МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра кібербезпеки

МОНІТОРИНГ ТА УПРАВЛІННЯ ІНФОРМАЦІЙНОЮ БЕЗПЕКОЮ

ОПОРНИЙ КОНСПЕКТ ЛЕКЦІЙ

для здобувачів освітньо-професійної програми
«Кібербезпека»
другого (магістерського) рівня вищої освіти
спеціальності «Кібербезпека та захист інформації»

Тернопіль – ЗУНУ
2024

Опорний конспект лекцій з курсу «Моніторинг та управління інформаційною безпекою» для здобувачів освітньо-професійної програми «Кібербезпека» другого (магістерського) рівня вищої освіти спеціальності «Кібербезпека та захист інформації» / Укл.: Касянчук М.М., Івасьєв С.В. – Тернопіль: ЗУНУ, 2024. – 21 с.

Відповідальний за випуск: Івасьєв С.В., к.т.н., доцент кафедри кібербезпеки

Рецензенти:

Загородна Н.В. к.т.н., доцент, завідувач кафедри кібербезпеки Тернопільського національного технічного університету імені Івана Пулюя

Батько Ю.М. к.т.н., доцент кафедри комп'ютерної інженерії факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету

Методичні вказівки розглянуті та схвалені на засіданні кафедри кібербезпеки, протокол № 5 від 8 листопада 2024 р.

ЗМІСТ

ТЕМА 1. ПОСТАНОВКА ЗАДАЧІ АНАЛІЗУ ЗАХИЩЕНОСТІ.

Аналіз захищеності (Security Assessment) – процес оцінки системи, мережі або додатку на предмет вразливостей і відповідності стандартам безпеки.

Вразливість (Vulnerability) – слабке місце в системі, що може бути використане для несанкціонованого доступу або шкоди.

VA (Vulnerability Assessment) – оцінка вразливостей системи без активного використання атак, мета – виявити слабкі місця.

Penetration Testing (PenTest) – контрольований тест на проникнення, імітація атаки з метою виявлення реальних ризиків.

Red Teaming – комплексне моделювання атак з боку «червоних команд» для перевірки готовності системи та персоналу.

Blue Teaming – захисна діяльність «синіх команд» щодо виявлення, реагування та усунення загроз.

Vulnerability Assessment (VA) – автоматизовані сканери, ручна перевірка конфігурацій.

Penetration Testing – тестування доступу до систем, соціальна інженерія, аналіз мережевого трафіку.

Red/Blue Teaming – комплексні сценарії атак та захисту, інтеграція з SOC та SIEM для навчання персоналу і перевірки процесів.

Міжнародні стандарти та фреймворки

Стандарт / Фреймворк	Опис	Офіційне посилання
ISO/IEC 27001:2022	Стандарт для систем управління інформаційною безпекою (ISMS), що визначає вимоги до встановлення, впровадження, підтримки та постійного вдосконалення ISMS.	iso.org/standard/27001
NIST Cybersecurity Framework (CSF 2.0)	Рамки для управління кіберризиками, що включають функції: ідентифікація, захист, виявлення, реагування та відновлення.	nist.gov/cyberframework
MITRE ATT&CK	Глобальна база знань про тактики та техніки атак, що допомагає в моделюванні атак та розробці методологій захисту.	attack.mitre.org

Законодавство України у сфері моніторингу інформаційної безпеки

Закон / Нормативний акт	Опис	Офіційне посилання
Закон України «Про захист інформації в інформаційно-телекомунікаційних системах»	Регулює відносини у сфері захисту інформації в ІТС, визначає вимоги до захисту інформації та відповідальність за порушення.	https://zakon.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80#Text
Закон України «Про захист персональних даних»	Встановлює вимоги до обробки персональних даних, права суб'єктів даних та обов'язки операторів даних.	https://zakon.rada.gov.ua/laws/show/2297-17#Text

Про внесення змін до деяких законів України щодо захисту інформації та кіберзахисту державних інформаційних ресурсів, об'єктів критичної інформаційної інфраструктури	Комплекс технічного захисту інформації. Авторизація з безпеки. Кіберзахист критичної інформаційної інфраструктури. Національна система реагування на кіберінциденти.	https://zakon.rada.gov.ua/laws/show/4336-20#Text

ТЕМА 2. ЗБІР ДАНИХ ДЛЯ МОНІТОРИНГУ БЕЗПЕКИ

Моніторинг безпеки – це постійний процес збору та аналізу інформації про події та стан системи з метою виявлення загроз та аномалій.

Сенсор безпеки – це пристрій або програмний модуль, який збирає інформацію про стан системи, мережі або застосунків для подальшого аналізу.

Network Tap (точка моніторингу мережі) – це апаратний пристрій, який дозволяє пасивно дублювати весь мережевий трафік без впливу на роботу мережі.

SPAN-порт (Switch Port Analyzer) – це спеціальний порт комутатора, через який відбувається дублювання трафіку для моніторингу.

Агенти – це програмні модулі, встановлені на хостах або пристроях, які збирають логи, метрики та мережевий трафік і передають їх у систему моніторингу.

Пасивний збір даних передбачає спостереження за системою та мережею без створення додаткового трафіку або впливу на об'єкти моніторингу.

Активний збір даних включає використання спеціальних запитів або тестів для отримання інформації про систему або мережу, наприклад, сканування портів або опитування SNMP.

Джерелами даних для моніторингу є журнали операційних систем, мережевий трафік, логи застосунків і сервісів, а також дані хмарних сервісів.

Системи SIEM (Security Information and Event Management) дозволяють централізовано збирати, корелювати та аналізувати події безпеки з різних джерел.

Центр управління безпекою (SOC) координує аналіз подій, реагування на інциденти та контроль за станом безпеки у масштабі всієї організації.

Для збору даних використовуються різні інструменти: Wireshark і tcpdump для аналізу трафіку, Zeek для виявлення аномалій, OSSEC або Wazuh для збору логів ОС, а також Fluentd, Logstash чи Filebeat для централізації логів у SIEM.

Для активного збору застосовують Nmap для сканування мережі та портів, SNMP Tools для опитування пристроїв, а Nagios, Zabbix або Prometheus – для моніторингу стану систем і сервісів.

У хмарних середовищах збір даних забезпечують AWS CloudTrail, Azure Monitor, Google Cloud Logging та інші платформи, які надають події та логи для аналізу безпеки.

Моніторинг безпеки – це постійний процес збору та аналізу інформації про події та стан системи з метою виявлення загроз та аномалій.

Сенсор безпеки – це пристрій або програмний модуль, який збирає інформацію про стан системи, мережі або застосунків для подальшого аналізу. Network Tap (точка моніторингу мережі) дозволяє пасивно дублювати весь мережевий трафік без впливу на роботу мережі. SPAN-порт (Switch Port Analyzer) – це спеціальний порт комутатора, через який відбувається дублювання трафіку для моніторингу. Агенти – це програмні модулі, встановлені на хостах або пристроях, які збирають логи, метрики та мережевий трафік і передають їх у систему моніторингу.

Пасивний збір даних передбачає спостереження за системою та мережею без

створення додаткового трафіку або впливу на об’єкти моніторингу. Активний збір даних включає використання спеціальних запитів або тестів для отримання інформації про систему або мережу, наприклад, сканування портів або опитування SNMP.

Джерелами даних для моніторингу є журнали операційних систем, мережевий трафік, логи застосунків і сервісів, а також дані хмарних сервісів. Системи SIEM (Security Information and Event Management) дозволяють централізовано збирати, корелювати та аналізувати події безпеки з різних джерел. Центр управління безпекою (SOC) координує аналіз подій, реагування на інциденти та контроль за станом безпеки у масштабі всієї організації.

Для збору та аналізу мережевого трафіку використовують Wireshark та tcpdump для захоплення і перегляду пакетів, Zeek для виявлення аномалій і Kismet для моніторингу безпровідних мереж. Для збору логів ОС та подій безпеки застосовують OSSEC або Wazuh, а також Fluentd, Logstash чи Filebeat для централізації логів у SIEM.

Для активного збору даних застосовують Nmap для сканування мережі та відкритих портів, SNMP Tools для опитування мережевих пристроїв, а Nagios, Zabbix і Prometheus дозволяють контролювати стан систем, сервісів та додатків.

