Zabezpieczenia w systemie embdeded

Dziś nikogo już nie trzeba przekonywać, że zabezpieczenie systemu embedded odgrywa ważną rolę, ponieważ rozmaite skomputeryzowane urządzenia towarzyszą nam w różnych dziedzinach życia i nierzadko „posiadają" zapisane w pamięci wrażliwe dane, które niekoniecznie mamy ochotę ujawniać. Dodatkowe wyzwania przed konstruktorami i programistami stawia technologia IoT. Wiele mówi się o niej w kontekście ochrony prywatności, ponieważ z założenia ma nas opleść sieć czujników, które będą przesyłały pomiędzy sobą różne, dotyczące nas dane. Ważnym i może bardziej dotyczących konstruktorów zagadnieniem jest ochrona urządzeń przed kopiowaniem – wytwarzaniem tak zwanych podróbek. Jest to proceder niebezpieczny, ponieważ oprócz strat finansowych, prawdziwy producent traci rynek, cierpi na tym jego wizerunek. Traci też konsument, który czasami kupuje co prawda tańsze urządzenie, ale nierzadko kiepsko wykonane, niespełniające norm bezpieczeństwa, bez wsparcia technicznego. Powodów, dla których zabezpieczamy urządzenia embedded może być wiele – w artykule zajmiemy się nie motywacją, ale omówimy potencjalne zagrożenia, metody stosowane przez „piratów produktowych" i podamy wskazówki, ułatwiające wykonanie poprawnego zabezpieczenia. Co prawda, nie opiszemy gotowych rozwiązań, ale wskażemy kierunki poszukiwań i podamy ogólne zalecenia dotyczące ich opracowywania.

Tak zwane piractwo produktowe dotyczy nie tylko urządzeń elektronicznych i zajmuje się tworzeniem ich kopii lub „podróbek". Współcześni „piraci" podrabiają również komponenty elektroniczne, kopiują oprogramowanie, ekstrahują unikatowe rozwiązania techniczne, które następnie stosują we własnych urządzeniach. Ofiarami piratów produktowych podają głównie przedsiębiorstwa, których produkty są kopiowane. Tracą w ten sposób pieniądze, rynek, cierpi wspierana przez nie marka oraz wizerunek producenta.

Współczesne urządzenia elektroniczne mogą być budowane w taki sposób, że płytka drukowana zawiera niewiele komponentów lub wręcz gotowe moduły, natomiast główną wartość stanowi oprogramowanie. Nierzadko do jego utworzenia niezbędna jest

wiedza i sporo czasu – inwestycję liczy się w latach spędzonych na eksperymentowaniu, zdobywaniu niezbędnego doświadczenia, wykonaniu oprogramowania prototypu, a następnie testowaniu i wyszukiwaniu błędów. W takiej sytuacji nikogo raczej nie dziwi fakt, że przedsiębiorstwo chce odzyskać zainwestowane pieniądze, zanim ktoś wpadnie na podobny pomysł, zduplikuje rozwiązanie lub po prostu je skopiuje.

Nie ma zabezpieczeń doskonałych, nie do obejścia czy złamania. Przekonał się o tym znany producent komponentów elektronicznych, który wykonał (jego zdaniem) doskonale zabezpieczony mikrokontroler przeznaczony dla wojska. Pamięć programu i danych wykonano w postaci statycznego RAMu zasilanego za pomocą baterii, a każda próba otwarcia obudowy układu kończyła się odłączeniem zasilania i utratą całej zawartości. Ktoś wpadł na pomysł, że w bardzo niskiej temperaturze nośniki ładunku nie są już tak ruchliwe, zamroził układ, otwarł obudowę i… odczytał jego zawartość. Podobno w ten sam sposób można odczytać zawartość dynamicznej pamięci RAM – w niskiej temperaturze zachowa ona dane pomimo odłączenia zasilania. Zabezpieczenie może być jedynie na tyle dobre, aby inwestycja czasu i środków na jego obejście po prostu nie opłaciła się.

Dostępne narzędzia służące do zabezpieczenia produktów są tak różne, jak zabezpieczane produkty. Najbardziej oczywistym i najtrudniejszym do obejścia zabezpieczeniem są posiadane przez przedsiębiorstwo wiedza i doświadczenie składające się na tak zwany „know how" związany z produktem i jego aplikacjami. Ochrona wiedzy stanowi odrębne zagadnienie, które wykracza poza ramy artykułu, aczkolwiek zapewne jej część może być pozyskana dzięki analizie funkcjonalności wyrobu.

Efektywne zabezpieczenie przed piractwem wymaga nie tylko ochrony wiedzy, ale również zabezpieczenia powstałego na jej bazie systemu (urządzenia). Dlatego na każdym etapie projektowania urządzenia za podstawowy cel trzeba uznać nie tylko funkcjonalność, ale również ochronę własności intelektualnej. Należy przy tym brać pod uwagę powszechną już dziś dostępność nowoczesnych narzędzi umożliwiających kompleksową analizę systemu, możliwości tzw. inżynierii wstecznej oraz niepożądane manipulacje komponentami lub ingerencję w oprogramowanie. Aby zaplanować skuteczną obronę, należy…

… Rozważyć scenariusz ataku

Scenariusze ataku są zależne od rodzaju produktu i mogą znacznie różnić się. Co ważne, w urządzeniach można wyróżnić komponenty bardziej lub mniej istotne, a więc rozpatrując je w kategoriach bezpieczeństwa – takie, na których ochronie będzie nam szczególnie zależało lub nie będzie zależało w ogóle. Na przykład, głównym zagrożeniem dla producenta konsoli do gier nie jest to, że ktoś wykona klon konsoli, ale to, że manipulowanie sprzętem lub oprogramowaniem umożliwi użytkownikom, również tym niemającym odpowiedniej wiedzy technicznej, korzystanie z nielegalnego, tak zwanego piackiego oprogramowania. To samo dotyczy urządzeń służących do nawigacji satelitarnej – producenci chcą sprzedawać najnowsze mapy do swoich urządzeń i dlatego są zainteresowani nie tylko zabezpieczeniem oprogramowania, ale również dystrybuowanych map przed kopiowaniem. Z drugiej strony, czasami urządzenie do nawigacji pochodzi od innego producenta (może to być np. smartfon lub tablet z odbiornikiem GPS), więc nie są aż tak bardzo zainteresowani zabezpieczeniem przed wykonywaniem podróbek.

Inną ciekawą grupę urządzeń stanowią czytniki kart płatniczych. Te dla zapewnienia najwyższego poziomu bezpieczeństwa wymagają nie tylko zabezpieczenie sprzętu, ale również komunikacji wewnętrznej, zewnętrznych protokołów komunikacyjnych i oprogramowania. Trzeba też dać użytkownikowi minimalną możliwość kontroli czy wszystko odbywa się zgodnie z zasadami. Znane są jednak przypadki czytników, wewnątrz których zainstalowano dodatkowy hardware. Jego zadaniem było zarejestrowanie numeru PIN i numeru karty, a następnie przesłanie ich do oszusta na przykład – za pomocą komunikatu SMS. W ten sposób, zanim posiadacz konta miał szansę rozpoznać zagrożenie, zostawał okradziony. W takiej sytuacji bardzo trudno jest udowodnić bankowi, że to nie my jesteśmy winni.

Istnieje znaczna różnica pomiędzy opisywaną metodą, w której rezultat osiągnięto za pomocą dodatkowego, zainstalowanego wewnątrz czytnika trojana sprzętowego, a tak zwanym skimmingiem, w którym instaluje się odpowiednie urządzenie na zewnątrz.

Możliwość przeprowadzenia takiego ataku świadczy o kiepskim poziomie zabezpieczeń sprzętowych i programowych oraz o niskim poziomie dystrybucji oraz obsługi posprzedażnej. Mimo pozornego braku związku z poziomem zabezpieczeń, właściwy serwis oraz sieć dystrybucji pomagają w unikaniu zagrożeń tego typu i umożliwiają szybką reakcję na zagrożenia.

