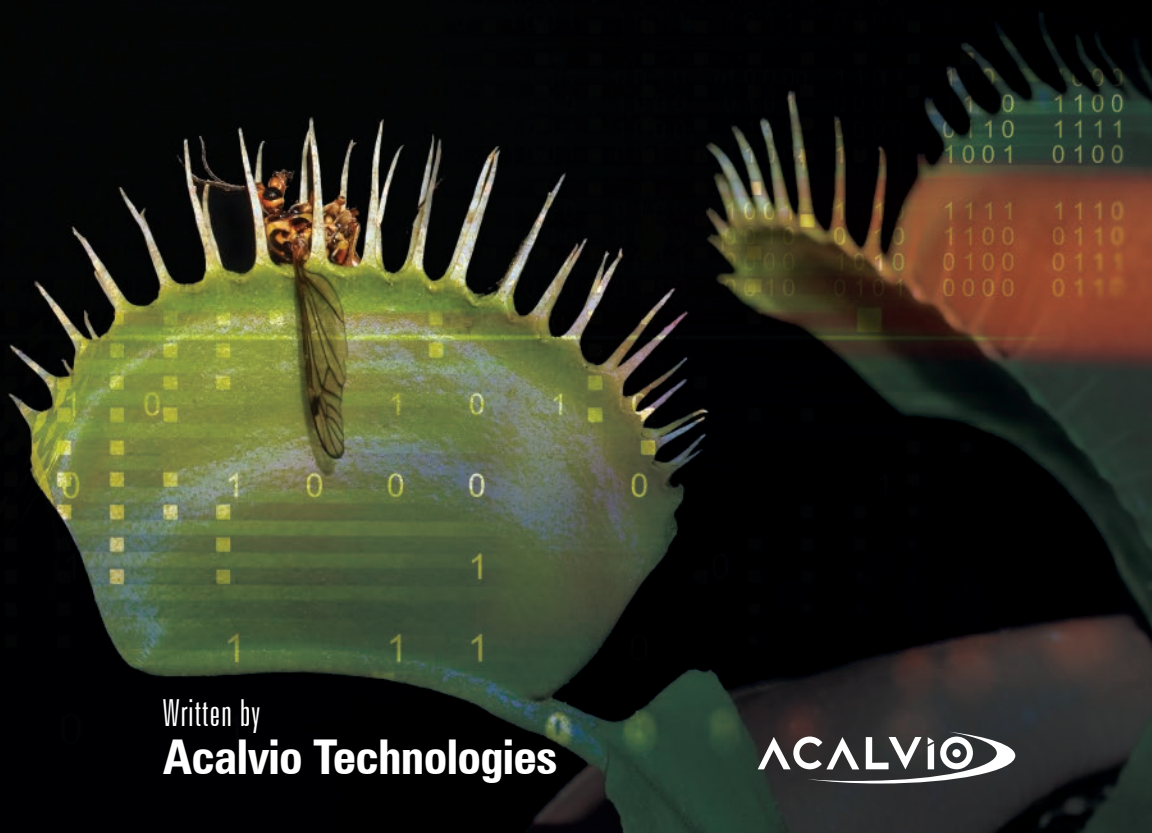
DEFINITIVE GUIDE TO DECEPTION 2.0

Cybersecurity Manual for Distributed Deception Solutions

Foreword by

Dr. Gerhard Eschelbeck

VP of Security and Privacy Engineering at Google

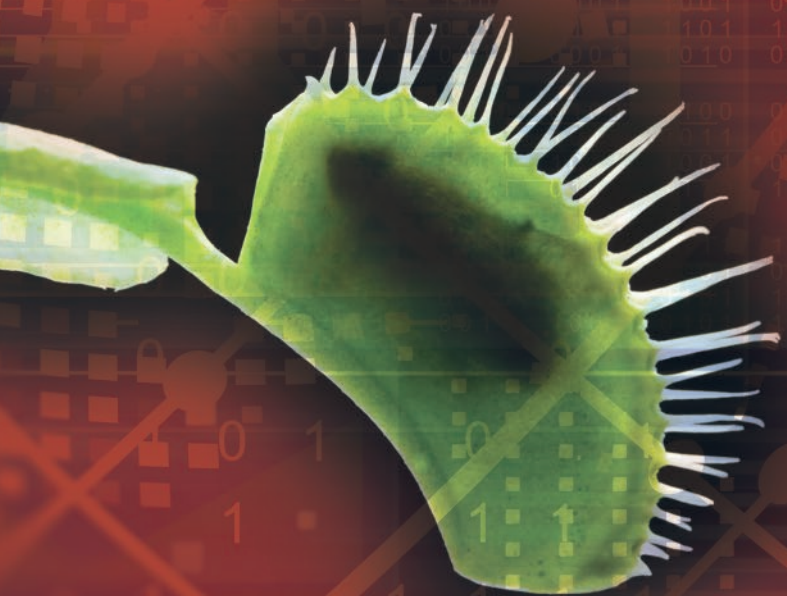


Written by

Acalvio Technologies

ACALVIO

Acalvio Technologies
2520 Mission College Blvd, #110, Santa Clara, CA 95054
www.acalvio.com



Foreword

I am an ardent believer that we can and should out-innovate the threat actor. I am also a huge believer in the power and potential of Deception technologies to delay, deflect and ensnare the threat actor in a high fidelity, timely and cost-effective fashion. Currently, there exists a fundamental asymmetry in the security industry – we must get it right all the time, the threat actor must get it right only once. Deception turns this asymmetry on its head to our benefit – with Deception, the bad guy must be wrong only once to get caught.

Having said that, there are several practical challenges in the design of effective Deception solutions. I have had the pleasure of getting to know the Acalvio team and I am very impressed with their innovative approach in getting Deception ready for the enterprise. They have thought through Enterprise Deception in a very comprehensive, holistic and practical manner.

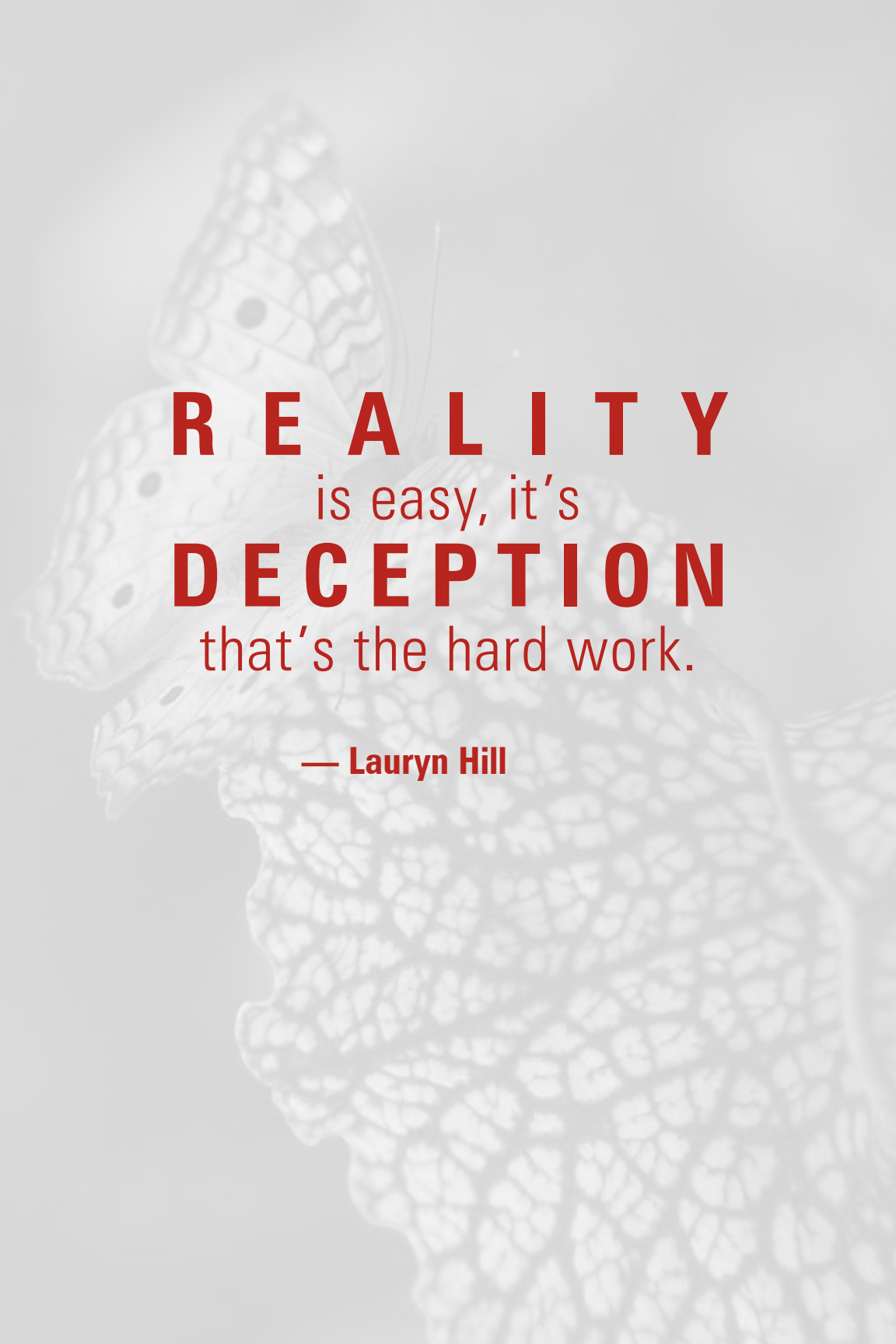
I believe you will find this book to be a very hands-on compendium of Deception technologies.

I would like to wish the Acalvio team the very best in their endeavor to make Deception technologies a mainstream aspect of the Enterprise Security Stack. Likewise, I would like to wish you the very best in leveraging the power of Deception to stay a few steps ahead of the bad guys.

Dr. Gerhard Eschelbeck
VP Security & Privacy Engineering at Google

DEFINITIVE GUIDE TO DECEPTION 2.0

CYBERSECURITY MANUAL FOR DISTRIBUTED DECEPTION SOLUTIONS


REALITY
is easy, it's
DECEPTION
that's the hard work.
— Lauryn Hill

1 THE ART + HISTORY OF DECEPTION

IN THIS CHAPTER:

1. **Use of deception as a part of natural selection in evolution**
2. **How deception has been adopted as an art by humankind**
3. **Failure of traditional security mechanisms in safeguarding "porous" network perimeters**
4. **Role of deception in cyber-security**

Evolution has often relied on deception as a survival technique. We start with deception in nature to provide an understanding of the nature of deception. For creatures with higher cognitive abilities, the capability to deceive is linked to creativity. The same cognitive skills that enable imagination go into telling a successful lie. Inventiveness of mankind has further improved upon nature's design and made deception into a form of art. We use warfare as an example to study this art of deception. Warfare in the 21st century has moved to information technology. The last section in this chapter looks at the history of this war and the role of deception as a survival technique.

DECEPTIONS IN NATURE

In the natural world, every organism is driven to survive and propagate. Deception has been employed in various forms as a successful strategy and has played an important role in the physical and behavioral adaptations of all organisms. As psychologist Harriet Lerner observed, "Deception and 'con games' are a way of life in all species and throughout nature. Organisms that do not improve their ability to deceive – and to detect deception – are less apt to survive".

A FEW EXAMPLES ILLUSTRATE THE VARIETY OF DECEPTIONS:

PROPAGATION: Western Skunk Cabbage emits a scent quite similar to that of skunk spray to draw in its pollinators – scavenging flies and beetles.

PROPAGATION: Rye used to be a weed in wheat fields, until it began mimicking the qualities of wheat and even surpassing it in some areas through a phenomenon called *Vavilovian mimicry*.

PREY DECEPTION: *Lamium albium* – known as "dead nettle" – evolved to look just like stinging nettle. It doesn't have the same painful sting, but it gets the same benefits by association and often grows near its doppelganger.

PREDATOR DECEPTION: Angler fish have a long filament on top of their head that it can wiggle to resemble a prey animal to lure other predators close.

PREY DECEPTION: Mimic octopus (shown here) can mimic up to 15 species of local marine organisms as a primary defense mechanism.

IMPORTANT POINTS TO NOTE FROM NATURE:

DECEPTIONS ARE VARIED. There are literally thousands of examples of deception.

DECEPTIONS ARE SPECIALIZED. Each type of deception works only for a specific species, in a specific environment for a particular threat or opportunity.

DECEPTIONS ARE NOT STATIC. The arms race between the predator and prey keeps throwing up newer deceptions as the opponent evolves to circumvent the previous deception.

Illustrations: The many disguises of the mimic octopus

Deception in plants and animals does not automatically imply a conscious act. For example, when rye used to be a weed in wheat fields, it was perennial and had smaller seeds. It was artificial selection by humans (by killing the weeds and winnowing the weed rye seeds from the wheat crop) that helped propagate rye to be annual with larger seed size and more rigid spindles.

MILITARY DECEPTION

Active deception requires a higher level of cognitive ability. In humans, deception has been related to cognitive development and has been observed even in children of 2-3 years old[1]. Deception requires the ability to mentally balance reality and fiction, in addition to the capacity to recognize the difference.

