Dreamlab Technologies

The SAPPAN-project (Sharing And Automation for Privacy Preserving Attack Neutralization) and utilization of MITRE for attack emulation

2021

> whoami

- Mischa Obrecht, 34
- IT-Security since 10 years, analyst, engineer, project / security manager, tester
- Pentester and project manager for Dreamlab
- Involved in SAPPAN as project coordinator for Dreamlab

To get in touch please use:

- mischa.obrecht@dreamlab.net

Outline

High level overview on SAPPAN

Utilization of MITRE for attack emulation in the context of SAPPAN

SAPPAN – General Information

Sharing and **A**utomation for **P**rivacy-**P**reserving **A**ttack **N**eutralization

H2020 Call SU-ICT-01-2018 (IA) - Dynamic countering of cyber-attacks:

- **scope:** Cyber-attacks management – advanced response and recovery.
- **timeline**: May 2019 until April 2022.

Abstract:

- **Platform for sharing** and automation of privacy preserving response and recovery using advanced data analysis and machine learning.
- **Decrease the effort required by a security analyst** to find optimal responses to and ways to recover from an attack.
- Within a single organization and across organisations through **privacy-preserving data processing** and sharing.

SAPPAN – Meet the Consortium

Coordinator:

Fraunhofer FIT

Industrial Partners:

Hewlett Packard Enterprise

cesnet

DREAMLAB TECHNOLOGIES

F-Secure®

Academia:

RWTH AACHEN UNIVERSITY

MUNI

Universität Stuttgart

High level overview

Motivation: Intrusion Detection Systems

Example scenario:

- Networks are monitored only within individual organizations.
- Suspicious patterns can trigger alerts.
- Alerts can be resolved by response and recovery actions (“playbooks”).
- New threats may cause new patterns.

Common IDS Challenges:

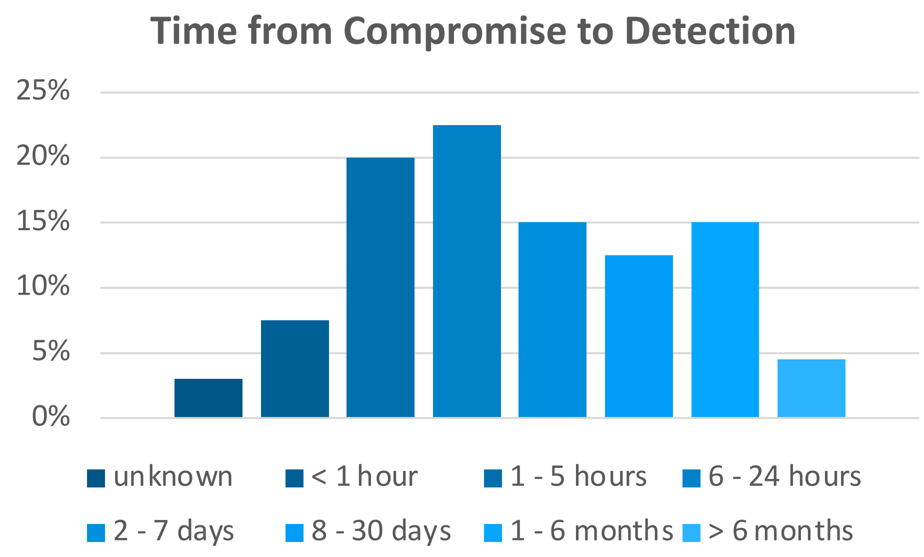
- Limited availability and processed data (e.g., SMEs have less IDS capabilities).
- Difficult to identify attacks with new patterns.
- Too many false positive alerts, security analysts get overwhelmed.
- Data sharing among organizations might lead to privacy/confidentiality leakage.

Motivation:
Long delay in detection of intrusions in the real world

Detection time takes more than 5 hours for two thirds of the cases.

