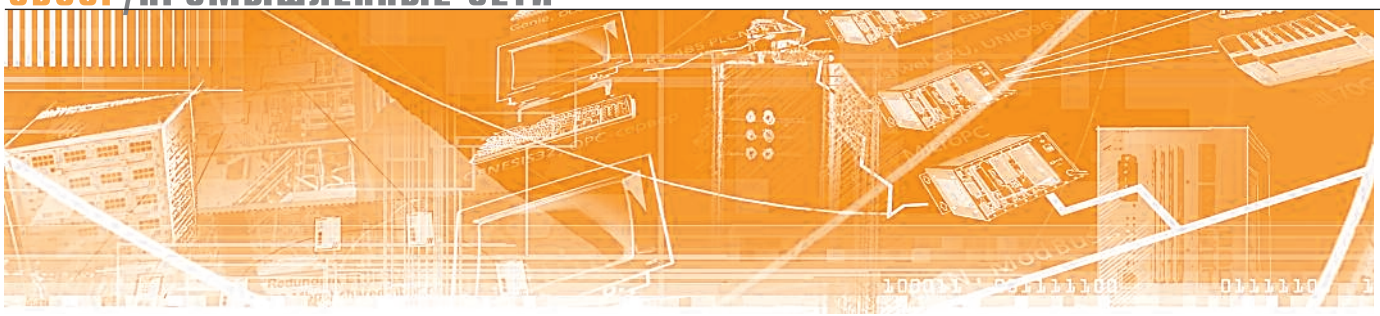

Сергей Воробьёв

“Defense in Depth” в действии. Уровень 4: защита промышленных протоколов

Часть 1

Данный материал служит продолжением цикла статей, посвящённых многоуровневой защите промышленных Ethernet-сетей на базе принципа “Defense in Depth”. В статье рассмотрен ряд базовых уязвимостей промышленных протоколов Modbus TCP и OPC Classic, а также методы защиты, основанные на глубокой инспекции трафика.

Введение

Тщательная и глубокая проверка данных, передаваемых по промышленной Ethernet-сети, является следующим уровнем защиты согласно принципу “Defense in Depth” [1]. Фактически это узконаправленный механизм защиты, который позволяет нейтрализовать угрозы, направленные на оконечные устройства.

Известно, что датчики, ПЛК, HMI – это устройства, которые функционируют на базе хорошо известных промышленных протоколов Modbus TCP, Ethernet/IP, DNP3 и других. И если анализировать IP-пакет данных, например, протокола Modbus TCP, с которым работает датчик или ПЛК, то можно констатировать, что вся необходимая служебная информация находится внутри пакета. Если злоумышленник или вредоносное ПО использует узкоспециализированные шаблоны атак, которые направлены на изменение передаваемых данных внутри промышленных протоколов, то это может привести не только к полной потере контроля над оконечными устройствами, но и к компрометации передаваемых данных. Практических примеров подобных инцидентов достаточно много, начиная от воровства топлива на АЗС путём передачи ложных данных о температуре окружающей среды, заканчивая выходом из строя критически важных узлов и агрегатов промышленного предприятия. Наглядной демонстрацией последнего может служить широко известный промышленный вирус Stuxnet, а также авария на сталелитейном заводе в Германии. В первом случае это привело к выходу из строя центрифуг для обогащения урана, во втором к застыванию доменной печи.

Методики защиты, описанные в предыдущих статьях цикла, позволяют обеспечить защиту от самых многочисленных и разнообразных угроз, но если атака направлена на протокол передачи данных, на изменение служебной информации, содержащейся в передаваемых пакетах, и атакующий уже получил доступ к сети, то нужен иной инструмент анализа и защиты. Средства, основанные на использовании классического L3- или L2-брандмауэра, не позволяют это реализовать, так как принцип их работы основан на проверке заголовка в начале пакета либо фрейма. Брандмауэр, функционирующий на уровне L3, согласно модели OSI позволит внести ограничение на уровне порта, например, полностью закрыть протокол Modbus TCP путём ввода ограничений на передачу по порту 502. Это даст возможность отключить множество ненужных клиентов от оконечных устройств, но не предотвратит передачу ложных данных. Необходим иной подход, который позволит «залезть» внутрь пакета и проанализировать передаваемые данные на уровне регистров протокола. Решение, которое поможет преодолеть данную проблему и защитить оконечное устройство, – это проверка пакетов данных (packet inspection). Осуществлять её необходимо непосредственно на верхних уровнях модели OSI при помощи специализированных брандмауэров. При работе подобного устройства каждый пакет передаваемых данных полностью распаковывается и проверяется на уровне протоколов и полезной нагрузки. А задержки, которые очень критичны в промышленной сети, должны быть сведены к минимуму.

