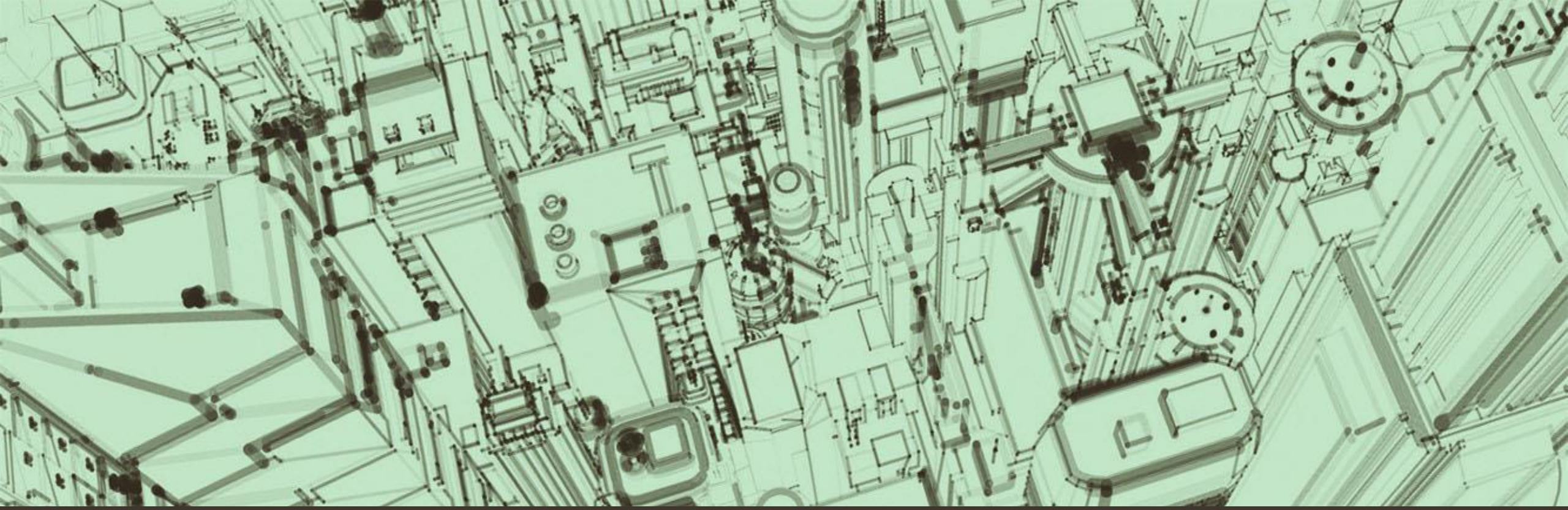


# CTI:A Threat Intel Story Telling

Tas / @tas-kmanager

# What we gonna talk about…

- The role of CTI
- Common workflows (reactive vs proactive)
- Skills and tools
- What it looks like on real world applications

# APT-GarudaPoutine

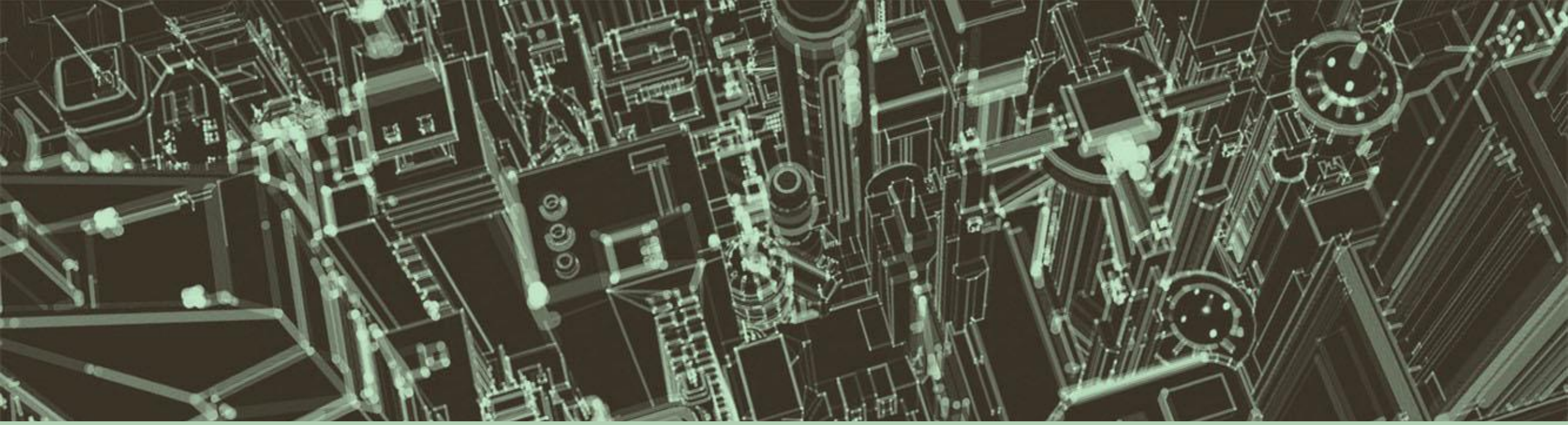
Tas / @tas-kmanager on ISS Discord

Sheridan ISS Graduate

## Affiliations

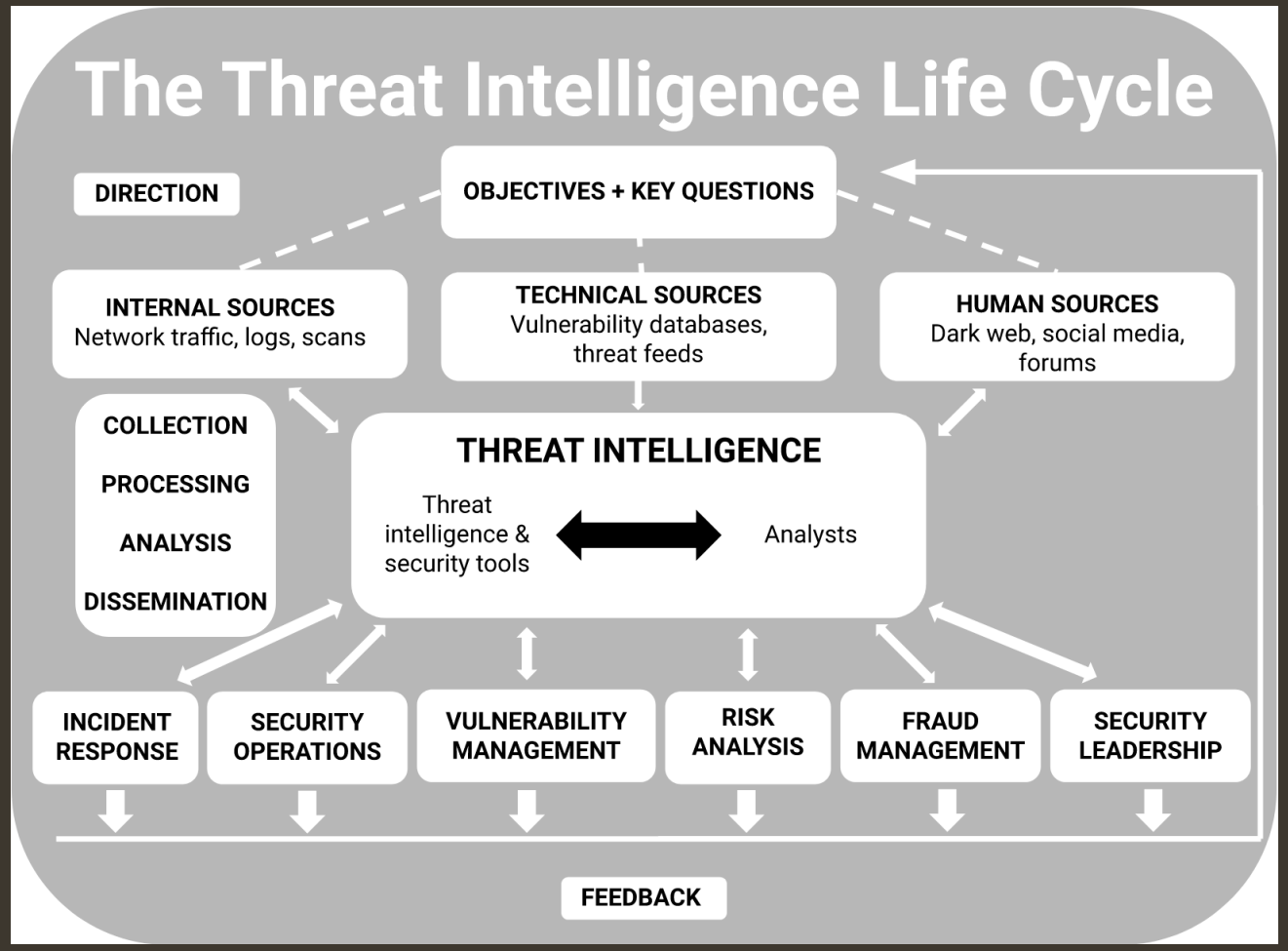
- Microsoft Securi arch – Cloud Security
- Curated Intel
- The DFIR
- CDEF.ID

Gaming, Cooking and Travel!



# The role of CTI

# Where does CTI sits?




https://www.sentinelone.com/cybersecurity-101/threat-intelligence/cyber-threat-intelligence/