Istotnym czynnikiem, który powinien być brany pod uwagę przy rozważaniu niezbędnego poziomu zabezpieczeń jest czas życia produktu. Na przykład smartfon „uzbrojony" w przełomowe technologie traci atrakcyjność po około od 2 do 4 lat, gdy zastosowana w nim technologia staje się przestarzała lub na rynku pojawia się więcej podobnie wyposażonych smartfonów. Daje to znacznie krótsze ramy czasowe dla osób chcących wyprodukować podróbkę – po prostu muszą one rozważyć czy opłaca się uruchomić produkcję. Z drugiej strony, na rynku jest szereg różnych urządzeń, których okres użytkowania jest znacznie dłuższy – są to na przykład kosztowne, częstokroć używane latami, urządzenia przemysłowe.

Jest oczywiste, że osoby chcące wykonać kopię urządzenia na najwyższym poziomie posługują się ujęciem produktowym, a na najniższym rozpatrują komponenty lub wręcz – naruszenie ich integralności. Na czym polegają oba podejścia? Na najwyższym poziomie „pirat produktowy" przygląda się np. obudowie, złączom, analizuje, które z zastosowanych produktów są standardowe i które będzie można nabyć w normalnej sieci dystrybucji. Jednocześnie rozpatruje czy do uzyskania podobnego efektu będzie wymagana jakaś specjalistyczna wiedza i czy inwestycja związana z jej zdobyciem opłaci się. Na tym etapie może też zapaść decyzja czy będzie wykonana dokładna kopia, czy po prostu podobnie funkcjonujące, ale identycznie wyglądające (lub podobne) urządzenie. Od tej decyzji będą zależały dalsze kroki oraz to czy zostanie podjęta próba odczytania zawartości pamięci programu i innych, które są niezbędne do funkcjonowania urządzenia.

Zależnie od złożoności systemu mogą być zaplanowane specyficzne strategie związane z zastosowanymi komponentami. Urządzenie można podzielić na poszczególne bloki, a następnie – używając odpowiednich przyrządów pomiarowych oraz dokonując analiz – atakujący mogą zrozumieć sposób, w jaki działają nawet najbardziej złożone komponenty. Na tym etapie rozważa się też, które części można zastąpić standardowymi, a które będą wymagały dalszej, pogłębionej analizy. Innymi słowy, wysiłki mogą być skupione nie na całości urządzenia, ale na najbardziej złożonej jego części i – jeśli to potrzebne – dla tych potrzeb można wynająć odpowiednich specjalistów.

Takie zredukowane podejście pozwala piratom produktowym na działanie na kilka sposobów:

Skupienie i specjalizacja. Jedną z opcji jest użycie specjalizowanych metod i narzędzi, które umożliwiają zastosowanie metod inżynierii wstecznej do poszczególnych komponentów składowych. Dla przykładu, obudowa może być skopiowana za pomocą skanera 3D. Jednocześnie ten sam skaner może ją zwymiarować i utworzyć model cyfrowy niezbędny dla oprogramowania EDA do zaprojektowania odpowiedniej płytki

1) Michael Stell „17 Mistakes Microsoft Made in Xbox Security System", 22nd Chaos Communication Congress

2) Heise News „Visa: Regelmäßiges Wiegen der Kartenterminals schütz vor Manipulationen", wiadomość umieszczona na portalu dnia 09/07/2010 pod adresem *http://heise.de/-1035169*

drukowanej. Na poziomie oprogramowania za przykład może posłużyć analiza kodu binarnego (wynikowego) umieszczonego w pamięci, która wymaga specjalistycznej wiedzy, oprogramowania i sprzętu, zależnych od systemu mikroprocesorowego.

Zastępowanie i modyfikowanie. Jeśli jakieś komponenty okazują się odporne na metody stosowane przez inżynierię wsteczną, może być zastosowana alternatywna technologia lub typowy komponent o zbliżonej funkcjonalności. W rezultacie, sklonowany wyrób może być gorszy, ponieważ piraci produktowi zwykle zastępują nowoczesną technologię starszą, mającą mniejsze możliwości lub co gorsze – niemającą już wsparcia. Znamienne jest też to, że spadek funkcjonalności produktu jest trudno dostrzec na pierwszy rzut oka, ponieważ sklonowany wyrób może podobnie wyglądać i mieć zbliżoną jakość wykonania, a różnice są jedynie funkcjonalne. Jeszcze inna sytuacja może wystąpić w wypadku naśladownictwa i próby wykonania kopii funkcjonalnej – tu może zdarzyć się pełne odstępstwo od oryginału, które będzie wywierało wpływ nie tylko na funkcjonalność, ale również na bezpieczeństwo użytkowania.

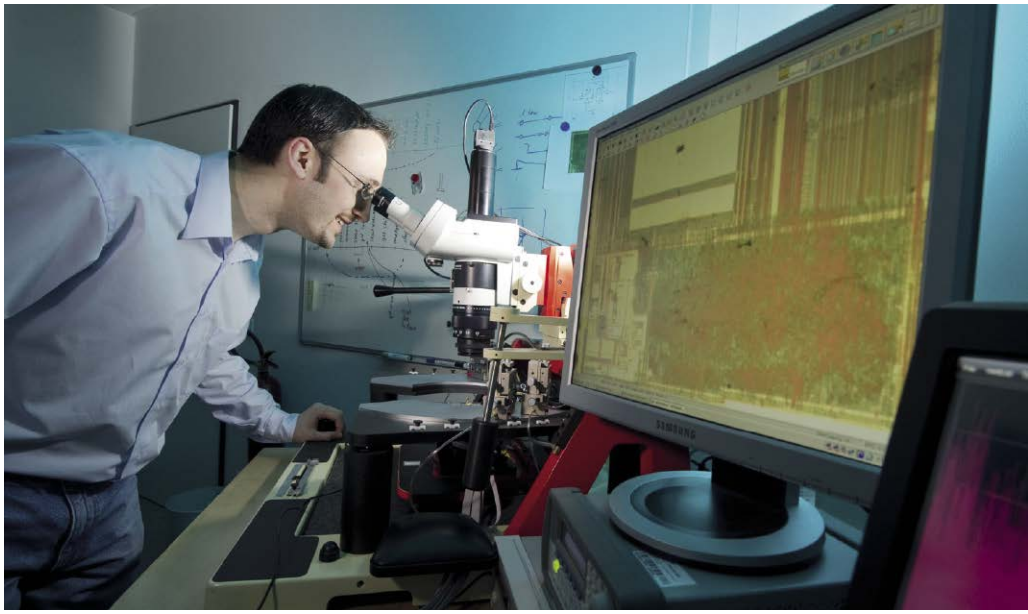
Jak łatwo domyślić się, wykonanie odpowiedniej kopii lub podróbki, pomimo najlepszych chęci i wiedzy projektanta, to czasami i przeważnie jedynie kwestia „czasu i pieniędzy". Czy wobec tego warto chronić się przed podróbkami i w jaki sposób? Aby skutecznie chronić swoje produkty, trzeba po pierwsze założyć jaki jest…

… Cel ochrony

Co oczywiste, pierwszym i zapewne podstawowym celem ochrony dla wielu przedsiębiorstw będzie zabezpieczenie dochodu związanego ze sprzedażą urządzenia oraz swojego udziału w rynku. Jednak – bazując na wspomnianym wcześniej przykładzie terminala do obsługi płatności – celem ochrony może też być zabezpieczenie przed nieautoryzowanym dostępem do różnych zasobów lub urządzeń. Celem może też być ochrona wizerunku producenta, który może stracić opinię solidnego, oferującego dobrze zabezpieczone i zaawansowane technologicznie produkty. Co paradoksalne, w określeniu celu ochrony mogą pomóc sami piraci – cennych wskazówek dostarczy analiza podróbek dostępnych w handlu. Trzeba jednak w tym miejscu wspomnieć, że zupełnie inny cel ochrony może być ustalony przez przedsiębiorstwo o ugruntowanej pozycji na rynku, a zupełnie inny przez firmę, która dopiero debiutuje. Zwykle doświadczenie podpowiada też, co może być głównym celem ataku i jak się przed nim zabezpieczyć.

Najczęściej używaną metodą kopiowania produktu jest drobiazgowa analiza komponentów. W takiej sytuacji wyrób jest rozkładany na najmniejsze „cegiełki" i jest przeprowadzana analiza zależności funkcjonalnych i technologicznych. Jednocześnie określa się, które komponenty są standardowe (dostępne w handlu), a które wykonywane na zamówienie. Zmywanie oznaczeń układów scalonych może tu utrudnić pracę, ale jedynie amatorom, niedysponującym odpowiednią wiedzą i sprzętem.

