1. Secure Communication Techniques Research

a) Notes on: Zero Trust Networks

Source: Zero-Trust Networks Lecture

In 2001, SANS made a list of the top 10 security mistakes made by individuals, which are ranked as follows:

- Poor password management
- Leaving your computer on and unattended
- Opening e-mail attachments from strangers
- Not installing anti-virus software
- Laptops on the loose
- Blabber mounts (file access open to the world)
- Plug and Play without protection
- Not reporting security violations
- Always behind the times (OS, application patches)
- Keeping an eye out inside the organization

In the past decade, none of these mistakes have really been fixed. When a hacker attempts an attack, their "Attack Goals" can be one or more of the following: Either their goal is to steal or disclose sensitive data, attack other sites using the hacked assets from your system, or destroy company data through deletion or ransomware.

In order to defend properly, it's important to think about what should really be defended. When we think of network security, there is often a mindset of: "It's a compromise for something to stay inside your network, but a breach only if it leaves". The reality is that you can't protect the network, because the network is hostile - it's impossible to control it.

The most important thing that should be defended is sensitive data - and having knowledge of where that data is located, constitutes the first step in protecting it

The motto behind Zero-Trust Networks is the concept of treating your network as hostile. The following quote by Chris TownShend is a good view of how our data fits in to all of this:

"As we move our data outside of the firewall, we have to adopt a Zero-Trust model. We are shifting our security enforcement out of the data itself, and you have to have a security policy that follows that user no matter where that user is or what device they are using to access the data."

He is essentially saying that our data becomes the border of our network.

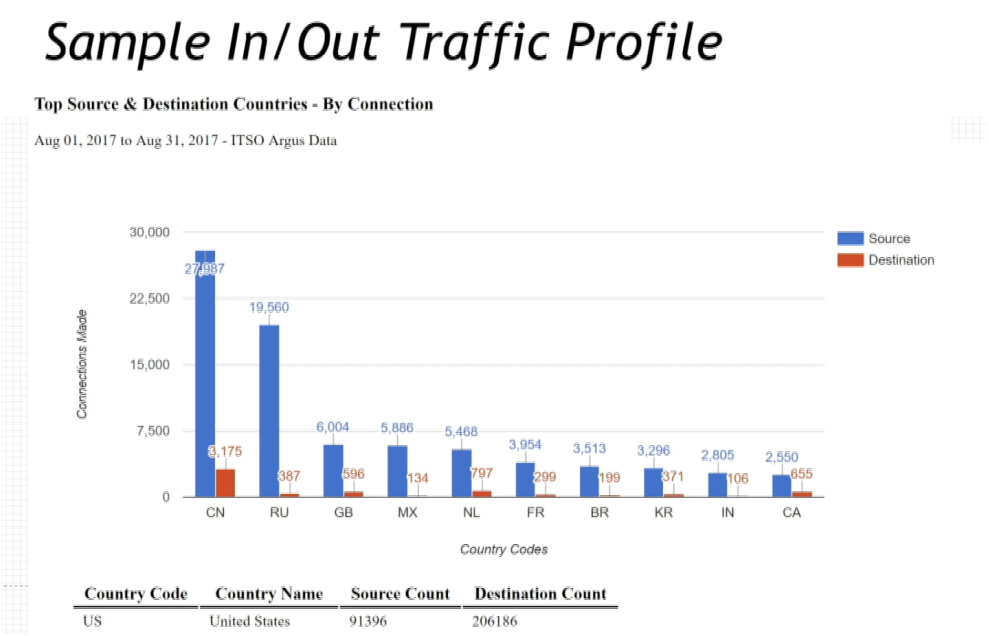

Figure 1: Above is a graph that examines the traffic coming in and out by country

A good way of understanding the network is to think of it like a museum. We have free flowing access through the network, but also must have additional barriers around high risk assets (in the case of a network - sensitive data). In the same way that we assume there are hostile people inside a museum, we must always assume that the network is hostile, and treat it as such.

