ADVERSARY MODELS IN LITERATURE (AND IN REALITY)

On the importance of adversary modelling

ON THE IMPORTANCE OF ADVERSARY MODELLING

WANTED BY THE FBI

KIM IL

Conspiracy to Commit Wire Fraud and Bank Fraud; Conspiracy to Commit Computer-Related Fraud (Computer Intrusion)

DESCRIPTION

Aliases: Julien Kim, Tony Walker	
Place of Birth: Democratic People's Republic of Korea (North Korea)	**Hair:** Black
Eyes: Brown	**Sex:** Male
Race: Asian	**Languages:** English, Korean, Mandarin Chinese

REMARKS

Kim is a North Korean citizen last known to be in North Korea. Kim has traveled to Singapore and Russia in the past and has reported a date of birth in 1994.

CAUTION

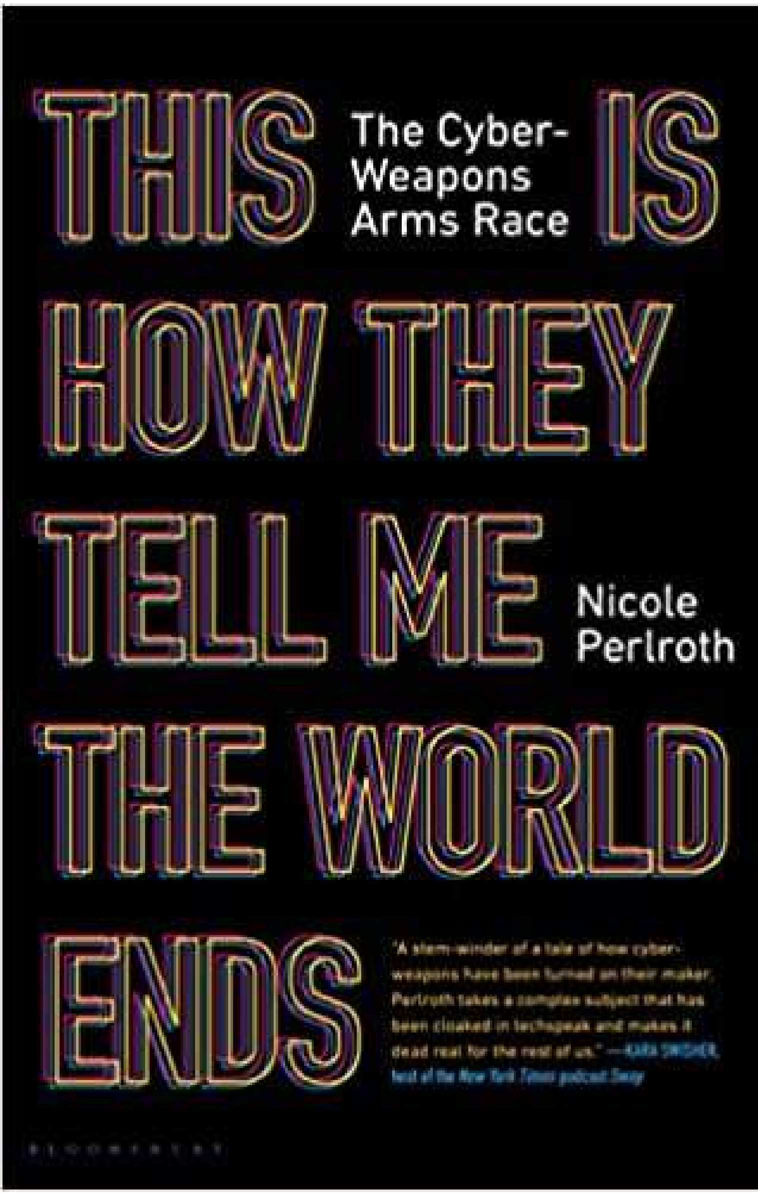
Kim Il is allegedly a state-sponsored North Korean hacker who is part of an alleged criminal conspiracy responsible for some of the costliest computer intrusions in history. These intrusions caused damage to computer systems of, and stole currency and virtual currency from, numerous victims.

Kim was alleged to be a participant in a wide-ranging criminal conspiracy undertaken by a group of hackers of the North Korean government's Reconnaissance General Bureau (RGB). The conspiracy comprised North Korean hacking groups that some private cybersecurity researchers have labeled the "Lazarus Group" and Advanced Persistent Threat 38 (APT38). For his part in the conspiracy, Kim is alleged to have been directly involved in the development and dissemination of a malicious cryptocurrency application, in cyber-enabled heists from financial institutions, and in the Marine Chain initial coin offering. On December 8, 2020, a federal arrest warrant was issued for Kim in the United States District Court, Central District of California, after he was charged with one count of conspiracy to commit wire fraud and bank fraud, and one count of conspiracy to commit computer fraud (computer intrusions).

If you have any information concerning this person, please contact your local FBI office or the nearest American Embassy or Consulate.

Field Office: Los Angeles

Not predicting profile



A really short adversary model

An attacker is interested in three things

- Reliabiilty of its tools
- Invisibility
- Persistence in the target system

Adversary model and work

- A robustness/resilience metric considers (predicts) the attacker work to defeat a system ie to build a successful intrusion
- We can measure the amount of work as the number of actions (including repetitions due to failures)
- The number of actions may not be enough because we need to measure the power of the work = how complex the work is
- Power of the work = overall complexity of all the actions
- Time as a measure of effort = robustness/resilience increase with the time to penetrate a system
- We compare systems or system versions in terms of the time and/or the work of a successful intrusion

Adversary Models

- An adversary model formalizes the behavior of an attacker in an intrusion. Depending on the detail level, a model may consist of
  - an algorithm
  - a set of attributes to describe the attacker capabilities and goals.
- There is a large amount of work on adversary models in crypto to formalize an attack on a system or protocol.
- One of the first and most popular model is Dolev–Yao where the adversary has the abilities to
  - listen to all traffic on a network
  - initiate a connection with and send data to any other network client on the network

