



# ACSC

AUSTRALIAN CYBER SECURITY CENTRE

# 2017

# CONFERENCE

14 – 16 March | Canberra

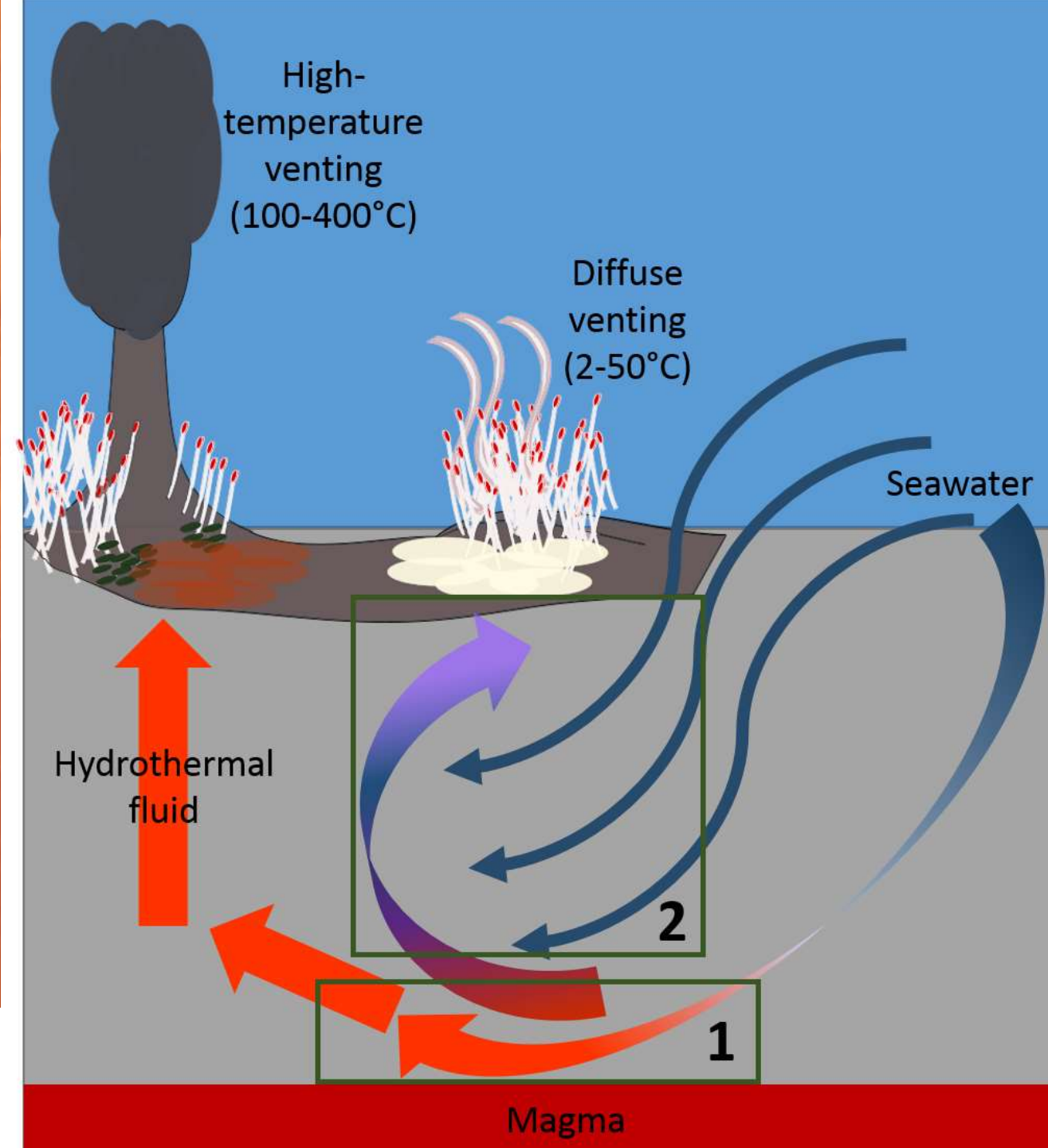


# Resident Hyperthermophiles of Kermadec Hydrothermal Vents

*Implications for astrobiology and the understanding of the limits of life*

# Hydrothermal systems

- Pressures and temperatures > 200 atm and 80°C
- Extremely rich in chemical energy
- Life is found in them, and around them



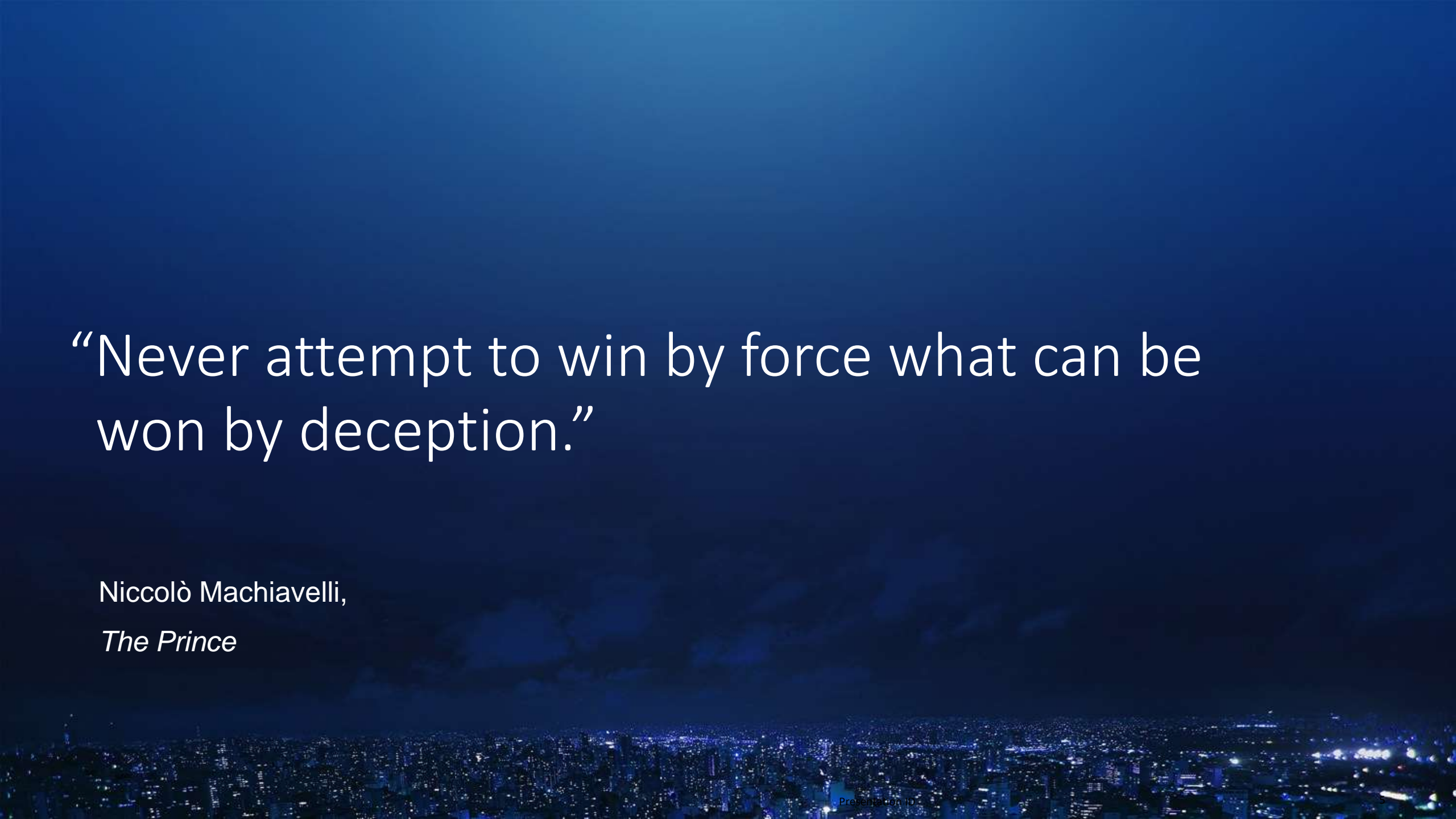
Stewart, Lucy C; - *Energetic Limitations Of Thermophilic Methanogens And Thiosulfate Reducers In The Subsurface Biosphere At Deep-Sea Hydrothermal Vents*. 2015



“...in my dream you play by my rules..”

Saito

*Inception (2010)*



“Never attempt to win by force what can be won by deception.”

Niccolò Machiavelli,

*The Prince*

# GASLIGHTING WITH HONEYPITS AND MIRAGES

DESTROYING DISCOVERY TO DEplete ATTACKERS

Catherine (Kate) Pearce

*Sr. Security Consultant, Cisco Security Services*



# PEOPLE - KATE



- Catherine (Kate) Pearce
  - @secvalve
- Sr. Security Consultant  
(Customer Focused) at Cisco
  - Break & report
  - Coach the builders
  - Research what's ahead
- Distinguishing Features:
  - Loud, Yellow
  - Or is that "Loud Yellow"?



# PEOPLE - KATE

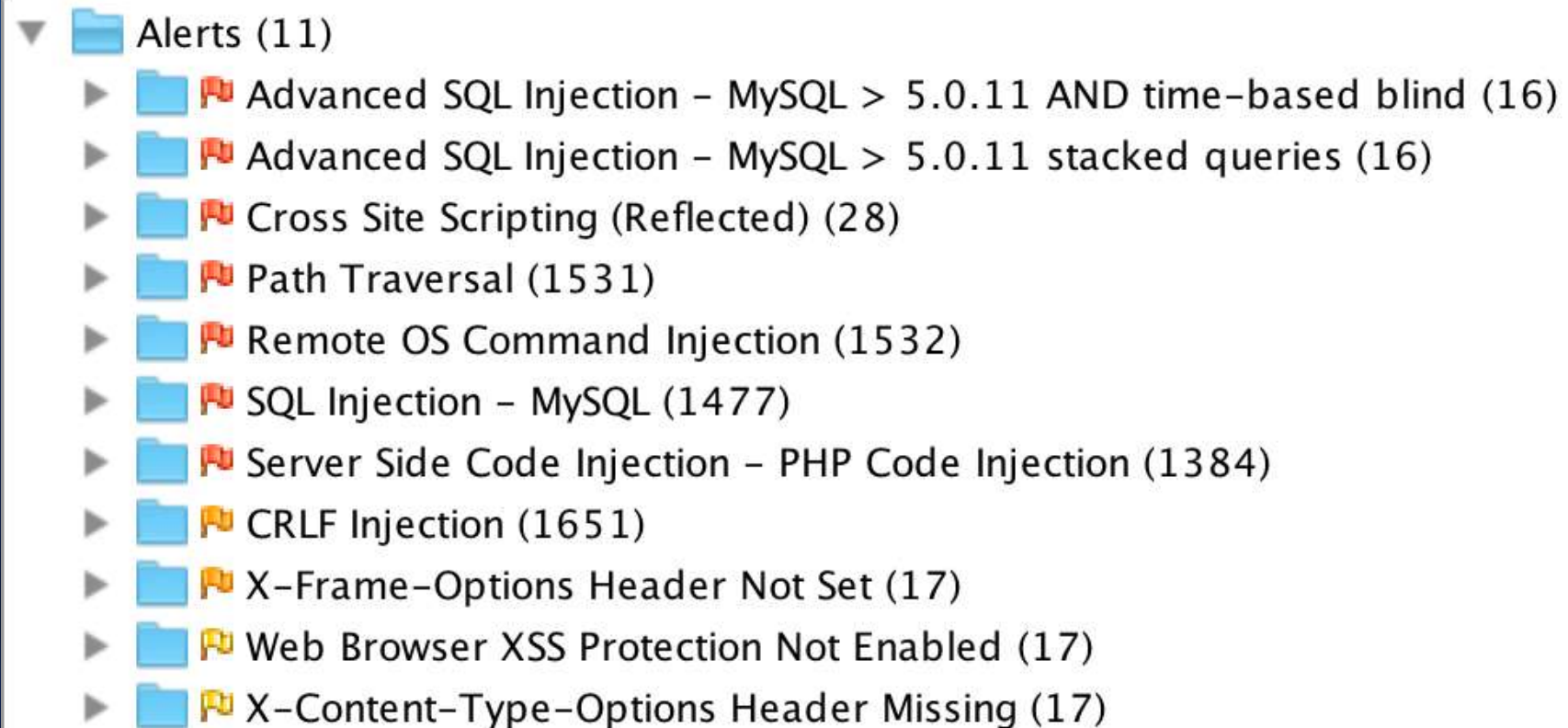
- Plays with fire, will never have a better photo taken in her life:





# IMAGINE ....

- Thousands of issues



IMAGINE ....

- Don't you hate false positives?












# IMAGINE ....

- More scanners,  
more problems

## Issues

- ▶  SQL injection [77]
- ▶  Cross-site scripting (reflected) [17]
- ▶  External service interaction (DNS) [55]
- ▶  External service interaction (HTTP) [54]
- ▶  Python code injection [77]
- ▶  XML injection [19]
  - i Frameable response (potential Clickjacking)
- ▶  Cross-site request forgery [5]

IMAGINE ....

- Can't exploit any of them?





IMAGINE ....

- Don't you hate Irreproducible vulns?



IMAGINE ....

- I made a reproducible vulnerability just for you





IMAGINE ....

- I made a reproducible vulnerability just for you
- **Nobody else gets it**



# INTRO

- This presentation makes me uncomfortable
- It should make you uncomfortable too
- There are some amazing frameworks for thinking about, planning, undertaking, measuring, and improving deception. I am not that.





