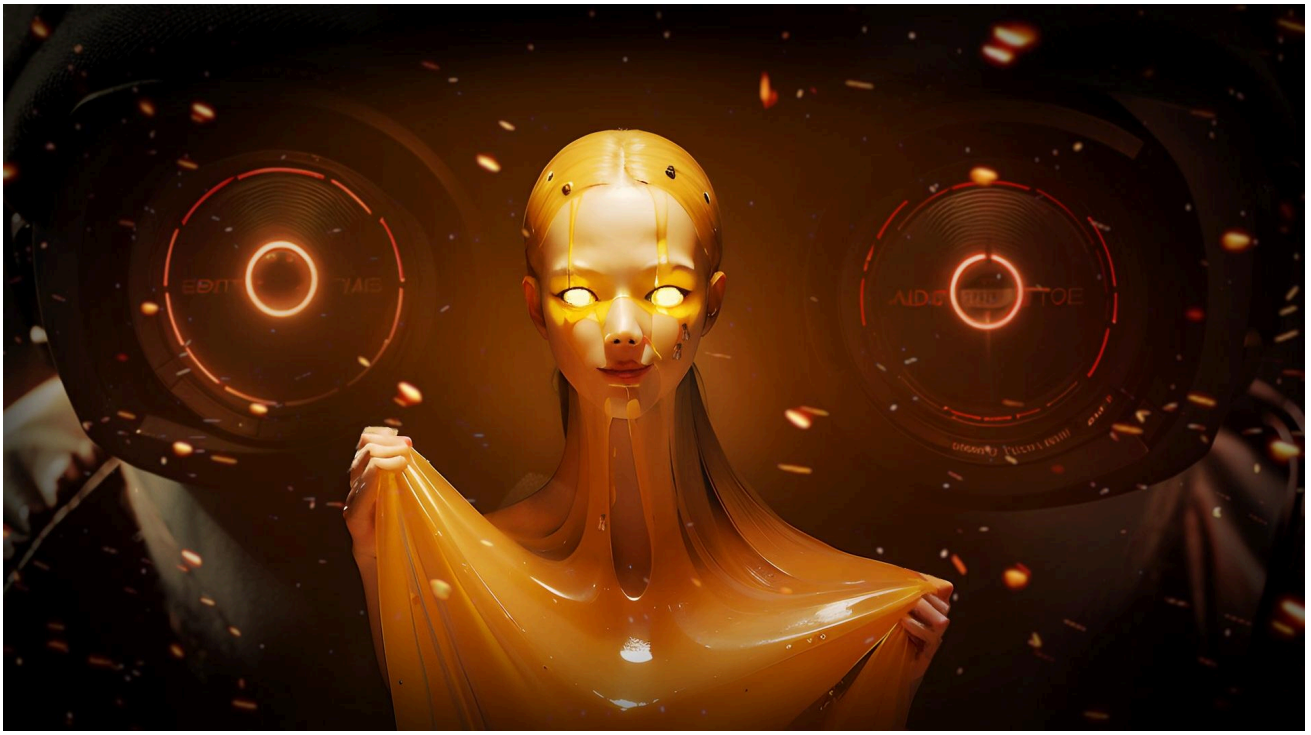
A GAME OF DECEPTION



"The one where we lure them in."

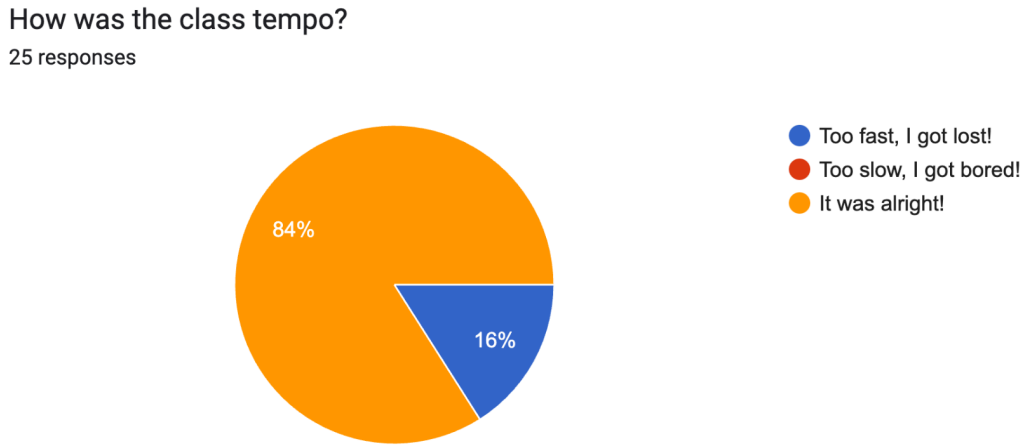
October 24th, 2024

Credits

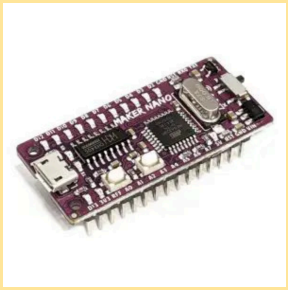
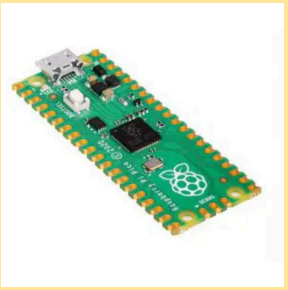
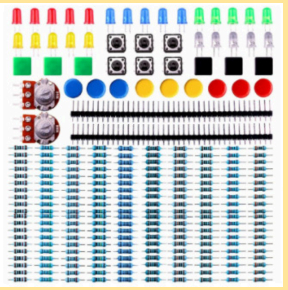
Content: Muris Sladić
Tim Pappa
Sebastian Garcia
Ondřej Lukáš
Maria Rigaki
Martin Řepa
Lukáš Forst
Veronica Valeros
Illustrations: Fermin Valeros
Design: Veronica Garcia, Veronica Valeros, Ondřej Lukáš
Music: Sebastian Garcia, Veronica Valeros, Ondřej Lukáš
CTU Video Recording: Jan Sláma, Václav Svoboda, Marcela Charvatová
Audio files, 3D prints, and stickers: Veronica Valeros

CLASS DOCUMENT	https://bit.ly/BSY2024-5
WEBSITE	https://cybersecurity.bsy.fel.cvut.cz/
CLASS MATRIX	https://matrix.bsy.fel.cvut.cz/
CLASS CTFD (CTU STUDENTS)	https://ctfd.bsy.fel.cvut.cz/
CLASS PASSCODE FORM (ONLINE STUDENTS)	https://bit.ly/BSY-VerifyClass
FEEDBACK	https://bit.ly/BSYFEEDBACK
LIVESTREAM	https://www.youtube.com/playlist?list=PLQL6z4JeTTQmu09ItEQaqjt6tk0KnRsLh
INTRO SOUND	https://bit.ly/BSY-Intro
VIDEO RECORDINGS PLAYLIST	https://www.youtube.com/playlist?list=PLQL6z4JeTTQk_z3vwSIvn6wIHMeNQFU3d
CLASS AUDIO	https://audio.com/stratosphere

Results from the survey of the last class (14:32)



Pioneer Prize for Assignment 4 (14:35)

1st Place	2nd Place	3rd Place
		
Cytron Maker Nano	Raspberry Pi Pico	MINI Elektro Starter Kit
Lukáš (novakl39)	**Ľuboslav (motoslub)**	**Jan (hoferjan)**

Parish Notices

- Remember that for CTU students, assignments 2, 3, 4, and 5 close on **November 7th, 2024 23:59**. If you haven't solved them by the deadline, you won't be able to solve them.

The main topics in this class

1. Deception, why it works, key components
2. Cyber Deception & steganography
3. Behavioral Analysis as an Adaptive Methodology for Cyber Deception Design - Guest Talk by Tim Pappa
4. Honeypots: honey systems, honey tokens, and honey services

Deception (14:40)

Goal: to learn what deception is, its basic concepts, and applications.

