


Accurate Modeling of the Siemens S7 SCADA Protocol for Intrusion Detection and Digital Forensics

Amit Kleinmann and Avishai Wool

School of Electrical Engineering, Tel Aviv University

Agenda

- Brief introduction to ICS and SCADA
- The Siemens S7 SCADA Protocol
- Modeling for intrusion detection
- Results
- Conclusions & future work

Industrial Control Systems (ICS)

- Automatic monitor/control of industrial facilities
- Electricity generation, water treatment, gas pipes, etc…
- Supervisory Control and Data Acquisition (SCADA)
  - Human-Machine Interface (HMI): operator control
  - Programmable Logic Controllers (PLC):
    - connected to sensors, actuators, running a control program

- Not isolated anymore
- Include standard IT components
- More features, more volume

No Security

Protecting ICS Systems

Threats

- Gaining access to the Control Network => Enables Attacks:
  - Violating: confidentiality, availability, integrity
    - E.g., DOS, manipulate net. protocols, Tamper with memory, Buffer Overflow
- Deny committing an attack (Attribution, Lack of forensic capabilities)
  - Violating non repudiation

Constraints

- Difficult/expensive to replace equipment
- Concerns over loss-of-control
  - false-positive => block legitimate traffic

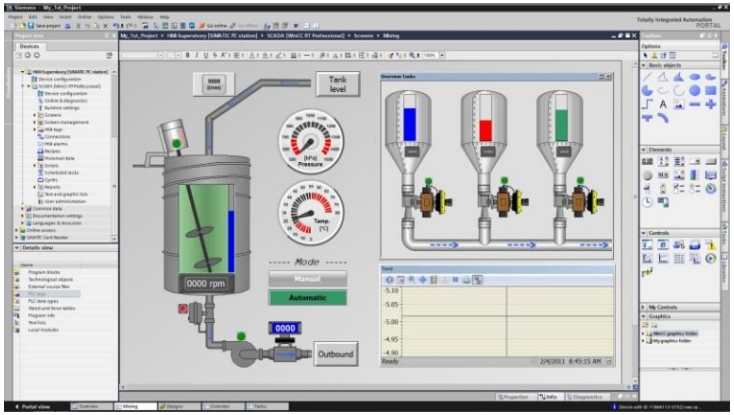
NIDS Network Intrusion Detection System

- **Signature** detection
  - Inherently lags behind attack development
    - E.g., obfuscation & polymorphism utilized by recent network attacks
  - Fails to protect from unknown and novel threats
- **Anomaly** detection

The S7 PLC Platform

Siemens TIA Portal:

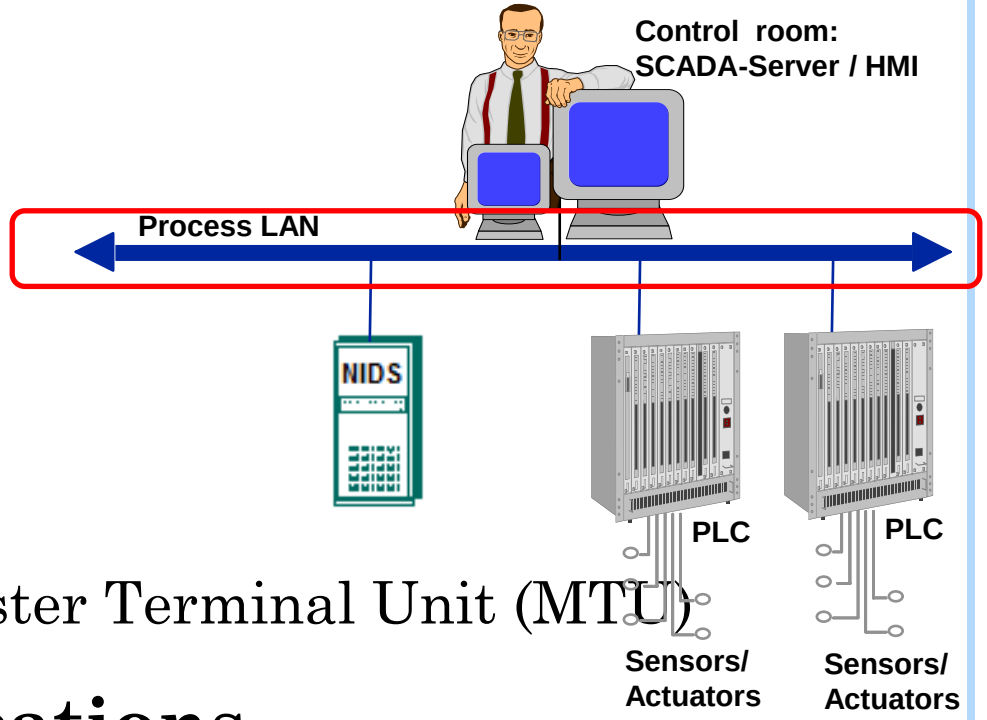
- STEP7 **Engineering WS** +
- WinCC **HMI** SW



