СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

Монографія

За загальною редакцією
д-ра екон. наук, професора В. С. Пономаренка

Харків
ХНЕУ ім. С. Кузнеця
2022

УДК 004(0.034)
С91

Авторський колектив: д-р техн. наук, професор Н. Г. Аксак – розділ 1; д-р пед. наук, професор Л. Е. Гризун, канд. техн. наук, доцент О. В. Щербаков – розділ 2; канд. техн. наук, доцент В. А. Золотарьов – розділ 3; канд. екон. наук, доцент І. О. Золотарьова – розділ 4; д-р техн. наук, професор М. М. Корабльов – розділ 5; канд. техн. наук, доцент В. П. Коцюба – розділ 6; канд. техн. наук, доцент М. Ю. Лосєв – розділ 7; канд. техн. наук, доцент О. В. Фролов – розділ 8; д-р техн. наук, професор С. В. Мінухін – розділ 9.

Рецензенти: завідувач кафедри репрографії Навчально-наукового видавничо-поліграфічного інституту Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського", д-р техн. наук, доцент *О. О. Палюх*; професор кафедри комп'ютерних технологій і мехатроніки Харківського національного автомобільно-дорожнього університету, д-р техн. наук *О. П. Алексієв*; професор кафедри комп'ютерних наук та інформаційних технологій Української академії друкарства, д-р техн. наук *О. В. Тимченко*.

Рекомендовано до видання рішенням ученої ради Харківського національного економічного університету імені Семена Кузнеця.
Протокол № 4 від 25.05.2022 р.

Самостійне електронне текстове мережеве видання

Сучасні інформаційні технології та системи [Електронний ресурс] : монографія / Н. Г. Аксак, Л. Е. Гризун, О. В. Щербаков та ін. ; за заг. ред. д-ра екон. наук, професора В. С. Пономаренка. – Харків : ХНЕУ ім. С. Кузнеця, 2022. – 271 с.
С91
ISBN 978-966-676-850-9

Розглянуто сучасний стан та перспективи розвитку сучасних інформаційних технологій і систем різних видів та різного прикладного характеру.

Монографія становить інтерес як для фахівців, сферу діяльності яких безпосередньо пов'язано з розробленням прикладних інформаційних технологій і систем, так і для більш широкого кола фахівців. Вона буде корисною викладачам, аспірантам і студентам, що спеціалізуються в галузі інформаційних технологій, і всім, хто серйозно цікавиться проблемами інформаційного суспільства.

УДК 004(0.034)

ISBN 978-966-676-850-9

Розділ 3
Захист інформації в інфокомунікаційній мережі за віддаленої роботи установи

3.1. Вступ і формулювання завдання

За період пандемії, коли багато організацій вимушені організовувати віддалену роботу, питання захисту інформації в інформаційно-телекомунікаційних мережах стало актуальним як ніколи. Масштабна кібератака 14 січня 2022 року, коли хакери атакували й частково зламали понад 70 державних вебсайтів України, зайвий раз довела слабкість кіберзахисту більшості установ і нехтування рекомендаціями CERT-UA – урядової команди реагування на комп'ютерні надзвичайні події України, яка функціонує у складі Державного центру кіберзахисту Державної служби спеціального зв'язку та захисту інформації України.

Причиною вдалої атаки вебсайтів імовірно стала вразливість системи управління вмістом сайтів October CMS, яку виявили ще у травні 2021 року. За сім місяців ніхто з відповідальних за кібербезпеку в установах, які зазнали атаки, не спромігся оновити програмне забезпечення.

Для складання рекомендацій для безпечної віддаленої роботи організації насамперед потрібно розглянути наявні загрози інфокомунікаційній мережі (кіберінциденти); потім дослідити сучасне шкідливе програмне забезпечення, яке застосовують зловмисники для кібератак і з'ясувати, які сучасні методи захисту використовують для його знешкодження.

Окремо слід дослідити питання захищеності вебсерверів: виявити найнебезпечніші кібератаки та вразливості, які вони використовують, запропонувати методи захисту, згідно з рекомендаціями CERT-UA [35].

3.2. Категорії кіберінцидентів

Закон України "Про основні засади забезпечення кібербезпеки України" визначає кіберінцидент (інцидент кібербезпеки) як подію або ряд несприятливих подій ненавмисного характеру (природного, технічного, технологічного, помилкового, зокрема внаслідок дії людського фактора) та/або таких, що мають ознаки можливої (потенційної) кібератаки, які становлять загрозу безпеці систем електронних комунікацій, систем управління технологічними процесами, створюють імовірність порушення

штатного режиму функціонування таких систем (зокрема зриву та/або блокування роботи системи, та/або несанкціонованого управління її ресурсами), ставлять під загрозу безпеку (захищеність) електронних інформаційних ресурсів.

Перелік категорій кіберінцидентів було розроблено CERT-UA [18], згідно з рекомендаціями Європейської агенції з кібербезпеки (ENISA Reference Incident Classification Taxonomy) і спільного документа ENISA та Європейського центру боротьби з кіберзлочинністю Європолу (Common Taxonomy for Law Enforcement and The National Network of CSIRTs). Усі відомі типи інцидентів було розподілено на 10 категорій. Кожній категорії й типу інциденту було надано двозначний код.

Категорія 01. "Шкідливий (образливий) вміст" (Abusive content), складається з одного типу інциденту – 01. Спам (Spam) – надсилання небажаних повідомлень або великої кількості повідомлень (флуд).

Категорія 02. "Шкідливий програмний код" (Malicious Code) містить такі типи інцидентів: 01. Зараження шкідливим програмним забезпеченням (Malware) infection – у системі виявлено шкідливе програмне забезпечення (ШПЗ); 02. Розповсюдження ШПЗ (Malware distribution) – розповсюдження ШПЗ, наприклад шляхом розсилання повідомлень електронної пошти, що містять укладення із ШПЗ або посилання на його завантаження; 03. Командно-контрольний центр (C2) (Command & Control (C2)) – система, яку використовують як точку управління ботнетом та/або слугує точкою для збирання інформації, викраденої ботнетами; 04. Шкідливе підключення (Malicious connection) – спроби з'єднання від/до IP/URL-адреси, пов'язаної з відомим ШПЗ, наприклад C2C або ресурсом розповсюдження компонентів, пов'язаних з активністю певної ботмережі.

Категорія 03. "Збирання інформації зловмисником" (Information Gathering) складається із трьох типів інцидентів: 01. Сканування (Scanning) – збирання інформації про системи або мережі; 02. Сніфінг (Sniffing) – несанкціоноване перехоплення (логічне або фізичне) та аналіз мережевого трафіка. Несанкціонований моніторинг та зчитування мережевого трафіка; 03. Фішинг (Phishing) – спроба збирання інформації про користувача чи систему за допомогою методів соціальної інженерії (масове розсилання електронною поштою, спрямоване на збирання даних, може містити посилання на фішингові сайти).

