



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Немања Малиновић

Етичко хаковање индустријских управљачких система употребом ControlThings платформе

ДИПЛОМСКИ РАД
- Основне академске студије -


Нови Сад, 2023

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:	
Идентификациони број, ИБР:	
Тип документације, ТД:	Монографска публикација
Тип записа, ТЗ:	Текстуални штампани документ/ЦД
Врста рада, ВР:	Завршни-bachelor рад
Аутор, АУ:	Немања Малиновић
Ментор, МН:	Проф. др Имре Лендак
Наслов рада, НР:	Етичко хаковање индустријских управљачких система употребом ControlThings платформе
Језик публикације, ЈП:	Српски(ћирилица)
Језик извода, ЈИ:	Српски/Енглески
Земља публикавања, ЗП:	Србија
Уже географско подручје, УГП:	Војводина
Година, ГО:	2023
Издавач, ИЗ:	Ауторски репринт
Место и адреса, МА:	Факултет Техничких Наука (ФТН), Д. Обрадовића 6, 21000 Нови Сад
Физички опис рада, ФО: (поглавља/страна/ цитата/табела/слика/графика/прилога)	5/26/0/2/21/0/0
Научна област, НО:	Електротехничко и рачунарско инжењерство
Научна дисциплина, НД:	Примењено софтверско инжењерство
Предметна одредница/Кључне речи, ПО:	Modbus, PLC, мрежа, етичко хаковање SCADA система
УДК	
Чува се, ЧУ:	Библиотека ФТН, Д. Обрадовића 6, 21000 Нови Сад
Важна напомена, ВН:	
Извод, ИЗ:	Рад се бави изучавањем мера безбедности индустријских управљачких система кроз офанзивну безбедност. Одабрано је неколико сценарија напада на индустријске управљачке системе и њихова примена. Детаљи ових сценарија су изложени у раду.
Датум прихватања теме, ДП:	
Датум одбране, ДО:	
Чланови комисије, КО:	Председник: Др Стеван Гостојић, редовни професор
	Члан: Др Александар Селаков, ванредни професор
	Члан, ментор: Др Имре Лендак, редовни професор
	Потпис ментора

KEY WORDS DOCUMENTATION

Accession number, ANO :			
Identification number, INO :			
Document type, DT :	Monographic publication		
Type of record, TR :	Textual material, printed/CD		
Contents code, CC :	Bachelor thesis		
Author, AU :	Nemanja Malinović		
Mentor, MN :	Imre Lendak, PhD, full professor		
Title, TI :	Ethical hacking of industrial control systems with the ControlThings platform		
Language of text, LT :	Serbian (cyrillic script)		
Language of abstract, LA :	Serbian/English		
Country of publication, CP :	Serbia		
Locality of publication, LP :	Vojvodina		
Publication year, PY :	2023		
Publisher, PB :	Author reprint		
Publication place, PP :	Faculty of Technical Sciences, D. Obradovića 6, 21000 Novi Sad		
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	5/26/0/2/21/0/0		
Scientific field, SF :	Electrical and computer engineering		
Scientific discipline, SD :	Power Software Engineering		
Subject/Key words, S/KW :	Modbus, PLC, network, ethical hacking of SCADA systems		
UC			
Holding data, HD :	Library of the Faculty of Technical Sciences, D. Obradovića 6, 21000 Novi Sad		
Note, N :			
Abstract, AB :	<p>The paper deals with the study of security measures of industrial control systems through offensive security. Several attack scenarios on industrial control systems and their application have been selected. Details of these scenarios are presented in the paper.</p>		
Accepted by the Scientific Board on, ASB :			
Defended on, DE :			
Defended Board, DB :	President:	Stevan Gostojić, PhD, full professor	Menthor's sign
	Member:	Aleksandar Selakov, PhD, associate professor	
	Member, Mentor:	Imre Lendak, PhD, full professor	

	УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА 21000 НОВИ САД, Трг Доситеја Обрадовића 6	Број:
	ЗАДАТАК ЗА ИЗРАДУ ДИПЛОМСКОГ (BACHELOR) РАДА	Датум:

(Податке уноси предметни наставник - ментор)

Врста студија:	<input type="checkbox"/> Основне академске студије
Студијски програм:	Примењено софтверско инжењерство
Руководилац студијског програма:	Доц. др Александар Селаков

Студент:	Немања Малиновић	Број индекса:	PR108/2019
Област:	Сигурност и безбедност		
Ментор:	Проф. др Имре Лендак		

НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА ЗАВРШНИ (Bachelor) РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА:

- проблем – тема рада;
- начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна;
- литература

НАСЛОВ ДИПЛОМСКОГ (BACHELOR) РАДА:

Етичко хаковање индустријских управљачких система употребом ControlThings платформе
--

ТЕКСТ ЗАДАТКА:

Задатак дипломског рада је анализа ControlThings платформе за испитивање безбедности индустријских контролних система. Експериментални део рада ослонити на употребу следећих алата: Wireshark, ModbusPal, mbtget i nmap.
--

Руководилац студијског програма:	Ментор рада:

Примерак за: <input type="radio"/> - Студента; <input type="radio"/> - Ментора
--

Образац Q2.НА.15-04 - Издање 2

Списак коришћених скраћеница

Скраћеница	Значење скраћенице
ICS	Industrial Control Systems (Индустријски Управљачки Системи).
USB	Universal Serial Bus (УСБ диск за пренос података).
SCADA	Supervisory Control And Data Acquisition (Систем за контролу и аквизицију података).
PLC	Programmable Logic Controller (Програмабилни логички контролер).
DCS	Distributed Control System (Дистрибуирани систем за контролу и управљање).
IoT	Internet of Things (Мрежа уређаја која је повезана и ти уређаји размењују информације једни са другим).
IIoT	Industrial Internet of Things (Мрежа индустријских уређаја која је повезана и ти уређаји размењују информације једни са другим).

Табела 1 Табела коришћених скраћеница.

Садржај

1. Увод.....	1
2. Опис индустријских управљачких система и платформе.....	2
3. Опис коришћених алата, технологија и протокола	3
3.1. Modbus протокол.....	3
3.2. Wireshark	4
3.3. Metasploit framework.....	4
3.4. Nmap	4
3.5. Mbtget наредбе.....	5
3.6. ModbusPal	5
4. Експерименти.....	6
4.1. Mbtget експеримент.....	6
4.2. Metasploit framework експеримент.....	8
4.3. Снимци мрежног саобраћаја	12
4.4. Nmap експеримент	14
4.5. Гугл SCADA хаковање.....	16
5. Закључак.....	18
Литература.....	19
Подаци о кандидату	20

1. Увод

У данашњем дигиталном добу, индустријски управљачки системи (*ICS*) играју кључну улогу у одржавању функционалности и безбедности различитих индустријских сектора. *ICS* системи омогућавају управљање разним процесима, од производње електричне енергије и управљања фабрикама, до транспорта и обраде воде. Са све већим повезивањем ових система са интернетом, њихова рањивост на разне претње постаје забрињавајућа ствар.

У прошлости безбедност *SCADA* система није привлачила велику пажњу програмера. Међутим, 2010. године догодио се значајан напад у области безбедности информационих система. Злонамерни компјутерски црв, познат под именом *STUXNET*, први пут је откривен те године и верује се да је нарушио ирански нуклеарни програм у значајној мери. Један од највероватнијих начина на који је црв унесен у систем био је *USB* диск. Током тог периода, примарни акценат био је стављен на физичку безбедност људи у постројењима, а мало пажње се обраћало на могуће нападе са удаљених тачака, као и на нападе у сајбер простору.

