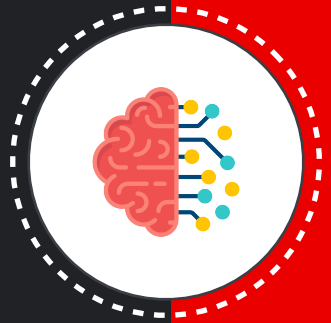


Psychology and Security

Demotivating Persistent Attackers

Jarrod Overson - @jsoverson
Director of Engineering, Shape Security

InfoSecon 2018

HOW DO YOU ENGAGE WITH ATTACKERS WHILE UNDER ATTACK?

HOW DO YOU KNOW YOU ARE UNDER

ATTACK

IN THE FIRST PLACE?

0 Imitation attacks and attacker sophistication

1 The economics of attacks

2 Flipping the economics in your favor

3 Case Studies

IN THE BEGINNING,

ACCESS

WAS GIVEN TO EVERYONE & EVERYTHING

THE MORE WALLS

WE PUT UP

THE HARDER IT BECAME TO TELL

HUMANS AND

ATTACKERS APART


PLEASE
CLOSE
GATE


GATE, YOU HAD ONE JOB

TO AFFECT BEHAVIOR,

YOU NEED TO

REMOVE THE

INCENTIVE

MANUAL ATTACKS

Sufficient when value is high

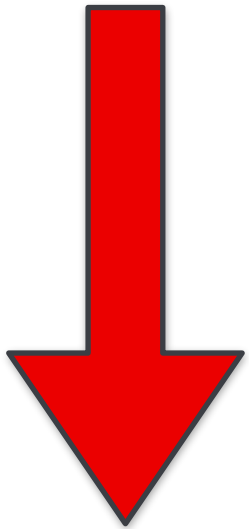
Can’t scale when value per attack is reduced

AUTOMATED ATTACKS

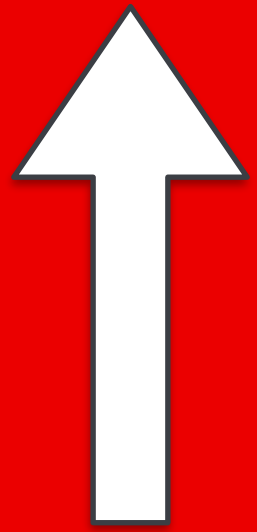
Sufficient when value is low

Can’t scale when cost per attack is increased

THE SECRET TO DEFEATING ATTACKERS

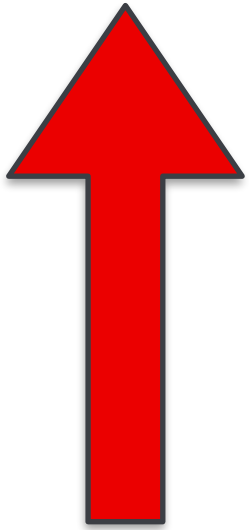


Decrease value

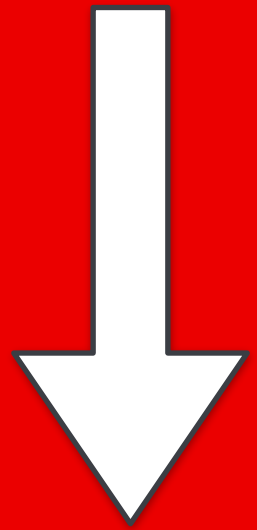


Increase cost

THE SECRET TO HAPPY USERS



Increase value



Decrease cost

0 Attacker sophistication & where we are

1 The economics of attacks

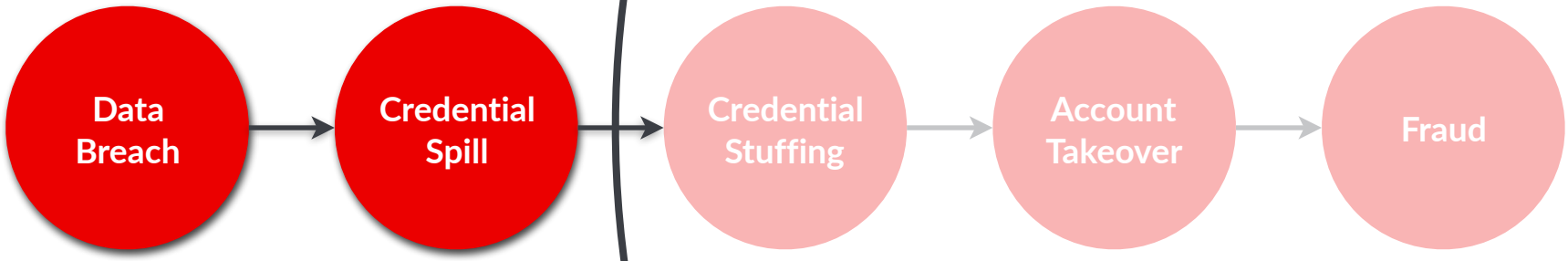
2 Flipping the economics in your favor

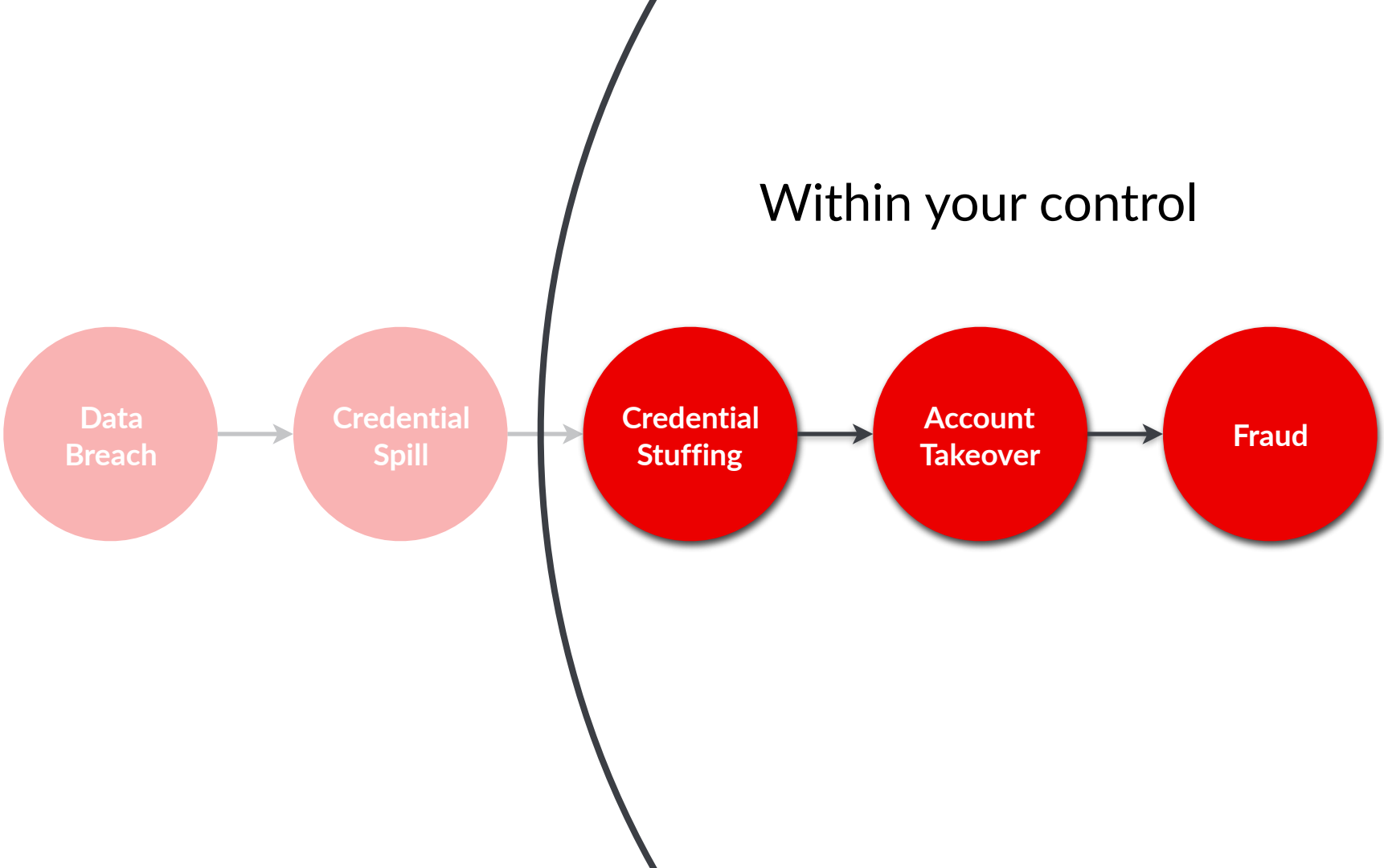
3 Case Studies

Attack Detail: **Credential Stuffing**

FROM DATA BREACH TO DAMAGE

Outside your control




Within your control
Data Breach
Credential Spill
Credential Stuffing
Account Takeover
Fraud

Your web applications and APIs are the battlefield.