Siemens SIMATIC S7 - **PLC**:

- Over 30% of the worldwide PLC market
- Models:
  - Standard (S7-200, S7-300, S7-400)
  - New generation (S7-1200, S7-1500)

SCADA Supervisory Control and Data Acquisition in a Nutshell



- **Data Acquisition**
  - Sensors
    - Located at various points
- **Control**
  - RTU/PLC
  - SCADA Server a.k.a. Master Terminal Unit (MTU)
- **Network Communications**
  - Point to point
  - Query-Response SCADA Protocol
    - E.g., Modbus, Siemens S7, IEC 60870-5-104, DNP3
- **Data Presentation**
  - Human Machine Interface (HMI)
  - Data Historian - centralized DB – archive data

The S7 Protocol 1/2

IP Header	TCP Header	TPKT Header	COTP Header	S7 (Header and PDU)

- TCP – Port 102
- ISO-TPKT: ISO-on-TCP - RFC 1006:
  - Emulates ISO COTP on top of TCP
    - Adds packet boundaries - header (4 bytes): version & length fields
- COTP - ISO 8073; RFC 905
  - TSAP addressing
    - Contains: device type and addresses (direct, or #rack + #slot)

TPKT = Transport Service on top of the TCP
COTP = Connection-Oriented Transport Protocol

The S7 protocol 2/2

- S7 is proprietary protocol

- Communication modes:
  - **P2P**: partner devices exchange unsolicited data
  - **Client-Server**: HMI – sends query
    PLC – sends respond

- Protocol flavors:
  - Standard (0x32)
  - New (0x72)

Introducing the GW Model

Developed at the Net. Security **Lab of TA University**

- By Niv Goldenberg and Avishai Wool
- Tested on recorded Modbus traffic

Observation – SCADA networks have:

- Relatively static topology
- Regular network traffic pattern

Learning phase: builds DFA for PLC-HMI connection

- State represents a valid query or response
- Symbol – PDU fields, e.g., function code, memory address

Enforcement phase

- Irregular query/response occurs => irregular behavior is detected
  - An alert may be triggered

GW Model advantages:

- Goes much deeper into the details of the SCADA protocol spec
- Captures inter-packet relationship

The GW Model: Transition Function

For each:

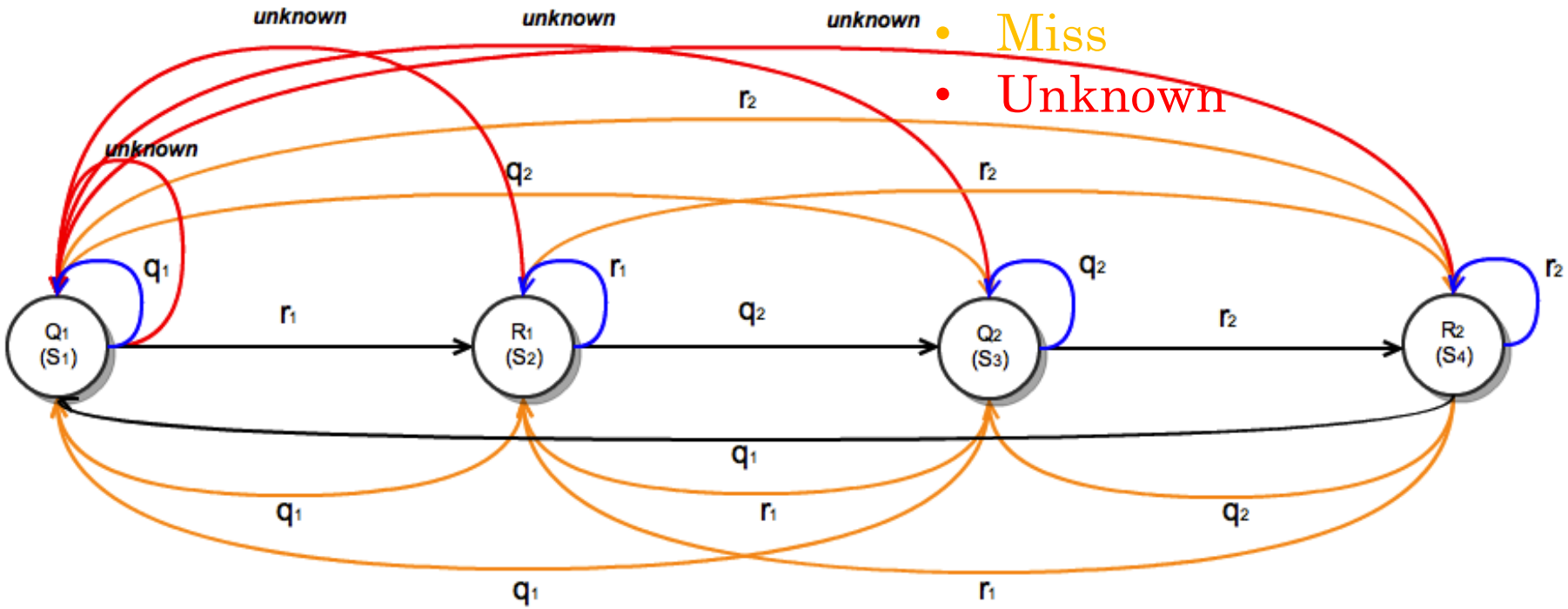
(Origin State, Input symbol)

matches:

(Destination state, operation)

4 Types (& corresponding counters):

- Normal
- Retransmission
- Miss
- Unknown



Unknown transition: Serious anomaly

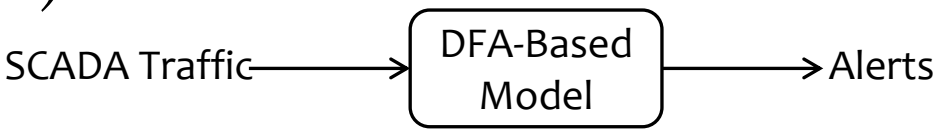
$s_i \neq \{q_1, r_1, q_2, r_2\}$

Raise ALARM(symbol)

The GW Model: Mealy DFA

Deterministic Finite Automata

Generated per channel (HMI IP, PLC IP, PLC Unit Id)



Design Choice: what constitutes a symbol?

1. Finite set of states (Q)

2. Finite set of input symbols
   - Called alphabet(Σ)

3. Transition function ($\delta: Q \times \Sigma \rightarrow Q$)

4. 'Start' state ($q_0 \in Q$)

5. Set of 'accept' (final) states ($F \subseteq Q$)

Is it Possible to model the S7 traffic with a DFA

Observations:

- The S7 traffic is periodic!
  with a short period
- The PDU is divided into three parts:
  - Fixed header
  - Parameters part
  - Data part

The S7-0x32 PDU

Fixed Header for ROSCTR 1 or 3, Func. Code 4 or 5 (Read/Write)

Protocol Id	ROSCTR	Reserved
Request Id		Parameter Length
Data Length		Error Code - only for ROSCTR 3
Function Code	Item Count	

ROSCTR = Remote Operating Service Control

Read/Write Request - Parameter Item

Item head (Variable Specification)

Var Spec Type	Var Spec Length	Var Spec Syntax Id	Transport size
Length		DB Number	
Area	Address		

- **Variable specification**: implies attribute settings to the item variable/s
  - E.g., permitted address range, supported data types and access permissions
- **Transport size** = data type => implies data length
  - Bit, Byte, Int, Real, etc.
- **Length**: the number of referenced variables
- **Data Block (DB)** memory area - stores internal state of the PLC program
  - Each DB contains data-item/s

Return code	Transport size	Data length (per transport size)

Header of Data Item

Applying the GW model to S7

What constitutes a symbol? Selected PDU fields:

- Header fields
  - ROSCTR
  - Parameter Length
  - Data Length
  - Error Code*
  - Function Code
  - Item Count
- All the parameter items*
- All the data header fields of data items*