У хмарних середовищах збір даних забезпечують AWS CloudTrail, Azure Monitor та Sentinel, Google Cloud Logging та Security Command Center, які надають події та логи для аналізу безпеки. Ці інструменти дозволяють організувати централізований моніторинг та інтеграцію з SIEM і SOC для повного управління подіями безпеки.

ТЕМА 3. ВИЗНАЧЕННЯ ТОПОЛОГІЇ ТА ІДЕНТИФІКАЦІЯ МЕРЕЖЕВИХ ОБ’ЄКТІВ.

Визначення топології мережі – це процес картографування її структури, включаючи вузли, маршрутизатори, комутатори та інші пристрої, щоб зрозуміти, як дані передаються між ними. Картографування мережі дозволяє виявляти потенційні точки вразливості та забезпечує основу для подальшого моніторингу безпеки.

Ідентифікація мережевих об’єктів – це процес визначення типу, функцій та характеристик пристроїв у мережі. Для цього використовують різні протоколи та інструменти, які дозволяють отримати інформацію про вузли, відкриті порти та служби, що на них працюють.

Протокол ICMP використовується для перевірки доступності вузлів і відстежування маршрутів даних. Команди ping та traceroute дозволяють визначати, які маршрути проходять пакети, і виявляти затримки чи втрати пакетів у мережі.

Протокол ARP застосовується для визначення відповідності IP-адрес фізичним MAC-адресам у локальній мережі, що допомагає ідентифікувати пристрої та виявляти можливі конфлікти адрес.

Сучасні інструменти, такі як Nmap і Masscan, дозволяють виконувати сканування мережі для визначення активних хостів, відкритих портів та сервісів. Вони також можуть використовуватися для виявлення активності IDS/IPS або налаштувань фільтрації трафіку.

Виявлення фільтрації та систем IDS/IPS є важливим етапом, оскільки ці системи можуть блокувати або сповільнювати трафік, що може вплинути на роботу мережі та результати моніторингу. Моніторинг їхньої активності дозволяє оцінити ефективність заходів захисту та виявити потенційні обмеження у зборі даних.

Для цієї теми використовують такі інструменти: Nmap для детального сканування мережі та виявлення відкритих портів, Masscan для швидкого сканування великих сегментів мережі, а також стандартні утиліти ping, traceroute та arp для аналізу маршрутів і відповідностей адрес. Для візуалізації топології мережі можуть застосовуватися програми типу Zenmap (графічний інтерфейс для Nmap) або платформи мережевого моніторингу, які підтримують автоматичне картографування.

Ці інструменти дозволяють системно збирати дані про мережу, формувати карту топології та готувати інформацію для подальшого аналізу в рамках SIEM або SOC.

Протокол ICMP використовується для передачі службових повідомлень у мережі. У сфері моніторингу він дозволяє перевіряти доступність вузлів за допомогою команди ping, що

допомагає виявити активні пристрої в мережі. ICMP також дає змогу визначати маршрути передачі пакетів через traceroute, що допомагає картографувати мережу і виявляти вузли, де пакети затримуються або блокуються. Аналіз відповідей ICMP дозволяє виявляти аномалії у роботі мережі, наприклад втрату пакетів або затримки, які можуть сигналізувати про атаки або некоректну роботу мережевих пристроїв.

Протокол ARP використовується для відображення відповідності між IP-адресами і фізичними MAC-адресами пристроїв у локальній мережі. У моніторингу він допомагає ідентифікувати активні пристрої та створювати карту вузлів у локальній мережі. ARP також дозволяє виявляти конфлікти адрес або підозрілі зміни MAC-адрес, що може свідчити про атаки типу ARP spoofing. Завдяки цьому протоколу дані сенсорів та агентів, які відстежують пристрої і трафік, залишаються актуальними та точними.

Для протоколу ICMP використовують утиліти, які дозволяють перевіряти доступність пристроїв і маршрути передачі пакетів. Інструменти на зразок ping і traceroute дозволяють визначати, чи доступний вузол у мережі і через які проміжні вузли проходять пакети. У сучасних системах моніторингу ці утиліти інтегровані у більш складні платформи, наприклад у SIEM чи мережеві монітори, де результати перевірок зберігаються, аналізуються та корелюються з іншими подіями безпеки.

Для протоколу ARP використовують утиліти, які дозволяють відображати відповідність IP-адрес фізичним MAC-адресам та перевіряти стан таблиці ARP на пристроях. Це дає змогу виявляти активні пристрої у локальній мережі та контролювати можливі конфлікти або підозрілі зміни адрес. Спеціалізовані системи моніторингу, такі як Wazuh або Zeek, можуть автоматично опитувати ARP-таблиці і фіксувати зміни для аналізу аномалій і підозрілої активності.

Використання таких засобів дозволяє поєднувати традиційні мережеві утиліти з сучасними платформами моніторингу, що дає змогу автоматично збирати, обробляти та аналізувати дані про доступність вузлів, маршрути трафіку та відповідність адрес у мережі. Це підвищує точність моніторингу і допомагає швидше виявляти потенційні загрози та аномалії.

ТЕМА 4. ІДЕНТИФІКАЦІЯ СЕРВІСІВ, ДОДАТКІВ ТА ОС.

Ідентифікація сервісів і додатків дозволяє визначати, які служби працюють на вузлах мережі, і оцінювати їхню конфігурацію та потенційні вразливості. Для цього використовують методи сканування TCP та UDP, які дають змогу перевірити відкриті порти і виявити активні служби.

Service Fingerprinting та Banner Grabbing дозволяють отримати детальну інформацію про тип сервісу, його версію та налаштування, аналізуючи відповіді серверів на мережеві запити. Це допомагає зрозуміти, які додатки працюють на віддалених хостах, і виявляти потенційні слабкі місця для подальшого аналізу.

Ідентифікація операційних систем (OS Fingerprinting) дозволяє визначати тип і версію ОС на віддалених вузлах. Для цього використовують інструменти на кшталт Nmap та SinFP, які застосовують як активні, так і пасивні методи. Активне визначення ОС передбачає відправку спеціальних мережевих пакетів і аналіз відповіді, а пасивне спостереження за трафіком дозволяє визначати ОС без взаємодії з хостом.

Сучасні методи визначення ОС та сервісів поєднують активне сканування, аналіз банерів і поведінкові підходи, що дозволяє підвищити точність і мінімізувати ризик виявлення моніторингових дій. Завдяки цьому ідентифікація сервісів, додатків і ОС стає ключовим етапом підготовки до аналізу захищеності мережі та оцінки її безпеки.

Ідентифікація сервісів і додатків дозволяє визначати, які служби працюють на вузлах мережі, і оцінювати їхню конфігурацію та потенційні вразливості. Для цього використовують методи сканування TCP та UDP, які дають змогу перевірити відкриті порти і виявити активні служби.

Service Fingerprinting та Banner Grabbing дозволяють отримати детальну інформацію про тип сервісу, його версію та налаштування, аналізуючи відповіді серверів на мережеві

запити. Це допомагає зрозуміти, які додатки працюють на віддалених хостах, і виявляти потенційні слабкі місця для подальшого аналізу.

Ідентифікація операційних систем (OS Fingerprinting) дозволяє визначати тип і версію ОС на віддалених вузлах. Для цього використовують інструменти на кшталт Nmap та SinFP, які застосовують як активні, так і пасивні методи. Активне визначення ОС передбачає відправку спеціальних мережевих пакетів і аналіз відповіді, а пасивне спостереження за трафіком дозволяє визначати ОС без взаємодії з хостом.

Сучасні методи визначення ОС та сервісів поєднують активне сканування, аналіз банерів і поведінкові підходи, що дозволяє підвищити точність і мінімізувати ризик виявлення моніторингових дій.

Для практичного застосування використовують Nmap, який дозволяє сканувати порти, визначати сервіси та робити OS Fingerprinting. SinFP застосовують для точного визначення операційних систем за характерними особливостями мережевих стеків. Для збору банерів і перевірки версій сервісів можна використовувати Netcat або спеціалізовані сканери типу Nessus. Крім того, сучасні платформи SIEM і SOC інтегрують дані сканування, що дозволяє корелювати інформацію про відкриті порти, сервіси і ОС з іншими джерелами безпеки, підвищуючи ефективність моніторингу та аналізу загроз.

