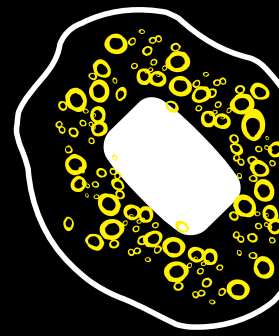
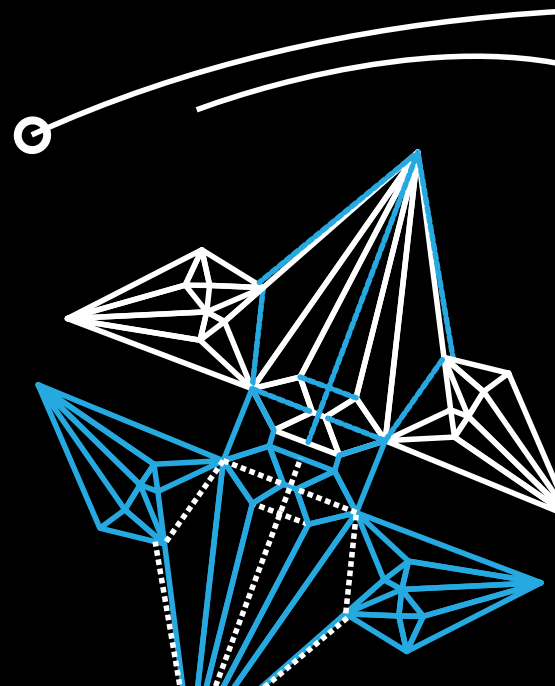
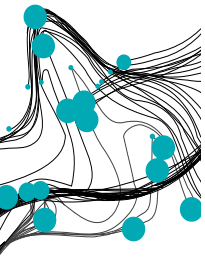
UNIVERSITY OF TWENTE.



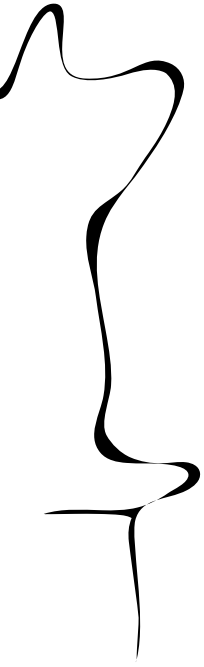
Automated reverse engineering

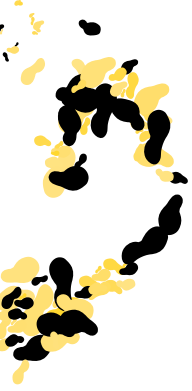
Active state machine learning

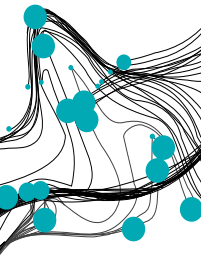




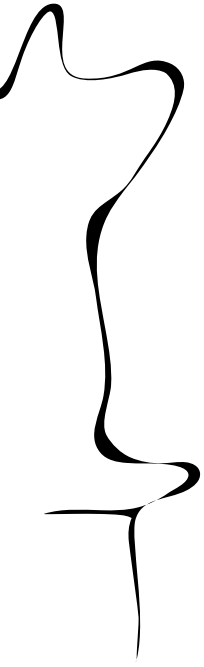
Overview





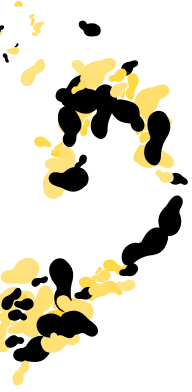


Manual Reverse Engineering

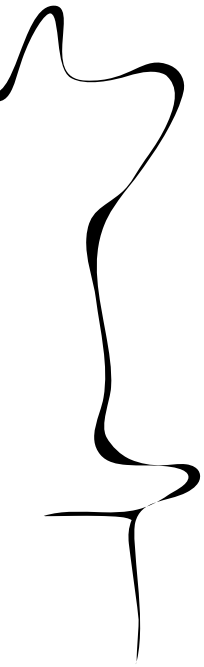


Manual Reverse Engineering and Software Testing has some problems:

- Analyzing complicated binary can be time consuming
- Obfuscated code is hard to analyze



What is active state machine learning



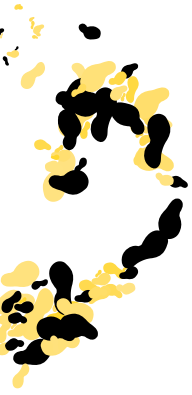
Given a software implementation get a software specification.

