УДК 621.391

ДОРОГИЙ Я. Ю.

ТС-СУМІСНА АРХІТЕКТУРА КРИТИЧНОЇ ІТ-ІНФРАСТРУКТУРИ

Критичні ІТ-інфраструктури можна моделювати як кібер-фізичні системи (cyber-physical systems), у яких програмні компоненти контролюють основні фізичні процеси з метою оптимізації системних вимог на основі фізичних властивостей та обмежень, а також поточного та майбутнього прогнозованого стану системи. Такі системи зазвичай вимагають гарантованої продуктивності та рівня безпеки, помилково отримані або пропущені команди можуть дестабілізувати роботу всієї ІТ-інфраструктури, окремих її систем або навіть припинити її роботу повністю. Тим не менш, забезпечення безпеки та стабільності роботи критичної ІТ-інфраструктури з неоднорідними компонентами все ще залишається відкритою та складною проблемою.

У цій статті запропоновану надійну архітектуру для стабільної роботи критичної ІТ-інфраструктури. Обговорюються питання моделі загроз, вразливості, доступності в режимі реального часу, цілісності під час виконання та показуються шляхи отримання стійкості від навмисного втручання у роботу та непередбачених недоліків при проектуванні, використовуючи компоненти з підтримкою довірених обчислень та структуру доступу до управління.

Ключові слова: Критична ІТ-інфраструктура, кібер-фізична система, архітектура, TCN, TPM

Введення в проблему

Вирішальний характер послуг, що надаються системами критичної ІТ-інфраструктури та потенційно руйнівні наслідки таких атак, як атака хробака Stuxnet, вказують на актуальність пошуку рішень, які забезпечують для ІТ-інфраструктури ефективний захист від кібер-загроз. Однак доступність в режимі реального часу також є життєво важливою. Контролери повинні мати доступ до інформації про стан в режимі реального часу: правильні повідомлення, доставлені в неправильний час можуть призвести до помилок та відмов цілої ІТ-інфраструктури чи окремих її систем. З метою усунення таких загроз як навмисного, так і випадкового характеру, потрібно розробити нові технології, які підвищують довіру до новостворених критичних ІТ-інфраструктур. Парадигма Trusted Computing (ТС) [1] підтримує платформи застосувань для безпечних розподілених програм і забезпечує цілісність виконання цілей. Використання парадигми запобігає виконанню ненадійного програмного забезпечення, а також допомагає вирішувати питання з інсайдерськими загрозами. Однак ТС-сумісна архітектура забезпечує лише статичний захист, тому треба розглянути можливість атак в реальному часі (під час виконання програм), які обходять захист ТС і будуються на вразливостях, що змінюють виконання машинного коду.

Аналіз існуючих рішень

Технології Trusted Computing використані у різних застосуваннях, що дозволили підвищити довіру до розподілених систем. У [2] автори інтегрують компоненти ТС з Kerberos [3] з метою здійснення авторизації методом атестації. У праці [4] Trusted Platform Module (TPM) використовується для забезпечення безпеки записів електронних машини для голосування шляхом обов'язкового прив’язування вибору виборців до ключів з надійного сховища.

Є велика кількість праць, які описують методи захисту систем від загроз часу виконання, але проблема все ще викликає стурбованість та не вирішена до кінця. Нещодавні пропозиції використовують аналіз уражень та моніторинг динамічної цілісності поведінки програмного забезпечення ІТ-інфраструктур під час виконання, щоб визначити і відслідкувати порушення безпеки [5, 6, 7, 8]. Існують різні методики, що дозволять включити відстеження, наприклад, інструменти коду, бінарні переписування та підтримка архітектурного відстеження. В деяких роботах, незважаючи на прикладені зусилля щодо оптимізації продуктивності процесу моніторингу систем ІТ-інфраструктури (шляхом виявлення

непотрібних перевірок, тощо), досі не вирішена нетривіальна проблема накладних витрат, пов'язаних з динамічним вимірюванням параметрів інфраструктури [6].