Категорія 04. "Спроби втручання" (Intrusion Attempts) містить два типи інцидентів: 01. Спроба експлуатації вразливості (Vulnerability exploitation

attempt) – спроба вторгнення з використанням вразливості в системі, компоненті чи мережі; 02. Спроби авторизації/входу в систему (Login attempts) – спроба входу до служб або механізмів автентифікації / доступу. Невдала спроба підбору автентифікаційних даних чи використання раніше скомпрометованих, уже неактуальних даних.

Категорія 05. "Утручання" (Intrusion) об'єднує два типи інцидентів: 01. Компрометація облікового запису (Account compromise) – фактичне вторгнення в систему, компонент або мережу шляхом компрометації облікового запису користувача або адміністратора; 02. Компрометація системи (System compromise) – фактичне вторгнення в систему чи її компонент, сервіс, застосунок через використання вразливості в компоненті або мережі. Несанкціонований доступ до системи або компонента в обхід системи контролю за доступом.

Категорія 06. "Порушення доступності" (Availability) складається із трьох типів інцидентів: 01. Атака на відмову в обслуговуванні (DoS/DDoS) – вплив на нормальне функціонування системи чи сервісу, що досягають спрямуванням з одного чи багатьох джерел до цільового ресурсу запитів для перенасичення пропускної спроможності чи системних ресурсів; 02. Саботаж / шкідливі дії (Sabotage) – дії (навмисні або ненавмисні), спрямовані на пошкодження системи, переривання процесів, зміну або видалення інформації тощо; 03. Перебій (Outage, no malice) – перебій у роботі системи чи її компонента без зловмисного втручання.

Категорія 07. "Порушення властивостей інформації" (Information Content Security) містить два типи інцидентів: 01. Несанкціонований доступ до інформації (Unauthorised access to information) – несанкціонований доступ до інформації й несанкціонований обмін конкретним набором інформації; 02. Несанкціонована модифікація (Unauthorised modification of information) – несанкціонована зміна або видалення певного набору інформації.

Категорія 08. "Шахрайство" (Fraud) складається з одного типу інцидентів – 01. Шахрайський сайт (Fraudulent site) – створення фішингових сайтів для збирання автентифікаційних чи інших даних користувачів. Використання ресурсів установи для цілей, відмінних від передбачуваних.

Категорія 09. "Відома вразливість" (Vulnerable) поєднує два типи інцидентів: 01. Уразливість (Vulnerability) – наявність у системі чи її компонентах відомих уразливостей, відкритих для експлуатації; 02. Некоректна конфігурація (Misconfiguration) – недоліки в налаштуваннях, що може бути використано зловмисником (налаштування за замовчуванням тощо).

Категорія 10. "Інше" (Other) складається з одного типу інцидентів – 01. Невизначений інцидент (Undetermined incident) – недостатньо даних для опрацювання інциденту.

Якщо код інциденту 02.04 – то це тип інциденту Malicious connection ("Шкідливе підключення").

Якщо інцидент зараховано до певної категорії, проте не визначено його тип, використовують код 00. Наприклад, код інциденту: 01.00; тип інциденту: не визначено; категорія: Abusive content.

3.3. Шкідливе програмне забезпечення

Шкідливе програмне забезпечення (ШПЗ) – це програмне забезпечення, яке за умови запуску може заподіяти шкоду різними способами, зокрема: призвести до блокування пристрою та його непридатності для використання; крадіжки, видалення або шифрування даних; використання пристроїв користувача для атак на інші організації; здобуття облікових даних, які дозволяють дістати доступ до систем або служб, якими користується суб'єкт атаки; майнінг криптовалюти; використання платних послуг на основі даних суб'єкта атаки (наприклад, телефонні дзвінки преміум-класу).

ШПЗ часто охоплює кілька категорій. Наприклад, програма може одночасно містити кейлогер, збирати паролі й бути "хробаком" для розсилання спаму. Далі наведено категорії, на які розподіляють більшість шкідливого програмного забезпечення:

Бекдор (backdoor) – шкідливий програмний код, який установлюють у систему, щоб надати зловмиснику віддалений доступ. Бекдори зазвичай дозволяють підключитися до комп'ютера з мінімальною автентифікацією або зовсім без такої та виконувати команди в локальній системі.

Завантажувач (downloader) – ШПЗ, єдиною метою якого є завантаження іншого шкідливого програмного коду. Зловмисники зазвичай установлюють завантажувачі за першого доступу до системи.

Викрадач інформації (stealer) – ШПЗ, яке збирає інформацію на комп'ютері жертви та, зазвичай, відправляє її зловмисникові. Як приклад можна навести програми, що збирають хеші паролів, перехоплювачі та кейлогери. Це ШПЗ зазвичай використовують для здобуття доступу до облікових записів інтернет-застосунків, як-от електронна пошта або інтернет-банкінг.

Руткіт (rootkit) – ШПЗ, що приховує наявність іншого коду. Руткіти зазвичай застосовують у поєднанні з іншим ШПЗ, як-от бекдор, що дозволяє

їм відкрити зловмисникові доступ до системи й ускладнити виявлення коду.

Залякувальне ПЗ (scareware) – створене для залякування атакованого користувача та спонукання його до купівлі чого-небудь. Зазвичай має графічний інтерфейс, схожий з антивірусом або іншою програмою, що забезпечує безпеку. Воно повідомляє користувачеві про наявність у його системі шкідливого коду й переконує його в тому, що єдиним виходом із ситуації є купівля певного програмного забезпечення.

Програма для розсилання спаму (spam-sending malware) – ШПЗ, яке заражає комп'ютер користувача й потім із його допомогою розсилає спам. Цей тип програм генерує дохід для зловмисників, дозволяючи їм продавати послуги з розсилання спаму.

Вірус-вимагач (ransomware) – тип шкідливого програмного забезпечення, що блокує доступ до системи або унеможливлює роботу з файлами (часто за допомогою методів шифрування), після чого вимагає від жертви викуп для відновлення вихідного стану.

Кейлоггер (keylogger) – програмне забезпечення, що реєструє кожну дію користувача, наприклад із пристроїв уведення (рух комп'ютерної мишки, натискання кнопок клавіатури). Дозволяє заволодіти даними користувача, що були введені після його встановлення.

Зазвичай зловмисники просять здійснити платіж (часто вимагають у криптовалюті), щоб розблокувати комп'ютер жертви. Однак у разі оплати немає гарантії, що жертва дістане доступ до своїх файлів. Беручи до уваги сказане раніше, важливо, щоб у режимі офлайн, завжди зберігали резервні копії найважливіших файлів і даних. CERT-UA рекомендує ігнорувати подібні вимоги, не переказувати кошти шахраям і повідомити про інцидент правоохоронні органи та CERT-UA.

