УДК 004.056
DOI: 10.31673/2409-7292.2023.020003

Білобровець І. В., Найман Г. Г.

ТЕХНОЛОГІЯ ВИЯВЛЕННЯ МЕРЕЖЕВИХ ЗАГРОЗ З ВИКОРИСТАННЯМ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ZABBIX

В статті проведено аналіз проблеми забезпечення інформаційної безпеки підприємства та необхідності застосування систем моніторингу. Проаналізовано сучасні існуючі системи моніторингу та принципу виявлення загроз. Доведено, що для виявлення загроз та підвищення інформаційної безпеки актуально використовувати програмне забезпечення Zabbix. Визначено призначення, характеристику та основні можливості Zabbix. Наведено приклади застосування програмного забезпечення для виявлення атак. На основі досліджень проведених в статті розроблено технологію виявлення загроз із застосуванням Zabbix і плагіну для нього. Показано ефективність застосування розробленої технології для виявлення загроз.

Ключові слова: інформаційна безпека, система моніторингу, атака, загроза, програмне забезпечення, Zabbix, плагін, технологія виявлення загроз.

Вступ

В даний час для забезпечення інформаційної безпеки важливе значення мають технології проактивного захисту інформації, націлені на запобігання порушенням інформаційної безпеки та здійснюють моніторинг, менеджмент інформаційної безпеки. Технології проактивного захисту ґрунтуються на своєчасному, оперативному зборі даних та метаданих про події безпеки, які фіксуються у записах журналів аудиту комп'ютерної інфраструктури, зберігання даних у спеціалізованому сховищі та подальшій обробці, включає процедури класифікації, кореляції, моделювання, вироблення попереджень та рішень щодо протидії атакам, а також інші найбільш ефективні оперативні процедури відновлення та надійне збереження безпеки інформації. Іншими словами, здійснюється управління інцидентами моніторинг інцидентів інформаційної безпеки.

Постановка проблеми

Останні десятиліття комп’ютерні мережі постійно стикаються з кіберзагрозами різної складності. Хакери викрадають конфіденційну інформацію. Головна складність для IT-фахівців - необхідність керувати багатоскладовою інфраструктурою, яка включає технології різного рівня від «стародавніх» до новітніх. За їх інтеграції часто виникають проломи, якими згодом користуються хакери. Наприклад, якщо в інфраструктурі організації використовуються системи, що більше не підтримуються виробником. Вищенаведені аргументи актуалізують тему даної статті, зміст якої становлять дослідження щодо можливості застосування програмного забезпечення Zabbix для виявлення загроз.

Мета статті – дослідити технологію виявлення загроз шляхом моніторингу систем з використанням програмного забезпечення Zabbix.

Архітектура Zabbix

Zabbix складається з кількох основних програмних компонентів:

1. Сервер. Zabbix сервер є основним компонентом, якому агенти повідомляють інформацію та статистику про доступність та цілісність. Сервер є основним сховищем, у якому зберігаються всі дані конфігурації, статистики, і навіть оперативні дані.

2. База даних. Як така вся інформація про конфігурацію, а також дані, зібрані Zabbix, зберігаються в базі даних.

3. Проксі. Для легкого доступу до Zabbix з будь-якого місця та будь-якої платформи, поставляється інтерфейс на основі Веб. Інтерфейс є частиною Zabbix сервера і зазвичай (але не обов'язково) працює на тій самій фізичній машині, що і сервер. При використанні SQLite Веб-інтерфейс Zabbix повинен бути запущений на цій же фізичній машині, що і сервер [1]. Zabbix проксі_може збирати дані про продуктивність та доступність від імені Zabbix сервера. Проксі є опціональною частиною Zabbix; однак він може бути корисним, щоб розподілити навантаження одного сервера Zabbix.

4. Агент. Zabbix агенти розгортаються на системах, що спостерігаються, для активного моніторингу за локальними ресурсами і додатками, і для відправки зібраних даних Zabbix серверу або проксі.

5. Потік даних. Крім того, важливо розглянути весь потік даних у Zabbix. Щоб створити елемент даних, який збиратиме дані, необхідно спочатку створити вузол мережі. Переміщаючись в інший кінець спектру Zabbix має бути елемент даних, щоб створити тригер. Далі вас бути тригер, щоб створити дію.

Таким чином, якщо необхідно отримувати сповіщення про занадто високе завантаження CPU на сервері X, спочатку необхідно створити запис про вузли мережі для сервера X, потім елемент даних для спостереження за CPU, потім тригер, який спрацює, якщо завантаження CPU буде занадто високим, а потім дія, яка відправить email. Хоча може здатися, що потрібно багато кроків, використання шаблонів значно спрощує процес. Однак така побудова системи дозволяє створювати дуже гнучкі інсталяції.