# 4 Product Types of Threat Intelligence

**Tactical**

- TTPs of Threat Actor
- Detection and Hunt

**Technical**

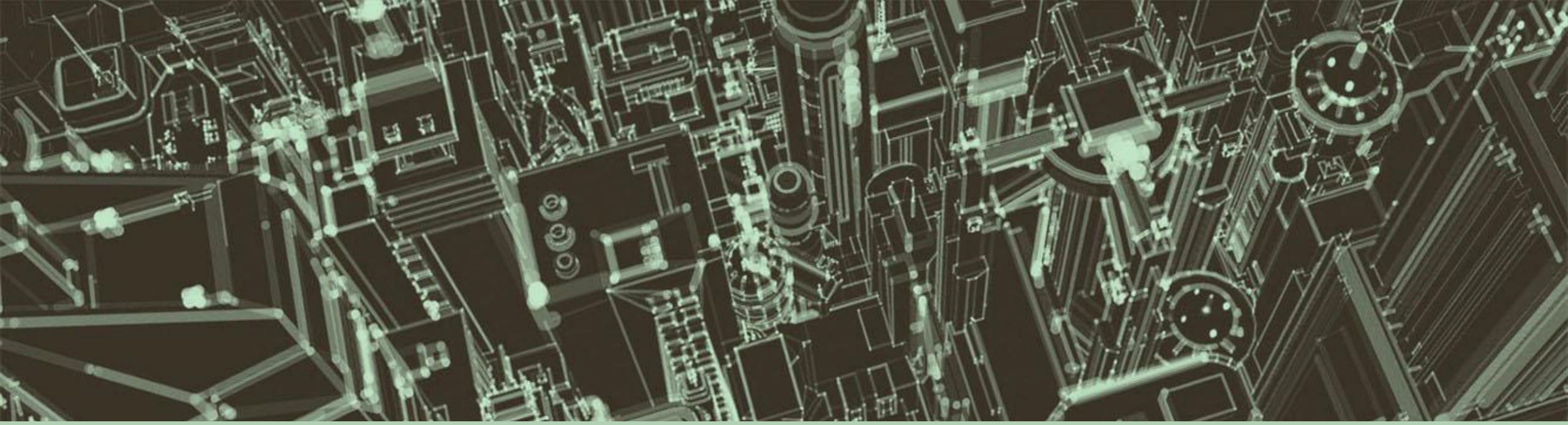
- IoCs
- Detection, Hunt and IR

**Operational**

- Nature, motive, timing, and how an attack is carried out
- Blue Team

**Strategic**

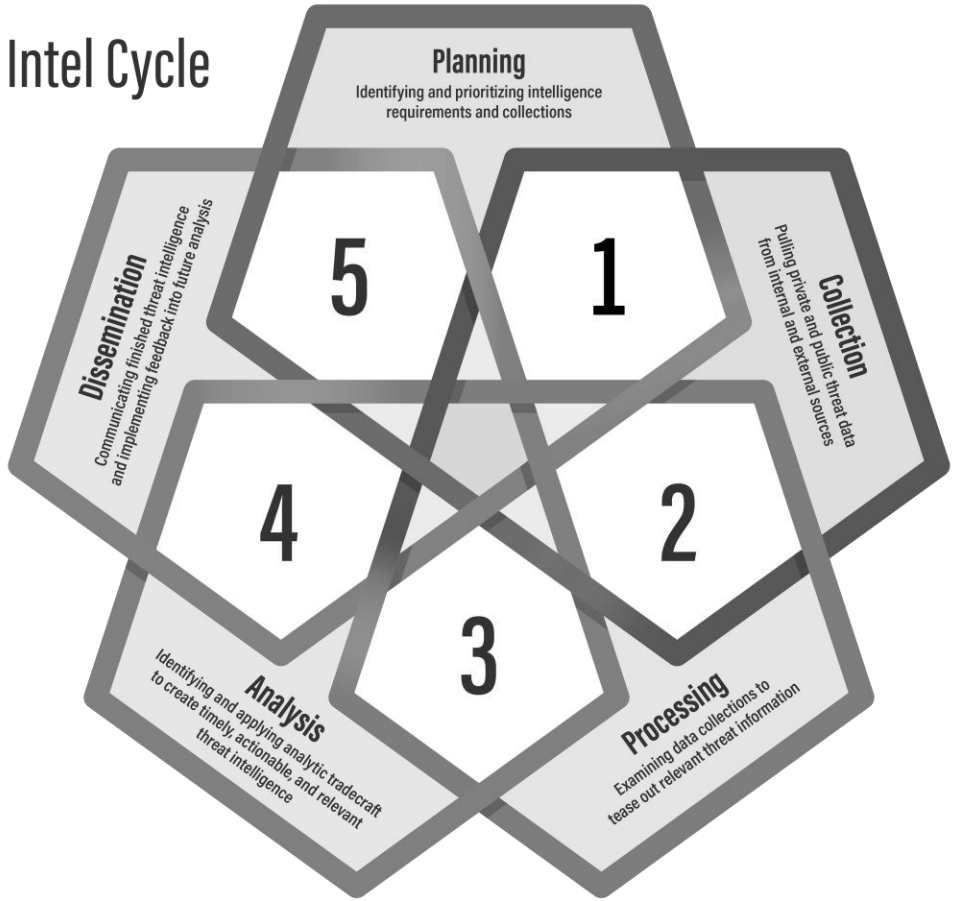
- High level analysis on trends
- Board & C levels



# Common Workflows

# The CTI Cycle

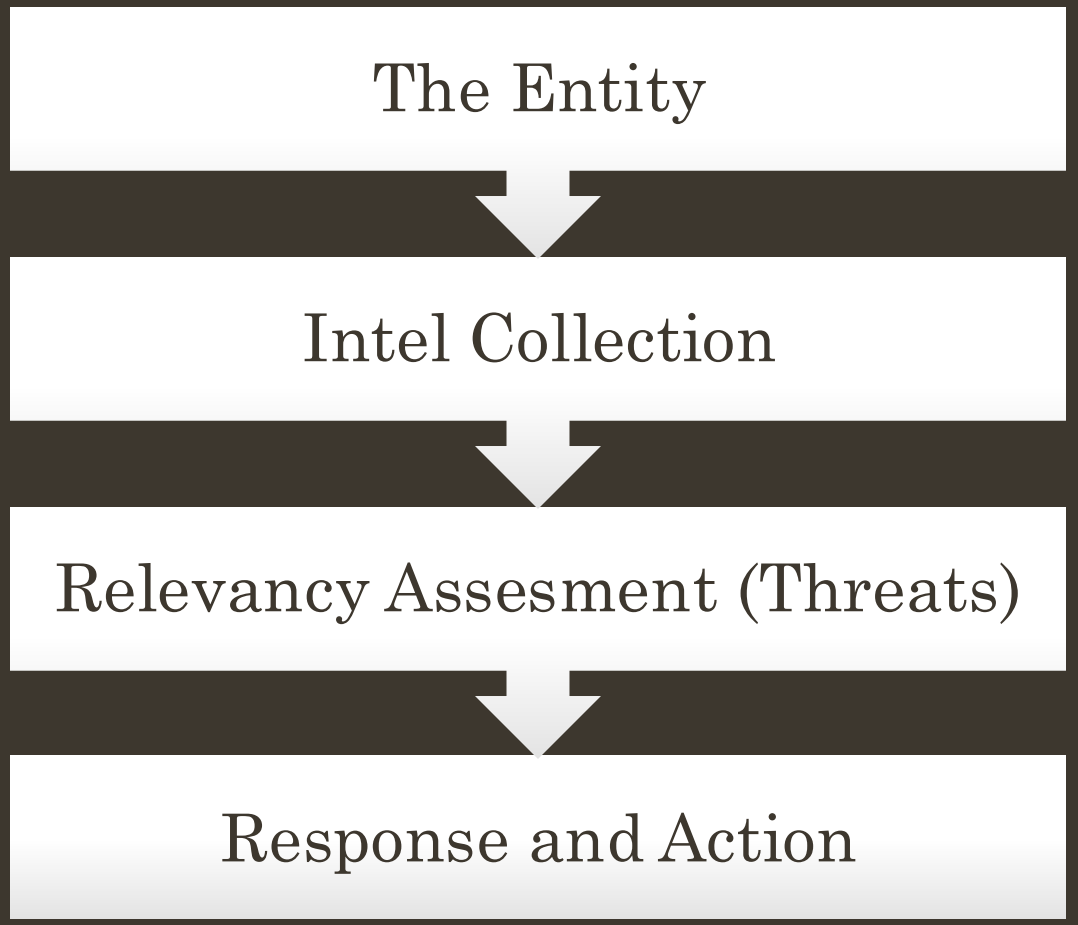
1.Planning
2.Collection
3.Processing
4.Analysis
5.Dissemination




https://www.cisecurity.org/insights/blog/what-is-cyber-threat-intelligence

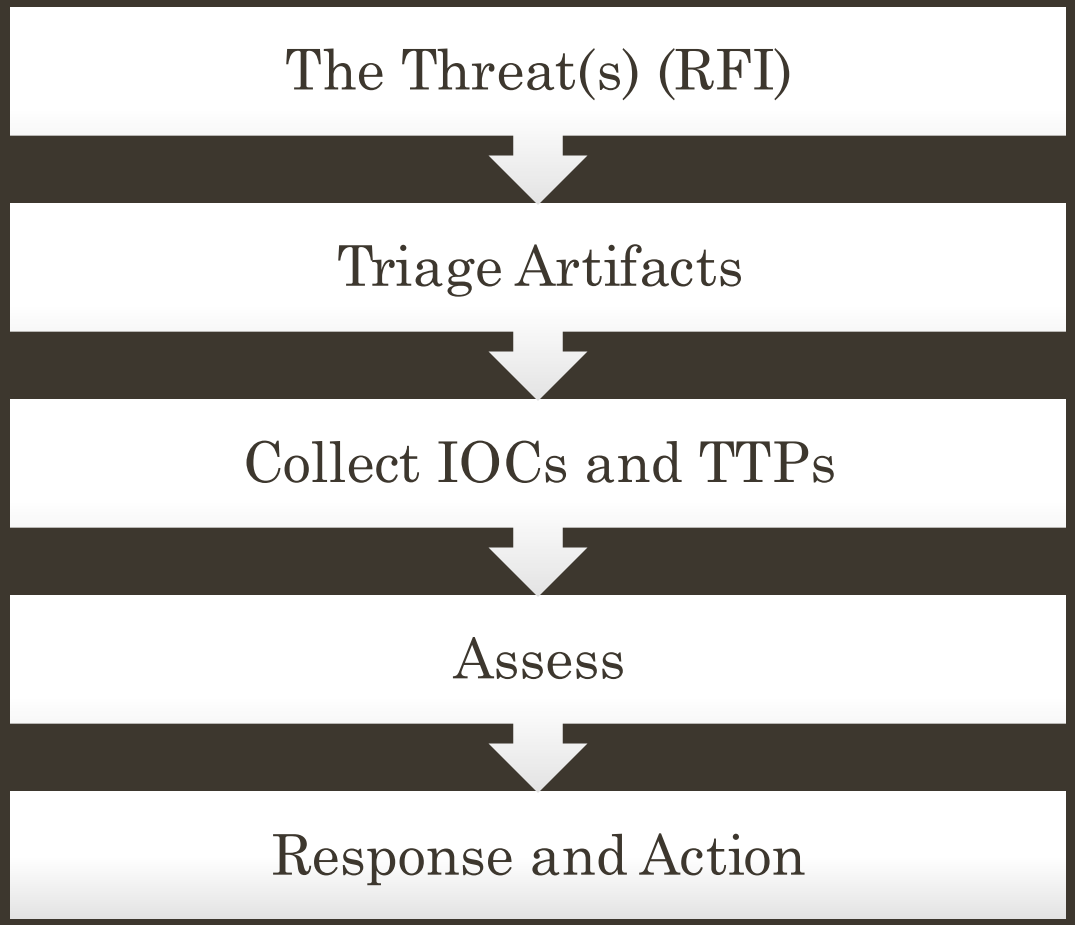
# Proactive CTI

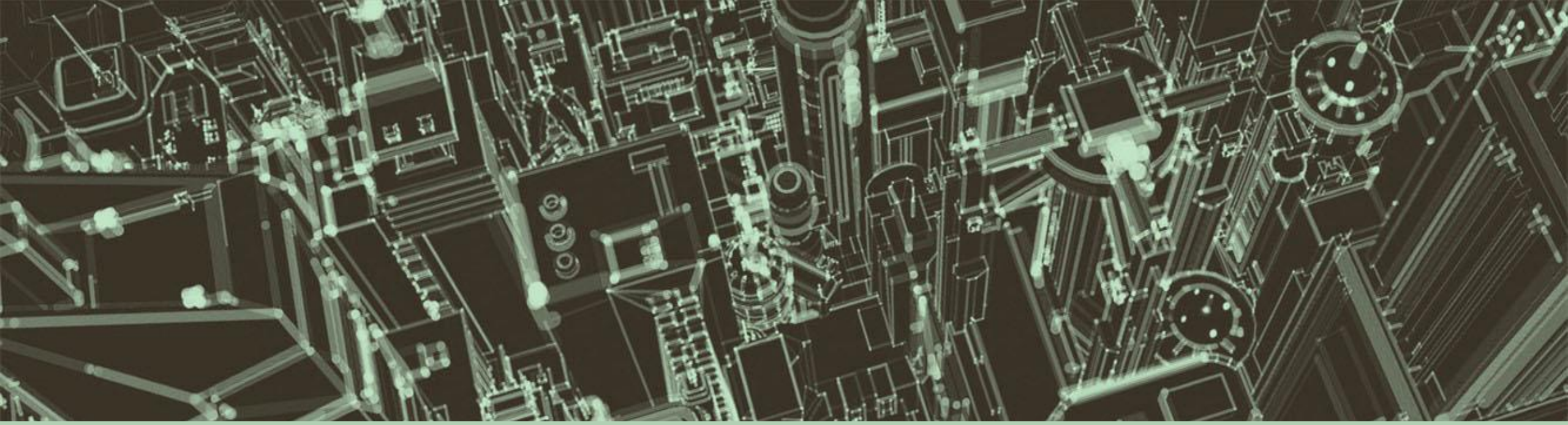
CTI initiated based on certain organization request to assess potential threats



# Reactive CTI

CTI initiated after an incident already occurred





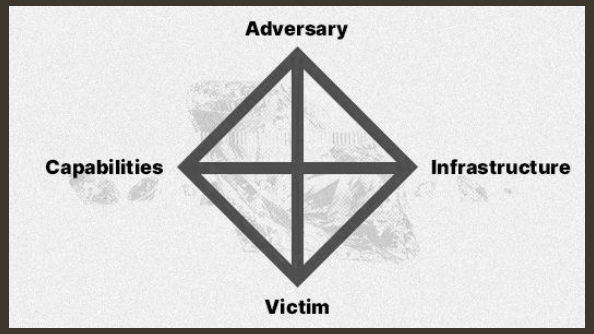
# Skills & Tools

# Tools and Frameworks

- OSINT Tools
  - Never ending list of tools, always something new
  - Awesome-threat-intelligence
  - Open source tools for CTI