this models complexity (power) of actions not time or the amount of work

Adversary Models, Bellare and Rogaway, 1993

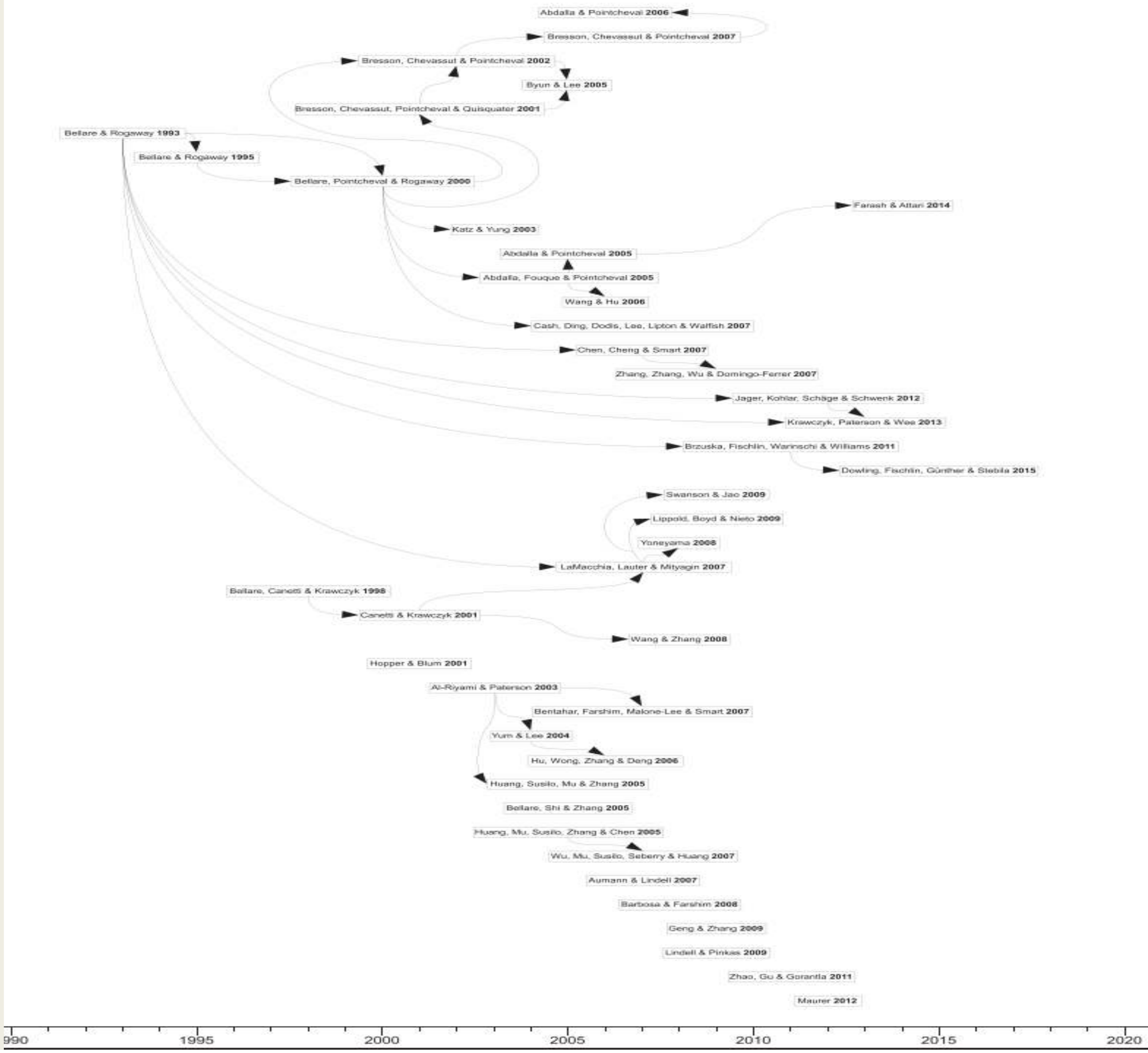
- The adversary controls all the communications by interacting with a set of oracles, each representing an instance of a principal in a protocol run.
- Interactions with the adversary are called oracle queries and the list is
  - *Send*(U, s, M): the adversary makes the principals run the protocol normally.
  - *Reveal*(U, s): the adversary's ability to find old session keys.
  - *Corrupt*(U, K): it returns the oracle's internal state and sets the long-term key of U to the value K the adversary chooses
  - *Test*(U, s): if an oracle has accepted a session key K the adversary can attempt to distinguish it from a random key to determine the protocol security. A random bit b is chosen;
    - if b= 0 return K
    - if b= 1 return a random string from the same distribution of session keys.
- What the adversary can do and its detailed abilities and we could deduce time but here time s not important

Further adversary models

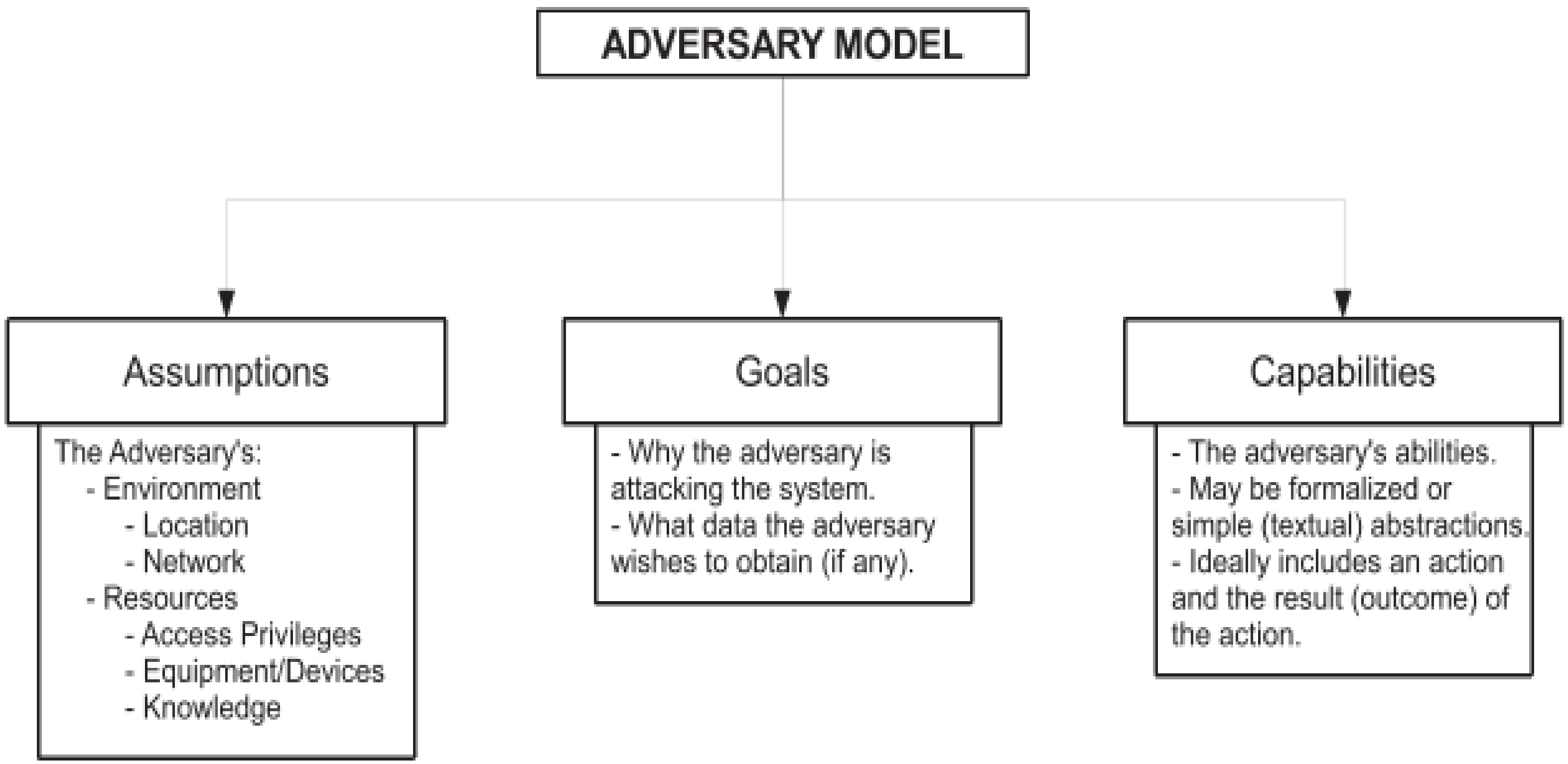
- The ability of delaying the messages that are exchanged
- Insider and outsider attackers (access to a master key)
- Extract some bits from the private key
- Resending a message
- Covert adversary that tries to hide its presence
- Little attention to physical attacks …

