Surreptitious Communication

CS 161 - Computer Security
Profs. Vern Paxson & David Wagner

TAs: John Bethencourt, Erika Chin, Matthew Finifter, Cynthia Sturton, Joel Weinberger
http://inst.eecs.berkeley.edu/~cs161/

April 26, 2010
Steganography

- Transmitting **hidden** messages using a **known** communication channel
  - Or hiding extra data inside known storage
- **Goal:** Sneak past a reference monitor (**“warden”**)
- **Examples?**
  - **Zillions**: tattooed heads of slaves, least-significant bits of image pixels, extra tags in HTML documents, …
  - All that’s necessary is **agreement** between writer of message & reader of message
- **Security?**
  - **Brittle**: relies on **security-by-obscurity**
    - Warden can extract/block messages if they know the trick
Covert Channels

• Communication between two parties that uses a hidden (secret) channel

• Goal: evade reference monitor inspection entirely
  – Warden doesn’t even realize communication is possible

• Example: suppose (unprivileged) process A wants to send 128 bits of secret data to (unprivileged) process B …
  – But can’t use pipes, sockets, signals, or shared memory; and can only read files, can’t write them
Covert Channels, con’t

• Method #1: **A** syslog’s data, **B** reads via `/var/log/…`
• Method #2: select 128 files in advance. **A** opens for read only those corresponding to 1-bit’s in secret.
  – **B** recovers bit values by inspecting access times on files
• Method #3: divide **A**’s running time up into 128 slots. **A** either runs CPU-bound - or idle - in a slot depending on corresponding bit in the secret. **B** monitors **A**’s CPU usage.
• Method #4: Suppose **A** can run 128 times. Each time it either exits after 2 seconds (0 bit) or after 30 seconds (1 bit).
• Method #5: …
  – There are zillions of Method #5’s!
Covert Channels, con’t

• Defenses?
• As with steganography, #1 challenge is identifying the mechanisms
• Some mechanisms can be very hard to completely remove
  – E.g., duration of program execution
• Fundamental issue is the covert channel’s capacity
  – Bits (or bit-rate) that adversary can obtain using it
• Crucial for defenders to consider their threat model
Side Channels

• **Inferring** information meant to be hidden / private by **exploiting** how system is structured
  – Note: unlike for steganography & covert channels, here we do *not* assume a cooperating sender / receiver

• Can be difficult to recognize because often system builders “abstract away” seemingly irrelevant elements of system structure

• Side channels can arise from physical structure …
Side Channels

• Inferring information meant to be hidden / private by exploiting how system is structured
  – Note: unlike for steganography & covert channels, here we do not assume a cooperating sender / receiver

• Can be difficult to recognize because often system builders “abstract away” seemingly irrelevant elements of system structure

• Side channel can arise from physical structure …
  – … or higher-layer abstractions
/* Returns true if the password from the user, 'p', matches the correct master password. */
bool check_password(char *p)
{
    static char *master_pw = "T0p$eCRET";
    int i;
    for(i=0; p[i] && master_pw[i]; ++i)
        if(p[i] != master_pw[i])
            return FALSE;

    /* Ensure both strings are same len. */
    return p[i] == master_pw[i];
}
Inferring Password via Side Channel

- Suppose the attacker’s code can call `check_password` many times (but not millions)
  - But attacker can’t breakpoint or inspect the code

- How could the attacker infer the master password using side channel information?

- Consider layout of `p` in memory:

  ```
  ... if(check_password(p))
    BINGO();
  ...wildGUEss
  ```
Spread $p$ across different memory pages:

```
spread across memory pages: wildGUEs$s
```

Arrange for this page to be paged out

If master password doesn’t start with ‘w’, then loop exits on first iteration ($i=0$):

```
for(i=0; p[i] && master_pw[i]; ++i)
  if(p[i] != master_pw[i])
    return FALSE;
```

If it *does* start with ‘w’, then loop proceeds to next iteration, generating a page fault that the caller can observe
T0p$eCRET?

Ajunk....

Bjunk....

Page fault!

Tjunk....

TAunk....

TBunk....

Page fault!

Fix?

T0unk....

T0Ank....