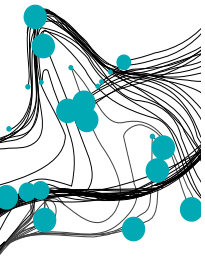
The problem that active state ML tries to solve is:

- Find the smallest deterministic finite-state automaton (DFA) that is consistent with the data

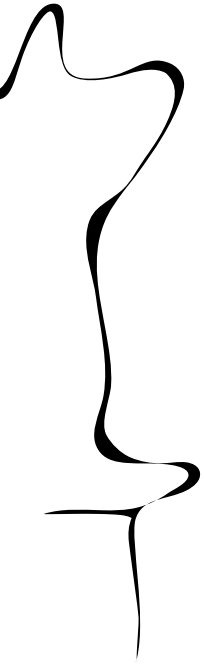
Input:

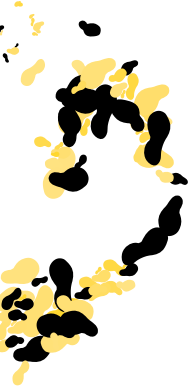
1. Labeled data
2. Positive and negative traces





Overview





What is L*?

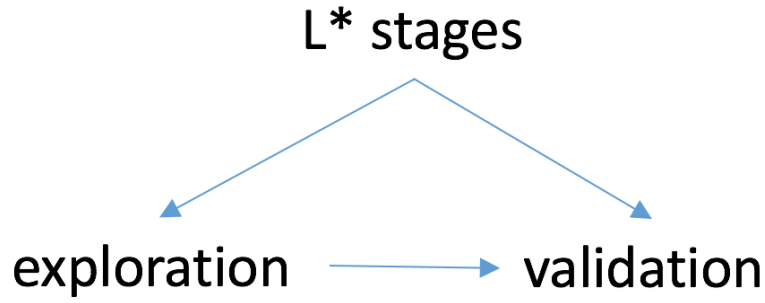
L^* is one of the most popular algorithms used to construct system's behavioral model

We will talk about L^*_M - L^* algorithm adapted for Mealy machines

Requirements:

- Predefined set of actions (input alphabet)
- All actions are executed separately

*L** overview



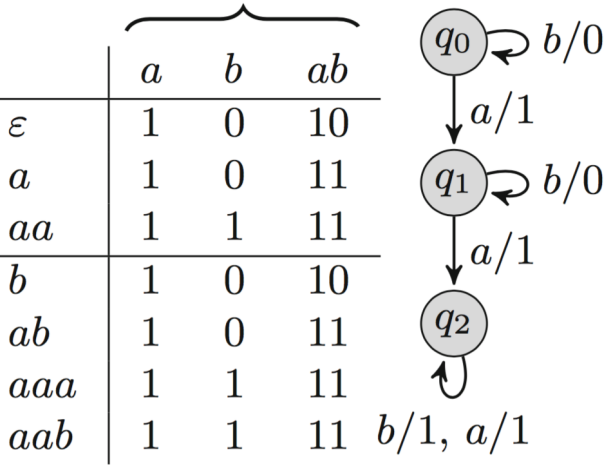
Exploration phase - sequences of symbols (*membership queries*) are executed to form a first hypothesis about SUL's behavior.

Validation phase - check whether hypothesis from exploration phase models system's behavior correctly (*equivalence query*).