History of adversary models

(and just for one system element)



Attacker (Adversary) Models

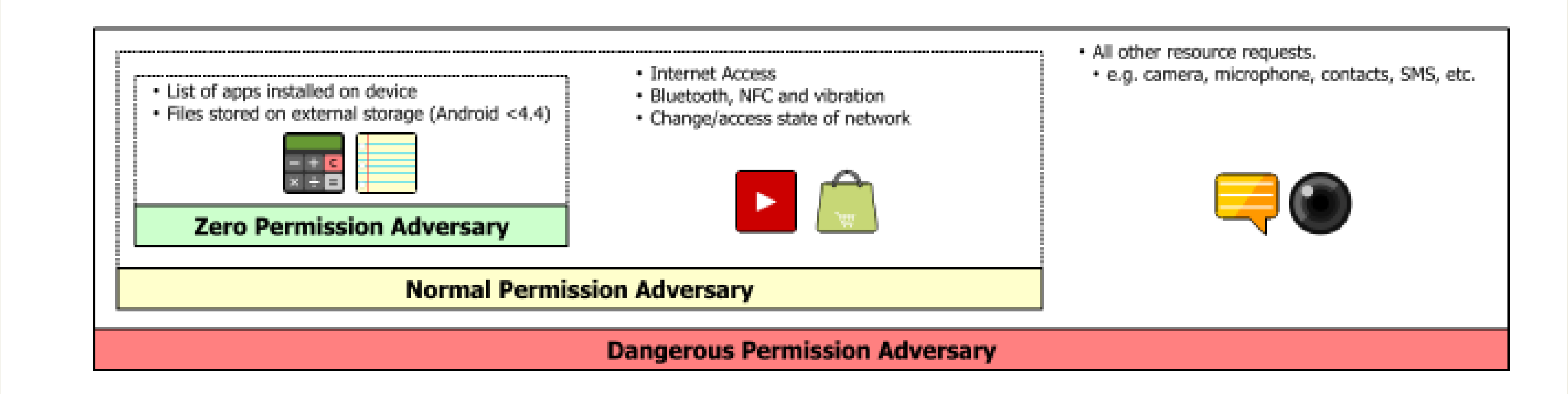


The role of the adversary model in applied security research

ELSEVIER
Computers & Security
Volume 81, March 2019, Pages 156-181

An example: mobile adversary

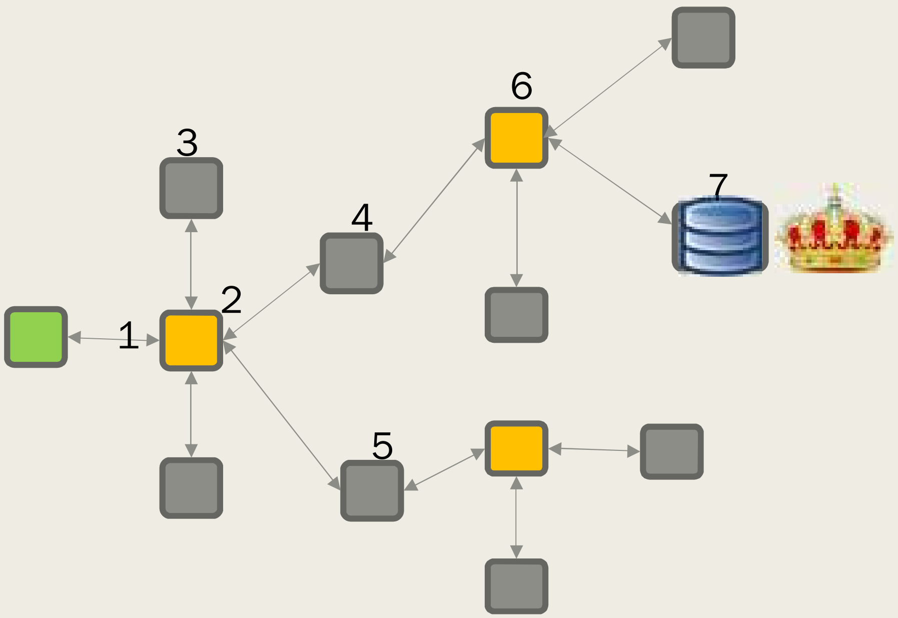
- Adversary assumptions
  - *zero permission,*
  - *normal permission*
  - *dangerous permission.*
- Adversary goals
  - *collection of sensitive user information*
  - *collection of user file*
- Adversary capabilities
  - *send and receive data via the Internet*
  - *accessing files stored on the device's external storage*



