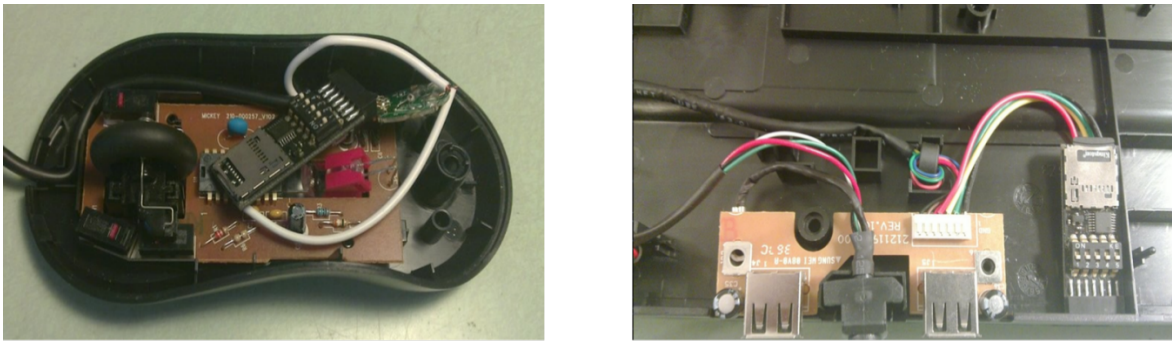
Trojany sprzętowe - niedoceniane zagrożenie
Autor: Adam Kostrzewa
Strona: https://adamkostrzewa.github.io/
Twitter: https://twitter.com/systemWbudowany
Artykuł przedstawia prywatne opinie i poglądy autora.

Czy można ufać naszym procesorom, chipom i układom scalonym? Wielu specjalistów pomija to pytanie z prostego powodu – ze względu na wysokie koszty produkcji sprzętu użytkownik nie ma żadnej (albo bardzo nikłą) możliwość wyboru produktów. W przypadku serwerów, stacji roboczych czy komputerów typu mainframe, firmy najczęściej decydują się zatem na zakup komponentów od sprawdzonych dużych dostawców, na przykład AMD, INTEL, NVIDIA, ASUS, licząc na uczciwość producentów a także na to że wysoka popularność doprowadzi do szybkiego wykrycia ewentualnego zagrożenia. Tak proste rozwiązanie nie jest już dostępne dla wielu podmiotów z branży telekomunikacyjnej czy rynku systemów wbudowanych (np. Internet-of-Things) a także elektroniki przemysłowej. Rynek zalewany jest dużą ilością taniej elektroniki z dalekiego wschodu a przez media przetaczają się co jakiś czas skandale które dotykają nawet takich potentatów jak NetGear czy Cisco.

Czy jest zatem powód do obaw? Czy układy scalone od dużego dostawcy są bezpieczne? Na czym polega zagrożenie stwarzane przez trojany sprzętowe i czy warto bliżej zapoznać się z tematem? W tym krótkim tekście postaramy się odpowiedzieć na te pytania.

Zasady i cele działania

Trojan sprzętowy, to ukryta przed użytkownikiem funkcja układu elektronicznego, która obok znanej funkcjonalności, może zostać wykorzystana przez osoby trzecie w niepożądanym celu. Trojan sprzętowy („hardware-owy") działa bardzo podobnie do trojana w oprogramowaniu. W przypadku oprogramowania użytkownik sam instaluje interesującą aplikację, która dodatkowo (poza podstawową funkcjonalnością) bez jego wiedzy np.: zamienia komputer w element botnetu. Podobnie jest ze sprzętem. Użytkownik kupuje interesujący go komponent, np. klawiaturę lub myszkę z interfejsem USB patrz Rysunek 1, która oprócz pobierania informacji o wciskanych klawiszach posiada jeszcze układ zdolny do generowania takich sygnałów [1].



a) b)