What is deception?

Deception is a deliberate attempt to conceal, fabricate, and manipulate factual information to create or maintain in others a belief considered false.

Deception is common in several areas of our society (and more):

- Advertising, propaganda, and misinformation
  - False sales
  - WW2 posters
- Business
  - Salesmen saying there is a high demand for a product
- Sports
  - Faking the direction of movement
- Entertainment
  - Magicians
- Politics
  - Highlighting only the positive or negative side to shape public opinion
- Military
  - Battle for the Wounded (Battle of Neretva) - https://en.wikipedia.org/wiki/Case_White
- Finances
  - Optimistic growth assumptions
- Internet
  - Social engineering
- Animal world
  - Opossums playing dead
  - Fake poisonous frogs

Unlike lying, **deception does not require a statement to be made**, and deception connotes success.

- Lying can be used for deception purposes, but it is not necessary.

Why does deception work?

- **Let's play a game!**
    - I need a volunteer to play
- Some experiments show that the success rate for detecting lies and deception is barely above 50%[1]
- Human and animal perception of the world and reality is based on patterns that are called **biases**[2].
- Biases influence decision-making, beliefs, and behavior.
    - **Personal biases**: from individual experience, education, traits...
    - **Cultural biases**: acquired during the evolution of cultures, widely held beliefs, traditions...
    - **Organizational biases**: biases that stem from organizational culture, leadership, policies...
    - **Cognitive bias:** intrinsic to our nature in perceiving and processing information. It is impossible to avoid and occurs involuntarily.
        - Derivative of human processing limitations. Our brains are lazy, we employ shortcuts.
        - We tend to perceive what we **expect** to perceive.
- Are there other things than biases that make us vulnerable to deception?
    - Emotions
    - Lack of Knowledge
    - Information Overload
    - Repetition - people are more likely to believe a false statement if they hear it repeatedly
    - Poor critical thinking skills

[1] A. Vrij. Detecting lies and deceit: Pitfalls and opportunities. 2nd Edition. John Wiley & Sons, 2008
[2] https://www.visualcapitalist.com/every-single-cognitive-bias/

- Most often, it is probably a combination of multiple factors that makes us vulnerable.
- One of the most important things to remember about deception, so let's repeat
    - We tend to perceive what we **expect** to perceive.

Components, principles, and variants of deception

Deceptions may differ in their complexity, but some components are common for all:

- Deceiver & Target
- Deception story
- Channels
- Target feedback

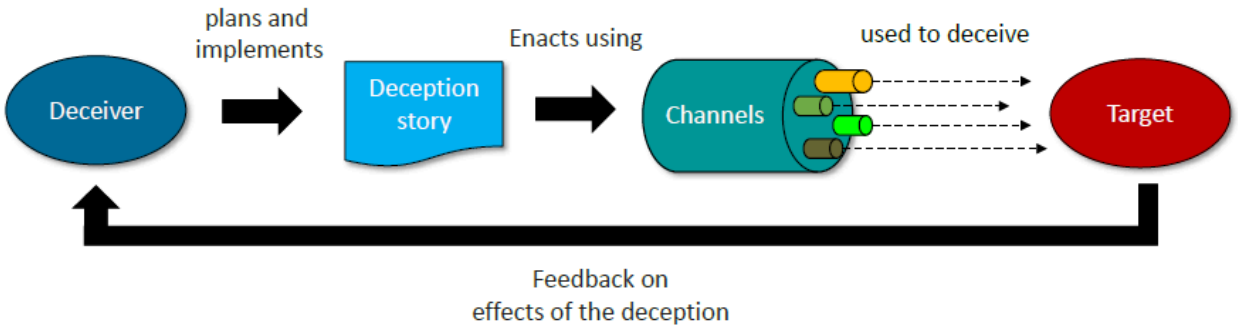


Figure 1. Strategic deceptive flow[3]

We can say that deception is governed by 4 basic **principles**:

- **Truth**: The default state of the world. Deception needs to be an exception to a non-deceptive world.
- **Denial**: Blocking the target's access to true data
- **Deceit**: Lying and/or simulating
- **Misdirection**: Drawing attention away from real assets and activities
- ***One extra*: Confusion** - degrade perceptual abilities

Variants of deception:

- **A-Type** (ambiguity-increasing) - target unsure of what to believe
- **M-Type** (misleading) - target believes in the false alternative

[3] C De Faveri, Modeling Deception for Cyber Security, 2021, https://run.unl.pt/bitstream/10362/148907/1/Faveri_2022.pdf

Deception can be either:

- passive or active.

Magruder's principles

Before attempting deception, it is vital to know:

- It is generally easier to induce a target to maintain a preexisting belief than to change it or create a new one.
- Following up on Magruder's principle, it has been asserted that
    - successful deception must contain 95% truth.

Recap: To create a good deception... Be imaginative, and creative, and understand the need and purpose, the setting, the target, and prepare well.

Cyber Deception

Goal: to learn how deception is used in cyberspace.

What is cyber deception?

It is a form of deception that leverages digital tools and the online environment to mislead, manipulate, or confuse a target.

It can be used by both attackers and defenders.

Deception for attackers

There are many examples of attackers using deception, for example:

- Phishing
- Evil Twin - Websites
    - https://www.shouldiclick.org/
- Evil twin - Rogue WiFi access point (MitM)
- Malvertising
- Social engineering (Impersonating)
- Deepfake technology
- Trojan horses
    - Spyware
- Steganography

Steganography - There is definitely nothing suspicious going on 😉

Steganography is the practice of representing information within a medium in such a manner that the presence of the information is not evident to human or machine inspection.

- Today, in cyberspace, Steganography is detected with tools and ML so it should also be resistant to that.
- One advantage of steganography is that fewer people try to find it and break it because they don't know something is there.
    - Good for situations where usual encryption may raise suspicion
    - However, it is security by obscurity and should never be the sole security solution.
- Another advantage is that it can use multiple types of media to transmit messages
    - Photos, video, text files, audio files etc.
- Some disadvantages are
    - Limited capacity of data that can be hidden
    - Carrier quality degradation

What are the differences with encryption?

They are actually not even related but easily confused:

Steganography	Encryption
Steganography hides the fact that information exists	Encryption does not hide this.
Steganography can be encrypted (does not have to be).	
Encryption is designed not to be broken	Steganography is designed not to be found.

How can attackers use steganography?

- Replacing pixels with malware code
- Send instructions to command and control servers
- Hide stolen data
    - Unauthorized data transfers

Let's try Steganography! When a photo is more than just a photo 👀

- Let's download a photo
    - `wget https://gcdnb.pbrd.co/images/YqFrPc65LTvj.jpg?o=1`
- Let's rename it so it's easier
    - `mv 'YqFrPc65LTvj.jpg?o=1' Poirot.jpg`
- Let's check the photo out by clicking the link.
    - https://gcdnb.pbrd.co/images/YqFrPc65LTvj.jpg?o=1
- How can we analyze it?
    - ExifTool is free and open-source software for reading, writing, and manipulating image, audio, video, and PDF metadata.
    - `apt install exiftool`
    - `exiftool Poirot.jpg`
    - Strings is a Linux command to print the strings of printable characters in files.
    - `strings Poirot.jpg`
    - *xxd* creates a hex dump of a given file or standard input.
    - `xxd Poirot.jpg`
- Using exiftool, strings, and xxd we found some hidden messages in the photo.
- How to hide messages like in the previous example?
    - `echo "New Message" >> Poirot.jpg`
    - Remember with Exiftool earlier, we found a message hidden in the comment section of jpg. You can write a message in a comment like this:

    - `exiftool -comment='New comment' photo.jpg`
    - Running this command will rewrite the old comment
    - More ways can be found online, Or ask ChatGPT ;)

Something for fun 🪄

- During the break, put a hidden message for your class colleagues in a photo of your choice. (Make sure the photo and the message do not violate any ethical guidelines of the class)
- Put the photo in the Matrix random channel for your colleagues to analyze and read your message.