# SECURITY ASYMMETRY

THE BALANCE TIPS BOTH WAYS

# SECURITY ASYMMETRY

- Resource Asymmetry
- Information Asymmetry
- Control Asymmetry
- Opportunity Asymmetry



# SECURITY ASYMMETRY

- “Defenders have one major advantage over the attacker. It is their castle.” – Monty McDougal, *Castle Warrior: Redefining 21st Century Network Defense*





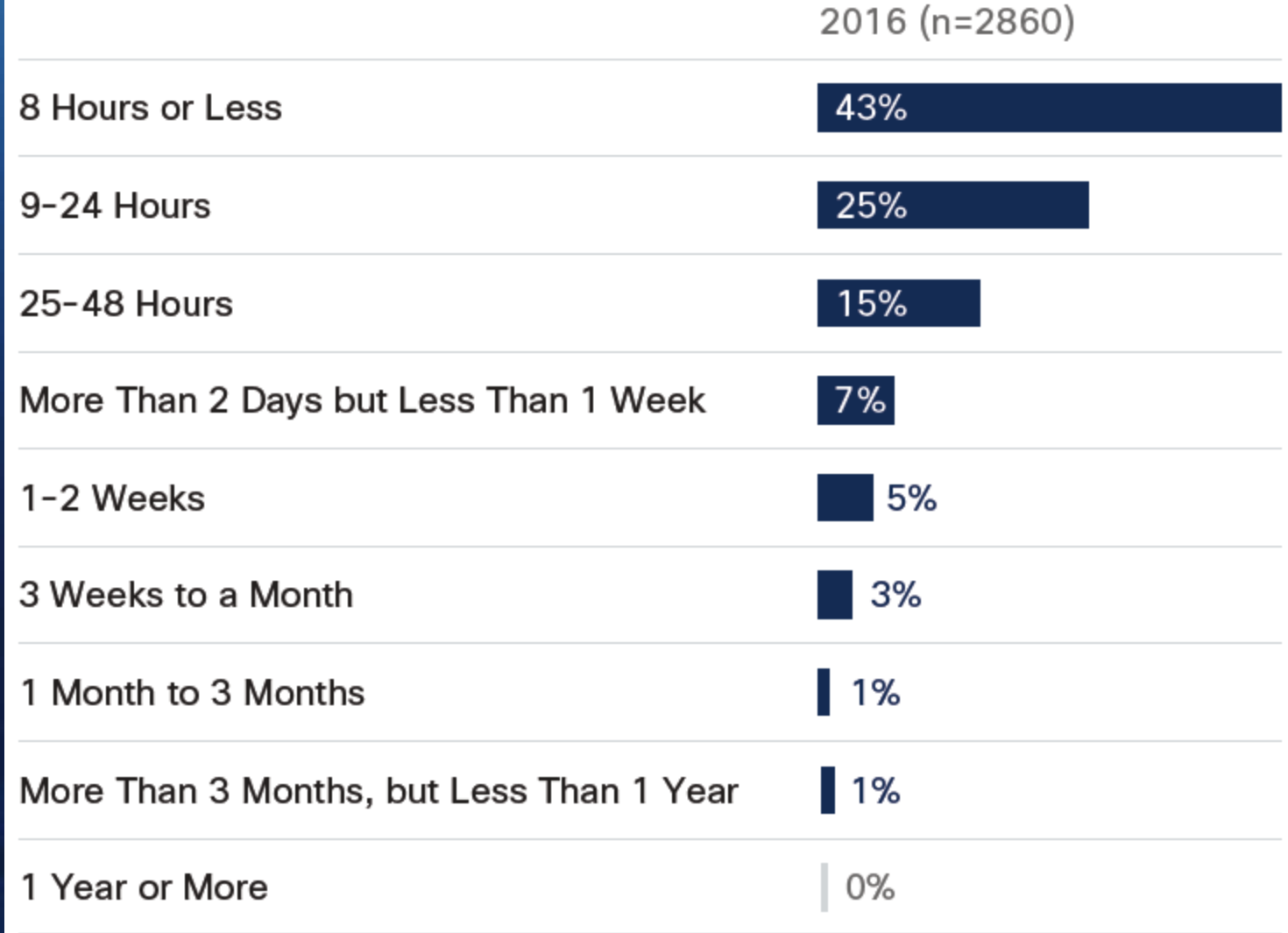
# THE DEFENDER'S ADVANTAGE

YES. I'M SERIOUS

# DOES THIS SOUND FAMILIAR?

- An attacker only needs to find one way in, but the defender needs to protect every possibility

**Figure 101 Average Time to Detect Security Breaches**



If it takes until the morning to find the ruse, it is too late

Source: Cisco 2017 Security Capabilities Benchmark Study



But... what if  
we can find it  
before the  
morning?

Time to  
Compromise

Time to  
Exfiltrate



Verizon DBIR 2016

Or slow attackers until  
the morning

## Time to Exfiltrate



# HOSTILE NETWORKS

- If the enemy is coming to get you on your own terrain use your knowledge against them
- If the enemy is coming to meet you in your own reality, use your mastery of it against them



“...in my dream you play by my rules..”

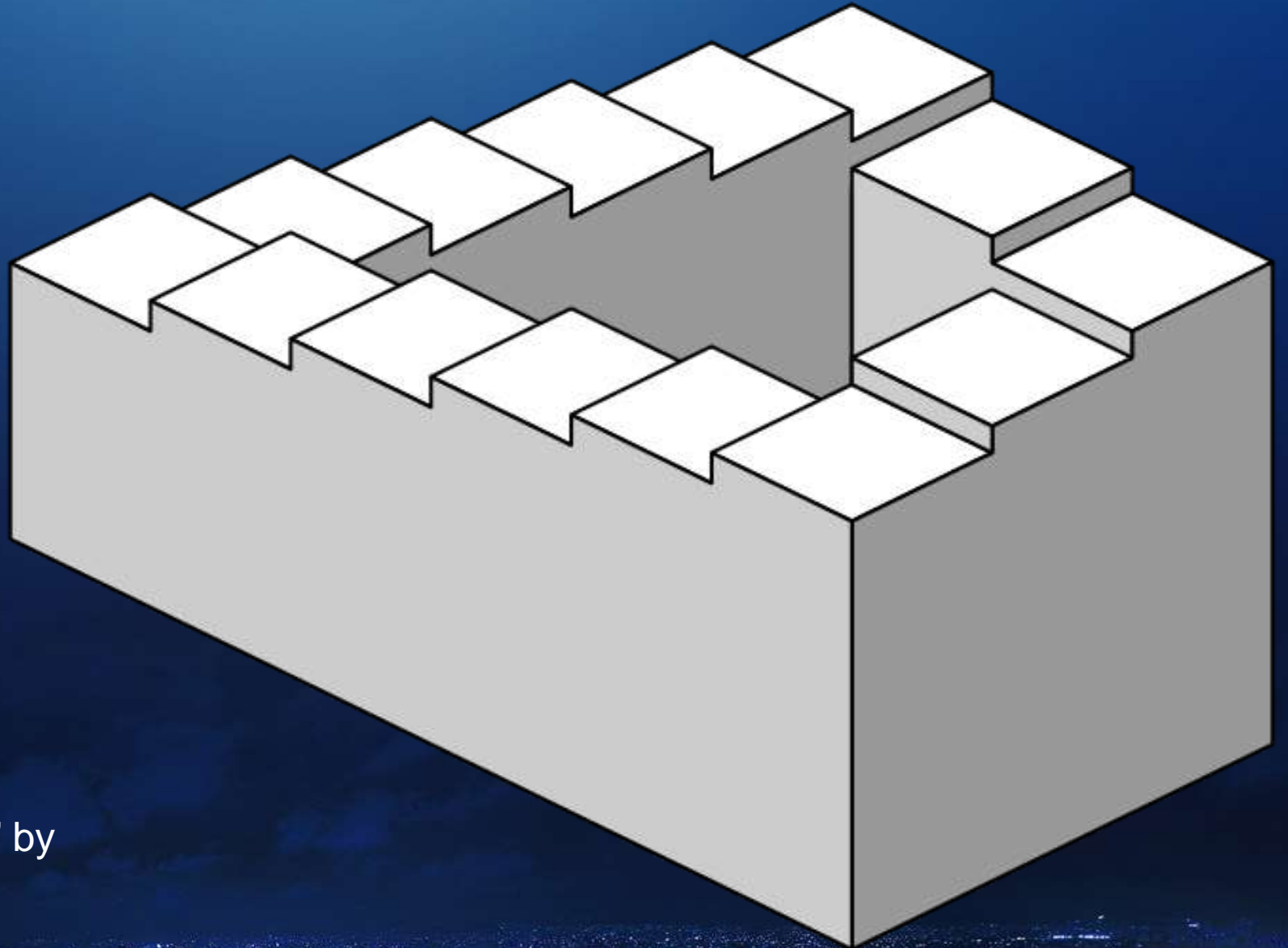
Saito

*Inception (2010)*

The background of the slide is a night-time photograph of a city skyline, densely packed with lights from buildings and streets. A dark blue gradient is applied over the image, starting from a lighter blue at the top and fading into a deep navy blue towards the bottom, where the city lights are most visible.

How do you attack a system that makes no sense?

# *Give Attackers Penrose Stairs*



“A "Penrose stairs" optical illusion” by  
Sakurambo



The background of the slide is a night cityscape. The top half of the image is a solid dark blue gradient. The bottom half shows a city skyline at night, with numerous lights from buildings and streets. The text is centered in the middle of the image.

How do you attack a system that never does what you expect

# DECEPTION OVERVIEW

BECAUSE IT'S ALL FUN AND GAMES WHEN YOU'RE THE ONE WINNING

# DECEPTION OVERVIEW

- Offensive Deception
  - E.g. Ambushes
- Defensive Deception
  - E.g. Traps





## Sun TZU

- All warfare is based on deception.
- Hold out baits to entice the enemy. Feign disorder, and crush him.
- Pretend inferiority and encourage his arrogance.
- Lure with bait; strike with chaos.

## Clausewitz

- "War is . . . an act of force to compel our enemy to do our will."
- Many intelligence reports in war are contradictory; even more are false, and most are uncertain....
- Disarming the enemy – Making them sacrifice (your forced conditions more acceptable than the sacrifice if they don't)
- Control the use of force

For a good discussion see Michael Handel - *Sun Tzu And Clausewitz: The Art Of War And On War Compared*  
<http://www.dtic.mil/dtic/tr/fulltext/u2/a239084.pdf>

# DECEPTION ELSEWHERE

EVERYBODY'S DOING IT

# DECEPTION EXAMPLES - OTHER

- Flares and Chaff
- False Flags
- Decoys
- Friend or Foe
- Uniforms
- Flags
- Smoke Generators
- Intel and Counter intel
- Canaries
- Anonymity
- Surveillance and sigint
- Double agents
- Radar detectors
- Radar detector detectors (X n)
- Counterintel, Counterinformation





# DECEPTION EXAMPLES – INFORMATION SECURITY

- Attack

- DDOS
- Fake AV
- Ransomware
- Trojans
- Slowloris
- Spoofing
- False Flag

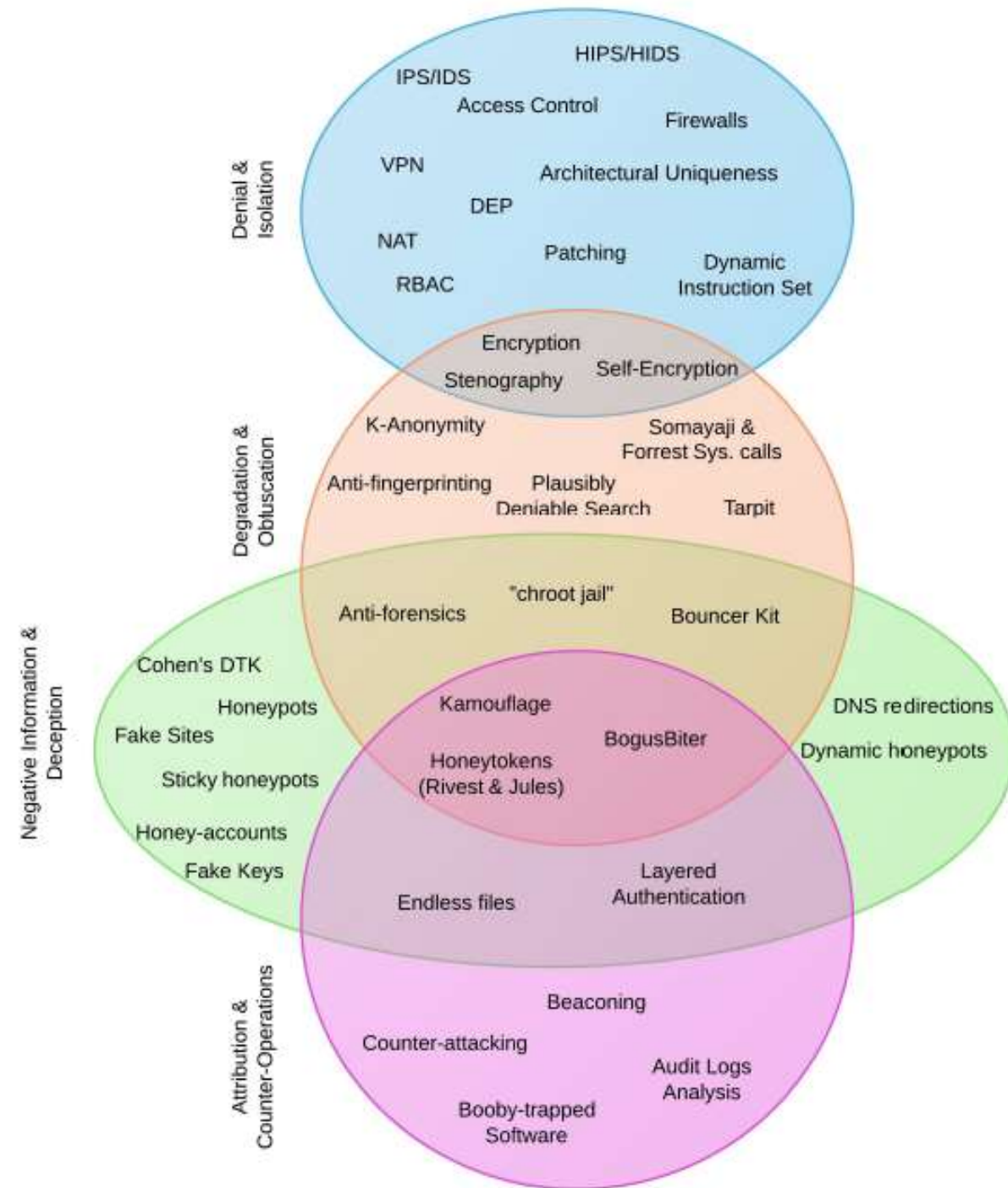
- Defence

- ASLR
- Moving Target Defences
- Canaries, honeypots
- Red Teaming
- Fake banners/headers
- Labrynth
- Honeypatches

- More to be discussed later

# DECEPTION EXAMPLES – INFORMATION SECURITY

- From Almeshekah



# DECEPTION EXAMPLE - BANNING AND SHADOWBANNING

- *“...blocking a user ...in such a way that the user does not realise that they have been banned.*

*If the user never becomes aware that they were banned, it will not occur to them to attempt to circumvent that ban.”* - [https://en.wikipedia.org/wiki/Stealth\\_banning](https://en.wikipedia.org/wiki/Stealth_banning)

- Similar to comment ghosting





# DECEPTION GOALS

IT'S NOT JUST SPORT, IT'S A CAREER SKILL

# DECEPTION GOALS

- Gain intel
- Waste Resources
- Plant False Intel
- Waste Time
- Burn Credibility
- Burn Sanity



# DEFENSIVE DECEPTION

THESE ARE NOT THE NETWORKS YOU ARE LOOKING FOR



# THE 7 D'S OF DEFENSIVE DECEPTION

Distract

Deter

Deny

Delay

Demoralize

[Mis]Direct

Devalue



# ATTACK CHAIN

EACH STEP OFFERS A PLACE TO MESS WITH THE ATTACKER

1. [Scan potential target area]
2. Identify Targets
3. Categorise Targets
4. [Scan target surfaces]
5. Identify surfaces
6. [Categorize Surfaces]
7. [Build hypotheses]
8. Probe target points (Test Hypotheses)
9. Assess Probes (Successful entry, y/n?)
- 10.[Payload]
- 11.[Post Exploit]
- 12.Desired outcome?
- 13.Next steps

# POSSIBILITIES

(IN)FINITE STATE

# POSSIBILITIES

- Attackers need to decide
  - What can I see?
  - What seems vulnerable?
  - What is actually exploitable?