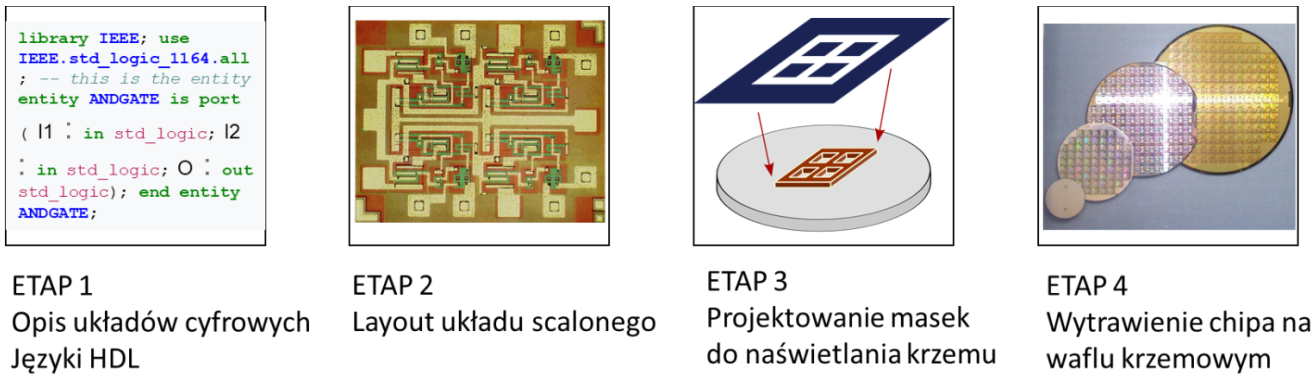
Rysunek 1 Trojan sprzętowy w myszce (a) i w klawiaturze (b) źródło JP Dunning ".ronin" BlackHat Asia 2014 [1]

Oczywiście ten „walor" produktu jest pominięty w dokumentacji. Następnie układ ten jest w stanie odtworzyć zapisane w pamięci flash makra i generować polecenia. System nie potrafi ustalić kto

wydaje polecenia i efektywnie trojan ma uprawnienia użytkownika który korzysta z klawiatury. Podobieństwa pomiędzy trojanami sprzętowymi i softwarowymi wynikają głównie z celów ich działania: infiltracji i eksfiltracji [2]. Infiltracja polega na wprowadzaniu zmian w infrastrukturze zakłócających poprawna prace systemu np.: modyfikację danych, modyfikację funkcjonalności czy bardziej wyrafinowane działania np. celowe osłabienie algorytmów kryptograficznych przez zmianę klucza na znany. Eksfiltracja polega na wysłaniu do atakującego informacji z systemu np: danych (projektowych, pracowników, logów etc), kluczy kryptograficznych, danych potwierdzających tożsamość, wartości startowych generatorów liczb pseudolosowych czy wreszcie kodu aplikacji bądź dokumentacji projektów.

Na czym polegają zatem różnice? Implementacja trojanów sprzętowych to zadanie skomplikowane, wymagające wysoko wykwalifikowanego personelu i zaawansowanych narzędzi a więc generujące wysokie koszty. Produkcja sprzętu jest procesem wielostopniowym w którym biorą udział oprócz podmiotów z branży elektronicznej również te z branży chemicznej czy nawet mechanicznej (systemy wbudowane w maszynach do pracy w wysokich temperaturach czy wibracjach). Dodatkowo wymaga dużych nakładów finansowych n.p.: pomieszczenia o wysokim stopniu czystości tzw. “cleanroom” w laboratoriach do produkcji krzemu, mikroskopy, lasery do cięcia wafli krzemowych etc. Z praktyki, np.: systemów wbudowanych, powszechnie wiadomo, że wiele funkcji można wykonać za pomocą dedykowanego układu elektronicznego (*ang. application-specific integrated circuit AISIC*) bądź softwarowo za pomocą programowalnego kontrolera. Różnica polega na koszcie i elastyczności rozwiązania. W przeciwieństwie do oprogramowania, w większości wypadków wprowadzonej funkcjonalności sprzętowej ciężko zmienić w procesie post-produkcyjnym tzn. po wprowadzeniu produktu na rynek [19]. W przypadku błędnej pracy układu na krzemie producent może próbować firmawerem ograniczać jego skutki jednak pełna wymiana bramek czy tranzystorów jest niemożliwe, szczególnie pryz procesach poniżej 32nm. To wymusza długą fazę testową dodatkowo zwiększając i tak już wysokie koszty produkcji. Dedykowany układ na kryemie jest drogi w projektowaniu i nie podlega zmianom dlatego też wymaga jasno określonej specyfikacji i długiej serii produkcyjnej amortyzującej kosztów. W odniesieniu do trojanów sprzętowych znaczy to, że ich wprowadzenie wiąże się dla podmiotu z dużym ryzykiem tak finansowym jak i wizerunkowym. Może to skutkować utratą zaufania do producenta a nawet jego bankructwem. Dobrym przykładem są chińskie firmy Huawei i ZTE które po fali krytki oskarżającej o stoswanie trojanów sprzętowych i softwarowych w produktach telekomunikacyjcnych [4], [5], [6] znalazla sie na celowniku kogresu USA. W 2012 kongres zarządał wykluczenia firmy z przetargów publicznych na terenie kraju [7],[8]. Wymowne nagłówki amerykanskiej prasy NYT „Panel kongresu US uznaje Huawei i ZTE za zagrożenie dla bezpieczeństwa narodowego” czy niemieckiej telewizji publicznej Arte „Huawei szpieg z Chin” najlepiej oddają sytuację.