Deception has always been a strategic component of warfare. Numerous historical references exists where ploys were designed to draw an enemy into weak position or to defeat or completely rout.

VARIOUS TYPES OF DECEPTIVE ACTIVITIES HAVE BEEN EMPLOYED THROUGHOUT THE HISTORY OF WARFARE, SUCH AS: [2]

FEIGNED RETREAT: Leading the enemy, through a false sense of security, into a pre-positioned ambush.

FICTIONAL UNITS: Creating entirely fictional forces or exaggerating the size of an army.

SMOKE SCREEN: A tactical deception involving smoke, fog, or other forms of cover to hide battlefield movements.

TROJAN HORSE: Gaining admittance to a fortified area under false pretenses, to later admit a larger attacking force. The most famous example being the subterfuge used by the Greeks to enter the city of Troy.

STRATEGIC ENVELOPMENT: A small force distracts the enemy while a much larger force moves to attack from the rear. A favored tactic of Napoleon.

These general tactics can be combined or customized to suit the need[2]. For instance, use of camp fires by George Washington to fool the British scouts, or the use of 15,000 dummy horses in the battle of Megiddo during World War I, or even the inflatable arsenal deployed by Russia in 2016[3] can all be seen as different forms of Deception.

In all the examples, successful use of deception in warfare depends on secrecy, concealment, originality and rapidity. The same deception can neither be used multiple times nor is appropriate in all circumstances. The deception must blend into the environment. When the dummy horses were used in the example above, they had real horses walk up and down to the nearest water source multiple times to create the illusion. A successful deception is always the result of an extremely well-orchestrated plan.

Illustration: Megiddo dummy horses in 1918; Russia's inflatable arsenal imitation T-80 tank

DECEPTION FOR CYBER DEFENSE

Information technology infrastructure has become the new battlefield in the twenty-first century. Despite the billions in financial and intellectual capital that has gone into IT security, we seem to be continually outsmarted by the threat actors. The recent spate of highly publicized security breaches, and scores that remain anonymous for obvious reasons, only go to establish that the threat actors are gaining ground, and at a rapid pace. The 2016 Verizon Data Breach Investigations Report[4] indicates that over 68% of the threat actors remain unnoticed within the Enterprise for days, weeks and months before successfully ex-filtrating valuable enterprise assets.

A detection system focuses on finding intruders or malware that have gotten past traditional perimeter defenses. Detection does not replace perimeter defenses, but provides an additional layer of protection. To use a household metaphor, in addition to locks to protect the front door, one employs motion sensors to detect and ensnare unwanted intruders in the house. Similar to motion sensors, detection solutions provide visibility into unauthorized and dangerous activity inside the network. This represents a big change from traditional "full trust" models of the enterprise network, where organizations have limited visibility into unusual activity once users gain access to organizational networks and systems.

Advanced detection solutions are based on either anomaly detection or deception. Anomaly detection solutions create a behavior baseline of various network components – hosts, data access, network traffic, user behavior etc. Anything anomalous from the baseline is flagged as an alert.

Anomaly-based solutions have two significant drawbacks:

1. Capturing, storing and distilling the data lake is complex and expensive.
2. A high rate of false positives adds significant burden to an already over-loaded security team.

Deception-based solutions offer a promising alternative. Any component in a computer network can be used for deception - a computer system, a service, a credential, a data item, etc. The deceptions are not part of the normal operation and revealed only by an attack. Any time an intruder expends the time and effort necessary to locate and access a deception set up specifically to invite attack, it is a positive affirmation of a compromise. In other words, in a deception-based solution, the malicious activity announces itself, thus making for a very high-fidelity signal.

The next chapter explores the history of using deception to detect adversaries. Just as in warfare, we will see early deception practitioners lay well-orchestrated traps to detect and respond to intrusions.

Anomaly-based solutions have two significant drawbacks:

1.

Capturing, storing and distilling the data lake is complex and expensive.

2.

A high rate of false positives adds significant burden to an already over-loaded security team.


All warfare is based on
DECEPTION.
— Sun Tzu

2 DECEPTION 1.0: HONEYPOTS ARE DEAD!

IN THIS CHAPTER:

1. **Illustrate the various ways deception can be used to detect and study threats**
2. **Development of honeypots as the first generation of deception**
3. **Types of honeypots, their placement and usage**
4. **Limitations with the honeypots**

The complete quote from Sun Tzu is, **"All warfare is based on deception. Hence, when we are able to attack, we must seem unable; when using our forces, we must appear inactive; when we are near, we must make the enemy believe we are far away; when far away, we must make him believe we are near".** Deception takes many forms and is effective only when it is consistent and appropriate for the adversary. Two of the early descriptions of using deception in cyber-defense come from Cliff Stoll's book[5] from 1989 and Bill Cheswick's paper[6] from 1992. We cover Stoll's account in some detail in the first section, since it illustrates the varieties and stages of deception. Early (and some of the current) deception solutions are based only on honeypots and the second section provides a summary. Finally, we go over different types of honeypots and discuss several drawbacks with the honeypot implementations.

THE CUCKOO'S EGG

One of the earliest detailed descriptions on how deception was used to detect and track a cyber-attack came from Cliff Stoll. His first person account, *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage*"[5], became a *New York Times* bestseller. The ingenuity and orchestration of his deceptions provide a framework on how to build an effective deception-based solution.

Cliff Stoll was an astronomer turned systems manager at Lawrence Berkeley Lab. On the second day in his new role in 1986, he was intrigued by a 75-cent accounting error in the billing for system usage. His efforts to unravel the discrepancy eventually became a year-long quest. We present a few highlights of his study.

HIGHLIGHTS FROM 'THE CUCKOO'S EGG' BY CLIFF STOLL:

The account causing the discrepancy was reported as dormant by the Unix system accounting, while the home-built application showed activity. Stoll concluded that there was a hacker in the system who must have acquired system privileges to modify the Unix accounting file, but was not aware of the home-built one.

To track the hacker's activity, Stoll physically wired four dozen teletypes and portable terminals to each of the modem lines, recording every keystroke that came through. This enabled Stoll to study the hacker without being detected.

1. The hacker used a bug in Gnu-emacs to gain system privileges.
2. The first thing the hacker did after gaining system privileges was to erase his tracks.
3. The hacker made sure no one was watching him.
4. The hacker was located about 6000 miles away, based on latency during file exfiltration.
5. The hacker kept track of everything he had done.
6. The hacker didn't succeed through sophistication. Rather he poked at obvious places, trying to enter through unlocked doors.
7. The hacker targeted military and defense contractors.

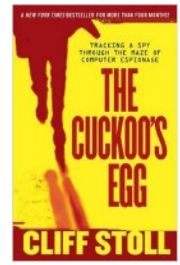
When the hacker tried to steal important files, **Stoll physically slowed down the transmission** by jangling his keys across the wires connected to the hacker's line to add noise. On other machines, he programmed trap doors that notified Stoll whenever known hacked accounts were touched.

The hacker was traced to Germany. German Bundespost set a trap on the account, but given the physical infrastructure, needed the hacker connected for at least two hours to trace the phone line.

To lure the hacker, Stoll invented a secretary who works for people doing a fake secret military project. He created lots of rough drafts, repetitive stuff, and interoffice memos that would take the hacker more than two hours to copy on a 1200-baud connection. The hacker fell for the deception and was traced and arrested.

The deceptions that Stoll used are of many varieties - computers, credentials, data (emails, documents, drafts, interoffice memos). Stoll created a complete believable persona of a secretary to lure the attacker and a fake project based on the hacker's behavior profile that would entice him to stay connected.

Stoll's investigation can be divided into three stages, based on what deception was used for.



1. **DETECT:** The first step in cyber-defense is to detect an attack.

2. **ENGAGE:** The engagement part of the deception started when Stoll decided to record the hacker's activity. By engaging with the hacker, Stoll built a profile of the hacker's exploitation techniques, other compromised accounts and servers, data he is interested in, and his behavior.

3. **RESPOND:** Stoll managed to protect multiple vulnerable installations by warning them of the hacker's activities. He slowed down exfiltration when the data was sensitive. He eventually trapped the hacker by creating deceptions based on the hacker's profile.

We will go over these stages in more detail in Chapter 3. All these are important pieces in creating an automated deception-based solution.

The cuckoo bird has the ability to lay eggs to match their prey's nest, using the host females to hatch the chicks, and later attacking the nest.

HONEYPOT DECEPTIONS

Stoll's deceptions were varied, dynamic, and precisely crafted to entice a specific adversary. Automating such elaborate deceptions needs machine intelligence and domain knowledge. Given the state of technology in the late 1990s, the early efforts on automating deception used only machine deceptions, called honeypots.

A honeypot is a computer system that is not part of the normal business processes. Theoretically no one should be interacting with a honeypot. Lance Spitzner, founder of the Honeynet Project, defines a honey pot as "a resource whose value lies in being probed, attacked or compromised". He succinctly summarizes the utility of honeypots [7]:

"A honeypot's greatest value lies in its simplicity, it's a device that is intended to be compromised. This means that there is little or no production traffic going to or from the device. Any time a connection is sent to the honeypot, it is most likely to be a probe, scan, or even attack. Any time a connection is initiated from the honeypot, this most likely means the honeypot was compromised. As there is little production traffic going to or from the honeypot, all honeypot traffic is, by definition, suspicious."

The early 2000's saw further development of honeypots, specialized for different tasks. A few of the specialized honeypots are covered below. But we first discuss the two basic flavors of honeypots, based on the level of interaction with an adversary, and where they provide value in an enterprise network.

All first-generation honeypot implementations chose low- or high-interaction as the underlying architecture, along with the associated pros and cons.

LOW-INTERACTION VS HIGH-INTERACTION

Honeypots are of two types, low and high-interaction, depending on how an adversary is engaged. Both types have their pros and cons.

LOW-INTERACTION HONEYPOTS are based on emulations of pieces of the operating system (e.g., the network stack) and any needed services. These honeypots are relatively easy to setup. The emulations are not resource intensive. Hence the main advantage of low-interaction honeypots is that a single computer can effectively emulate many honeypots. The emulations do not allow an adversary to gain access to the system, thereby avoiding the situation where the honeypots themselves are compromised by the adversary. However, this results in the biggest drawback of low-interaction – they are limited to capturing known activity. Another failing of the low-interaction honeypots is the limited set of emulations. An enterprise network usually has many complex software systems offering a variety of services, some of which may be bespoke. The low-interaction honeypots offer only the standard network services. Even a standard service, for example FTP, usually has several versions and the same version may be implemented a little distinctly (maybe individualized greeting banners) on different systems. The simplicity of the emulations (canned responses, incomplete emulations etc.) allows adversaries to easily fingerprint low-interaction honeypots and bypass them.

HIGH-INTERACTION HONEYPOTS are actual systems, not just emulations. They are no different from the systems used in the normal business processes, with actual operating systems and offering real services, and allow login access to an attacker. However, any data in this honeypot is fake and the operating system and services are instrumented to capture a detailed account of the attacker's actions on the system. The honeypot is also usually wired to capture all network traffic. The detailed attack data helps in understanding the attacker's intentions, systems compromised, exploits used, and identify any command and control center. The downside of high-interaction honeypots is that these systems are hard to configure correctly, deploy and maintain. It needs significant effort to change the nature of a high-interaction honeypot, for example to provide an upgraded operating system, modify the services or the counterfeit data. Since the attacker is allowed access to the honeypot, there is also a risk that the attacker may compromise the honeypot and use it to launch attacks against other servers. Lastly, high-interaction honeypots are costly since they are real systems. However, there is no alternative to high-interaction if an enterprise wants to understand targeted attacks and mount a proper defense.

In Chapter 4, we will see how the two are combined in the second iteration of deception technology.

HONEYPOT PLACEMENT

Honeypots can be placed either outside the firewall in an enterprise network or inside the firewall. Each of the two placements serves a different purpose.

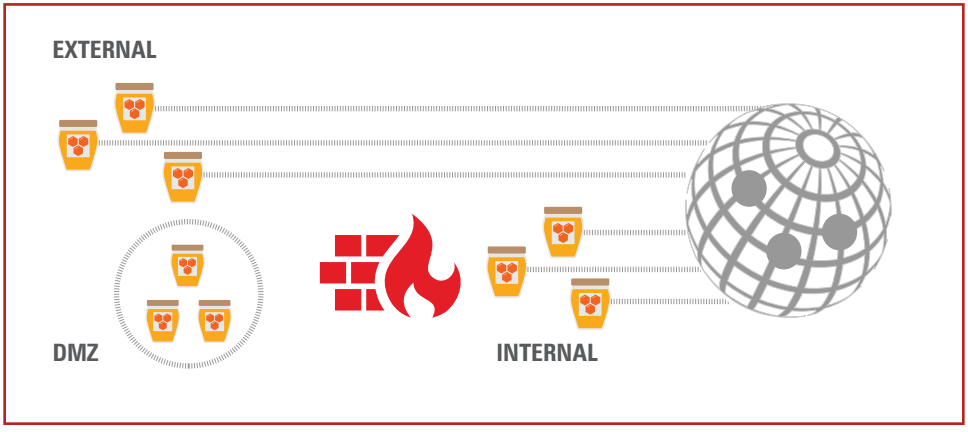
EXTERNAL: FACING THE INTERNET

The honeypot is directly connected to the internet, without any firewall protection. These are called "research" honeypots, since the main goal is to gather intelligence on the threats an enterprise may face. The honeypots share the same public IP address subnet as the production network. These honeypots are typically high-interaction systems and aid an enterprise to understand who is targeting them, their exploitation techniques and what the goals are after compromising the system. This intelligence helps the enterprise prevent the attacks by fixing targeted vulnerabilities and building appropriate detection.

