Building a Weaponized Honeybot

Mohamed Bedewi - Senior Security Researcher at DTS Solution

Introduction and Facts

- Honeypots are decoy monitored systems configured to be intentionally vulnerable.
- The detection of a honeypot can get tricky but once it happens it can get abused.
- Honeypots can decrease the gap between security threats and security solutions.
- The technical price of deploying and maintaining a honeypot is pretty expensive.
- Honeypots may function as a stress reliever in case of persistent targeted attacks.
- The first publically available honeypot was Fred Cohen's Deception Toolkit in 1998.

Classification of Honeypots as per Design

- **Low-interaction Honeypots**

They work by emulating certain services and operating systems and have limited interaction when it comes to functionality.

The intruder’s interactions are limited to the level of emulation provided by the honeypot which makes it easily detected.

They consume relatively few hardware resources and don’t take much effort in terms of deployment and maintenance.

- **High-interaction Honeypots**

They work by fully emulating any service or operating system and have full extensive interaction when it comes to functionality.

The intruder’s interactions are limited to the level of his imagination which makes this type a time waster and hard to detect.

They consume relatively more hardware resources and take much time and effort in terms of deployment and maintenance.

Honeybots need to be well configured otherwise they can pose a risk to your network

Why We Need Honeybots?

- Honeybots can frustrate malicious users at some extent and save network resources.
- Honeybots can effectively identify internal compromised hosts and malicious users.
- Honeybots can integrate with the deployed security solutions for less false positives.
- Honeybots can robustly feed the deployed SIEM solution with accrued attack cases.
- Honeybots can easily detect encoded and encrypted network malicious payloads.

Bots are around us but we barley notice!

If you mess with my stuff, I will mess with you while I am sleeping because I am too lazy!

A Weaponized Honeybot, Really?

- Because why to frustrate malicious users when you can make their lives miserable.
- Because why to get your valuable information stolen without knowing who really did it.
- Because why to collect limited information about malicious users when you can get it all.
- Because why to be reactive to cyber breaches when you can be proactive and control it.

Passive Defense Vs. Active Defense

Detect and Attack at the Network Level

- **Detect a Network Offense**

Malicious traffic gets routed from the network security solutions to the honeybot once any indication of an attack gets detected.

The honeybot starts by validating if the ongoing attack is automated or manually conducted and once validated it will activate passive defense.

The honeybot will respond to an offense in case of a manual break-in attempt and will delay any other automated or semi-automated attempts.

- **Respond to a Network Offense**

Malicious traffic gets routed once more from the honeybot to the exploitation environment where all ports will be closed except for port 80.

The intruder will voluntarily change his attack from a network attack to a web application attack and this is when he turns into a clueless a victim.

The intruder will get fingerprinted and exploited automatically in the middle of his attempts to exploit the discovered web application.

Detect and Attack at the Application Level

- **Detect an Application Offense**

Malicious traffic can get routed from the application security solutions or custom traps can be implemented to enhance the detection ratio.

Honey Tokens: they're fake database records which can be used to identify a database breach.

Honey Pages: they're hidden sprinkled web pages which can be used to identify a persistent attack.

Honey Domains: they're dummy DNS published sub-domains which can be used to identify recon.

