

# 2IMS20 — Cyberattacks Crime & Defenses

## Lecture Notes

Arthur Geel, 0907552

Publication date:

31.01.2020

### Table of Contents

- 1 — Webapps and Attack Principles
- 2 — SQL Injections and XSS
- 3 — Targeted Attacks (In General)
- 4 — Targeted Attacks on Critical Infrastructure
- 5 — Case Study: IT Attacks
- 6 — IoT and Detection
- 7 — Monitorability
- 8 — Defensive Policies
- 9 — Case Study on Smart Buildings
- 10 — Cybercrime Markets Operation
- 11 — Cybercrime & Malware Commodification
- 12 — Social Engineering and Advanced Phishing Attacks
- 13 — Network and Security Trends in Automotive Systems

# 1 — Webapps and Attack Principles

## Web apps

A few years ago, attacks used to be on the operating systems. Nowadays, it's easier to attack the web applications. This is due to the core flaws of web technologies: HTTP is a connectionless, **stateless protocol**.

**All information required** to process your requests on any website (i.e. proof of who you are, information about what you have done so far) is **stored in your browser** in the form of cookies.

An **HTTP request** contains parameters: host, referer, cookies, username, password. There are two kinds of parameters: GET and POST. where GET are disclosed in the URL, POST are hidden, and therefore more secure.

**SSL** (Secure Sockets Layer) takes care of securing the communication (TCP/IP). This means that all data that is passed between the web server and the browser is **encrypted**, meaning that nobody besides these two parties can see what is sent.

## OWASP Top 10 Application Security Risks — 2017

1. **Injection** flaws such as SQL, NoSQL and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
2. **Broken Authentication** occurs when authentication and/or session management are implemented incorrectly, allowing attackers to compromise passwords, keys or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.
3. **Sensitive Data Exposure** occurs when sensitive data (financial, healthcare, etc.) is not protected properly. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft or other crimes.
4. **XML External Entities (XXE)** can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution and denial of service attacks. This happens due to older or poorly configured XML processors processing XML documents.
5. **Broken Access Control** — restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.
6. **Security Misconfiguration** is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries

and applications be securely configured, but they must be patched and upgraded in a timely fashion.

7. **Cross-Site Scripting (XSS)** allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface websites, or redirect the user to malicious sites. It occurs whenever an application includes untrusted data in a new webpage without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript.
8. **Insecure Deserialization** often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.
9. **Using Components with Known Vulnerabilities** may undermine application defenses and enable various attacks and impacts due to the fact that they run with the same privileges as the application.
10. **Insufficient Logging and Monitoring** allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data.

## Cookies

Cookies are used to store data from sessions, and are set by HTTP requests. Each cookie has a domain and a path, and can have an expiration moment. Cookies can only be retrieved if the user is on that specific domain and subdirectory.

On a higher level, internet service providers and larger web services use **Supercookies** to better track you, which are persistent, meaning that if you remove cookies or browse in private mode, they still persist.

- Browsers do not allow arbitrary (java)scripts to read cookies.
- Browsers do not allow cookies to be transmitted to third parties.

## Watering Hole Attack

Watering Hole Attacks are a type of strategy where the victim is part of a particular group (organization, industry or region). The attacker infects websites / tools they use with malware, which sees the victim infected whenever they access the watering hole, usually with a dropper. The dropper on the target contacts the CC (Command and Control site) to download the full malware. The malware carries out the exploitation (data exfiltration, keylogging, etc).

## Phishing

Phishing is a strategy of obtaining sensitive information such as usernames, passwords and likewise by disguising themselves as a trustworthy entity in electronic communication.

## **DDoS (Distributed Denial of Service)**

A DDoS attack is a cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the internet.

### **Further Reading:**

- OWASP, T. (2017). *Top 10-2017. The Ten Most Critical Web Application Security Risks*. OWASP™ Foundation. The free and open software security community. URL: [https://www.owasp.org/index.php/Top\\_10-2017\\_Top\\_10](https://www.owasp.org/index.php/Top_10-2017_Top_10).

## 2 — SQL Injections and XSS

### SQL Injection

SQL injection occurs when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

#### How to defend against SQLi

- *Avoid the interpreter entirely, or*
- Use an interface that supports *bind variables* (e.g. prepared statements or stored procedures)
- *Encode all user input* before passing it on to the interpreter
- Always perform *white list input validation* on all user supplied input
- Always *minimize database privileges* to reduce the impact of a flaw.

### XSS (Cross-Site Scripting)

XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface websites, or redirect the user to malicious sites. It occurs whenever an application includes untrusted data in a new webpage without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript.

There are two types of XSS:

- **Stored** or **persistent** stores the malicious code on the web page / email message. When the victim accesses the content, the malicious code is executed.
- **Reflected** or **non-persistent** is 'supplied' by the victim themselves, i.e. a link. This contains malicious code that is reflected by the server once it is loaded.

How to defend against XSS:

- **Signature-based filters** recognize and block attacks based on a set of signatures.
- **Input sanitization** encodes the input, so that it does not trigger the script on the browser's side.
- **Input truncation** prevents input of a large length to be fully sent, reducing the opportunity to XSS.

### 3 — Targeted Attacks (In General)

#### Lockheed Martin Cyber Kill Chain

Typical cyber attacks have seven steps:

1. **Reconnaissance:** Intelligence gathering on the victim(s). This can include harvesting email addresses, conference information, etc.
2. **Weaponization:** Preparation of the attack, gathering of tools.
3. **Delivery:** Delivering weaponized bundle to the victim via phishing or watering hole.
4. **Exploitation:** Exploiting a vulnerability to execute code on the victim's system.
5. **Installation:** Installing the full malware on the asset.
6. **Command and Control:** Command channel for remote manipulation of the victim.
7. **Action on Objectives:** With 'Hands on Keyboard' access, intruders accomplish their original goals (i.e. data exfiltration, etc).

The problem is that targeted attacks are **relatively easy to carry out**. Until a few years ago, you needed to craft your own attack, but now you can simply buy the malware to do so.