INTERNAL: BEHIND THE FIREWALL

The second option for honeypot placement is inside the network. These honeypots are also called "production" honeypots, since they add to the security of the production network. Production honeypots act as a warning system for any exploits that have crossed the perimeter defenses. They can be configured to only detect an attack (low-interaction) or engage and analyze the exploit (high-interaction). As we will see in the next chapter, these honeypots can also be used to respond to an attack. In our discussion in this book, we focus primarily on production honeypots as our goal is to detect threats inside the network.

One other possible placement of the honeypots is in the DMZ to detect any exploitation activity within the DMZ. The same issues that applies to production honeypots applies to DMZ honeypots.



SPECIALIZED HONEYPOTS

Over the last decade, several specialized honeypots were built to help with security in specific environments. We cover a few of these bespoke or custom honeypots to illustrate the power of deception.

1 Gaspot

GasPot is a research honeypot for gasoline-monitoring systems. This uses deception technology to emulate supervisory control and data acquisition (SCADA) and ICS systems that monitor gasoline in storage tanks. Since several tank-monitoring systems opened to internet have suffered attacks, the researchers came up with a virtualized system with similar functionality and controls. The attacks observed and the results are detailed in[8].

2 Conpot

According conpot.org, Conpot is a "low interactive server side Industrial Control Systems honeypot designed to be easy to deploy, modify and extend." Conpot provides emulations for a range of common industrial control protocols like Modbus and SNMP, which can be used to emulate complex industrial infrastructures.

3 Phoneypot

Phoneypot is a telephony honeypot that is used to understand telephony scams based on robocalling, voice phishing, caller ID spoofing etc. The researchers report[9] that their honeypot received 1.3 million calls from 250K unique sources over a period of seven weeks and that they detected several debt collectors and telemarketers calling patterns and an instance of a telephony denial-of-service attack.

These adaptations show that deception is an effective technique for cyber defense beyond traditional computer networks.

LIMITATIONS AND EXCLUSIONS APPLY

Honeypots, as the first generation of deception technologies, were successful in demonstrating the utility of deception as part of layered security. Open-source projects, especially Honeyd and Honeynet, helped many organizations experiment with honeypots.

THE NEED FOR ENTERPRISE-GRADE DECEPTION BROUGHT UP SEVERAL CHALLENGES WITH THE FIRST ITERATION:

LOW-INTERACTION VS HIGH-INTERACTION. Enterprise deception solutions need scale, and provide a detailed understanding of the attack. Whether low- or high-interaction, honeypot implementations are limited by their architecture and could provide only one of the two.

SIMPLICITY OF LOW-INTERACTION EMULATIONS. It is a huge task to develop and customize emulations to match the breadth and characteristics of the services in every organization.

SCALE OF DEPLOYMENT. For effective detection, Gartner recommends a ratio of 10 to 1, honeypots to real systems. This level of scaling is tough to achieve even with low-interaction honeypot solutions for any decent-sized enterprise network.

FINGERPRINTING OF HONEYPOTS. First-generation honeypots exhibited distinct characteristics that never changed. This made them easily recognizable and enabled attackers to fingerprint and identify the honeypots and avoid them. Many of the open-source solutions have well-publicized fingerprints.

EASE OF USE. Honeypots were not designed for enterprises, and it made administration at scale impractical. Even if it took a minute to setup a honeypot, setting up a thousand honeypots would take more than 16 hours! Maintaining them was nearly impossible and attackers could identify them since the honeypots were static.

RISK OF COMPROMISE. Honeypots were implemented as part of the enterprise network. An attacker can use a compromised honeypot as a base to attack other systems or organizations. The risk is significant for high-interaction honeypots since the attacker is allowed login access.

HONEYPOT IS JUST ONE OF THE MANY DIFFERENT TYPES OF DECEPTIONS THAT ONE NEEDS TO EMPLOY. Honeypots are worthless if no one touches them. As seen in Stoll's work, effective deception should be in every part of the enterprise network. Deception must extend to files, emails, cached credentials, network shares, database systems, routers, printers etc.

STAND-ALONE PRODUCT. Honeypots are mostly stand-alone products and did not interact with the security ecosystem typically present in any enterprise network.

THREAT ANALYSIS. The high-interaction honeypots were used primarily to capture malware payloads, though there were some advances to study the attack patterns and identify the command and control centers.

The honeypots, as we know from the first generation, more than served their purpose. They showed the tremendous potential for deception, even in limited form, in cyber-security. Subsequent chapters discuss second generation of deception and how it alleviates the challenges identified above.

He who knows **WHEN** he can **FIGHT** and when he **CANNOT** will be **VICTORIOUS.**

— Sun Tzu

It is a double pleasure to
D E C E I V E
the **DECEIVER.**

— Jean De La Fontaine

3 ADVANCED THREATS

IN THIS CHAPTER:

1. **Advanced threats – kill chain model**
2. **Detect, Engage & Respond – the three stages of deception**
3. **Pervasive deception as a strategy to detect and mitigate attacks**

In spite of heightened attention on perimeter security, advanced cyber-attacks have been increasingly successful year over year.

In this chapter, we outline the nature of advanced threats and the different dimensions along which deception can be used to detect and respond to these threats. Thereby, we outline the approaches required for next generation deception technologies.

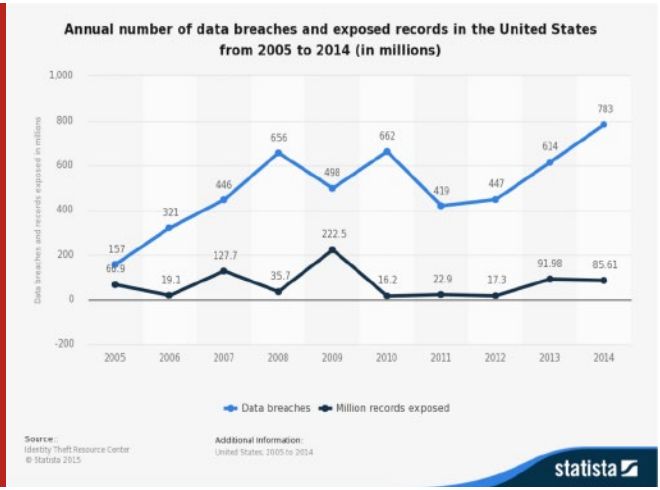
We will use the kill-chain model to understand how advanced threats operate. The next section discusses how deception can be employed against advanced threats using three progressive levels of interaction. The last section expands the definition of "deception" to encompass any resource that is not part of the normal business process and introduces the notion of pervasive deception. We also illustrate how pervasive deception is used in the different stages of a "kill chain".

CYBER THREATS ARE INCREASINGLY SUCCESSFUL

The network perimeter has become "porous" by design. Cloud computing, BYOD (bring your own device), IoT (Internet of Things) devices and applications have made the traditional cyber security perimeter defenses largely ineffective. In 2014, speaking to CBS' "60 Minutes", FBI Director James Comey said, *"there are two kinds of big companies in the United States. There are those who've been hacked…and those who don't know they've been hacked".*

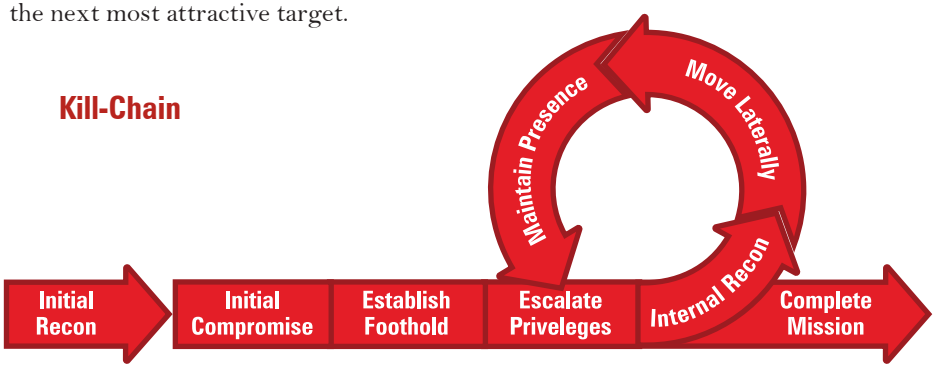
The chart shows the rise in the number of data breaches from 2005 to 2014. The average total cost of a data breach has also increased to reach $4 million in 2016[16].

In 2014, speaking to CBS' "60 Minutes", FBI Director James Comey said, "there are two kinds of big companies in the United States. There are those who've been hacked … and those who don't know they've been hacked".



THE KILL-CHAIN MODEL

The *"kill-chain"* model captures the various stages of an advanced threat attack scenario. Perimeter security systems try to prevent the initial compromise, but a persistent attack ultimately succeeds, through social engineering, phishing, or other means of compromise, and establishes a foothold. Once the threat is inside the perimeter, it spreads by escalating privileges, doing a reconnaissance of the network neighborhood and moving laterally to the next most attractive target.



DETECT, ENGAGE & RESPOND

Advanced threats establish multiple footholds using several exploits. Effective mitigation requires a detailed understanding of the *tactics, techniques and procedures* (TTP) of the threat. **Deception solutions should interact with a threat at three levels** (analogous to Stoll's account in Chapter 2) to provide detection and mitigation.

DETECT

Any access to a deception reveals a threat with a high degree of confidence. The deception could be a fake privilege inserted in an endpoint, a fake network share, a honeypot server or any mimicked resource.

Detection requires only a low-interaction deception and cannot identify all the TTPs used. Hence detection is necessary but not sufficient to completely eliminating threats.

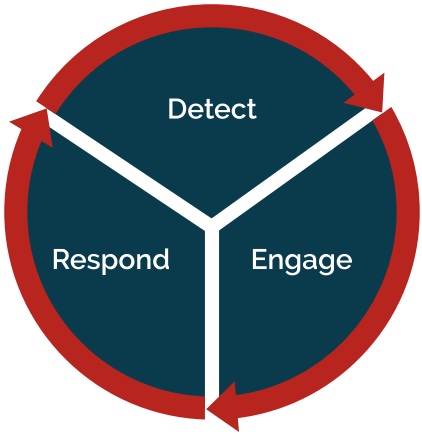
ENGAGE

Once a threat is detected, deception enters the engage phase and starts interacting with the threat to gather information. Typically this requires a high-interaction deception, maybe a server, a router, a software service etc., which is essentially a copy of the production resource.

The engage phase collects detailed threat TTPs, specifically the payload, lateral movement exploits, command and control centers, accounts compromised and the goal of the attack.

RESPOND

Understanding the attacker's techniques and goals helps in both slowing down the attack and remediating all vulnerabilities. This needs automated intelligence to correlate the threat TTPs collected in the engage phase and devise a proper response strategy. A typical response can be shutting down access to the external exfiltration sites.



PERVASIVE DECEPTION

Deception takes many forms to detect and engage the threat at every step of the kill chain. Deceptions can be broadly divided into four types.

DECOYS

A decoy is a fabricated system or a software service that presents an attractive target to the attacker. A honeypot is a type of decoy. Other decoys can be routers, printers, a database service etc. A decoy usually is more tempting than the real production network neighbors, with interesting data and known vulnerabilities.

BREADCRUMBS

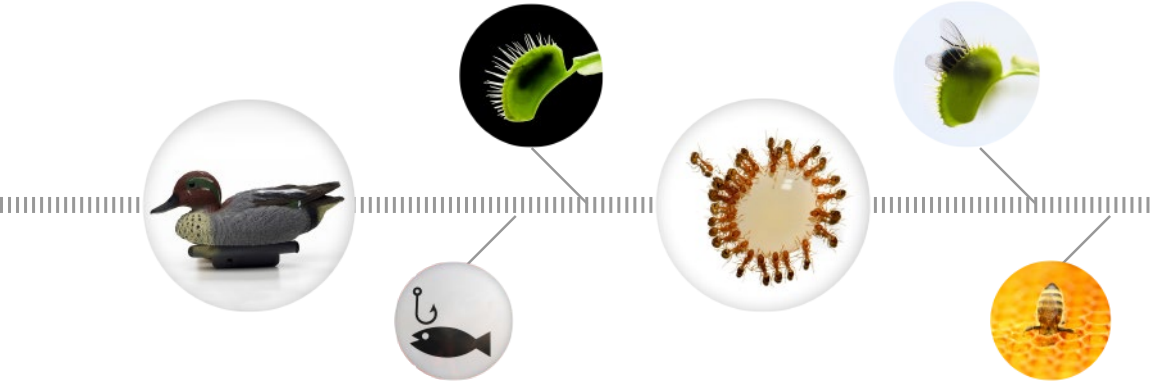
Breadcrumbs are used to lead an attack to a decoy. These are important since the initial compromise is usually an enterprise endpoint. When an attacker does a reconnaissance, breadcrumbs on the endpoints and in the network point to decoys as interesting targets.