W niektórych sytuacjach może pomóc samodzielne analizowanie produktu. Można wykonać te same czynności, których spodziewamy się po „piratach": rozłożyć produkt na poszczególne komponenty, odrębnie poddać każdy z nich analizie określając ważność, stopień komplikacji i wymagany poziom zabezpieczenia. Przy analizowaniu takiego urządzenia istotne będzie też to, w jaki sposób komponenty są połączone ze sobą i jak wymieniają dane – czy protokół komunikacyjny jest jawny, czy szyfrowany, jak ważna jest komunikacja dla danego komponentu i czy mogą na tym skorzystać osoby próbując wykonać kopię lub dokonać włamania czy zmodyfikować urządzenie. Warto w tym miejscu wspomnieć, że te działania może za nas wykonać wynajęta firma, specjalizująca się w ocenie ryzyka kopiowania urządzeń.



Trzeba zwrócić uwagę na fakt, że takie ujęcie może doprowadzić do myślenia kategoriami „zabezpieczenia modularnego", co może być niebezpieczne. Wydaje nam się wtedy, że skoro poszczególne moduły urządzenia są zabezpieczone, to całość też musi być bezpieczna. Nic bardziej mylącego – zabezpieczenia poszczególnych modułów, jedno po drugim, jest stosunkowo łatwo usunąć lub zastąpić moduł podobnym, mającym zbliżoną lub tę samą funkcjonalność. Dlatego dobrze zabezpieczone urządzenie powinno stanowić całość, rodzaj współpracujących i współzależnych modułów, komunikujących się za pomocą bezpiecznego protokołu.

Znaczny wpływ na bezpieczeństwo produktu ma jego charakterystyka technologiczna. Co należy rozumieć pod tym pojęciem? Po pierwsze, jakich technologii użyto do jego wyprodukowania. Tu, co prawda, trudno zabezpieczyć się, ponieważ na rynku jest dostępnych wiele firm oferujących zaawansowane usługi montażu lub obróbki mechanicznej, ale czasami można wpaść na jakiś pomysł, który będzie trudny do skopiowania przez innych. Po drugie, jakich technologii – w rozumieniu komponentów elektronicznych – użyto w celu osiągniecia danej funkcjonalności. Czy są to technologie ogólnie dostępne i dobrze znane, czy nowoczesne, opracowane przez przedsiębiorstwo, będące dopiero w fazie upowszechniania? Jak łatwo domyślić się, starsze technologie zwykle są dostępne w handlu i mogą być używane przez każdego. Dlatego często myślenie kategoriami redukcji kosztów (np. zastosujmy coś, co jest tanie, dobrze znane i już od dawna dostępne w handlu) może być w dłuższej perspektywie zgubne dla przedsiębiorstwa i produktu. Dla podkreślenia wagi użytych technologii oraz komponentów, posłużę się przykładem związanym z analizowaniem kodu wynikowego w pamięci programu. Jeśli już podejmującemu atak uda się odczytać tę pamięć, to zależnie od zastosowanego mikroprocesora będzie miał dostępnych wiele narzędzi softwareowych i sprzętowych (w tym dla niektórych rdzeni całkowicie za darmo) lub tylko kilka, dostępnych od wybranych producentów i niekoniecznie za darmo, więc zastosowanie mniej popularnego mikrokontrolera w naturalny sposób ogranicza możliwości hakera.

Ważnym zagadnieniem jest również rozpoznanie czy trudno jest wyekstrahować

3) Andreas Henke, Mark Jansen „Vergleich Kathrein UFS 910 Original mit China Klone", opracowanie dostępne w formacie PDF pod adresem *http://goo.gl/IsptSp*

wiedzę z produktu. To znaczy, jak trudno jest pozyskać cały know how, który włożono w jego opracowanie, który może dotyczyć zarówno komponentów mechanicznych, jak i elektronicznych, być też związany z doświadczeniami firmy jako producenta i dystrybutora. Co gorsze, wzajemne zależności pomiędzy poszczególnymi częściami składowymi, pomimo jakby zupełnie innych, niezwiązanych ze sobą domen, mogą ujawniać szczegóły dotyczące funkcjonowania urządzenia lub jego oprogramowania. Dla zilustrowania omawianych zagadnień, posłużmy się przykładem…

… Klonowania odbiornika cyfrowego Kathrein UFS 910

Niemiecka firma Kathrein opublikowała ciekawy dokument opisujący klon odbiornika UFS 910 wykonany w 2008 r. W dokumencie opisano różnice pomiędzy kopią a oryginalnym odbiornikiem. Dokument jest świetną ilustracją sposobu, w jaki działają piraci produktowi.

Większość z konsumentów identyfikuje produkt po opakowaniu i obudowie. Dlatego też imitatorzy poświęcili im szczególną uwagę, aby nie wzbudzać niepokoju klientów już przy pierwszym kontakcie ze skopiowanym odbiornikiem. Obudowę kopii wykonano tak starannie, że różnice można było zauważyć przez porównanie stojących obok siebie wyrobów: oryginalnego i podróbki. Podobnie starannie wykonano opakowanie. Znamienne było to, że numer seryjny zamieszczony na tabliczce na urządzeniu nie odpowiadał naklejce na opakowaniu. Kopia była również nieco spóźniona, ponieważ urządzenie wyglądało tak, jak starsza wersja produktu. Inaczej (w porównaniu z oryginałem) rozmieszczono otwory wentylacyjne. Występowały również niewielkie różnice kolorów obudowy (oryginalna była mniej błyszcząca) i elementów manipulacyjnych (kanty, niestaranne wykończenie), jednak nie dało się tego zauważyć nie mając obok oryginału.

Zwykle – ze względu na kiepski proces technologiczny mający zastosowanie przy produkcji pierwszy kopii – łatwo było wykryć podróbki poprzez uważne przyjrzenie się odbiornikowi, ale po pewnym czasie ich wygląd poprawił się na tyle, że na pierwszy rzut oka, nie dokonując dokładniejszej analizy, bardzo trudno było rozpoznać czy ma się do czynienia z oryginałem, czy z kopią.

Podróbki korzystają dzięki dostępności narzędzi, takich jak kamery o dużej rozdzielczości, skanery i drukarki 3D oraz innym urządzeniom tego typu. Z tego powodu producentom jest bardzo trudno zabezpieczyć zewnętrzny wygląd produktu. W związku z tym uciekają się do metod znakowania obudów i ważniejszych komponentów za pomocą trudnych do podrobienia hologramów, etykiet odczytywanych elektronicznie, atramentów wytwarzanych przy użyciu technologii chronionej tajemnicą, a ostatnio nawet do znakowania za pomocą nanotechnologii (mikrocząstki, molekuły DNA, powierzchnie skanowane za pomocą lasera, znaki wodne możliwe do sprawdzenia jedynie za pomocą specjalnych czytników), znaczników RFID i innych metod. Te metody pozwalają co prawda na ustalenie czy mamy do czynienia z wyrobem oryginalnym, czy z podróbką, ale w żaden sposób nie zabezpieczają własności intelektualnej producenta. Wróćmy jednak do odbiornika HDTV firmy Kathrein.

Jeśli bliżej przyjrzymy się jego wnętrzu, można dostrzec pewną liczbę standardowych komponentów, którymi zastąpiono te używane w oryginalnym wyrobie. Są wśród nich złącza USB i CI (wyglądające identycznie, ale dostarczane przez inną firmę), złącze zasilania niezgodne z wymaganiami norm DIN-EN, inny moduł SCART, inny wyłącznik sieciowy. Oprócz nich zastosowano też inny wyświetlacz, od innego producenta, mający znaki o nieco innej wielkości.

