УДК 004.056.53
doi:10.31799/1684-8853-2026-1-61-76
EDN: OIUQDB

Научные статьи
Articles

Методика противодействия сетевой разведке объекта критической информационной инфраструктуры злоумышленником на основе IoC-анализа

А. А. Шевченко*[а], *канд. техн. наук, доцент, orcid.org/0000-0001-9113-1089
В. А. Липатников*[б], *доктор техн. наук, профессор, orcid.org/0000-0002-3736-4743, lipatnikovanl@mail.ru
В. А. Задбоев*[б], *младший научный сотрудник, orcid.org/0009-0003-9362-1307
П. И. Кузин*[в], *канд. техн. наук, доцент, orcid.org/0000-0003-0880-6204
[а]*Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича, Большевиков пр., 22-1, Санкт-Петербург, 193232, РФ*
[б]*Военная академия связи им. Маршала Советского Союза С. М. Буденного, Тихорецкий пр., 3, Санкт-Петербург, 194064, РФ*
[в]*Санкт-Петербургский государственный лесотехнический университет им. С. М. Кирова, Институтский пер., 5, Санкт-Петербург, 194021, РФ*

Введение: *развитие новых способов воздействия на объект критической информационной инфраструктуры со стороны злоумышленников побуждает к поиску актуальных методик противодействия.* ***Цель:*** *путем исследования способов реализации сетевой разведки разработать методику противодействия данному типу воздействия злоумышленника на основе IoC-анализа, которая позволит повысить оперативность обнаружения вероятных угроз объекта критической информационной инфраструктуры.* ***Результаты:*** *с использованием системного подхода проведен анализ механизмов реализации сетевой разведки (ICMP, TCP/UDP, ARP, DNS, SNMP) и инструментов (nmap, arp-scan, DNSenum, snmpwalk) для выявления активных устройств, открытых портов, операционных систем и служб. Результаты анализа стали основой при разработке модели указанного типа атаки, позволившей установить конкретные IoC, которые возможно зафиксировать в сети на каждом этапе реализации данного воздействия злоумышленника. Синтез полученных знаний о ключевых IoC, таких как аномальные значения TTL, всплески ICMP и SYN-пакетов, повышенный DNS-трафик и повторные попытки аутентификации и способов обеспечения защищенности сети, дал возможность разработать методику противодействия сетевой разведке объекта критической информационной инфраструктуры на основе анализа IoC и ряд инструментальных подходов для оперативного выявления аномалий сетевого трафика и попыток несанкционированного доступа с использованием Python-библиотеки Scapy. Реализация отдельных этапов предложенной методики в виде программного обеспечения способствовала проведению анализа ее результативности в ряде экспериментов.* ***Практическая значимость:*** *определяется возможностью использовать предложенную методику при разработке информационно-управляющих систем обеспечения информационной безопасности объектов критической информационной инфраструктуры.*

Ключевые слова *— информационная безопасность, объект критической информационной инфраструктуры, несанкционированный доступ, анализ трафика, IoC, модель, ранняя нейтрализация угроз.*

Для цитирования: Шевченко А. А., Липатников В. А., Задбоев В. А., Кузин П. И. Методика противодействия сетевой разведке объекта критической информационной инфраструктуры злоумышленником на основе IoC-анализа. *Информационно-управляющие системы*, 2026, № 1, с. 61–76. doi:10.31799/1684-8853-2026-1-61-76, EDN: OIUQDB
For citation: Shevchenko A. A., Lipatnikov V. A., Zadboev V. A., Kuzin P. I. Methodology based on the IoC analysis for countering attacker network reconnaissance on a critical information infrastructure facility. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2026, no. 1, pp. 61–76 (In Russian). doi:10.31799/1684-8853-2026-1-61-76, EDN: OIUQDB

Введение

Безопасная информационная инфраструктура является основой устойчивого функционирования государства и общества. Все более глубокая цифровизация этих систем делает их уязвимыми к компьютерным атакам (КА), последствия которых могут привести не только к экономическим потерям, но и к угрозе жизни людей.

Количество КА на российскую инфраструктуру с начала специальной военной операции увеличилось в разы, в том числе со стороны проправительственных хакерских групп Украины, которые атакуют российские предприятия оборонно-промышленного комплекса, также отмечается рост скорости исполнения атак. Поэтому разработка эффективных методик противодействия сетевой разведке объектов критической информационной инфраструктуры (КИИ) становится приоритетной задачей национальной и информационной безопасности (ИБ) [1–3]. В современных условиях реализация КА носит многоэтапный характер, причем в большинстве случаев невозможно определенно установить, в какой конкретный момент времени злоумыш-

ленник перешел к следующей стадии эскалации информационного конфликта. Одним из решений данной проблемы является внедрение в системы обеспечения ИБ объектов КИИ средств анализа индикаторов компрометации (Indication of Compromise, IoC). IoC — это признак, свидетельствующий о возможном инциденте ИБ в объекте КИИ, системе или устройстве, т. е. цифровой «след», по которому возможно выявить вредоносную активность на ранней стадии развития кризисной ситуации.

Ученые во всем мире активно разрабатывают способы к защите объектов КИИ с использованием указанного подхода. Так, в работах [4–6] предложили комплексные стратегии обеспечения ИБ, учитывающие внедрение передовых технологий, таких как искусственный интеллект, машинное обучение и блокчейн, в основные подпроцессы (аудит безопасности и реагирование на инциденты). Авторами в [7–9] описана концепция интеграции искусственного интеллекта и машинного обучения, в частности глубокого обучения, в систему обнаружения вторжений. В [10, 11] предлагается метод оценки уровня уязвимости сети от появления тех или иных IoC с использованием STIX-графа для структуризации информации об угрозах. Данные работы заложили основу для понимания природы угроз и путей обнаружения IoC.

Однако описанные подходы фокусируются на мониторинге и выявлении атак, уделяя меньше внимания практическим способам их безопасной нейтрализации. В связи с этим разработка и верификация комплексных методик противодействия сетевой разведке объекта КИИ, сочетающих технические, организационные и аналитические меры для обеспечения непрерывности и устойчивости критически важных процессов, является актуальным направлением исследования.

С учетом вышесказанного поставлена задача: исследовать способы реализации сетевой разведки, разработать методику противодействия указанной угрозы объекта КИИ путем раннего выявления IoC. Сетевая разведка выбрана ввиду того, что является одной из основных атак, с которой злоумышленник начинает воздействие на объект КИИ.

Моделирование сетевой разведки объекта КИИ и выявление IoC

Сетевая разведка — это один из видов КА на объекты КИИ, который заключается в исследовании сети в целях сбора информации о ее структуре, активных устройствах, открытых портах, запущенных службах и других характеристик [12, 13]. Это может быть как легитимное действие (например, при проведении аудита безопасности), так и злонамеренное (при подготовке к атаке).

Основные цели сетевой разведки:

1) обнаружение активных устройств — определение IP-адресов, MAC-адресов и других идентификаторов устройств в сети;

2) идентификация открытых портов — поиск портов, на которых работают сетевые службы (например, HTTP, FTP, SSH);

3) определение операционных систем и служб — установление типов операционных систем и версий программного обеспечения (ПО), работающих на устройствах;

4) построение карты сети — создание схемы сети, включая маршрутизаторы, коммутаторы, серверы и другие устройства;

5) выявление уязвимостей — поиск слабых мест в конфигурации сети или ПО.

Методы сетевой разведки включают различные подходы для исследования и анализа сетевой инфраструктуры. Одним из основных методов является перехват сетевого трафика, который позволяет наблюдать за обменом данными между узлами сети, выявлять используемые протоколы и типы передаваемой информации, скрывая свое присутствие в сети. В рамках этого метода используются специализированные анализаторы пакетов и средства мониторинга, обеспечивающие сбор, фильтрацию и интерпретацию трафика, например с помощью утилит *tshark* или *tcpdump*.

Следующим методом является ping-сканирование (ICMP-сканирование), которое используется для обнаружения активных устройств в сети. В рамках этого метода отправляются ICMP-запросы на целевые IP-адреса, что позволяет определить, какие устройства отвечают. Для выполнения таких задач часто применяются инструменты, такие как утилиты *ping* или *fping* [14].

Еще одним важным методом является сканирование портов, которое позволяет проверить состояние портов (открыт, закрыт или фильтруется). В рамках этого метода используются различные типы сканирования, например отправка TCP-пакетов с флагами SYN, ACK, FIN и др.

Также проводится UDP-сканирование, которое проверяет UDP-порты, часто используемые для таких служб, как DNS и DHCP [15]. Одним из популярных инструментов для сканирования портов является утилита *nmap*.

Для более глубокого анализа сети применяется сканирование операционных систем и служб. Этот метод позволяет определить тип операционной системы и версии запущенных служб на основе анализа ответов от устройств. Утилиты,

такие как *nmap -O* и *nmap -sV*, помогают в реализации этого подхода и поиске IoC.

