A Master Attack Methodology for an AI-Based Automated Attack Planner for Smart Cities

GREGORY FALCO

ARUN VISWANATHAN

CARLOS CALDERA

HOWARD SHROBE

Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109 USA

IEEE Access, 2018

Outline

- Introduction
- Background
- Design
- Evaluation
  - A case
- Conclusion

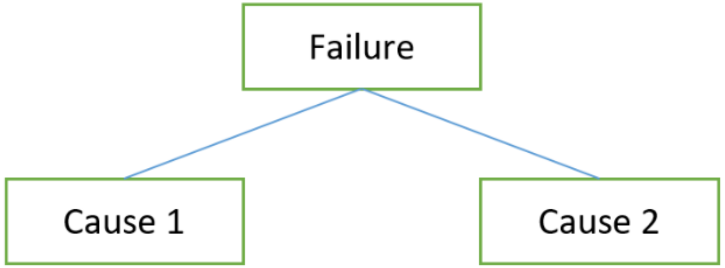
Introduction

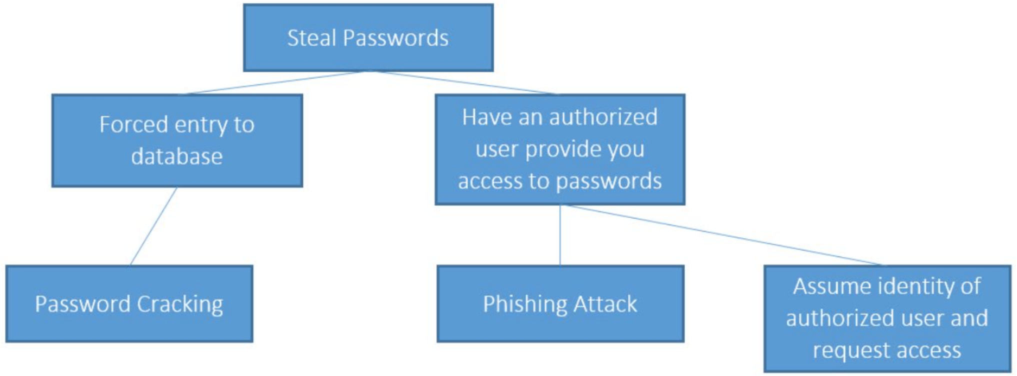
- ICS(Industrial Control Systems)
  - Highly specialized computers used in smart cities
- IIoT(Industrial Internet of Things)
  - Consists of ICSs and sensors
- ICS is vulnerable
  - One in every five ICSs is attacked each month
  - Administrators have not been active participants in cybersecurity
- Traditional approach: enumerating attack vectors
  - Creating attack trees → tedious and requires highly technical knowledge
- Cooperating with AI
  - This paper focuses on “industry sector agnostic”

A methodology for creating attack trees based on AI-based planners

Background

- Attack tree comes from the Fault tree
  - use qualitative measures to score each leaf
- Attack tree's benefits
  - Help to structure the complex problem of defending against cyberattacks ……
  - Common attack trees are reusable
- Attack tree challenges
  - Need to be prepared by an expert who has both full knowledge of the system and a comprehensive understanding of how best to attack the system
  - Semantic idiosyncrasies in different researchers




Steal Passwords
Forced entry to database
Have an authorized user provide you access to passwords
Password Cracking
Phishing Attack
Assume identity of authorized user and request access

Background

- Shrobe and Howard developed an automated attack tree generator using classical planning
  - Classical planning is a branch of artificial intelligence.

