BlackPerl

JR. SECURITY ANALYST

COOKBOOK

Practical Threat Intelligence

1.Define your requirements. Understand international relations and the geopolitical context.

2. Collect & Classify intelligence reports:

- Advanced Persistent threat, Threat Actor
- Tactics, Techniques and Procedures
- Vulnerability reports

3. Collect & Classify Indicators of Compromised (IOC):

- Incident Response
- Open-Source Intelligence (OSINT)
- Threat Hunting

4. Analyze & Triage IOCs:

- Malware and/or vulnerability analysis
- Infrastructure mapping. New domains

5. Hunt & pivot for new attacks:

- Create Yara, Sigma, Snort Rules
- Identify code similarities
- Search for infrastructure overlap & passive DNS
- MassScanning to uncover new CSs
- Set up honeypots
- Get information from private sources

6. Understand victimology:

- Who/where are the targets? Which sectors?
- Make the connections to past attacks.
- Find a link with the geopolitical context.

7. Share intelligence, dispatch IOCs, improve the knowledge base.

8. Iterate & improve the process

Diamond Model Of Intrusion Analysis

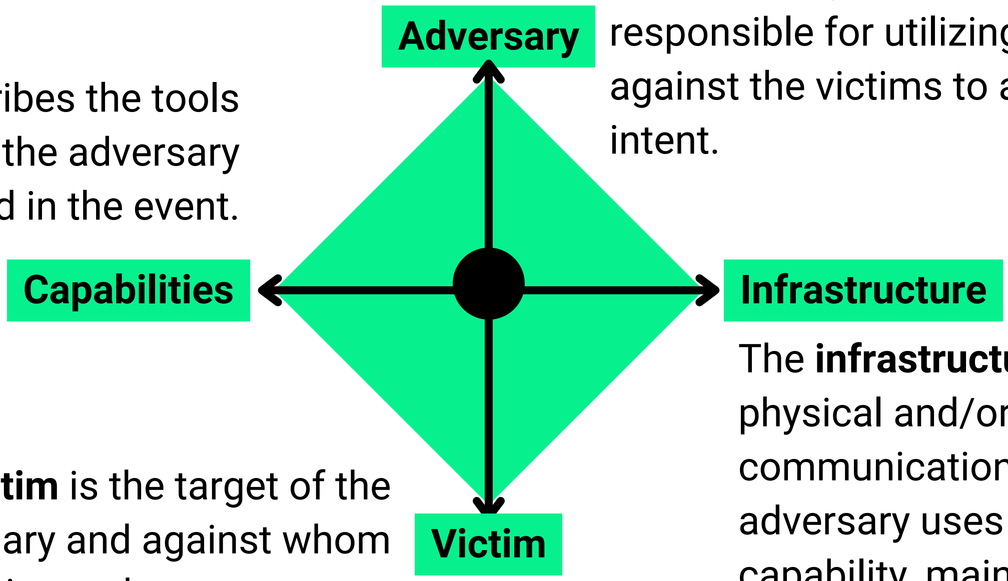
The **Diamond Model** is an approach to conducting intelligence on network intrusion events.

This model relates **four basic elements** of an intrusion: Adversary, capabilities, infrastructure and victim.

An **intrusion event** is defined as how the attacker demonstrates and used certain capabilities and techniques over infrastructure against a target.

An **Adversary** is the actor responsible for utilizing a capability against the victims to achieve the intent.

The **capability** describes the tools and techniques of the adversary used in the event.



Adversary

Capabilities

Infrastructure

Victim

The **infrastructure** describes the physical and/or logical communication structures, the adversary uses to deliver a capability, maintain control of capabilities (C2) and effect results from the victim.

A **victim** is the target of the adversary and against whom vulnerabilities and exposures are exploited and capabilities used.

