Social Engineering in the Internet of Everything

By Ryan Heartfield and Diane Gan

It is well known that social engineering attacks are designed to target the user-computer interface, rather than exploiting a systems technical vulnerability, to enable attackers to deceive a user into performing an action that will breach a system's information security. They are a pervasive and existential threat to computer systems, because on any system the user-computer interface is always vulnerable to abuse by authorised users, with or without their knowledge.

Historically, social engineering exploitations in computer-systems were limited to traditional Internet communications such as email and website platforms. However, in the Internet of Things the threat landscape includes vehicles, industrial control systems and even smart home appliances. Add to this mix naive users and default passwords that are extremely weak and easily guessed and the threat becomes greater. As a result, the effects of a deception-based attack will now no be longer limited to cyberspace (stealing information, compromising a system, crashing a web service ... etc.), but can also result in physical impact, ranging from manufacturing plants being damaged, trains and tram signalling disrupted causing death and injury, water treatment plants discharging sewage to damage to a nuclear power plants, e.g. *STUXNET*.

In December 2014 damage was caused to a German steel mill furnace when hackers used targeted phishing emails to capture user credentials in order to gain access to the back office and ultimately the production network with devastating consequences. Another example occurred when households in Ukraine suffered a blackout on 23rd December 2015 caused by an attack which brought down the power grid. Again, the attackers used phishing emails to trick users at the electricity company into clicking on an attachment in an email, purportedly from the Prime Minister of Ukraine. This is thought to be the first cyber-attack which brought down an entire power grid leaving 80,000 homes without electricity.

The more effective such cyber-physical attacks prove, [1], the more the deception attack surface continues to grow. For example, in the near future, fake tire pressure alerts shown on a car's dashboard or gas leakage warnings on a SMART heating system's GUI may be used to achieve deception in a manner not too dissimilar to current scareware pop-up alerts experienced by today's mobile and desktop users. In the extreme, attackers may even begin to target medical devices (such as pacemakers or mechanical syringes delivering insulin) via near field communications or wireless sensor networks, in an approach analogous to ransomware. This has already occurred through the IoT using conventional hacking techniques (SSH vulnerabilities and unpatched systems with default hardwired passwords) and is commonly known as a *MEDIJACK* attack. The major problem with these devices is that they remain unpatched throughout their life-time and at the moment this is also the situation within the IoT. In figure 1 an overview of current and future IoT user-to-system interfaces provide a snapshot of the potential social engineering threat space.

Would your Fridge lie to you?

Prior to the advent of the IoT, an email or instant message purporting to originate from your fridge would seem ludicrous. Nowadays, however, the concept does not seem so absurd. In fact, it is exactly this change in our expectations from the way we use technology and the increasing capabilities of system-to-system communication that poses the most risk. Today's users expect greater visibility and control over their environment; leading a proliferation of distributed interfaces attached to what were traditionally isolated systems, sharing new types of data across a cyber-physical boundary. The result, an augmented attack surface at the disposal of willing cyber criminals. Since IoT devices themselves may not always be directly exploited, instead it is the distributed functionality and associated behaviour integrated into new and existing

Figure 1 - The Internet of Everything: SMART devices, cars, homes, cities, people...

systems that can be targeted. For example, it would not be unreasonable to imagine an attacker crafting a spoofed instant message from a user's fridge, reporting that the fridge is running low on milk and asking whether they would like to place an order; with the Amazon style "one-click" ordering button (which conveniently leads to a drive-by download). But how did the attacker know their fridge was empty? Well, in the IoT they simply sniffed seemingly unimportant, unencrypted sensor node data sent from their fridge to their home automation controller; which connects to the user over the Internet via their home broadband router. Here, the attacker has exploited platform functionality that interfaces with the IoT device, in this case a fridge, by manipulating the perceived behaviour of the system as opposed to the device itself. In practice, such an attack can lead to a conventional exploitation such as system compromise or theft of banking credentials. It is not a great leap to envision that your fridge could be held to ransom by ransomware. Pay up or your fridge won't turn on.

Unlike phishing emails claiming to originate from financial institutions and banks (which have existed for nearly 30 years), users are not sensitive to malicious behaviour originating from home/city automation systems, smart devices or social media platforms providing access to e-health, emergency or public services. To a large extent, this is because the physical appearance of such systems do not require significant change to become compatible with the Internet of Things; normally it is only the data these platforms generate that is shared. Specifically, the IoT is enhancing data accessibility which is further augmenting the attack landscape for attackers seeking to develop convincing social engineering attacks.