As a recap, let's compile the characteristics of Zero-Trust Networks:

1. The network is hostile (or is assumed to be so)
2. Assume that there are already hostiles inside of the network
3. Segmentation is not enough to decide trust in the network
4. Network flow - user and devices must be authorized and authenticated
5. Policies are dynamic, and calculated from as many data sources as possible
6. The user's data (identity) is the border of the network - the device is not the border anymore
7. New disruptors of traditional security architectures include: containers, serverless and cloud computing
8. Mobile: users, apps, and storage

In actuality, the Zero-Trust Network theory is easier said than done. Though we are surrounded by components of Zero-Trust, they may be hard to recognize (logging, firewall detection, network monitoring tools)

i. Components of a Zero-Trust Network

The Control Plane: Grants access to the network resources, processes requests from (Data Plane) devices to do so.

User/device authorization and authentication are completed here, as well as stronger authentication for high risk resources.

The Data Plane: Components contained in the Data Plane include: applications, firewalls, proxies, routers processing network traffic. The Data Plane works to handle higher traffic rates.

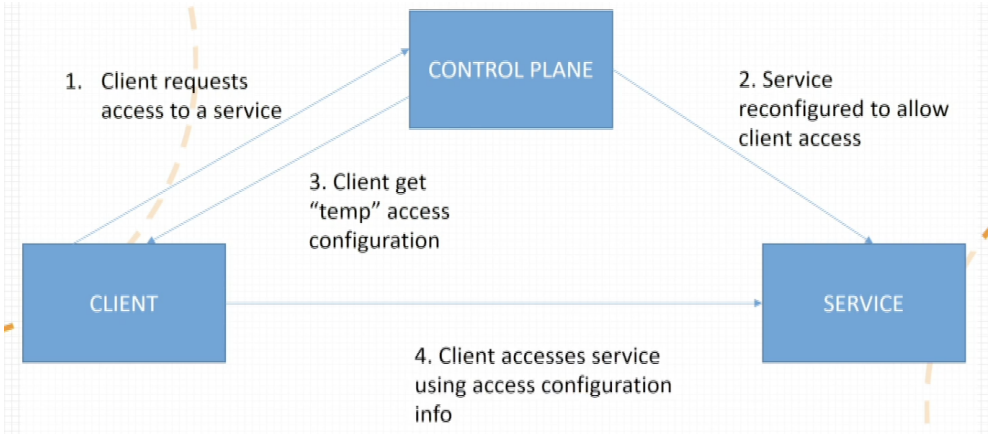

Figure 2: Above is a graph that represents how client-control plane interactions occur

One final note for ZTNs: What goes into the network is not as important as what comes out.

b) Notes on: Network Security 101 - Full Workshop

Source: Network Security 101 Workshop

i. Man-in-the-middle (MitM)

Man-in-the-middle attacks occur when the hacker inserts himself between the user and the web server, often by diverting the user's traffic through him/herself before it reaches the web server.

In order to understand how situations such as MitM attacks occur, let's look at a few scenarios.

ARP - Address Resolution Protocol

This protocol is used to locate a physical address for a corresponding IP address. It works by sending out a message, to which only one system (the owner of the IP address) responds.

Since this protocol is initially based on trust, a hacker can proxy the actual gateway and lie - sending out a response in place of the real IP address, asserting their system to be the correct address. This is one way for a man-in-the-middle to come about. In this scenario, all traffic is going through this hacker. One constraint to this method, however, is that the hacker must be connected to the same local network in order to accomplish this.
Note: Ettercap is just one of many tools that are able to accomplish this.

This method is called ARP Cache Poisoning.

What is TOR?

TOR is an anonymizing proxy that allows a user to communicate with other websites or users without being easily traced. This is achieved through randomized interior jumps between three nodes that user A will connect to in order to communicate with user B. The last node that connects to user B is called the exit node, which has contact to services that exist outside of the TOR cloud.

The final method for man-in-the-middle interception: WiFi Pineapple
Karma Attack: When you turn on any wireless device, it (the network stack) immediately checks every place it's previously been and searches for available WiFi networks. What WiFi Pineapple does is it responds to your device's search and acts as if it is the network the device has been looking for. As long as you have an encrypted access point (WPA2), this attack won't succeed. However, it is good to remember that not all public access points are encrypted, leaving you open to a MitM attack, because your device will still remember these open networks and search for them when you're in the area.