BlackPerl

Log Parsing Cheat Sheet

GREP	GREP allows you to search patterns in files. ZGREP for GZIP files. $grep <pattern> file.log	-n:Number of lines that matches -i: Case insensitive -v: Insert matches -E: Extended regex -c: Count number of matches -l: Find filenames that matches the pattern
NGREP	NGREP is used for analyzing network packets. $ngrep-I filecap	-d: Specify network interface -i: Case insensitive -x: Print in alternate hexdump -t: Print timestamp -I: Read pcap file
CUT	The CUT command is used to parse fields from delimited logs. $cut -d “.” -f 2 file.log	-d: Use the field delimiter -f: The field numbers -C: Specifies characters position
SED	SED (Stream Editor) is used to replace strings in a file. $sed s/regex/replace/g	s: Search g: Replace d: Delete w:Append to file -e: Execute command -n: Suppress output
SORT	SORT is used to sort a file. $sort foo.txt	-0: Output to file -r: Reverse order -n: Numerical sort -k: Sort by column -c: Check if ordered -u: Sort and remove -f: Ignore case -h: Human sort
UNIQ	UNIQ is used to extract uniq occurrences. $uniq foo.txt	-c: Count the number of duplicates -d: Print duplicates -i: Case insensitive
DIFF	DIFF is used to display differences in files by comparing line by line. $diff foo.log bar.log	How to read output? a: Add c: Change d: Delete #: Line numbers <: File 1 >: File 2
AWK	AWK is a programming language use to manipulate data. $awk {print $2} foo.log	Print first column with separator “.” $awk -F: “{print $1}” /etc/passwd Extract uniq value from two files: awk ‘FNR--NR {a[$0]++, next} |($0 in a)’ f1txt f2txt

BlackPerl

Log Parsing Cheat Sheet

HEAD	HEAD is used to display the first 10 llines of a file by default. $head file.log	-n: Number of lines to display -c: Number of bytes to display
TAIL	TAIL is used to display the last 10 lines of a file by default. $tail file.log	-n: Number of lines to display -f: Wait for additional data -F: Same as -f even if file is rotated
LESS	LESS is used to visualize the content of a file, faster than MORE. ZLESS for compressed files. $less file.log	space: Display next page /: Search n: Next g: Beginning of the file G: End of the file +F: Like tail -f
COMM	COMM is used to select or reject lines common to two files. $comm foo.log bar.log	Three columns as output: Column 1: lines only in file 1 Column 2: lines only in file 2 Column 3: lines in both files -1, -2, -3: Suppress columns output
CSVCUT	CSVCUT is used to parse CSV files. $csvcut -c 3 data.csv	-n: Print columns name -c: Extract the specified column -C: Extract all columns except specified 1 -X: Delete empty rows
JQ	JQ is used to parse JSON files. $jq. foo.json	jq . f.json: Pretty print jq ‘[ ]’ f.json: Output elements from arrays jq ‘[0].<keyname>’ f.json
TR	TR is used to replace a character in a file. $ tr “;” “,” < foo.txt	-d: Delete character -s: Compress characters to a single one Lower to upper every character: tr “[:Lower:]” “[:upper:]” < foo.txt
CCZE	CCZE is used to color logs. $ccze < foo.log	-h: Output in html -C: Convert Unix timestamp -l: List available plugins -p: Load specified plugin

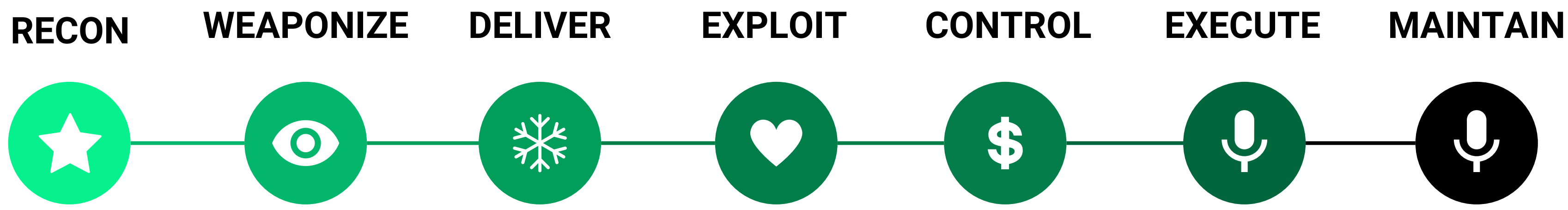
Tactics Techniques and Procedure (TTP)

TTP is a military term describing the operations of enemy forces

In **InfoSec TTP** is an approach for profiling and contextualizing cyberattack operations.

Being able to break down **complex TTP attacks** will make detection much easier to understand.

Attack Lifecycle - MITRE



RECON → WEAPONIZE → DELIVER → EXPLOIT → CONTROL → EXECUTE → MAINTAIN

Tactics

Tactics describes how an attacker operates during his operation. (Infrastructure reused, amount of entry point, compromised targets.)

Techniques

Techniques describes the approach used to facilitate the tactical phase (Tools used, malware, phishing attacks..)

Procedures

Procedures describes a special sequence of actions used by attackers to execute each step of their attack cycle.