УДК 004.056

АНАЛИЗ БЕЗОПАСНОСТИ ПРОТОКОЛА SNMP ПО СООТНЕСЕНИЮ MIB-ОБЪЕКТОВ С БИНАРНОЙ РЕАЛИЗАЦИЕЙ ПРОШИВОК СЕТЕВЫХ УСТРОЙСТВ

Яцкевич А.А.[1], Скляренко Е.П.[1], Андрушкевич Д.В.[1]
Научный руководитель – кандидат технических наук, Андрушкевич Д.В.[1]
[1]ВКА им. А.Ф. Можайского
vka@mil.ru

Введение

В данной работе рассматривается протокол Simple Network Management Protocol (SNMP), применяемый для мониторинга и удаленного управления сетевыми устройствами. Доступ к параметрам устройства обеспечивается через объекты базы управляющей информации (MIB), организованные в виде дерева идентификаторов объектов (OID). В типовой архитектуре SNMP управляющий узел инициирует запросы к агенту, встроенному в сетевое устройство, а агент формирует ответы на основании значений, получаемых из внутреннего состояния системы [1].

С точки зрения безопасности в работе обосновывается необходимость рассматривать SNMP как удаленно доступный интерфейс управления. Указывается, что любой объект MIB, к которому выполняется запрос, соответствует участку выполняемого кода агента и связанным с ним данным, принимается в качестве базового. Следовательно, свойства безопасности определяются не столько декларативным описанием MIB, сколько реализацией SNMP-агента в прошивке. Для проверки предположений о поведении агента и оценки рисков требуется анализ бинарного образа прошивки методами обратной инженерии [3].

В рамках данной работы предлагается подход к анализу безопасности, основанный на соотнесении объектов MIB (OID) с их программной реализацией в бинарном образе прошивки. Такое соотнесение позволит систематизировать поверхность атаки SNMP и выделить участки кода, требующие приоритетной проверки.

Основная часть

Обработка SNMP-запроса включает разбор сообщения, извлечение OID, поиск соответствующего объекта в иерархии и выполнение обработчика, который формирует значение (операции чтения) или изменяет состояние устройства (операции изменения). Таким образом, в прошивке устройства, поддерживающего SNMP, должны присутствовать: компоненты обработки протокольных сообщений, структуры представления OID и механизм сопоставления «OID – объект», процедуры обработки операций чтения и изменения.

Помимо запросно-ответных операций чтения и изменения, в SNMP используется механизм уведомлений – ловушки (traps). Ловушка представляет собой сообщение, которое агент отправляет менеджеру при наступлении заданного события, передавая набор переменных (varbinds), идентифицируемых OID [1]. С точки зрения безопасности этот механизм расширяет контур удаленного взаимодействия: в прошивке присутствуют процедуры формирования уведомлений, выбора состава передаваемых данных и привязки событий к OID. При анализе рисков целесообразно учитывать корректность формирования набора переменных, ограничения частоты отправки и устойчивость к ситуациям, в которых злоумышленник может вызвать массовую генерацию ловушек (например, через провоцирование повторяющихся событий) [2].

Для задач безопасности в работе проводится разграничение декларативного и реализованного уровней MIB. Декларативный уровень задается MIB-описаниями и

определяет ожидаемый перечень объектов управления. Реализованный уровень определяется фактическим наличием обработчиков запросов в прошивке. Именно реализованный уровень формирует удаленно доступную поверхность атаки, поскольку только для реализованных объектов выполняется код агента.

Соотнесение MIB-объектов и бинарной реализации позволяет выделить практические случаи, значимые для анализа безопасности: объекты, описанные в MIB, но не поддерживаемые реализацией; объекты, поддерживаемые реализацией, но отсутствующие в доступных описаниях; объекты, для которых поведение, права доступа или обработка входных параметров отличаются от ожидаемых. Наиболее критичными с точки зрения рисков являются объекты, допускающие изменение состояния (операции SET), а также табличные объекты, предполагающие обработку индексов и выборку элементов, поскольку такие сценарии повышают требования к корректной валидации входных данных.

Методологически привязка OID к коду прошивки естественным образом опирается на приемы реверс-инжиниринга и анализа бинарных программ: выделение значимых компонентов, восстановление связей между данными и функциями, сопоставление наблюдаемого поведения с участками кода.

В прикладном плане соотнесение MIB и реализации SNMP-агента поддерживает: инвентаризацию реализованных объектов управления; ранжирование объектов по потенциальному риску (например, приоритет SET-объектов); сравнение версий прошивок по изменению реализованного множества объектов; подготовку обоснованных гипотез о потенциально опасных обработчиках для последующей углубленной проверки.

Выводы

Соотнесение MIB-дерева SNMP с бинарной реализацией прошивки позволит рассматривать OID как идентификаторы удаленно доступных точек выполнения кода. Такой подход переводит анализ SNMP из уровня описаний в уровень реализованных обработчиков и обеспечивает более строгую основу для оценки поверхности атаки. Применение методик анализа бинарного кода и реверс-инжиниринга повышает проверяемость выводов и создает предпосылки для сопоставимого анализа версий прошивок.

В дальнейшем планируется автоматизировать процесс привязки OID к участкам бинарного кода SNMP-агента (в том числе с выделением обработчиков запросов и генерации уведомлений), а также расширить методику на другие протоколы и модели управления сетевой конфигурацией и состоянием устройств, включая NETCONF/YANG и RESTCONF.

Литература

1. Одом У. Официальное руководство Cisco по подготовке к сертификационным экзаменам CCENT/CCNA ICDN 100- 101/ У. Одом – Вильямс, 2017. – 912 с.
2. Таненбаум, Э., Уэзеролл Д. Компьютерные сети: учеб. пособие / Э. Таненбаум, Д. Уэзеролл. – СПб.: Питер, 2013. – 960 с.
3. Израилов К. Е. Методология реверс-инжиниринга машинного кода. Часть 1: подготовка объекта исследования // Труды учебных заведений связи. 2023. – URL: https://doi.org/10.31854/1813-324X-2023-9-5-79-90 (дата обращения: 15. 01. 2026).
4. Падарян В. А., Гетьман А. И., Соловьев М. А., Бакулин М. Г., Борзилов А. И., Каушан В. В. Методы и программные средства, поддерживающие комбинированный анализ бинарного кода // Труды Института системного программирования РАН. 2014. – URL: https://doi.org/10.15514/ISPRAS-2014-26(1)-8 (дата обращения: 20. 01. 2026).