BAITS

Baits are honey tokens, for example counterfeit data or fake credentials to a service, which the attacker finds worthwhile to steal. Baits are laid carefully so that ordinary IT procedures or normal user behavior do not touch them. An attack can be detected by monitoring the access or usage of the bait.

LURES

A lure makes a decoy, a breadcrumb, or a bait more attractive than the actual enterprise network assets. For example, to make a software service decoy attractive, it can be set with factory default credentials. A file used as bait may contain fabricated enterprise financial information.



Pervasive deception is required at every stage of the kill chain in order to disrupt the attacker.

The list below provides a few examples of deceptions to disrupt the kill-chain.

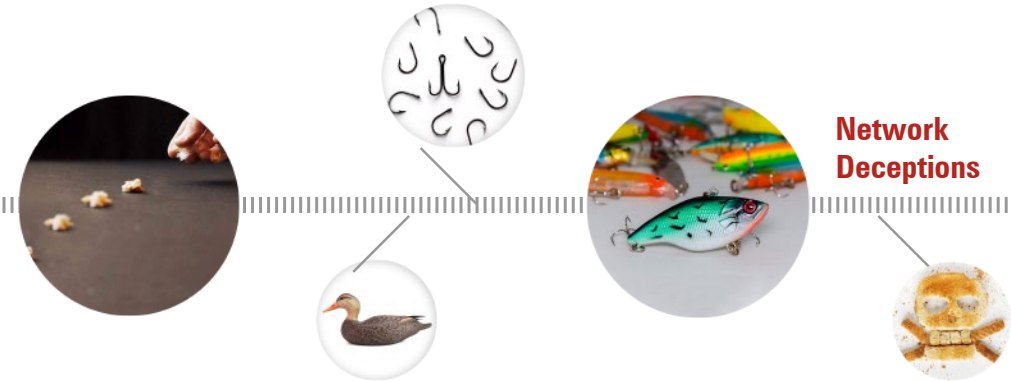
ESCALATE PRIVILEGES – counterfeit user credentials in OS caches (bait)

INTERNAL RECON – FTP/RDP/SSH links & credentials that direct to honeypots (breadcrumb)

MOVE LATERALLY – fake network shares (decoy)

MOVE LATERALLY – honeypots (decoy)

COMPLETE MISSION (Data Exfiltration) – honey data files in honeypots (bait)



NEVER attempt to **WIN BY FORCE** what can be **WON** by **DECEPTION.**

— Machiavelli

4 DECEPTION 2.0: LONG LIVE HONEYPOTS!

IN THIS CHAPTER:

1. **Introduce Deception 2.0 to alleviate the challenges involved in First Generation Deception solutions**
2. **"Fluid Deception" and "DevOps for Deception" form the core concepts of Deception 2.0**
3. **Concept of "Projection Points" and "Deception Farms" to enable Deception at scale and mitigate the risk of compromise**
4. **Understand the nature of the attack through engagement and analysis**

The first iteration of deception technologies, even with their limitations, showed tremendous potential in cyber-defense. Deception 2.0 addresses the limitations of the first iteration and extends deception beyond honeypots into enterprise-scale pervasive deception.

Deception 2.0 is about the industrialization of deception technology. Industrialization of any technology requires two key ingredients:

SCALE – ability to deploy thousands of deceptions.

EASE OF USE – both in configuring and managing thousands of deceptions.

We begin by looking at **three important innovations** that underlie the industrialization of deception. Next we address the drawbacks of using emulations in low-interaction decoys.

We move on to cover how threat analysis is done using a high-interaction decoy, and cover how deception is part of the security ecosystem, both leveraging the data from and providing intelligence to the other layers. The last section recaps Deception 2.0 by showing how the limitations identified in Deception 1.0 are addressed.

FLUID DECEPTION

Industrialization of deception technology requires deploying deception across networks with potentially thousands of machines. Low-interaction decoys are the only solution to achieve cost-effective enterprise-scale. However, as discussed in Chapter 3, both "engage" and "respond" phases require **high interaction** to identify and neutralize the attacks' *tactics, techniques and procedures* (TTP). Acalvio's patented solution, called *"fluid deception"*, combines scale and depth by gradually escalating the level of interaction of a deception dynamically as needed.

We will use honeypot decoys as an example to explain fluid deception. The solution uses a scaled-out deployment of a multitude of low-interaction honeypots, backed by only a few high-interaction ones. During any scanning or reconnaissance phases, honeypots stay in low-interaction mode, and send back appropriate responses to scans. But when an attacker tries to interact more meaningfully — via Secure Shell (SSH) or Remote Desktop Protocol (RDP), for example — with a specific low-interaction honeypot, only that honeypot automatically is morphed into a full-blown high-interaction honeypot running the real OS and services the intruder expects to find. This happens in milli-seconds, while preserving the characteristics of the original low-interaction honeypot, including hostname, IP address, open ports and services.

The above example shows only two modes of interaction, but fluid deception is a powerful idea that is also applicable for other types of decoys and provides variable-interaction. We will use a database decoy to illustrate variable interaction. In low-interaction mode, the database decoy provides an appropriate interface and collects only the login details. If the attack is deemed serious (may be because the provided user id and password are valid), the interactivity is increased and the attack is allowed login access. The database decoy will show appropriate fabricated tables with fake statistics for millions of rows to determine what the attack is aiming for and records any exploits used. If the attack tries to exfiltrate data, further interactivity is provided, where rows for the accessed tables are generated on the fly to keep the attack engaged. By varying interaction only as needed, fluid deception optimizes resource utilization.

DEVOPS FOR DECEPTION

DevOps is defined as "building, managing and operating rapidly-changing resilient systems at scale". Industrialized deception requires a devops approach to configuring, deploying and managing deceptions. An enterprise network evolves continuously. Threat scenarios keep changing quite frequently. Deception 2.0 analyzes changing ecosystem and adapts dynamically.

Deceptions that don't blend into the target environment are detected easily. Change is a constant in an enterprise network. For example, in an endpoint subnet, machines get connected and disconnected frequently. The number of connected machines also changes based on the time of the day or the day of the week. **Staleness is the enemy of deception in such environments.** Deceptions follow the same patterns as the real machines to be effective.

Deception 2.0 also understands the context in which it runs, for example the differences between healthcare and financial systems, and knows that a Windows 8.1 system looks different from a Windows 10 system. The deceptions conform to and change with the nature of the real machines in the environment.

Threat intelligence provides another input to second-generation deception technology. As the cyber attacker methods and exploits change, the deployed deceptions must keep changing as well. If the latest malware targets a specific service or exploits a particular vulnerability, there should be more deceptions providing that service or displaying that vulnerability. All these considerations form the basis of devops.

DevOps is enabled by combining machine intelligence with domain expertise. In the next chapter, we discuss how data science enables devops for deception.

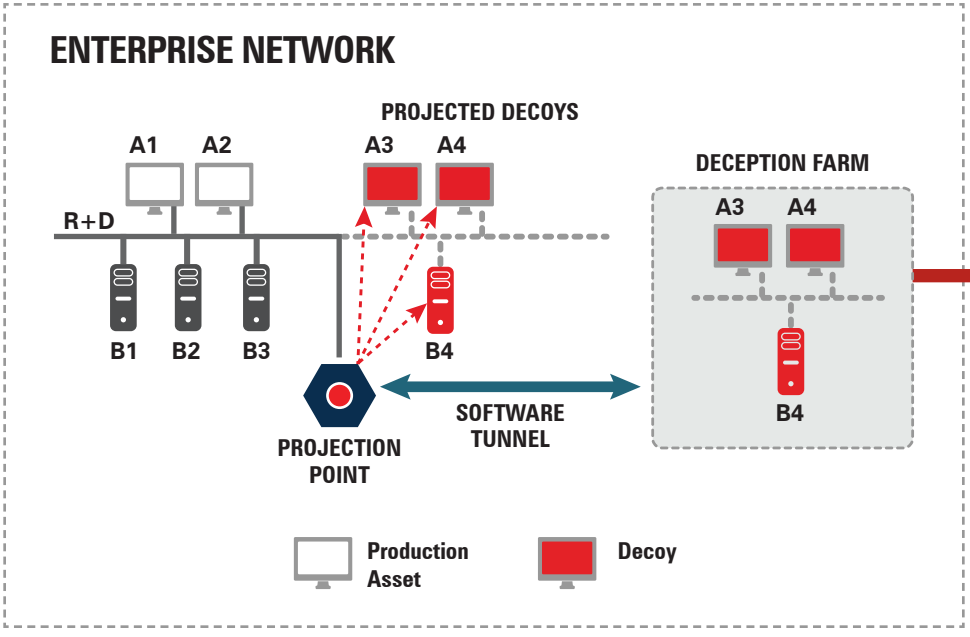
Fluid deception relies on technology that responds intuitively to the attack methods and exploits, changing deceptions from 'Low' to 'High' as needed.

DECECEPTION FARMS

A high-interaction decoy is meant to engage and study the attacker by allowing unrestricted access to the decoy. The decoy may also be easier to exploit, may be set with known vulnerabilities or factory default settings, than the neighboring production assets to attract the attacker. However, once the attacker takes control of a decoy, he may use the decoy as a base and launch attacks against regular production assets. To avoid this risk, the decoys must be separated from the production network. But the decoys also need to be intertwined with and be part of the production network to blend. Deception Farms is a way to satisfy both contradictory requirements.

The basic concept underlying deception farms was described for honeypots[11] by Lance Spitzner. His idea was to move all the high-interaction honeypots from the customer network into a separate consolidated location. The same idea is extended in Deception 2.0 to all types of decoys, and to both low-interaction and high-interaction.

Deception farms have two parts – projection points and a centralized deception farm with low and high interaction decoys. A projection point sits in the enterprise network and is essentially a "worm hole" to the deception farm. The projection point builds a software tunnel to the deception farm and, using the advancements in software defined networks (SDN), provides logical layer 2 adjacency to the enterprise network for deceptions in the central deception farm. All the decoys are deployed in the deception farms, but appear to be adjacent to the resources in multiple subnets in the enterprise network through the projection points (see illustration).



CUSTOMIZED LOW-INTERACTION SERVICES

As discussed in Chapter 2, building and customizing emulations in low-interaction decoys to match all the services in a given enterprise is a non-trivial task. As a result, emulations fail to provide coverage for all services and also are easy to fingerprint.

Fluid deception manages both low and high interaction decoys. The low-interaction decoys take advantage of the full services available in the high-interaction decoys to provide bespoke low-interaction services. The deceptions completely avoid emulations and project services identical to the customized services in the enterprise network. Since actual services are leveraged, the interaction is not just low or high, but can vary on a continuous scale based on the attack. For example, by leveraging an actual ftp service, the deception can stop the interaction immediately after login, continue until a payload is detected or engage the attack with attractive data for a long duration.

An attacker on a network cannot distinguish whether a neighbor is physically or logically adjacent.

Any attack on one of these projected decoys is moved through the tunnel to the deception farm in the cloud. The tunnel is "invisible" to the attacker and is used to monitor traffic generated by the attack. Network traffic through the tunnel is controlled to ensure that any attack or malware is allowed only one way from the enterprise network to the deception farm, but blocked if it tries to move the other direction. This ensures that any compromise stays in the deception farm, and is disposed off by recycling the decoy.

THREAT ENGAGEMENT & ANALYSIS

Threat engagement is part of the "engage" phase and collects all attack artifacts. Threat analysis extracts intelligence from the collected artifacts and provides the input to the "respond" phase. We will again use a honeypot to explain the process, but this applies to any high-interactive decoy.

A high-interaction honeypot is different from a sandbox, both in requirements and functionality. A honeypot is an interactive system, designed for both malware and threat actors, while a sandbox is primarily for malware detonation. Both honeypots and sandboxes collect detailed attack information. However a honeypot is part of a networked environment and is uniquely positioned to study how the attack plays out beyond the system, for example interactions with DNS, DHCP servers or Active Directory.

A HIGH-INTERACTION HONEYPOT MUST SATISFY THE FOLLOWING CONDITIONS TO ENGAGE IN EFFECTIVE THREAT ENGAGEMENT.

