Cyber Attack Models for Smart Grid Environments☆

Peter Eder-Neuhauser*, Tanja Zseby, Joachim Fabini, Gernot Vormayr

Institute of Telecommunications, TU Wien, Gusshausstraße 25 / E389, 1040 Vienna, Austria

Abstract

Smart grids utilize communication technologies that make them vulnerable to cyber attacks. Because the power grid is a critical infrastructure, it is a tempting target for sophisticated and well-equipped attackers. Cyber attacks are usually based on Malicious Software (malware) that must communicate with a controlling entity over the network to coordinate and propagate.

In this paper we investigate communication and spreading of malware in smart grids, proposing a comprehensive, generic model for cyber attack life-cycles, and addressing the specific characteristics of smart grid environments. The generic model includes the building blocks for all major known malware types as well as different propagation methods, access vectors, scanning techniques, control structures, attack methods, triggers, and cleanup mechanisms. Supported by an extensive review of earlier work, we examine the techniques of many different existing malware types with respect to their potential impacts on smart grids, and then discuss countermeasures. Toward this end, we analyze and evaluate a variety of types of malware – well-known but persistent malware, malware featuring outstanding or innovative concepts, as well as very recent malware – with respect to metrics that are fundamental to the generic model. We then introduce three novel superclasses of malware that are particularly suited for smart grid attacks, and evaluate their methods and impacts. Our model provides a basis for the detection of malware communication and extrapolates from existing technologies in order to predict future malware types. The smart grid specific malware types thus extrapolated provide insight into new threats and help utility companies to prepare defenses for future attacks.

Keywords: Communication Networks, Malware, Smart Grids, Cyber-Physical Systems, Cyber Attacks

1. Introduction

Smart grids, i.e., networked power grid control equipment, depend on Information & Communication Technology (ICT) for managing power flux and energy balance. A smart grid hosts several types of devices, including but not limited to measurement equipment (e.g., Phasor Measurement Units (PMU) and smart meters), actuators (e.g., breaker-switches and disconnectors), and networking equipment (e.g., gateways and control nodes). These critical devices are just as susceptible to Malicious Software (malware) as are classical Internet technologies and consumer electronics. However, unlike consumer electronics, traditional power grid environments have a focus on long-term stability and plan for hardware life-spans of 10 years or more. As devices age, unknown vulnerabilities of hardware, operating system, software, and protocols emerge. Such vulnerabilities pose a serious threat to the infrastructure. While consumer electronics need not fulfill the same life-cycle requirements of industrial devices, their base technology is similar.

Today, few malware implementations that have actually caused severe physical damage to cyber-physical systems are known. However, during the last decade the number of such highly evolved malware, capable of orchestrated cyber-physical attacks has increased. The complexity and sophistication of these malware implementations lead to the assumption that massive resources were invested in their development and that they may be financed by large stakeholders such as nation states [1–3]. The detection and analysis of existing malware is of paramount importance for the implementation of countermeasures, which are critical to the safe operation of smart grids. However, the danger of extensively documenting malware algorithms is that it decreases the effort required to craft novel malware. Malware families could be created that combine existing mechanisms with unpublished, highly effective zero-day-exploits (zero-days) and then be exploitable by less equipped adversaries to attack critical infrastructures.

The remainder of this paper is structured as follows. Section 2 reviews the state of the art of modern malware. Section 3 identifies the most important characteristics of smart grid environments with respect to networking and security, placing particular emphasis on the differences from the classical Internet. Section 4 proposes a generic model for the life-cycle of malware-based cyber attacks, describing all involved stages. Several subsections then explain the specific parts in more detail. A classification in Section 5 identifies similarities, benefits, shortcomings, and differentiating features of existing malware.

☆This is a pre-print version for personal use of the manuscript accepted and published by the Elsevier Journal on Sustainable Energy, Grids and Networks, Volume 12, December 2017, Pages 10-29. The published manuscript version is available at `https://doi.org/10.1016/j.segan.2017.08.002`.

*Corresponding author

Email addresses: peter.eder-neuhauser@tuwien.ac.at (Peter Eder-Neuhauser), tanja.zseby@tuwien.ac.at (Tanja Zseby), joachim.fabini@tuwien.ac.at (Joachim Fabini), gernot.vormayr@nt.tuwien.ac.at (Gernot Vormayr)

We concentrate on sophisticated modern as well as older but prevailing malware. Section 6 investigates the most distinct features that are effective in smart grid environments and their possible evolution into future threats. Section 6.2 discusses conceivable attack types specific to smart grid environments, and Section 6.3 the corresponding defense strategies. Section 7 introduces three new malware superclasses specifically designed for smart grid attacks, based on knowledge from existing malware. They include descriptions and characteristics based on the features that have been isolated in earlier sections. Section 8 concludes the paper and presents an outlook on future work.

2. State of the Art

According to Line et al. [4], the energy industry is traditionally well-prepared for threats such as physical damage, accidents, natural disasters, or equipment failure as long as they affect small, restricted areas. However, coordinated cyber attacks can do significant damage, yet are still inadequately addressed, due to their low probability of occurrence. Using coordinated, distributed resources, cyber attacks can target sufficient critical power control equipment simultaneously to originate cascading effects and eventually cause the system to collapse. Several incidents of attacks on the energy industry have been reported by the Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) [5] that demonstrate how organized groups with motivation, resources, and competence can cause serious damage. They report that about 60% of all cyber attacks in the year 2013 targeted the energy sector, albeit not necessarily power control networks.

Traditional cyber attacks target individuals, businesses, intelligence services, or military adversaries. However, the latest successful attack on the Ukrainian power grid [6] manifests a new impact level that affects the civil population. Attackers managed to infiltrate utility companies and cause damage to power switching equipment that led to wide ranging blackouts. We later discuss the methods used by the attackers, which could in part have been counteracted if awareness had been higher.

Line et al. [4] consider Cyber-Situational-Awareness (CSA) to be a future cornerstone in protection against intruders. CSA attempts to mitigate threats before attacks are conducted successfully. The goal is the awareness and comprehension of:

- The current situation inside the network.
- The situational evolution during an attack.
- The causes of the current situation and its implications.
- The quality of the collected information.
- The impact of attacks on critical equipment.
- The attacker behavior before, during and after an event.
- The possible future developments and recovery plans.

Cyber attacks involve networked devices whose software becomes infected with malware. According to [7–10], a number of different malware types that are capable of infecting hosts exist. Malware can spread either actively in the form of worms or in botnets, or passively in the form of viruses. Furthermore, they can utilize functional malware such as trojan horses, spyware, adware, spammers, sniffers, crypto-lockers, backdoors, logic bombs, and more in modular extensions. Although these malware types behave vastly different, they all at some point utilize a payload that exploits a vulnerability to change the host's behavior. Attackers may use any of a variety of infection vectors, e.g., software exploits, compromised removable drives, or manipulated emails. However, in the remainder of this paper we will use the term *"malware"* to refer to all malware-types. For instance, worms often extend their functionality by downloading modules, which could be seen as its own malware type, it is not useful to categorize them into any more detail, as shown in [1, 9, 11].

Malware instances often communicate with each other, forming a botnet, i.e., a remote Command & Control (C&C) infrastructure, with a botmaster. This allows the malware and its controlling entity to perform flexible commands, in order to react to the environment or adjust to different attack goals. Extortion schemes, Distributed Denial-of-Service (DDoS) attacks, espionage campaigns, repurposing attacks, and lately also cyber-physical attacks are examples of attacks on networks that can be triggered. Section 4.6 and Table 1 elaborate on them.

Modern cyber attacks are increasingly conducted by so-called Advanced Persistent Threats (APT). These groups often utilize stealth techniques in order to remain concealed for long periods and seem to be increasing the complexity, versatility and potential damage of their attacks. According to Thonnard et al. [12], most APTs utilize zero-days. This further lowers the chance that affected defenders can discover an attack. Thus, attackers are now able to further obfuscate operations and utilize attack methods that are highly difficult to defend against.

We considered several attack models proposed by existing work, and have based our generic life-cycle model presented in Section 4 on these. Lockheed Martin [13] introduced the cyber-kill-chain as part of a framework for intelligence-driven defense to prevent network intrusions. It comprises the following stages: reconnaissance, weaponization, delivery, exploitation, installation, control, and action. The SPARKS project [14] proposed a similar linear kill chain composed of the following successive stages: intrusion, installation, lateral movement, exploitation, pivoting to the control network, deployment, and attack. Another linear approach was taken by Matrosov et al. [1] using a client-server model with focus on the client side. With social engineering as the intrusion vector, this approach cites local exploits and malicious infection as methods of gaining persistent access. Schneier [15] introduced the concept of "attack trees" that branch off at each decision an attacker makes. This formal model is well known for its versatility, as it is able to describe any attack according to its scalability. All four approaches provide a clear sequence of events that occur during a cyber attack. We considered these models in developing our own generic approach by reusing parts of their main concepts.

In addition, Gollmann et al. [16] have discussed cyber attacks in which each stage is connected to the others. These stages include access, discovery, control, damage, and cleanup. Rather than following a fixed sequence, they argue that malware can move from any stage to another at any given time. We provide details on each stage in our model and extend this approach in several aspects. We also elaborate on logical sequences these stages can follow under different circumstances. Li et al. [7] discussed several characteristics in target-finding, propagation and transmission schemes that we also took into account in our work. Trullols et al. [17] have researched large-scale vehicular networks and introduce attack stages and target discovery schemes that complement those scanning techniques we include in our model.

Since smart grids are a critical infrastructure, they deserve progressive protection from cyber attacks. Consequently, security updates should be deployed in a timely manner as unpatched vulnerabilities increase their attack surface. Vulnerabilities based on shortcomings in the hardware may even persist for prolonged periods, as replacing widely deployed hardware is accompanied by financial and staffing issues. This is why, aside from APTs, malware that uses readily available technologies is a realistic threat to modern power grid control systems. The attacker can choose among a multitude of targets in the communication infrastructure and the electricity grid. Targeted equipment includes power switches, transformer stations, and field devices, e.g., smart meters or PMUs [9].

The **contributions** of the present work include the following:

- It suggests a generic life-cycle model for cyber attacks on smart grid environments and discusses several considerations related to malware propagation behavior and technologies.
- It provides an in-depth analysis of building blocks of malware and a comparison of existing malware capabilities, including propagation vectors, access methods, scanning behavior, attack goals, and defensive strategies that correspond to all metrics discussed in the generic model.
- It proposes three conceptual models for potential smart grid enabled malware based on the knowledge of existing malware capabilities. Additionally, it presents outlooks that go beyond technologies available today.

The proposed life-cycle model provides a basis for the detection of malware communication and enables the prediction of future malware types. The three conceptual smart grid specific malware types provide insight into new threats and can help utility companies to prepare for potential future attacks.

3. Smart Grid System Model

Due to the critical role that electricity grids play in a nations' society and the increasing interconnectivity of smart grid devices via ICT, incentives and opportunities to attack smart grids are increasing. Furthermore, the entire range of sophisticated malware known from the Internet today is increasingly suited for deployment against smart grids. Section 3.1 describes the basic smart grid environment and identifies essential properties that differentiate it from classical Internet communication. Section 3.2 defines the capabilities and characteristics of attackers, and Section 3.3 elaborates on security assumptions that the proposed attack life-cycle model relies upon. These discussions establish the general operational environment of this work.

3.1. Smart Grid Communication Model

Smart grid ICT infrastructures differ in several aspects from classical Internet communication. Smart grid communication is mainly Machine-to-Machine (M2M) communication, which makes it more predictable than traditional (Human-to-Human (H2H) or Human-to-Machine (H2M)) Internet traffic [18, 19]. Predictability simplifies anomaly detection, but challenges remain depending on observation points and protocols in use. In addition, smart grid environments often favor homogeneous structures. Utility providers purchase components from one or few vendors, which results in monocultures of devices, software, protocols and network-topologies. This can lead to large populations of devices with identical vulnerabilities that provides an ideal environment for malware spreading [17, 18].

Furthermore, security and privacy concerns may lead to conflicting goals. For instance, smart meters that collect data on energy-consumption and monitor user behavior also raise privacy concerns. This data can be combined with personal information collected about each user on the Internet. Many smart meter types are also capable of remote disconnection. Such features raise security concerns, because they can be abused in order to disconnect many households at once [18]. Based on [18] and [19] we identify the following characteristics that distinguish smart grids from Internet communication.

- The predominance of M2M communication, instead of H2H or H2M communication.
- Homogeneity (monocultures) in the choice of devices.
- Differences in network requirements across several device types (smart meters, gateways, sensors, actuators, etc.).
- Physical access to field devices by non-trusted parties.
- Huge planned life span of installations in the field, e.g., utility companies plan for smart meters to be operational for more than 10 years.
- Pre-authorization of field devices to local servers, e.g., pre-configured certificates or authentication tokens that are stored in field devices.
- A need for remote monitoring, maintenance, and updates. In particular the requirement for remote upgrade support.