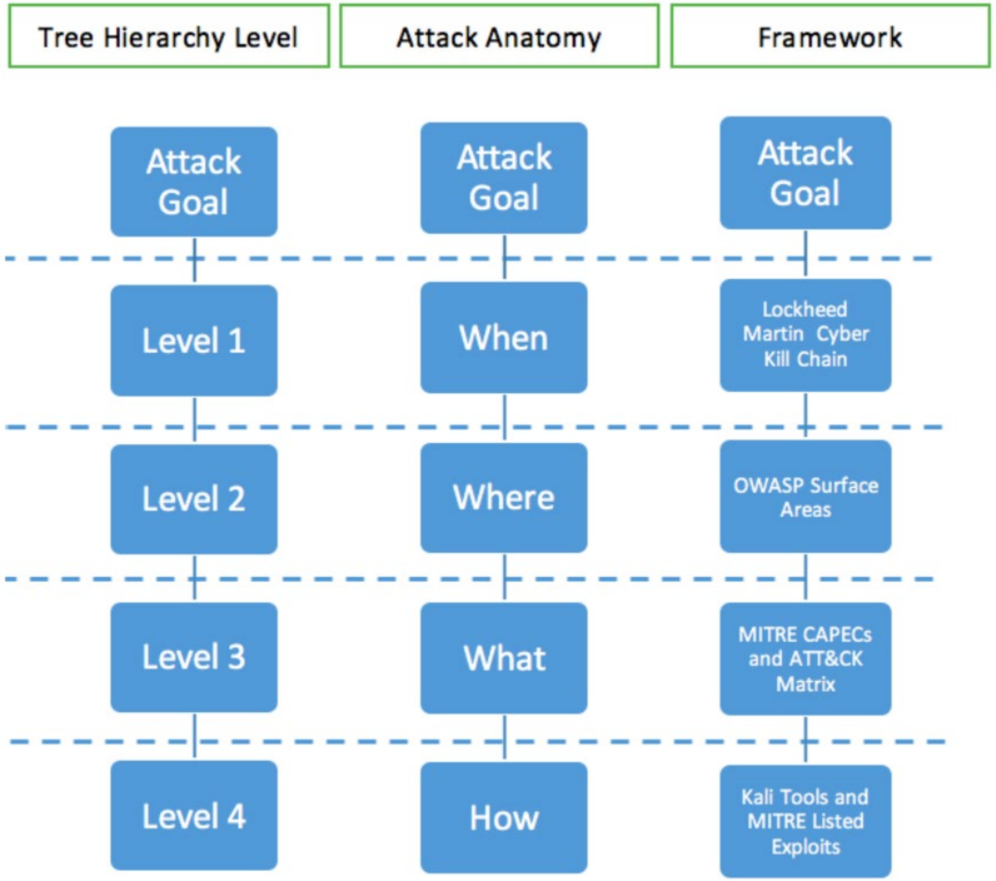
- Classical planning generator components:
  - An abstracted rule set describing methods
  - A detailed system description

- Classical planning generator challenges
  - Do not incorporate standardized language from the cybersecurity community into the trees
  - Do not cover all modern systems – especially with the recent surge of IoT and IIoT systems.

Design

A methodology for creating attack trees

- Hierarchy level:
  - “when” + “where” + “what” + “how”
- When: The Cyber Kill Chain
  - Sequence of phases for waging attacks
- Where: OWASP Surfaces
  - Surface area for waging an attack
- What: CAPECs & MITRE ATT&CK
  - Actions required for waging an attack
- How: Kali & MITRE Exploits
  - Tools needed for waging an attack

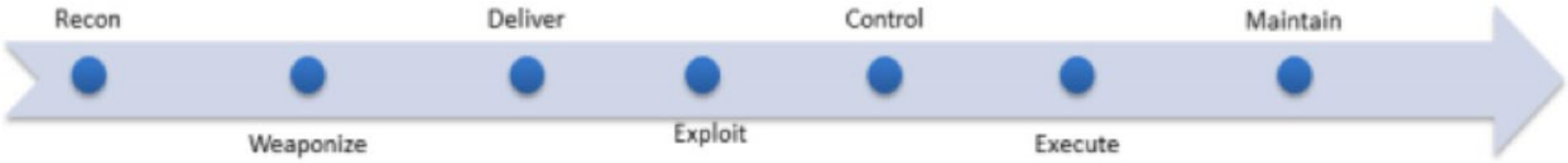


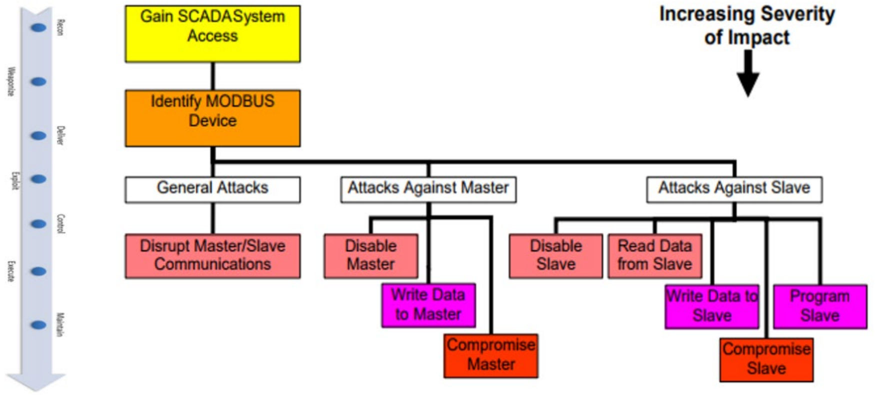
Tree Hierarchy Level	Attack Anatomy	Framework
Attack Goal	Attack Goal	Attack Goal
Level 1	When	Lockheed Martin Cyber Kill Chain
Level 2	Where	OWASP Surface Areas
Level 3	What	MITRE CAPECs and ATT&CK Matrix
Level 4	How	Kali Tools and MITRE Listed Exploits