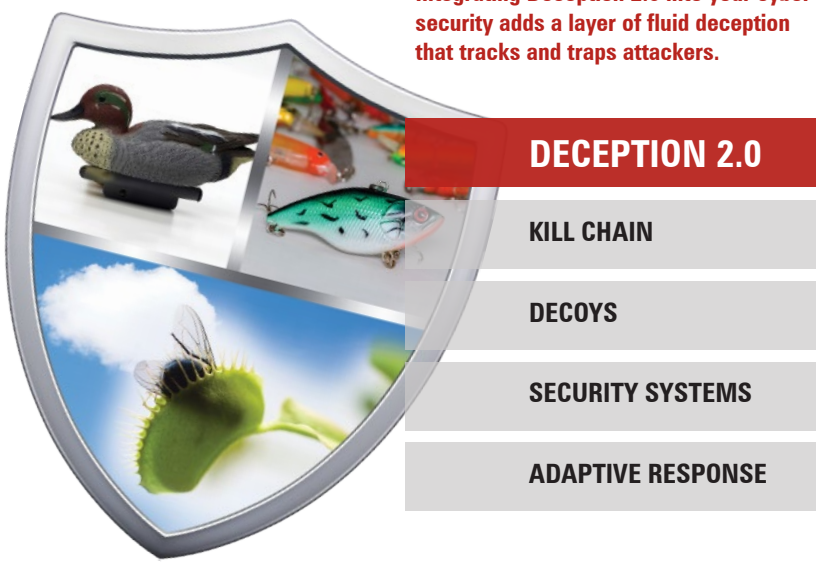
1. The honeypot should have multiple decoys, breadcrumbs and baits installed to determine the exploits used and lateral movement paths. Lateral movement should lead to other decoys.
2. The collection mechanism should be invisible to the attacker.
3. Network traffic must be collected and correlated to the attacker's actions. This helps determine command and control center, exfiltration activity etc.
4. It should record all attacker actions, for example through instrumented binaries, screen capture, keyboard logger etc.
5. It should be able to review Memory dumps to determine any files not saved to disk.
6. It should track downloaded or dropped files.

Threat analysis is done in real-time as the threat artifacts are collected. Analysis figures out the malicious actions being carried out by the attacker and also the goal of the attack (exfiltrate data, encrypt data, botnet etc.). Once the attack is analyzed, data science (covered in next chapter) determines the appropriate response.

LAYERED DEFENSE

Layered defense is the practice of combining multiple mitigating security controls to protect resources and data. Given the "kill-chain" model, a combination of security systems, each using different techniques, is required to deal with persistent threats. Layering security systems allows vulnerabilities in one system being covered by another. Splunk's adaptive response initiative[10], of which Acalvio is a part, is one of the first attempts to integrate the security systems.

Deception 2.0 uses a completely new technique to provide a new layer for cyber-defense. Threat intelligence, both external and internal from other security systems, provides input to the devops of deception. The threats could be an anomaly detected by an IDS (intrusion detection system) or an alert raised by a database firewall on failed connection attempts. Leveraging the devops model, Deception 2.0 can quickly deploy relevant additional deceptions around the alert to lure the attack. Once an attack is engaged and studied, the analysis can be shared with the other systems. For example, an IOC (indicator of compromise) can be generated for the command and control center used by the attack and shared with the perimeter firewall to automatically stop all exfiltration immediately..



Integrating Deception 2.0 into your cyber security adds a layer of fluid deception that tracks and traps attackers.

REVISITING "LIMITATIONS AND EXCLUSIONS"

Below we recap the challenges encountered in Deception 1.0 and how they are addressed in Deception 2.0.

HOW THE CHALLENGES OF DECEPTION 1.0 ARE ADDRESSED IN DECEPTION 2.0:

LOW-INTERACTION VS HIGH-INTERACTION:
Fluid Deception solves this by combining the best of both and provides a range of interactivity from low to high.

SIMPLICITY OF LOW-INTERACTION DECEPTIONS:
Customized low-interaction services avoid emulations to provide complete coverage and resist fingerprinting.

SCALE OF DEPLOYMENT: Also addressed by fluid deception.

FINGERPRINTING OF HONEYPOTS: DevOps model deploys dynamic deceptions to avoid staleness, which in turn reduces fingerprinting risk. Also, the use of real services instead of emulations prevents signature based fingerprinting.

EASE OF USE: DevOps model deploys and maintains deceptions automatically.

RISK OF COMPROMISE: Use of deception farms and projection points mitigates the risk of compromise.

HONEYPOTS ARE THE ONLY DECEPTION TYPE:
Deceptions now mimic all enterprise resources.

STAND-ALONE PRODUCT: Deception 2.0 actively gets input from and provides intelligence to the security ecosystem.

THREAT ANALYSIS: Threat engagement and analysis is an integral part of Deception 2.0.

The supreme art of war is to **SUBDUE THE ENEMY** without **FIGHTING.**

— Sun Tzu

Oh what a **TANGLED WEB** we weave when first we practice to **DECEIVE.**

— Walter Scott

5 ADDING INTELLIGENCE TO DECEPTION: THE ROLE OF DATA SCIENCE

IN THIS CHAPTER:

1. **Understand role of data science in Deception 2.0**
2. **A brief look at the history of security data science**
3. **How data science guides "Fluid Deception" and "DevOps for Deception"**
4. **How data science can leverage deception alerts to respond to attacks**

Data Science is an inter-disciplinary field that seeks to extract insight from data. Data can often tell what is happening and data science is the language to understand it. Data Science seeks to extract insight from data in all its many forms, both structured (as found in a database or other orderly collection of records) and unstructured (as found in emails, documents, or other content). It combines tools and analytical approaches from many fields of data analysis, including statistics, machine learning, data mining, deep learning, predictive analytics, and more. The first section provides an overview of how the role of data science in cyber security has evolved.

COMBINING DATA SCIENCE WITH DECEPTION

Data science provides automated intelligence to deception. The building blocks of Deception 2.0, "fluid deception" and "DevOps for deception", are guided by data science (illustration). We then explore how data science is leveraged for building enterprise-scale deception.

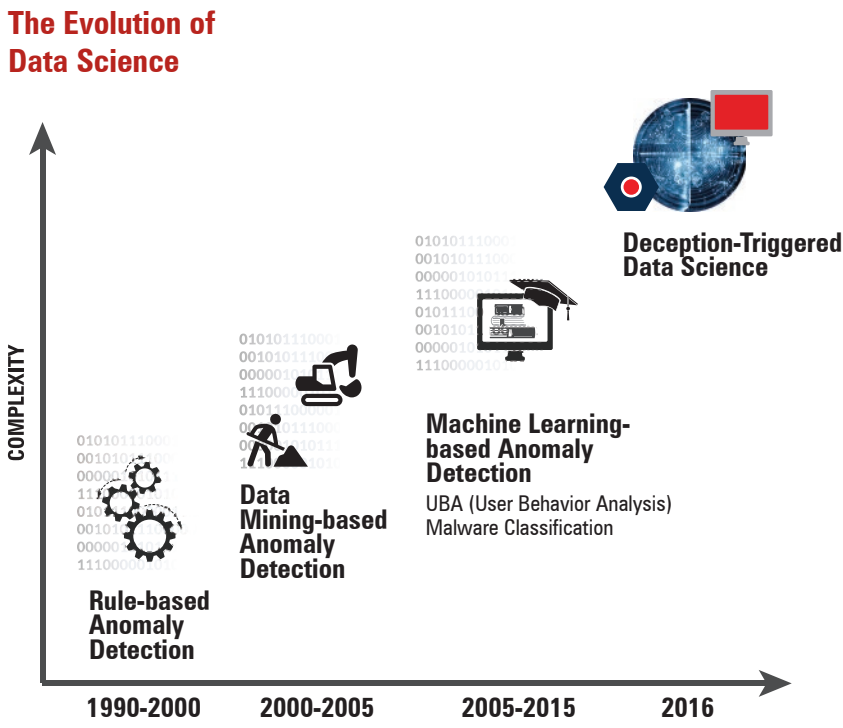
Threat engagement and analysis (Chapter 3) can be used to build a detailed profile of the adversary behavior. *Adversary Behavior Analytics* (ABA)™ analyzes this profile to build an effective response to the attack. ABA is computed by deception-triggered data science. The deception-triggered data science is a novel concept that is different from traditional security data science and brings more insights about the adversary behavior by correlating deception alerts with other security events in *Security Information and Event Management* (SIEM) database. A detailed explanation of deception-triggered data science is provided later in this chapter.

Combining Deception and Data Science

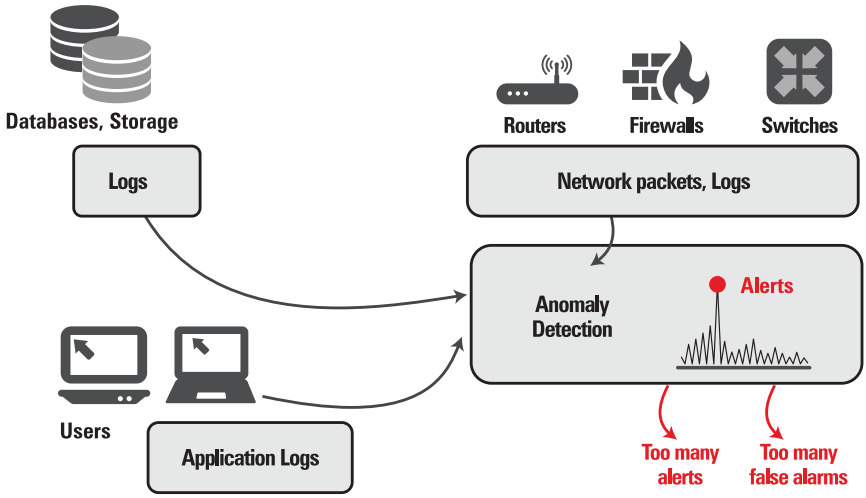
SECURITY DATA SCIENCE

Data Science draws upon techniques, algorithms and theories from multiple academic disciplines. These comprise an extensive list that embraces statistics, machine learning, data mining, mathematics and optimization, plus a great many subfields from the information and computer sciences. Ultimately, data science is about examining and analyzing data and gaining insight and intelligence from the data. This could mean discovery of interesting trends or new emerging patterns or phenomenon. It can involve using visualizations to aid understanding, or creating powerful mathematical models to explain all kinds of things.

Since the 1990s, data science has played an increasingly important role in information security. This started with rules-based approaches to finding anomalies in intrusion detection systems (IDSs) and intrusion prevention systems (IPSs). The Evolution of Data Science illustration shows how data science has gained importance and momentum in information security.



Traditional Security Data Science Using Anomaly Detection



In the first decade of the 21st century, IDSs and IPSs evolved to incorporate control charts and data-mining-based anomaly detection. Later iterations saw complex systems built to gather and store vast amounts of log and event data in the SIEM database. User and entity behavior analytics (UEBA) employed machine learning algorithms which distill the SIEM data to profile and baseline every user and network element in the enterprise IT environment. Any significant deviations from the baselines were triggered as anomalies that further raised alerts to be investigated by the security analysts (illustration). UEBA enhanced the detection of insider threats, albeit to a limited extent. However, the anomaly-based solutions have the drawback of generating a large number of false-positive alerts. Each investigation of a false-positive alert adds significant burden to an already over-loaded security team.

DECEPTION GUIDED BY DATA SCIENCE

Industrialized deception requires intelligence to configure, deploy and manage thousands of deceptions. The deception should change as the enterprise environment shifts, and as the threat landscape evolves. Effective deception needs to be customized to blend into the enterprise IT environment and lure attackers. Data science provides the intelligence to automate the deception deployment in an enterprise such that it blends within the enterprise IT infrastructure.

Data science leverages the security ecosystem to get various network data as input, including network traffic patterns (e.g., a density of the network traffic, whether any of the network traffic is encrypted, source and destination addresses, etc.), the value of assets such as hardware resources, data, and so on in the network, previous attack patterns, and current alerts from network security devices, among others and build a deception recommendation model. The model can be used to determine the number, position, and configuration of deceptions to deploy. The recommendation model may be revised periodically based on new or modified inputs and the effectiveness of the previous deployment strategies. A couple of examples:

1. **If the neighborhood has a mix of services and operating systems installed, the honeypots will show the same mix.**
2. **If threat intelligence warns of new attacks on SQL server databases, more fabricated SQL Server database services are deployed.**

Data science is also used by fluid deception to decide when to escalate and de-escalate the interactivity of a decoy. Escalation triggers may depend on the service accessed, existing threat intelligence, etc., and keep changing with the environment. By building intelligence into deception escalation, data science helps to minimize the overhead of high-interaction decoys.

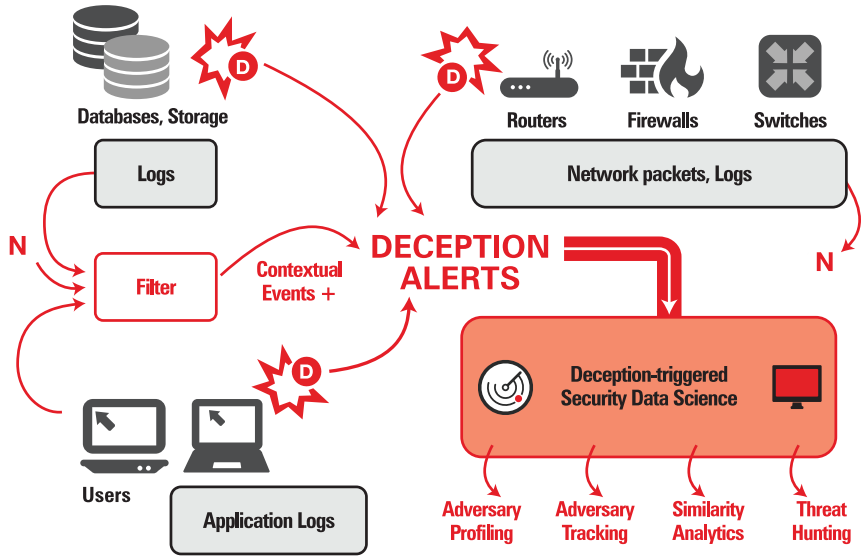
DECEPTION-TRIGGERED DATA SCIENCE

Deception-triggered data science is significantly different from conventional security data science. Conventional security data science primarily leverages anomaly detection techniques to identify anomalous behavior in network traffic, user or host or network element behavior. In sharp contrast, deception-triggered data science starts with a real attack, i.e. anomaly announced by the deception event. Consequently, it does not require expensive and elaborate anomaly detection algorithms.