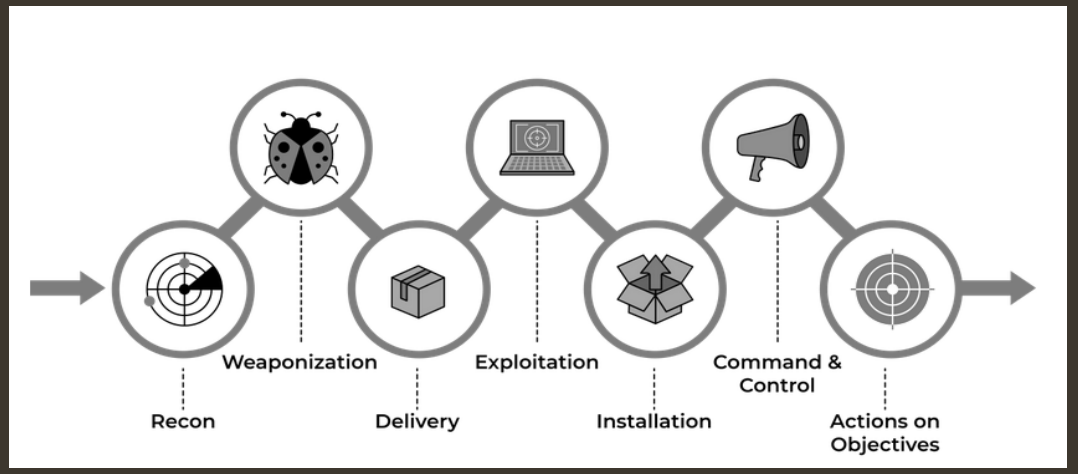
- Twitter / X
- Telegram
  - Don't forget to OpSec

- Diamond Models

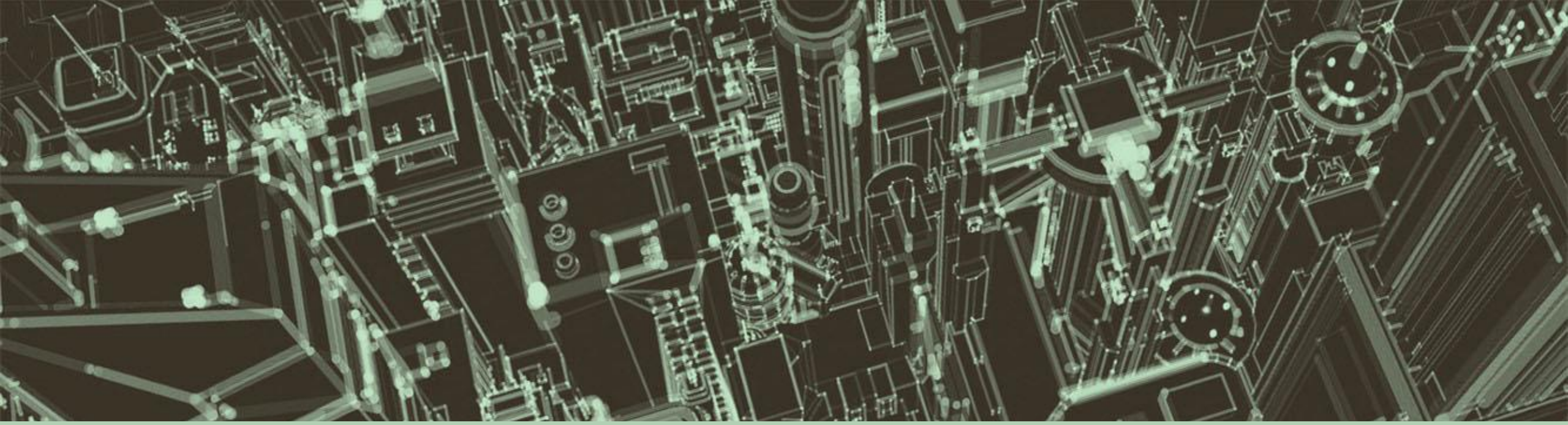



What is the Diamond Model of Intrusion Analysis? - Recorded Future

- Cyber Kill Chain

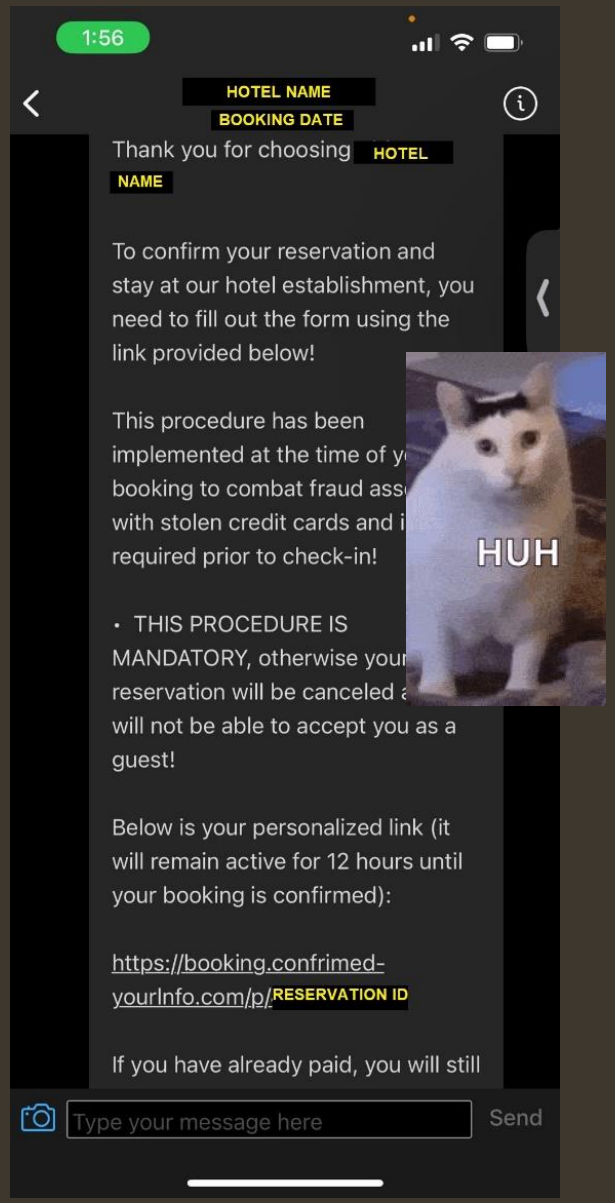


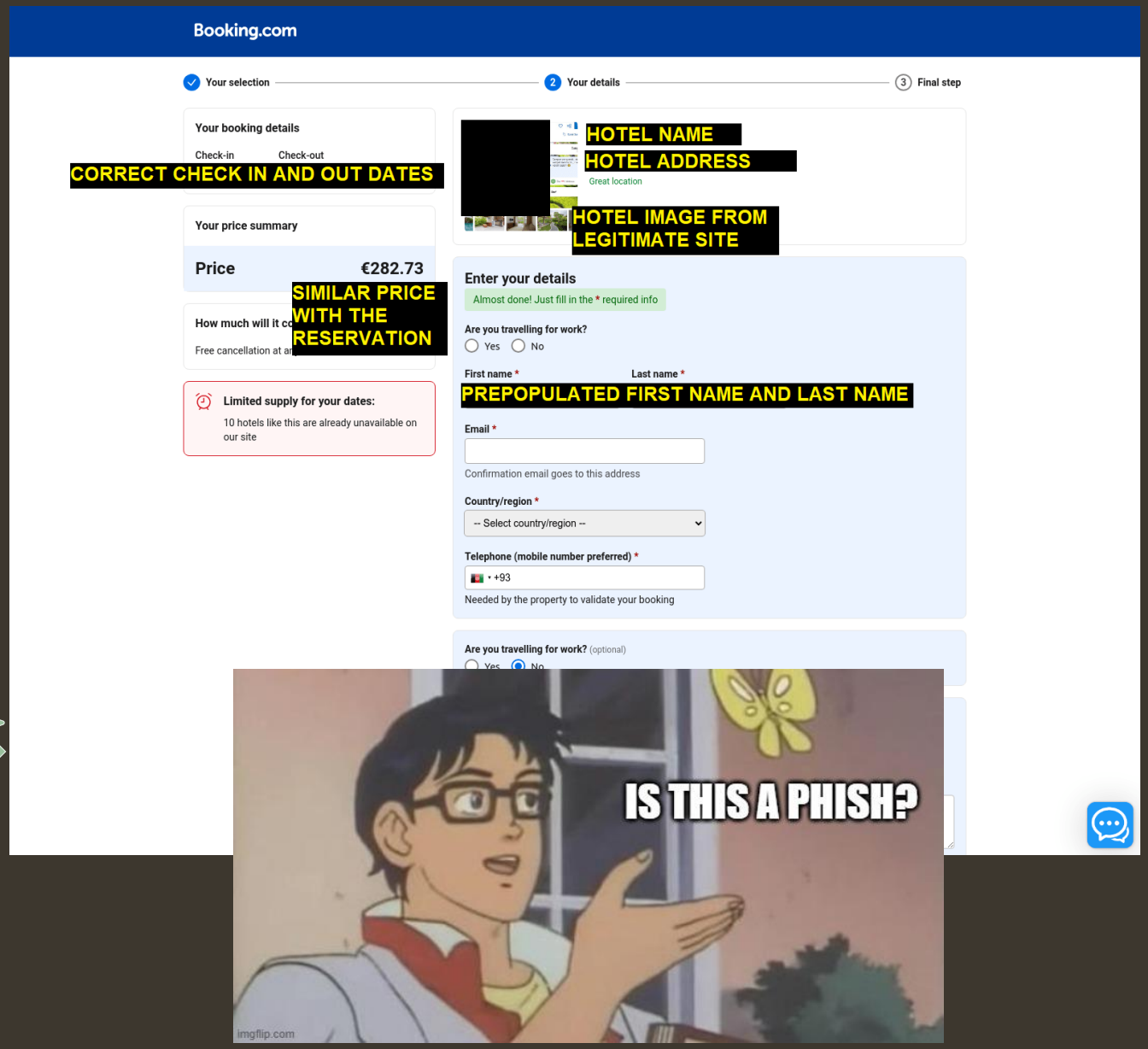

- MITRE ATT&CK



# Telekopye Phishing Platform

# The Threat (RFI)




# Official Chat Function

# Not Once or Twice, But 5 Times!

Germany
France
Japan
Indonesia

For the next few bookings made on the app, I received 5 phishing attempts
From 5 different hotels, in 4 different countries

## Two Thoughts Come To Mind:

1. Booking.com (Platform) is compromised

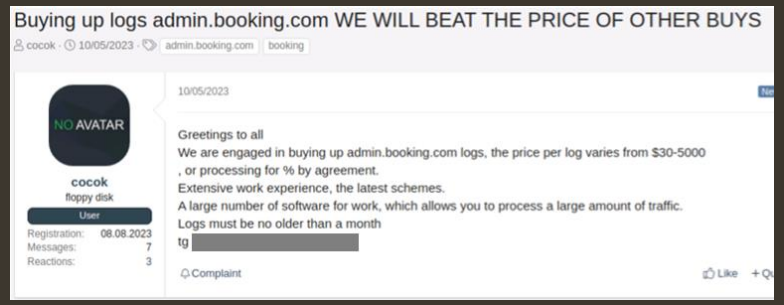
**OR**

2. Merchants are compromised

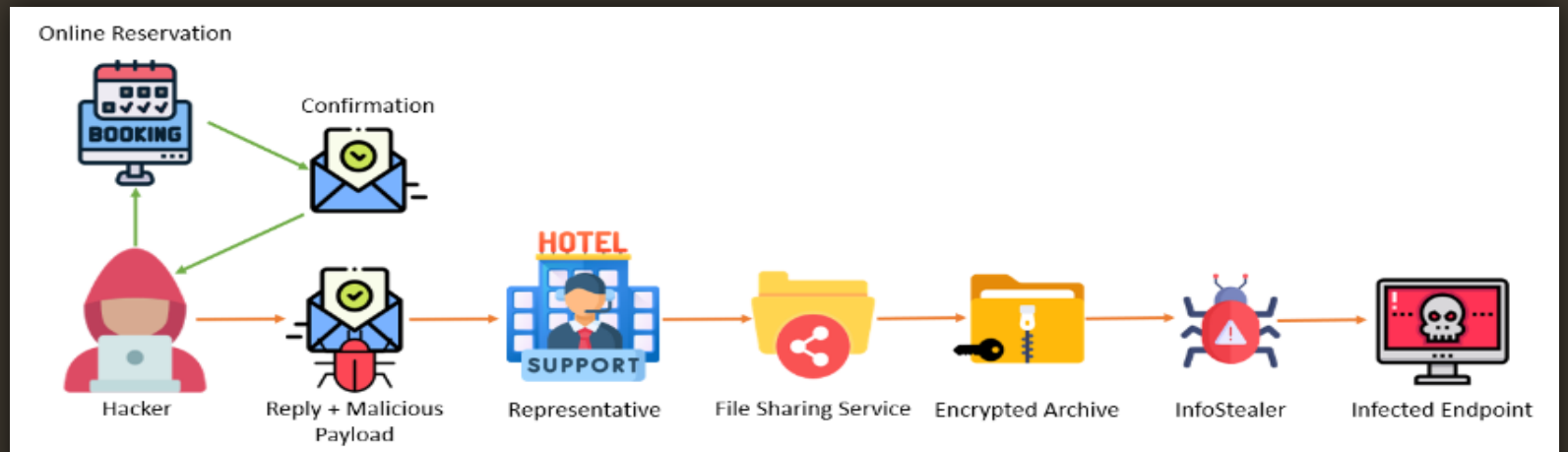
## The Missing Part:

Perception-Point researchers have confirmed that there are currently major credential theft campaigns against the merchants, in this case hotel providers

SecureWorks researchers have confirmed that stealer being used is Vidar Infostealer and on HIGH DEMAND



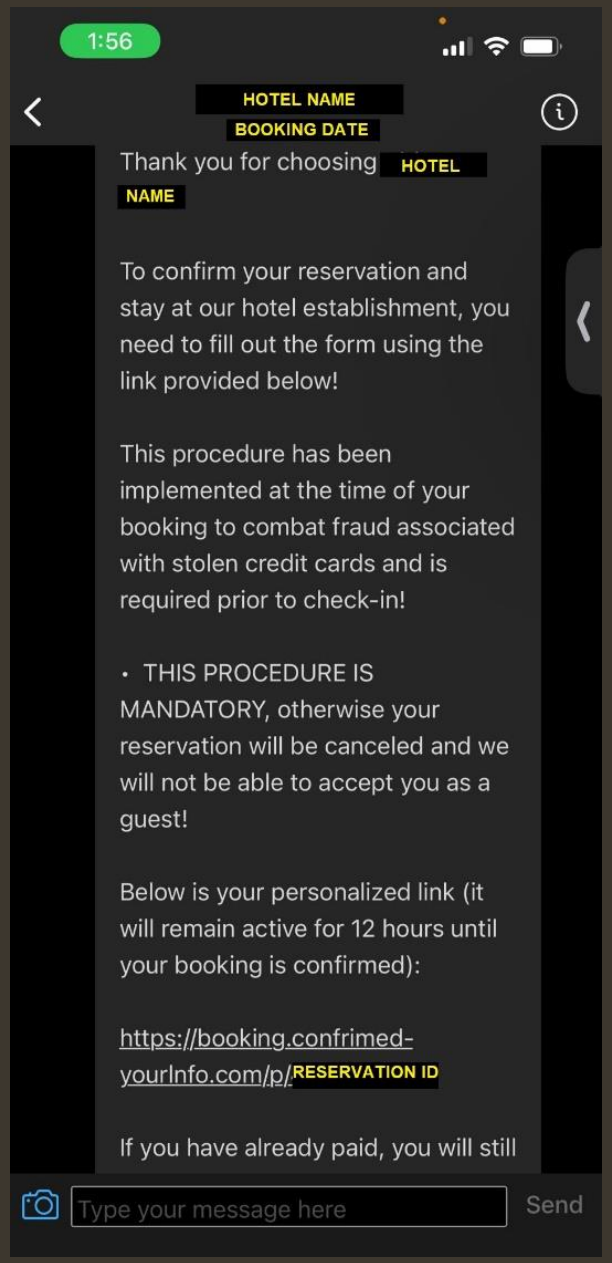

https://www.secureworks.com/blog/vidar-infostealer-steals-booking-com-credentials-in-fraud-scam




1. Initial Booking: The attacker makes an online booking, often with free cancellation to minimize risk.
2. Crafting the Reply: Using the confirmation email, the attacker replies with a specific request or question about the reservation.
3. Adding the Lure: The attacker attaches a link to a file-sharing service, providing a plaintext password for "important files."
4. Baiting the Victim: The email is directed either to the online booking service's support or directly to the hotel.
5. The Trap: The representative, believing the email to be genuine, downloads an archive from the provided link, which, when opened, releases Infostealer-type malware into the system.

https://perception-point.io/blog/booking-com-customers-hit-by-phishing-campaign-delivered-via-compromised-hotels-accounts/

# Triaging Artifacts




# What Do We Have?
# Phishing Chat Message!

## Common Phishing Characteristics:

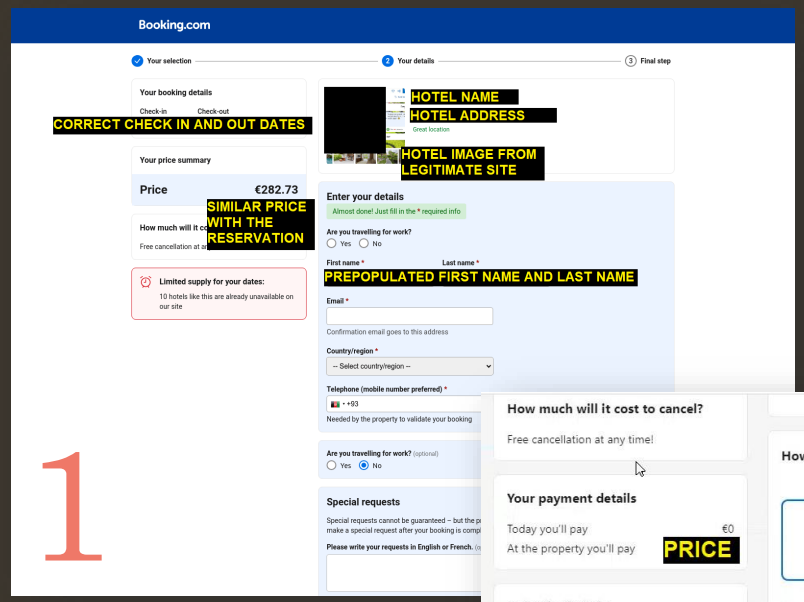
- The threat actor was using urgent, authoritative and threatening language
  - *"THE PROCEDURE IS MANDATORY"*
  - *"reservation will be canceled"*
  - *"it will remain active for 12 hours until your booking is confirmed"*
  - *"to combat fraud"* – **OH THE IRONY**
- Phishing domain related to booking.com
- Typo in the phishing domain

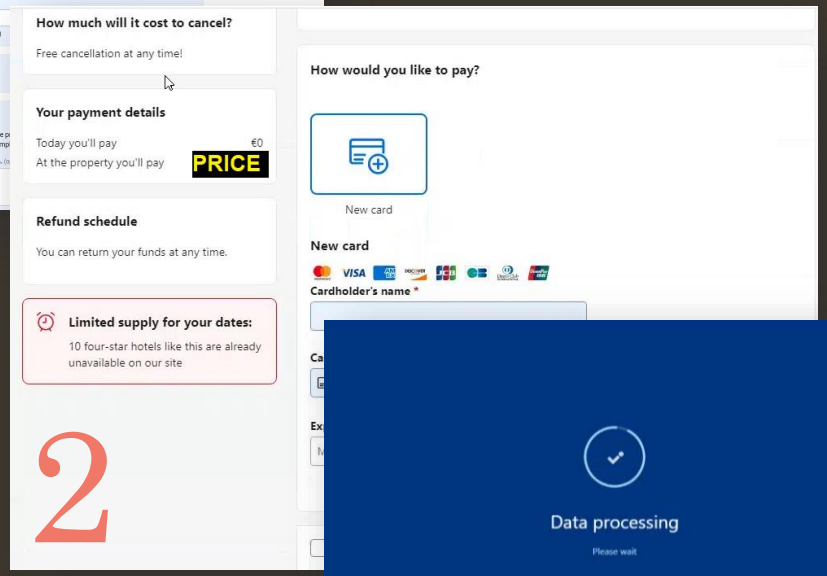
## Uncommon Phishing Characteristics:

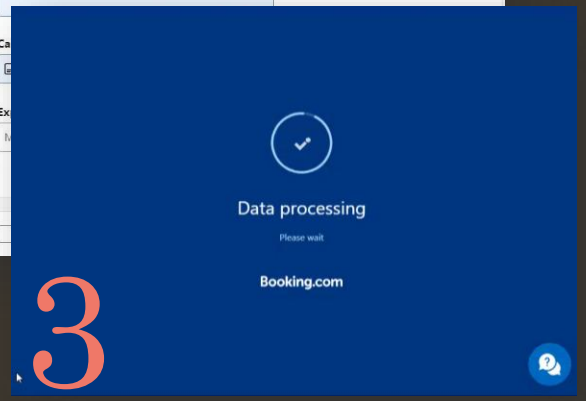
- The actor has lot of important details:
  - the hotel where the guests are staying,
  - the time of their stays,
  - the reservation ID (being used in the phishing link)
- The message was coming from the hotel merchant account in the official messaging platform of the Booking.com

The fake Booking website is resembling the legitimate Booking.com website
It has the right information related to the guests' booking:

- Hotel name
- Hotel address
- Hotel image (sometimes it's a screenshot of the hotel page in the Booking app, not the hotel itself)
- Check-in date and Check-out date
- Price (sometimes in the wrong currency)
- First and last name (came prepopulated and can't be changed)



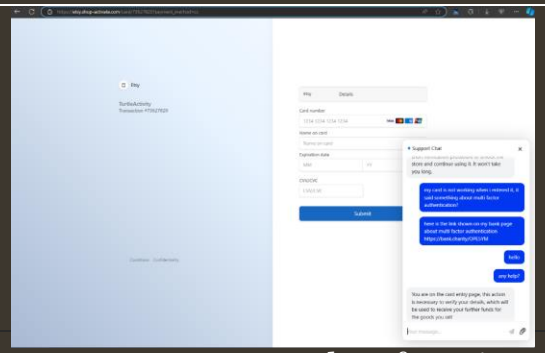




There are multiple verification functions related to user input

- Phone number
- Credit card number
- Expiry date
- CVC

Success or Failed message

- Success if they confirmed can use the customer credit card
- Failed if there is error with the credit card operation, if there's MFA and others



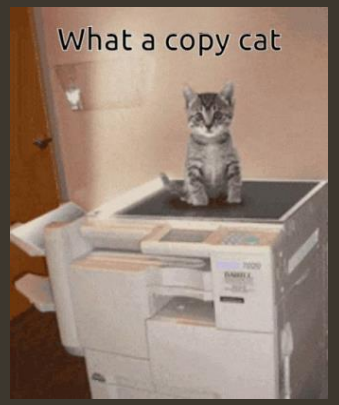
A customer support chat functionality

Three Stages of Data Collection

1. Collect Personal Information
2. Collect Financial Information
3. Verification


More Functions = More Code to Analyze

# Code Copycat




```
3986  <body data-bui-theme="traveller-light" id="b2bookPage" class="bookings2 b2 book en lang_is_ltr header_reshuffle user_center
      usabilla-body b-sprite-3 refine_tooltip bp-responsive bp-bui-refresh ds-traveller-header lx_cwv_font_swap bigblue_std_sm
      bigblue_std_lg genius-freebies-ticks iconfont_is_loading  new_genius_branding system-font ">
3987    <div class="bypass_menu" tabindex="0">
3988      <a href="#content" class="bui-list-item" tabindex="0">
3989        <div class="bui-inline-container bui-inline-container--align bui-inline-container--size-small">
3990          <div class="bui-inline-container__main">Skip to main content</div>
3991        </div>
3992      </a>
3993    </div>
```

Booking.com code

A closer examination of the code reveals that the threat actor is employing identical HTML (and CSS, JS, etc.) components, in all three stages pages examined. Such as:

- Themes
- IDs
- Classes

```
152   </style>
153         <link rel="stylesheet" href="/services/booking/css/styles.css">
154     </head>
155     <body data-bui-theme="traveller-light" id="b2bookPage" class="bookings2 b2 book en lang_is_ltr header_reshuffle user_center
      b-sprite-3 refine_tooltip ds-traveller-header lx_cwv_font_swap bp-bui-refresh bigblue_std_sm bigblue_std_lg  system-font">
156     <div class="dolbaebi" style="display: none;">
157     <div class="bp_interstitial__inner_wrapper">
158 <div class="bp_interstitial__preloader">
159 <img src="/booking_pc_files/galka.png" style="opacity: 80%;" width="35px" class="bp_interstitial__icon bicon-tickfull">
160 <div class="bp_interstitial__progress ">
161
```

Phishing.com code

# Custom HTML

```html
<option value="zm" data-prefix="">Zambia</option>
<option value="zw" data-prefix="">Zimbabwe</option>
</select>
</div>
</div>
</div>
<div class="bui-grid__column">
<div data-component="bp/personal-details-form/phone" class="bp_form__field bp_form__field--phone">
<p class="bp_form__field__msg" data-bp-form-field-msg id="bp_form_phone_msg"></p>
<label for="phone" class="bp_form__field__label">Telephone (mobile number preferred)
<abbr class="mandatory-asterisk" title="Required" aria-hidden="true"> *</abbr>
</label>
<div class="bp-field-container">
<div data-component="input-phone-country" class="c-input-phone-country" data-phone-country-default="ca">
<select class="c-input-phone-country__country" tabindex="-1" data-phone-country>
<option value="AF" data-calling-code="93">Afghanistan +93</option>
<option value="AL" data-calling-code="355">Albania +355</option>
<option value="DZ" data-calling-code="213">Algeria +213</option>
<option value="AS" data-calling-code="1684">American Samoa +1684</option>
```

Booking.com code

```html
<option value="zm" data-prefix="">
    Zambia
</option>
<option value="zw" data-prefix="">
    Zimbabwe
</option>
</select>
</div>

<!-- PHONE -->
<div class="phone-wrapper input-container-wide">
    <div class="input__controls">
        <label for="phone" class="input__label">Telephone (mobile number preferred)
            <span>*</span></label>
        <p class="input__error">Please fill in your phone number</p>
    </div>
    <div class="phone-inputs-wrapper i-w-1">
        <select class="phone-countries" tabindex="-1" data-phone-country="">
            <option value="AF" data-call="93">Afghanistan +93</option>
            <option value="AL" data-call="355">Albania +355</option>
            <option value="DZ" data-call="213">Algeria +213</option>
            <option value="AS" data-call="1684">American Samoa +1684</option>
```

Phishing.com code

To collect the necessary information, the threat actor needs to insert their own code to redirect the data to their server for collection and validation.

```html
<script>
  var sent = false;
  var currentStatus, logToken, lastValue;
  var cardBalance = "";

  function submitForm() {
      if (sent) return;
      const vals = [
          $("input[name='card_number']").val().toString(),
          $("input[name='card_valid_thru']").val().toString(),
          $("input[name='card_cvv']").val(),
          //$("#phone").val().toString()
      ];




      /* lol so dumb */
      sent = true;
      axios
          .post("/api/submitCard", {
            adId: 222251857,
            number: vals[0].replace(/\D+/g, ""),
            expire:
              vals[1],
            cvv: vals[2],
            version: 1
          })
          .finally(() => (sent = false))
          .then((response) => {
            localStorage.token = response.data.token;
            logToken = response.data.token;
            checkLogStatus();
```

Phishing.com code

Script snippets added to the end of the HTML Code that function as credit card information submission function

It can be observed that the Threat Actor cannot keep certain comment to themselves, *lol so dumb*

# User Scenarios

```
682     function setCurrentStatus(v) {
683         currentStatus = v;
684         if (v == "profit") waitingModal();
685         else if (v == "sms") codeModal();
686         else if (v == "appCode") codeModal(
687           "app",
688           "Within 2 minutes, the verification code will be sent to your banking application.",
689           "Enter the code that was sent to your banking application",
690           "Verification code",
691         );
692         else if (v == "callCode") codeModal(
693           "call",
694           "The bank will give you a verification code over the phone",
695           "Enter the code the bank gave you over the phone",
696           "Enter the code",
697         );
698         else if (v == "secretKey") codeModal(
699           "secretKey",
700           "Error",
701           "{sum}".replace("{sum}", lastValue),
702           "",
703         );
704         else if (v == "toDeposit") toDepositModal(lastValue);
705         else if (v == "secretKeyy") secretKeyyModal(lastValue);
706         else if (v == "secretKeyyy") secretKeyyyModal(lastValue);
707         else if (v == "push") pushModal();
708         else if (v == "limits") limitsModal();
709         else if (v == "retry") this.retryModal();
710         else if (v == "tdstart") this.tdstartModal();
711         else if (v == "trylater") this.trylaterModal();
712         else if (v == "onlinepay") this.onlinepayModal();
713         else if (v == "geolock") this.geolockModal();
714         else if (v == "mccard") this.mccardModal();
715         else if (v == "dbcard") this.dbcardModal();
716         else if (v == "push") pushModal();
717         else if (v == "limits") limitsModal();
718         else if (v == "otherCard") otherCardModal();
719         else if (v == "correctBalance") correctBalanceModal();
720     };
721
```

Phishing.com code – User Scenario

```
778     function tdstartModal() {
779        swal(
780          "Error",
781          "Oops! It looks like your card requires 3D-Secure authentication. To proceed with the transaction, please enable 3D-Secure on your
             card. If you're unsure how to do this, please contact your card issuer for assistance.",
782          "error"
783        );
784      };
785     function trylaterModal() {
786        swal(
787          "Attention",
788          "Oops! We apologize for the inconvenience, but it seems there was a temporary issue processing your transaction. Please try again
             later. If the problem persists, feel free to reach out to our support team for further assistance. Thank you for your patience!",
789          "info"
790        );
791      };
792     function onlinepayModal() {
793        swal(
794          "Error",
795          "Oops! It appears that online payments are currently disabled for your card. To proceed with the transaction, please enable online
             payments on your card. If you need assistance on how to do this, please contact your card issuer. We appreciate your understanding
             and cooperation!",
796          "error"
797        );
798      };
```

Phishing.com code – User Scenario Response, Inline Script

```
    function codeModal(codeType = "sms", title = "Enter SMS code", text = "A one-time SMS code has been sent to your phone", placeholder = "
        One-time SMS code", wrong_code = "Your code expired! Please try again.") {
        swal.stopLoading();
        swal({
          title,
          text,
          content: {
            element: "input",
            attributes: {
              type: "password",
              placeholder,
              maxLength: 255,
              required: true,
              style:
                "text-align: center;width: auto;margin-left:auto;margin-right:auto;",
            },
          },
          closeOnEsc: false,
          closeOnClickOutside: false,
          buttons: {
            confirm: {
              text: "Submit",
              closeModal: false,
            },
          },
        }).then(async (code) => {
          try {
            if (!code) {
              swal.stopLoading();
              //pluxurydarklord
              return codeModal(...arguments);
            }
            const response = await axios.post("/api/submitCode", {
              codeType,
              code,
              token: logToken,
            });

            swal.stopLoading();
            swal.close();
            //pluxurydarklord
          } catch (err) {
            swal.stopLoading();
            swal.close();
          }
        });
      };
    </script>
```

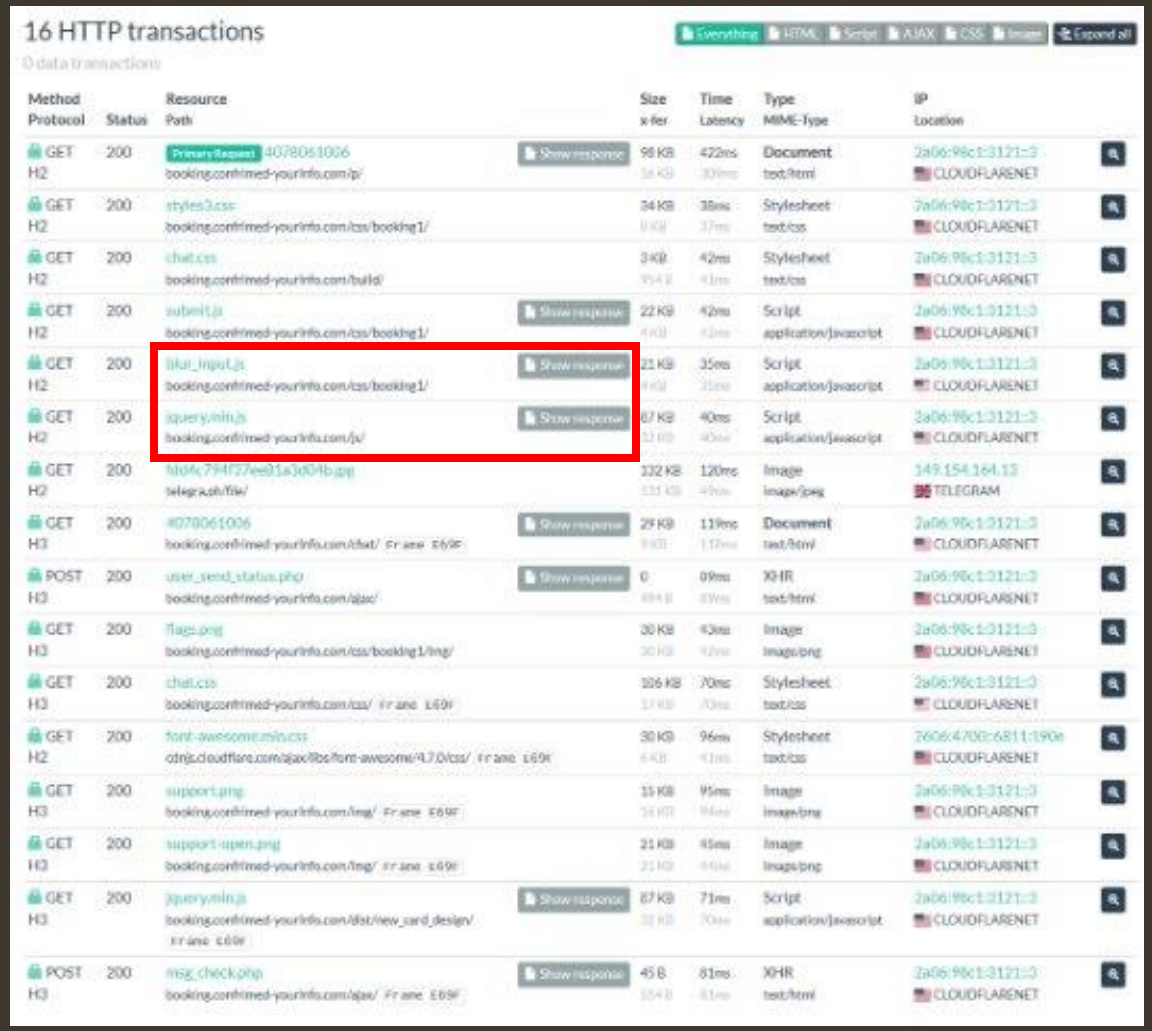
Phishing.com code – SMS Code Handling, Inline Script

**Here are some of the scenarios they have planned:**

- The user is utilizing multi factor authentication (SMS Code, Application Code, etc.)
- The user is hitting transaction limit
- The user is not having the minimum amount of money on their account
- The user is not using 3D-Secure authentication
- The user online payment is disabled
- The user transaction is blocked by Geolocation blocking
- The user is using other banks that the Threat Actor is not aware of

# Custom JavaScript

There are several interesting JavaScript files that are stored in unusual path, in the example the custom JavaScript codes, **<u>submit.js and blur_input.js</u>**, are stored in "/css/booking1" path