ТЕМА 5. МЕТОДИ ІДЕНТИФІКАЦІЇ ВРАЗЛИВОСТЕЙ. PASSIVE FINGERPRINTING.

Методи ідентифікації вразливостей дозволяють оцінювати стан безпеки систем і мереж та виявляти слабкі місця, які можуть бути використані зловмисниками. Passive Fingerprinting – це підхід, при якому інформація про ціль збирається без активного сканування, шляхом спостереження за існуючим трафіком, поведінкою пристроїв або публічно доступною інформацією. Цей метод дозволяє уникнути виявлення активністю моніторингу, що особливо важливо при оцінці безпеки у великих або чутливих мережах.

Банерні перевірки дають змогу отримати інформацію про сервіси та додатки, аналізуючи їхні відповіді на мережеві запити. За допомогою банерів можна визначити версії програмного забезпечення, налаштування служб і можливі вразливості, не створюючи великого навантаження на систему. Цей метод часто використовується у поєднанні з іншими пасивними технологіями для підвищення точності визначення сервісів.

Локальні перевірки систем проводяться безпосередньо на хості і дозволяють оцінити стан ОС, встановлені програми та конфігурацію служб. Вони включають аудит журналів подій, перевірку налаштувань безпеки та пошук відомих вразливостей. Цей підхід особливо ефективний для внутрішнього аудиту або тестових середовищ, де можна безпечно отримати повну інформацію про стан системи.

Аналіз TCP/IP стеку дозволяє визначати особливості роботи мережевих протоколів, які можуть бути використані для пасивного визначення ОС та сервісів. Різні реалізації стеку IPv4 та IPv6 мають характерні ознаки у поведінці пакетів, що дозволяє визначати тип і версію операційної системи без активного втручання. Також цей аналіз допомагає виявляти аномальні або підозрілі патерни трафіку, що можуть свідчити про атаки або неправомірні дії у мережі.

Для практичного збору і аналізу даних застосовують сучасні інструменти, такі як Wireshark, який дозволяє перехоплювати та детально аналізувати пакети мережевого трафіку, і Zeek, який автоматично обробляє трафік, виявляє аномалії і підготовлює структуровані дані для подальшого аналізу. Ці інструменти дозволяють проводити кореляцію подій, визначати потенційно вразливі вузли, виявляти незвичні шаблони трафіку і підозрілу активність користувачів чи пристроїв.

Завдяки методам Passive Fingerprinting і використанню сучасних засобів моніторингу, можна отримати цінну інформацію про мережеві об’єкти, сервіси та операційні системи без створення активного навантаження на систему. Це робить методи пасивного збору даних безпечними, ефективними і дозволяє організаціям проводити оцінку стану безпеки без ризику порушення роботи мережі.ефективним для оцінки стану захищеності.

ТЕМА 6. СКАНЕРИ ВРАЗЛИВОСТЕЙ. NESSUS, OPENVAS.

Сканери вразливостей дозволяють автоматично перевіряти системи та мережі на наявність відомих слабких місць, помилок конфігурації та потенційних загроз. Вони є ключовим інструментом для оцінки безпеки і допомагають організаціям виявляти вразливості до того, як ними можуть скористатися зловмисники.

Nessus – це комерційний сканер вразливостей, який дозволяє проводити детальне сканування серверів, робочих станцій, мережевих пристроїв і баз даних. Він використовує велику базу плагінів для визначення різних типів вразливостей, включно з помилками конфігурацій, відомими експлойтами та слабкими паролями. Nessus дозволяє проводити сканування за різними профілями, перевіряти відповідність стандартам безпеки (CIS, PCI DSS), а також аналізувати політики безпеки і генерувати звіти з пріоритизацією ризиків. Додатково він підтримує інтеграцію з платформами SIEM та SOC для централізованого моніторингу та кореляції подій безпеки.

OpenVAS – це відкритий сканер вразливостей, який надає подібні можливості і дозволяє проводити перевірку на відповідність до стандартів безпеки. Він підтримує активне і пасивне сканування, регулярно оновлює базу вразливостей через Greenbone Vulnerability Management (GVM), і дозволяє генерувати детальні звіти про виявлені проблеми. OpenVAS може виконувати сканування окремих хостів або цілих мереж, перевіряти конфігурації операційних систем, сервісів і веб-додатків, а також інтегруватися з іншими інструментами для централізованого збору і аналізу даних про безпеку.

Сканери вразливостей використовують як активні, так і пасивні методи. Активне сканування передбачає відправку запитів на хости і аналіз відповідей для виявлення слабких місць, а пасивне сканування дозволяє збирати інформацію про систему без створення великого навантаження на мережу. Поєднання обох методів дає змогу отримати повну картину стану безпеки і оцінити ефективність захисних заходів.

Інтеграція сканерів з платформами SIEM та SOC дозволяє централізовано контролювати виявлені проблеми, корелювати дані з іншими джерелами та оперативно реагувати на потенційні загрози. Це підвищує ефективність моніторингу і забезпечує системний підхід до управління вразливостями у корпоративному середовищі.

ТЕМА 7. МОВА ОПИСУ АТАК NASL.

NASL (Nessus Attack Scripting Language) – це спеціалізована мова сценаріїв, яка використовується для створення тестів і плагінів у сканері Nessus. Вона дозволяє автоматизувати перевірку систем і мереж на наявність відомих вразливостей, а також налаштовувати поведінку сканера під конкретні завдання.

За допомогою NASL можна описувати структуру атаки, визначати умови, при яких вона вважається успішною, і формувати результати для подальшого аналізу. Мова підтримує логічні конструкції, роботу з мережевими протоколами, аналіз відповідей серверів і обробку даних, що дозволяє створювати як прості перевірки на віддалений доступ або слабкі паролі, так і складні експлойти для тестування безпеки.

NASL активно використовується для створення плагінів, які оновлюються разом з базою Nessus, що дозволяє завжди мати актуальні перевірки на нові вразливості. Це робить мову важливим інструментом для адаптації сканера до конкретного середовища і специфічних завдань безпеки.

Наприклад, простий сценарій NASL може перевіряти відкритий порт 80 на веб-сервері і виявляти банери, які містять інформацію про версію сервера. У сценарії задається цільовий хост, порт і команда для отримання банера. Після отримання відповіді скрипт аналізує її і виводить повідомлення про версію сервера і можливі вразливості, які відповідають даній версії.

Використання NASL дозволяє фахівцям з безпеки створювати власні перевірки, адаптовані до конкретних мереж, служб і політик безпеки, а також інтегрувати результати в централізовані системи моніторингу та управління вразливостями, такі як SIEM або SOC.

```
# Простий NASL скрипт для перевірки банера веб-сервера на порту 80

include("compat.inc");

# Вказуємо порт для перевірки
port = 80;

# Перевіряємо, чи порт відкритий
if(get_port_state(port) != PORT_OPEN) exit(0);

# Відправляємо простий HTTP-запит для отримання банера
banner = open_sock_tcp(port);
send(banner, "HEAD / HTTP/1.0\r\n\r\n");
response = read_buf(banner, 1024);
close(banner);

# Аналізуємо банер та виводимо результат
if (response) {
   security_message("Веб-сервер відповів банером:\n" + response);
} else {
   security_message("Веб-сервер не відповів або банер не отримано.");
}
```

ТЕМА 8. ІНШІ ІНСТРУМЕНТИ АНАЛІЗУ (XSPIDER, BURP SUITE, NIKTO, SQLMAP).

Інструменти для аналізу вразливостей допомагають дослідникам безпеки та адміністраторам швидко оцінювати стан мережі, серверів і веб-додатків. XSpider, наприклад, дозволяє проводити комплексне сканування мереж і вузлів, збираючи інформацію про відкриті порти, працюючі сервіси та можливі слабкі місця. Він підтримує як локальні перевірки, так і віддалене сканування, що дозволяє створювати детальні звіти про стан безпеки корпоративної мережі.

Burp Suite застосовується для аналізу веб-додатків. Він дозволяє перехоплювати та модифікувати HTTP/HTTPS-запити, досліджувати поведінку сервера та тестувати захист від типових веб-атак, таких як XSS чи SQL-ін’єкції. Наприклад, під час тестування веб-форми на сайті можна виявити слабкі місця у валідації даних або неправильно налаштовані кукі та заголовки безпеки.