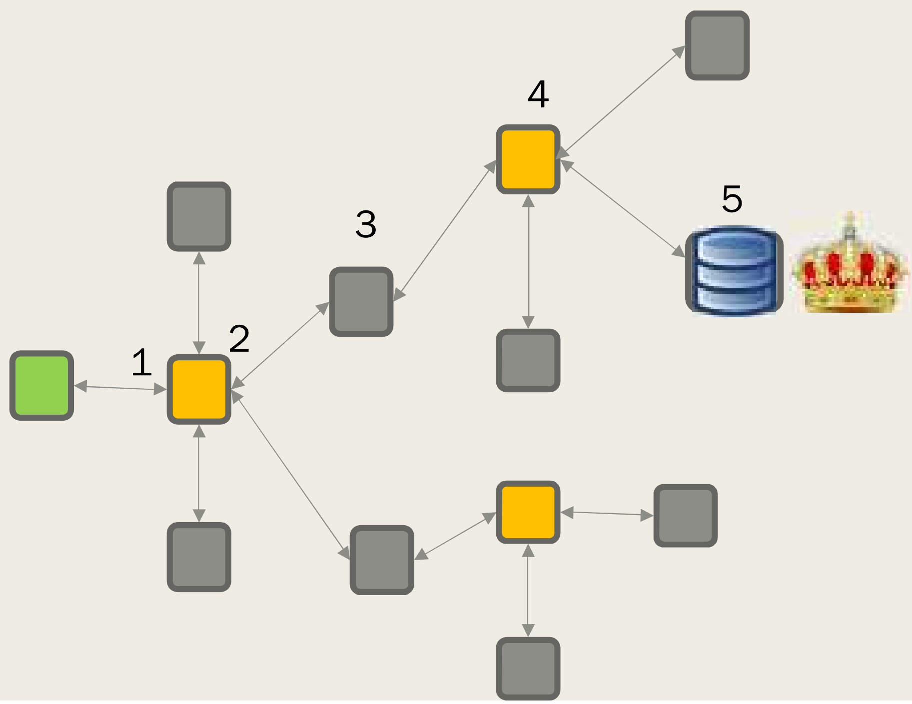
An important issue

- Some models cannot support emulation as they describe what an adversary can do but not how or when it does it
- Starting from adversary actions we can define alternative adversaries by pairing the actions with the strategy to select the action to execute
- Actions and strategy are important to evaluate time and hence non functional properties such as «how long this system can resist?»
- The attacker strategy is of little interest when, as in crypto,
  - *we are interested in knowing only if a module or a protocol can be attacked*
  - *we assume the adversary knows the optimal sequence of actions if there is one*

The importance of the strategy



Improve the strategy



Time changes

The attacker strategy

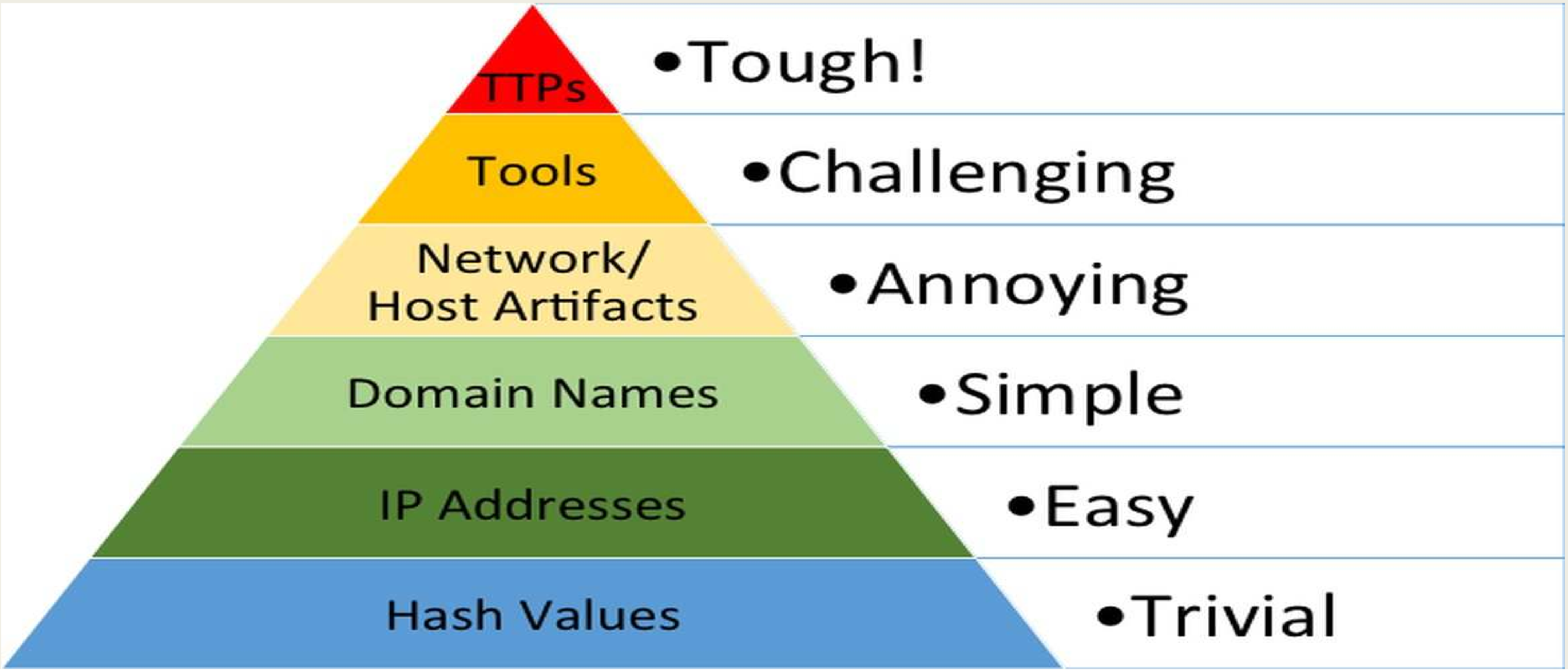
- To know how long an intrusion takes we consider both
  - *the set of actions the attacker can implement*
  - *the strategy to compose these actions ie to select the one to execute at each step of the intrusion*
- The attacker rights determine the freedom degrees of the selection at any time
- The strategy models the attacker priorities
  - *Tolerated noise*
  - *Available resources*
  - *Time to goal*
  - *Persistence or not persistence into the system*
- Anyway, the strategies of intelligent adversaries share some properties
  - *Avoid useless repetitions of an action*
  - *Minimal effort to reach the intrusion goal*

On the importance of time

- The strategy determines how long it take the adversary to reach its goal
- Any system includes some mechanisms/modules (IDS and/or SIEM) to discover intrusions, anomalies or system manipulations such as
  - *Delaying messages*
  - *Accessing the master keys*
  - *....*
- These mechanisms define an upper bound on the time of an intrusion and strongly contribute to the overall robustness and resilience
- The time to execute the intrusion determines the success or failures of the attacker and is also fundamental to evaluate robustness and resilience
- This problem is usually neglected in cryptography

How to describe the actions

“The Pyramid of Pain”, introduced in 2013 by security professional David J Bianco shows there are several way to describe an attacker



The pyramid of pain

- Hashes = signatures of the modules and tools the attacker uses. Easy to defeat by permutation of the source code
- IP addresses = nodes the attacker uses to launch its attacks or in its C2 network, Easy to defeat by creating a new botnet and by hiding through stepping stones
- Domain names = domain name generation algorithm where the malware and the command&control network can meet
- Network artifacts = the protocols supporting interaction with the target system
- Host artifacts = registry keys or values known to be created by the attacker, files or directories created in certain places or using certain names
- Tools = software adversary uses e.g Tor, Windows Task Scheduler, GCC, Powershell, etc. The software itself might not be directly malicious, but the specific use, time or location might be indicative of malicious or at least suspicious

