USER-BASED PERMISSIONS
ADVERSARIAL APPS

MALWARE: Actively harms the user

GRAYWARE: Pushes the boundaries of legality

LEGITIMATE: Benign, but it has unwanted features
APP-BASED PERMISSIONS

Application Permissions
The application Mr. Number Callblock is attempting to access phone information.

Do not ask again

Allow  Deny

Vendor Modules >>
Permissions are supposed to help users control their applications.
Do you want to allow the following program from an unknown publisher to make changes to this computer?

Program name: updater.exe
Publisher: Unknown
File origin: Hard drive on this computer

Show details

Change when these notifications appear
95% of study participants consented to fake UAC dialogs

Motiee 2010
PANDORA

Accept & download

PERMISSIONS

Phone calls
Read phone state and identity

System tools
Bluetooth administration, change Wi-Fi state, change network connectivity, prevent phone from sleeping
42% of study participants had never noticed the permissions

Felt 2012

Bluetooth administration, change Wi-Fi state, change network connectivity, prevent phone from sleeping
How do we ask users to make informed decisions about app permissions?
GUIDING PRINCIPLES
DUAL PATH MODEL

- **optimizing**
  - reads the question
  - retrieves memories
  - makes a decision

- **satisficing**
  - does not understand
  - retrieves cues
  - relies on heuristics

DUAL PATH MODEL
Conserve user attention
Don’t interrupt the user’s primary task for security
PERMISSION-GRANTING MECHANISMS
**PRO**
Does not consume user attention

**CON**
Only appropriate for revertible or low-severity permissions

AUTOMATIC GRANT
TRUSTED UI

**PRO**
Non-interruptive

**CON**
Only appropriate for user-initiated actions
**PRO**
Applicable to almost all permissions

**CON**
Interruptive, habit-forming, not useful for advance approval

RUNTIME CONSENT DIALOGS
**PRO**
Applicable to all permissions, even advance approval

**CON**
Interruptive, looks like EULAs, habit-forming
Instead of choosing one mechanism for a system, choose the best for each permission
Can it be undone?

Is it an annoyance?

Does the user initiate it?

Can the user approve it when it’s needed?

Automatic

Trusted UI

Runtime

Install-time
Can it be undone?

Is it an annoyance?

Does the user initiate it?

Can the user approve it when it’s needed?

Yes → Trusted UI

RECORDING WITH THE MIC
<table>
<thead>
<tr>
<th>Permission Assignment</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatically granted</td>
<td>60%</td>
</tr>
<tr>
<td>Trusted UI</td>
<td>21%</td>
</tr>
<tr>
<td>Runtime consent dialogs</td>
<td>11%</td>
</tr>
<tr>
<td>Install-time warnings</td>
<td>7%</td>
</tr>
</tbody>
</table>

76 platform-neutral permissions
Which is better, run-time or install-time?
Which is better, run-time or install-time?

Wrong question!
We propose guidelines for how to ask users for permission with minimal interruptions.
QUESTIONS?
adriennefelt@gmail.com

www.adrienneporterfelt.com