CREDENTIAL STUFFING

cre·den·tial stuff·ing
/krəˈden(t)SHəl ˈstəfiNG/

The automated replay of breached username/password pairs across many sites in order to take over accounts where passwords have been reused.

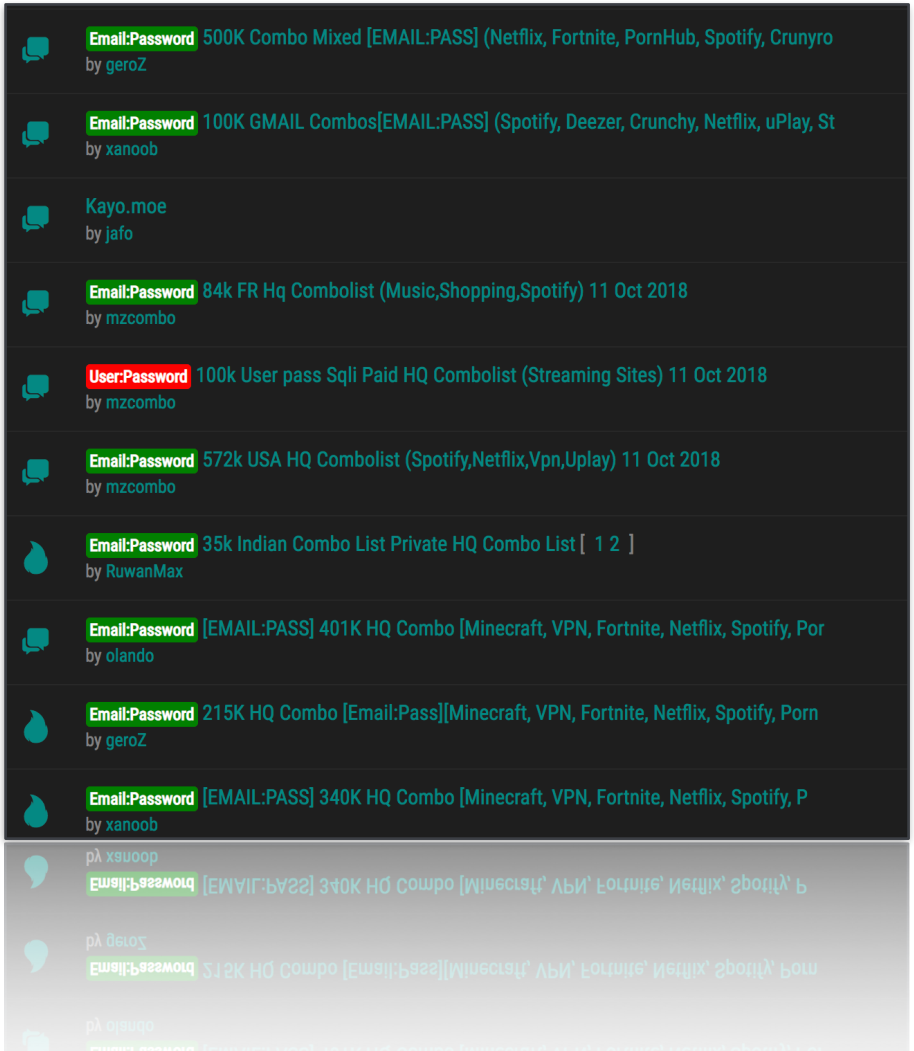
A STEP BY STEP GUIDE

1. Get Credentials
2. Automate Login
3. Defeat Automation Defenses
4. Distribute Globally

CREDENTIAL STUFFING

1

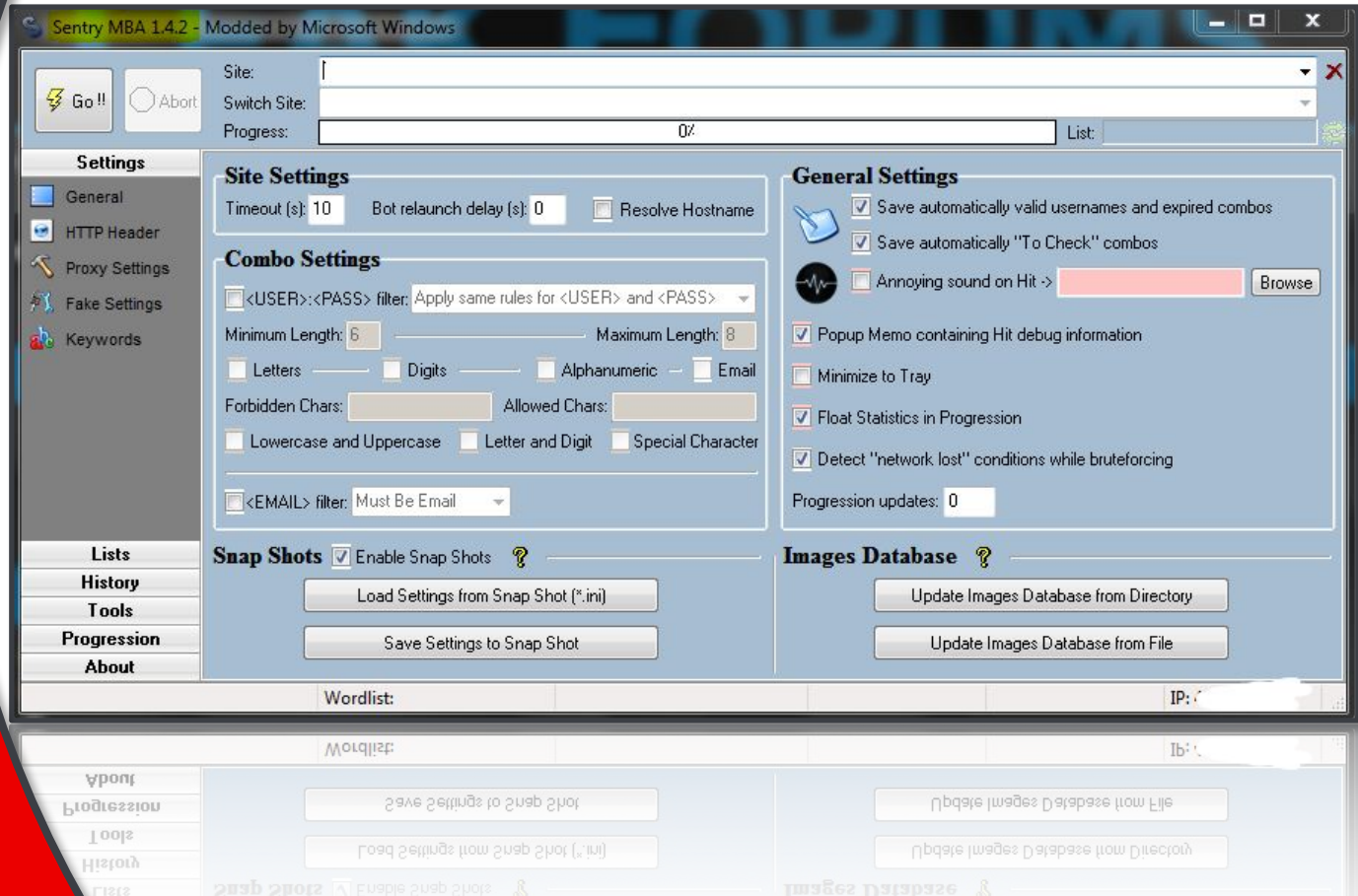
1. Get Credentials