Индустријски управљачки систем користи сензоре и актуаторе који омогућавају интеракцију са физичким светом (нпр. сензор за притисак, вентили, мотори). *PLC* је дигитални компјутер (контролер) који се користи за аутоматизацију у реалном времену. Има већи број аналогних и дигиталних улаза и излаза, имуни су на вибрацију, температуру, прашину и сл.

Још један аспект безбедности односи се на комуникационе протоколе који се користе у оваквим системима. На примеру *Modbus* протокола, сигурносне заштите нису интегрисане у протокол, што може омогућити неовлашћен приступ и пренос података у читљивом тексту.

У складу са растућим значајем *ICS*-а, питање етичког хаковања ових система постаје све актуелније. Етичко хаковање се користи за идентификацију рањивости и слабости система како би се унапредила њихова безбедност. У том контексту, платформа *ControlThings* представља моћан алат који омогућава истраживачима, инжењерима и безбедносним стручњацима да анализирају, тестирају и унапређују *ICS* системе.

Треба напоменути да етичко хаковање или пентестирање обухвата активности испитивања и анализе информационих система и мрежа са дозволом и сагласношћу власника или одговорне организације. Експерименте треба изводити у контролисаним условима.

Овај дипломски рад ће истражити тему етичког хаковања индустријских управљачких система употребом платформе *ControlThings* и разматрати сценарије напада на овакве системе користећи методе офанзивне безбедности.

2. Опис индустријских управљачких система и платформе

Индустријски управљачки системи представљају кључни сет компјутерских хардверских и софтверских компоненти који су одговорни за надгледање, контролисање и управљање индустријским процесима и системима. Они су од суштинског значаја за различите области индустрије, укључујући производњу, енергетику, снабдевање водом и отпадним водама, транспорт и многе друге.

Елементи ових система укључују:

- Сензори и актуатори: Овај део система обухвата физичке компоненте које су одговорне за прикупљање података из околине (сензори) и извршавање одређених акција на основу добијених информација (актуатори).
- Регулатори: Они обезбеђују контролу над различитим процесима коришћењем података прикупљених од сензора и корисничких упутстава.
- *HMI – (Human Machine Interface)*: Представља софтверску или хардверску компоненту која омогућава комуникацију између оператера и индустријског управљачког система.
- Комуникациони системи: Омогућавају размену података између компоненти индустријских управљачких система.
- Контролни системи: Обухвата хардвер и софтверске делове који обрађују податке, доносе одлуке и издају команде.

Ови системи су кључни за функционисање различитих критичних система. На пример, индустријски управљачки систем користи се за надгледање и контролу производних постројења, енергетских централа и сличних постројења.

Пошто су ови системи од изузетног значаја за индустрију, њихова безбедност је такође критична. Због специфичних захтева ових система, њихова заштита и тестови морају бити познати и адаптирани за све околности, што изискује специјализоване алате и вештине.

ControlThings платформа је *Kali Linux* дистрибуција која је специјално опремљена за процењивање безбедности и пенетрационо тестирање индустријских управљачких система. Поседује алате које садржи свака линукс дистрибуција, као што су *metasploit*, *nmap*, и слично. Поред ових алата, на платформи се налази неколико фолдера:

- *Crypto*: фајлови за крипто вежбе и експерименте.
- *Datasheets*: примери табела за стандардне чипове.
- *Firmware*: за разне хардверске уређаје које *ControlThings* платформа може користити.
- *Memory*: примери дампова (*eng. Dumps*) од *EEPROM*, *Flash* и *RAM*.
- *Protocols*: опис протокола који се користе на платформи и снимци мрежног саобраћаја.
- *RadioFrequencies*: конфигурација и примери за радио-фреквентну комуникацију.
- *ReadingRoom*: водич кроз документе о сајбер безбедности из разних извора.
- *Simulators*: конфигурације за симулационе софтвере који се користе на платформи.
- *UserInterfaces*: примери нестандартних корисничких интерфејса.

У овом раду биће коришћени фолдери *Crypto*, *Protocols* и *Simulators*.

3. Опис коришћених алата, технологија и протокола

Овај део рада садржи комплетан преглед коришћених алата и технологија који су имали кључну улогу у извршавању анализе и тестова. Од софтверских платформи до специјализованих апликација, сваки коришћен алат ће бити детаљно описан. Од многих протокола који се користе у индустријским управљачким системима биће разматран само *Modbus* протокол и то *TCP* верзија.

3.1. *Modbus* протокол

Modbus протокол је захтев-одговор (*eng. request-response*) протокол који је имплементиран користећи везу господар-слуга (*eng. master-slave*). Један уређај иницира захтев па затим чека одговор где је уређај који иницира (господар) одговоран за сваку интеракцију. Овај протокол је формиран у касним седамдесетим годинама прошлог века. Типови *Modbus* објеката могу бити *Coil*(читање и упис једног бита), *Discrete Input*(само читање једног бита), *Input Register*(само читање шеснаест битова - читање речи), *Holding Register*(читање и упис шеснаест битова - читање и упис речи).

Верзија овог протокола која је значајна за овај рад је *Modbus TCP*. На слици испод је приказан *Modbus TCP* фрејм формат. Порт који овај протокол користи је 502.

Назив	Дужина (у бајтима)	Функција
Идентификатор трансакције	2	За синхронизацију порука између клијента и сервера.
Идентификатор протокола	2	0 за <i>Modbus/TCP</i> .
Дужина поља	2	Број преосталих бајтова у овом оквиру.
Идентификатор јединице	1	Адреса сервера (255 ако није у употреби).
Код функције	1	Кодови разних функција.
Бајтови података	n	Подаци који представљају одговор или команду.

Слика 1 *Modbus/TCP* формат оквира.

Неки кодови за функције (*eng. Function codes*) су: 1 за *READ_COILS*, 5 за *WRITE_SINGLE_COIL*, 3 за *READ_HOLDING_REGISTERS*, 6 за *WRITE_SINGLE_REGISTER*, 16 за *WRITE_MULTIPLE_REGISTERS*. Постоје још многи кодови за функције али ово су кодови од највећег значаја за овај рад.

Овај протокол има широку примену и данас али у основној верзији није безбедан. У овом раду слабости аутентификације и слабост чистог текста (*eng. Clear text*) биће показане кроз експерименте.

3.2. Wireshark

Представља бесплатан пакет анализатор, отвореног кода. Користи се за снимање и анализу мрежног саобраћаја.

У овом раду биће коришћен за аквизицију мрежног саобраћаја *Modbus TCP* протокола.

3.3. Metasploit framework

Metasploit framework пружа информације о безбедносним рањивостима и помаже у тестирању и пенетрацији и развоју система детекције упада (*eng. Intrusion Detection System*). У власништву је компаније *Rapid7* из Бостона у Масачусетсу, САД. Састоји се из већег броја модула који имају широк спектар употребе. У овом раду биће узети у обзир модули који се тичу *SCADA* система. Ови могући се могу видети у табели испод:

Име модула	Опис
auxiliary/analyze/modbus_zip	Извлачење .zip из <i>Modbus</i> комуникације.
auxiliary/scanner/scada/modbus_banner_grabbing	„ <i>Modbus banner grabbing</i> “ техника за добијање основних информација о <i>Modbus</i> уређајима у мрежи.
auxiliary/scanner/scada/modbusclient	Омогућава кориснику да комуницира са уређајима који користе <i>Modbus</i> протокол.
auxiliary/scanner/scada/modbus_findunitid	Користи се да би се установила комуникација са конкретним уређајем пре него што се пошаљу <i>Modbus</i> команде.
auxiliary/scanner/scada/modbusdetect	Користи се за откривање верзије <i>Modbus</i> -а.

Табела 2 *Metasploit* модули за *SCADA* системе и њихов опис.

