# Žad Deljkić: Android USB vektor napada

## Dnevnik rada

### 1. tjedan (28.3.2016. - 3.4.2016.)

#### Dosadašnji rad

Do sada sam razvio prototip aplikacije za Android uređaje sa instaliranim Kali NetHunterom koja može preko USB-a detektirati OS te pokrenuti USB HID napad, te sam napisao nacrt rada na engleskom.

Napisao sam plan rada koji je okvirno dogovoren na početnom sastanku u petak 24.3. Trenutno istražujem što je moguće sa USB napadima, načine detektiranja OS-a preko USB-a te moguće zaštite protiv USB napada.

Dok sam istraživao zaštite protiv USB napada općenito, došao sam opet i do Qubes-a, OS-a koji je općenito fokusiran na sigurnost te ju postiže gotovo ekstremnom izolacijom, te sam ga sada i instalirao i trenutno ga isprobavam. Autori su svjesni raznih napada, između ostaloga i USB napada. No ako korisnik koristi USB tipkovnicu i/ili miš, ne postoji nikakva ugrađena zaštita od USB napada. I dalje je vjerojatno moguće nekako ručno zabraniti novim USB uređajima da se spoje (što opet nije idealno rješenje). USB HID napad (isprobao sam simuliranje tipkovnice) funkcionira na Qubes-u, moguće je direktno napasti dom0, najsigurniju domenu, no detektiranje OS-a ne funkcionira, računalo se ne spaja automatski na mrežu sa mobitelom.

#### Daljnji koraci

Jednom kada je istraživanje gotovo, osmisliti daljni smjer rada - što je točno problem kojim će se rad baviti, te kako ćemo ga rješiti.

Plan je napraviti nešto novo. Jedna opcija je nastaviti s već razvijenom aplikacijom, završiti ju te joj potencijalno proširiti funkcionalnosti. Druga opcija je osmisliti i implementirati novi USB napad koji do sada nije bio istražen, ukoliko pronađemo nešto takvo.

### 2. tjedan (4.4.2016. - 10.4.2016.)

#### Dosadašnji rad

Prikupio sam literaturu koja bi mogla biti vezana uz ovaj rad te ju sortirao u kategorije:

- USB napadi sa Androida na PC
- Napadi na Android (kojima bi se mobitel mogao “zaraziti” tako da kasnije može napasti računala s kojima se spoji)
- Obrane od USB napada
- Detekcija OS-a preko USB-a

Literatura se nalazi u Zotero grupi.

#### Daljnji koraci

Pročitati i odrediti koja je literatura najrelevantnija, te ovisno o postojećim radovima odrediti daljnji smjer ovog rada kako bi se napravilo nešto novo.

Za sada okvirnim pregledom svih radova nisam naišao na ništa slično aplikaciji koju sam razvio, tako da mi se čini da bi nastavak u tom smjeru bio dobar.

### 3. tjedan (11.4.2016. - 17.4.2016.)

#### Dosadašnji rad

Pročitao sam literaturu, organizirao ju te napisao bilješke o bitnim dijelovima (sve je u Zotero grupi).

Postoji velika količina relevantnih radova te je dosta toga već istraženo, uključujući i USB napade sa Androida te fingerprintanje OS-a preko USB-a, čak i sa Androida (u svrhe forenzike).

No ne postoji nikakvo rješenje sa već objavljenim source kodom za fingerprintanje OS-a, niti neko rješenje koje integrira više koraka napada, primjerice detekcija OS-a te napad ovisno o rezultatu detekcije (no spominje se).

#### Daljnji koraci

Ja mislim da najviše ima smisla fokusirati daljni smjer rada na sljedeće točke te izrada aplikacije/jednostavnog frameworka za Android USB napade u tom duhu:

- USB napadi imaju neke velike prednosti - zaobilaze sve sadašnje forme zaštite, pod pretpostavkom da uspijemo uštekati ili nekoga navesti da ušteka naš zlonamjerni uređaj
- Android smartphone je odlična platforma za USB napade jer:
  - Relativno lagano ga je reprogramirati da radi što želimo
  - Ima *izrazito* puno mogućnosti, potpuno neusporedivo sa drugim uređajima za USB napade:
    - Moguće je koristiti ogroman broj alata razvijenih za linux, npr. sa Kali NetHunterom imamo pristup svim alatima pakiranim u Kali Linux-u (nmap, čak metasploit, itd.)
      - Jedna velika prednost korištenja takvih postojećih popularnih alata je što nije potrebno izrađivati/održavati vlastite alate, npr. nmap OS scan će samo postajati bolji s vremenom, detektirati će i nove OS-ove kako se pojavljuju itd., dok drugi pristupi za detektiranje OS-a s Androida zahtjevaju određenu količinu posla i inicijalno i za održavanje
    - Uz USB, obično imamo 3G, wifi, bluetooth, mikrofon…
      - Mogu se koristiti kao kanali za eksfiltraciju podataka neovisni o situaciji
      - Općenito, vlastiti pristup internetu otvara velike mogućnosti (npr. command&control server)