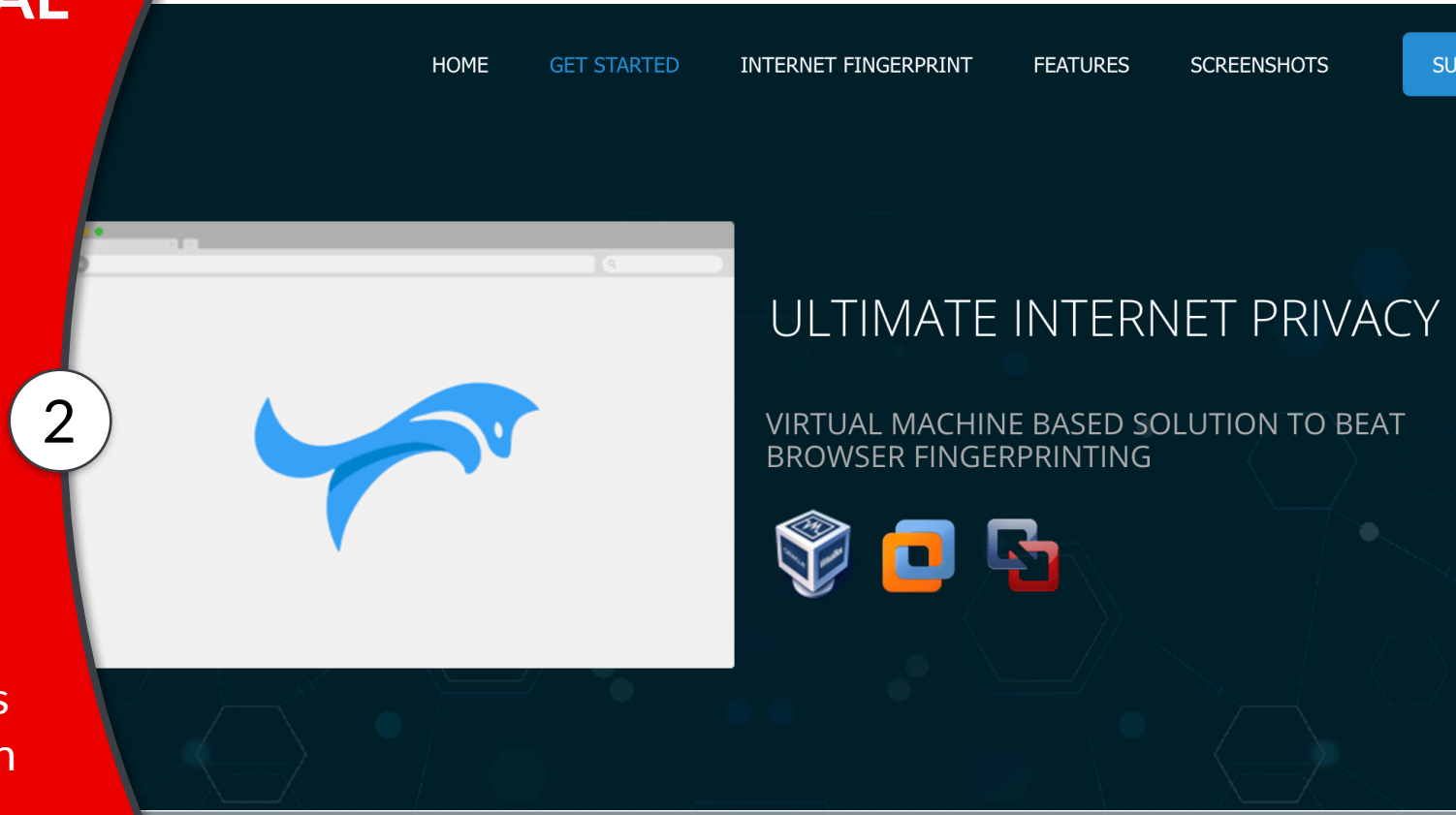
CREDENTIAL STUFFING

1. Get Credentials
2. Automate Login

2



CREDENTIAL STUFFING

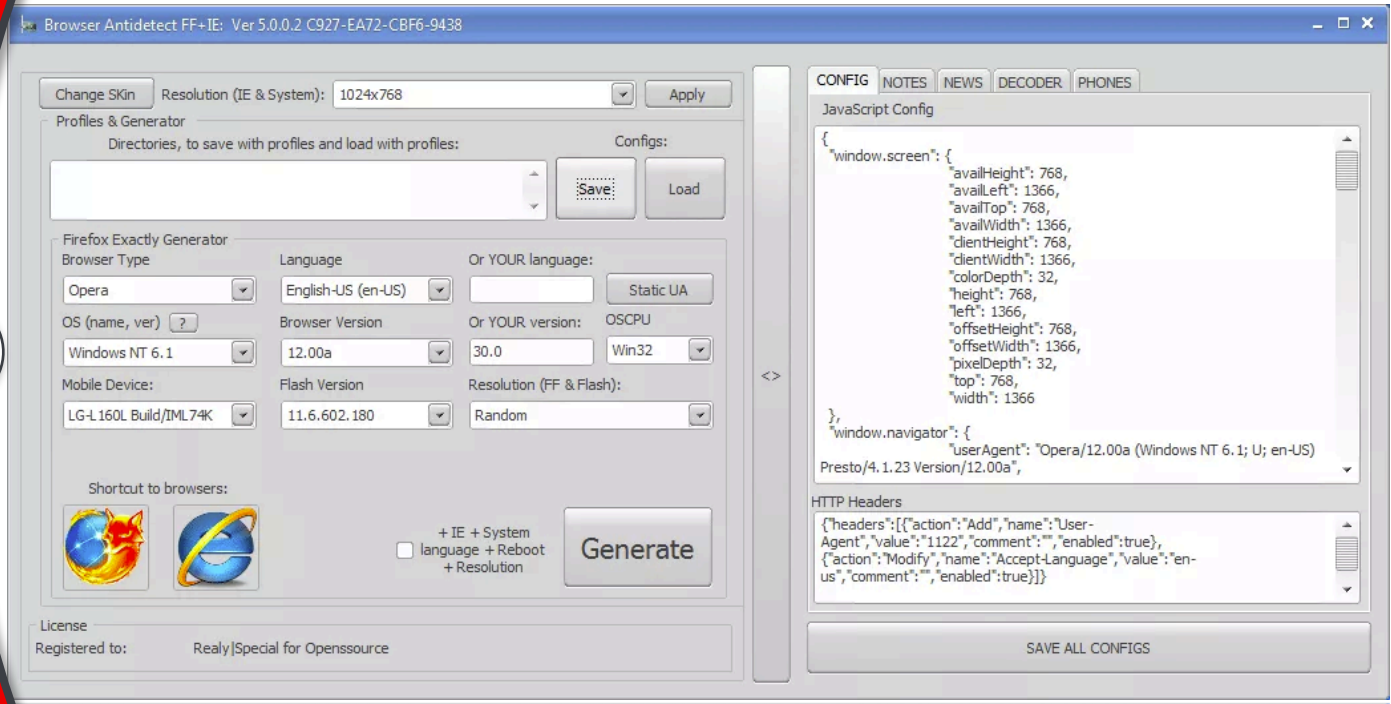


1. Get Credentials
2. Automate Login

CREDENTIAL STUFFING

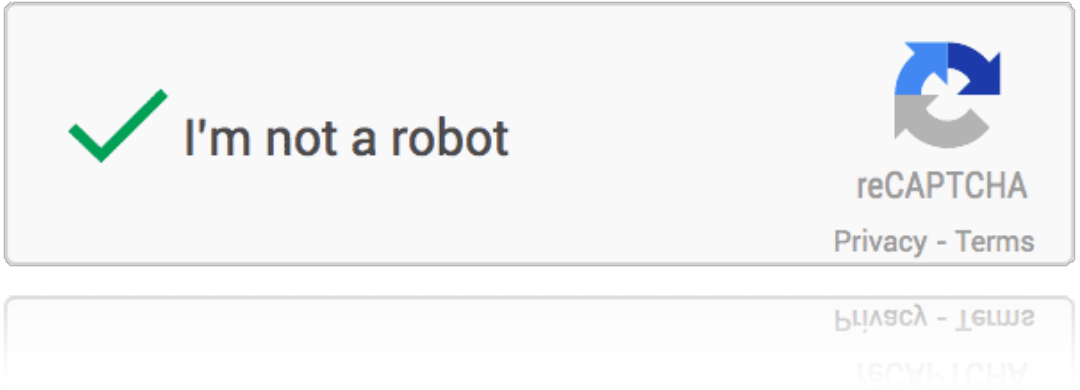
2

1. Get Credentials
2. Automate Login



CREDENTIAL STUFFING

3

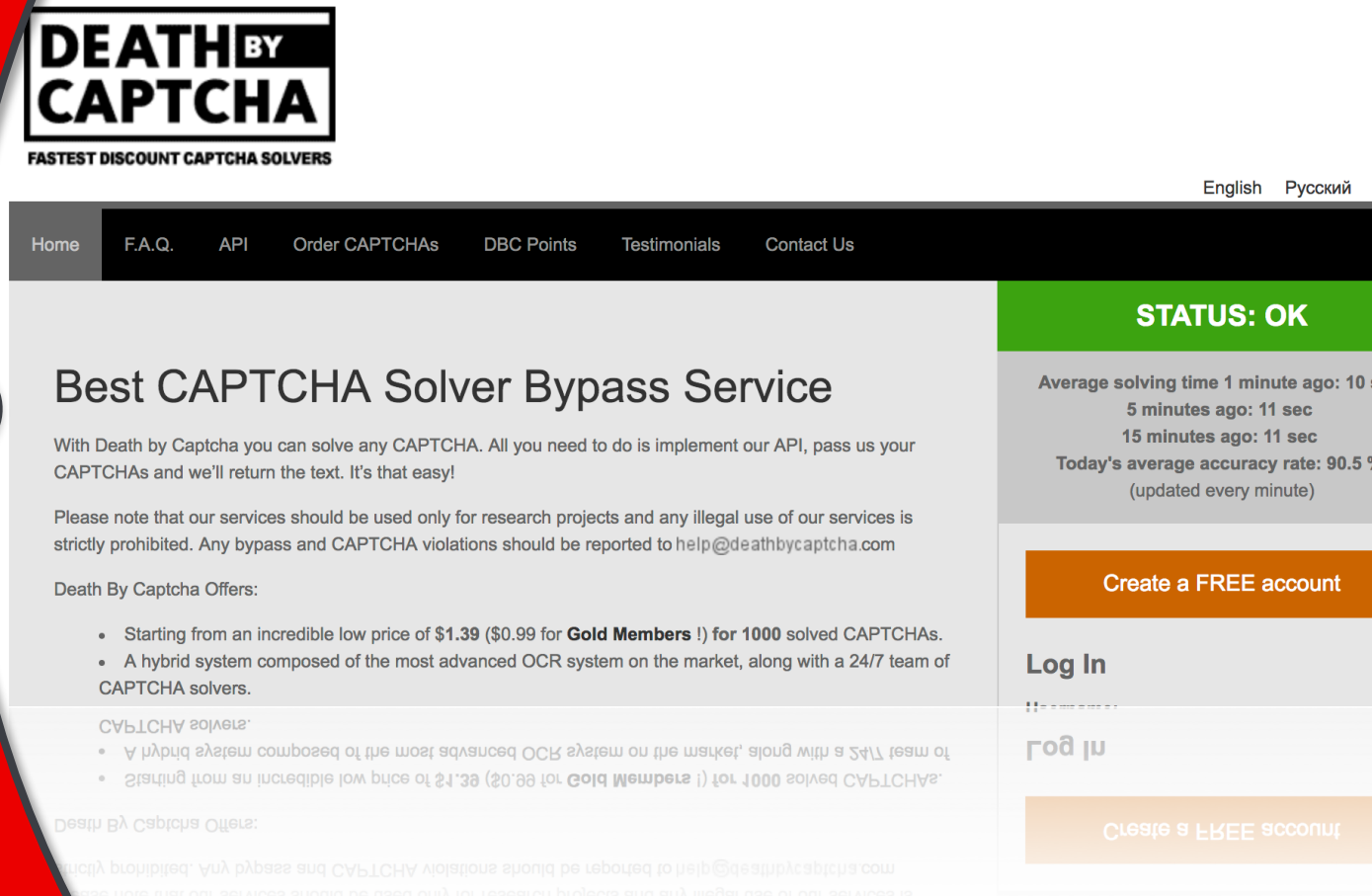


1. Get Credentials
2. Automate Login
3. Defeat Defenses

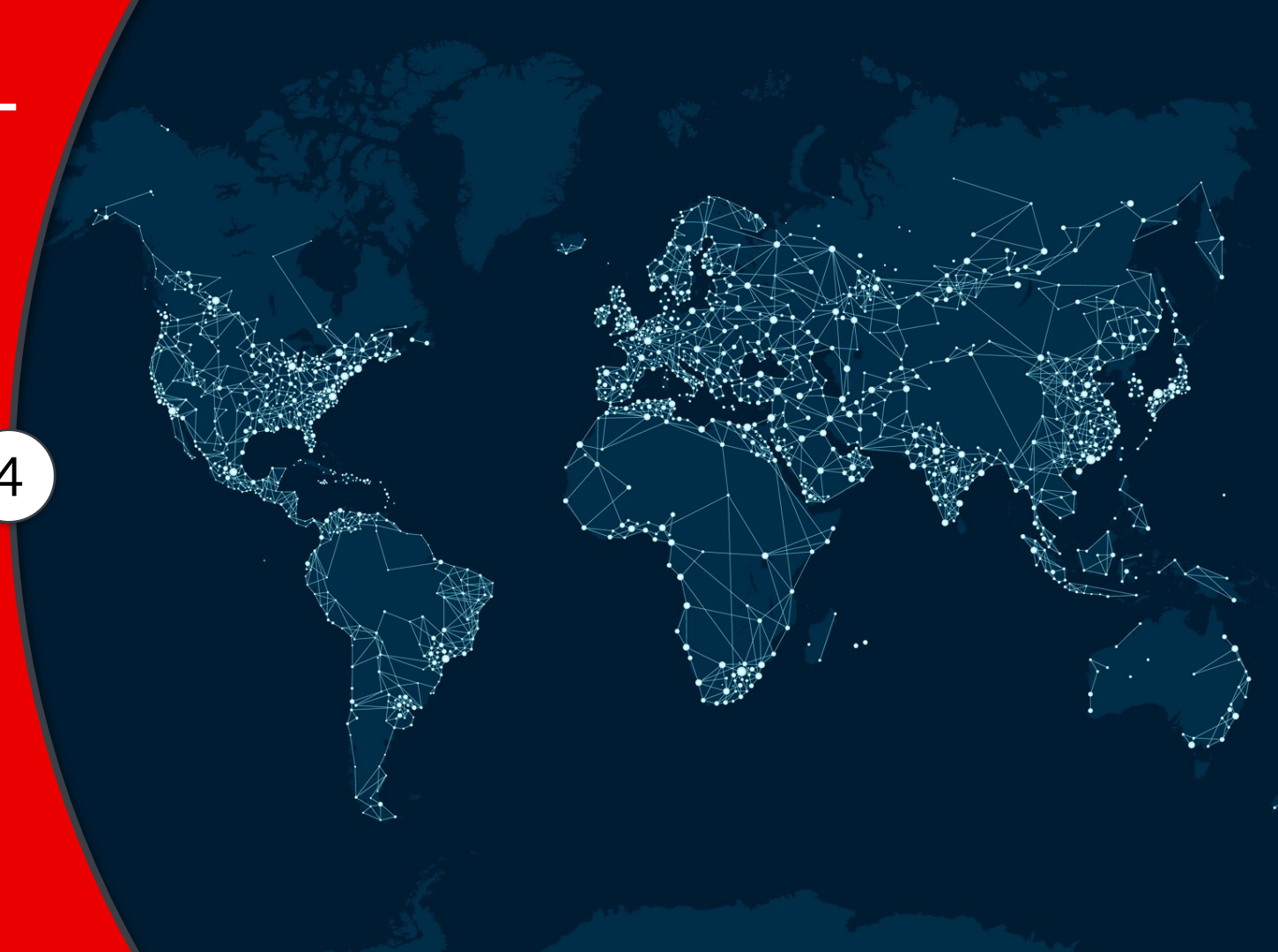