* If exist/s

- Number of symbol bits ≈ Length of PDU fields

⇒ **Symbol = hash of selected PDU fields**

  - E.g., 64 least significant bits of sha-1

- Configuration:
  - Max pattern length: 75 symbols
  - Validation window: 300 symbols

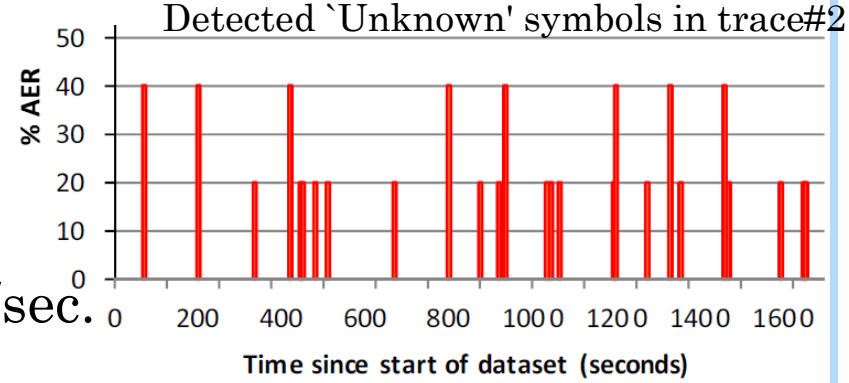
Collected datasets

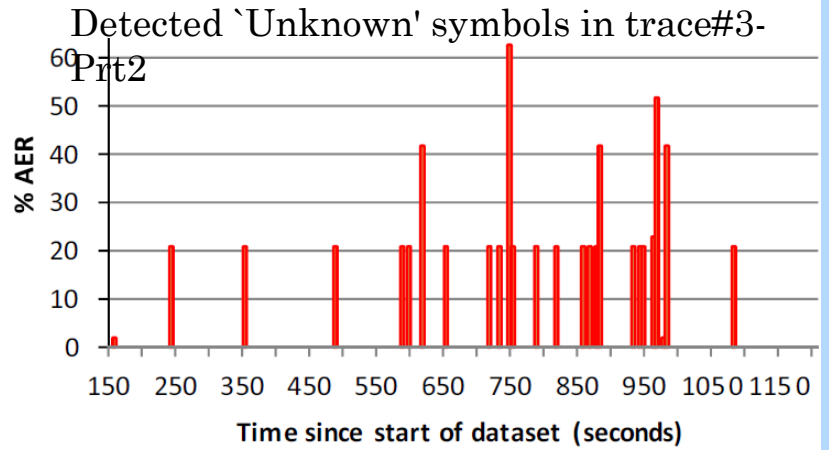
Collected from ICS that controls:

- manufacturing: raw materials are weighed/measured in a mixer
  - Single channel between the HMI and an S7-compliant VIPA PLC
- Waste Water: tank levels, flow rates, and temperatures/pressures
  - Starting and stopping of pumps, opening and closing of valves, etc.
  - Three channels between HMI and 2 Siemens S7-200 PLCs

#	Description	Duration	TCP Packets	S7 Packets
1	Building material manufacturing plant	3242 Sec.	56843	35747
2	Waste water treatment facility	1691 Sec.	38962	21670
3	Waste water treatment facility	1204 Sec.	42465	25379

Results of applying the basic model on S7 traffic





- Definitions:
  - Average Event Rate (AER): sym/sec.
  - S7 quiescent period
    - A time frame during which only `Normal‘ symbols were exhibited
- Results:
  - Patterns are short
  - Pattern lengths ≈ AER
    - Indicating that the system has inherent 1-second periods
  - **Over 98.16% of all packets were identified as `Normal' packets**

Dataset	AER	Pattern
#1 Part 1	8.01	8
#1 Part 2	11.99	12
#1 Part 3	12.09	12
#2 All	10.96	12
#3 Part 1	10.81	12
#3 Part 2	20.13	22

Diagnosis of The Unknown-Symbols Symptom

bservation:

- Changes in the grouping of S7 items within the S7 PDU
  - Reminder: a pattern has N packets each with ~20 data items
  - Some of the items are grouped differently into PDUs

- Reflected in the basic model by

Day 1

Day 2

Splitting an S7 packet into separate items

We avoid these `false alarms‘ by
applying the model at a finer granularity

Create an artificial packet per each of the PDU items
Apply the model to the artificial packets