From The Hacker's POV

MitM attacks allow the hacker to:

- Steal and/or store login information
- Sniff (passively take) information
- Inject false information/Manipulate connection

Being a MitM gives one complete control over a user's online experience. And from the user's perspective, figuring out whether you are being hacked or not is very difficult.

Cryptography

By definition, cryptography is the science of encoding and decoding information.
There are two different types of cryptography: Symmetric and Public Key

Symmetric Crypto:
The encoded information is referred to as cyphertext - which can be encoded by using an arbitrary, secret key. So, if user A wants to send an encrypted message M to user B, they can run an encryption algorithm using their message M and key K

$$E(K, M) = C \tag{1}$$

With C = cyphertext

Now, when user A sends C to user B, B will need to decode the message. we can denote the inverse to be the original message M:

$$E^{-1}(K, C) = M \tag{2}$$

Note: A and B both have the same secret key K

Something to notice is that his method seems to only be efficient if you are both A and B. One concern is the following: Is there any convenient way for key K to be shared without it falling into the hands of the MitM?

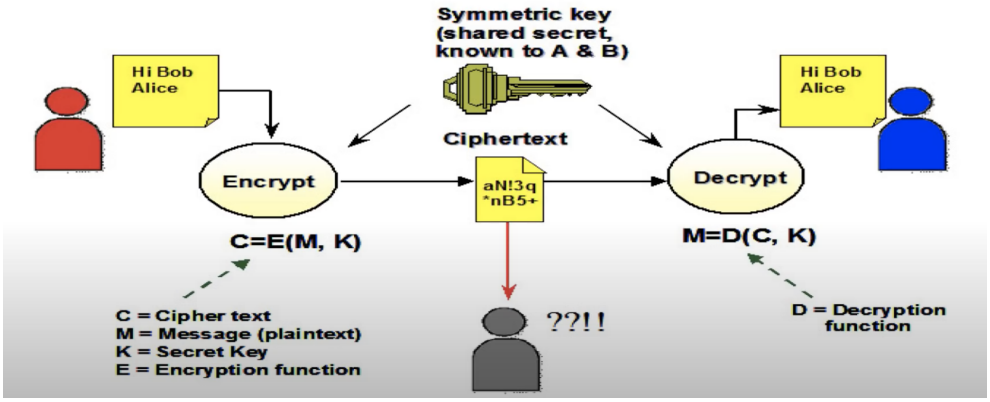

Figure 3: Above is a graph to represent the process of encoding and decoding with Symmetric Crypto

Note: The encoding algorithm E and decoding algorithm E^{-1} are publicly known - but in order for a MitM to be able to successfully decode this message, they would have to guess the right key out of 2^{128} possible keys.

Message Authentication Code - Authenticates that user A is really the one who sent this message to user B, essentially like a signature.

Public Key Crypto:

This system is not symmetric, and therefore doesn't assume user A and user B have a shared key K. In Public Key Crypto, you have a public key PK_B, and a secret key SK_B, for a single user B. As the names indicate, there is no danger in sharing the public key, but the private key should be protected and kept secret. There is a mathematical relationship between these two keys, but we cannot derive one from the other and there is no way to obtain SK by only knowing PK.

You could theoretically break RSA by using brute force if you were able to factor large composites, which would require an algorithm of exponential time, and there is no amount of time where one would be able

to do this. Therefore, it is not a realistic threat to this MitM method. [Quantum computers are able to break RSA in polynomial time, but their one drawback is that they unfortunately do not exist.]

In the case of Public Key Crypto, if user A wants to send a message M to user B: A would compute (3) and then send C to user B, and B would then decode with (4) for the original message.

$$E(PK_B, M) = C \tag{3}$$

$$D(SK_B, C) = M \tag{4}$$

For Public Key Crypto, PK encrypts and SK decrypts. The reason RSA is so difficult to break is because even user A can't undo what they just did - A cannot decode their message once they encode it - at that point, only user B can see M.