Można zauważyć również kilka zmian wynikających z niewiedzy lub nieznajomości norm spełnianych przez oryginalny wyrób. Na przykład, pomiędzy zasilaczem a modułem RS232 w oryginalnym produkcie zastosowano przekładkę izolacyjna po to, aby finalny wyrób był zgodny z normami bezpieczeństwa obowiązującymi w Europie. Podobną przekładkę zastosowano też w podróbce, jednak została ona niepoprawnie zamontowana. W zasilaczu oryginalnego wyrobu zintegrowano obwód zapewniający minimalny pobór energii w trybie czuwania (*deep standby circuit*). Tego obwodu nie ma podróbce (zasilacz w ogóle wygląda przez to inaczej), co może nie wpływa na jej funkcjonalność, ale obniża jakość gotowego wyrobu. Na fotografiach rzucają się też w oczy kable połączeniowe – te oryginalne są w postaci taśm, te podrabiane są poprowadzone jako pojedyncze przewody. Co ciekawe, w niektórych napisach są literówki. Na przykład, oryginalne złącze interfejsu zewnętrznego jest „DATA I/O", natomiast na podróbce widnieje napis „DITA I/O".

Inaczej wygląda też nadajnik zdalnego sterowania – oryginalny jest o kilka milimetrów krótszy, ma nieco inne napisy (na podróbce są większe, mniej starannie nadrukowane czcionki), różni się też ozdobnymi detalami i nadrukami. Po otwarciu pokrywy baterii, w podróbce zobaczymy kontakty w postaci sprężyn z drutu, natomiast w oryginale są blaszki. To są jednak naprawdę drobne różnice i nie mając obok siebie oryginału i podróbki, trudno jednoznacznie osądzić czy wyrób jest oryginalny, czy nie.

Ochrona systemu embedded

Aby osiągnąć żądany poziom zabezpieczenia urządzenia embedded, powinno się odrębnie przeanalizować oprogramowanie i warstwę sprzętową. Z drugiej strony, obie te domeny muszą ze sobą współgrać, być skoordynowane, aby urządzenie mogło osiągnąć maksymalny poziom zabezpieczeń.

Powody, które skłaniają do zajęcia się inżynierią wsteczną mogą być różne, od zwykłej ciekawości jak coś jest zrobione lub ile można zyskać wprowadzając na rynek dany produkt (można w ten sposób oszacować koszt produkcji i odnieść go do ceny rynkowej), do chęci kradzieży własności intelektualnej. Zdobyta w ten sposób wiedza może być wykorzystana w procesie opracowywania własnych produktów i/lub porównania ich z konkurencją.

Jeśli firma chce rozszerzyć ofertę swoich produktów inżyniera wsteczna w odniesieniu do dostępnych na rynku wyrobach konkurencyjnych jest częstokroć łatwiejsza i tańszą metodą pozyskiwania wiedzy i informacji, niż inwestowanie czasu i pieniędzy w prace badawcze oraz opracowywanie nowych produktów i technologii. Z prawnego punktu widzenia, inżynieria wsteczne leży w „szarej strefie", ponieważ z jednej strony nikt nie zabrania zajrzenia do wnętrza właśnie nabytego przez nas urządzenia i zapoznania się z tym, jak działa, ale z drugiej dobrze wiemy, że jeśli chcielibyśmy wykorzystać zdobytą wiedzę w celach komercyjnych, to taka działalność jest niezgodna z prawem i może naruszać prawa patentowe. Niemniej jednak istnieją przedsiębiorstwa, takie jak Canadian Chipworks, które specjalizują się w inżynierii wstecznej w celu zapewnienia swoim usługobiorcom ochrony patentowej. W publikacji z 2009 r. firma wymieniła procedury, które są przez nią używane do analizowania układów scalonych:

- Rozpoznanie produktu – identyfikacja układu scalonego, jego obudowy, płytek wewnętrznych i komponentów.
- Poziom analizy systemowej – analiza funkcjonalności, ścieżek sygnałowych i zależności czasowych.
- Poziom analizy procesowej – rozpoznanie zastosowanej technologii.
- Ekstrakcja obwodów – rekonstrukcja obwodów układu scalonego.

Takie same lub bardzo podobne czynności są zapewne używane przez piratów produktowych i dlatego warto się z nimi zapoznać, aby zrozumieć jak działają piraci i opracować dzięki temu strategię zabezpieczenia własnych wyrobów.

Rozpoznanie produktu. Rozpoznanie produktu lub inaczej – identyfikacja jego komponentów, jest na szczycie procesu inżynierii wstecznej. Interesujące są wszystkie

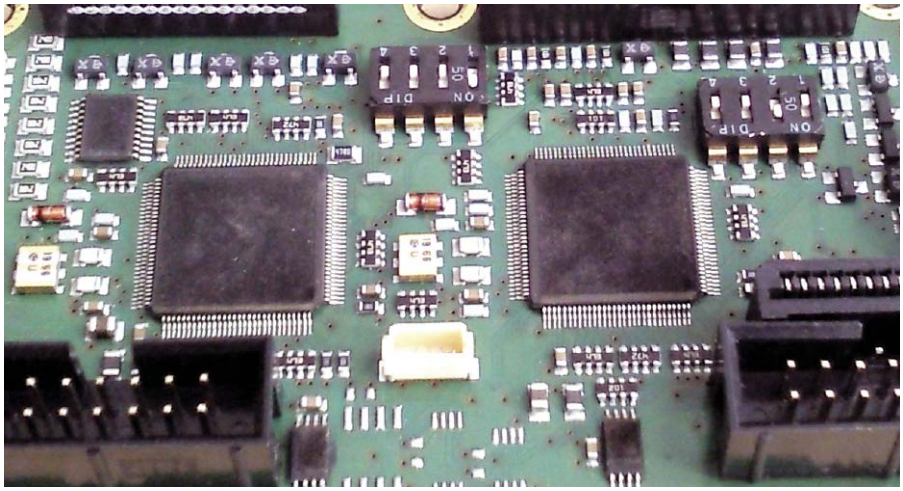
4) *http://goo.gl/SBkRNx*

elementy obwodu, ponieważ dzięki nim można nie tylko określić parametry techniczne, ale również oszacować koszt produkcji. Płytka urządzenia jest fotografowana, a komponenty są identyfikowane na podstawie obudowy, umieszczonych na niej napisów i sposobu połączenia (np. linii zasilających).

Trudno zabezpieczyć urządzenie przed analizą na tym poziomie. Można utrudnić zadanie atakującym np. zmieniając napisy na obudowach komponentów, usunąć je (za pomocą lasera, rozpuszczalnika lub wręcz ścierając) lub zamawiając komponenty bez nadruku (**fotografia 1**). Strukturę lub obudowę układu scalonego można zalać żywicą epoksydową lub twardym tworzywem sztucznym. Dodajmy jednak (o czym już wspomniano), że taka metoda może utrudnić rozpoznanie urządzenia amatorowi, ale nie jest przeszkodą dla fachowca, który ma do dyspozycji odpowiedni sprzęt (aparat rentgenowski, mikroskop, sprzęt do precyzyjnego szlifowania). Obudowę układu scalonego można rozpuścić, a na jego strukturze jakże często producent umieszcza napisy i symbole, które uławiają atakującemu rozpoznanie układu i wybranie odpowiednich metod oraz narzędzi. Przykład – fotografie struktur mikrokontrolerów – pokazano na **fotografii 2**.

Poziom analizy systemowej. Kolejnym krokiem jest analiza systemowa polegająca na rozpoznaniu, w jaki sposób komponent „współgra" z otoczeniem. Na przykład, większość procesorów komunikujących się za pomocą interfejsu asynchronicznego wymaga użycia rezonatora kwarcowego w celu zapewnienia stałości częstotliwości taktowania interfejsów komunikacyjnych. Oczywiście jest to wskazówka, a nie reguła. A więc na początek „wyciąga się" wszystkie informacje, które mogą być zauważone lub zmierzone za pomocą przyrządów. W tym celu używa się nie tylko sygnałów dostępnych w systemie, ale również dołącza sondy w pewne charakterystyczne punkty na płytce, wymusza sekwencje sygnałów (zarówno spodziewanych w systemie, jak i specjalnych, normalnie niewystępujących w trakcie eksploatacji) i testuje odpowiedzi za pomocą analizatora stanów logicznych, oscyloskopu i innych przyrządów. Ta część nazywa się analizą funkcjonalności. Jako kolejną przeprowadza się analizę systemu.