Nikto спеціалізується на скануванні веб-серверів і допомагає визначати відомі уразливі файли, небезпечні скрипти та конфігураційні помилки. За допомогою цього інструменту можна швидко отримати інформацію про сервер, наприклад про відкрите тестове середовище чи небезпечні HTTP-заголовки, що підвищує ризик атаки.

Nikto використовується через командний рядок і дозволяє швидко перевіряти веб-сервери на наявність відомих уразливостей. Наприклад, щоб просканувати сервер на базі HTTP, можна виконати команду, яка вказує адресу хоста і використовує стандартний набір тестів. Після виконання сканування Nikto виводить список знайдених слабких місць, таких як небезпечні скрипти, старі версії серверного ПЗ або відомі конфігураційні помилки.

Якщо потрібно зберегти результати сканування у файл для подальшого аналізу, можна вказати формат вихідних даних, наприклад HTML або CSV. Це дозволяє легко інтегрувати результати в звіти або імпортувати їх у системи управління вразливостями.

Nikto також підтримує сканування з використанням проксі або SSL, що дає змогу перевіряти веб-сайти, які працюють через HTTPS або знаходяться за проміжними серверами. Крім того, можна вказати конкретні порти або URL, щоб обмежити область сканування до певних додатків або ресурсів.

Приклад команд:

Скана серверу на стандартному порту HTTP:
nikto -h http://example.com

Скана з використанням порту 8080:
nikto -h http://example.com:8080

Збереження результатів у HTML-файл:
nikto -h http://example.com -o report.html -Format html

Використання HTTPS та проксі:
nikto -h https://example.com -useproxy http://127.0.0.1:8080

Ці приклади показують базові можливості Nikto. Інструмент дозволяє швидко оцінювати безпеку веб-серверів і готує інформацію, яку можна використовувати для подальшого аналізу в SIEM, SOC або при плануванні заходів щодо усунення вразливостей.

Sqlmap дозволяє автоматично виявляти та експлуатувати SQL-ін'єкції у веб-додатках. Він аналізує параметри запитів, перевіряє наявність вразливих полів і може демонструвати, які дані можна отримати з бази даних. Наприклад, при тестуванні форми логіну на веб-сайті sqlmap може показати, які таблиці та записи доступні через вразливість, що дозволяє оцінити серйозність проблеми.

Sqlmap – це інструмент для автоматичного виявлення та експлуатації SQL-ін'єкцій у веб-додатках. Він аналізує параметри запитів, перевіряє наявність вразливих полів і може демонструвати, які дані можна отримати з бази даних. Для початку сканування досить вказати URL веб-сторінки або форму з параметрами, які потрібно перевірити.

Наприклад, базова команда для перевірки наявності SQL-ін'єкції в параметрі id веб-сайту виглядає так:

sqlmap -u "http://example.com/page.php?id=1"

Інструмент автоматично аналізує запит і визначає, чи можна його використати для витягування даних з бази.

Sqlmap також дозволяє вказувати тип бази даних, якщо вона відома, щоб підвищити точність перевірки. Для збереження результатів або отримання більш детальної інформації можна включити опції для виведення даних у файл або виводу усіх тестів, які проводилися.

Наприклад, щоб отримати повний список таблиць і колонок у вразливій базі даних, можна використати команду:

sqlmap -u "http://example.com/page.php?id=1" --dbs

sqlmap -u "http://example.com/page.php?id=1" -D target_db --tables

sqlmap -u "http://example.com/page.php?id=1" -D target_db -T users --columns

Якщо потрібно отримати реальні дані з таблиць, наприклад логіни та паролі, sqlmap дозволяє це зробити через опцію --dump:

sqlmap -u "http://example.com/page.php?id=1" -D target_db -T users --dump

Sqlmap також підтримує перевірку через проксі, роботу з автентифікацією і HTTPS-запитами. Це робить його потужним інструментом для оцінки безпеки веб-додатків і демонструє, які дані можуть бути витягнуті через SQL-ін'єкції, що дозволяє організаціям вчасно усувати уразливості.

Всі ці інструменти можна використовувати як окремо, так і в комбінації для проведення комплексного аналізу безпеки. Результати їх роботи дозволяють створювати звіти, пріоритизувати ризики і планувати заходи щодо усунення вразливостей, а також інтегрувати їх у системи SIEM або SOC для централізованого моніторингу і контролю безпеки.

ТЕМА 9. ЛОКАЛЬНИЙ АНАЛІЗ ТА ЗАХИСТ СУБД.

Аналіз захищеності на рівні вузла — це детальна перевірка окремого серверу або

робочої станції з метою виявлення конфігураційних помилок, невиправлених вразливостей, так званих «слабких місць» у правах доступу та ознак компрометації. Такий аналіз поєднує аудит конфігурації, перевірку цілісності файлів, сканування локальних сервісів та пошук шкідливих артефактів, і його результати часто стають входом для робіт SOC або процесу управління вразливостями.

Для Unix/Linux-систем корисні інструменти, що роблять глибокий аудит конфігурації й безпеки: наприклад, Lynis запускають командою lynis audit system і отримують детальний звіт по налаштуваннях, сервісах і рекомендаціям. Для перевірки цілісності файлів застосовують AIDE, який створює базу контрольних сум і дозволяє пізніше виявити нетипові зміни; звичайно процес виглядає як ініціалізація бази aide --init і подальша перевірка aide --check. Для пошуку відомих руткітів і підозрілих бінарників використовують rkhunter або chkrootkit, які дають швидку індикацію можливих ознак компрометації.

Локальний пошук небезпечних дозволів і SUID/SGID-файлів дає просте, але ефективне уявлення про ризики: наприклад, команда find / -perm /4000 -type f 2>/dev/null покаже файли з SUID-бітами, які варто проаналізувати на предмет необхідності й коректності. Також важливо перевіряти автозапуски і cron-джоби — у Linux це systemctl list-unit-files і огляд /etc/cron.* — бо багато нападів залишають персистентні модулі саме там.

На Windows- вузлах аудит ведеться через аналіз подій (Windows Event Logs) і інструменти Sysinternals; для збору автозапусків зручно використовувати Autoruns від Sysinternals, а журнали подій переглядати через Get-WinEvent у PowerShell. Для централізованого збору логів і правил відповідності застосовують агенти — наприклад, Wazuh/OSSEC агент на Windows і Linux збирає події, перевірки цілісності та виконує кореляцію з правилами детекції.

Спеціалізовані локальні сканери вразливостей, такі як Assuria Auditor (якщо використовується у вашій інфраструктурі), дозволяють робити глибоку перевірку конфігурацій під конкретні політики і генерувати нормативні звіти; подібні інструменти часто мають модулі для перевірки БД, файлових систем і рівнів доступу. Для СУБД корисно виконати локальні перевірки налаштувань та прав (наприклад, аудит привілейованих облікових записів в Oracle/MSSQL/Postgres), а також застосувати спеціалізовані сканери і конфіг-чеки.

Аналіз журналів і тимчасових артефактів є ключовим: локальні логи системи, логи аутентифікацій (auth.log / Windows Security), журнали застосунків та крон-виконань дають картинку останньої активності. Інструменти на кшталт Logstash/Filebeat або Wazuh agent дозволяють відправляти ці логи до SIEM для кореляції. Для швидкого локального триажу корисно запускати пошук незвичних мережевих з’єднань через ss або netstat і аналізувати відкриті сокети та процеси, що їх тримають.

Практичні лабораторії з аналізу вузла можуть включати аудит конфігурації з Lynis, ініціалізацію AIDE і перевірку відхилень, сканування rkhunter, пошук SUID-бінарів, аналіз автозапусків і агрегування логів через Wazuh агент із подальшим оглядом в Kibana. Важливо відточувати методику безпеки: кожна зміна має документуватися, а дії тестувальника виконуватися тільки в дозволеному середовищі або з письмовим погодженням власника системи.

Завжди пам’ятайте про юридичну та етичну сторону локального аналізу: виконання локальних перевірок на чужих системах без дозволу є незаконним, а під час навчання необхідно використовувати лабораторні віртуальні машини або окремі тестові середовища.