TTPs = MITRE ATT&CK matrix

- A knowledge base of the methods that attackers use against enterprise systems, cloud apps, mobile devices, and industrial control systems.
- A de facto standard used in tech reports about real intrusions
- ATT&CK, which stands for Adversarial Tactics, Techniques, and Common Knowledge, to understand how cyber attackers think and work.
  - Tactics describe their goals, like getting inside *a network* or stealing credentials (short term goals to achieve the global one)
  - *Techniques show how a tactics can be achieved*
  - Procedures are highly detailed examples of the tools and actions of specific attacker
- MITRE has defined several matrices that depend upon the target system, the attacker status and the technological domains (Linux, Windows)

MITRE ATT&CK matrix flavors

- ATT&CK Enterprise— The most commonly referenced matrix. It mostly contains techniques for the post-exploitation portion of an intrusion. The information is broken into these platforms:
  - *Operating systems—Microsoft Windows, macOS and Linux*
  - *Cloud platforms—Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform*
  - *Cloud services—Microsoft Office 365, Microsoft Azure Active Directory and generic SaaS platforms*
- ATT&CK Mobile—It covers techniques involving access and networkbased effects that adversaries can use without device access. It encompasses techniques for Android and iOS.
- ATT&CK ICS—ATT&CK for ICS is the knowledge base specific to the tactics and techniques that attackers may use while operating within an ICS network.
- PRE-ATT&CK—Other ATT&CK matrices aim to enumerate tactics and techniques used as part of the post-exploitation attack stages, PRE-ATT&CK enumerates the tactics for pre-exploitation
  - *Technical Weakness Identification,*
  - *Target Selection and Technical Information Gathering,*

the 2020 roadmap indicates that MITRE aims to merge these matrices into a single model.

The tactics x tecniques ICS matrix

Initial Access	Execution	Persistence	Evasion	Discovery	Lateral Movement	Collection	Command and Control	Inhibit Response Function	Impair Process Control	Impact
Data Historian Compromise	Change Program State	Hooking	Exploitation for Evasion	Control Device Identification	Default Credentials	Automated Collection	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Drive-by Compromise	Command-Line Interface	Module Firmware	Indicator Removal on Host	I/O Module Discovery	Exploitation of Remote Services	Data from Information Repositories	Connection Proxy	Alarm Suppression	Change Program State	Denial of Control
Engineering Workstation Compromise	Execution through API	Program Download	Masquerading	Network Connection Enumeration	External Remote Services	Detect Operating Mode	Standard Application Layer Protocol	Block Command Message	Masquerading	Denial of View
Exploit Public-Facing Application	Graphical User Interface	Project File Infection	Rogue Master Device	Network Service Scanning	Program Organization Units	Detect Program State		Block Reporting Message	Modify Control Logic	Loss of Availability
External Remote Services	Man in the Middle	System Firmware	Rootkit	Network Sniffing	Remote File Copy	I/O Image		Block Serial COM	Modify Parameter	Loss of Control
Internet Accessible Device	Program Organization Units	Valid Accounts	Spoof Reporting Message	Remote System Discovery	Valid Accounts	Location Identification		Data Destruction	Module Firmware	Loss of Productivity and Revenue
Replication Through Removable Media	Project File Infection		Utilize/Change Operating Mode	Serial Connection Enumeration		Monitor Process State		Denial of Service	Program Download	Loss of Safety
Spearphishing Attachment	Scripting					Point & Tag Identification		Device Restart/Shutdown	Rogue Master Device	Loss of View
Supply Chain Compromise	User Execution					Program Upload		Manipulate I/O Image	Service Stop	Manipulation of Control
Wireless Compromise						Role Identification		Modify Alarm Settings	Spoof Reporting Message	Manipulation of View
						Screen Capture		Modify Control Logic	Unauthorized Command Message	Theft of Operational Information
								Program Download		
								Rootkit		
								System Firmware		
								Utilize/Change Operating Mode		

Some differences with crypto adversaries …

Att&ck Matrix for Containers

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Impact
Exploit Public-Facing Application	Container Service	Implant Internal Image (NAME CHANGE)	Escape to Host	Build Image on Host	Brute Force	Container Resource Discovery	Endpoint Denial of Service
External Remote Services	Deploy Container	Scheduled Task/Job	Scheduled Task/Job	Deploy Container	Brute Force: Password Guessing		Network Denial of Service
Valid Accounts	Scheduled Task/Job	Scheduled Task/Job: Container Orchestration Job	Scheduled Task/Job: Container Orchestration Job	Masquerading	Brute Force: Password Spraying		Resource Hijacking
Valid Accounts: Local Accounts	Scheduled Task/Job: Container Orchestration Job	Valid Accounts	Valid Accounts	Masquerading: Match Legitimate Name or Location	Brute Force: Credential Stuffing		
	User Execution	Valid Accounts: Local Accounts	Valid Accounts: Local Accounts	Valid Accounts	Unsecured Credentials		
	User Execution: Malicious Image			Valid Accounts: Local Accounts	Unsecured Credentials: Credentials in Files		
					Unsecured Credentials: Container API		

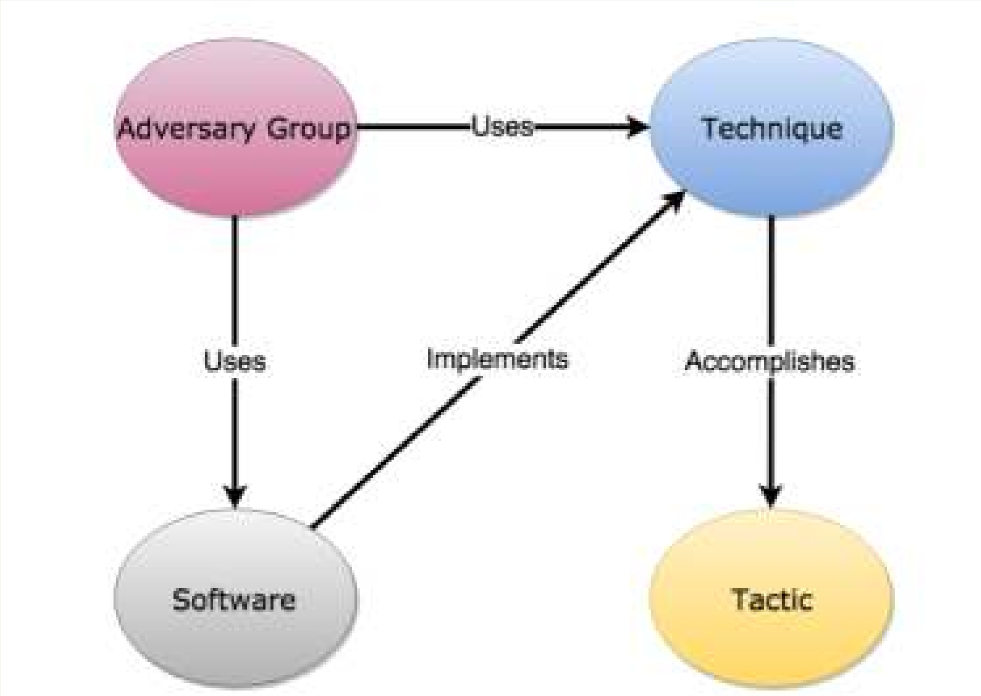
Proposed new techniques and sub-techniques