3.4. Методи й засоби захисту від шкідливого програмного забезпечення

Основними методами захисту мережі від шкідливого програмного забезпечення є такі: резервне копіювання; запобігання розповсюдженню ШПЗ у мережі; запобігання запуску ШПЗ на пристроях; обмеження впливу ШПЗ; коректна робота під час ураження мережі ШПЗ.

Розгляньмо їх докладніше.

Резервне копіювання даних. Ключові дії, які потрібно вжити, щоб знизити рівень шкоди, яку може заподіяти шкідливе програмне забезпечення, – це забезпечити наявність актуальних резервних копій важливих файлів, за їхньої наявності можливо відновити свої дані, ігноруючи вимоги зловмисників. Резервне копіювання є ефективним заходом зниження ризиків від впливу ransomware. Слід регулярно здійснювати резервне копіювання даних, зберігати резервні копії на зовнішніх носіях інформації (SDD, HDD тощо) та налаштовувати функцію "відновлення системи". Потрібно періодично перевіряти можливість відновлення даних із резервних копій.

Хмарні сервіси (cloud service), що використовують синхронізацію (наприклад, Dropbox, OneDrive та SharePoint або Google Drive) не слід використовувати як єдине середовище для збереження резервних копій. Недоліком цих систем є те, що вони можуть автоматично синхронізуватися відразу після зараження файлів, і тоді можливо втратити й резервні копії також.

Здебільшого зашифровані файли не можуть бути розшифрованими ким-небудь. Не варто витрачати свій час чи гроші на послуги, які обіцяють це зробити. У деяких випадках фахівці з кібербезпеки можуть надати програми, які можуть розшифровувати файли через недоліки шкідливого програмного забезпечення. Рекомендуємо не використовувати програми для дешифрування даних із неперевірених джерел.

Можливо знизити ймовірність розповсюдження шкідливого програмного забезпечення в мережі за допомогою:

створення політик, що дозволить завантаження лише файли тих типів, які мають надходити (наприклад заборонити отримання чи передавання .EXE-файлів);

блокування вебсайтів, які є шкідливими;

перевірки антивірусними програмами файлів, що викликають підозру, у разі відсутності ліцензійного антивірусу рекомендуємо використовувати безкоштовний сервіс VirusTotal чи Cuckoo sandbox;

використання сигнатур для блокування відомого шкідливого коду.

Зазвичай названі раніше функції формуються системами на кшталт мережевих екранів, а не пристроями користувачів. Як приклад:

фільтрація пошти (у поєднанні з фільтруванням спаму), яка може блокувати шкідливі повідомлення електронної пошти та видаляти підозрілі вкладення;

використання засобів, які блокують відомі шкідливі вебсайти за відповідними списками;

використання засобів із функціями інформаційної безпеки, які можуть перевіряти вміст даних щодо відомих зловмисних програм;

під час використання віддаленого доступу дозволити підключення лише визначеним користувачам за допомогою білого списку (IP whitelisting).

Також слід ужити заходів для запобігання запуску ШПЗ. Необхідні кроки можуть бути різними для кожного типу пристроїв та операційних систем, але слід звернути увагу на такі методи захисту:

централізоване управління пристроями підприємства, щоб: дозволяти встановлювати лише те програмне забезпечення, яким довіряє організація (як приклад використання AppLocker); дозволяти запускати програми лише з надійних джерел чи ті, що мають відповідні сертифікати розробників;

використання антивірусного програмного забезпечення з технологією евристичного аналізу та вчасне оновлення його бази сигнатур;

унеможливлення підключення флеш-пристроїв та зовнішніх дисків, якщо не має повної довіри до їхнього джерела;

вимкнення або обмеження використання макросів (використовують у багатьох офісних продуктах, наприклад Microsoft Office, CorelDRAW, Notepad++).

Слід регулярно підтримувати безпечне налаштування та своєчасно встановлювати оновлення пристроїв, дотримуючись таких рекомендацій: установлювати оновлення безпеки, як тільки вони стануть доступними, щоб виправити недоліки, що використовують на ваших пристроях; увімкнути автоматичні оновлення для операційних систем, програм та мікропрограмного забезпечення, за можливості використовувати найновіші версії операційних систем та застосунків, щоб скористатися найновішими функціями безпеки.

Виконання таких заходів забезпечить швидке реагування та відновлення системи:

Використовувати двофакторну автентифікацію (2FA) для автентифікації користувачів всюди, де це можливо. Якщо облікові дані викрадено шкідливим програмним забезпеченням, це ускладнить можливість їхнього несанкціонованого використання.

За необхідності використання установою застарілих платформ (операційних систем і застосунків), рекомендуємо належним чином відокремити їх від основної частини мережі.

Не слід зберігати дані для автентифікації в легкодоступних місцях (наприклад, на робочому столі).

Потрібно використовувати для зберігання паролів менеджери паролів (наприклад KeePass, LastPass) і стійкі парольні фрази.

Потрібно регулярно переглядати та перевизначати права користувачів, щоб обмежити можливість розповсюдження ШПЗ. Шкідливі програми можна розповсюдити лише в ті місця мережі, до яких мають доступ облікові записи заражених користувачів.

Потрібно налаштовувати відповідні політики мережі, щоб використовували тільки необхідні порти, інтерфейси; використовувати програмні міжмережеві екрани (брандмауери) та штатні засоби захисту ОС від шкідливого програмного забезпечення; установлювати стабільні версії оновлень.

Якщо інфокомунікаційна мережа організації вже заражена шкідливим програмним забезпеченням, ці кроки можуть допомогти обмежити вплив вірусу: у певних випадках може бути необхідним негайне відключення заражених комп'ютерів, ноутбуків чи планшетів від усіх мережевих підключень, незалежно від проводового чи безпроводового, проте не вимикати сам пристрій; повідомити правоохоронні органи та CERT-UA.

Із метою збереження доказів несанкціонованого впливу, лише після завершення дій правоохоронних органів: слід змінити облікові дані, включно з паролями (особливо для адміністраторів); перш ніж відновити дані з резервної копії, потрібно переконатися, що копію створено до факту інфікування; за необхідності потрібно перевстановити операційні системи; слід оновити та виконати запуск антивірусного програмного забезпечення; здійснити перевстановлення операційної системи та застосунків, включно з базами даних програмного забезпечення та їхніми сигнатурами, має відбуватися в довіреному сегменті мережі; потрібно постійно відстежувати мережевий трафік щодо підозрілої мережевої активності.