Модули се могу користити искључиво у *metasploit* конзоли. Наредба коју је потребно унети у линукс терминал за покретање ове конзоле је *msfconsole*. Након покретања конзоле, модул можемо користити тако што искористимо наредбу *use <имеМодула>*.

Након коришћења модула можемо покренути команду *show actions* како би смо видели које све функције овај модул садржи. Командом *set action* дефинишемо која функционалност ће бити искоришћена. Употреба ових функционалности ће бити детаљно приказана у поглављу експерименти.

3.4. Nmap

Nmap је скенер за откривање хостова и сервиса на рачунарским мрежама, чиме се ствара „мапа“ мреже. *Nmap* шаље специјално креиране пакете на хост а затим анализира одговоре. Злоупотреба овог алата је законом кажњива.

Овај алат и његове функције биће разматрани детаљно у поглављу експерименти.

3.5. Mbtget наредбе

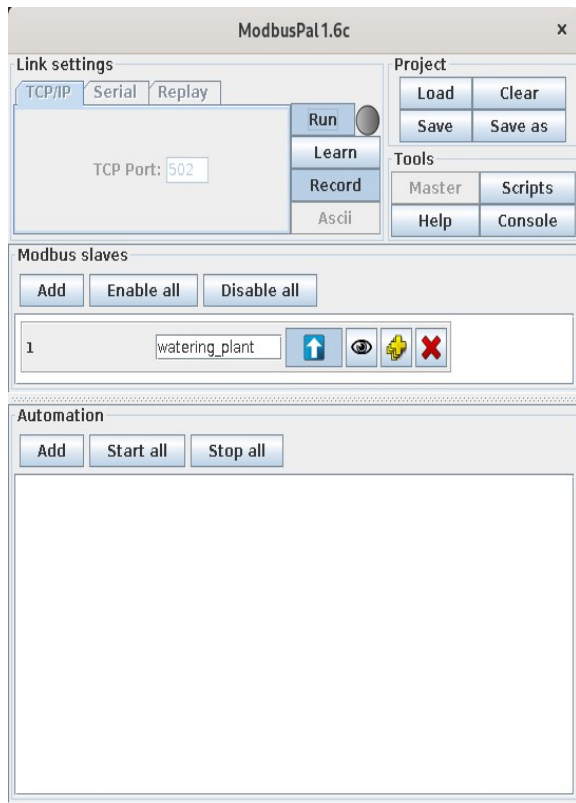
Mbtget је специјално дизајниран за тестирање и за интеракцију са *Modbus* уређајима. Дозвољава читање и упис података из/у *Modbus* регистре. Експлоатише слабост аутентификације *Modbus* протокола, на исти начин као и *metasploit* конзола, тако што омогућава приступ и читање података из *Modbus* уређаја.

Употреба ових наредби ће бити објашњена кроз експерименте.

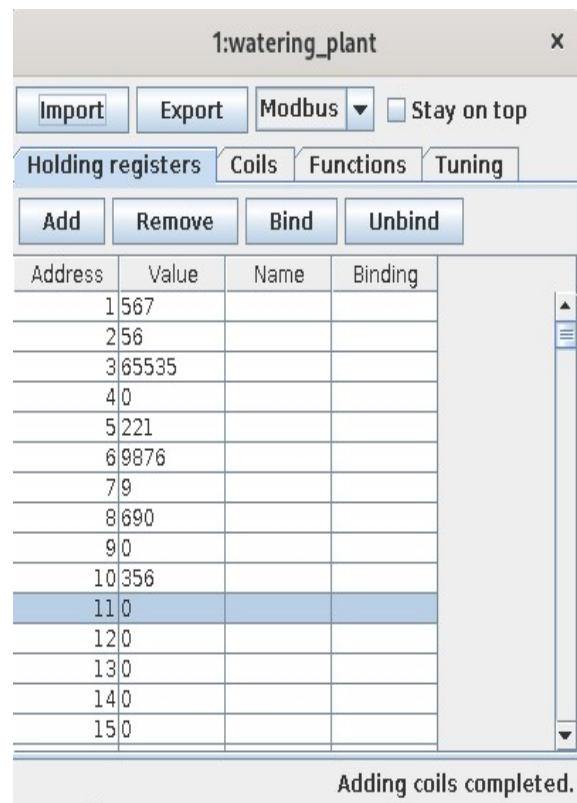
3.6. ModbusPal

Ова софтверска апликација представља алат за симулацију уређаја који користе *Modbus* протокол за комуникацију. Ова симулациона алатка омогућава корисницима да тестирају и развијају своје апликације или системе које користе *Modbus* комуникацију без потребе за реалним уређајима. Омогућава емуляцију разних типова уређаја као што су сензори, актуатори, регулатори и сличне компоненте. Корисници могу подесити вредности различитих регистара и статуса које би уређаји користили у реалном времену. На сликама 2 и 3 се може видети изглед апликације.

У овом раду, овај симулатор је коришћен за имитирање слуга (*eng. Slaves*) којима ће неовлашћено приступати нападач-тестер преко *metasploit* конзоле и *mbtget* наредби.



Слика 2 Интефејс симулатора.



Слика 3 Интерфејс за унос вредности регистара.

4. Експерименти

У поглављу експерименти, фокус ће бити на конкретним истраживачким активностима и тестовима који су извршени у овом раду. Овде ћемо разматрати методологију, поступке и резултате сваког експеримента са циљем дубљег разумевања и анализе аспеката сајбер безбедности индустријских управљачких система.

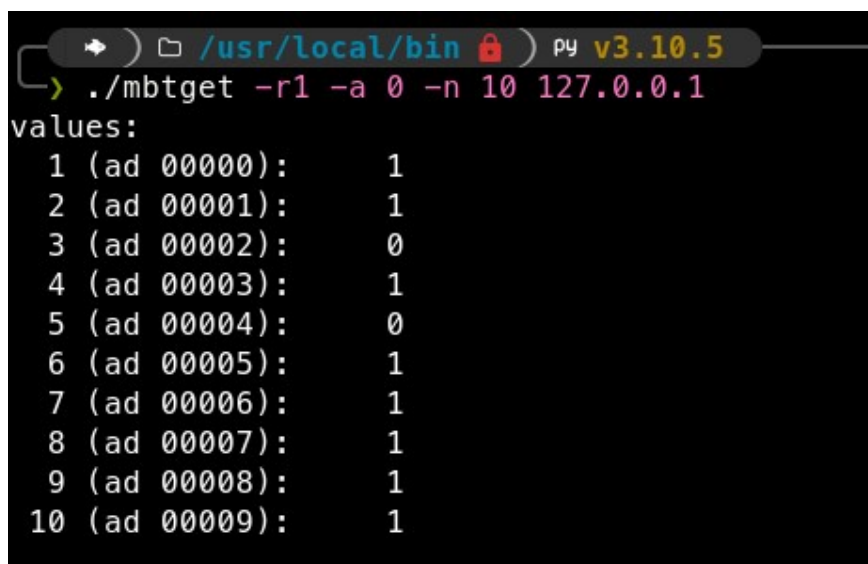
При извршавању сваког експеримента, примењене су одређене технике, алати и праксе са циљем остваривање конкретних исхода и тестова. У наставку су изложени детаљи и резултати сваког експеримента, како би се тема етичког хаковања што боље разумела.

4.1. *Mbtget* експеримент

Циљ овог експеримента је експлоатисање слабости *Modbus* протокола. То је планирано учинити као неаутентификовани нападач који приступа *ModbusPal* симулатору и покушава да прочита и измени критичне вредности регистра симулатора које симулирају реалне критичне вредности са којима рукују *PLC* контролери у реалном систему користећи пакет *mbtget*.

Приступ овом симулатору се врши тако што апликација *ModbusPal* постоји и ради на порту 502., *IP* адреса ове апликације је *Loopback* адреса. Вредност 127.0.0.1. представља вредност *loopback* адресе.