3.2. Smart Grid Attack Model

Many of the malware we investigated is APT supported. Therefore, our attack model is based on well financed and highly skilled adversaries. However, we also take less-equipped attackers into account, who are able to reorganize publicly released malware and retrofit new features. Our attack life-cycle model thus intentionally includes many conceivable attack vectors, familiar from the Internet landscape. However, we exclude malicious insiders, instead concentrating on outside threats, as the former can best be defended against through increased awareness and strict company policies (e.g., user management) rather than by technical means. We assume the attackers are capable of utilizing zero-days and operate on distributed network resources. We present several malware types and their properties in Section 5, as well as discussing their impact on smart grids. Although some do not utilize zero-days, we assume incorporating their features in more capable malware is feasible and will present a major challenge in the future. Furthermore, we assume that attackers cannot interfere with properly implemented defense mechanisms, except for known evasion techniques that are discussed in Section 5.4.

3.3. Smart Grid Security Model

Well-established security guidelines to prevent predictable attacks exist from classical Internet security, as do standards on power control equipment, and smart grid implementations. Although the reality in existing installations is still alarmingly deficient, for our analysis we assume that these measures are in place. The German Federal Office for Information Security (BSI) [20, 21] has developed a network architecture based on a smart meter gateway that significantly narrows the attack surface. However, these strict rules for smart metering do not protect other types of devices and services hosted in smart grids. PMUs or other sensors often operate time-critical services that cannot tolerate retransmission. With the use of network segmentation for instance, a clear separation of the objective of such services is possible. Furthermore, field devices have to cope with an increasing number of vulnerabilities over time [18]. Therefore, we expect that attacks on power infrastructures will become more common upon the implementation of automated smart grid control. We aggregate several guidelines from [14, 18–22] and summarize the most important of them. In some countries, these are compulsory by law, in others they are only suggestions. For this study we assume that the following measures have been implemented. We also discuss additional security measures in Section 6.3 for future examples.

- *Security updates:* Regular and timely updates for devices in business and industrial networks prevent vulnerabilities and minimize the window of opportunity for attackers.
- *User management*: All users, including administrators, are restricted to environments suitable for their role.
- *Password policy*: Strong password policy ensures the use of long non-repeating high-entropy pass-phrases.
- *Anti-virus*: Modern anti-virus software is used based on heuristics and remote reputation services.
- *Network segmentation*: Subnetworks with distinct objectives are separated, e.g., segmentation exists between administrative network and industrial control environment.
- *Restricting remote access*: Since segmentation can be circumvented by remote access, the latter is strictly controlled and limited to trusted parties, if permitted at all.
- *Strict firewall rules:* All access is prohibited by default except for white-listed hosts and services, protecting users from Internet threats.
- *Decentralization of critical services:* Decentralization strengthens resilience against attacks. However, explicit countermeasures are required to counteract propagation methods that do not depend on a functioning network, e.g., infected removable drives.
- *Dimensioning hardware for future software updates*: Smart grid devices remain in service for more than 10 years. Whenever resource-constrained hard- or software is integrated into modern equipment as part of a modular design, the entire system security may be compromised with regard to sophisticated attacks. Therefore, these devices are prepared for future demands and provide sufficient resources to support updates.
- *User education*: One of the most basic counter measures is user education, which protects against many simple access vectors. In combination with strong passwords, this can significantly impede propagation.
- *Ensuring data integrity, confidentiality, and availability*: The correct implementation of standard protocols prevents, e.g., packet-integrity-attacks or sniffing.

4. A Generic Life-Cycle of Malware Attacks

This section proposes a generic model for multiple stages in the life-cycle of cyber attacks and malware communications. We consider the existing approaches discussed in Section 2 and reuse some of their concepts in our model. However, rather than confining our model to a strictly linear approach, we argue that loop-back cycles more accurately reflect the greater flexibility that is intrinsic to modern malware. All phases in our model revolve around *access* to resources. The proposed model begins with a *discovery-propagation-access* cycle for the network side of target-discovery and propagation. This is followed by an *infection-access* cycle for host infection. A *control-access* cycle represents the remote-control infrastructure that allows for functional updates, as in C&C-triggered updates and remote commands. Subsequently, the model proposes *attack*, *trigger*, and *cleanup* stages, which are deployed once sufficient access to resources is attained and the attacker is capable of commencing the attack. Figure 1 and the following subsections provide detail on all stages:

- *Access*: The centerpiece of any attack is direct or remote access to critical resources. Initial access is often of inferior quality and requires privilege escalation. When administrator access to a resource is available, further propagation or the commencement of an attack may be possible. Applicable access methods are summarized in Section 4.1.
- *Discovery*: If insufficient access is available to achieve a specific goal, discovery and scanning methods are used to locate new victims in the network. Attackers thus gain

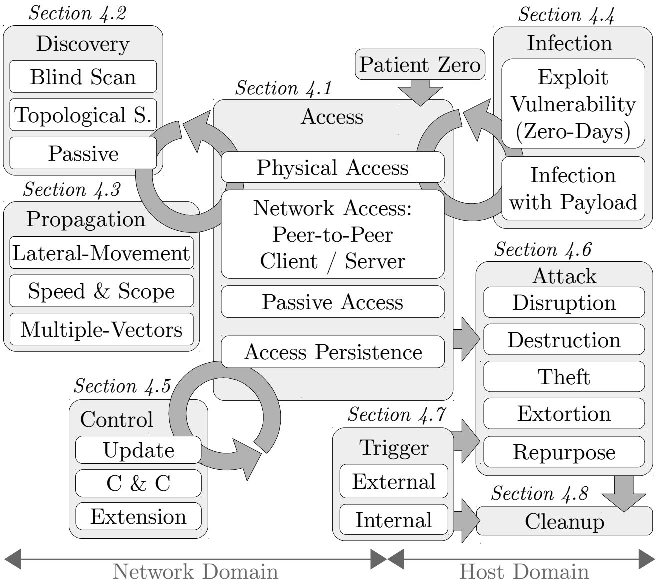

Figure 1: Generic Stages of Malware-based Cyber Attacks

knowledge. Techniques range from noisy to covert scanning and are summarized in Section 4.2. Topics of interest include topological security and heterogeneity of devices.

- *Propagation*: After discovering new targets, malware propagates to new hosts using exploits. Once transmission is complete, the malware has gained access to the target. Propagation techniques range from direct connections, to highly covert ones, as elaborated in Section 4.3. Details concerning transmission can also be found there. It is worth noting that all propagation techniques are also suitable as access techniques.

- *Infection*: After gaining access to a new host, malware infects it and escalates user rights to a higher level, increasing its access quality. Section 4.4 contains details on this.

- *Control*: Most modern malware implementations are controlled externally. In addition to loading new modules and controlling the spread as seen in many cases [1, 2, 23–25], other methods are described in Section 4.5.

- *Attack*: Successful attacks depend on sufficient access to compromised resources on the network. The direct transition from the access block to the attack block in this model reflects this dependency. A variety of types of attacks are possible including service disruption, physical destruction, theft, extortion, or repurposing (cf. Section 4.6).

- *Trigger*: Complex attacks require coordinated action and orchestration. Attack triggers can be hard-coded or remotely activated, as described in Section 4.7.

- *Cleanup*: Many adversaries conduct covert operations [26, 27] and may be interested in concealing their technology. Therefore, hiding their tracks by removing parts of the malicious code can be effective in combination with other persistence mechanisms as discussed in Section 4.8.

4.1. Access

In this section we provide detail on access methods to resources such as hosts and networks. Furthermore, we point out similarities among those access methods that are also used as propagation methods as a prerequisite for all other blocks in the generic model. At the end of this section, we discuss the possibility of long-term persistence of malware in spite of countermeasures implemented by defenders. Initially a foothold inside the network is established, often via an office PC (patient zero). After infection, the attacker escalates its privileges to a higher level of access by exploiting a vulnerability allowing for further steps such as *discovery*, discussed in Section 4.2, and *propagation*, discussed in Section 4.3. We generally distinguish between host access and network access. However, both are required for successful infiltration of networks.

The following list summarizes the considered access vectors and provides examples:

- *Physical access* refers to methods by which an attacker reaches the target host directly and modifies hard- or software. Examples include direct data manipulation via Universal Serial Bus (USB) drives, hard drive exchange, live disc reboot or direct installation of malware.

- Active *Peer-to-Peer (P2P)* network access refers to P2P protocols and file-sharing, in which each host acts as a server and a client at the same time.

- Active *Client / Server* network access designates typical structures with separated clients and servers.

  - *Server-to-Server (S2S)* access refers to the use of vulnerabilities that enable the infection of other servers, for instance via buffer overflow attacks to insert backdoors or server exploits [28].
  - *Server-to-Client (S2C)* access includes all methods that allow the compromising of a client system from an infected server, e.g., watering hole attacks [29] where groups of users are targeted by compromised web services or rapid client reinfection by persistently infected servers [24].
  - *Client-to-Server (C2S)* access refers to all methods which, for instance inject exploit code into websites [30].
  - *Client-to-Client (C2C)* access allows the infection of other clients via remote code execution or auto-run shared files, without infecting the server [3].

- *Passive access* includes methods by which the attacker gains remote control over resources through social engineering and deceptive abuse of a persons trust:

  - *Host-to-Network-Share (H2NS)*: This group subsumes all access vectors that utilize vulnerabilities to infect files in trusted network shares, including backup drives and shared folders [31].

- *Removable Drives (RD)*: Infected USB drives, external hard drives, or other removable media provide effective propagation and access methods by utilizing auto-run exploits on the target host. They can, for instance, be strategically placed so they attract individuals who unknowingly infect the target network. Another example is the infection of employee USB drives by the host. These methods benefit from being invisible on the network, yet exhibiting effective reach across air-gaps. Some sophisticated malware even disinfect removable drives in order to increase their stealthiness [26].
- *Phishing emails*: The bulk dispatch of emails containing infected attachments or malicious web links is a method heavily used to gain initial access. Some malware uses phishing alone for propagation, but most resort to alternative methods after establishing a foothold. Several email protocols are used in phishing methods and victims typically have to open the attachment or web link to commence the infection process [32].

Faulhaber et al. [33] argue that passive access vectors, which require user interaction, rank among the most effective. They account for about 88% of all Microsoft Windows-based system infections. Of these, phishing emails with malicious attachments or web links account for approximately 45%, auto-run features on removable drives for another 26% and H2NS infections for 17%. The authors conclude that exploits on non-updated hosts account for below 6% and zero-days even less.

Although zero-days seem to be rare, it is a mistake to infer from this that the need for countermeasures is any less. Advanced malware have been observed in the wild on several occasions [2, 11, 25, 26] and examples are expected to be seen even more frequently in the future. Since substantial resources are required for their development, adversaries capable of funding such campaigns can afford and often do utilize zero-days. They can immediately unhinge access rights restrictions, granting the attacker system access at will.

Lee et al. [6] elaborate on APT-made malware based on an analysis of a recent cyber attack on the Ukrainian power grid, that occurred in December 2015. They found that phishing emails were used for initial access, then a backdoor established C&C to the infected hosts and allowed further propagation inside the network. After establishing persistence, stealing certificates, and creating administrator accounts the attackers pivoted to the control network of the power grid. Section 4.6 elaborates on the extensive attack methodology, confirming that standard security measures do not suffice to repel advanced attackers [3, 25, 26, 31].

Physical access to field devices is one of the most obvious means of entry specific to smart grid control networks. Furthermore, two-way communication between field devices and servers opens vectors toward higher levels in the ICT hierarchy. These field devices are connected to a critical network that must be protected using adequate intrusion detection, which can identify and hinder malware from spreading. Border gateway protection is insufficient once attackers are inside the network and can move freely. This is the case if physical access or secondary targets such as trusted partners or external services provide lateral access vectors, as seen during the Aurora attack [34]. Therefore, intrusion and anomaly detection across strictly segmented sub-networks becomes increasingly important [6, 19, 35].

Concerning *Access Persistence* we discuss defensive methods malware authors use to ensure continuous access, even beyond security measures implemented on the host. Based on [1, 3, 11, 23, 24, 26, 27, 31, 32, 36–40], many malware types establish persistence using a variety of methods. These include but are not limited to stealing credentials or injecting malicious code into core processes. Beyond that, malware often uses anti-detection mechanisms such as code obfuscation, encryption, memory residency, or detection of anti-virus software henceforth referred to as anti-malware tools. The capabilities of modern malware are increasing in complexity, transforming them into multipurpose attack platforms. This development is true for both host-based and network-based access forms including C&C, scanning, remote code execution, and propagation.

Based on the aforementioned sources, we can enumerate several persistence methods:

- Manipulation, or deactivation of host-based anti-malware tools helps the malware evade detection.
- Code obfuscation increases stealthiness.
- Multi-layer encryption techniques increase stealthiness.
- Local recompilation changes appearance of the malware.
- Rapid reinfection upon disinfection increases persistence.
- Stealing credentials allows administrator access.
- Memory residency helps in evading detection.
- Service injection into system processes can survive restart.
- Cleanup mechanisms prevent forensic analysis.

