АЛГОРИТМИЧЕСКОЕ РЕШЕНИЕ ЗАДАЧИ СИНТЕЗА ПРАВИЛ ДОСТУПА НА ОСНОВЕ ТЕЛЕМЕТРИИ СЕТЕВЫХ ПОТОКОВ

Редькин Ю.В., канд. техн. наук, доцент,
Рябуха А.Г., магистрант,
Пешков А.В., магистрант
ФГБОУ ВО «ГМУ им. адм. Ф.Ф. Ушакова»,
г. Новороссийск, Россия

Аннотация. В работе предложен метод автоматического синтеза правил пакетного фильтра pf на основе анализа потокаNetFlow,получаемого от программного шлюза FreeBSD. Представлен скрипт фильтра, выделяющий устойчивые каналы связности графа локальной вычислительной сети при наличии фонового сканирования портов Nmap. Приведен пример лога nfdump, структура рабочего кода и результаты статистических исследований тестируемого стенда. Показано, что предлагаемый метод позволяет изолировать узлы локальной сети внутри широковещательного домена и исключает попадание сканирований в разрешающие правила.

Ключевые слова: микросегментация, NetFlow, FreeBSD, списки доступа ACL, сканирование портов.

В настоящее времяосновной подход к защите локальных сетей базируется на концепции «замковой защиты» (Castle-and-Moat), при которой основной комплекс защитных мер концентрируется на внешнем периметре с помощью межсетевых экранов (МЭ), систем обнаружения и предотвращения вторжений (IDPS) и VPN-шлюзов. При этом внутреннее пространство сети, как правило, сегментируется лишь укрупненно – путем выделения нескольких виртуальных локальных сетей (VirtualLocalAreaNetwork, VLAN).

Разбив сеть на подсети тегами 802.1Q, удается защититьее периметр, но при этом остается открытым весь внутренний L2-домен: узлы свободно

обращаются по MAC-адресам через общий коммутатор. Таким образом, методы разграничения сетевого доступа, базирующиеся на изоляции широковещательных доменов посредством виртуальных локальных сетей VLAN и статических межсетевых экранов, не предотвращают развитие атак с горизонтальным перемещением нарушителя (lateralmovement) внутри периметра.Как следствие, если на сетевой компьютер попадает вредоноснаяпрограмма, то он сразу начинает опрос соседей по портам 445 (SMB), 135 (RPC) и 3389 (RDP). Для портовых терминалов и судовых сетей,объединяющих навигационные мостики (IBS), автоматику контроля энергоустановок и терминалы спутниковой связи в общую локальную сеть Ethernet, такое горизонтальное перемещение нарушителя оказывается критичным[6]. Создать жесткие списки доступа (ACL) вручном режиме сетевому инженеру, как правило,не позволяет отсутствие документации на используемыепрограммные продукты, а попытки блокировать порты экспериментальным путем приводят к простоям [1].

Для решения указанной проблемы предлагается использовать эвристический алгоритм автоматизированного синтеза списков контроля доступа, включающий этапы пороговой фильтрации сетевого шума, кластеризации ролей хостов и генерации минимально достаточных разрешающих правил в сочетании с терминальным запретом по умолчанию (DefaultDeny). Такой подходпозволяет сократить потенциальную поверхность сетевой атаки при полном сохранении доступности легитимных сервисов, что повышает надежность работы локальной сети[2].Вместо зеркалирования трафика (SPAN), перегружающего накопители PCAP-дампами гигабайтных размеров, предлагается использовать штатный экспорт потоков NetFlowv9 / IPFIX [3]. При этом экспортер собирает на сетевых интерфейсах шлюза только заголовки сессий (IP, порты, TCP-флаги, объемы пакетов).

Однако анализнеобработанных (“сырых”) потоков при формировании правил сталкивается с особенностями работы демонов экспорта. Так, в эксперименте на стенде под управлением FreeBSD 13.2 (виртуальная машина

VirtualBox под управлением дистрибутива МЭ OpenSense: 2 vCPU, 2048 МБ RAM) для экспорта применялся демон softflowd 1.0.0, транслирующий данные на локальный коллектор nfdump 1.7.1. Запуск демона на внутреннем интерфейсе em0 производился строкой:

```
# softflowd -i em0 -n 127.0.0.1:2055 -v 9 -t maxlife=1800s -t expint=30s
```

Проведенные исследования показали, что дефолтное значение inactivetimeout (expint=15s) разрывает длинные соединения: веб-клиент делает паузу при чтении страницы, сокет не закрывается, но экспортер считает поток завершенным и отправляет запись в коллектор. При возобновлении обмена создается новый поток с тем же кортежем адресов. Увеличение expint до 30 секунд позволило объединять такие сессии без лишней фрагментации базы потоков.