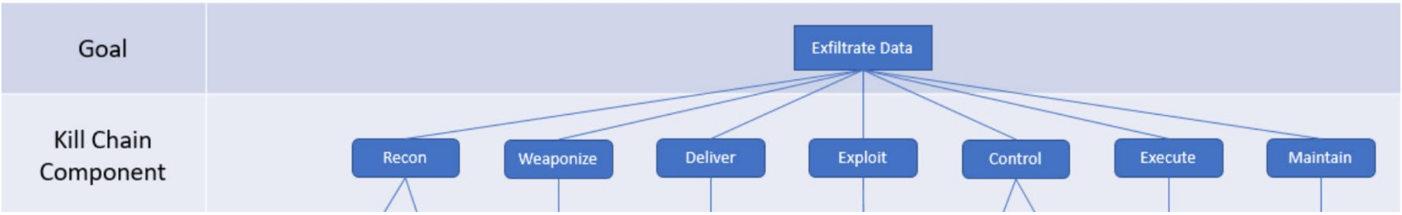
Design

A methodology for creating attack trees

- When: The Cyber Kill Chain
  - Sequence of phases for waging attacks





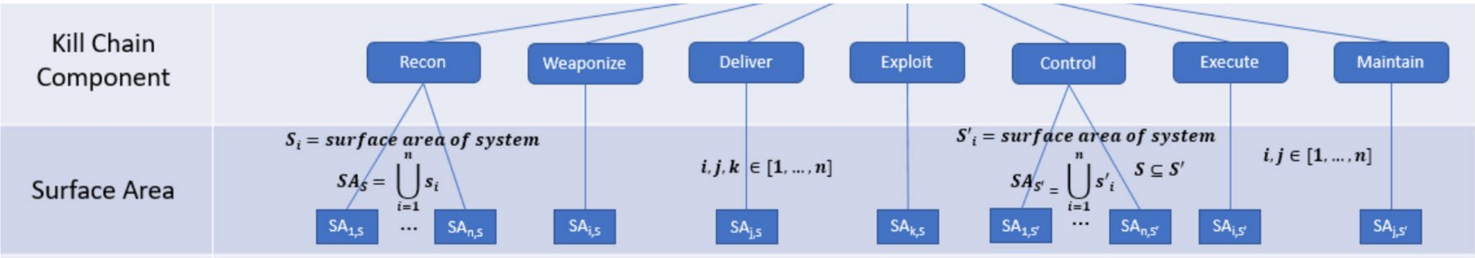


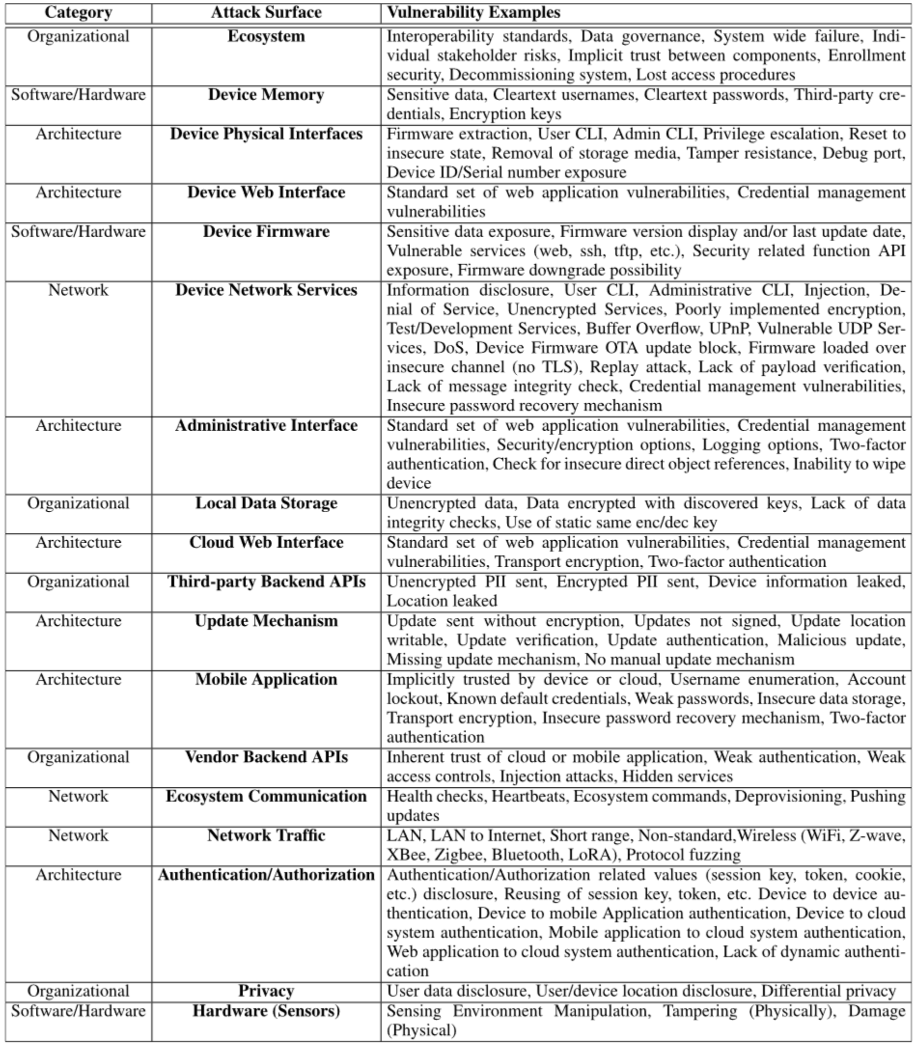
Design

A methodology for creating attack trees

- Where: OWASP Surfaces
  - Surface area for waging an attack

- Categories
  - Software/Hardware
  - Architecture
  - Network
  - Organizational



