Etherify, Sonify
Soft TEMPEST w domu i w zagrodzie

Jacek Lipkowski SQ5BPF

https://github.com/sq5bpf/misc/blob/master/omh2020.pdf

~$ whoami

- Jacek Lipkowski SQ5BPF@lipkowski.org
  Hobby:
- Krótkofalarstwo. Znak SQ5BPF
  (od ponad 25 lat, licencja z roku 1993)
- Unixy, sieci. I ich psucie :) (od ponad 20 lat)
- Elektronika (od zawsze)

Pracuje w Pekao Financial Services Sp. z o.o.
Prezentacja ta jest moja i nie wyraża poglądów pracodawcy.

Co jest w prezentacji

Radio to obecnie modny temat w bezpieczeństwie :)

- Co to jest TEMPEST
- Co to jest Soft TEMPEST
- Jak mogą wyciekać dane
- Jak robią to ONI
- Jak moglibyśmy zrobić to my

Każdy z tych tematów zasługuje na osobny wykład, więc niestety zostaną one potraktowane bardzo skrótowo.

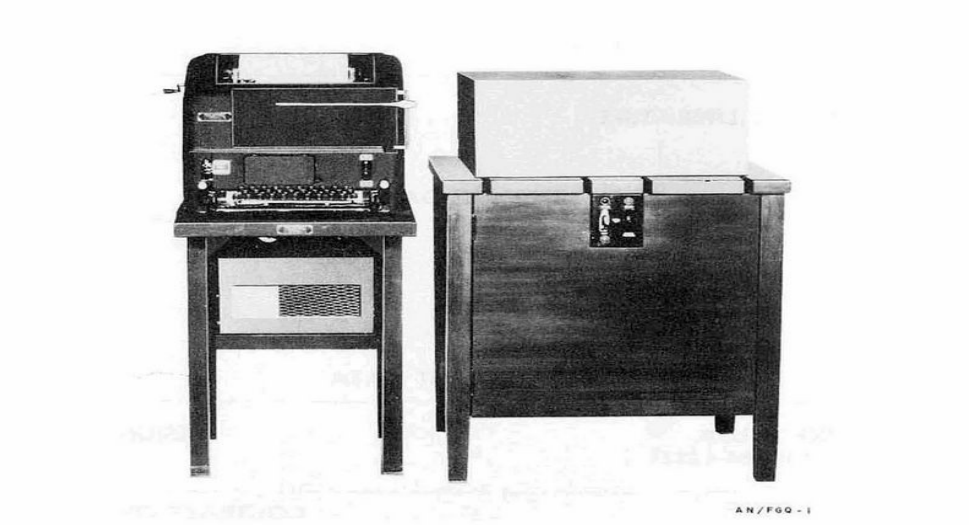
https://github.com/sq5bpf/misc/blob/master/omh2020.pdf

Trochę Historii

Trochę historii (od NSA)

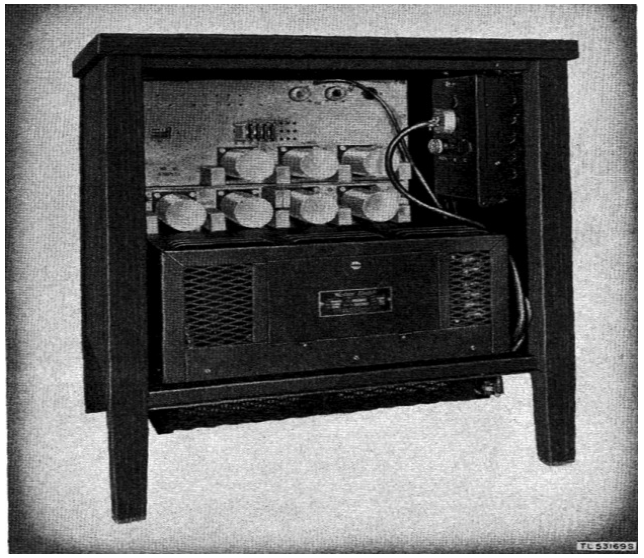
W 1943, podczas II wojny światowej, w Bell Labs odkryto że ich urządzenie szyfrujące do dalekopisów podwoduje zakłócenia radiowe. Tak jak wiele urządzeń elektronicznych, nic specjalnego.