CREDENTIAL STUFFING

1. Get Credentials
2. Automate Login
3. Defeat Defenses

3



CREDENTIAL STUFFING



4

1. Get Credentials
2. Automate Login
3. Defeat Defenses
4. Distribute

CREDENTIAL STUFFING

1. Combolists starting at $0
2. $50 per site configuration
3. $1.39 per 1000 CAPTCHAs
4. $2 for 1000 global IPs

Less than $200 for 100,000 ATO attempts

0 Attacker sophistication & where we are

1 The economics of attacks

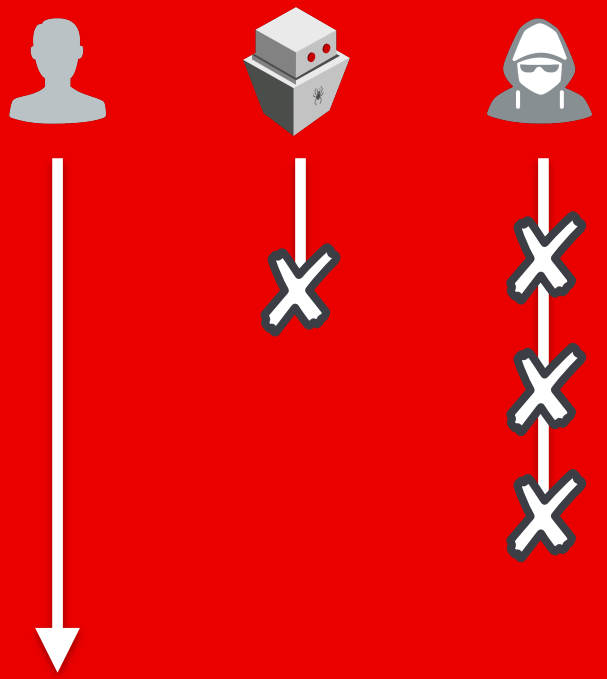