Configuration:

- Max pattern length: 1500 (artificial) symbols
- Validation window: 6000 (artificial) symbols

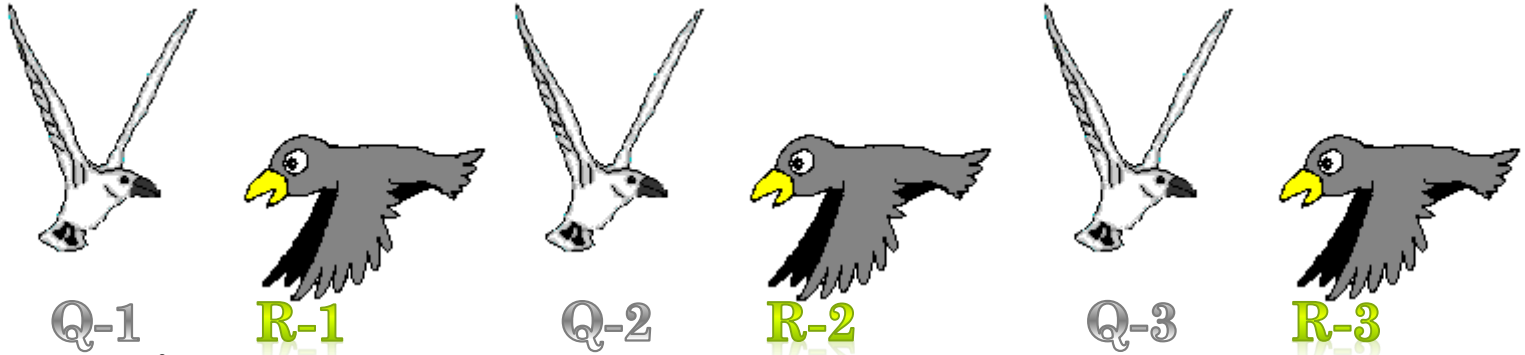
Still a tiny challenge

Dataset	AER	Pattern	# Normal	# Unknown	# Miss	%Normal	%Unknown	%Miss
#1 Part 1	45.89	46	39667	28	1	99.92	0.07	0.01
#1 Part 2	63.78	64	61824	42	0	99.93	0.07	0.00
#1 Part 3	58.18	58	82535	86	0	99.90	0.10	0.00
#2 All	180.24	198	304949	0	565	99.82	0.00	0.18
#3 Part 1	178.19	198	23150	0	16	99.93	0.00	0.07
#3 Part 2	370.20	406	389335	272	950	99.69	0.07	0.24

Simultaneous Inflight Request and Response Packets

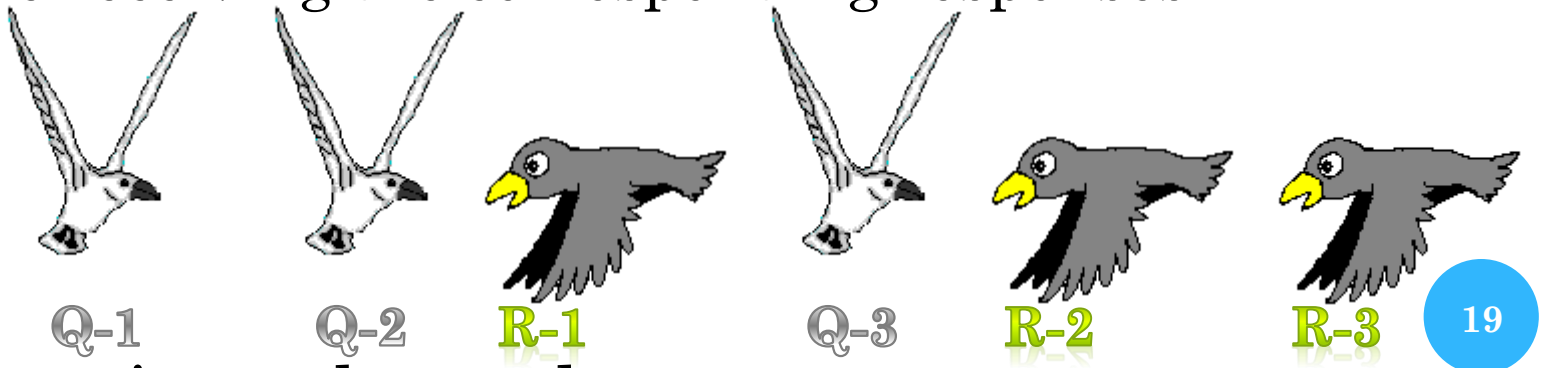
The **basic GW model implicitly assumes** that:

in a clean capture of benign traffic, each query packet is succeeded by its corresponding response packet



