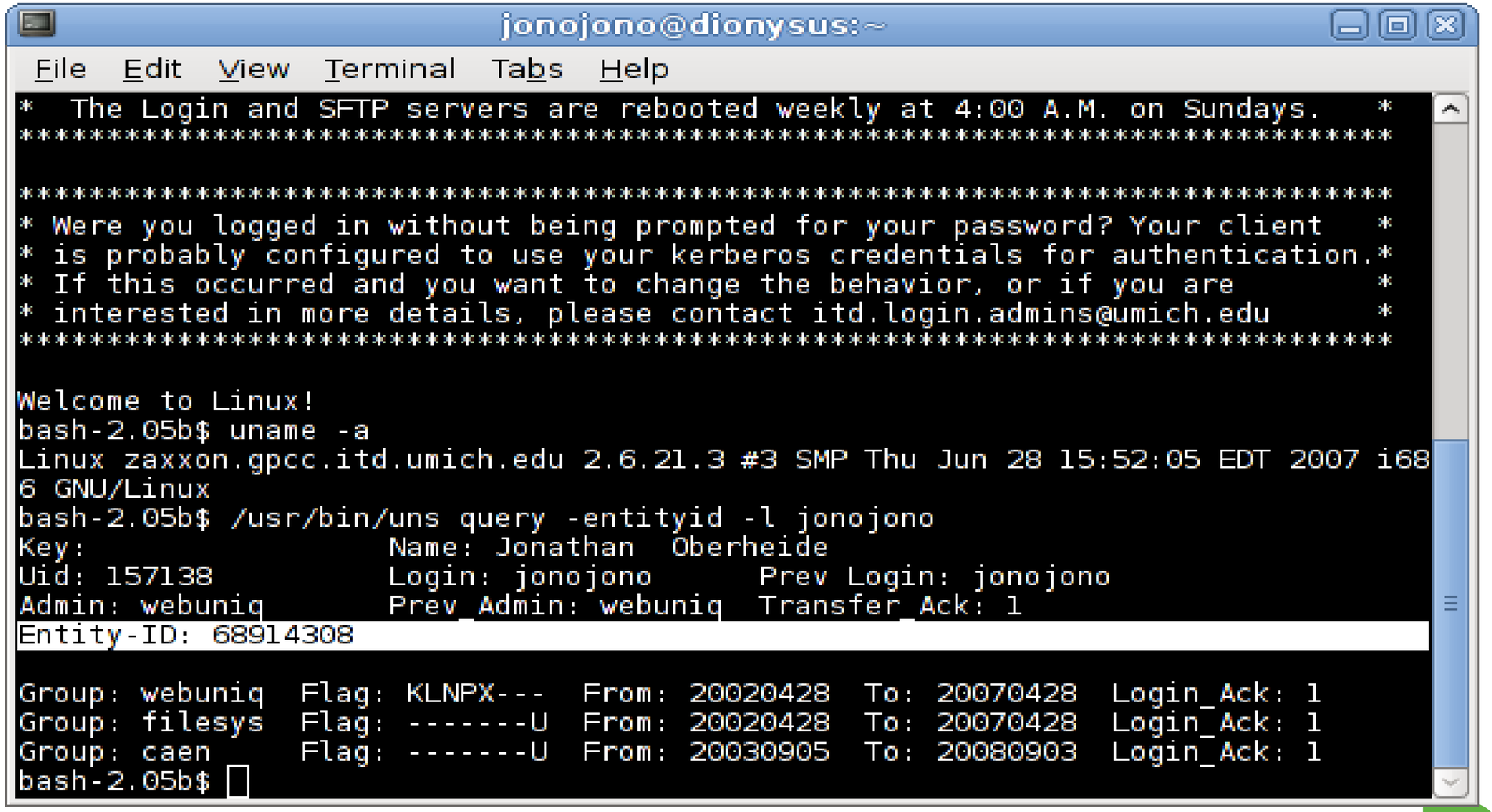
Find Target's Uniqname



Find Target's Uniqname



Find Target's UMID

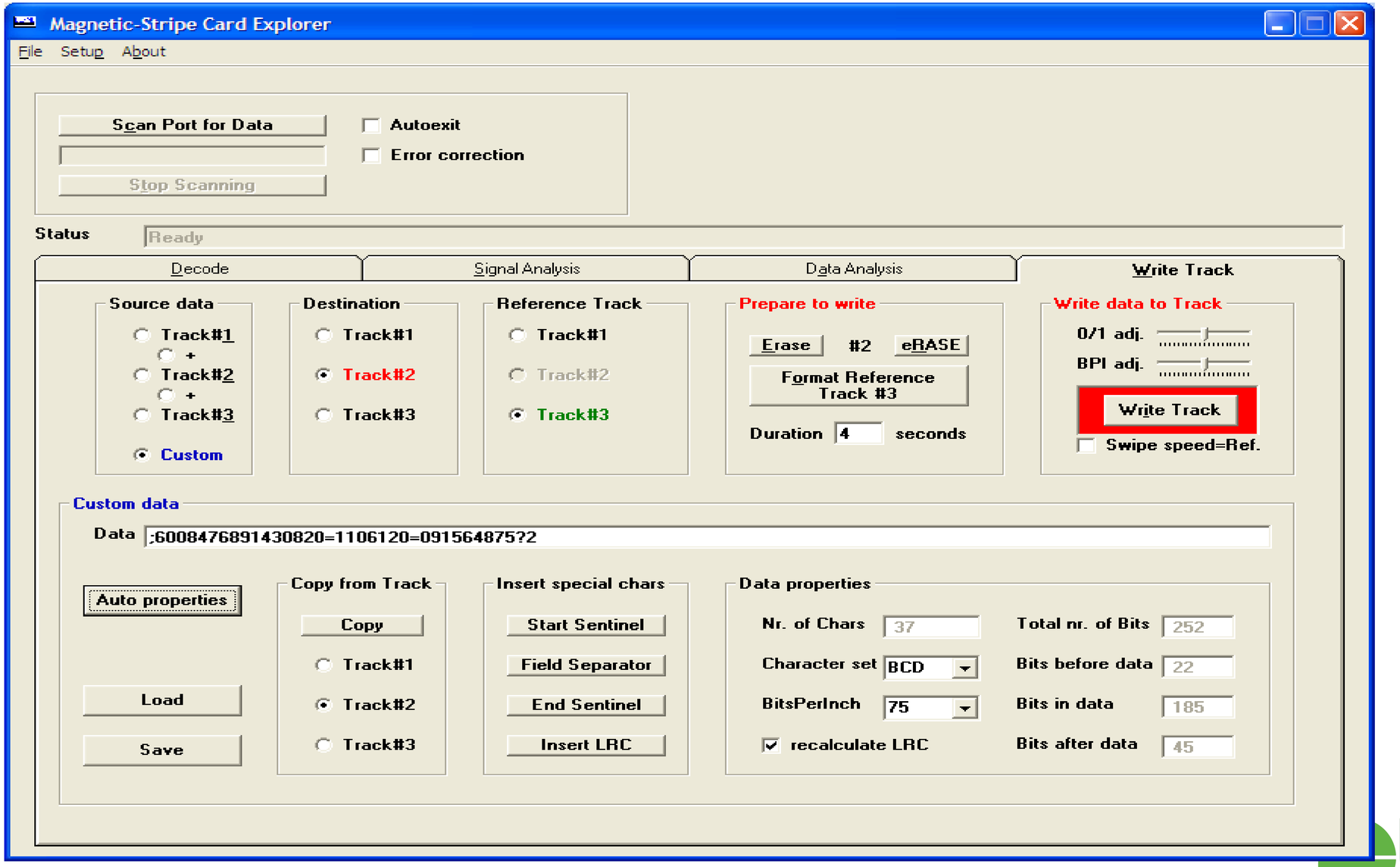


Derive Card Number

CARD NUMBER:

600847 + UMID + Revision + Luhn

Write Magnetic Card



PROFIT!!!

BUT HOW???

Badness

Steal Entree Plus/meal plans

Buy iPods (Ugo's, showcase)

Break into dorms and wreak havoc

- Frame your enemies!
- Graffiti incident

Gain physical access as your favorite Umich official

- marysuec, grue, mabdelah

Steal expensive equipment

- Target building/facilities managers

More Badness

Arbor Lakes

Chemicals, hospital drugs, etc

- Hopefully protected by another layer of secure access?

TCF-linked ATM access

- Mcard acts as ATM card at TCF bank
- Already compromised “what you have”
- Fairly easy to obtain “what you know”
- ATM shoulder surfing + social engineering

Solutions

Vulnerability stems from predictability

- Don't just read the card number
- Add extra random data for verification

9 random digits added on track 2

- 10^9 = 1 billion tries to brute force
- Making a card takes ~5 minutes, attack infeasible!