2 Flipping the economics in your favor

3 Case Studies

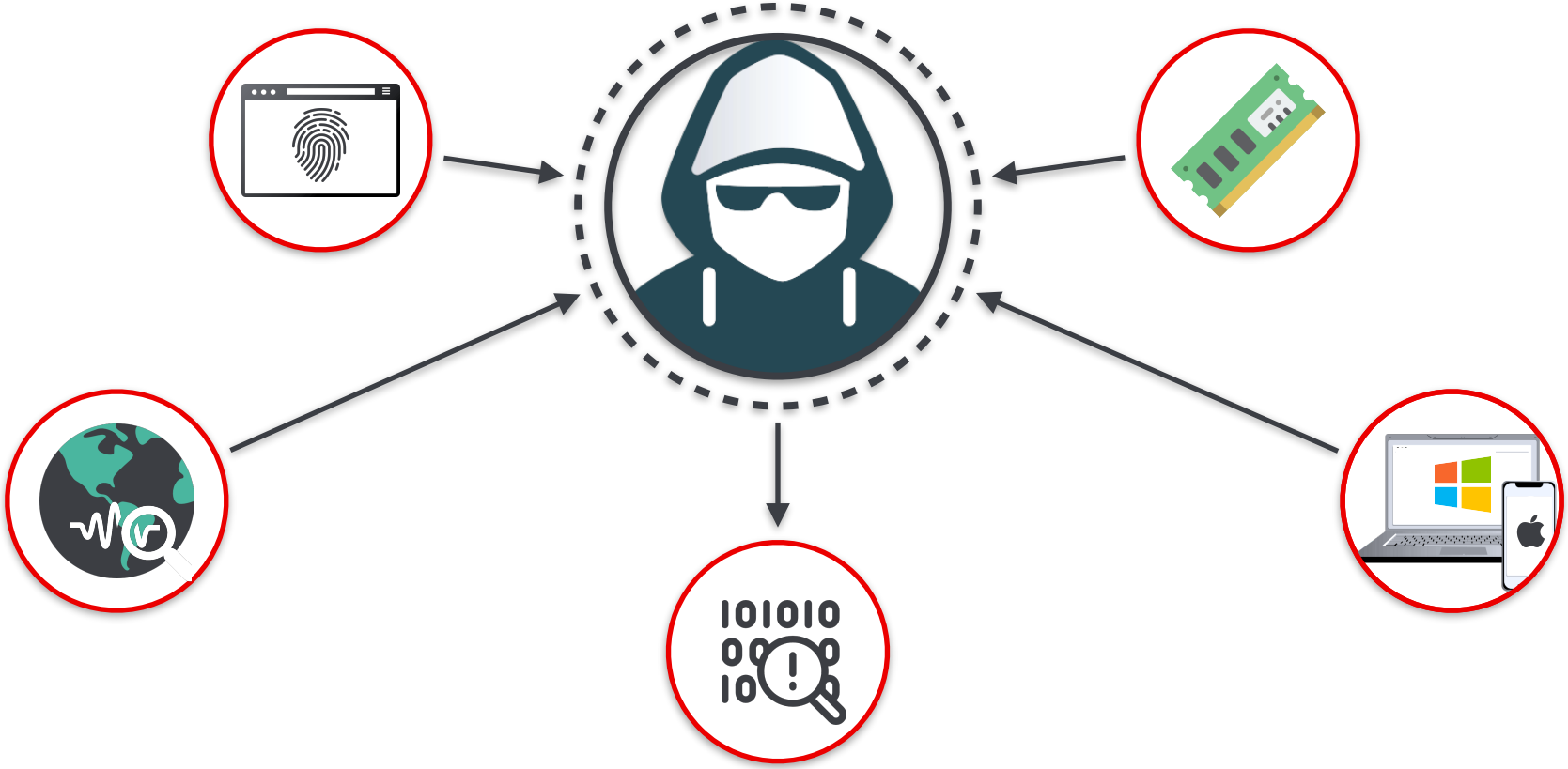
DETECTION

VS

MITIGATION

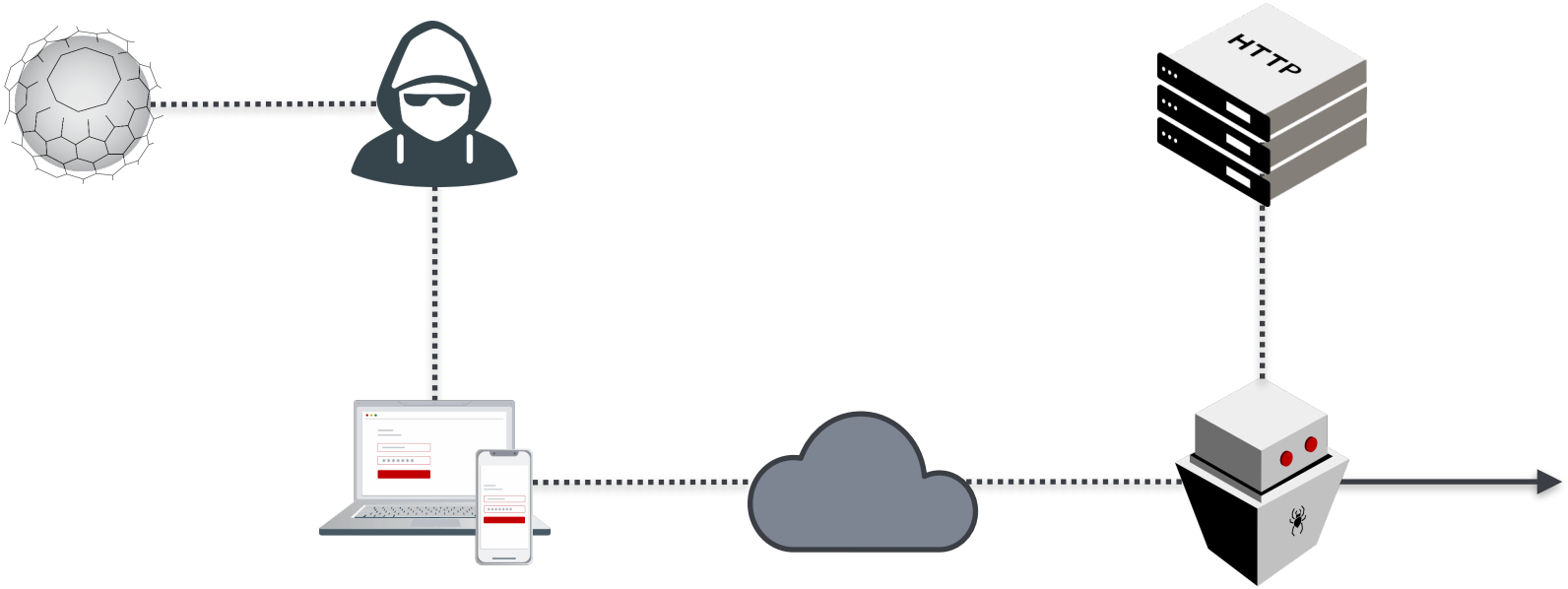


Have multiple ways to detect bad actors



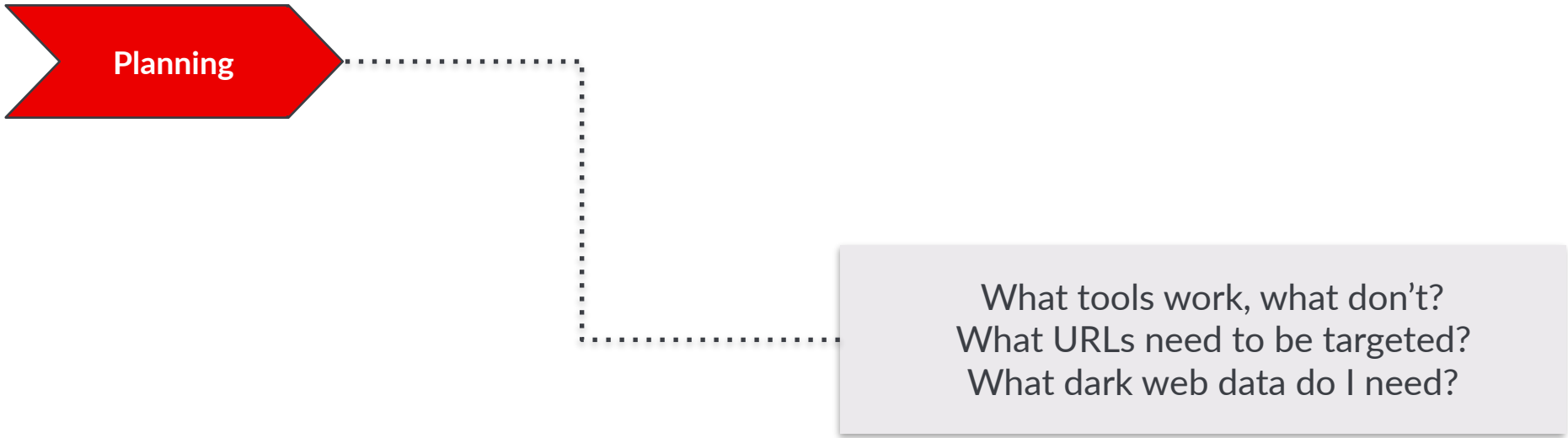
Also have ways to detect when they've started to pry open your systems

TARGET WHAT HURTS THE MOST

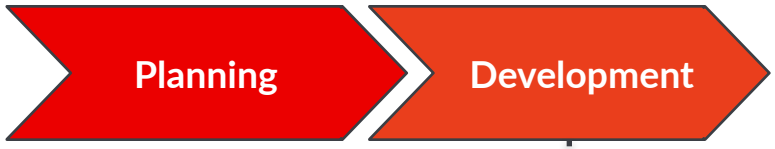


It’s not as simple as blocking a baddie.
You need to target what will hurt the most.