A **Vulnerability** is a weakness which can be exploited by an attacker to perform unauthorized actions within a computer system.

A **Zero-Day vulnerability** is a computer-software vulnerability that is unknown to, or unaddressed by, those who should be interested in mitigating the vulnerability. Until this is done, hackers can exploit it to adversely affect computer programs, data, additional computers or a network.

**Living off the Land** is making use of tools already installed on targeted computers, or running simple scripts and shellcode directly in memory.

#### Further Reading:

- The Real Story of Stuxnet, Posted 26 Feb 2013, by DAVID KUSHNER  
<https://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet>
- Falliere, N., Murchu, L. O., & Chien, E. (2011). W32. stuxnet dossier. White paper, Symantec Corp., Security Response, 5(6), 29. Retrieved from  
<https://nsarchive2.gwu.edu/NSAEBB/NSAEBB424/docs/Cyber-044.pdf>
- T. Conway, R. M. Lee, M. J. Assante. Analysis of the cyber attack on the Ukrainian power grid. SANS ICS, 2016.  
[https://ics.sans.org/media/E-ISAC\\_SANS\\_Ukraine\\_DUC\\_5.pdf](https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf)

## 4 — Targeted Attacks on Critical Infrastructure

### SCADAs and PLCs

The industry has special types of computers that are part of the infrastructure. These perform critical tasks in support of the business. There are two general types:

- **SCADA** (Supervisory Control and Data Acquisition). These send control commands to remote devices such as PLCs. Example commands can be: *'set the level of water to value x'*, or *'give me the current water flow rate'*.
- **PLC** (Programmable Logic Controller). These process the commands sent by the SCADA. They make decisions on its control program in order to produce the output required.

SCADAs and PLCs communicate via (proprietary) industrial protocols such as MODBUS, DNP3, IEC-104, etc.

### Problem Statement

The inherent problem with SCADAs is that they were designed ~20 years ago. They were isolated, with no security in mind, and access was local (no internet). Nowadays, they are **connected through the back-office**, and allow remote control.

Lots of proprietary protocols make them **very hard to patch** (vendors are very reluctant to do so), hence their vulnerability.

However, we see relatively few attacks on SCADA because these are **not scalable**. As they have proprietary controls, the setting they are in is different, little standardization. All these make attacks hard to replicate, and for a non scalable business model.

### StuxNet

StuxNet was a cyberattack on an Iranian nuclear enrichment facility that took place in ~2009-2010. It was one of the first of its kind, and previously unseen in magnitude: it utilized four zero-day vulnerabilities, which means it had to have been done by a state-sponsored entity.

The initial attack vector was a **USB-based worm**, which spread on back-office computers of the enrichment plant. It spread, hid and updated itself (through C&C communication).

Then, after penetrating the network using zero-day exploits and privilege escalation, it attacked the Siemens SCADA and PLC systems. It replaced PLC code which changed the rotation frequencies periodically, yet hid itself with the **first PLC rootkit ever**. It fed fake data back to the SCADAs, where everything seemed to be fine. However, the centrifuges kept destroying themselves.

### Duqu

Duqu was a cyberattack with the same building platform of StuxNet. The dropper was a MS word document which contained a 0-day exploit. Its purpose was **espionage**: information gathering on the PC. It contained a keylogger. Duqu was harder to study as by default it destroyed itself after 30 days.

### Flame

Flame was information-collecting malware, discovered in May 2012. It was also similar to StuxNet, although much larger and more complex. To this day, it is unknown what the spreading mechanism was, although it used the same **print spooler** and **LNK exploits** as StuxNet. Flame also was used for espionage: collecting passwords, screenshots, audio, skype conversations and pdf and autocad files.

If Flame had no network available, it saved the gathered intelligence on USB sticks, through which it can infect other computers and use their network connections to communicate with the C&C servers.

### Gauss

Gauss had the same platform as Flame, but different modules (gauss, lagrange, gödel, taylor). It was discovered in June 2012, and partly a banking trojan. However, it was also very targeted: one of the modules was encrypted in such a way that it can only be decrypted on its target system. As such, it has not been decrypted yet.

(This was done to protect its vulnerabilities and exploits)

### Red October

Infection method "low profile", but organizationally huge and well-orchestrated. It has been active since 2007, focusing on diplomatic and governmental targets. It's resistant to C&C server takeover, and has a resurrection module.

### Havex/Dragonfly

Advanced malware: spread through multiple vectors. It could be the first step of a cyberwarfare campaign, targeted towards US and European energy firms since 2013. The attack vector were **phishing emails + watering hole attacks**, and **trojanization of legitimate software bundles**.

Havex appears to be a state-sponsored operation (high degree of technical capability), expected to be based in Eastern Europe. Havex can be detected in two ways:

- 1) The moment it performs a network scan.
- 2) The moment it communicates with C&C servers.



### 2015 Ukraine Power Grid Attack

In December 2015, a major Ukrainian power grid was attacked. This affected up to 225,000 customers in three different distribution-level service territories, and lasted several hours.

The infection spread in a **spear phishing email** with malware ridden word-document, containing a version of **Blackenergy 3**. This connected the C&C, which established persistent access to the targets. It also harvested credentials that allowed it to **connect via VPN**, making it more stealthy.

- The attackers accessed the networks for a longer period of time, which is where they **prepared** the attack. Through this connection they explored the Oblenergo networks and customized their tactics on their different **Distribution Management Systems** (DMS).
- They learned to use the three DMS's, and developed **malicious firmware** for the serial-to-ethernet devices. For this, they must have had access to **in-house devices** to test their work-in-progress.
- Finally, in their **attack** they opened the Oblenergo breakers, stopping the power output. Then, they ran the **KillDisk** application, destroying any local backups. Finally, they performed **telephonic floods** against the customer support.

How could we have detected it? **Phishing** is difficult to counter, and the **watering hole** too. The communication to the **C&C** could have been picked up, but only if the workstation in question usually does not connect to the internet.

