

# Observations on Deception-Related Changes in Attack Behavior & Future Directions

---

IARPA ReSCIND PROPOSERS' DAY 2023

Merve Sahin

 @mervesahin

# Application-layer vs. Network-layer defense

- Initial access often gained via:  
Exploit of public-facing applications,  
phishing, use of valid accounts
- Web application and API attacks are continuously rising

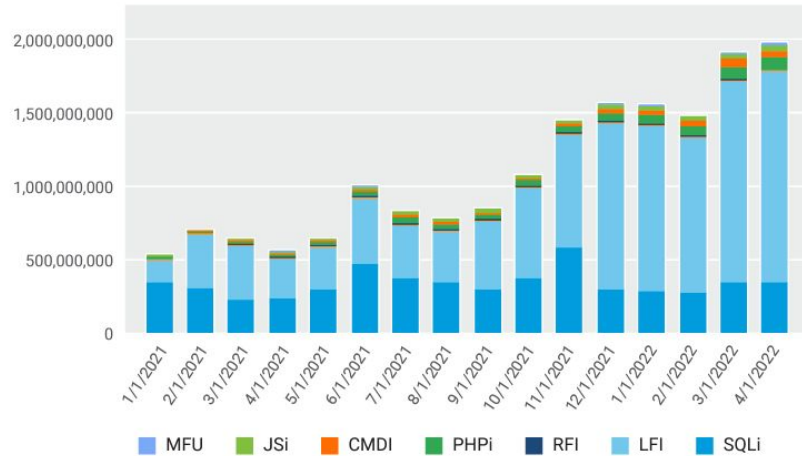


Fig. 1: Number of attacks by attack vector

[Akamai Technologies Threat Report'23]

## Initial Access

9 techniques

Drive-by Compromise	
Exploit Public-Facing Application	→
External Remote Services	
Hardware Additions	
Phishing (3)	→
Replication Through Removable Media	
Supply Chain Compromise (3)	
Trusted Relationship	
Valid Accounts (4)	→

[MITRE ATT&CK Map'22]

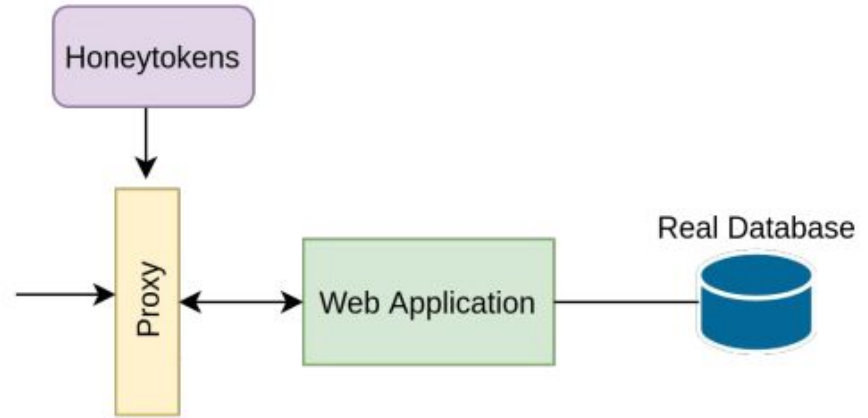
# Outline

- Our relevant work on:
  - Automation of deception for web applications
  - Observations on changes in attack behavior
- Possible future directions

# Web application layer deception

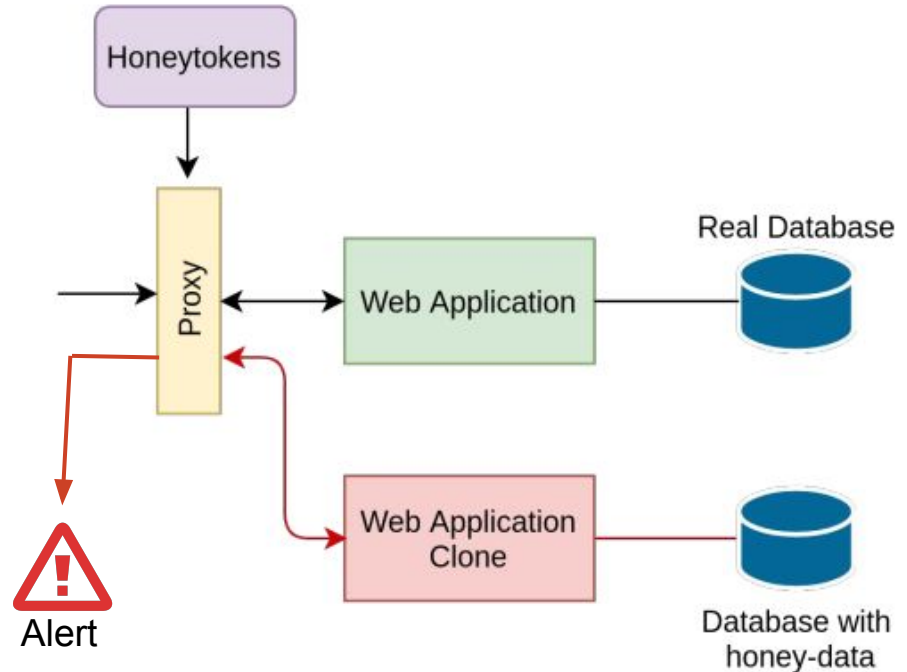
- Use of a reverse proxy to add & remove honeytokens on the fly
- Honeytokens can be in form of, e.g.,

