Computer Network Defense: Compromise Detection Prototype

Carl Beisel, Jim Jones, Christian Iivari

NATIONAL SECURITY • ENERGY & ENVIRONMENT • HEALTH • CYBERSECURITY

The Problem

- Zero Day Attacks:
  - How and when does a novel, previously unknown attack first get discovered?   Can that attack be detected and stopped before affected systems are compromised and exploited?
- Problem:
  - Signature based detection patterns are based on having discovered, evaluated and defined patterns for the attack. Behavior based detection has high false positives.
- Approach:
  - Non-signature, non-behavior based detection
  - Attack Modeling: reason over observables (indicators, anomalies, second-order effects, etc.)

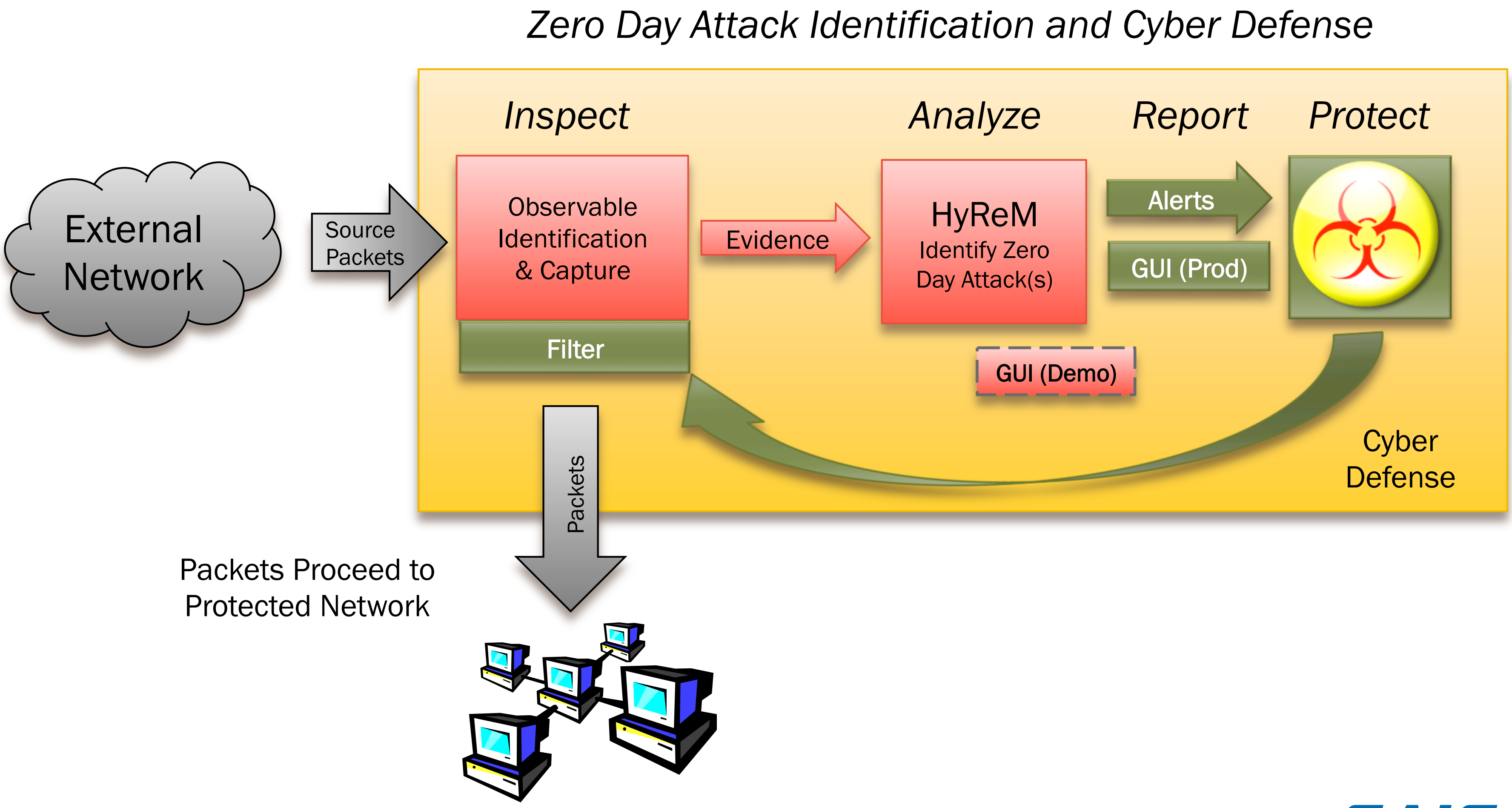
Technical Approach

- **Perform Deep Packet Inspection** of network traffic and capture of packets of interest matching one of our 16+ observable rules
  - Observables represent pieces of evidence relevant to the activities an attacker may perform during an attack as represented by the following transition states:
    - Vulnerability Research
    - Exploit Development and Testing