However...

- Impractical to reissue over 110k cards
- Gradual replacements for high risk, but no flag day

Agenda

Cosign authentication bypass

Mcard forgery attack

Physical security of CSE

Other things...

CSE Building

- Let's look at a real world example
  - CSE building!
- But, with a hypothetical scenario
  - Plutonium in my office desk drawer
  - Evil terrorist CSE scholars want to obtain it
- How robust is the physical security that protects our valuable research in CSE?
  - Hard balance between an open university building and a locked down supermax facility
- PHOTO TOUR!

Entrance

Vomit Covered Reader?

Inside!

Security Cameras

Security Cameras: DoS

- We don't want to be recorded on video when breaking into CSE!
- Most of these cameras are very simple
  - Take video, fling across network
  - Slim processing capabilities
- But usually have some sort of controlling interface (web, telnet, whatever)
- Simple DoS will make it fall over, drop frames/packets, etc

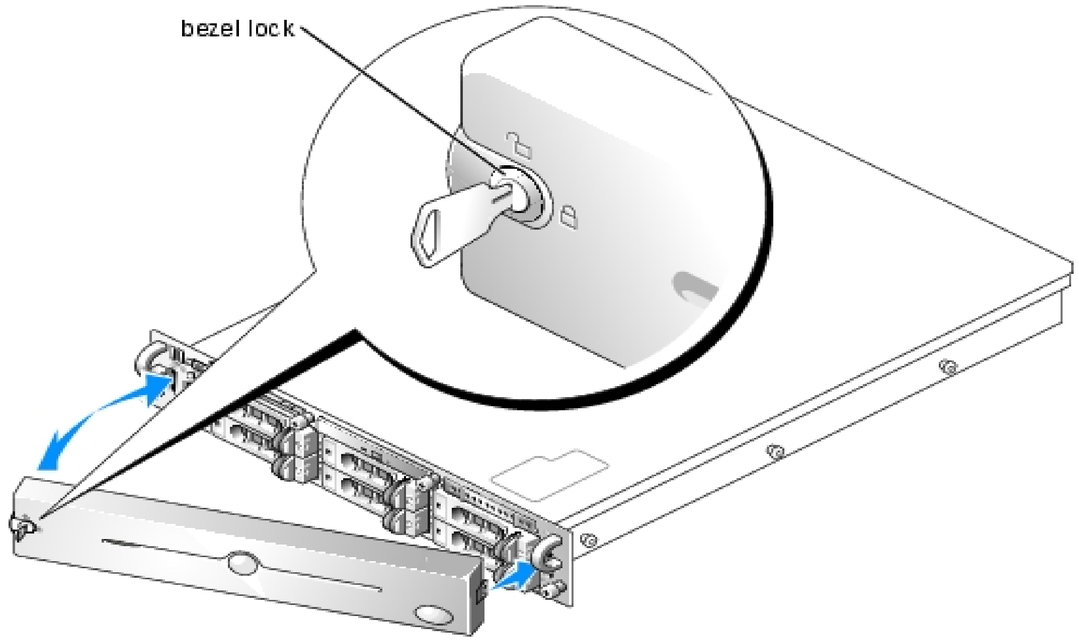
Security Cameras: 802.11 Attack

- Like ethernet, 802.11 has no link-layer authentication
  - In most situations…
- How to kick someone off WiFi network?
- Just spoof 802.11 Deassociation frame
- Camera will have to reassociate and be unable to transmit video

Elevator Access

Elevator Access

- Has special key for maintenance, individual service, lockdown, etc
- Could we pick this lock?
  - Certainly!
- But why pick it when we already have keys to operate it?!?
  - Dell server bezel keys
  - Pretty much any small key can rake it



Elevator Action!

Or, get past locked barriers

- Stairway barrier protected with simple padlock
- Shimming is easy, and fun!
- Bonus points if you use a pop can!

4th Floor

Hallway

Office Lock

Office Locks

- The locks on all our offices (and University-wide locks) are fairly good!
- Schlage Everest with restricted keys

Schlage Everest

- 6 pin - tumbler locks (B145)
- Restricted keys
  - You CAN'T get key blanks
  - Distributed directly by Schlage
  - Also, restricted by patents to prevent 3rd-party blanks
- So master-key creation attacks are foiled
  - Unless we mill our own blanks, but VERY hard
- But can't we still pick the tumbler?

