Smart-Phones for Two-Factor Authentication?

- Passwords, credit card numbers etc. neither safe for entry into smart-phones nor into PCs.

- Will cost web retailers too much
  - already too much fall-off of customers during check-out phase of web-purchases (where’s my wallet?) – requiring another device makes things much worse (now where’s my phone – in the car?)

- Un-workable within enterprises – e.g. financial institutions
  - are IT groups going to support myriad of employee-owned devices – where are the phones – at home, in the car, in a meeting room, in a restaurant etc.? What happens when lost, stolen, damaged, disconnected, given to kids?
  - are IT groups going to purchase company smart-phones for employees – have them carry personal and business devices – where are they - are they fully charged etc.

*Un-workable, costly band-aid, to underlying flawed design*
Asimov’s Law privileges humans over smart machines

- Law 1: robots must not harm humans (so neither should computers)

People care strongly about Asimov’s Law

- Elon Musk on Artificial Intelligence (AI): “summoning the demon”
- Stephen Hawking on AI: “will destroy humanity”
- ordinary users currently take Law into their own hands: will tape over camera lenses on laptops, will tape foam over microphones – as they assert ultimate privilege of determining what gets recorded or not in their environment – not the smart machine

Will reveal computer designs that adhere to Asimov’s Law
- smart machines will not be privileged to handle: passwords, credit card numbers etc.
Protection of Credentials & Transactions

Credentials transactions bypass CPU and malware

- New NIC functionality and connection redirects credentials request message (e.g. http 401 message) around CPU/malware to new display on keyboard
- Conventional keyboard connection is blocked until transaction is complete
- **No smart-machine/malware access to credentials**
Internalized configuration needs external lamp to indicate security mode operation.

Pop-up display overlay and external lamp are *exclusively* under control of NIC.

- Display overlays common on monitors – e.g. brightness/contrast controls
- Display overlays not accessible to OS – e.g. Print Screen can’t capture
- For tablets, lamp indicates keyboard touch-input is disconnected from CPU
Humans Validate/Complete Transactions

- Prevents smart-machine/malware from completing transactions
  - once user signs into bank/broker, require user to confirm risky transactions
  - especially needed for transfers to 3rd party accounts or stock purchases
  - not strictly necessary for modest transactions to well-established billers
- Prevents malware from tampering with purchase and shipping details
  - user alerted to unintended purchases and delivery elsewhere
- Memory for frequently used sites/certificates can alert to phishing

- [www.bank1.com] Transfer $128.64 to 8192-4096-2048 – enter PIN: <f1 - help>
- [www.broker1.com] Purchase 400 shares of Telus Preferred – enter PIN: <f1 - help>
- [chapters.indigo.ca] $64.16 purchase to 128 Byte St. – enter MasterCard: <f2 - more>
- ????www.bank1.com???? Enter password: <?? f8 – warnings ??>
User Termination of Port 0* – Solves the Turing Test

Port 0 dedicated to secure keyboard & display, CPU and O/S forever denied access to port 0

- Placing service on separate Port 0, makes it general, aiding many services
  - akin to DNS Port 53: helper for HTTP, HTTPS, FTP, POP, SIP etc.
- NIC functions block malware from ever using UDP/TCP Port 0
- Critical transactions involving Port 0 cannot be completed without human intervention and oversight!
- Secure user-agent can have factory certificate from dedicated CA
  - allows server to confirm PC has secure hardware
Credentials not actioned within server – not accessible by CPU/malware
  - therefore cannot exploit services on other servers using valid credentials

When server requires user authentication, it sends message to NIC to conduct transaction, and only receives a success/fail response
  - returned (http) authentication headers stripped by NIC – cannot reach CPU
  - could enhance CMU to clear credit-card transactions away from CPU

Port 0 separation would allow connection to central ID service
User can confirm web site name and/or welcome phrase, certificate

Can verify transaction details & confidently submit credentials

Provides two-factor authentication – conveniently within same device via independent means & “2nd factor” is fixed circuitry – not “smart”

Critical transactions just can’t complete without human oversight and explicit participation – only humans have this elevated privilege
  - CPU-OS can never complete a transaction – even if it were to know credentials!