В локальных сетях часто используется сканирование на основе ARP, которое позволяет обнаруживать устройства по их MAC-адресам. Для этого применяются такие инструменты, как *arp-scan* [16]. Также важным методом является сканирование на основе DNS, которое направлено на сбор информации о доменных именах, поддоменах и связанных IP-адресах. Инструменты, такие как *DNSenum* и *dig*, помогают в выполнении этой задачи.

Еще одним методом является сканирование на основе SNMP, которое использует протокол SNMP для получения информации о таких сетевых устройствах, как маршрутизаторы и коммутаторы [17, 18]. Для этого может использоваться инструмент *snmpwalk*.

Пример использования инструмента *nmap* представлен на рис. 1.

Для углубленного анализа сетевой разведки и выявления IoC, которые появляются в сети на каждом этапе реализации атаки, необходимо провести ее моделирование, например, с использованием модели, разработанной корпорацией Lockheed Martin для описания жизненного цикла КА Cyber Kill Chain [19]. В данной модели процесс реализации КА декомпозирован на семь этапов. В случае сетевого сканирования процесс реализации злоумышленником выглядит следующим образом:

1) разведка — пассивный сбор информации о сети, изучение активных хостов и сервисов;

2) вооружение — подготовка инструментов для сканирования сети;

3) доставка — активное сканирование сети;

4) заражение — анализ полученных данных и выявление уязвимостей;

5) инсталляция — попытка установить соединение, если найдены уязвимые сервисы;

6) получение управления — попытка установить канал связи, если получен доступ к системе;

7) выполнение действий — проведение дальнейших атак с использованием собранной информации.

По совершенным злоумышленником действиям на каждом этапе реализации КА в сети возможно выявить «следы» его присутствия — IoC [20] (рис. 2).

```
nmap -sP 192.168.1.0/24   # Ping-сканирование сети
nmap -sS 192.168.1.1      # TCP SYN-сканирование портов
nmap -O 192.168.1.1       # Определение операционной системы
nmap -sV 192.168.1.1      # Определение версий служб
```

■ ***Рис. 1.*** Основные команды для сканирования сети

■ ***Fig. 1.*** Basic commands for network scanning

Установление IoC позволяет приступить к разработке методики противодействия сетевой разведке объекта КИИ, которая обеспечит раннее обнаружение и предотвращение данной атаки на разных этапах ее реализации.

Разработка методики противодействия сетевой разведке объекта КИИ на основе IoC-анализа

С учетом того, что были установлены IoC, которые появляются в сети на каждом этапе реализации КА, была поставлена задача по разработке методики противодействия сетевой разведке объекта КИИ, которая должна позволить в режиме времени, близком к реальному, проводить анализ трафика, подсчет статистики по портам, детектирование IoC (аномальные TTL, всплески ICMP/SYN-пакетов, повышенный DNS-трафик, повторные попытки аутентификации) и выработку предложений по немедленному реагированию на развитие кризисной ситуации.

Ввиду вышеизложенного для противодействия сетевой разведке объекта КИИ предлагается следующая методика, подробное описание которой приведено на рис. 3–7 в виде алгоритма, ее реализующего.

Этапы выполнения предлагаемой методики.

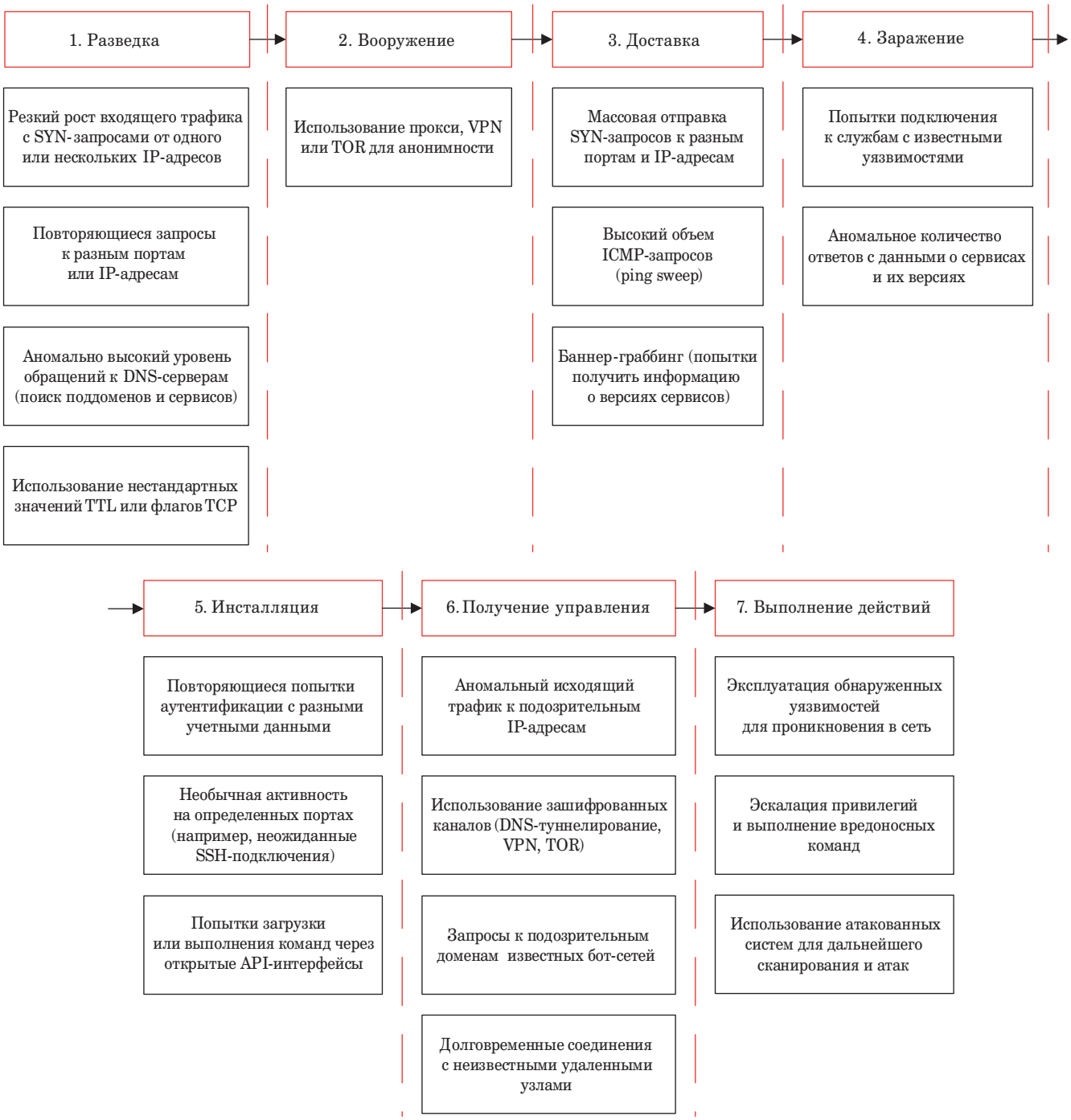
1. Сформировать массивы данных, которые необходимы для работы методики (рис. 3): {*I*} — ресурсы объекта КИИ; {*S*} — средства защиты информации (СЗИ); {*ADM*} – учетные данные о пользователях с правами «Администратор»; {*ADM-Q*} — карантинные данные о пользователях с правами «Администратор»; {*DNS-Q*} — данные, которые не соответствуют записям на сервере DNS; {*X-Q*} — информация об аномальных запросах к удаленным узлам объекта КИИ; {*I-Q*} — информация об удаленных узлах объекта КИИ, помещенных в карантин.

Массивы данных представляют собой наборы информации о ресурсах объекта КИИ, СЗИ и учетных данных пользователей, содержащие такие данные, как внутренний и внешний IP-адреса, маски подсети, MAC-адреса сетевой карты, DNS-серверы, наименование провайдеров и др.

2. Во время информационного обмена провести:

— анализ потоков данных на ресурсах объекта КИИ;

— выявление IoC (аномальные TTL, всплески ICMP/SYN-пакетов), которые соответствуют первому и второму этапам реализации сетевой разведки;



■ ***Рис. 2.*** IoC сетевой разведки объекта КИИ
■ ***Fig. 2.*** IoC network intelligence of a critical infrastructure facility

– устранение уязвимости объекта КИИ, связанной с использованием открытых данных о нем (рис. 4).