Consider an attacker that is able to passively capture data from a wireless sensor in a workplace bathroom, where the sensor reports when the automated lighting is activated.

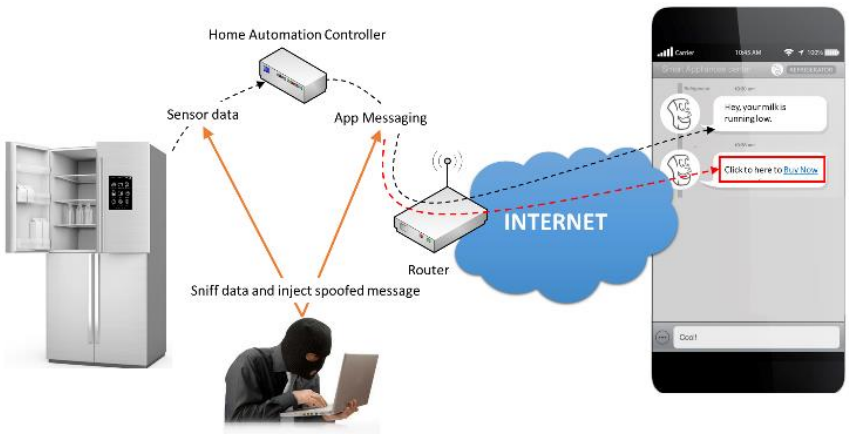

Figure 2 - Attacking a SMART fridge through intercepting and injecting spoofed application messages

Here the attacker then uses the employee bathroom data collected to profile users and send targeted phishing emails promoting cosmetic products, such as makeup or shaving discount voucher links at specific times during the day.

Data leakage: No data is too BIG or SMALL

Just as the Internet of Things expands the different types of user-interfaces that can be targeted by attackers, consequently, the different types of data (previously hidden from attackers) that can be acquired is also increased. It is well known that attackers are adept at gathering user data and utilising this information as a mechanism to target a user and to better design an attack specific to the user's system or to improve the credibility of the deception techniques that are used. Nowadays, social networks are used by hackers to obtain personal data about a user, for example your children's names, pet's name, dates of birth, where you graduated, etc. By detecting and exploiting systems which are of high value and using your "pattern of life" data, cyber-criminals can develop effective deception mechanisms by manipulating information the user is has shared and is therefore is and very familiar with has little reason to repudiate. Data leakage is exacerbated when geolocation is turned on in IoT devices. For example, anyone can then determine the exact location where a Smart phone picture was taken, which can be a problem if this identifies your home and you have just tweeted that you are going away on holiday. Burglars use Twitter as well!

Recent research by the C-SAFE team at the University of Greenwich has demonstrated the ease with which an individual can be profiled through their leaked personal data using only social networks (Facebook, Twitter, LinkedIn, Instagram …etc.) [2]. A series of experiments were undertaken to determine how much information could be extracted about three subjects using only social networking sites. By utilising three freely available tools (Twitonomy, Streamd.in, Creepy) that harvest information from Twitter, the data revealed where the three subjects lived, worked, the route they took to work each day and in one case

where their parents lived and even where and when another subject went to the gym. It was also possible to follow each of them through cyber space to other sites such as Facebook, LinkedIn, Foursquare and Instagram where information missing from their "profile" was quickly filled in. The experiment demonstrated how easy it is for cyber-criminals to gather personal data to construct social engineering attacks which an individual would find credible.

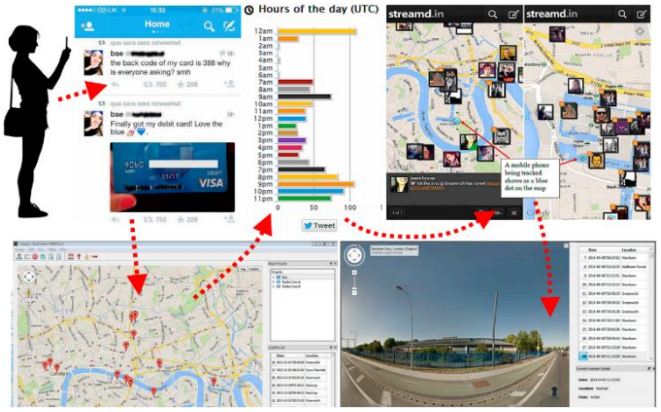

Figure 3 - Example of cyber stalking experiment monitoring and geo-locating Tweets from Twitter user, Twitter feed (top left, middle), Creep.py (bottom left, right), Streamd.In (top right)

"SMART"er Attacks

Social engineering attacks against IoT devices are by no means "hypothetical" and exploitations abusing functionality in SMART devices have already been observed in the wild.