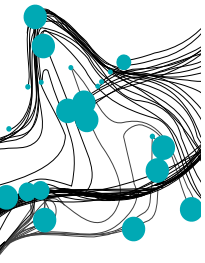
L^* Overview - Output

An output of the L^* is an *Observation Table*

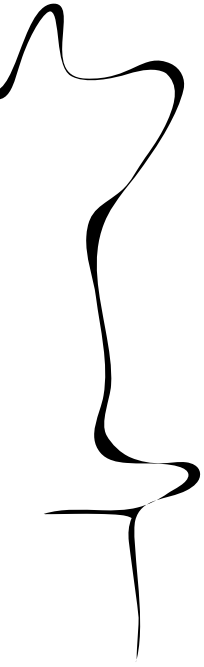
		$\mathcal{D}$		
		a	b	ab
$\mathcal{S}p$	ε	1	0	10
	a	1	0	11
	aa	1	1	11
$\mathcal{L}p$	b	1	0	10
	ab	1	0	11
	aaa	1	1	11
	aab	1	1	11

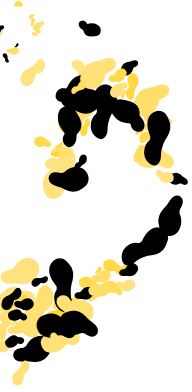


A cell contains the outcome of executing the concatenation of a prefix and a suffix on the system



Overview





Applications

Active state machine learning has many different uses and can be used in a wide range of applications. Especially in the field of **black-box testing** and **protocol inference**.

Both applications are ideal to perform with active state machine learning because these methods are originally:

1. Labor intensive
2. Error-prone
3. Time-consuming

Active machine learning can automate these applications and efficiently solve them.

Protocol state fuzzing of TLS implementations

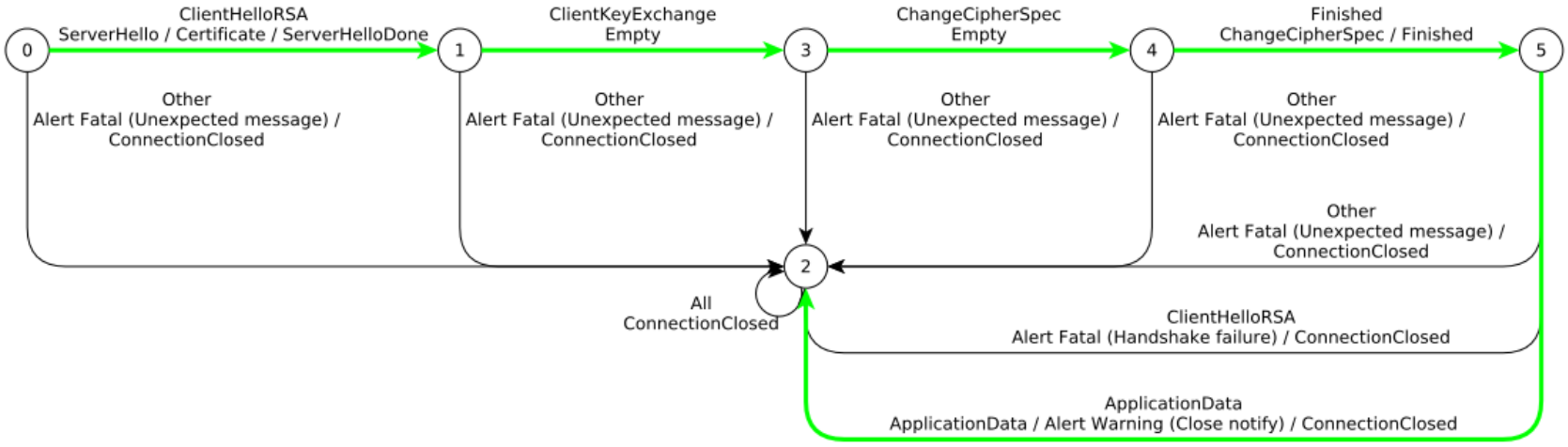
J. de Ruiter and E. Poll used active state machine learning to find vulnerabilities within TLS implementations.