У *ModbusPal* симулатору креирано је десет *coils* регистра и десет *holding* регистра. У наставку видимо употребљену команду.



```

> ./mbtget -r1 -a 0 -n 10 127.0.0.1
values:
 1 (ad 00000):      1
 2 (ad 00001):      1
 3 (ad 00002):      0
 4 (ad 00003):      1
 5 (ad 00004):      0
 6 (ad 00005):      1
 7 (ad 00006):      1
 8 (ad 00007):      1
 9 (ad 00008):      1
10 (ad 00009):      1

```

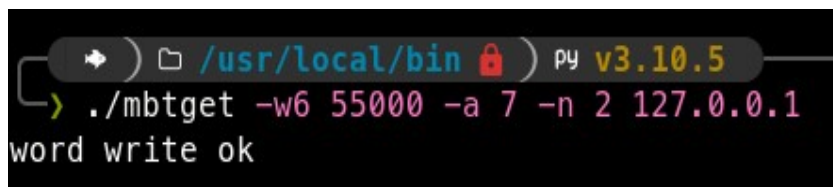
Слика 4 Команда за читање 10 *coils* регистра и резултат.

Команда са претходне слике има параметре:

- Извршни фајл „./mbtget“.
- Функција за читање *coils* регистра „-r1“.
- Адреса на уређају од које се врши читање. Овде је адреса нула. „-a 0“.
- Број регистра који ће бити прочитан. Овде читамо десет. „-n 10“.
- *IP* адреса уређаја са ког ће бити прочитани подаци. „127.0.0.1.“.

Ова команда тражи од *Modbus* уређаја на локалном рачунару (*IP* адреса 127.0.0.1) да прочита вредности десет регистара почевши од адресе нула користећи функцију за читање *coils* регистара.

Биће приказан још један случај овог пута са командом за упис у *holding* регистре. Покушај да се модификује нека критична вредност може се видети на слици испод.

A screenshot of a terminal window with a dark background. The prompt shows the current directory as /usr/local/bin and the Python version as 3.10.5. The command entered is ./mbtget -w6 55000 -a 7 -n 2 127.0.0.1. The output of the command is word write ok.

```
➡ ) /usr/local/bin ) py v3.10.5  
> ./mbtget -w6 55000 -a 7 -n 2 127.0.0.1  
word write ok
```

Слика 5 Команда за упис у *holding* регистре.

Команда са претходне слике извршава упис вредности 55000 у регистар на адреси број седам на *IP* адреси 127.0.0.1. Као резултат ове операције видимо да је вредност успешно уписана односно измењена.

Тестирањем на овај начин, види се да било ко са знањем да рукује оваквим алатом може да чита и модификује критичне вредности са којима овај систем управља. На пример, ако је вредност коју нападач може да измени критична вредност одговорна за регулацију температуре у нуклеарној електрани. Тада непрописана и недозвољена могућност за приступ и измену критичне вредности може имати катастрофалне последице и може угрожавати безбедност људи и животне средине.

Због оваквих потенцијалних опасности, од изузетне важности је применити безбедносне мере и механизме који ће спречити или открити потенцијалне нападе.

4.2. Metasploit framework експеримент

У оквиру овог експеримента, фокус ће бити на експлоатисању слабости *Modbus* протокола користећи *msfconsole*. Циљ је истраживање потенцијала овог алата и његових модула у контексту индустријских управљачких система као и како овај алат може бити примењен за процену безбедности таквих система.

Након што је конзола покренута врши се претрага свих модула везаних за индустријске управљачке системе (*SCADA* системе). Ово је могуће урадити уз помоћ наредбе *search modbus* која ће приказати све модуле који користе *Modbus* протокол, као што је приказано у табели 1 из поглавља 3.3.

Први модул који је коришћен је *auxiliary/scanner/scada/modbusclient*. Овај модул омогућава кориснику да комуницира са уређајима који користе *Modbus* протокол. У овом случају *ModbusPal* симулатор симулира један *modbus* уређај. Акције које овај модул садржи излистане су наредбом „*show actions*“ и могу се видети на слици испод:

```
msf6 auxiliary(scanner/scada/modbusclient) > show actions

Auxiliary actions:

  Name                Description
  ----                -
  READ_COILS           Read bits from several coils
  READ_DISCRETE_INPUTS Read bits from several DISCRETE INPUTS
  READ_HOLDING_REGISTERS Read words from several HOLDING registers
  READ_ID              Read device id
  READ_INPUT_REGISTERS Read words from several INPUT registers
  WRITE_COIL           Write one bit to a coil
  WRITE_COILS          Write bits to several coils
  WRITE_REGISTER       Write one word to a register
  WRITE_REGISTERS      Write words to several registers
```

Слика 6 Излистане акције модула *auxiliary/scanner/scada/modbusclient*.

Симулатор је покренут и врши се покушај читања критичних вредности из *holding* регистара. Потребно је сетовати одговарајућу акцију наредбом „*set action READ_HOLDING_REGISTERS*“. Свака акција има своје опције које је потребно поставити на одговарајуће вредности. Да би се приказале све могуће опције једне акције потребно је унети наредбу „*show options*“. На следећој слици су излистане све могуће опције за акцију читања холдинг регистара.

Module options (auxiliary/scanner/scada/modbusclient):

Name	Current Setting	Required	Description
DATA		no	Data to write (WRITE_COIL and WRITE_REGISTER modes only)
DATA_ADDRESS		yes	Modbus data address
DATA_COILS		no	Data in binary to write (WRITE_COILS mode only) e.g. 0110
DATA_REGISTERS		no	Words to write to each register separated with a comma (WRITE_REGISTERS mode only) e.g. 1,2,3,4
HEXDUMP	false	no	Print hex dump of response
NUMBER	1	no	Number of coils/registers to read (READ_COILS, READ_DISCRETE_INPUTS, READ_HOLDING_REGISTERS, READ_INPUT_REGISTERS modes only)
RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	502	yes	The target port (TCP)
UNIT_NUMBER	1	no	Modbus unit number

Auxiliary action:

Name	Description
READ_HOLDING_REGISTERS	Read words from several HOLDING registers

Слика 7 Опције за акцију **READ_HOLDING_REGISTERS**.

Порт који је таргетован је аутоматски постављен на порт на ком функционише *Modbus* протокол (502). Потребно је поставити *IP* адресу уређаја који треба тестирати, у овом случају то је *loopback* адреса јер се хост налази на виртуелној машини са адресом 127.0.0.1. Како би ово било учињено потребно је употребити наредбу „set RHOSTS 127.0.0.1“. Потребно је поставити адресу на *Modbus* уређају од које ће вредности бити прочитане. Ово се извршава наредбом „set DATA_ADDRESS“. Након тога уколико је потребно читати више вредности регистара од једног, мора се изменити опција *NUMBER* употребом команде „set NUMBER“. Када су све опције и параметри успешно постављени на жељене вредности потребно је употребити команду *run* или *exploit*. Обе команде имају исто значење, а то је да покрену акцију и изврше читање критичних вредности из *holding* регистара. Резултат ове акције се може видети на слици испод.

```
msf6 auxiliary(scanner/scada/modbusclient) > set NUMBER 9
NUMBER => 9
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 127.0.0.1

[*] 127.0.0.1:502 - Sending READ HOLDING REGISTERS...
[+] 127.0.0.1:502 - 9 register values from address 1 :
[+] 127.0.0.1:502 - [690, 17899, 5678, 65535, 0, 1, 55000, 4566, 13434]
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/scada/modbusclient) > exploit
[*] Running module against 127.0.0.1
```

Слика 8 Резултат акције **READ_HOLDING_REGISTERS**.