# POSSIBILITIES - "TRUTH" TABLE

True/False – Is the result correct or not?  
Positive/Negative – What was the result?

True?	Result		True	
True	Positive	True Positive	Issue Found	Vulnerable
	Negative	True Negative	There is no issue	Secure state
False	Positive	False Positive	Found Non-existent issue	Annoys Everyone
	Negative	False Negative	Issue Unidentified	"Mitigated"

# POSSIBILITIES - DECEPTION

But what if somebody is actively messing with your assessments?

The defender uses their control/visibility to affect the attacker's effectiveness

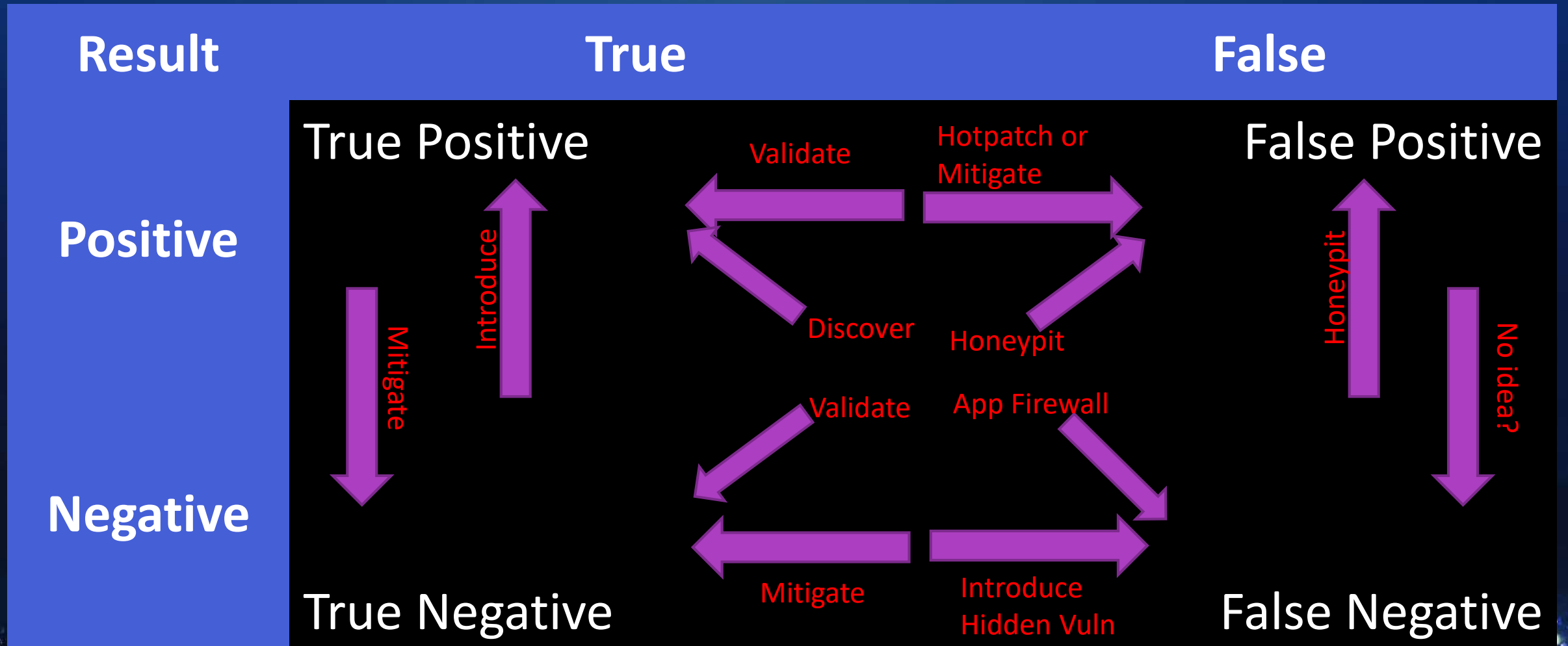


# POSSIBILITIES - “TRUTH”/REALITY TABLE

True/False – Is the result correct or not?  
Positive/Negative – What was the result?  
Real/Fake – Are you assessing the actual state or a deceptive one

Real?	Correct?	Result	True
Real	True	Positive	Real True Positive
		Negative	Real True Negative
	False	Positive	Real False Positive
		Negative	Real False Negative
Fake	True	Positive	Fake True Positive
		Negative	Fake True Negative
	False	Positive	Fake False Positive
		Negative	Fake False Negative

# POSSIBILTIES - TRANSITIONS





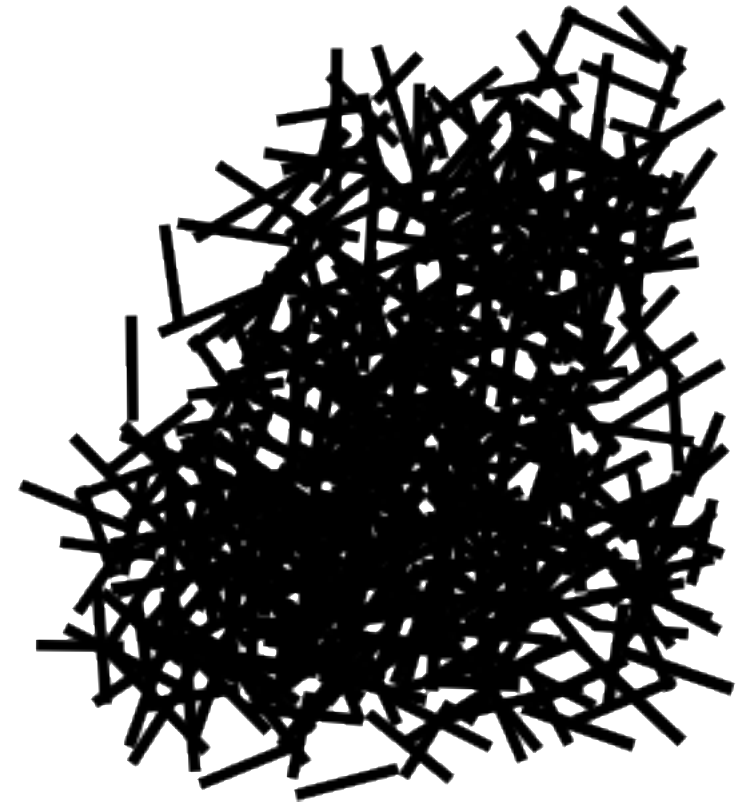
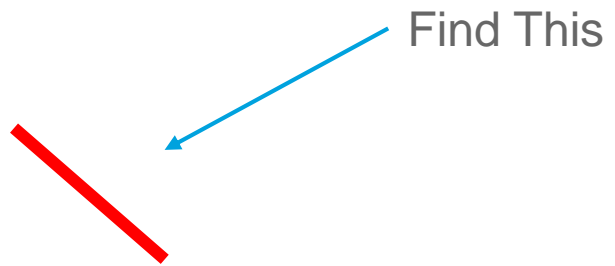
# PROBABILITIES

HOW SURE AREN'T YOU?

# (PRIOR) PROBABILITIES

- Attackers need to know:
  - What is the chance something can be found?
  - What is the chance it is exploitable?
  - What is the chance that a reading is true?

# Needles, Haystacks



# Needles, Haystacks

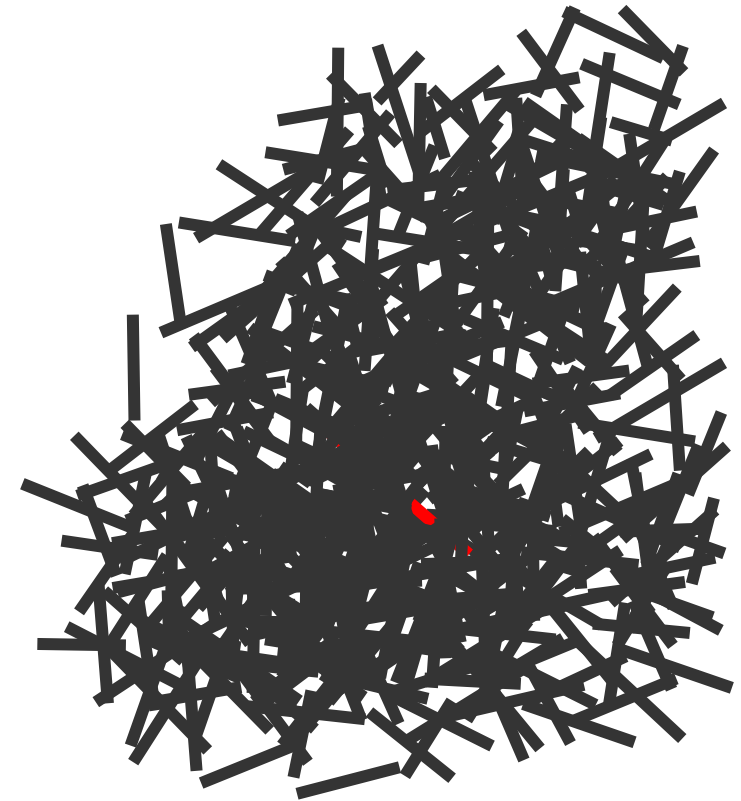
- Chance of finding randomly =

*NumRed*

---

*NumBlack*

*= 1 / 600*

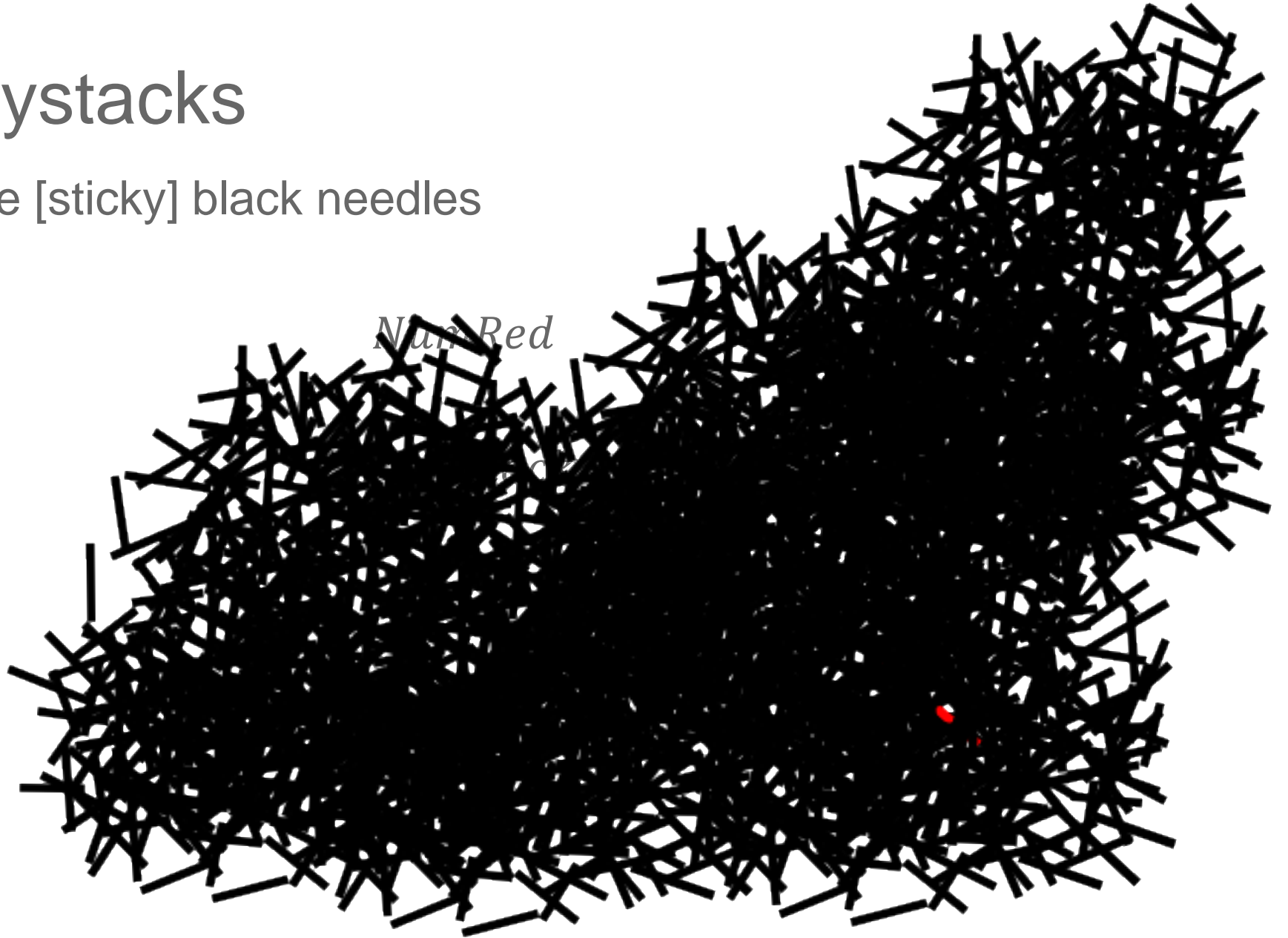




# Needles, Haystacks

- A tarpit adds more [sticky] black needles

$= 1 / 4000$



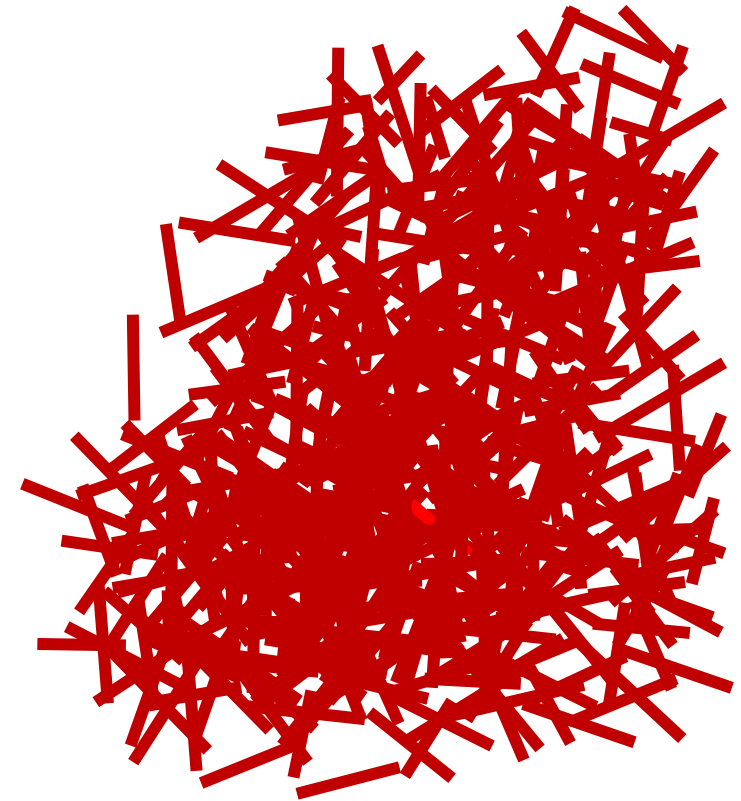
# Needles, Haystacks

- A Honeypot adds fake red needles
- $= (\text{Real} / \text{Fake}) * 1/600$
- $= 1/20 * 1/600$
- $= 1/12000$



# Needles, Haystacks

- A honeypit turns all black needles dark red
- $= (\text{Real} / \text{Fake}) * 1/600$
- $= 1/600 * 1/600$
- $= 1/360\ 000$



# NEEDLES, HAYSTACKS

- Tar pits –  $O(n)$  increase in complexity
- Honeypots –  $O(n^2)$  ?
- Honeypits –  $O(n!)$  ?
- My algorithmic efficiency is rusty and probably wrong.