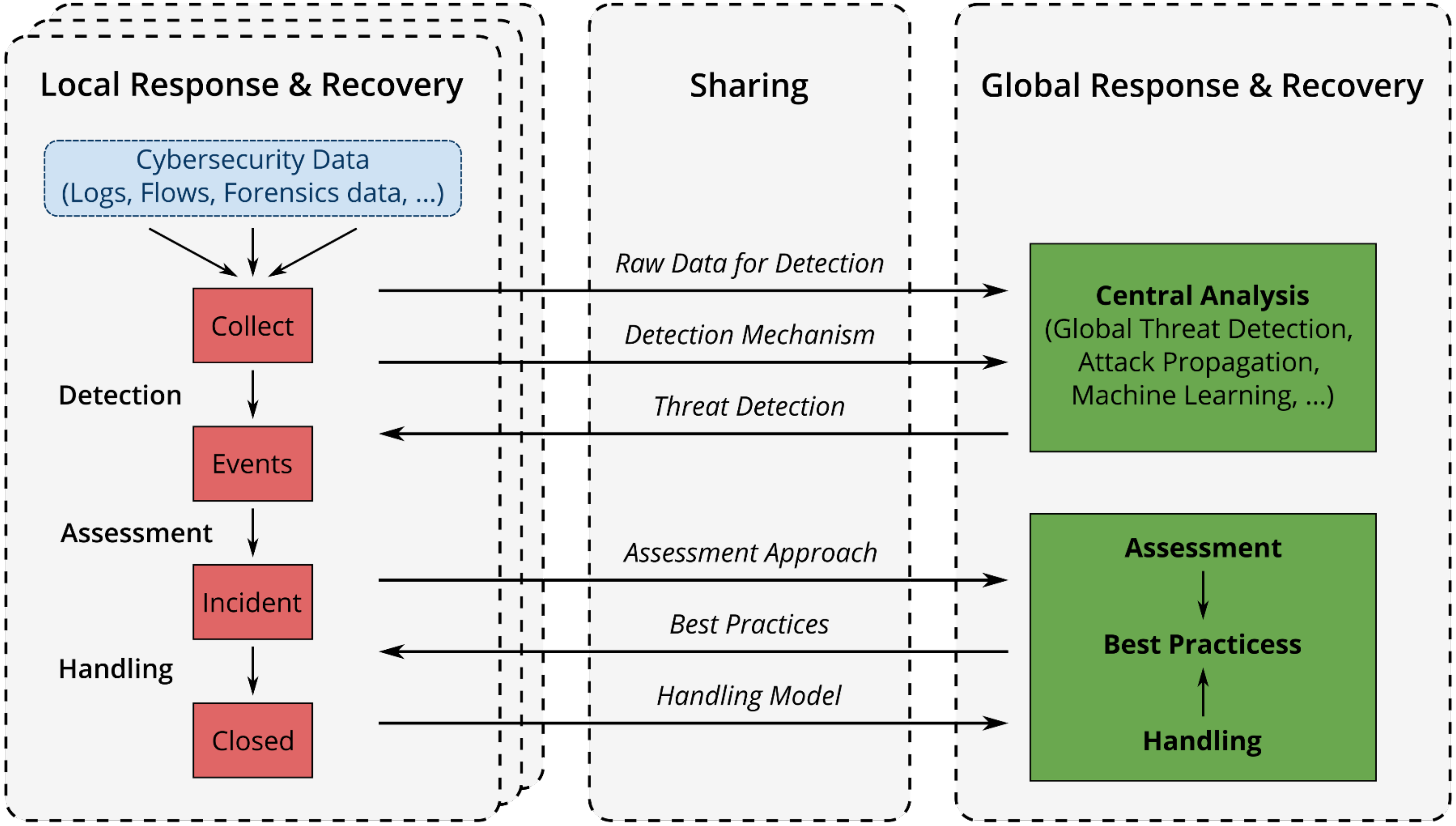
For 20% of cases detection takes more than one month.

Sharing of detection models and actionable response and recovery information between companies could drastically reduce detection and response time.



The Show must go on – A SANS Survey by Matt Bromiley – Published 2017 by SANS Institute

SAPPAN Concept



Local Response & Recovery

Cybersecurity Data (Logs, Flows, Forensics data, ...)