Interesting note: It is possible that there exist backdoors within crypto algorithms, most likely put in place by the NSA, in a way that they are not noticeable to the public, but reveal the keys/plain-text/etc. under certain circumstances.

Furthermore, Public Key Crypto is significantly slower than Symmetric Key Crypto, and RSA is the most famous PK cryptosystem (it is based on the difficulty of factoring).

Public Key Authentication - Similar to Message Authentication Code, RSA is used for authenticating Public Key. In this case, we have a private signing key and a public verification key. The message itself will not have a signature, but the hash of the message will.

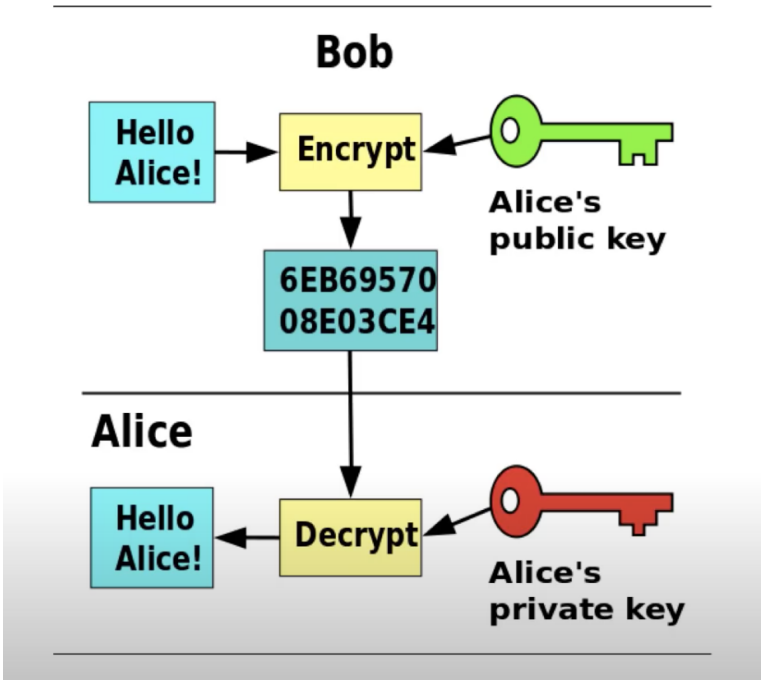

Figure 4: Above is a representation of how Public Key Crypto encodes and decodes a message

SSLStrip

Realistically, nobody actually types "https" into the browser when they want to go to Google. Instead, most people type in "google" or "google.com" and let port 80 redirect them to port 443, which is Google using https. This is called a 302 redirect.

SSLStrip keeps your browser from being redirected to port 443. Instead, the hacker keeps the user on port 80 with an (outwardly) almost identical, unencrypted version of the site. SSLStrip is one MitM attack method that is directed toward SSL, and is one of the only attacks that still succeed, since Certified Authentication (CA) protects SSL from these attacks. [HSTS (Http Strict Transport Security) is the defensive response to SSLStrip]

c) Notes on: Network Security Vulnerabilities

Source: Common Types Of Network Security Vulnerabilities In 2021

Network vulnerabilities are weaknesses within the software, hardware, or organizational processes, that when exploited, results in a breach of security.

Nonphysical vulnerabilities, such as outdated OS without the latest patches, involve software/data. Physical vulnerabilities involve physically protecting an asset (such as sensitive information). This can include securing entry with a turnstile or locking a server in a rack closet.

Different types of vulnerabilities include:

1. Malware - malicious software, such as viruses, Trojans, or worms
2. Social Engineering Attacks - psychologically manipulating others into giving up sensitive data (this may be a username/password)
3. Outdated or Unpatched Software - exposing certain applications and possibly the whole network
4. Misconfigured OS/Firewalls

Let's look at them each individually:

Malware:
Malicious software unknowingly downloaded or purchased by the user.

Often times, users don't even know that their system is infected with malware. If the system has malware installed, it will: run significantly slower, randomly reboot, or start unknown processes/send emails without the user. Hackers often target users through email, sending links to websites and embedding attachments.

