Napad TEMPEST (»Van Eck phreaking«)

Peter Aleksander Bizjak, Pavel Remic-Weiss

Uvod

TEMPEST (*Telecommunications Electronics Materials Protected from Emanating Spurious Transmissions*) je specifikacija ameriške nacionalne varnostne agencije in certifikat zveze NATO, ki se nanaša na vohunjenje za informacijskimi sistemi zaradi uhajanja emanacij, vključno z nenamernimi radijskimi ali električnimi signali, zvoki in vibracijami. TEMPEST zajema obe metodi vohunjenja drugih in zaščito opreme pred takšnim vohunjenjem. Prizadevanja za zaščito so znana tudi kot varnost emisij (EMSEC), kar je podmnožica komunikacijske varnosti (COMSEC).

Zgodovina

Zgodovina napada TEMPEST se je začela leta 1943, ko je inženir korporacije Bell Telephone odkril, da naprave pri prenosu podatkov puščajo sled prenesenega signala.

Drugemu raziskovalcu v laboratoriju se je osciloskop odzval vsakič, ko je bilo poslano šifrirano pismo. Po natančnejšem pregledu je ugotovil, da je te špice mogoče prevesti v navadno sporočilo, ki ga stroj obdela. Čeprav inženir tega takrat verjetno ni vedel, je s tem ugotovil, da vsi stroji za obdelavo informacij za seboj puščajo elektromagnetno valovanje.

Napad TEMPEST je nepričakovano preprost. Vsak stroj, ki obdeluje informacije – naj bo fotokopirni stroj, električni pisalni stroj ali prenosni računalnik –, ima v notranjosti dele, ki oddajajo elektromagnetno energijo. Natančen napadalec lahko zajame pravo frekvenco, analizira podatke glede vzorcev in povrne surove podatke, ki so jih obdelovale naprave, ali celo zasebne šifrirne ključe znotraj naprave.

Uvod

Fizični napadi stranskih kanalov izkoriščajo nenamerno uhajanje informacij z nizko stopnjo fizičnega vedenja računalniških naprav, kot so elektromagnetno sevanje, poraba energije, električni potencial, zvočna emanacija in toplotna nihanja. Ti so bili uporabljeni za razbijanje številnih kriptografskih izvedb.

Majhne naprave, kot so pametne kartice, oznake RFID, FPGA, mikrokontrolerji in enostavne vgrajene naprave, so bile v preteklosti deležne veliko raziskovalne pozornosti s številnimi objavljenimi napadi stranskih kanalov. Vendar pa je pri bolj zapletenih napravah (prenosni računalniki, namizni računalniki, strežniki itd.) teh raziskav manj. Napadi na osebne računalnike

sprožijo nove in hude izzive v primerjavi z napadi na majhne naprave: zapletenost strojne in programske opreme povzroča nepredvidljivo vedenje in hrup; visoke taktne hitrosti nekaj gigahercev na primer. Efektivna merilna pasovna širina je veliko nižja od urne frekvence ciljnega procesorja, zato je težko razlikovati posamezna navodila in zahtevati nove, glede algoritma kriptovalitične tehnike.

Raziskave z univerze v Tel Avivu so pokazale napad TEMPEST: raziskovalci so pridobili zasebni ključ od prenosnika, ki poganja GnuPG. Med dešifriranjem so merili puščanje EMV. Uporabljena oprema je vključevala anteno, ojačevalnike, programsko določen radio in prenosni računalnik.