Zabbix сервер – центральний процес програмного забезпечення Zabbix. Функціонал базового сервера Zabbix розділений на три окремі компоненти; це: Zabbix сервер, веб-інтерфейс та сховище у базі даних. Будь-які зміни в веб-інтерфейсі Zabbix будуть відображені у розділі останніх даних із затримкою до двох хвилин.

Приклади параметрів командою рядка:

```
    shell> zabbix_server -c /usr/local/etc/zabbix_server.confshell>
zabbix_server –
    help
    shell> zabbix_server –V
```

Опції керування роботою сервера представлені у вигляді табл. 1.

Таблиця 1

Опції керування роботою сервера Zabbix

Опція	Опис	Ціль
	Перезавантаження конфігурації кешу. Ігнорується, якщо кеш вже завантажується зараз.	- Ідентифікатор процесу (1 до 65535) **тип процесу** - Усі процеси зазначеного типу (наприклад, **тип процесу, N** - Тип процесу та номер (наприклад, poller,3)
	Запуск процедури очищення бази даних. Ігнорується, якщо процедура очищення виконується на даний момент.	
log_level_increase[=<мета>]	Збільшення рівня журналювання діє на всі процеси, якщо мета не вказана.	
log_level_decrease[=<мета>]	Зменшення рівня журналування діє на всі процеси, якщо мета не вказана.	

Допустимий діапазон PID зміни рівня журналування одного процесу з 1 до 65535. На системах з великими значеннями PID опція <тип процесу,N> може використовуватися для зміни рівня журналування окремих процесів [1].

У зв'язку з вимогами безпеки та критично важливого характеру роботи сервера, UNIX є єдиною операційною системою, яка може забезпечити необхідну продуктивність, відмовостійкість та гнучкість. Zabbix працює із провідними на ринку версіями операційних систем. Zabbix сервер протестований на наступних платформах [1]: Linux; Solaris; AIX; HP-UX; Mac OS X; FreeBSD; OpenBSD; NetBSD; SCO Open Server; Tru64/OSF1.

Zabbix агенти розгортаються на цілях для активного моніторингу локальних ресурсів і додатків (статистика жорстких дисків, пам'яті, процесорів і т.д.). Агент локально збирає

оперативну інформацію та надсилає дані Zabbix серверу для подальшої обробки. У разі проблем (таких як відсутність вільного місця на жорсткому диску або аварійного завершення процесу сервісу), сервер Zabbix може швидко повідомити адміністраторів конкретного сервера, який повідомив про помилку. Zabbix агенти є надзвичайно ефективними, тому що використовують рідні системні виклики для збору інформації статистики. Zabbix агенти можуть виконувати пасивні та активні перевірки. В разі пасивної перевірки агент відповідає запит даних. Zabbix сервер (або проксі) запитує дані, наприклад, завантаження CPU, і агент Zabbix повертає результат. Активні перевірки вимагають складнішої обробки. Агент спочатку отримує список елементів даних для незалежної обробки від сервера Zabbix. Далі він періодично надсилатиме нові значення серверу. Вибір між пасивною та активною перевіркою здійснюється вибором відповідного типу елемента даних. Zabbix агент обробляє елементи даних типів 'Zabbix агент' та 'Zabbix агент (активний)'. Zabbix агент підтримується на наступних платформах [1]: Linux; Solaris; AIX; HP-UX; Mac OS X; FreeBSD; OpenBSD; NetBSD; SCO Open Server; Tru64/OSF1; Windows: Server 2008, 7, 8, 10, 11.

Параметри командного рядка, які можуть бути використані з агентом Zabbix, наведені у вигляді табл. 2.

Таблиця 2

Параметри командного рядка, які можуть бути використані з агентом Zabbix

Параметр	Опис
UNIX та Windows агент	
-c --config <файл-конфігурації>	Абсолютний шлях до конфігураційного файлу. Можна використати цю опцію, щоб задати файл конфігурації, розміщеному в папці відмінної від заданої за замовчуванням. У UNIX, шлях за промовчанням /usr/local/etc/zabbix_agentd.conf або як задано під час компіляції змінними --sysconfdir або –prefix. У Windows, шлях за промовчанням
	Виведення відомих елементів даних та вихід. Також для отримання результатів параметрів користувача, можна вказати файл конфігурації (якщо він знаходиться поза папкою, заданою за замовчуванням).
-t --test <ключ елемента даних>	Тестування зазначеного елемента даних та вихід. Також для отримання результатів параметрів користувача, можна вказати файл конфігурації (якщо він знаходиться поза папкою, заданою за замовчуванням).
	Виведення довідкової інформації.
	Виведення номера версії
Тільки для UNIX агента	
-R --runtime-control <опція>	Виконання адміністративних функцій.
Тільки для Windows агента	
	Використання кількох примірників агента (з -i,-d,-s,-x функціями). Для відділення імені примірників служб, кожне ім'я служби буде у значенні Hostvalue із зазначеного конфігураційного файлу.
Тільки для Windows агента (функції)	
	Установка Zabbix агента службою
	Видалення служби Zabbix Windows агента
	Запуск служби агента Zabbix Windows
	Зупинка служби агента Zabbix Windows