Collect

Detection

Events

Assessment

Incident

Handling

Closed

Sharing

Raw Data for Detection

Detection Mechanism

Threat Detection

Assessment Approach

Best Practices

Handling Model

Global Response & Recovery

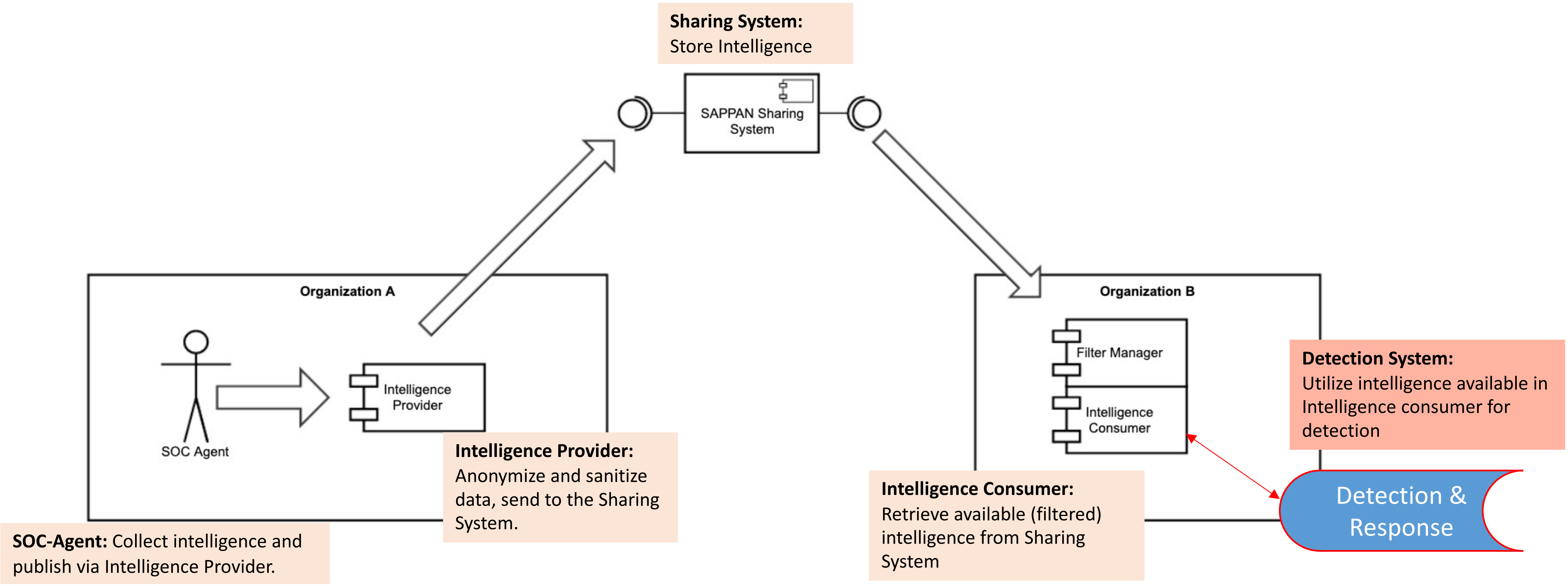
Central Analysis (Global Threat Detection, Attack Propagation, Machine Learning, ...)

Assessment

Best Practicess

Handling

SAPPAN Architecture



Sharing System: Store Intelligence

Intelligence Provider: Anonymize and sanitize data, send to the Sharing System.

SOC-Agent: Collect intelligence and publish via Intelligence Provider.