- **Respond to an Application Offense**

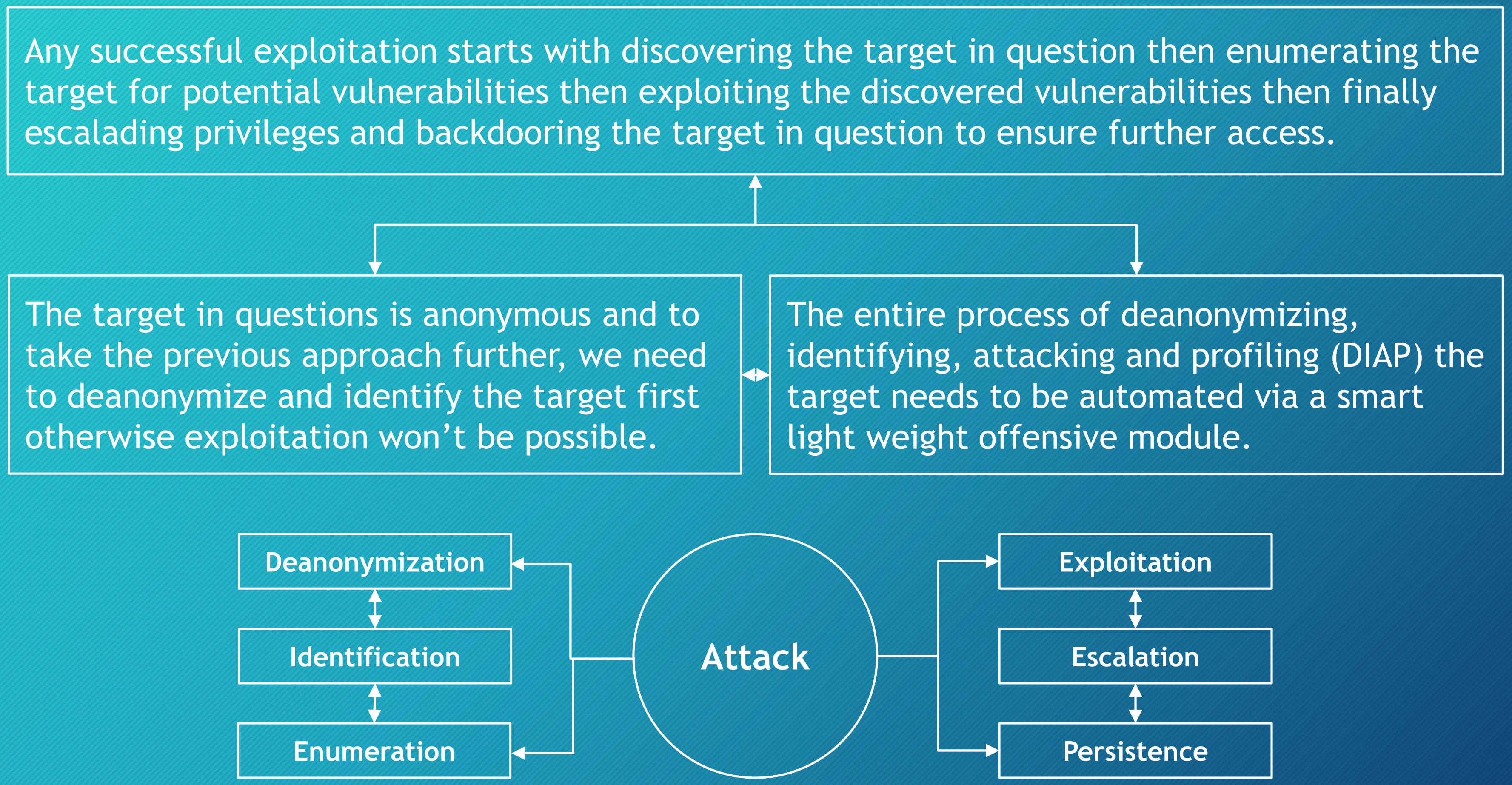
Malicious traffic gets routed once more from the honeybot or directly from the web application to the web application exploitation environment.

The intruder will get fingerprinted and exploited automatically in the middle of his attempts to discover the web page he got redirected to.

Active Defense = Offensive Security + Artificial Intelligence

I can’t believe that you’re trying to pull this off

How to logically pull this off?



Any successful exploitation starts with discovering the target in question then enumerating the target for potential vulnerabilities then exploiting the discovered vulnerabilities then finally escalading privileges and backdooring the target in question to ensure further access.

The target in questions is anonymous and to take the previous approach further, we need to deanonymize and identify the target first otherwise exploitation won’t be possible.

The entire process of deanonymizing, identifying, attacking and profiling (DIAP) the target needs to be automated via a smart light weight offensive module.