Problem w tym że z tych zakłóceń można było odtworzyć plaintext.



Dalekopis AN/FGQ-1

(terminal szeregowy)



Szyfrator Bell 131-B2

(XOR w pudełku)

Trochę historii

Wedle artykułu NSA [1] wkrótce zapomniano o sprawie, w 1951 odkryło ją ponownie CIA (też na 131-B2). Nadano tajny kryptonim TEMPEST.

Wkrótce odkryto że istnieją inne kanały wycieku danych niż kanał elektromagnetyczny. Akustyczny, optyczny itd..

Problem zakłóceń elektromagnetycznych nie jest nowy, nie wiadomo czy nie zostało to wcześniej odkryte (ile razy, przez kogo) [2].

Wiadomo że służby wielu krajów chętnie wykorzystywały TEMPEST [4]

TEMPEST to kryptonim, ale chętnie nadal jest używany w literaturze. Po polsku: emisje ujawniające (tłum. „compromising emanations”)

Kryptonim TEMPEST został oficjalnie odtajniony w 2007 (64 lata po 1943!)

Cywilny TEMPEST

W 1985 Wim van Eck opublikował artykuł opisujący zdalne przechwytywanie obrazu z monitora. W skrócie: monitor generuje zakłócenia modulowane amplitudowo sygnałem video, można je odebrać, zdemodulować, dodać impulsy synchronizacyjne i w ten sposób uzyskać sygnał który można odtworzyć na drugim monitorze.

Działa to nie tylko na monitorach analogowych, ale też na cyfrowych. Przykład w [6]

Obecnie znamy wiele przykładów bocznych kanałów: np sidechannel attack na urządzenia wykorzystujące kryptografię co umożliwia odzyskanie klucza (pokrewne TEMPEST).

Literatura

[1] „TEMPEST: a signal problem” (1972) https://www.nsa.gov/Portals/70/documents/news-features/declassified-documents/cryptologic-spectrum/tempest.pdf

[2] „How old is TEMPEST?” http://cryptome.org/tempest-old.htm

[3] „TEMPEST 101” http://www.tscm.com/TSCM101tempest.html

[4] „Spycatcher” Peter Wright (pol. „Łowca szpiegów”)

[5] „Electromagnetic Radiation from VideoDisplay Units: An Eavesdropping Risk?” Wim van Eck http://www.tscm.com/vaneck85.pdf

[6] „TEMPEST HDMI demo” Oona Räisänen https://www.youtube.com/watch?v=BpNP9b3aIfY

„Boczne kanały”

„Boczny kanał” (brzydkie tłumaczenie sidechannel):

- Elektromagnetyczny (klasyczny TEMPEST) [1]
- Akustyczny (dźwięk, wibracje) [1] [10]
- Optyczny (światło z monitora, lampki sygnalizacyjne) [7]
- Termiczny (temperatura, emisja w dalekiej podczerwieni) [8]
- Moc pobierana z zasilania [9]
- Inne :) [11]

Literatura cd.

[7] https://www.researchgate.net/publication/327173348_Optical_TEMPEST

[8] „BitWhisper:Covert Signaling Channel be-tweenAir-Gapped Computersusing Thermal Manipulations” https://arxiv.org/pdf/1503.07919.pdf

[9] „PowerHammer:Exfiltrating Data from Air-GappedComputers through Power Lines” https://arxiv.org/pdf/1804.04014.pdf

[10] „Acoustic Surveillance of Physically Unmodified PCs” http://seclab.illinois.edu/wp-content/uploads/2011/03/LeMayT06.pdf

[11] „ODINI : Escaping Sensitive Data fromFaraday-Caged, Air-Gapped Computers viaMagnetic Fields” https://arxiv.org/pdf/1802.02700.pdf

[12] https://www.cl.cam.ac.uk/~mgk25/ih98-tempest.pdf

Soft TEMPEST

TEMPEST zajmuje się niezamierzoną emisją informacji.