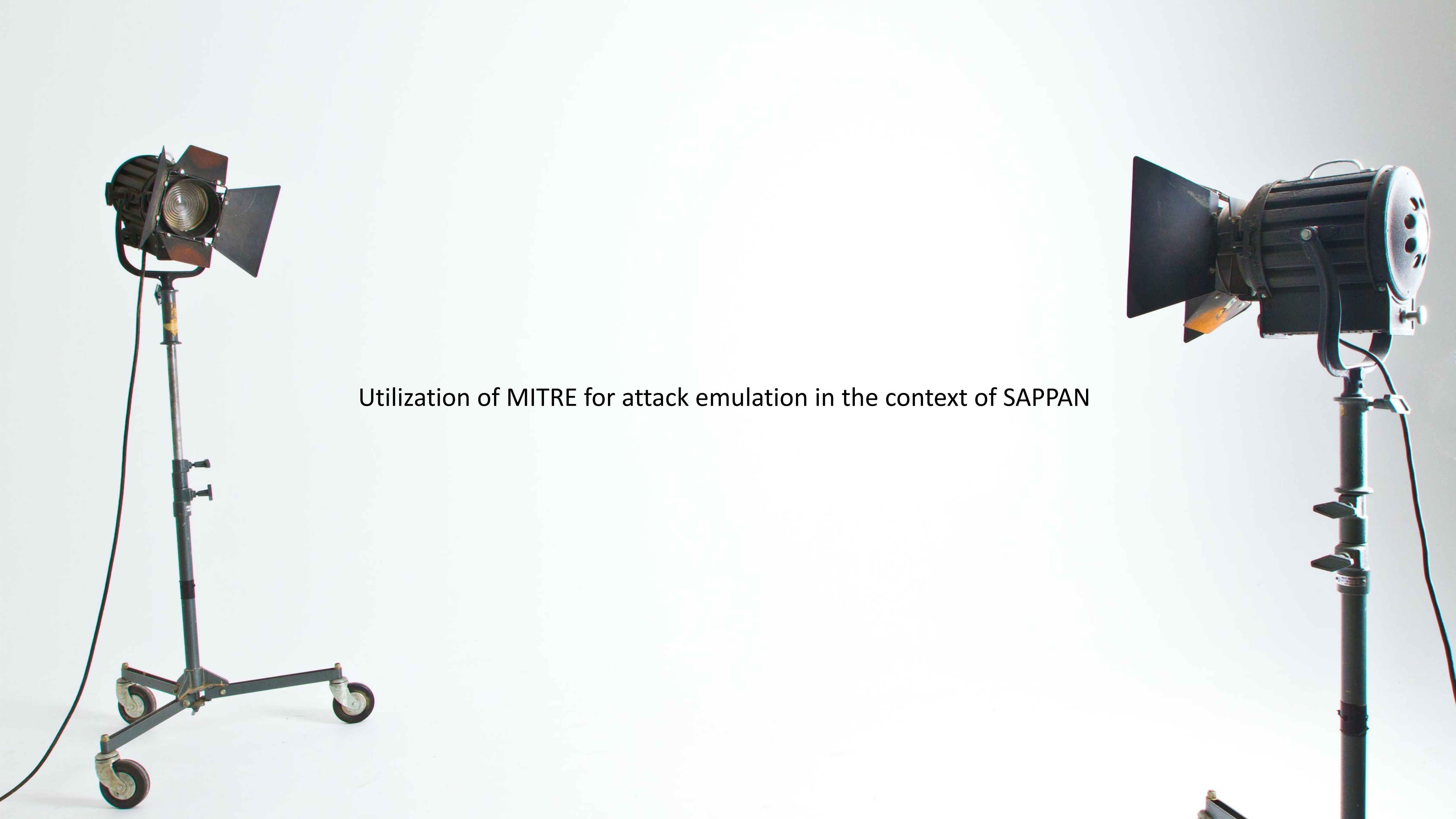
Intelligence Consumer: Retrieve available (filtered) intelligence from Sharing System

Detection System: Utilize intelligence available in Intelligence consumer for detection

SAPPAN – Current Progress

M27/M36

- …
- Framework for machine readable playbooks containing response and recovery information
- Research on local detection methods
    - DGA-Detection
    - Classification of phishing URLs
    - Host- and application profiling based on network and endpoint-data
    - Anomaly detection based on network and endpoint-data
- Research on automation of playbooks for remediation of identified incidents
- Research on anonymization for sharing of information
- Research on federated machine learning
- …



Utilization of MITRE for attack emulation in the context of SAPPAN

Anomaly detection based on network and endpoint-data

All SAPPAN stories start with local detection of security incident

Anomaly detection is a good, general approach to detect nefarious activities

New local detectors that utilize

- network (netflow, connection logs, and full-packet capture) data and
- endpoint data (e.g. process launches)