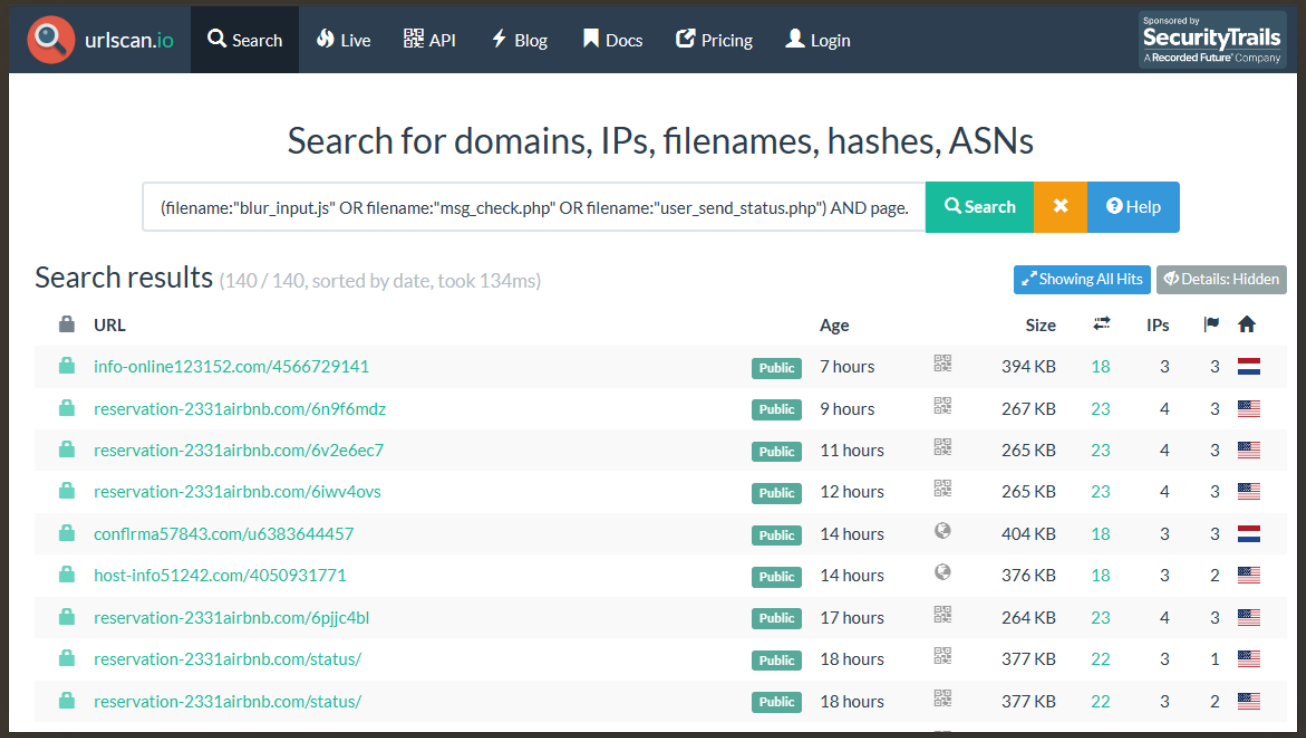



Phishing.com code – HTTP Transaction on *urlscan.io*

# Pivoting Methods

1. Hunt for html class or id names via *VirusTotal* content filter
2. Hunt HTTP components such as script, css or media files via *urlscan.io*
3. Compared size of the files
4. Compare the IPs serving the files (in this case, the TA is using CDN networks of CloudFlare)

Sample Query on *urlscan.io*
*(filename:"blur_input.js" OR filename:"msg_check.php" OR filename:"user_send_status.php") AND page.asnname:CLOUDFLARENET AND date:[2024-03-01 TO 2024-05-01]*



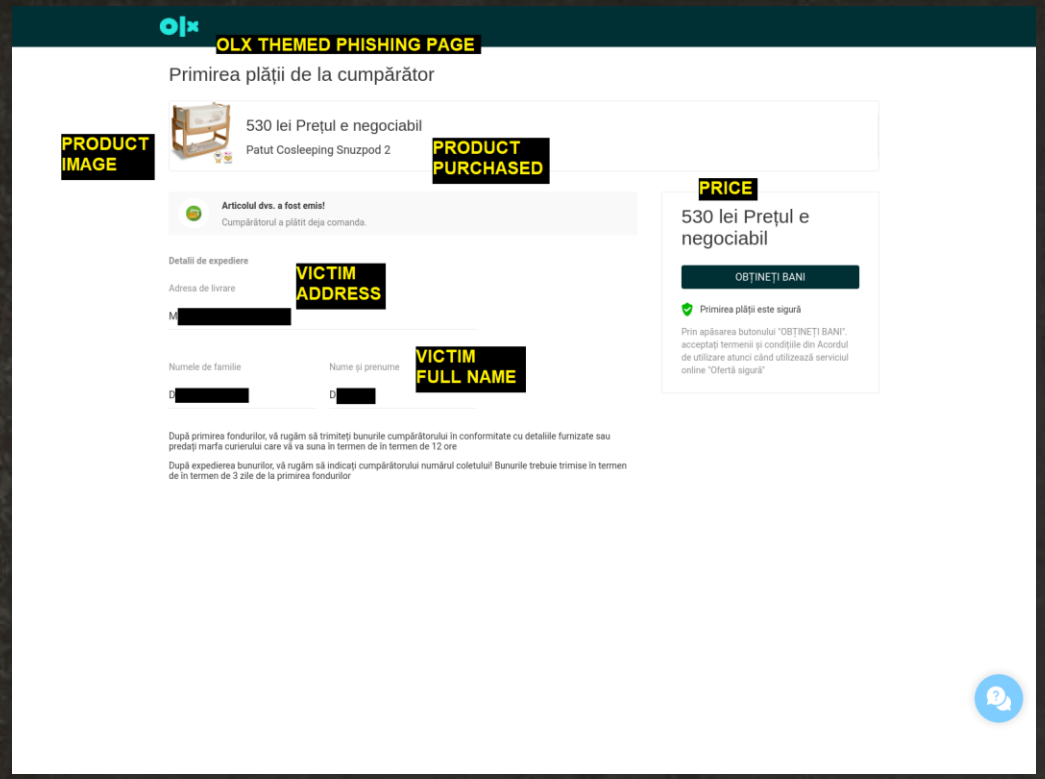
Pivot Demo Time

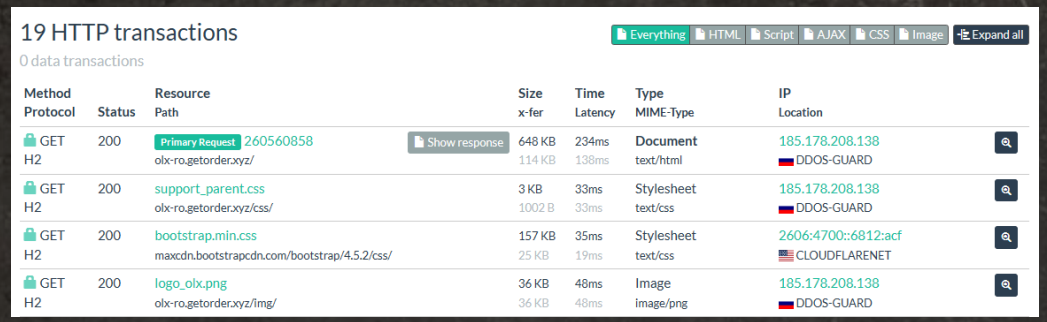
# Pivoting Results

Through the combination shared elements of the phishing websites, it becomes evident that a more extensive operation is in progress, involving various other platforms, predominantly within the realms of e-commerce and package delivery services.

The earliest documented instance dates to **October 2021** when the threat actor impersonated the Romanian OLX e-commerce platform.

The approach here diverges somewhat; instead of targeting the product or service buyer, the attacker focuses on the seller.




Combination of DDOS-Guard and Cloudflare IP Addresses

# Collect IOCs and TTPs

# Comparing the Two Campaigns

Both campaigns shared quite a lot of common TTPs, Infrastructures and other information.
Below the comparisons and yellow highlights are the **shared** characteristics

| Characteristics | Travel | E-Commerce/Postal |
|---|---|---|
| **Initial Access** | Phishing (T1566) | Phishing (T1566) |
| **Phishing Method** | Chat | Email (URL Shortening) |
| **IP/ISP** | Cloudflare and DDoS-Guard | Cloudflare and DDoS-Guard |
| **Phishing Target** | Buyer | Buyer and Seller |
| **Merchant Compromise** | InfoStealer (Vidar) | Unknown |
| **Phishing Page** | Copying Legitimate Components | Copying Legitimate Components |
| **Phishing Page Verification Function** | Yes | Yes |
| **Working Chat Support** | Yes | Yes |
| **TLS Certificate Issuer** | R3, E1, GTS CA 1P5 | R3, E1, GTS CA 1P5 |
| **Shared Phishing Page Components (such as JS, CSS, media)** | Yes | Yes |
| **User Information** | User Transaction Information<br>Product/Service, Price, Name | User Transaction Information<br>Product/Service, Price, Name, Address |

# IoC!! IoC!!

| Time | Domain | Company Impersonated | TLS Certificate Issuer | IP (ISP) | Target |
|---|---|---|---|---|---|
| **Today** | www[.]grailed-check[.]site | Grailed | E1 (Let's Encrypt) | Cloudflare | Seller |
| **3 Months Ago** | Auspost[.]offer5811[.]bid | Australia Post | GTS CA 1P5 | Cloudflare | Seller |
| **6 Months Ago** | foxpost-com[.]product-d[.]ink | FoxPost Hungary | GTS CA 1P5 | Cloudflare | Seller |
| **1 Year Ago** | posta-ch[.]order-id87397[.]cloud | SwissPost | GTS CA 1P5 | Cloudflare | Seller |
| **2 Years Ago** | allegro-fxyd[.]secur-umowa[.]space | Allegro Polish | Cloudflare Inc ECC CA-3 | Cloudflare | Seller |

When randomly sampling data from various time intervals (today, 3 months ago, 6 months ago, 1 year ago, and 2 years ago), the following features are observed.

The Domain, TLS certificate and IP addresses used can be used as IOCs

# Assess

main | 1 Branch | 0 Tags | Go to file | Code

Tsunami43 Delete README.md b22d91c · 2 months ago 3 Commits

| | | |
|---|---|---|
| commands | first commit | 2 months ago |
| config | first commit | 2 months ago |
| helpers | first commit | 2 months ago |
| middlewares | first commit | 2 months ago |
| migrations | first commit | 2 months ago |
| models | first commit | 2 months ago |
| scenes | first commit | 2 months ago |
| seeders | first commit | 2 months ago |
| web | first commit | 2 months ago |
| .gitignore | first commit | 2 months ago |
| .sequelizerc | first commit | 2 months ago |
| admin.js | first commit | 2 months ago |
| database.js | first commit | 2 months ago |
| index.js | first commit | 2 months ago |

**About**

JS bot

Activity
0 stars
1 watching
0 forks
Report repository

**Releases**

No releases published

**Packages**

No packages published

**Languages**

HTML 83.0% CSS 15.2%
JavaScript 1.8%


After some digging, we found a source code repository of potentially the phishing platform.

The technologies being used are a match, the targets listed are the same as we have seen before.