Далее в качестве примера подобного брандмауэра рассмотрим функциональность программно-аппаратного комплекса Tofino Xenon от компании Hirschmann (рис. 1), который позволяет защитить не только промышленные, но и проприетарные протоколы различных устройств, работающих по Ethernet-сети.

SPI и DPI: в чём различие?

Начнём с того, что сейчас встречаются два достаточно близких термина, относящихся к проверке данных, которые содержатся в IP-пакетах, это SPI – Stateful Packet Inspection и DPI – Deep Packet

Рис. 1. Внешний вид брандмауэра Tofino Xenon

Inspection. SPI можно дословно перевести как инспекция пакетов с хранением состояния. Брандмауэр, в котором заявлена поддержка SPI, является пакетным фильтром, анализирующим данные на транспортном уровне модели OSI.

Изначально технология SPI создавалась, исходя из необходимости защитить сессию протокола TCP/IP. Когда протокол TCP создаёт сессию с другим сетевым устройством, используется определённый порт на устройстве с противоположной стороны, также открывается порт на исходном устройстве-отправителе. В соответствии со спецификацией TCP-порт отправителя будет некоторым числом, большим чем 1023 и меньшим чем 16384. Порт назначения на удалённом устройстве имеет фиксированный номер. Например, для SMTP это будет 25, для Modbus TCP – 502. Смысл SPI – это реализация пакетного фильтра, который должен разрешать либо запрещать трафик по определённым портам. Один из примеров настройки правила для пакетного фильтра (не SPI и DPI) – это разрешение на пропуск всего входящего трафика для портов с большими номерами, так как это будут возвращаемые пакеты от системы назначения. Но подобное открытие портов создаёт риск несанкционированного проникновения в локальную сеть [2].

Брандмауэры с поддержкой SPI решают эту проблему путём создания списка для исходящих TCP-соединений, соответствующих каждой сессии. Данный список затем используется для проверки допустимости любого входящего трафика. В сущности, если у брандмауэра заявлена функциональность SPI, это добавляет анализ транспортного уровня в архитектуру пакетного фильтра. Пример подобного брандмауэра был описан в статье [3]. Как правило, подобная функциональность необходима на границе сети.

Немного иной принцип работы у брандмауэров с поддержкой Deep Packet Inspection. DPI – это глубокая проверка данных не только на сетевом и транспортном уровне, как в случае с SPI, но и проверка на всех вышестоящих уровнях модели OSI, включая прикладной (рис. 2). Это очень ресурсозатратный про-цесс, который, как правило, требует существенных вычислительных мощностей. При этом зачастую возникает вопрос относительно того, откуда брать данные и что анализировать? Сейчас существует ряд подходов, которые используют разработчики DPI-систем, один из самых распространённых – это получение данных из SPAN-порта коммутатора (Switch Port Analyzer). Подключив к нему мощную платформу для проверки и анализа трафика, можно выявить многие процессы, происходящие в сети. Но это решение не всегда является хорошим, так как при анализе данных необходимо чётко понимать структуру сети, какие именно данные проходят в данном сегменте сети, что является входящим и исходящим потоком данных, а также откуда их нужно брать. При этом знание протокола передачи должно быть достаточно глубоким и применимым в реальной системе. Если рассмотреть популярный промышленный протокол Ethernet/IP (EIP – Ethernet Industrial Protocol), который используется в сетях ControlNet и DeviceNet, то можно констатировать, что для анализа данных, помимо стандартного ряда параметров, присущих любой Ethernet-сети, необходимо учитывать достаточно большое количество служебной информации, передающейся в пакете (Class ID, Member ID, Service ID, Connection Point, Port Seg Number, Data Seg Number, CIP service, Instance ID и т.д.) Подобный перечень индикаторов потенциально вредоносных действий можно обозначить практически для любого промышленного протокола. Помимо Ethernet/IP сюда можно отнести Modbus TCP, OPC Classic, DNP3, IEC104, GOOSE и т.д., и достаточно большое количество проприетарных протоколов, данные которых могут быть скомпрометированы, например, WAGO CODESYS, S7-COMM, Emerson DeltaV и т.д.

В итоге можно констатировать, что DPI – это комплексная и сложная проверка на уровне данных, которые несут полезную нагрузку в промышленных протоколах. Для её реализации необходимо чётко понимать структуру передаваемой информации, как на уровне стандартов, так и на уровне возможных отклонений от них, а также присутствие их в том или ином сегменте сети. При этом помимо проверки данных необходимо анализировать полученную информацию о нетипичном поведении протоколов, вовремя сигнализировать об этом поведении и предотвращать подобные ситуации.