Steghide[4]

Steghide is a tool to hide text messages in images. Very useful!

- `apt install steghide`
- `wget https://upload.wikimedia.org/wikipedia/commons/3/3a/Cat03.jpg`
- You can check the image by clicking on the URL
    - https://upload.wikimedia.org/wikipedia/commons/3/3a/Cat03.jpg
- Let's prepare file we want to hide
- `echo "hi there" > text.txt`
- `steghide embed -ef text.txt -cf Cat03.jpg`
    - Enter the password
- Copy the stego-cat to your computer. From your computer:
    - CTU students:
        - `scp -O -P <your_docker_port> root@<YOUR-DOCKER-IP>:/root/Cat03.jpg Cat03.jpg`
            - `-O - pass options to SSH; we use it here to pass the port number`
            - `-P - your SSH docker port`
    - Online students:

[4] shetzl @chello.at, S.H. (2024). Steghide. [online] Sourceforge.net. Available at: https://steghide.sourceforge.net/ [Accessed 22 Oct. 2024].

        - `scp -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -P 2222 root@127.0.0.1:/root/Cat03.jpg Cat03.jpg`
            - `-o` - pass options to SSH
            - `StrictHostKeyChecking=no` - do not confirm identity of the remote server
            - `UserKnownHostsFile=/dev/null` - do not store remote host's public key in known_hosts file
    - See the image!
    - Does it look strange?
- Desteganize
    - `steghide extract -sf Cat03.jpg -xf outside.txt`

Briefly on Steganalysis

Steganalysis is the study of detecting messages hidden using steganography. It is analogous to cryptanalysis applied to cryptography.

Hard, but mostly it involves checking that the protocol/format was not respected or finding statistical anomalies regarding the use of parts of a protocol that should not be used so much (too many spaces in text, for example).

- Some things to look for
    - Inconsistencies in data compression
    - Distortion of neighboring pixels

It is so hard that companies sell it as a service[5] (Example)

Deception for defenders

Why do defenders use deception in cyber security?

- Enhance protection & gather intelligence on attackers
- Mitigate **cyber kill chain**

[5] WetStone Technologies (2020). StegoCommandTM [online] Available at: https://www.wetstonetech.com/products/stegocommand-steganography-detection/ [Accessed 22 Oct. 2024].

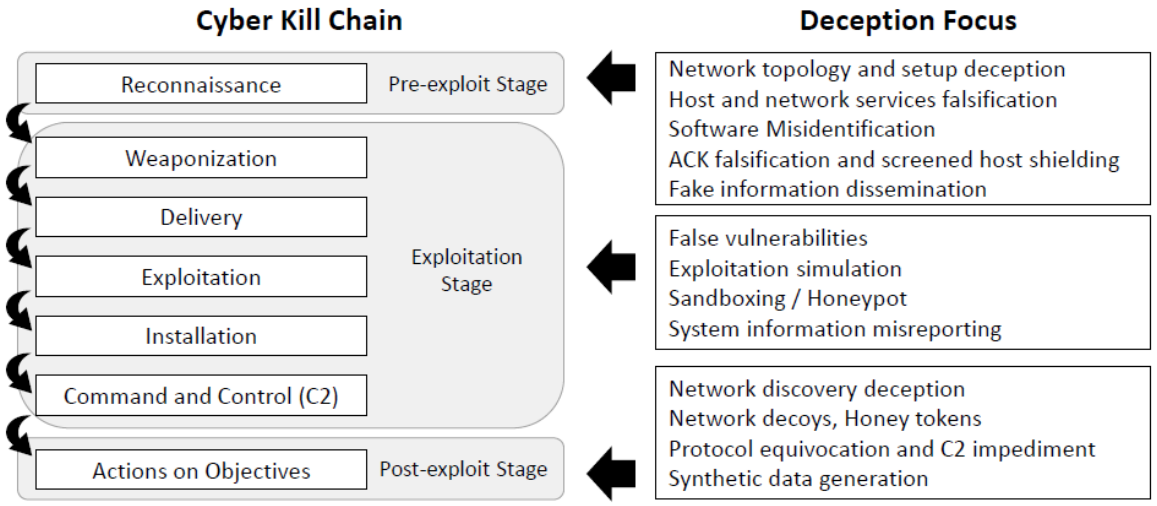


Figure 2. Defensive deception and the cyber kill chain[6]

There are many examples of deception used by defenders, for example:

- Honeypots - fake systems designed to lure in attackers
    - Honeytokens
    - Honeyfiles
    - Honeywords
- Notifying about deception
    - How does this work? you might ask
    - Very interesting research by Ferguson et. al.[7]
- Some examples from real-life stories:
- Clifford Stoll
    - The Cuckoo's Egg
    - Ted talk: https://www.youtube.com/watch?v=Gj8IA6xOpSk

[6] C De Faveri, Modeling Deception for Cyber Security, 2021, https://run.unl.pt/bitstream/10362/148907/1/Faveri_2022.pdf

[7] Ferguson-Walter, K., Major, M., Johnson, C. and Muhleman, D. (2021). *Examining the Efficacy of Decoy-based and Psychological Cyber Deception Examining the Efficacy of Decoy-based and Psychological Cyber Deception*. [online] Available at: https://www.usenix.org/system/files/sec21-ferguson-walter.pdf

Break I (15:25-15:35)

Behavioral Analysis as an Adaptive Methodology for Cyber Deception Design - Guest Talk by Tim Pappa (15:35)

Tim Pappa is an Incident Response Engineer - Cyber Deception Strategy, Content Development, and Marketing, Cyber Deception Operations, Walmart Global Tech. Before Walmart, Tim was a Supervisory Special Agent and profiler with the Federal Bureau of Investigation's (FBI) Behavioral Analysis Unit (BAU), specializing in online influence and cyber deception. Tim is also a Senior Behavioral Consultant for Analyst1 and an Adjunct Professor at Capitol Technology University. **LinkedIn**

Background

- Studying and applying communication
- Understanding behavior and exploring how to influence behavior online and offline
- Becoming an industry cyber deception practitioner

Exercise 1: Student feedback

A: What do you think the purpose is of this object?

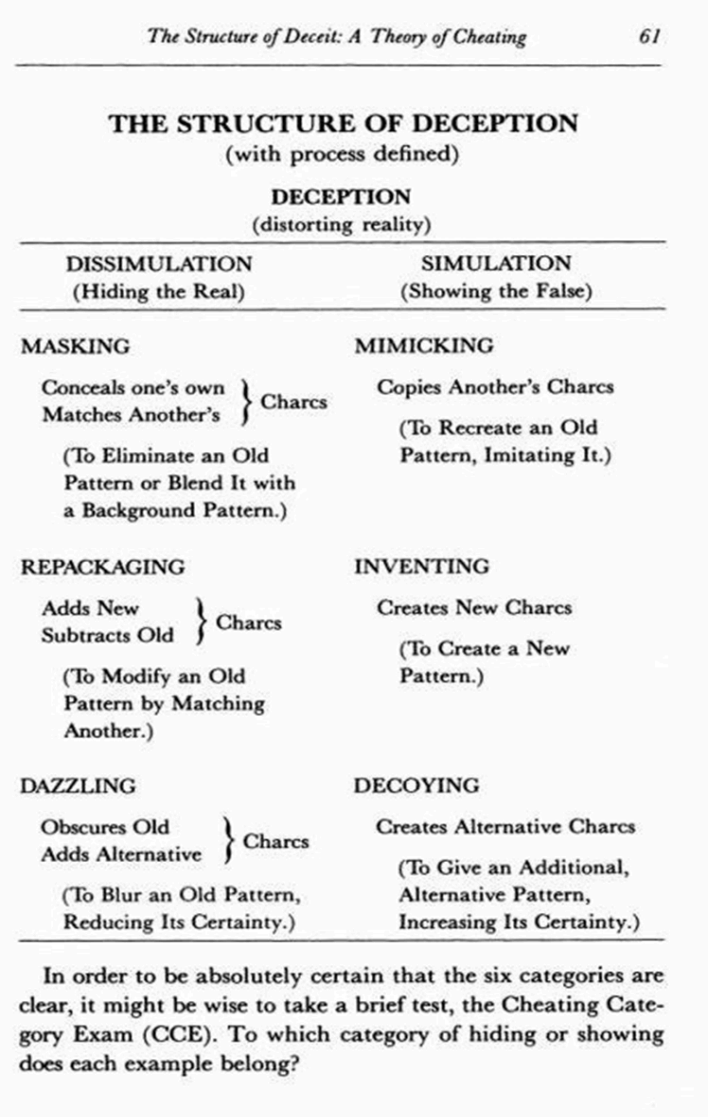
B: If you wanted to purchase golf balls on a reseller site like eBay, for example, which golf balls below would you make a bid on or try to purchase?