Observation:

The HMI sometimes issues several requests, before receiving the corresponding responses

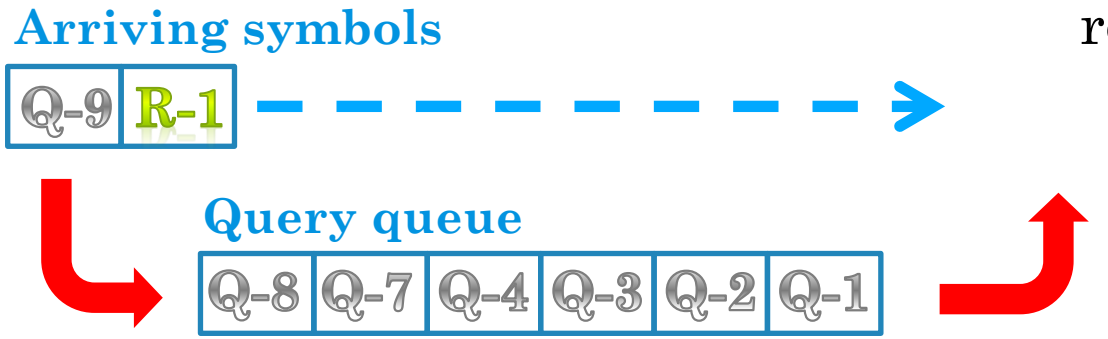


Variations in packet order are reflected in the model by `Miss' events

Synchronizing the Inflight Request and Response Packets

Avoid the false alarms **by synching** each **request with** its associated **response**, before taking the DFA step

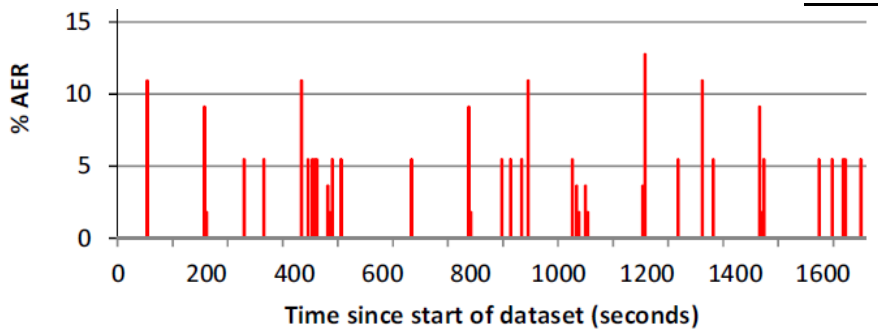
- Applied during both learning phase and enforcement phase
- Implementation:
  - Queue the request packets, and
  - Handle request when its corresponding response is captured
  - Response that results in `Miss' event => take out of queue all requests whose corresponding responses were missed