4.2. Discovery

This section discusses known discovery techniques such as network scanning, as well as the protocols in use for these techniques. Sufficient access to a host and local privileges are a prerequisite for discovery. Consequently, scanning for new victims precedes the propagation block in Figure 1. Li et al. [7], Staniford et al. [10], and Riley et al. [41] discuss several scanning methods and cluster them into the following categories:

- *Blind scanning* targets randomly generated address ranges and commences sequential, clustered or random discovery. This approach produces a high rate of failed connections and thus anomalies, which are easily detected. Malware employing blind scanning generally spreads fast, yet imprecisely [7, 41].
- *Routing scan* methods use the Border Gateway Protocol (BGP) to decrease the scanning space, resulting in a better hit rate and lower background noise, which can be used by anomaly detection algorithms. Routing scans generally target countries or regions [7, 41].

- *Topological scanning* improves the hit rate further by obtaining information about the network from the host. It operates more stealthily and produces even fewer anomalies. Examples mentioned in [10] show that CodeRed2 is capable of preferring local networks even in a random-scan. Furthermore, it scans outside its local network only with certain restrictions, producing fewer anomalies than its sibling CodeRed1 [7, 41].

- *Passive scanning* does not probe the network actively but rather waits for native connections to be initiated. Such malware types spread slowly, but no scanning anomalies are produced. Infectious packets may be sent by initiating a session with known hosts, yet could trigger anomaly detection. Alternatively, covert attachment to active transmissions may decrease the anomaly output further. Such a threat may spread faster in a homogeneous environment such as a P2P-network, where a host is both server and client, likely running a single dominant implementation [7, 10, 41].

- *Hitlist scanning* requires pre-scanning for a list of initial targets. This list may be created and updated through the use of a botnet. Hitlist scanning produces few anomalies in a network increasing stealthiness. It is a technique for accelerating the initial spread, which is an important measure for the early stages of infection.

- *Permutation scanning* is an augmentation for histlist scanning that distributes the target list between parent and child malware. This method increases stealthiness by decreasing the amount of rescanning, each instance appearing to scan randomly and separately. Hitlist and Permutation Scanning make a malware fast enough to attack most vulnerable hosts Internet-wide in under one hour [7, 10, 41].

- *Distributed scanning* was described by Dainotti et al. [37] and Staniford et al. [10]. An example of a large-scale stealth scan showed that approximately 3 million infected hosts were used to scan the Internet Protocol Version 4 (IPv4) address range randomly in reversed byte order. This method was considered impressively stealthy, as hardly any source addresses rescanned a similar address range. Therefore, few anomalies are visible in traffic. Compared to older scanning methods which generally originate from a small set of sources, this method is more capable of scanning covertly thanks to its sheer size.

Typical techniques used for scanning in the IPv4 address space utilize the following protocols: User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Address Resolution Protocol (ARP), File Transfer Protocol (FTP), or Internet Control Message Protocol (ICMP). See Roger and Tan [42, 43] for details on their scanning methods: Vanilla TCP Connect Scan, TCP SYN (Half Open) Scan, TCP FIN Scan, TCP Reverse Ident Scan, TCP XMAS Scan, TCP NULL Scan, TCP ACK Scan, UDP ICMP Port Scan, ARP Scan, FTP Bounce Scan, and ICMP Ping-Sweeping Scan.

With Internet Protocol Version 6 (IPv6), active scanning is futile due to the large address space. According to RFC5157 [44], even search space reduction techniques can prove futile, although passive scanning could be used instead. An attacker could do so by hosting a harmless and free web service, to attract unsuspecting users [29]. IPv6 addresses of target hosts, including ones that are generated with the privacy extensions, can then be extracted from the generated logs. This method was used by the security search engine Shodan [45, 46] by contributing time servers to the debian Network Time Protocol (NTP) pool [47], and was discovered in 2016. Every network packet to one of the contributed servers resulted in a network scan of the origin. Even the limited lifetime of IPv6 privacy extensions (generally one day) provides a large enough window of opportunity to infect a host.

4.3. Propagation

This section elaborates on propagation methods between hosts via non-network or network transmission. Knowledge of potential victims is required and generated during the discovery phase (cf. Section 4.2), after which successful propagation to a new host allows low level access (cf. Section 4.1). Propagation then leads to wider access and ultimately better knowledge of the environment, which is essential for commencing attacks.

All methods except for physical access as discussed in Section 4.1, are usable as propagation methods. The decision *not to* utilize a method is an implementation choice of the malware in question, which can refrain from using a technically feasible option. For instance, Locky [48] utilizes email for initial access but not for propagation. Analog to Section 4.1, we generally distinguish between active and passive propagation:

- *Active* methods enable self-propagation, usually by exploiting remote code execution or auto-run features. Active propagation is generally very fast and requires no user-interaction. They include P2P, S2S, S2C, C2S, and C2C.

- *Passive* propagation, on the other hand, can be slower yet is more effective at bridging air-gapped networks, e.g., ones in which users physically carry removable drives into segmented networks and infect hosts. Many malware types nowadays have air-gap capabilities [1, 24–27, 40], effectively moving and communicating between seemingly secure networks. Propagation via H2NS or phishing emails requires user interaction in the form of opening files, attachments, or web links.

With regard to secure networks, [10, 11, 33] elaborate on cyber attacks increasingly targeting homogeneous and widely used services. Propagation can commence across vast distances in rapid succession, as displayed in aggressive malware types such as Slammer [7] or CodeRed2 [30]. Man-in-the-middle attacks on monocultures are also familiar from Flame [25], whose fake update servers infect hosts (S2C). Aside from anomaly detection, no feasible network based method exists for ensuring the intended behavior of hosts. There are, however, a number of host-based detection mechanisms that are resource-intensive, for which field devices should provide adequate resources. One particular application for smart grids concerns smart meter updates that run on older models. According to Li et al. [7], malware generally consists of two parts, a payload and a dropper, that may propagate in one of the following schemes:

- *Self-carried*: Payload and dropper are transferred in one file-set. Typical examples include Code Red 1 and 2 [28, 30, 41], and Nimda [10, 30].
- *Second-channel*: The payload is downloaded later through a backdoor, following an infector, i.e., dropper file. Typical examples of second-channel malware are Regin [23], Duqu, Duqu2 [11, 24, 49], and BlackEnergy3 [2].
- *Embedded*: The payloads differ for each instance and masquerade communication as normal traffic, as mentioned by Staniford et al. [10].

According to [24, 26, 34], modern malware increasingly utilizes *modular* propagation schemes such as second-channels that only acquire the modules needed at any given time. They also allow multiple propagation paths, which makes them more diverse (cf. Table 2). Only a few utilize either passive propagation or an unknown embedded method that may depend on the use of *covert channels*. This, however, renders them invisible to network-based detection. Staniford et al. introduced such a method in their "contagion" model [10]. Real world examples include Locky [48, 50], Gauss [11, 26], and Equation [27] which are known for their prolonged covert behavior. There are indications that Gauss utilizes zero-days for propagation, which may imply the existence of unknown covert techniques.

All malware propagation relies on common network- and transport-layer protocols such as Internet Protocol (IP), UDP or TCP. The choice of protocol can have substantial influence on propagation characteristics and performance:

- *UDP*-based malware communication can saturate the maximum link-bandwidth [7, 41]. Such communication often has a small packet size, which enables fast infection. UDP is connectionless, meaning that packets may be dropped or denied leaving no guarantee of delivery. However, broadcast in the network commences regardless. Examples of malware capable of UDP transmission are Nimda [10, 30], Slammer [7, 41], Sality [37, 38], Conficker [33, 37, 51], or Regin [23]. Except for Slammer, the others also use TCP, employing each protocol for different parts of the scanning and propagation cycle [37].
- *TCP*-based malware has the advantage of reliable transmission but is constrained by the number of parallel connections and certain latency limitations. Through parallelization the number of connections may be increased, and simultaneous infection becomes possible. Since TCP connections require a handshake, these malware types are limited by the average Round-Trip Time (RTT) of a network. Real world examples include: Code Red 1 and 2 [28, 30, 41], Nimda [10, 30], Sality [37, 38], Conficker [33, 37, 51], Regin [23], Aurora [1, 27], Stuxnet [1, 40], Duqu and Duqu2 [11, 24, 49], Flame [11, 25], BlackEnergy3 [2], CozyDuke [31], and PLC-Blaster [52, 53].

4.4. *Infection & Exploit*

According to Li et al. [7], most older malware types only target a single operating system and few vulnerabilities. However, Matrosov et al. [1], Symantec [23], and Bencsath et al. [11] show that in the recent past, digital warfare has evolved into a more complex environment. Attacks involving multiple exploits, infection vectors, and payloads are common and must be considered to be the state of the art. The National Institute of Standards and Technology (NIST) [54] and the Mitre Corp. [55] provide lists of known vulnerabilities containing a number of possibilities to exploit all kinds of operating systems.

Increasing modularity in modern malware also allows for a multitude of payloads, thus adding flexibility; for instance, Duqu has only six modules whereas Duqu2 has over one hundred modules [11, 24], increasing its range of possible uses.

According to a number of sources [1, 2, 23–27, 31, 39], most APT-authored malware utilizes zero-days. They are, due to their undisclosed nature, nearly impossible to defend against, which makes them a valuable asset for attackers. This demand has lead to black markets for zero-days. The defense systems for critical infrastructures must be designed to repel skilled attackers. Well-understood threats may be circumvented by standard methods, but deterring APTs requires elaborate security considerations.

Ször et al. [56] and Li et al. [7] categorize malware according to three distinct payload types. Additional features are added to malware according to the following methods:

- *Monomorphic* malware may vary in size through the use of padding and evade detection by fragmentation. Yet all instances produce the same signature and are easily detectable using anti-malware tools. Examples include CodeRed1 and 2, Nimda, Slammer, and PLC-Blaster [7, 10, 28, 30, 41, 52].
- *Polymorphic* malware scrambles its payload through encryption. Therefore, every instance has a different signature and size. However, when the payload or parts of it are decrypted on the local host, every instance has the same signature. This method provides better evasion properties, yet may be detectable using sophisticated anti-malware tools. Examples include Conficker, Regin, Stuxnet, Duqu, and BlackEnergy [1–3, 11, 23, 33, 37, 40, 49, 51].
- *Metamorphic* malware are able to create new instances that appear to be different from their parent. They vary in shape, size, encoding, and encryption and utilize recompilation on the host system to change their appearance. They do not carry a decryptor; instead, all data is carried in one single code body. These types of malware are notoriously difficult to detect. An example is AdWind [32].

4.5. *Control*

Modern malware is generally controlled by a C&C infrastructure, which allows the botmaster to influence them and also enables modular extension and updates. Although, intrusion detection systems try to match C&C traffic patterns found in the communication, signature-based systems such as Snort [57] can be defeated by encrypting communication. Therefore, malware such as BlackEnergy, that used clear text for C&C traffic in early versions [58], employs full encryption in the latest version [2]. Anomaly detection systems may, however, still be capable of detecting suspicious encrypted traffic.

The outcome of our literature review suggests that non-encrypted or non-obfuscated C&C methods have largely been replaced by fully encrypted and obfuscated methods [2, 11, 23, 31]. Some examples even utilize complex multi-layer encryption [36] or highly obfuscated methods [24] which can now be considered to be the state of the art. Another method that helps obfuscate C&C is the use of highly distributed C&C architectures to avoid detection, as do Regin, Aurora, Equation, and Locky [23, 27, 43, 48].

Recent developments, however, push the boundary toward covert communication. Mazurczyk et al. [59] and Kaspersky [26] discuss such behavior in modern malware. While Duqu can attach itself to harmless communication using JPEG images, Regin recently also acquired covert capability. It seems that covert channels are now gaining momentum toward becoming economically viable as most outbound traffic is allowed to pass unrestricted: that is, ICMP traffic can be abused by entering information into the data fields of echo requests and replies. However, today many routers block ICMP traffic for security reasons. While ICMPv4 can be blocked completely, critical functions in ICMPv6 prevent routers from blocking it. RFC4890 [60] and RFC2979 [61] discuss firewall guidelines for IPv6 and IPv4. Even though covert techniques are not yet very sophisticated or wide spread, development is commencing, making them a future threat.

Several sources [1–3, 10, 11, 24–28, 30–32, 36–38, 40, 41, 48, 50–53] elaborate on what protocols, interfaces, and evasion techniques modern malware utilize for C&C. These protocols include: Internet Relay Chat (IRC), ICMP, Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS), P2P protocols, Virtual Private Network (VPN), and Server Message Block (SMB). The following have been used as interfaces for C&C architectures: WinAPI, USB, Network Pipes, Mailslots, and backdoors of older malware. Several methods, including, for instance, Domain Name System (DNS) flux, IP flux, encryption, obfuscation, highly distributed C&C architectures, and covert channels have been used as evasion techniques.