3. Если нелегитимная активность на объекте КИИ не прекратилась, то провести:

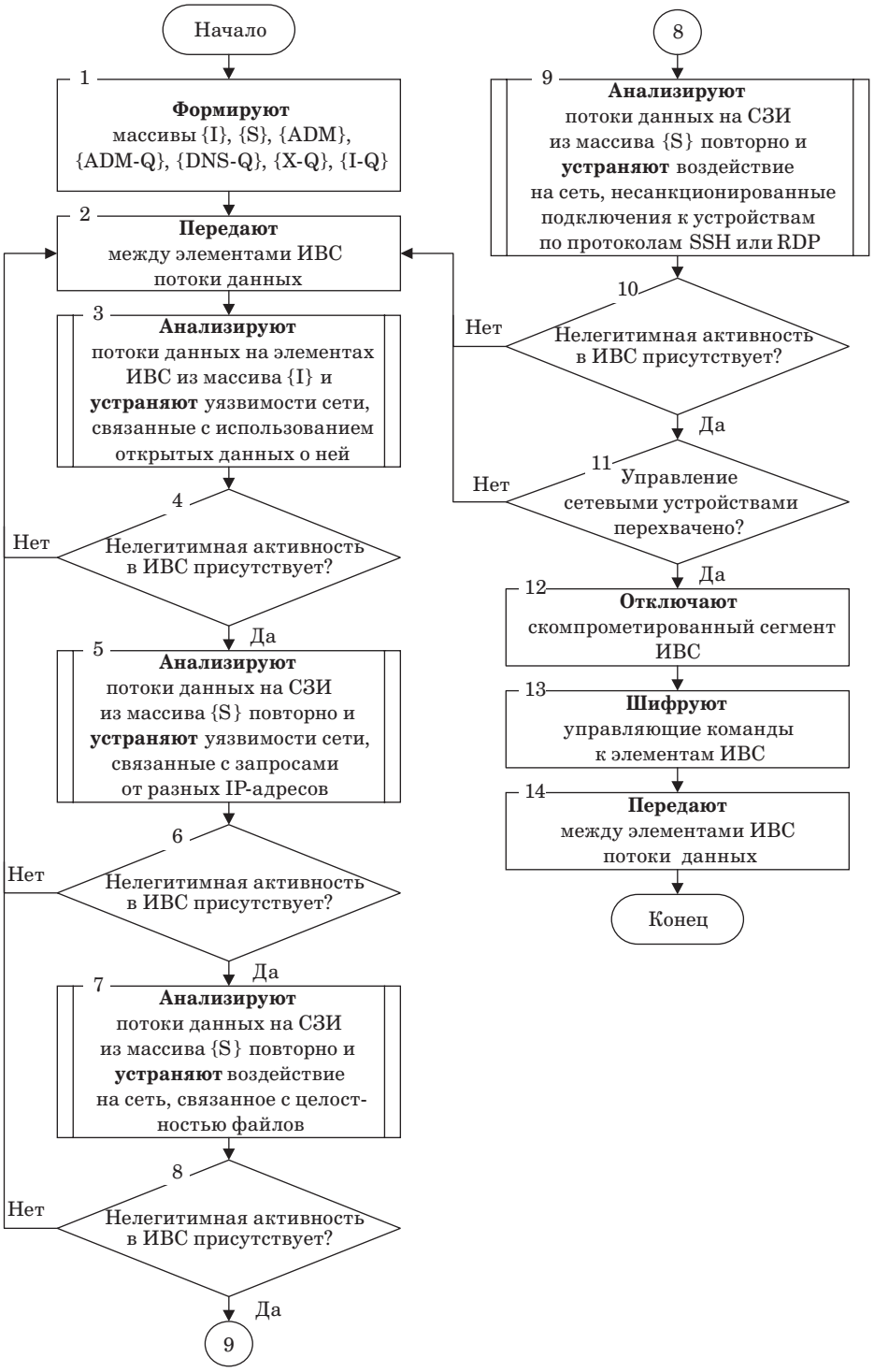
– анализ потоков данных на СЗИ;

– выявление IoC (всплески SYN-пакетов, повышенный DNS-трафик), которые соответствуют третьему и четвертому этапам реализации данной КА;

– устранение уязвимости сети, связанной с запросами от разных IP-адресов (рис. 5).

4. Если нелегитимная активность на объекте КИИ все же не прекратилась, то проводится:

– повторный анализ потоков данных на СЗИ;

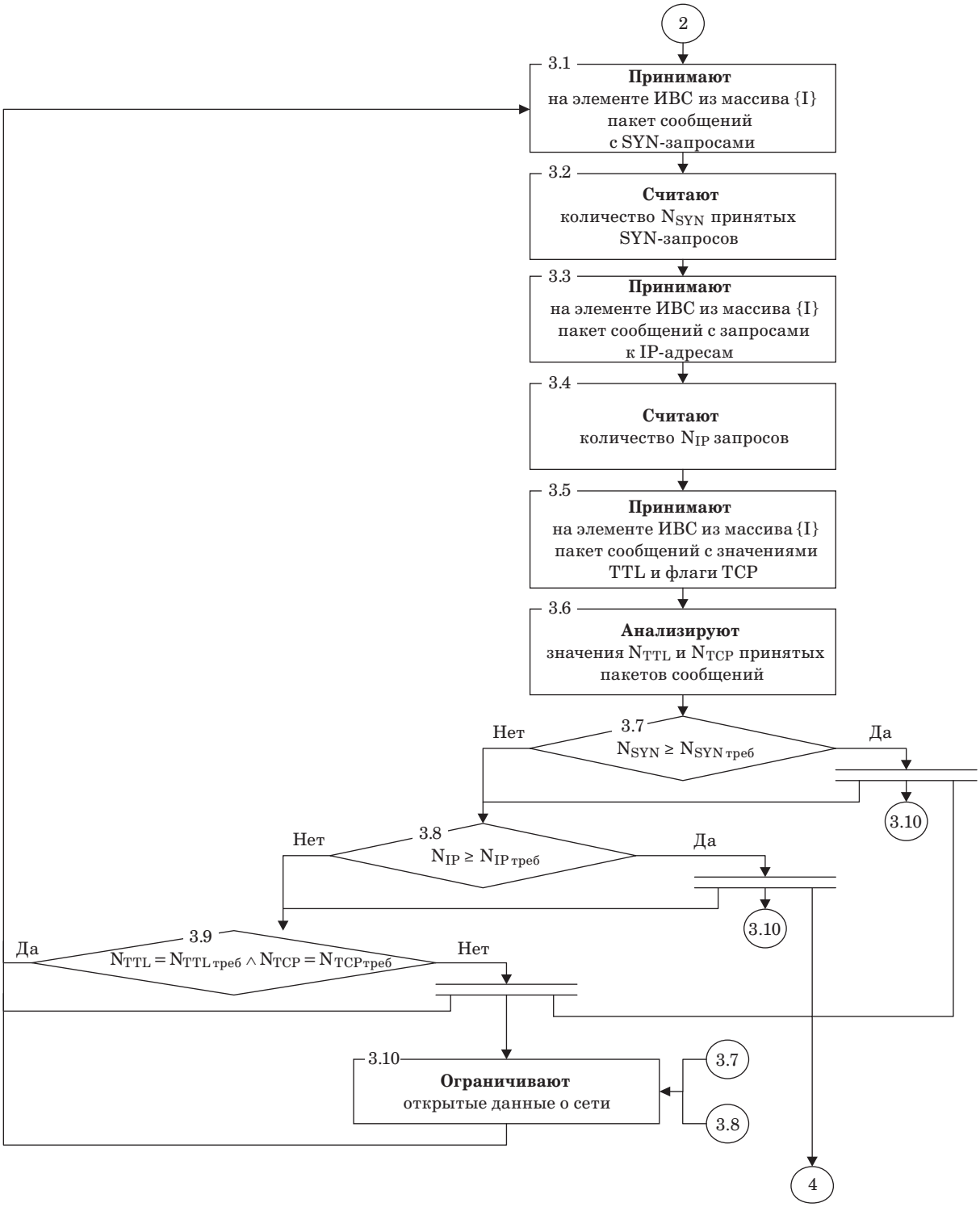


■ ***Рис. 3.*** Блок-схема алгоритма, реализующего методику противодействия сетевой разведке на основе IoC-анализа: *ИВС* — информационно-вычислительная сеть

■ ***Fig. 3.*** Flowchart of an algorithm implementing a method for countering network intelligence based on IoC analysis: *ИВС* — information and computing network

— выявление IoC (изменение учетных данных о пользователях с правами «Администратор», повторные попытки аутентификации, всплески SYN-пакетов, загрузки или выполнения команд через открытые API-интерфейсы), которые соответствуют пятому этапу реализации данной КА;

— устранение воздействия на сеть, связанного с целостностью файлов (рис. 6).