Tofino Xenon – невидимый защитник промышленных протоколов

Одним из примеров устройства, в котором реализована технология DPI, является программно-аппаратный комплекс **Tofino Xenon**. Комплекс состоит из аппаратной платформы и программного обеспечения. Аппаратная платформа реализована в промышленном исполнении, при этом является конфигурируемой и позволяет подстроиться под существующую сетевую инфраструктуру. Комплекс устанавливается в разрыв сети и может быть оснащён как оптическими, так и медными портами типа RJ-45 со скоростью до 100 Мбит/с. Из важных особенностей аппаратной части можно отметить пропускную способность, которая составляет 2000 пакетов в секунду, а также то, что комплекс является полностью прозрачным на сетевом уровне,

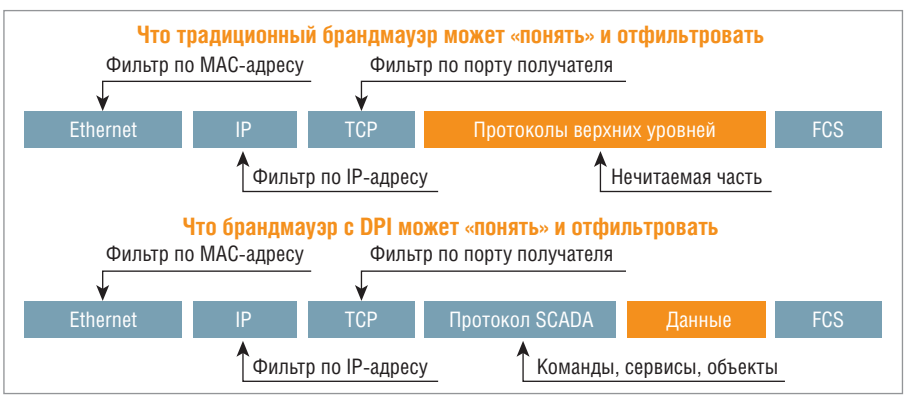

Рис. 2. Отличия принципа работы DPI-брандмауэра от традиционного

Таблица 1

Краткий список протоколов, поддерживаемых Tofino Xenon

Производитель	Серия	Серверы и приложения	Протоколы
1	2	3	4
ABB	AC400	Engineering Workstation	ABB Time
	AC800M	Operator Workspace	MI
		Controller Network Interface	RNRP
		Server Network Interface	RemSys
		Aspect Server	
		Domain Controller	
		Multisystem Integration Server	
Allen-Bradley	CompactLogix		Ethernet/IP (CIP)
	ControlLogix		Rockwell CSP (TCP and UDP)
	PLC-5		
	SLC-5		
Cisco	1600, 1800, 2600, 2800, 3800 routers		
	ASA&PIX firewalls		
Emerson	DeltaV		DeltaV
	Provox		
GE	90-30		GE QuickPanel Configuration Protocol
	90-70		GE SRTP
	VersaMAX		MOST/PAC8000API
	VersaMAX Micro		
	PAC8000 SafetyNet		
Промышленные IT-протоколы		Wonderware HMI	DNP3
			FF Fieldbus Message Specification
			FF system management
			GOOSE-IEC61850
			IEC MMS
			IEC 60870-5-104
			IEEE 1588 PTP
			ISO networklayer protocol;
			MRP (Media Redundancy Protocol)
			OPC Classic TCP
IT-протоколы			DHCP (клиент и сервер)
			DNS
			FTP
			HTTP (web)
			HTTPS
			ICMP (ping)
			IGMP
			Intel NIC Teaming Protocol
			IPv6
			Kerberos Authentication
			LDAP
			LLDP
			LLMNR
			NETBIOS Datagram
			NETBIOS Name Resolution
			NETBIOS Session Service
			Network Time Protocol (NTP)
			Novell Netware Protocol
			Remote Replication Agent
			Reverse ARP
			SMB
			SNMP
			SNMP Trap
			Spanning Tree Protocol (STP)
			SSH Secure Shell Protocol
			Symantech AV
			Telnet
			TFTP
			UPnP (TCP and UDP)
			VRRP
			WS-Discovery
			WSUS