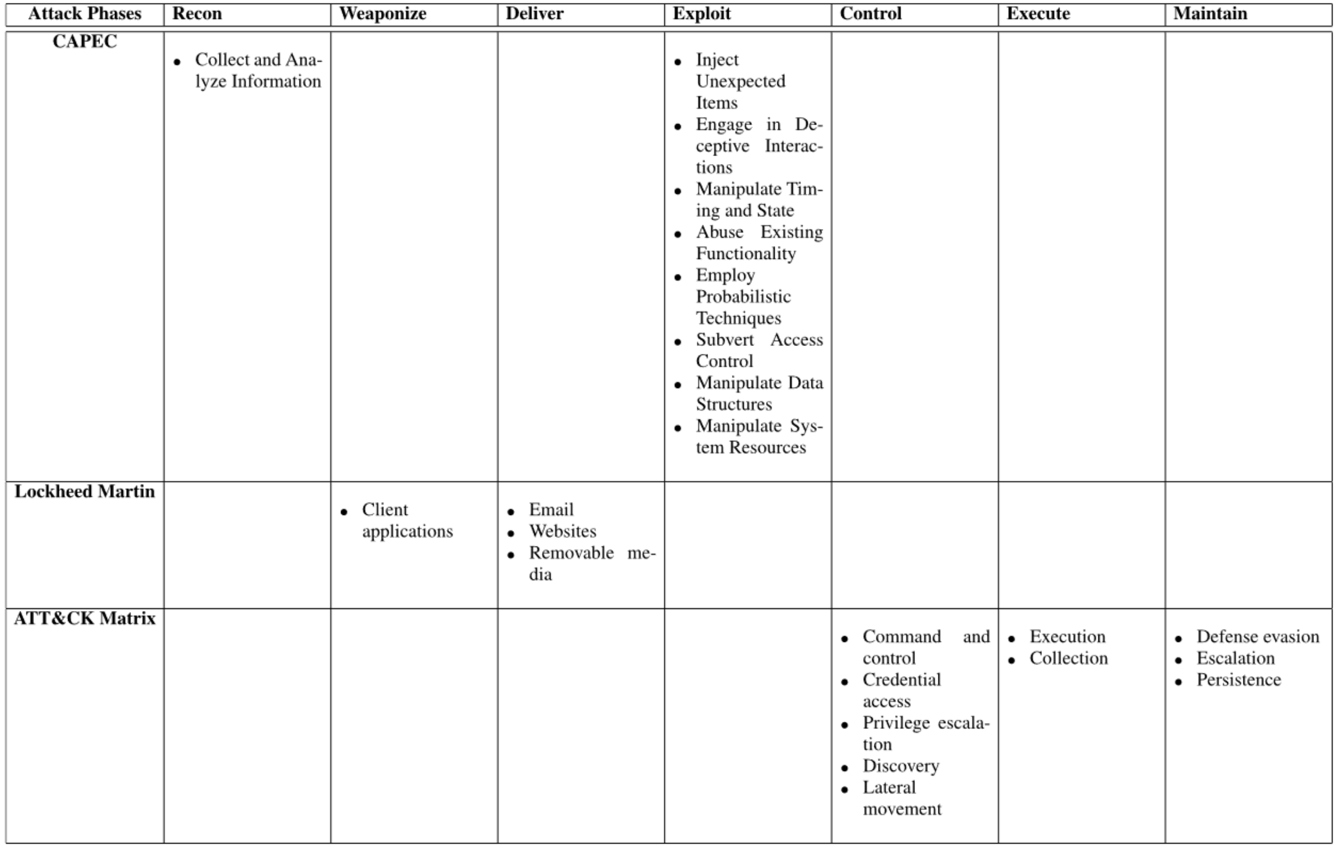


Category	Attack Surface	Vulnerability Examples
Organizational	**Ecosystem**	Interoperability standards, Data governance, System wide failure, Individual stakeholder risks, Implicit trust between components, Enrollment security, Decommissioning system, Lost access procedures
Software/Hardware	**Device Memory**	Sensitive data, Cleartext usernames, Cleartext passwords, Third-party credentials, Encryption keys
Architecture	**Device Physical Interfaces**	Firmware extraction, User CLI, Admin CLI, Privilege escalation, Reset to insecure state, Removal of storage media, Tamper resistance, Debug port, Device ID/Serial number exposure
Architecture	**Device Web Interface**	Standard set of web application vulnerabilities, Credential management vulnerabilities
Software/Hardware	**Device Firmware**	Sensitive data exposure, Firmware version display and/or last update date, Vulnerable services (web, ssh, tftp, etc.), Security related function API exposure, Firmware downgrade possibility
Network	**Device Network Services**	Information disclosure, User CLI, Administrative CLI, Injection, Denial of Service, Unencrypted Services, Poorly implemented encryption, Test/Development Services, Buffer Overflow, UPnP, Vulnerable UDP Services, DoS, Device Firmware OTA update block, Firmware loaded over insecure channel (no TLS), Replay attack, Lack of payload verification, Lack of message integrity check, Credential management vulnerabilities, Insecure password recovery mechanism
Architecture	**Administrative Interface**	Standard set of web application vulnerabilities, Credential management vulnerabilities, Security/encryption options, Logging options, Two-factor authentication, Check for insecure direct object references, Inability to wipe device
Organizational	**Local Data Storage**	Unencrypted data, Data encrypted with discovered keys, Lack of data integrity checks, Use of static same enc/dec key