Сбор дампа выполнялся утилитой nfcapd в циклический буфер:

```
# nfcapd -w -D -p 2055 -u netflow -g netflow -B 200000 -l /var/log/netflow
```

Как показал проведенный эксперимент в файлевыводанfdump фиксируются не только завершенные сессии, но и служебный шум операционной системы вместе со следами полуоткрытого сканирования портов. На рис.1 приведен фрагмент текстового дампа с коллектора при обращении клиента к веб-серверу на фоне работы сканера Nmap:

```
Date first seen          Duration Proto      Src IP Addr:Port          Dst
IP Addr:Port   Packets    Bytes Flags
2026-04-18 09:14:22.104    1.420 TCP     192.168.10.11:51240 ->
192.168.20.10:80         14     1820 .AP.SF
2026-04-18 09:14:25.881    0.000 TCP     192.168.10.99:43112 ->
192.168.20.25:5432        1       44 ....S.
2026-04-18 09:14:26.885    0.000 TCP     192.168.10.99:43112 ->
192.168.20.25:5432        1       44 ....S.
2026-04-18 09:14:27.115    0.038 UDP     192.168.10.12:61004 ->
192.168.10.1:53           2      148 ......
2026-04-18 09:14:28.012    0.000 UDP     192.168.10.13:137   ->
192.168.10.255:137          1       78 ......
```

Рис.1. Фрагмент коллектора при обращении клиента к веб-серверу на фоне работы сканера Nmap

Из анализа содержимого дампарис.1 следует, что при сканировании закрытого или фильтруемого порта (.20.25:5432) Nmap через секунду делает повторную отправку SYN-пакета (строки 2 и 3, Flags:....S.). Легитимный HTTP-запрос (строка 1) сопровождается полным набором флагов (.AP.SF). При этом в потоке постоянно присутствуют «мусорные» широковещательные пакеты WindowsNetBIOS (строка 5, порт 137 на адрес .255).

Для обработки получаемого лога на языке Python 3.10 был разработан скриптфильтра, выделяющего устойчивые каналы связности графа локальной вычислительной сети при наличии фонового сканирования портов Nmap.Данные выгружаются из бинарных файлов nfcapd командой nfdump -r<файл> -ojson. Для построения ребер графа связности сети по методике [4] адреса отправителей и сервисные порты назначения группируются по ключу K = ⟨src_ip, dst_ip, dst_port, proto⟩. При этом динамические порты клиентов (49152–65535) игнорируются, так как пакетный фильтр привязывается к сервису.

```
  def build_pf_ruleset(aggregated_flows: dict) -> list:
    whitelist_udp = {53, 123, 161}
    blacklist_ports = {137, 138, 5353, 5355, 1900}
    rules = ['set block-policy drop', 'set skip on lo0']
    for key, stats in aggregated_flows.items():
        try:
            src, dst, port, proto = key
            port = int(port)
            if port in blacklist_ports or dst.endswith('.255'):
                continue
            if proto == 'tcp' and stats['packets'] >= 5 and 'S' in
stats['flags']:
                rules.append(f'pass in quick proto tcp from {src} to {dst} '
                             f'port {port} keep state label "acl_{src}_{port}"')
            elif proto == 'udp' and port in whitelist_udp and stats['packets']
>= 2:
                rules.append(f'pass in quick proto udp from {src} to {dst} '
                             f'port {port} keep state')
        except (KeyError, ValueError):
            continue
    rules.append('block in quick all label "default_deny"')
    return rules
```

Рис.2. Функцияpfфильтрации и генерации синтаксиса

Логика работы фильтра и генерации синтаксиса pfбыла реализована в виде отдельной функции с валидацией кортежей и обработкой исключений (рис.2).

При этом использование механизма keepstate обязательно: он заставляет ядро FreeBSD заносить открытую сессию в таблицу состояний и автоматически пропускать обратный трафик с сервера клиенту без ручного добавления обратных правил[5].

