

Data analytics for Cyber security

-Types of Cyberattacks-

Vandana P. Janeja

©2022 Janeja. All rights reserved.



Outline

Types of Attacks

Example: Social Engineering

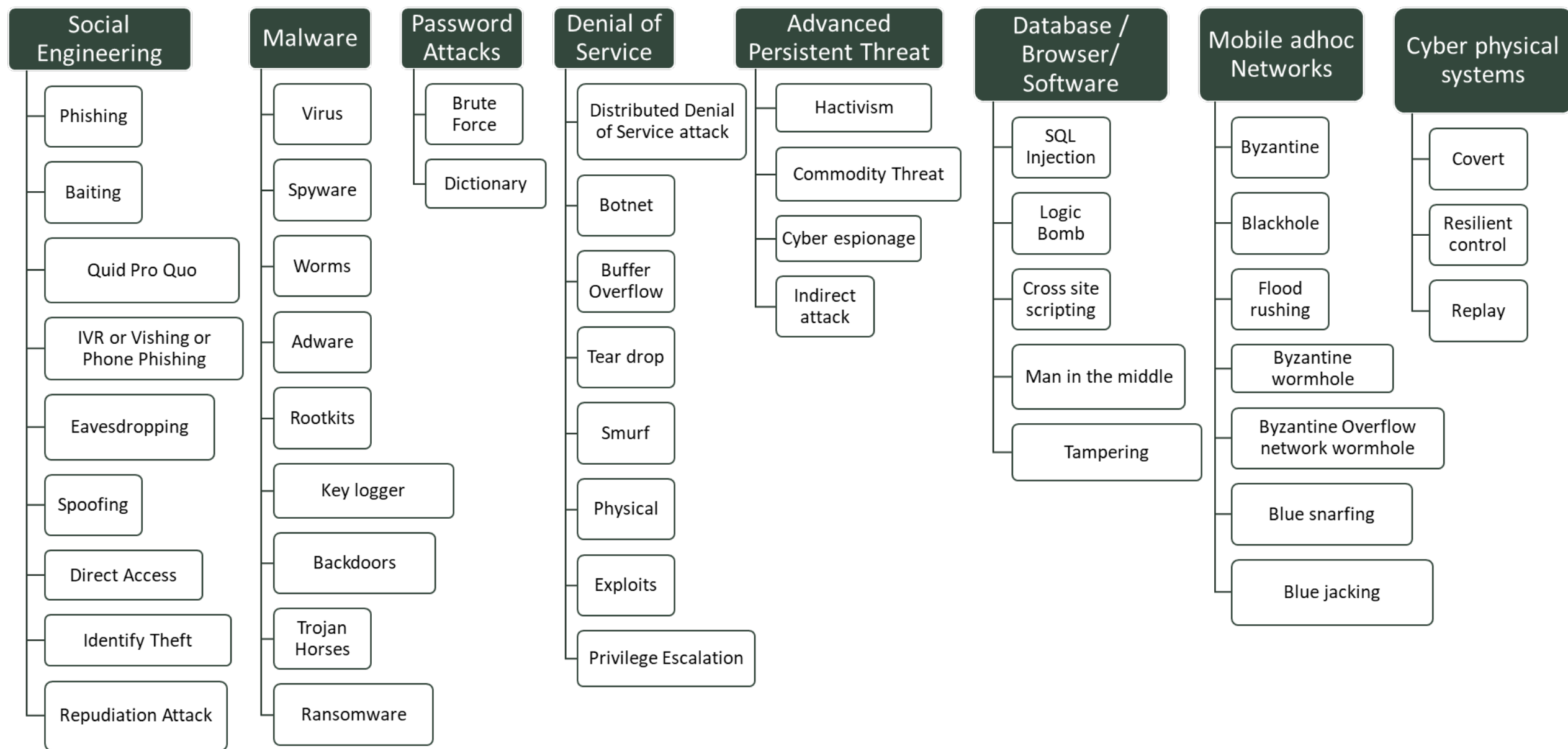
- Computational Data Model for Social Engineering

Example: Advanced Persistent Threat (APT)