However, **Network whitelisting** in the internal network would have reported the upload of malicious firmware to the serial-to-ethernet devices.

### 2016 Ukraine Power Grid Attack

A year later, a similar attack was performed. This resulted in a power cut that hit part of Kiev. The blackout lasted over an hour.

This time, the **Industroyer malware framework** was used, which was more automatized when compared to the previous attack. This malware is specifically designed to attack energy companies, and is **modular**.

### Further Reading:

- Bruce Schneier on the Equifax Hack  
<https://www.schneier.com/cryptogram/archives/2017/1115.html>
- Symantec Internet Security Threat Report 2019. Retrieved from  
<https://www.symantec.com/content/dam/symantec/docs/reports/istr-24-2019-en.pdf>
- Symantec Internet Security Threat Report 2018. Retrieved from  
<https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-2018-en.pdf>
- The 6 biggest ransomware attacks of the last 5 years. Retrieved from  
<https://www.csoonline.com/article/3212260/ransomware/the-5-biggestransomware-attacks-of-the-last-5-years.html>
- *Petya ransomware and NotPetya malware: What you need to know now*. URL:  
<https://www.csoonline.com/article/3233210/ransomware/petya-ransomwae-and-notpetya-malware-what-you-need-to-know-now.html>
- The Hacking Team Hack) Retrieved from <https://pastebin.com/kHUzWWm9>
- Stress Testing the Booters: Understanding and Undermining the Business of DDoS Services, by Mohammad Karami, Youngsam Park, Damon McCoy, 2015  
<https://arxiv.org/pdf/1508.03410v1.pdf>
- T. Conway, R. M. Lee, M. J. Assante. Analysis of the cyber attack on the Ukrainian power grid. SANS ICS, 2016.

## 5 — Case Study: IT Attacks

**Ransomware** is malware that encrypts files and/or partitions of a hard disk in order to extract a ransom of the victim, usually in cryptocurrency.

Notable ransomware attacks include

- Cryptolocker (2013), the first ransomware, similar to how StuxNet was for attacks on infrastructure.
- TeslaCrypt (2015)
- Symplelocker, an Android ransomware
- Petya (2016), which overwrites the master boot record and installs its own boot loader, making the computer unusable.
- WannaCry (2017), which spread like a worm. It used the EternalBlue exploit which was stolen from the NSA.
- NotPetya (2017), using EternalBlue and EternalRomance. Not really ransomware, as the encryption key is random.

### Data Leakage

Most notably, Equifax had a data leakage in 2017, as they used an Apache framework. On March 6, 2017, Apache published a security advisory regarding **manipulated HTTP headers**, which allowed hackers to access affected systems.

Despite the patch being released one day later, Equifax failed to install this until July, allowing hackers free reign for months. This exposed the personal data of more than **143 million consumers**.

### DDoS (Distributed Denial of Services)

Attacks ranging from 30 seconds to an indefinite time, where the victim network is receiving such heavy traffic that it is unreachable for real users. DDoS usually use previously identified misconfigured amplification servers, which are sent to the victim.

## 6 — IoT and Detection

**Internet of Things (IoT)** is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

The market of IoT devices is growing, which comes with a number of problems.

- Firstly, IoT devices are not always developed with **security** in mind. For example, a lot of IoT devices are used with **standard credentials** or on **standard ports**, making them easy to access. Furthermore, they often are riddled with vulnerabilities that allow attackers to access the network.
- IoT devices get discontinued often. While it may still function, it will never be patched anymore.

### How is IoT Exploited?

Hacked IoT devices often are used in botnets, to carry out DDoS attacks using the compromised network. However, IoT can also act as a vulnerability in the network, allowing attackers to escalate privileges and move laterally across the network.

### How Can We Stop/Detect Cyber-Threats?

#### Rejection-based

- **Negative model.** Is able to recognize an attack when they see one (e.g. anti-viruses, blacklisting, signature-based systems).
  - Can be done using **signatures** (part of a payload of a known attack) and **heuristics** (when you see something that resembles a malicious entity).
- **Cons** — false-negatives problem.
  - You have to know the attack, 0-day vulnerabilities have no signatures.
  - It takes a long time to devise and deploy signatures (expensive + time).
  - Non-mainstream systems (SCADAs) have no good signatures.
  - Signatures and heuristics are relatively easy to evade (polymorphic virus)
- **Pros** — very low false positives.
  - Can be used as blocking system, intrusion prevention.
  - Knows what it is once they detect something.
  - Does not need to be reconfigured when the system is reconfigured.

## Rejection-based

- **Positive model.** Is able to recognize what is normal behaviour of the system. Anything abnormal is considered an attack. (e.g. firewalls, whitelisting systems)
  - Can be based on "flows" (quantitative analysis)
  - Can be based on analysis of payload (qualitative analysis, neural networks)
- **Cons.**
  - Resource-intensive — systems change, whitelisting systems need to adapt with them.
  - Relatively easy to circumvent unless they are very accurate.
  - Provide little information about the attack, if they detect one.
- **Pros.**
  - You don't have to know the attack to block it.
  - Low false positives if correctly configured - can be used for blocking purposes.

## Anomaly Detection

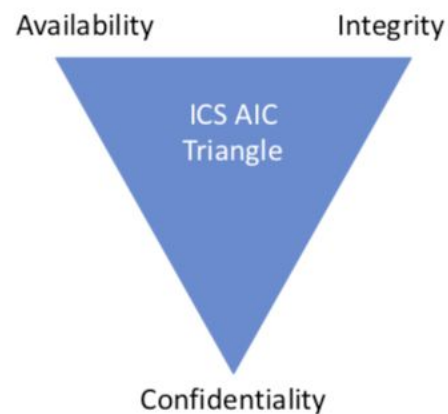
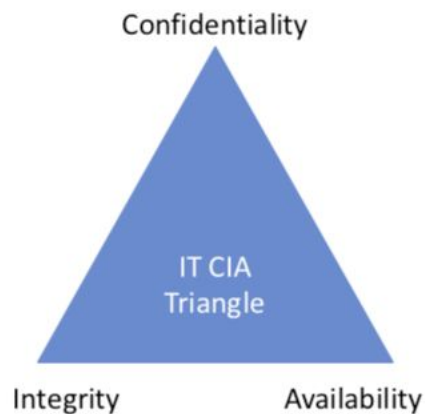
- **Quantitative** (flow-based)
  - Tells you when too many things are happening. Important for situational awareness, though includes a lot of false positives.
- **Qualitative** (payload-based)
  - Tells you when a single information unit is anomalous.
- **Cons.**
  - Its applicability depends heavily on the particular instance of the target system.
  - A lot of false positives.
  - Provides little information about the attack, even less than whitelisting.
- **Pros.**
  - Little/no setup costs.
  - May allow you to see 0-day attack.