**Fully compliant with Asimov’s design law**
- smart machines never privileged to handle password, credit card etc.
Going Forward Plan
Formula for Insertion into Industry Ecosystem

- Involves a limited amount of funding
- Requires only a modest amount of readily available talent
- Has an inconsequential barrier to entry
- Has only modest technical challenge & few/quick steps
- Follows a proven & well-traveled road to success
Zero barrier to IETF participation – Universities very common
Outline of IETF Submission Activity

- Partnering with McGill University
  - working with Advanced Networking Research Lab (ANRL)
  - Carlton Davis is lead researcher – has extensive protocol experience

- Drafting a protocol for secure credentials exchange around secure keyboard-NIC reference configuration
  - work is fully funded and well underway
  - “Port 0” protocol based on DTLS

- Looking for additional support to back IETF submission
  - e.g. from financial industry/other
**Server: S**

- Conventional HTTP/S connection established at port 80/443

**Client: C**

- Ignore Upgrade header

**Flow Chart Details:**

1. **Need credentials**
   - If No, **Client has Sec-NIC?**
     - If Yes, **Send Upgrade header**
     - If No, **Continue with conventional connection**

2. **Has Sec-NIC?**
   - If Yes, **Inform server**
   - If No, **Ignore Upgrade header**

**Flow Paths:**

- **Send Upgrade header**
  - **Upgrade header**
  - **Version and port #**
  - **Version & port received?**
    - If Yes, **Send version # and port #**
    - If No, **Send response**

- **Response received**
  - **Yes**
    - If No, **Send Upgrade header**
  - **Version & port received?**
    - If Yes, **Send version # and port #**
    - If No, **Send 303 message**

- **Version & port received?**
  - **Yes**
    - If No, **Version and port #**
  - **Version and port received?**
    - If Yes, **Send version # and port #**
    - If No, **303 Message received?**
      - If Yes, **Yes**
      - If No, **No**
Connection redirected to secure “Port 0” using DTLS

**Server: S**
- Send credentials request (e.g. 401 msg)
  - (wait)
- Received handshake?
  - Yes → Create DTLS tunnel
    - (wait)
  - No → Credentials received?
    - Yes → Send session cookie
    - No → Send credentials request
- Received request?
  - Yes → Send handshake
    - (wait)
  - No → Initialize handshake
    - (wait)
- Tunnel msg received?
  - Yes → Instantiate tunnel
    - Send credentials
    - Credential reply formed
      - (wait)
      - Yes → Cookie received?
        - Yes → Return to conventional connection with new session cookie
        - No → Return to conventional connection with new session cookie
      - No → Return to conventional connection with new session cookie
    - No → Send credentials
      - Credential reply formed
        - (wait)
        - Yes → Cookie received?
          - Yes → Return to conventional connection with new session cookie
          - No → Return to conventional connection with new session cookie
        - No → Return to conventional connection with new session cookie
  - No → Send tunnel creation message
- Instantiate tunnel
- Send credentials
- Credential reply formed
- Cookie received?
  - Yes → Return to conventional connection with new session cookie
  - No → Return to conventional connection with new session cookie
- (wait)

**Client: C**
- Send credentials request
- Received request?
  - Yes → Send handshake
  - No → Initialize handshake
    - (wait)
- Tunnel msg received?
  - Yes → Instantiate tunnel
    - Send credentials
    - Credential reply formed
      - (wait)
      - Yes → Cookie received?
        - Yes → Return to conventional connection with new session cookie
        - No → Return to conventional connection with new session cookie
      - No → Return to conventional connection with new session cookie
    - No → Send credentials
      - Credential reply formed
        - (wait)
        - Yes → Cookie received?
          - Yes → Return to conventional connection with new session cookie
          - No → Return to conventional connection with new session cookie
        - No → Return to conventional connection with new session cookie
  - No → Send tunnel creation message
- Instantiate tunnel
- Send credentials
- Credential reply formed
- Cookie received?
  - Yes → Return to conventional connection with new session cookie
  - No → Return to conventional connection with new session cookie
- (wait)
MILS: Multiple Independent Layers of Security
- NEAT: Non-by-passable, Evaluate-able, Always-invoked and Tamperproof

Research activity defined with McGill University
- designing embedded systems to be robust against exploits
- directly applicable to making NIC-keyboard tamper-proof
- demonstrate resilience/fail-soft against buffer-overflow attacks etc.
- many get-started ideas within IEEE World-CIS paper from 2012
- also applicable to internet-of-things

Full funding has been put in place at McGill University
- currently in getting-organized phase
Formula for Insertion into Industry Ecosystem

- Involves a limited amount of funding
- Requires only a modest amount of readily available talent
- Has an inconsequential barrier to entry
- Has only modest technical challenge
- Follows a proven & well-traveled road to success

Activity well underway: A Cybersecurity Game-Changer
Kevin O’Connor ...
- successful serial entrepreneur & zillionaire
- co-founder of DoubleClick
- seed investor in HotJobs, MeetUp etc.

... advises that to validate proposition value:
- ask the right questions and look for “of course” answers

Is there value to:
- keep passwords/credentials away from malware
- keep financial transactions away from malware
- being able to distinguish network requests emitted from a PC, as either coming from: malware on a CPU or a human at the keyboard

Of Course!
Firewalling: Passwords, Financial Transactions and Human Privileges from CPU Resident Malware

Jim McAlear, Carlton Davis – Dec. 11, 2014

jim.mcalear@ieee.org
carlton@cs.mcgill.ca