Istnieje ogromna różnica pomiędzy analizą funkcjonalności, a analizą systemu. Przystępując do analizy systemu wykonuje się dokumentację fotograficzną urządzenia „przed" oraz opisuje rozpoznane komponenty i ich połączenia. Następnie, płytka drukowana jest dosłownie rozrywana na pojedyncze warstwy, a każda z tych warstw jest skanowana, fotografowana i przenoszona do programu, który umożliwia utworzenie mapy połączeń na płytce drukowanej i w konsekwencji – narysowanie schematu



Fotografia 1. Usunięcie napisów z obudowy układu scalonego może znacznie utrudnić jego identyfikację

urządzenia. Co oczywiste, im więcej warstw ma płytka drukowana, tym trudniejsza jest jej analiza.

Analiza systemu może być użyta nie tylko do określenia funkcjonalności urządzenia lub układu scalonego, ale również funkcjonalności programu uruchomionego na mikrokontrolerze lub układzie FPGA. Występuje przy tym podobna zależność, jak przy analizowaniu płytki drukowanej – im bardziej złożony system, im więcej komponentów ma wewnątrz układ scalony i im bardziej złożone oprogramowanie, tym ta analiza jest trudniejsza. Ale… Będzie o tym mowa dalej.

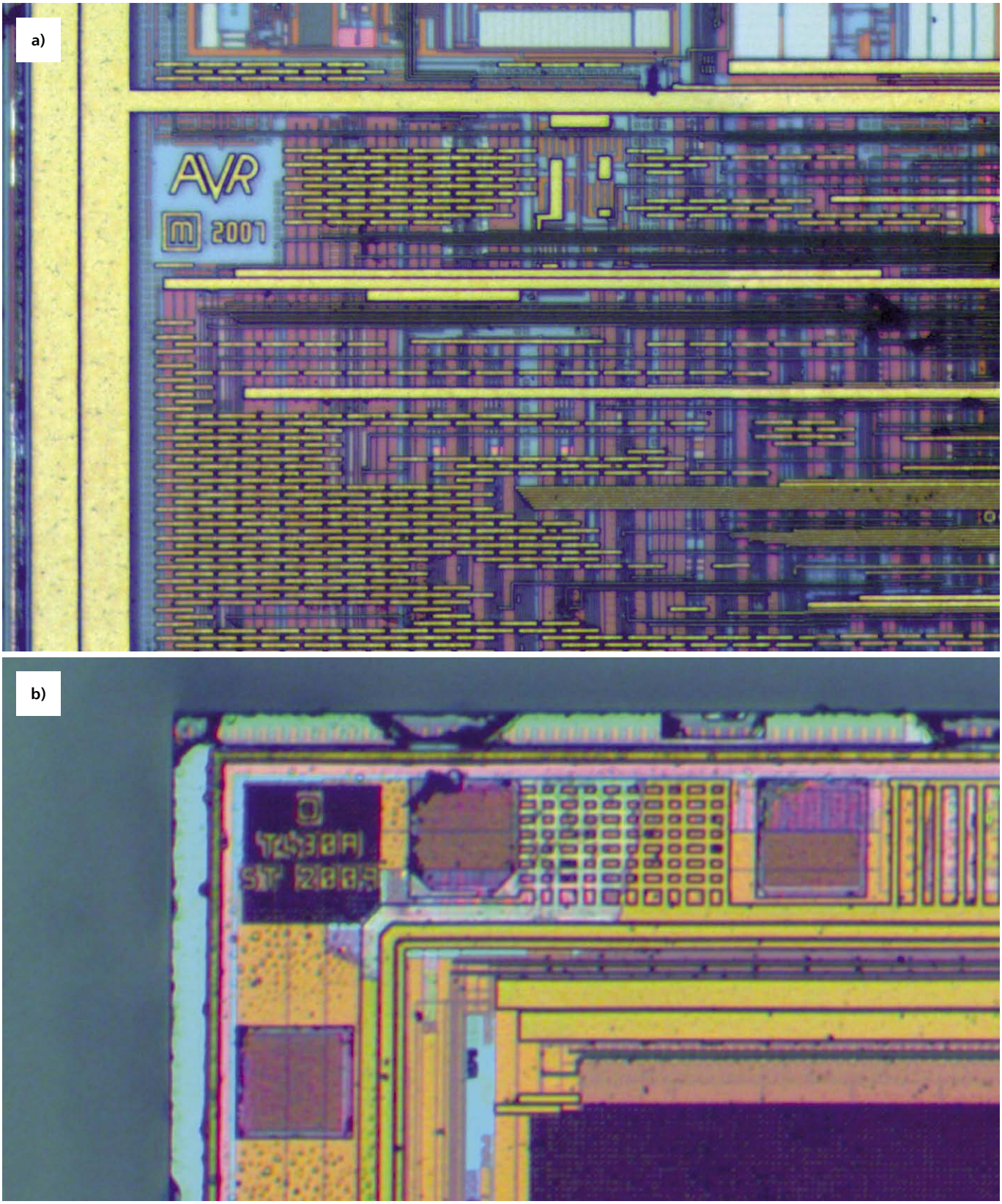
Słabym punktem większości systemów embedded są pamięci. Przechowują one dane, oprogramowanie i konfigurację FPGA. Konstruktor urządzenia i programista muszą tym układom poświęcić szczególną uwagę, a zwłaszcza przechowywanym przez nie tzw. wrażliwym danym, np. kluczom szyfrowania.

Układy pamięci mogą być czytane za pomocą wielu urządzeń – programatorów, debugerów, interfejsu JTAG. Podobnie współpracujące z nimi procesory lub mikrokontrolery. Budując system embedded przechowujący wrażliwe dane nie wolno o tym zapominać, że osoba atakująca może szukać jakiejś ścieżki obejścia, odczytać klucz szyfrowania za pomocą rejestru procesora lub w inny sposób. Stosowane powszechnie zabezpieczenia w postaci programowanych bitów bezpieczników są niewystarczające i podatne na ataki: bity w pamięci Flash można ustawić za pomocą lasera (**fotografia 3**), niektóre pamięci nie są odporne na szybkozmienne napięcie występujące na liniach zasilania, natomiast przepalane ścieżki lub druty można odnaleźć za pomocą mikroskopu i mostkować z użyciem mikrosond (**fotografia 4**). Stosowane są też inne metody, na przykład, w razie potrzeby za pomocą wiązki jonów można w strukturze układu scalonego wykonać dodatkowe tranzystory, które pozwolą na ominięcie zabezpieczeń. Owszem, wymaga to odpowiedniego oprzyrządowania i wiedzy, ale aparaturę można po prostu wynająć, a fachowców – zatrudnić lub zlecić im odpowiednie prace.

Producenci układów scalonych przeznaczonych do zastosowań specjalnych zabezpieczają strukturę przed tego rodzaju atakami umieszczając wokoło niej siatkę z drutu i/lub membranę, które są elektrycznie połączone ze strukturą układu. Przy próbie otwarcia obudowy następuje uszkodzenie siatki lub membrany, co skutkuje zniszczeniem całej funkcjonalności układu i elementów jego struktury.

Ze względu na rosnącą liczbę zastosowań układów FPGA, również ich producenci muszą nierzadko mierzyć się problemem zabezpieczenia przez kopiowaniem. Współcześnie są używane głównie dwa typy układów FPGA: przechowujące konfigurację w pamięci SRAM najczęściej ładowanej z zewnętrznej pamięci nieulotnej po załączeniu zasilania (chociaż są też już dostępne układy z wbudowaną nieulotną pamięcią konfiguracji), oraz FPGA, które mają konfigurację zapisaną na stałe w procesie produkcji. Jak zapewne czujemy intuicyjnie, jeśli jest stosowana zewnętrzna pamięć przechowująca konfigurację, to „rozmowa" pomiędzy nią a FPGA może być łatwo zarejestrowana i poddana analizie. Wytwórcy FPGA polegają na zabezpieczeniu w postaci ciągu bitowego, charakterystycznego dla danego układu, który co prawda zabezpiecza transmisję, ale nie układ FPGA przed sklonowaniem, ponieważ do interfejsu, za pomocą którego jest odczytywana konfiguracja można dołączyć równolegle taki sam układ i wczytać konfigurację również do niego. Z tego powodu często klucz szyfrowania jest dzielony na dwie części, z których jedna jest przechowywana przez pamięć, a druga jest zapisana w FPGA. Do poprawnej transmisji FPGA musi „znać" obie części klucza.