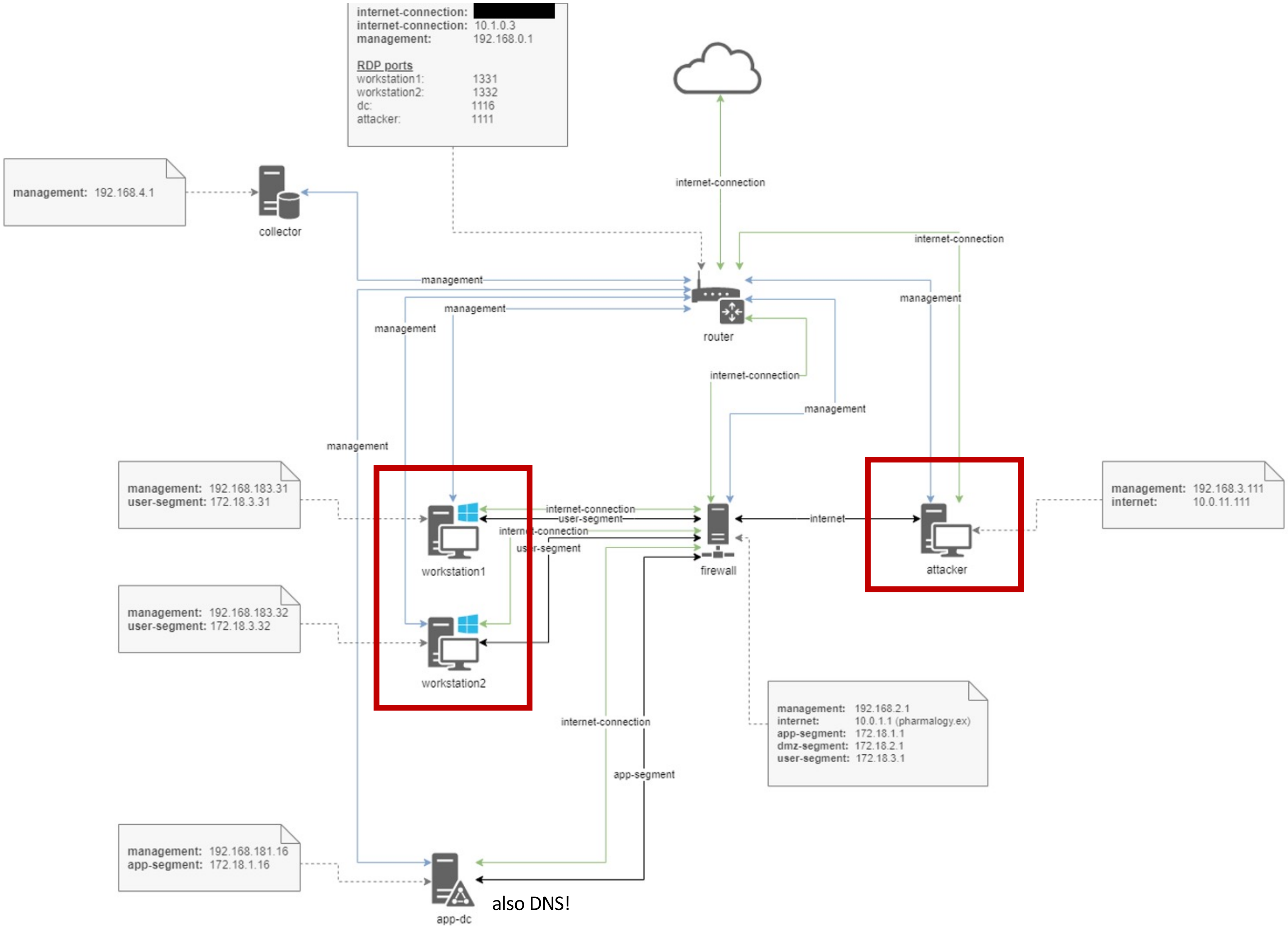
are developped.

Reliable, labeled test-data turned out to be somewhat hard to come by ➔ Make our own

Based on what?

- Simple exploitation of SMB (Eternal Blue) and Drupal (Drupalgeddon) vulnerabilities
- “Internal” Scenario based on: https://attackevals.mitre-engenuity.org/enterprise/APT29/

Network topology for experiment



internet-connection: [redacted]
internet-connection: 10.1.0.3
management: 192.168.0.1

RDP ports

workstation1:	1331
workstation2:	1332
dc:	1116
attacker:	1111

management: 192.168.4.1

management: 192.168.183.31
user-segment: 172.18.3.31

management: 192.168.183.32
user-segment: 172.18.3.32

management: 192.168.181.16
app-segment: 172.18.1.16

management: 192.168.3.111
internet: 10.0.11.111

management: 192.168.2.1
internet: 10.0.1.1 (pharmalogy.ex)
app-segment: 172.18.1.1
dmz-segment: 172.18.2.1
user-segment: 172.18.3.1

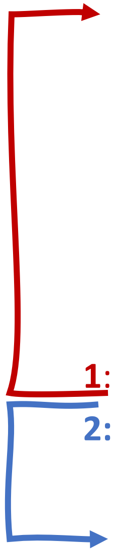
also DNS!