5 principles for a threat-based approach to network security (Underlying philosophy)

- Use a Threat-based Model – An accurate and well-scoped threat model is necessary to ensure that detection activities are effective against realistic and relevant adversary behaviors.
- Include Post-Compromise Detection – Previously effective perimeter and preventative defenses may fail to keep persistent threats out of a network. Post-compromise detection capabilities are necessary when a threat bypasses defenses or uses new means to enter a network.
- Focus on Behavior – Signatures and indicators are useful with a priori knowledge of adversary infrastructure and tool artifacts, but defensive tools that rely on known signatures may become unreliable when signatures become stale due to a changing threat. Sophisticated defenses also incorporate detecting and learning from post-compromise adversary behavior.
- Iterate – Adversarial tool and technique are constantly evolving. A successful approach requires constant, iterative evolution and refinement of security models, techniques, and tools to account for changing behavior and to understand how networks are compromised by an APT.
- Develop and Test in a Realistic Environment – Analytic development and refinement should be performed in an environment that matches realistic conditions as closely as possible. Behavior generated by real users should be present to account for the expected sensor noise generated by standard use. Whenever possible, detection capabilities should be tested by emulation of adversary behavior within that environment.

Tactics and techniques

- Tactics
  - *represent the “why” of a technique. It is the adversary’s tactical objective: the reason for performing an action.*
  - *serve as useful contextual categories for individual techniques and cover standard notations for things adversaries do in an intrusion, such as persist, discover information, move laterally, execute files, and exfiltrate data*
  - *remain relatively static over time because adversary goals are unlikely to change.*
- Techniques
  - *the foundation of ATT&CK and represent the individual actions adversaries make or pieces of information the adversary learns by performing an action*
  - *represents “how” (the action) an adversary achieves a tactical objective. For example, dump credentials to gain access to useful credentials within a network. Distinct techniques can achieve the same objective and each category includes multiple techniques .*
  - *may also represent “what” an adversary gains by performing an action, each applies to multiple platforms as describes a general platform agnostic behavior. The description is kept general, and details are provided with references to the examples from the different platforms as needed.*



ATT&CK MATRIX RELATIONSHIP

MATRIX ABSTRACTION LEVEL

MITRE ATT&CK matrix tactics (entreprise)

Reconnaissance	to gather information to use to plan future operations.
Resource Development	to establish resources to support operations.
Initial Access	to get into your network.
Execution	to run malicious code
Persistence	to maintain a foothold in your network
Privilege Escalation	to gain higher-level permissions
Defense Evasion	to avoid being detected
Credential Access	to steal account names and passwords
Discovery	to figure out your environment
Lateral Movement	to move through your environment
Collection	to gather data of interest to their goal
Command and Control	to communicate with compromised systems to control them
Exfiltration	to steal data
Impact	to manipulate, interrupt, or destroy your systems and data

MITRE ATT&CK matrix tactics (ICS)

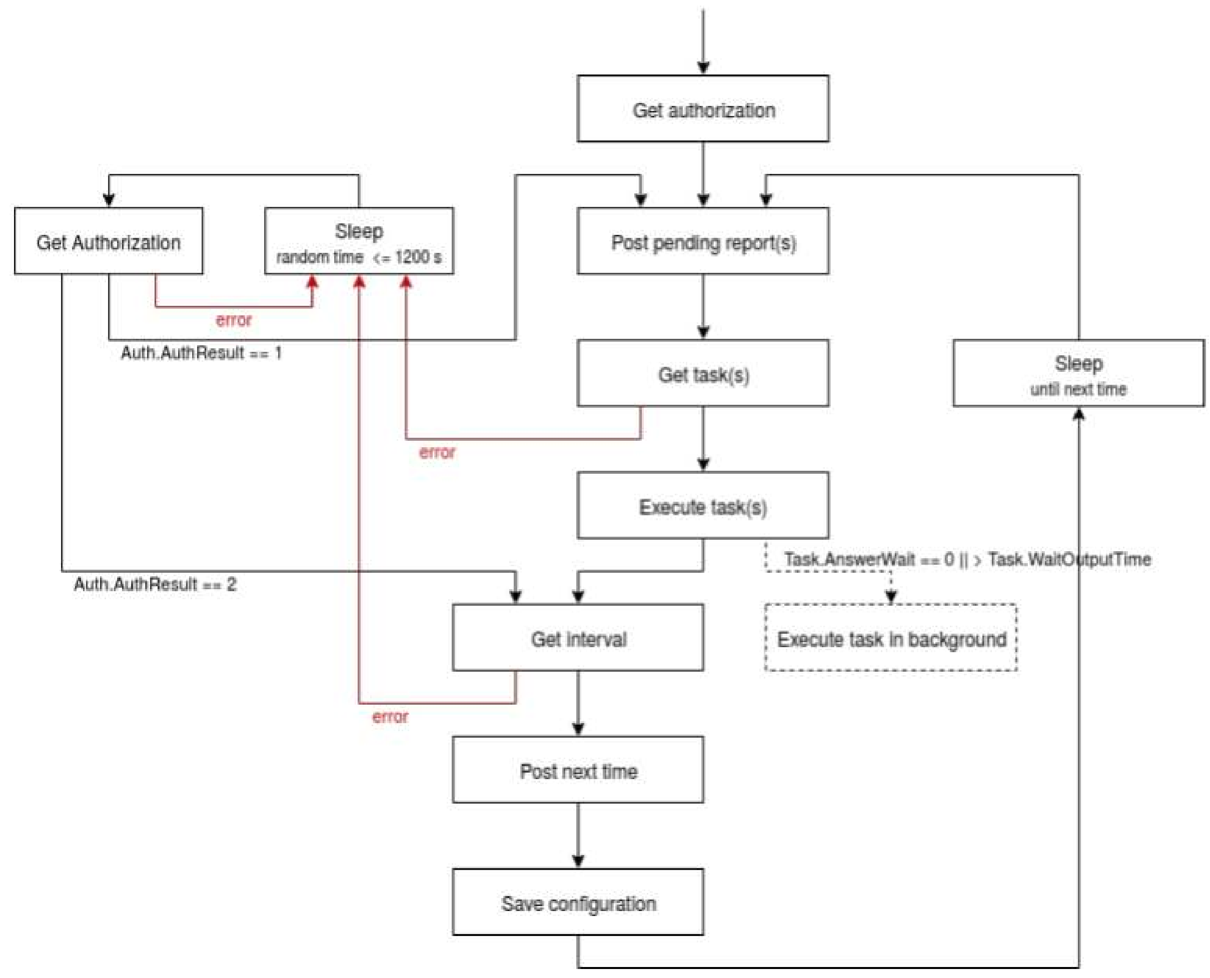
Initial Access	to get into your network.
Execution	to run malicious code
Persistence	to maintain a foothold in your network
Privilege Escalation	to gain higher-level permissions
Defense Evasion	to avoid being detected
Discovery	to figure out your environment
Lateral Movement	to move through your environment
Collection	to gather data of interest to their goal
Command and Control	to communicate with compromised systems to control them
Impact	to manipulate, interrupt, or destroy your systems and data
Inhibit Response Function	
Impair Process Control	