Сканування мережі та портів

Nmap — гнучкий сканер хостів/портів, fingerprinting сервісів і ОС.

Masscan — дуже швидкий масовий сканер портів для великих мереж.

Сканери вразливостей

Nessus — комерційний сканер з великою базою плагінів і звітністю.

OpenVAS / Greenbone (GVM) — відкритий сканер вразливостей з оновлюваною базою.

Аналіз веб-застосунків

Burp Suite — інтерактивний проксі/фреймворк для тестування веб-додатків

(інтерцепція, модифікація запитів, сканер).

OWASP ZAP — відкритий аналог для перевірки веб-додатків.

Nikto — швидкий сканер веб-серверів на відомі проблеми.

Acunetix / Qualys WAS — комерційні рішення для глибокого сканування веб-додатків.

Інструменти для експлуатації / пентесту

Metasploit Framework — платформа для розробки та виконання експлойтів і тестів проникнення.

sqlmap — автоматизований інструмент для виявлення та експлуатації SQL-ін'єкцій.

Пакетний і поточний мережевий аналіз

Wireshark — інтерфейсний аналізатор мережевих пакетів для детального розбору трафіку.

tcpdump — консольний захоплювач пакетів.

Zeek (Bro) — потужна платформа для пасивного мережевого моніторингу та аналізу трафіку.

ntopng / nProbe — аналіз NetFlow/sFlow і статистики трафіку.

Бездротові мережі

Kismet — пасивний сканер і сенсор для Wi-Fi (rogue AP, survey, GPS).

Aircrack-ng — набір інструментів для тестування захисту Wi-Fi (захоплення handshake, інжекція).

IDS / IPS

Snort — сигнатурний IDS/IPS.

Suricata — високопродуктивний IDS/IPS з підтримкою багатьох форматів.

SIEM / SOC / централізований збір логів

Wazuh — HIDS + SIEM-платформа (агенти, кореляція, інтеграція з ELK).

Elastic Stack (Elasticsearch, Logstash, Kibana) — збір, індексація, візуалізація логів.

Splunk — комерційна SIEM/платформа для аналізу логів і подій.

Агенти та локальні інструменти

OSSEC — HIDS/лог-агент (відкритий).

Filebeat / Fluentd / Logstash — відправники/агенти логів до SIEM.

Auditd (Linux) — системний аудит подій.

Аудит конфігурацій та жорстка політика

Lynis — аудит безпеки Linux/Unix-систем.

CIS-CAT / Benchmarks (інструменти CIS) — перевірки відповідності CIS-бенчмаркам.

Перевірка цілісності, пошук руткітів

AIDE — інструмент для перевірки цілісності файлів.

rkhunter / chkrootkit — інструменти для первинної перевірки на руткіти.

Крейкінг паролів і тестування стійкості аутентифікації

Hashcat — GPU-прискорений крейкер хешів.

John the Ripper — універсальний інструмент для перебору паролів.

Форензика та елементи IR

The Sleuth Kit / Autopsy — інструменти для дискової криміналістики.

Volatility — аналіз пам'яті (memory forensics).

Контейнерна та хмарна безпека

Trivy — сканер вразливостей контейнерних образів і файлів.

Clair / Anchore — сканування образів контейнерів.

AWS CloudTrail / Azure Sentinel / GCP Security Command Center — логування і моніторинг у хмарі.

Інструменти для управління вразливостями і оркестрації

Vulnerability Management Platforms (Tenable.sc, GVM, Rapid7 InsightVM) — управління результатами сканувань, трекінг виправлень.

SOAR-інструменти (Cortex XSOAR, TheHive/Shuffle) — автоматизація реагування на інциденти і оркестрація.

ТЕМА 10. МЕТОДОЛОГІЇ ОЦІНКИ БЕЗПЕКИ (ETHICAL HACKING, RED/BLUE/PURPLE TEAMING).

Ethical Hacking — це систематичний підхід до перевірки захищеності системи шляхом імітації дій зловмисника, але з дозволу власника ресурсу і в межах погодженого обсягу робіт. Мета — знайти і віддати до виправлення вразливості до того, як ними скористаються реальні нападники, при цьому мінімізувавши ризик шкоди під час тестування.

Методологія етичного хакінгу зазвичай включає кілька послідовних етапів: планування і погодження (scope, ROE), розвідка і збір інформації, сканування та ідентифікація вразливостей, експлуатація (за потреби), пост-експлуатацію і збирання артефактів для підтвердження, а також підготовку звіту і рекомендацій. Кожен етап має чіткі правила і логіку, щоб тестування було керованим і відтворюваним.

На етапі планування важливо погодити обсяг робіт, часові вікна, контакти для ескалації і правила взаємодії — «rules of engagement». Правильно оформлений дозвіл і погоджені межі робіт зменшують юридичні ризики та дозволяють команді тестувальників діяти швидко й відповідально під час інцидентів.

Розвідка включає збір відкритих даних (OSINT), картографування мережі, ідентифікацію цілей і сервісів. На цьому етапі застосовують пасивні і активні методи — від аналізу публічних записів і банерів до сканування портів і сервісів. Результатом є карта атакувальних поверхонь і пріоритетна цільова модель.

Сканування та ідентифікація вразливостей дає набір «точних підозр» — це результат роботи автоматичних сканерів і ручних перевірок. Важливо комбінувати аутентифіковані (credentialed) і неаутентифіковані (non-credentialed) сканування, оскільки перші дають глибший погляд у внутрішній стан систем, а другі показують, що може бачити зовнішній нападник.

Експлуатація робиться лише за суворо погоджених умов і зазвичай у контрольованому середовищі. Навіть якщо експлойт успішний, команда документує кроки, збирає докази і повертає систему в попередній стан. Пост-експлуатація (pivoting, збір секретів, перевірка впливу) допомагає оцінити бізнес-ризики від компрометації.

Підготовка звіту є не менш важливою частиною хакінгу: звіт повинен містити підтвердження виявлених вразливостей, інструкції з відтворення, оцінку ризику (CVSS + бізнес-контекст), рекомендації з усунення та план перевірки виправлень. Хороший звіт орієнтований і на технічних фахівців, і на менеджмент.

Централізоване управління вразливостями — це процес, що починається з інвентаризації активів і закінчується підтвердженням виправлення. Ключові елементи — точний реєстр активів, регулярні і планові сканування, триаж і пріоритизація результатів, розподіл завдань по відповідальним особам, відстеження статусу виправлень і перевірка ремедіації.

Пріоритизація вразливостей повинна базуватися не тільки на CVSS, а й на бізнес-контексті: експлуатація на системах з критичними даними, відкриті мережеві інтерфейси або наявність експлойтів у вільному доступі підвищують пріоритет. Автоматизовані правила в системах управління вразливостями (Vulnerability Management Platforms) і SOAR-процеси допомагають прискорити триаж і відправку тасків до IT-підрозділів.

Інтеграція з SIEM/SOC робить управління вразливостями набагато ефективнішим: події SIEM можуть корелюватися з даними сканерів, щоб виявити експлуатацію відомої вразливості в реальному часі. Автоматизація (playbooks) дозволяє створювати тикети, надсилати повідомлення і навіть виконувати превентивні дії, наприклад тимчасове блокування IP чи вимикання вразливого сервісу.

Практично важливі аспекти — частота сканувань (регулярні і після змін), різниця між аутентифікованими та неаутентифікованими сканами, планові «вікна безпеки» для великих оновлень і контроль змін (change control), а також метрики ефективності: час від виявлення до виправлення (MTTR для вразливостей), кількість відкритих критичних вразливостей у часі, відсоток ремедіації в SLA.

Законність і етика лежать в основі практики: тестування без письмового дозволу заборонене, необхідно дотримуватись політик конфіденційності та збереження даних, а також

мати план резервного відновлення на випадок, якщо тест викликає збої. Для робіт з критичними системами рекомендується спочатку виконувати тестування у дзеркальних або тестових середовищах.

ТЕМА 11. КОНТРОЛЬ ЗАХИЩЕНОСТІ Wi-Fi.