Окончание таблицы 1

1	2	3	4
Hirschmann	OpenRail		Hiper Ring Protocol
	RSR		Hirschmann redundant Ring Coupling
	MICE		
	BAT Wireless		
	Octopus		
	RS40		
	MACH 100		
	MACH 1000		
	MACH 3000		
	MACH 4000		
	LION		
	RAIL Video		
Honeywell	C200		Honeywell CDA
	C300		Honeywell FTE
			Honeywell Safety Manager
			PLANTSCAPE
Mitsubishi			Mitsubishi MELSCQNA
Omron			FINS (UDP)
OSISoft		PI Data Historian	PI Data Historian
Schneider Electric	Momentum		Modbus TCP
	Premium		Modbus UDP
	Quantum		
	Twido Nano		
Siemens	SIMATIC S7/200/1200/		
	300/400(FH)/C		
WAGO	750-842 PLC		WAGO CODESYS
Yokogawa	Cendum		Yokogawa Stardom
	Stardom		Vnet/IP

Таблица 2

Модули Tofino Xenon

Наименование	Краткое описание программного модуля
Tofino Firewall LSM	Базовый модуль, включающий возможность анализа промышленных, IT и проприетарных протоколов
NetConnect	Модуль для возможности удалённого конфигурирования
Tofino Modbus TCP Enforcer LSM	Модуль для глубокого анализа протокола Modbus TCP
Tofino OPC Classic Enforcer LSM	Модуль для глубокого анализа протокола OPC Classic
Tofino Xenon IEC 104 Enforcer LSM	Модуль для глубокого анализа протокола IEC104
Tofino Xenon DNP3 Enforcer LSM	Модуль для глубокого анализа протокола DNP3
Tofino EtherNet/IP Enforcer LSM	Модуль для глубокого анализа протокола Ethernet/IP
Tofino Xenon GOOSE Enforcer Loadable Security Module (LSM)	Модуль для глубокого анализа протокола GOOSE

у него нет IP-адреса и определить его наличие в сети практически невозможно. При этом у устройства присутствует режим тестирования, который помогает проверять правила трафика без какого-либо риска случайного блокирования сообщений, имеющих решающее значение для работы оконечных устройств.

Конфигурирование Tofino Xenon можно осуществить как удалённо, так и записав конфигурацию на специализированный USB-носитель. Программная часть состоит из ПО для конфигурирования – Tofino Configurator, которое предназначено для комплексной настройки параметров безопасности сети и программных модулей. Программные модули определяют функциональность межсетевого экрана Tofino Xenon. При этом для каждого устройства возможно сформировать индивидуальный набор модулей, в зависимости от требований, предъявляемых к конкретному сегменту сети. Программных модулей несколько, и они предназначены для различных промышленных протоколов, рассмотрим более подробно каждый из них.

Tofino Firewall LSM

Данный модуль является базовым для всех устройств Tofino и фактически позволяет управлять трафиком, пропускать либо блокировать его. В процессе работы происходит проверка всех коммуникаций в сети по контрольному списку данных, куда входит IP- и MAC-адрес, а также, что немаловажно, тип сетевого протокола, в том числе промышленного. Любое сообщение, которое не входит в список разрешённых, будет заблокировано, а информация о нём отправлена в виде log-файла.

Данный модуль содержит предварительно определённые шаблоны для более чем 25 популярных промышленных ПЛК, включая правила для защиты устройств с известными уязвимостями (табл. 1).

Регистратор событий (Event Logger) также по умолчанию включён в данный модуль. Регистратор событий контролирует события, которые происходят в промышленной сети, а также отправляет сигналы тревоги. Система регистрации событий может как отправлять сообщения об угрозах на удалённый сервер syslog, так и хранить список в энергонезависимой памяти Tofino. Также в составе комплекса Tofino Xenon могут быть установлены модули группы Enforcer, которые позволяют проводить анализ и тонкую настройку фильтров для промышленных протоков (табл. 2).

Tofino Modbus TCP Enforcer LSM

Модуль Modbus TCP Enforcer позволяет осуществить анализ трафика Modbus TCP на достаточно глубоком уровне – уровне регистров передаваемых данных. Необходимость подобной проверки связана с тем, что Modbus, наверно, самый «хороший» пример по уязвимости именно промышленных протоколов. Это связано с тем, что протокол Modbus, который применяется на огромном количестве предприятий, известен ещё с 70-х годов прошлого века. Впервые спецификация была опубликована компанией Modicon в 1979 году, сейчас эта компания называется Schneider Electric. И начиная с момента публикации спецификации, этот протокол набирал популярность и проник практически во все сферы промышленности. Можно сказать, что сейчас Modbus – это стандарт де-факто среди промышленных протоколов. При этом протокол достаточно простой и лёгкий в реализации. Огромное количество производителей ПЛК и оконечных устройств используют его (Schneider Elecric, Advantech, ABB, FASTWEL, Emerson, WAGO и т.д.). При этом Modbus настолько популярен, что многие noname-производители имеют поддержку именно Modbus. И было бы всё хорошо, но в те далёкие годы, когда спецификация протокола была разработана, никто в принципе не думал о безопасности. В качестве линии для передачи данных использовались RS-232/485, всё было изолировано внутри промышленного объекта либо цеха.