Architecture	**Cloud Web Interface**	Standard set of web application vulnerabilities, Credential management vulnerabilities, Transport encryption, Two-factor authentication
Organizational	**Third-party Backend APIs**	Unencrypted PII sent, Encrypted PII sent, Device information leaked, Location leaked
Architecture	**Update Mechanism**	Update sent without encryption, Updates not signed, Update location writable, Update verification, Update authentication, Malicious update, Missing update mechanism, No manual update mechanism
Architecture	**Mobile Application**	Implicitly trusted by device or cloud, Username enumeration, Account lockout, Known default credentials, Weak passwords, Insecure data storage, Transport encryption, Insecure password recovery mechanism, Two-factor authentication
Organizational	**Vendor Backend APIs**	Inherent trust of cloud or mobile application, Weak authentication, Weak access controls, Injection attacks, Hidden services
Network	**Ecosystem Communication**	Health checks, Heartbeats, Ecosystem commands, Deprovisioning, Pushing updates
Network	**Network Traffic**	LAN, LAN to Internet, Short range, Non-standard,Wireless (WiFi, Z-wave, XBee, Zigbee, Bluetooth, LoRA), Protocol fuzzing
Architecture	**Authentication/Authorization**	Authentication/Authorization related values (session key, token, cookie, etc.) disclosure, Reusing of session key, token, etc. Device to device authentication, Device to mobile Application authentication, Device to cloud system authentication, Mobile application to cloud system authentication, Web application to cloud system authentication, Lack of dynamic authentication
Organizational	**Privacy**	User data disclosure, User/device location disclosure, Differential privacy
Software/Hardware	**Hardware (Sensors)**	Sensing Environment Manipulation, Tampering (Physically), Damage (Physical)

