Beware of Traps!

Honey token

A Honey token can take a variety of forms: file, URL, API key, environment variable, or any login information.

A Honey token is a piece of information that will not be used by legitimate users but is likely to be used by hackers. When an attacker interacts with this token, an alert is triggered to inform security administrators that suspicious activity has taken place.

> *For instance:*
>
> Security administrators of a web service place fake credentials in their database. When an attacker tries those credentials in the web service, the authentication page detects the usage of a honeytoken which raises an alarm. Security administrators are now aware that the database has probably leaked.

How to detect a honey token

Detecting a honeytoken is not an easy task. Therefore, in the context of CTF, Honeytoken has been made less stealthy to help players to found the traps.

Some honeytokens have clear identification to help security administrators distinguish honeytokens and real pieces of information. Honeytoken may be obvious.

> {id: root, password: hon3yPassword!} is an obvious honeytoken

Some honeytokens may not work at all. In this context, the attacker won't have any impact on the service because the honeytoken exists but has no interaction with the real system.

> ID and password may not work but raise an alarm when tried to authenticate in a service

Some honeytoken may lead to services that strangely behave. These are advanced honeytoken as attackers may not notice they have been detected. The main goal of these tokens is to help security administrators to capture more activity linked to the honeytoken and so, to the attacker.

> New processes or scripts are executed when a resource is accessed. The attacker will still be able to read the resource or a part of it while they think they remain undetected.
>
> Honeytoken may bring the attacker in a copy of the service, which has no real information.

Some honeytoken may provide less security to decoy attacks from real systems to fake ones.

> For example, no multifactor-authentication on a fake service.