Soft TEMPEST, to to samo tyle że zrobione celowo za pomocą oprogramowania :) [12]

- Eksfiltracja danych z niepodłączonego komputera za pomocą czegoś co tam „podrzuciliśmy”, a użytkownik jest tego nieświadomy
- Źródło testowej emisji dla osób zajmujących się obroną przed atakami TEMPEST
- Boczny kanał transmisji danych łączący różne urządzenia (którego użytkownik nie zawsze jest świadomy jeśli nie przeczyta drobnego druku :)

Już wymienione źródła piszą o celowej emisji [8] [9] [10] [11]

Badania akademickie

Ta jedna prezentacja w świetny sposób wyczerpuje temat:

„The Air-Gap Jumpers” Black Hat US 2018

Mordechai Guri, PhD The Head of R&D, Cyber-Security Research CenterBen-Gurion University of the Negev, Israel

https://i.blackhat.com/us-18/Wed-August-8/us-18-Guri-AirGap.pdf

Przykład:

„GSMem: Data Exfiltration from Air-Gapped Computers over GSM Frequencies”
https://www.usenix.org/system/files/sec15-paper-guri-update_v2.pdf

- Malware na PC
- Odbiornik w basebandzie telefonu GSM
- Brak źródeł
- Granty ($$$), dostęp do sprzętu (drogiego), stajnia z profesorami, tony specjalistycznego sprzętu itd

Badania „Służb”

[redacted]

Przykład:

[redacted]

-
- [redacted]
-
- Niewyobrażalna kasa, dostęp do sprzętu niedostępnego dla zwykłych śmiertelników, bataliony profesorów, technologie z UFO itd.

[13] NSA ANT Catalog https://www.eff.org/files/2014/01/06/20131230-appelbaum-nsa_ant_catalog.pdf

„ONI”

https://www.433aw.afrc.af.mil/News/Photos/igphoto/2000835639/

A może by też byśmy mogli?

- Małe fundusze
- Brak drogiego sprzętu
- Jesteśmy amatorami, nie wiemy co robimy

- Jesteśmy amatorami, nie wiemy że czegoś się nie da zrobić.
- Możemy opublikować źródła
- SDR-y są teraz tanie
- Mamy złom elektroniczny, srebrną taśmę i mnóstwo inwencji
- Spróbujmy sami to zrobić praktyczną implementację Soft TEMPEST

Radio!

(w końcu)

Software Defined Radio

- Kiedyś musielibyśmy zrobić lub kupić odbiornik.
- Klasyczny odbiornik (superheterodynowy, są inne): obwody wejściowe, przemiana na niższą częstotliwość, filtrowanie, demodulatory (osobny dla każdej emisji).
- Odbiornik SDR (hybrydowy): obwody wejściowe, przemiana na niższą częstotliwość, filtrowanie, przetwornik AD. Reszta w oprogramowaniu.
- Odbiornik SDR: obwody wejściowe (albo i nie), przetwornik AD. Reszta w oprogramowaniu.

- Obserwujemy tyle pasma ile ma przetwornik AD (analogowo-cyfrowy)
- Cała obróbka sygnału w oprogramowaniu (są gotowce, nie trzeba wymyślać koła od nowa), odbiór innej emisji/protokołu wymaga jedynie zmiany oprogramowania.
- Można przetwarzać wiele sygnałów w paśmie naraz
- Problem w tym że fabryczne odbiorniki SDR były drogie (np. USRP, WinRadio)

SDR :)

- Sytuacja się zmieniła. Bajecznie tanie SDR z odbiornika DVB-T (24-1700MHz dla tunera R820T)
- max około 2.5MHz pasma, ADC 8-bitowy (hack używający nidokumentowanego trybu dekodera DVB-T RTL2832)
- Dużo oprogramowania: gqrx, GNUradio, rtl_fm (pakiety np w debianie) i in. (jest też opgrogramowanie pod windows, mac os, android itp)
- 30 lat temu to byłaby ściśle kontrolowana technologia wojskowa (a teraz mamy ją za 60zł)