Контроль захищеності безпровідних мереж починається з розуміння специфіки самої технології Wi-Fi. На відміну від дротових мереж, де доступ обмежений фізичним підключенням, у бездротових середовище розповсюджується у повітрі, і сигнал може перехоплюватися сторонніми особами навіть за межами будівлі. Це створює додаткові вектори атак і підвищує ризики.

Основою захисту Wi-Fi є протоколи шифрування і автентифікації. WEP вважається застарілим і небезпечним, WPA був кроком уперед, але теж має відомі слабкості. Сьогодні стандартом де-факто є WPA2, а найбільш захищеним — WPA3, який усуває частину вразливостей попередніх версій і застосовує сучасні криптографічні методи. Використання відкритих або слабкозахищених мереж піддає організацію суттєвим ризикам.

Перевірка безпеки бездротових мереж проводиться за допомогою спеціалізованих інструментів. Kismet дозволяє пасивно виявляти точки доступу, клієнтів та підозрілу активність у ефірі. Aircrack-ng використовується для тестування стійкості ключів, відновлення паролів і перевірки, наскільки легко можна зламати конкретну конфігурацію Wi-Fi. Використання цих інструментів у навчальних лабораторіях допомагає зрозуміти практичні аспекти атак і оборони.

Важливим етапом є виявлення несанкціонованих точок доступу, так званих rogue AP. Нападник може підняти фальшиву точку з ідентичним SSID, спровокувати користувачів на підключення і перехоплювати їхній трафік. Тому системи моніторингу повинні не лише бачити відомі корпоративні точки, але й виявляти підозрілі пристрої у радіоефірі.

Аналіз захищеності включає також перевірку стійкості до атак типу deauthentication або jamming. Ці атаки не завжди націлені на викрадення даних, але можуть створити відмову в обслуговуванні або змусити клієнтів переключитися на небезпечні точки доступу. Для моніторингу таких атак застосовуються як апаратні датчики, так і програмні агенти.

Пасивний спектральний і пакетний моніторинг — базовий спосіб контролю: сенсори в режимі monitor ловлять усі пакети в ефірі й зберігають їх для аналізу. Це дає змогу виявляти нові або неавторизовані SSID, підключені клієнти, аномалії в сигналі і підозрілі патерни (масові deauth, probe flood тощо) без активного втручання в роботу мережі. Пакети з сенсорів потім аналізують у Wireshark для детального розбору або через Zeek/Kismet для автоматичної генерації подій.

Активні інструменти для тестування міцності конфігурацій використовують контрольовані атаки: захоплення WPA/WPA2-handshake, перевірка стійкості паролів, перевірка налаштувань аутентифікації. Aircrack-ng та подібні утиліти дозволяють демонструвати слабкість ключів і перевіряти, наскільки на практиці можна зламати конкретну конфігурацію. У навчальних заняттях такі інструменти застосовують лише в тестових мережах і з письмовим дозволом.

Системи WIDS/WIPS (Wireless IDS/IPS) виконують постійний моніторинг ефіру і вміють не тільки сигналізувати про інциденти, а й виконувати превентивні дії: наприклад ідентифікувати rogue AP і блокувати його на рівні контролера або ініціювати політику на NAC. Комерційні WIPS-рішення часто інтегруються з контролерами провайдерів Wi-Fi і дають централізовану панель управління для швидкого реагування.

Аутентифікація і контроль доступу — критично важлива складова контролю: рішення на базі RADIUS/802.1X (і NAC-платформи) дозволяють забезпечити централізовану перевірку пристроїв, політики доступу і автоматичне сортування трафіку (гостьова мережа проти корпоративної). Ця інфраструктура дає змогу не лише авторизувати користувачів, але й застосовувати правила відключення або обмеження для підозрілих пристроїв.

Інструменти спектрального аналізу потрібні для діагностики інтерференції і джемінгу: апаратні аналізатори і програмні рішення показують, де в ефірі виникають

перешкоди, які частоти перевантажені, і допомагають правильно вибрати канали та потужність передавача. Це важливо, бо багато проблем, що виглядають як атаки, насправді викликані перешкодами від сторонніх пристроїв.

Розгортання розподілених сенсорів — ефективна практика для великих територій: легкі пристрої (наприклад Raspberry Pi з USB-адаптером у режимі monitor та Kismet drone) встановлюються по периметру і надсилають події в центральну систему. Таким чином отримують постійну картину радіоефіру і можуть корелювати події між зонами для швидкого виявлення атак або несанкціонованих точок доступу.

Інтеграція з SIEM/SOC є тим елементом, що перетворює локальні події в операційну інформацію. Події від WIDS/WIPS, сенсорів, контролерів та RADIUS надходять у SIEM, де корелюються з логами хостів і мережевими подіями; при виявленні критичних сценаріїв SOC аналізує інцидент і, запровадивши playbook, може автоматично створити інцидент, відправити завдання або ініціювати блокування через NAC чи контролер.

ТЕМА 12. ВИЯВЛЕННЯ АТАК НА БЕЗПРОВІДНІ МЕРЕЖІ

Виявлення атак у безпровідних мережах спирається на поєднання пасивного моніторингу ефіру і кореляції подій з інших джерел, тому сенсори, що слухають радіоефір, повинні працювати постійно і зберігати метадані (BSSID, MAC, SSID, канал, RSSI, таймінги запитів/відповідей). Аналіз великих потоків probe-запитів, масових deauth-повідомлень або частих змін SSID у просторі дає першу індикацію про можливу атаку, навіть якщо пакети самі по собі не несуть змістовних даних.

Типова деаутентифікаційна атака проявляється різким стрибком в кількості deauth/Disassoc-пакетів від одного або кількох MAC-адрес, з одночасним підвищенням частоти повторних підключень клієнтів і зниженням RSSI. Виявлення ґрунтується на порогових правилах і часових вікнах — коли кількість деаутів від одного джерела перевищує нормальний рівень за короткий інтервал, сенсор генерує alert і передає подію в SIEM для кореляції.

Evil-Twin або SSID-спуфінг легше виявити, порівнюючи атрибути мережі: якщо з’являється точка з тим самим SSID, але іншим BSSID, каналом чи низькою автентичністю (відсутність 802.1X), система повинна позначити її як підозрілу. Кореляція з RADIUS-логами допомагає зрозуміти: чи було масове падіння автентифікації, чи користувачі переходять на нову точку — це підказка для SOC, що має відкрити інцидент і відправити завдання на фізичну перевірку.

Атаки на 802.1X і автентифікацію проявляються аномаліями в RADIUS-логах: збільшення числа невдалих запитів, повторні повторні запити від однієї MAC-адреси або нечакайні тайм-аути під час EAP-сеансів. Інтеграція RADIUS-подій в SIEM дає змогу корелювати ці логи з подіями WIDS і виявляти спроби підбору, брютфорсу або relay-атаки на автентифікацію.

KRACK та інші атаки на протокол рівня шифрування часто важко виявити без глибинного аналізу handshake-послідовностей; тут корисний збір 4-way handshake metadata і аналіз відхилень у часових мітках або повторних спробах пересинхронізації. Сенсори, що зберігають pcap-фрагменти handshake, дають можливість детально розібратися при підозрі й підтвердити експлуатацію вразливості.

Виявлення клієнтських атак та пост-компрометаційних ознак опирається на одночасний аналіз мережевого трафіку з хост-логами: незвичні DNS-запити, з’єднання з відомими C2-сервери, або нетиповий обсяг трафіку з мобільного пристрою — усе це корелюється в SIEM і дає підставу для триажу інциденту. Wazuh/Host IDS може підсилити датчики ефіру, надсилаючи alert якщо на хості виявлені процеси, що створюють підозрілий вай-фай-трафік.

Для виявлення джемінгу або шуму корисний спектральний аналіз: апаратні сенсори, що вимірюють енергетичну щільність по каналах, дозволяють відрізнити штучний джем від природних перешкод. Раптове зниження SNR на кількох каналах одночасно або аномально висока зайнятість певної смуги вказують на активний вплив в ефірі і повинні сигналізувати

SOC про можливу атаку на наявність сервісу.

Підхід на основі підписів корисний для відлову відомих шаблонів атак (типові deauth-фрейми, певні патерни probe requests), але сучасні мережі виграють від поведінкового виявлення: моделі нормальної активності клієнтів і точок доступу дозволяють піднімати аномалії, навіть якщо вони не співпадають із заздалегідь відомим підписом. Поєднання обох підходів у SIEM підвищує покриття і зменшує фейкові спрацювання.