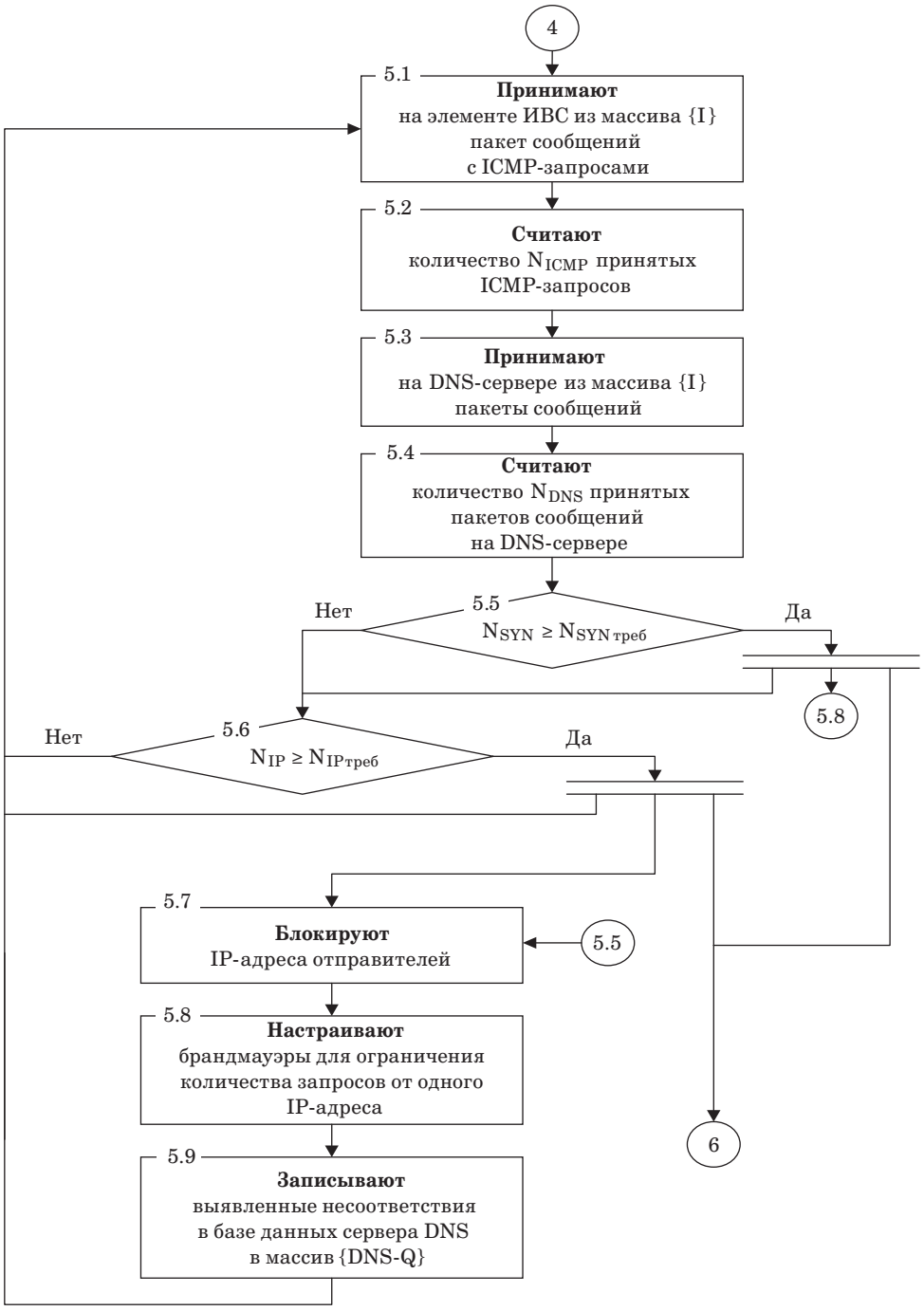
■ ***Рис. 4.*** Блок-схема алгоритма, реализующего методику противодействия сетевой разведке на основе IoC-анализа (*продолжение рис. 3*)

■ ***Fig. 4.*** Flowchart of an algorithm implementing a method for countering network intelligence based on IoC analysis (*continued Fig. 3*)

5. Если нелегитимная активность на объекте КИИ все еще не прекратилась, то проводится:

— повторный анализ потоков данных на СЗИ;

— выявление IoC (аномальные задержки, подозрительные запросы к удаленным узлам объекта КИИ, подключение к карантинным удален-



■ ***Рис. 5.*** Блок-схема алгоритма, реализующего методику противодействия сетевой разведке на основе IoC-анализа (*продолжение рис. 3*)

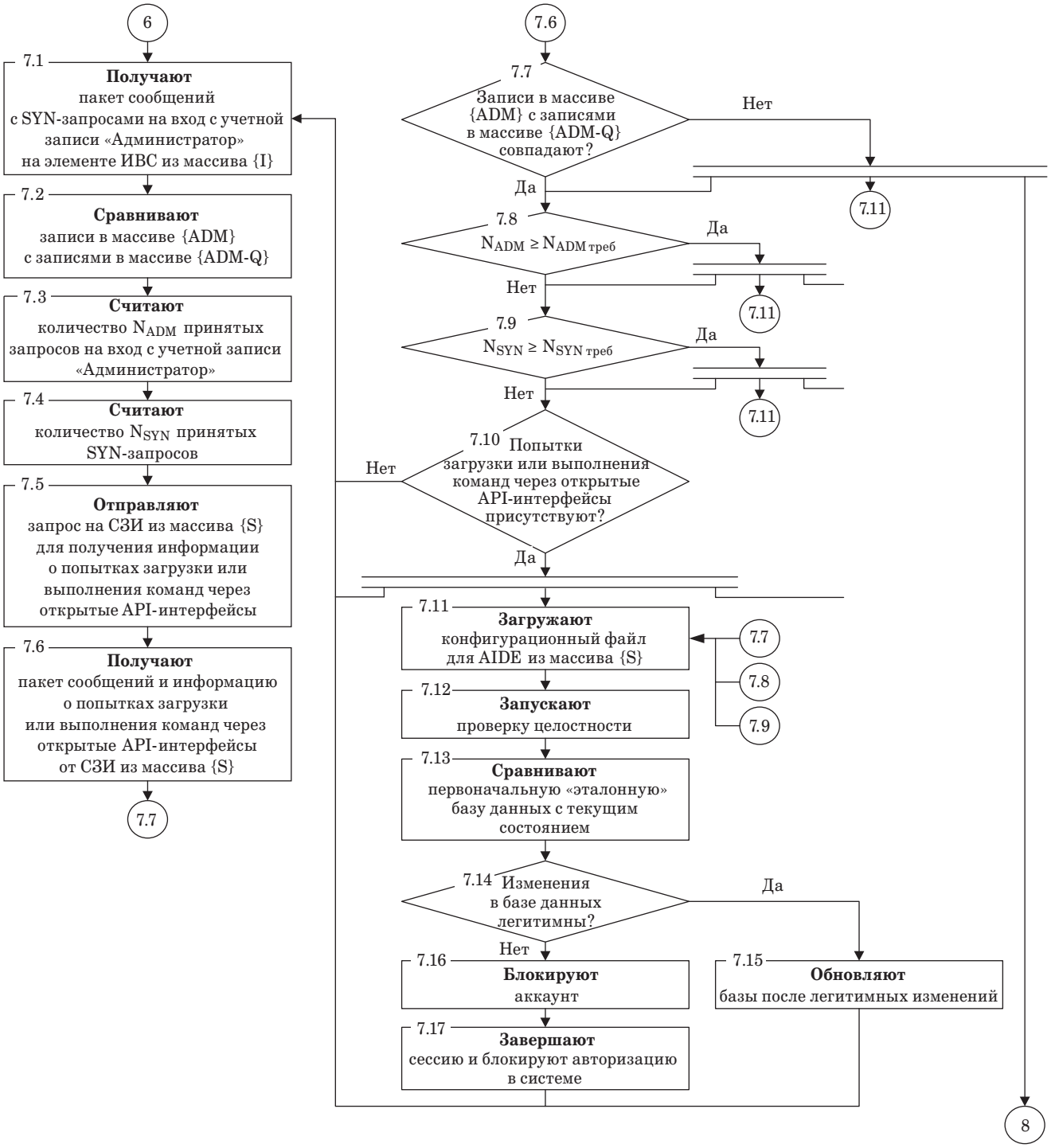
■ ***Fig. 5.*** Flowchart of an algorithm implementing a method for countering network intelligence based on IoC analysis (*continued Fig. 3*)

ным узлам объекта КИИ), которые соответствуют шестому этапу реализации данной КА;

– устранение несанкционированных подключений к устройствам по протоколам SSH или RDP (рис. 7).

6. Если развитие КА находится на финальном этапе, то проводят отключение скомпрометированного сегмента объекта КИИ и осуществляют шифрование управляющих команд (см. рис. 3).

Таким образом, предложена методика противодействия сетевой разведке, учитывающая все этапы эскалации информационного конфликта между злоумышленником и объектом КИИ для незамедлительного его завершения.