    - Reconnaissance
    - Exploit Execution
    - Cleaning Activities
    - Back Door Installation
  - Outputs observables for analysis by the reasoning model

- **Assess the likelihood of an attack** using HyReM
  - Use Bayesian Network model to accumulate and assess evidence and calculate the likelihood of a successful attack (i.e., a compromise).

- **Provide graphical output** to a user indicating the likelihood of an attack.
  - Graphical depiction of analysis and calculated 'Likelihood of System Compromise'
  - Can be enhanced to interface with widely used network monitoring and alert tools.

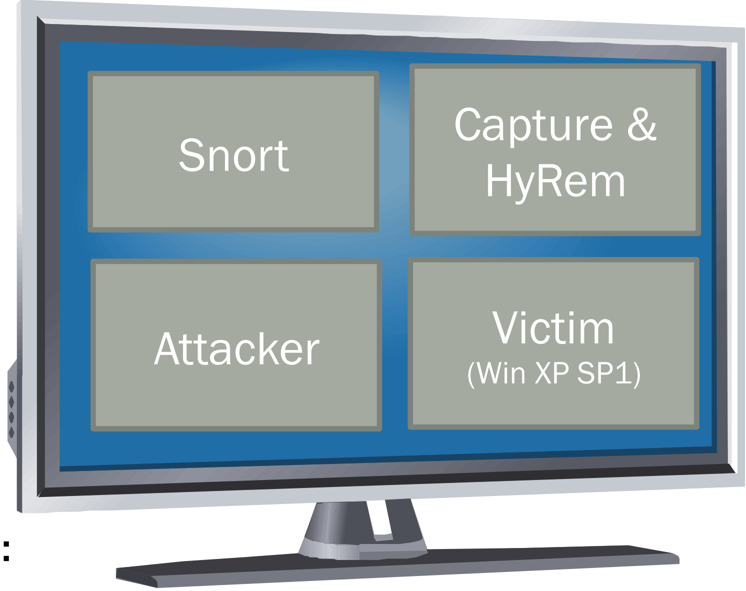
Architecture – Inspect, Analysis, Report, Protect


Zero Day Attack Identification and Cyber Defense
External Network
Source Packets
Inspect
Observable Identification & Capture
Filter
Evidence
Analyze
HyReM
Identify Zero Day Attack(s)
GUI (Demo)
Report
Alerts
GUI (Prod)
Protect
Cyber Defense
Packets
Packets Proceed to Protected Network

Experimental Environment

Virtual Machine Configuration: Four VMs to monitor and analyze network traffic, attacker and victim.