# MENTAL MODELS

NOPE. YOU MAKE THE JOKE FOR ME.

# THEORY OF MIND

- Simple attackers don't have a theory of mind
- Some do

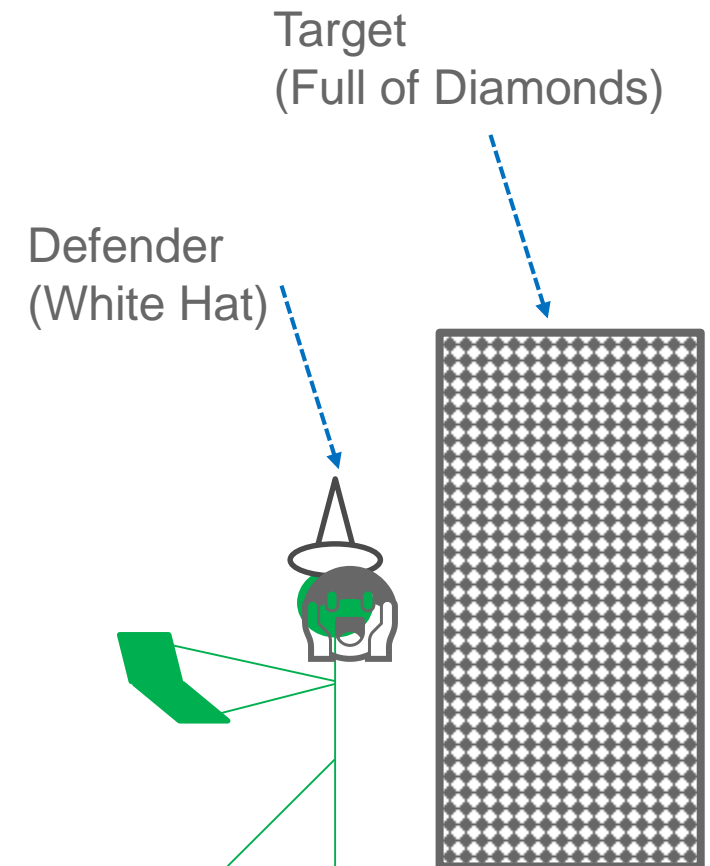
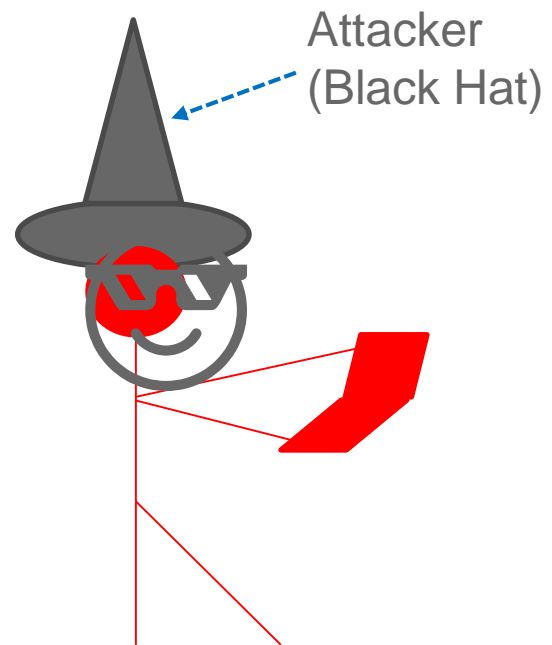


# MENTAL MODELS

- Attacker Models Target
  - Attacker Models Defender
  - Attacker Models Attacker
  - Defender Models Defender
  - Defender Models Attacker
  - Defender Models Target to Defend
- 
- Either mess with the attacker's read of the world, model of themselves, model of the target, model of the defender, or model of themselves!

# Mental models

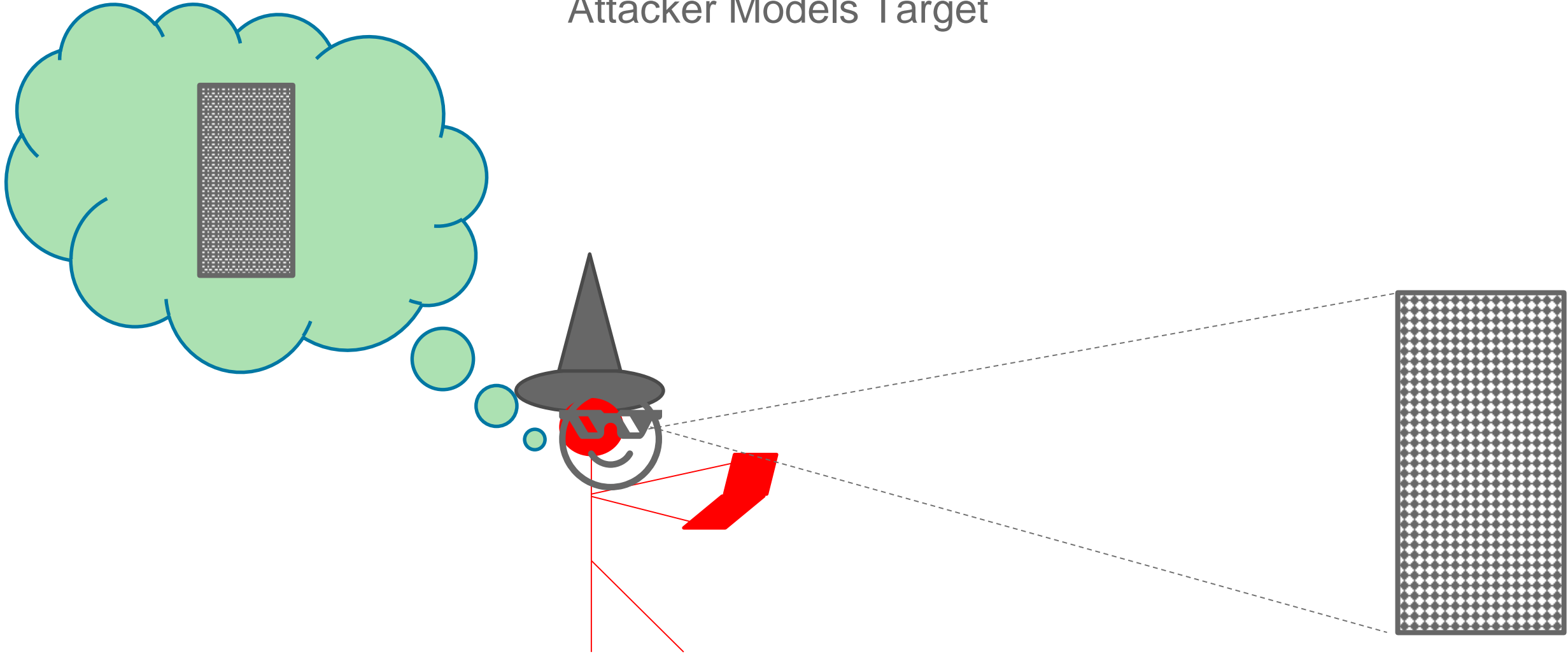
“Stakeholders”