Trojany sprzętowe mają jednak niepodważalne zalety. Ich wykrycie i analiza działania jest zdecydowanie trudniejsza niż w przypadku oprogramowania [2]. Inżynieria wsteczna układów scalonych wymaga mikroskopów laboratoriów chemicznych oraz bardzo wysokich nakładów finansowych. Trojanów sprzętowych często nie da się blokować software'owo tzn. atakujący nie musi obawiać się nowej poprawki czy wersji systemu łatającej podatność [20]. Dodatkowo cykl życia układów elektronicznych jest zdecydowanie dłuższy niż oprogramowania [19]. Często jednym z podstawowych wymagań klientów jest kompatybilność wsteczna pozwalająca na wykorzystanie tego samego produktu przez wiele lat np. terminale na lotniskach, w bankomatach czy systemach wbudowanych. 15 letnia jednostka licząca zamontowana w samochodzie czy ośmioletni laptop nie jest czymś wyjątkowym. Pozwala to atakującemu na rozciągnięciu całego ataku w czasie i przeprowadzanie bardzo skomplikowanych ataków.



Rysunek 2 Etapy powstawania dedykowanego układu elektronicznego na krzemie, źródła wikipedia commons.

Gdzie i kiedy można wprowadzić trojana sprzętowego?

Proces powstawania współczesnej elektroniki jest bardziej złożony niż proces powstawania oprogramowania [19]. Składa się on z czterech podstawowych etapów przedstawionych na Rysunek 2. Celem pierwszego etapu jest zaprojektowanie i przetestowanie logicznych funkcji układu. Podstawowym narzędziem są tutaj języki opisu sprzętu (*ang Hardware Description Language, HDL*) np. VHDL, Verilog czy systemVerilog. W zależności od celów projektant może koncentrować się tylko na wysokopoziomowym opisie układu np. sygnały na wejściach i wyjściach poszczególnych elementów, albo zdefiniować bloki logiczne włączając proste funkcje logiczne jak i skomplikowane elementy np. pamięci, rejestry. Możliwe jest też mieszanie tych dwóch podejść. Dla łatwiejszego zrozumienia, pewną analogią z oprogramowania jest podejście w którym programista albo używa języków obiektowych pozwalających na pracę na wyższym poziomie abstrakcji albo używa języków niskopoziomowych (assemblera) gdy np.: chce uniknąć zbyt dużej ingerencji kompilatora w konkretny fragment kodu. Jednak w obu przypadkach projekt musi zostać “przetłumaczony” na język maszynowy.

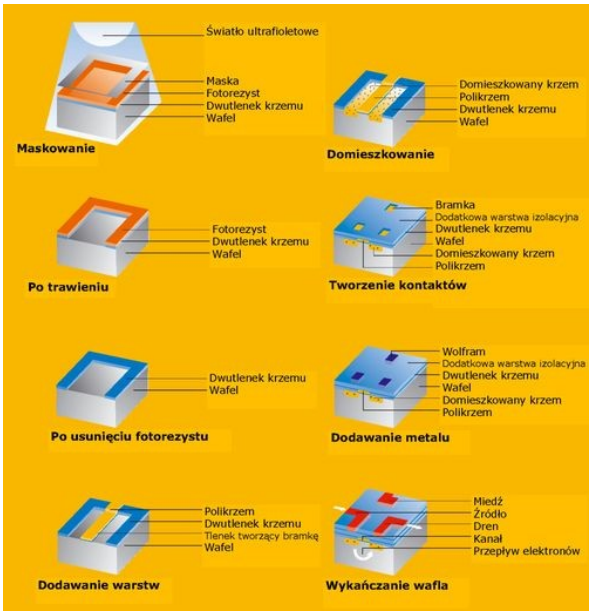
W przypadku układu elektronicznego, celem “tłumaczenia” jest uzyskanie konkretnych układów elektronicznych (tranzystorów, bramek logicznych etc.). Odbywa się to w drugim etapie zwanym potocznie syntezą. Rezultatem jest konkretny układ (ang. layout) elementów, który może zostać wykonany w krzemie. W dzisiejszej praktyce przemysłowej synteza konkretnych obwodów elektronicznych odbywa się często automatycznie za pomocą narzędzi np.: Vivavdo, Xilinx ISE - podobnie do procesu kompilacji.