- Data Analytics Methods for Persistent Threats

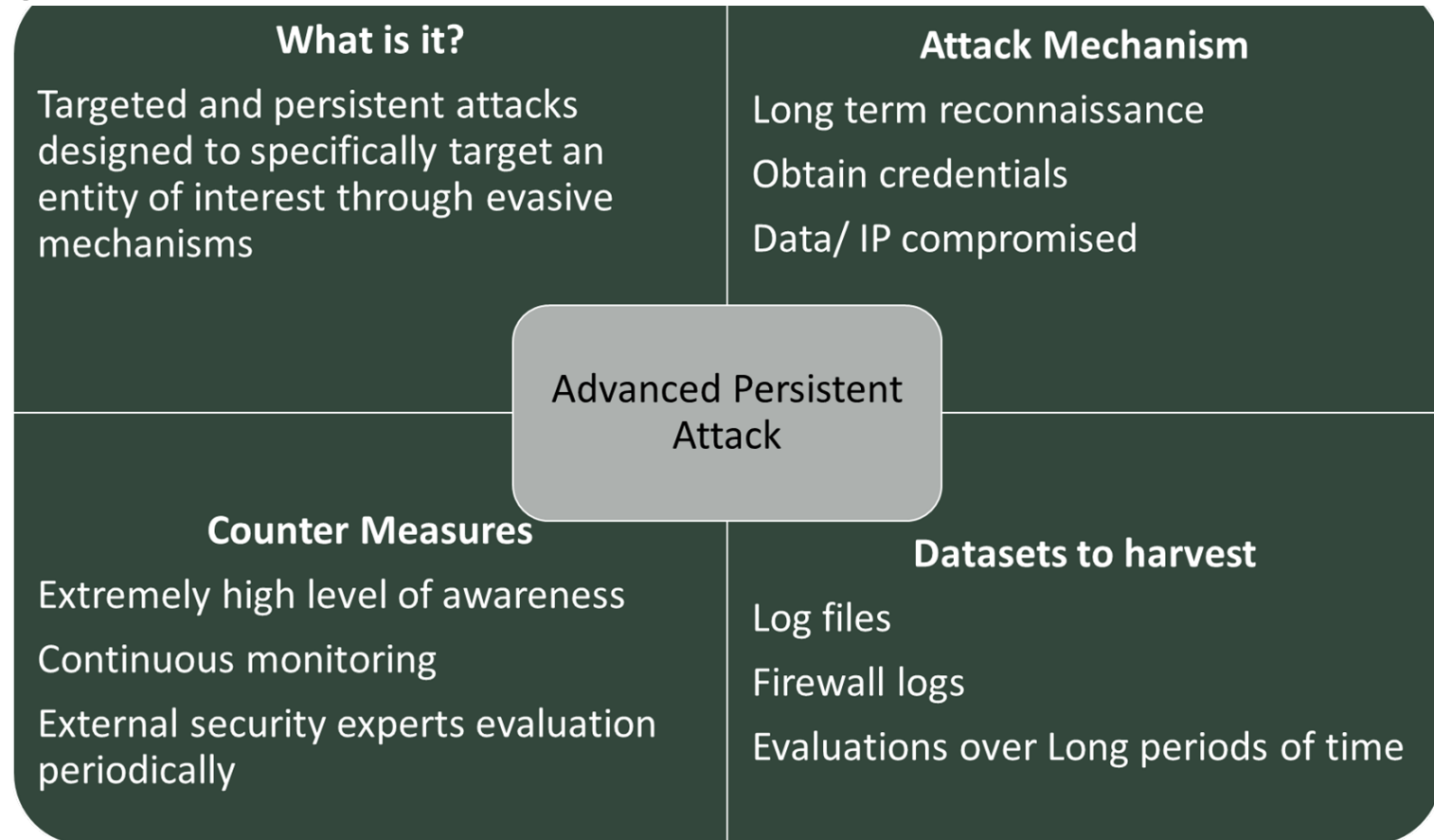
Types of Attacks

- Threat taxonomies : ENISA threat taxonomy (ENISA 16), Cyber physical incident catalog (Miller 2014), AVOIDIT cyber attack taxonomy (Simmons 2009), (Quader et. al 2022)
- Examples:
 - Social engineering: relies on understanding the social interactions of the individual and trying to gain access to user credentials. Through phishing an individual may be motivated to provide their user name and password which will allow a hacker to gain access to a system
 - Phone Phishing: phishing through a phone system where the user is coaxed to provide their credentials
 - Malware : class of attacks which install a malware such as virus, spyware, Trojans etc., on a user's system with the intent of gaining access and information from a user's system.
 - Database or software based attacks: caused due to bugs in the software such as in the case of SQL injection attack where the front end user form can be used to initiate a query request to the back end database to provide the user name and password credentials.
 - Mobile adhoc attacks: caused where the information flow is disrupted such as in the case of the blackhole attack where the network traffic is redirected and may be lost.
 - Replay attack: valid data is sent maliciously but repeatedly with the intent to cause delay or block the traffic.