В результате увеличения скоростей передачи данных и доли интеллектуальных устройств Modbus решили перевести на стек TCP, и в результате этого в XXI веке Modbus представлен в виде протокола Modbus TCP. Но что изменилось? Да в принципе изменений совсем немного. Modbus TCP – это по-прежнему протокол типа «запрос-ответ», на установку соединения в нём ничего не завязано. В спецификации, конечно, есть пункт про установку соединения и дальнейшую его поддержку, где указано, что не следует разрывать его после каждого ответа, но это рекомендуется делать только для оптимизации, чтобы избежать «торможения». Если заглянуть внутрь пакета, то существенные значения: идентификатор устройства, код операции и данные (зависят от операции) – остались без изменения. И вроде бы поле «идентификатор устройства» логически должно отвечать за базовую безопасность, но, увы, оно используется не для защиты, а лишь только для адресации. Это поле используется и в протоколе Modbus RTU, и в Modbus TCP. В случае с TCP-версией оно либо игнорируется при непосредственном подключении, либо в дальнейшем используется шлюзом для маршрутизации. Относительно поля «код операции» можно сказать, что в TCP-версии при ответе устройства в нём может содержаться код ошибки, либо может быть полное дублирование отправленной информации, что наиболее вероятно. Получается, что при переходе к TCP-версии протокол фактически не изменился. Modbus-данные упаковываются в Ethernet-пакет, и используется порт 502. Все плюсы и минусы, присущие изначальной версии протокола, остались неизменными. Шифрование, аутентификация, авторизация – это всё не про Modbus. Но есть один момент с безопасностью, который всё-таки прописан в спецификации: указано, что на критически важных объектах связывающиеся узлы должны проверять друг друга по IP-адресу.

Если рассмотреть функциональность Modbus, то можно выделить три большие группы функций, которые позволяют нам узнать про устройство: стандартные (прописаны в спецификации), зарезервированные и пользовательские, последние вендор использует по своему усмотрению. В разрезе безопасности и защиты протокола стоит рассматривать лишь первый тип. К нему относится доступ к данным – чтение/запись из регистров. Также стандартно доступен достаточно большой список диагностических функций, который различен для разных каналов связи. Для TCP-версии наиболее интересна функция device identification, то есть система присвоения уникального идентификатора устройству. В стандарте прописано, что устройство должно сообщить о себе ряд обязательных (vendor name, product code, MajorMinorRevision) и необязательных данных (vendorUrl, ProductName, ModelName, UserApplicationName). Но стандарты зачастую не соблюдаются. Кто-то из производителей передаёт их, кто-то не передаёт, а кто-то передаёт, но другими способами. Например, используя ПО Modbus Device Identificator, можно просканировать всю сеть и определить абсолютно все Modbus-устройства, которые там используются. При этом подобных утилит очень много: пара приложений ModSim/ModScan, fuzzing-утилиты для поиска уязвимостей и ряд других, не стоит забывать про всем известные Wireshark и Python. Последняя позволяет очень просто создать скрипт (листинг 1), передающий Modbus-данные в Ethernet-сеть.

В итоге можно сделать вывод, что, имея доступ к Ethernet-сети без каких-либо дополнительных уровней защиты, можно довольно просто просканировать все Modbus-устройства и, например, обнулить все регистры, либо точечно передать ложные данные.

Ситуацию может спасти DPI-проверка данных. Но и тут не всё так однозначно, простота и удобство Modbus-протокола несут определённые сложности в реализации его защиты. Важно понимать, что для TCP протокол Modbus – это конвейер данных, информация передаётся байт за байтом. И устройство, вы-

Листинг 1. Пример скрипта для передачи информации по протоколу Modbus TCP

```
# скрипт, позволяющий передать ложные данные
from pyModbusTCP.client import ModbusClient
import time
c = ModbusClient()
c.host("192.168.1.1")
c.port(502)
c.open()
while True:
    address = 12288
    while 1 < 2:
        c.write_single_coil(address,0)
        c.write_single_register(12293,0)
        c.write_single_register(12291,100)
        regs=c.read_input_registers(12291,4)

    if regs is not None:
        print(regs)
    else:
        print("Fail!")
    time.sleep(1)
```