Dominującą na rynku metodą produkcji układów krzemowych jest naświetlanie i zmywanie. Proces tworzenia tranzystorów jak i połączeń pomiędzy nimi odbywa się za pomocą domieszkowania prowadzonego tylko w konkretnych miejscach krzemowego wafla. Ten efekt uzyskuje się za pomocą naświetlania powierzchni wafla poprzez maski osłaniające lub odkrywające wybrane fragmenty chipu. Za tworzenie masek jest odpowiedzialny etap trzeci produkcji układu. Dla zainteresowanych czytelników więcej informacji tutaj.
Etap czwarty to fizyczne wykonanie układu w krzemie za pomocą wyżej wspomnianej metody i przygotowanych masek. Powszechną praktyką przemysłową jest specjalizacja firm i ośrodków. W rezultacie, każdy z tych etapów jest najczęściej wykonywany przez inny podmiot często w innym miejscu na świecie.

Powstawanie trojana

Dodanie trojana sprzętowego może odbyć się na każdym etapie procesu wytwarzania chipu [3], [2], . Praca na pierwszym etapie produkcji jest najmniej kosztowna, ale i najtrudniejsza do ukrycia. Wprowadzenie backdoor-a polega po prostu na dodaniu nowej funkcjonalności w języku HDL, podobne do dodania nowej klasy w projekcie softwarowym. Implementacja i testy są relatywnie tanie gdyż atakujący pracuje na wysokim poziomie abstrakcji np.: może wspierać się wysokopoziomową symulacją w Modelsimie albo SystemC. Trudno jest jednak ukryć takie działania gdyż w opis logiczny układu zaangażowana jest zazwyczaj spora grupa pracowników i personelu firmy. Pewną pomocą jest tutaj celowe komplikowanie kodu by był on jak najmniej czytelny dla osób niezwiązanych bezpośrednio z funkcjonalnością. W procesie produkcji oprogramowania działania takie nazywa się procesem zaciemniania kodu (także obfuskacja, z ang. *obfuscation*). Chodzi o takie przekształcanie elementów układu scalonego by zachowana była semantyka, ale znacząco utrudnione rozumienie.

Na drugim etapie produkcji, zmian dokonuje się już na poziomie obwodów elektrycznych. Można dodać nowe elementy, połączyć istniejące bądź wykorzystać pewne schematy pracy narzędzi syntezy. Pewną analogią softwarową jest praca reversera który modyfikuję pracę już skompilowanego kodu (aplikacji) na poziomie języka niskopoziomowego. W przypadku sprzętu jak i oprogramowania zazwyczaj wymaga to wykwalifikowanego i doświadczonego personelu co podwyższa koszty. Zmiany są jednak trudniejsze do wykrycia.



Rysunek 3 Etapy powstawania chipu na waflu krzemowym, źródło fraz.pc i Intel.

Zmiany na trzecim etapie są najtrudniejsze do wykonania a przez to najbardziej kosztowne. Znając specyfikę procesu trawienia wafla krzemowego można modyfikować maski tak, aby dodać nowe elementy bądź modyfikować już istniejące. Jest to jednak bardzo skomplikowane i czasochłonne ze względu na złożoność procesu chemicznego. Najlepszym przykładem jest liczba niezbędnych operacji, przedstawiona na Rysunek 3: maskowanie, domieszkowanie, trawienie, tworzenie kontaktów, usunięcie fotorezystu, dodawanie metalu, dodawanie warstw wykańczanie wafla. Wprowadzenie na trzecim etapie produkcji układu daje jednak atakującemu możliwość ingerencji w produkty innych firm i podmiotów. Firmy takie jak NXP/Phillips, Sony, Microsoft często zlecają trawienie krzemu

podwykonawcom. Są to konkretne wyspecjalizowane fabryki (*ang. foundries*) z których przeważająca ilość znajduje się w Azji: Chinach, Tajwanie, Japonii, Korei. Brak własnych fabryk rodzi zatem niebezpieczeństwo utraty kontroli nad funkcjonalnością produktu mimo obniżenia kosztów produkcji. Dodatkowo proces wprowadzenia funkcjonalności sprzętowej prowadzącej do powstania trojana może być rozłożone pomiędzy różnymi etapami produkcji. Jako przykład, w etapie pierwszym różne fragmenty funkcjonalności implementują różne grupy robocze. Nie wzbudza to podejrzeń gdyż funkcjonalności te nie są same w sobie niebezpieczne. Dopiero połączenie ich w całość, które odbywa się za pomocą małych modyfikacji na etapie drugim bądź trzecim, tworzy podstawową logikę i prawdziwe zagrożenie. W ten sposób można ograniczyć niebezpieczeństwo wykrycia na etapie produkcyjnym.