Implementacija

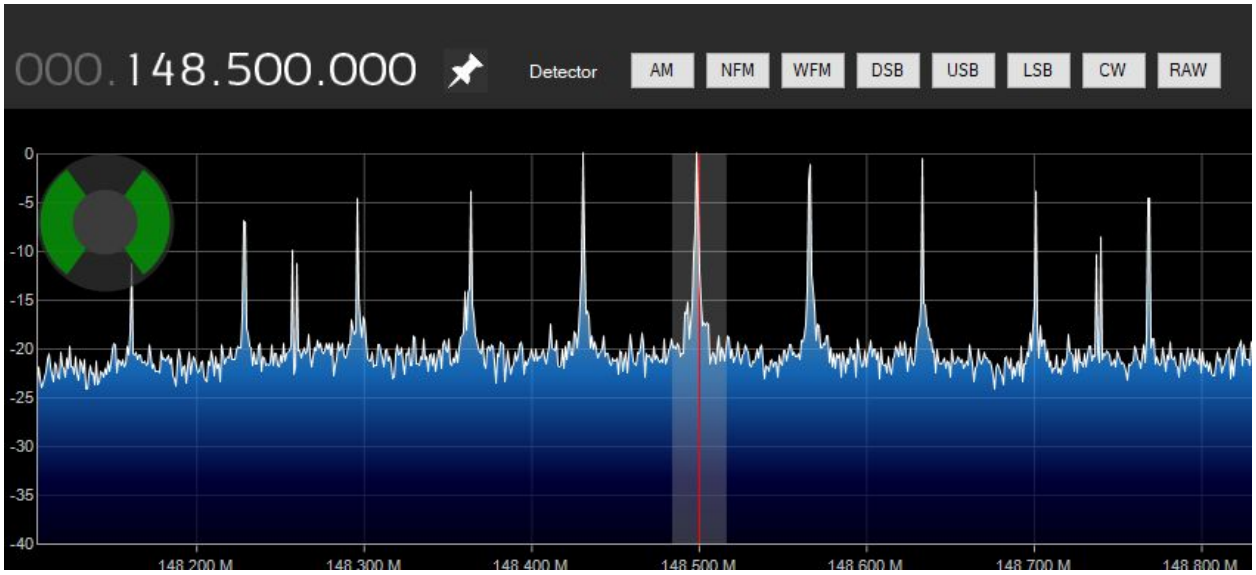
Teorija

»Vektorjev« za napad TEMPEST je, kot smo že omenili, več. Hotela sva najti najbolj preprost način za prenos podatkov iz izoliranega računalnika na drugega. Uporabila bi lahko:

- moduliranje obremenjenosti procesorja,
- moduliranje obremenjenosti grafične kartice,
- …

Vsi ti primeri bi posledično modulirali EMV na napajalniku.

A dejansko se je za najbolj ranljivega izkazal neobičajen člen: **monitor**.
Monitor, ko riše piksle, lahko dela to na tri načine: vrstico za vrstico (**progressive**), mrežno (interlaced) ali pa piksel za pikslom (CRT oz. katodne cevi). Večina modernih monitorjev riše vrstico za vrstico, 60-krat na sekundo. Vsak monitor pa hkrati vsebuje neko uro (clock), t. i. pixel clock, ki z neko frekvenco sinhronizira signal na monitor z grafične kartice. Torej je za naš fiksni primer **1920x1080 @ 60Hz** pixel clock torej 1920x1080x60, kajne? Ne, saj moramo upoštevati še »blanking«, čas, ko se na zaslonu ne kaže nič (koncept overscana). Pri našem zaslonu se izkaže, da je to okoli 20 % oz. **0,193**. Tako da pridemo na končno pixel clock frekvenco 148,5 MHz, kar se tudi lepo vidi na SDR# z RTL-SDR kot neko interferenco:



Amplituda signala je večja, kolikor več informacij (nečrnih pikslov) pošlje grafična kartica. Frekvenca pixel clocka je torej naš nosilec.

Sprejem

Za sprejem uporabljava RTL-SDR palčko:

RTL-SDR antena je tipa »dipole« in za našo rabo jo je treba namestiti na okoli 1 m dolžine, kar je tudi maksimalna dolžina te antene. Po namestitvi pravih gonilnikov sva lahko uporabila program SDR# za zajem signala.

BASK Modulacija

Za implementacijo in preizkus napada sva se za začetek odločila predvajati vsaj osnovni ton. Glede na to, da lahko zadevo amplitudno moduliramo, lahko z amplitudno modulacijo zgeneriramo ton. Ampak kako? Odločila sva se risati črne in bele črte, kar je tudi praksa za testiranje in identifikacijo pixel clocka pri napadih TEMPEST.

Za generiranje tona x Hz moramo vedeti o monitorju samem nekaj stvari:

- vertikalno ločljivost (**h** (npr. 1080)),
- frekvenco osveževanja (**fr** (npr. 60)).

$k_{č} = \frac{x}{f_r}$ - koliko črt na frame potrebujemo

$h_{č} = \frac{h}{k_{č}}$ - višina posamezne črte

Iz teh podatkov nato lahko narišemo eno belo in eno črno črto $k_{č} \cdot 2$ krat.

Ko sva spisala program, ki to dela, sva SDR# nastavila na frekvenco pixel clocka (**148,5 MHz**, ki je nosilec našega AM signala) in sva lahko poslušala ton x Hz.

https://photos.app.goo.gl/K1bvjddjcHLsjFC1A (video primera modulacije pesmi)

Raziskala sva, kam lahko to pelje. Naredila sva poseben program, ki zajame tipke na tipkovnici in nato izvede BASK modulacijo pritisnjenega binarnega ASCII znaka + preamble (1010). To je zametek malwara, ki bi lahko posledično bil uporabljen za pridobivanje podatkov na zlonamerni način.

Naslednja stvar je pa, kako daleč lahko izvedeva to reč oz. kako hitro lahko prenašava podatke. Na začetku sva uporabljala zamik 500 ms, zanimalo pa naju je, kje je meja. Teoretično je meja okoli 16,7 ms na znak, torej da bi 12 bitov prenesla v 0,2 s (teoretično – ne upoštevava SNR in BER), kar ni hitro, ampak veliko bolj zadovoljivo.

Izvorna koda se nahaja na naslovu:
https://gitlab.com/coffee2stuff/bask-demodulation/-/tree/master/tempest_js

BASK demodulacija

Amplitude-shift keying (ASK) je vrsta amplitudne modulacije, ki predstavlja binarne podatke v obliki variacij amplitude signala. Binarni simbol 1 je predstavljen s prenosom nosilnega vala s fiksno amplitudo in fiksno frekvenco v danen časovnem obdobju.