Спеціальні приклади використання параметрів командного рядка: відображення всіх вбудованих елементів даних із їх значеннями; тестування параметра користувача з ключем “mysql.ping” заданим у вказаному файлі конфігурації; інсталювати службу “Zabbix агента” у Windows, використовуючи шлях за промовчанням до файлу конфігурації c:\zabbix_agentd.conf; установка служби “Zabbix Agent [Hostname]” у Windows з використанням файлу конфігурації zabbix_agentd.conf, розміщеного в тій же папці що і бінарний файл агента та визначення унікального імені служби, використовуючи значення Hostname з файлу конфігурації:

```
    shell> zabbix_agentd –print
    shell> zabbix_agentd -t "mysql.ping" -c
/etc/zabbix/zabbix_agentd.conf
    shell> zabbix_agentd.exe –i
    shell> zabbix_agentd.exe -i -m -c zabbix_agentd.conf
```

Використовуючи опції адміністративних функцій, можна змінити рівень журналування процесів агента (табл. 3).

Таблиця 3

Опції адміністративних функцій агента Zabbix

Опція	Опис	Ціль
log_level_increase[=<мета>]	Збільшення рівня журналування. Діє на всі процеси, якщо ціль не вказана.	Ціль можна вказати за допомогою: - ідентифікатор процесу (від 1 до 65535) тип процесу - всі процеси вказаного типу (наприклад, poller) тип процесу, N - тип процесу та номер (наприклад, poller,3)

Zabbix проксі – це процес, здатний збирати дані моніторингу з одного або декількох пристроїв, що спостерігаються, і відправляти цю інформацію Zabbix серверу, таким чином проксі працює від імені сервера. Усі зібрані дані локально буферизуються і потім надсилаються Zabbix серверу, якому належить цей проксі. Розгортання проксі не обов'язково, але може бути дуже корисним для розподілу навантаження одиночного сервера Zabbix. Якщо проксі збирають дані, обробка цих даних на сервері навантажує CPU і I/O диска. Zabbix проксі - ідеальне рішення для централізованого моніторингу віддалених об'єктів, філій та мереж, де відсутні локальні адміністратори [2].

Для Zabbix проксі потрібна окрема база даних. Zabbix проксі підтримує наступні бази даних SQLite, MySQL та PostgreSQL. Можна також використовувати Oracle або IBM DB2. Починаючи з Zabbix 2.0 з'явився новий даемон Zabbix, званий Zabbix Java gateway, що забезпечує нативну підтримку моніторингу JMX додатків.

Zabbix Java gateway – це даемон, написаний мовою Java. Коли сервер Zabbix хоче знати значення конкретного JMX лічильника вузла мережі, він опитує Zabbix Java gateway, який використовуючи API управління JMX_опитує віддалений додаток. Додаток не потребує додаткового програмного забезпечення, він просто повинен бути запущений з опцією командного рядка Dcom.sun.management.jmxremote. Java gateway приймає вхідні підключення від сервера Zabbix або проксі і може бути використаний тільки як "пасивний проксі". Але, на відміну від Zabbix проксі, Java gateway може використовуватися з Zabbix проксі (тоді як один Zabbix проксі не може працювати через інший Zabbix проксі).

У Zabbix сервері та проксі є спеціальний тип процесів, які підключаються до Java gateway, їх кількість налаштовується опцією StartJavaPollers. Внутрішньо Java gateway

© Білобровець, І. В., & Найман, Г. Г. (2023). Технологія виявлення мережевих загроз з використанням програмного забезпечення Zabbix. Сучасний захист інформації, 2(54), 19–28. https://doi.org/10.31673/2409-7292.2023.020003.

запускається кількома потоками, що налаштовуються опцією START_POLLERS. На стороні сервера, якщо з'єднання займає більше Timeout секунд, воно буде завершено, але Java gateway може залишатися зайнятим отриманням значення JMX лічильника. Щоб вирішити цю проблему, Java gateway починаючи з Zabbix 2.0.15, Zabbix 2.2.10 та Zabbix 2.4.5 підтримують опцію TIMEOUT, що дозволяє вказати час очікування мережевих операцій JMX [2].

Zabbix сервер та проксі намагатимуться максимально об'єднати запити до однієї мети JMX (залежить від інтервалів оновлення елементів даних) та надсилати їх у Java Gateway за одне підключення для кращої продуктивності. Zabbix sender - це утиліта командного рядка, яка може бути використана для відправки даних про продуктивність сервера Zabbix для подальшої їх обробки. Зазвичай ця утиліта використовується в довгострокових скриптах для періодичної відправки даних про доступність і продуктивність. Починаючи з версії Zabbix 1.8.4, утиліта Zabbix_sender була покращена в плані відправлення даних у реальному часі. Мається на увазі, що велика кількість значень, отримана за короткий проміжок часу, накопичуватимуться у тимчасовому стеку і потім буде відправлено серверу за одне з'єднання. Дані, що прийшли через менше, ніж 0,2 с після попереднього значення накопичуються в одному стеку, але максимальний час їх накопичення до відправки все ж 1 секунда.