As mentioned above, modern malware may use VPNs stacked on top of other protocols. Regin [23], for instance, supports custom UDP, TCP, SMB, network pipes, HTTP, HTTPS, and ICMP protocols as its base. The protocols are negotiated between infected nodes whereas the control messages use the overlaid VPN. For this reason the observable malware traffic disappears within native traffic encountered in the network. Regin is not only modular, but written from scratch as a service-oriented architecture [62]. This means that modules on the same host have distinct VPN addresses that can only be controlled via network communication. This potentially allows the attack to be distributed over multiple hosts, where every host just carries out a small part, reducing visibility.

4.6. Attack Methods

This section discusses and categorizes attack methods. We consider all levels of complexity, but focus on APT-orchestrated attacks, as they are hardest to defend against.

Li et al. [7] point out that high-complexity attacks are expensive to develop and difficult to coordinate. APTs, however, are capable of investing substantial funds in the development of the required methods. As a consequence, defending against them becomes increasingly costly and difficult. Furthermore, the defender's response time may be slow due to the increased complexity in comparison to less sophisticated attacks. Multiple vulnerabilities lead to greater damage in a shorter time and even though such attacks are of a low probability, they are, according to Line et al. [4], not to be underestimated.

According to Bencsáth et al. [11], an adversary has the advantage of using off-the-shelf products and choosing from a number of methods to fine-tune an attack. Furthermore, cyber attacks are usually conducted anonymously, and, in the case of failure, there are few consequences for the attacker. However, defenders have to fend off all possible attacks, which leads to asymmetries of knowledge. Defenders can, in the best case, mitigate an attack, but never win against anonymous attackers.

We distinguish five general attack goals against networks and communication technology that will be elaborated on with examples of smart grid attacks in Section 6.2.

- *Disruption attacks* aim to suppress a service or the production of a commodity for the period the attack lasts, e.g., flooding a target with unsolicited queries. Such DDoS attacks usually cause no physical damage, yet can result in monetary loss or outages. Real-world examples are discussed in [7, 10, 28, 30, 41].

- *Destruction attacks* generally target industrial equipment, with the goal of destroying its infrastructure, effectively halting production for a prolonged period and necessitating replacement. We examined two well-known examples, namely the Stuxnet and Ukraine attacks [1, 2]. The authors behind Stuxnet targeted uranium enrichment centrifuges mechanically destroying them by manipulating revolution speeds. BlackEnergy3 managed to destroy Ukrainian power management equipment, leaving parts of the power grid without automatic-restore functionality. This attack was mounted in multiple stages. Legacy communication equipment and firmware was manipulated simultaneously, to prevent automatic recovery. The backup batteries were discharged and the hard drives deleted. Finally, essential power switches on the transformers were opened remotely, and a DDoS attack on the telephone hotlines prevented customers from reporting blackouts. Furthermore, this incident provides a blueprint for imitation.

- *Theft* likely ranks among the most common attack types according to descriptions in [23, 24, 26, 27, 38, 51]. They range across all sectors, from companies and industries to politics, academia, or military. Data theft (espionage) is typically conducted over long periods and with very low visibility.

- *Extortion schemes* have recently made a resurgence in the form of crypto-lockers. During such attacks host-devices are infected, all accessible files encrypted and all backups removed. Generally, local and remote files are affected, leading to a broad impact through shared network folders. Ransom is demanded from the victim to obtain the decryption key, as seen in the case of Locky [48, 50]. There are, however, several other possibilities for extortion of operators of critical infrastructures, as discussed in Section 6.2.

- *Repurpose attacks* change the behavior of a host from its intended function; for instance, infected hosts act as stepping stones or proxies to obfuscate the attacker's infection and C&C paths.

4.7. Trigger

Triggers for executing attacks may be hard-coded (internal) or remote-controlled (external). One particularly interesting example of a hard-coded trigger is the "Gödel" module in Gauss [11, 26], known for its highly targeted nature. This means that it can only decrypt and execute on hosts with a certain hardware and software setup. Triggers can further be divided into time-based, remote-triggered (C&C), version numbered, or autonomous triggers on e.g., successful target infection.

4.8. Cleanup

Malware authors may need to cover their tracks, for which a number of clean up functionalities have been developed, e.g., self-disinfecting removable drives [26], or self-removal after Time-to-Live (TTL) has expired. Concealing valuable technology by obfuscating or removing parts of the malicious code can significantly impede forensic analysis, as was seen in the case of Gödel [11, 26], which remains encrypted to this day.

5. Classification of Existing Malware Types

In this section we evaluate 19 representative instances of malware using metrics that correspond to the modules in the generic life-cycle model introduced in Section 4. Our samples include well-researched malware types from the Internet that represent a wide range of features relevant to smart grid attacks. Some types included in the sample are older but prevailing malware at various levels of sophistication. Others are modern, highly evolved APT-produced malware. We aim to provide insight into recent developments and extrapolate future threats, but are aware that there are many more samples available for analysis. Therefore, we have selected examples that represent the most important classes, showing variants of modern techniques and the most recent representatives. Since the primary difference between smart grids and the Internet is the former's predominant use of, e.g., M2M communication as well as the homogeneity of its devices, cf. Section 3.1, this selection takes into account lateral attack vectors via the office network into the control network. The large number of possible attacks allows us to infer similar behavior to that of Internet malware.

All metrics are processed in Tables 1 through 4 and sorted chronologically by year of detection. We provide an overview of the metrics below and reference the corresponding sections within the generic model.

- General information (Table 1)
  - Access vectors, cf. Section 4.1
  - Attack goals, cf. Section 4.6
  - References for all malware types
- Propagation (Table 2)
  - Propagation vectors, cf. Section 4.3
  - Payload types, cf. Section 4.3
- Communication patterns (Table 3)
  - Discovery methods, cf. Section 4.2
  - Network protocols, cf. Section 4.3
  - C&C methods, cf. Section 4.5
- Persistence and defense-mechanisms (Table 4)
  - Morphism level, cf. Section 4.4
  - Evasion of anti-malware tools, cf. Section 4.1
  - Memory residency, cf. Section 4.1
  - Code obfuscation, cf. Section 4.1
  - Service injection, cf. Section 4.1
  - Uninstall mechanisms, cf. Section 4.1
  - Rapid reinfection, cf. Section 4.1
  - Cleanup mechanisms, cf. Section 4.1

We further supplement the aforementioned metrics with the following four qualitative metrics:

- Level of complexity, cf. Table 1
- Propagation speed, cf. Table 2
- Propagation scope, cf. Table 2
- Modularity, cf. Table 2

In this comparison we generally disregard infection methods and thus the known vulnerabilities, as we could only speculate which vulnerabilities will emerge in future. The present examples show that only a few malware instances share the same vulnerabilities, which means they have access to a large set of possibilities. But exceptions can still be found in which new malware recycles known vulnerabilities or members of the same family take advantage of the same vulnerabilities, which is the case for most Stuxnet derivatives. Since we consider only well-researched malware types, the vulnerabilities used here can today be counteracted by keeping security implementations up-to-date. However, new vulnerabilities arise daily, and malware authors are becoming increasingly creative.

Table 1: General Malware Information

Name	Year of Detection	Level of Complexity	Access Methods	Attack Goal	References
CodeRed1	2001	Low	S2S	Disruption	[10, 28, 30]
CodeRed2	2001	Low	S2S	Disruption	[10, 28, 30, 41]
Nimda	2001	Low	Email, C2S, S2C, H2NS	Disruption	[10, 30]
Slammer	2003	Low	C2S, S2C	Disruption	[7, 41]
Sality	2003	Medium	H2NS, RD, P2P	Theft(Espionage)	[37, 38]
Conficker	2008	Low	C2C, P2P, RD	Theft (Espionage)	[33, 37, 51]
Regin	2008	High	P2P, C2C	Theft (Espionage)	[3, 23]
Aurora	2010	Medium	Email, S2C	Theft (Espionage)	[1, 27, 39]
Stuxnet	2010	High	RD, C2C, P2P	Destruction, Theft (Espionage)	[1, 39, 40]
Duqu	2011	High	Email, C2C, P2P, H2NS	Theft (Espionage)	[11, 24, 39, 49]
Flame	2012	High	C2C, RD	Theft (Espionage), Repurpose	[11, 25, 39]
Gauss	2012	High	RD	Theft (Espionage)	[11, 26, 39]
Duqu2	2014	High	Email, C2C, S2C	Theft (Espionage)	[11, 24, 39]
Equation	2014	High	C2C, RD, S2C	Theft (Espionage)	[27, 39]
BlackEnergy3	2015	High	RD, Email, C2C	Destruction, Theft (Espionage)	[2, 39]
Cozy Duke	2015	High	Email, S2C, H2NS	Theft (Espionage)	[31, 39]
AdWind	2015	High	Email	Theft (Espionage)	[32, 36, 39]
Locky	2016	Low	Email, H2NS	Extortion, Disruption	[48, 50]
PLC-Blaster	2016	Low	C2C	Destruction, Repurpose	[52, 53]

Legend: (P2P) Peer-to-Peer, (S2S) Server-to-Server, (S2C) Server-to-Client, (C2S) Client-to-Server, (C2C) Client-to-Client, (H2NS) Host-to-Network-Share, (RD) Removable Drive (non-network), (Email) Phishing emails through active dispatch or passive third party (several protocols).

5.1. General Malware Information

Section 4.1 and Table 1 describe detailed examples of access methods. Furthermore, the chronological evolution is a strong indication that recent malware development is shifting away from disruption attacks and rather focusing on espionage and data theft. However, it is worth noting that this evolution does *not* mean that DDoS attacks are no longer feasible or not of interest. The modular concept of today's malware allows for easy addition of new modules at any time. This means that DDoS capability could be added as part of an upgrade to existing malware.

Based on our investigation, we argue that future malware may increasingly have the capability of installing modules with the goal of physical destruction. Attacks that target cyber-physical systems, e.g., Stuxnet or BlackEnergy [1, 2] utilize extensive espionage capabilities along with a destructive payload. This helps the attacker to map the environment, as it has to get through the defenses of industrial networks. Scanning and password or credential theft are among the methods utilized. Technically, most sophisticated espionage malware only lacks modules for targeting cyber-physical equipment as well as certain protocols in order to become a threat to smart grids. This means that utilities are well advised to defend against espionage-class malware based on the suspicion that cyber-physical attack capabilities may be implemented in the future.

Extortion attacks have been highly present in recent news and are also feasible as smart grid attacks, as critical infrastructures provide strong leverage against communities. Extortion could therefore precede destruction-attacks, as a modular extension. Furthermore, modern attacks are increasing in complexity, making them more challenging to defend against. Some examples of malware show that the development of low-complexity and thus, sub-APT-level malware is still conducted for monetary gain [32, 36, 48, 50]. Although, they are not to be underestimated, development is generally moving in the direction of enabling more universal capabilities.

Based on Table 1, the following findings can be seen:

- Overall malware complexity is increasing substantially.
- Attack goals are shifting increasingly toward more complex espionage and away from disruption.
- Disruption attacks are still feasible as part of a modular upgrade in more complex attack environments or low-complexity efforts.
- Destruction attacks are expected to become more common in the near future.
- Sophisticated espionage malware currently lacks modules for cyber-physical attack capability and specific smart grid protocols to be deployed in smart grids. However, such features may be implemented in the future.
- Development is generally moving toward enabling more universal capabilities. Modularity enables fast upgrades.
- Access vectors are becoming more sophisticated and often depend on the availability of zero-day exploits.

5.2. Malware Propagation

Table 2 reviews malware network activity in terms of scope, speed, propagation vectors, payload construction, and modularity. The propagation scope, which in some cases is global, is shifting increasingly toward targeted attacks. The scope could be sub-categorized and refined in more detail but we omit any finer granularity as to targeted persons, companies, or states. The BlackEnergy and Regin attacks [2, 3, 6] demonstrated that all involved groups and persons were spied upon regardless of their level of protection. Attacks against smart grid operators

Table 2: Malware Propagation

Name	Scope	Speed	Vectors	Payload	Modularity
CodeRed1	Global	Aggressive	S2S	Self-Carried	None
CodeRed2	Global	Aggressive	S2S	Self-Carried	None
Nimda	Global	Aggressive	Email, C2S, S2C, H2NS	Self-Carried	None
Slammer	Global	Aggressive	C2S, S2C	Self-Carried	None
Sality	Global	Aggressive	H2NS, RD, P2P	Second-Channel	Low
Conficker	Global	Aggressive	C2C, P2P, RD	Self-Carried	Low
Regin	Targeted	Moderate	P2P, C2C	Second-Channel	Medium
Aurora	Targeted	Aggressive	Email, S2C	Second-Channel	Low
Stuxnet	Targeted	Moderate	RD, C2C, P2P	Second-Channel	Medium
Duqu	Targeted	Moderate	Email, C2C, P2P, H2NS	Second-Channel	Medium
Flame	Targeted	Moderate	C2C, RD	Second-Channel	High
Gauss	Targeted	Passive	RD	Embedded	High
Duqu2	Targeted	Moderate	Email, C2C, S2C	Second-Channel	High
Equation	Targeted	Passive	C2C, RD, S2C	Embedded	High
BlackEnergy3	Targeted	Moderate	RD, Email, C2C	Second-Channel	Medium
Cozy Duke	Targeted	Moderate	Email, S2C, H2NS	Second-Channel	Medium
AdWind	Targeted	Passive	Email	Second-Channel	High
Locky	Global	Passive	Email, H2NS	Second-Channel	Low
PLC-Blaster	Targeted	Moderate	C2C	Self-Carried	None

Legend: (P2P) Peer-to-Peer, (S2S) Server-to-Server, (S2C) Server-to-Client, (C2S) Client-to-Server, (C2C) Client-to-Client, (H2NS) Host-to-Network-Share, (RD) Removable Drive (non-network), (Email) Phishing emails through active sending or passive third party (several protocols).

will almost certainly be of a targeted nature, as utilities are expected to implement strong defenses. Yet, there remains the possibility that non-targeted malware finds its way into a utility companies' network as occurred in the nuclear power plant in Gundremmingen, Germany [63]. This incident, however, did not result in a cyber-physical attack thanks to well-implemented network segmentation.

