Czy Kaspersky Lab pracuje nad własnym systemem operacyjnym? Potwierdzamy pogłoski i ucinamy spekulacje!

Witajcie!

W dniu dzisiejszym chciałbym poruszyć temat przyszłości. Niestety, niezbyt świetlanej przyszłości pełnej masowych cyberataków na instalacje jądrowe, przerw w dostawach energii, zaburzeń w funkcjonowaniu transportu, awarii systemów finansowych i telekomunikacyjnych. Ogólnie, będę mówił o przyszłości obfitującej w incydenty mające na celu unieruchomienie tego, co nazywamy krytyczną infrastrukturą przemysłową. Wyobraźcie sobie to, co działo się w Szklanej pułapce 4 – gdzie atak na infrastrukturę pogrążył w chaosie prawie cały kraj – o tym właśnie chciałem dzisiaj porozmawiać.

Niestety, w realnym świecie nie mamy Johna McClane'a, który migiem rozwiązałby problem wrażliwych systemów przemysłowych, a nawet gdybyśmy go mieli, to obawiam się, że typowe dla niego metody działania nie byłyby pomocne. Niezrażeni tym, pracujemy nad rozwijaniem technologii bezpiecznego systemu operacyjnego – technologii mającej na celu ochronę krytycznych systemów kontroli przemysłowej (ICS). Ostatnio w internecie pojawiło się kilka doniesień na temat naszego projektu, więc myślę, że nadszedł czas, aby uchylić rąbka tajemnicy, która spowija nasze tajne działania i poinformować Was o tym, co naprawdę się dzieje.

Ale najpierw pozwolę sobie na małe wprowadzenie Was w krainę podatnych na ataki systemów przemysłowych i wyjaśnienie, dlaczego świat *naprawdę* potrzebuje naszego nowatorskiego podejścia do problemu.

Bezbronność systemów przemysłowych

Mimo że na pierwszy rzut oka przemysłowe systemy IT i, powiedzmy, typowe biurowe sieci komputerowe mają ze sobą sporo wspólnego, to w rzeczywistości są to kompletnie inne twory – głównie pod względem priorytetów bezpieczeństwa i użyteczności. W typowej firmie najważniejsza jest poufność danych. Jeżeli zatem – na przykład – na firmowym serwerze plików wykrywany jest trojan, najprostszym rozwiązaniem jest odłączenie zainfekowanego systemu od sieci i spokojne rozpoczęcie rozwiązywania problemu w odizolowanym środowisku.

W systemach przemysłowych powyższa metoda nie ma racji bytu, ponieważ tutaj priorytetem jest utrzymanie ciągłości pracy za wszelką cenę. Nieprzerwana ciągłość produkcji ma znaczenie nadrzędne w każdym obiekcie przemysłowym na świecie, a ochrona jest zawsze spychana na drugi plan.

Prowadzi to do sytuacji, w której oprogramowanie infrastruktury IT w instalacji przemysłowej jest uaktualniane tylko po dokładnym sprawdzeniu tolerancji błędów, tak, aby nie przerywać realizacji najważniejszych zadań. A ponieważ taka kontrola wymaga mnóstwa wysiłku (nie dając i tak gwarancji braku awarii), wiele firm po prostu nie fatyguje się, aby uaktualnić ICS – pozostawiając przestarzałe oprogramowanie na całe dekady. Niedawno czytałem ciekawy artykuł, który wymienia 11 zasad bezpieczeństwa ICS; jedną z pierwszych zasad jest „Nie dotykać. Nigdy". Czy potrzeba bardziej obrazowego przykładu?!

Aktualizacja oprogramowania może być również wyraźnie zabroniona przez przemysłowe lub infrastrukturalne reguły bezpieczeństwa firmy. Nawet jeżeli istniałaby możliwość aktualizacji oprogramowania i załatania „dziur", to niewiele by to pomogło. Producenci specjalistycznego oprogramowania nie są zainteresowani stałą analizą kodu źródłowego i tworzeniem poprawek łatających dziury. Jak pokazuje doświadczenie, nakłady finansowe na tego rodzaju działalność są zazwyczaj ucinane, a poprawki są publikowane okazjonalnie dla wykrytych exploitów, które zostały umieszczone w internecie i stały się powszechnie dostępne. W rzeczywistości jest to prawdą nie tylko dla specjalistycznego oprogramowania, ale także dla ogółu tworów programistycznych, niemniej dziś mówimy konkretnie o oprogramowaniu przemysłowym.