Picking Everest

- Yes, but harder than normal
- The dreaded finger pin!

Picking Everest

- Just need special tool
  - Make your own or buy one for ~$30

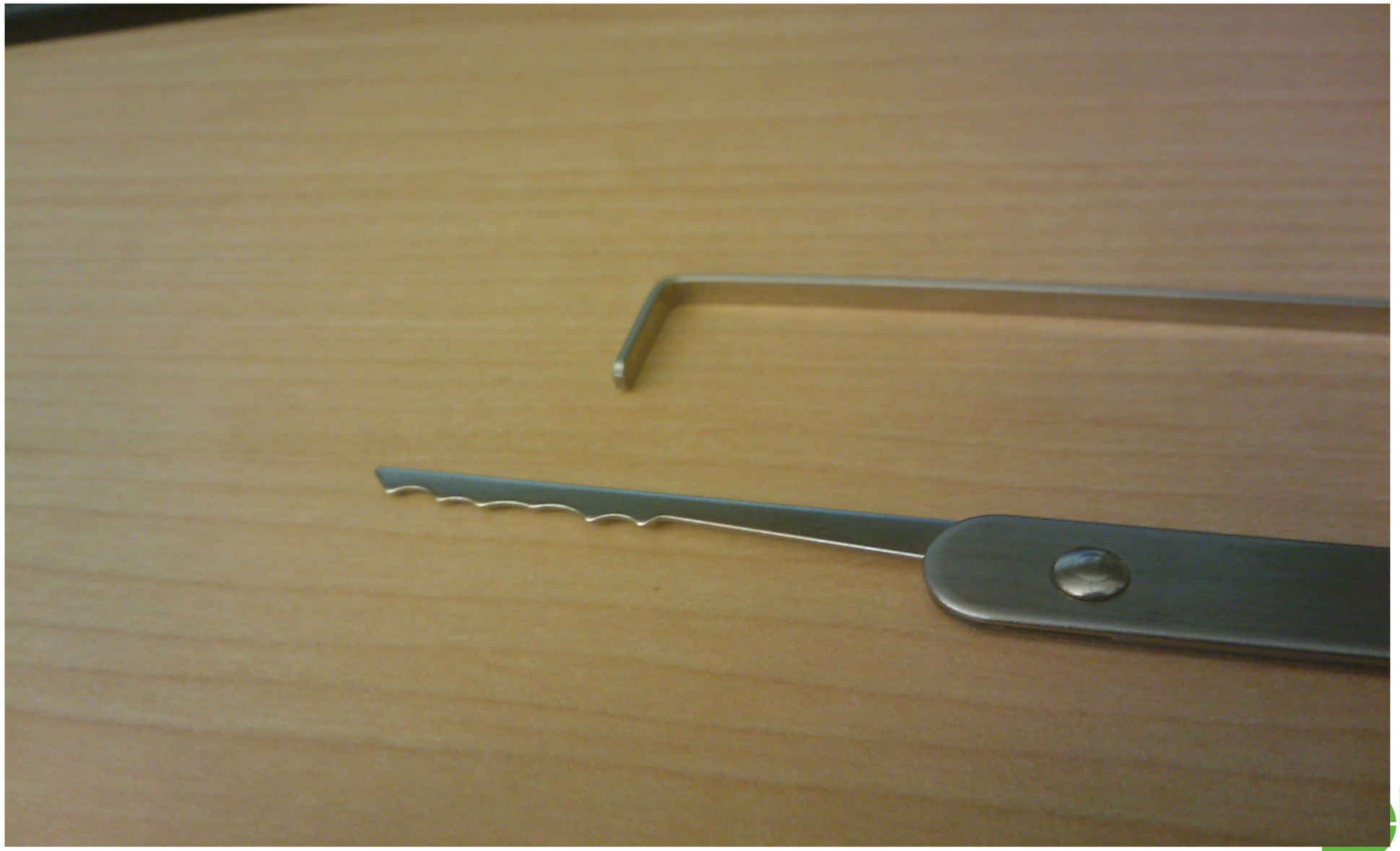
Everest Pick

Open Sesame

Terrorist Bunker

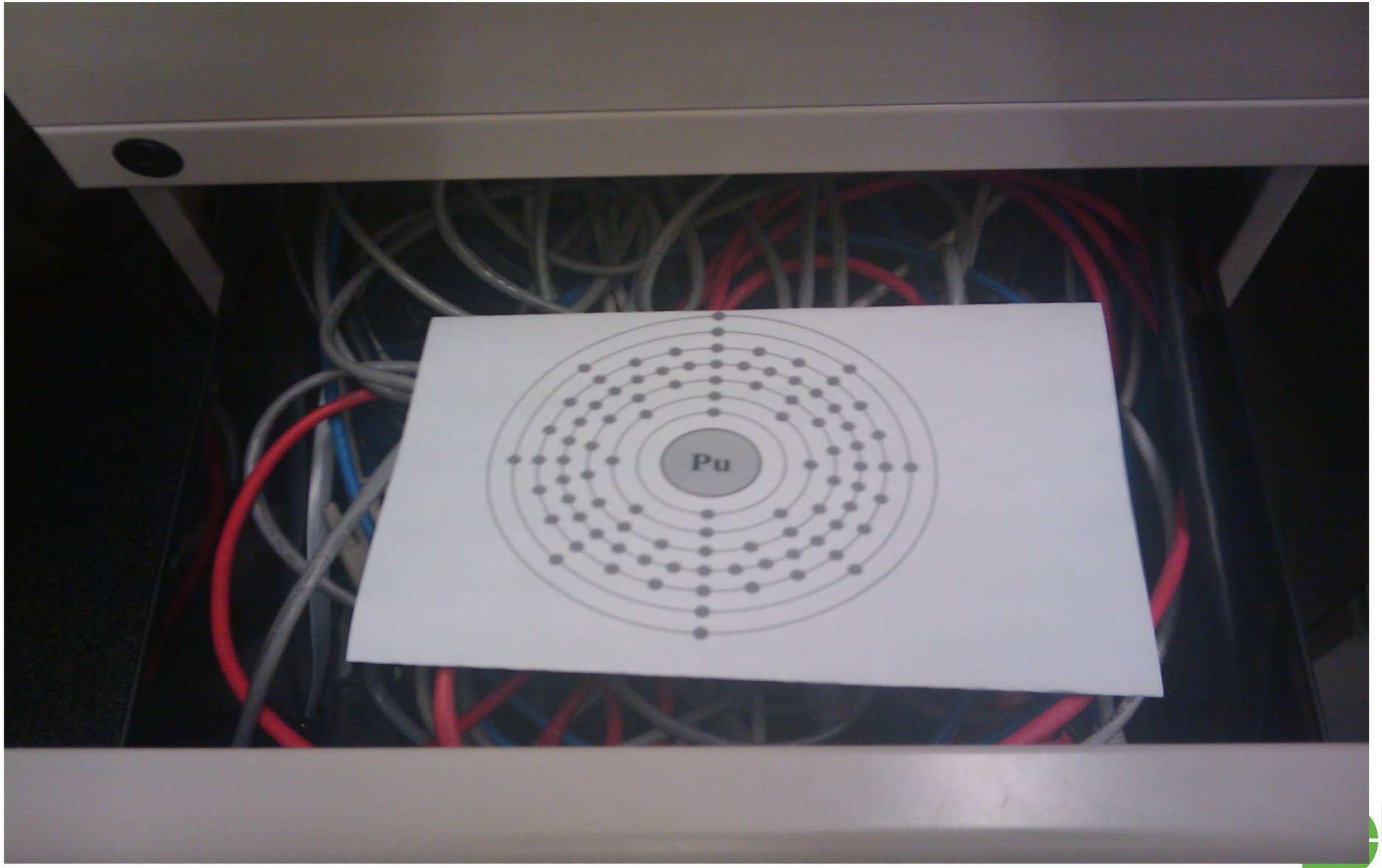
Locked WMD Office Drawer

Rake Pick



Boink

Plutonium!



Agenda

Cosign authentication bypass

Mcard forgery attack

Physical security of CSE

Other things...

Duo Tech Talks

^ Car hacking
from these clowns

http://www.meetup.com/Duo-Tech-Talks/

ARBSEC

http://arbsec.org

A2 New Tech Meetup

http://a2newtech.org

Thank you

QUESTIONS?

Jon Oberheide
jono@duosecurity.com
https://www.duosecurity.com