Przykładowy atak

Opiszmy zatem zagrożenie na **teoretycznym/spekulatywnym** przykładzie ataku inspirowanego ostatnimi doniesieniami prasowymi, np. lukami w CClenearze czy płytami głównymi DELLa. Po pierwsze, ze względu na specyfikę musimy uwzględnić wysoki stopień wyrafinowania atakującego rozumiany jako doświadczenie techniczne i posiadane zasoby. Załóżmy zatem że plan operacji jest przygotowany przy współudziale/współpracy producenta podzespołów elektronicznych z dalekiego wschodu. Atakujący ma profesjonalny zespół inżynierów i zasoby finansowe pozwalające na rozłożenie całego przedsięwzięcia w czasie - nawet na kilka lat. Pierwszym krokiem będzie wprowadzenie zainfekowanego sprzętu na rynek. Duże firmy, np. opisane wcześniej skandale z ZTE [5] i HUAWEI [6] czy Xiaomi[9], teoretycznie mogą (tzn. mają takie możliwości techniczne) od razu wbudować go w swój produkt - pierwszy etap. To jednak wiąże się z dużym ryzykiem utraty zaufania klientów w przypadku wykrycia. Dodatkowo raz zmodyfikowanych produktów nie sposób wycofać z rynku. Dlatego atakujący może zdecydować się na alternatywne rozwiązanie: wprowadzenie w module dostarczonym przez podwykonawcę. Może się to odbyć przy jego współpracy na pierwszym bądź drugim etapie produkcji, bądź też bez jego wiedzy lub zgody na drugim i trzecim etapie, jak opisano wcześniej. Przykładem takiego ataku jest historia związana z oświadczeniem firmy DELL która ostrzegała użytkowników, że w niektórych produktach (płytach głównych) występuje sprzętowy najprawdopodobniej wprowadzony przez podwykonawcę. By dodatkowo zabezpieczyć się na wypadek wykrycia, atakujący projektuje funkcjonalność tak aby można ją było uznać za błąd projektowy. Słynnym przykładem są dynamicznie zmienialny mikrokod AMD które według producenta miały służyć łataniu (pathowaniu) działającego procesora. Okazało się jednak że umożliwiają atakującemu wprowadzenie tylnej furtki (backdoora) w procesie aktualizacji i ominięcie systemu ochrony sprzętowej rodząc wiele spekulacji i kontrowersji. Oczywiście AMD uznało sprawę za błąd projektowy czli tzw. bugdoor.

Następnie gotowy produkt jest rozprowadzany na rynku. Załóżmy że atakujący na tym etapie nie aktywuje tylko czeka i pozwala użytkownikom cieszyć się podstawową funkcjonalnością układu. Po pewnym, czasie np. od pół/roku do dwóch lat następuje nasycenie rynku. Produkt pojawia się w naturalny sposób (w wyniku zakupu, napraw gwarancyjnych, podmiany czy czasowej wymiany sprzętu) w wybranych obiektach np. bankach, fabrykach, instytucjach rządowych.

Wtedy atakujący mógłby przystąpić do właściwej częsci operacji. Istnieje wiele sposobów aktywacji i komunikacji z trojanami: przez tylne furtki w protokole np. [10], przez manipulacje protokołem np. [11], atak bocznym kanałem np. [12] czy w końcu przez wprowadzenie błędu np. [13]. Tylne furtki w protokole to jedna z popularniejszych metod, (np. [14], [15], [16], [17], [18]) która polega na wbudowaniu sterowania trojanem w już istniejącą komunikację używając znaków wodnych bądź

technik stegano tzn. wybrane i znane tylko atakującym fragmenty transmisji służą do wydawania poleceń złośliwemu układowi bądź odbioru danych.

Nasz atakujący wybiera zatem pierwszy sposób. Aktywacja i wyprowadzanie danych może nastąpić jako element procesu instalacji / aktualizacji sterowników. Połączenie z serwerami producenta odbywa się wtedy za wiedzą i zgodą użytkownika. Warto wziąć pod uwagę i bardziej wysublimowane formy ataku. Możliwe jest zakupienie praw do produktu firmy trzeciej niepowiązanej bezpośrednio z atakującym. Świetnym przykładem jest niedawny skandal z aplikacją CCleaner. Znany i lubiany program CCleaner przez prawie miesiąc (od 15 sierpnia, do 12 września kiedy kolejny update usunął podatność) infekował miliony komputerów swoich użytkowników. Badacze z firmy Talos, którzy jako pierwsi publicznie opisali atak, opublikowali raport na temat celów ataku wymieniając fabrykę sprzętu AGD Samsunga w Polsce.