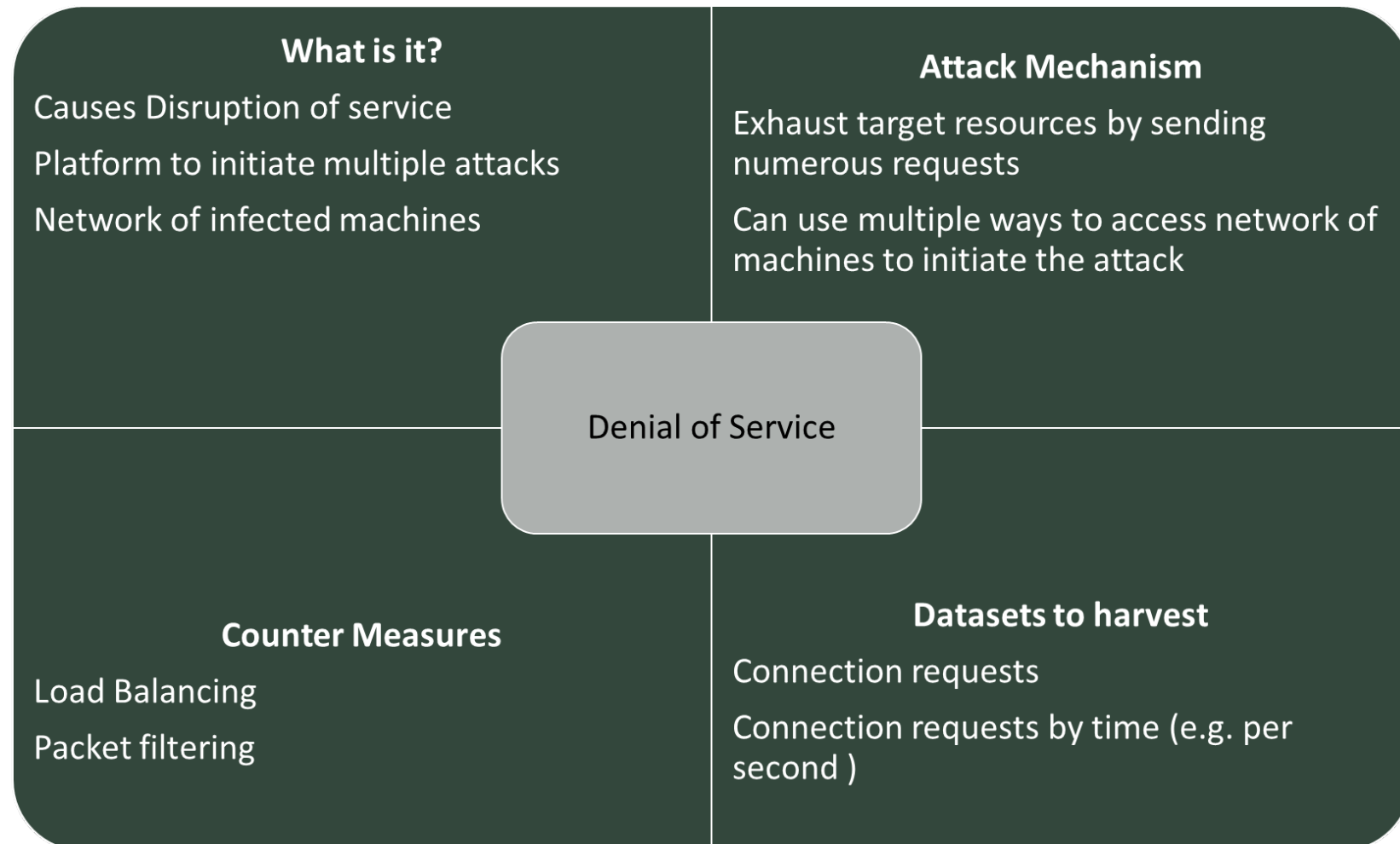
Advanced Persistent Threat

- Persisted through coordinated long term attacks and are mostly originated from organizations with long term interest in the assets on the hacked system
- These could be initiated from nation states
- Generally carried out by exploiting vulnerability or by gaining access to a system through a social engineering attack
- Mostly spear phishing is the root cause of majority of other types of attacks
- Security is more than an engineering challenge as people are essential part of the critical infrastructure
- Understanding and addressing human behavior is essential to building a security culture