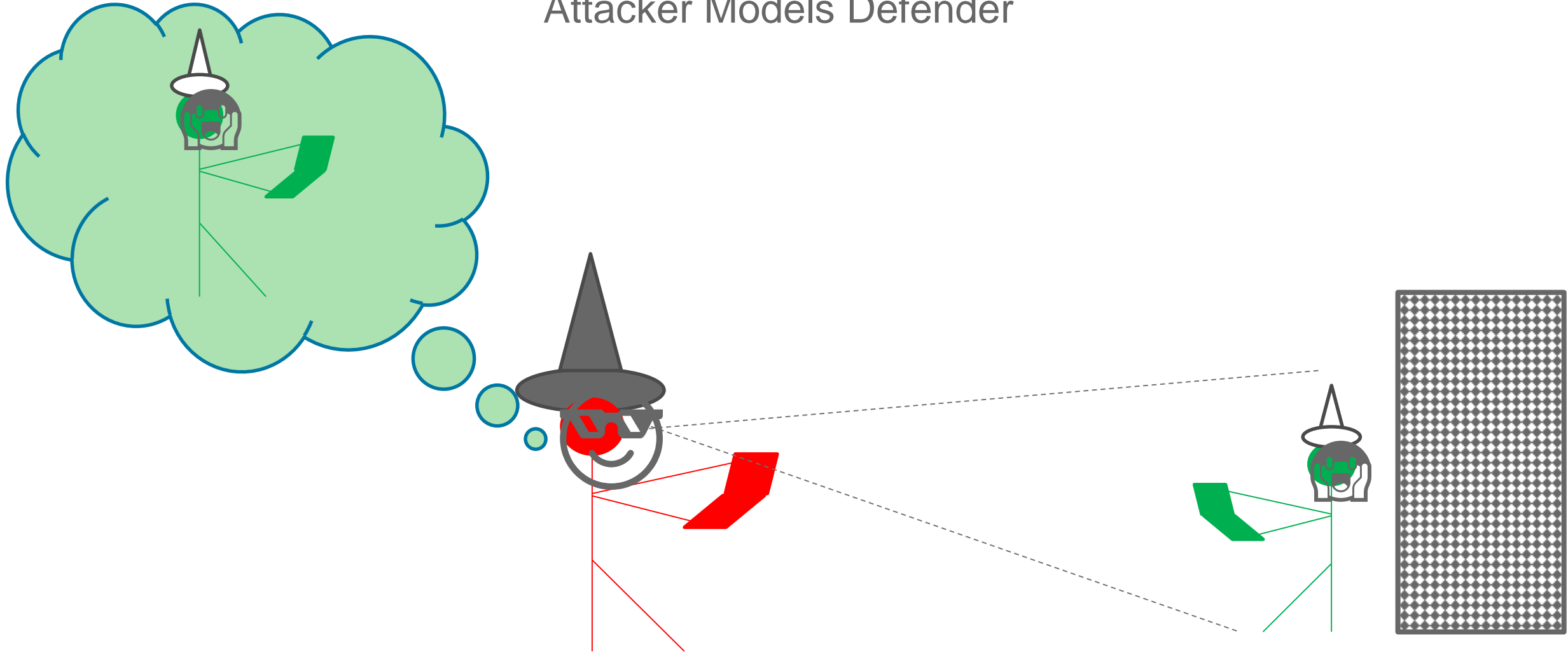
# Mental models

Attacker Models Target

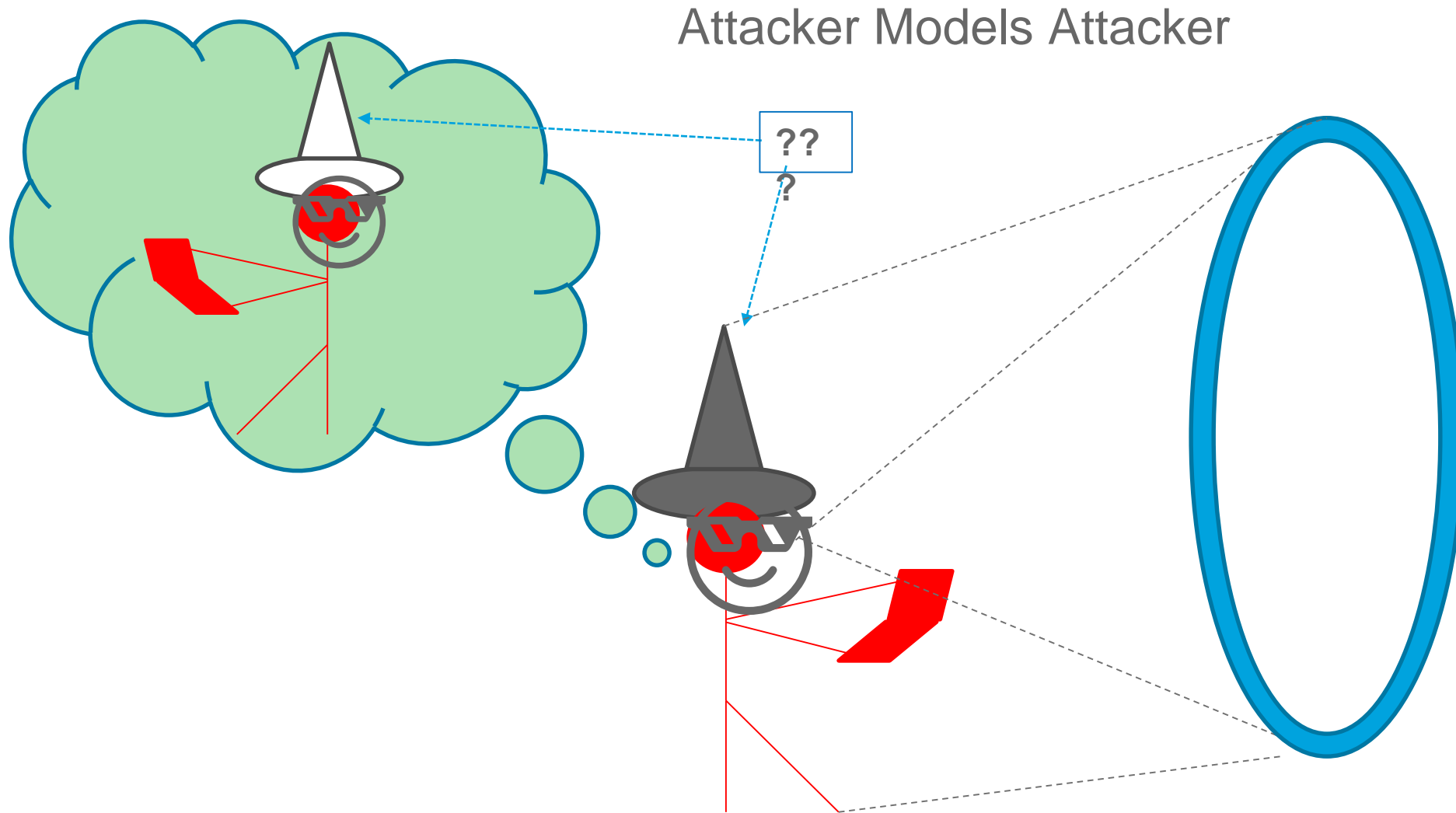


# Mental models

Attacker Models Defender

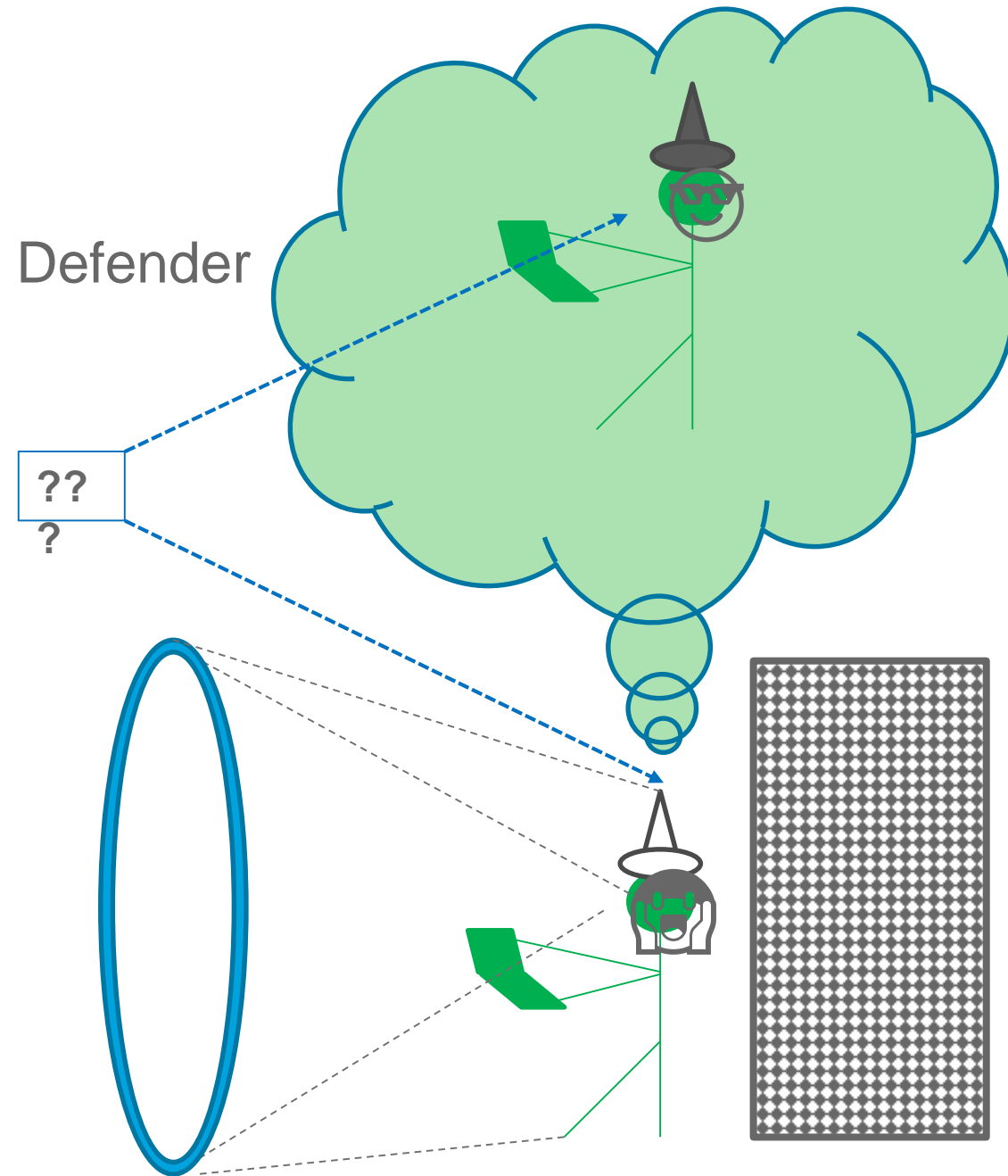


# Mental models



# Mental models

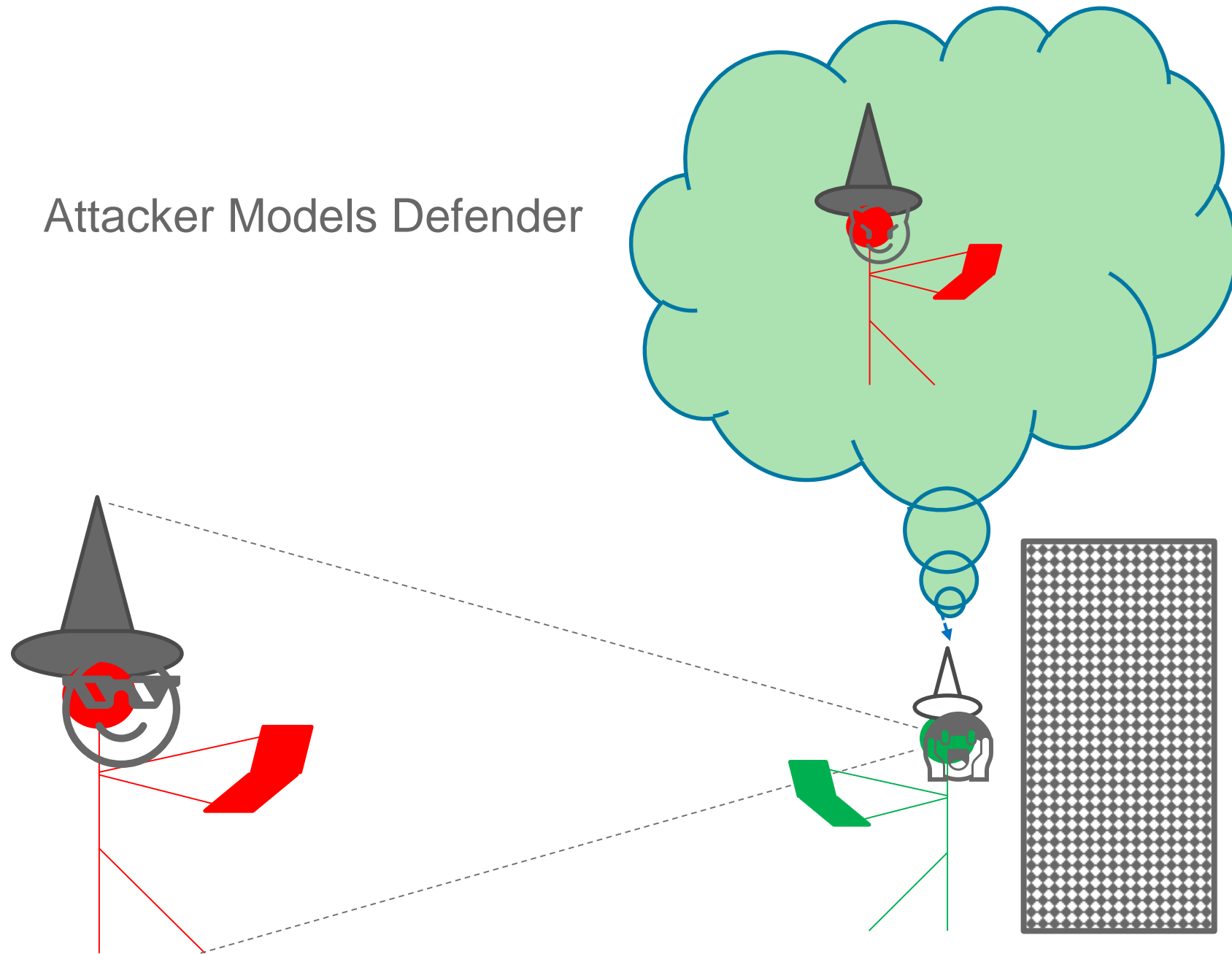
- Defender Models Defender





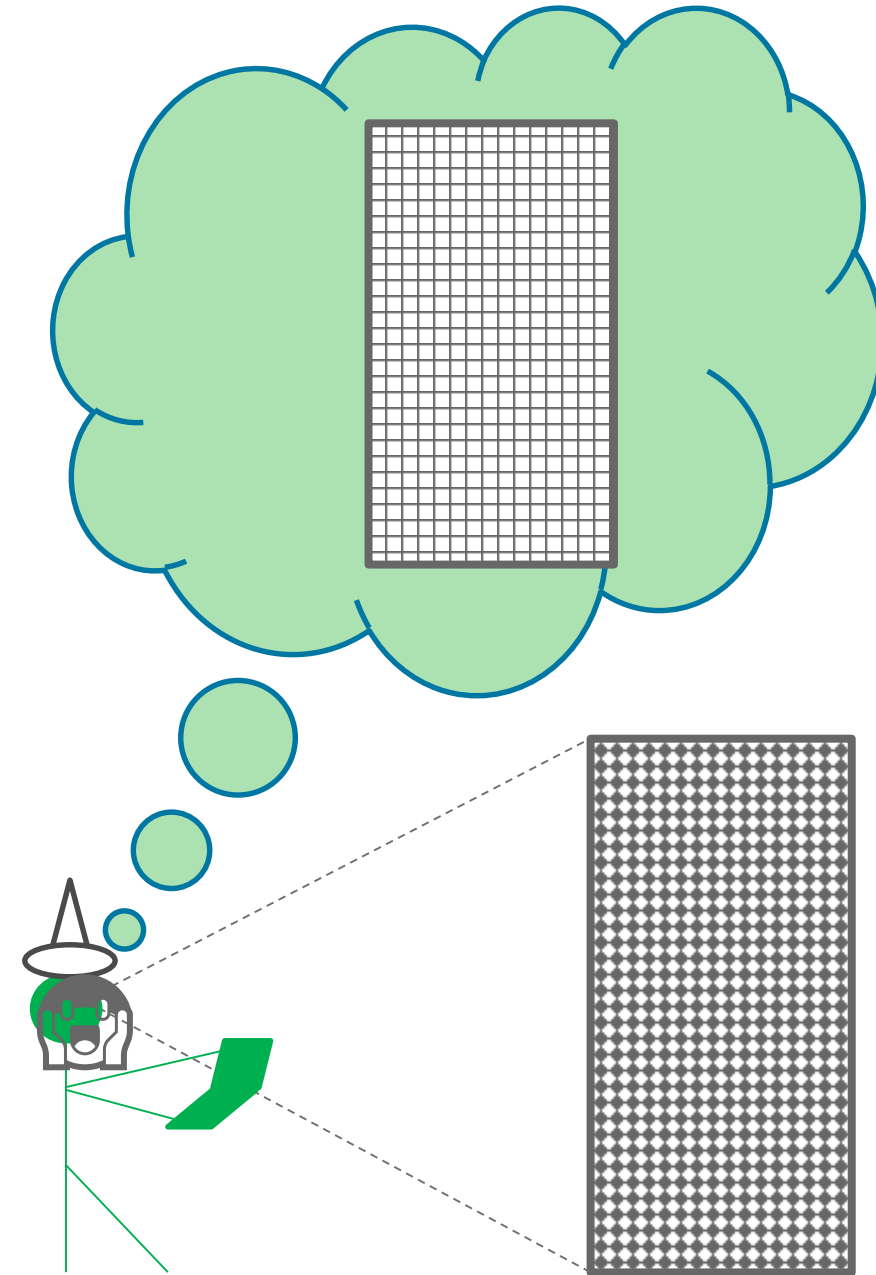
# Mental models

Attacker Models Defender



# Mental models

Defender Models Target



# SENSING

MURDER? NOT SO MUCH

# SENSING

- The world is a black box?  
What you did  
What answered you  
[Hypothesis] what the answer means
- Joke.





# DECISION MAKING

DECIDE ON A JOKE TO GO HERE?

# DECISION MAKING

- Decisions are made on the basis of some input
  - From Memory
  - From Sensing

# MEMORY

I FORGET WHAT WE WERE GOING TO PUT HERE

# SENSING

- Memory is what we use to link the sensed together to form a model

LAYERS OF DECEPTION

LAYERS OF DECEPTION

LAYERS OF DECEPTION

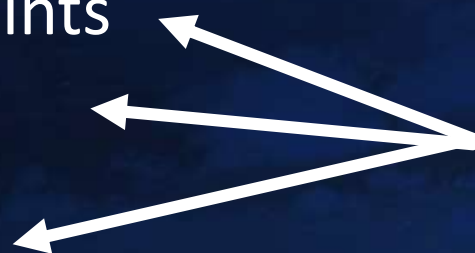
LAYERS OF DECEPTION



# THE DECEPTION STACK

- Networks
- Systems
- Protocols
- Ports
- Services
- Applications
- Application Points
- Vulnerabilities
- Data

For now we're only  
looking at these



# TOOLS AND TECHNIQUES

PROCEDURES ARE FOR POLICE TV SHOWS AFTER ALL

# EXISTING

- Tarpits
- Honeypots, anti-honeypots, anti-anti-honeypots
- Canaries
- Port Knocking
- Embedded honeypots
- Deception technology, Mykonos

- ADHD -
  - Portspooof -
  - Labrea -
  - Conpot -
  - Dionaeea -
  - Thug -
  - Glastopf, snare, tanner (<http://mushmush.org/>)
- 
- Kippo et al

# DECOYS

LOOK, THEY'RE BEHIND YOU



# DECOYS

- We made some decoys
  - Data Decoys
  - Defined Response
  - Behavioural Decoys
  - Dummy Parameters

# DATA DECOYS

- Requested files
  - Fake files
    - /etc/shadowbanned
    - C:\windows\win.ini
- Direct Object reference:
  - If it's not yours return a dummy one that looks real

# VULN 0 – PATH TRAVERSAL

```
//vuln 0
if (get_vuln_enabled($vuln_str,0)){
    //echo "<br>1 – path traversal enabled for: " . $key . "<br>";
    //Path issues
    $passwd_file = "cat /etc/passwd\nroot:x:0:0:root:/root:/bin/bash\nbin:x:1:1:bin:/b
    $win_ini_file = "; for 16-bit app support\n[fonts]\n[extensions]\n[mci extensions]
    $web_xml_file = "<web-app xmlns=\"http://java.sun.com/xml/ns/javaee\" version=\"2.

    if (preg_match("~.*etc.*passwd.*~i",$value) ){
        echo $passwd_file;
    }
    if (preg_match("~.*win.ini.*~i",$value) ){
        echo $win_ini_file;
    }
    if (preg_match("~.*web.xml.*~i",$value) ){
        echo $web_xml_file;
    }
}
```

# VULN 1 – COMMAND INJECTION

```
//vuln 1
if (get_vuln_enabled($vuln_str,1)){
    //echo "<br>2 - command injection enabled for: " . $key . "<br>";

    //command injection
    $ls_slash = "bin\tetc\tlib\tmedia\tproc\tsbin\tsys\tvar\nboot\thome\tlib64\tmnt\troot\tsnap\ttmp\tvml
    $dir_c = "C:\>dir\n
    Volume in drive C has no label.\n
    Volume Serial Number is F4AC-9851\n

    Directory of C:\\n

    09/02/2015  12:41 PM      <DIR>          SysReset\n05/30/2016  06:22 PM          93 HaxLogs.txt\n
    if (preg_match("~.*ping.*127\.0\.0\.1.*~i",$value) ){
        sleep(5);
    }
    if (preg_match("~.*ls\ \/*~i",$value) || preg_match("~.*ls\ \\*~i",$value)){
        echo $ls_slash;
    }
    if (preg_match("~.*dir\ \/*~i",$value) || preg_match("~.*dir\ \\*~i",$value) ){
        echo $dir_c;
    }
    if (preg_match("~.*cmd\%3D\%22dir\+\%5C.*~i",$value)){
        echo $dir_c;
    }
}
```