For example, over the period of December 2013 to January 2014, security provider Proofpoint identified a cyber-attack that was originating from the IoT, where three times a day, in bursts of 100,000, malicious emails targeting businesses and individuals was sent out. In total, the global attack consisted of more than 750,000 malicious emails originating from over 100,000 everyday consumer gadgets, 25 percent of which originated from smart TVs, home routers, and even one fridge [3]. Crucially, the attack demonstrated that botnets are now IoT botnets, capable of recruiting almost any device with a network connection and messaging software.

In the following two hypothetical social engineering attack scenarios, each attack is practically facilitated by the functionality provided in the IoT.

Attack Case A: IoT Phishing in Smart Homes

Smart homes are becoming more common as people connect up numerous devices and "things" within their home. All these IoT "things" and devices connect to a network, be it wireless or wired and eventually connect to a routing device. Individually they may not offer any obvious value to cybercriminals, however they can provide a user interface which an attacker can manipulate to execute a social engineering attack. The following attack considers a threat actor who has gained control of a brand of IoT Smart meter cloud-based services platform; bundled with the product to deliver updates or new content. Here, the attack can either monitor (what may be) unencrypted communication between the cloud services and the smart meter and inject information into existing data flows, or potentially send direct messages to the meters if they have gained complete control over the cloud environment.

In both examples the attack triggers a message to all the smart meters which is displayed when the heating sensor indicates that the users are home (e.g. it has been turned up/down): "Software Upgrade Required, Go to: www.heaterupgrades.com/smartupgrade", Run the patch from a Windows computer on this network". If the user complies then they have been phished.

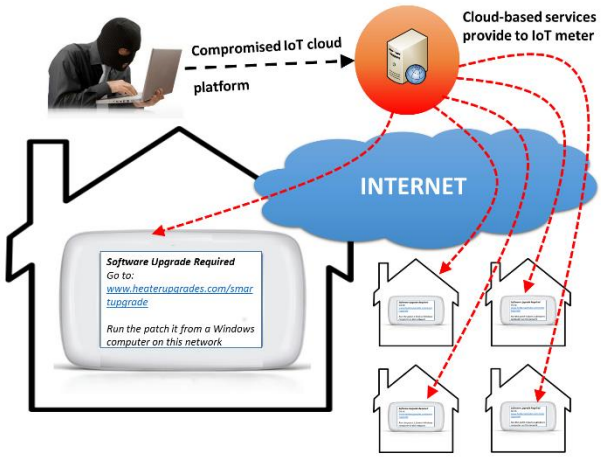

Figure 4 - Example of a smart meter phishing attack via compromised update and content services in the cloud

Attack Case B: The Internet of Social Things

Social networking and media is at the heart of the IoT, where it is no longer only people that share information with other people, but also "things" that are able to communicate with users or with other "things". Think back to your fridge kindly advising that you are low on milk. Your car might even want to tell your Facebook friends that its carbon footprint is less than 4 other cars on the road this week (e.g. in-product advertising across social media). The following attack considers a threat actor scanning Twitter, looking for status posts that include meta-data from IoT picture frames. IoT picture frames often come bundled with an app that allows a user to automatically download and upload pictures to popular social media platforms. In this example, the attacker finds a tweet containing the meta-data, however it is a re-tweet from an open Twitter account following a particular user who owns the target picture frame. Next the attacker sends a direct tweet to the user (who's account privacy settings were locked down), from a spoofed Twitter account pertaining to be the picture frame's manufacturer. The tweet contains a shortened URL to a Twitter app that will allow the user to install video functionality on their picture frame for free. In reality the Twitter app gives the attacker's account rights to download all the pictures from the users IoT picture frame, which they plan to use as ransomware data or to craft future phishing attacks.