3.5. Безпека вебресурсів

Є багато авторитетних ресурсів з рекомендаціями щодо захисту та коректного налаштування вебресурсів [28; 31; 32; 33 35; 39]. На авторів думку, найкраще ризики для вебресурсів та рекомендації щодо їхнього усунення описано у проєкті OWASP Top-10 [39], який роз'яснює

розробникам, проєктувальникам, архітекторам, менеджерам та організаціям наслідки, до яких призводять найвразливіші місця безпеки вебресурсів. У десятці найкращих подано основні методи захисту від цих проблем високого ступеня ризику, а також рекомендації щодо подальших дій із їхнього усунення.

Злам типу "Вставлення інструкцій", наприклад вставлення інструкцій SQL, ОС та LDAP, відбувається, коли ненадійні дані відправляють на інтерпретатор даних як частину команди, або запиту. Ворожі дані зловмисника можуть призвести до того, що інтерпретатор почне виконувати довільні команди, або зловмисник здобуде доступ до даних без належної авторизації.

Найкращий спосіб визначити, чи уразливий застосунок до вставлення інструкцій – це перевірити, чи всі інтерпретатори, що використовують, чітко відокремлюють сумнівні дані від команд або запитів. Для звернень SQL це означає використання присвоєних змінних у всіх підготовлених операторах та збережених процедурах, а також уникнення динамічних запитів.

Перевірка коду – це швидкий та надійний спосіб визначити, чи безпечно використовує застосунок "інтерпретатори". Інструменти аналізу кодів можуть допомогти аналітику безпеки визначити спосіб використання інтерпретаторів та відстежити опрацювання даних у застосунку. Спеціаліст із проникнення в системи може перевірити ці питання шляхом моделювання вторгнення, що підтвердить уразливість.

Автоматичне динамічне сканування застосунку може показати, чи є можливість вставлення інструкцій шляхом, придатним для використання. Сканери не завжди доходять до інтерпретаторів, і їм важко виявити, чи була атака успішною. Неналежне опрацювання помилок спрощує виявлення вставки інструкцій.

Процес запобігання вставленню інструкцій передбачає відокремлення сумнівних даних від команд та запитів. Найкращий варіант – використовувати безпечний ІПП (інтерфейс прикладного програмування), який узагалі не використовує інтерпретатор або забезпечує інтерфейс із заданими параметрами. Будьте обережними з такими ІПП, як збережені процедури, що є налаштованими, проте все ще можуть приховано виконати вставлення інструкцій. Якщо ІПП із заданими параметрами є не доступним, користувачу слід уникати певних символів шляхом використання спеціального синтаксису уникнення для інтерпретатора.

Позитивна перевірка даних, що вводять, або білий перелік, також рекомендують, проте не є повним захистом, оскільки багато застосунків потребують спеціальних символів під час уведення.

Некоректна автентифікація та управління сеансами – функції застосунку, пов'язані з автентифікацією та управлінням сеансами, часто є некоректно впровадженими, що дозволяє зловмисникам обходити паролі, ключі чи сеансові ідентифікатори або використовувати інші типи зламів для здобуття інших ідентифікаторів користувачів.

Ресурси управління сеансами, як-от облікові дані користувачів та ідентифікатори сеансу, можуть бути уразливими, якщо: облікові дані користувача для автентифікації не захищено під час збереження за допомогою хешування або шифрування; облікові дані можна підібрати або перезаписати в разі слабких функцій управління обліковими записами (наприклад, створення облікового запису, зміни пароля, відновлення пароля, слабких індивідуальних номерів (ІН) сеансів зв'язку); незахищені ІН сеанси зв'язку у URL (наприклад, переписування URL; ІН сеансів є уразливими для атаки; ІН сеансів є не обмеженими за часом або ідентифікатори користувачів чи автентифікації, особливо ідентифікатори "Технології єдиного входу до системи (SSO)", не перевіряють належним чином під час реєстрації; ІН сеансів не змінюють після успішного входу до системи; паролі, ІН сеансів та інші облікові дані відправляють незашифрованим з'єднанням.

Основна рекомендація для організацій для захисту від атаки "некоректна автентифікація й управління ресурсами" – це надати розробникам єдиний набір елементів сильного контролю за автентифікацією та управлінням сеансами. Такі елементи контролю мають відповідати всім вимогам до автентифікації й управління сеансами, визначеним у стандартах; мати простий інтерфейс для розробників. Задля емуляції, використання або як основи гарним прикладом є автентикатор та ІПП користувача ESAPI. Крім того, необхідно докладати всіх зусиль задля запобігання атакам XSS, що використовують для крадіжки індивідуальних номерів сесій.

Міжсайтове виконання сценаріїв (XSS) – атаки XSS відбуваються, коли застосунок здобуває ворожі дані та відправляє їх до веббраузера без належної перевірки. Атаки XSS дозволяють зловмисникам виконувати сценарії в браузері жертви, у результаті яких вони можуть перехоплювати сеанси користувача, видозмінювати вебсайти або переспрямовувати користувачів на інші шкідливі сайти.

Сервіс буде вразливим, якщо власник не забезпечить, щоб усі дані, що вводять користувачі, належним чином фільтрували, або якщо власник не перевіряє їх щодо безпеки за допомогою перевірки даних, що вводять, перед тим, як додати такі дані на сторінку, що випадає. Якщо Ajax використовують для динамічного оновлення сторінки, чи власники впевнені, що використовуєте безпечні ІПП JavaScript. Для небезпечних ІПП JavaScript також необхідно використовувати шифрування або перевірку.

Автоматизовані інструменти можуть виявляти деякі проблеми XSS автоматично. Однак кожний застосунок по-різному створює вихідні сторінки та використовує різні інтерпретатори браузера, JavaScript, ActiveX, Flash та Silverlight, що ускладнює автоматичне виявлення. Відповідно, повний захист потребує поєднання ручного перегляду коду та тестування на проникнення на застосунок до автоматичних підходів. Вебтехнології 2.0, Ajax, ускладнюють виявлення атак XSS за допомогою автоматизованих інструментів.

Запобігання атакам XSS потребує відокремлення сумнівних даних від умісту активного браузера. Найкращим варіантом є фільтрування всіх сумнівних даних, основаних на контексті HTML (тіло, атрибути, JavaScript, CSS або URL), у який будуть уносити дані. Позитивна перевірка даних, що вводять, або білий перелік, також рекомендують, проте вона не є повним захистом, оскільки багато застосунків потребують спеціальних символів під час уведення. Така перевірка має, наскільки це можливо, містити перевірку довжини даних, символів, формату та правил дій щодо таких даних перед тим, як приймати дані, що вводять.

Небезпечні прямі посилання на об'єкти – пряме посилання на об'єкт відбувається, коли розробник залишає незахищеним посилання на внутрішній об'єкт застосунка, як-от файл, каталог або ключ до бази даних. Без перевірки прав доступу або іншого захисту зловмисники можуть маніпулювати такими посиланнями, із метою несанкціонованого доступу до даних.