Behavioral Analysis as a Method of Grounded Theory

- Grounded psychological theories of victimology and criminology, but applied to individuals and their relationships
- Digital criminalistics and cyber pathways
- Application of relevant theories and concepts to online engagements
- Finding behavioral vulnerabilities to exploit

Bell-Whaley's Expanded Deception Framework

- There are *qualitative* and *quantitative* attributes of misperception
- Whaley emphasized the simultaneous nature of *simulation* and *dissimulation*
- Whaley was also a practitioner and scholar of communication and influence, perhaps more than deception



The Structure of Deceit: A Theory of Cheating 61

THE STRUCTURE OF DECEPTION
(with process defined)

DECEPTION
(distorting reality)

DISSIMULATION (Hiding the Real)	SIMULATION (Showing the False)
MASKING Conceals one's own / Matches Another's } Charcs (To Eliminate an Old Pattern or Blend It with a Background Pattern.)	**MIMICKING** Copies Another's Charcs (To Recreate an Old Pattern, Imitating It.)
REPACKAGING Adds New / Subtracts Old } Charcs (To Modify an Old Pattern by Matching Another.)	**INVENTING** Creates New Charcs (To Create a New Pattern.)
DAZZLING Obscures Old / Adds Alternative } Charcs (To Blur an Old Pattern, Reducing Its Certainty.)	**DECOYING** Creates Alternative Charcs (To Give an Additional, Alternative Pattern, Increasing Its Certainty.)

In order to be absolutely certain that the six categories are clear, it might be wise to take a brief test, the Cheating Category Exam (CCE). To which category of hiding or showing does each example belong?

A Practitioner's Approach to Cyber Deception Design

- Doing more than just 'annoying' attackers
- Increasing calls for behaviorally-based design of cyber deception responses
- Applying a transdisciplinary approach to custom builds or repurposing content and infrastructure
- The Bell-Whaley (expanded) deception framework is my foundation

Integrating this Method and Approach

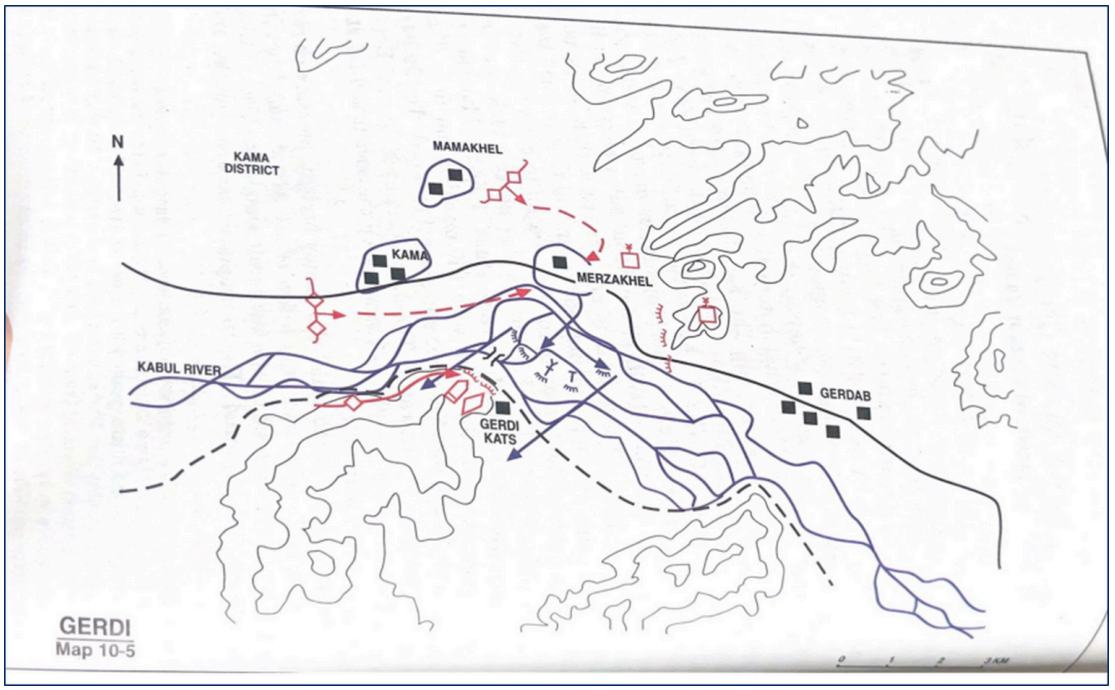
- Behavioral analysis in context is your starting point

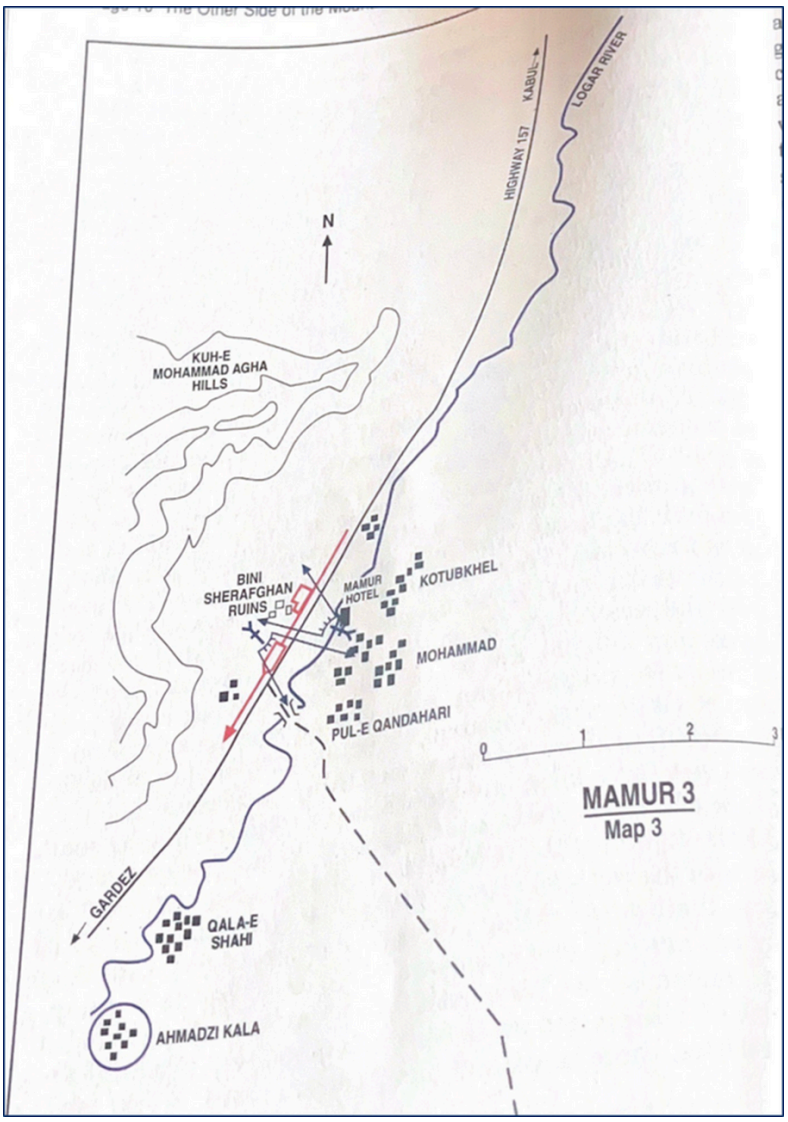
- Considering far and near analogies when designing cyber deception to influence behaviors
- Concentrating design on observed and known behaviors