Резултат је враћен у облику вредности 9 регистара са почетком од адресе број 1. Адресе на *modbus* уређају почињу од адресе 0. Као што се може видети, поново је експлоатисана слабост аутентификације овог протокола где било ко, са знањем да рукује овим алатом, може приступити и прочитати осетљиве вредности процесних променљивих.

У наставку овог експеримента биће показана акција уписа у *holding* регистар *WRITE_REGISTER*. Ова акција поново има своје опције које је потребно попунити. Потребно је поставити податке за следеће опције:

- *RHOSTS* је потребно поставити поново на 127.0.0.1.
- *DATA* представља податак који треба да се унесе у регистар.
- *DATA_ADDRESS* представља адресу на коју ће податак бити унесен.

На пример, уколико се покуша унос вредности 5355 на адресу 3, након валидно постављених параметара и покретањем акције уз помоћ наредби *run* или *exploit* видљив је следећи резултат.

```
msf6 auxiliary(scanner/scada/modbusclient) > set ACTION WRITE_REGISTER
ACTION => WRITE_REGISTER
msf6 auxiliary(scanner/scada/modbusclient) > set DATA 5355
DATA => 5355
msf6 auxiliary(scanner/scada/modbusclient) > set DATA_ADDRESS 3
DATA_ADDRESS => 3
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 127.0.0.1
[*] 127.0.0.1:502 - Sending WRITE REGISTER...
[+] 127.0.0.1:502 - Value 5355 successfully written at registry address 3
[*] Auxiliary module execution completed
```

Слика 9 Резултат акције *WRITE_REGISTER*

Поново је видљива слабост овог протокола јер је недозвољено измењена критична вредност регистра.

Следећи модул који је коришћен је *auxiliary/scanner/scada/modbus_banner_grabbing*. „Modbus banner grabbing“ техника се користи за добијање основних информација о *Modbus* уређајима у мрежи. Уколико постоји *modbus* уређај у мрежи тада ће га модул детектовати. Како на овој патформи има само један *modbus* уређај(симулатор) овај модул би требао да га детектује уколико је незаштићен. Ово може бити врло корисно нападачима у фази извиђања (*eng. reconnaissance*) добијају информације о уређајима у систему за који планирају даљи напад. У наставку се може видети примена овог модула са његовим опцијама.

```
msf6 auxiliary(scanner/scada/modbus_banner_grabbing) > show options
Module options (auxiliary/scanner/scada/modbus_banner_grabbing):

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS    127.0.0.1        yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT     502              yes       The target port (TCP)
  THREADS   1                yes       The number of concurrent threads (max one per host)
  TIMEOUT   2                yes       Timeout for the network probe

msf6 auxiliary(scanner/scada/modbus_banner_grabbing) > set RHOSTS 127.0.0.1
RHOSTS => 127.0.0.1
msf6 auxiliary(scanner/scada/modbus_banner_grabbing) > run
[-] 127.0.0.1:502 - MODBUS - Network error during payload: The connection with 127.0.0.1:502 timed out.
[*] 127.0.0.1:502 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Слика 10 Резултат коришћења модула *auxiliary/scanner/scada/modbus_banner_grabbing*.

Постављањем *RHOSTS* поља и покретањем команде *run* овај модул проналази информације о *Modbus* уређајима у мрежи.

Следећи модул се употребљава искључиво за проверу да ли се у систему користи *Modbus* протокол. На следећој слици су приказане опције и извршавање овог модула.

```
Module options (auxiliary/scanner/scada/modbusdetect):
```

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	502	yes	The target port (TCP)
THREADS	1	yes	The number of concurrent threads (max one per host)
TIMEOUT	10	yes	Timeout for the network probe
UNIT_ID	1	yes	ModBus Unit Identifier, 1..255, most often 1

```
msf6 auxiliary(scanner/scada/modbusdetect) > set RHOSTS 127.0.0.1
RHOSTS => 127.0.0.1
msf6 auxiliary(scanner/scada/modbusdetect) > run

[+] 127.0.0.1:502 - 127.0.0.1:502 - MODBUS - received correct MODBUS/TCP header (unit-ID: 1)
[*] 127.0.0.1:502 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Слика 11 Резултат коришћења модула *auxiliary/scanner/scada/modbusdetect*.

Ово може бити врло корисно нападачима у фази извиђања (*eng. reconnaissance*) где нападач анализира и сакупља информације које ће му помоћи у следећим фазама напада. Извршено је идентификација и откривање поверљивих информација и анализирана је и експлоатисана рањивост. Добро извиђање може значајно повећати шансе за успех сајбер напада. Од изузетне је важности имати јаке безбедносне мере и практиковати одговарајућу обуку како би се спречили овакви видови напада.

Последњи модул који ће бити изучаван у овом раду кроз експерименте јесте *auxiliary/scanner/scada/modbus_findunitid*. Користи се да би се установила комуникација са конкретним уређајем пре него што се пошаљу *Modbus* команде. Омогућава проналажење свих идентификатора уређаја који се налазе у мрежи која је таргетована. Могуће је навести опсег у ком желимо да тражимо идентификаторе и он је у распону од 1 до 254, као и тајмаут између скенираних идентификатора. У наставку је приказана примена овог модула и његових опција.

```
Module options (auxiliary/scanner/scada/modbus_findunitid):
```

Name	Current Setting	Required	Description
BENICE	1	yes	Seconds to sleep between StationID-probes, just for beeing nice
RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	502	yes	The target port (TCP)
TIMEOUT	2	yes	Timeout for the network probe, 0 means no timeout
UNIT_ID_FROM	1	yes	ModBus Unit Identifier scan from value [1..254]
UNIT_ID_TO	254	yes	ModBus Unit Identifier scan to value [UNIT_ID_FROM..254]

```
msf6 auxiliary(scanner/scada/modbus_findunitid) > set RHOSTS 127.0.0.1
RHOSTS => 127.0.0.1
msf6 auxiliary(scanner/scada/modbus_findunitid) > run
[*] Running module against 127.0.0.1

[+] 127.0.0.1:502 - Received: correct MODBUS/TCP from stationID 1
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 2 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 3 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 4 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 5 (probably not in use)
```

Слика 12 Резултат коришћења модула *auxiliary/scanner/scada/modbus_findunitid*.

Ово скенирање ће се наставити све док не претражи свх 254 идентификатора. Као што је приказано, успешно је детектован симулатор који се налази у систему са идентификатором број 1.

Уколико је потребно поништити неку постављену вредност тада се користи наредба *unset*. За излаз из *msfconsole* потребно је унети наредбу *exit*.