Deception alerts are high fidelity alerts and data science correlates other security event data with these high fidelity alerts to generate a lot of insights about the adversary behavior. In this approach we collect and describe the context around a deception alert instead of looking for anomalies like needles in haystacks. Instead, this kind of data science can focus on capturing everything about how an attack begins and proceeds as it progresses.

To draw on a metaphor, deception-triggered data science to brute force security data science; is boiling a cup of tea instead of boiling the ocean – the former is practical, clever and elegant; the latter is expensive, impractical and impossible. Deception triggered data science significantly reduces the false positives thereby reducing the overall infrastructure and maintenance cost associated with security related chores. Analysis of attacker behavior also lets security data science make recommendations on the types of deception to offer intruders, and can use statistical analysis to help determine the number of deceptions to include in an enterprise environment. Moreover, based on the deception triggered, this can be dynamically modified either to obfuscate the attacker or alienate him away from the production environment.

Deception-triggered Data Science



In a deception-based environment, analysis starts after deception sensors get tripped. Deception-triggered alerts is then used to correlate with alerts from other security data sources, such as Network Flows, User Authentication, Asset Data, SIEM, etc. to generate improved threat intelligence. Such deception context data provides insights into adversary behavior, movements, and tactics.

Deception-triggered security science enables multitudes of use cases:

ADVERSARY PROFILING: Adversary profiling provides details about intruders based on deception-context data, along with hosts visited, vulnerabilities exploited, and tools or malware used. Based on this information analyst can weigh different response scenarios.

ADVERSARY TRACKING: Adversary tracking correlates deception event data with historical NetFlow data or user authentication data, to construct the probable path that led the intruder to a deception sensor. Armed with this knowledge, analyst can trace the attacker's path through the system. He/she gets insights about the vulnerable access points and paths which may otherwise go undetected.

SIMILARITY ANALYTICS: Similarity analytics permits hosts similar to those compromised to be identified and then remediated. Analyst can see similarity at various dimensions enabling him to get a snapshot of possible compromised machines and many more.

THREAT HUNTING: By correlating deception alerts with IDS alerts, Firewall alerts, Vulnerabilities one can build InfoSec Playbooks that can assist security analysts to hunt for threats in their enterprise environment.

In summary, data science creates opportunities to seek out patterns in deception events, and to anticipate future adversaries or patterns of compromise. Moreover, it helps in seamless deployment and operational efficiency which gives security professional a significant edge over the threat actors.

DECEPTION is EVERYWHERE.

— James Sanborn

6 DECEPTION IN THE CLOUD

IN THIS CHAPTER:

1. **Cloud-based deception farms**
2. **Deception farms for hybrid cloud networks**

Cloud computing has become ubiquitous, with enterprises increasingly adopting both private and public cloud solutions. Deception 2.0 consolidates Deception Farms in the Cloud (Chapter 4), thus streamlining the operations overhead involved in the deployment. We start the chapter by describing this architecture and the advantages of cloud-based deception farms.

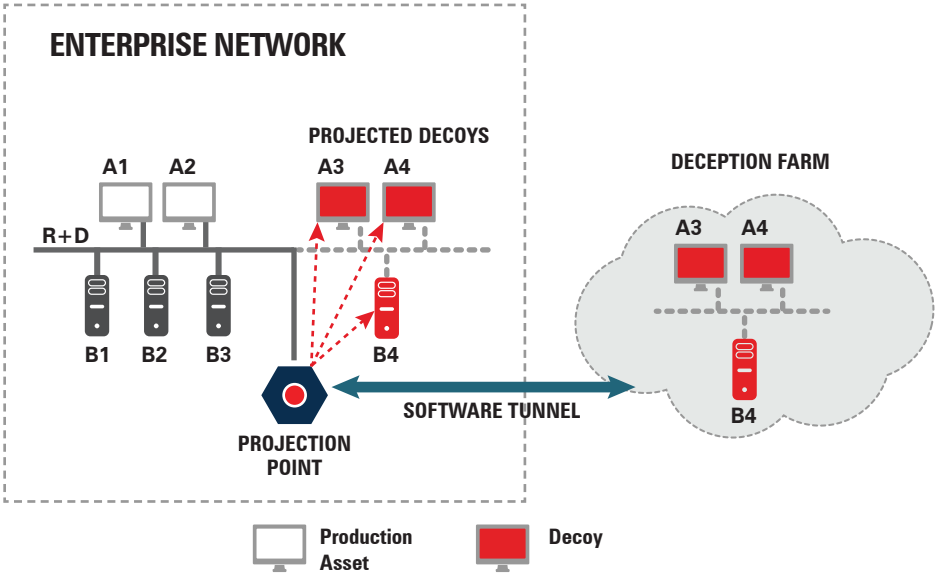
A number of large enterprises will migrate some or all of their compute load to public clouds (AWS, Azure, Google Cloud, etc.) in the near future. Public clouds come with their own unique vulnerabilities. Hybrid clouds, which combine both on-premises and cloud services, are also prevalent and have additional points of susceptibility based on the connectivity between the two component networks. The second section in this chapter discusses how deception can be extended to enterprise networks based either completely or partially in the cloud.

CLOUD-BASED DECEPTION FARMS

Deception farms provide the ability to consolidate all low-interaction and high-interaction decoys in a single location (as described in Chapter 4). Projection points, deployed in the on-premises enterprise networks, build software tunnels to the central deception farms and provide logical layer 2 adjacency to the enterprise network for deceptions in the central deception farm.

The centralized deception farms concept is ideally suited to be based in cloud (illustration). By moving the deception farms to the cloud, the risk of compromise is further reduced. The vulnerable decoys are now separate from the enterprise network and any compromise can be cordoned off in the cloud. Cloud-based decoys are also not physically accessible in the enterprise, providing an additional level of security.

Deception Farms



Cloud-based deception farms have other significant advantages compared to on-premise deception farms

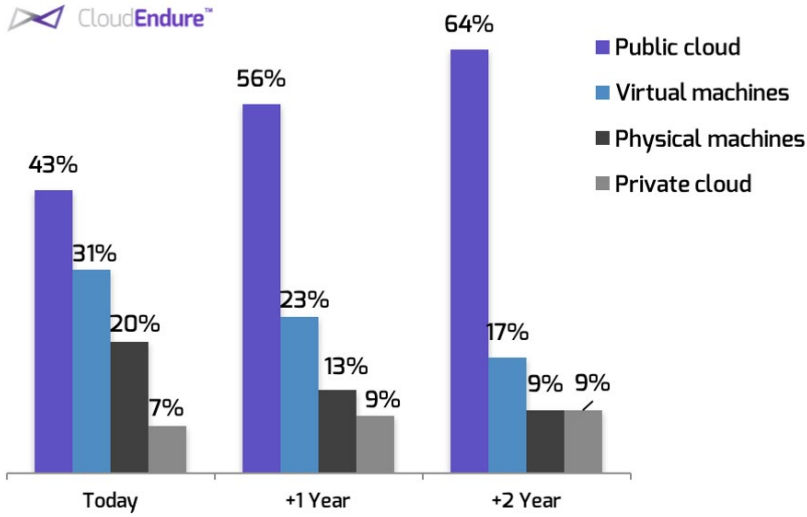
Providing scale in a cloud-based solution is much easier. Cloud is elastic and hence the deception farms can scale up/down as the need arises.

Cloud-based deception farms are cheaper. Fluid deception uses a resource intensive high-interaction decoy only when needed. But the costs for on-premises deception farms is fixed since it has to be deployed in an appliance. In cloud-based deception farms, costs are variable and are incurred only when the high-interaction decoy is deployed.

DECEPTION FOR NETWORKS IN CLOUD

Public clouds are becoming popular due to scalability, cost effectiveness and reliability. According to CloudEndure 2016 Report[14], more than half of the large enterprises plan to migrate to public clouds within the next two years (see chart), while the share of on-premises virtual and physical machines is set to decrease.

Share of Public Clouds Is Set to Increase



Many enterprises also have a hybrid cloud infrastructure, with a part of the network on-premises and another part on the cloud. An enterprise-scale deception solution should provide a uniform interface for both on-premises and cloud networks, make deception easy to install and manage on both and scale equally well across both. Deception farms and projection points, introduced in the previous section, make this possible. Projection points can be installed in both on-premises and public cloud networks. The centralized deception farms, which may be located on-premises or on the same/different cloud, projects customized deceptions through the projection points based on network segments. The architectural and deployment flexibility of Deception 2.0 make extending deception to hybrid workloads possible.

Public cloud networks have their own peculiarities that can be exploited. The connectivity between on-premise and cloud networks also introduces additional vulnerabilities. The Microsoft Security Intelligence Report[15] details multiple attack methods unique to public clouds. The pervasive deception (Chapter 3) strategy of Deception 2.0 includes deception to detect and engage these new vectors of compromise.

The secret access keys to the cloud network are the target of many attacks. Source code repositories or hacked emails may contain the secret keys. Deception can help detect these attacks by strategically placing fabricated secret access keys in places likely to be targeted by the attackers.

A persistent brute force attack on public-facing endpoints in the cloud can succeed. Pivot-back attacks start with compromised servers in the public cloud and laterally move into the on-premises network. For cloud networks with public-facing endpoints, research honeypots (Chapter 2) can help with generating threat intelligence specific to that cloud network. By engaging with the attacker, a research honeypot can provide information on the tactics, techniques and procedures of the attackers even before they manage to break into the network.

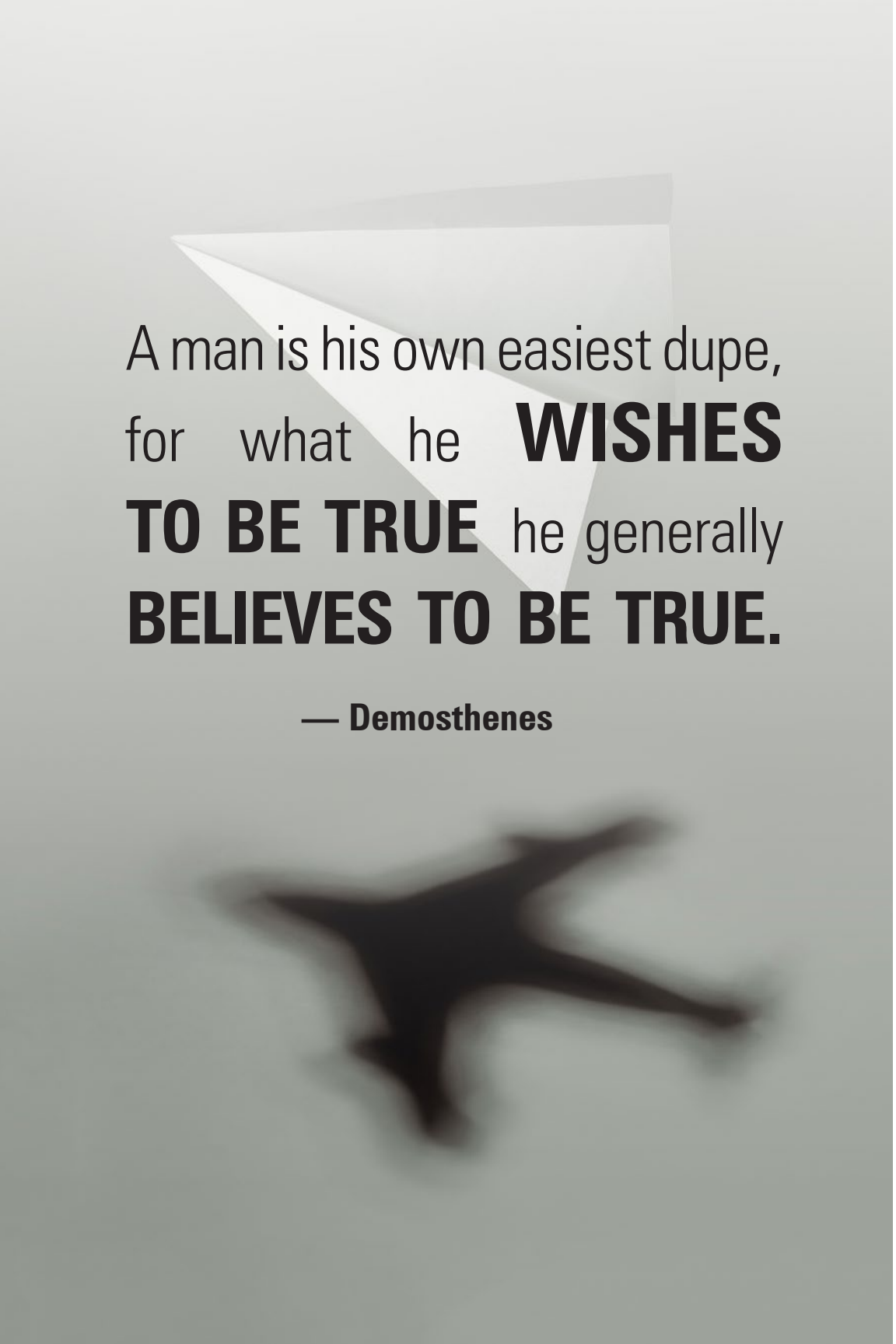
In an on-premises network, the lateral movement is from machine to machine – an attacker compromises a machine, escalates privileges, does a recon and moves to the next attractive machine. In the cloud, the movement may also be from service to service[15]. The attacker compromises a service, does a recon for credentials for other services and uses them to move laterally to the next service. **A Deception 2.0 solution, with decoy services and breadcrumbs, can detect the lateral movement in such situations.**

Cloud computing is going through significant improvements and changes on a regular basis.