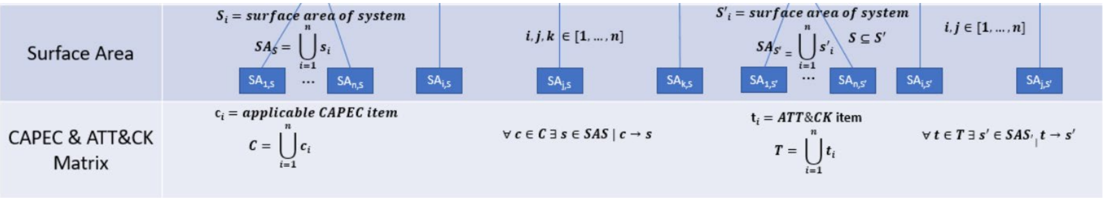
Design

A methodology for creating attack trees

- What: CAPECs & MITRE ATT&CK
  - Actions required for waging an attack



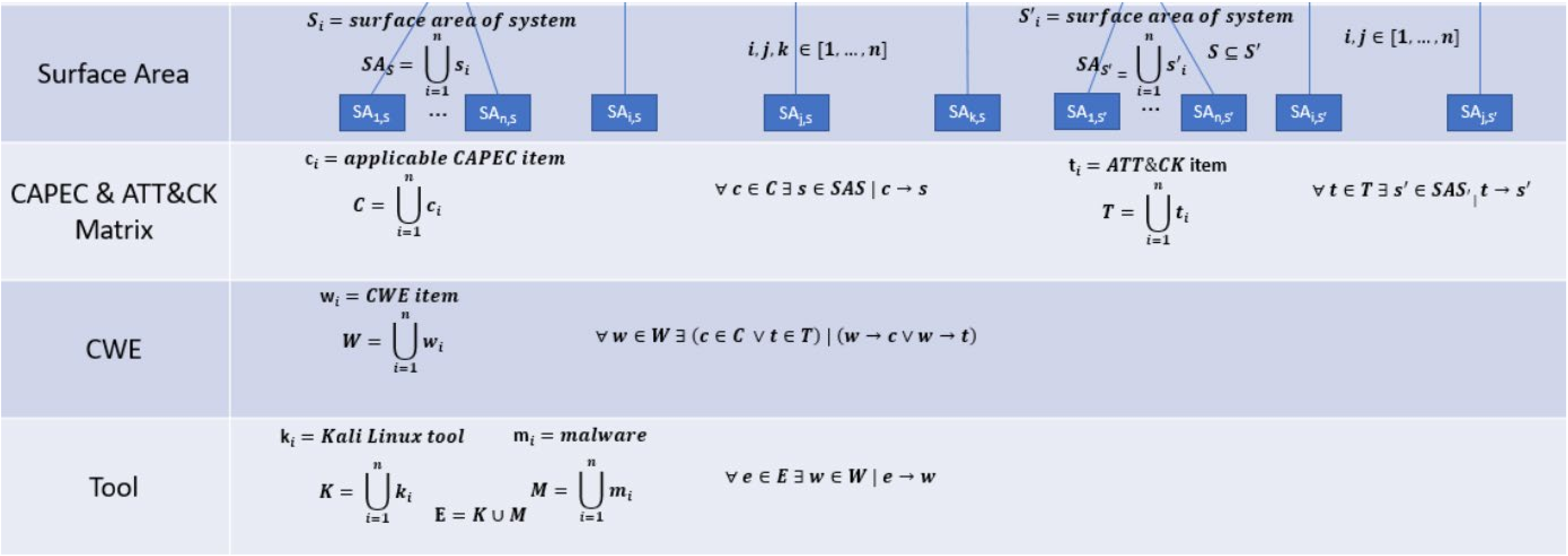
Attack Phases	Recon	Weaponize	Deliver	Exploit	Control	Execute	Maintain
CAPEC	• Collect and Analyze Information			• Inject Unexpected Items • Engage in Deceptive Interactions • Manipulate Timing and State • Abuse Existing Functionality • Employ Probabilistic Techniques • Subvert Access Control • Manipulate Data Structures • Manipulate System Resources			
Lockheed Martin		• Client applications	• Email • Websites • Removable media				
ATT&CK Matrix					• Command and control • Credential access • Privilege escalation • Discovery • Lateral movement	• Execution • Collection	• Defense evasion • Escalation • Persistence


Surface Area
$S_i = surface\ area\ of\ system$
$SA_S = \bigcup_{i=1}^{n} s_i$
$SA_{1,s}$ … $SA_{n,s}$
$SA_{i,s}$
$i, j, k \in [1, \dots, n]$
$SA_{j,s}$
$SA_{k,s}$
$S'_i = surface\ area\ of\ system$
$SA_{S'} = \bigcup_{i=1}^{n} s'_i$
$S \subseteq S'$
$SA_{1,s'}$ … $SA_{n,s'}$
$SA_{i,s'}$
$i, j \in [1, \dots, n]$
$SA_{j,s'}$
CAPEC & ATT&CK Matrix
$c_i = applicable\ CAPEC\ item$
$C = \bigcup_{i=1}^{n} c_i$
$\forall c \in C \exists s \in SAS \mid c \rightarrow s$
$t_i = ATT\&CK$ item
$T = \bigcup_{i=1}^{n} t_i$
$\forall t \in T \exists s' \in SAS_{'} \mid t \rightarrow s'$