# VULNERABILITY DECOYS

- Defined Responses
  - SQLi
  - XSS



# VULN 6 – SQLI

```
//vuln 6
if (get_vuln_enabled($vuln_str,6)){
    //echo "<br>6 - SQL injection enabled for: " . $key . "<br>";

    //SQL error processing
    if (preg_match("~.*sleep\(20\).*~i",$value) ){
        sleep(20);
    }
    if (preg_match("~.*sleep\(5\).*~i",$value) ){
        sleep(5);
    }

    $numdbchars = substr_count ($value ,"'");
    //echo $numdbchars;
    if ($numdbchars > 0 && $numdbchars % 2 == 1){
        echo "http.StatusText(500) org.gjt.mm.mysql com.mysql.jdbc.exceptions The used SELECT
    }
}
```

# VULN 8 – ZAP PHP INJECTION

```
//vuln 8
if (get_vuln_enabled($vuln_str,8)){
    //echo "<br>7 - PHP injection enabled for: " . $key . "<br>";

    //php injection
    $zap_token = "zap_token";
    //$zap_token_php =
    "chr(122).chr(97).chr(112).chr(95).chr(116).chr(111).chr(107).chr(101).chr(110)";
        if
    (preg_match("~.*chr\(122\)\.chr\(97\)\.chr\(112\)\.chr\(95\)\.chr\(116\)\.chr\(111\)\.chr\
    (107\)\.chr\(101\)\.chr\(110\).*~i",$value)){
        echo $zap_token;
    }
}
```

# VULN 10 – ZAP CRLF

```
//vuln 10
if (get_vuln_enabled($vuln_str,10)){
    //echo "<br>10 - CRLF/Cookie injection enabled for: " . $key . "<br>";

    //CRLF injection

    //Set-cookie%3A+Tamper%3Dd7394596-c91f-4519-a3a2-bd46b9457878
    $num_matches =
preg_match_all("~Set-cookie%3A+Tamper%3D[A-F0-9]{8}(:-[A-F0-9]{4}){3}-[A-F0-9]{12}~i",$
value,$match_cookie_1);
    //echo $match_cookie_1[0];
    if($num_matches > 0){
        header($match[0][0]);
    }
    $num_matches = preg_match_all("~Set-cookie:
Tamper=[A-F0-9]{8}(:-[A-F0-9]{4}){3}-[A-F0-9]{12}~i",$value,$match);
    if($num_matches > 0){
        header($match[0][0]);
    }
}
}
```

# DECOYS – DUMMY PARAMETERS

- Doesn't actually do anything except look vulnerable with decoys

```
<?php
    //insert honey params

    for( $i = 0; $i<$numRandomParams; $i++ ) {

        $randstr_val = random_str(32);
        $randstr_name = random_str(12);
        echo "<input type=\"hidden\" name=\"";
        echo $randstr_name;
        echo "\"value=\"";
        echo $randstr_val;
        echo "\"/>";

    }
?>
```

# DECOYS – UNCRACKABLE HASHES

- Static
  - Easy
- Generated
  - MD5(dev/urandom)



# DECOYS – WHITELIST BENIGN VULNERABILITIES

- Let it pass if it's harmless

# VULN 7 – XSS

```
//vuln 7
if (get_vuln_enabled($vuln_str,7)){
    //echo "<br>7 - XSS injection enabled for: " . $key . "<br>";

    //xss handling
    if (preg_match("~.*0W45pz4p.*~i",$value)){
        echo "0W45pz4p";
    }
    if (preg_match("~.*\<script\>alert\\(1\)\\;\<\/script\>.*~i",$value)){
        echo "<script>alert(1);</script>";
    }
    if (preg_match("~.*\%3Cscript\%3Ealert\%281\%29\%3B\%3C\%2Fscript\%3E.*~i",$value)){
        echo "<script>alert(1);</script>";
    }
}
```

# DECOYS - VULNERABILITY / EXPLOIT EMULATION

- Play Along

# VULN 3 – BURP COLLABORATOR HTTP GET

```
//vuln 3
if (get_vuln_enabled($vuln_str,3)){
    //echo "<br>3 - Burp Collaborator get file Enabled for: " . $key . "<br>";

    //burp collaborator get file
    if (preg_match("~.*\.burpcollaborator\.net.*~i",$value) ){
        error_log("Found a burpcollaborator url in: " . $value);
        preg_match_all('~https?:\/\/(www\.)?[0-9a-z]{2,64}\.burpcollaborator\.net~', $value, $match);
        if($match[0]){
            error_log("matching on" + $match[0][0]);
            file_get_contents($match[0][0]);
        }
    }
}
```

# DECOYS - VULNERABILITY / EXPLOIT PASSTHROUGH

- SQL Injection vulnerable parameters
- But they go to a dummy DB with Dummy Data
- “Mirages”

# HONEYPITS AND MIRAGES

STICKY SUGARY BADNESS



# MIRAGES

Mirages occur when light is bent, showing things different to where they are



# MIRAGES



# MIRAGES

- Things that only exist till you reach them
- Different to a pure decoy,
  - A decoy is *believable*
  - A mirage becomes *less believable the closer you get*

# HONEYPITS

- Honeypot
- Tarpit
- Portmanteau
- Sweet sweet vulnerabilities that you get bogged down in





**IMAGINING  
ICE AGE L.A.**

During the last Ice Age, the Los Angeles region was home to a variety of prehistoric animals, including mammoths, bison, and horses. These animals roamed the area for thousands of years before the arrival of humans. The exhibit features large-scale sculptures of these animals, allowing visitors to see what life might have been like in Ice Age L.A.

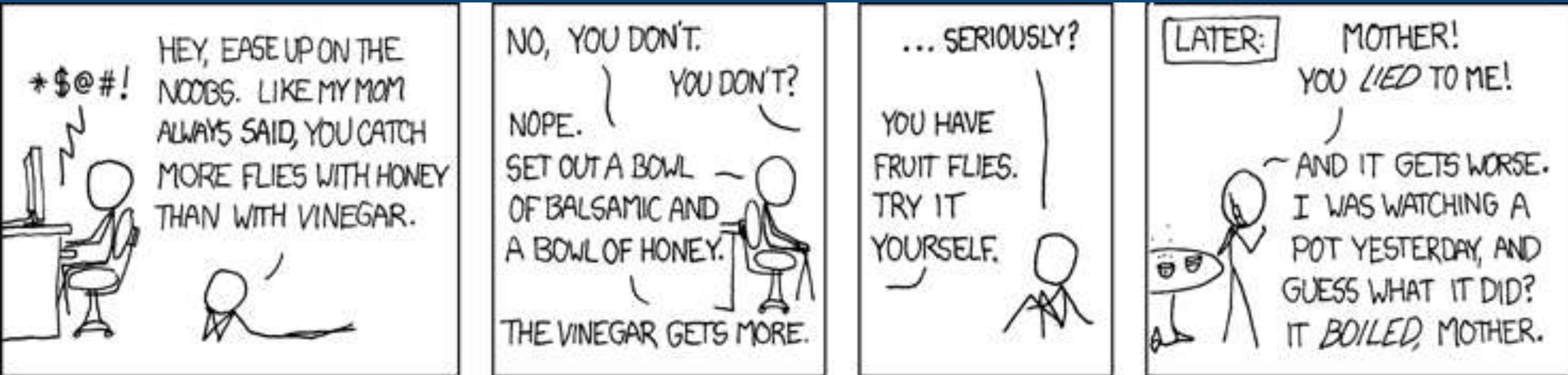








# HONEYPIT



<https://xkcd.com/357/>

# HONEYPITS AND MIRAGES

- Mirages – It's there until you reach it
- Honeypits – There's more there than you could ever deal with

# HONEYPIT LOGIC FLOW (CURRENT)

- Assess input type:
  - Reflect
  - Defined Response
    - String
    - File Contents
  - Transformed Response
    - Encoded/Decoded
  - Emulated Response
    - Timing
    - Logic

# HONEYPITS





















- Static
  - Decoys
- Whitelist
  - Safe only
- Triggered
  - FYO
- Redirection
  - Error Handling
- Emulation

Dancing with the  
langsec devil....

# HONEYPITS

- Burp of DVWA unmodified

## Issues

- ▶  SQL injection [4]
- ▶  File path manipulation [4]
- ▶  Cross-site scripting (stored)
- ▶  Cross-site scripting (reflected) [16]
- ▶  Cleartext submission of password [4]
- ▶  External service interaction (DNS)
- ▶  OS command injection
- ▶  File path traversal
- ▶  Session token in URL [20]
- ▶  Password submitted using GET method [3]
- ▶  Cross-site request forgery [4]
- ▶  Cookie without HttpOnly flag set
- ▶  Cross-domain Referer leakage [8]
- ▶  Cross-domain script include
- ▶  File upload functionality
- ▶  Private IP addresses disclosed
- ▶  Path-relative style sheet import [15]
- ▶  User agent-dependent response
- ▶  Frameable response (potential Clickjacking) [16]
- ▶  Directory listing

# HONEYPITS

- **Level 0 - Everything has same number of issues**



# DVWA

- Burp, all vulnerabilities injected on all params
- Some performance issues. This took over 20 times as long to scan...

## Issues

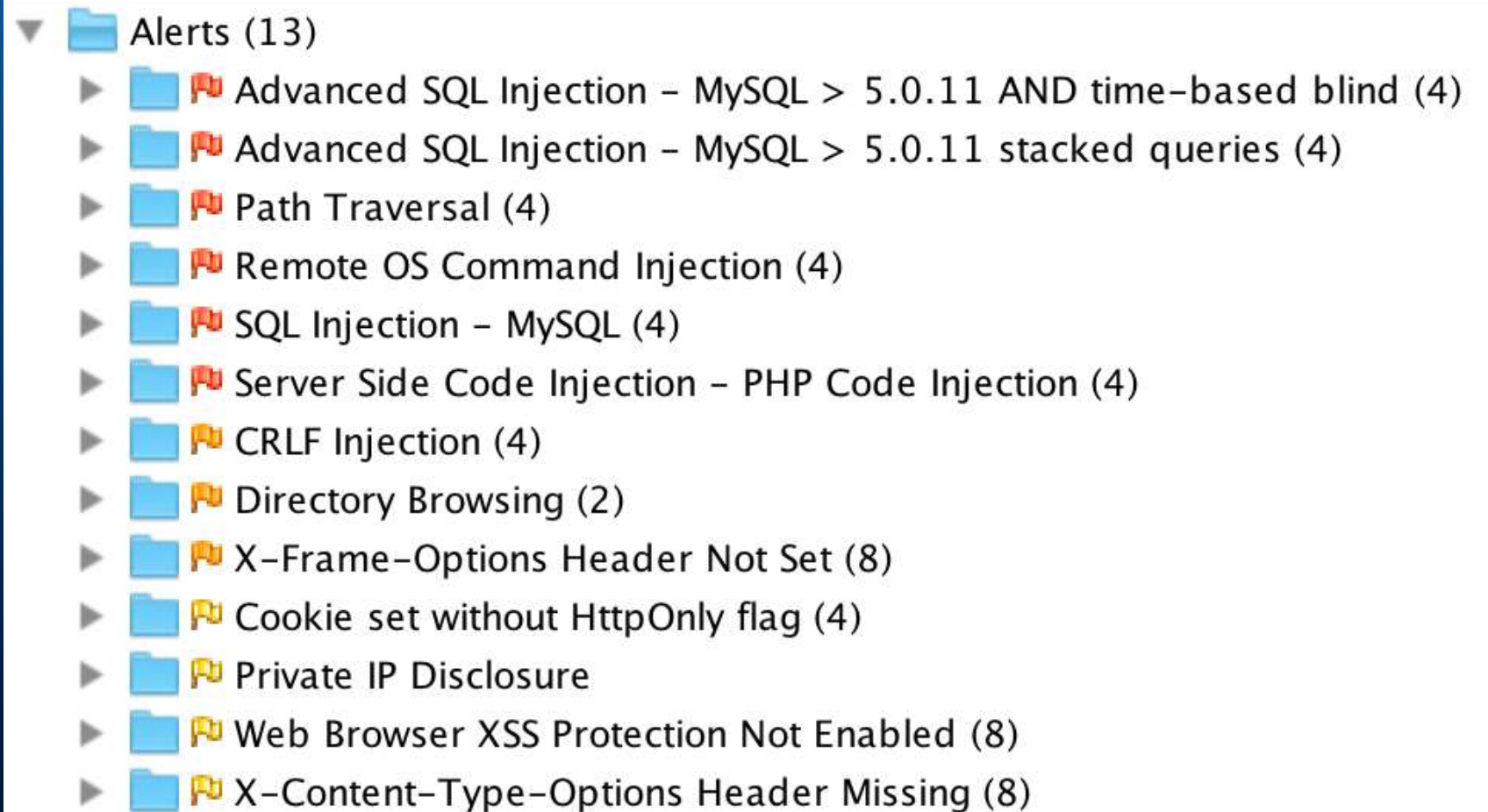
- ❗ OS command injection
- ▶ ❗ SQL injection [26]
- ❗ File path manipulation
- ▶ ❗ Cleartext submission of password [3]
- ▶ ❗ External service interaction (DNS) [71]
- ▶ ❗ External service interaction (HTTP) [71]
- ▶ ! Python code injection [22]
- ▶ ⚠ XML injection [23]
- ▶ ⚠ Cross-site scripting (reflected) [4]
- ? XPath injection
- ▶ ⚠ Password submitted using GET method [2]
- ▶ i Cross-domain Referer leakage [3]
- i Cross-domain script include
- i Private IP addresses disclosed
- ▶ i Path-relative style sheet import [6]
- ▶ i Frameable response (potential Clickjacking) [8]
- i Directory listing

# HONEYPITS

- **Level 0 - Everything has same number of issues**
- Level 1: randomness in fake issues, so only “real” issues are stable
- Level 2: fake parameters added but only “real” params are stable
- ...

# DVWA

- ZAP, all vulnerabilities injected on all params



# HONEYPITS AND MIRAGES

- Gaslighting is the next step
  - It's there, but only for you. Nobody else can see it.

# PSYOPS - GASLIGHTING

I'M NOT GASLIGHTING YOU, YOU'RE HYSTERICAL

# PSYOPS - GASLIGHTING

- Everybody's viewpoint is different
- The trick is to make this difference in viewpoint make the target question which is real



# PSYOPS - GASLIGHTING

- M-deception (Misdirection) tries to make the target believe a picture different from the reality
- A-deception (Ambiguity) deception reduces the ability to assess the reality

# GASLIGHTING

- Consistent (M-deception)
- Inconsistent (A-Deception)
- Hybrid
  - Consistent, but only for you. ← This is the really evil one

# PSYOPS - GASLIGHTING

- “ Gaslighting is now used to refer to any attempt to make another person doubt their sense of reality.” -  
<http://berkeleysciencereview.com/call-me-crazy-the-subtle-power-of-gaslighting/>
- Named after a 1930s play and 1940s films “Gas Light”.



**GASLIGHT**

The word "GASLIGHT" is rendered in a large, bold, serif typeface with a double outline. The letters are set against a dark, textured background. A bright, stylized flame or light source is positioned directly above the letter 'S', casting a glow and creating a sense of depth. The overall aesthetic is reminiscent of classic horror or thriller movie titles.