Найкращий шлях виявлення, чи є застосунок уразливим для небезпечних прямих посилань на об'єкти – це перевірити, чи всі посилання на об'єкти належним чином захищено. Для цього для прямих посилань на обмежені ресурси слід перевірити, чи застосунок виконав перевірку прав доступу користувача саме до ресурсу, щодо якого подано запит. Якщо посилання є непрямим, перевірити чи прив'язування до прямого посилання виконало обмеження значень, на які має право поточний користувач. Аналіз коду застосунка може швидко перевірити, чи безпечно

впроваджено кожен зі шляхів. Тестування також є ефективним для визначення прямих посилань на об'єкти та того, чи є вони безпечними. Автоматизовані інструменти, зазвичай, не визначають таких недоліків, оскільки вони не можуть розпізнати, що потребує захисту або є безпечним чи небезпечним.

Запобігання небезпечним прямим посиланням на об'єкти потребує вибору підходу до захисту кожного об'єкта, до якого має доступ користувач (наприклад, номер об'єкта, назва файлу). По-перше, використання непрямих посилань на об'єкти для користувача або сеансу. Це запобігає спробам зловмисника націлюватися безпосередньо на недозволені ресурси. Наприклад, замість використання ключа до бази даних ресурсу, у контекстному переліку шести ресурсів, дозволених для поточного користувача, використовуються числа від 1 до 6, щоб указати, яке значення вибрав користувач. Застосунок має перетворити непряме посилання для користувача назад у реальний ключ до бази даних на сервері. Перевірка доступу передбачає, що кожне використання прямого посилання на об'єкт із сумнівного джерела має містити перевірку контролю за доступом, щоб переконатися, що користувач має право доступу до запитаного об'єкта.

Небезпечна конфігурація оточення – належна безпека потребує визначення та використання безпечної конфігурації для застосунків, середовища розроблення, сервера застосунку, вебсервера, сервера бази даних та платформи. Необхідно визначати, упроваджувати та підтримувати безпечні налаштування, оскільки типові налаштування є, зазвичай, небезпечними. Крім того, ПЗ має бути оновленим.

Для того щоб перевірити, чи мають усі частини стека застосунку належну безпеку, насамперед слід перевірити таке: чи використовують будь-яке застаріле програмне забезпечення (таке програмне забезпечення містить ОС), вебсервер/сервер застосунку, СУБД, застосунки та всі бібліотеки коду; з'ясувати, чи є активованими або встановленими будь-які непотрібні елементи (наприклад, порти, сервіси, сторінки, облікові записи, привілеї); чи є активованими та незміненими стандартні облікові записи та паролі до них; чи ваша система опрацювання помилок відображає користувачам послідовність викликів функцій (трасу стека) або іншу надмірну інформацію в повідомленнях про помилки; чи налаштування безпеки у ваших інструментах розроблення застосунків (наприклад, Struts, Spring, ASP.NET) та бібліотеках установлено на безпечні значення.

Без забезпечення погодженого, постійного процесу безпеки конфігурації застосунку, системи перебувають під високим ризиком.

Основні рекомендації із захисту передбачають постійний процес посилення безпеки, що забезпечує швидке та просте розгортання іншого, належним чином заблокованого середовища. Середовище розроблення, контролю за якістю та експлуатацією мають бути однаково налаштованими (із різними паролями в кожному середовищі). Такий процес має бути автоматизованим для мінімізації зусиль, необхідних для підготовки нового, безпечного середовища.

Витік критичних даних – багато вебзастосунків неналежним чином захищають такі критичні дані, як дані платіжних карток, індивідуальні податкові номери й облікові дані для перевірки автентичності. Зловмисники можуть вкрасти або змінити такі слабо захищені дані та здійснити шахрайські операції із платіжними картками, украсти особисті дані або вчинити інші кримінальні правопорушення. Критичні дані слід додатково захищати шляхом шифрування під час збереження або передавання, а також необхідно дотримуватися певних застережень під час обміну такими даними із браузером.

Перше, що необхідно зробити – це визначити, які саме дані є критичними та потребують додаткового захисту. Наприклад, паролі, номери платіжних карток, медичні картки й особисті дані слід захищати. Для всіх таких даних слід з'ясувати таке: чи будь-які такі дані зберігають у вигляді незашифрованого тексту впродовж тривалого часу, включно з резервними копіями таких даних; чи будь-які такі дані передають у вигляді незашифрованого тексту, внутрішньо або зовнішньо (інтернет-трафік становить особливу загрозу); чи використовують будь-які старі/слабкі криптографічні алгоритми; чи генерують слабкі криптографічні ключі; чи є належним управління ключами; чи є ротація; чи є будь-які відсутні директиви безпеки або головні мітки браузера під час надання/відправлення чутливих даних до браузера.

Повний перелік ризиків, пов'язаних із небезпечною криптографією, використанням SSL та захистом даних не входить до обсягу десятки найбільших загроз вебсайтам. Тобто всі рекомендації, указані далі, є мінімально необхідними діями щодо критичних даних:

упевнитися, що шифруються всі критичні дані, що зберігають та передають, відповідно до визначених загроз;

не зберігати непотрібні критичні дані, видаляти їх так швидко, як це можливо, бо не можна вкрасти дані, яких немає;

забезпечити використання стійких стандартних алгоритмів і ключів, а також належне управління ключами, зважаючи на стандарт FIPS-140;

переконатися, що паролі зберігають за допомогою алгоритмів, призначених спеціально для захисту паролів, наприклад, bcrypt, PBKDF2 або scrypt;

відключити автоматичне заповнення форм, що збирають критичні дані та кешування для сторінок, що містять критичні дані.

Відсутність контролю за доступом до функціонального рівня: більшість вебзастосунків перевіряють права доступу до функціонального рівня перед тим, як відображати відповідну функцію в інтерфейсі користувача. Однак застосункам необхідно виконувати аналогічні перевірки контролю за доступом на сервері, коли здійснюють доступ до кожної функції. Якщо запити не перевіряють, зловмисники можуть підробляти їх для доступу до функцій без відповідної авторизації.

Найкращий шлях визначити, чи контролює застосунок доступ до функціонального рівня – це перевірити кожну функцію застосунку: чи відображає інтерфейс користувача посилання на недозволені функції; чи є перевірки автентифікації або авторизації з боку сервера; чи перевірки з боку сервера здійснюють виключно на основі інформації, що надає зловмисник.

За допомогою програми-посередника перегляньте ваш застосунок із привілейованою роллю. Потім повторно зайдіть на обмежені сторінки з менш привілейованою роллю. Якщо реакція сервера є однаковою, ви, напевно, є уразливими. Деякі програми-посередники для тестування безпосередньо підтримують такий тип аналізу.