Design

A methodology for creating attack trees

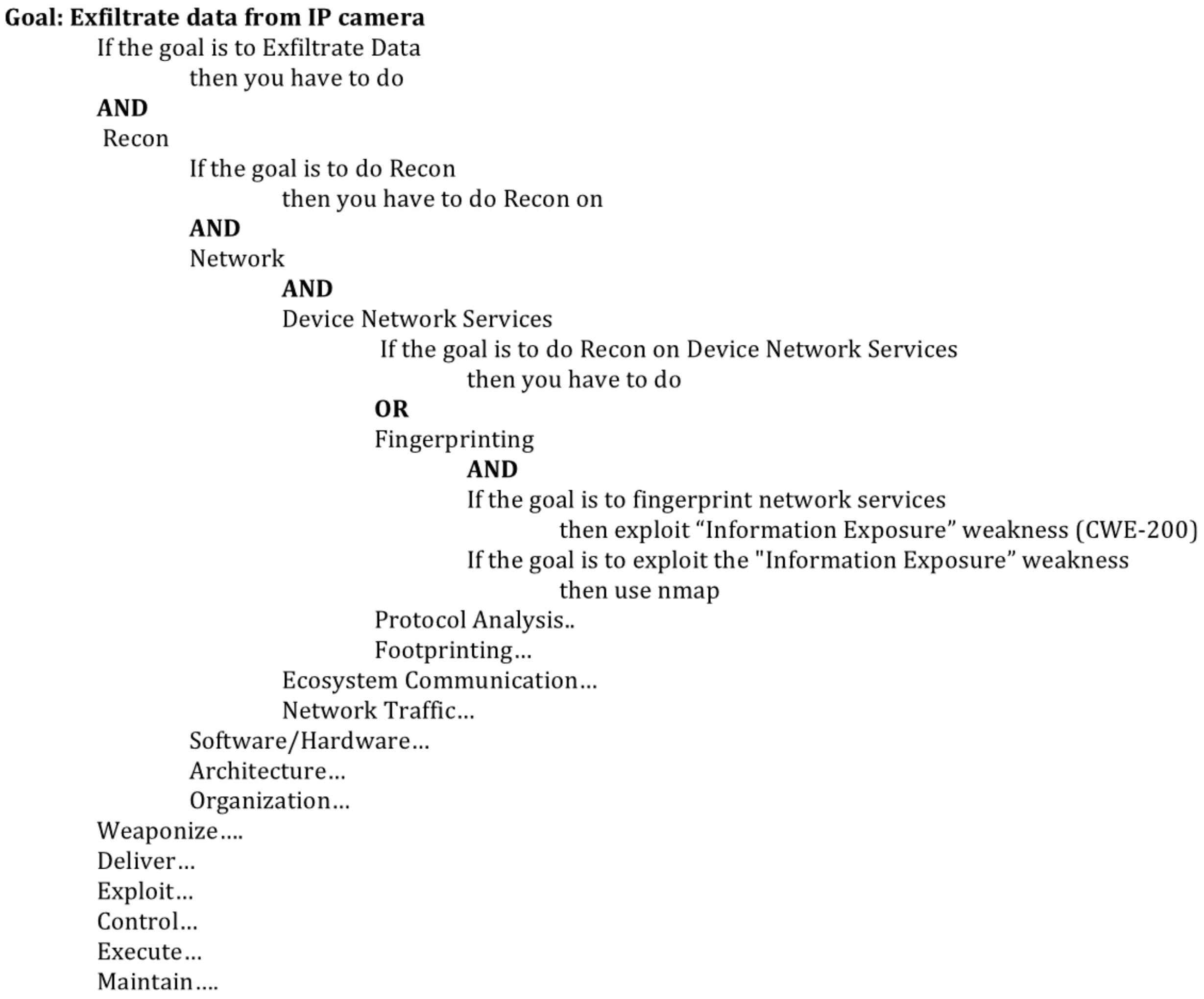
- How: Kali & MITRE Exploits
  - Tools needed for waging an attack



Surface Area	$S_i = surface\ area\ of\ system$ $SA_S = \bigcup_{i=1}^{n} s_i$ $SA_{1,S}$ … $SA_{n,S}$ $SA_{i,S}$	$i, j, k \in [1, \dots, n]$ $SA_{j,S}$ $SA_{k,S}$	$S'_i = surface\ area\ of\ system$ $SA_{S'} = \bigcup_{i=1}^{n} s'_i$ $S \subseteq S'$ $SA_{1,S'}$ … $SA_{n,S'}$	$i, j \in [1, \dots, n]$ $SA_{i,S'}$ $SA_{j,S'}$
CAPEC & ATT&CK Matrix	$c_i = applicable\ CAPEC\ item$ $C = \bigcup_{i=1}^{n} c_i$	$\forall c \in C \exists s \in SAS \mid c \rightarrow s$	$t_i = ATT\&CK$ item $T = \bigcup_{i=1}^{n} t_i$	$\forall t \in T \exists s' \in SAS_{'} \mid t \rightarrow s'$
CWE	$w_i = CWE\ item$ $W = \bigcup_{i=1}^{n} w_i$	$\forall w \in W \exists (c \in C \vee t \in T) \mid (w \rightarrow c \vee w \rightarrow t)$		
Tool	$k_i = Kali\ Linux\ tool$ $m_i = malware$ $K = \bigcup_{i=1}^{n} k_i$ $E = K \cup M$ $M = \bigcup_{i=1}^{n} m_i$	$\forall e \in E \exists w \in W \mid e \rightarrow w$		