## Environments







Industrial Control System (ICS) and Information Technology (IT)

## ICS network priorities

- Different systems, different protocols, different threats
- → Different priorities: CIA vs. AIC



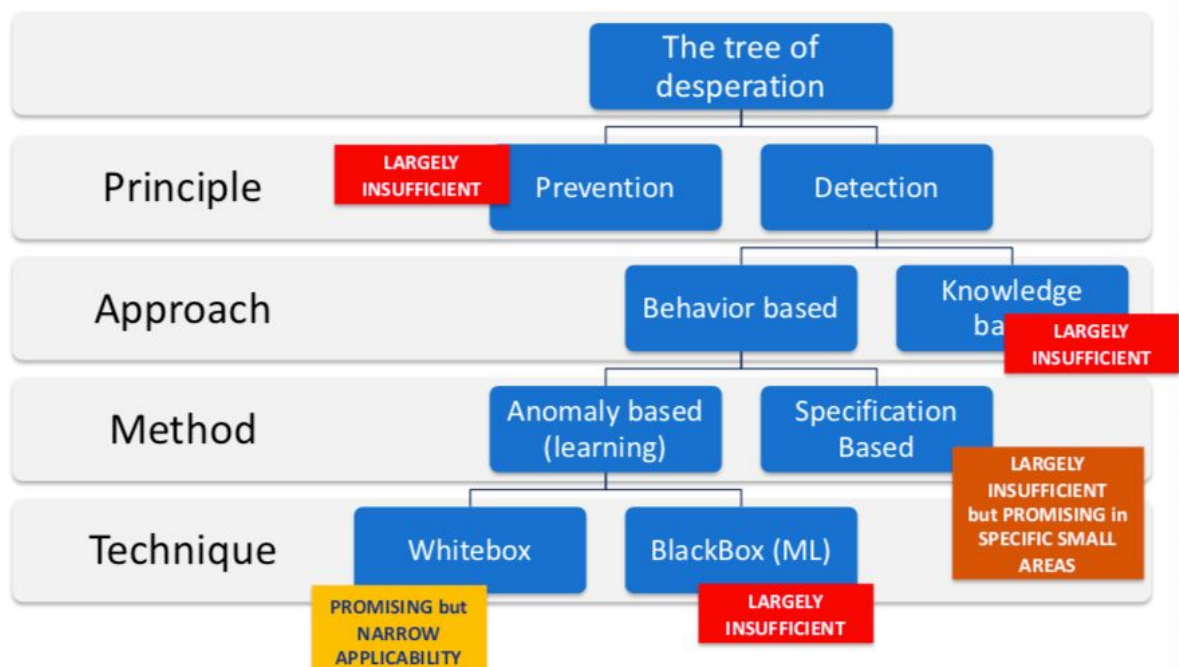
## IT vs. ICS networks

IT network	ICS network
 <p>Thousands of hosts, hundreds of applications</p>	 <p>Limited number of hosts</p>
 <p>Devices' behavior might be difficult to predict</p>	 <p>Same operations repeated over and over</p>
 <p>Behavior patterns change frequently</p>	 <p>Changes are less frequent</p>

## 7 — Monitorability

### Problem Statement

- Attacks are difficult to counter because present systems are hard to monitor.



There are two ways to deal with attacks: **Prevention** and **Detection**. However, the former is not sufficient as it is impossible to make software bug free. Additionally, if it were bug-free, it would be used in some way that would make it vulnerable.

Therefore, **Detection** seems to be the way. **Intrusion Detection Systems (IDS)** are created to help. These have the following crucial criteria: **false negatives**, **false positives**, **actionability**, **adaptability** and **scalability**. However, academia fails because it simply is not feasible to create- and maintain an accurate system that meets all three criteria.

**IDS** can be **knowledge-based** and **behaviour-based**. However, the former are not always suitable due to the fact that they detect a fraction of the attacks, are unsuited to 'non-mainstream' systems, and are not cost effective.

Therefore, we delve deeper into **behaviour-based** systems. They have two main methods of doing so: from **specification-based** and **anomaly-based**. However, the former once more is not suitable to all areas due to the lack of *adaptability* and *scalability*. However, they serve a use case for some subparts of systems (i.e. IoT).

Anomaly-based has two techniques: **black-box** and **white-box**. B-B use machine learning approaches: the semantics used by the detection system are unrelated to the semantics of the target system. However, in doing so, they are unable to transfer their results into actionable reports. You know that there is an anomaly, but have no follow-up.

Therefore, WhiteBoxing seems to be the way to go. This is based on e.g. understanding the communication protocol, extracting command and setpoints and whitelisting them. This can be useful for SCADA/ICS, but is not fool-proof.

- *Most IT systems are simply not understandable.*

Therefore, we got stuck. We can only make things better if we **change the way we write software** to make it more amenable to **monitoring**. Software should be supervisable and privacy-preserving by design.

Important to consider is the preservation of privacy. This can be done for some enterprises, but not for all (i.e. internal processes may disclose sensitive information regarding how something is made or done).