- HTTP parameter
  - Cookie
  - User account
  - Application endpoint
  - Honey-link
- Monitored for tampering of the value
- Monitored for login attempts
- Monitored for incoming HTTP requests



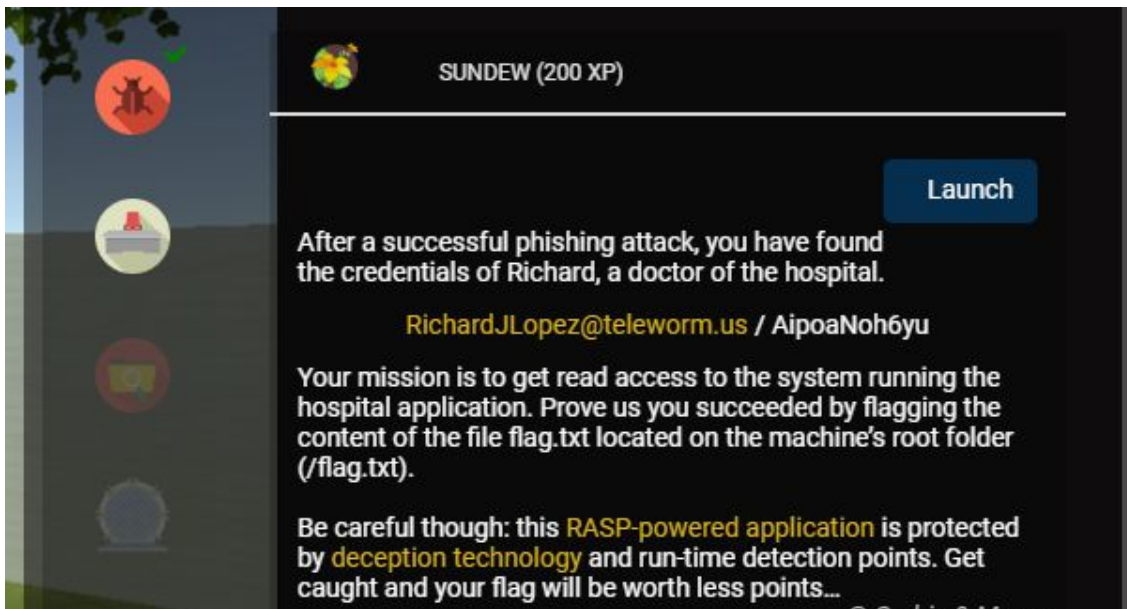
# Web application layer deception: Attack response

- Once an honeypot is triggered:
  - Alert
    - High fidelity, fast detection
  - Automatic redirection to a clone serving fake data
    - **Wasting attacker's time & effort**
    - **Cast doubt on any finding**



# Experiment #1: A Capture The Flag (CTF) challenge [3]

- 98 CTF participants informed about deception
  - Post-challenge survey evaluating participants' experience and attack behavior



# Experiment #2: Survey on real vs. deceptive parameters [4]

## API Specification

POST /oauth/token			
DESCRIPTION			
REQUEST BODY			
application/x-www-form-urlencoded			
REQUEST PARAMETERS			
Name	Description	Type	Data type
client_id		formData	string
client_secret		formData	string
redirect_uri		formData	string
code		formData	string
uaa		formData	string
grant_type		formData	string

&

## Survey listing the parameters

Form Parameters

	Deceptive	Genuine
client_id	<input type="radio"/>	<input checked="" type="radio"/>
client_secret	<input type="radio"/>	<input checked="" type="radio"/>
redirect_uri	<input type="radio"/>	<input checked="" type="radio"/>
code	<input type="radio"/>	<input checked="" type="radio"/>
uaa	<input type="radio"/>	<input checked="" type="radio"/>
grant_type	<input type="radio"/>	<input checked="" type="radio"/>

Clear selection

# Observations - Experiment #1

- 85% of participants reported that deception affected their attack strategy
- Most common reaction was to avoid automated attacks (e.g., brute-forcing, scanning, fuzzing, automation tools)

## [Participants' comments]

- I was very careful / cautious,
  - I avoid to use brute force attack.
  - especially I didnt try tampering with the cookies .
  - I investigated everything client side and interacted normally in the beginning.
  - I tried not to access .git and stuff, but finally still used dirbuster as I wasnt successful otherwise after some hours.
- At the beginning, I tried to be quiet, without scanning the webserver and focused purely on the svg upload. But after a while, none of my payload worked out, so I started with the scanning, which might be loud on server side.
- I avoided automated attacks/scanning (like port scan).
- I tried not to access things that I was sure wasn't authorized, like an ID that didn't appear. Also, avoided XSS in the text fields.
- I was focusing only on the target file, not other files in the system.
- I used the URL of a colleague to try riskier stuff
- It scared me.