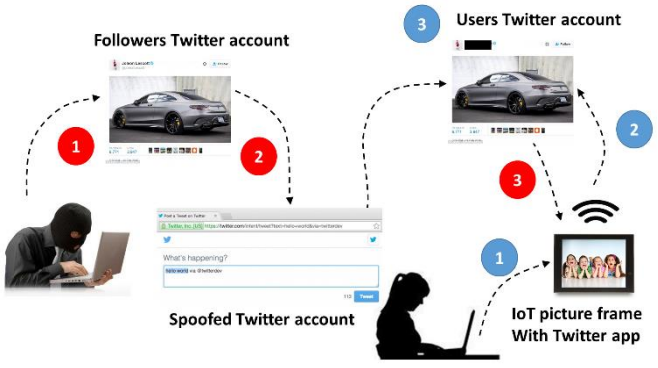

Figure 5 - Example of attacker exploiting Internet of Social Things contagion to deliver social engineering attack

Defence Recommendations

Principally, in order to instil confidence and encourage uptake in smart technologies that underpin the Internet of Things and for them to be usable in the long-term, it is necessary for the security of these devices to be robust, scalable and above all practical. Here, four approaches to defence are explored.

Generic Attack Classification

Since deception-based attacks in the IoT can be launched in either cyber or physical space identifying the source of a deception attempt and the structure of a social engineering attack can be extremely challenging. For developers, the challenge of building an effective defence that addresses a range of deception vectors would appear insurmountable when one takes into consideration all of the different platforms that may been involved in an attack. It is more practical to employ a generic classification criterion to breakdown down attacks into parametrised, components parts. This approach can be used to reveal shared characteristics between attacks; which then aids the design of defences that address multiple threats sharing similar traits. Using the taxonomy proposed by Heartfield and Loukas [4] and summarised by each root category in Figure 6, the following recommendations can help developers capture the multiple variables involved in the construction, delivery and execution of a social engineering attack, by applying criteria that are independent of the attack vectors used.

Figure 6 - A high level summary of taxonomic classification criteria for social engineering attacks in the Internet of Things

Orchestration:

Target Description [TD]: How is the target chosen? Determine an attack's targeting parameters to define which user and/or system features a defence system should focus on. A targeted attack is likely to exploit a specific user's attributes leaked by their IoT footprint (e.g. a toll payment spear phishing email based on the tweets mapped to the geolocation of their vehicle) as part of the deception. Whereas promiscuous targeting is opportunistic and random (e.g. an attacker plants a malicious QR code in a shopping centre).

Method of Distribution [MD]: How does the attack reach the target? Investigate the method in which the attack's deception is distributed and where it is executed to identify the platforms that are involved in the attack. Whether it is a remote (hence involving a network) or local system that requires monitoring and defending.

Mode of Automation [MA]: Is the attack automated? Recognising whether an attack is automatically or manually executed will help determine the most suitable response mechanism or the type of data that can be meaningful to collect about it. It may be possible to fingerprint a fully automated attack based on patterns of previously observed behaviour, while a fully manual attack may need to focus on the attacker's behaviour instead.

Exploitation:

Deception Vector [DV]: Is it looks or behaviour that deceive the user? A defence mechanism needs to pinpoint mechanisms by which an attacker can deceive the user into a false expectation by manipulating visual and/or system behaviour aspects of a system. Within the IoT, it is not just graphical user interfaces that can be abused, but the physical appearance or state of a sensor node in a home/work/city automation system as well (e.g. heating thermometer, heart beat monitor, vehicle speed, traffic lights ... etc.)

Interface Manipulator [IM]: Is the platform used in the deception only (ab)used or also programmatically modified? Depending on the system involved in an attack, it may be impractical or impossible to patch directly (e.g. pacemaker, legacy actuator ..etc.). In order to reduce the scope of a defence, developers need to establish whether the deception vector in an attack occurs in code (e.g. embedded within the system or external), or if the attack abuses intended user space functionality built into the platform by design.

Execution:

Execution Steps [ES]: Does the attack complete the deception in one step? Model the effect that a single user action can have on the integrity of a platform, as it may be necessary to build in extra user authentication steps to commit actions; especially in e-health services or industrial controls systems. An attack that relies on multiple user response steps may be detected earlier and more easily than a single-step attack, and before it completes by looking for traces of its initial steps.