## 8 — Defensive Policies

### Encryption in ICS Networks

Sometimes we see encryption being used inside ICS networks. However, this may not be **wise at all**. Here's why:

- Encryption in internal networks **does not yield extra SCADA security**.
  - Infection in StuxNet was through **USB sticks**. Encryption would not have helped in containing the virus.
  - Infection in Ukraine Blackout was through **phishing emails** on corporate networks.
  - Encryption does not stop any of the known SCADA attacks because they went through the endpoint, not the cable.
- Encryption in internal networks **can negatively affect security**.
  - In most observed SCADA incidents, a proper network monitoring solution could have detected the attack as an anomaly. However, encryption of data would make this impossible to do.
- Encryption in internal networks **can complicate troubleshooting**.
  - Encryption limits or slows traffic monitoring for troubleshooting.
- Encryption comes with the costs of managing, distributing and revoking keys and certificates.

However, these **defensive policies are certainly needed**:

- User/device authentication, and message authenticity.
- Integrity of the communication.

Encryption could be **useful** for:

- Protecting long-haul communications along untrusted networks (i.e. internet) or in adversarial environments.
- Guaranteeing confidentiality when it is crucial (i.e. privacy of smart meters)

### Economics of Security Policies

Users' rejection of security advice is entirely rational from an economic perspective: password advice is outdated and does not address actual threats, 100% of certificate error warnings appear to be false positives, and the time spent reading URLs to avoid phishing would be 100 times higher than the phishing losses caused by them.

## Why are we not all attacked

Attacks usually are no zero-sum: more value is lost than created. (i.e. indirect costs). However, large-scale attacks must be profitable in expectation, facing a sum-of-effort rather than a weakest-link defense. The various reasons why attackers leave us alone are:

- Average success rate is too low.
- Average value is too low.
- Attacks (and attackers) may collide too often.
- Attack is expensive with regard to alternatives.

## Types of attackers

- Criminals (Cost < benefit)
- Hacktivists (Cost < fixed limit)
- Nation states (No constraints)
- Occasional (Typically: insiders)

## Common (flawed) advice for passwords

1. Length
2. Composition (e.g. digits, special characters)
3. Dictionary membership (any language)
4. Don't write it down
5. Don't share it with anyone
6. Change it often
7. Don't reuse passwords across sites

## Attacks on passwords:

1. Phishing
2. Keylogging
3. Brute-force attack on the user's account
4. Bulk-guessing attack on all accounts on the server
5. Special-access attacks.

These make most of the advice useless.

The root reason why many of our security policies are erroneous is due to their unfalsifiability. **Things can be declared insecure by observation**, but not by the reverse. Furthermore, policy-creep further accumulates wrong policies.

### **Myths on Cybersecurity:**

- *"Deployment of SSL prevents you from all types of attacks."*
- *"Implementation of a firewall as a network perimeter defense makes the environment bulletproof."*
- *"Deployment of an Intrusion Prevention System (IPS) or Intrusion Detection System (IDS) prevents malicious code from entering my network."*
- *"Usage of Two-Factor Authentication (2FA) protects from all types of fraudulent activities."*
- *"Deployment of security policies eradicates the risks."*
- *"Malware is distributed primarily through shady and rogue websites such as torrents and warez."*
- *"Email filtering mechanisms only allow secure and verified attachments to be delivered with emails."*
- *"Malware infections are specific to certain operating systems."*
- *"Mobile devices are completely secure."*
- *"Virtualization technologies are untouched by malware."*

## 9 — Case Study on Smart Buildings

*Guest lecture by Elisa Costante, ForeScout.*

### Three Trends That Make Breaches Difficult To Prevent

1. Growth of Devices and Platform Diversity
  - a. Innumerable device-specific operating systems.
  - b. Cannot get agents onto new devices.
  - c. Cannot write agent-based software for every OS.
2. New Threats
  - a. Better funded actors (i.e. nation states).
  - b. Advanced malware.
  - c. Malicious use of OT protocols and features.
3. OT Convergence with IT Heightens Risk
  - a. OT networks are no longer physically separated.
  - b. Threats moving between cyber & physical dimensions.
  - c. Assets are highly vulnerable and rarely can be patched.

**Smart Buildings** have Building Automation Systems (BAS) that may take care of access control, HVAC, solar panels, lighting, lifts and fire alarms. However, future smart buildings may be interconnected with full **smart cities**: people, other buildings, power grids, cars, etc. etc. This opens up pandora's box.

*Three reasons why we should worry:*

1. Legacy Systems
  - a. 60% of buildings have systems that are >20 years old.
  - b. There is no encryption.
  - c. There is no authentication.
2. Connectivity
  - a. More connections = more vulnerabilities.
  - b. Ports may be open by default.
  - c. Passwords may be default.
3. Critical Buildings
  - a. Airports.
  - b. Data centers.
  - c. Hospitals, malls and public spaces.

### Reducing Security Risks:

- Network monitoring (gain visibility)
- Asset inventory (know your devices)
- Anomaly detection (new hosts and links)
- Threat detection (indicators of compromise)

### Key takeaways

- Landscape
  - Critical infrastructure relies on legacy systems.
  - Cyber risks for critical infrastructure are on the rise.
  - Critical infrastructure networks are vulnerable.
- Visibility
  - Understand what network devices are doing.
  - Assess risks, threats and vulnerabilities.
  - Understand the current resilience state of the network.
- Detection
  - Catch known and unknown threats.
  - Pinpoint weak spots and current inefficiencies.
  - Gather all evidence required for incident response.

## 10 — Cybercrime Markets Operation

### About Cybercrime

Cybercrime is an extension of traditional crime, but it happens in forms **vehiculated by a technological system**. Technological systems can be a:

- **Vehicle to crime** if the target is outside of the technological-IT domain;
- **Vehicle & target of crime** if the impact is suffered by a (group of) technological systems. This can extend to the users of those systems.

Cybercrime is related to malware/attacks/data, drugs, weapons, human trafficking, terrorism, child exploitation, social disturbance, etc.