Є можливість вказати вхідний файл, який містить значення для відправки на сервер Zabbix. Сторінка допомоги по Zabbix sender_містить правила, як правильно форматувати записи у вхідному файлі в розділі --input-file. Нижче наведено приклади того, як значення будуть збережені в базі даних при використанні різних методів укладання в лапки (табл. 4).

Таблиця 4

Збереження даних Zabbix

Значення у вхідному файлі	**Результат у базі даних**	**Повідомлення про помилку на екрані**
		Warning: [line 1] 'Key value' required
“C:\My Documents”		

Нижче наведено приклад відправки 300 значень із вхідного файлу:

```
# zabbix_sender -z 127.0.0.1 -i /tmp/trapper.txt Info з
сервера: "Процісовано 250 Failed 0 Total 250 Seconds spent
0.002668"Info з сервера: "Processed 50 Failed 0 Total 50 Seconds
spent 00" skipped: 0; total: 300
```

Zabbix get – це утиліта командного рядка, яка підключається до Zabbix агенту і отримує від нього запитувану інформацію [3]. Утиліта зазвичай використовується для діагностики

агентів Zabbix. Приклад виконання Zabbix get UNIX для отримання значення завантаження процесора від агента:

```
shell> cd binshell> ./zabbix_get -s 127.0.0.1 -p 10050 -k
"system.cpu.load[all,avg1]"
```

Ще один приклад виконання утиліти Zabbix get для отримання рядка з веб-сайту:

```
shell> cd bin shell> ./zabbix_get -s 192.168.1.1 -p 10050 -k
"web.page.regexp[www.zabbix.com,,,USA: ,,\1]"
```

Можливості застосування Zabbix для виявлення загроз

Як вже зазначалося раніше, мережа може стати об'єктом DoS-атак чи спроб вторгнення, що є серйозним ризиком. Щоб вирішити цю проблему, ISO (Міжнародна організація зі стандартизації) розробила зведення правил, яким повинні дотримуватися користувачі, тим самим підвищуючи безпеку та якість операцій.

Відповідно до стандарту ISO ризик визначається як: ефект невизначеності цілей. Це пов'язано з потенціалом, за якого загрози можуть досліджувати вразливості активу і, отже, завдати шкоди. Наприклад, уразливості, виявлені на серверах, можна швидко виявити й усунути за допомогою інструментів моніторингу. У стандарті ISO/IEC 27005 розроблено елементи управління та рекомендації з моніторингу, з яких можемо виокремити таке: необхідний постійний моніторинг, щоб можна було виявити зміни. При попереджувальному моніторингу за допомогою Zabbix можна уникнути деяких ризиків інформаційної безпеки, як зазначено в [4].

Відома загроза серед команд безпеки – програми-вимагачі. Найпоширеніші способи поширення програм-вимагачів: експлуатація вразливості в невиправленій системі; через нефільтровані порти брандмауера. За допомогою Zabbix команда з інформаційної безпеки може активно виявляти програми-вимагачі, наприклад, контролюючи наявність останніх оновлень на сервері WSUS, налаштовуючи попередження для незвичайних відкритих портів, попереджаючи про виконання відомих процесів програм-здирників тощо.

У наступному прикладі, який наведено у [4], налаштовано Zabbix для надсилання попереджень, коли порти, які використовуються програмою-вимагачем wancray (9003, 9101 і 9001), відкриваються в брандмауері (рис. 1).

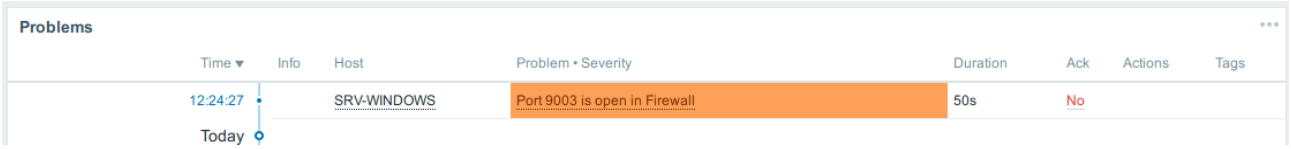

Рис. 1. Налаштування Zabbix для надсилання попереджень

Налаштувати таке сповіщення в Zabbix можна, використовуючи ключ net.tcp.port. Для цього потрібно перейти в "Конфігурація"> "Хост"> "Елемент" і натиснути "Створити елемент".