Problem polega na tym, że *luki w oprogramowaniu kontrolnym, programowalnych kontrolerach i przemysłowych sieciach komunikacyjnych powodują, że operatorzy systemów przemysłowych i infrastrukturalnych nie mają możliwości otrzymywania wiarygodnych informacji o całkowitym stanie operacyjnym poszczególnych systemów!* Teoretycznie możliwa jest sytuacja, w której np. atakowany jest system dystrybucji energii elektrycznej, a w wyniku tego ataku awaria pojawia się w jakiejś odległej instalacji, usytuowanej w innej części kraju. Ale centrum sterowania nic o tym nie wie, bo napastnicy zawczasu przesłali fałszywe dane na odpowiednie komputery.

Przykłady

Nie trzeba się bardzo natrudzić, aby znaleźć przykłady opisanej powyżej sytuacji w realnym świecie. Przykładem cybersabotażu, w jego potencjalnie najbardziej niebezpiecznej formie, był bezpośredni atak na systemy SCADA, przeprowadzony w 2000 r. w Australii. Pracownik podwykonawcy, zajmujący się systemami kontroli Maroochy Shire Council, zdołał w 46 (!) atakach spowodować kompletne zatrzymanie lub nieprawidłową pracę pomp kanalizacyjnych. Nikt nie mógł zrozumieć, co się dzieje, i dlaczego naruszona została komunikacja wewnątrz systemu kontroli. Dopiero po *miesiącach* od całego incydentu władzom przedsiębiorstwa i lokalnym organom ścigania udało się rozpracować, co właściwie zaszło. Okazało się, że pewien pracownik bardzo chciał dostać pracę w firmie kanalizacyjnej, a kiedy jego aplikacja została odrzucona, postanowił on zalać ściekami ogromny obszar Queensland!

Istnieje wiele takich przykładów, a nie słyszymy o nich na co dzień, ponieważ po prostu nie przedostają się one do mediów. Poza tym, firmy padające ofiarą naruszeń bezpieczeństwa na ogół nie są zbyt chętne do ogłaszania całemu światu, że ich systemy zostały naruszone. Występuje tak wiele incydentów, że często nawet same ofiary nie wiedzą o ataku. Nie tak dawno temu w routerach przemysłowych RuggedCom wykryto lukę, która pozwalała atakującym na podniesienie własnych praw dostępu do poziomu administratora i przejęcie kontroli nad urządzeniem. Można tylko spekulować - przez kogo, kiedy, jak i gdzie luka mogła być wykorzystana? Podobnie spekulować można odnośnie liczby luk, które przemykają niezauważone...

Dla lepszego zrozumienia tematu proponuję poczytać (na przykład tutaj) o atakach na ICS, które odniosły sukces w realizacji szkodliwych zadań.

Więc kto jeszcze – oprócz szantażystów, niezadowolonych kandydatów do pracy itd. – chciałby uzyskać dostęp do kodu źródłowego oprogramowania ICS, sterowników i systemów operacyjnych? Oczywiście, odpowiednie władze publiczne. Takie, które posiadają departamenty odpowiedzialne za zaświadczanie, że oprogramowanie do systemów jest ważne i spełnia określone wymogi, ale również i władze rozwijające

(w ostatnich latach) cyberbroń do atakowania systemów przeciwnika, kimkolwiek by on nie był.

Tak, mówię tutaj o rzeczach takich jak Stuxnet oraz jego kuzyni: Dugu, Flame i Gauss – mówię o szkodliwym oprogramowaniu tak bardzo skomplikowanym, że musiało zostać opracowane przy wsparciu rządowym. I nie ma to znaczenia, kto jest obecnie celem; istotne jest to, że takie cyberbronie są w ogóle opracowywane i wdrażane. A kiedy otworzy się puszkę Pandory, nie ma sposobu na jej ponowne zamknięcie. Budowanie uzbrojenia do ataków na instalacje przemysłowe i wrogie infrastruktury *wcześniej czy później dotknie nas wszystkich*. Tak więc okazuje się, że największe zagrożenie dla naszej planety nie pochodzi dzisiaj od zwykłego „cybernetycznego pif – paf", ani nawet nie ze strony zorganizowanych grup cyberprzestępczych, ale od sponsorowanych przez rządy twórców cyberbroni.

Ochrona dzisiaj: niestety, nieefektywna
Przedsiębiorstwa i rozmaite agencje nie od dziś przygotowują plany awaryjne na wypadek cyberataków. Ale w jaki sposób faktycznie to robią?