4.3. Снимци мрежног саобраћаја

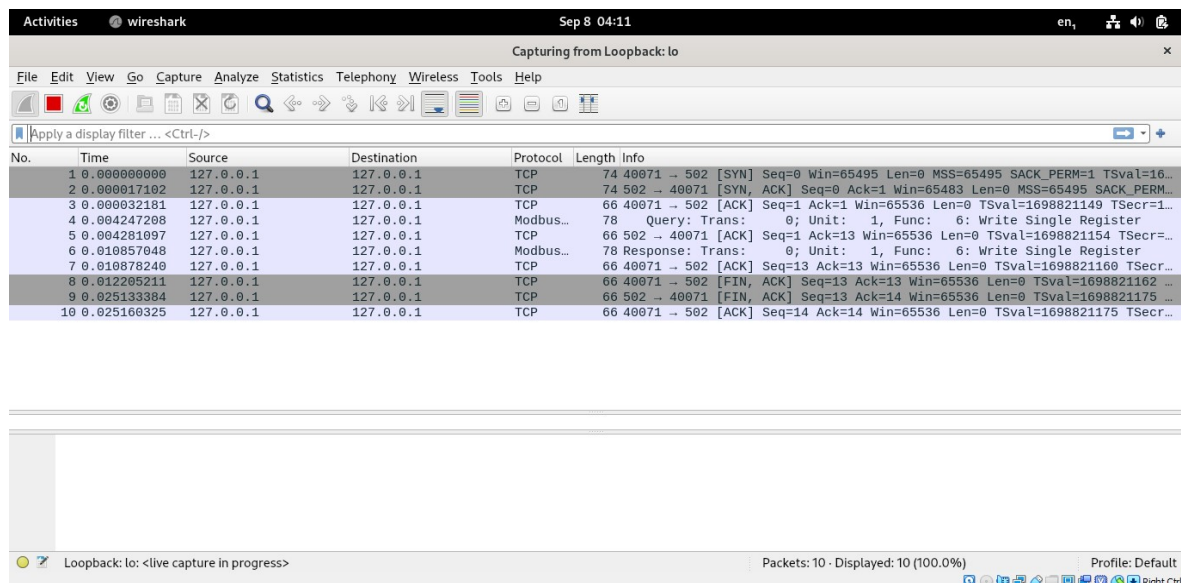
У овом поглављу биће речи о анализи и интерпретацији снимака мрежног саобраћаја. Овај корак има изузетну важност у разумевању како различите компоненте међусобно комуницирају у оквиру индустријских управљачких система. Кроз детаљну анализу мрежног саобраћаја добијен је увид у безбедносне претње које могу утицати на функционисање ових система. Биће проучено како се снимци мрежног саобраћаја могу употребити као снажан алат за анализу и дијагностиковање индустријских управљачких система.

За анализу мрежног саобраћаја коришћен је *Wireshark* алат за снимање и анализу мрежних пакета. Кроз овај експеримент, биће приказана још једна слабост *Modbus/TCP* протокола која представља слабост чистог текста. Постоји безбедна верзија *Modbus/TCP* протокола која користи енкрипцију и аутентификацију, али је у многим околностима још увек присутан старији протокол без заштите. Ова слабост може довести до могућности за неовлашћен приступ и манипулацију подацима у индустријским управљачким системима.

Овај експеримент илуструје како ова слабост може бити искоришћена од стране потенцијалних нападача да приступе и манипулишу подацима у реалном времену. Ово наглашава значај примене безбедносних мера и усавршавање протокола како би се заштитили овакви системи од потенцијалних напада.

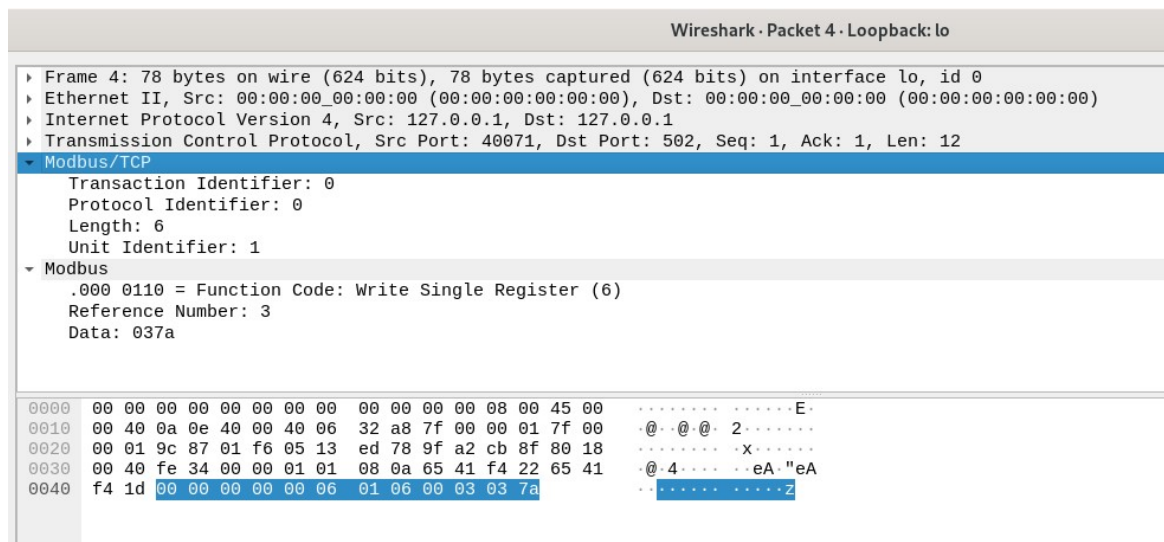
Укључивањем *Wireshark-a* и снимањем мрежног саобраћаја на ИП адреси на којој се налази симулатор (127.0.0.1) ухваћен је одређен број пакета.

Када се деси операција уписа или измене критичних вредности, у апликацији за хватање пакета приказани су следећи резултати:



Слика 13 Ухваћени пакети током операције уписа вредности у *Modbus* уређај

Наравно, *TCP* пакети су за успоставу и обустављање везе. Од значаја су ухваћени *Modbus* пакети. Нападачу који користи овај алат и зна да изврши анализу *Modbus* пакета видљиве су критичне информације које могу да се злоупотребе. Ове информације су најкорисније за фазу извиђања. Извршена је идентификација и откривање поверљивих информација и анализирана је и експлоатисана рањивост.



Слика 14 Приказан изглед једног *Modbus* пакета и читљиви подаци.

Прегледом пакета закључује се да се ради о операцији уписа, конкретно, вредност 3 је уписана у регистар.

Ова слабост се такође може користити и за пресретање и анализу комуникације која се одвија преко *Modbus/TCP* протокола (нпр. нуклеарна енергетика, хемијска индустрија и сл.). Ово отвара врата за могуће шпијунаже и прикупљање осетљивих података који се преносе у реалном времену.

4.4. Nmap експеримент

Са овим експериментом, биће спроведено извиђање на SCADA системе користећи скенирање портова и *nmap* скрипте. Ово спада у активнији приступ који проналази SCADA системе и обаља активно извиђање зарад добијања више информација.

Nmap је један од оних алата који су неопходни сваком хакеру или пентестеру. *Nmap* има много врста и опција за скенирање, међу којима су можда најкориснији и најпопуларнији:

- Т – ово је конекциони скенер. Отвара *TCP* тросмерно руковање са циљним системом, нудећи на тај начин најпоузданије резултате, али најмању сакривеност, пошто систем бележи тросмерно руковање.
- S – прикривено или „*SYN*“ скенирање шаље пакет са постављеном „*SYN*“ заставицом и тиме отвара везу, али не довршава тросмерно руковање. Дакле, није евидентиран, али је прилично поуздан.
- U – претходна два скенирања пружају информације о *TCP* портовима, али не и о *UDP* портовима. Ово скенирање посебно тражи *UDP* портове.
- А – ово скенирање поставља „*ACK*“ заставицу која би нормално указивала на текућу *TCP* конекцију. Може се користити за збуњивање и превазилажење неких заштитних зидова без памћења стања (*eng. Stateless firewalls*).

Типичан облик *nmap* команде је: „*nmap -s<type of scan> <IP address>*“.

Проналажење информација о индустријском управљачком систему помоћу овог алата се може извршити на следећи начин:

```
root@ctp:~# nmap -sT 127.0.0.1 -p 502
Starting Nmap 7.92 ( https://nmap.org ) at 2023-09-08 04:54 MDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00031s latency).

PORT      STATE SERVICE
502/tcp   open  mbap

Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
```

Слика 15 Употреба *nmap* команде за детекцију *Modbus* уређаја

Као што је приказано резултатом претходне команде, *nmap* команда је пронашла порт 502 који је отворен на овом систему. Ово указује на то да је порт доступан и да постоји апликација која је повезана на тај порт.