THE SOFTWARE DEVELOPMENT LIFECYCLE

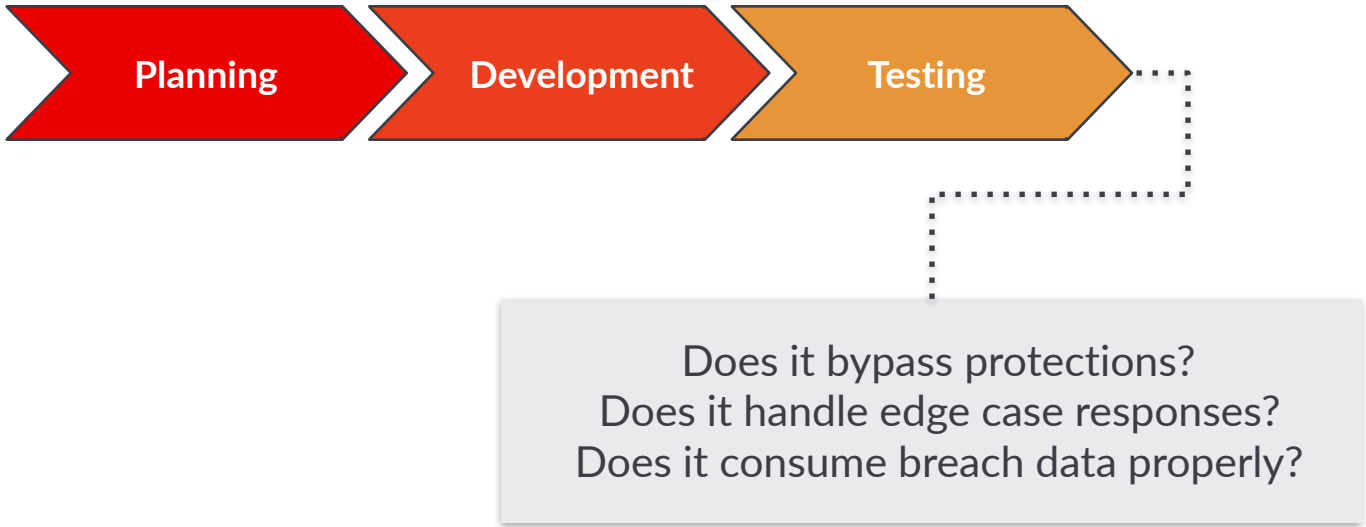


THE SOFTWARE DEVELOPMENT LIFECYCLE

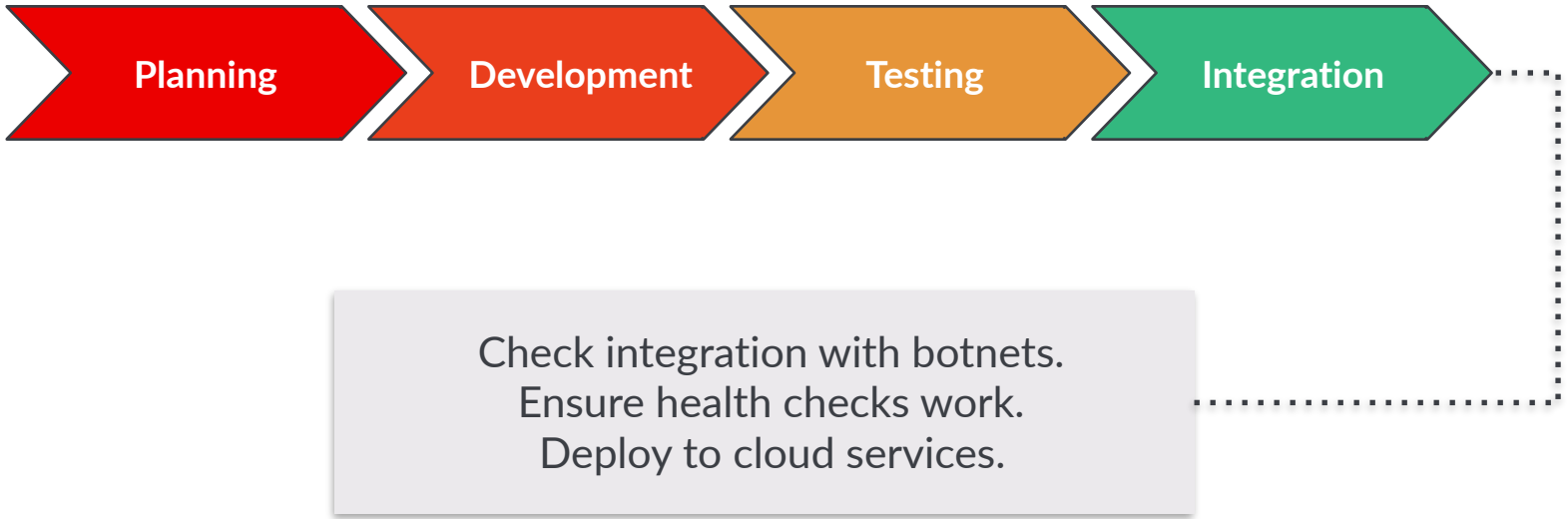


Investment in a framework of choice.
Custom development against a site.
Building in proxy/botnet hooks.