- SNORT Monitor (Backtrack 5.0 on Ubuntu Linux):
  - Snort IDS (Live monitoring)
- Zero Day Attack Monitor (Ubuntu Linux):
  - Capture Tool (LibPCAP)
  - HyReM – Zero Day Attack Analysis/GUI (Demo)
- Attacker (Backtrack 5.0 on Ubuntu Linux):
  - Attacks (Metasploit, Minishare)
- Victim (Win XP SP1 a, has numerous vulnerabilities):
  - Clean snapshot re-instated after each experiment



Experiment Results

One result per test.

						HyReM	Snort			
Test #	Pcap File	Pcap Packet Cnt	Description	Obs File	Total Obs Cnt	Likelihood of Compromise	Snort Alerts	Priority 1	Priority 2	Priority 3
1	1.pcap	2179	Metasploit exploit ms04-011	1_obs.pcap	89	0.9740	13	2	8	3
2	2.pcap	2588	Metasploit exploit ms03-026	2_obs.pcap	105	0.9905	12	2	8	2
3	3.pcap	2420	Metasploit exploit ms08-067	3_obs.pcap	102	0.9742	12	2	8	2
4	4.pcap	1669	Minishare - noisy	4_obs.pcap	90	0.9753	13	2	8	3
5	5.pcap	1642	Minishare - moderate	5_obs.pcap	103	0.9752	10	2	6	2
6	87.pcap	311+5339	Minishare - quiet	8_obs.pcap	6	**0.7803**	0	0	0	0
7	6.pcap	325	Clean 1	6_obs.pcap	2	0.0141	0	0	0	0
8	7.pcap	5339	Clean 2	7_obs.pcap	0	0.0100	5	0	5	0
9	14.pcap	3848	Chronological merge of files 1 and 4	14_obs.pcap	181	0.9970	23	4	16	3
10	26.pcap	2913	Chronological merge of files 2 and 6	26_obs.pcap	108	0.9915	12	2	8	2
11	46.pcap	1994	Chronological merge of files 4 and 6	46_obs.pcap	92	0.9792	13	2	8	3
12	57.pcap	6981	Chronological merge of files 5 and 7	57_obs.pcap	128	0.9752	15	2	11	2
13	347.pcap	9428	Chronological merge of files 3, 4, and 7	347_obs.pcap	219	0.9970	31	4	21	6