Практично важливо мати готові playbook-процедури: при виявленні підозрілої точки SOC повинен мати алгоритм перевірки, ізоляції, блокування через контролер або NAC і ескалації на інженерів. Автоматизація відповіді (наприклад тимчасове блокування BSSID або обмеження доступу) полегшує реакцію, але її слід використовувати обережно, щоб уникнути false positive-блокувань у робочій мережі.

З точки зору даних, корисно збирати не лише події, а й контекст: геолокацію сенсора, RSSI, часові інтервали, історію появи SSID, відповідні записи RADIUS і DHCP. Такий контекст дозволяє відрізнити, наприклад, мобільний користувач, що рухається крізь кампус, від активного атакуючого пристрою. SIEM-панелі з дашбордами і timeline-переглядом прискорюють роботу аналітика SOC.

Не забувайте про етику і законність: тестування контрзаходів, ін’єкції чи активні перевірки робіть лише в тестовому середовищі або за письмовим дозволом. Впровадження виявлення і повязаних реагувань треба документувати у внутрішніх процедурах і узгоджувати з політиками безпеки, щоб уникнути юридичних і оперативних проблем.

ТЕМА 13. ДЖЕРЕЛА ДАНИХ ДЛЯ IDS/IPS.

Система виявлення атак найефективніше працює, коли отримує дані з різних рівнів інфраструктури: мережеві потокові дані та перехоплений трафік дають уявлення про рух пакетів і поведінку протоколів, а журнали з хостів показують, що відбувається всередині системи — процеси, аутентифікації і системні помилки.

Логи додатків, веб-серверів і баз даних є критично важливими, бо в них видно спроби авторизації, помилки запитів і специфічні ознаки атак на прикладному рівні; одночасно дані про аутентифікацію з RADIUS/AD дають контекст, хто саме і звідки намагається увійти, а записи DHCP/DNS допомагають зв’язати IP із хостом чи пристроєм.

Потокові дані NetFlow/sFlow/IPFIX і метадані Zeek/Suricata дозволяють швидко визначати аномалії в обсягах і патернах трафіку, виявляти C2-зв’язки і масштабні сканування, тоді як повні pcap-записи корисні для форензики і підтвердження інцидентів при триажі.

Ендпоінтна телеметрія (логі EDR, індикатори виконання, знімки процесів і мережевих з’єднань) відкриває видимість у поведінку кінцевих точок і дозволяє виявляти пост-компрометаційні ознаки, які мережа сама по собі може не помітити.

Хмарні сервіси і платформи SaaS генерують власні логи: CloudTrail, CloudWatch, Azure Monitor або журнали GCP містять сліди дій адміністратора, зміни прав і аномальні API-виклики, і ці джерела потрібно інтегрувати до загального потоку подій.

Спеціалізовані джерела теж важливі — сигнатури IDS/IPS, події контролерів Wi-Fi, SNMP-триґери та журнали мережевих пристроїв дають технічні індикатори, а дані від сканерів вразливостей і threat-intelligence додають контекст щодо відомих слабких місць і IOC (indicator of compromise).

Щоб система працювала коректно, потрібно стежити за якістю даних: синхронізовані часові позначки, валідність форматів, нормалізація полів і збереження контексту (наприклад, прив’язки IP→MAC→хост) роблять кореляцію точною і дозволяють мінімізувати хибні спрацювання.

Архітектурно дані збираються через сенсори та агенти, через Network TAP або SPAN-порти, і передаються колекторами й форвардерами (Filebeat, Logstash тощо) у SIEM/кореляційний рушій, де відбувається нормалізація, збагачення даними з threat-intelligence і застосування правил детекції.

Пам’ятайте про зберігання й політики ретеншену: зберігати сирі pcap-файли довго дуже дорого, тому варто комбінувати короткострокове збереження повних записів і

довгострокове архівування лише метаданих і критичних подій. Також питання конфіденційності і законності збору логів має бути врегульовано політиками організації та відповідними дозволами.

ТЕМА 14. МЕТОДИ ВИЯВЛЕННЯ АТАК

Признаки атак у сфері моніторингу інформаційної безпеки проявляються у вигляді відхилень від звичайної поведінки систем і користувачів. Це можуть бути часті помилки автентифікації, раптові зміни у патернах мережевого трафіку, незвичні DNS-запити чи підозрілі звернення до зовнішніх IP-адрес. Навіть дрібні зміни в журналах системи або додатків можуть вказувати на спробу експлуатації вразливості. Саме ці сигнали стають базою для виявлення інцидентів, коли вони правильно корелюються і аналізуються.

Методи виявлення атак поділяються на кілька підходів. Традиційний сигнатурний аналіз дозволяє зіставити відомі зразки шкідливої активності з подіями у системі. Це швидкий і надійний метод для виявлення відомих атак, але він не здатен виявити нові загрози. На противагу йому поведінковий аналіз базується на побудові моделей звичайної активності користувачів і систем, щоб фіксувати відхилення від норми. Цей підхід допомагає знаходити раніше невідомі або складні атаки, зокрема атаки нульового дня.

Сучасні системи часто комбінують ці два підходи з використанням евристики та машинного навчання. Наприклад, IDS може сигналізувати про підозрілу кількість ICMP-запитів, а SIEM — автоматично перевірити, чи це пов’язано з легітимним тестуванням або з ознаками сканування. Така багаторівнева інтеграція дозволяє точніше визначати справжні інциденти та знижувати кількість хибних спрацювань.

У практичному вимірі методи виявлення атак спираються на різні джерела даних. Аналіз мережевого трафіку допомагає побачити спроби сканування портів, перебору паролів або використання незвичних протоколів. Журнали операційних систем показують дії користувачів і процесів, що дозволяє виявляти спроби підвищення привілеїв або використання небезпечних команд. Логи додатків, наприклад веб-серверів, дають змогу фіксувати SQL-ін’єкції чи спроби обходу автентифікації.

Wazuh дозволяє централізовано збирати і аналізувати події з різних джерел, перетворюючи їх у корисну інформацію для виявлення атак. Він приймає логи з хостів, мережевих пристроїв, додатків і систем безпеки, нормалізує їх і порівнює з набором правил, щоб визначити ознаки підозрілої активності.

При спробах брутфорсу на сервері Wazuh аналізує журнали автентифікації і може підняти тривогу, якщо кількість невдалих входів перевищує заданий поріг за короткий проміжок часу. Крім простого лічильника, система враховує IP-адресу, користувача та контекст попередніх подій, що дозволяє відокремлювати випадкові помилки від реальної атаки.

Виявлення шкідливого програмного забезпечення також можливо завдяки аналізу змін у файловій системі і процесах. Wazuh відстежує зміни у системних каталогах, нові або модифіковані виконувані файли та сумнівні запуски процесів. Якщо ці події відповідають сигнатурам шкідливих програм, генерується інцидент і відправляється alert у SIEM або SOC.

Для мережевих атак Wazuh інтегрується з IDS/IPS, такими як Suricata чи Snort. Система приймає їх події, нормалізує і корелює з іншими джерелами: наприклад, якщо Suricata зафіксувала сканування портів одночасно з підозрілими спробами входу на хості, Wazuh може згенерувати високопріоритетний інцидент.

Wazuh також дозволяє виявляти атаки на веб-додатки через інтеграцію з журналами веб-серверів. Система аналізує HTTP-запити, виявляє шаблони, характерні для SQL-ін’єкцій, XSS чи LFI/RFI. Комбінація цих даних з поведінковим аналізом користувачів дає змогу відокремити легітимні запити від потенційних атак.

При виявленні атак на безпровідні мережі Wazuh може обробляти дані WIDS/WIPS або спеціалізованих сенсорів. Події про масові deauth-пакети, появу rogue AP або аномальні підключення клієнтів потрапляють у систему, де вони корелюються з іншими логами, наприклад RADIUS чи DHCP. Таким чином аналітик SOC отримує повну картину інциденту і може приймати обґрунтовані рішення щодо реагування.