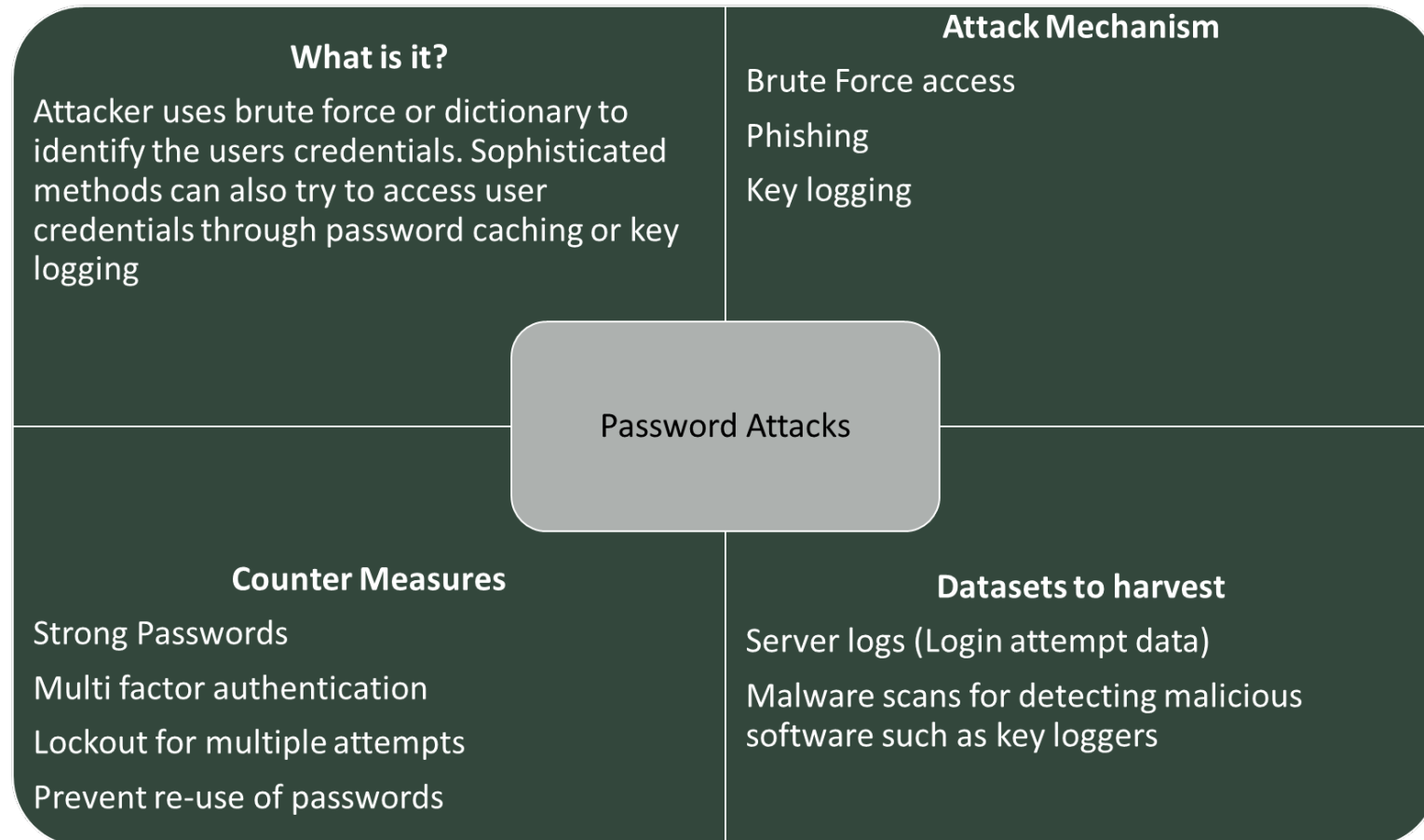
Denial of Service

- Denial of Service (DoS) attacks are caused when the malicious users tend to block legitimate traffic by sending too many requests to a server
- Distributed versions of these attacks cause a coordinated attack to take place such as in the case of a botnet where several machines are carrying out these requests being sent



Password Attacks

- Focus on getting user credentials either through a brute force attack or by checking the passwords against a dictionary of words
- Hacker trying out different passwords based on a set of passwords such as from a dictionary
- To counter such attacks complex passwords and passwords which are not repeated from one site to the next are recommended



Example: Social Engineering

- A mechanism to gain access to computer systems and the information that resides there in an unauthorized manner through human or computer mediated methods targeting the social aspects of a user's data
- Compromising information about a user's credentials can be obtained through social connections or impersonating social connections
- An example of impersonating social connection is spear phishing where an email that appears to be from a legitimate social connection or authority figure is requesting credentials
- The goal of social engineering is to obtain information that will allow the hacker to gain unauthorized access, via human mediated or computer mediated methods, to a system in order to compromise the systems integrity and potentially leading other types of attacks
- Human mediated methods may include impersonation, impersonating helpdesk, third-party authorization, phishing through impersonating trusted authority figure, snail mail to name a few
- The question addressed here is how do we detect common social engineering attacks in the network using computational models?

Computational Data Model for Social Engineering

- Computational data models are derived on intuitions of how these attacks affect the patterns in the data collected on the network
- In order to build a computational data model for social engineering attacks, specifically password theft, log data from Intrusion detection systems such as Snort log data can be analyzed
- The objective is to derive a data model by which we can detect an anomaly in the network which emulates the behavior of a password theft
- This model can be applied to any Snort IDS log data or even firewall or server log data to identify potential password theft attack

Computational Data Model for Social Engineering - Intuition

- If a password is stolen the intent could be to generate more attacks inside the network through this compromised system
- If the intent is data theft then several datasets might be moved from this compromised system
- In either case the traffic would substantially increase as compared to before the password was stolen
- To distinguish such a change in data pattern of the user we can differentiate this from other traditional attacks by eliminating any probes conducted on this system, since traditional hacks into a system are done after a probe and scan in the network
- The **intuition** is that if there is no probe or scan in the network and the system under study and the network pattern has substantially changed over time then it leads to an observation that this may potentially be a password theft
- An exploratory analysis as a human network administrator would have to evaluate further whether the identified system indeed had a password theft or there is another event signifying change in the traffic patterns
- These types of intuitions allow the administrator to focus on a narrower swath of data rather than very large network traffic at large
- **Condition 1:** Password theft in this study of social engineering attack is a non-probing scenario where the intruder may gain access to the password via non-probing means such as getting hold of password through human mediated methods such as impersonation as an IT security personnel.

Computational Model

Social Engineering: Password theft

Model: Computational Data model to identify password theft attack

Conditions:

Condition 1: Non-probe Condition in log data

- Count of compromised destination node IP (n_1) is less than threshold (ρ)
- \neg Probe

Count (destination (n_1)) $< \rho$

$$\left. \begin{array}{l} n_2 \rightarrow n_1 \\ n_3 \rightarrow n_1 \\ n_4 \rightarrow n_1 \end{array} \right\} < \rho$$

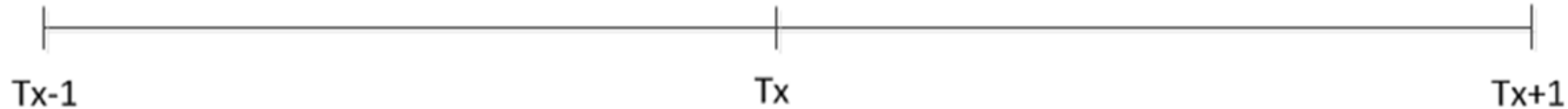
Computational Data Model for Social Engineering - Intuition

- **Condition 2** is about significant activity increase on the same machine (IP) as a result of having access to that machine from the intruder via password theft
- Once the target machine becomes a victim, the same machine is used as a source (IP) to potentially attack other machines or send out data
- The activities on the source IP significantly increases as a result of password theft.
- In this model, the behavior of the destination IP at time T_{x-1} when this IP has normal behavior is compared to the behavior of the same IP as a source at time T_x when the IP becomes a victim of password compromise.
- ρ is the threshold below which a particular destination is identified as having a non-probing scenario
- δ is the threshold below which the source IP depicts normal activities while the value above the threshold would identify the source to be sending out large amount of communication to other destinations (for example, data or attacks).
- Given a set of IP addresses $N=\{n_1, \dots, n_n\}$, a compromised IP n_1 is source and accessing destination n_i more than a certain threshold δ .