Wykrycie ataku będącego kombinacją sprzętowo softwarową byłoby jeszcze trudniejsze niż w przypadku CCleanera. Po pierwsze, jak opisaliśmy wcześniej, całość funkcjonalności służącej ominięciu zabezpieczeń systemu jest wykonana w sprzęcie. Jak pokazują wyniki uzyskane przez badaczy zabezpieczeń sprzętowych, np. backdoorow w procesorach, warstwa oprogramowania jest praktycznie bezbronna. Większość mechanizmów zabezpieczających od lat 70tych opiera się na założeniu, że sprzęt na którym pracują zachowuje się według zasad ściśle określonych w specyfikacji np. pierścienie ochrony pracy procesora.
W rezultacie, atakujący musi jedynie zamaskować komunikację z trojanem dokonywaną w celu infiltracji albo eksfiltracji. Techniki kryptograficzne np. stegano czy znaki wodne, dowodzą że można wykonać to nie wzbudzając podejrzeń - nie znając protokołu komunikacyjnego jest niezwykle trudno wykryć przebieg i cel połączenia. W konsekwencji atakujący może odczytać i interesujące go wartości i prowadzić sterowanie np. za pomocą danych diagnostycznych wysłanych za wiedzą i zgodą użytkownika.

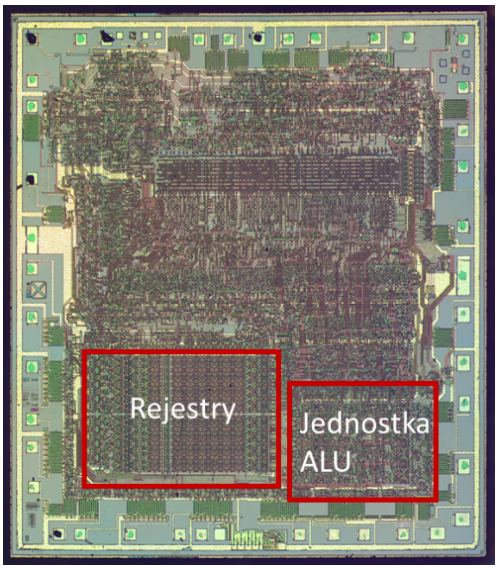
Metody i koszty obrony

Proces detekcji trojanów sprzętowych jest drogi i skomplikowany. Pierwszą barierą są kwestie techniczne. Narzędzia do rewersu i analizy oprogramowania są stosunkowo łatwo dostępne a niezbędne prace można przeprowadzić na zwykłej stacji roboczej. Zazwyczaj wymaga disassemblera wspieranego przez narzędzia do wstecznej analizy kodu np. IDAPro czy radare. W przypadku chipów krzemowych jest to proces zdecydowanie bardziej skomplikowany. Po pierwsze krzem sprzedawany jest w szczelnych plastikowych (epoksydowych) obudowach które trzeba usunąć by móc fizycznie zobaczyć układ. Proste metody mechaniczne, np. szlifowanie, mogą uszkodzić chip. Nawet mała rysa w przypadku technologii wykonania poniżej 90nm może zdecydowanie utrudnić, bądź wręcz uniemożliwić, analizę wsteczną. W profesjonalnych laboratoriach stosuje się zatem silnie żrące kwasy rozpuszczające wszystko oprócz krzemu. Ich użycie wymaga jednak odpowiedniej infrastruktury np. odpowiednie nawiewy, narzędzia (pipety, odczynniki), a co ważniejsze odpowiednio przeszkolonego i doświadczonego personelu. Fizyczny dostęp do powierzchni układu to dopiero pierwszy krok. W następnym, do wykonania analizy niezbędne są silne mikroskopy. Już sama wielkość układu np. 27nm jest barierą nie do przeskoczenia dla wielu hobbystów ze względu na wymagane powiększenie i jakość obrazu. Dodatkowo wiele układów produkowanych jest w technologii 3D tzw. wielowarstwowej. W tym wypadku trzeba przekroić bądź wypalić warstwę by zobaczyć ukryte fragmenty do czego niezbędne są precyzyjne lasery.