FIG. 7. VARYING HARE IN WINTER
Camera-trap photograph taken on a runway in the snow in Canadian life-zone at 1,800 feet, Fern Ridge, Monroe County, Pennsylvania, at 5 p.m., December 30, 1922. Exposure $\frac{1}{15}$ second. The camera-trap wires may be dimly seen.

FIG. 5. FLORIDA OPPOSSUM, SEIZING WOODPECKER BAIT
Photographed near Lake Harney, Orange County, Florida, at 8:53 p.m., August 2, 1910. Note the white face. Is it a freak? Exposure speed $\frac{1}{15}$ second. The first super-speed camera-trap photograph of an animal ever made.

FIG. 6. BEAVER ON TOP OF DAM
Photographed in Monroe County, Pennsylvania, August, 1924. The beavers are "coming back" in parts of Pennsylvania, having been virtually extinct throughout the state for many years.





An Adaptive Methodology

- **Step 1:** Behaviorally assess the cybercriminal, including their attitudes, life experiences, and relationships
- **Step 2:** Consider the victimology of that cybercriminal's targeting and how that cybercriminal behaved throughout an attack cycle
- **Step 3:** Come up with a design plan for a deceptive approach that incorporates simulation and dissimulation that you believe the cybercriminal will be most behaviorally responsive to
- **Step 4:** Design/create or repurpose infrastructure and content to influence the cybercriminal inside and outside the network

Exercise 2: Application to Theoretical Scenarios

Pick one scenario. Come up with a brief design plan that is behaviorally based for a response to one of the following theoretical attacker scenarios. Apply the adaptive methodology to your proposed design and creation process. The design plan should include consideration of both infrastructure builds (i.e. Cowrie repurposing) and content creation (i.e. communication of a vulnerability to outside audiences, use of sock puppets, etc.).

- **Scenario 1:** Repeated Business Email Compromise (BEC) Attempts
    - Variations of business email compromise attempts are reported in your organization, generally targeting your finance department personnel. Some emails appear to include typos in cyrillic.
        - BEC is a type of cyberattack where criminals use email fraud to deceive businesses into transferring money or sensitive information.
- **Scenario 2:** An Access Broker is Claiming They Have Network Access for Sale
    - An unknown access broker with limited activity on a dark web market has posted a starting bid for access to what he or she claimed is access to your network.
        - Access Broker is a type of cybercriminal or intermediary who specializes in gaining unauthorized access to networks, systems, or databases and then selling that access to other criminals, such as ransomware groups, hackers, or cyber espionage actors.
- **Scenario 3:** A Ransomware Gang has Pledged a Bounty for Admin Credentials to Your Network
    - A ransomware developer with dozens of affiliates has pledged a bounty award for anyone who can obtain admin credentials to your network.

Questions for Tim 🧐

You can ask questions for Tim in the Matrix room **tim-pappa-live-questions**.

Honeypots

Goal: to learn what honeypots are and how to install and use them.

Honeypots are "security resources whose value lies in being probed, attacked, or compromised"[8]. All honeypots share four key characteristics: they "are ***deceptive***, ***discoverable***, ***interactive*** and ***monitored***"[9]. They can be anything from a file to an entire operating system.

- **Any attempt** to communicate with the honeypot **is, by definition, an attack** since no legitimate users or applications should use the honeypot system[10].
- "By watching attackers break into and control a honeypot, we learn how [they] operate and why"[11]

Honeypots allow defenders to:

- **Track attackers:** from commands to operators
- **Learn attackers' actions:** new attacks, new exploits, new strategies
- **Gain knowledge:** who attacks, where attacks come from, and what the attacks are

Deploy your first honeypot!

Let's try to deploy our first honeypot from our containers and see what we can learn:

1. Login to your containers
2. Create a directory that we will share on the web server:
    a. `mkdir /tmp/publicshare/`
    b. `cd /tmp/publicshare/`
    c. `echo "Not a Honeypot" > /tmp/publicshare/README.md`
3. Create a Python web server
4. CTU students

[8] L. Spitzner (2002) Honeypots: Tracking Hackers. Addison-Wesley Longman Publishing Co., Inc.
[9] C. Sanders (2020). Intrusion Detection Honeypots: Detection Through Deception. Applied Network Defense.
[10] Supra note 2
[11] Supra note 1

   a. `screen -d -m -S FirstHoneypot bash -c ' python3 -m http.server 80 --directory /tmp/publicshare'`
      i. -d → daemon (detached) mode
      ii. -m → long formatting
      iii. -S → screen session name
      iv. bash -c '' → command to execute on the screen
      v. python3 -m http.server 80 → simple HTTP server
      vi. --directory /tmp/publicshare → share the content of this directory

5. Online students
   a. `tmux new-session -d -s FirstHoneypot bash -c 'python3 -m http.server 80 --directory /tmp/publicshare'`

6. CTU students let's check each other web servers. Online students, let's find web servers in your own docker:
   a. `nmap  -p 80 -sV 172.20.0.0/24`
      i. -p 80 → Only port 80
      ii. -sV → Attempt to identify the service version
   b. `nmap -p 80 -sV 127.0.0.1` - command for online students

7. Let's check our honeypot output:
   a. `screen -r FirstHoneypot`
   b. `tmux a -t FirstHoneypot` - command for online students

From the logs of our web server, shown below, we can learn which IP attacked us, when the connections happened, which methods were used, what HTTP resources were requested, and what was the status code of the response.

```
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
172.16.1.68 - - [01/Nov/2023 14:50:31] "GET / HTTP/1.1" 200 -
172.16.1.68 - - [01/Nov/2023 14:52:18] "GET / HTTP/1.0" 200 -
172.16.1.68 - - [01/Nov/2023 14:52:18] code 501, message Unsupported method ('POST')
172.16.1.68 - - [01/Nov/2023 14:52:18] "POST /sdk HTTP/1.1" 501 -
172.16.1.68 - - [01/Nov/2023 14:52:18] code 404, message File not found
172.16.1.68 - - [01/Nov/2023 14:52:18] "GET /nmaplowercheck1698850338 HTTP/1.1" 404 -
172.16.1.68 - - [01/Nov/2023 14:52:18] "GET / HTTP/1.0" 200 -
172.16.1.68 - - [01/Nov/2023 14:52:18] code 404, message File not found
```

What are the problems and limitations of this basic honeypot? To start with, it runs as root, it is not separated from the real system, it is easy to identify as honeypot, and logs are not stored. There are many more problems.