Computational Model

Condition 2: Significant increase in activity from the compromised source node after potential password theft

Time \longrightarrow



$\text{Count}(\text{source}(n_1), T_{x-1}) < \delta$ and $\text{Count}(\text{source}(n_1), T_x) > \delta$

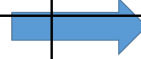
$\left. \begin{array}{l} n_1 \rightarrow n_2 \\ n_1 \rightarrow n_3 \\ n_1 \rightarrow n_4 \end{array} \right\} > \delta$


Computational Data Model for Social Engineering - Computing thresholds

- Data from log files can be divided into several bins based on the time stamp and for each bin the count for each destination IP can be computed and sorted in ascending order
- The Mean and Standard Deviation of the counts can be computed
- Mean can be considered as the threshold value for ρ and δ , alternatively mean + standard deviation can also be used as a slightly restrictive threshold
- Example data is shown for both time bins and a $\mu + \sigma_1$ and $\mu + \sigma_2$ thresholds have been computed to indicate an anomalous communication at time T_x for ip n5
- In general, if subsequent bins show low values of threshold followed by high values of the counts as compared to thresholds then this may indicate a period worth investigating

Computational Data Model for Social Engineering- Example

- Notice, the total count for source IP n1 before the password theft on $T_x - 1$ is 2, which is less than our threshold of 2.28 ($\mu + 2\sigma$). The total count of the source n1 goes up significantly after the password is potentially compromised on T_x , which is 5. This count is greater than the threshold of 4.9 ($\mu + 2\sigma$)
- As a preprocessing task thresholds can be identified from historic datasets. Overall interesting conditions are when $\text{Count}(\text{destination}(n_1), T_{x-1}) < \rho$ and $\text{Count}(\text{source}(n_1), T_x) > \delta$
- Before the password theft occurs, the count (activity) for that IP (as a destination) is going to be lot less at time T_{x-1} and the count (activity) for that same IP (as source) after the password theft is going to be a significantly high at T_x time
- Discovering frequent rules
 - Association rule mining can be used to validate the rules where the activities of the compromised IP increases significantly with a high Lift value
 - Association rule mining provides the rules where the compromised source IP targets many destination IP's once the intruder gains access to the source machine via password theft
 - This source IP from ARM rules should match up with results from the computational data model and show this IP attacking many targets (n_2, n_3, n_4) as a result it should show up in the Association rules
 - The Lift value gives us the performance measure and the confidence for the ARM rules gathered from the algorithm

Condition1	Source	Destination		Destination	Count
Time $T_x - 1$	n4	n2		n1	2
	n1	n6		n2	2
	n2	n1		n3	1
	n3	n1		n4	1
	n4	n2		n5	1
	n5	n3		n6	1
	n6	n4			
	n2	n5		μ	1.33
				σ	0.471405
			ρ	1σ	1.80
			ρ	2σ	2.28

Condition2	Source	Destination		Source	Count
Time T_x	n1	n2		n1	5
	n1	n3		n2	1
	n1	n4		n3	2
	n1	n5		n4	0
	n2	n6		n5	0
	n3	n2		n6	0
	n1	n6			
	n3	n4		μ	1.33
	n2			σ	1.795055
			δ	1σ	3.13
			δ	2σ	4.92

Example: Advanced Persistent Threat (APT)

- One of the fundamental goals of an Advanced Persistent Threat (APT) is to remain undetected on the network for a long period of time in order to carry out their mission
- The desire to remain hidden, and the sophisticated measures taken to avoid detection, makes it extremely difficult to identify, analyze, and categorize APTs
- **Difficult to detect:** Use of custom tools that do not have a signature for detection are very challenging to discover. Zero-day exploits avoid detection by signature-based systems. Legitimate third-party applications leveraged for command and control hide in plain sight. Evidence that data has been transferred from the system is deleted or overwritten.
- **Difficult to prevent intrusion:** Attackers target the weakest link that is the users of the systems, via socially engineered spear-phishing emails. Up-to-date systems are still vulnerable to zero-day exploits. Attackers have been known to target additional, outside organizations in order to access their true intended victim
- **Difficult to remove:** Attackers install backdoors throughout victim networks to maintain footholds and hence they are very difficult to remove.



APT Charechtersitics

Attack vectors through which they are initiated

Exploits through which they are carried out

Tools used to establish the threat

Targets that make it inviting for an APT attack

Command and control, and persistence of the attack which is longer lasting than other traditional attacks

APT

I. Attack Vector

- APTs typically follow three primary attack vectors for gaining access to a target's system
- No vector is necessarily more sophisticated than any other; however, the level of sophistication is inherent in the quality and thoroughness of the social engineering and spear-phishing techniques
- Several APTs have broadened their potential victim base and use an attack vector known as a “*watering hole*”
- Attackers first identify a vulnerable website and insert malicious code into the site that will redirect visitors to another malicious site hosting the APT's malware
- *SQL injection/Other*: This is probably the least utilized vector by APTs. Attackers may use SQL injection to exploit a company's web servers and then proceed with the attack