# PSYOPS - GASLIGHTING

- “\_ Gaslighting is now used to refer to any attempt to make another person doubt their sense of reality.”\_  
<http://berkeleysciencereview.com/call-me-crazy-the-subtle-power-of-gaslighting/>
- Named after a 1930s play and 1940s films “Gas Light”.
- In it, a man convinces a woman she is delusional in order to steal from her

**A STORY THAT  
REVEALS**

**A MAN'S *SECRET*  
*AND UNHOLY*  
DESIRES.....**



AND PROBES INTO  
THE STRANGE  
*Emotional Depths*  
OF ONE WOMAN'S  
HEART...

# PSYOPS - GASLIGHTING

- As a manipulation technique, it is basically evil.
- Make sure you aren't evil. Evil is bad.
- Seriously, don't be evil

# PSYOPS – GASLIGHTING STEPS

- 1. Make them comfortable and feeling in charge or loved
- 2. Make them feel they understand the rules
- 3. Let them act as if that is true
- 4. Either
  - A. Demonstrate the "untruth" dramatically
  - B. Continually invalidate their grasp of the situation

# HONEYPITS - GASLIGHTING

- Level 1: Everything has same number of issues
- Level 2: randomness in fake issues, so only “real” issues are stable
- Level 3: fake parameters added but only “real” params are stable
- Level 4: sticky randomness in fake issues, so have to filter them out somehow with secondary tests
- Level 5: Sticky randomness in fake issues, but only for you
- //Level 6: Enter the matrix. Also Commenting out slide points doesn't work

# FOR YOU ONLY (FYO) – GENERAL STEPS

- Form a fingerprint of the source/request/target
- Hash that fingerprint with a secret
- Use the hash to determine which vulnerabilities appear

# FOR YOU ONLY (FYO) - FINGERPRINT PARAMETERS

- Time
- Target Parameter
- Target Filename
- Address (source)
  - Specific or range
- Address (Dest)
  - Specific or range
- Account
- Window
- Cookies
- Fingerprinting (Browser)
- Fingerprinting (Stack)
- Fingerprinting (Network/latency)
- Fingerprinting (Social)



# FOR YOU ONLY (FYO) – OUR STEPS

- Steps
  - Have a set of Vulnerability emulators with IDs
  - Form a fingerprint of the source/request/target
  - Hash that fingerprint with a secret GUID
  - Use entropy in the hash to determine which vulnerabilities appear
  - Use bit 1 for vulnerability ID 1, bit 2 for ID 2, etc

# FOR YOU ONLY (FYO) – OUR STEPS

- For this demo, we used stateless
  - Params (PHP):
    - IP Address
    - Filename target
    - Secret Token
    - Parameter
    - Cookies
  - 11 Vulnerability Emulators

# FOR YOU ONLY (FYO) – OUR STEPS

```
foreach($_POST as $key => $value){  
    // echo $key . " : " . $value . "<br />\r\n";  
    process_param($key,$value);  
}
```

```
foreach($_GET as $key => $value){  
    // echo $key . " : " . $value . "<br />\r\n";  
    process_param($key,$value);  
}
```

# FOR YOU ONLY (FYO) – OUR STEPS

```
function process_param($key,$value){
    global $list_enabled_vulns;
    $paramName = $key;
    $vuln_str = get_vuln_md5($paramName);
    if($list_enabled_vulns){
        echo "<br><br>Param name: " . $key . " enabled vulns: ";
        for( $i = 0; $i<10; $i++ ) {
            if(get_vuln_enabled($vuln_str,$i)){
                echo $i . ", ";
            }
        }
        echo "<br>";
    }

    //vuln 0
    if (get_vuln_enabled($vuln_str,0)){
        ...
    }

    //vuln 1
    if (get_vuln_enabled($vuln_str,1)){
        ...
    }
}
```

# FOR YOU ONLY (FYO) – OUR STEPS

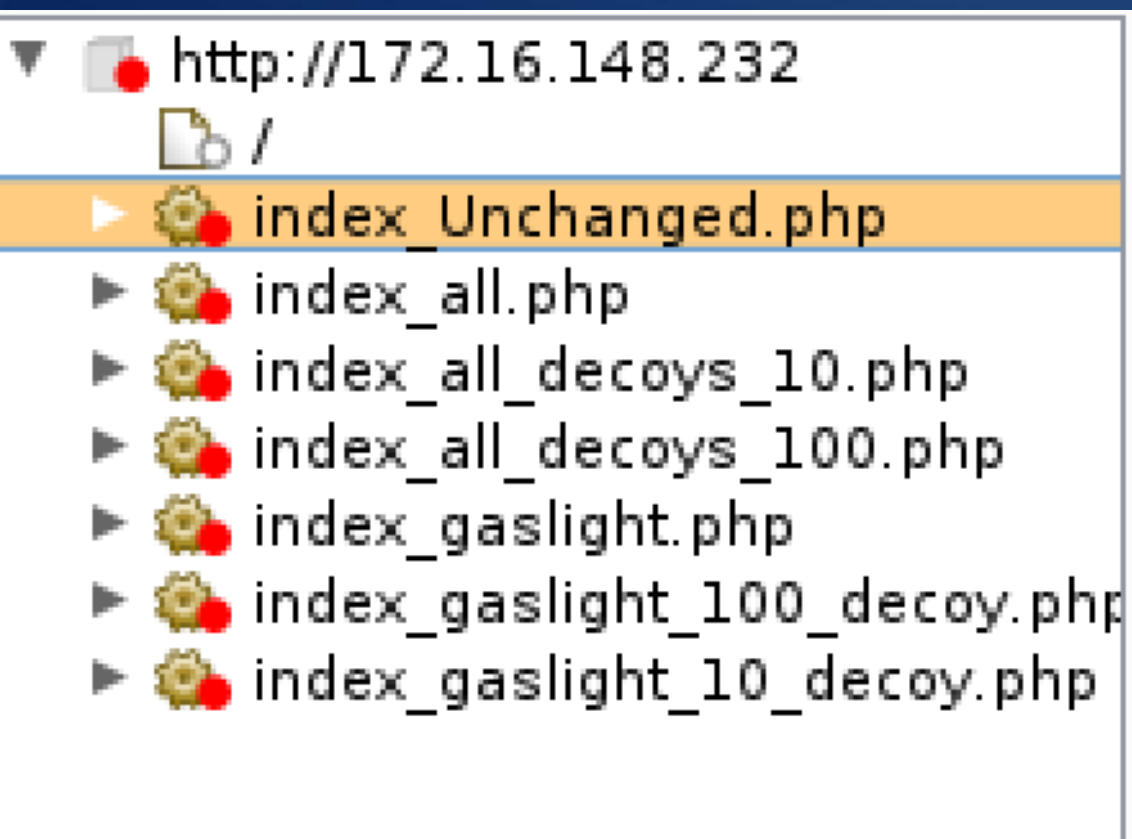
```
$user_ip = $_SERVER['REMOTE_ADDR'];  
$cookies = $_COOKIE;  
$basename_val = basename($_SERVER['REQUEST_URI']);
```

```
function get_vuln_md5($paramName){  
    global $user_ip, $secret_token, $cookies, $basename_val;  
    $strgen = $user_ip . $basename_val . $secret_token . "X" . $paramName . "X" . implode(',', $cookies);  
    $md5_raw = md5($strgen, true);  
    return $md5_raw;  
}  
  
function get_vuln_enabled($md5_raw, $bitID){  
    global $enable_all_vulns;  
    if($enable_all_vulns){  
        return true;  
    }  
  
    $byteNum = (int) ($bitID / 8);  
    $bitOffset = $bitID % 8;  
    return (ord($md5_raw[$byteNum]) & (1 << $bitOffset) );  
}
```




# FOR YOU ONLY (FYO) – DEMOS

## Basic Case



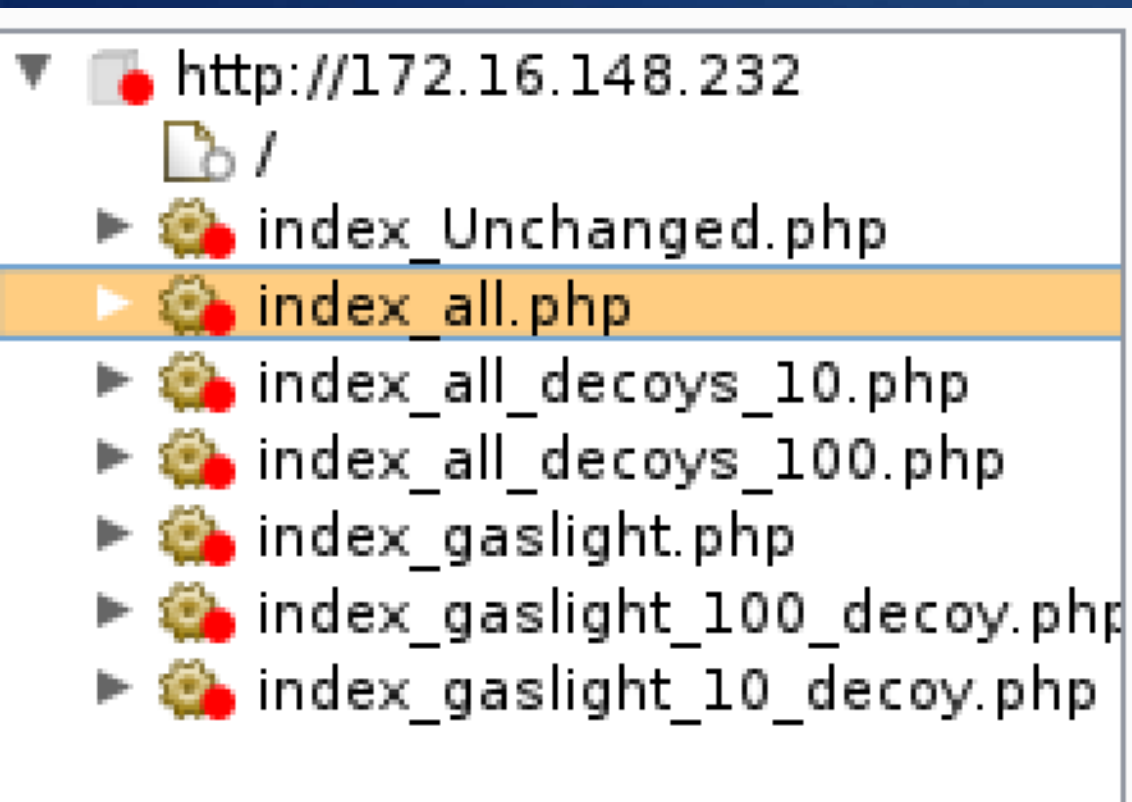
## Issues

- ▶  Cross-site scripting (reflected) [3]
  - i Cross-site request forgery



# FOR YOU ONLY (FYO) – DEMOS

Injecting all on all



















## Issues








- ❗ SQL injection
- ▶ ❗ Cross-site scripting (reflected) [3]
- ▶ ❗ External service interaction (DNS) [2]
- ▶ ❗ External service interaction (HTTP) [2]
- ❗ Python code injection
- ❗ XML injection
- ℹ Cross-site request forgery

# FOR YOU ONLY (FYO) – DEMOS

Injecting 10 dummy parameters AND all vulns on all

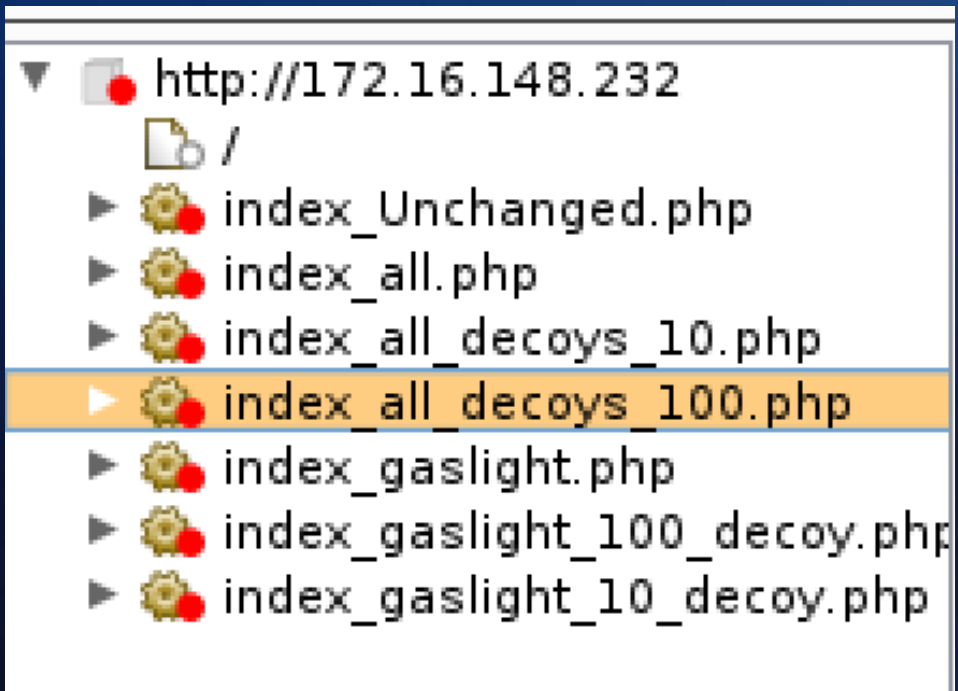
- ▼  http://172.16.148.232
  -  /
  - ▶   index\_Unchanged.php
  - ▶   index\_all.php
  - ▶   index\_all\_decoys\_10.php
  - ▶   index\_all\_decoys\_100.php
  - ▶   index\_gaslight.php
  - ▶   index\_gaslight\_100\_decoy.php
  - ▶   index\_gaslight\_10\_decoy.php

## Issues

- ▶  SQL injection [15]
- ▶  Cross-site scripting (reflected) [3]
- ▶  External service interaction (DNS) [25]
- ▶  External service interaction (HTTP) [25]
- ▶  Python code injection [15]
- ▶  XML injection [8]
  -  Cross-site request forgery

# FOR YOU ONLY (FYO) – DEMOS

Injecting 100 dummy parameters AND all vulns on all



## Issues

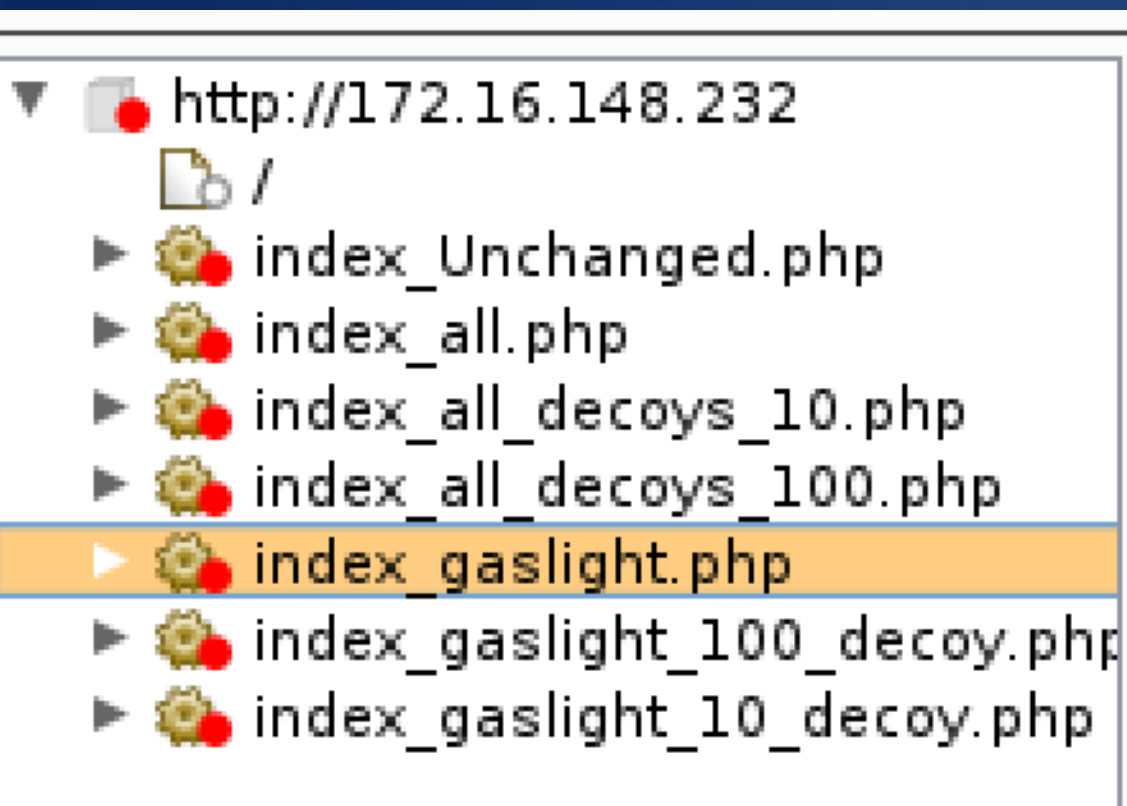
- ▶ **SQL injection [51]**
- ▶ **Cross-site scripting (reflected) [2]**
- ▶ **Python code injection [51]**

It didn't finish before I got bored...