Pomimo stosowanych zabezpieczeń, przechowywanie klucza w sposób



Fotografia 2. Zdjęcie struktur mikrokontrolerów z widocznym logo producenta i oznaczeniem rodziny: a) AVR (ATtiny13), b) STM32F103VGT6

uniemożliwiający jego odtworzenie przez atakującego nie jest zadaniem łatwym. Istnieją metody, które pozwalają na obliczenie klucza na podstawie zjawisk fizycznych (np. skoków poboru prądu, interwałów czasowych pomiędzy realizacją poszczególnych poleceń i inne) występujących w czasie obliczania klucza. Układ „niezabezpieczony kryptograficznie" może wymagać mniej lub więcej czasu na odszyfrowanie wiadomości zależnie od – na przykład – jej długości i używanego klucza szyfrowania. Wynika to z prostego faktu mniejszej lub większej złożoności obliczeniowej (mniejsza lub większa liczba dostępu do pamięci, częściej lub rzadziej wywoływane instrukcje itd.). Jeśli wiadomość, na przykład – zaszyfrowana zawartość pamięci konfiguracyjnej FPGA, jest znana, to klucz szyfrowania może być rozpoznany za pomocą obliczeń statystycznych. W ten sam sposób, klucz może być obliczony z użyciem wahań poboru mocy zasilania lub zmian promieniowania elektromagnetycznego układu.

NOTATNIK KONSTRUKTORA

Większość współczesnych układów scalonych jest wykonywanych w technologii CMOS. Jak wiadomo, tranzystor w takim układzie pobiera energię jedynie w czasie zmiany stanu. Aby zdekodować sygnał, pewna liczba tranzystorów musi zmienić stan. Przełączenia (różnice poboru mocy lub promieniowania elektromagnetycznego) mogą być odczytane i użyte do rozpoznania klucza. Oprócz tego układ kryptograficzny może być „zmuszany" (za pomocą promieniowania podczerwonego, nagłych zmian napięcia zasilającego lub częstotliwości zegarowej i w inny sposób) do popełniania błędów. Następnie klucz jest wyznaczany dzięki porównaniu poprawnych i błędnych danych wyjściowych.

Producenci jednostek kryptograficznych w układach scalonych bronią dane utrudniając zadanie hakerom poprzez losowe wywoływanie funkcji niemających wpływu na przetwarzane dane, maskowanie danych, instalowanie dodatkowego układu kryptograficznego, który wykonuje te same operacje, ale jakby „w przeciwnej fazie", dzięki czemu nie ma skoków poboru mocy lub wbudowując źródła sygnału zakłócającego.

Mimo stosowanych zabezpieczeń, ocena ryzyka jest bardzo trudna, ponieważ współczesne urządzenia elektroniczne są wręcz wypełnione standardowymi układami, z którymi dzięki temu można eksperymentować do woli, odkryć słabe punkty i sprawić, by przemówiły ujawniając przechowywane tajemnice.

Poziom analizy procesowej. Na tym poziomie są pozyskiwane informacje na temat procesu produkcji, zastosowanego materiału i struktury układu scalonego. Narzędzia niezbędne do tej analizy są niemal powszechnie dostępne, ponieważ są stosowane przez producentów układów półprzewodnikowych do kontroli poprawności procesu produkcyjnego oraz analizy błędów. Pierwszym krokiem jest usunięcie obudowy układu za pomocą oparów kwasu lub – w wypadku obudowy ceramicznej – poprzez szlifowanie. Po odsłonięciu struktury układu scalonego analizuje się ją za pomocą różnych mikroskopów oraz odczynników chemicznych (zmiany materiału, domieszki, wielkość struktury, liczba warstw, domieszki itp.).

Ekstrakcja obwodów. Niegdyś słyszałem od przedstawiciela pewnej znanej firmy, nieistniejącego już pod dawną marką producenta półprzewodników, że użytkują swego rodzaju relikt „zimnej wojny" – precyzyjną szlifierkę (chociaż przy grubości zbieranej warstwy raczej należałoby tę maszynę nazwać polerką), która jest w stanie z wielka precyzją szlifować strukturę układu scalonego. Po zdjęciu ultracienkiej warstwy wykonuje się zdjęcie – wiele tych fotografii wykonywanych warstwa po warstwie służy do stworzenia trójwymiarowego obrazu struktury układu scalonego, który to jest analizowany. To było ponad 20 lat temu, a współcześnie?

Fotografia 3. Próba manipulowania zawartością pamięci Flash za pomocą lasera (źródło: materiały firmy Fraunhofer)

Okazuje się, że tę samą metodę stosuje się i dziś. Po zeszlifowaniu warstwy fotografuje się i identyfikuje elementy obwodu: cewki, rezystory, tranzystory, diody, przelotki, połączenia lub ich ewentualny brak itd. Następnie szlifuje się kolejną warstwę i powtarza cały proces aż do samego podłoża układu scalonego. Ten przedstawiciel wspominał również o tym, że używa się kilku struktur badanego układu, które są szlifowane w taki sposób, aby zapewnić różne przekroje, które są następnie ze sobą porównywane – zapewne tak też postępuje się współcześnie.

Analiza połączeń w obrębie struktury współczesnego układu scalonego, zwłaszcza wielkiej skali integracji, musi być bardzo trudna. Myślę też, że próbując rozpoznać poszczególne miniaturowe komponenty oraz ich połączenia popełnia się sporo błędów, więc analiza bardziej będzie ograniczała się do bloków funkcjonalnych, niż do tworzących je tranzystorów i innych elementów. Jest to pewnie tym łatwiejsze, że na przykład raz zaprojektowany i sprawdzony blok interfejsów jest przez producenta układu scalonego stosowany w różnych wyrobach.

Rozpoznawanie funkcjonalności oprogramowania

Dostępność zawartości pamięci programu lub zawartości pamięci konfiguracji układu FPGA w dużej mierze ułatwia atakującemu pozyskanie wiedzy niezbędnej do wykonania kopii urządzenia. Współczesne oprogramowanie disasemblujące zawartość pamięci (nierzadko dostępne za darmo) jest na tyle doskonałe, że poradzi sobie z większością sztuczek stosowanych przez programistów. Ponadto, często kompilatory języków wysokiego poziomu odciskają w kodzie swój ślad, co dodatkowo ułatwia osobie atakującej rozpoznawanie pewnych standardowo stosowanych bloków asemblera, będących w istocie odzwierciedleniem funkcji wysokiego poziomu. Języki programowania mają też pewne charakterystyczne mechanizmy służące do przekazywania argumentów i pobierania rezultatów przetwarzania przez funkcje, którymi też może posłużyć się atakujący w celu odtworzenia programu źródłowego.

Pamiętajmy jednak, że to nie są jedyne cele hakerów. Czasami odtworzenie programu źródłowego jest zupełnie niepotrzebne. Wystarczy – jak w terminalu płatniczym – znaleźć „dziurę", która umożliwi przechwycenie komunikacji, odczytanie istotnych parametrów lub manipulowanie przesyłaną informacją.