II. Exploits

- Before an attacker can surreptitiously install their chosen backdoor on a victim machine, vulnerability needs to be available to exploit. Vulnerabilities can be found in many software applications.
- *Zero day*:. A zero day vulnerability is not known by the software developer or security community; it is typically only known by the attacker or a small group of individuals. A zero day is valuable in that any victim machine running that software will be vulnerable to the exploit
- *Known, but unpatched vulnerabilities*: A small window of time exists between the discovery of vulnerability and the time it takes to develop and release a patch. An attacker may take advantage of this time to develop an exploit and use it maliciously
- *Known, patched vulnerabilities*: Since software developers have to release updates to patch any vulnerability, an opportunity exists for attackers to take advantage of users who do not regularly update their software

III. Tools

- Many attackers use similar, publicly available tools to navigate a victim network once they've established backdoors on victim systems. The difference between APTs lies in the tools they use to maintain access to victims
- *Custom tools*: Some APTs use custom tools that are not publicly available. Since a considerable amount of time, expertise, and money is needed to develop a family of malware, custom tools will highlight the potential expertise and backing of a group
- *Publicly available tools*: Many APTs use tools that are publicly available and easily downloadable from many websites dedicated to hacking



APT

IV. Targets

- APT targets can be Government/Military, Non-governmental Organization, Commercial Enterprises or Defense Industrial Base.

V. Command and Control - ways in which an attacker communicates with implanted machines:

- *Trojan-initiated call backs*: This type of communication is marked by the malicious program initiating the connection to an external server. Most attackers use this technique in order to defeat firewalls, which typically allow outbound connections from computers within the network
- *Encryption*: The usage of encryption to hide C2 communications and make it more difficult to detect.
- *Third Party Applications*: Leveraging third party applications, like MSN Messenger, to communicate with victims. Using trusted applications makes it more difficult for network defenders to identify C2 communications
- *Standard/Default Malware Connections*: Many publicly available backdoors come with default settings. For example, Poison Ivy uses port 3460 by default. Communications over default settings are more easily detected because they are known by network defenders

VI. Persistence

- Persistence is characterized by the length of time a particular APT has been conducting operations such as more than five years, between two and five years or less than two years

APT Characteristics



APT charechteristics

Name	Attack Vector	Exploit	Tools	Targets	Command And Control	Persistence
APT1	Spear Phishing		<i>Custom</i>	Commercial Enterprises, Defense Industrial Base	<i>Trojan-initiated call backs, Third Party Applications</i>	More than five years
Shady Rat	Spear Phishing	<i>Known patched vulnerabilities</i>		Government/Military, Non-governmental Organization , Commercial Enterprises, Defense Industrial Base		Less than two years
Aurora / Elderwood	Spear Phishing <i>Watering hole</i>	<i>Zero day</i>	<i>Custom</i>	Non-governmental Organization, Commercial Enterprises, Defense Industrial Base	<i>Trojan-initiated call backs, Encryption</i>	Between two and five years
RSA Hack	Spear Phishing	<i>Zero day</i>	<i>Publicly available</i>	Commercial Enterprises	<i>Trojan-initiated call backs, Standard/Default Malware Connections</i>	Less than two years
Lurid	Spear Phishing	<i>Known but unpatched vulnerabilities</i>	<i>Publicly available</i>	Government/Military, Commercial Enterprises, Defense Industrial Base	<i>Trojan-initiated call backs</i>	
Night Dragon	Spear Phishing <i>SQL injection/Other</i>	<i>Known patched vulnerabilities</i>	<i>Publicly available</i>	Commercial Enterprises	<i>Trojan-initiated call backs</i>	Between two and five years
Ghost Net	Spear Phishing	<i>Known patched vulnerabilities</i>	<i>Publicly available</i>	Government/Military, Non-governmental Organization		Less than two years
Sykipot	Spear Phishing <i>Watering hole</i>	<i>Known, but unpatched vulnerabilities</i>	<i>Custom</i>	Government/Military, Defense Industrial Base	<i>Trojan-initiated call backs, Encryption</i>	More than five years
Nitro	Spear Phishing <i>Watering hole</i>		<i>Publicly available</i>	Non-governmental Organization, Commercial Enterprises	<i>Trojan-initiated call backs, Encryption</i>	Less than two years

Data Analytics Methods for Persistent Threats

- How data analytics can help to discover persistent threat patterns such as those exhibited in an APT in the network data.
- Evaluate the Intrusion detection log data along with other raw log data files and study it over time using data discretization
- Once the temporal bins are created frequent patterns using the Association Rule Mining can be identified and the overlapping and non-overlapping rules across these bins can be determined
- The high priority unusual persistent threats can be isolated, that is the rules which are frequently occurring but are non-overlapping across time periods

Data Analytics Methods for Persistent Threats - Intuition

- An advanced persistent threat is generally unusual and uses stealth mechanisms to not be discoverable
- APT's also exhibit signs of system access at unusual times, high level of backdoor access and unusual information exchange
- These events can be unusual yet frequent
- This unusualness comes from the times when these events happen or in some cases the specific systems being targeted
- Association rules that are repeatedly occurring across the multiple time periods represent the daily chatter in the log files
- Rules which are frequent but are non-overlapping across time periods can be deemed unusual and considered as potential persistent threats
- For example, frequent logins at unusual times. Keeping this intuition in mind let us consider the mining strategy.