Група Trusted Computing Group (TCG) [1] опублікувала специфікації для архітектур та інтерфейсів для деяких обчислювальних реалізацій. Платформи, що побудовані на базі цих специфікацій вважаються такими, що відповідають функціональним вимогам та надійності комп'ютерних систем і забезпечують підвищену довіру до них. Таким чином, вони добре підходять для підтримки та захисту критичної IT-інфраструктури. Дві з таких платформ - це модуль Trusted Platform (TPM) [9] та компонент Trusted Network Connect (TNC) [10].

TPM зв'язує дані з платформою конфігурації обладнання з метою підвищення безпеки програмного забезпечення. Він має дві основні можливості: віддалена атестація і надійне зберігання, а також підтримка низки криптографічних примітивів. TPM впроваджує надійний набір інструментів - вузли довіри, з метою забезпечення довіри до очікуваної поведінки систем критичної IT-інфраструктури. Довіра заснована на цілісності захищеного процесу завантаження системи, в якій виконується файл. Код та пов'язані з ним дані конфігурації оцінюються раніше ніж він виконується. Для віддаленої атестації TPM використовує атрибут «ідентифікаційний ключ атестації» для затвердження поточного стану програмного середовища шляхом перевірки у третьої сторони підписаних значень параметрів конфігурацій. Закрите сховище використовується для захисту криптографічних ключів. TCG вимагає, щоб TPM були фізично захищені від втручання. Така вимога включає в себе зв'язування TPM з фізичними частинами платформи (наприклад, материнською платою).

TNC – це архітектура Trusted Computing для надійних мережевих програм. Вимога перевірки конфігурацій операційної системи клієнта та сервера до створення комунікаційного каналу між ними відрізняє TNC від інших архітектур забезпечення сумісності. Довірений канал між клієнтом і сервером встановлюється лише, якщо:

1) Сутність клієнта та сервера встановлена і перевірена. Інфраструктура відкритих ключів використовується для створення довіреного каналу між Root Authority і TPM клієнта та сервера.

2) Клієнт має в доступ до сервера в режимі реального часу.

3) Клієнт та сервер автентифіковані. Вузли довіри на TPM обох сторін видають необхідні ключі для виконання протоколу рукостискання [10].

4) Цілісність передачі даних, а при необхідності і конфіденційність, гарантуються TPM.

TPM запобігає запуску скомпрометованих компонентів. В результаті, якщо виключити загрози часу виконання, шкідливі загрози зводяться лише до DoS-атак, які можна подолати шляхом резервування компонентів системи. Є два види втручань, які можуть вплинути на ТС-сумісну критичну IT-інфраструктуру: природні (аварії) та змагальні (навмисні, зловмисні, інсайдерські).

Природні втручання можна передбачити шляхом розрахунку ймовірності виникнення таких втручань. З метою зменшення цієї ймовірності нижче прийнятного порогу використовується те ж саме резервування. В цій роботі зроблено припущення про те, що система має достатній рівень резервування компонентів і ймовірність природних втручань або успішної DoS-атаки досить мала.

Мета роботи

Метою даної роботи є розробка ТС-сумісної архітектури критичної IT-інфраструктури та її математичної моделі.

Модель критичної IT-інфраструктури

З метою моделювання запропонованої в [11] архітектури критичної IT-інфраструктури як ТС-сумісної архітектури С потрібно побудувати (описати) наступні компоненти:

1) Модель реального часу, яка відображає її функціональність, включаючи розподіл помилок F кожного її компоненту (ПК, ЦОД, Сервери і т. ін.).

2) Набір специфікацій SP, політик P та вимог безпеки R, які описують всі вразливості V, від яких пропонована IT-інфраструктура має бути захищена. На кожному рівні архітектури цей набір буде іншим.

