Safe Passage for Passwords and Other Sensitive Data

Jonathan M. McCune       Adrian Perrig
Carnegie Mellon University / CyLab

Michael K. Reiter
University of North Carolina at Chapel Hill

February 11, 2009
Input Security on the Web

- Keylogger
- or
- Screen Scraper

My info is going to my bank and only to my bank.

S - e - c - r - e - t

www.suntrust.com/
Input Security on the Web

My info is going to my bank and only to my bank?

Is my input really safe?

Keylogger
- or -
Screen Scraper

Trusted Monitor

S - e - c - r - e - t

www.suntrust.com/
Web-Input Security Problems

• Host-based malware
  – Rootkits, keyloggers, screen scrapers, …
  – May capture input pre-SSL

• On-screen security indicators cannot be trusted
  – Malware may forge them

• SSL offers network protections only
  – Was never intended for malicious host
Our Solution: Bumpy

• Protect user input from malware
  – Software keylogger, screen scraper
  – Compromised OS, web browser

• Offer assurance that input is protected
  – User feedback via a Trusted Monitor
  – Optional: feedback to web server via attestation

• Degrade gracefully to today’s input system for legacy applications
  – Retain seamless user experience
Bumpy Approach (1/3)

- User decides which fields are sensitive
- Secure Attention Sequence @@ [RJMBM2005]
Bumpy Approach (2/3)

- Trusted Monitor assures user that input protections are in place
- Physically separate device
  - Display, long-term storage, comm., crypto-capable
- Display indicates
  - Application name
  - SSL hostname
  - Favicon
Bumpy Approach (3/3)

• Post-Processor executes on client to process sensitive input for web server

1. PoPr may be standard / widely deployed
   – No changes to server: PwdHash [RJMBM05]

2. Web server provides PoPr
   – Ex: End-to-end encryption
   – Remote attestation proves PoPr used
Bumpy Architecture

• Input devices encrypt all events
• Protected (isolated) input processing
  – Pre-Processor (PreP) to decrypt events
  – Post-Processor (PoPr) packages events for web server
• Logical Flow:
Input Flow for @@

1. User types @@
2. Keystrokes encrypted
3. OS handles ciphertext
4. OS invokes Pre-Processor
5. PreP releases @@ to OS / App and signals TM
Sensitive Keystroke Flow

1. User presses key / button
2. Keystroke encrypted
3. OS handles ciphertext
4. OS invokes Pre-Processor
5. PreP releases decoy event to OS / App
Inside the Pre-Processor

- Decrypt and enqueue input events
- Invoke PoPr upon receiving “Blur”
Input Flow
Per Field

6. PoPr invoked with queue

7. PoPr output handled by web browser

8. Web server receives PoPr output

Encrypting Input Devices

USB Interposer

Legacy Operating System

PreP Q PoPr

Protection!

Browser Extension

Web Server

Internet

Trusted

Untrusted
PreP, PoPr Protection: Flicker

- Isolate security-sensitive code execution from all other code and devices [McPaPeReIs2008]
  - Runs directly on hardware, except for the shim
- Attest to security-sensitive code and its arguments and nothing else
- Convince a remote party that security-sensitive code was protected
- Add < 250 SLoC to the software TCB
Flicker Execution Flow

- Part of AMD Secure Virtual Machine (Intel TXT)
- Measured launch and isolation
- Please see the paper for full details
External Verification

• PreP informs Trusted Monitor of @@@ receipt and PoPr origin
  – Trusted Monitor presents to user the origin of PoPr for subsequent secret input

• Upon form submission, web server may receive attestation to PoPr
  – Covers PreP, PoPr, and protected keystrokes
  – Relevant when web server provides PoPr
Bumpy Implementation

- Commodity workstation with AMD SVM
  - HP dc5750 with Broadcom v1.2 TPM
- USB Interposer
  - 141 +/- 15 ms overhead per keystroke
  - C program (~500 SLoC) for embedded Linux
- Trusted Monitor
  - C++ smart phone application (~2K SLoC)
- Firefox 2 extension
Trusted Monitor

• Indicates when protected input is active
Limitations

• Incompatible with some Phishing defenses
• Non-textual input fields unprotected
  – Drop-down lists, radio buttons, …
  – Ex: Credit card expiration date
• User forgets to employ @@ prefix
• Confusing form fields on malicious page
  – “Enter your password: @@__________”
• Mouse position information is revealed
• Input timing information is revealed
Subtleties

• Active input field in browser
  – Focus: untrusted hints from browser
    • Field label included in PoPr input
  – Blur: infer from input stream
    • Prevents browser from ending protection early

• Device association
  – PreP to input device(s)
  – PreP to Trusted Monitor

• Public computers
Some Related Work

- VMM-based input protection
  - NetTop [MeSi 2000], TIP [BoPr 2007], Garriss et al. [2008]
- Mobile devices as “smart cards”
  - Balfanz et al. [1999], Ross et al. [RHCJCB 2002], Sharp et al. [2008], ZTIC [IBM 2008]
- Secure Window Managers
  - NitPicker [FesHel 2005], EROS [ShVaNoCh 2004], Epstein et al. [1990s]
- Browser Security: PwdHash [RJMBM 2005]
Conclusions

- Sensitive input inaccessible from OS
- Users indicate which input is sensitive
- Web server can define processing for sensitive input intended for that server
- Attestation used to convince web server its PoPr is in use
- Trusted monitor assures user
- Feasible today on commodity hardware
Thank You

• jonmccune@cmu.edu

• Questions?