1 **Beágyazott rendszerek biztonsága**

2 **Mi is a beágyazott rendszer?**

- A beágyazott rendszer olyan számítógép, ami cél feladatot lát el.
- Lehetséges alkotó elemek:
  - Mikrovezérlő – Nincs operációs rendszer, nehéz támadni a felépítésből adódóan
  - SOC – Operációs rendszert futtat & Neumann architektúra
  -

3 **USB**

- USB eszközöket minden nap alkalmazunk
- Könnyű használni, mivel Plug&Play
- Számos mikrovezérlő rendelkezik natív USB támogatással
- Szoftver oldalról:
  - 127 eszközt lehet egyszerre kezelni
  - Az eszközt a VID és PID azonosítók azonosítják be
  - Számos periféria osztály
  - Egy USB eszköz több perifériát is megvalósíthat

4 **USB**

- USB Architektúrában nem egyenlő rangúak a felek kommunikáció során:
  - HOST: Inteligens, kommunikációt vezérli
  - SLAVE: Buta eszköz, „csak" kommunikációra képes
- A támadások a HOST gép ellen irányulnak.

5 **USB**

- Az előadásban bemutatásra kerülő megoldások többsége az USB buszrendszerhez köthető
- Ez nem feltétlen jelenti azt, hogy az USB rossz
- Az USB jó, csak kellő bizalmatlansággal kell hozzáállni

6 **USB és Arduino Leonardo**

- A Leonardo modell volt az első Arduino natív USB támogatással
- A gyári bootloader a következő eszközöket biztosítja:
  - billentyűzet
  - egér
  - soros port (program teszteléshez)
- Külső könyvtárak segítségével akár pendrive is készíthető vele

7 **Furcsán viselkedő egér**

- A legelterjedtebb USB periféria: egér
- Mivel a Leonardo gyárilag rendelkezik egér emulációs képességekkel, ezért könnyű megírni az alábbi programot:
  - Véletlenszerűen várjon x percet
  - Ezután véletlenszerűen y másodpercig mozgassa az egeret véletlenszerűen generált pontokra

8 **Furcsán viselkedő egér**

- Konkrétan semmi rosszat nem csinál, csak bosszant

- Érdekes eredményeket lehet produkálni vele, ha a támadott célszemély masszívan online játékos…
- Kb. 3000Ft-os Bluetooth adapterrel kiegészítve még érdekesebb eredmények érhetőek el…

9 **Második billentyűzet**

- Minden operációs rendszer rendelkezik gyorsbillentyűkkel adott feladat elvégzésére
- Ezen gyorsbillentyűk lenyomása könnyen emulálható
- Probléma: akár teljes műveletsor is automatizálható
  - Futtatás ablak megnyitása
  - Parancssor indítása
  - Minden adat és meghajtó törlése

10 **Második billentyűzet**

- Operációs rendszertől függetlenül működik
- Vírus telepítésére is felhasználható:
  - Olyan parancsot futtatunk, amely az internetről letölti és telepíti a kártékony kódot
- mikrovezérlő sebessége > emberi reakcióidő
  - Ebből adódóan szinte észrevétlen gyorsasággal történhetnek a dolgok
-

11 **Hardveres Keylogger**

- A Leonardo nyílt forráskódú terve alapján készíthető egy olyan eszköz, amely:
  - Rendelkezik PS/2 vagy USB bemenettel és SD kártya foglalattal
  - A mikrovezérlő programja nem csinál mást, csak a bemenetre kötött billentyűzet eseményeit továbbítja a gép felé, valamint ezeket az SD kártya felé is továbbítja.
  - Szintén operációs rendszertől független megoldás
  - Kellően kicsire megcsinálható, így szinte észrevehetetlen

12 **Hardveres Keylogger**

- A vezeték nélküli billentyűzetek még könnyebb célpontot jelentenek
- Sok gyártó nem alkalmaz az átvitel során titkosítást, illetve ha alkalmaz is, az igen primitív
- Ebből adódóan építhető olyan eszköz, amely a vezeték nélküli billentyűzeteket naplózza
- Létezik ebből is nyílt forrású megoldás: KeyKeriki