Concerning the propagation speed, development is converging toward stealthy malware types. These accept penalties in propagation speed in order to improve their stealthiness and persistence. Equation and Gauss [26, 27] in particular stand out for their very slow spreading speed in support of sophisticated stealthiness. Then, on the other hand, CodeRed and Nimda [10, 28, 30, 41] propagate at incredible speeds, producing many anomalies that make them easily detectable. In between, however, there are many moderate types of malware, that prioritize stealthiness in the interest of prolonged persistence.

The propagation vectors of malware are a key feature in cyber attacks; here impeding metrics, e.g., scope and speed, play a crucial role. Propagation vectors correspond to the access methods in Table 1 and affect the movement of malware. Some only allow passive propagation via phishing emails, cf. AdWind and Locky [32, 48], or via removable drives, cf. Gauss and Equation [26, 27]. Many other types are capable of network-enabled self-propagation, which opens up many opportunities for rapid malware spread, e.g., Conficker [51]. The latter is capable of infecting other hosts via remote execution of a dropper file on networks using P2P networks, whereas Stuxnet [1] and Flame [25] partially utilize the same vulnerabilities, yet are more advanced in network propagation. They can also propagate via removable drives that allow access into air-gapped networks.

All propagation vectors have promising candidates for networks with homogeneous equipment types, as are expected in smart grids. Flame [25], for example, is capable of disguising itself as an update server while infecting hosts. Such a vector is essentially feasible with every update service. Given the fact that modern malware most often has a second-channel payload type, the increasing modularity fits in well as can be seen in Table 2. The analysis reveals that self-carried malware is rarely modular, whereas second-channel malware is predominantly modular. The payload types are discussed in detail in Section 4.3. Although there are still some self-carried payload types the trend is clearly moving toward second-channel types.

This comparison shows that modern highly complex and APT-made malware is increasingly modular and extensible. The number of functions seems to increase with every generation. Some modern malware types already have about 100 modules as exemplified by Duqu2 [24], whereas older malware types exhibit less flexibility. There are some exceptions, most notably PLC-Blaster [52], which is in an early stage of development yet highly specialized toward, in this case, industrial equipment. If such threats are incorporated into a feature-rich modular malware comparable to [24], power utilities will face highly challenging adversaries.

Based on Table 2, the following findings can be seen:

- Propagation scope is shifting toward targeted attacks.
- Payloads and increasing modularity are advancing stealthiness.
- Propagation speed is converging toward slower attacks with a stronger focus on sophisticated stealthiness.
- Propagation vectors include promising candidates for networks with homogeneous characteristics.
- Modular malware with second-channel payload types are now the state of the art.

While malware propagation is positively influenced by increasing speed, scope, and availability of attack vectors, these

Table 3: Malware Communication Patterns

Name	Discovery	Protocols	C&C Method	Distributed C&C
CodeRed1	Blind Scan	TCP	None	No
CodeRed2	Topological Scan	TCP	None	No
Nimda	Blind Scan	TCP, UDP	None	No
Slammer	Blind Scan	UDP	None	No
Sality	Distributed Blind Scan	TCP, UDP, HTTP	P2P	No
Conficker	Topological Scan	TCP, UDP, HTTP, SMB	DNS flux, HTTP, P2P	Yes
Regin	Topological Scan	TCP, UDP, HTTP, HTTPS, SMB, ICMP	HTTP, HTTPS, SMB	Yes
Aurora	None	TCP, HTTPS, TLS, Email	WinAPI	Yes
Stuxnet	Topological Scan	TCP, HTTP	P2P, WinAPI	No
Duqu	Topological Scan	TCP, HTTP, Email	P2P, Obfuscation [1]	No
Flame	Topological Scan	TCP, HTTP	HTTP, USB	No
Gauss	None	None	USB, HTTP (inactive)	No
Duqu2	Topological Scan	TCP, HTTP, HTTPS, Network pipes, Email	HTTP, Network Pipes, Mailslots, Obfuscation [1]	No
Equation	None	None	USB	Yes
BlackEnergy3	Topological Scan	TCP, HTTPS, Email	HTTPS	No
Cozy Duke	Topological Scan	TCP, HTTP, HTTPS, Email	HTTP, HTTPS	No
AdWind	None	HTTP, Email	TLS	No
Locky	None	HTTP, Email	IP flux, HTTP	Yes
PLC-Blaster	Topological Scan	TCP, S7CommPlus	TCO	No

Legend: (1) Obfuscating C&C among Standard traffic by choosing native protocols.

properties have a negative impact on all metrics that consider the stealthiness of malware. Detectability on the part of defenders therefore increases with the availability and successful application of anomaly detection that can find patterns and protocols used for propagation and for the C&C architecture.

5.3. *Malware Communication Patterns*

Table 3 provides a classification with respect to detectable communication patterns. The table first scrutinizes discovery methods (scanning) utilized for mapping. It furthermore illustrates that some malware types do not scan networks. The recent development is shifting toward less conspicuous methods including highly distributed scanning, advanced topological preference scanning or alternative channels that do not require network scanning. This development may be the result of the fact that companies have started to deploy advanced network anomaly detection.

Malware uses several network and transport-layer protocols, which are discussed in Section 4.2. However, according to Table 3, today's malware implementations show a clear preference for the use of TCP as transport protocol, moving away from UDP. There has also been a notable shift toward encrypted communication. Since anomaly detection is expected to perform well in homogeneous infrastructures, several pieces of malware investigated here obfuscate C&C messages among native network traffic. See Section 4.5 for details on Regin's [3] obfuscation capabilities. Monitoring for the following malware activities can aid in network-based anomaly detection:

- Forbidden communication attempts that highlight unsolicited connections which can be slowed or blocked.
- Scanning behavior, which indicates malware presence.
- Specific communication protocols that stand out in native traffic indicating the presence of malware.

Anti-malware tools can complement network-based detection with host-based methods when deployed on field devices.

We find it noteworthy that substantial differences in variants of C&C infrastructures are an indication that this may be a starting point for anomaly detection in predominantly homogeneous infrastructures. However, highly distributed structures as seen in [37, 38] may be difficult to detect. At this point, many malware instances still use standard protocols but encrypt all C&C traffic. Hence, anomaly detection may discover unsolicited traffic, which can be slowed or blocked. There are historic examples [64] in which covert channels were used for C&C, yet modern malware currently resorts to encryption instead of stealthiness tactics. The vast differences in the architectures and communication patterns of C&C structures seem to provide enough stealthiness to hide such traffic from typical heuristic comparison algorithms.

Based on Table 3, the following findings can be seen:

- Network discovery is shifting toward complex low-visibility, high-distribution, modern topological or no-scanning methods, and away from aggressive blind-scan and simple topological-scan methods.
- Malware increasingly obfuscates C&C communication streams among native network traffic.
- Instead of using covert techniques, malware communication is usually encrypted as the vastness of communication patterns seems to provide enough obfuscation that such traffic is overlooked.
- There is a clear preference for TCP over UDP.
- Although standard protocols are used, C&C traffic is mostly encrypted and obfuscated. Hence, modern anomaly detection may discover unsolicited traffic.

Table 4: Malware Persistence and Defense-Mechanisms

Name	Morphism	Evasion of Anti-Malware-Software	Memory Residency	Code Obfuscation	Service Injection	Uninstall Mechanism	Rapid Reinfection	Cleanup [4] Mechanism
CodeRed1	Monomorph	No	Yes	None	No	No	Yes	No
CodeRed2	Monomorph	No	No	None	No	No	No	No
Nimda	Monomorph	No	No	None	No	No	No	No
Slammer	Monomorph	No	No	None	No	No	Yes	No
Sality	Polymorph	Yes	Yes	Simple	Yes	No	No	No
Conficker	Polymorph	Yes	Yes	Complex [3]	No	Yes	Yes	No
Regin	Polymorph	No	No	Simple	No	No	No	No
Aurora	Polymorph	Yes	No	Simple	Yes	No	No	No
Stuxnet	Polymorph	Yes	Yes	Simple	Yes	Yes	No	Yes
Duqu	Polymorph	Yes	No	Simple	Yes	No	No	Yes
Flame	Polymorph	Yes	No	Simple	Yes	No	No	No
Gauss	Polymorph	Yes	No	Simple	Yes	Yes	No	Yes
Duqu2	Polymorph	Yes	Yes	Simple	No	No	Yes	No
Equation	Polymorph	Yes	Yes [2]	Simple	Yes	No	No	No
BlackEnergy3	Polymorph	Yes	No	Simple	Yes	No	No	No
Cozy Duke	Polymorph	Yes	No	Simple	Yes	No	No	No
AdWind	Metamorph [1]	Yes	Yes [2]	Complex [3]	No	Yes	No	Yes
Locky	Polymorph	No	No	Complex [3]	Yes	Yes	No	Yes
PLC-Blaster	Monomorph	No	No	None	No	No	No	No

Legend: (1) Utilize recompilation, padding, code obfuscation and multi-layer encryption. (2) Decrypt modules into memory or stores them in encrypted form on the hard drive. (3) Use complex multi-layer code obfuscation and encryption. (4) Impede malware detection and analysis.

- Anomaly detection is expected to perform well in homogeneous infrastructures to detect the selective use of protocols, forbidden communication, and scanning behavior.
- C&C is generally not widely distributed, although there are examples of it as well.

5.4. *Malware Persistence and Defensive Mechanisms*

Table 4 discusses the defense mechanisms of malware that increase its persistence against removal efforts. Morphism as discussed in Section 4.4 denotes the ability of a malware to autonomously change and adapt its appearance, including bit patterns in memory and for communications. A malware's morphism type can be used as a metric that characterizes its stealthiness and self-defense mechanisms. The payload structure has moved away from monomorphic toward polymorphic types, that represent by far the largest group in our sample. There is only one metamorphic example, i.e., AdWind [32, 36], which shows promising obfuscation features, but only in homogeneous environments. AdWind runs in JAVA and is capable of installing its native environment on multiple platforms. Future developments may, however, shift toward metamorphic types as AdWind is well known for its difficulty to detect and prolonged persistence although it uses no defense mechanisms other than recompilation and encryption on a local host.

Evasion of anti-malware tools was not a common feature in older malware. However, as malware's complexity increases, the huge effort spent in development is a high incentive to hide such evolved malware. Modern malware often evades and more recently even dismantles anti-malware tools by injecting malicious code into running system processes, effectively rendering them useless against a particular threat. Should a version of anti-malware tools be installed that cannot be counteracted, retreat and cleanup tactics are often used in order to protect the intellectual property such advanced malware represents.

Several malware types are also memory residents that are altogether unreachable for common anti-malware tools. There are a number of ways of residing in memory or even partially on the hard drive in encrypted form, decrypting only the parts required into memory.

Besides encryption, most modern malware also obfuscates its code, making it even less detectable by heuristic-based anti-malware tools. This behavior is characteristic for polymorphism (cf. Section 4.4), and goes as far as to use heavy multi-layer obfuscation with encryption and periodic re-iteration in metamorphism [32, 36]. This is an indication that malware authors do not yet perceive the need for such heavy defenses and represents an advantage to attackers, as the technology for obfuscation exists already but is not yet widely used.

Service injection enables malware to survive restarting the host, which is an indication that increasing malware complexity supports greater persistence. Most sophisticated types are capable of hijacking running drivers, maintaining their functionality while obtaining autostart capabilities. Duqu2 [24], as a notable example, shows no client-side persistence. Using an alternative approach, it resides in local servers and rapidly reinfects hosts after the boot sequence.

Finally, self-uninstall or clean-up mechanisms are rarely found. As discussed in Section 4.8 there are variants capable of disinfection upon triggers such as time triggers, or successful target detection, or on specific C&C commands.