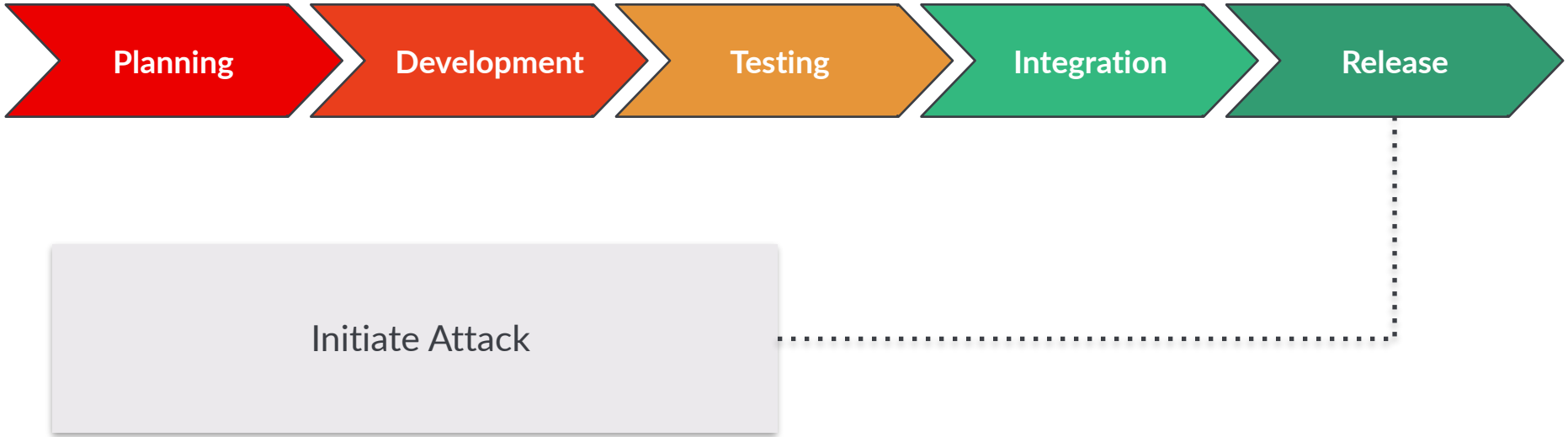
THE SOFTWARE DEVELOPMENT LIFECYCLE



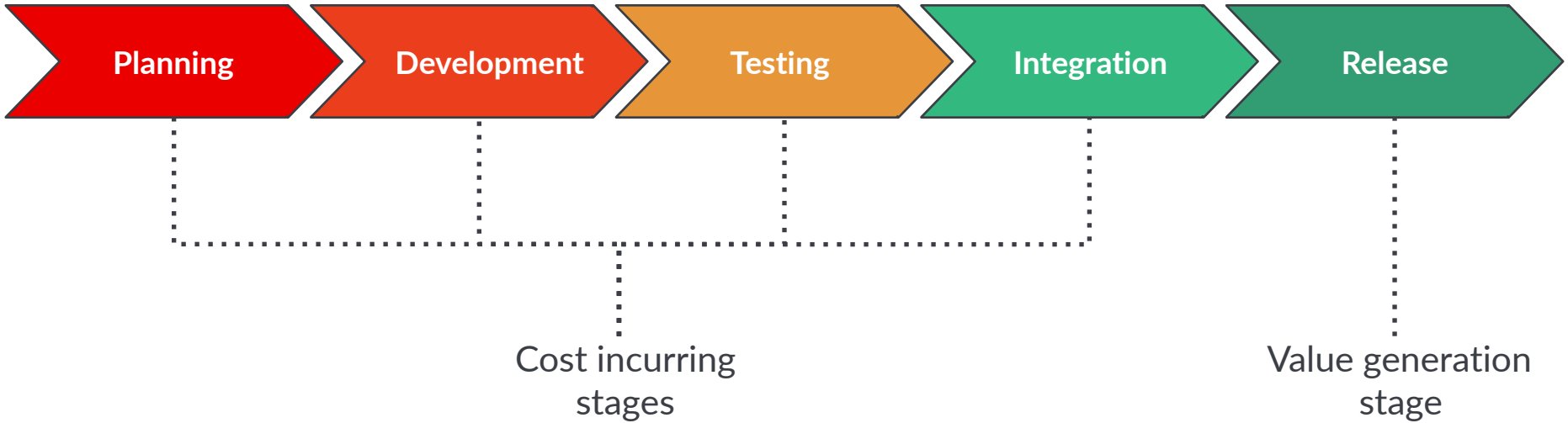
THE SOFTWARE DEVELOPMENT LIFECYCLE



THE SOFTWARE DEVELOPMENT LIFECYCLE



THE SOFTWARE DEVELOPMENT LIFECYCLE



0 Attacker sophistication & where we are

1 The economics of attacks

2 Flipping the economics in your favor

3 Case Studies

Case Study 1

Damaging Reputation

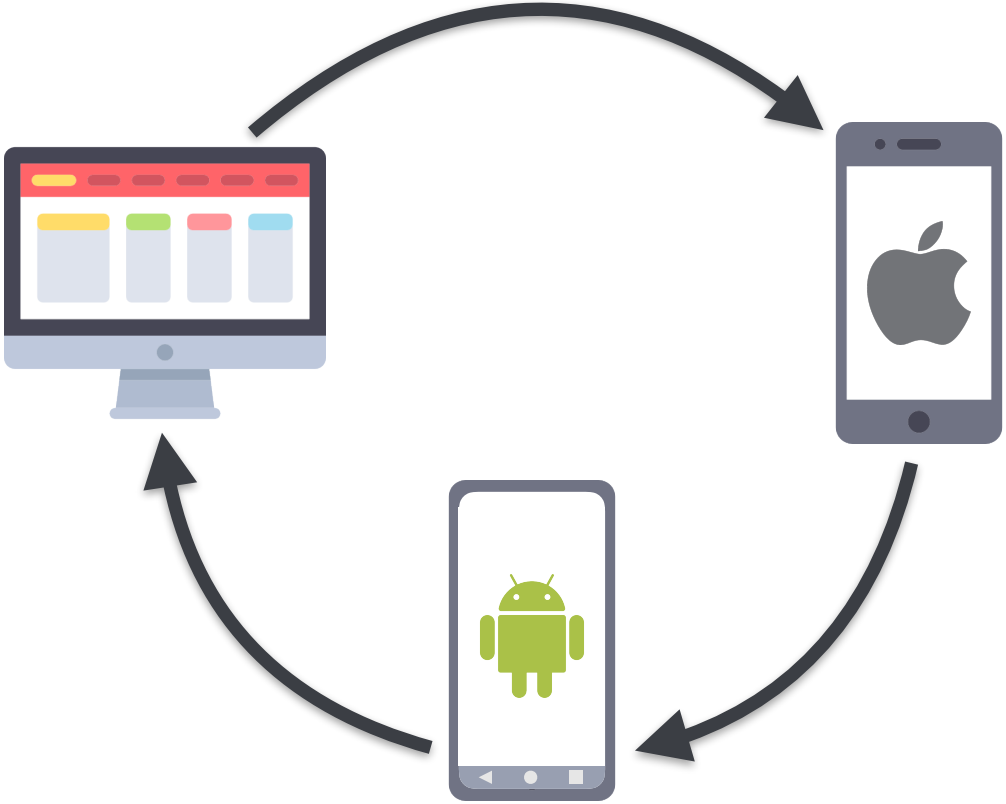
Scenario

Well funded scraper

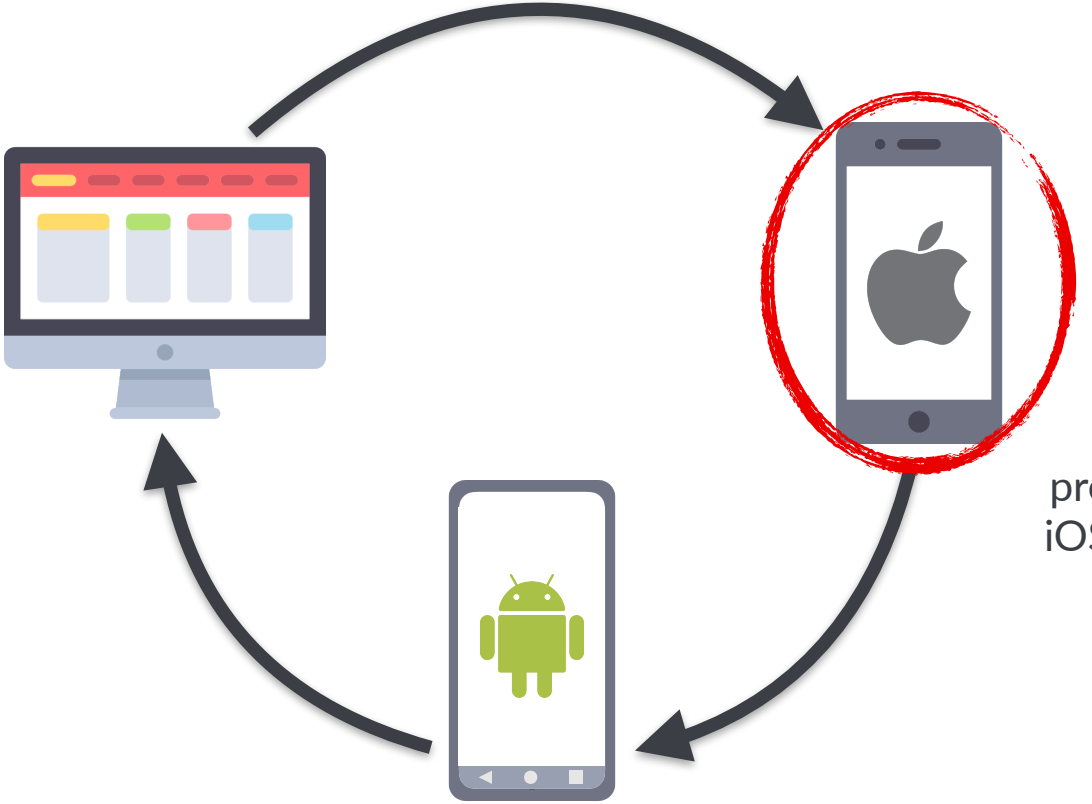
VS

Big US Bank

The actor cycled through the softest targets

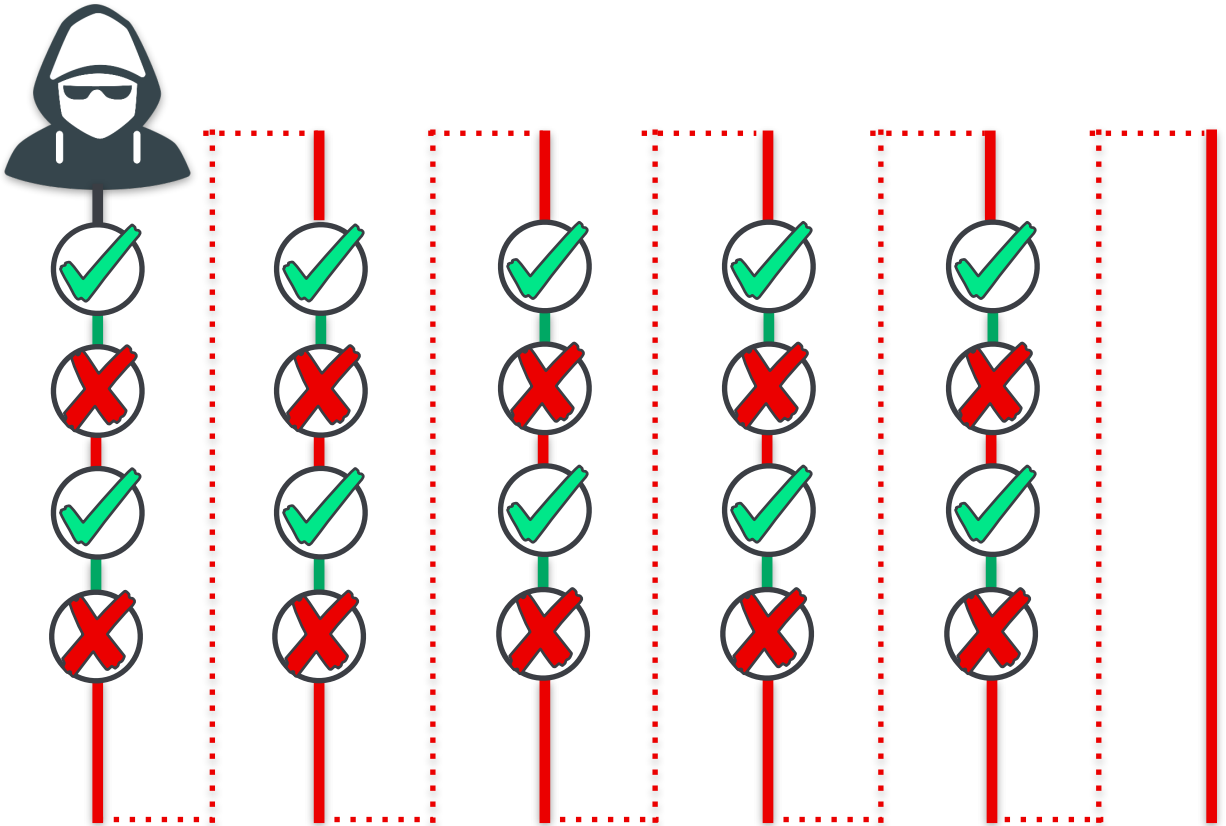


Finally committed to one to dive deeper

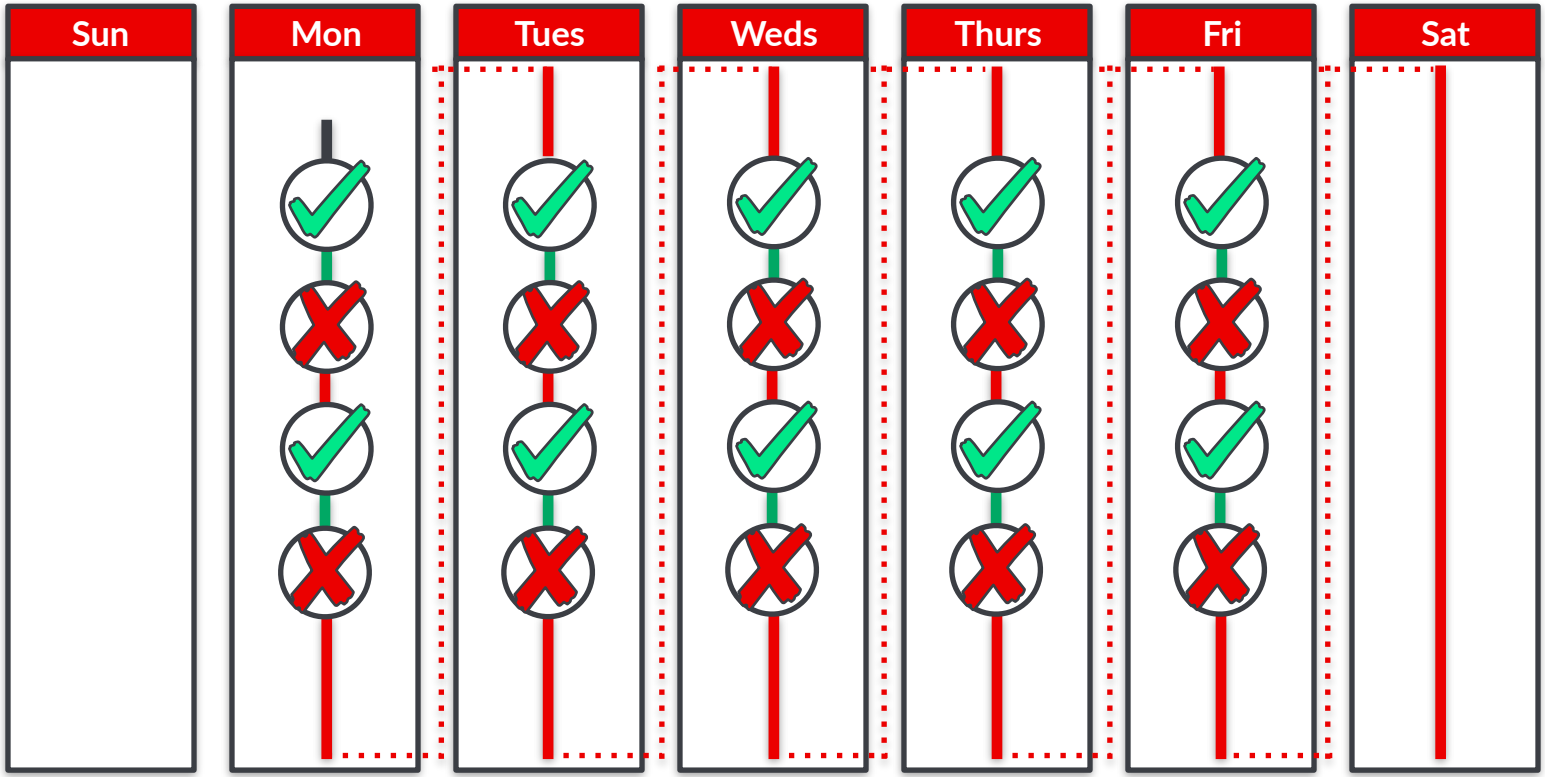


The threat found prolonged success on iOS due to version lag

Got around defenses regularly, though not durably



Recognizing patterns in behavior



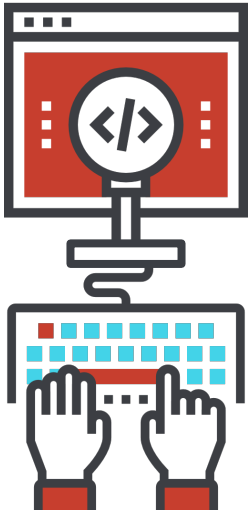
Analysis

1. Regular working schedule
2. Actor’s consumers were notified upon success
3. Failure was met with downstream frustration
4. Prolonged failure provoked distress

Plan of action

1. Target defense out of working schedule
2. Turn on defenses when damage would be highest
3. Turn off primary mitigation during working schedule
4. Cycle through defenses even when still working

Case Study 2



Github Kiddies

Scenario

Credential Stuffer
and Account taker-overer

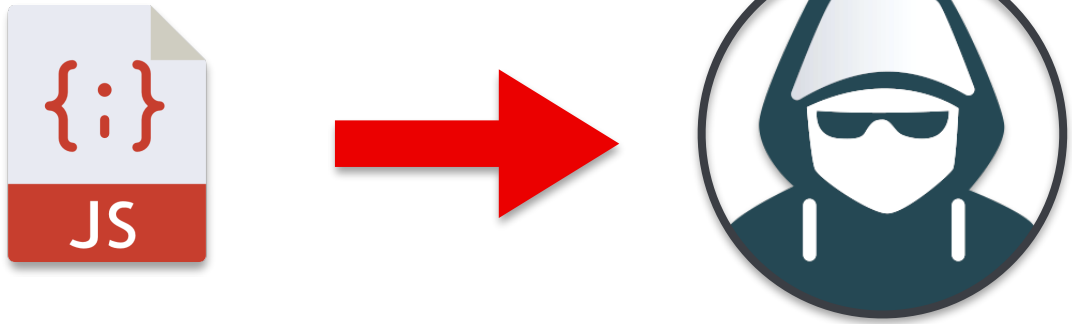