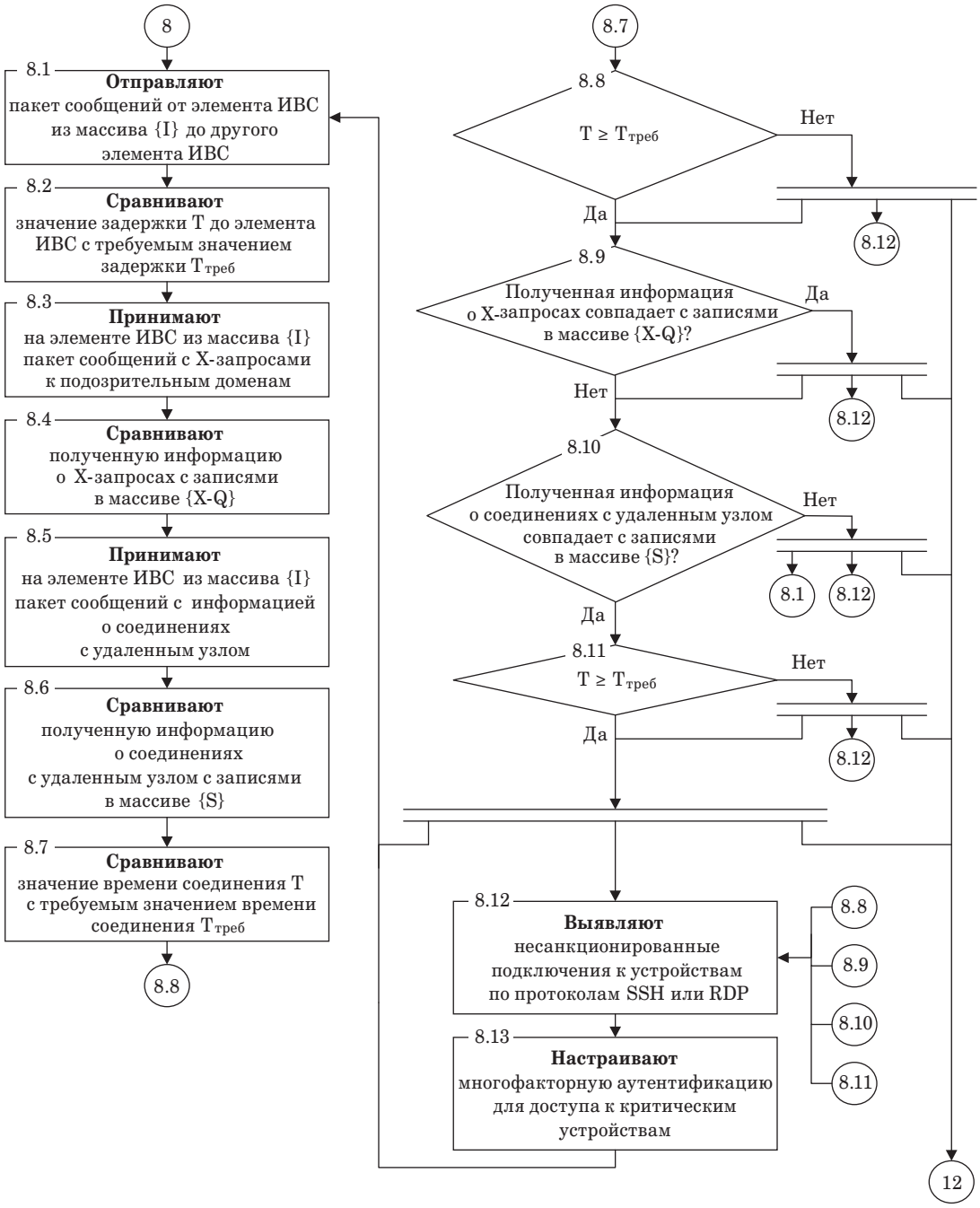


■ ***Рис. 6.*** Блок-схема алгоритма, реализующего методику противодействия сетевой разведке на основе IoC-анализа (*продолжение рис. 3*)

■ ***Fig. 6.*** Flowchart of an algorithm implementing a method for countering network intelligence based on IoC analysis (*continued Fig. 3*)

Реализация разработанной методики в виде ПО

Для того чтобы провести анализ результативности предложенной методики в экспериментах, необходимо разработать ряд инструментальных подходов для автоматизированного выявления IoC и попыток несанкционированного доступа, таких как SYN-запросы, высокий уровень обращения к DNS-записям,



■ ***Рис. 7.*** Блок-схема алгоритма, реализующего методику противодействия сетевой разведке на основе IoC-анализа (*окончание рис. 3*)

■ ***Fig. 7.*** Flowchart of an algorithm implementing a method for countering network intelligence based on IoC analysis (*completion Fig. 3*)

нестандартное значение TTL, высокий объем ICMP-запросов, повторяющиеся попытки аутентификации.

Для отслеживания состояний IoC, связанных с трафиком, необходимо проводить анализ приходящих и исходящих пакетов. В связи с этим для разработки ПО предлагается использовать язык программирования Python и библиотеку Scapy.

Scapy — это мощная интерактивная библиотека на Python для создания, отправки, перехвата и анализа сетевых пакетов. Она сочетает в себе функциональность генератора пакетов, анализатора трафика и декодера протоколов,

при этом предоставляя богатый программный API для автоматизации сетевых экспериментов, тестирования и прототипирования. Scapy давно используется как в исследовании безопасности и «пентестинге», так и в сетевой диагностике и обучении.

Проверка на аномальное поведение TTL происходит путем отправки в функцию самого пакета и массива стандартных значений; если такового нет, то функция передает True, что означает необходимость отправки события на сервер (рис. 8).

Проверка на аномальное количество SYN-запросов происходит путем подсчета количества пакетов с флагом «*S*»; если их количество превышает переданный предел *THRESHOLD* во времени *TIME_WINDOW*, то функция передает True, что означает необходимость отправки события на сервер (рис. 9).

Для того чтобы проверить аномальное количество ICMP/DNS-пакетов, а также трафика на других портах, был создан список экземпляров класса Port, в который записываются значение порта, имя порта и предельное количество проходящих байт в секунду (рис. 10).

Далее на каждый экземпляр класса Port в списке *ports* используется метод этого класса *paket_analyze*(), который содержит набор правил, проверяющих, подходит ли проходящий пакет под характеристики, заданные при инициализации класса, и если подходит, то производится перерасчет среднего значения байт этих пакетов и их схожесть (рис. 11).

Для того чтобы проверить количество попыток входа в систему, необходимо проверить записи во встроенной системе журналирования операционной системы. В различных операционных системах путь к нужному журналу аутентификации разный, например в *Astra Linux* — это */var/log/auth.log* (рис. 12).

Таким образом, было разработано ПО, которое в автоматизированном режиме выявляет IoC и попытки несанкционированного доступа. На рис. 13 представлены результаты регистрации нестандартных значений TTL и высокого объема ICMP-запросов.

```python
from scapy.all import IP

# Определяем стандартное значение TTL

def ttl_check(packet, standart_ttl):
    if IP in packet:
        ttl = packet[IP].ttl
        # Если TTL нестандартный, выводим сообщение
        if ttl not in standart_ttl:
            print(f"Нестандартный ttl: {ttl}")
            return True
    return False
```

■ ***Рис. 8.*** Листинг кода, реализующего выявление аномальных значений TTL

■ ***Fig. 8.*** Listing of code implementing the detection of abnormal TTL values

```python
ports = []

ports.append(Port(23, 'telnet', 25000))
ports.append(Port(22, 'ssh', 60000))
ports.append(Port(21, 'ftp'))
ports.append(Port(80, 'http', 1000))
ports.append(Port(443, 'https', 20000))
ports.append(Port(0, 'icmp', 1000))
ports.append(Port(161,'SNMP'))
ports.append(Port(162,'SNMP'))
ports.append(Port(10161,'SNMP_tls'))
ports.append(Port(10162,'SNMP_tls'))
ports.append(Port(179,'BGP', 100))
ports.append(Port(53, 'DNS', 10000))
```

■ ***Рис. 10.*** Список анализируемых портов

■ ***Fig. 10.*** List of analyzed ports

```python
def syn_check(packet, THRESHOLD, TIME_WINDOW):
    if packet.haslayer(TCP) and packet[TCP].flags == "S":
        src_ip = packet[IP].src
        now = time.time()
        # Добавляем временную метку
        syn_requests[src_ip].append(now)
        # Удаляем старые метки (вне окна TIME_WINDOW)
        syn_requests[src_ip] = [t for t in syn_requests[src_ip] if now - t < TIME_WINDOW]
        if len(syn_requests[src_ip]) > THRESHOLD:
            print(f"[ALERT] SYN flood detected from {src_ip} — {len(syn_requests[src_ip])} SYNs!")
            return True
        else:
            return False
```