Break II (16:30-16:40)

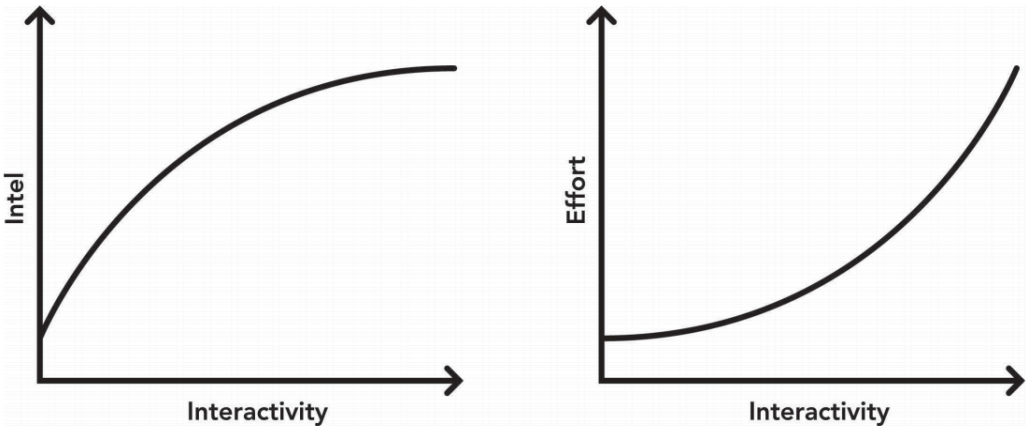
Characterizing Honeypots (16:40)

Honeypots, as we defined them, can take many forms. We can distinguish between:

- **Honey systems:** mimics an entire operating system with all its underlying services
    - Linux or Windows OS, power grid control systems, navigation systems
- **Honey services:** mimics a specific protocol or software
    - SSH, SMB, RDP, Telnet, smart contracts, etc.
- **Honey tokens:** mimic specific data or real data modified to inform about attack
    - Documents (PDFs, Word, etc), users, passwords (honeywords), tables in a DB, URLs, special credit card numbers etc.

Honeypots can present various levels of interaction. We differentiate between:

- **High interaction honeypots:** the honeypot allows attackers to perform any actions the same way as a real system would. High-value intelligence. High risk.
- **Low interaction honeypots:** the honeypot allows attackers very limited actions. Lower value intelligence—low risk.
- **Medium interaction honeypots:** anywhere in between the other two extremes. This is often the most common type of honeypot, balancing intelligence collection and risk.



Relationship between interactivity, intelligence value, and defender effort[12].

❓ What type and what interaction level did the honeypot we deployed before have?

Let's do an experiment!

- You need just your browser to play this. No other tool is needed.
- Both CTU students and online people - go to this link
  - https://shellsevaluation.thesis.stratosphereips.org/
- Leave your email address in the "email address" field
- When you submit your email address, you will be randomly assigned a system
  - Real Ubuntu or a Honeypot
    - Response times of both were modified to simulate a slow IoT system
  - Half of you will get a real Ubuntu system, and half of you will get a honeypot. It is randomly decided with a 50/50 chance.
- Your goal is **not to get detected by the honeypot** **and** to find a hidden crypto-wallet key.
- It is more important **not to get detected** than to find the key. But getting the key is the only way to win.
- You will have 10 minutes.

[12] C. Sanders (2020). Intrusion Detection Honeypots: Detection Through Deception. Applied Network Defense.

- If you find the key, put it in the submission field; Otherwise, select 'Give Up' option
- When you finish, there will be a survey for you to fill
- Wait for a few seconds for the survey to load
- **Do not reload the page before you fill in and submit the survey!**
- Please do not communicate with your colleagues throughout the game

Let's take a look at some results that we are getting right now.

- We use Google Forms for survey so let's check results

Thank you for participating!

What are Honey Systems

Honey systems are complex and rare as they require a lot of effort to set up, and monitor, and they are full systems that often introduce higher risks. The easiest cases include virtual machines with the OS exposed to the internet.

Let's Explore Honey Tokens

- **Active Directory users**[13]: create and inject fake users in the Active Directory structure, luring attackers away from other targets. Early detection of attacks.
- **Server users**: create fake users in a system and monitor logins to that account.
- **URL tokens**[14]: create fake URLs to use as early warning systems, detect insider threats, data exfiltration, and more:
- Let's visit this link and create a URL honey token
    - https://canarytokens.org
    - Token: http://canarytokens.com/images/traffic/npqmbiiu59t2ar6d8qkgnojal/post.jsp

[13] Lukas, O. and Garcia, S. (2021). Deep Generative Models to Extend Active Directory Graphs with Honeypot Users. *Proceedings of the 2nd International Conference on Deep Learning Theory and Applications*, [online] pp.140–147. doi:https://doi.org/10.5220/0010556601400147..
[14] Thinkst Applied Research (2024). *Know. Before it matters*. [online] Canarytokens. Available at: https://canarytokens.org/nest/generate [Accessed 22 Oct. 2024].

    - Logs: https://canarytokens.org/nest/history/d26e9d336ea5d1e8d54992f79d07c1f1/npqmbiiu59t2ar6d8qkgnojal

- Many, many more! Explore!

Let's Explore Honey Services

The Infinite Webpage

The Infinite Webpage[15] is a simple honeypot web server that delivers an infinite web page to anyone asking anything from it. It has "sticky" properties that aim to retain the attacker as much as possible on the website and also fill its entire disk if the download of content is automatic.

Cowrie: Telnet and SSH

"Cowrie is a medium to high interaction SSH and Telnet honeypot designed to log brute force attacks and the shell interaction performed by the attacker"[16]. It's highly customizable and probably one of the most used honeypots in the industry.

Cowrie can give us information about:

- **Who is attacking**: IP address, human, bot, etc.
- **What credentials** were used for attacking?
- **What actions** do the attackers attempt to do on the honeypot: commands, downloads, reverse shells, scripts, lateral movement, etc...

Installing Cowrie

Let's install and run Cowrie in our containers to try and experience it:

- Install dependencies - already done in your dockers
    - `apt-get install git virtualenv libssl-dev libffi-dev build-essential libpython3-dev python3-minimal authbind`
- Add the non-root cowrie user
    - `adduser --disabled-password cowrie`
- Change to user Cowrie: the rest of the actions will be run as this user

[15] Stratosphere Laboratory (2015) stratosphereips/theinfinitewebpage [online] Available at: https://github.com/stratosphereips/theinfinitewebpage. Accessed on October 22, 2024.
[16] Cowrie (2015) Cowrie SSH/Telnet Honeypot [online] Available at: https://github.com/cowrie/cowrie. Accessed on October 22, 2024.

    - `su - cowrie`
- Clone the Cowrie repository:
    - `git clone http://github.com/cowrie/cowrie`
    - `cd cowrie`
- Create a virtual Python environment:
    - `virtualenv --python=python3 cowrie-env`
        - As root: `apt install virtualenv`
    - `source cowrie-env/bin/activate`
- Install more packages
    - `python3 -m pip install --upgrade -r requirements.txt`
- Create a Cowrie configuration file:
    - `cp etc/cowrie.cfg.dist etc/cowrie.cfg`
- Edit the new configuration file:
    - `vim etc/cowrie.cfg`
    - Enable Telnet:
        - Search for the [telnet] section (line 653)
            - In vim just press ":653" to go to line 653
        - Change enabled = false to enabled = true

```
652 # Enable Telnet support, disabled by default
653 enabled = true
654
```

- Start Cowrie
    - `bin/cowrie start`
    - /home/cowrie/cowrie/bin/cowrie start
    - There are other commands: status, stop, restart
- Cowrie will run in the background in ports 2222/TCP (SSH) and 2223/TCP (Telnet)
    - Check that it is running: `netstat -anp`

        - `-a` → show both listening and non-listening sockets
        - `-n` → numeric values
        - `-p` → show the associated program or service
    - Check that it is running: `ps afx`

Playing with Cowrie

Find a friend to attack you in the class. Share your IP address in the Matrix chat, and be sure somebody attacks you. If nobody attacks you, tell us! Get some IP to attack!