The Deception 2.0 approach of using deception farms and projection points helps address future advances in cloud computing. For example, one of the emerging trends is the use of software containers in cloud deployments. All the major public cloud providers like Amazon, Microsoft and Google already provide software container services. Centralized deception farms allows easy addition of container decoys to the deception farms and automatically have them projected across any network segment.

A man is his own easiest dupe, for what he **WISHES TO BE TRUE** he generally **BELIEVES TO BE TRUE.**

— Demosthenes



7 EXTENDING DECEPTION TO THE INTERNET OF THINGS

IN THIS CHAPTER:

1. **Introduction to cyber-threats in Internet of Things and Industrial Control Systems**
2. **Security challenges specific to Industrial Control Systems and how deception addresses them**
3. **Deception 2.0 applied to Internet of Things**

Internet of Things (IoT) is a network of smart devices in seamless interaction with themselves and with people. Any device, a coffee maker, a door lock or a jet engine, becomes smart if it has the hardware, software and network connectivity to collect and exchange data. A smart device has one or more sensors (to collect data) and may, optionally, have actuators (to act on a signal). The smart devices are linked through wired or wireless networks, and often use the existing internet infrastructure and the Internet Protocol (IP).

In the first section, we first trace the development of connected devices through Industrial Control Systems (ICS). We discuss the potential for immense harm through cyber-attacks in critical infrastructure and how Deception 2.0 is one of the few applicable defenses. The second section details the security issues in the vast, and still developing, IoT market and the use of the second generation deception technology to detect and study attacks

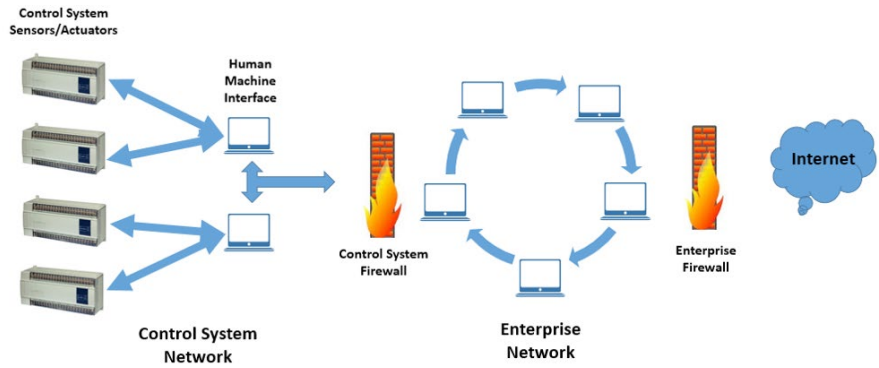
INDUSTRIAL CONTROL SYSTEMS

Industrial Control Systems (ICS) are used to monitor and control critical infrastructure components in industrial processes. ICS systems have been implemented in industries ranging from chemical factories, power grids, oil and gas pipelines to sewage management.

An ICS system is composed of essentially two parts:

1. **Remote Terminal Units (RTU):** RTUs are usually Programmable Logic Controllers (PLC), with sensors and actuators. They also have communication ports to allow them to receive and send signals (data and commands).
2. **Human-Machine Interface (HMI):** The RTUs are connected to operator workstations, called HMI, and provide continuous, real-time information on the process state. HMI is used by the operators to monitor and control RTUs.

Two-Firewall Control System Architecture



The ICS systems used to be air-gapped (not physically connected) to the enterprise network. As a result, the ICS systems were built for reliability and safety, but not for security. Over time, ICS systems were internet-enabled for integrating ICS data into other functionality in the enterprise network (for example, production data can automatically update inventories). Standardizing on the Internet Protocol (IP) simplified the networking of ICS components. The illustration shows a typical two-firewall control system architecture, where the control system network is protected by a second firewall from the enterprise network.

The perils of unsecured critical infrastructure control systems was demonstrated by the Idaho National Laboratory in the *Aurora Generator Test* in 2007[12]. The experimental cyber-attack caused a 2.25 MW generator to self-destruct in less than three minutes. The attack used a computer program to rapidly open and close the diesel generator's circuit breakers out of phase from the rest of the grid and caused it to explode.

ICS systems suffer from significant challenges not seen in enterprise networks.

1. RTUs run on small processors with limited computing power, which precludes running any antivirus software.
2. A significant percentage of the RTUs are old and are rarely patched for vulnerabilities. The patches may not be available or may make the system unstable.
3. Inline network security systems may not work for ICS systems as false positives, or even delays in transmission, result in loss of view or control, which have the potential to cause harm to life or environment in critical infrastructure services.
4. Many ICS system protocols are not built for security. For example, Modbus is a communications protocol used in most ICS systems. First designed in 1979, this protocol has no security features. If an attacker can reach a device with Modbus protocol, he can read and write anything on that device.

The challenges above limit the security systems that can be deployed in an ICS environment. Deception technology is one of the few that satisfy the requirements. It is not inline and does not affect the regular control system processes. Deception technology also generates few false positives.

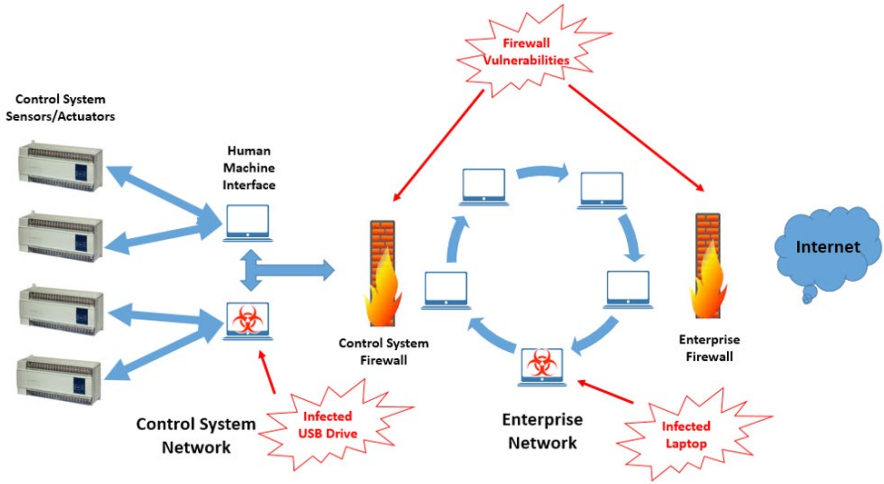
Early honeypot experiments listed in (Chapter 2), Gaspot and Conpot, deal exclusively with ICS systems. Conpot simulated a basic Siemens PLC, which is used in many critical infrastructure in-dustries, to support Modbus and SNMP protocols for ICS. Gaspot concentrated on a non-critical infrastructure (gas tanks). These first generation deception solutions, even with simple emula-tions, demonstrated the interest in the ICS systems among the hacker community and the effec-tiveness of deception.

INDUSTRIAL CONTROL SYSTEMS (CONTINUED)

Deception 2.0 solution, in an ICS environment, considers multiple attack entry points and automatically deploys deception throughout the network. The illustration below shows a few possible entry points in a two-firewall control system architecture. For example, the infamous Stuxnet worm used infected USB drives as the entry point to attack an air-gapped Iranian nuclear facility. Deceptions should be in the enterprise network, human-machine interfaces (HMIs) and the PLC controls.

Deception in the enterprise network shows the existence of multiple counterfeit control system firewalls using well-placed breadcrumbs in the servers in the network. Any access to a counterfeit firewall detects an attack. In addition, fabricated HMIs, with fake traffic, blend in with the real HMIs. The fabricated HMIs connect to realistic PLC emulators and display continuous process state. The real control system network is also populated by many low-interaction PLC emulations. Fluid deception is used to engage any access to one of the low-interaction decoys by dynamically morphing it into a high-interaction decoy. Since the PLC systems are relatively cheap and each one implements the network protocols a little differently, a few real PLC systems (not connected to any infrastructure) are used as the high-interaction decoys. Pervasive deception enables the attacks to be detected at each of the entry points.

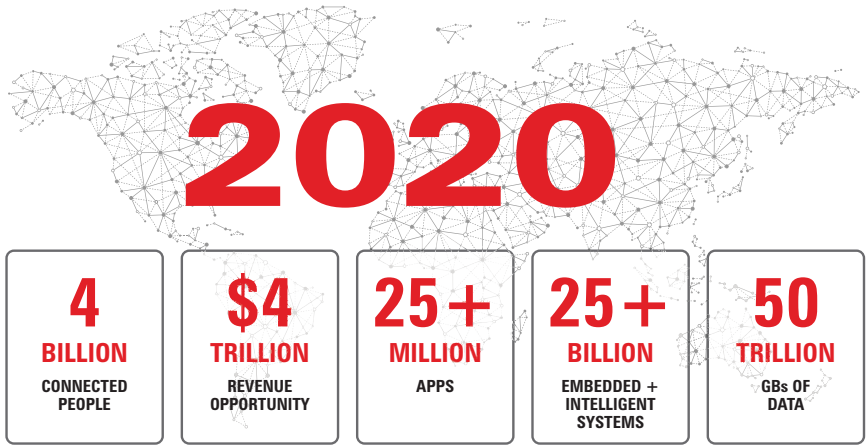
Possible ICS Attack Entry Points



INTERNET OF THINGS

Internet of Things (IoT) is a network of connected smart objects. The sensors and actuators, similar to those in the ICS systems, are now embedded in physical objects – from devices in homes and offices to devices implanted in humans – making them smart and able to communicate. These new information networks are driving a huge and fundamental shift in creating new products and services. Conservative estimates place the number of smart objects at 25+ Billion and foresee a $4 Trillion market opportunity by 2020.

Explosion of IoT



Conservative estimates place the number of smart objects at 25+Billion and foresee a $4 Trillion market opportunity by 2020.

The security challenges that exist in ICS systems are increased manifold in the IoT arena.

1. The IoT devices contain more and more complex and powerful technology. As processors and storage get faster and cheaper, smart objects become more powerful with every iteration.

2. Smart devices are installed everywhere - homes, offices, cars, roads, bodies etc. Many of them are accessible by their IP addresses, without any firewall protection.

3. The devices are built by thousands of manufacturers. The market opportunity is huge and developing, which encourages more companies to participate. For many of these manufacturers, security is an afterthought.

4. The IoT systems, unlike enterprise networks or ICS systems, are usually managed by people who have little background or appreciation of cyber-security.

The combination of these factors makes IoT devices easy to compromise. The Mirai botnet attack in September 2016[13] illustrates the above challenges. Mirai scans the internet for IoT devices that use factory default usernames and passwords. This simple initial compromise enabled Mirai to establish a foothold and infect hundreds of thousands of devices, mainly cameras and DVRs. Using these devices, Mirai launched, and still launches, massive distributed denial of service (DDOS) attacks.

Many of the deception technologies introduced in the previous chapters are required to provide effective security for an IoT network. We use a "smart home" network to describe how Deception 2.0 works in this environment. A smart home begins with smart household devices and appliances. These residential devices, which control heating, lighting, air conditioning etc., and appliances, such as washer/dryers, ovens or refrigerators/freezers, have sensors, actuators and embedded intelligence for automation. They communicate with each other, usually over WiFi, and can be controlled from a central hub.

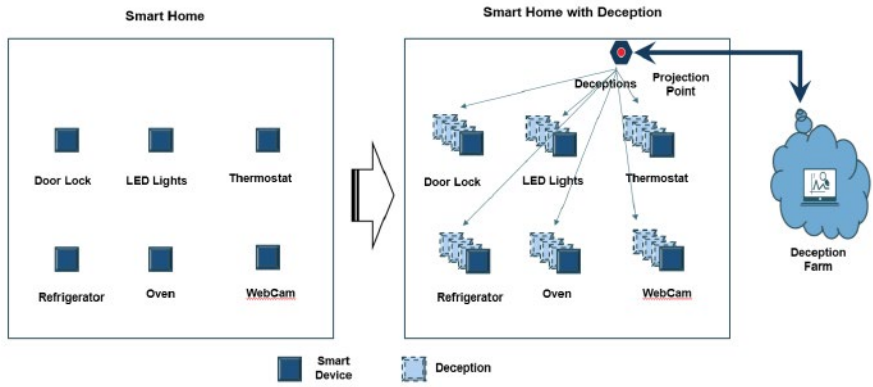
Deception farms and projection points, as described in Chapter 6, are ideal for providing security to the smart home networks. A single projection point can project multiple deceptions of the smart objects and appliances in the house, as the second panel in the illustration shows. The deceptions themselves exist in the deception farms in the cloud and only appear to be in the network neighborhood.