■ ***Рис. 9.*** Листинг кода, реализующего проверку на аномальное количество SYN-запросов

■ ***Fig. 9.*** Listing of code implementing a check for an abnormal number of SYN requests

```python
def packet_analyze(self, packet) -> None:
    """Анализирует входящий пакет и решает, учитывать ли его в статистике."""
    self.analyze_counter += 1

    try:
        layers = {layer.name: layer for layer in packet.layers()}

        rules = [
            lambda l: "ICMP" in l and self.port_num == 0,
            lambda l: "TCP" in l and (
                l["TCP"].sport == self.port_num or l["TCP"].dport == self.port_num
            ),
            lambda l: "UDP" in l and (
                l["UDP"].sport == self.port_num or l["UDP"].dport == self.port_num
            ),
            lambda l: "DNSQR" in l,
        ]

        for rule in rules:
            if rule(layers):
                self._handle_packet(packet)
                break

    except Exception as e:
        print(f"[WARN] Ошибка анализа пакета на порту {self.name}: {e}")
```

- ***Рис. 11.*** Листинг кода, реализующего анализ входящих пакетов для подсчета статистики
- ***Fig. 11.*** Listing of code implementing the analysis of incoming packets for calculating statistics

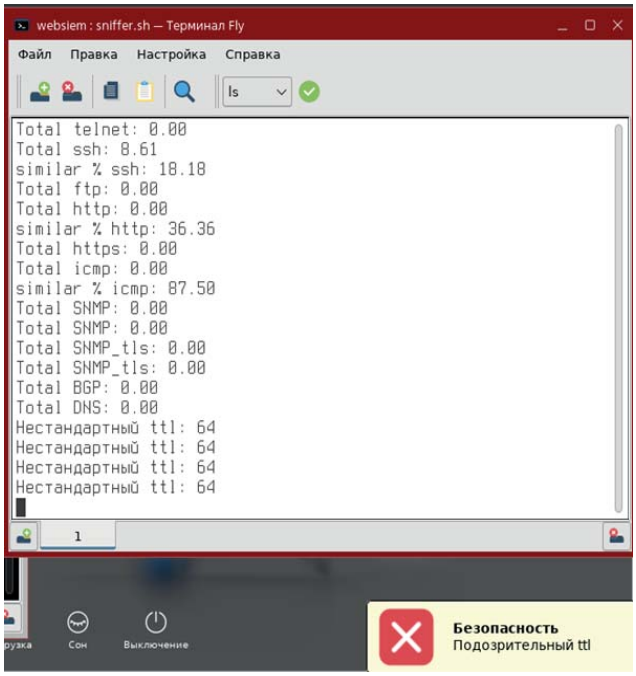
```python
try:
    with open(log_file_path, 'r') as log_file:
        log_file.seek(0, 2)  # Перейти в конец файла
        # old_code = ""
        while True:
            line = log_file.readline()
            if not line:  # Если нет новых строк, подождать 1 секунду
                time.sleep(1)
                continue
            if ('authentication failure' in line):
                # old_code = code
                message = "Найдена ошибка аутентификации: " + line.strip()
                print("Найдена ошибка аутентификации:", line.strip())
                thr = ThreatInfo(message, device='astra', date=datetime.datetime.now(), level=4)
                event["event_type"] = thr.name
                send_event(conf.address, conf.token, event)
                popup_message(message[:90], notif_timer)

except FileNotFoundError:
    print(f"Файл {log_file_path} не найден.")
except PermissionError:
    print(f"Нет доступа к файлу {log_file_path}.")
```

- ***Рис. 12.*** Листинг кода, реализующего проверку журнала на ошибки аутентификации
- ***Fig. 12.*** Listing of code implementing log checking for authentication errors

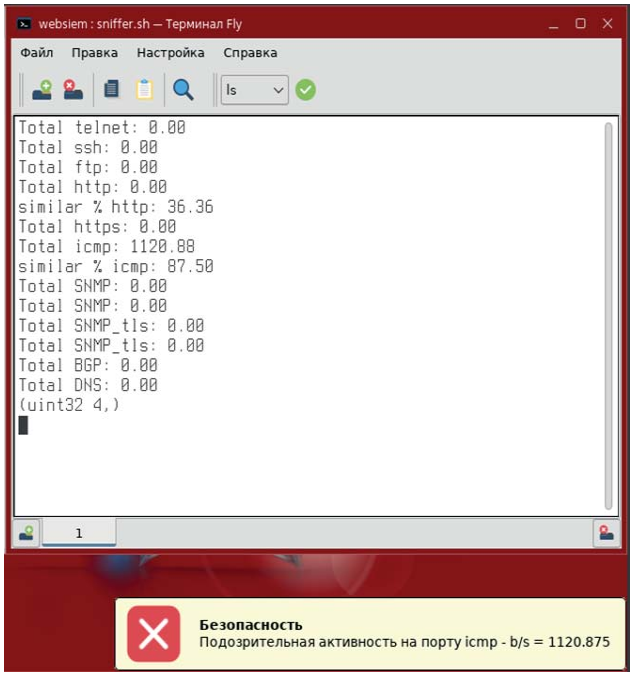
Оценка степени достижения цели

Для оценки степени достижения цели был проведен ряд эксперементальных проверок на лабораторном стенде в виде сети виртуальных машин с одним ядром центрального процессора и 8 ГБ оперативной памяти. В данной виртуальной сети проводилось моделирование сетевой разведки. Анализ трафика проводился с помощью специализированного ПО Wireshark и разработанного ПО на основе предлагаемой методики. Сравнительный анализ времени обнаружения IoC данными средствами представлен в таблице, причем в первом продукте для выяв-

а)



б)



■ ***Рис. 13.*** Результаты экспериментальной проверки разработанного ПО: *а* — регистрация нестандартных значений TTL; *б* — регистрация высокого объема ICMP-запросов

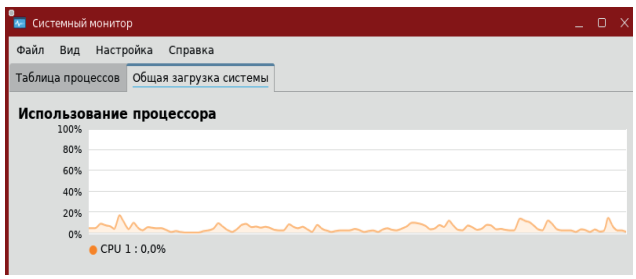
■ ***Fig. 13.*** Results of experimental testing of the developed software: *а* — logging of non-standard TTL values; *б* — logging of high volume of ICMP requests

■ Сравнительный анализ времени обнаружения IoC

■ Comparative analysis of IoC detection times

IoC	Время обнаружения	
	ПО Wireshark	ПО, реализующее предлагаемую методику
Подозрительный TTL	<1 с	<1 с
Подозрительное количество SYN-запросов	>2 с	<1 с
Подозрительный объем трафика на порт	<2 с	<1 с
Одинаковые пакеты	–	<1 с

а)



б)

■ ***Рис. 14.*** Использование ресурсов виртуальной машины во время работы: *а* — ПО Wireshark; *б* — ПО, реализующего предлагаемую методику

■ ***Fig. 14.*** Virtual machine resource usage during operation: *а* — Wireshark software; *б* — software implementing the proposed method

ления IoC (подозрительный TTL, подозрительное количество SYN-запросов, подозрительный объем трафика на порт) необходимо настраивать отдельные фильтры, а во втором продукте данные IoC выявляются в автоматизированном режиме. Данный факт и результаты анализа указывают на то, что при использовании предлагаемой методики оперативность реагирования на угрозы выше, чем при использовании традиционных подходов. Также с помощью ПО Wireshark не представляется возможным провести подсчет одинаковых поступающих сетевых пакетов на объект КИИ в автоматизированном режиме.

Дополнительно проведено сравнение использования ресурсов на виртуальной машине между ПО Wireshark и разработанным ПО. В режиме анализа трафика ПО Wireshark использует от 5 до 20 % ресурса процессора виртуальной машины (рис. 14, *а*), а разработанное ПО в режиме активной обработки и отправки событий использует от 1 до 10 % ресурса процессора (рис. 14, *б*). Ввиду этого можно сделать вывод, что разработанное ПО во время работы задействует в два раза меньше ресурсов виртуальной машины, что позволяет оптимально использовать ограниченный вычислительный ресурс.