Jak znaleźć boczny kanał radiowy

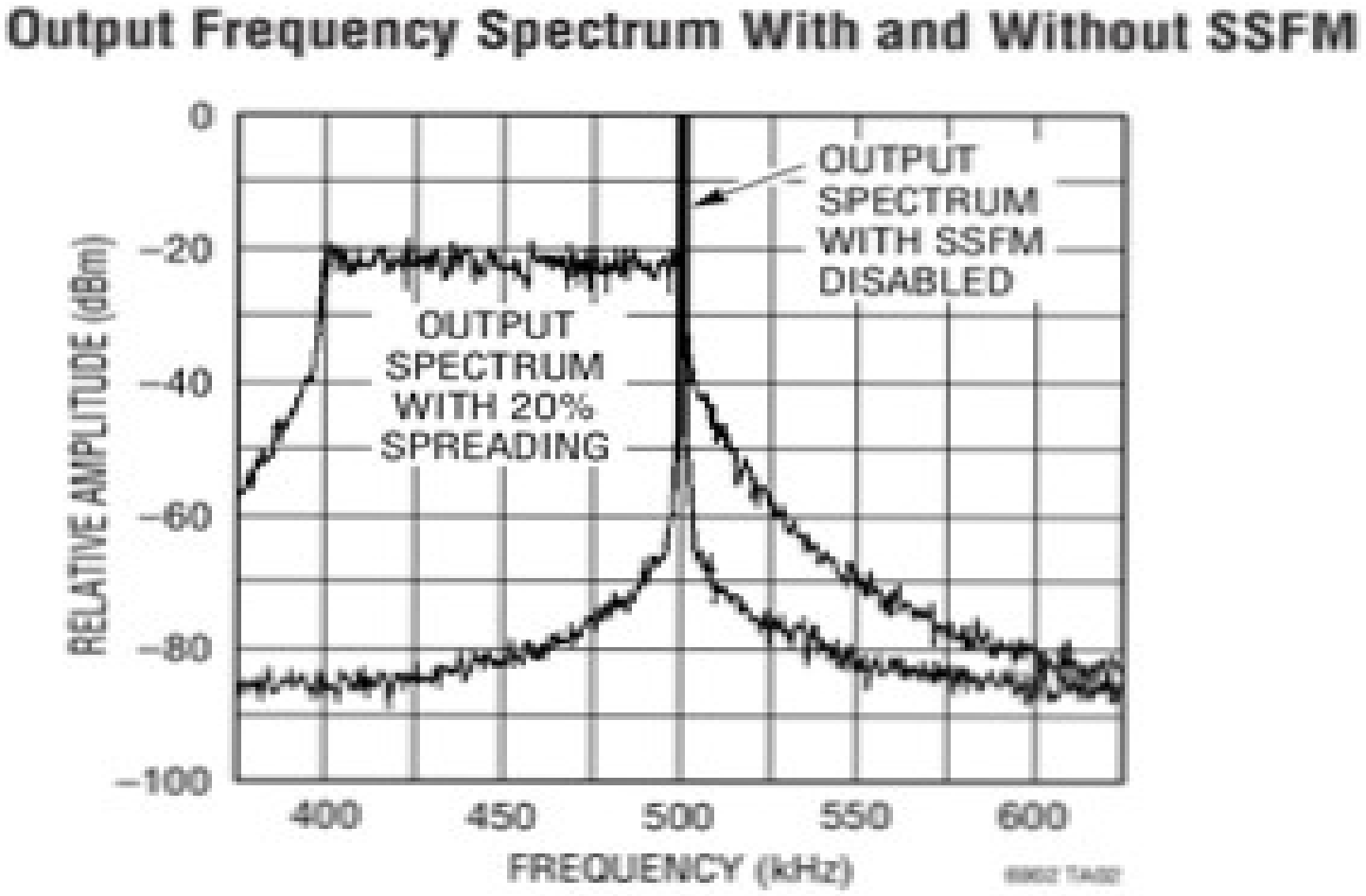
Prosto:

- Szukamy jakiegoś zegara albo szyny danych taktowanej częstotliwością rzędu przynajmniej parudziesięciu MHz (im więcej tym zwykle lepiej promieniuje)
- Patrzymy czy można jakoś to modulować (zmienić częstotliwość, wyłączyć itp.)
- Piszemy coś co moduluje to w takt danych
- Słuchamy na tej częstotliwości i na jej harmonicznych.
- I już :)

Ale:

- Urządzenia są robione tak aby nie zakłócać (bardziej niż wymagają tego przepisy, niekoniecznie lokalne)
- Często stosowane jest rozproszenie widma częstotliwości taktującej albo sygnału

Rozproszenie widma



- Rozproszenie widma „spread spectrum”
- Rozproszenie widma w dół „downspread”

https://www.analog.com/en/technical-articles/spread-spectrum-frequency-modulation-reduces-emi.html

Ethernet

Skrętka – skręcona linia symetryczna 100 omów (ang. twisted pair)

10base-T – kodowanie Manchester, zegar 20MHz, napięcie +-2.5V

100base-T – kodowanie 4B5B i NRZI, zegar 125MHz, napięcie +-1V

1000Base-T – kodowanie 4D-PAM5, zegar 125MHz, napięcie +-1V

Nie można zastosować rozproszenia widma ponieważ odbiornik po drugiej stronie kabla musi łatwo odzyskać sygnał zegarowy (synchronizując się do transmisji).

Linia symetryczna w kablu niekoniecznie jest idealnie symetryczna, transceiver ethernetowy niekoniecznie jest idealny, sygnal może wyciekać jeszcze jakąś inna drogą.

Może by się udało cos z tego nadać? :)

Etherify 1

- Najprostsza modulacja – zmieniamy prędkość interfejsu za pomocą ethtool
- Bitem 1 będzie sygnał na częstotliwości 125MHz - 100base-T, bitem 0 brak sygnału - 10base-T
- Nadajemy kodem Morse'a (bo można ocenić stosunek sygnał/szum na ucho i na ucho zdekodować jeśli ktoś potrafi). Pozatym tak jest fajnie :)
- Implementacja w shellu: etherify1.sh https://github.com/sq5bpf/etherify
- Odbieramy na 125MHz, albo dedykowanym odbiornikiem albo za pomocą SDR

Wynik: 2 Raspberry PI 4 połączone kablem 2m ze starterkitu Raspberry PI można odebrać z odległości 100m :)

Problemem jest czas przełączania prędkości interfejsu, RPI 4 – od razu, switch cisco catalyst 9200 – około 3s, laptop Dell Lattitude 6220 z debian buster, driver e1000e – około 5s.

Zakłócenia od pobliskich sieci ethernetowych.

Opis: https://lipkowski.com/etherify

DEMO Etherify 1

Etherify 2

- Najprostsza modulacja – ustawiamy prędkość 1Gbps
- Nadawanie dużego strumienia danych powoduje minimalną zmianę zegara w interfejsie ethernetowym, najprawdopodobniej spadek napięcia przy większym obciążeniu . Nadajnik: ping -f
- Nadajemy kodem Morse’a (bo można ocenić stosunek sygnał/szum na ucho i na ucho zdekodować jeśli ktoś potrafi). Pozatym tak jest fajnie :)
- Implementacja w shellu: etherify2.sh https://github.com/sq5bpf/etherify
- Odbieramy na 125MHz (lub na harmonicznych, dobrze działa 500MHz), albo dedykowanym odbiornikiem albo za pomocą SDR

Wynik: 2 Raspberry PI 4 połączone kablem 2m ze starterkitu Raspberry PI można odebrać z odległości 30m za pomocą prostej anteny kierunkowej :)

Problemem jest to że jest to specyficzne dla Raspberry PI 4, inne interfejsy też wykazują zmianę widma sygnału przy obciążeniu dużym ruchem, ale każdy inaczej.

Zakłócenia od pobliskich sieci ethernetowych.