Based on Table 4, the following findings can be seen:

- Development has moved away from monomorphism toward polymorphism. Some metamorphic types exist that are acknowledged for their prolonged persistence and may represent an indication of future developments [32].
- Increasing complexity in advanced malware provides an incentive to protect its intellectual property.

- The evasion of anti-malware tools is extensively implemented and should be considered to be the state of the art.
- Several malware types are capable of memory residency, increasing their persistence. However, there are other effective methods to achieve persistence using local storage.
- Traditional methods such as heuristics are becoming less efficient at detecting malware yielding their place to anomaly detection.
- Persistence through service injection must be considered to be the state of the art.
- Neither rapid reinfection nor cleanup mechanisms are yet widely deployed.

6. Future Evolution of Malware for Smart Grids

This section considers the evolution of Internet malware targeting smart grid environments. We expand on the perspective we opened in Section 3 and provide specific examples as to what future smart grid malware may look like, considering differences between Internet and smart grid communications. We first extract the specific novelties and strengths that existing types provide. In this way, we gain insights into functionalities that might be used in construction kits for future malware. Then, we examine trends and summarize features that we expect will continue to be present and evolve. We elaborate on attack vectors specific to future smart grid environments in Section 6.2 and discuss defensive strategies in Section 6.3. Table 6 maps the attack scenarios corresponding to these effective defensive measures. Based on this, we define three novel hypothetical malware types suited for smart grid environments.

The first step in designing novel advanced (hypothetical) malware for future smart grids is to determine the most highly evolved features that can be found in existing malware. These outstanding capabilities are summarized in Table 5. However, the investigated examples are not limited to these features and we expand on them for future attacks in the following sections.

Table 5: Distinct Strengths of Malware

Name	Distinct Strengths
CodeRed1	Small size (4 kbyte) and speed of spreading
CodeRed2	Designed to spread much faster than CodeRed1 using a localized scanning strategy
Nimda	Small size (60 kbyte) and speed of spreading
Slammer	Extremely small file size (404 bytes) and even higher speed of spreading than CodeRed2
Sality	Highly distributed scanning by > 3 million hosts
Conficker	Fixes vulnerabilities and uses them as backdoor
Regin	Only utilizes VPN between modules, even locally
Aurora	Watering hole attack on secondary targets, in order to gain access to primary target
Stuxnet	Physical destruction, stealthiness & prolonged persistence
Duqu	Infects internal processes to dismantle anti-malware tools
Flame	Hosts a fake Windows update-service in order to spread
Gauss	Silent on networks, highly covert, slow spreading
Duqu2	Rapid reinfection by unusual S2C-vector without any persistence on the hosts
Equation	Silent on networks, infects hard drive firmware
BlackEnergy3	Uses legal signatures, physical destruction
Cozy Duke	Watering hole attack to steal credentials
AdWind	Highly obfuscated, uses multi-layer encryption and utilizes local recompilation - metamorph
Locky	Obfuscation, encryption and disinfection features
PLC-Blaster	Selectively targets Siemens industry PCs

6.1. Malware Trends

Future *trends we expect to see* concern both the Internet and smart grids but we focus on the latter. These include:

- Decreasing importance of propagation speed to the benefit of stealthiness and prolonged persistence in networks.
- Evolution of malware to evade signature-based detection methods, e.g. by:
  - Metamorphism replacing modern polymorphism.
  - Sophisticated multi-layer encryption.
  - Complex code obfuscation.
  - Unobtrusive network scanning methods.
  - Obfuscation of propagation in native network-traffic and increased passive propagation.
  - Covert and/or distributed C&C communication.
- Increasing versatility and complexity of modern anti-detection mechanisms beyond signature-based detection to the benefit of persistence and stealthiness in order to defeat anti-malware tools through the use of:
  - Rapid local recompilation.
  - Sophisticated multi-layer encryption.
  - Complex code obfuscation.
  - Increased use of memory residency or other effective means to avoid anti-malware tools.
- Increasing modularization & customization adding flexibility to all areas of the generic model introduced in Section 4.
- Increasing targeted attacks, targeting cyber-physical systems due to their excellent potential as leverage against companies or communities.
- Increasing profitability of zero-day vulnerability research and increasing number and complexity of access and propagation vectors.

The list above includes several anti-detection methods. However, we also took into account an obfuscation method by Beunardeau et al. [65] which leaves the code's functionality computationally unchanged, yet unreadable for reverse engineering. Arguably, this has the potential to strongly impede research. Paired with known encryption and recompilation methods, sophisticated metamorphic malware may arise, or already be in use. Initially, this method was intended to protect against software piracy. It could however, protect malware code and create huge challenges for researchers and defenders if misused.

6.2. Attack Types Specific to Smart Grids

This section investigates potential attack methods specific to smart grid environments. As mentioned in Section 4.6, we dis-

Table 6: Mapping Applicable Security Measures to Attack Scenarios

	Attack Types (Section 6.2)																						
	Disruption				Destruction					Theft								Extortion			Repurpose		
Defense (Sections 3.3 & 6.3)	a	b	c	d	a	b	c	d	e	a	b	c	d	e	f	g	h	a	b	c	a	b	c
Baseline Security	✓	~	✓	✓	✓	~	~	~	~	~	~	~	~	~	~	✓	✓	~	✓	~	~	~	~
Reduction of Service	–	~	~	–	–	~	~	–	–	–	–	–	–	–	–	~	~	–	–	~	~	~	~
Anomaly Detection [1]	–	–	–	~	~	✓	~	~	~	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	~	✓	✓	✓
Physical Security	–	–	–	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	–	–	~	–	–
Removable Media	–	–	–	~	~	~	~	–	–	~	~	~	~	~	~	~	~	~	–	~	~	–	–
Trusted Environment	–	–	–	~	~	✓	~	~	~	✓	✓	✓	✓	✓	✓	✓	✓	~	–	~	✓	~	~
Content Filter and Firewall	✓	~	✓	~	~	~	~	–	–	~	~	~	~	~	~	~	~	~	✓	–	~	✓	~
Leakage Protection	–	–	–	–	–	✓	~	✓	~	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	~	–	–	–
Backup / Recovery	–	–	–	–	–	✓	✓	–	–	–	–	–	–	–	–	✓	✓	✓	–	✓	~	~	~

Legend: (✓) Effective, (~) Partially effective, (–) Ineffective, (1) Effective in combination with measures that are invoked upon attack detection.

tinguish between five general attack goals. Furthermore, we include examples that we consider feasible in the near future.

6.2.1. Disruption Attacks

This attack type aims to suppress a service for a certain period, leaving it unavailable for critical decision processes.

(a) *DDoS-attacks from outside targeting inside assets (Inbound attacks)*: Such attacks can result in delays, outages, and monetary loss. Examples have been discussed in [7, 10, 28, 30, 41]. Targets inside smart grids could be meter-data-management-servers, key-servers, email-servers, network-attached storage, or critical firewalls, among other things.

(b) *DDoS-attacks from inside targeting inside assets (Internal attacks)*: Such attacks can originate from compromised assets inside the utility network and can target central or higher level components e.g., control center. The attacks may result in outages, depending on the resilience of the system and fallback strategies in place.

(c) *DDoS-attacks from inside attacking targets outside (Outbound attacks)*: Compromised assets, e.g., smart meter networks, represent a substantial number of devices that can be used to run outbound DDoS-attacks.

(d) *DDoS-attacks on certain user groups (Selective harassment)*: Rather than targeting central assets to disrupt all service the selective harassment of user groups through abuse of remote-disconnect features on compromised smart meters can impact the reputation of utility companies.

6.2.2. Destruction Attacks

There are several precedents from the recent past, cf. Section 4.6, in which industrial equipment was targeted and destroyed, delaying production considerably. The affected process then has to be rebuilt. The difference between a disruption and a destruction is that once the attack ends (e.g. a DDoS) the equipment recovers and becomes reusable again, whereas in a destruction attack one needs to restart or even rebuild the infrastructure to continue operation. Potential use cases include:

(a) *Disconnect households:* Compromised smart meters with remote power-disconnect features may drop many households at once, aggravating cascading effects or blackouts in the power grid. Furthermore, power reconnect features can destabilize the power grid by oscillate-switching households and negatively impacting its stability.

(b) *Destroy power management:* Central power management and switching equipment can be attacked and destroyed, as demonstrated in the Ukraine incident [6]. Details about the access methods were discussed in Section 4.1 and the attack vectors in Section 4.6.

(c) *Influence critical electrical nodes in the Field:* The targeting of power-switching equipment in the field can be optimized as demonstrated in the examples devised by Wang et al. [66] and Rosas-casals et al. [67]. They introduce metrics for calculating critical electrical nodes, relieving effort on the attackers side. In combination with DDoS or targeted cyber attacks potent attack vectors are opened.

(d) *Modify sensor data:* Compromised sensors or man-in-the-middle attacks on sensor data may generate false decisions in the control center, leading to blackouts.

(e) *Tamper with clock synchronization:* The time synchronization and power measurement data on sensors (e.g., PMUs) can be manipulated to drift apart, triggering emergency switching [68].

6.2.3. Theft

Stealing a commodity, e.g., information, affects the target indirectly, e.g., by reputation. Information revealed to competitors or leaking critical information about topology or credentials can improve attack efficiency and preparation. Theft is among the most common attack types according to [1, 23, 24, 26, 27, 38, 51].

(a) *Espionage:* Eavesdropping on user data, e.g., power consumption or billing data, collect information on persons or groups, especially when complemented with Internet data.

(b) *Ruin credibility of users:* Legal actions may be forced upon victims, effectively occupying their time, e.g., if they become suspect after an attacker modifies meter readings or profiles.

(c) *Ruin reputation of providers:* Satisfaction ratings of utility companies can be negatively impacted by stealing information and manipulating billing.

(d) *Ruin reputation of manufacturers:* Satisfaction ratings of manufacturers can be affected via data manipulation and information theft, leading utility companies to reconsider future investments.
(e) *Sell long term data:* Historical consumption data can be sold for statistics, e.g., to competitors.
(f) *Sell live data:* Live energy consumption data can be sold for home intrusion optimization.
(g) *Market manipulation:* Energy markets can be manipulated by compromising smart meter networks with switchable loads through exfiltrated information.
(h) *Bill manipulation (Attack as a service):* Attackers can offer reduced energy bills as a service, effectively manipulating billing with false energy data.

6.2.4. Extortion Schemes

Extortion attempts aim to demand ransom for releasing a captured commodity or service [48, 50].

(a) *Threat of destruction:* Compromised smart meters harbor the potential to affect power grids. Extortion of utility companies may be used as a prerequisite for destruction attacks, yet may also reveal the attackers' presence.
(b) *Threat of DDoS:* DDoS attacks can nowadays be accompanied by ransom notes. In the case of critical infrastructures cascading effects may serve as excellent leverage.
(c) *Crypto-locker:* The infiltration of crypto-locker malware into utility networks may lead to devastating data loss through the encryption of important files.

6.2.5. Repurpose Attacks

Repurposing includes all methods that change the behavior of a host from its intended function.

(a) *Fake servers:* Infected hosts can act as fake update servers as demonstrated by Flame [25], in order to increase the spreading of the infection. This method is also conceivable in smart grid networks.
(b) *Proxies:* Infected hosts can be used as proxies to obfuscate C&C infrastructures.
(c) *Distributed computing:* Compromised smart meter networks host numerous devices that can be re-purposed for distributed computing (e.g., Bitcoin mining, or brute forcing hashes).

Most attacks mentioned above, are hypothetical yet are based on the increasing complexity of malware technology, cf. Tables 1 through 4. Sophisticated stealthiness and persistence can be highly effective in mapping targets. Extensive espionage therefore usually precedes cyper-physical attacks [1, 2, 40, 67].

6.3. Security Measure Extensions for Smart Grids

It is worthy noting that ICT architecture plays a crucial role in terms of overall system resilience [35, 66]. This is in particular true for critical systems such as power grids and future smart grid implementations that utilize communication networks. Alone, however, secure architectures and baseline measures (cf. Section 3.3) will not suffice to repel attackers. This list complements the previously established security model. We aim to provide solutions for the above attack scenarios, based on measures suggested in [14, 18–22]. Extended security measures that must be implemented in order to protect critical infrastructures include:

- *Reduction of services*: Services should be restricted to applications that are necessary in the respective network segment, e.g., office networks support email, print-servers, or IP-telephony, but these are not required in industrial control networks and should therefore be omitted. Reducing the number of active services decreases the chance for vulnerabilities and aids in the detection of anomalies as a supporting measure. Therefore, this measure is only partially effective in our qualitative assessment in Table 6.
- *Anomaly detection*: Anomaly- and event correlation complement other reactive measures and can detect suspicious network behavior, e.g., network scanning, C&C or unsolicited traffic within native traffic. It can provide better results compared to heuristic-based detection and is especially effective against stealthy attacks.
- *Physical security*: Access to field devices is possible at any given time. However, physical security should be established according to technological feasibility as it is often used during an early part (initial access) of a larger attack.
- *Removable media*: The bridging of air-gaps across critical environments should be prevented by disabling / reducing the number of interfaces. Similar to restricted physical access this measure decreases the attack surface, yet cannot alone stop cyber attacks.
- *Trusted environment*: Key management, signature verification and multi-factor authentication establish trusted environments that are under the control of utility companies. They can effectively hinder attackers from entering or propagating in a critical network (cf. Table 6).
- *Extended firewall use*: Inbound and outbound application and content filtering improves trusted environments.
- *Leakage protection*: This ensures that no data is leaked by means of active integrity checks, periodic vulnerability analysis, and anomaly detection.
- *Data backup and fallback strategies*: Several stages of short-term and redundant long-term backups can preserve working environments for fallback and reconstruction processes. These measures are effective in case attackers delete information or destroy important devices.