**Cybercrime Markets** are the obvious result of cybercrime. On here, you'll find:

- Hacking products:
  - **First-order**, enabling the hack with vulnerabilities, exploits..
  - **Second-order**, after the hack, credit cards, banking info, trade secrets,
  - Trade occurs largely **online**. Product selection, contract and delivery.
- Non-hacking products:
  - Drugs, weapons, stolen hardware, etc.
  - Mostly physical goods
  - Trade is a **mix of online and offline**.

**Agency theory** stipulates that this market only operates if the contracts are valid or can be satisfied. I.e.: *how do economic agents stipulate contracts?*.

- The 'Principal' asks for a service;
- The 'Agent' provides the service.

The information that the principal and the agent have is **key** to achieve a sustainable **market equilibrium**. However, if there is an **information asymmetry**, the market is bound to fail. This can happen in two ways:

- 1) The Agent/service provider knows much more about the good/service than the Principal/buyer.
- 2) The Principal/buyer has no viable way of verifying that the Agent/service has respected the "contract".

If the market offers no ways of addressing information asymmetry, "good" principals/agents will be **pushed out of the market**. A fundamental problem of criminal markets is the following: as virtually all interactions are criminal, **enforcing contracts is very hard**. You cannot go to the police and report scammers.

Within criminal markets, there are two core types:

- **Forum-based**, where most of the malware is traded.
  - Niche market, repeated trade is limited.
  - Large problems with **adverse selection** and **moral hazard**.
    - How can you evaluate an exploit or a malware before buying it?
    - It's very difficult to test/prove it without giving it away.
- **E-commerce like**, which are generally used for drugs, weapons and other physical goods.
  - Larger market, customers come back to circle of trusted vendors.
    - Non-technical goods, expectations are lower.
    - Repeated purchases create **rich feedback history** for buyers and sellers.
  - **Anonymity** is a key factor.
    - Quick entry / exit from the market reduces exposure.

Forum-based markets of malware have mechanisms to regulate them:

Market Mechanism	Evidence	Rationale
<u>Moral Hazard</u>	Fear of punishment	Trials and active discussion hinders unfair behaviour
	High cost of entry	Being banned is expensive
	History of trade	Backs up seller's trustworthiness / quality of goods
	Reputation system	Reputation levels match punishment
<u>Adverse Selection</u>	Product samples	Sellers provide tests, trials, videos of the exploit in operation for buyers to assess
	Feedback mechanisms	Products are reviewed by buyers, positive/negative feedback

**Further Reading:**

- Akerlof, George A. "The market for "lemons": Quality uncertainty and the market mechanism." *Uncertainty in Economics*. 1978. 235-251.  
<http://www.perishablepundit.com/docs/market-for-lemons.pdf>
- Herley, C., & Florencio, D. (2010). Nobody sells gold for the price of silver: Dishonesty, uncertainty and the underground economy. *Economics of Information Security and Privacy*. URL:  
[https://www.researchgate.net/profile/Tyler\\_Moore2/publication/216757809\\_Economics\\_of\\_Information\\_Security\\_and\\_Privacy/links/542c332e0cf29bbc126c4349/Economics-of-Information-Security-and-Privacy.pdf#page=48](https://www.researchgate.net/profile/Tyler_Moore2/publication/216757809_Economics_of_Information_Security_and_Privacy/links/542c332e0cf29bbc126c4349/Economics-of-Information-Security-and-Privacy.pdf#page=48)
- OPT: Europol. Drugs and the darknet. Perspectives for enforcement, research, and policy. 2017. Retrieved from  
[http://www.emcdda.europa.eu/system/files/publications/6585/TD0417834E\\_NN.pdf](http://www.emcdda.europa.eu/system/files/publications/6585/TD0417834E_NN.pdf)



## 11 — Cybercrime & Malware Commodification

### "Hacker" Evolution

Initially (1990s—onwards), "hackers" were simply **security enthusiasts** that tried to innovate threats. Goals were:

- Improving security;
- Security testing;
- Software evaluation;
- Hardware flaws.

Later (2000-2010), **script-kiddies** entered the scene. Their activities were to automate trivial attacks originally developed by security enthusiasts. Their goals were to create havoc and service disruption. However, these had limited technical understanding so they were unable to innovate.

Finally, from 2010 onwards we see **economic attackers**. These have created a business case for hacking, innovating threats, delivering at scale in order to create a malware economy. Their goals are:

- Data exfiltration (CCNs, banking, accounts, ...);
- Computational power;
- Ransomware.

The resulting **underground economy** shifted from a *"one-do-it-all" business* model to a *"composite" model*. The complexity and scale of the attacks is too big to have a profitability if the attacker has to do all Lockheed Martin steps by themselves. Therefore, **product composition** occurs where the activity is facilitated by multiple services.

A list of **composite cybercrime business models**:

- Spam to sell counterfeit products;
- Ransomware campaigns;
  - Scareware;
- Click fraud;
- Finance/banking fraud;
- Booter services (DDoS);
- Exploit-as-a-service and Pay-Per-Install.

**Exploit Kits** are websites that serve vulnerability exploits and ultimately malware. They drop malware upon successful exploitation. They only work if they receive victim traffic (direct links, ads, iframes, redirections).

The underground economy has services that trade connections (maladvertising, spam, iframes on legitimate websites). The attacker buys connections from specific users with specific configurations (to maximise stealth).

Many exploit kits **defend themselves** against antivirus / robot detection. Mostly through **payload and malware obfuscation**.