Можна перевірити контроль за доступом у коді. Спробуйте відстежити один привілейований запит у коді та перевірити механізм авторизації. Потім визначте базу кодів, щоб виявити, де механізм не дотримується. Автоматизовані інструменти є не схильними до виявлення таких проблем.

Застосунок мусить мати постійний та легкий для аналізу механізм авторизації, що викликають з усіх функцій. Часто такий захист забезпечено одним або кількома компонентами, що є зовнішніми щодо коду застосунку. Слід проаналізувати процес управління дозволами та упевнитися, що ви можете легко оновлювати та контролювати його. Не перевантажуйте код. Механізм(и) виконання має відхиляти всі запити на стандартний доступ, вимагати чіткий дозвіл на доступ до кожної функції, відповідно до конкретної ролі. Якщо функція є залученою до послідовності дій,

що виконують, перевірте та впевніться, що всі умови для доступу зазначено належним чином.

Більшість вебзастосунків не відображають посилання та кнопки до недозволених функцій, однак такий "контроль за доступом на рівні відображення" не забезпечує реального захисту. Вам також необхідно впровадити перевірки в логічну частину контролера або програмний код, що реалізує функціональність застосунку.

Підроблення міжсайтових запитів (CSRF) – атака CSRF змушує підключений до системи браузер жертви автоматично відправляти підроблені запити HTTP, включно із фрагментом даних (кукізом) сеансу жертви та іншу інформацію щодо автентифікації до уразливого веб-застосунку. Це дає зловмисникам змогу змусити браузер жертви створювати запити, які уразливий застосунок уважає правомірними запитами жертви.

Для перевірки того, чи уразливий ваш застосунок, перевірте, чи всі посилання та форми мають непередбачувані ключі CSRF. Без таких ключів зловмисники можуть підробляти шкідливі запити. Альтернативний спосіб захисту – це вимагати від користувачів підтвердження наміру подати запит шляхом повторної автентифікації або будь-яким іншим шляхом підтвердження того, що вони є справжніми користувачами (наприклад, CAPTCHA). Слід звернути особливу увагу на посилання та форми, що викликають функції зміни стану, оскільки вони є найважливішими цілями CSRF. Потрібно перевіряти багатоетапні транзакції, оскільки вони не є захищеними як такі. Зловмисники можуть легко підробити ряд запитів за допомогою складних тегів. Зауважте, що фрагменти даних сеансів, IP-адреси джерел та інша інформація, що автоматично відправляють браузером, не забезпечують захист від CSRF, оскільки такі дані також додають до підроблених запитів.

Запобігання CSRF, зазвичай, потребує включення непередбачуваних ключів CSRF у кожний запит HTTP. Такі ключі мають бути, як мінімум, унікальними для кожного сеансу. Найкращий варіант – уставити унікальний ключ CSRF у приховане поле. Це приводить до того, що значення відправляється в тілі HTTP-запиту, запобігаючи його включенню до URL, який є більш схильним до розкриття. Унікальний ключ CSRF можна також додати в сам URL або параметр URL. Однак таке розміщення підвищує ризик розкриття URL зловмиснику, ставлячи під загрозу секретність ключа. Захист від CSRF можна також забезпечити шляхом повторної автентифікації або підтвердження особи користувача (наприклад, за допомогою CAPTCHA).

Використання компонентів із відомими вразливостями – такі компоненти, як бібліотеки, середовища розроблення та інші модулі програмного забезпечення майже завжди працюють із повними привілеями. Якщо використовують уразливий компонент, така атака може сприяти втраті критичних даних або підміні сервера. Застосунки, що використовують компоненти з відомими вразливостями, можуть знизити рівень захисту та сприяти різноманітним атакам і наслідкам.

Теоретично має бути легко виявити, чи використовуєте будь-які вразливі компоненти або бібліотеки. На жаль, звіти про вразливості в комерційному або відкритому програмному забезпеченні не завжди чітко зазначають, яка саме версія компонента є вразливою. Крім того, не всі бібліотеки використовують зрозумілу систему нумерації версій. Найгіршим є те, що не про всі вразливості повідомляють до центру обміну інформацією, у якому легко здійснити пошук; стає легше визначити такі сайти, як CVE та NVD. Для визначення того, чи є сервіси вразливими, необхідно шукати такі бази даних, а також стежити за переліком розсилань проєкту та оголошеннями про будь-що, що може бути вразливим. Якщо один із ваших компонентів має вразливість, вам слід ретельно оцінити, чи й справді ви є вразливими, шляхом перевірки, чи ваш код використовує частину компонента із вразливістю та чи може така вразливість вплинути на вас.

Один із варіантів захисту від використання компонентів із відомими вразливостями – це не використовувати компоненти, які ви не писали. Однак це майже нереально. Більшість проєктів зі створення компонентів не створюють вставки для вразливостей для старих версій. Замість цього, вони просто виправляють проблему в наступній версії. Отже, оновлення до таких нових версій є просто критичним.

Проєкти з розроблення програмного забезпечення мають забезпечити процес визначення всіх компонентів і версій, що ви використовуєте, включно зі всіма залежностями (наприклад, додаткових модулів для версій); контролю за безпекою таких компонентів у публічних базах даних, реєстрах розсилання проєкту та реєстрах розсилання безпеки й забезпечення їхнього оновлення; створення політики безпеки, яка буде регулювати використання компонентів, наприклад, потребувати певних методів розроблення програмного забезпечення, проходження тестів на безпеку та прийнятні ліцензії; де доречно, необхідно зважити на додавання безпечних обгорток, щоб відключити непотрібні функції та/або захистити слабкі або вразливі частини компонентів.

Небезпечні переадресування – вебзастосунки часто переспрямовують користувачів на інші сторінки та вебсайти, а також використовують сумнівні дані для визначення цільової сторінки. Без належної перевірки зловмисники можуть переспрямувати жертв до фальшивих чи шкідливих сайтів або використовувати переадресування для доступу до несанкціонованих сторінок.

Найкращий спосіб виявити, чи має ваш застосунок будь-які небезпечні переадресування, це:

аналізувати код для всіх переадресувань (називають передаванням у .NET). Для кожного разу використання визначте, чи включений цільовий URL у будь-які значення параметра. Якщо це так і якщо цільовий URL не перевіряється в білому переліку – ви є вразливими;

крім того, слід індексувати сайт, щоб виявити, чи він не створює переадресування (HTTP-коди відповіді 300 – 307, зазвичай, 302). Погляньте параметри, визначені до переадресування, та перевірте, чи є вони цільовим URL або частиною такого URL. Якщо так, то змініть цільовий URL і подивіться, чи сайт переадресує вас до іншої цілі;

якщо код є недоступним, перевірте всі параметри, щоб виявити, чи вони мають вигляд частини переадресування з URL-призначенням, та перевірте відповідність дій.