3) SPPR-профайлер, який емулює поведінку С в реальному часі в рамках застосованих до неї SP, P, R.

Опишемо модель критичної ІТ-інфраструктури засобами розмічених транзиційних систем (РТС) [12, 13]. Модель має наступний вигляд:

$$C = (S, s_0, L, T, \tau, F, SP, P, R, V)$$

де S - скінченна множина станів критичної ІТ-інфраструктури C. Множина поділяється на наступні множини: S_{sf} - множину безпечних станів, S_{cr} - множину критичних станів та S_t - множину термінальних станів;

s_0 - початковий стан, $s_0 \in S_{sf}$;

L - скінченна множина відміток переходів, що включає спеціальний символ †;

T - функція переходів, що активуються часом, $T \subset SxSxL$;

τ - диспетчер часу;

F - розподіл втручань в роботу компонентів;

SP - набір специфікацій;

P - набір політик;

R - набір вимог безпеки;

V - набір вразливостей.

Наступний запис:

$$T(t_i): s \xrightarrow{t_i, l} s' \text{ для } (s, s', l) \in T \text{ та } t_i \in \tau$$

описує зміну стану, яку ініціює дія l в момент часу t_i. Функція переходів T детермінована у випадку, коли $l \in L \setminus \{\dagger\}$, або ймовірнісна, у випадку, коли $l = \dagger$. В останньому випадку, значення стану формується випадково згідно розподілу F.

Взаємодіючими учасниками в такій моделі є:

- розумні пристрої (телекомунікаційні, промислові і т. ін.);
- оператори систем критичної ІТ-інфраструктури;
- супротивники;
- середовище.

Їх логіка функціонування описується наборами специфікацій SP, політик P та вимогами безпеки R.

Середовище контролює часові та просторові аспекти всіх подій в моделі та диспетчиризує всі зміни станів відповідно до τ, використовуючи для цього розподіли F з метою створення стратегій виходу з ладу доступних компонентів, а також, для вирішення проблем одночасного виникнення подій в критичній ІТ-інфраструктурі та їх обробки.

Імплементація архітектури

Для демонстрації ефективності запропонованої архітектури покажемо її використання для захисту взаємодії системи автоматизації підстанції (SAS) електромережі в реальному часі при наявності загрози цілісності під час виконання. Для надійності зв'язку SAS слід використати пропоновану ТС-сумісну архітектуру, що об'єднує наступні компоненти:

- TPM з вбудованим сервісом довіри;
- сервіс аутентифікації Kerberos;
- систему керування доступом на основі атрибутів в реальному часі;
- криптографічні служби для AES та HMAC на базі алгоритму SHA2.

Захист від атак під час виконання гарантується у випадку, якщо все довірене програмне забезпечення розроблене на належному рівні (відсутні баги і т. ін.). Це припущення є прийнятним для критичної ІТ-інфраструктури з мінімальною ОС.

Структура макета для експериментальних досліджень складається з наступних компонент:

- профайлеру IEEE C37.118 [14];
- 5 робочих станцій з TPM на базі ОС Ubuntu Linux;
- комутатора CISCO Catalyst 3750;
- серверу з встановленим Kerberos 5 Release 1.15.2 [15].

Емуляція захищеного доступу до SAS досягається за допомогою використання TPM та TNC, а також серверу доступу Kerberos. Такий підхід дозволяє виключити скомпрометовані компоненти та гарантує доступність в реальному часі для високопріоритизованих пакетів від SAS.

Автентифікація клієнтів, передача 1000 повідомлень від SAS до клієнтів та у зворотному напрямку, їх криптографічна обробка (шифрування, хешування) показали затримку

передачі на рівні 0,6 мс. Такий результат є досить прийнятним.