#### Further Reading:

- Huang, Kurt Thomas Danny Yuxing, et al. "Framing Dependencies Introduced by Underground Commoditization." In Proceedings of WEIS 2015. URL: <https://storage.googleapis.com/pub-tools-public-publication-data/pdf/43798.pdf>
- Van Wegberg, Rolf, et al. "Plug and prey? measuring the commoditization of cybercrime via online anonymous markets." 27th {USENIX} Security Symposium ({USENIX} Security 18). {USENIX} Association, 2018. URL: [https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-van\\_wegberg.pdf](https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-van_wegberg.pdf)
- Allodi, Luca. "Economic factors of vulnerability trade and exploitation." Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017. Retrieved from [https://dl.acm.org/doi/pdf/10.1145/3133956.3133960?casa\\_token=bHfxPC\\_hj7v0AAAAA:5RrSws8htZf9CMmU-B4kFvsb3mtp237ZN4U6tL3bxkHpJ47xz964Ix3XSRdxbby\\_EtxrYSD8OV62](https://dl.acm.org/doi/pdf/10.1145/3133956.3133960?casa_token=bHfxPC_hj7v0AAAAA:5RrSws8htZf9CMmU-B4kFvsb3mtp237ZN4U6tL3bxkHpJ47xz964Ix3XSRdxbby_EtxrYSD8OV62)
- OPT: Grier, Chris, et al. "Manufacturing compromise: the emergence of exploit-as-a-service." Proceedings of the 2012 ACM conference on Computer and communications security. ACM, 2012. Retrieved from [https://dl.acm.org/doi/pdf/10.1145/2382196.2382283?casa\\_token=h6e4GwUB6z8AAAAA:EIK5AnLmQWFycrPuE7-qS9\\_WD1hyo--IeBiZ9fhrDcbzFPnJ2qx75LAVjf-gCLZEuPUGFpYZwmiJ](https://dl.acm.org/doi/pdf/10.1145/2382196.2382283?casa_token=h6e4GwUB6z8AAAAA:EIK5AnLmQWFycrPuE7-qS9_WD1hyo--IeBiZ9fhrDcbzFPnJ2qx75LAVjf-gCLZEuPUGFpYZwmiJ)

## 12 — Social Engineering and Advanced Phishing Attacks

### Sources of Malware Infections and Attacks

There are two main sources of infections and attacks:

- **Technological**
  - Software vulnerabilities are the main vector of infection.
  - These allow for unwanted actions from the attacker. (C. I. and A. impacts)
  - Some still require user interaction.
    - E.g. open attachments in mails that THEN exploits the vulnerability.
- **Human**
  - Social Engineering
    - **External** attacker obtains information from internal user.
    - Attacker obtains access (i.e. passwords, malware execution, information leaks)
    - The art of persuasion.
  - 'Disgruntled Employee'
    - **Internal** attacker misbehaves.
    - Attacker already has some access (i.e. damaged systems, leaked information, disrupts processes).

**Social Engineering** is a wide set of attacks that exploit human nature to (usually) breach data confidentiality. They seek to **persuade** a human being in performing actions solicited by the attacker.

The **Elaboration Likelihood Model (ELM)** is borrowed from psychology to explain how persuasion in social engineering works. The model proposes two 'routes' through which we assess information:

- **Central route:** within this route, we extensively think and consider information. This is cognitively expensive, hence our brains seek to bootstrap this process.
- **Peripheral route:** this route requires no significant cognitive effort, and is essentially an 'automatic pilot'.
  - It's a **shortcut**: we analyze "relevant" cues to evaluate the information, while these cues may not be trustworthy at all.
  - **Persuasion** happens here: through adjunct elements (i.e. likability of the subject, trust and physical attractiveness) we may be persuaded to perform behaviour or accept facts that we may otherwise not have.

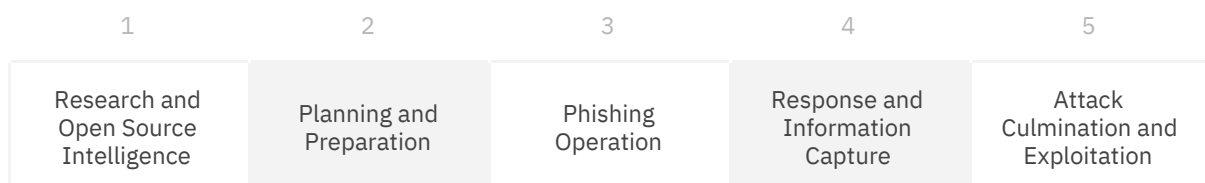
Therefore, **social engineering** and **phishing** occur within the peripheral route.

## Hacking a Human: 6 Factors

There are six main psychological factors that are exploited heavily within social engineering. A lot of academic research into these has been conducted to further the discipline of marketing.

1. **Reciprocation:** we are prone to making normative commitments to our peers: if they do something nice for us, we try our best to reciprocate. We often experience cognitive dissonance if we don't, which pushes us to play along instead.
2. **Consistency:** we are prone to making continuance commitments: we try to act rationally, hence we tend to repeat patterns of the past. This is linked with loss aversion and the sunk cost fallacy.
3. **Social Proof:** we are prone to making affective commitments: if we see social proof of peers partaking in a certain activity or train of thought, we are bound to follow.
4. **Likeability:** for some reason, we easily extend our trust in certain people or instances beyond their relevant scope.
5. **Authority:** this factor is often used to give the message extra credibility, or impose fear upon the victim to collaborate. A well-known experiment is Milgram's 1960's study on using authority to persuade/force people to give shocks to victims.
6. **Scarcity:** this factor makes us react quickly, and makes us experience pressure which makes uninformed decisions more likely. In cybercrime we see this with e.g. Ransomware: you have a set amount of time to collaborate before the damage is irreversible.

There are different approaches to social engineering. The most simple form is known as a **single-stage attack**, and is usually aimed at collecting sensitive information about general targets. The stages are shown below:



Of course, more complicated attacks also exist. These are known as **two- or multiple stage attacks**. These types of attacks involve an initial reconnaissance that gathers information needed for the second stage (i.e. to increase the credibility of the attack).

A schematic of the steps included in such an attack is shown below. If the attack has more than two stages, **phases 5 through 7 are repeated**.