Ключовим елементом є кореляція подій і створення правил, які відповідають реальним загрозам організації. Wazuh підтримує кастомні правила і інтеграцію з threat intelligence, що дозволяє виявляти атаки нульового дня або підозрілі патерни, навіть якщо вони не відповідають жодному відомому підпису.

Wazuh дозволяє централізовано збирати і аналізувати події з різних джерел, перетворюючи їх у корисну інформацію для виявлення атак. Він приймає логи з хостів, мережевих пристроїв, додатків і систем безпеки, нормалізує їх і порівнює з набором правил, щоб визначити ознаки підозрілої активності.

При спробах брутфорсу на сервері Wazuh аналізує журнали автентифікації і може підняти тривогу, якщо кількість невдалих входів перевищує заданий поріг за короткий проміжок часу. Крім простого лічильника, система враховує IP-адресу, користувача та контекст попередніх подій, що дозволяє відокремлювати випадкові помилки від реальної атаки.

Виявлення шкідливого програмного забезпечення також можливо завдяки аналізу змін у файловій системі і процесах. Wazuh відстежує зміни у системних каталогах, нові або модифіковані виконувані файли та сумнівні запуски процесів. Якщо ці події відповідають сигнатурам шкідливих програм, генерується інцидент і відправляється alert у SIEM або SOC.

Для мережевих атак Wazuh інтегрується з IDS/IPS, такими як Suricata чи Snort. Система приймає їх події, нормалізує і корелює з іншими джерелами: наприклад, якщо Suricata зафіксувала сканування портів одночасно з підозрілими спробами входу на хості, Wazuh може згенерувати високопріоритетний інцидент.

Wazuh також дозволяє виявляти атаки на веб-додатки через інтеграцію з журналами веб-серверів. Система аналізує HTTP-запити, виявляє шаблони, характерні для SQL-ін’єкцій, XSS чи LFI/RFI. Комбінація цих даних з поведінковим аналізом користувачів дає змогу відокремити легітимні запити від потенційних атак.

При виявленні атак на безпровідні мережі Wazuh може обробляти дані WIDS/WIPS або спеціалізованих сенсорів. Події про масові deauth-пакети, появу rogue AP або аномальні підключення клієнтів потрапляють у систему, де вони корелюються з іншими логами, наприклад RADIUS чи DHCP. Таким чином аналітик SOC отримує повну картину інциденту і може приймати обґрунтовані рішення щодо реагування.

Ключовим елементом є кореляція подій і створення правил, які відповідають реальним загрозам організації. Wazuh підтримує кастомні правила і інтеграцію з threat intelligence, що дозволяє виявляти атаки нульового дня або підозрілі патерни, навіть якщо вони не відповідають жодному відомому підпису.

Таким чином, Wazuh є універсальним засобом для централізованого виявлення атак, кореляції даних з різних джерел і автоматизації реагування у рамках SOC. Він дозволяє поєднати поведінковий і сигнатурний підходи, забезпечуючи видимість подій на рівні хостів, мережі та застосунків.

ТЕМА 15. ІНТЕГРАЦІЯ ЗАСОБІВ У ЄДИНУ СИСТЕМУ.

Сучасні організації стикаються з величезними обсягами подій, що надходять з операційних систем, мережевих пристроїв, додатків та засобів захисту. Для їх обробки та виявлення загроз використовується інтеграція засобів у єдину систему, яка реалізується на базі SIEM-рішень.

Системи класу SIEM, такі як Splunk, ELK чи IBM QRadar, виконують централізований збір і нормалізацію логів, забезпечують кореляцію подій та побудову зрозумілих дашбордів для аналітиків SOC. Splunk відомий потужним механізмом пошуку та візуалізації даних, ELK-стек є гнучким і з відкритим кодом рішенням, тоді як QRadar інтегрує SIEM з розвинутим механізмом управління інцидентами та підтримкою великих корпоративних мереж.

Окрім класичного збору та аналізу подій, все більшого значення набувають системи SOAR, які дозволяють автоматизувати реагування на інциденти. SOAR забезпечує створення сценаріїв (playbooks), які автоматично виконують дії у відповідь на події. Наприклад, при

виявленні масових невдалих спроб входу SOAR може автоматично заблокувати IP-адресу на фаєрволі, створити інцидент у системі управління заявками та повідомити відповідальну особу. Це значно знижує час реагування на загрози.

Ключовим елементом інтегрованих систем є кореляція даних. Вона дозволяє пов'язати події з різних джерел у єдиний інцидент, наприклад з'єднати факт сканування портів, невдалі спроби входу та спробу запуску підозрілої програми на сервері в одну атаку. Завдяки цьому аналітик отримує цілісне уявлення про ланцюжок дій зловмисника та може оцінити рівень ризику для організації.

Побудова SOC (Security Operations Center) передбачає організацію процесів моніторингу, виявлення та реагування на інциденти в реальному часі. SOC базується на SIEM та SOAR, інтегрує IDS/IPS, системи моніторингу мережі, захисту кінцевих точок, WAF, DLP та інші рішення. У результаті формується єдиний центр управління безпекою, який забезпечує як технічну, так і організаційну підтримку інформаційної безпеки.

Таким чином, інтеграція засобів у єдину систему дозволяє ефективно керувати інформаційною безпекою, скорочувати час на виявлення і реагування, а також будувати комплексну архітектуру SOC, що відповідає сучасним викликам кіберзагроз.

Рішення	Тип ліцензії	Основні можливості	Переваги	Недоліки	Приклади використання
Splunk Enterprise Security	Комерційна	Збір і нормалізація логів, потужний пошук SPL, аналітика, кореляція подій, інтеграція з SOAR	Швидкий пошук і візуалізація, масштабованість, велика спільнота	Висока вартість, складність налаштування	Великі корпорації, банки, телеком
ELK Stack (Elasticsearch, Logstash, Kibana) + Beats	Open Source (з комерційними опціями Elastic)	Централізований збір логів, збереження, пошук і візуалізація, гнучка інтеграція	Безкоштовна базова версія, висока гнучкість, велика кількість плагінів	Відсутність готових правил кореляції, потребує налаштування й адміністрування	Стартапи, університети, SOC на базі Open Source
IBM QRadar	Комерційна	SIEM + UEBA + управління інцидентами, інтеграція з IDS/IPS і фаєрволами, автоматизація	Вбудовані правила кореляції, зручний інтерфейс для SOC, інтеграція з IBM Security	Висока вартість, складна підтримка	Державні структури, фінансові установи, великі підприємства
Wazuh (як SIEM)	Open Source	Збір логів з агентів, HIDS, кореляція подій, інтеграція з OSSEC, підтримка cloud	Безкоштовний, інтеграція з ELK, простіше у розгортанні ніж "чистий ELK"	Менш розвинена аналітика ніж у Splunk/QRadar, потребує кастомізації	Невеликі компанії, освітні заклади, лабораторії SOC

Wazuh виступає як система збору даних і аналізу безпеки на хостах, а ELK-стек використовується для зберігання, пошуку і візуалізації цих даних.

Агенти Wazuh встановлюються на робочі станції, сервери та інші вузли. Вони збирають журнали подій операційних систем, інформацію про процеси, зміни у файловій системі, дані з антивірусів і системних служб.

Менеджер Wazuh отримує дані від агентів, нормалізує їх і застосовує правила кореляції. На цьому рівні вже можливо виявлення brute-force атак, підозрілих змін у конфігурації чи несанкціонованих дій користувачів.

Filebeat передає згенеровані алерти у Elasticsearch, де вони індексуються та зберігаються у вигляді структурованих даних. Це дозволяє швидко здійснювати пошук та аналіз.

Kibana використовується для візуалізації: на дашбордах SOC-аналітики бачать статистику подій, графіки атак, heatmap за IP-адресами, таймлайни інцидентів. Це значно полегшує моніторинг у реальному часі.

Wazuh може приймати події з IDS/IPS (Suricata, Snort), фаєрволів, SIEM-агентів чи cloud-сервісів. Це дає можливість створювати багаторівневу картину атак.

SOC-рівень реагування. На основі алертів в Kibana аналітики можуть створювати правила автоматичної реакції, наприклад блокування IP на фаєрволі чи відправку повідомлення у Slack/Telegram. У більш розвинених сценаріях Wazuh інтегрується з SOAR-рішеннями для повної автоматизації.