# Observations - Experiment #1

- 85% of participants reported that deception affected their attack strategy
- Most common reaction was to avoid automated attacks (e.g., brute-forcing, scanning, fuzzing, automation tools)
- Participants fall back to the conventional strategies if they don't find a way out

## [Participants' comments]

- I was very careful / cautious,
  - I avoid to use brute force attack.
  - especially I didnt try tampering with the cookies.
  - I investigated everything client side and interacted normally in the beginning.
  - I tried not to access .git and stuff, but finally still used dirbuster as I wasnt successful otherwise after some hours.
- At the beginning, I tried to be quiet, without scanning the webserver and focused purely on the svg upload. But after a while, none of my payload worked out, so I started with the scanning, which might be loud on server side.
- I avoided automated attacks/scanning (like port scan).
- I tried not to access things that I was sure wasn't authorized, like an ID that didn't appear. Also, avoided XSS in the text fields.
- I was focusing only on the target file, not other files in the system.
- I used the URL of a colleague to try riskier stuff
- It scared me.

# Observations - Experiment #2

- Anchoring bias: Participants find deception even when it doesn't exist  
(Also observed by Ferguson et al. & Gutzwiller et al. [5, 7])
- Uncertainty: Is it just due to bad API design practices, or due to deception?

## [Participant's comment]

I would be extra careful in a situation like this and mark things [that maybe are not deceptive] as deceptive just in case. Taking into account that programmers are not perfect, they may create parameters that are not needed. So I think this is not needed, but is it because it is deceptive or it was done like this in reality... My general approach when doing tampering is, just touch what you are sure of.

# Directions for Future Work

Some empirical evidence on cognitive effects,

but we need a more systematic approach!

# Directions for Future Work

#1 Understanding attackers' cognitive biases:

## Mapping cognitive biases to attackers' sequence of actions

- The commonalities in the **initial attack steps** can relate to the **thin slicing bias**,
- Attackers' **persistence on failed exploit attempts** can refer to **sunk cost fallacy**,
- If an attacker is **stuck in one attack path**, despite additional findings or evidence, this can refer to **anchoring bias**,
- If an attacker chooses a very **difficult/unlikely attack path**, they might be incorrectly predicting their abilities (**Dunning-Kruger effect**),
- If an attacker is **over-complicating** a solution, this can refer to the **Einstellung** effect.


# Directions for Future Work

## #2 Exploiting attackers' cognitive biases:

- Thin slicing bias: Contradict attackers' common assumptions and expectations
- Sunk cost fallacy: Simulate fake attack progress
- Dunning Kruger effect: Decrease perceived risk (e.g. no visible detection),  
increase attacker's self-confidence
- Anchoring bias: Embed hints on simulated vulnerabilities
- Einstellung effect: Artificially increase the attack surface with known vulnerabilities

# Challenges

- Creating a **realistic environment**
  - Designing multi-stage attacks
  - Recording & analysis of all steps
- Simulating genuine **attack motivation**
  - Intrinsic / extrinsic motivations?
- Simulating '**risk**'
  - What is the risk for an attacker?
    - Losing access
    - Vulnerabilities being patched
- **Human subjects**
  - Security experience
  - Attacker mindset



Need for multidisciplinary approaches & collaborations

# References

[1] Akamai Web Application and API Threat Report, 2022.

<https://www.akamai.com/resources/research-paper/akamai-web-application-and-api-threat-report>

[2] ATT&CK Matrix for Enterprise, 2022. <https://attack.mitre.org/>

[3] M. Sahin, C. Hebert, A. Santana de Oliveira. *Lessons Learned from SunDEW: A Self Defense Environment for Web Applications*. In Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb'20) co-located with NDSS'20.

[4] M. Sahin, C. Hebert, R. Cabrera Lozoya. *An Approach to Generate Realistic HTTP Parameters for Application Layer Deception*. In proc. of the 20th International Conference on Applied Cryptography and Network Security (ACNS'22).

[5] R. S. Gutzwiller, K. J. Ferguson-Walter, and S. J. Fugate. "Are cyber attackers thinking fast and slow? Exploratory analysis reveals evidence of decision-making biases in red teamers," Proc. of the Human Factors and Ergonomics Soc., vol. 63, pp. 427-431. 2019.

[6] C. K. Johnson, R. S. Gutzwiller, J. Gervais and K. J. Ferguson-Walter, "Decision-Making Biases and Cyber Attackers," 2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW), 2021,

[7] K. J. Ferguson-Walter, M. M. Major, C. K. Johnson, and D. Muhleman. "Examining the efficacy of decoy-based and psychological cyber deception," USENIX Security Symposium. 2021