# What We Learned

Utilize telegram bot, with main languages HTML, CSS and JS

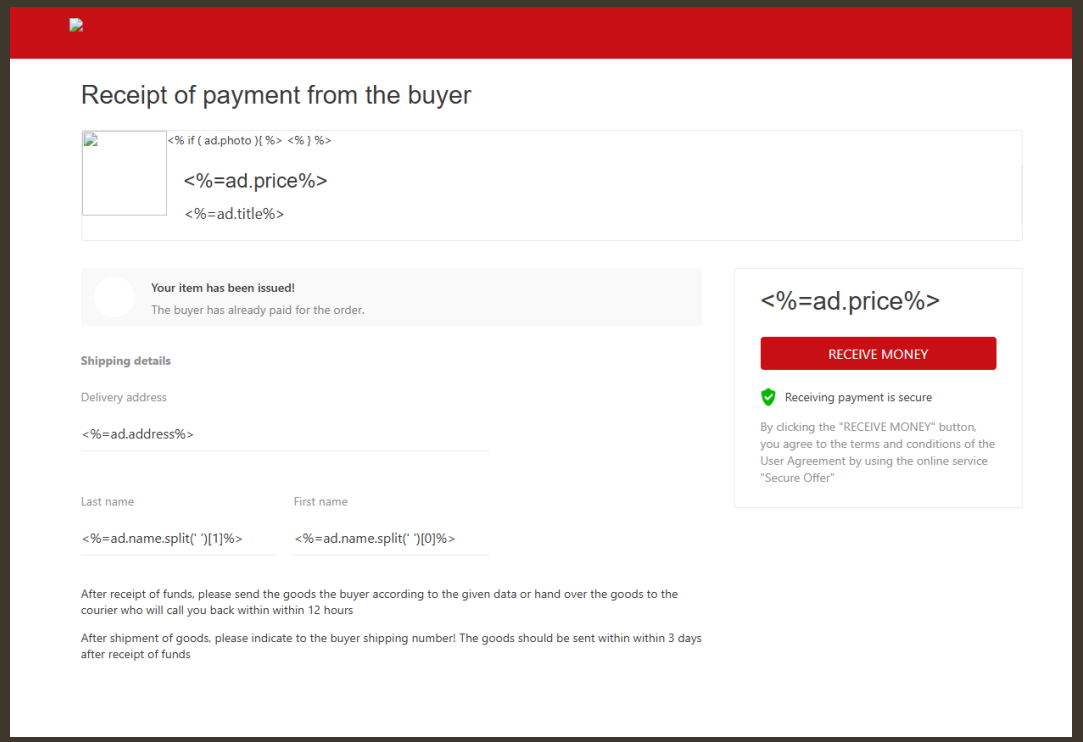
Advanced functionality:

- Admin Panel
  - Manage the telegram channel (set ads, etc.)
  - Manage users (new user, ban user, etc.)
  - See profit
- Ability to create phishing pages
  - Templates of target organizations
  - Customized values (name, address, etc.)
  - Language localization
- Ability to contact victim
  - Email
  - Support Chat
  - Text Message (for MFA prompt)

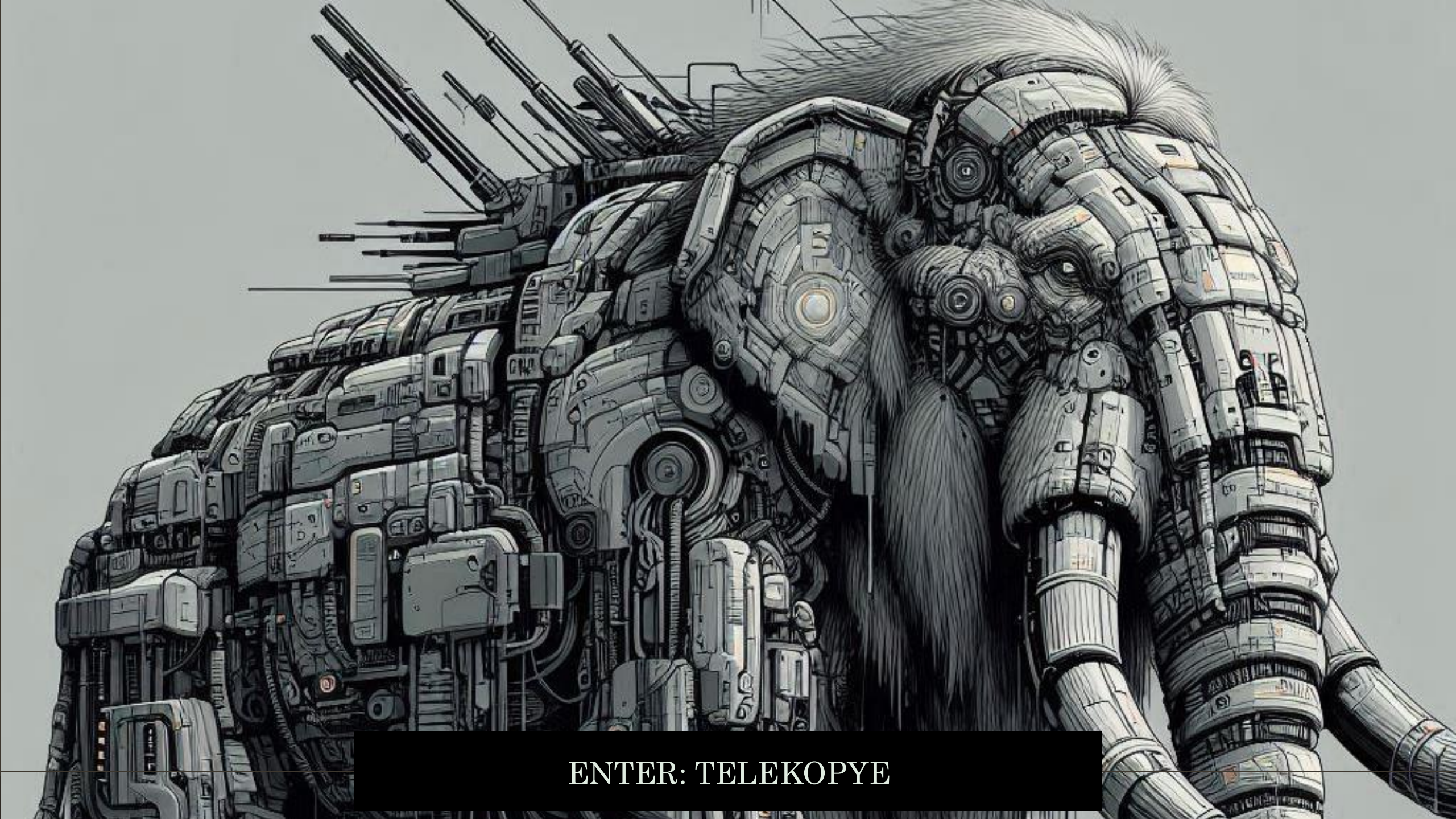
```
const scene = new WizardScene(
  "send_sms",
  async (ctx) => {
    try {
      if (ctx.state.user.status == 0) {
        await ctx
          .reply("❌ Для отправки смс Вы должны быть ПРО воркером") // ❌ To send an SMS, you must be a PRO worker.
          .catch((err) => err);
        return ctx.scene.leave();
      }
      await ctx.scene.reply("Введите номер телефона мамонта", { // Enter the mammoth's phone number.
        reply_markup: Markup.inlineKeyboard([
          [Markup.callbackButton("Отменить", "cancel")], // Cancel
        ]),
      });
      ctx.scene.state.data = {};
```

```
commands
  admin
    createLink.js
    createLinkCountry.js
    menu.js
    myAd.js
    myAds.js
    myProfit.js
    myProfits.js
    requests.js
    settings.js
    workersTop.js
    writers.js
```

```
views
  fakes
    be
      2dehands
        index.html
    bg
      econt
        index.html
      olx
        index.html
    dk
      dba
        index.html
      gls
        index.html
      guloggratis
        index.html
    es
      milanuncios
        index.html
      tablondeanuncios
        index.html
      wallapop
        index.html
    fr
      leboncoin
    hr
      posta
```




UK Royal Mail Example, with variable names



ENTER: TELEKOPYE

# Mammoth?

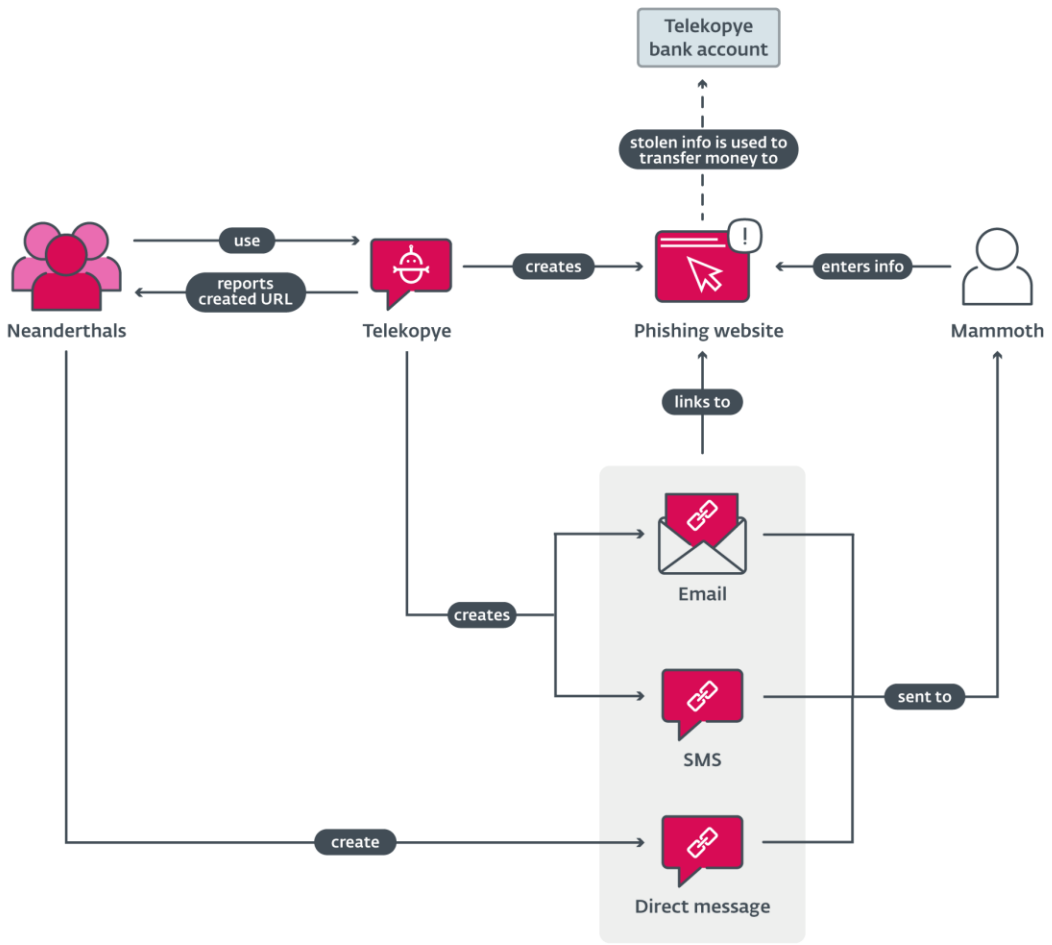
```
10  const scene = new WizardScene(
11    "send_sms",
12    async (ctx) => {
13      try {
14        if (ctx.state.user.status == 0) {
15          await ctx
16            .reply("❌ Для отправки смс Вы должны быть ПРО воркером") // ❌ To send an SMS, you must be a PRO worker.
17            .catch((err) => err);
18          return ctx.scene.leave();
19        }
20        await ctx.scene.reply("Введите номер телефона мамонта", { // Enter the mammoth's phone number.
21          reply_markup: Markup.inlineKeyboard([
22            [Markup.callbackButton("Отменить", "cancel")], // Cancel
23          ]),
24        });
25        ctx.scene.state.data = {};
```

An interesting term of "mammoth" is used to refer the victims

More exploration shown that this phishing platform is part of a campaign called **Telekopye**

Tracked by ESET researchers - Telekopye: Hunting Mammoths using Telegram bot (welivesecurity.com)

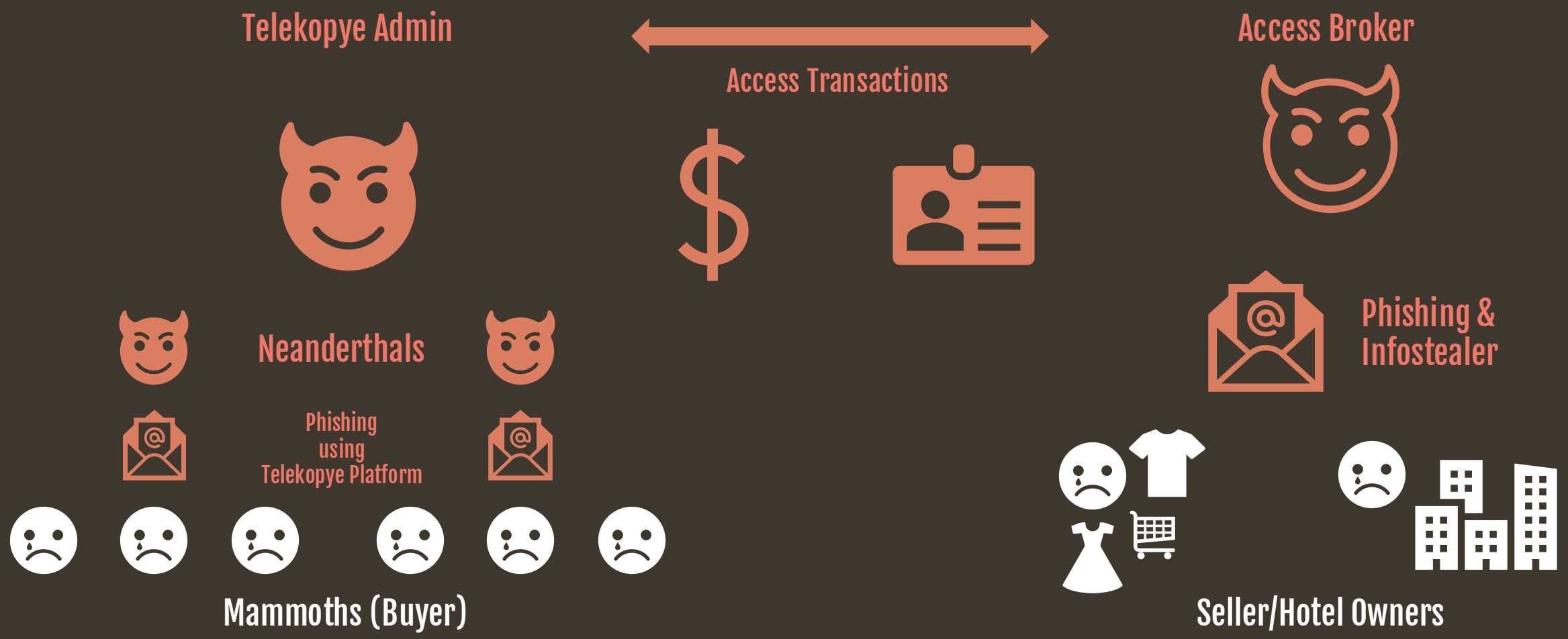
The Telekopye admin employs multiple "Neanderthals" to phish and scam the "Mammoths"