DEMO Etherify 2

Ultradźwięki

- Człowiek słyszy częstotliwości od 20Hz do maksymalnie 20kHz (bardziej 15kHz)
- Karta dźwiękowa ma częstotliwość próbkowania zwykle min. 48KHz (lepsze 96kHz, 192 kHz)
- Zgodnie z twierdzeniem Nyquista można wygenerować sygnał do ½ częstotliwości próbkowania, czyli 24kHz (dla próbkowania 48kHz)
- Możemy korzystać z całego pasma 20kHz – 24kHz do transmisji danych.
- Głośniki w laptopie są małe, dobrze przenoszą wysokie częstotliwości.
- Nikt tego nie usłyszy :)

Sonify 1

- Generujemy sygnał na częstotliwości 21.5kHz
- Nadajemy kodem Morse’a (bo można ocenić stosunek sygnał/szum na ucho i na ucho zdekodować jeśli ktoś potrafi). Pozatym tak jest fajnie :)
- Implementacja w shellu: sonify1.sh https://github.com/sq5bpf/sonify
- Odbieramy za pomocą karty dźwiękowej (można z zewnętrznym mikrofonem kierunkowym), programem który zamieni częstotliwość 21.5kHz na 650Hz (słyszalna). Np. DL4YHF Spectrum Lab

  Wynik: 2 Dell Latitude 6220 można odebrać 10m w innym pokoju 10m dalej za pomocą wbudowanego mikrofonu w Dell Latitude 5310 :)

  Problemem jest echo i doppler (nie ruszać się, nie spacerowac z laptopem).

  Opis: https://lipkowski.com/sonify

DEMO Sonify 1

Co można poprawić?

- Lepsza modulacja: inna przepływność, łatwiejsze do dekodowania maszynowo
- Lepsze urządzenie odbiorcze: antena o większym zysku, mikrofon kierunkowy
- Pamiętajmy że to tylko demo :)

Czy to się zdarza w rzeczywistości?

- Ultradźwięki: „Talking Behind Your Back: Attacks and Countermeasures of Ultrasonic Cross-Device Tracking” BlackHat 2016

  https://www.blackhat.com/docs/eu-16/materials/eu-16-Mavroudis-Talking-Behind-Your-Back-Attacks-And-Countermeasures-Of-Ultrasonic-Cross-Device-Tracking.pdf

  BadBIOS: https://en.wikipedia.org/wiki/BadBIOS (nie potwierdzone)
- Ethernet: Nie znalazłem odpowiednika etherify w literaturze

- Co mają do powiedzenia na ten temat „służby”:

  [redacted]

  [redacted]

  Brak informacji o pasywnej eksfiltracji, ale na pewno są zainteresowani dowolną eksfiltracją danych np. (z NSA ANT Catalog [13]):

  SURLYSPAWN - Data RF retro-reflector. Provides return modulated with target data (keyboard, low data rate digital device) when illuminated with radar.

  Unit Cost: $30

Z ostatniej chwili: Etherify 3 i 4

Etherify 3 https://lipkowski.org/etherify3

- Okazuje się że ethernet Raspberry PI 4 generuje fatalne zakłócenia
- Niepodłączony do ethernetu raspberry pi 4 można odebrać z 50m
- Demo (jeśli jest czas)

Etherify 4 https://lipkowski.org/etherify4

- Implementacja na „normalny” sprzęt (np porządny laptop)
- Do transmisji używa wolnego alfabetu morsea (QRSS CW)
- Możliwa implementacja na sprzęt sieciowy (via SNMP, tclsh itp.)
- Demo (jeśli jest czas)

Linki

Etherify:
https://lipkowski.com/etherify
https://github.com/sq5bpf/etherify

Sonify:
https://lipkowski.com/sonify
https://github.com/sonify

SQ5BPF@lipkowski.org

PYTANIA?

VY 73

Jacek Lipkowski SQ5BPF
SQ5BPF@lipkowski.org

Link do prezentacji:
https://github.com/sq5bpf/misc/blob/master/omh2020.pdf