Persistence

An important tactics, attackers are interested in persisting years into a system

Exaramel is a backdoor an attacker uses to remote control a system after it has been successfully attacked

It receives a command, execute it, sleeps and then receive another one …



Exaramel Persistence

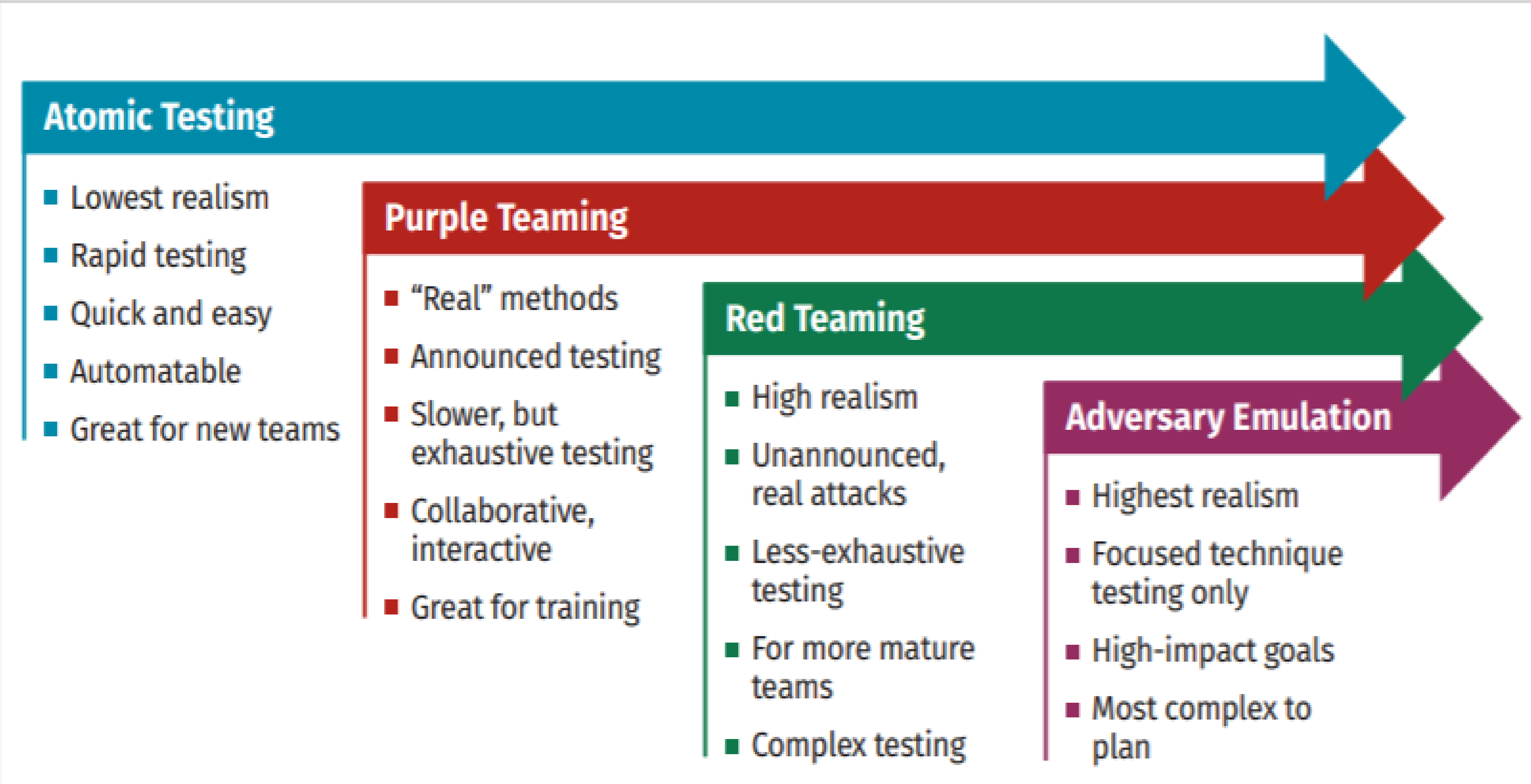
not run by root : command from C&C center

- Installation: add two entries to the user crontab, one that restarts Exaramel every minute and another one that starts Exaramel at the system start (@reboot).
- Deletion: delete every entries of the user's crontab.
- Check: searche for $EXARAMEL_PATH in the crontab entries of the user

run by root and the startup system is systemd

- Installation: create the file /etc/systemd/system/syslogd.service, enables the new unit (systemctl enable syslogd.service) and reloads the systemd manager configuration (systemctl daemon-reload). At this time, the new unit is not active, it will be restarted the next time system reboot.
- Deletion disable the unit then deletes the file /etc/ systemd/system/syslogd.service, reloads several time the systemd manager daemon and stops the. Every crontab entries of the root user are also deleted.
- Check: Exaramel tests if the file /etc/systemd/system/syslogd.service exists

Adversary and Matrix



Att&ck Matrix: A de facto standard

FIREEYE Products Mandiant Solutions Customers Partr

Threat Research

New SUNSHUTTLE Second-Stage Backdoor Uncovered Targeting U.S.-Based Entity; Possible Connection to UNC2452

March 04, 2021 | by Lindsay Smith, Jonathan Leathery, Ben Read

MALWARE UNC INTELLIGENCE BACKDOOR

Executive Summary

- In August 2020, a U.S.-based entity uploaded a new backdoor that we have named SUNSHUTTLE to public malware repository.
- SUNSHUTTLE is a second-stage backdoor written in GoLang that features some detection evasion capabilities.
- Mandiant observed SUNSHUTTLE at a victim compromised by UNC2452, and have indications that it linked to UNC2452, but we have not fully verified this connection.
- Please see the Technical Annex for relevant MITRE ATT&CK techniques (T1027, T1027.002, T1059.003

Appendix: MITRE ATT&CK Framework

Technique	Description
T1027	Obfuscated Files or Information
T1027.002	Software Packing
T1059.003	Windows Command Shell
T1071.001	Web Protocols
T1105	Ingress Tool Transfer
T1140	Deobfuscate/Decode Files or Information
T1573.001	Symmetric Cryptography