Висновки

В ході дослідження визначена термінологія та вимоги до безпечного, надійного функціонування критичної ІТ-інфраструктури як ТС-сумісної архітектури. Запропонована узагальнена модель критичної ІТ-інфраструктури у вигляді розміченої транзиційної системи, визначені її компоненти та їх функціональні можливості. Проведено експериментальне дослідження запропонованої архітектури. В подальшому планується реалізувати запропоновану модель у середовищі моделювання Matlab для дослідження пропонованої архітектури та порівняння результатів моделювання з результатами, що отримані на віртуальному макеті.

Список посилань

1. «TCG,» [Онлайновий]. Режим доступу: http://www.trustedcomputinggroup.org.
2. Leicher A. Implementation of a trusted ticket system / A. Leicher, N. Kuntze, and A. U. Schmidt // IFIP Advances in Information and Communication Technology, vol. 297, pp. 152-163, 2009.
3. Neuman C. The Kerberos Network Authentication Service (V5) / C. Neuman, T. Yu, Hartman, and K. Raeburn // RFC 4120, July 2005.
4. Fink R.A. TPM meets DRE: reducing the trust base for electronic voting using trusted platform modules / R. A. Fink, A. T. Sherman, and R. Carback // IEEE Transactions on Information Forensics and Security, vol. 4, no. 4, pp. 628-637, 2009.
5. Levin D. TrInc: small trusted hardware for large distributed systems / D. Levin, J. R. Douceur, J. R. Lorch, and T. Moscibroda // in Proc. 6th USENIX Symp. Networked Systems Design and Implementation. - USENIX Association, Berkeley, CA, USA. , 2009.
6. Chang W. Efficient and extensible security enforcement using dynamic data flow analysis / W. Chang, B. Streiff, and C. Lin // in ACM Conf. Comp. and Comm. Security, 2008, pp. 39.50.
7. Cheng W. TaintTrace: Efficient flow tracing with dynamic binary rewriting / W. Cheng, Q. Zhao, B. Yu, and S. Hiroshige // in Proc. 11th IEEE Symp. Computers and Communications (ISCC '06), 2006, pp. 749-754.
8. Qin F. Lift: A low-overhead practical information flow tracking system for detecting security attacks / F. Qin, C.Wang, Z. Li, H. seop Kim, Y. Zhou, and Y.Wu // in 39th IEEE/ACM Int. Symp. Microarchitecture, (MICRO-39), 2006, pp. 135-148.
9. TCG. TPM Structures, Level 2, Ver. 1.2, Rev. 116 // in Communication networks and systems for power utility automation, March 2011.
10. Trusted Network Connect Architecture for Interoperability; Specification 1.3; Rev. 6. – April 2008.
11. Dorogyy Y.Y. MANAGEMENT OF CRITICAL IT-INFRASTRUCTURES/Y.Y.Doroogyy // Information and Telecommunication Sciences, no. 1, pp. 10-15, 2014.
12. Дорогий Я.Ю. ПРОЕКТУВАННЯ КРИТИЧНИХ ІТ-ІНФРАСТРУКТУР З ВИКОРИСТАННЯМ РОЗМІЧЕНИХ ТРАНЗИЦІЙНИХ СИСТЕМ / Я.Ю.Дорогий // Матеріали V заочної наукової конференції «Фундаментальні та прикладні дослідження в сучасній науці", Харків, Україна, 31 жовтня, 2017.
13. Дорогий Я.Ю. Моделювання критичних ІТ-інфраструктур за допомогою розмічених транзиційних систем / Я.Ю.Дорогий // Матеріали VI Міжнародної науково-практичної конференції, Чернівці, 9-11 листопада 2017 року.
14. "pyPMU - Python implementation of the IEEE C37.118 synchrophasor standard," [Online]. Режим доступу: https://github.com/iicsys/pypmu. [Accessed 03 10 2017].
15. "Kerberos V5 Release 1.15.2 - current release (2017-09-25)," [Online]. Режим доступу: http://web.mit.edu/kerberos/dist/#krb5-1.15. [Accessed 03 10 2017].