13 **KeyKeriki**

- 27Mhz-es vezeték nélküli billentyűzetek lehallgatására
- Létezik 2,4Ghz-es változata is modernebb billentyűzetekhez
- Teljesen nyílt forráskódú
- Egy nagyobb zsebben simán elfér

14 **Védekezés a keyloggerek ellen**

- A hardveres megoldások nem igen detektálhatóak szoftverből
- Biztonságilag kritikus helyen ne alkalmazzunk vezeték nélküli megoldásokat
- Ellenőrizzük a számítógép és a billentyűzet közvetlen kapcsolatát

15 **BAD USB**

- A hordozható USB meghajtók a melegágyai voltak már korábban is a vírusoknak,

köszönhetően az autorun lehetőségnek
- Újabb Windows rendszerek esetén az autorun nem igazi autorun, mivel felhasználói beavatkozást igényel.
- Operációs rendszer függő támadás

16 **BAD USB**
- A legtöbb USB pendrive két részből áll: NAND flash + USB vezérlő
- A vezérlő illeszti a flash memóriát az USB rendszerhez
- Továbbá tartalmazza a rossz szektorok táblázatát
- Sok esetben a vezérlő egy általános processzor, ami 8085 vagy ARM alapú
- A flash memória tartalmazza a működtető firmware-t is.

17 **BAD USB**
- Ez gyártói szempontból azért jó, ha a flash memórián van a firmware, mert egyszerűbb a gyártás: kész lap bedug egy cél gépbe, ami USB-n felprogramozza.
- Egyszerű gyártás, de hatalmas biztonsági kockázat, mivel ha USB-n megy rá a program, akkor USB-n le is lehet szedni…
- Ha le lehet szedni, akkor vissza lehet fejteni… Ha vissza lehet fejteni, akkor át lehet írni…

18 **BAD USB**
- Menet közben tetszőlegesen válthat az USB eszköz eszközosztályt
- Pl: HP nyomtatók először CD meghajtóként látszanak, amíg nincs fent az illesztő, utána nyomtatóként használhatóak.
- Tehát ha kellően elszántak vagyunk, akkor egy tetszőleges pendrive átprogramozható
- Pl: használat közben váltson át billentyűzetre és töltsön le és telepítsen egy vírust, vagy egyéb programot…

19 **BAD USB**
- De nem csak erre használható a dolog…
- A pendrive akár megvalósíthat egy hálózati eszközt is, amin egy DHCP és DNS szolgáltatás fut
- Innentől kezdve pedig a támadás hasonló a Pi esetén említett DHCP megoldáshoz…

20 **BAD USB**
- Akár boot „vírus” is írható, mivel a BIOS/UEFI rendszerek más módon olvassák a meghajtót, mint az operációs rendszerek.
- Ebből adódóan detektálható, hogy újratelepítés lesz…
- Mivel a Windows telepítők nagyjából ugyanazt a fájl sémát követik
- Igen csúnya dolgok készíthetők

21 **BAD USB**
- Pl. Telepítéskor eleve fel lehet tenni egy rootkit, vagy keylogger programot egy módosított install.wim fájl kiszolgálásával, ami normál módon nem látható.
- Linux és OS-X rendszerekre is adaptálható a megoldás
- A lehetőségek ténylegesen végtelenek…

22 **BAD USB**
- További probléma, hogy a szükséges eszközök, amik kellenek a pendrive-ok módosításához, nyíltan elérhetőek az interneten.
- Elvileg bármilyen flash meghajtó jó, de macera visszafejteni és leszedni a gyári kódot

- A kiadott kódok pár meghajtó típust támogatnak jelenleg

23 **BAD USB**

- Pillanatnyilag védekezni az ilyen támadások ellen nem lehet, mivel ehhez az USB újragondolása kellene.
- Gyártói oldalról úgy lehetne védekezni, hogy a firmware programozás után csak olvashatóvá váljon -> hardver módosítás kell
- Szoftver oldalról kérdéses a dolog, mivel a VID és PID azonosítók szabadon változtathatóak, nem elég egyediek -> nem lehet blokkolni bizonyos eszközöket

24 **Raspberry Pi alapú telefon másoló**

- Minden okostelefon USB töltéssel rendelkezik*
- Ez szép és jó, de igen nagy biztonsági kockázat
- Több napos fesztiválok szervezésekor a szervezők lehetőséget biztosítanak a telefonok töltésére töltőállomásokon
- Gond: nem látni, hogy hova megy a kábel…