Крім того адміністратори вебресурсів мають забезпечувати:

управління оновленнями програмного забезпечення – необхідно постійно стежити за версіями операційної системи, системи управління контентом (CMS), менеджера пакетів, фреймворків або іншого ПЗ, що забезпечують роботу вебресурсу, та регулярно оновлювати їх. Водночас краще використовувати тільки LTS-версії;

використання HTTPS – використання протоколу HTTPS гарантує цілісність і конфіденційність взаємодії із сервером, захищає дані користувачів під час передавання в мережі "Інтернет". Сертифікат має бути виданим центром сертифікації. Необхідно використовувати TLS останньої версії (SSL має недоліки та вразливості й не є прийнятним для безпечного зв'язку). Гарною практикою є налаштування механізму HSTS (HTTP Strict Transport Security) для примусового використання HTTPS, навіть у разі переходу за посиланнями з явним зазначенням протоколу HTTP;

моніторинг журнальних файлів (логів) на підозрілі події – якщо у вебресурсу відсутня своя система журналювання (що описано в пункті A10 OWASP Top-10 2017), потрібно відстежувати журнальні файли

вебсервера (access.log). У журнальних файлах потрібно звертати увагу на POST-запити та код відповіді сервера на них. Особливу увагу варто приділяти POST-запитам до сторінок, які не мають приймати ніякі дані або яких взагалі не має бути. Це може свідчити про несанкціоновані дії з вебресурсом;

періодична перевірка директорій на сервері вебресурсу, із метою виявлення підозрілих файлів (пошук вебшелів), – зазвичай після зламу вебресурсів зловмисники залишають на сервері бекдори (вебшели) для віддаленого доступу до сервера сайта. Рекомендуємо періодично переглядати директорії вебзастосунку для пошуку таких бекдорів. Для цього можна використовувати спеціальні скрипти або банально перевіряти наявність нових файлів у директоріях. Виявлення створеного сторонніми особами файлу буде свідчити про злам вебресурсу та дає можливості для подальших дій із пошуку вразливостей, які було використано;

управління правами доступу – налаштуйте дозволи для файлів і каталогів. Розподіляйте права доступу до файлів на сервері й окремих розділів сайта, відповідно до завдань користувачів. Доцільно розмежувати розташування скриптів і програм, даних, призначених тільки для читання, та даних, призначених для зміни відвідувачами;

уникайте вразливих конфігурацій вебсервера – злам сайта починається зі збирання інформації про сервер. Приховування версій використовуваного ПЗ – це один з елементів убезпечення вебсервера. Знання версій цих програм може полегшити завдання зловмисника з пошуку відомих для цієї версії вразливостей і, як наслідок, досягнення основної мети – проникнення. Тому необхідно приховувати службові сторінки (наприклад phpinfo.php, temp.php, test.php.) і службову інформацію, що виводять у повідомленнях про помилки. Вимкніть непотрібні сервіси. Заблокуйте порти, що не використовують, налаштуйте міжмережевий екран та/або Web Application Firewall (WAF). Обмежте доступ до панелі адміністратора з мережі "Інтернет" і мереж загального користування. Регулярно змінюйте паролі доступу до сайта та сервера. Використовуйте захищені методи доступу до сервера для передавання файлів та управління ним (SFTP, SSH та ін.). Налаштуйте фільтрацію вхідних даних у вебформах. Регулярно здійснюйте резервне копіювання сайта та БД (якщо така наявна);

розмежування вебзастосунків – досить часто фахівці CERT-UA спостерігають розташування на одній віртуальній машині кількох вебресурсів, які не належать один до одного. Наприклад, вебсайт і стара версія

вебсайта або нова тестова версія. Стара версія не підтримується і має старі вразливості, тестова версія є недопрацьованою і також вразливою, водночас до них є доступ із мережі "Інтернет". Через уразливості цих вебресурсів зловмисники здобувають несанкціонований доступ до основного вебресурсу.

3.6. Рекомендації з безпечної організації доступу віддаленого до інформаційних ресурсів організації

Часто віддалена робота передбачає доступ працівників до інформаційних ресурсів організацій, розміщених у внутрішній мережі. Для цього використовують віртуальні приватні мережі VPN, програмне забезпечення для віддаленого доступу (AnyDesk, TeamViewer, RDP). Фахівці не рекомендують організовувати віддалену роботу за допомогою RDP без використання VPN із шифруванням.

Якщо в організації немає технічної можливості організувати віддалене підключення з використанням VPN із шифруванням необхідно дотримуватися таких правил:

Пароль до RDP має бути стійким.

Слід фільтрувати доступ до RDP. Визначити IP-адреси, із яких працівники організації працюють віддалено. Відфільтрувати доступ до комп'ютера із RDP за віддаленими IP-адресами працівників організації. Доступ з усіх інших IP-адрес слід заборонити. Це можливо реалізувати за допомогою брандмауера Windows.

Не рекомендовано надавати до директорій спільний доступ із мережі "Інтернет". Слід або фільтрувати доступ або взагалі заборонити. Працівники організації можуть здобути до неї доступ після підключення до RDP із внутрішньої мережі. Це можливо реалізувати за допомогою брандмауера Windows.

Потрібне постійне журналювання та моніторинг RDP-з'єднань. Працівники, відповідальні за інформаційну безпеку, мають періодично переглядати журнальні файли на наявність підозрілих записів (наприклад з'єднання працівників уночі).

Рекомендованим методом організації віддаленої роботи є використання віртуальних приватних мереж (VPN) із шифруванням. Є велика кількість комерційних і безкоштовних рішень. Одним із найпопулярніших є OpenVPN. Організація такого типу віддаленого доступу передбачає

наявність сервера, до якого приєднуються клієнти (працівники) за допомогою спеціально згенерованих сертифікатів, після чого їхній трафік переспрямовують до внутрішніх інформаційних систем.

Під час організації віддаленої роботи за допомогою VPN потрібно дотримуватися таких правил: налаштувати автентифікацію на VPN-сервері за допомогою сертифіката та пароля; VPN-сертифікати та паролі до них мають зберігати в захищеному середовищі. Якщо зловмисники здобудуть до них доступ, вони здобудуть доступ до мережі організації.

Здійснювати постійне журналювання та моніторинг з'єднань до VPN-сервера. Усі популярні VPN-сервіси мають функціонал журналювання подій. Працівники, відповідальні за інформаційну безпеку, мають періодично переглядати журнальні файли на наявність підозрілих записів (наприклад з'єднання працівників уночі).

Фільтрація доступу до VPN-серверу. Слід визначити IP-адреси, із яких працівники організації працюють віддалено. Потрібно відфільтрувати доступ до VPN-сервера за віддаленими IP-адресами працівників організації. Доступ з усіх інших IP-адрес заборонити.