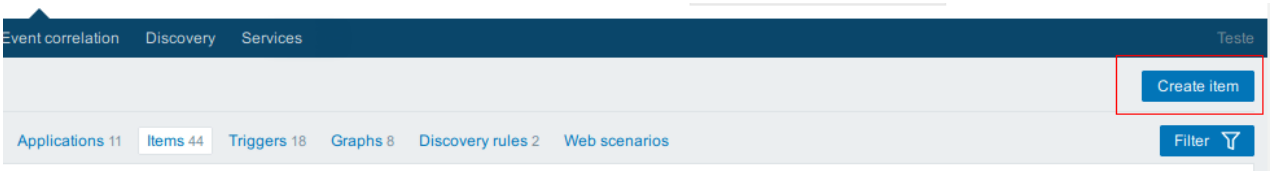

Рис. 2. Налаштування сповіщення в Zabbix

Далі необхідно задати ім'я, тип і ключ, як у прикладі, наведеному на рис. 3.

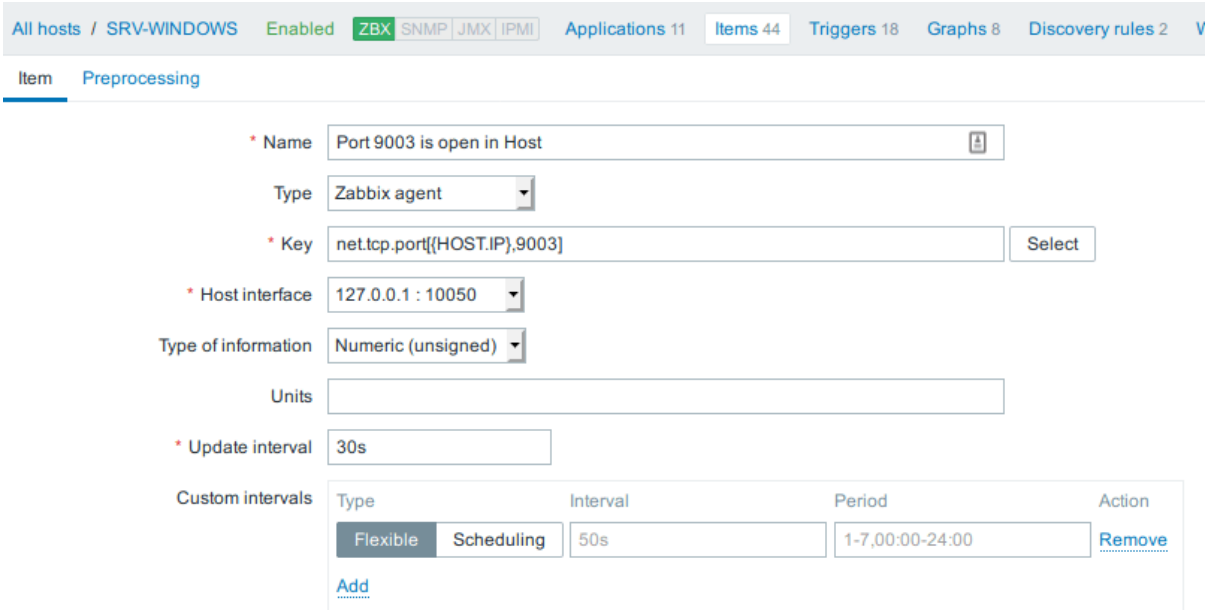

Рис. 3. Налаштування ім'я, типу і ключа

Далі необхідно натиснути «Додати».

Тепер потрібно створити тригер. У Host необхідно перейти до Trigger > Create Item. Встановити ім'я і вираз як показано на рис. 4.