1	2	3	4	
Research and Open Source Intelligence	Planning and Preparation	Phishing Operation	Response and Information Capture	—
	5*	6*	7*	8
—	Replanning and Preparation	Spear Phishing Operation	Response and Information Capture	Attack Culmination and Exploitation

A more **elaborate overview of the activities and patterns** seen in the previously discussed attacks can be seen below:

Pattern Phase	Typical Activities	Pattern Interactions
1. Research and Open Source Intelligence	Search for open source intelligence Establish attack objectives Identify opportune targets	Attacker researches and strategizes about potential targets and specific objectives
2. Planning and Preparation	Develop attack strategy including means to avoid detection and mitigation by UIT organization Prepare phishing attack artifacts	Attacker plans phishing attack and creates phishing artifacts (e.g. phishing mail, mobile text message, phony website, malware to be implanted)
3. Phishing Operation	Release phishing artifact via email, cellphone, rogue website, or other means Wait for response	Attacker initiates phishing attack through email, cellphone, rogue website or other means
4. Response and Information Capture	Gain access and/or privileges to obtain greater information reach Implant malware to achieve information objectives Identify other opportune UIT targets and internal system information, and capture guarded and sensitive information	One or more targets unwittingly respond to phishing artifact and become a UIT Attacker detects or is alerted to UIT response and obtains initial information directly from UIT data entry. Attacker implants malware on victim's machine or network. Attacker obtains desired information via malware.

The **Impersonation Model** captures the different strategies that cybercriminals use in their phishing attempts. Four levels are recognised:

- Address Spoofer (manipulates SMTP protocol to spoof the email address)
- Name Spoofer (wrong email, but correct name)
- Previously Unseen Attacker (authoritative name of fake company)
  - Relatively hard to block.
- Lateral Attacker (uses actual identity of compromised person)

In an effort to resolve these, a number of technologies and protocols were developed. One is the **DomainKeys Identified Mail (DKIM)**, which has a public key and signature that are shared between the sending- and receiving email services to verify the identity.

Additionally, we have the **Domain-based Message Authentication, Reporting & Conformance (DMARC)**. Before receiving an email, the recipient's client performs validation tests (IP blocklists, reputation, rate limits) and applies the DMARC policy. Only if the content passes these, it is sent to the recipient.

However, relating these back to the four types in the impersonation model, **only Address Spoofing** is combated by **DKIM and DMARC**.

In order to defend against **Name Spoofers** and **Previously Unseen Attackers**, some form of **Human Judgement** is needed. However, **Lateral Attackers** are not defended against well by any mechanism.

#### Further Reading:

- Stevens Le Blond. A Look at Targeted Attacks Through the Lense of an NGO. Usenix Security 2014. Retrieved from <https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-blond.pdf>
- Workman, Michael. "Wisecrackers: A theory-grounded investigation of phishing and pretext social engineering threats to information security." Journal of the Association for Information Science and Technology 59.4 (2008): 662-674. <https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.20779>

## 13 — Network and Security Trends in Automotive Systems

### About our Modern Cars

Cars have undergone a great shift since their introduction in the late 19th century. The paradigms have shifted from **Research and Development** (1966-1995) to **Embedded Systems** (1995-2002), to **Infotainment** (2007-2012) to **V2X** (2012-Ongoing). V2X signifies the connection between the Vehicles and Anything.

Modern cars are built up from Electronic Control Units (ECUs), which pretty much are interconnected computers that contain **Sensors** and **Activators**. Here is an overview of core car subsystems:

- **Powertrain** (engine power, transmission, gear control)
- **Chassis** (in-vehicle active safety, ABS, suspension system)
- **Body** (in-vehicle body / climate control)
- **Passive Safety** (airbags, seat belt pretensioners)
- **Telematics** (infotainment, gps, cd player)

Modern cars are considered a **Cyber Physical System**: *"An intelligent system combining electronics and software, which is connected to the real world through sensors and actuators, and is also connected to each-other and the internet."*

Modern car manufacturers usually are now 'compilers' of the cars. Rather than manufacture each of the individual components themselves, they purchase them from OEMs. This results in a considerable network complexity.

Typically, a car has **over 100 ECUs**, connected to different buses. Most buses operate using a **Controller Area Network (CAN)**, which seems to be the root of the difficulties.

This is due to its broadcasting architecture: it contains a **CAN ID** and **priority**, and the **payload**. However, information on the sender and recipient are not there, making it vulnerable.

The **OBD-II Port** (on-board diagnostic) can be used to read CAN data streams, and therefore can be used to manipulate the system (hack it).

### Problem Statement

- Cars as Cyber Physical Systems.
- Complexity (>100 ECUs, 10M+ lines of code, a lot of bugs included)
- Life cycle (>15 years)
- Patching (hard to do)

## Attack Vectors

Most interesting ECUs are on the CAN bus. Therefore, hackers use CAN message injection: where they send arbitrary identifiers and data. This is facilitated by the broadcast nature of the CAN protocol.

*How do you reach the CAN bus?*

1. **Using the OBD-II port** (easy).
2. **Through OEM vulnerabilities** (difficult, but becoming more mainstream)
  - a. Indirect physical (Media players, aftermarket gear, charging)
  - b. Short-range wireless (Wi-fi, bluetooth, TPMS, keyless entry, DSRC)
  - c. Long-range wireless (Cellular, HD radio, DAB, RDS)

## Cars' Hackability Triad:

- Network Architecture
- Remote Attack Surface
- Cyber Physical

## Who's Hacking Cars?

- Criminals/Terrorists: causing physical harm and wide-spread damage.
- Manufacturer's competition: vehicle theft/copy the vehicle's designs and specs.
- Second-hand car dealers: vehicle notification compression and/or avoidance of incurring replacement expenses, targeting the safety of the vehicle.
- Intelligence Agencies: espionage for tracking and recording sensitive information. (Through GPS, passenger calls, etc.)

However, the main challenge to hack cars is in its **limited profitability**. So far, car hacking does not scale (CAN bus outputs differ per car, and take months to reverse-engineer. Also: exploitable vulnerabilities differ per car). Therefore, the **effort required > potential gains**.

Nevertheless, **modern research** to reduce the 'hackability' of cars is focused on:

- Intrusion detection systems;
- ECU identification with certificates;
- Use of MAC of data integrity and authentication;
- Network segregation with trusted group;
- Attestation based security architecture;
- On-chip Trusted Platform Module (TPM);
- ...