VS

Big US Retailer

We were down to a fraction of our normal ability to detect

We needed more data

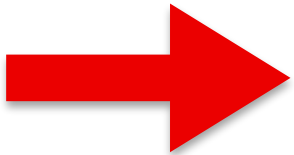
We had enough of a grip to deliver a targeted payload



This allowed us to inspect the retooling effort in real time

```
> function doBadStuff() {
    if (iCanHazAccounts) {
      stealAllAccounts();
    } else {
      injectMaliciousScripts();
    }
  }
```

What we learned



Analysis

1. Actor was a competent developer
2. Still relied on community to get around problems
3. Bypassed defenses via trial and error
4. Actor was been lucky, not wildly skilled

Plan of action

1. Build up defenses based on the tool he was using
2. Provide variable feedback during retooling phase
3. Turn on just enough to be infuriating. No more, no less
4. Create new countermeasures that act differently during retooling phase

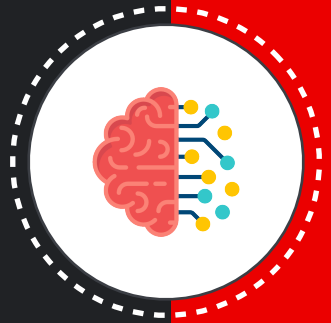
Recap

1. Treat detection and mitigation separately.
2. Protect the data used to detect.
3. Understand what is incentivizing your attackers.
4. Work with product to build in app-level defenses.

Tactics is knowing what to do when there is something to do.

Strategy is knowing what to do when there is nothing to do.

- Savielly Tartakower



Psychology and Security

Demotivating Persistent Attackers

Jarrod Overson - @jsoverson
Director of Engineering, Shape Security

InfoSecon 2018