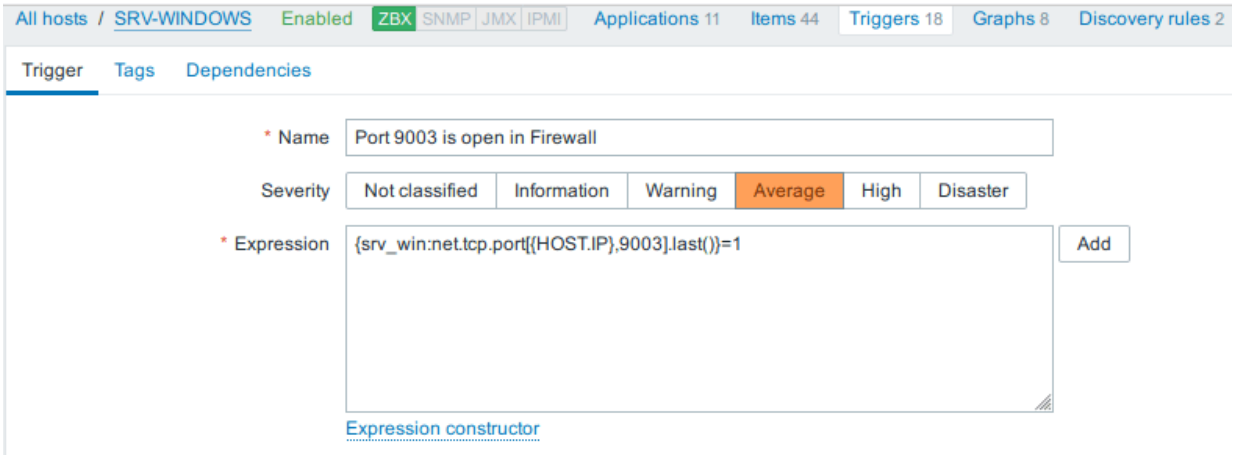

Рис. 4. Встановлення ім’я та виразу

У деяких випадках зловмисники змінюють важливі файли операційної системи, такі як журнали, паролі або конфігурації служб. Моніторинг цілісності файлів має першорядне значення, і Zabbix може вирішити цю проблему, перевіривши контрольні суми (рис. 5).

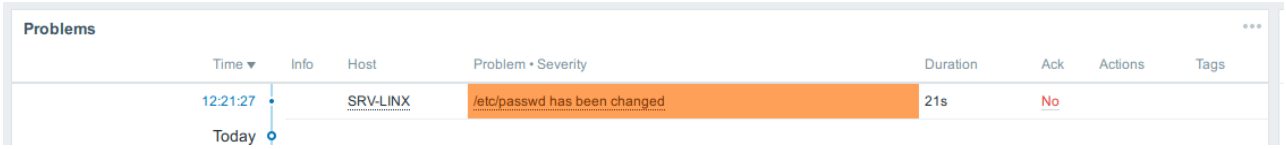

Рис. 5. Перевірка контрольних сум

Налаштування цього в Zabbix досить тривіальне за допомогою ключа vfs.file.cksum. Для цього просто необхідно перейти у "Конфігурація" > "Хост" > "Елемент" і натиснути "Створити елемент" [4].

Далі необхідно задати Name, Type і Key як у наступному прикладі (рис. 6).

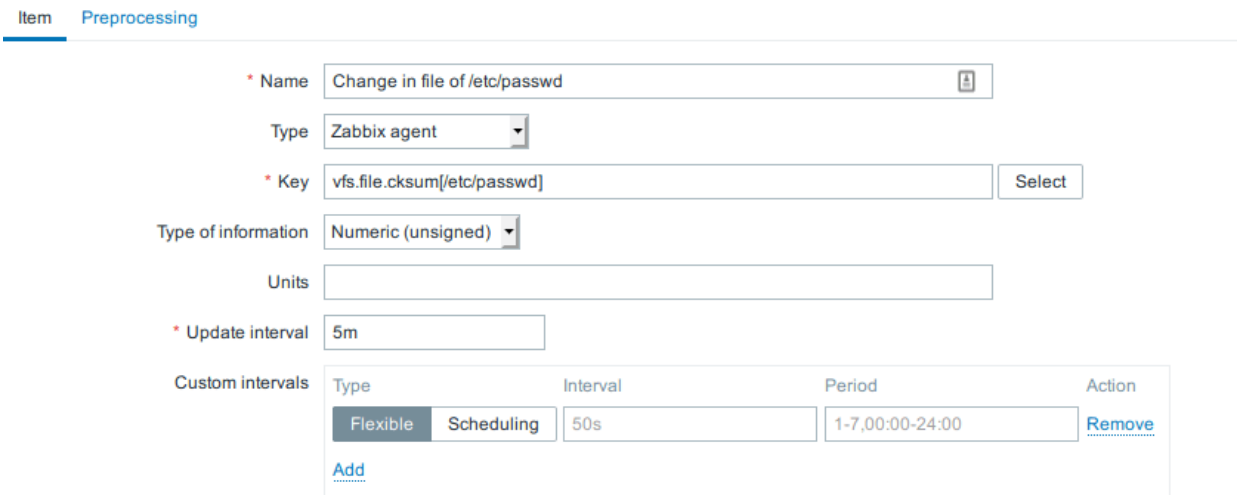

Рис. 6. Задання Name, Type і Key

Також потрібно буде створити тригер, тому необхідно перейти у "Тригер" > "Створити елемент". Далі можливо встановити Ім'я, Вираз, як у прикладі (рис. 7).