Noisy Attack

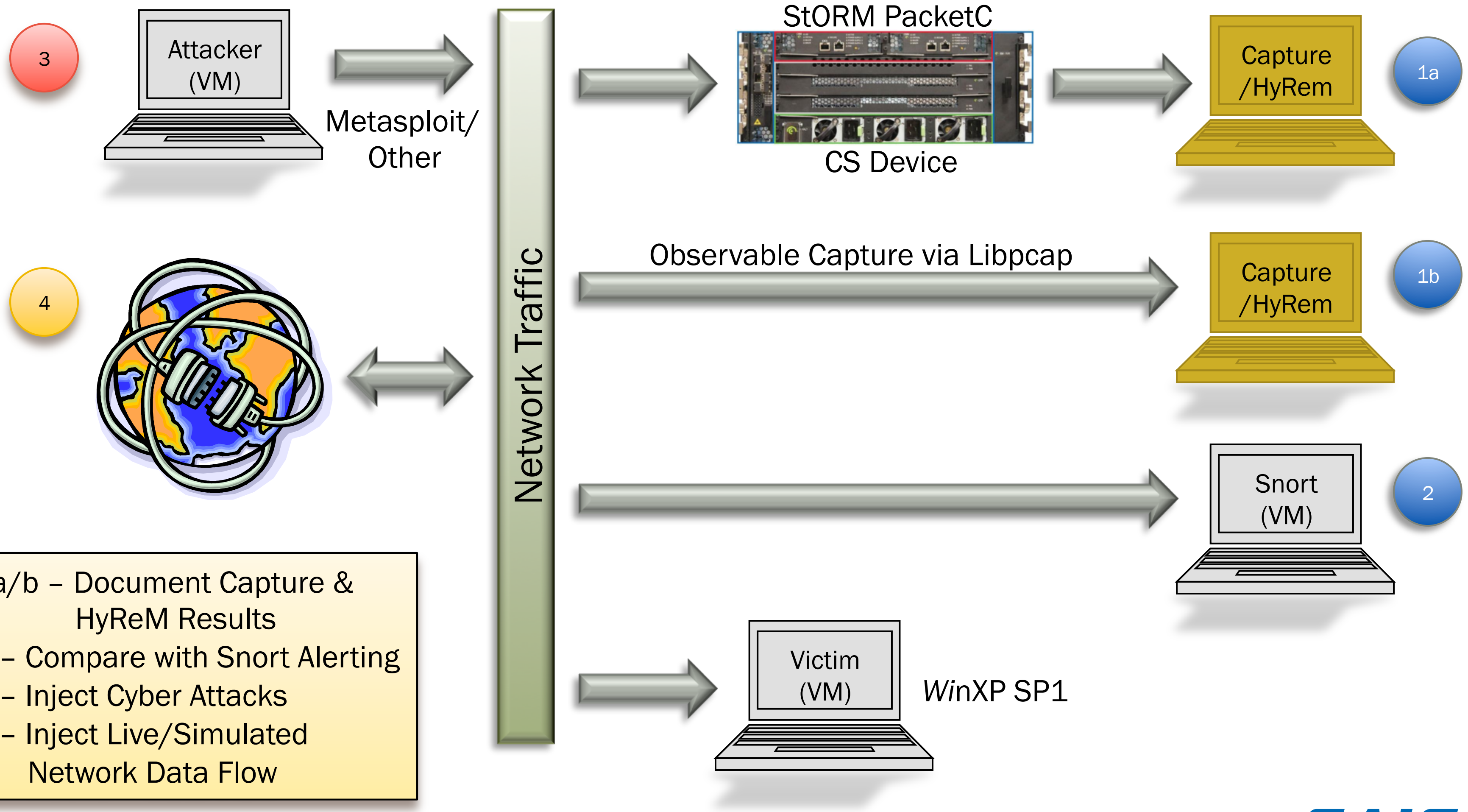
Quiet Attack

Clean

False Positive

Our approach found an attack that the standard toolset missed

Test Environment – Optimal Configuration


3

Attacker (VM)

Metasploit/ Other

4

Network Traffic

StORM PacketC CS Device

Capture /HyRem

1a

Observable Capture via Libpcap

Capture /HyRem

1b

Snort (VM)