Następną kwestią utrudniającą inżynierie wsteczną jest współbieżność pracy większości obwodów tzn. elementy procesora takie jak stopnie potoku czy rejestry pracują równolegle ze sobą w czasie. Z teorii

programowania wiadomo, że debug aplikacji wielowątkowej jest czasochłonny i kosztowny. Proces ten wymaga nie tylko znajomości wartości sygnałów stymulujące pracę układu, ale także ich przebiegu w czasie i współzależności. Te same problemy w sposób bezpośredni odnoszą się do obwodów elektronicznych. Dodatkowo w przypadku prostych układów metoda, np. jednostki 8081, metoda testowania wszystkich możliwych zestawów wejść – analogowych, cyfrowych jak i programowalnych (ISA) choć pracochłonna jest jednak możliwa. W przypadku nowoczesnych jednostek np. procesorów Intela, jest już bardzo trudna.

Dodatkowo na końcowy kształt układu elektronicznego mają też wpływ czynniki ekonomicznych takie jak cena pamięci czy podzespołów, które dynamicznie zmieniają się w czasie. Jak wiadomo z rewersu starszych generacji procesorów budowa wielu elementów układu nie musi być oczywista i prosta do zrozumienia z przyczyn czysto praktycznych. Dobrym przykładem są procesory z lat 70tych i 80tych które ze względu na wykonanie za pomocą starych procesów technologicznych (8000 – 1500 nm) są częstym obiektem analizy wstecznej dokonywanej przez amatorów i profesjonalistów. W procesorze Z80 jednostka arytmetyczna według specyfikacji jest 8 bitowa jednak jej wykonanie odbywa się za pomocą czterech rejestrów w dwustopniowym potoku, patrz Rysunek 4. Pozwoliło to projektantom zaoszczędzić cenne miejsce na chipie a użytkownik / programista nie odczuwa różnicy, gdyż wykonanie programu jest zdefiniowanie przez wydajność innych fragmentów układu i dlatego stopnie potoku są transparentne. Kolejnym problemem jest kompatybilność wsteczna, która często wymusza istnienie artefaktów projektowych tzn. elementów które są umieszczane na chipie ze względu na starsze programy a nie obecny stan techniki. Dobrym przykładem jest proces bootowania (startu) architektury x86 który jest skomplikowany i trudny do zrozumienia dla wielu użytkowników.



Rysunek 4 Zdjęcie struktury układu Z80, źródło WIKIPEDIA oznaczenia autora.

Na koniec pozostają jeszcze błędy projektowe. Dobrym przykładem jest niewłaściwe zabezpieczenie złączy serwisowych i diagnostycznych. Ich podstawowym celem jest umożliwienie producentowi na wprowadzanie aktualizacji czy odzyskanie danych po awarii. Bardzo często błędy w zabezpieczeniach umożliwiają atakującym wprowadzenie tą drogą złośliwego oprogramowania infekującego podłączone maszyny np. J-TAG port w routerach umożliwiających podmianę firmware-u i zamianę maszyny w „Rogue Access Points“, czy złącze serwisowe w dysku twardym które umożliwiające atakującemu na bezpośrednie odczytanie jego zawartości.

W rezultacie odróżnienie elementów budowy trojana od pozostałych fragmentów układu jest zadaniem bardzo trudnym technicznie wymagającym nie tylko wysokich nakładów finansowych i dobrej infrastruktury, ale co ważniejsze bardzo dużego doświadczenia a często i intuicji.