# FOR YOU ONLY (FYO) – DEMOS

## Injecting gaslight vulns

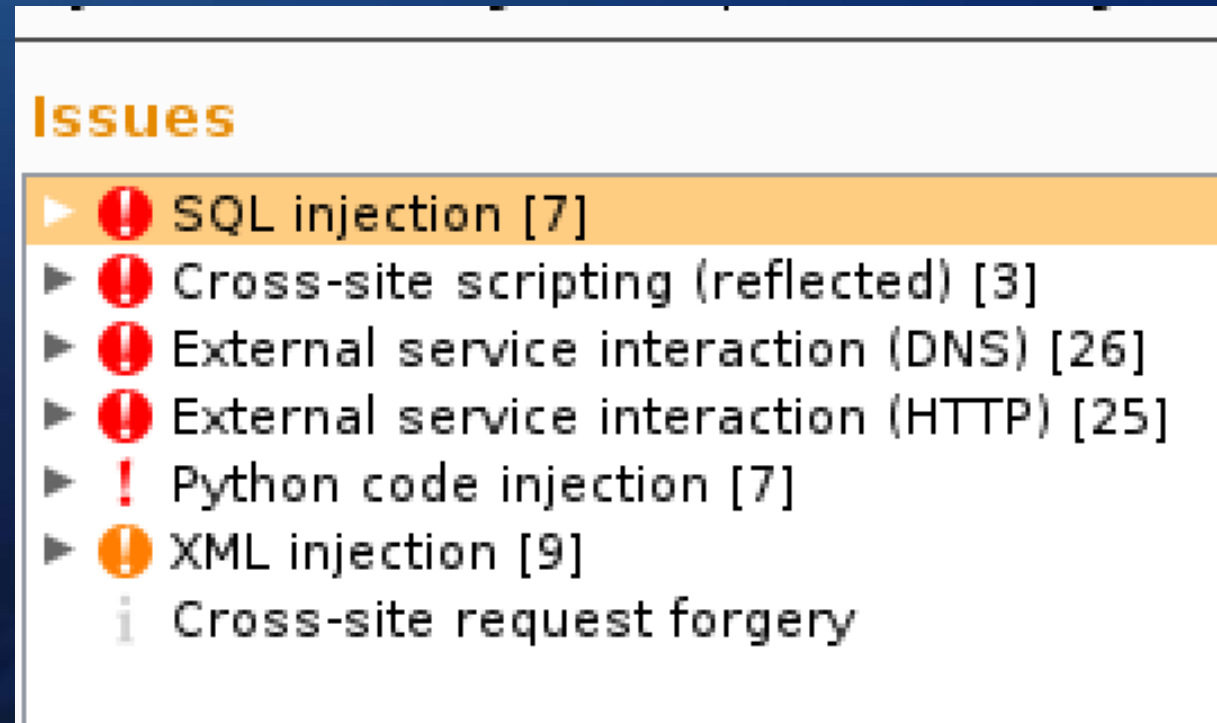
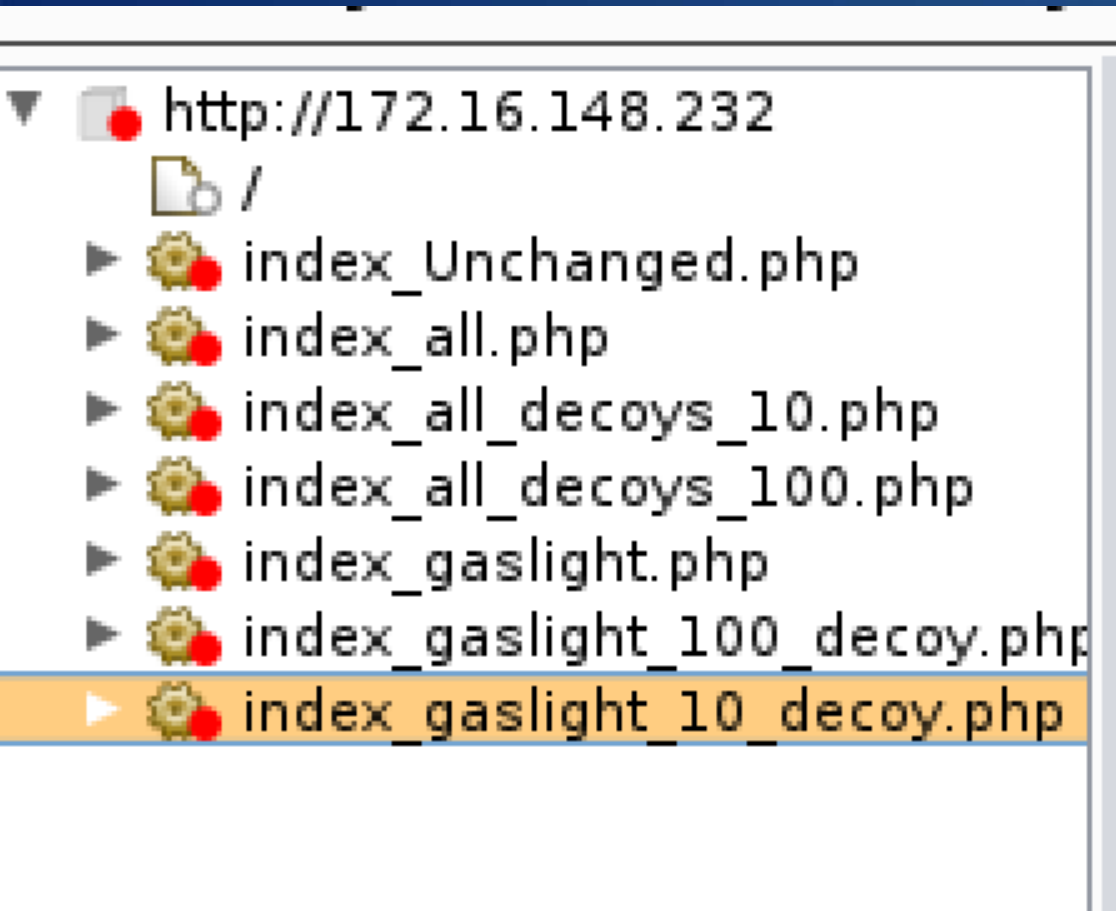


### Issues

- ▶ SQL injection [3]
- ▶ Cross-site scripting (reflected) [3]
- ▶ External service interaction (DNS) [2]
- ▶ External service interaction (HTTP) [2]
- ▶ Python code injection [3]
- ▶ XML injection
  - Cross-site request forgery

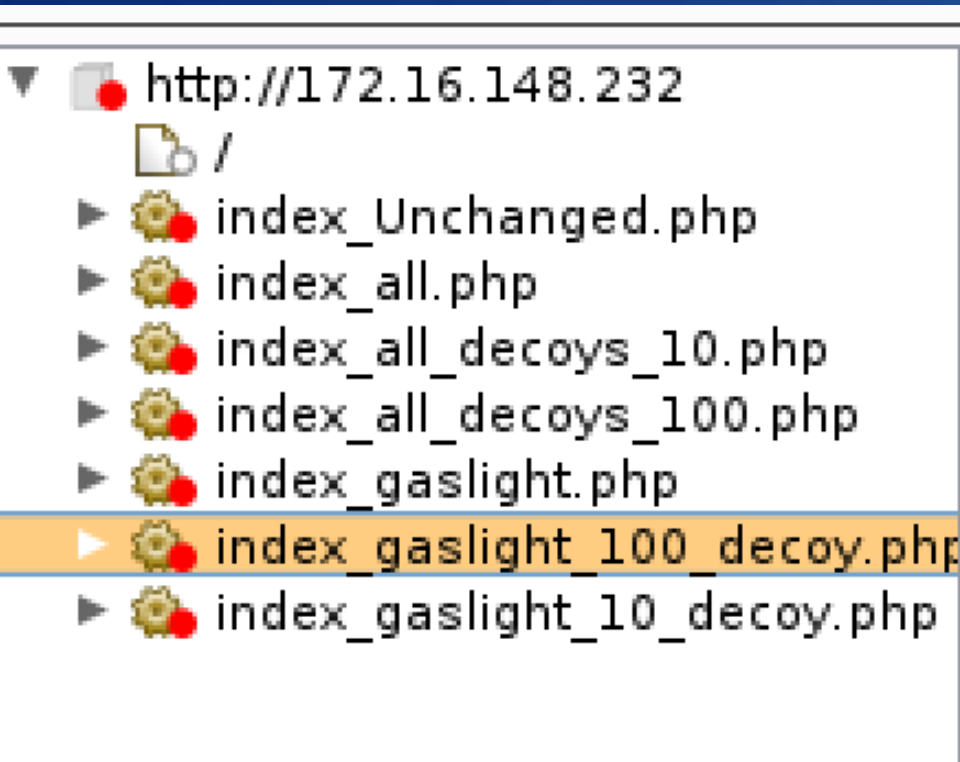
# FOR YOU ONLY (FYO) – DEMOS

Injecting gaslight vulns and 10 dummy parameters



# FOR YOU ONLY (FYO) – DEMOS

Injecting gaslight vulns and 100 dummy parameters



## Issues

- ▶ **SQL injection [47]**
- ▶ **Cross-site scripting (reflected) [3]**
- ▶ **Python code injection [47]**

It didn't finish before I got bored...  
and the scanner breaks with this  
many params



# FOR YOU ONLY – OTHER OPTIONS

- Services
- Systems
- Networks
- Parallel Universes
  - Take the red pill, see how deep the rabbit hole goes
  - Hit the red pill, get trapped in \_\_\_\_\_NARNIA??\_\_\_\_\_

# TRANSIENT GASLIGHTING

- Security tools (and people) don't handle NAN causal errors very well

# TRANSIENT PROBLEMS - MISHEARING

- Return unassociated error or response conditions
  - E.g. integer parse errors on text
  - Library errors for libraries not used
  - Headers for different systems

# TRANSIENT PROBLEMS – MISSPEAKING

## ANSWER PRETTY MUCH AS EXPECTED BUT NOT QUITE

- Random errors
  - Random timing errors
  - Random omissions
  - Random bitflips
  - Random endian changes
  - Random number changes
  - Badly signed things
  - Badly encrypted things
  - Random wrong content
- Nonrandom errors to break things
  - Invalid characters/bytes
  - Terminal command characters
  - Random “unallocated” memory
  - Bad pointer values
  - Filesystems of the wrong type
  - Impossible filenames
  - Timing “errors”
  - “Omissions”

# FUTURE

- More Vulnerability Fakes
- Limited Time Vulnerabilities
  - Roll vulnerabilities with a time bin (e.g. hour, day)
- Limited Count Vulnerabilities
  - Requires state tracking
- Cache scanner test results in advance
  - Often Deterministic / predictable

# GASLIGHTING - MORE

- Uncrackable Hashes
- Decoy Systems, Ports, Services
- Manufactured Vuln Emulation
  - E.g. MS08-067?
- Decoy Vulns (static)
- Decoy Vulns (non exploitable {buffer overflow in managed lang})
- Nondeterministic Existence
- For you only existence
- Transient Vulns
- Transient Systems, Ports, Services
- Vuln neutering
- Vuln Chains leading nowhere
- Benign Passthrough
- Honeypot Passthrough
- Trickster passthrough
- One time Vulnerability Generation
- One time vulns as canaries
- Answering questions you never asked
- Answering different questions
- Fake answers
- Fake Data
- Silent Failure (denying you ever agreed)
- Rewriting page format dynamically to break validation and cscripting



# OTHER

- Can I get attackers to crack hashes for me?
  - Have a “vulnerable” system with a database that can be dumped
  - Have hashes assigned to accounts
  - If attackers dump, crack, and successfully log in them save the output.

Twitter: @secvalve

End

Or is it?

# REFERENCES

- <https://fas.org/irp/doddir/army/fm90-2/toc.htm>
- [http://www.au.af.mil/au/awc/awcgate/cia/tradecraft\\_notes/note\\_10.htm](http://www.au.af.mil/au/awc/awcgate/cia/tradecraft_notes/note_10.htm)
- “Deception for the Cyber Defender: To Err is Human; to Deceive, Divine” – Shmoocon 2015
- “From Patches to Honey-Patches: Lightweight Attacker Misdirection, Deception, and Disinformation”
- “Planning and Integrating Deception into Computer Security Defenses” - Almeshekah, Mohammed H and Spafford, Eugene H, *Proceedings of the 2014 workshop on New Security Paradigms Workshop* -  
[https://www.researchgate.net/profile/Eugene\\_Spafford/publication/267748217\\_Planning\\_and\\_Integrating\\_Deception\\_into\\_Computer\\_Security\\_Defenses/links/5463a1a80cf2c0c6aec4f5c6.pdf](https://www.researchgate.net/profile/Eugene_Spafford/publication/267748217_Planning_and_Integrating_Deception_into_Computer_Security_Defenses/links/5463a1a80cf2c0c6aec4f5c6.pdf)
- *Lots more to come soon*