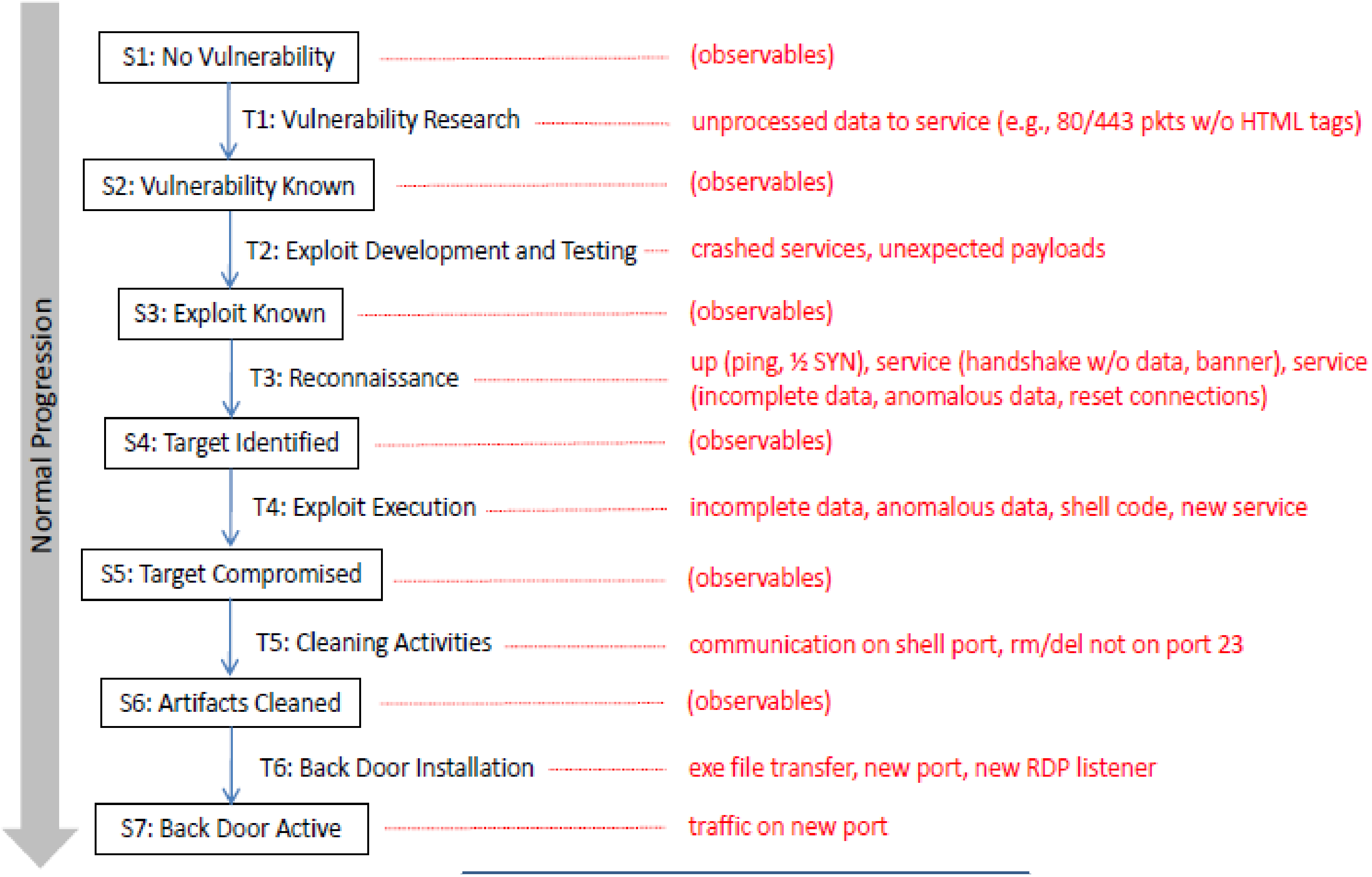
2

Victim (VM)