Adoption of APT-29 Emulation Plan for SAPPAN experiment

- We utilized Scenario 1
- PoshC2 instead of Pupy RAT
- Utilization of ”living off the land” binaries to avoid detection
  - Rundll32.exe
  - RuntimeBroker.exe

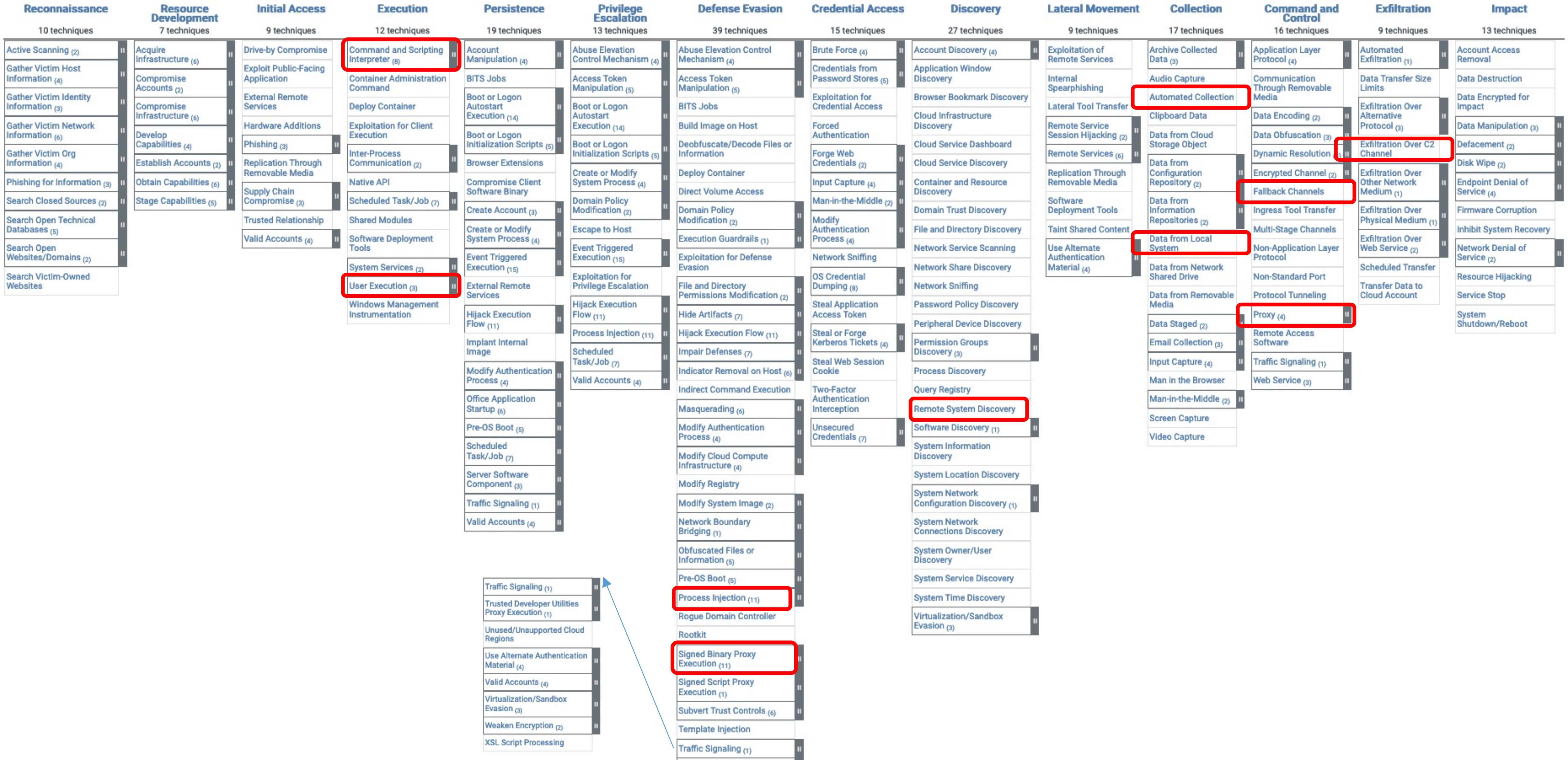
Red team experiment in SAPPAN

- Initial breach through user execution of malicious executable on **workstation 1** (T1204.002)
  - Posh_V4_dropper_x86_migrate.exe, configured to migrate into RuntimeBroker.exe through process injection (T1055.001)
  - Execution of Posh-dropper through rundll32.exe (signed binary proxying method, T1218.011)
- Utilization of communications-rotation for C2-beaconing using list of predefined URLs to avoid detection (T1008, T1090.002)
- Collection of interesting files (smash and grab) (T1005, T1119)
- Exfiltration of collected data through Posh C2-channel (T1041)
- Enumeration of additional machines by querying AD (T1018)
- RCE on **Workstation 2** through Powershell (“Invoke-Command”) to download and execute PoshC2-dropper (T1059.001)
- Killing all implants and end of experiment