7. Three Types of Smart Grid Enabled Malware

In this section we propose three hypothetical malware types that are optimally suited for smart grid attacks. They are intended as examples to serve utility companies and researchers in their efforts to develop proactive and reactive security measures for critical infrastructures.

The classification relies on three main sources. We first consider analogies to the works of Li et al. [7], which proposed the establishment of a Cyber Center for Disease Control. We then use definitions and terms from the Centers for Disease Control and Prevention (CDC) [69] in order to draw analogies between the spreading of malware in cyberspace and the spreading of diseases in the real world. Lastly, we draw from Staniford et al. [10], who propose the "contagion" model that hardly leaves a trace for detection mechanisms to pick up on.

7.1. Pandemic Malware

A pandemic is defined [69] as the rapid spreading of a disease across an extensive geographic space (across the globe), usually affecting many individuals. We draw an analogy from this example to the realm of malware and reuse this term to refer to the first category of smart grid malware. This group is generally composed of rapidly spreading malware, with the objective of aggressive propagation. The main goal of pandemic malware is the almost immediate infection of large areas of the network before utilities can adopt countermeasures.

According to our investigations in Section 5, aggressive and noisy malware were preferred in the past, but recent developments have shifted towards stealthy, highly complex and modular types. Although techniques of rapid spreading are not obsolete, modern detection mechanisms can today better identify and counteract them. Strong security implementations likely leave little attack surface for such malware. Monocultures, however, increase propagation speeds and could therefore encourage a new generation of aggressive malware. In combination with hypothetical widely available zero-day vulnerabilities, such aggressive malware harbors the potential to spread quickly. This could have catastrophic consequences if critical infrastructures are affected. Zero-days may, however, be too expensive for malware of such a low sophistication level. Yet delayed updating of recently discovered vulnerabilities typically leaves a window of opportunity.

Characteristics native to pandemic malware types include noisy scanning methods such as blind scanning or topological scanning (Section 4.2) as well as less sophisticated payload construction such as monomorphism (Section 4.4). The evolution of malware shows that such globally acting aggressive types are often optimized for speed and do not implement much if any modularity, decreasing its overall complexity. We consider pandemic malware to be viable in the Internet landscape with a number of widely deployed services, but less likely to be so in a smart grid. Although it may find its way into a smart grid control network through improperly configured security mechanisms, modern heuristic detection should be capable of detecting such attacks.

References and Significant Features. Real-world malware exhibiting features relevant to pandemic malware include: Code Red 1 and 2 [28, 30, 41], Nimda [10, 30], Slammer [7, 41], and Conficker [33, 37, 51].

The following list summarizes the most significant features:

- High propagation speeds at the expense of stealthiness.
- Aggressive scanning methods.
- Self-carried payload type and a simple, monomorphic architecture.
- Few propagation vectors predominantly in homogeneous topologies.
- Simple or older vulnerabilities.
- Presumably optimized for extortion or disruption attacks because espionage requires much greater stealthiness.
- No modularity, low complexity, therefore low investments.

Countermeasures.

- Regular security updates prevent low-complexity attacks by closing known vulnerabilities.
- High analytic speed in modern reactive security measures are of relevance due to the extremely high speed of infection rates expected.
- Heuristic detection should suffice to contain such types.
- Building smart grids as monocultures should be avoided.
- Strong resilience measures help mitigate attacks.
- Fallback systems help mitigate attacks.
- Emergency restoration methods help mitigate attacks.
- Strict firewall rules with application white-listing decreases the versatility of malicious environments.
- Network segmentation prevents propagation.
- Strict access management confines users to specific controlled environments.

7.2. Endemic Malware

An endemic refers to the constant presence and prevalence of a disease in a population within a geographic area [69]. We use this term to name the next category of malware that behaves more stealthily than pandemic malware.

Endemic malware types are modular, polymorphic, and have multiple propagation vectors. Modularity provides stealthiness features by minimizing the footprint depending on local conditions and allows for a multitude of payload types with various attack goals. Even though modules for smart grid devices have not yet been seen in the wild, one must assume they will exist in the future. Furthermore, modularity at a smart meter scale can, in part, overcome the challenges of targeting heterogeneous architectures through the use of multiple modules.

Various encryption and code obfuscation methods, e.g., polymorphism, augment defensive features and various propagation vectors increase reach, especially across air-gaps. Most endemic malware types considered in Section 5 are capable of network scanning and propagation, yet utilize discovery methods that do not produce an excessive amount of noise such as hitlist-, or distributed scanning. They are, however, detectable by modern reactive measures, which should spur utilities toward implementing anomaly detection.

Our samples exhibit many persistence mechanisms. They are, however, sophisticated in the sense that endemic malware is difficult to purge from a network. Examples are discussed in Table 4, in which we argue that evasion of anti-malware tools, memory residency, code obfuscation, multi-layer encryption, and reinfection are the most evolved features in this category.

References and Significant Features. This group represents generally stealthy, modular and persistent malware, such as Sality [37, 38], Regin [23], Aurora [1, 27], Stuxnet [1, 40], Duqu 1 and 2 [11, 24, 49], Flame [11, 25], BlackEnergy3 [2], and CozyDuke [31]. The following list summarizes the most significant features of endemic malware:

- Sacrifice of propagation speed for increased stealthiness.
- Highly developed modularity allowing for a smaller footprint and adding some stealthiness.
- Polymorphism, which increases the stealthiness through code obfuscation and encryption.
- Multiple scanning methods allowing less conspicuous discovery.
- Multi-vector propagation (using zero-day vulnerabilities).
- Sophisticated persistence mechanisms such as detection-evasion, code-injection, or memory-residency.

Countermeasures. This malware type presumes that all measures discussed in the pandemic model are included and supplemented by the following:

- Reactive measures, e.g., anomaly detection and event correlation, that avoid the drawbacks of heuristic detection.
- Permanent network segmentation and strict firewall rules to prevent straight forward propagation.
- Content filtering to prevent host infections via, for instance, watering hole attacks or other drive-by downloads.
- Social engineering education to prevent unwanted access.

7.3. Contagion Malware

The contagion malware type builds upon the concepts of Staniford et al. [10] and propagates in a manner that is hardly or not at all traceable by network-detection mechanisms. Such methods either include removable drives or highly obfuscated channels appended onto legitimate traffic. There are few suitable real world examples; for this reason, we include all malware types that are notoriously hard to detect.

References and Significant Features. The following list summarizes the most significant features of contagion malware. Some features are recycled ones that contagion malware builds upon, taking stealthiness and persistence features to a new level. Real world examples include Gauss [11, 26, 39], Equation [27, 39], and AdWind [32, 36, 39]. The main features of contagion malware include:

- Highly developed modularity adding stealthiness and minimizing the footprint on networks and hosts.
- Multiple covert network scanning methods or captured network-information from the infected host.
- Multi-vector covert propagation via zero-day vulnerabilities in an embedded payload.
- Covert propagation via non-network channels, e.g., hidden sectors on removable drives.
- Sacrifice of even more speed for increased stealthiness compared to endemic malware.
- Metamorphism and therefore highly sophisticated stealth features, encryption, obfuscation, and recompilation, further increasing persistence.
- Other sophisticated persistence mechanisms, including detection evasion, memory residency, service injection, or cleanup mechanisms.
- Firmware infection for added persistence against detection, even beyond host system recovery.

Countermeasures. This malware type presumes that all measures discussed in the pandemic and endemic model are included and supplemented by the following:

- Anomaly detection and event correlation extend across multiple segmented networks.
- Permanent network segmentation with respect to the type of service used along with anomaly detection, i.e., only specific services may run inside smart grid VPNs.

7.4. Comparison of Smart Grid Enabled Malware

Unlike the pandemic model, the endemic and contagion models share one common characteristic. They are hard to detect, which gives them an advantage in terms of persistence at the cost of speed. Table 7 applies the metrics from Section 5 onto the newly introduced smart grid malware types and lists countermeasures discussed in the aforementioned sections.

8. Conclusion and Future Work

We investigated 19 types of malware with respect to metrics introduced in our generic cyber attack model, i.e., propagation mechanisms, detectability, targets, persistence, and countermeasures. We then analyzed their relevance to smart grid attacks and provided 3 hypothetical classes of malware that in future could target smart grids, namely pandemic, endemic, and contagion malware. Our aim is to raise awareness on the defenders' side of the need to build defenses against, at the very least, the newly introduced endemic malware classes. Since attackers only need to exploit one particular vulnerability, while the defenders have to defend against all potential vulnerabilities, an asymmetry in resources and knowledge is apparent.

As discussed in Section 6, the most significant trends in future malware creation have been moving toward stealthiness at the cost of speed. Also as expected, modern malware shifts

Table 7: Three Types of Smart Grid Enabled Malware - Significant Features & Countermeasures

	Metric	Pandemic	Endemic	Contagion
	Complexity	Low	Medium	High
Propagation	Speed	High	Medium	Low
Propagation	Scope	Global	Targeted	Targeted
Propagation	Payload	Self-carried	Second-channel	Embedded
Propagation	Vectors	Few	Many	Many
Propagation	Modularity	None	Modular	Modular
Detection	Scanning	Aggressive	Stealth	None
Detection	Transmission	Any	Stealth	Covert
Detection	C&C	Any	Stealth [1]	Covert
Detection	Morphism	Monomorph	Polymorph	Metamorph
Defensive Measures	Evasion of Anti-Malware-Tools	No	Yes	Yes
Defensive Measures	Memory Resident	No	Maybe	Yes
Defensive Measures	Obfuscation	No	Yes	Yes
Defensive Measures	Encryption	No	Yes	Yes [2]
Defensive Measures	Service Injection	No	Yes	Yes
Defensive Measures	Reinfection	No	Yes	Yes
Defensive Measures	Cleanup / Uninstall	No	No	Yes
Target	Unpatched Vulnerabilities	Yes	Yes	Yes
Target	Zero-day	No	Yes	Yes
Countermeasures	Security Updates	Yes	Yes	Yes
Countermeasures	Heuristic Detection	Yes	Yes	Yes
Countermeasures	Avoid Monocultures	Yes	Yes	Yes
Countermeasures	Resilience Measures [3]	Yes	Yes	Yes
Countermeasures	Fallback Systems	Yes	Yes	Yes
Countermeasures	Emergency Restore	Yes	Yes	Yes
Countermeasures	Anomaly Detecion	Yes	Yes	Yes
Countermeasures	Strict Firewall Rules	Yes	Yes	Yes
Countermeasures	Access Management	Yes	Yes	Yes
Countermeasures	Content Filtering	No	Yes	Yes
Countermeasures	Social Engineering Education	No	Yes	Yes
Countermeasures	Network Segmentation & Event Correlation	No	Yes	Yes
Countermeasures	Network Segmentation to Type of Service	No	No	Yes

Legend: (1) Encryption and obfuscation, (2) Encryption and recompilation, (3) E.g., redundancies, secure topologies or fallback-strategies.

away from simple, aggressive types towards complex, modular and sophisticated ones with a considerable set of capabilities. Recently, such malware types have been heavily involved in espionage campaigns. However, there are also several precedents for cyber-physical attacks and an increasing trend toward highly versatile malware.

The complexity is ever increasing as demonstrated by the evolution of members of the same malware family within few years – for instance Stuxnet (6 modules) and Duqu2 (> 100 modules). Duqu2, among others, has gained many functions including new exploits, espionage-, persistence-, and propagation methods. The addition of one single module could enable such real world examples to launch a cyber-physical attack. The needed technology being readily available, we consider this trend to be of particular concern for utility companies, and encourage them to raise the bar for attackers.

Pandemic malware (cf. Section 7.1) with its simple attack methods may be defeated by standard IT-security measures found in best-practice guidelines, c.f. Section 3.3. Endemic types, however, require more scrutiny in order to be defended against. This can include anomaly detection and specific employee training, cf. Section 7.2. Furthermore, we consider endemic malware to be the current state of the art and therefore the most immediate threat. Contagion malware, cf. Section 7.3, is an even greater challenge due to its increased stealthiness and persistence. It may in part be counteracted with the same technologies required for the defense against endemic types. In addition to network segmentation, anomaly detection and honeypots could be feasible approaches.