- Connect to Cowrie using SSH (user: root, password: *try any password*)
    - `ssh root@<friends' IP> -p 2222`
    - `ssh root@127.0.0.1 -p 2222 - online students`
- Connect to a Cowrie using Telnet
    - `telnet <friends' IP> 2223`
    - `telnet 127.0.0.1 2223 - online students`
    - `ncat <friends' IP> 2223`
    - `ncat 127.0.0.1 2223 - online students`
- What do you do?

Cowrie's Logs and TTY sessions

Cowrie's logs are located at var/log/cowrie:

- Access the logs:
    - `cd /home/cowrie/cowrie/var/log/cowrie`
        - `cat cowrie.log | less -S`
        - `cat cowrie.json |less -S`
- Search for new connections:
    - Search for "New connection" in less. Use /

        - `cat cowrie.log | less -S`
        - `/New connection`
    - `cat cowrie.log |grep -i "new connection" | less`
- See attacks in real-time:
    - `tail -f cowrie.log`

Cowrie also records the TTY sessions:

- Go to the Cowrie home folder:
    - `cd /home/cowrie/cowrie`
- Activate the virtual environment (only if you haven't done so already):
    - `virtualenv --python=python3 cowrie-env`
    - `source cowrie-env/bin/activate`
- Check that you have sessions to replay:
    - `ls -lh var/lib/cowrie/tty/`
- Replay one of your sessions:
    - `bin/playlog var/lib/cowrie/tty/<id>`

What is the problem with honeypots?

- Identification and Fingerprinting[17] [18]
- Vulnerabilities
- Do they attract more attacks? Nobody knows
- Would you put it in your production servers? Probably not.

Modifying Cowrie

Cowrie is highly configurable, and almost anything on it can be changed. Yes, including accepted users, passwords, commands, etc:
https://github.com/cowrie/cowrie

[17] Srinivasa, S., Pedersen, J.M. and Vasilomanolakis, E., 2021. Gotta catch'em all: a Multistage Framework for honeypot fingerprinting. *arXiv preprint arXiv:2109.10652*.

[18] Vetterl, A. and Clayton, R., 2018. Bitter harvest: Systematically fingerprinting low-and medium-interaction honeypots at internet scale. 12th *USENIX Workshop on Offensive Technologies (WOOT 18)*. URL: https://www.usenix.org/conference/woot18/presentation/vetterl

Let's change the passwords that are valid to enter:

a. Go to the Cowrie home folder:
   - `cd /home/cowrie/cowrie`

b. First, copy the default User DB to a real one:
   - `cp etc/userdb.example etc/userdb.txt`

c. Second edit this file to add a new user:
   - `vim etc/userdb.txt`
   - Add a new line with:
     `bsy:x:*`

d. Restart the Cowrie service to load the new file and its changes:
   - `bin/cowrie restart`

e. Connect with SSH:
   - `ssh bsy@172.16.1.xx -p 2222`

CTU students clean-up (after you finish assignment 5!!!)

a. How to exit a virtual environment
   - `deactivate`

b. How to stop cowrie
   - `bin/cowrie stop`

c. How to delete a virtual environment: just delete the directory
   - `rm -r cowrie-env`

d. Delete the extra users!

shellM

Let's start by logging into this system and trying to answer these questions.

ssh -p 1337 tomas@olympus.felk.cvut.cz
password: tomy

What can you tell us about this system?

- Is this a honeypot? If yes, what gave it away?
- Is this a honey system or a honey service?
- What level of interaction would you say the honeypot has?

We can try some commands like:

- `ls`
- `cd Desktop`
- `touch testfile.txt`
- `echo “Let’s try this!” >> testfile.txt`
- `cat testfile.txt`
- `cd /`
- `cat /etc/passwd`
- `cat ~/Desktop/test.txt`

👾 Try some more. Play with it and share your thoughts or findings in the Matrix chat or in person if you are in the class.

shelLM is a honeypot framework to generate believable honeypot environments dynamically on the fly using Large Language Models (LLMs).[19][20]

- The user never interacts with a real system.
- LLM can be taught to mimic any service.
- The more real the behavior mimicked by the LLM the more users interact with it.

We can go beyond shelLM. How about a tool for Honey services? Well, lucky for you, we have it. It is called **VelLMes** (Slavic god of, amongst other things, trickery -Veles; + LLMs). shelLM is a part of it, but also MySQL, POP3, and HTTP.

- https://github.com/stratosphereips/VelLMes-AI-Honeypot/tree/main

[19] Sladić, M., Valeros, V., Catania, C., & Garcia, S. (2024, July). Llm in the shell: Generative honeypots. In 2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW) (pp. 430-435). IEEE.
[20] Sladić, Muris (2023) LLM in the Shell: Generative Honeypots Demo [online] Available at: https://www.youtube.com/watch?v=0ysdHanr-jA. Accessed on November 1, 2023.

Recap: What makes a honeypot successful?

A honeypot is successful when attackers probe it, attack it, and access it. Honeypot deception can be planned using the see-think-do deception methodology[21].

- **SEE**: we want attackers to find and see the honeypot services, systems, and tokens.
- **THINK**: we want attackers to think these honeypots are valuable targets.
- **DO**: we want attackers to use the systems to do some interactions with the systems.

If defenders control all three of these, they have a high chance of detecting intruders early and increasing the intelligence collected from attackers.

Extra: Honeypots we love

- Dionaea: is a low-interaction honeypot that captures attack payloads and malware[22]
- Glutton: proxy honeypot to log all attacks between the internet and other honeypots[23]
- All-in-one T-Pot: multi-honeypot containerized framework. Dozens of honeypots. IDS/IPS. Monitoring[24].
- All-in-one Chameleon: customizable honeypots to gather intelligence. 19 honeypot services are supported.

[21] JP 3-13.4 (2012) Military Deception [online] https://info.publicintelligence.net/JCS-MILDEC.pdf. Accessed on 11/01/2023.
[22] Dionaea – Catching bugs – The Honeynet Project, https://www.honeynet.org/projects/active/dionaea/
[23] An analysis of Glutton — All Eating honeypot | by Muhammad Tayyab Sheikh (CS Tayyab) | Medium, https://cstayyab.medium.com/an-analysis-of-glutton-all-eating-honeypot-625adf70a33b
[24] Installing T-Pot Honeypot Framework in the Cloud — Stratosphere IPS, https://www.stratosphereips.org/blog/2020/10/10/installing-t-pot-honeypot-framework-in-the-cloud

Assignment 5 (6 Points)

1. You must have a Cowrie installation at your docker running **SSH on port 2222** (Like we did in class).
2. Attackers will try to connect to this port and detect if it is a honeypot or a real service.
3. You need to try and make the honeypot look more realistic.
4. More details in CTFd

Class Feedback

By giving us feedback after each class, we can make the next class even better!