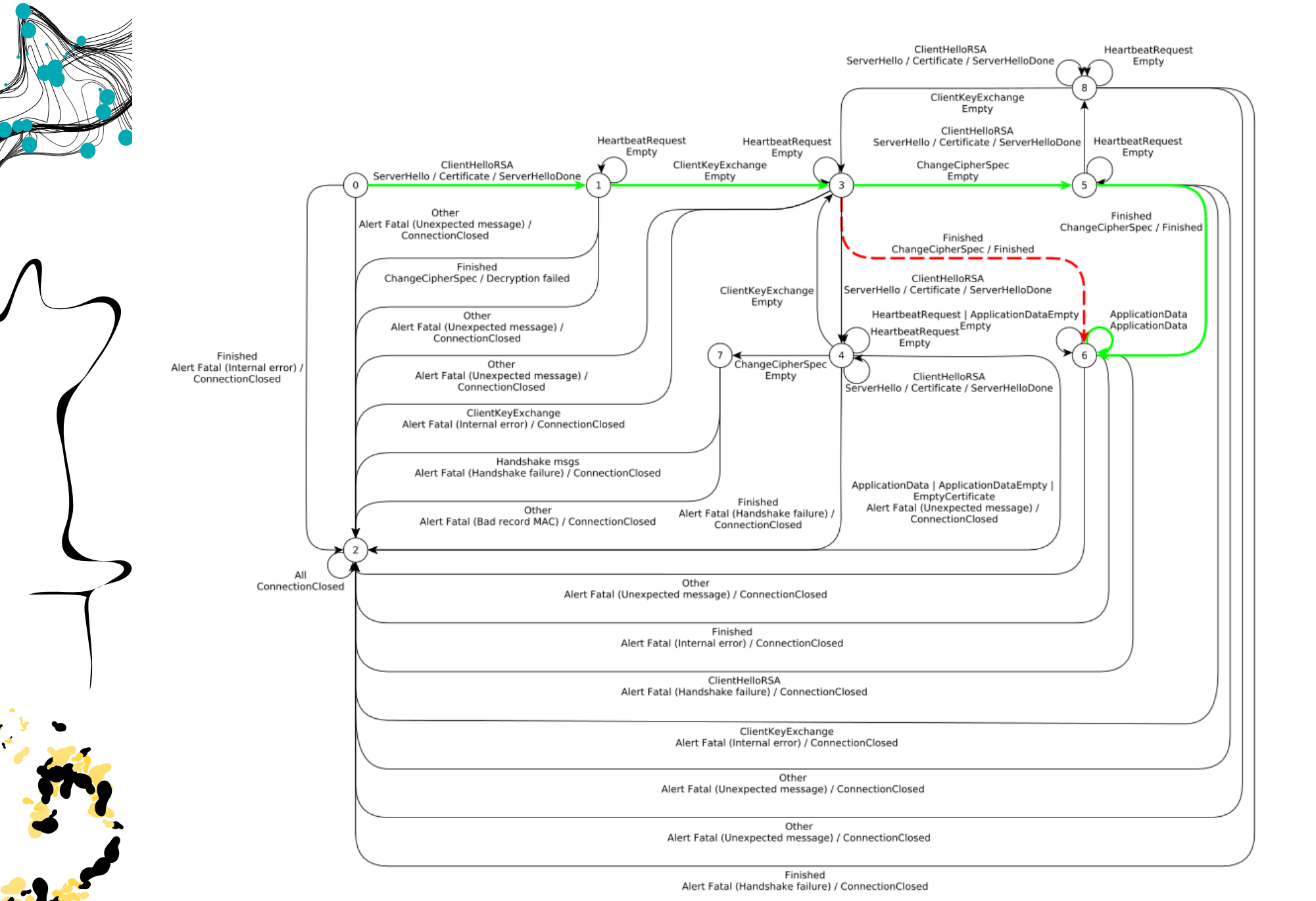
- Used *LearnLib*, implementation of L* Algorithm
- *Input alphabet* is composed of valid TLS messages