        - Snažne performanse
    - Široko su dostupni, često ih nije potrebno kupiti već ih gotovo svatko posjeduje (gotovo svi imaju smartphone, Android ima ~80% market share)
        - Ali opet u pitanje dolazi koliko tih mobitela je moguće rootati, te za koliko njih su podržani neki alati relevantni ovom radu (primjerice kernel patch za HID napade ili cijeli Kali NetHunter)
    - Uvijek ih nosimo sa sobom što je praktično te smo uvijek “spremni”, za razliku od raznih USB rubber duckya i sličnih rješenja, jer smartphone-i imaju i izrazito korisnu svrhu osim USB napada
        - Uz USB napade, moguće je sa Androida lansirati razne druge napade pa je takav mobitel praktičan kao općenito platforma za napade
    - Nije “sumnjiv”, tj. često je korišten kao primjerice i USB memory stick-ovi te postoje legitimni razlozi zašto ga uštekati u PC
        - Uštekava se u PC između ostaloga za prijenos podataka i punjenje baterije, tako da nije sumnjivo vidjeti ga uštekanog
        - “Možeš li staviti moj mobitel na punjenje na tvoj PC?”

U raznim radovima su već predložene neke zanimljive ideje, no nigdje to nije skupljeno i implementirano.

Funkcionalnosti koje su potrebne za efektivan napad bi bile:

- Način detektiranja kada korisnik nije za računalom, kako bi se tada izvršio napad koji bi korisnik inače primijetio
- Detektiranje OS-a (ili čak i detaljnije fingerprintanje PC-a) kako bi se napad mogao prilagoditi
- Razni načini napadanja PC-a

Zato sam mislio kako bi bilo dobro pretvoriti aplikaciju koju radim u nekakav framework za USB napade, koji bi podijelio otprilike ovako:

- **Trigger** - okidači za započinjanje napada (prvenstveno kako ih korisnik ne bi primijetio), npr.:
    - Čekanje fiksne količine vremena - npr. pričekaj 5min
        - I ako se nešto dogodi na računalo što korisnik primijeti, ako je prošlo neko vrijeme od uštekavanja uređaja manje će sumnjati da je on to uzrokovao
    - Čekanje određenog vremena - npr. napadni u 22:00
        - Ako se primjerice uređaj ušteka u računalo u nekoj firmi gdje ga nitko ne dira (npr. svi misle da je nečiji tuđi), moguće mu je reći da pričeka 22:00 kada nikoga više neće biti tamo te da pokrene napad - tada nitko neće moći vidjeti kako se napad odvija niti vjerojatno reagirati do jutra
    - Čekaj naredbu za napad - npr. napadni kada ti preko interneta C&C server to kaže
        - “Možeš li uštekati ovaj mobitel na punjenje u neko računalo?” te kada napadač primijeti da nitko nije oko računala, preko interneta (ili na neki drugi način) pošalje naredbu da mobitel izvrši napad
    - Čekaj dok korisnik ne ode od računala - potencijalni načini implementacije:
        - mobitel MITM-a internet promet od računala - po prometu može saznati otprilike je li korisnik aktivan (barem na internetu) - čak nije potrebno sav promet MITM-at, može se samo mobitel zadati kao DNS server preko DHCP-a no ne i default gateway, te pratiti DNS promet
        - bilokoji USB uređaj može sniffati promet poslan sa PC-a drugim uređajima na istom hub-u - npr. ako korisnik prebacuje nešto na USB drive, uređaj uštekan u isti hub može vidjeti to i pretpostavit da je korisnik aktivan
- **Scan** - skeniranje računala

    - Raspoznavanje OS-a po razlikama na razini USB protokola (postoje radovi o tome)
    - Povezivanje mobitela i računala na istu mrežu + mrežno skeniranje (npr. nmap)
        - Spomenuto je u jednom radu no ne i implementirano (u mojoj aplikaciji već je)
- **Attack** - razne implementacije USB napada, prilagođene ovisno o rezultatu skeniranja
    - HID napad (simuliranje tipkovnice, miša, joysticka)
    - MITM - simuliranje USB ethernet adaptera, preusmjeravanje svog internet prometa preko uređaja ili samo DNS-a
    - Razno exploitanje:
        - Exploitanje USB driver-a
        - Exploitanje koda koji parse-a filesystem na USB drive-u
        - Exploitanje koda koji prikazuje thumbnailove/ikone datoteka na USB drive-u
        - Zločudni office dokumenti, pdf-ovi, bilošta što će korisnik potencijalno otvoriti...

Naravno ne bih implementirao sve ovo, no velik dio toga je moguće lagano implementirati.

Sama aplikacija bi bila izrazito jednostavna - u njoj bi bilo moguće podesiti koje shell skripte treba pokrenuti pri sljedećem USB spajanju na PC, te bi se glavna funkcionalnost implementirala u njima (tako više manje za sad i funkcionira).