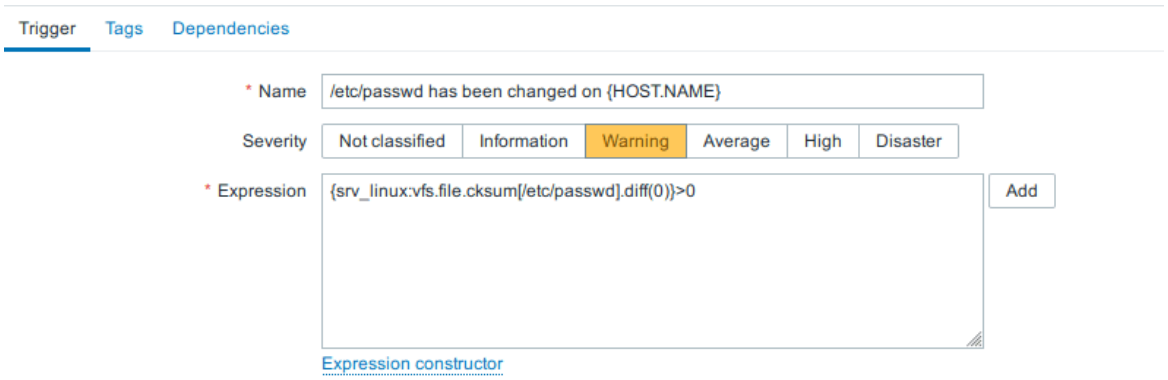

Рис. 7. Параметри налаштування Zabbix [4]

Як вказується в [4], можна почати з написання простих перевірок, як у наведених вище прикладах, а потім розвивати підхід у міру необхідності. Також зазначається, що, незалежно від того, чому віддається перевага - глибокому налаштуванню або готовим рішенням, Zabbix може застосовуватися для виявлення загроз. Щоб відповідати ISO 27005, деякі організації використовують Zabbix як найкраще рішення для моніторингу своїх серверів, брандмауерів, хмарних серверів і загальної інфраструктури. При цьому Zabbix підтримує співробітників інформаційної безпеки в ухваленні рішень і попереджувальному виявленні аномалій, які можуть виникнути в інфраструктурі.

Приклад застосування Zabbix

Розглянемо приклад, який наведено в [5]. В даній роботі відбувався експеримент, де жертвою став сервер V.Server з IP-адресою 178.128.119.51, а ZABBIX-Server – був сервер Zabbix з IP-адресою 209.97.164.25.

В роботі була проведена установка на сервер-жертву агента Zabbix, який мав збирати дані і відправляти дані на сервер Zabbix за допомогою sudo apt install zabbix-agent. Далі автори увійшли в Zabbix з обліковими даними користувача за замовчуванням і зробили необхідну зміну пароля. Активні перевірки дозволяють Zabbix збирати дані для моніторингу сервера-жертви. Налаштування «пасток» на обробку активних перевірок в Zabbix виконувалося командою шляхом запуску ловушок в файлі /etc/zabbix/zabbix_server.conf.

Для запуску атаки ping flooding було відкрито термінал і виконано команду ping з опціями -n і -i. Опція –n показала IP замість імені хоста, а опція -i задавала інтервал між успішними передачами пакетів. Для імітації атаки ping flood, яку необхідно було виявити, треба було відправити 10 і більше ping-запитів в секунду. Опція -i дозволила це зробити, встановивши її в 0,1. За допомогою команди ping -n –i 0.1 , 10 пінгів будуть відправлятися в секунду на ціль. Цю команду, однак, потрібно виконувати від імені користувача root, виконавши команду sudo su. Це виконає вимоги тригера для Zabbix, щоб показати пінги як можливу атаку переповнення пінгами.

Атака переповнення SYN була змодельована за допомогою інструменту hping3, доступного в дистрибутивах Linux. Щоб встановити hping3 в Ubuntu, було виконано команди sudo apt update і sudo apt install hping3. Для запуску атаки використано команду hping3 --flood --rand-source --destport --syn -d 120 -w 64. Виходячи з встановлених тригерів, будь-який IP відправляє 10 і більше пакетів з увімкненим прапором SYN і отримує від цілі пакет SYN+ACK без відправки ACK у відповідь, що вважається спробою SYN-флуду. Щоб перевірити, чи була спроба з'єднання потенційною атакою SYN flood, було використано netstat для перевірки стану сокета на наявність статусу SYN_RECV. Це вказує на те, що сервер отримав початковий SYN-пакет, відправив власний SYN+ACK-пакет, і чекає відповіді ACK від машини зловмисника. Коли сервер отримує велику кількість з'єднань, він чекає відповіді від клієнта, залишаючи з'єднання відкритими і виснажуючи ресурси сервера, що призводить до відмови в обслуговуванні.

Перший тест, виконаний у межах роботи [5] передбачав проведення атаки ping flooding, щоб спостерігати сплеск на графіку і проблему згенерованого на панелі управління Zabbix. Другий тест проводився шляхом проведення SYN flooding-атаки, і очікуваним результатом було те, що проблема з'явиться на Zabbix. Третій тест полягав в оцінці можливостей системи оповіщення під час ping та SYN flooding. Цей тест проводився шляхом запуску атак і фіксації часу початку атаки та часу отримання адміністратором оповіщення.

На сервер-жертву (178.128.119.51) було запущено три спроби ping-флуда. Zabbix зміг виявити ці атаки за тригером 10 і більше пінгів від одного джерела в секунду. Адміністратор отримував сповіщення про атаку відразу після того, як вона з'являлася на панелі моніторингу Zabbix. Сповіщення, що надсилається адміністратору, містить наступну інформацію: проблема, час початку проблеми, ім'я проблеми, хост, на якому виникла проблема, ступінь серйозності, операційні дані і початковий ID (рис. 8) [5].