Utilized MITRE-Att&ck Tactics

- T1204.002 - User Execution: Malicious File
- T1055.001 - Process Injection: DLL-Injection
- T1218.011 - Signed Binary Proxy Execution: Rundll32
- T1008: Fallback Channels
- T1090.002 - Connection Proxy: External Proxy
- T1119: Automated Collection
- T1005: Data from Local System
- T1041: Exfiltration Over Command and Control Channel
- T1018: Remote System Discovery
- T1059.001: Command and Scripting Interpreter: Powershell



Reconnaissance
10 techniques
- Active Scanning (2)
- Gather Victim Host Information (4)
- Gather Victim Identity Information (3)
- Gather Victim Network Information (6)
- Gather Victim Org Information (4)
- Phishing for Information (3)
- Search Closed Sources (2)
- Search Open Technical Databases (5)
- Search Open Websites/Domains (2)
- Search Victim-Owned Websites

Resource Development
7 techniques
- Acquire Infrastructure (6)
- Compromise Accounts (2)
- Compromise Infrastructure (6)
- Develop Capabilities (4)
- Establish Accounts (2)
- Obtain Capabilities (6)
- Stage Capabilities (5)

Initial Access
9 techniques
- Drive-by Compromise
- Exploit Public-Facing Application
- External Remote Services
- Hardware Additions
- Phishing (3)
- Replication Through Removable Media
- Supply Chain Compromise (3)
- Trusted Relationship
- Valid Accounts (4)

Execution
12 techniques
- Command and Scripting Interpreter (8)
- Container Administration Command
- Deploy Container
- Exploitation for Client Execution
- Inter-Process Communication (2)
- Native API
- Scheduled Task/Job (7)
- Shared Modules
- Software Deployment Tools
- System Services (2)
- User Execution (3)
- Windows Management Instrumentation

Persistence
19 techniques
- Account Manipulation (4)
- BITS Jobs
- Boot or Logon Autostart Execution (14)
- Boot or Logon Initialization Scripts (5)
- Browser Extensions
- Compromise Client Software Binary
- Create Account (3)
- Create or Modify System Process (4)
- Event Triggered Execution (15)
- External Remote Services
- Hijack Execution Flow (11)
- Implant Internal Image
- Modify Authentication Process (4)
- Office Application Startup (6)
- Pre-OS Boot (5)
- Scheduled Task/Job (7)
- Server Software Component (3)
- Traffic Signaling (1)
- Valid Accounts (4)

Privilege Escalation
13 techniques
- Abuse Elevation Control Mechanism (4)
- Access Token Manipulation (5)
- Boot or Logon Autostart Execution (14)
- Boot or Logon Initialization Scripts (5)
- Create or Modify System Process (4)
- Domain Policy Modification (2)
- Escape to Host
- Event Triggered Execution (15)
- Exploitation for Privilege Escalation
- Hijack Execution Flow (11)
- Process Injection (11)
- Scheduled Task/Job (7)
- Valid Accounts (4)

Defense Evasion
39 techniques
- Abuse Elevation Control Mechanism (4)
- Access Token Manipulation (5)
- BITS Jobs
- Build Image on Host
- Deobfuscate/Decode Files or Information
- Deploy Container
- Direct Volume Access
- Domain Policy Modification (2)
- Execution Guardrails (1)
- Exploitation for Defense Evasion
- File and Directory Permissions Modification (2)
- Hide Artifacts (7)
- Hijack Execution Flow (11)
- Impair Defenses (7)
- Indicator Removal on Host (6)
- Indirect Command Execution
- Masquerading (6)
- Modify Authentication Process (4)
- Modify Cloud Compute Infrastructure (4)
- Modify Registry
- Modify System Image (2)
- Network Boundary Bridging (1)
- Obfuscated Files or Information (5)
- Pre-OS Boot (5)
- Process Injection (11)
- Rogue Domain Controller
- Rootkit
- Signed Binary Proxy Execution (11)
- Signed Script Proxy Execution (1)
- Subvert Trust Controls (6)
- Template Injection
- Traffic Signaling (1)

- Traffic Signaling (1)
- Trusted Developer Utilities Proxy Execution (1)
- Unused/Unsupported Cloud Regions
- Use Alternate Authentication Material (4)
- Valid Accounts (4)
- Virtualization/Sandbox Evasion (3)
- Weaken Encryption (2)
- XSL Script Processing