Стање сервиса такође може бити „*filtered*“ што значи да је порт доступан на овом систему али да зид (*eng.firewall*) блокира приступ том порту. Овде је поново приказана слабост незаштићености посматраног уређаја (симулатора).

Додатно, поред тога што је одличан алат за скенирање портова, *nmap* има могућност употребе разних скрипти. Ово додаје значајну могућност *nmap* -у преко „*Lua*“ скриптног језика. *Nmap* механизам за скриптовање је једна од *nmap*-ових најмоћнијих и, најфлексибилнијих карактеристика. Омогућава корисницима да пишу сопствене скрипте и деле те скрипте са другим корисницима. Ове скрипте се могу користити за:

- Откривање мреже (*eng. Network discovery*).
- Софистицираније и прецизније откривање верзије оперативног система.
- Откривање рањивости.
- Откривање позадинских врата (*eng. Backdoor detection*).
- Искоришћавање рањивости.

Nmap скрипте обично имају фајл екстензију „*.nse*“. Како би смо пронашли доступне скрипте на нашој *ControlThings* платформи користи се команда „*locate *.nse*“ у линукс терминалу са администраторским правима. Постоје многобројне *nmap* скрипте од којих је за овај експеримент важна скрипта „*modbus-discover.nse*“.

```
root@ctp:~# locate *.nse
/home/control/Samples/Protocols/BACnet/Nmap_bacnet-discover-enumerate.nse
/home/control/Samples/Protocols/CODESYS/Nmap_codesys-v2-discover.nse
/home/control/Samples/Protocols/DNP3/Nmap_dnp3-info.nse
/home/control/Samples/Protocols/S7comm/Nmap_S7comm_Enumerate.nse
/usr/share/nmap/scripts/acarsd-info.nse
/usr/share/nmap/scripts/address-info.nse
/usr/share/nmap/scripts/afp-brute.nse
/usr/share/nmap/scripts/afp-ls.nse
/usr/share/nmap/scripts/afp-path-vuln.nse
/usr/share/nmap/scripts/afp-serverinfo.nse
/usr/share/nmap/scripts/afp-showmount.nse
/usr/share/nmap/scripts/ajp-auth.nse
/usr/share/nmap/scripts/ajp-brute.nse
/usr/share/nmap/scripts/ajp-headers.nse
/usr/share/nmap/scripts/ajp-methods.nse
/usr/share/nmap/scripts/ajp-request.nse
/usr/share/nmap/scripts/allseeingeys-info.nse
/usr/share/nmap/scripts/amqp-info.nse
/usr/share/nmap/scripts/asn-query.nse
/usr/share/nmap/scripts/auth-owners.nse
/usr/share/nmap/scripts/auth-spoof.nse
/usr/share/nmap/scripts/backorifice-brute.nse
/usr/share/nmap/scripts/backorifice-info.nse
/usr/share/nmap/scripts/bacnet-info.nse
```

Слика 16 Приказ неколико *nmap* скрипта доступних на *ControlThings* платформи.

Употребом скрипте „*modbus-discover.nse*“ добијамо следеће резултате:

```
root@ctp:~# nmap --script modbus-discover.nse --script-args='modbus-discover.aggressive=true' -p 502 127.0.0.1
Starting Nmap 7.92 ( https://nmap.org ) at 2023-09-08 05:29 MDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000044s latency).

PORT      STATE SERVICE
502/tcp   open  modbus
| modbus-discover:
|   sid 0x1:
|_  error: ILLEGAL FUNCTION
Nmap done: 1 IP address (1 host up) scanned in 102.72 seconds
```

Слика 17 Приказ резултата употребе „*modbus-discover.nse*“ скрипте

Када је скрипта успешно извршена, добавила је резултате о свим *modbus* чворовима на систему. Како је тренутно на платформи покренут само један симулатор, добавила је његове информације.

Симулатор нема назив који ова скрипта може да детектује али, у реалном систему ова скрипта добавља идентификациони назив реалног *PLC*-а што не би требало да се деси. Ово поново представља последицу незаштићености протокола. Ова информација може бити довољна за нападаче да почну планирати њихов напад на ову инфраструктуру.

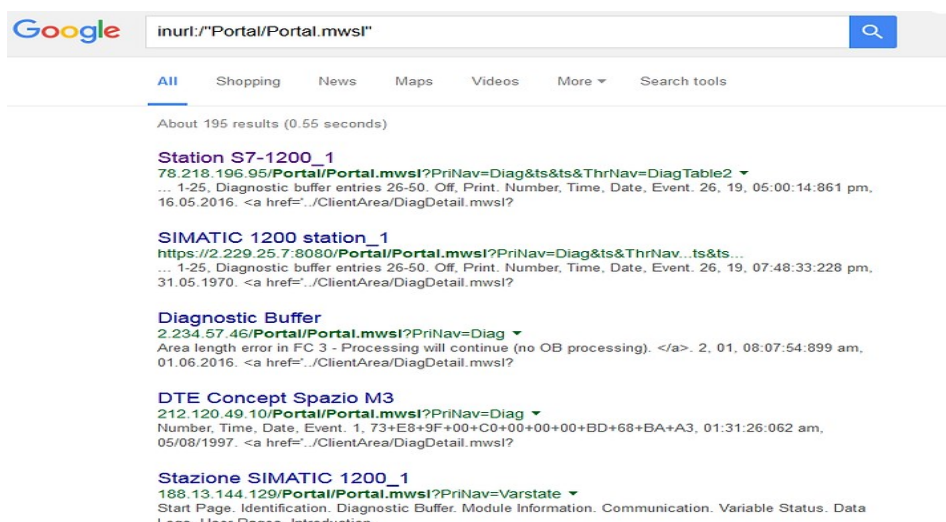
4.5. Гугл *SCADA* хаковање

SCADA системи који користе веб интерфејс омогућавају оператерима да нагледају и контролишу индустријске управљачке системе путем веб прегледача. Ово представља велику предност у управљању и мониторингу приступа систему, како из интерне мреже компаније, тако и удаљено са било ког места са интернет конекцијом. Веб интерфејси обично обезбеђују интуитиван и кориснички прилагођен приказ података и могућности за управљањем истим. Ово их чини савршеном метом за сајбер нападе. Из тог разлога, поред свих предности, важно је осигурати адекватну заштиту оваквих веб интерфејса.

Овај експеримент представља налажење рањивих *SCADA* система помоћу Гугл хаковања. Као што већина претпоставља, Гугл претражује и складишти и индексира информације које пронађе на скоро свакој веб локацији и страници. Међутим, мало је познато да Гугл има језик за извлачење тих информација осим тражења кључних речи.

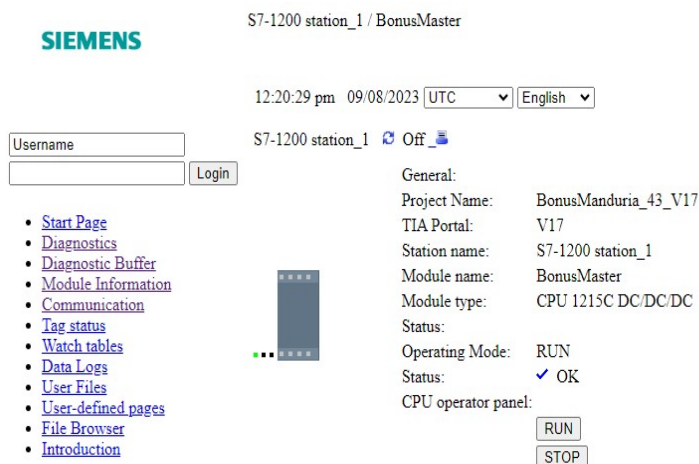
Када је познат овај начин манипулације са „*Google Dorks*“ *SCADA* системи са веб интерфејсима великих компанија који се тичу индустријских управљачких система постају велика мета ових напада (нпр. *Siemens*, *Shneider Electir*, *General Electric*, *Rockwell Automation* и многе друге).