Types of malware include: Viruses, Keyloggers, Worms, Trojans, Ransomware, Logic Bombs, Bots/Botnets, Adware and Spyware, Rootkits.

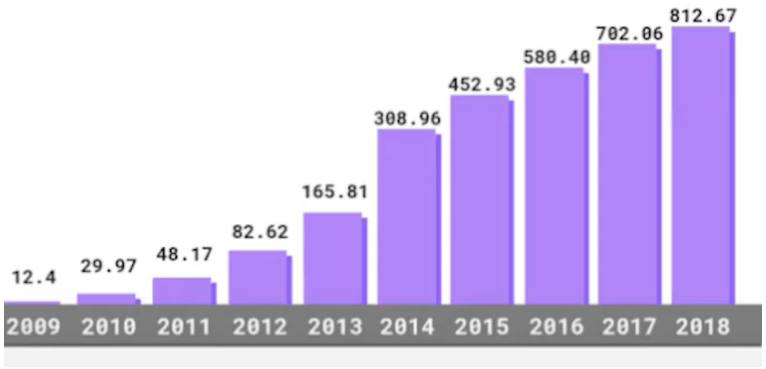

Figure 5: In 2018, malware exploitation hit a new record high with 812 million infected devices

- Viruses - the most common of the Malware attacks, they must be clicked or copied to a host in order to compromise the system. They can spread to multiple systems by email, messaging, downloads, USB, and even network connections.
- Keyloggers - also known as keyboard capturing, will log the user's keystrokes and send them to the hacker. This is commonly used to steal login information.
- Worms - similar to viruses, they can spread over multiple systems. However, worms do not need a host in order to replicate.
- Trojans - malware pretending to be credible software. They lay dormant on your system until activated, and can then be used to spy, steal sensitive information, or gain backdoor entry to the system.
- Ransomware - locks the user out of the system until they pay a ransom. CryptoMalware, a type of Ransomware, encrypts the user's files and usually requires payment in Bitcoin.
- Logic Bombs - malware that is dormant until triggered. The trigger can be a specific predefined time or date, or it can activate when some other condition is satisfied. Logic Bombs can be very damaging, and in some cases they have even rendered hard drives unreadable.
- Bots/Botnets - Botnet stands for robot network, referring to a group of bots - any computer system attached to a compromised network.
- Adware and Spyware - Adware serves advertisements in a browser, while Spyware looks to gain access to the system. Though annoying, Adware is harmless, but Spyware collects the user's information.
- Rootkits - backdoor programs that give the hacker control over the system without the user's knowledge. The hacker is able to remotely log files, spy, execute files, and change system configurations.

Social Engineering Attacks:

Internal users are the greatest risk to an organization. Those who are uneducated may be easily manipulated through social engineering, and can allow unwanted access into the system by accidentally downloading attachments, clicking links, etc.

Common types of SE attacks include: Phishing Emails, Spear Phishing, Whaling, Vishing, Smishing, Spam, Pharming, Tailgating, Shoulder Surfing, Dumpster Diving

- Phishing Emails - a malicious email sent by a hacker that looks to be legitimate. These emails aim to trick the user into giving a username/password, downloading/opening an application, or transferring money. This attack is reliant on a false sense of trust.
- Spear Phishing - similar to phishing, but uses personal information to further bait the user into clicking a link.
- Whaling - targets a user of higher status with more high-risk data, such as a manager or business executive. These fake emails/messages are hard to tell apart from legitimate emails/messages.
- Vishing - voice phishing, an attack taking place over the phone. Vishing is one of the fastest growing SE attack methods, as there was a 57% increase in these attacks between 2017 and 2018.
- Smishing - an attack using SMS text messaging to trick users into providing sensitive data. These messages might also include malicious links.
- Spam - the sending of mass emails to many users. The emails can either be harmless or they can be scams.
- Pharming - directs a user's traffic (website traffic) to a false site. Code is installed on the system to modify the destination URL to that of a hacker's.
- Tailgating - a form of attack where someone gains physical entry to a facility by following the user inside - it can be as simple as holding a door open for someone.
- Shoulder Surfing - securing sensitive information by directly observing the user's activity. This can be done by glancing over the target's shoulder.
- Dumpster Diving - SE reconnaissance. The hacker is looking for any and all information they can find to later use for spear phishing/whale attacks. While DD is generally legal, different states have specific laws for this.