Podsumowanie

Wprowadzenie „koni trojańskich" do komercyjnie dostępnego układu scalonego jest technicznie możliwe. Dowodzą tego publikacje, prace praktyczne a także doniesienia prasowe. Profilaktyka tego zagrożenia jest trudna i droga. W rezultacie, ze względu na swą specyfikę jest ona w pełni możliwa jedynie dla dużych koncernów bądź instytucji na szczeblu krajowym np. urzędy, politechniki. Zagrożenia tego typu są przede wszystkim groźne dla dużych instytucji: fabryk, banków, agend rządowych czy systemów wojskowych. W tych przypadkach potencjalne rezultaty ataku mogą uzasadniać wysiłek i ryzyko związane z wbudowaniem trojana w produkt komercyjny. Teoretycznym interesującym rozwiązaniem problemu mogło by być zdelegalizowanie ukrytych funkcjonalności w produktach elektronicznych. W tym wypadku producent byłby prawnie zobowiązany do przekazania całości dokumentacji. W praktyce jednak rozwiązanie takie jest trudne do zrealizowania. W grę wchodzą ograniczenia patentowe (producent może nie zgadzać się na ujawnianie całości rozwiązania) jak również formalne (jaka instytucja i w jakim trybie miała by podejmować operacje weryfikacyjne). Dlatego dla wielu firm jedynym rozwiązaniem pozostaje dokładne testowania kupionych/zamówionych komponentów, szukanie najdrobniejszych anomalii, porównywanie ze specyfikacjami a także w przypadku wątpliwości zamawianie drogich ekspertyz. Żadna z tych metod nie gwarantuje jednak 100% bezpieczeństwo. Ten cel można osiągnąć jedynie poprzez całkowitą kontrolę nad całością procesu produkcyjnego tzn. własną produkcję. Taką decyzję z przyczyn politycznych podjęło już wiele krajów w tym nasi sąsiedzi. Niemcy w dziedzinie telekomunikacji (np. inicjatywa „Email made in Germany") czy Rosja z własnymi stacjami roboczymi i procesorami wspierającymi architektury x86. W kuluarach zaczyna się mówić o procesie podziału rynku sprzętu elektronicznego w którym o wyborze dostawcy decydują kwestie polityczne a nie walory techniczne czy ekonomicznie. Warto o tym pamiętać czytając prasowe doniesienia o kolejnych udanych atakach z dalekiego wschodu na doskonale zabezpieczone rządowe czy przemysłowe systemy.

Odnośniki

[1] „Building Trojan Hardware at Home|, JP Dunning ".ronin", BlackHat Asia 2014
[2] „Stopping Hardware Trojans in Their Tracks|", Subhasish Mitra, H.-S. Philip Wong And Simon Wong, IEEE Spectrum 2015
[3] „Hardware-Trojaner in Security-Chips", Peter Laackmann, Marcus Jank, 32C3 CCC Media Congress 2015
[4] http://www.nytimes.com/2012/10/09/us/us-panel-calls-huawei-and-zte-national-security-threat.html
[5] https://www.theguardian.com/technology/2012/oct/08/china-huawei-zte-security-threat
[6] https://www.forbes.com/forbes/welcome/?toURL=https://www.forbes.com/sites/simonmontlake/2012/10/08/u-s-congress-flags-chinas-huawei-zte-as-security-threats
[7] http://www.spiegel.de/netzwelt/netzpolitik/us-kongress-will-chinas-telekom-firmen-huawei-und-zte-aussperren-a-860014.html
[8] http://info.arte.tv/de/huawei-der-spion-aus-china
[9] http://www.chip.de/news/Diese-Smartphones-spionieren-Sie-garantiert-aus-Android-Handys-mit-vorinstallierter-Spyware-entdeckt_82763375.html

[10] Y. Dodis, C. Ganesh, A. Golovnev, A. Juels, T. Ristenpart, A Formal Treatment of Backdoored Pseudorandom Generators, EUROCRYPT, Sofia, Bulgarien, 27.04.2015.
[11] M. G. Kuhn, R. J. Anderson, Soft Tempest - Hidden Data Transmission Using Electromagnetic Emanations, Information Hiding, LNCS 1525, Springer-Verlag, Berlin, 1998.
[12] G. T. Becker, F. Regazzoni, C. Paar, W. P. Burleson, Stealthy Dopant-Level Hardware Trojans, CHES, Santa Barbara, USA, 22.08.2013.
[13] R. Kumar, P. Jovanovice, W. P. Burleson, I. Polian, Parametric Trojans for Fault-Injection Attacks on Cryptographic Hardware, FDTC, Busan, Korea, 23.09.2014.
[14] A. L. Young, Building Robust Backdoors in Secret Symmetric Ciphers, BlackHat, Las Vegas, USA, 28.07.2005.
[15] A. Mishra, Cryptographic Backdoors: Subverting the RSA, COCON, Kochi, Indien, 22.08.2014.
[16] S. Bhasin, J. L. Danger, S. Guilley, T. Ngo, L. Sauvage, Hardware Trojan Horses in Cryptographic IP Cores, FDTC, Santa Barbara, USA, 29.08.2013.
[17] J. P. Aumasson, SHA1 Backdooring and Exploitation, BSIDES, Las Vegas, USA, 05.08.2014.
[18] D. Kern, Understanding and Implementing Encryption Backdoors, CSC7002 Project Paper, 31.03.2012.
[19] David A. Patterson and John L. Hennessy. 2013. Computer Organization and Design, Fifth Edition: The Hardware/Software Interface (5th ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
[20] https://danluu.com/cpu-backdoors/