EtherWAN
www.etherwan.com

Высокоскоростные удлинители Ethernet
с питанием по сигнальной линии

PoE-камера

IEEE 802.3at / IEEE 802.3af

Питание +48/55 В

Модель ED3538T – удлинитель Ethernet по VDSL с передачей питания по сигнальному кабелю

Модель ED3538R – удлинитель Ethernet по VDSL с питанием от сигнального кабеля и передачей PoE-питания конечному устройству

- Передача питания для обратного преобразователя и конечного устройства на расстояние до 1300 м
- Скорость передачи данных по технологии Ethernet-over-VDSL до 100 Мбит/с
- Передача до 30 Вт на конечное устройство по PoE
- Удлинение Ethernet по двухжильному кабелю на расстояние до 2200 м
- Работа при температурах –40…+75°C

Характеристики моста ED3538T – ED3538R с включенным питанием по сигнальной линии

Дистанция между удлинителями (м)	Скорость передачи данных по VDSL (Мбит/с)	Мощность для конечного PoE-устройства (Вт)
300	100	30
600	60	14
800	45	9,5
1200	20	5

Характеристики моста ED3538T – ED3538R с автономным питанием каждого удлинителя

Дистанция между удлинителями (м)	Скорость передачи данных по VDSL (Мбит/с)	Мощность для конечного PoE-устройства (Вт)
1400	15	30
1600	10	30
1800	33	0
< 2200	13	0

полняющее DPI, должно поверять каждый Modbus-пакет в Ethernet-пакете, фактически разворачивать и собирать пакет данных с полезной нагрузкой, как матрёшку. Ведь ложные данные могут содержаться в последовательности передаваемых Modbus-пакетов. Помимо механизмов фильтрации также необходимо проводить аналитику тех событий, которые происходят в сети при использовании промышленных протоколов.

В Modbus TCP индикаторами наличия вредоносных действий могут выступать следующие события:

- наличие Modbus-соединений, которые являются нетипичными для данной зоны;
- наличие неудачных попыток установки TCP/UDP-соединения по порту 502;
- сканирование порта 502 в широком диапазоне адресов;
- наличие команды сканирования от slave-устройства;
- использование команд, специфичных для различных производителей;
- поток пакетов данных ADU (Application Data Unit) с множеством различных команд;
- нетипичные команды;
- непоследовательная история полученных данных в ответах устройств;
- передача Modbus-данных в обход DMZ;
- трафик Modbus с использованием протокола UDP.

В модуле Tofino Modbus TCP Enforcer LSM реализована достаточно богатая функциональность, которая позволяет защитить протокол Modbus TCP. Фактически, используя данный модуль, можно обеспечить защиту данных на уровне регистров. Для каждого оконечного устройства можно задать ряд правил, которые будут включать разрешение или запрет доступа.

Создаваемое для протокола «правило на доступ» будет включать следующие параметры:

- связка IP/MAC-адрес;
- перечень значимых регистров;
- тип возможных операций (запрет доступа, только чтение, только запись, запись/чтение);
- идентификатор устройства;
- наличие проверки базовых команд (1–6, 15, 16, 20–24);
- установка контроля соединения;
- политики исключений;
- реакция в случае блокировки сообщения.

Сформировав данный список разрешённых запросов (рис. 3), можно защитить Modbus- устройство. Tofino Modbus TCP Enforcer LSM является «инспектором» контента для Modbus-протокола. При работе происходит полная DPI-проверка каждого Modbus-запроса и ответа, как для входящего, так и для исходящего трафика. Любая команда, которая не находится в списке разрешённых, или любая попытка доступа к регистрам данных, которая находится за пределами разрешённого диапазона запросов, блокируется, о чём сообщается на специальный IP-адрес. В итоге установка устройства Tofino Xenon с модулем Modbus TCP Enforcer LSM позволит не только защитить сеть на уровне протокола, но и повысить надёжность и снизить нагрузку сети.