Outdated or Unpatched Software:

Vulnerabilities are inevitable when it comes to software development. As software is patched, the vulnerabilities are handled, and new features are often added as well.

Misconfigured OS/Firewalls:

Having your internal network or servers exposed to the internet is a significant risk to an organization. If this occurs, hackers have access to the user's traffic/data, and can compromise the network.

A Firewall's purpose is to serve as a barrier between the network and the internet. It tracks in/outbound traffic, filtering what comes in and leaves the network.

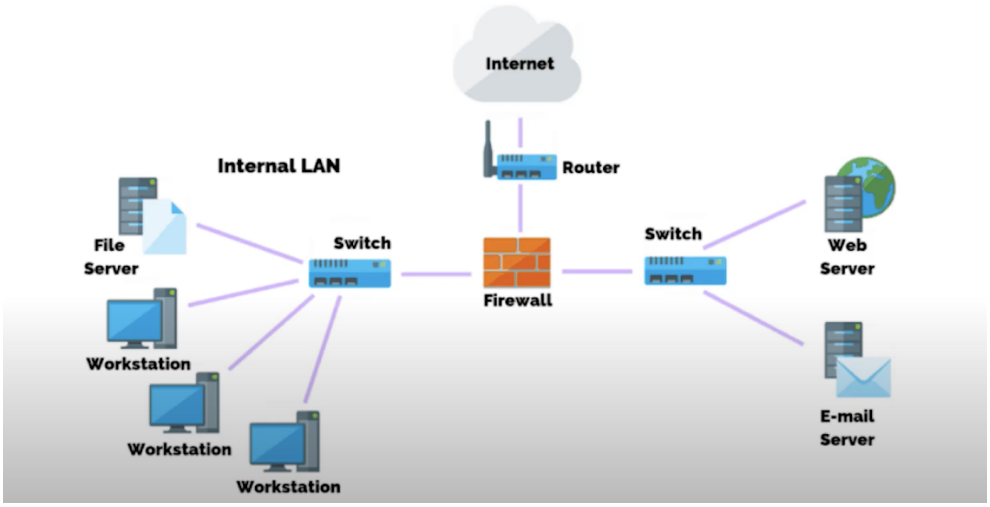

Figure 6: Above is a graph representing the purpose of a firewall

d) Introduction to the CDP's Security Protocol

Source: CDP wiki - Home

ClusterDuck was created as a mobile mesh network and is a firmware, making use of Long Range radio, WiFi, Bluetooth, and other sensor integrations. Without needing any pre-downloaded software or specific hardware, CDP gives the ability to establish networks for communication in case of emergency.

i. Setup - The Ecosystem of the CDP

As previously stated, ClusterDuck is a firmware, which can be loaded into any IOT (Internet-of-Things) electronics available - turning that device into a Duck.

These Ducks are nodes that connect to form the full CDN (ClusterDuck Network). There is no need for pre-downloaded software, since any one present at the scene of emergency can connect through Bluetooth/WiFi with either a computer or smartphone to send messages through the network.

ii. The Different Roles of Ducks

First, let's overview the unique roles of these Ducks for transmitting and receiving data. There are three primary Ducks outlined below:

DuckLink: They are edge nodes and serve only to transmit data. DuckLinks are either remote sensors or additional access points for a Captive Portal.
MamaDuck: MamaDucks can both transmit and receive data. They take the messages they receive (from a DuckLink or another MamaDuck) and repeat them until they reach a PapaDuck.
PapaDuck: PapaDuck's collect all the data and either push it to the cloud, or store it.

iii. The Data Structures of The CDP

Message Handling

To briefly touch on the past methods of message handling, there were 8 (at least) different message types coming into the network. Each used ACII text strings for identification and needed to be parsed - which was problematic as they could be accidentally altered.