- Tested popular implementations such as:
  - mbed TLS (PolarSSL)
  - openSSL
  - JSSE
- Equivalence query by: **Chow's method**: *Testing correctness of state-model*

De Ruiter, Joeri, and Erik Poll. "Protocol state fuzzing of TLS implementations." USENIX Security. 2015

State diagrams




0
1
2
3
4
5
6
7
8
ClientHelloRSA
ServerHello / Certificate / ServerHelloDone
HeartbeatRequest
Empty
ClientKeyExchange
Empty
ChangeCipherSpec
Empty
Other
Alert Fatal (Unexpected message) /
ConnectionClosed
Finished
ChangeCipherSpec / Decryption failed
Finished
ChangeCipherSpec / Finished
HeartbeatRequest | ApplicationDataEmpty
ApplicationData
ApplicationData
Finished
Alert Fatal (Internal error) /
ConnectionClosed
ClientKeyExchange
Alert Fatal (Internal error) / ConnectionClosed
Handshake msgs
Alert Fatal (Handshake failure) / ConnectionClosed
Other
Alert Fatal (Bad record MAC) / ConnectionClosed
Finished
Alert Fatal (Handshake failure) /
ConnectionClosed
ApplicationData | ApplicationDataEmpty |
EmptyCertificate
Alert Fatal (Unexpected message) /
ConnectionClosed
All
ConnectionClosed
Other
Alert Fatal (Unexpected message) / ConnectionClosed
Finished
Alert Fatal (Internal error) / ConnectionClosed
ClientHelloRSA
Alert Fatal (Handshake failure) / ConnectionClosed
ClientKeyExchange
Alert Fatal (Internal error) / ConnectionClosed
Other
Alert Fatal (Unexpected message) / ConnectionClosed
Other
Alert Fatal (Unexpected message) / ConnectionClosed
Finished
Alert Fatal (Handshake failure) / ConnectionClosed

Botnet C&C model analysis

Y. Cho, et all. theorized that knowing more about the internal state model of a C&C systems of a botnet that it would be easier to perform a successful takedown attempt.