Заключение

Представленное исследование демонстрирует, что сетевая разведка — многообразное по методам и целям явление, которое может выступать как инструментом легитимного аудита, так и одним из этапов при проведении многоэтапных атак. Ключевыми задачами являлись исследование методов сетевой разведки и моделирование процесса реализации данной атаки, разработка методики противодействия указанной угрозы объекта КИИ на основе раннего выявления IoC и немедленного реагирования на складывающуюся кризисную ситуацию. Решение данных задач требует сочетания различных техник (ICMP/ARP/TCP/UDP/DNS/SNMP) и специализированных инструментов (*nmap, arp-scan, snmpwalk, DNSenum* и др.). Установлены IoC (аномальные значения TTL, всплески ICMP/SYN-пакетов, повышенные DNS-запросы, повторные попытки аутентификации), которые позволяют формализовать критерии обнаружения риска и служат основой для автоматизированного мониторинга.

Предложенная методика проверена в ряде экспериментов с помощью разработанного на ее основе ПО, которое представляет собой анализатор трафика и коллектора журналов аутентификации с использованием библиотеки Scapy в Python и простой логики пороговых проверок — подтверждает применимость подхода «сбор + корреляция»: анализ сетевых пакетов в сочетании с локальными логами повышает уверенность в выявлении инцидентов и снижает число ложных срабатываний. Использование классов для агрегирования статистики по портам и реализованные правила детектирования (анализ TTL, подсчет SYN/ICMP-пакетов, контроль объема трафика) дают модульную архитектуру, удобную для расширения и интеграции с системами SIEM/IDS.

Научная новизна заключается в комплексном подходе к выявлению сетевой разведки и аномального поведения трафика в сочетании с анализом сетевых пакетов и локальных журналов аутентификации. Предложенная методика детектирования IoC отличается от традиционных методов защиты объектов КИИ тем, что позволяет анализировать трафик, подсчитывать статистику по портам, выявлять аномалии и нейтрализовать КА данного типа в режиме времени, близком к реальному, что повышает оперативность обнаружения угроз.

Теоретическая значимость заключается в развитии методологических положений теории управления ИБ объектов КИИ за счет:

— использования системного подхода для анализа механизмов реализации сетевой разведки и ее моделирования, что позволило установить конкретные IoC, которые возможно зафиксировать в сети на каждом этапе реализации данного воздействия злоумышленника;

— синтеза полученных знаний о IoC и способов обеспечения защищенности сети, что позволило разработать методику противодействия сетевой разведке объекта КИИ.

Практическая значимость методики определяется возможностью ее использования при разработке перспективных информационно-управляющих систем обеспечения ИБ объектов КИИ с применением технологий искусственного интеллекта.

Положительный эффект разработанной методики заключается в том, что ее применение позволяет повысить оперативность обнаружения вероятных угроз объекта КИИ.

Литература

1. *Федеральный закон от 26.07.2017 N 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации»*. https://gossopka.ru/doc/npa/federalnye-zakony/federalnyy-zakon-ot-26072017-n-187-fz-o-bezopasnosti-kriticheskoy-informacionnoy-infrastruktury-ross-37407.html (дата обращения: 10.10.2025).
2. *Методика оценки угроз безопасности информации*. https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/metodicheskij-dokument-ot-5-fevralya-2021-g.html (дата обращения: 05.11.2025).
3. **Липатников В. А., Шевченко А. А., Мелехов К. В., Задбоев В. А.** Метод активной защиты объектов критической информационной инфраструктуры от кибератак на основе прерывания процесса воздействия нарушителя. *Информационно-управляющие системы*, 2025, № 2 (135), с. 37–49. doi:10.31799/1684-8853-2025-2-37-49, EDN: PVFXD
4. **Ajayi O. O., Alozie C. E., Abieba O. A.** Enhancing cybersecurity in energy infrastructure: Strategies for safeguarding critical systems in the digital age. *Trends in Renewable Energy*, 2025, vol. 11, no. 2, pp. 201–212. doi:http://dx.doi.org/10.17737/tre.2025.11.2.00192
5. **Akinbolaji T. J.** Advanced integration of artificial intelligence and machine learning for real-time threat detection in cloud computing environments. *Iconic Research and Engineering Journals*, 2024, vol. 6, no. 10, pp. 980–991. doi:10.5281/zenodo.13963676
6. Пат. 2839562 C1 Российская Федерация, МПК G06F 12/14, H04L 12/22. *Способ защиты информационно-вычислительной сети от вторжения,* В. А. Задбоев, В. А. Липатников, К. В. Мелехов, А. А. Шевченко. № 2024104981; заявл. 27.02.2024; опубл. 06.05.2025.
7. **Shukla P.**, **Krishna C. R., Patil N. V.** Kafka-Shield: Kafka Streams-based distributed detection scheme for IoT traffic-based DDoS attacks. *Security and Privacy*, 2024, vol. 7, iss. 12. doi:10.1002/spy2.416
8. **Ичетовкин Е. А., Котенко И. В.** Модели и алгоритмы защиты систем обнаружения вторжений от атак на компоненты машинного обучения. *Computational nanotechnology*, 2025, т. 12, № 1, с. 17–25. doi:10.33693/2313-223X-2025-12-1-17-25. EDN: LSJCNO
9. **Sarhan M., Layeghy S., Moustafa N., Portmann M.** *Netflow Datasets for Machine Learning-Based Network Intrusion Detection Systems*. In: Big Data Technologies and Applications. Springer International Publishing, 2021, pp. 117–135.
10. **Chen S. S., Hwang R. H., Ali A., Lin Y. D., Wei Y. C., Pai T. W.** Improving quality of indicators of compromise using STIX graphs. *Computers & Security*, 2024, vol. 144, Art. 103972. doi:https://doi.org/10.1016/j.cose.2024.103972
11. **Kartak V., Bashmakov N.** Method for selecting indicators of data compromise. *2022 International Siberian Conference on Control and Communications*, 2022, pp. 1–5. doi:10.1109/SIBCON56144.2022.10002962
12. **Everson D., Cheng L.** A survey on network attack surface mapping. *Digital Threats: Research and Practice*, 2024, vol. 5, no. 2, pp. 1–25. doi:https://doi.org/10.1145/3640019
13. **Tundis A., Modo Nga E. M., Mühlhäuser M.** An exploratory analysis on the impact of Shodan scanning tool on the network attacks. *ARES 2021: The 16th International Conference on Availability, Reliability and Security*, 2021, pp. 1–10. https://doi.org/10.1145/3465481.3469197
14. **Nasereddin M., ALKhamaiseh A., Qasaimeh M.** A systematic review of detection and prevention techniques of SQL injection attacks. *Information Security Journal: A Global Perspective*, 2023, vol. 32, no. 4, pp. 252–265. doi:10.1080/19393555.2021.1995537
15. **Velankar M. R., Mahalle P. N., Shinde G. R.** *Machine Thinking: New Paradigm Shift*. In: Cognitive Computing for Machine Thinking. Innovations in Sustainable Technologies and Computing, 2024. 98 p.
16. **Chernyagin A. S., Svetunkov S. G.** Study of the concept of Bayesian optimization and practical use of its algorithms in the Python programming language. *Technoeconomics*, 2023, vol. 2, no. 4 (7), pp. 4–15. doi:https://doi.org/10.57809/2023.2.4.7.1
17. **Задбоев В. А., Абрамова Н. И., Москалев В. С.** Противодействие внешним вторжениям в информационно-вычислительной сети. *Телекоммуникации и связь*, 2025, № 5 (8), с. 18–23. doi:10.21681/3034-4050-2025-5-18-23, EDN VSBDMK
18. **Липатников В. А., Мелехов К. В., Задбоев В. А.** Способ активной защиты информационно-вычислительных сетей от многоэтапных атак. *Региональная информатика и информационная безопасность: сб. тр. СпБ междунар. конф.*, Санкт-Петербург, 23–25 октября 2024 г. СПб., 2024, с. 112–114.
19. **Sun N., Ding M., Jiang J., Xu W., Mo X., Tai Y.** Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives. *IEEE Communications Surveys & Tutorials*, 2023, vol. 25, no. 3, pp. 1748–1774. doi:10.1109/COMST.2023.3273282
20. **Parashchuk I., Levshun D., Kotenko I.** Proactive complex security analysis of IoT-based critical infrastructure facilities. *2025 International Russian Smart Industry Conference*, 2025, pp. 52–57. doi:10.1109/SmartIndustryCon65166.2025.10986286

UDC 004.056.53
doi:10.31799/1684-8853-2026-1-61-76
EDN: OIUQDB

Methodology based on the IoC analysis for countering attacker network reconnaissance on a critical information infrastructure facility

A. A. Shevchenko[a], PhD, Tech., Associate Professor, orcid.org/0000-0001-9113-1089
V. A. Lipatnikov[b], Dr. Sc., Tech., Professor, orcid.org/0000-0002-3736-4743, lipatnikovanl@mail.ru
V. A. Zadboev[b], Junior Researcher, orcid.org/0009-0003-9362-1307
P. I. Kuzin[c], PhD, Tech., Associate Professor, orcid.org/0000-0003-0880-6204
[a]The Bonch-Bruevich Saint-Petersburg State University of Telecommunications, 22-1, Bolshevikov Pr., 193232, Saint-Petersburg, Russian Federation
[b]S. M. Budenny Military Academy of Communication, 3, Tikhoretskii Pr., 190064, Saint-Petersburg, Russian Federation
[c]Saint Petersburg State Forest Technical University named after S. M. Kirov, 5, Institutsky per., 194021, Saint-Petersburg, Russian Federation