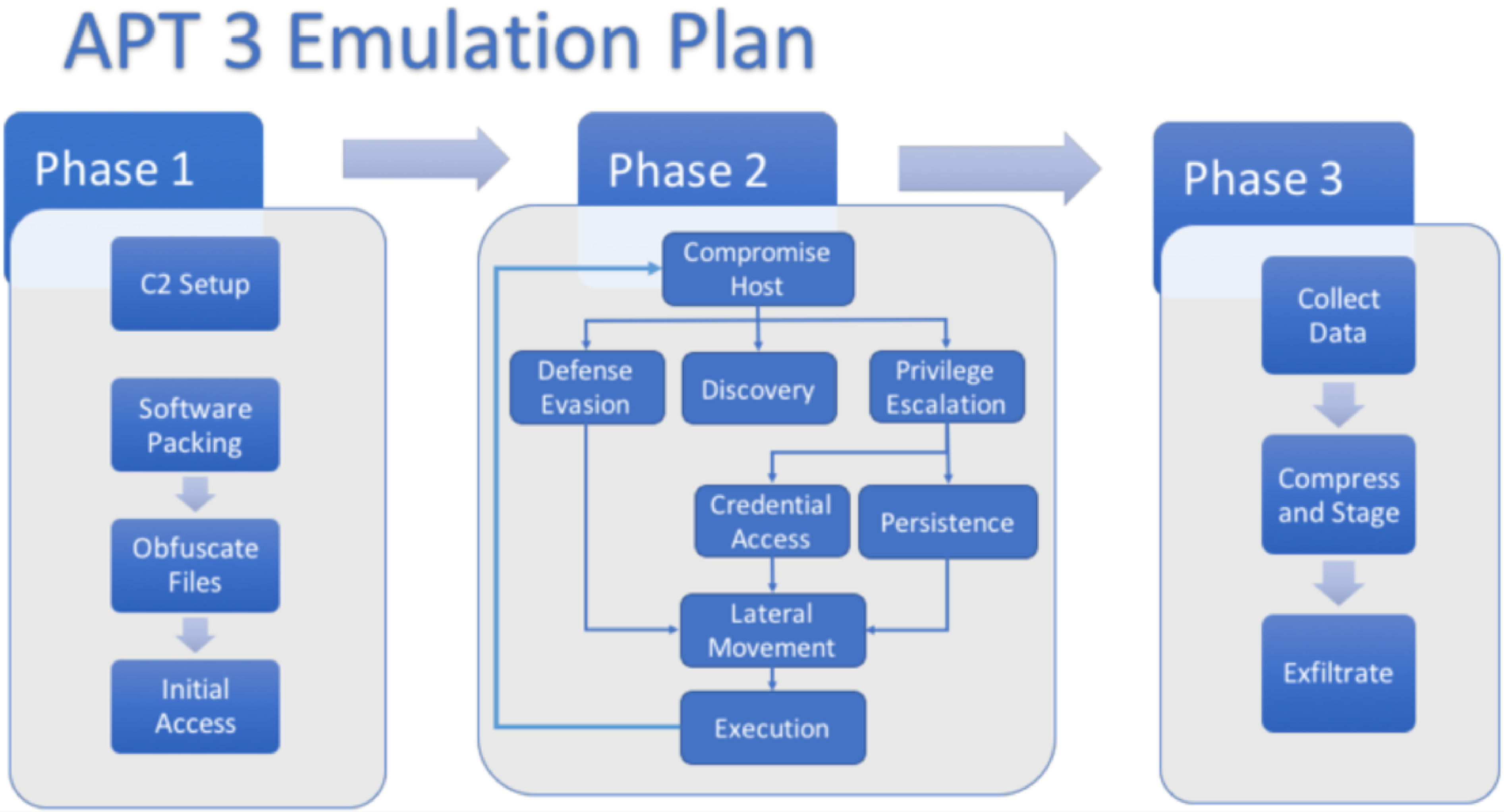
Att&ck Matrix: emulation plans

- These are prototype documents of what can be done with publicly available threat reports and ATT&CK.
- This activity aims to enable the modeling of adversary behavior, as described by ATT&CK, to allow defenders to more effectively test their networks and defenses
- This is part of a larger process to help more effectively test products and environments, as well as create analytics for ATT&CK behaviors rather than detecting a specific indicator of compromise (IOC) or specific tool.

PROBLEM

> Att&ck Matrix focuses on intrusion detection/discovery hence it neglects the attacker strategy to focus on the attacker TTPs because their execution order is unessential (you can detect actions but not strategy)

Att&ck Matrix: emulation plans



Strategies based on Att&ck Matrix

(ANALYSIS OF AUTOMATED ADVERSARY EMULATION TECHNIQUES, *SummerSim-SCSC, 2017 July 9-12, WA, USA***)**

- A simulation environment (a target environment) with three main components:
  - the *objects* in the environment,
  - the *properties* between the objects,
  - the *settings* that control objects and properties.
- Objects
  - hosts,
  - domains,
  - accounts
  - vulnerabilities
- Properties: Static do not change, Dynamic may change

connected(*X*,*Y*)	Static	Network traffic between hosts *X* and *Y* is allowed.
localAdmin(*A*,*X*)	Static	Account *A* is a local administrator on host *X*.
domainAdmin(*A*,*D*)	Static	Account *A* is an administrator for domain *D*.
remote(*A*,*X*)	Static	Account *A* is authorized for remote login on host *X*.
inDomain(*X*,*D*)	Static	Host *X* is in Windows domain *D*.
vulnerable(*X*,*E*)	Static	Host *X* is vulnerable to exploit *E*.
activeCreds(*A*,*X*)	Dynamic	Host X stores the credentials for account *A*
activeConnection(*X*,*Y*)	Dynamic	An active network connection between hosts *X* and *Y* .

Strategies based on Att&ck Matrix

- A simulation environment with three main components:
  - the *objects* in the environment,
  - the *properties* between the objects,
  - the *settings* that control objects and properties.
- **Settings** Rules to generate a world with the objects and properties listed above.

  Each setting can be classified in one of three ways:
  - *fixed :* fixed network parameters e.g., number of personal/shared workstations/domains,
  - *perobject maximum:* specifies the maximum value for an object property e.g., maximum number of domains a host can be in, of admins for a shared workstation By default, personal workstations have exactly one local administrator and are members of exactly one domain; shared workstations typically have more.
  - *Probabilistic*: the likelihood a randomly selected pair will have a property
    - the probability a host will have a vulnerability,
    - the probability two hosts are connected.

Strategies based on Att&ck Matrix

- Each simulation is run as a *game* in a world built according the previous description. There are three participants:
  - a gray agent = user activity,
  - a defender = its only ability is to detect the attacker
  - an attacker.
- Initially the attacker a low-privilege foothold somewhere on the network.
- Then, the simulation iterates for a given number of times
  - the attacker makes a move,
  - the gray agent makes a batch of moves,
  - the defender makes a move
- The attacker has *imperfect information* regarding its state in the network. At any time, the attacker view of the network is limited to what it can see.
- At the beginning of a simulation, all the attacker knows is that it has compromise a single host, (= the goal) with no knowledge regarding other hosts or user accounts.