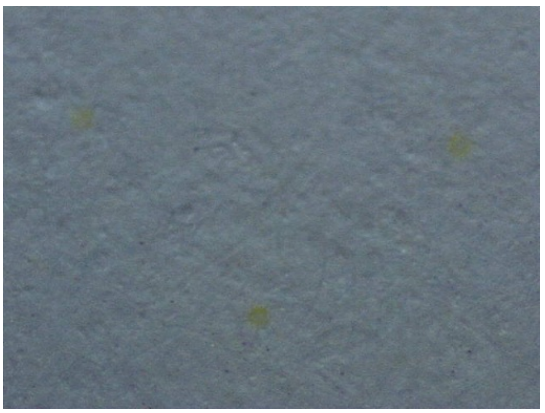
bit.ly/BSYFeedback

Side Dishes

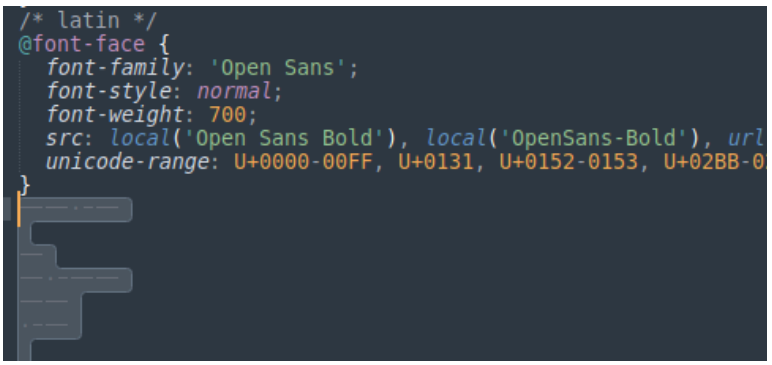
More on Steganography: History and uses

- The first reference was in 440 BC in Greece when Herodotus mentions two examples in his Histories. Histiaeus sent a message to his vassal, Aristagoras, by shaving the head of his most trusted servant, "marking" the message onto his scalp, and then sending him on his way once his hair had regrown.
- Mostly military use since then.
- Invisible ink.
- Morse code in yarn knitting. Common in WWII. Here.
- In documents and prints, the message could be hidden by using two or more different typefaces, such as normal or italic.
- During and after World War II, espionage agents used photographically-produced microdots to send information back and forth.
- Velvalee Dickinson was a US citizen spying for Japan. She embedded info in the manufacturing orders of dolls shipped to Argentina. It was discovered when the Argentinian moved, and the letters bounced back. [25]
- Jeremiah Denton repeatedly blinked his eyes in Morse code during the 1966 televised press conference that he was forced into as an American prisoner-of-war by his North Vietnamese captors, spelling out "T-O-R-T-U-R-E".
- 911 Attack: “bin Laden and others ‘are hiding maps and photographs of terrorist targets and posting instructions for terrorist activities on sports chat rooms, pornographic bulletin boards and other websites, U.S. and foreign officials say.’”
    - https://www.wired.com/2001/02/bin-laden-steganography-master/
- And in your home, too
    - Some modern printers use steganography, including Hewlett-Packard and Xerox brand color laser printers. The printers add tiny yellow dots to each page. The barely visible dots contain encoded printer serial numbers and date and time stamps. (https://www.eff.org/press/archives/2005/10/16 , and https://w2.eff.org/Privacy/printers/docucolor/)

[25] https://en.wikipedia.org/wiki/Velvalee_Dickinson



- Malware Attacks in 2021!
  - Malware is hidden inside images. (Example)
  - Used steganography to embed RATs within the embedded images.
- LokiBot: The Famous Image Steganography Attack
  - LokiBot malware uses steganography to hide its malicious files.
  - The malware installs itself as two files: a .jpg file and a .exe file.
  - The .jpg file opens, unlocking data that LokiBot needs when implemented.
  - The malware places the image and the .exe file into a directory that it creates, along with a Visual Basic script file that runs the LokiBot file.
  - The script uses a decryption algorithm to extract the encrypted code from the image, enabling the VBScript file interpreter to execute the malware.
- Malware in CSS
  - It fetches a remote file called fonts.css, from which it retrieves JavaScript code (Example).
  - But how? The .css file contained a few lines of valid CSS code, but there were also non-visible characters such as spaces, tabs, and newlines.



Snow

It is a tool to hide **text** *in text*!

- `apt install stegsnow`
- `echo “oh my god” > in.txt`
- `stegsnow -C -m "Good Morning" -p "pass" in.txt out.txt`
  - -m message
  - -p password
- Can you see where the info is stored?
  - `cat out.txt`
- What about
  - `xxd -C out.txt`
- Lets read it
  - `stegsnow -C -p "pass" out.txt`

Differences with polyglots

What are polyglots? A person who can speak many languages. Well, in our case, it is a *file* that includes many languages and can be read/processed by many different tools.

For example, a **PDF** document that is also a **ZIP** archive and a **Bash script** that runs a **Python webserver** that hosts **Kaitai Struct’s WebIDE,** which allows you to **view the file’s *own* annotated bytes.** (Here)

Or a **PDF** that's also a valid **ZIP** and a valid **firmware** for the Apollo Guidance Computer. (Here).

Let's try a hands-on example with an image:

- We will analyze the following image: https://twitter.com/David3141593/status/1057042085029822464
- Download it to your container:
  - `wget "https://pbs.twimg.com/media/DqteCf6WsAAhqwV?format=jpg&name=120x120" -O image.jpg`
- Check the type of file downloaded:
  - `file image.jpg`
- Find the content by analyzing it byte by byte with binwalk
  - `binwalk image.jpg`
- Unzip the JPG

    - `mkdir test`
    - `mv image.jpg test`
    - `cd test`
    - `zip -FFv image.jpg --out image-fix.jpg` (to fix)
    - `unzip image-fix.jpg`
- Unrar the files that you got from the ZIP that you got from the JPG
    - `unrar x shakespeare.part001.rar`

More Details about The Infinite Webpage

The Infinite Webpage[26] is a simple honeypot web server that delivers an infinite web page to anyone asking anything from it. It has "sticky" properties that aim to retain the attacker as much as possible on the website and also fill its entire disk if the download of content is automatic.

- Clone the repository
    - `git clone https://github.com/stratosphereips/theinfinitewebpage.git`
    - `cd theinfinitewebpage`
- Create a virtual Python environment[27]:
    - `virtualenv --python=python3 theinfinitewebpage-env`
    - `source theinfinitewebpage-env/bin/activate`
- Install more packages
    - `pip install -r requirements.txt`
- Run the honeypot:
    - `screen -d -m -S theinfinitewebpage bash -c 'python3 /root/theinfinitewebpage/the_infinite_website.py'`
- Check the logs:
    - `tail -f theinfinitewebsite.log`
- Connect to the webserver of a random student:
    - `wget "http://172.20.0.$(( ( RANDOM % 94 ) + 1 )):8800"`
- Attach and stop the screen:
    - `screen -r theinfinitewebpage`
    - `CTRL+C`
- Deactivate the virtual environment:

[26] Stratosphere Laboratory (2015) stratosphereips/theinfinitewebpage [online] Available at: https://github.com/stratosphereips/theinfinitewebpage. Accessed on November 1, 2023.
[27] https://docs.python.org/3/library/venv.html

    - `deactivate`

Heralding: Multiservice Credential Harvesting

Heralding[28] is a low-interaction honeypot dedicated to credential harvesting across 13 different services, including SSH, Telnet, HTTP, POP3, etc.. Let's try it in our dockers!

- Install dependencies (already done in your containers)
    - `apt install libpq-dev`
- Clone the repository
    - `git clone https://github.com/johnnykv/heralding.git`
    - `cd heralding`
- Create a virtual Python environment[29]:
    - `virtualenv --python=python3 heralding-env`
    - `source heralding-env/bin/activate`
- Install more packages
    - `pip install -r requirements.txt`
- Install heralding
    - `pip install heralding`
- Copy and edit the configuration file
    - `cp heralding/heralding.yml .`
    - `vim heralding.yml`
        - In the SSH section change the port to 2228
- Make the binary executable:
    - `chmod +x bin/heralding`
- Start the honeypot
    - `screen -d -m -S heralding bash -c '/root/heralding/bin/heralding'`
- Let's attack each other:
    - `ssh user@172.20.0.xx -p 2228`
    - `ftp 172.20.0.xx`
    - `ncat 172.20.0.xx 25`
    - Try other services: telnet, http, https, pop3, pop3s, imap, imaps, vnc, postgresql, and socks5.
- View the logs:
    - `tail -f log_auth.csv`
    - `tail -f log_session.json`

[28] Johnnykv (2016) heralding: Credentials catching honeypot [online] Available at: https://github.com/johnnykv/heralding. Accessed on October 24, 2024.
[29] Python documentation. (2024). *venv — Creation of virtual environments*. [online] Available at: https://docs.python.org/3/library/venv.html [Accessed 24 Oct. 2024]