Attack Persistence [AP]: Does the deception persist? Persistent deception attempts can be modelled by a learning-based defence system to identify its pattern of behaviour in order to block it. At the same time, it may also have a higher chance of success against the target. One-off deception attempts are by definition more difficult to detect and may be missed by a defence that is only looking for patterns in system behaviour or if the pattern is as yet unknown, i.e. a zero-day vulnerability.

S-SDLC

It is important that IoT platform developers have a detailed understanding of how their system will interface with users, as well as how system functionality may affect the wider ecosystem in which the system is be deployed. The Secure Software Development Life cycle (S-SDLC) provides developers with a guideline framework for the design and implementation of system software by integrating security considerations systematically into the core requirements and design of the software's architecture.

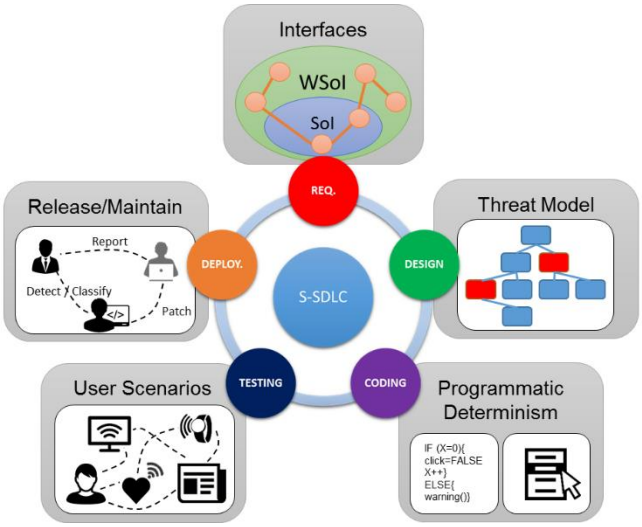

Figure 7 - Key concepts in the S-SDLC lifecycle for developing resistance to deception-based attacks in the IoT

Within the S-SDLC framework, see Figure 7, under each life cycle stage the following key concepts can aid the development of IoT platforms and functionality that are resistant to deception-based attacks.

Requirements.

Identify the attack surface for an IoT platform by clearly defining the intended functionality and its expected limitations. Document the system-to-system and system-to-user interfaces forming the overall system of interest (SoI) and identify how these communicate and effect interfaces within the wider SoI (e.g., the deployment environment).

Design

Develop threat models that run through different features of the platform's design and WSoI interactions. Pinpoint weak spots in the user interface that can be abused, or vulnerabilities in data transfer and network communications that may allow attackers to inject malicious data, code or gather information about the user.

Coding

Employ static code analysis to determine whether the platforms programmatic features are deterministic to ensure spoofed or injected data does not force the platform to exhibit a deceptive behaviour towards the user. Similarly, evaluate user interface controls (whether graphical or physical e.g. a button) to identify whether these can be (ab)used through intended functionality.

Testing

Design and implement scenarios where different user behaviour is arbitrarily executed (e.g., fuzzing), in order to identify anomalous situations when the user interface or functionality can become part of a deception-based attack. In testing, developers should generate and execute random input parameters, physical and logical, against the IoT platform in an attempt to elicit unhandled or anomalous behaviour that may lead to exploitable vulnerabilities.

Release/Maintain

Establish monitoring or reporting functionality within the platform deployment environment to help detect attacks that will facilitate continuous patching and security hardening of the specific platform and/or external platforms that have lower security features.

Attack classification and Defence

By applying each taxonomy criteria against each of the two IoT attack cases, classification is used to employ S-SDLC 6 principles that help suggest a single approach to defence that would prevent both attacks.

Case A:

TD. **Promiscuously targets** any user who owns the smart meter, by flooding connected devices with messages and commands (e.g., malicious updates) via the cloud.

MA. Functions as an **automated message** sent from the cloud-based service.

MD. Distributed to execute deception via **local software** on smart meter.

DV. Deception is both **cosmetic and behaviourally** convincing as the user would expect communications from the cloud platform.

IM. Injecting malicious messages through the cloud attack the **programmatic interface** of the smart meter by adjusting the internal code to display a deceptive message.

ES. The user must exercise **multiple steps** in order for the deception to be successful, first step downloads the supposed patch, the second step then requires the user to install the patch.

AP. The message's particular deception **is one-off** as it is unlikely the attacker will reissue the same phishing message to preserve the attack's integrity.