Tofino OPC Classic Enforcer LSM

Далее перейдём к модулю, который предназначен для защиты OPC-сервера. OPC (Open Platform Communications) – семейство программных технологий,

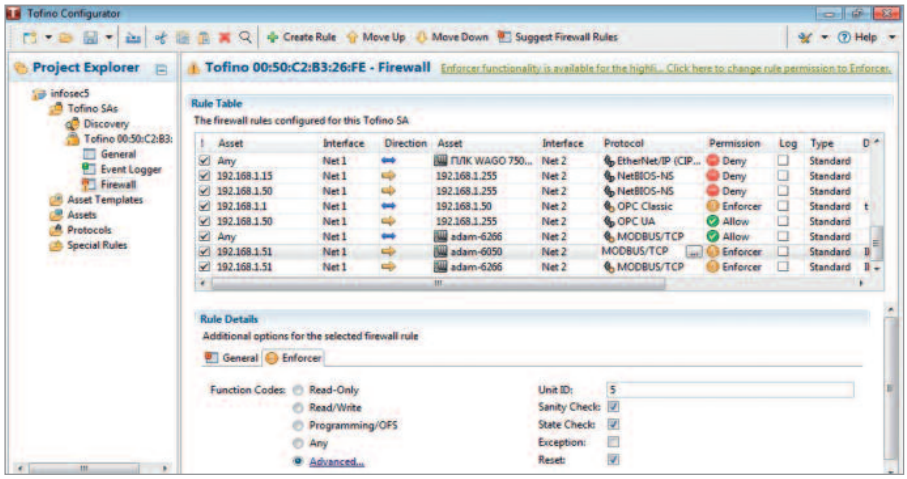

Рис. 3. Настройка DPI-фильтра для протокола Modbus TCP

предоставляющих единый интерфейс для управления объектами автоматизации и технологическими процессами. Технология OPC подразумевает работу по принципу клиент-сервер. В качестве OPC-клиента выступает программа диспетчеризации (SCADA) либо HMI (Human-Machine Interface), а OPC-сервер служит связующим звеном между клиентом и оконечными устройствами. OPC-серверы взаимодейству-ют с коммуникационными протоколами (Modbus, Profibus, Interbus, CAN-Bus и т.д.). Эта технология позволяет организовать доступ к данным промышленных систем автоматизации. Другими словами, благодаря стандартизации интерфейса стало возможным подключение любого физического устройства к любой SCADA-системе, если оно соответствует стандарту OPC. Но как обстоят дела с безопасностью сервера? Ведь он, по сути, является ключевой фигурой для связи между устройствами. Начнём с того, что сейчас доступны две технологии: OPC UA и OPC Classic. Первая является принципиально новым набором спецификаций, которая имеет достаточно широкий диапазон средств безопасности, от простой аутентификации с помощью пароля и обмена цифровыми подписями до полного шифрования передаваемых сообщений [4].

А вот с технологией OPC Classic всё достаточно непросто, особенно в плане безопасности. OPC Classic не определяет безопасность как часть каких-либо спецификаций интерфейса. По умолчанию OPC Classic построен на основе «транспорта» DCOM/COM от Microsoft (начиная с 1996 года). Сильно упрощая, можно сказать, что сервер экспортирует функции, которые клиент может вызывать, вынуждая сервер выполнить то или иное действие [5].

При этом связь организуется при помощи динамического распределения портов. Это и является основной уязвимостью OPC Classic. Протокол Modbus TCP использует порт 502, а HTTP использует порт 80, их редко кто-то меняет. А у OPC Classic диапазон номеров возможных портов может варьироваться от 1024 до 65535. При каждом открытии се-