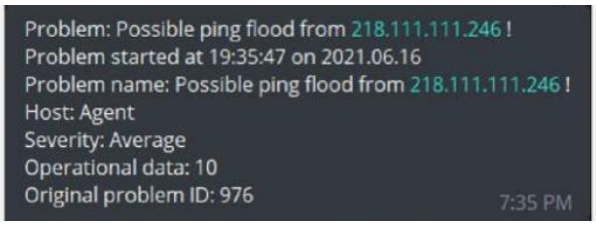

Рис. 8. Отримані повідомлення адміністратором [5]

Під час атаки на графіку (рис. 9) видно, що отримані пінги об'єднувалися в кластери і підсвічувалися червоним відтінком. Zabbix використовує ICMP ping, зареєстрований в

tcpdump.log, для побудови графіка, де кожна точка представляє отриманий пінг. Інформаційна панель Zabbix ілюструє атаки ping flood, запущені на ICMP ping запитів з різних джерел.

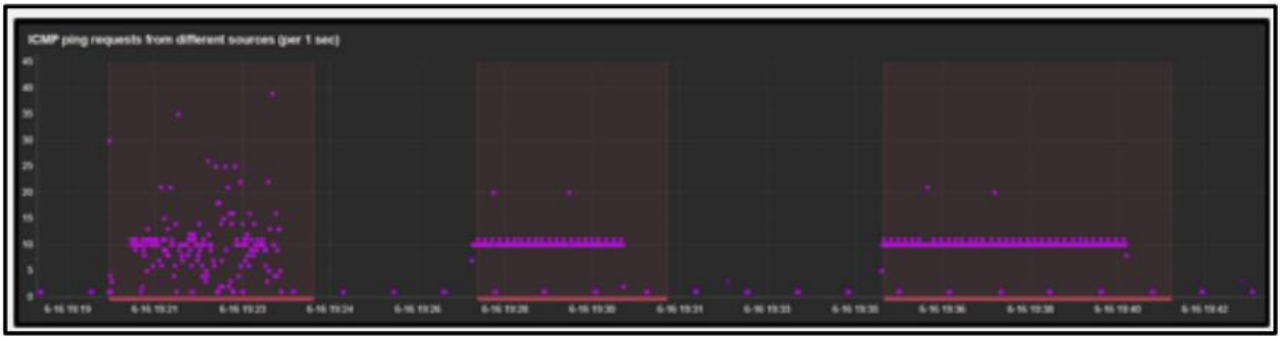

Рис. 9. ICMP Ping Graph [5]

Виходячи з рис. 9, коли сервер-жертва піддавався атаці, на графіку спостерігалося збільшення або сплеск точок, що представляють пінги, які він отримував.

Висновок

У статті досліджено програмне забезпечення Zabbix а також показані приклади його застосування для виявлення загроз і тим самим підвищення інформаційної безпеки. Дана технологія полягає у застосування плагіна для Zabbix, який необхідно правильно налаштувати для виявлення загроз. Детально показаний процес налаштування до ведено, що система виявлення загроз на основі Zabbix працює і може бути застосована для підвищення інформаційної безпеки.

Перелік посилань

1. Моніторинг обладнання [Електронний ресурс]: Режим доступу: https://hs.whitehat.one/uk-ua/blog/monitorynh-pratsezdatnosti-vashoho-obladnannya (дата звернення: 15.11.2022).

2. Далле Вакке А. Zabbix. Практичне керівництво. Дмк Пресс, 2016. – 400 с.

3. Overview of Zabbix [Електронний ресурс]: Режим доступу: https://www.zabbix.com/documentation/1.8/en/manual/about/overview_of_zabbix (дата звернення: 15.11.2022).

4. Using Zabbix for Risk Management [Електронний ресурс]: Режим доступу: https://blog.zabbix.com/using-zabbix-for-risk-management/6570/ (дата звернення: 15.11.2022).

5. Mohd F., Nur F., Muhammad N. Integrated Network Monitoring using Zabbix with Push Notification via Telegram. Journal of Computing Research and Innovation (JCRINN) Vol. 7 No. 1, 2022. – pp. 158-166.

6. Касовська І.В., Шаповаленко О.Д., Луценко І.М. Програмні комплекси мережевого моніторингу для підвищення ефективності захисту мереж // Сучасний захист інформації, №1(45), 2021. – С. 47–52.

7. Білобровець І. В. Розробка структури системи захисту хмарної бази даних від вторгнень // Зб. тез доповідей на наук.-практ. конференції «Цифрова трансформація кібербезпеки» 20 квітня 2022 року. – К.: ДУТ. – С. 6–8.

Надійшла: 06.02.2023 Рецензент: д.т.н., професор Гайдур Г.І.