Credential Access
15 techniques
- Brute Force (4)
- Credentials from Password Stores (5)
- Exploitation for Credential Access
- Forced Authentication
- Forge Web Credentials (2)
- Input Capture (4)
- Man-in-the-Middle (2)
- Modify Authentication Process (4)
- Network Sniffing
- OS Credential Dumping (8)
- Steal Application Access Token
- Steal or Forge Kerberos Tickets (4)
- Steal Web Session Cookie
- Two-Factor Authentication Interception
- Unsecured Credentials (7)

Discovery
27 techniques
- Account Discovery (4)
- Application Window Discovery
- Browser Bookmark Discovery
- Cloud Infrastructure Discovery
- Cloud Service Dashboard
- Cloud Service Discovery
- Container and Resource Discovery
- Domain Trust Discovery
- File and Directory Discovery
- Network Service Scanning
- Network Share Discovery
- Network Sniffing
- Password Policy Discovery
- Peripheral Device Discovery
- Permission Groups Discovery (3)
- Process Discovery
- Query Registry
- Remote System Discovery
- Software Discovery (1)
- System Information Discovery
- System Location Discovery
- System Network Configuration Discovery (1)
- System Network Connections Discovery
- System Owner/User Discovery
- System Service Discovery
- System Time Discovery
- Virtualization/Sandbox Evasion (3)

Lateral Movement
9 techniques
- Exploitation of Remote Services
- Internal Spearphishing
- Lateral Tool Transfer
- Remote Service Session Hijacking (2)
- Remote Services (6)
- Replication Through Removable Media
- Software Deployment Tools
- Taint Shared Content
- Use Alternate Authentication Material (4)

Collection
17 techniques
- Archive Collected Data (3)
- Audio Capture
- Automated Collection
- Clipboard Data
- Data from Cloud Storage Object
- Data from Configuration Repository (2)
- Data from Information Repositories (2)
- Data from Local System
- Data from Network Shared Drive
- Data from Removable Media
- Data Staged (2)
- Email Collection (3)
- Input Capture (4)
- Man in the Browser
- Man-in-the-Middle (2)
- Screen Capture
- Video Capture

Command and Control
16 techniques
- Application Layer Protocol (4)
- Communication Through Removable Media
- Data Encoding (2)
- Data Obfuscation (3)
- Dynamic Resolution
- Encrypted Channel (2)
- Fallback Channels
- Ingress Tool Transfer
- Multi-Stage Channels
- Non-Application Layer Protocol
- Non-Standard Port
- Protocol Tunneling
- Proxy (4)
- Remote Access Software
- Traffic Signaling (1)
- Web Service (3)

Exfiltration
9 techniques
- Automated Exfiltration (1)
- Data Transfer Size Limits
- Exfiltration Over Alternative Protocol (3)
- Exfiltration Over C2 Channel
- Exfiltration Over Other Network Medium (1)
- Exfiltration Over Physical Medium (1)
- Exfiltration Over Web Service (2)
- Scheduled Transfer
- Transfer Data to Cloud Account

Impact
13 techniques
- Account Access Removal
- Data Destruction
- Data Encrypted for Impact
- Data Manipulation (3)
- Defacement (2)
- Disk Wipe (2)
- Endpoint Denial of Service (4)
- Firmware Corruption
- Inhibit System Recovery
- Network Denial of Service (2)
- Resource Hijacking
- Service Stop
- System Shutdown/Reboot

Take aways

- MITRE Emulation Plans can be leveraged by a technically competent reader to simulate realistic attacks
- MITRE Emulation Plans helped our purpose by allowing for efficient adoption and customization

Next steps

- If necessary further red team experiments, e.g. compromise of active directory
- Utilization of the gathered data for detection experiments (based on network as well as endpoint data)
- Experimentation regarding automated remediation of detected attacks

Want to get involved? Participate in SAPPAN’s end user commitee

Become part of the SAPPAN end user committee!

What we need your help with:

- Interview after demonstration of SAPPAN results and discussion of achievements

How much time it all takes:

- 2 surveys + demonstration, 2 hours each

What you can expect in return:

- No cash
- Early access to results (papers)
- Early access to practical implementations (if open source)
- Access to new detectors as they are developped in showcases

What to do if you are interested:

- Hit me up: mischa.obrecht@dreamlab.net

Thank
You!