25 **Raspberry Pi alapú telefon másoló**

- Mivel minden telefon USB alapú, így a fájlokhoz hozzá lehet férni USB-n: PTP, MTP, vagy Mass Storage üzemmódban.
- A RaspberryPi igen kicsi, ebből adódóan könnyen készíthető egy kisméretű eszköz, amely:
  - A telefon csatlakoztatása után vár ~2 percet
  - Felcsatolja a telefont meghajtóként, majd egy USB merevlemezre átmásolja a tartalmat
  - A feladat végeztével lecsatolja a készüléket

26 **Raspberry Pi alapú telefon másoló**

- Pi helyett jobb választás lehet a BananaPi, mivel ezen van SATA, ami gyorsabb hozzáférést tesz lehetővé
- Mivel sok telefon Android alapú, így bővíthető a megoldás ADB támogatással is.
- Ha a telefonon be van kapcsolva az USB hibakeresés, akkor hozzá lehet férni mindenhez: telefonkönyv, üzenetek, böngészési előzmények, stb…

27 **Védekezés a telefon másolás ellen**

- Speciális USB adapterrel
- Megszakítja az USB adatkapcsolatot a telefon és a gép között
- $10 áron beszerezhető
- Akár otthon is elkészíthető

28 **RaspberryPi, mint DHCP szerver**

- A DHCP lehetővé teszi a végpontok központi és gyors konfigurálását.
- Ideális esetben egy hálózaton egy DHCP szerver van.
- Nem ideális eset: 2db DHCP szerver
- Ebből egy „gonosz” lesz ☺

29 **RaspberryPi, mint „gonosz” DHCP szerver**

- Kis mérete miatt könnyen elrejthető
- A hálózati beállításokkal megegyező DHCP adatokat oszt ki, „csak” a DNS szervert piszkálja meg.

- A DNS szerver is a Pi, amiben a támadni kívánt lapok kezdőoldalait átirányítjuk a Pi-re.
- A kezdőoldalakat módosítjuk úgy, hogy a felhasználók adatait naplózza
- A naplók hálózaton is megoszthatók a támadónak

30 **RaspberryPi, mint „gonosz" DHCP szerver**

- Nem minden gép fog csatlakozni hozzá, mivel versenyhelyzet van a hálózaton a 2 DHCP között.
- Ennek ellenére komoly adatbázis építhető pár óra alatt.
- Főleg, ha a hálózat iskolában/egyetemen van…
- Megoldás: switch-ek zárt rack szekrényben, lehetőleg nem eldugott helyeken
- Illetve a switch-ek esetén port security konfigurálása, ha lehetőség van rá

31 **Telefonok biztonsága**

32 **Telefonok biztonsága**

- Minden telefon futtat operációs rendszert.
- Az OS többségében Android
  - Linux kernel + Google Dalvik vm + etc goggle stuff.
  - Nyílt forráskód => Gyártói barmolások.

33 **Android**

- A barmolások miatt kiszámíthatatlan és pontosan nem tudott, hogy mennyi kritikus sebezhetőség van és mennyi készüléket érint.
- Sok gyártó nem frissíti rendszeresen a szoftvert már fél év után.
- Biztonságilag nem a legjobb választás
- Rootolás tovább növeli a kockázatot

34 **IOS**

- Biztonság? Az meg mi ? LOL
- Kritikus hibák esetén nem a legjobb a reagálása az apple-nek, de fejlődést mutatnak.
- Eszköz támogatás szűkös. Kb 2 év / eszköz, ami igen jó lenne, ha tudnának szoftvert írni jól és nem csak árakat szabni.
- Jailbreak növeli a biztonsági kockázatokat.
-

35 **Windows Phone**

- Alapvetően buta a rendszer, mint a föld, de legalább működik.
- Alacsony elterjedés miatt nem igen ismertek a sebezhetőségek
- Nincs ROOT lehetőség, csak Developer mode.
- Rendszeres frissítések a 8-as szériához. 7-es széria már nem támogatott.

36 **A Legbiztonságosabb telefonok ☺**