Results Before and After Applying Splitting and Synchronization

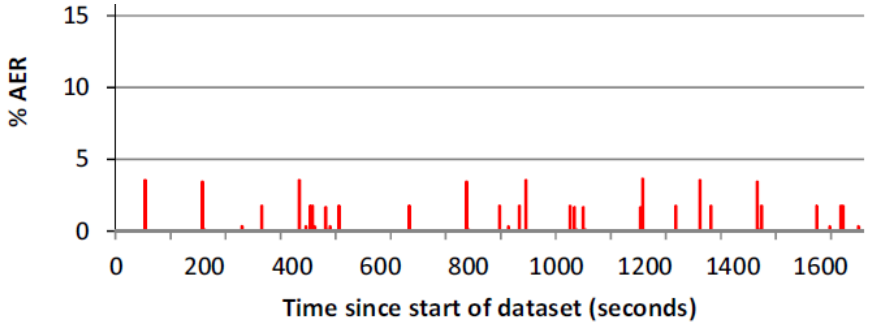
Detected `Miss' symbols on trace #2

Basic model



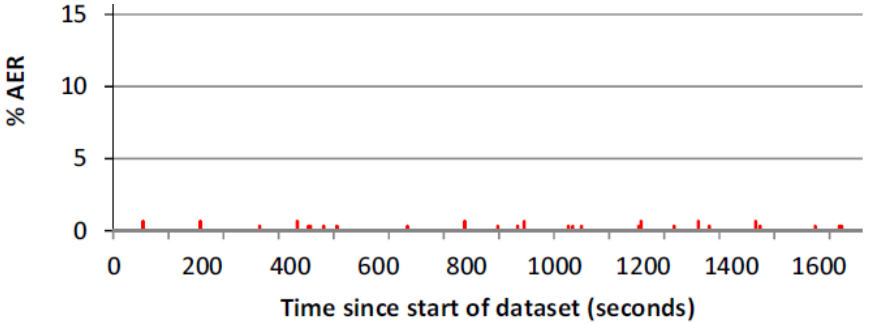
Dataset	% Unkwn	% Miss
#2 All	2.01	0.66
#3 Prt 1	0.78	0.21
#3 Prt 2	3.04	0.47

After splitting



Dataset	% Unkwn	% Miss
#2 All	0.00	0.18
#3 Prt 1	0.00	0.07
#3 Prt 2	0.07	0.24

After splitting and synchronization

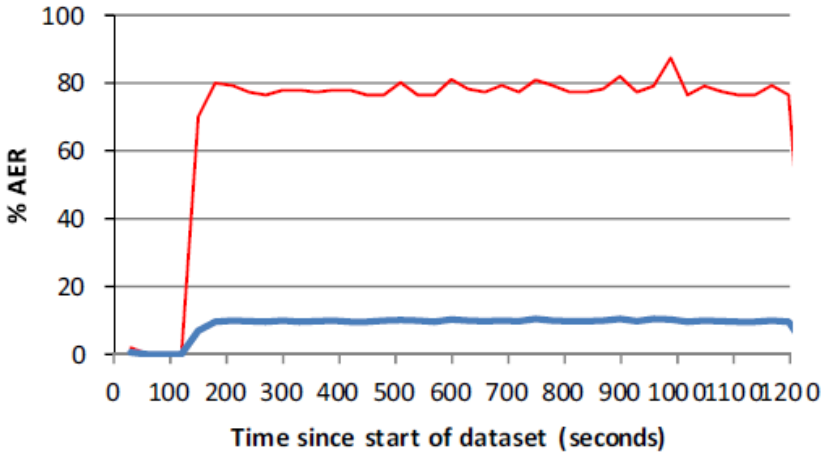


Dataset	% Unkwn	% Miss
#2 All	0.00	0.03
#3 Prt 1	0.00	0.01
#3 Prt 2	0.07	0.11

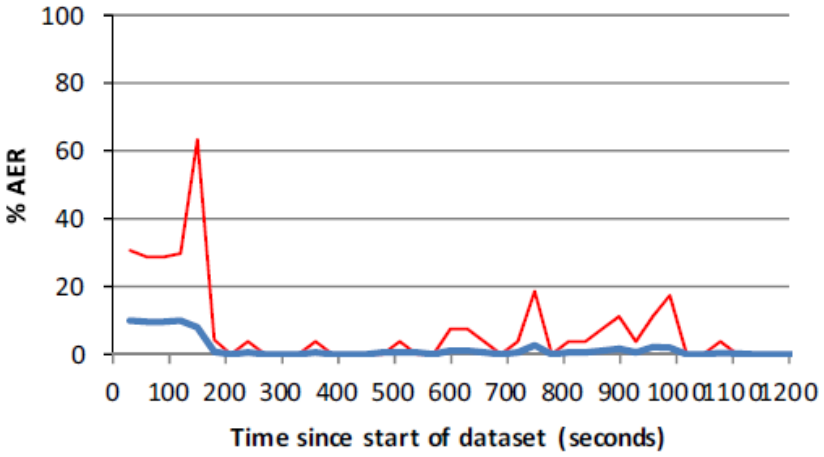
Conclusions and Future Work

Our DFA model has:

- **Very low `false positive'** rate over S7 benign traffic
- Remaining **challenge:**
  - Multiple phases with irregular periods between them
    - Periods (phases) each with its own distinct pattern



(a) Trace#3-Prt1



(b) Trace#3-Prt2


Q&A