Употребом „*Google Dorks*“ за нпр. *Siemens* компанију, можемо циљно тражити индустријски контролни систем са доступним интерфејсом за манипулацију. „*Google Dorks*“ наредба изгледа: „*inurl:/Portal/Portal.mwsl*“. Резултат извршавања ове наредбе је следећи:



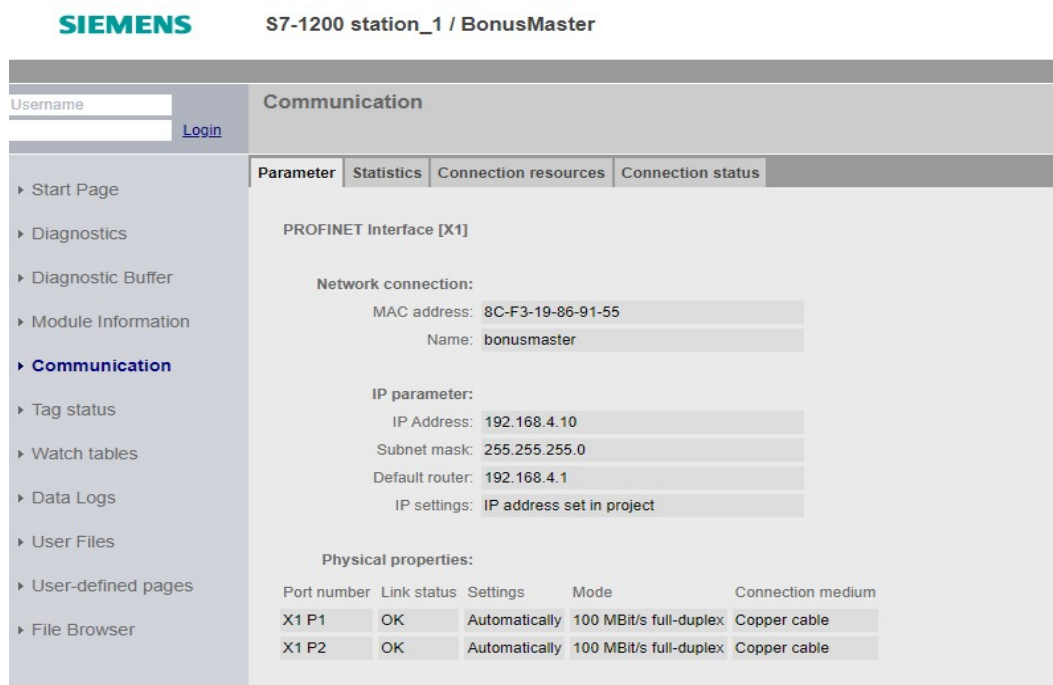
Слика 18 Резултат претраге помоћу „*Google Dorks*“

Кликом на први резултат претраге добијени су следећи резултати:



Слика 19 Информације о контролеру

Доступне су све информације о овом контролеру. Кликом на дугме *Communication* доступна је IP и MAC адреса.



Слика 20 Детаљне информације о комуникацији контролера

Ови једноставни „Google Dorks“ могу променити ове системе из невидљивих у јавно доступне свима на планети. Чак и хакер са основним вештинама може пронаћи ове системе и, ако има злонамерне планове, приступити овим контролним системима и изазвати хаварију.

5. Закључак

Овај рад је истражио важност и комплексност сајбер безбедности у индустријским управљачким системима, који су кључни за безбедно и ефикасно функционисање различитих индустријских сектора, укључујући производњу, енергетику и транспорт. Њихова сигурност је од виталног значаја.

Платформа *ControlThings*, је значајан корак напред у обезбеђивању сајбер безбедности за индустријске управљачке системе. Ова линукс дистрибуција је специјализована за тестирање оваквих система, укључујући и *SCADA*, *DCS*, *IoT*, *IIoT* системе, као и процесну и комуникациону опрему. Платформа је лако прилагодљива и доступна разним корисницима. Ова платформа је од изузетне важности за све организације које се баве индустријским управљачким системима, пружајући им могућност да проактивно идентификују и реше сигурносне рањивости пре него што постану мета потенцијалних нападача.

Напредак у области сајбер безбедности индустријских управљачких система захтева ангажовање стручњака, константно праћење нових претњи и технолошких решења, као и примену најбољих пракси у овој области. Уз адекватне мере заштите, овакви системи могу остати отпорни на потенцијалне сајбер претње и обезбедити поуздан рад у индустријским окружењима.

Кроз експерименте коришћења алата, у платформи *ControlThings*, попут *mbtget* и *metasploit framework*-а показано је како неосигурани *Modbus/TCP* протокол може представљати озбиљан сигурносни ризик за индустријске управљачке системе. Ова слаба тачка омогућава потенцијалним нападачима неовлашћен приступ и манипулацију критичним подацима. Методе заштите од ових претњи представљају употребу сигурнијих и безбеднијих верзија протокола које обухватају енкрипцију и аутентификацију. Приказане су неке од најчешћих метода напада на ове системе.

Сprovedени су различити експерименти са циљем анализе сигурности индустријских управљачких система. Коришћени су алати као што су *mbtget* и *Metasploit framework* како би се идентификовале потенцијалне слабости у *Modbus/TCP* протоколу. Исто тако, анализиран је мрежни саобраћај са циљем откривања могућих рањивости у комуникацији.

Конечно, ова истраживања су показала колико је важно посветити пажњу сајбер безбедности индустријских управљачких система и колико је потребно константно радити на унапређењу њихове сигурности. Само кроз заједничке напоре стручњака и индустријске заједнице може се осигурати сигурна и поуздана будућност индустријских управљачких система.

Литература

- [1] *ControlThings* платформа, доступно на <https://www.controlthings.io/home>
- [2] *Wireshark*, доступно на <https://www.wireshark.org/>
- [3] *Nmap*, доступно на <https://nmap.org/>
- [4] *Modbus*, доступно на <https://en.wikipedia.org/wiki/Modbus>
- [5] *Google Dorks*, доступно на <https://www.sababasecurity.com/hacking-finding-scada-in-the-network-using-google-dorks/>
- [6] *Mbtget* алат, доступно на платформи *ControlThings*
- [7] *Metasploit framework*, доступно на платформи *ControlThings* и на <https://www.hackers-arise.com/post/2016/11/25/scada-hacking-metasploit-scada-modules>
- [8] Информације о *ICS*, доступно на https://en.wikipedia.org/wiki/Industrial_control_system
- [9] Информације о етичком хаковању, доступно на [https://en.wikipedia.org/wiki/White_hat_\(computer_security\)](https://en.wikipedia.org/wiki/White_hat_(computer_security))
- [10] Информације о *SCADA* безбедности и етичком хаковању, доступно на <https://www.hackers-arise.com/scada-hacking>
- [11] *Clint E. Bodungen, Bryan L. Singer, Aaron Shebeeb, Stephen Hilt, Kyle Wilhoit, „Hacking Exposed Industrial Control Systems: ICS and SCADA Security Secrets & Solutions“ 1st Edition*
- [12] *Jon Erickson, „Hacking: The Art of Exploitation“ 2nd Edition*

Подаци о кандидату

Кандидат Немања Малиновић је рођен 2000. године у Новом Саду. Завршио је Средњу Школу ЕТШ „Михајло Пупин“ у Новом Саду, 2019. године. Факултет Техничких Наука у Новом Саду је уписао 2019. године. Испио је све обавезе и положио је све испите предвиђене студијским програмом.



Слика аутора рада