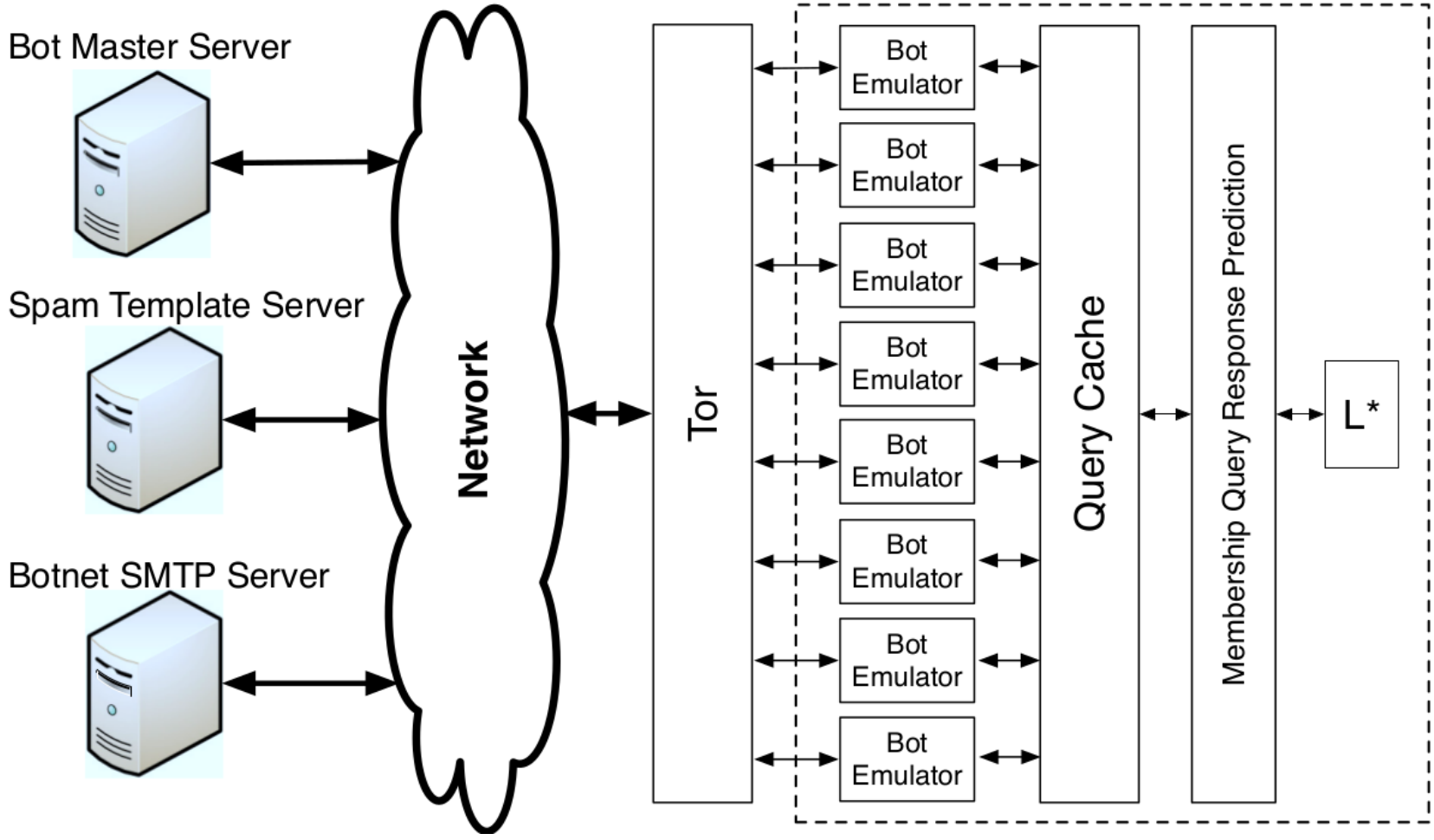
The researched was performed on the MegaD botnet, which is mainly used for spamming.

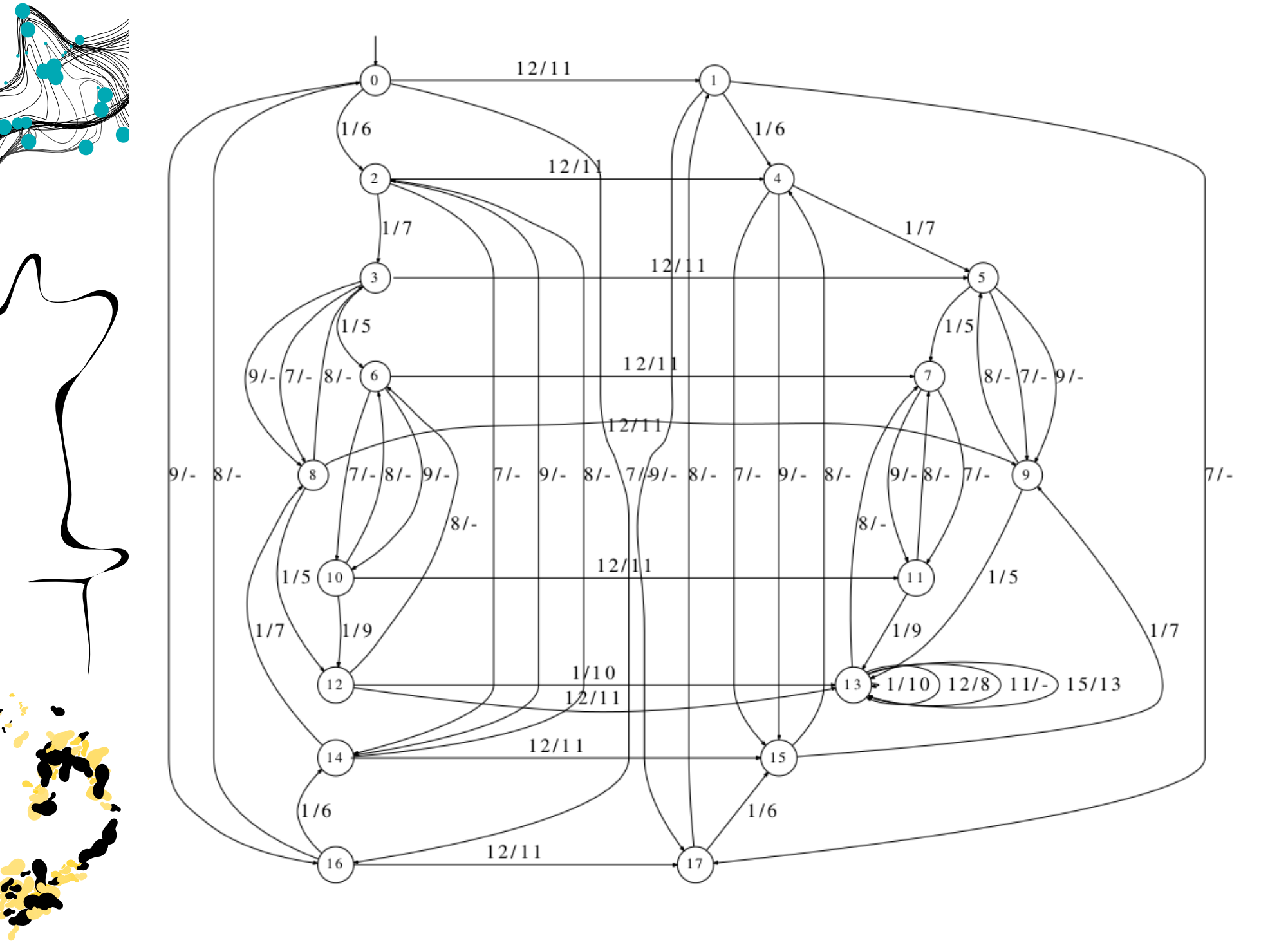
Using the L* Algorithm Y. Cho, et all. researched the following properties:

1. **Identifying Critical Links**
2. **Identifying Design Flaws**
3. **identifying Background-Channels**
4. **Identifying Implementation Differences**

Cho, Chia Yuan, Eui Chul Richard Shin, and Dawn Song. "Inference and analysis of formal models of botnet command and control protocols."

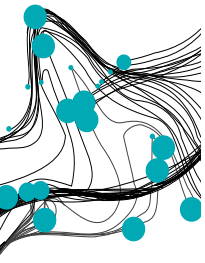
Setup


Bot Master Server
Spam Template Server
Botnet SMTP Server
Network
Tor
Bot Emulator
Bot Emulator
Bot Emulator
Bot Emulator
Bot Emulator
Bot Emulator
Bot Emulator
Query Cache
Membership Query Response Prediction
L*

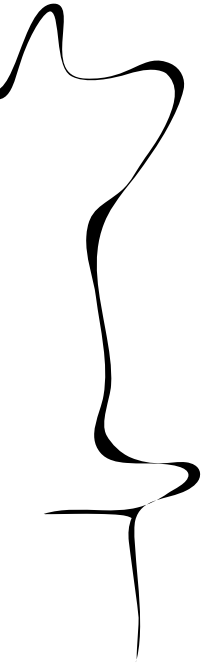

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
12/11
1/6
1/7
1/5
1/9
1/10
9/-
8/-
7/-
12/8
11/-
15/13

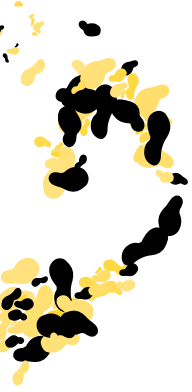
Results

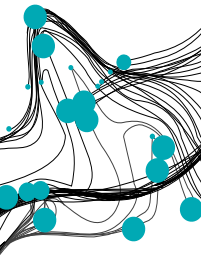
1. **Identifying Critical Links**
   Every bot relies on the same template server. Therefore it is more efficient to only takedown templates servers.
2. **Identifying Design Flaws**
   It is possible to retrieve the new spamming templates prior to other bots, thereby creating valid spamming filters.
3. **identifying Background-Channels**
   There does exists background-channel because states were different when bot visited master node prior to connecting to template server.
4. **Identifying Implementation Differences**
   The state models of the botnet SMTP protcol difference from default `Postfix SMTP 2.5.5` state model.