Data Analytics Methods for Persistent Threats - Data discretization

- Segregates data into time segment bins representing the multiple time periods, which makes it easier to evaluate the persistence of the threats across time periods represented by bins
- *Given a set of temporally ordered alerts $A=\{a_1, \dots, a_n\}$ (from the IDS log files such as from SNORT logs) where each a_i has a set of features $f_i=\{f_{i1}, f_{i2}, \dots, f_{im}\}$, Bin is a set of temporally ordered segments from A such that $B = \{b_1=(a_1, \dots, a_x), b_2=(a_{x+1}, \dots, a_y), \dots, b_r=(a_{i+1}, \dots, a_n)\}$ where $|b_1| = |b_2| = \dots = |b_z| = z$ where z is the size of the bin*
- *Consider equal sized bins in this case where z is fixed frequency.*
- Network traffic, which is suspicious, is captured through Intrusion detection System (IDS) in an alert log capturing the source, destination IP with timestamps and the priority
- Here priority indicates the severity of an alert (high, medium or low level alert)
- The high priority threats can be obvious suspicious activities and the low priority threats can be potential alerts that may or may not be real
- Such high priority alerts when looked at in combination over a long period of time may turn out to be persistent
- In general, an IDS does not capture threats over a period of time but views each individual alert and assigns priority
- It is important to evaluate the alert priorities to see, if combined over time, what the overall threat is.

Data Analytics Methods for Persistent Threats - Frequent Patterns

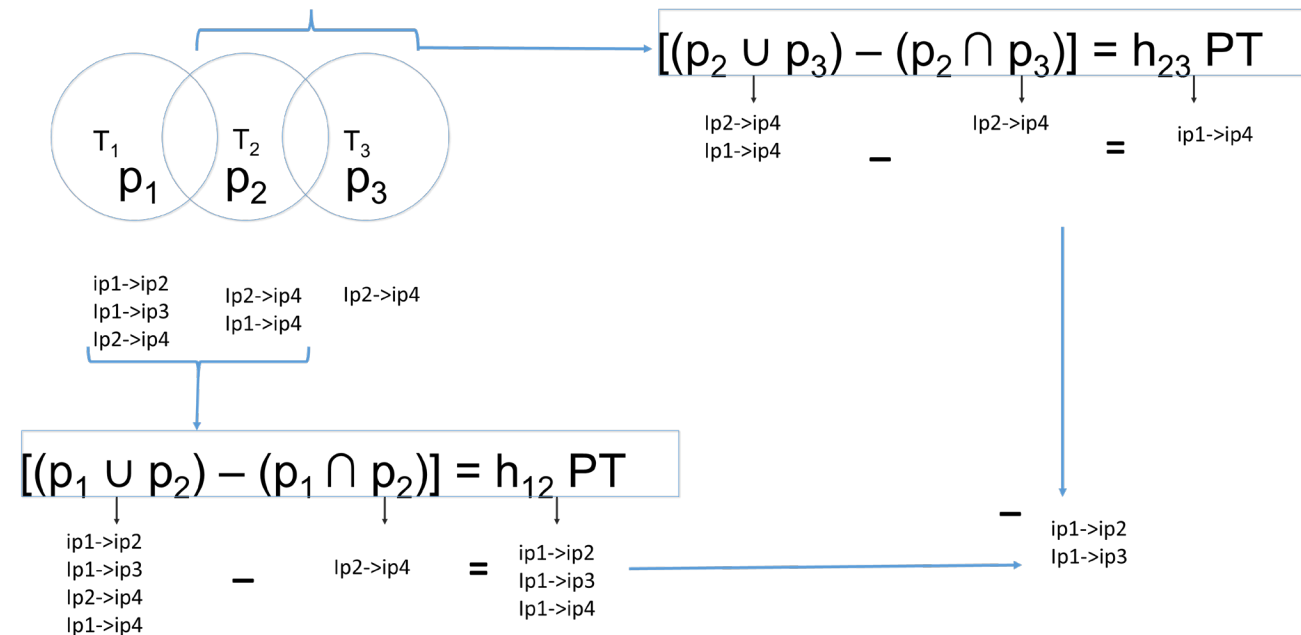
- Analyzing threats individually may not provide the overall persistence of a threat
- It is important to study the data from the overall perspective and identify frequent patterns on a given timeframe to discover persistent threats
- Thus, the data can be mined using Association Rule Mining techniques and isolate the Persistent High Priority Threats (PHPT) from the individual high priority threats.
- The PHPT are consistent and may indicate APT behaviors
- Now given a set of discrete bins B in the log data, frequent pattern set $P = \{p_1, \dots, p_s\}$ is a set of association rules where each $p_i = \{f_{ij} \rightarrow f_{ik}, f_{il} \rightarrow f_{im} \dots, f_{ip} \rightarrow f_{iq}\}$ such that each rule set p_i corresponds to each bin b_i and where $f_i = \{f_{i1}, f_{i2} \dots f_{im}\}$ are features of alert a_i in b_i

Data Analytics Methods for Persistent Threats - Persistent Threats

- Persistent threats are high priority, unusual threats that stay consistent over a long period of time
- The binned datasets and their corresponding frequent patterns can be intersected with each other to isolate the non-intersecting high priority threats to detect the potential persistent threats
- Persistent threat pattern have the following key characteristics:
 - Consistent over time (*occurs repeatedly*)
 - Single source (*same key players*)
 - Individually each is a threat
 - Unusual (*pattern may be repeated at an unusual time of the day or single source accessing different sources repeatedly at the same time of the day*)
 - Non-obvious (*non-overlapping*)