Розмежування доступу працівників до внутрішніх ресурсів. Це можливо реалізувати за допомогою фаєрволів, мережевих екранів, віртуальних мереж.

Для розгортання VPN слід використовувати оновлене ліцензійне програмне забезпечення, завантажене з офіційних ресурсів.

Особливу увагу слід приділити безпеці домашніх персональних комп'ютерів працівників організації. Шкідливе програмне забезпечення часто розповсюджується з використанням фішингу та методів соціальної інженерії, що знижує пильність користувачів. Антивірусне програмне забезпечення може захистити персональний комп'ютер від вірусів та інших видів шкідливого програмного забезпечення, а також небажаного контенту в мережі "Інтернет". З огляду на це, працівникам варто дотримуватися кількох порад, щоб повною мірою скористатися перевагами антивірусного програмного забезпечення:

Слід увімкнути захист у режимі реального часу. Захист у режимі реального часу – це функція багатьох типів антивірусного програмного забезпечення, що відповідає своїй назві. Він автоматично сканує дані під час завантаження та в разі виявлення підозрілого вмісту блокує їх.

Потрібно сканувати зовнішні пристрої на наявність шкідливого програмного забезпечення. Багато користувачів використовують антивірусне програмне забезпечення лише для сканування фізичних дисків свого

персонального комп'ютера. Радимо сканувати всі флеш-носії на наявність шкідливого програмного забезпечення.

Як і все програмне забезпечення, антивірусне програмне забезпечення потребує своєчасних оновлень. Усі типи антивірусного програмного забезпечення працюють, використовуючи бази даних, які складаються з відомих загроз та їхніх відповідних індикаторів. У разі несвоєчасного оновлення ефективність виявлення актуальних загроз значно знижується.

Ще однією функцією, яку слід увімкнути, – це ведення журналу. Під час своєї роботи антивірусне програмне забезпечення автоматично буде реєструвати всі події – від звичайного сканування до виявлення шкідливого навантаження. Ця інформація може бути корисною в разі виявлення вектора зараження чи побудови схеми зараження. Журналювання може допомогти вам ідентифікувати шкідливе програмне забезпечення та його переміщення на вашому комп'ютері.

Деякі антивірусні програмні забезпечення більш ефективно захищають від загроз, ніж інші. Використовуйте антивірусне програмне забезпечення, що розповсюджується відомими вендорами у сфері кібербезпеки та яким довіряєте. Використовуйте ліцензійні рішення. Якщо не має змоги встановити антивірус на домашній комп'ютер, використовуйте вбудовані в ОС Windows функції захисту Windows Security та Microsoft Security Essentials.

Головна поширена проблема, яку допускають користувачі, – слабкий пароль адміністратора, який дозволяє здобути доступ до налаштувань домашнього WiFi. Саме використання встановленого за замовчуванням розробниками паролю може дати зловмисникам безпосередній контроль над WiFi-роутером. Приклад паролів за замовчуванням: 1111, root, user, admin тощо.

З огляду на попередній пункт, важливо розуміти, що WiFi без пароля робить вас уразливими. У такому разі необхідно встановити WPA2 шифрування та використовувати надійний пароль.

Так званий Broadcast SSID можна приховати в налаштуваннях роутера. Так ідентифікатор клієнтської мережі не буде видно стороннім особам. Це ускладнює можливий злам зловмисниками. Водночас під час підключення клієнту необхідно буде кожен раз уводити цей ідентифікатор.

Сучасні роутери можуть підтримувати різні протоколи, які використовують "розумні" пристрої. Так користувач стає потенційною жертвою зловмисників. Адже в цих пристроях можуть бути активними відкриті

вразливості. Якщо користувач не використовує цю можливість – відключіть UPnP.

Використовуйте оновлення версії вбудованого програмного забезпечення. Саме оновлення позбавляє користувачів від уразливостей, що стали відомими розробнику. Завдяки оновленням, розробник виправляє помилки, які дають можливість зловмиснику дістати дані із клієнтської мережі, здобути безпосередній доступ та управління.

Відключення функції WPS. Ця функція дозволяє без уведення пароля швидко підключитися до безпроводової мережі, тому її слід вимкнути.

3.7. Висновки

У дослідженні описано 20 типів кіберінцидентів, які, згідно з методиками CERT-UA [18], було об'єднано в 10 категорій (шкідливий уміст; шкідливий програмний код; збирання інформації зловмисником; спроби утручання; втручання; порушення доступності; порушення властивостей інформації; шахрайство; відома вразливість; інше), що допоможе користувачам виявити актуальні кіберзагрози.

Досліджено вісім видів найактуальнішого шкідливого програмного забезпечення й запропоновано методи захисту від кожного з них: резервне копіювання; запобігання розповсюдженню ШПЗ у мережі; запобігання запуску ШПЗ на пристроях; обмеження впливу ШПЗ; коректна робота під час ураження мережі ШПЗ. На підставі цього аналізу слід вибирати програмні методи захисту мережі.

Розглянуто 10 типів атак на вебресурси: уставлення інструкцій; некоректна автентифікація та управління сеансами; міжсайтове виконання сценаріїв (XSS); небезпечні прямі посилання на об'єкти; небезпечна конфігурація оточення; витік критичних даних; відсутність контролю за доступом до функціонального рівня; підроблення міжсайтових запитів (CSRF); використання компонентів із відомими вразливостями; небезпечні переадресування. Для кожного типу атак з'ясовано вразливості, що дозволяють їх здійснити, та запропоновано програмно-апаратні методи захисту.

Було розроблено рекомендації щодо захисту інформаційно-інфокомунікаційних мереж за віддаленої роботи установи: організації безпечного віддаленого доступу до інформаційних ресурсів компанії з робочого місця співробітника; ефективного використання антивірусного на домашніх персональних комп'ютерах; збереження чутливих даних/паролів; безпечного використання домашніх роутерів.

Зміст

НАУКОВЕ ВИДАННЯ

Аксак Наталія Георгіївна
Гризун Людмила Едуардівна
Щербаков Олександр Всеволодович та ін.

СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

Монографія

За загальною редакцією
д-ра екон. наук, професора В. С. Пономаренка

Самостійне електронне текстове мережеве видання

Відповідальний за видання *І. О. Ушакова*

Відповідальний редактор *О. С. Вяткіна*

Редактор *О. Г. Доценко*

Коректор *О. Г. Доценко*

План 2022 р. Поз. № 9-ЕНВ. Обсяг 271 с.

Видавець і виготовлювач – ХНЕУ ім. С. Кузнеця, 61166, м. Харків, просп. Науки, 9-А

Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру
ДК № 4853 від 20.02.2015 р.