Dostęp do zawartości pamięci umożliwia jej replikowanie w identycznym układzie, nawet bez konieczności rozpoznawania, jak działa program. Zabezpieczeniem przed

Transformacja obfuskcyjna (obfuscation)

Zaciemnianie kodu to technika przekształcania programów, która zachowuje ich semantykę, ale znacząco utrudnia zrozumienie. Istnieją narzędzia modyfikujące kod źródłowy, pośredni bądź binarny w celu utrudnienia inżynierii wstecznej programu. Wyróżniamy 3 typy transformacji obfuskacyjnych:

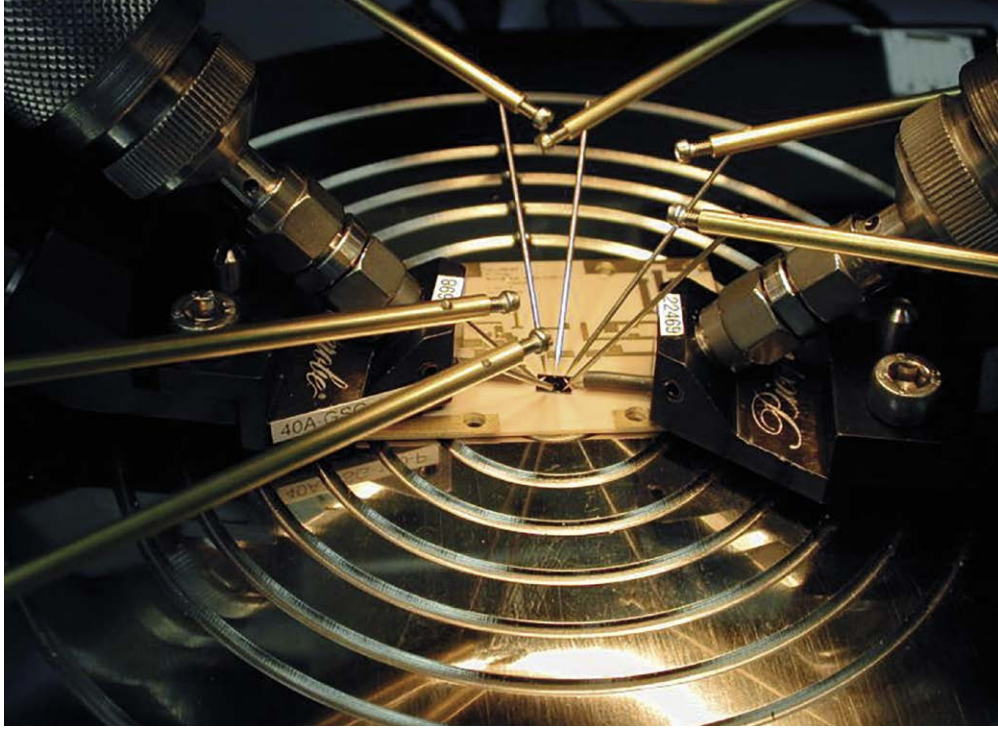
- Transformacja wyglądu (*Layout Transformation*) – zmiany nazw identyfikatorów, zmiana formatowania, usuwanie komentarzy.
- Transformacja danych (*Data Transformation*) – rozdzielenie zmiennych, konwersja statycznych danych do procedury, zmiana kodowania, zmiana długości życia zmiennej, łączenie zmiennych skalarnych, zmiana relacji dziedziczenia, rozłam/łączenie tablic, zmiana porządku instancji zmiennych/metod/tablic.
- Transformacja kontroli (*Control Transformation*) – zmiana przebiegu, rozszerzenie warunków pętli, zmiana kolejności komend/pętli/wyrażeń, metody inline, ogólnikowe wyrażenia, klonowanie metod.

takim przenoszeniem oprogramowania jest sprawdzanie numeru seryjnego lub innego „znaku wodnego” systemu czy układu, na którym jest uruchomiony program, ale jest to raczej uciążliwe, trudne do implementacji i raczej rzadko stosowane.

Wydaje się, że nie ma dobrej metody zabezpieczenia oprogramowania przed atakiem. Dobre zabezpieczenie przed „szpiegowaniem” i inżynierią wsteczną wymaga współpracy wsparcia programu przez hardware. Jeśli jest dostępna kompletna zawartość pamięci, to każdy aspekt zabezpieczenia softwareowego może być przeanalizowany i usunięty. W takiej sytuacji nic nie da zaangażowanie sprzętowego modułu kryptograficznego, ponieważ klucz szyfrowania jest zapamiętany w programie i może być łatwo podejrzany i/lub usunięty. Oprogramowanie będące nota bene integralnym składnikiem systemu jest podatne na modyfikowanie, może być emulowane i monitorowane również w kontrolowanych warunkach. Można je dla przykładu uruchomić na symulatorze (maszynie wirtualnej) bez rzeczywistego otoczenia sprzętowego.

Aby właściwie zabezpieczyć program i zapewnić szyfrowane połączenia w obrębie systemu, jednostka kryptograficzna nie może mieć klucza dostarczanego przez oprogramowanie, ale musi on być zapisany na stałe np. w pamięci układu kryptograficznego. Dzięki temu klucz nie da się odczytać z kodu binarnego programu ani poprzez analizę sposobu jego działania, ani metodami statystycznymi czy za pomocą emulowania pracy systemu w maszynie wirtualnej. W takiej sytuacji sprzęt i oprogramowanie tworzą jednostkę funkcjonalną i nie jest możliwe analizowanie zaszyfrowanych komponentów programu w jakikolwiek sposób bez właściwego otoczenia sprzętowego.

Trzeba przy tym zauważyć, że zabezpieczenie sprzętowe nie jest skutecznie ot tak, samo z siebie. Za przykład może nam posłużyć klucz sprzętowy dołączany do gniazda urządzenia, na przykład do interfejsu USB komputera PC. Oprogramowanie sprawdza czy klucz jest dołączony i jeśli tak, to pozwala na dostęp i użytkowanie. Jeśli oprogramowanie kontrolne jest źle napisane i można w nim łatwo zlokalizować blok odpowiedzialny za sprawdzenie zabezpieczenia oraz odnaleźć występujący na końcu testu rozkaz asemblerowy „skocz, jeśli zero”, to prosta zmiana tego polecenia na „skocz, jeśli różne od zera” spowoduje, że urządzenie w ogóle nie będzie kontrolowało obecności klucza sprzętowego i przyzna każdemu prawo dostępu. Co prawda, współczesne zabezpieczenia są dostarczane z odpowiednimi bibliotekami programowymi, które niełatwo oszukać, ale gdybyśmy coś „kombinowali” na własną rękę, to może zdarzyć się taka sytuacja.



Fotografia 4. Mikrosondy dołączone do struktury układu scalonego

Łatwo zauważyć, że stosując klucz sprzętowy trzeba zabezpieczyć program przed możliwością manipulacji, aby jego użycie miało sens. Jest to trudne do wykonania np. pod kontrolą systemu operacyjnego komputera PC ze względu na dostępność różnych narzędzi oraz bardzo dobrze udokumentowane środowisko pracy i nieco łatwiejsze w systemie embedded, zwłaszcza, jeśli komponenty pochodzą od jednego producenta, z którym można ustalić pewne cechy ważne z punktu widzenia funkcjonowania zabezpieczeń. Mogą to być na przykład specjalne kody zapisane na stałe w pamięci w procesie wytwarzania układu oferowanego tylko i wyłącznie nam.

Kompilatory tworzące kod dla debugera bardzo często umieszczają w pamięci procesora kompletną tablicę symboli, jeśli podczas kompilowania nie wyłączyło się opcji „debug”. Autorowi artykułu z autopsji znana jest sytuacja, w której osoba atakująca usunęła zabezpieczenie przed odczytem pamięci i uzyskała dostęp do pamięci programu zapisane właśnie takim kodem. Mówiąc krótko – atakującemu podano na tacy dostęp do wszystkich zmiennych i symboli stosowanych w programie. W takiej sytuacji zrozumienie działania programu było tylko formalnością…

Oprogramowanie również podlega ochronie prawnej i jest własnością intelektualną twórcy lub podmiotu, dla którego zostało wykonane. Nierzadko oprogramowanie stanowi główną wartość urządzenia, a na jego wykonanie jest wymagane sporo czasu, wiedza i pieniądze, i z tego punktu widzenia jest ono warte ochrony. Głównym celem ochrony oprogramowania powinno być uniemożliwienie atakującemu zapoznania się z przeznaczeniem i funkcjonalnością poszczególnych komponentów programowych i funkcji. W tym celu stosuje się technikę „zaciemniania kodu” (obfuscation – patrz wyjaśnienie w ramce).

Technika obfuskacji jest stosowana przede wszystkim przy programowaniu w językach wysokiego poziomu, takich jak Java czy C, które nie tyle tworzą natywny kod binarny, ile mapują kod źródłowy na pewnym kodzie pośrednim pełniącym rolę pomostu pomiędzy asemblerem, a tekstem programu. Jak już wspomniano, funkcje języka są reprezentowane przez pewne stałe bloki i łatwo rozpoznać czy mamy do czynienia z operacjami matematycznymi, pętlami, czy z innymi operacjami. W czasach, gdy pisałem dużo programów dla mikrokontrolerów z użyciem jednego kompilatora, potrafiłem bez trudu rozpoznać gdzie w wygenerowanym kodzie źródłowym są pętle, gdzie są wywoływane funkcji itd. Myślę, że i atakującym nie sprawia to specjalnej różnicy, stąd pomysł na „zaciemnianie” kodu.