*Win*XP SP1

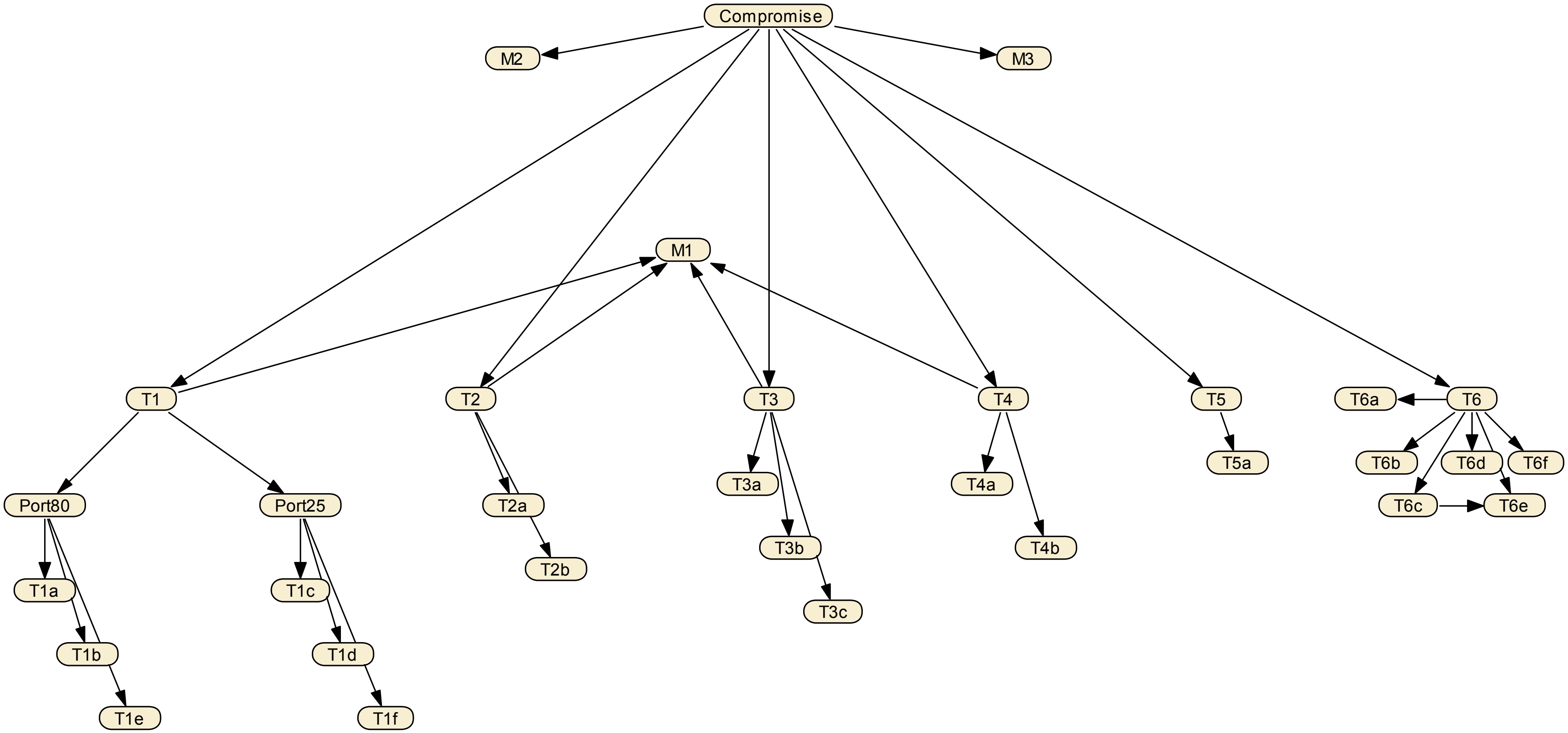
1a/b – Document Capture & HyReM Results
2 – Compare with Snort Alerting
3 – Inject Cyber Attacks
4 – Inject Live/Simulated Network Data Flow

Compromise Model and Observables



S1: No Vulnerability	(observables)
T1: Vulnerability Research	unprocessed data to service (e.g., 80/443 pkts w/o HTML tags)
S2: Vulnerability Known	(observables)
T2: Exploit Development and Testing	crashed services, unexpected payloads
S3: Exploit Known	(observables)
T3: Reconnaissance	up (ping, ½ SYN), service (handshake w/o data, banner), service (incomplete data, anomalous data, reset connections)
S4: Target Identified	(observables)
T4: Exploit Execution	incomplete data, anomalous data, shell code, new service
S5: Target Compromised	(observables)
T5: Cleaning Activities	communication on shell port, rm/del not on port 23
S6: Artifacts Cleaned	(observables)
T6: Back Door Installation	exe file transfer, new port, new RDP listener
S7: Back Door Active	traffic on new port

Observable Modeling


Compromise
M2
M3
M1
T1
T2
T3
T4
T5
T6a
T6
Port80
Port25
T2a
T3a
T4a
T5a
T6b
T6d
T6f
T6c
T6e
T1a
T1c
T2b
T3b
T4b
T1b
T1d
T3c
T1e
T1f