Data Analytics Methods for Persistent Threats - Persistent Threats

- The PT patterns are located in the non-overlapping area.
- Each circle in the figure represents the association rules for a specific bin
- If the association rules overlap they are consistent across bins and may be more like probing which is not consistent with the characteristics of an APT
- If the association rules do not overlap across bins and are unique to certain time periods then they can be considered them as potential persistent patterns of interest
- Three time periods T_1 , T_2 and T_3 with their corresponding rule sets p_1 , p_2 and p_3 .
- p_1 has associations in the form of IP1-> IP2
- This could imply a source destination relationship where IP1 is sending data to IP2
- A Union of two consecutive sets is taken and the intersection is subtracted from this set, as a result the non-overlapping or non-intersecting part of the two sets is discovered.
- After subsequent operations the most unusual frequent set is identified which is non-intersecting across multiple bins.



References

- ENISA, Threat Taxonomy A tool for structuring threat information, European Union Agency For Network And Information Security January 2016,
- Miller, W. B. (2014). *Classifying and Cataloging Cyber-Security Incidents Within Cyber-Physical Systems* (Doctoral dissertation, Brigham Young University).
- Simmons, C., Ellis, C., Shiva, S., Dasgupta, D., & Wu, Q. (2009). AVOIDIT: A cyber attack taxonomy.
- Yang, Sang-Chin, Wang, Yi-Lu. System Dynamics Based Insider Threat Modeling. 2011. International Journal of Network Security & Its Applications (IJNSA). Vol. 3, No. 3, May 2011
- Trend Micro Incorporate. Detecting APT Activity with Network Traffic Analysis. 2012. Available from <http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-detecting-apt-activity-with-network-traffic-analysis.pdf>.
- Binde E. Beth, McRee Russ, O'Connor Terrance. Assessing Outbound Traffic to Uncover Advanced Persistent Threat . May 2011. Available from <http://www.sans.edu/student-files/projects/JWP-Binde-McRee-OConnor.pdf>.
- Blasco, J. (2013, Mar 21). New Sykipot Developments. Alient Vault Labs. Retrieved from <http://labs.alienvault.com/labs/index.php/2013/new-sykipot-developments>.
- Leyden, J. (2012, Mar 29). NSA's top spook blames China for RSA Hack. The Register. Retrieved from http://www.theregister.co.uk/2012/03/29/nsa_blames_china_rsa_hack/.
- O'Gorman, G. & McDonald, G. (2012). The Elderwood Project. Symantec. Retrieved from http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-elderwood-project.pdf.
- The SecDev Group. (2009). Tracking GhostNet: Investigating a Cyber Espionage Network. Retrieved from <http://www.nartv.org/mirror/ghostnet.pdf>.
- Websense. Advanced Persistent Threat and Advanced Attacks: Threat Analysis and Defense Strategies for SMB, Mid-Size, and Enterprise Organizations Rev 2. 2011. Available from <http://www.websense.com/assets/white-papers/whitepaper-websense-advanced-persistent-threats-and-other-advanced-attacks-en.pdf>.
- Alperovitch, D. (2011). Revealed: Operation Shady RAT. McAfee. Retrieved from <http://www.mcafee.com/us/resources/white-papers/wp-operation-shady-rat.pdf>.
- Blasco, J. (2013, Mar 21). New Sykipot Developments. Alient Vault Labs. Retrieved from <http://labs.alienvault.com/labs/index.php/2013/new-sykipot-developments>.
- Mandiant. APT1: Exposing One of China's Cyber Espionage Units. 2013. Available from http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf.
- McAfee Labs. (2011). Global Energy Cyberattacks: "Night Dragon". Retrieved from <http://www.mcafee.com/us/resources/white-papers/wp-global-energy-cyberattacks-night-dragon.pdf>.
- McAfee Labs (2010). Protecting Your Critical Assets. Retrieved from <http://www.mcafee.com/us/resources/white-papers/wp-protecting-critical-assets.pdf>.
- O'Gorman, G. & Chien, E. (2011). The Nitro Attacks. The Nitro Attacks. Symantec. Retrieved from http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the_nitro_attacks.pdf.
- The SecDev Group. (2009). Tracking GhostNet: Investigating a Cyber Espionage Network. Retrieved from <http://www.nartv.org/mirror/ghostnet.pdf>.
- Thakur, V. (2011, Dec 8). The Sykipot Attacks. Symantec. Retrieved from <http://www.symantec.com/connect/blogs/sykipot-attacks>.
- Villeneuve, N. & Sancho, D. (2011). The "Lurid" Downloader. Trend Micro. Retrieved from http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp_dissecting-lurid-apt.pdf.
- Quader, F., Janeja, V., & Stauffer, J. (2015, May). Persistent threat pattern discovery. In *Intelligence and Security Informatics (ISI), 2015 IEEE International Conference on* (pp. 179-181). IEEE.
- Quader, F., & Janeja, V. (2014). Computational Models to Capture Human Behavior in Cybersecurity Attacks.
- Quader, F., & Janeja, V. (2017), A Survey of Cyber Attacks & The Leading Factors 2017, Techniical Report.
- Stauffer J., & Janeja, V. (2017), A Survey of Advanced Persistent Threats and The charechteristics 2015, Technical Report.