Niestety, transformacja obfuskacyjna może powodować powstawanie nowych błędów, które mogą przemknąć niezauważone aż do konsumenta powodując dodatkowe koszty związane z usunięciem usterki oraz poważny uszczerbek na wizerunku. Może ona też powodować wzrost wymagań odnośnie do pamięci, szybkości procesora oraz utrudnić aktualizację oprogramowania i usuwanie błędów.

Do zabezpieczania oprogramowania jest również stosowana również kompresja i szyfrowanie danych zawartych w pamięci (niektóre z metod szyfrowania/kompresji są unikatowe i stosowane wyłącznie przez konkretnego producenta).

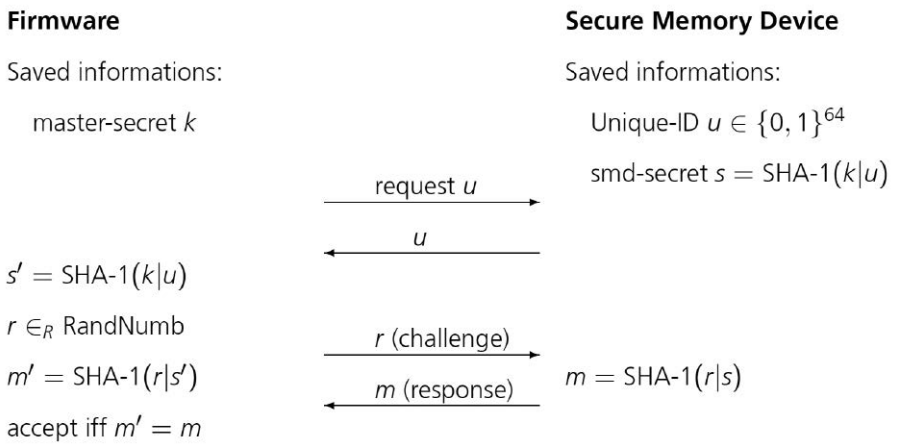
Aczkolwiek w świetle omówionych metod poziom zabezpieczenia oprogramowania może być bardzo wysoki, to w konsekwencji użyte metody mogą spowodować spadek ogólnej wydajności urządzenia, wzrost wymagań sprzętowych oraz dodatkowego know

how związanego z implementowaniem zabezpieczeń (zarządzanie kluczami szyfrowania, wybór optymalnej metody, aspekty implementacji zabezpieczeń w zmiennym środowisku sprzętowym, zabezpieczenie przed atakiem metodą pośrednią). Osobie niemającej doświadczenia trudno jest nawet ocenić, ile dodatkowego wysiłku będzie wymagała odpowiednia implementacja zabezpieczeń.

Łączenie zabezpieczenia sprzętowego i programowego

Przykładem zabezpieczenia sprzętowo-programowego chroniącego oprogramowanie przez nielegalnym kopiowaniem lub identyfikującym uprawnienia użytkownika w systemie jest klucz sprzętowy, tzw. dongle. Te klucze są sprzedawane z odpowiednimi bibliotekami programowymi, które można zintegrować z utworzonym przez siebie oprogramowaniem. Klucze szyfrowania zapisane w pamięci klucza sprzętowego są chronione zarówno przed odczytem z zewnątrz, próbami „podsłuchu", jak i atakiem za pomocą oprogramowania. Nowoczesny klucz sprzętowy ma wielkość pendrive, jest dołączany do USB i ma wbudowaną pamięć zawierającą odpowiednie drivery oraz pozwalającą na zintegrowanie oprogramowania kryptograficznego bezpośrednio w kluczu. Znacznie utrudnia to atakującemu zlokalizowanie odpowiednich funkcji obsługi oraz przesyłanie kwerend do klucza. Do testowania obecności klucza sprzętowego może być używany specjalny mechanizm uwierzytelniający – przykładowy, zaczerpnięty z materiałów firmy Fraunhofer, pokazano na **rysunku 5**. Oprócz tego, mechanizmy kryptograficzne zawarte w kluczu mogą być używane do kontroli integralności oprogramowania i zabezpieczenia go w ten sposób przed manipulacją. Współczesne, nowoczesne klucze sprzętowe są trudne do oszukania i oferują dobry poziom zabezpieczenia. Szyfrowanie zabezpiecza też oprogramowanie przed możliwością podglądu kodu, więc atakujący zanim będzie miał możliwość zajrzenia do kodu programu, musi je odszyfrować.

Klucze sprzętowe dobrze nadają się do zabezpieczenia komputerów PC, ale w związku z miniaturyzacją sprzętu coraz więcej z popularnych pecetów ma wymiary pudełka papierosów lub mniejsze i dlatego



Rysunek 5. Sposób działania specjalnego mechanizmu uwierzytelniającego (źródło: materiały firmy Fraunhofer)

te klucze mogą być używane również w systemach embedded.

Pewne układy elektroniczne mają funkcję *secure memory device*. W dużym uproszczeniu taka pamięć działa podobnie jak dongle, ale zamiast dołączenia jej do zewnętrznego interfejsu, można ją zintegrować jako moduł na etapie projektowania urządzenia. Wspólnie z pamięcią wewnętrzną, która może przechowywać indywidualne parametry lub mieć unikalny identyfikator, te moduły mają użyteczne, podstawowe funkcje kryptograficzne, które mogą znaleźć zastosowanie do ochrony systemu – zarówno sprzętu, jak i oprogramowania.

Głowni producenci programowanych układów logicznych, takich jak FPGA, oferują mechanizm ochrony dla poszczególnych serii produktów (*design security*). Niektóre z tych mechanizmów używają dodatkowej pamięci *secure memory*, a pewna liczba tych układów jest wyposażana w jednostki kryptograficzne zintegrowane wewnątrz ich struktury.

Metodą zabezpieczenia własności intelektualnej stosowaną przez największych producentów sprzętu jest zastosowanie specjalizowanych układów ASIC. Z drugiej strony, taki układ również jest podatny na opisane wcześniej metody, może być otwarty, przeanalizowany i zastąpiony odpowiednikiem funkcjonalnym albo blokiem realizującym jego funkcje. Mało tego, w pewnych okolicznościach ASIC może być łatwiejszy do przeanalizowania niż zwykły układ FPGA, który przechowuje konfigurację w pamięci zewnętrznej i jest dzięki temu odporny na obserwacje i analizowanie za pomocą mikroskopu. Nie ma też znaczenia czy konfiguracja jest załadowana, czy też nie, ponieważ układ i tak będzie wyglądał tak samo.

Dodatkowo do opisanych wcześniej modułów bezpiecznych pamięci, wytwarzane są również mikrokontrolery, które mogą być używane jako moduły do implementowania zabezpieczeń. Stosując jednak zabezpieczenia tego typu trzeba dobrze rozważyć, kiedy są one po prostu sztuką dla sztuki i czy urządzenie – pomimo zastosowania zaawansowanej kryptografii – nie jest podatne na jakiś drobiazg wynikający chociażby z błędu użytkownika.

Na koniec

Wyścig pomiędzy piratami produktowymi a producentami nadal trwa. Nie ma zabezpieczenia idealnego, którego nie da się zastąpić lub podrobić żadną z metod. Z drugiej strony, producent nie może też przesadzić, aby nie utrudnić sonie życie uniemożliwiając serwis i aktualizowanie oprogramowania w jakiś normalny sposób. Konstruując zabezpieczenie za każdym razem trzeba wziąć pod rozwagę takie czynniki, jak na przykład czas funkcjonowania produktu. Warto też pomyśleć, co tak naprawdę chcemy zabezpieczyć. Może na przykład nie będzie nam przeszkadzało to, że ktoś wykona klon naszej przemysłowej drukarki atramentowej, jeśli to właśnie my będziemy dostarczali do niej atrament?

Jacek Bogusz, EP