Tak naprawdę istnieją tylko dwa sposoby. *Pierwszy* – polega na izolacji obiektów krytycznych: odłączeniu ich od internetu lub fizycznej izolacji od otaczającego świata, przeprowadzonej w jakiś inny sposób. Jednak, jak pokazuje doświadczenie, jeśli technik podczas nocnej zmiany chce oglądać filmy z zainfekowanego dysku USB na komputerach sterujących – nic nie jest w stanie go powstrzymać (aktualnie mamy już opracowane metody blokujące takie działania, ale o tym kiedy indziej).

Drugi sposób to poufność. Zbiorowe i prowadzone na dużą skalę próby zachowania w tajemnicy wszystkiego i wszędzie. Twórcy ICS trzymają w tajemnicy kod źródłowy, właściciele fabryk i infrastruktury umieszczają na schematach stempel „ŚCIŚLE TAJNE", rodzaje stosowanego oprogramowania są trzymane w tajemnicy itd. Jednak, w tym samym czasie informacje o podatnościach na ataki występujących - na przykład - w większości popularnych systemów SCADA są ogólnie dostępne w internecie. A jeśli poszukamy głębiej – okazuje się, że już od kilku lat pracuje otwarty silnik wyszukiwania SHODAN – przeznaczony m.in. do poszukiwania słabych punktów w systemach przemysłowych (ze szczególnym uwzględnieniem SCADA), których właściciele decydują się podłączyć je do internetu lub zapomnieli odłączyć je od sieci.

Jednocześnie specjaliści z organizacji przemysłowych i infrastrukturalnych stosują również tradycyjne metody ochrony wrażliwego oprogramowania i systemów operacyjnych, polegające na kontroli użytkowania programów, a także na monitorowaniu działań użytkowników. Ale i w tym aspekcie nie ma 100% gwarancji bezpieczeństwa - tym razem ze względu na luki w oprogramowaniu sprawującym kontrolę. A dla krytycznej infrastruktury gwarancja jest tym, co jest potrzebne najbardziej.

Ochrona taka, jakiej naprawdę potrzebujemy
W idealnym przypadku całe oprogramowanie ICS powinno zostać przepisane od nowa, łącząc w sobie istniejące technologie zabezpieczeń oraz biorąc pod uwagę nowe realia cyberataków. Niestety, również taki kolosalny wysiłek w połączeniu z ogromnymi inwestycjami, które są wymagane przy testach i dostrajaniu parametrów ochrony, nie gwarantuje dostatecznie stabilnego działania systemów przemysłowych.

Istnieje alternatywa: bezpieczny system operacyjny. System, na którym może zostać zainstalowane oprogramowanie ICS, i który może być wbudowany w istniejącą infrastrukturę – platforma zapewniająca sterowanie i gwarantująca otrzymanie wiarygodnych raportów o stanie innych podsystemów.

Najpierw odpowiem na najbardziej oczywiste pytanie: w jaki sposób Kaspersky Lab może stworzyć bezpieczny system operacyjny, skoro żaden z gigantów, takich jak Microsoft czy Apple, ani żadne środowisko „open source" - nie dało rady tego zrobić? A to naprawdę jest proste...

Po pierwsze: nasz system jest ściśle dopasowany do sprecyzowanych wymogów, opracowany do spełniania bardzo konkretnego zadania, a nie przeznaczony do grania w Half – Life'a, montażu filmów z wakacji czy ćwierkania na Twitterze. *Po drugie*: pracujemy nad sposobami pisania oprogramowania, które z definicji nie będzie w stanie przeprowadzić żadnego ukrytego działania czy innej nielegalnej operacji. Brak możliwości wykonania kodu osób trzecich, włamania się do systemu lub uruchomienia nieuwierzytelnionych aplikacji jest tu kluczową kwestią i na naszym systemie dokładnie tak to wygląda.

Na zakończenie, w oczekiwaniu na wiele pytań od kolegów, partnerów, przedstawicieli mediów i zwyczajnie ciekawych ludzi, podaję jasny komunikat: rozwijamy chronione środowisko. Jest to wyrafinowany projekt, prawie niewykonalny bez aktywnego współdziałania z operatorami i dostawcami ICS. Nie możemy ujawnić szczegółów tego przedsięwzięcia, ponieważ obowiązuje nas klauzula poufności. Również my nie chcemy mówić o pewnych rzeczach, aby konkurencja nie rozszyfrowała naszego „know-how". I wreszcie, należy mieć na uwadze fakt, że pewne szczegóły pozostaną na zawsze niedostępne dla opinii publicznej – tylko dlatego, aby nie dopuścić do aktów cyberterroryzmu. Ale zapewniam, że jak tylko pojawiają się jakieś możliwości, opowiem Wam o projekcie w sposób bardziej szczegółowy.

Do następnego razu!
Jewgienij Kasperski