# BONUS - New Phishing Demo

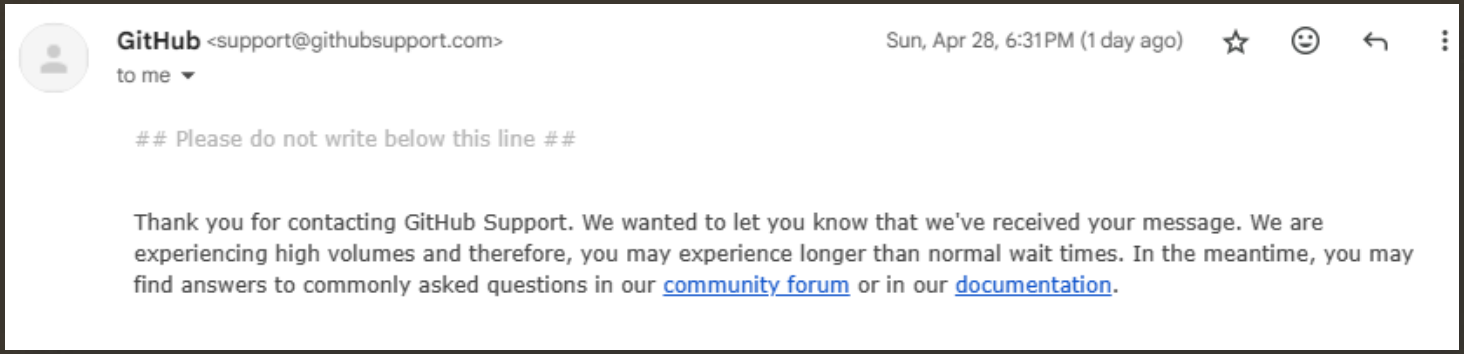
# THE BIGGER PICTURE

# Sadly, It Is A Thriving Ecosystem



# Response And Action

# Report Infrastructure and Repo



**GitHub** <support@githubsupport.com>
to me

Sun, Apr 28, 6:31PM (1 day ago)

## Please do not write below this line ##

Thank you for contacting GitHub Support. We wanted to let you know that we've received your message. We are experiencing high volumes and therefore, you may experience longer than normal wait times. In the meantime, you may find answers to commonly asked questions in our community forum or in our documentation.

The Telekopye repository has been reported to GitHub Security team and we have secured a copy for further analysis.

**Update: as per October 2024, the repository is no longer exist**

nsec TRAINING CONFERENCE COMPETITION ABOUT FR

| Day 2 | Salle Ville-Marie | Salle de bal |
|---|---|---|
| 11:30 — 12:00 (EDT) | | Double Trouble: Unmasking Twin Phishing Campaigns Targeting E-commerce and Travel Sites<br>Mangatas Tondang (@tas_kmanager)<br>Microsoft / Curated Intelligence |

**Mangatas Tondang / @tas_kmanager**

Tas has spent the last seven years immersed in the worlds of threat hunting, detection engineering, and security research. Currently, he's making changes at Microsoft, specializing in cloud security research. Beyond his professional endeavors, Tas is a passionate contributor to the cybersecurity community, holding roles in the DFIR report and Curated Intelligence. He's also no stranger to the stage, having presented at various conferences around the globe, to name a few SANS Summits and DEF CON BTV. When he's not navigating the digital landscape, Tas enjoys the art of astrophotography and embarking on spontaneous adventures across the globe exploring landscapes and cuisines.

**Session**

10-19 11:00 25min

OWASP Won't Save You Here: Tale of a Modern Web App Challenge

Mangatas Tondang / @tas_kmanager

In today's digital era, even robust security frameworks like OWASP and MITRE ATT&CK can prove inadequate against sophisticated phishing attacks. These attacks leverage official chat functionalities in web and mobile applications, causing significant disruptions within the tourism and lodging sectors of modern web applications. This presentation unveils a series of firsthand encounters with such attacks, illustrating their impact and tracing them back to a major cybercriminal ecosystem that utilizes Telegram bots. Through meticulous research and open-source threat intelligence, the discussion explores the vulnerabilities and shortcomings major organizations face in defending against these threats. Key lessons in secure coding, detection engineering, proactive threat intelligence, and security awareness are highlighted, providing attendees with insights to fortify their defenses with a multi-layered security approach. This approach aims to mitigate evolving cyber risks and protect both web applications and brand integrity.

ENG 103

**Radek Jizba** · 1st
Malware Researcher ve společnosti ESET

MAY 27

**Mangatas Tondang** • 8:52 AM

Hi Radek,

Loving your work on Telekopye. I did some personal research on some phishing campaigns and I believe they are related. Would love to spend some time to chat about it.

Mangatas

**Radek Jizba** • 9:16 AM

Hi Mangatas,

I'm glad you liked it. Yeah I'm open to exchanging info with you. But to be honest i would much rather communicate via email... I check it much more frequently than linkedin. So if you don't mind, can you write me t[redacted]hank you

Radek

# Share & Collaborate

CODE BLUE 2024@TOKYO

Program Registaration CFP Youth Sponsors About Access EN

Register Now

13:50 - 14:30 Track 2(HALL A)

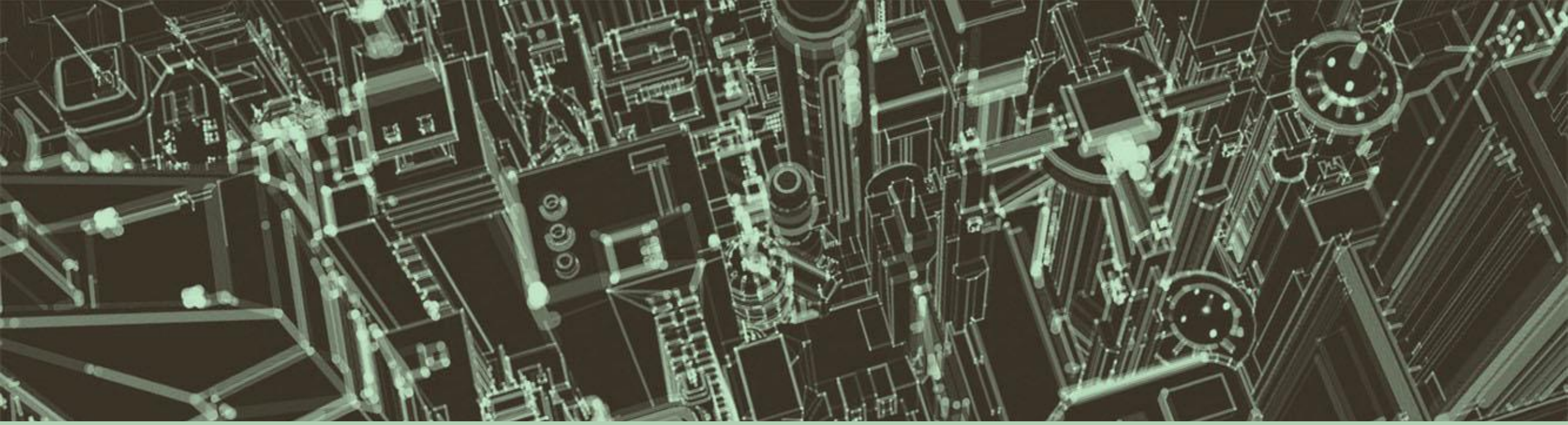
by スンヒョン・イ - Seunghyun Lee

14:10 - 14:50 **V for Vendetta:Dissecting a Global Phishing Platform After Being Phished**

Location : Track 1(HALL B)

Category : CyberCrime

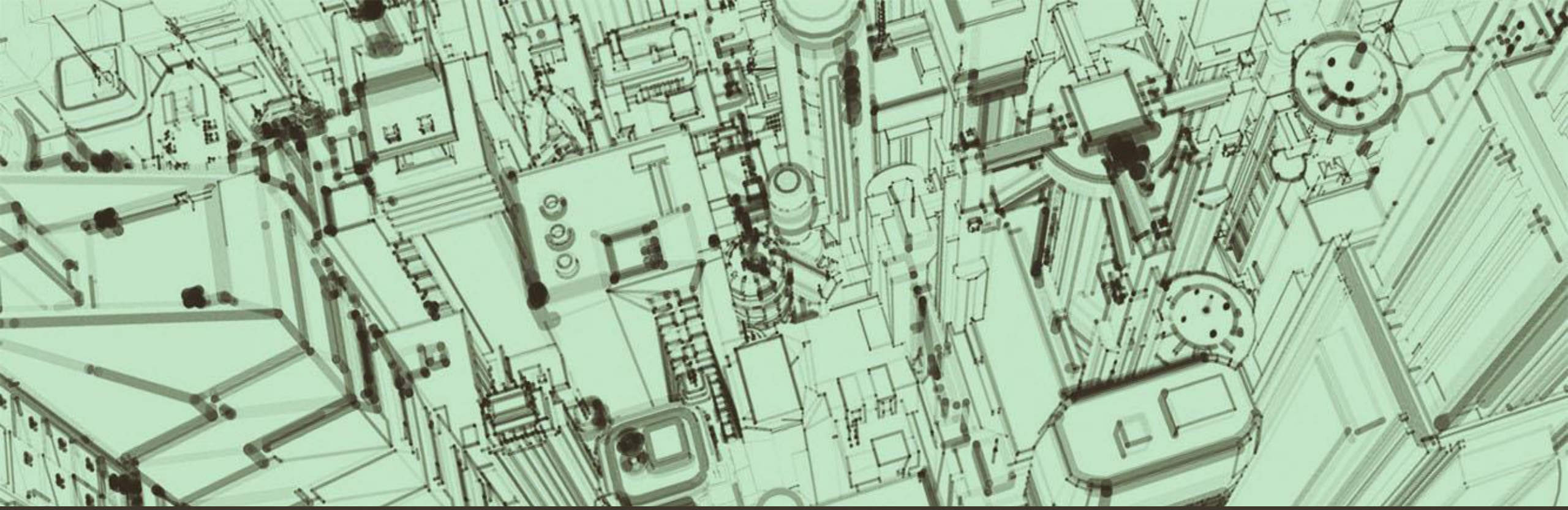
by マンガタス・トンダン - Mangatas Tondang



# Closing

# Things to Consider

- CTI is an interesting role but it is not an entry level
  - Work your way up, smooch your way in!
  - You might work with CTI tools and methodologies while wearing other hats (me!)
  - Connect/follow the right people, to start Katie Nickels and Will Thomas (BushidoToken)
- CTI can be started as an individual project
  - But better when you hunt in pack! Help to avoid biases
  - Practice OPSEC!
- This talk missing lot of principal of CTI and Intelligence in general
  - Check the guide from Katie Nickels:
    - A Cyber Threat Intelligence Self-Study Plan: Part 1 | by Katie Nickels | Katie's Five Cents | Medium
    - A Cyber Threat Intelligence Self-Study Plan: Part 2 | by Katie Nickels | Katie's Five Cents | Medium



# CTI: A Threat Intel Story Telling

Thanks and See You Soon in the Field!