В ходе проведения эксперимента в течение 12 часов тестирования стенда пользователи выполняли обычную работу с веб-сервером и синхронизацию времени, а с хоста Kali запускалось сканирование nmap -sS -p 1-10000 и попытки зондирования через Metasploit. Исходный текстовый дамп коллектора составил 16 321 строку (2,1 МБ). При этом в логе были зафиксированы аномалии: 14 поврежденных записей из-за переполнения буфера сокета nfdump

при интенсивном сканировании, а также повторные SYN-пакеты. Результатыобработки лог-файла сведены в табл. 1.

Табл. 1. Результаты селекции потоков и синтеза правил стенда

Категориятрафика	Объем в “сыром” NetFlow	Результатобработки
Штатный веб-трафик (клиенты Nginx)	8 412 записей	2 правила (TCP 80, 443)
Служебныйобмен (Nginx PostgreSQL)	2 145 записей	1 правило (TCP 5432)
DNS-запросы к локальному резолверу	1 120 записей	1 правило (UDP 53)
Широковещательный шум (NetBIOS, LLMNR, mDNS)	2 318 записей	0 (отфильтрован black-листом)
ICMP-эхо (диагностика pingмежду клиентами)	415 записей	0 (заблокированdefault _deny)
Служебные ICMP-ошибки (unreachable/TTL-exceeded)	68 записей	0 (отброшенкактранзиентный)
Сканирование Nmap (-sS, включая повторные SYN)	1 829 записей	0 (отсеяно по порогу S_pkt< 5)
Ошибки парсинга / неполные записи дампа	14 записей	0 (сброшеныблоком try-except)
Итого по “сырому” дампу / сгенерировано ACL	16 321 строка	4 правила PASS + 1 DENY

Конфигурация загружалась в МЭ командой: # pfctl -f /etc/pf.conf.synthesized. Проверка состояния соединений через pfctl -sstate подтвердила штатный пропуск сессий веб-приложения к базе данных. Попытки

повторного сканирования портов СУБД с машины Kali блокировались, а счетчик блокировок в pfctl -sinfo стабильно увеличивался.

Таким образом,результаты исследований показали, что: во-первых, обработка потоков NetFlow средствами nfdump с раздельной пороговой селекцией TCP и UDP позволяет генерировать списки доступа pf без ручного аудита портов; во-вторых, предлагаемый подход позволяет решить задачу изоляции узлов внутри широковещательного домена и исключает попадание сканирований в разрешающие правила.Главное ограничение рассматриваемогометода изоляции связано со сквозным шифрованием: при использовании внутренних VPN-туннелей (IPsec/WireGuard) между хостами сенсор фиксирует только единый транспортный поток туннеля, скрывая внутренние порты. Кроме того, ICMP-трафик требует явного описания типов сообщений (echoreq, echorep) в случае проведения регулярной диагностики доступности.

Литература

1. Репетий Е. О. Микросегментация сети как ключевой элемент стратегии Zero Trust // Научный аспект. 2024. Т. 44, № 4. С. 5788-5792.

2. Baranchikov A. I., Yakovlev I. I., Redkin Yu. V. Improving the reliability of automation systems using data model analysis // AIP Conference Proceedings, Krasnoyarsk. Vol. 2402. Melville, New York, United States of America: AIP Publishing, 2021. P. 50023.

3. Claise B., Trammell B., Aitken P. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information // RFC 7011. InternetEngineeringTaskForce, 2013. 74 p.

4. Khleborodov D. S. High-Granular Micro-Segmentation in Campus Networks based on Downloadable Access Control Lists // ProgrammnayaIngeneria. 2021. Vol. 12, No. 9. P. 443-449. DOI 10.17587/prin.12.443-449.

5. Peter N.M. Hansteen. The Book of PF: A No-Nonsense Guide to the OpenBSD Firewall. 3rd ed. San Francisco: No Starch Press, 2015. 250 p.

6. Redkin Y. V. Redundancy features of real-time maritime information systems based on Ethernet technology // Modern informatization problems in the technological and telecommunication systems analysis and synthesis : Proceedings of the XX-th International Open Science Conference, Yelm, WA, USA / Editor in Chief Dr. Sci., Prof. O.Ja. Kravets. Yelm, WA, USA: Science Book Publishing House LLC, 2015. P. 376-382.