No page fault

No page fault

No page fault
bool check_password2(char *p)
{
    static char *master_pw = "T0p$eCRET";
    int i;
    bool is_correct = TRUE;

    for(i=0; p[i] && master_pw[i]; ++i)
        if(p[i] != master_pw[i])
            is_correct = FALSE;

    if(p[i] != master_pw[i])
        is_correct = FALSE;

    return is_correct;
}

Note: still leaks length of master password
Side Channels in Web Surfing

• Suppose Alice is surfing the web and all of her traffic is encrypted
• Eve can observe the presence of Alice’s packets but can’t read their contents or destination
• How can Eve deduce that Alice is visiting FoxNews (say)?
FoxNews.com Meets iPhone

The No. 1 name in news — in the palm of your hand. Download the Fox News iPhone app for latest news, streaming video, and radio — and it’s all free.

- The Five Best Apple iPad Apps
- SLIDESHOW: Killer Apps for the Apple iPad

Wall Street Battle Brews
Democrats are resisting Republican appeals for a broad compromise on a financial overhaul bill as vote nears.
- SEC to Investigate Timing of Goldman Suit | VIDEO
- FoxBusiness: Goldman Stands Behind ‘Fabulous Fab’
- Buffett Presses for Eased Curbs on Derivatives

GOP Urges Border Insecurity Solution
Republican lawmakers and Arizona residents call on Capitol Hill to put immigration aside and secure the border.
- Arizona Lawmaker: U.S. Should Fight Immigration Law
- Mexico: Cartels Turning Attacks on Authorities

Wrong Guy, Place to Tell a Jewish Joke?
URGENT: National Security Adviser James Jones issues apology for joke told in front of pro-Israel group.
- YOU DECIDE: Was Jones’ Joke Inappropriate?
Eve “fingerprint” web sites based on the specific sizes of the items used to build them.
Side Channels in Web Surfing

- Suppose Alice is surfing the web and all of her traffic is encrypted.
- Eve can observe the presence of Alice’s packets but can’t read their contents or destination.
- How can Eve deduce that Alice is visiting FoxNews (say)?
- What about inferring what terms Alice is searching on?
Exploiting Side Channels For Stealth Scanning

• Can attacker using system \textbf{A} scan the server of victim \textbf{V} to see what services \textbf{V} runs …
• … without \textbf{V} being able to learn \textbf{A}’s IP address?

• Seems impossible: how can \textbf{A} receive the results of probes \textbf{A} sends to \textbf{V}, unless probes include \textbf{A}’s IP address for \textbf{V}’s replies?
IP Header Side Channel

ID field is supposed to be unique per IP packet.

One easy way to do this: increment it each time system sends a new packet.

<table>
<thead>
<tr>
<th>4-bit Version</th>
<th>4-bit Header Length</th>
<th>8-bit Type of Service (TOS)</th>
<th>16-bit Total Length (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>16-bit Identification</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3-bit Flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>13-bit Fragment Offset</td>
</tr>
<tr>
<td>8-bit Time to Live (TTL)</td>
<td>8-bit Protocol</td>
<td>16-bit Header Checksum</td>
<td></td>
</tr>
</tbody>
</table>

32-bit Source IP Address

32-bit Destination IP Address

Payload
A (Tacker) — P (atsy) — V (ictim)

- **Attacker**
  - Echo request, reply, ID=3
  - Echo request, reply, ID=4
  - Echo request, reply, ID=5
  - TCP SYN, src=P, dst port=24
  - TCP SYN-ACK
  - Echo request, reply, ID=6
  - Echo request, reply, ID=7
  - TCP SYN, src=P, dst port=25
  - TCP RST, ID=8

- **Patsy**

- **Victim**
  - no listener on port 24, RST generated
  - listener exists on port 25, SYN-ACK generated

*P has no state for this connection, so generates a RST, which increments the IP ID sequence.*
UI Side Channel Snooping

- Scenario: Ann the Attacker works in a building across the street from Victor the Victim. Late one night Ann can see Victor hard at work in his office, but can’t see his CRT display, just the glow of it on his face.

- How might Ann snoop on what Victor’s display is showing?
CRT display is made up of an array of phosphor pixels.

640x480 (say)
Electron gun sweeps across row of pixels, illuminating each that should be lit one after the other.
When done with row, proceeds to next. When done with screen, starts over.
Thus, if image isn’t changing, each pixel is **periodically** illuminated at its own unique time.
Illumination is actually short-lived (100s of nsec).
Photomultiplier + high-precision timing + deconvolution to remove noise
CAN YOU READ THIS?

This image was captured with the help of a light sensor from the high-frequency fluctuations in the light emitted by a cathode-ray tube computer monitor which I picked up as a diffuse reflection from a nearby wall.

Markus Kuhn, University of Cambridge, Computer Laboratory, 2001