Attack

Deanonymization

Identification

Enumeration

Exploitation

Escalation

Persistence

Now since we know how it should be done let me introduce to you dynamicDetect

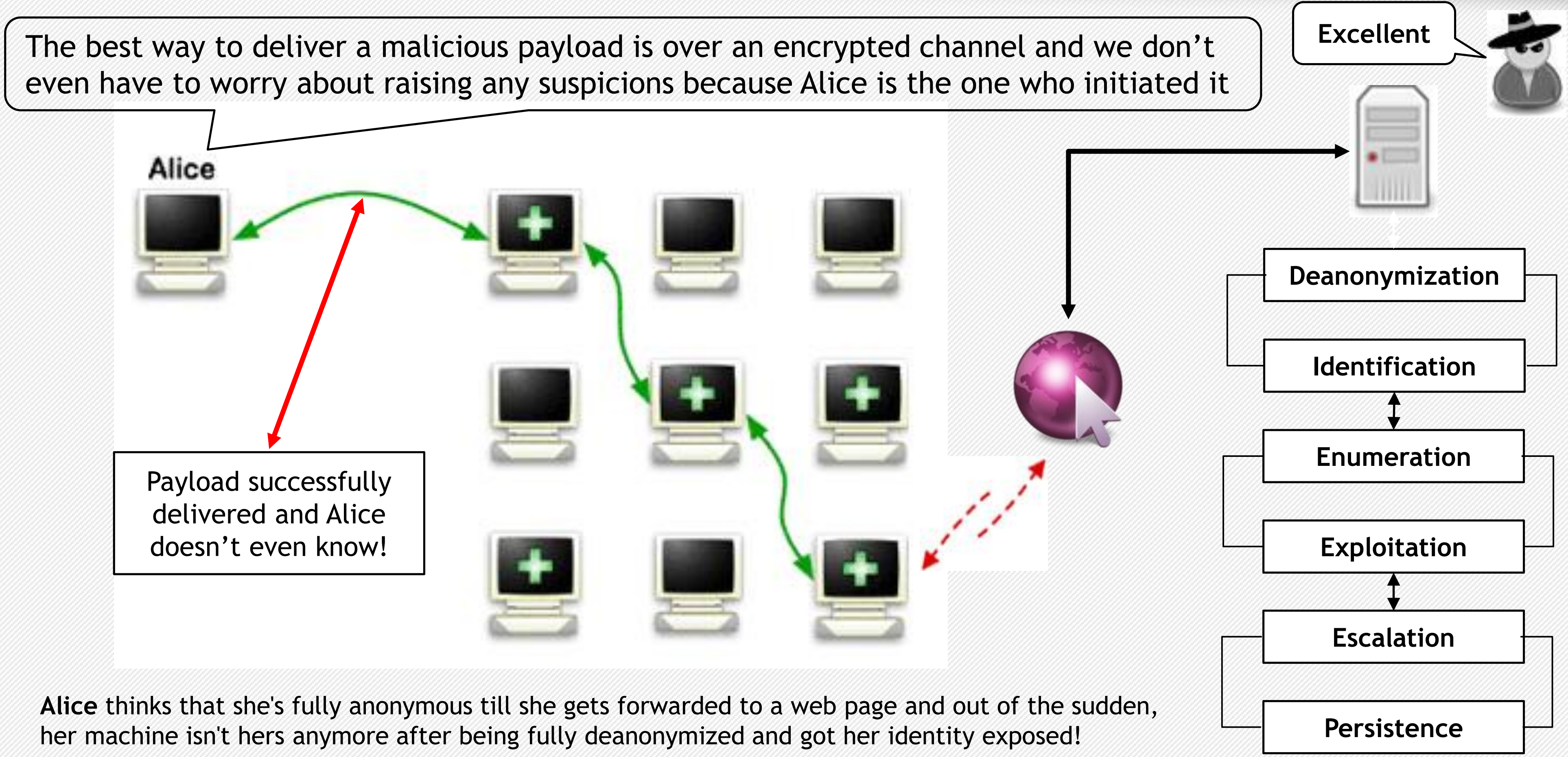
What is dynamicDetect?

Despite it’s friendly name, dynamicDetect is a very sophisticated offensive module which can effectively and robustly deanonymize, identify, attack and profile (DIAP) malicious users basically behind TOR, VPN and Proxies automatically and with zero human interaction.

dynamicDetect Technical Features:

- Capable of deanonymizing any malicious user flawlessly and accurately on the fly.
- Capable of identifying the malicious user’s IP address, country, city and coordinates.
- Capable of enumerating the malicious user’s machine to spot every single weakness.
- Capable of exploiting every single weakness identified in the malicious user’s machine.
- Capable of escalading privileges under most systems despite deployed security controls.
- Capable of maintaining access and staying stealthy even in the most strict environments.
- Capable of profiling every and each malicious user efficiently with detailed activity log.

How dynamicDetect Works?



The best way to deliver a malicious payload is over an encrypted channel and we don’t even have to worry about raising any suspicions because Alice is the one who initiated it

Excellent

Payload successfully delivered and Alice doesn’t even know!

Deanonymization

Identification

Enumeration

Exploitation

Escalation

Persistence

Alice thinks that she's fully anonymous till she gets forwarded to a web page and out of the sudden, her machine isn't hers anymore after being fully deanonymized and got her identity exposed!

Theories are easy and talk is cheap, show us dynamicDetect and let’s break this technically