DevOps for deception helps to dynamically create and manage deceptions based on the devices in the smart home and developing threat profiles. If a particular exploit like Mirai is known to exploit cameras with factory defaults, devops mode projects similar cameras in the smart home to detect if the malware has entered the home network.

Fluid deception can escalate any observed attacks on the projections to high-interaction decoys located in the deception farms. For example, a projected camera can be dynamically replaced by a real camera to engage a suspected *Mirai* attack, capture the payload and understand the variations of the exploit. Fluid deception becomes very cost-effective to implement in the context of deception farms, since thousands of projected low-interaction decoys across thousands of residences can be successfully backed by only a few high-interaction decoys.

Internet of Things is "the next big thing" that has already arrived. Powerful computing and storage resources are being interwoven into everyday life. With the advent of nano-technologies, the near future will see smart devices being used to repair and remediate inside the human body. As IoT spreads within the organization, perimeter defense systems' ability to effectively thwart the threat actor becomes increasingly less effective. The concepts and technologies introduced in Deception 2.0 provide one of the few viable and effective cyber-defense solutions.

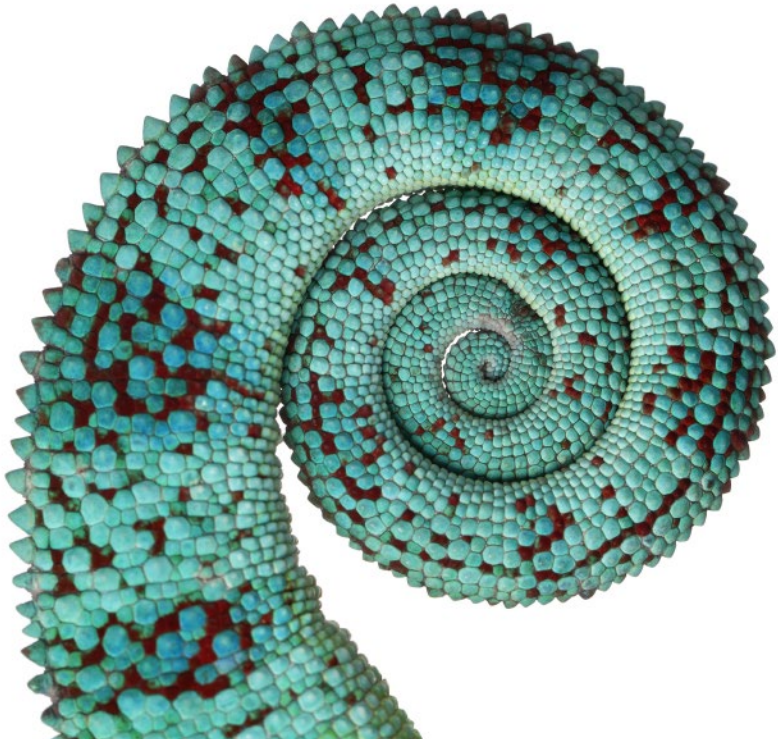
IoT: The Smart Home



The ability of perimeter defenses is rapidly being diminished as IoT spreads.

To know
HOW TO DISGUISE
is the
KNOWLEDGE OF KINGS

— Cardinal Richelieu



8 RECOGNIZING A GOOD DECEPTION SOLUTION

IN THIS CHAPTER:

A persistent cyber-attack usually has very good odds of success – it only has to succeed once in penetrating the perimeter defenses. Deception is unique in that the threat actor needs to be wrong only once before he gets detected. An effective enterprise-scale cyber-deception solution lays carefully crafted decoys to detect and study an attack at every step of the internal recon and lateral movement.

THE FOLLOWING ARE THE TEN ESSENTIAL REQUIREMENTS OF A POTENT CYBER-DECEPTION SOLUTION.

1 Deception must provide both SCALE AND DEPTH.

Enterprise deception solutions should scale up cost-effectively to deploy thousands of decoys, while also providing the ability to engage and respond to the attack.

Solutions that provide only one or the other are based on the first iteration of the deception technologies.

2 Deceptions must be DYNAMIC.

Staleness is the enemy of deception. As the network and threat environments evolve, deception must adapt.

Solutions with static deceptions are easy to fingerprint and are of little value.

3 Deceptions must be PERVASIVE.

Effective deception needs various kinds of decoys, baits and breadcrumbs.

Solutions that are decoy-only (or even worse honeypot-only) or breadcrumbs-only are partial, incomplete and marginally effective solutions.

4 Deceptions must be AUTOMATIC.

An enterprise-scale deception solution needs to lay out a multitude of deceptions and manage them dynamically. Automation of every step is a requirement for practical decep-tion at scale.

Solutions that require manually deploying or managing deceptions do not scale.

5 Deceptions must NOT INTRODUCE NEW RISK.

Deception technologies, by design, introduce vulnerable systems in the enterprise network to lure and engage attacks. A vulnerable system increases the risk of compromise as the threat actor can use this as a pivot point to launch attacks against other systems in the network.

Solutions that physically locate high-interaction decoys in the enterprise network (connected directly to an access port or a trunk port) run the risk of compromise pivoting to the enterprise servers.

6 Deception must BE INTELLIGENT.

Data science is an integral part of an effective deception solution. Machine intelligence is imperative for automation.

Beware of solutions that do not leverage machine intelligence. The effort involved to design, deploy, manage, monitor deceptions and correlate threat data is near untenable without the uncanny leverage of Data Science.

7 Deceptions must BLEND into the ENTERPRISE.

A deception should not look any different from the network neighborhood. This applies to all decoys, baits and breadcrumbs. Requires dynamic deception to keep up with the changes in the network.

Solutions that need manual setup for blending do not scale.

8 Deceptions must be DATA-DRIVEN.

A deception solution must be driven by the vulnerabilities in the network and the current threat landscape. Integration with the SIEM and cyber threat feeds is essential for effective deception.

Solutions that do not integrate with the SIEM cannot be dynamic.

9 Deceptions must STUDY THE ATTACK.

Threat engagement and analysis is an intrinsic part of a complete deception solution. A thorough understanding of the attack helps fix all vulnerabilities targeted by the attack and close all back doors to completely neutralize the attack.

Solutions that do not provide attack TTPs are equivalent to low-interaction solutions.

10 Deceptions must be part of the LAYERED DEFENSE

A deception solution cannot function in isolation. It needs to integrate with the security ecosystem to both provide effective deception and quick response.

Solutions that do not interact with the security ecosystem cannot respond to the attacks.

REFERENCES

1. "Emergence of Lying in Very Young Children"
   Angela D. Evans, Kang Lee
   Dev Psychol. 2013 Oct; 49(10): 1958–1963

2. "Military deception"
   https://en.wikipedia.org/wiki/Military_deception

3. "New Weapon in Russia's Arsenal, and It's Inflatable"
   http://www.nytimes.com/2016/10/13/world/europe/russia-decoy-weapon.html

4. "2016 Data Breach Investigations Report"
   http://www.verizonenterprise.com/verizon-insights-lab/dbir/2016/

5. "The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage"
   Clifford Stoll, 1989

6. "An Evening with Berferd In Which a Cracker is Lured, Endured, and Studied"
   Bill Cheswick, In Proc. Winter USENIX Conference 1992

7. "The Value of Honeypots"
   Lance Spitzner, Oct 2001
   https://www.symantec.com/connect/articles/value-honeypots-part-one-definitions-and-values-honeypots

8. "The GasPot Experiment: Unexamined Perils in Using Gas-Tank-Monitoring Systems"
   Kyle Wilhoit, Stephen Hilt, Aug 2015
   http://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/the-gaspot-experiment

9. "Phoneypot: Data-driven Understanding of Telephony Threats"
   Payas Gupta, Bharat Srinivasan, Vijay Balasubramaniyan, Mustaque Ahamad
   Network and Distributed System Security Symposium, February 2015

10. “Splunk’s Adaptive Response Initiative”
https://www.splunk.com/en_us/solutions/solution-areas/security-and-fraud/adaptive-response-initiative.html

11. “Honeypot Farms”
Lance Spitzner, August 2003
https://www.symantec.com/connect/articles/honeypot-farms

12. “Staged cyber-attack reveals vulnerability in power grid”
CNN, September 26, 2007
http://www.cnn.com/2007/US/09/26/power.at.risk/

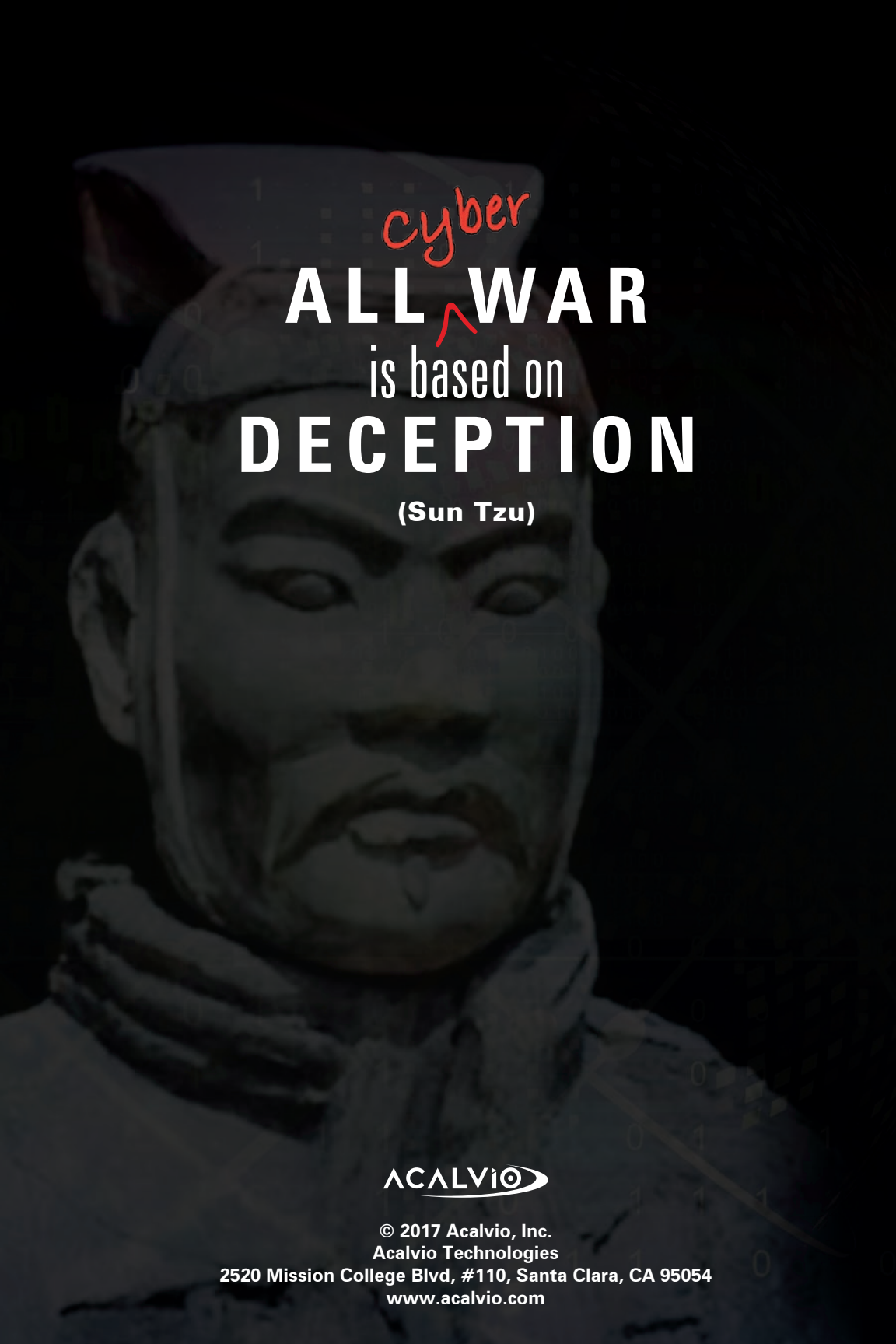
13. “Mirai”
https://en.wikipedia.org/wiki/Mirai_(malware)

14. “The 2016 Cloud Migration Survey Report”
http://info.cloudendure.com/2016-Cloud-Migration-Survey.html

15. “Microsoft Security Intelligence Report”
Volume 21, January through June 2016
https://www.microsoft.com/security/sir/default.aspx

16. “2016 Cost of Data Breach Study: Global Analysis”
June 2016
Sponsored by IBM, Independently conducted by Ponemon Institute LLC


ACALVIO



ACALVIO

Acalvio Technologies
2520 Mission College Blvd, #110, Santa Clara, CA 95054
www.acalvio.com