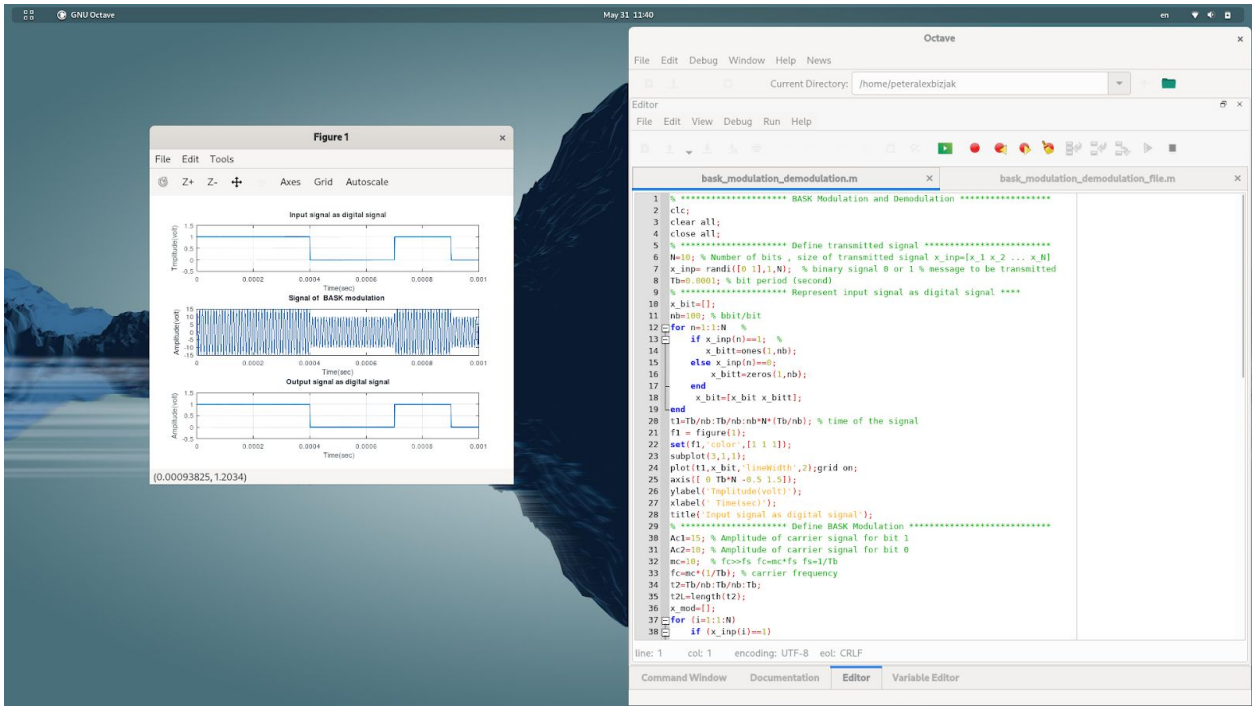
Obstajata dve vrsti demodulacije ASK: asinhrona in sinhrona.

Asinhroni detektor ASK je sestavljen iz polovičnega usmernika, nizkoprepustnega filtra in primerjalnika. Modulirani ASK signal se odda usmerniku, ki oddaja pozitiven polovični izhod. Nizkoprepustni filter zavira višje frekvence in daje zaznaven izhod v ovojnici, iz katere primerjalnik odda digitalni izhod.

Sinhroni detektor ASK je sestavljen iz kvadratnega detektorja, nizkoprepustnega filtra, primerjalnika in omejevalnika napetosti. Nizkoprepustni filter minimizira višje frekvence. Primerjalnik in omejevalnik napetosti pomagata dobiti čist digitalni izhod.

Naša implementacija demodulatorja BASK se uporablja v povezavi s temo napada TEMPEST v smislu, da z njeno pomočjo obdelamo ujeti vhodni signal in identificiramo binarno zaporedje, ki ga ta predstavlja.

Izvorna koda se nahaja na danem GitLab repozitoriju:
https://gitlab.com/coffee2stuff/bask-demodulation.



TempestSDR

TempestSDR je odprtokodno orodje, ki omogoča, da uporabimo kateri koli SDR za prejemanje nenamernega sevanja signala z zaslona in pretvorimo ta signal nazaj v živo sliko. Tako je mogoče brez fizičnih povezav videti, kaj je na zaslonu.

S pomočjo RTL-SDR.com nama je uspelo pridobiti delujočo različico TempestSDR ter jo nastaviti in zagnati v operacijskem sistemu Windows. Za uporabo programske opreme je treba v najboljšem primeru poznati ločljivost in hitrost osveževanja ciljnega monitorja. Če tega ne poznate, pa obstajajo grafi samodejne korelacije, ki dejansko pomagajo predvideti ločljivost in hitrost slikanja. Povezava do skladišča GitHub: https://github.com/rtlsdrblog/TempestSDR.

Pred kratkim je izšla GNU Radio, ponovna implementacija TempestSDR z imenom gr-tempest: https://github.com/git-artes/gr-tempest.

Še hiperpovezava do temelja programske opreme: https://github.com/martinmarinov/TempestSDR/blob/master/documentation/acs-dissertation.pdf