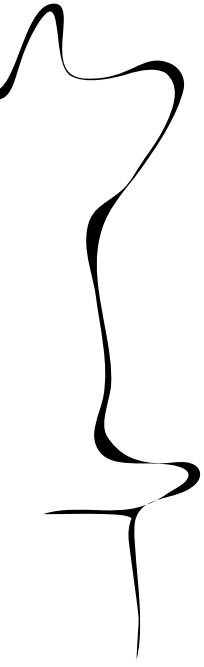
Overview



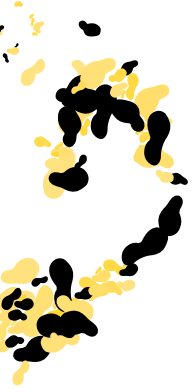


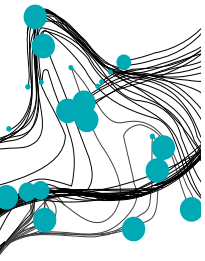


Conclusion

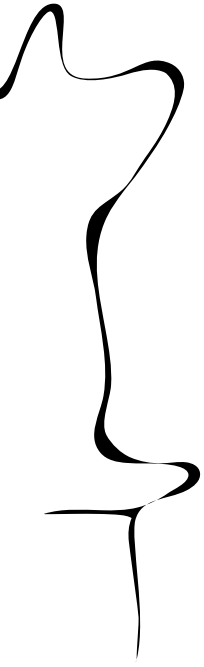


Active state machine learning is a very useful technique for multiple applications. Especially in the field of black-box testing and protocol inference



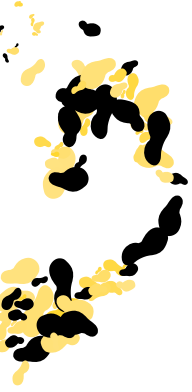


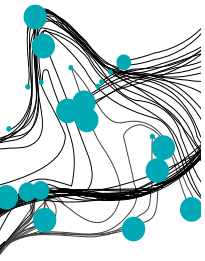
Conclusion

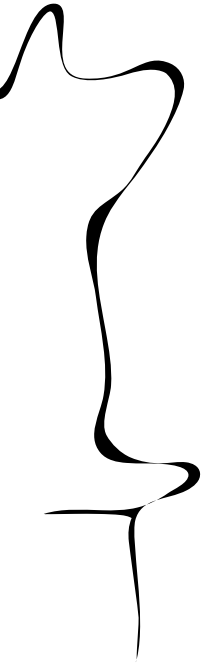


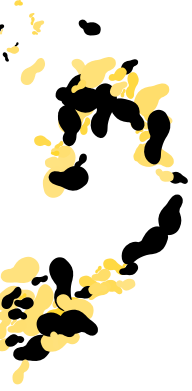
Active state machine learning is a very useful technique for multiple applications. Especially in the field of black-box testing and protocol inference

And although it requires some effort in applying Active state machine learning in a specific situation, it has shown that the uncovered security vulnerabilities are worth the effort.



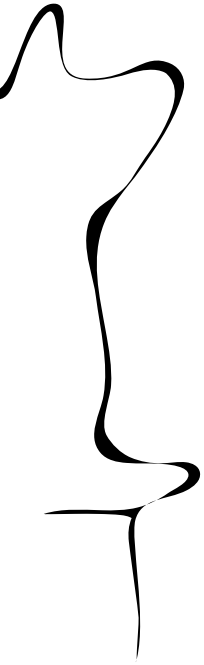






Questions!

Backup slides - Mealy Machine



In the theory of computation, a Mealy machine is a finite-state machine whose output values are determined both by its current state and the current inputs.