Introduction: The spread of new ways to target critical information infrastructure facilities developed and used by attackers encourages the search for relevant countermeasures. **Purpose:** Analyzing various ways of network reconnaissance on critical information infrastructure facilities, to develop a methodology based on the IoC analysis for countering this type of attacker influence in order to improve the efficiency of detecting potential threats to a critical information infrastructure facility. **Results:** Using a system approach, we analyze network reconnaissance mechanisms (ICMP, TCP/UDP, ARP, DNS, SNMP) and tools (nmap, arp-scan, DNSenum, snmpwalk) for detecting and identifying active devices, open ports, operating systems, and services. On the basis of this analysis, we develop a model for this type of attack, which allows identifying specific IoCs that can be recorded in the network at each stage of an attacker's invasion. Synthesis of the acquired knowledge about key IoCs, such as abnormal TTL values, bursts of ICMP and SYN packets, increased DNS traffic, and repeated authentication attempts, with network security methods enables developing a methodology based on the IoC analysis for countering network reconnaissance on a critical information infrastructure facility, and a number of instrumental approaches for the rapid detection of network traffic anomalies and unauthorized access attempts using the Scapy Python library. The implementation of individual stages of the proposed methodology in the form of software has facilitated the analysis of its effectiveness in a number of experiments. **Practical relevance:** The proposed methodology can be used for developing information management systems which ensure the information security of critical information infrastructure facilities.

Keywords — information security, critical information infrastructure facility, unauthorized access, traffic analysis, IoC, model, early threat mitigation.

For citation: Shevchenko A. A., Lipatnikov V. A., Zadboev V. A., Kuzin P. I. Methodology based on the IoC analysis for countering attacker network reconnaissance on a critical information infrastructure facility. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2026, no. 1, pp. 61–76 (In Russian). doi:10.31799/1684-8853-2026-1-61-76, EDN: OIUQDB

References

1. *Federal'nyj zakon ot 26.07.2017 N 187-FZ "O bezopasnosti kriticheskoj informacionnoj infrastruktury Rossijskoj Federacii"* [Federal Law of 26.07.2017 N 187-FZ "On the Security of Critical Information Infrastructure of the Russian Federation"]. Available at: https://gossopka.ru/doc/npa/federalnye-zakony/federalnyy-zakon-ot-26072017-n-187-fz-o-bezopasnosti-kriticheskoy-informacionnoy-infrastruktury-ross-37407.html (accessed 10 October 2025).
2. *Metodika ocenki ugroz bezopasnosti informacii* [Information security threats assessment methodology]. Available at: https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/metodicheskij-dokument-ot-5-fevralya-2021-g.html (accessed 5 November 2025).
3. Lipatnikov V. A., Shevchenko A. A., Melekhov K. V., Zadboev V. A. Active protection method against cyberattacks for the objects of critical information infrastructure based on the interruption of the process of an intruder's impact. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2025, no. 2, pp. 37–49. doi:10.31799/1684-8853-2025-2-37-49, EDN: PVFXD.
4. Ajayi O. O., Alozie C. E., Abieba O. A. Enhancing cybersecurity in energy infrastructure: strategies for safeguarding critical systems in the digital age. *Trends in Renewable Energy*, 2025, vol. 11, no. 2, pp. 201–212. doi:http://dx.doi.org/10.17737/tre.2025.11.2.00192.
5. Akinbolaji T. J. Advanced integration of artificial intelligence and machine learning for real-time threat detection in cloud computing environments. *Iconic Research and Engineering Journals*, 2024, vol. 6, no. 10, pp. 980–991. doi:10.5281/zenodo.13963676.
6. Zadboev V. A., et al. *Sposob zashchity informacionno-vychislitel'noj seti ot vtorzheniya* [A method for protecting an information and computing network from intrusion]. Patent Russian Federation, no. 2024104981, 2025.
7. Shukla P., Krishna C. R., Patil N. V., Kafka-Shield: Kafka Streams-based distributed detection scheme for IoT traffic-based DDoS attacks. *Security and Privacy*, 2024, vol. 7, iss. 12. doi:10.1002/spy2.416
8. Ichetovkin E. A., Kotenko I. V. Models and algorithms for protecting intrusion detection systems from attacks on machine learning components. *Computational nanotechnology*, 2025, vol. 12, no. 1, pp. 17–25 (In Russian). doi:10.33693/2313-223X-2025-12-1-17-25. EDN: LSJCNO
9. Sarhan M., Layeghy S., Moustafa N., Portmann M. *Netflow Datasets for Machine Learning-Based Network Intrusion Detection Systems*. In: *Big Data Technologies and Applications*. Springer International Publishing, 2021, pp. 117–135.
10. Chen S. S., Hwang R. H., Ali A., Lin Y. D., Wei Y. C., Pai T. W. Improving quality of indicators of compromise using STIX graphs. *Computers & Security*, 2024, vol. 144, Art. 103972. doi:https://doi.org/10.1016/j.cose.2024.103972
11. Kartak V., Bashmakov N. Method for selecting indicators of data compromise. *2022 International Siberian Conference on Control and Communications*, 2022, pp. 1–5. doi:10.1109/SIBCON56144.2022.10002962
12. Everson D., Cheng L. A survey on network attack surface mapping. *Digital Threats: Research and Practice*, 2024, vol. 5, no. 2, pp. 1–25. doi:https://doi.org/10.1145/3640019
13. Tundis A., Modo Nga E. M., Mühlhäuser M. An exploratory analysis on the impact of Shodan scanning tool on the network attacks. *ARES 2021: The 16th International Conference on Availability, Reliability and Security*, 2021, pp. 1–10. https://doi.org/10.1145/3465481.3469197
14. Nasereddin M., ALKhamaiseh A., Qasaimeh M. A systematic review of detection and prevention techniques of SQL injection attacks. *Information Security Journal: A Global Perspective*, 2023, vol. 32, no. 4, pp. 252–265. doi:10.1080/19393555.2021.1995537
15. Velankar M. R., Mahalle P. N., Shinde G. R. *Machine Thinking: New Paradigm Shift*. In: *Cognitive Computing for Machine Thinking*. Innovations in Sustainable Technologies and Computing, 2024. 98 p.
16. Chernyagin A. S., Svetunkov S. G. Study of the concept of Bayesian optimization and practical use of its algorithms in the Python programming language. *Technoeconomics*, 2023,

vol. 2, no. 4 (7), pp. 4–15. doi:https://doi.org/10.57809/2023.2.4.7.1
17. Zadboev V. A., Abramova N. I., Moskalev V. S. Countering external intrusions into the information and computing network. *Telecommunications and Communications*, 2025, no. 5 (8), pp. 18–23 (In Russian). doi:10.21681/3034-4050-2025-5-18-23, EDN VSBDMK
18. Lipatnikov V. A., Melekhov K. V., Zadboev V. A. A method for actively protecting information and computing networks from multi-stage attacks. *Sbornik trudov Sankt-Peterburgskoj mezhdunarodnoj konferencii "Regional'naya informatika i informacionnaya bezopasnost"* [Proceedings of the St. Petersburg International Conference "Regional informatics and information security"]. Saint Petersburg, 2024, pp. 112–114 (In Russian).
19. Sun N., Ding M., Jiang J., Xu W., Mo X., Tai Y. Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives. *IEEE Communications Surveys & Tutorials*, 2023, vol. 25, no. 3, pp. 1748–1774. doi:10.1109/COMST.2023.3273282
20. Parashchuk I., Levshun D., Kotenko I. Proactive complex security analysis of IoT-based critical infrastructure facilities. *2025 International Russian Smart Industry Conference*, 2025, pp. 52–57. doi:10.1109/SmartIndustryCon65166.2025.10986286

ПАМЯТКА ДЛЯ АВТОРОВ

Поступающие в редакцию статьи проходят обязательное рецензирование.

При наличии положительной рецензии статья рассматривается редакционной коллегией. Принятая в печать статья направляется автору для согласования редакторских правок. После согласования автор представляет в редакцию окончательный вариант текста статьи.

Процедуры согласования текста статьи могут осуществляться как непосредственно в редакции, так и по e-mail (ius.spb@gmail.com).

При отклонении статьи редакция представляет автору мотивированное заключение и рецензию, при необходимости доработать статью — рецензию.

Редакция журнала напоминает, что ответственность за достоверность и точность рекламных материалов несут рекламодатели.