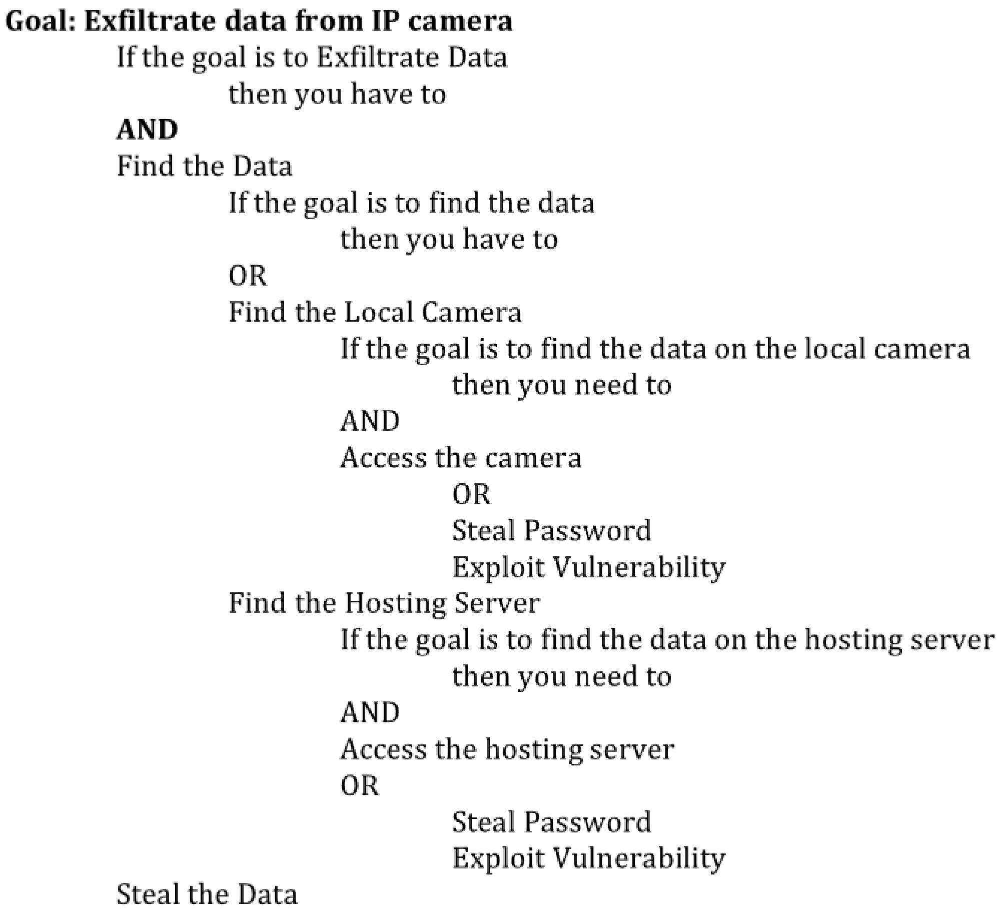
Evaluation

- Automatically generated



```
Goal: Exfiltrate data from IP camera
    If the goal is to Exfiltrate Data
        then you have to do
    AND
    Recon
        If the goal is to do Recon
            then you have to do Recon on
        AND
        Network
            AND
            Device Network Services
                If the goal is to do Recon on Device Network Services
                    then you have to do
                OR
                Fingerprinting
                    AND
                    If the goal is to fingerprint network services
                        then exploit "Information Exposure" weakness (CWE-200)
                    If the goal is to exploit the "Information Exposure" weakness
                        then use nmap
                Protocol Analysis..
                Footprinting…
            Ecosystem Communication…
            Network Traffic…
        Software/Hardware…
        Architecture…
        Organization…
    Weaponize….
    Deliver…
    Exploit…
    Control…
    Execute…
    Maintain….
```

- Hand-drawn



```
Goal: Exfiltrate data from IP camera
    If the goal is to Exfiltrate Data
        then you have to
    AND
    Find the Data
        If the goal is to find the data
            then you have to
        OR
        Find the Local Camera
            If the goal is to find the data on the local camera
                then you need to
            AND
            Access the camera
                OR
                Steal Password
                Exploit Vulnerability
        Find the Hosting Server
            If the goal is to find the data on the hosting server
                then you need to
            AND
            Access the hosting server
            OR
                Steal Password
                Exploit Vulnerability
    Steal the Data
```

Conclusion

- Developing attack trees has operational challenges
- Using AI planning can ease this operational burden.
- Existing works has not generated a comprehensive attack rule set that can be used across disparate critical infrastructure sectors
- By combining attack frameworks, we have developed a master attack method