Case B:

TD. **Promiscuously targets** any user who owns an IoT picture frame with social media app functionality.

MA. Functions as a **manual operation** by searching for tweets, then creates a custom twitter account and tweets once a target is found.

MD. Distributed to execute deception via **remote software** on the Twitter platform.

DV. Deception is **behaviourally convincing** as product suppliers often communicate with customers via social media, as to gain customer data analytics. It is unlikely the Twitter account is visually credible (e.g., there are little or no followers, and as the account is not official tweets are not authenticated (no blue tic!)).

IM. Here the attacks simply (ab)uses the **user interface** functionality of the Twitter platform.

ES. The deception completes in **multiple steps**, as the user must click on the URL and then add the malicious twitter app permissions to their account.

AP. The messages particular deception is one-off as it is unlikely the attacker will reissue the same phishing message to preserve the attacks integrity.

By applying the taxonomy classification to each attack case we establish that a number of similar traits are shared in the orchestration, exploitation and execution phases. Firstly, both attacks target users promiscuously, so it would appear the attacker is seeking to build the deception around a vulnerability in an IoT platform and its use case; rather than a specific user's platform profile. Both attacks are behaviourally deceptive, irrespective of whether they are visually convincing or not, and both attacks are one-off in their deception, but require multiple user steps to complete the deception and exploitation. By identifying that both attacks focus on the IoT product behaviour, rather than the users, it is clear that the **S-SDLC requirements** and **testing** stages would play a pivotal role in helping to mitigate these attacks. Crucially, it is the system-to-system interfaces of each IoT platform and their interaction with the ecosystem's wider system of interest (e.g. Case A: cloud-based services over the Internet, Case B: Twitter application add-ons) that needs addressing.

Analysis of each of the IoT devices, their interface contracts between other IoT platforms/devices and the functionality they extend should be clearly defined and then evaluated against different user deployment scenarios. By doing so, developers can identify specific functionality supplied by the system which is vulnerable to manipulation. Here, the manipulation of features supplied by the IoT devices in each attack case could easily be highlighted by reviewing each interface contract, then conducting a robust test of its functionality in different user deployment scenarios. Since both attacks deceptions are one-off they may be hard to identify and prevent, therefore it is even more important to rationalise system interface requirements before providing the users with functionality that the developers are not able (or willing) to protect. Where each attack requires multiple user steps to complete, integration of further authentication mechanisms for more significant functionality requests between interfaces should be enforced and reviewed through testing. This approach can help to identify if extra security procedures should

be enforced before a user commits a potentially compromising action (e.g. force a user to review a warning or confirm their identity through multi-factor authenticating).

User Susceptibility Profiling

In order to provide a robust defence against social engineering attacks, responsibility cannot solely be laid upon the shoulders of system developers or organisations providing access to a computer system, whether that is an IoT platform connected to the Internet, a local area network or near-field communications medium. On the contrary, the users of the system are just as important, if not relied upon even more to act and use the computer securely to ensure that their actions do not inadvertently result in information security compromise. Remember, there is no silver-bullet for protecting against human-error.

However, identifying a key set of user attributes that can be measured can help to provide a basis for modelling which type(s) of user profile are more or less likely to be susceptible to a deception-based attack. Such attributes could be used to define features for predicting and estimating user susceptibility when using a specific platform or range of platforms. Crucially, access to a user susceptibility profile provides the basis to apply a threshold in which the probability of user susceptibility triggers security enforcing actions aimed to minimise and/or mitigate exploitation.

Human as a Sensor (HaaS)

The concept of the human as a sensor has been employed extensively and successfully for the detection of threats and adverse conditions in physical space, for example to report road traffic anomalies, detect unfolding emergencies and improve the situational awareness of first responders through social media [10]. In a similar manner, human sensing can be applied to detect and report threats in cyber space as well. In fact, as the IoT crosses the cyber-physical boundary, the ability for users to report suspected attacks, both cyber and physical, may help to detect attacks initiated in one space that results in an effect on the other. In this respect, it then becomes particularly important to be able to tell to what extent users can correctly detect deception-based security threats; leveraging the intelligence provided by users to augment IoT cyber situational awareness.