But, due to changes in the CDP library, there are new methods for handling messages. Now, topics are single binary characters, which can encode no more than 256 other topics. There are specific reserve topic numbers for emergency medically-related messages. These topics can never be used in applications.

An example:

PING - "are you alive", PONG - "yes I'm alive"

Message topics are meant to be as short as possible, in order to prevent over-exhausting the mesh.

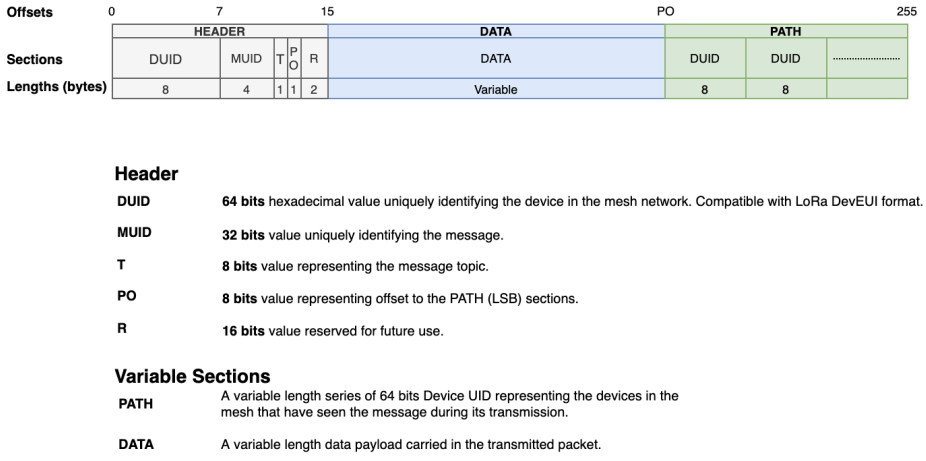

Figure 7: Above is a graph of how the new messaging system is laid out

What is The Captive Portal's Purpose?

The Captive Portal is what makes it possible for the CDP to operate without needing any pre-downloaded software. When a wireless device is able to connect to a Duck's WiFi, the page that comes up is the Captive Portal. This page will ask the user questions regarding their state of emergency.

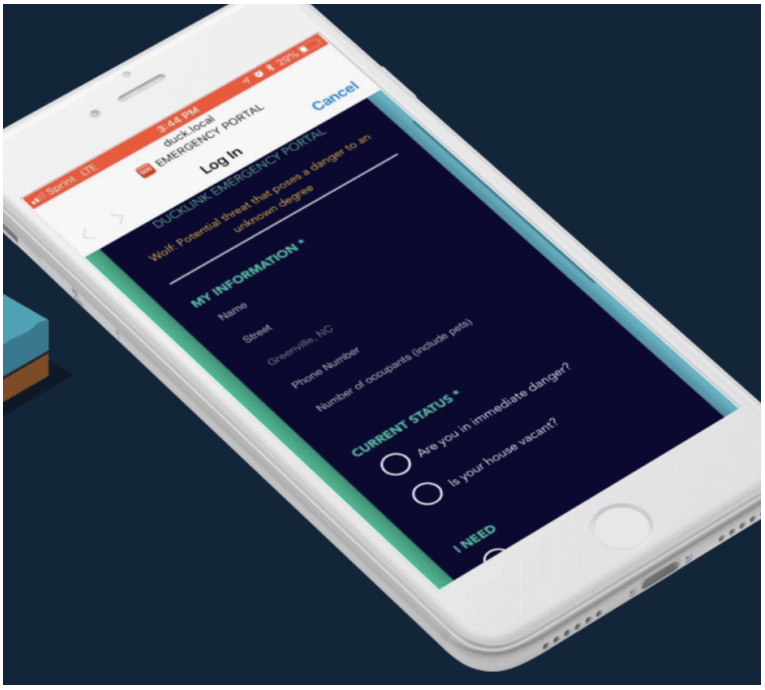

Figure 8: The Captive Portal