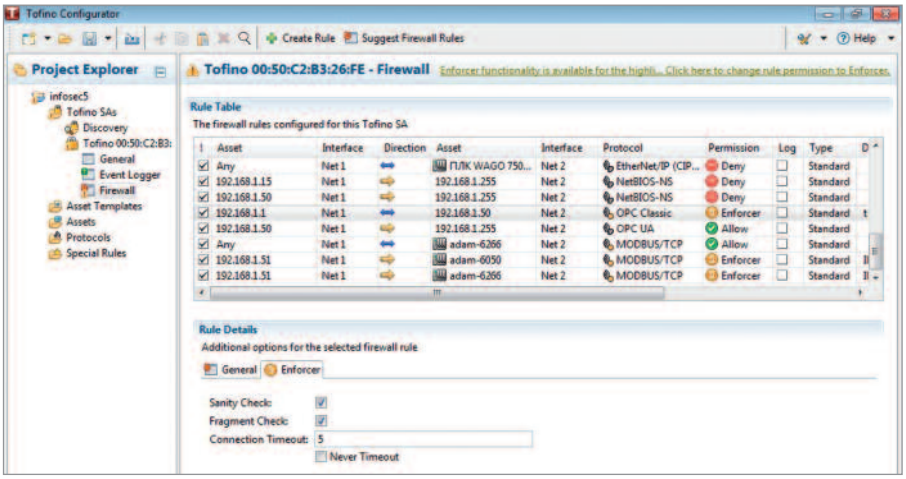

Рис. 4. Настройка DPI-фильтра для протокола OPC Classic

анса связи OPC-клиентом происходит динамическое назначение нового номера порта. Подключившись к OPC-серверу, OPC-клиент запрашивает номер TCP-порта, который должен быть использован для этой сессии. Затем производится новое соединение и заново отправляется запрос на номер свободного порта. По этой причине протокол OPC сложно и почти невозможно защитить с помощью классических стандартных брандмауэров, так как при настройках нужно будет открыть большой диапазон портов. Кроме того, в OPC Classic сервер должен иметь возможность инициировать связь с клиентом для обратных запросов, требующих доступа с сервера к клиенту. Эта функциональность приводит к тому, что все OPC-клиенты также настроены так, как будто бы они были OPC-сервером, а все OPC-серверы настроены так, как если бы они были OPC-клиентами.

Другими словами, для возможности работы OPC необходимо открыть практически все порты брандмауэра в обоих направлениях. В целом сервер OPC Classic может быть сконфигурирован так, чтобы обеспечить достаточную степень безопасности, но всегда стоит помнить, что она обеспечивается функциональностью DCOM/COM.

Один из вариантов решения данной проблемы – это контроль удалённого вызова процедур со стороны клиента (RPC – Remote Procedure Call, класс технологий, позволяющих вызывать функции или процедуры в другом адресном пространстве). На транспортном уровне RPC используют в основном протоколы TCP и UDP. RPC может быть настроен так, чтобы либо ограничивать диапазон используемых портов, либо статически назначать фиксированный порт данному серверу OPC Classic. Но назначение фиксированного порта может не работать на различных версиях OPC Classic. В итоге такое решение необходимо дополнительно тестировать, чтобы убедиться в его работоспособности. Иной вариант решения – это контроль пользователем OPC каждого соединения с помощью DPI-инспекции. Необходимо контролировать за-

просы на подключение к серверу, обратные ответы к клиенту, фактически контролировать всю сессию, которая включает такие параметры, как тип сообщения, номер порта, адрес OPC-сервера, адрес OPC-клиента. При подобном подходе можно обеспечить защиту OPC-сервера.

Примером подобного решения, которое может контролировать сессию OPC Classic, основанную на технологии Microsoft DCOM/COM, является программный модуль Tofino OPC Classic Enforcer LSM в составе программно-аппаратного комплекса Tofino Xenon. Tofino OPC Classic Enforcer LSM проверяет, отслеживает и защищает каждое соединение, созданное приложением OPC. Комплекс динамически открывает только TCP-порты, необходимые для каждого соединения, и только между конкретным OPC-клиентом и сервером, который создал соединение (рис. 4). При настройке данных нет необходимости изменений конфигурации на клиентах и серверах OPC.

В качестве дополнительной защиты со стороны OPC-клиента (как правило, это машина на которой установлена SCADA-система) желательно использовать хороший антивирус, ограничивающий на уровне системы управления эфирные окна сеанса обмена запросами и ответами между OPC-клиентами и серверами.

Заключение

Глубокая проверка данных (DPI) на уровне промышленных протоколов является защитой, способной распознать самые изощрённые угрозы. Многие промышленные протоколы, такие как Modbus и OPC Classic, имеют ряд уязвимостей, которые могут привести к печальным и очень затратным последствиям.

Один из вариантов защиты – это глубокий анализ специфичных данных, присущих тому или иному протоколу. Программно-аппаратный комплекс Tofino Xenon – один из примеров устройства, которое может обеспечить реальную защиту промышленных протоколов и оконечных устройств.

В данной статье были рассмотрены модули для защиты протоколов Modbus TCP и OPC Classic. В следующей части статьи будут описаны типичные уязвимости для протоколов Ethernet/IP, IEC104, DNP3 и GOOSE, а также механизмы их защиты. ●

Литература

1. Воробьёв С. Глубокая защита промышленного сетевого периметра // Современные технологии автоматизации. – 2017. – № 4.
2. Классификация firewall'ов и определение политики firewall'а [Электронный ресурс] // Режим доступа : https://www.intuit.ru/studies/courses/20/20/lecture/625?page=6.
3. Воробьёв С. "Defense in Depth" в действии. Уровень 1: защита границы сети // Современные технологии автоматизации. – 2017. – № 4.
4. Спецификация OPC UA [Электронный ресурс] // Режим доступа : http://www.bookasutp.ru/Chapter9_2_4.aspx.
5. Введение в COM/DCOM [Электронный ресурс] // http://www.delphikingdom.ru/asp/viewitem.asp?catalogid=1108.

Автор – сотрудник фирмы ПРОСОФТ
Телефон: (495) 234-0636
E-mail: info@prosoft.ru