Within a smart city, users are likely to be exposed to many different IoT interfaces such as advertising, multimedia and wireless multicast feeds in the local geographic area (e.g. local car park capacity, what's on at the cinema, popular restaurants ...etc.). Should any of these interfaces be targeted by an attacker using social engineering, users become an important source of information if a deception attempt is identified. In this example, the user can open their HaaS tool within their smart phone to report any suspected attacks, which can then be directly fed to the Smart cities security monitoring system. Free car parking might even be an incentive for correctly reported attacks!

Conclusion

The IoT promises to synergize technology in new and innovative ways and in doing so presents major social, business and economic benefits for modern society. Equally, for cyber criminals, the IoT promises significant rewards if a social engineering attack is executed successfully, because hacking the user can provide access to all the "things" that they control. The more successful social engineering attacks against the IoT are, the more user confidence in the IoT's security is undermined, ultimately delaying its adoption and the realization of its potential benefits.

Fundamentally, protecting the integrity of the IoT is a two-way street. System developers should ensure that they employ best practice frameworks for producing secure IoT platforms. The example provided here is the S-SDLC. However, the wider message for IoT platforms is that security should be treated as an enabler of system functionality and not be a cost based bolt-on or ignored completely. Equally, users are a crucial firewall in detecting social engineering threats in the IoT and it is important that they are empowered to report potential threats, especially as they will be familiar with their own environment and more sensitive to its anomalous behaviour. Of course, at the same time, it is helpful to be able to measure whether users will be deceived by social engineering attacks in an IoT ecosystem and therefore as part of security awareness it is crucial that the IoT is factored into training material.

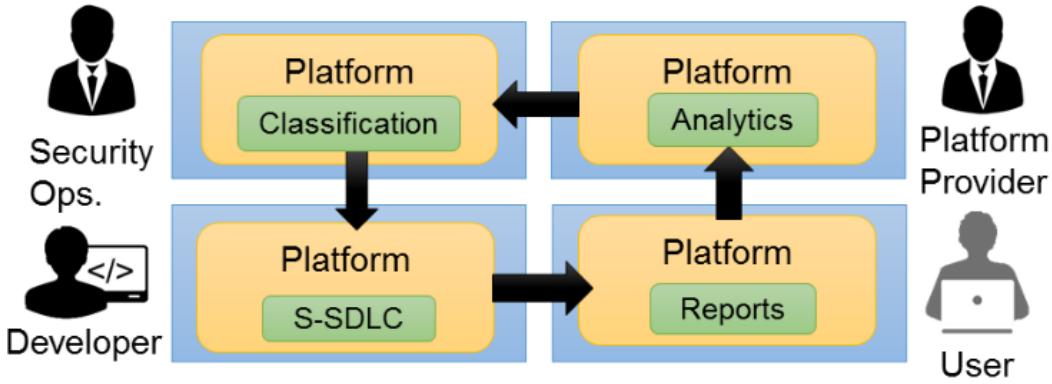

Figure 8 - A four phase approach to through life management of user interfaces in an Internet capable platform

Finally, each of these approaches, as shown in Figure 8, provide complimentary tools that help provide a through life defence architecture against social engineering attacks in the IoT. To improve IoT security, system developers must empower user threat detection with a mechanism to report suspected attacks and review/analyze user reports to determine their credibility. If they decide an attack report is credible, they can then apply a generic classification to determine the key aspects of the attack and finally integrate these attack vectors as patch parameters within the platform 'release/maintain' phase of the S-SDLC.

As Bruce Schneier once said, “People don’t understand computers. Computers are magical boxes that do things. People believe what computers tell them.”

Trust lies at the heart of securing the IoT against deception-based attacks, and thus in order to instill trust, it is device integrity that must be protected to prevent user compromise.

References

[1] G. Loukas, Cyber-Physical Attacks: A Growing Invisible Threat. Butterworth-Heinemann (Elsevier), 2015.

[2] D. Gan and L. R. Jenkins, “Social networking privacywhos stalking you?” Future Internet, 2015.

[3] Proofpoint, “Proofpoint uncovers internet of things (iot) cyberattack,” 2014. [Online].

[4] R. Heartfield and G. Loukas, “A taxonomy of attacks and a survey of defence mechanisms for semantic social engineering attacks,” ACM Computing Surveys, vol. 48, no. 3, 2016.

[5] S. K. Boddhu, R. Dave, R. Williams, M. McCartney, and J. West., “Augmenting situational awareness for first responders using social media as a sensor,”, 2013