Finally, zero-day vulnerabilities offer an attack surface utility companies cannot be expected to counteract. However, in addition to proper update- and segmentation policies, anomaly detection at several checkpoints and strong organizational processes may prevent social engineering and sabotage. Appending on those security measures, we urge utility providers to diversify their stock of devices and security implementations, i.e., to ensure heterogeneity in order to mitigate malware spreading that exploits one specific vulnerability.

Summarizing, the main findings of this paper are:

- The *generic life-cycle model* formalizes the stages of malware-based cyber attacks and enables us to investigate existing malware by dissecting it into recurring cycles (cf. Section 4). This allows a detailed comparison of characteristics in existing malware and provides a useful basis for predicting future developments.
- The *investigation of existing malware* shows a clear trend toward increasingly stealthy cyber attacks, with the goal of infiltrating highly secure networks (cf. Section 5). This trend should be a warning for utility companies to extend their defensive capabilities.
- The *prediction of potential future attacks* (cf. Section 6.2) based on the analysis of existing malware leads to three conceptual models (cf. Section 7), which can be used as basis to develop defensive strategies.
- The *mapping of security measures* to all future attack vectors assesses the usefulness in mitigating specific attacks by a qualitative assessment (cf. Tables 6 and 7). Furthermore, the mapping identifies where gaps still exist that can be exploited by future malware.

Future requirements beyond this work include:

- *Adjustment:* Knowledge collected in future attack incidents has to be incorporated in predictions of upcoming malware. For instance if malware with a novel spreading or scanning method is observed, this information has to be shared to prepare defense strategies for future threats.
- *Information sharing:* Cooperation is an important defense strategy. Incidents should be reported. Detailed information about attacks and attack preparation should be shared in the community. It may be required to provide incentives and technical solutions for controlled data sharing or even to enforce it to resolve concerns in organizations about sharing information with potential competitors.

- *Lightweight detection:* New lightweight detection methods are required to separate legitimate from illegitimate network traffic without exceeding resource demands and costs. They should guarantee long term functionality and maintenance in devices that operate over periods of more than 10 years.
- *Hiding techniques:* Malware may use sophisticated hiding techniques such as covert channels to prevent detection of malware communication. Future detection systems need to prepare for this and should incorporate covert channel detection methods.
- *Cyber-physical systems research:* Further research into a holistic view on the coherences between cyber-physical systems provide better understanding of the processes to minimize the attack surface and harden them against attacks. Here especially the interrelations between different systems is relevant for potential attack spreading.

Acknowledgments

This work was supported by the research projects URBEM funded by Wiener Stadwerke and SIMULTAN funded by the Austrian Research Funding Association (FFG).

References

[1] A. Matrosov, E. Rodionov, D. Harley, Stuxnet under the Microscope, Final Report No. 1.31, eset (Jan. 2011).

[2] F-Secure Labs, Blackenergy & Quedagh - The Convergence of Crimeware and APT Attacks, White Paper No. 1030745, F-Secure Labs, Helsinki (Sep. 2014).

[3] Kaspersky Labs, The Regin Platform: Nation-State Ownage of GSM Networks, White Paper, Moscow (Nov. 2014).

[4] M. Line, A. Zand, G. Stringhini, R. Kemmerer, Targeted Attacks against Industrial Control Systems: Is the Power Industry Prepared?, in: SEGS '14 Proceedings of the 2nd Workshop on Smart Energy Grid Security, ACM Press, Arizona, US, 2014, pp. 13–22. doi:10.1145/2667190.2667192.

[5] National Cybersecurity and Communications Integration Center, Industrial Control Systems Cyber Emergency Response Team, ICS-CERT Monitor, White Paper, Department of Homeland Security, USA (Oct. 2013).

[6] R. Lee, M. Assante, T. Conway, Analysis of the Cyber Attack on the Ukrainian Power Grid, A Defense Use Case, E-ISAC Electricity Sector Information Sharing & Analysis Center (Mar. 2016).

[7] P. Li, M. Salour, X. Su, A Survey of Internet Worm Detection and Containment, Communication Surveys and Tutorials 10 (1) (2008) 20–35. doi:10.1109/COMST.2008.4483668.

[8] I. A. Saeed, A. Selamat, A. M. A. Abuagoub, A Survey on Malware and Malware Detection Systems, International Journal of Computer Applications 67 (16) (2013) 25–31. doi:10.5120/11480-7108.

[9] C. Bruno, L. Guidi, A. Lorite-Espejo, D. Pestonesi, Assessing a Potential Cyberattack on the Italian Electric System, IEEE Security and Privacy 13 (5) (2015) 42–51. doi:10.1109/MSP.2015.99.

[10] S. Staniford, V. Paxson, N. Weaver, How to 0wn the Internet in Your Spare Time, in: USENIX Security Symposium, San Francisco, 2002, pp. 149–167.

[11] B. Bencsáth, G. Pék, L. Buttyán, M. Félegyházi, The Cousins of Stuxnet: Duqu, Flame, and Gauss, Future Internet 4 (4) (2012) 971–1003. doi:0.3390/fi4040971.

[12] O. Thonnard, L. Bilge, G. O'Gorman, S. Kiernan, M. Lee, Industrial Espionage and Targeted Attacks: Understanding the Characteristics of an Escalating Threat, in: D. Balzarotti, S. J. Stolfo, M. Cova (Eds.), Research in Attacks, Intrusions, and Defenses, no. 7462 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 64–85, dOI: 10.1007/978-3-642-33338-5_4.

[13] Lockheed Martin Corporation, Seven Ways to Apply the Cyber Kill Chain with a Threat Intelligence Platform, White Paper PIRA# CMK201507003 (2015).

[14] K. McLaughlin (QUB), Smart Grid Protection Against Cyber Attacks (SPARKS): High Level Design Documentation and Deployment Architecture for Multi-Attribute SCADA Intrusion Detection Systems, Project Deliverable D4.1 Contract No. 608224, AIT Austrian Institute of Technology (Aug. 2015).

[15] B. Schneier, Attack Trees: Modeling Security Threats, Dr. Dobb's Journal 24 (12).

[16] D. Gollmann, P. Gurikov, A. Isakov, M. Krotofil, J. Larsen, A. Winnicki, Cyber-Physical Systems Security: Experimental Analysis of a Vinyl Acetate Monomer Plant, in: Proceedings of the 1st ACM Workshop on Cyber-Physical System Security, CPSS '15, ACM, New York, NY, USA, 2015, pp. 1–12. doi:10.1145/2732198.2732208.

[17] O. Trullols-Cruces, M. Fiore, J. Barcelo-Ordinas, Understanding, Modeling and Taming Mobile Malware Epidemics in a Large-Scale Vehicular Network, in: 14th International Symposium and Workshops on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEE, Madrid, 2013, pp. 1–9. doi:10.1109/WoWMoM.2013.6583402.

[18] T. Zseby, Is IPv6 Ready for the Smart Grid?, in: 2012 International Conference on Cyber Security (CyberSecurity), Fraunhofer Inst. for Open Commun. Syst. (FOKUS), Washington, DC, 2012, pp. 157–164. doi:10.1109/CyberSecurity.2012.27.

[19] NIST, Guidelines for Smart Grid Cybersecurity, Tech. Rep. IR 7628r1, dOI: 10.6028/NIST.IR.7628r1 (Sep. 2014).

[20] Federal Office for Information Security (BSI), Protection Profile for the Security Module of a Smart Metering System, Common Criteria Protection Profile BSI-CC-PP-0077-2013, Germany (Dec. 2013).

[21] Federal Office for Information Security (BSI), Protection Profile for the Security Module of a Smart Meter Gateway, Common Criteria Protection Profile BSI-CC-PP-0077-V2-2015, Germany (Jan. 2015).

[22] Australian Signals Directorate (ASD): Cyber Security Operations Center, Strategies to Mitigate Targeted Cyber Intrusions, Attack Mitigation Strategy, Australian Government: Department of Defence (Feb. 2014).

[23] Symantec, Regin: Top-Tier Espionage Tool Enables Stealthy Surveillance, White Paper, Symantec (Nov. 2014).

[24] Kaspersky Labs, The Mystery of Duqu 2.0 a Sophisticated Cyberespionage Actor returns, White Paper, Moscow (Jun. 2015).

[25] Kaspersky Labs, The Flame: Questions and Answers (May 2016). URL `https://securelist.com/blog/incidents/34344/the-flame-questions-and-answers-51/`

[26] Kaspersky Labs, Gauss: Nation-State Cyber-Surveillance meets Banking Trojan (May 2016). URL `https://securelist.com/blog/incidents/33854/gauss-nation-state-cyber-surveillance-meets-banking-trojan-54/`

[27] Kaspersky Labs, Equation Group: Questions and Answers, White Paper Report No. 1.5 #EquationAPT, Moscow (Feb. 2015).

[28] D. Moore, C. Shannon, J. Brown, Code-Red: A Case Study on the Spread and Victims of an Internet Worm, in: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment, ACM SIGCOMM/USENIX Internet Measurement Workshop, Marseille, France, 2002, pp. 273–284.

[29] J. Schultz, Watering Hole Attacks an Attractive Alternative to Spear Phishing, on Cisco Security Blogs (2016). URL `http://blogs.cisco.com/security/watering-hole-attacks-an-attractive-alternative-to-spear-phishing`

[30] M. K. Chavan, P. V. Madane, Modelling and Detection of Camouflaging Worms - A Survey, International Journal of Emerging Technology and Advanced Engineering 2 (10) (2012) 564–569.

[31] F-Secure Labs, CozyDuke, White Paper No. 1030745, F-Secure Labs, Helsinki (2015).

[32] V. Kamluk, A. Gostev, Adwind - A Cross Plattform RAT, White Paper V. 3.0 #Adwind, Kaspersky Labs (Feb. 2016).

[33] J. Faulhaber, D. Felstead, H. Henry, J. Jones, Microsoft Security Intelligence Report - Zeroing in on Malware Propagation Methods (Jun. 2011).

[34] K. Kruglov, Security Policies: Misuse of Resources, Kaspersky Labs (May 2016). URL `https://securelist.com/blog/security-policies/`

35945/security-policies-misuse-of-resources/

[35] P. Eder-Neuhauser, T. Zseby, J. Fabini, Resilience and Security: A Qualitative Survey of Urban Smart Grid Architectures, IEEE Access 4 (2016) 839–848. doi:10.1109/ACCESS.2016.2531279.

[36] Idiom, Binaryforest - AlienSpy Java Rat Overview (2016). URL http://blog.idiom.ca/2015/03/alienspy-java-rat-overview.html

[37] A. Dainotti, A. King, K. Claffy, F. Papale, A. Pescape, Analysis of a "/0" Stealth Scan From a Botnet, IEEE/ACM Transactions on Networking 23 (2) (2015) 341–354. doi:10.1109/TNET.2013.2297678.

[38] A. Thigpen, E. Chien, W32.Sality, Security Response, Symantec (2013). URL https://www.symantec.com/security_response/writeup.jsp?docid=2006-011714-3948-99

[39] Kaspersky Labs, Targeted Cyberattacks Logbook (May 2016). URL https://apt.securelist.com/#secondPage

[40] N. Falliere, L. O. Murchu, E. Chien, W32. Stuxnet Dossier, White Paper, Security Response, Symantec Vol. 5 (2011) p. 6.

[41] G. Riley, M. Sharif, W. Lee, Simulating Internet Worms, in: The IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2004. (MASCOTS 2004). Proceedings, IEEE, 2004, pp. 268–274. doi:10.1109/MASCOT.2004.1348281.

[42] C. Roger, Port Scanning Techniques and the Defense Against Them, White Paper, SANS Institute (Oct. 2001).

[43] K. Y. Tan, How can attacker use ICMP for reconnaissance?, SANS - Information Security Resources (Apr. 2016). URL https://www.sans.org/security-resources/idfaq/how-can-attacker-use-icmp-for-reconnaissance/3/13

[44] T. Chown, IPv6 Implications for Network Scanning, Request for Comments RFC5157, IETF (Mar. 2008).

[45] Shodan, IoT Search Engine (Jun. 2016). URL https://www.shodan.io/

[46] G. Ollmann, Blocking Shodan (Jan. 2016). URL https://ipv6.net/news/blocking-shodan/

[47] J. Ullrich, Targeted IPv6 Scans Using pool.ntp.org . (Jun. 2016). URL https://isc.sans.edu/forums/diary/Targeted+IPv6+Scans+Using+poolntporg/20681/

[48] F. Sinitsyn, Locky: The Encryptor Taking the World by Storm, Kaspersky Labs (May 2016). URL https://securelist.com/blog/research/74398/locky-the-encryptor-taking-the-world-by-storm/

[49] The Laboratory of Cryptography and System Security (CrySyS), W32.Duqu - The Precursor to the next Stuxnet, White Paper, Symantec Security Response (Nov. 2011).

[50] AV-TEST Threat Research Team, Locky/ Dridex/ Cryptolocker Analysis, White Paper, AV-Test, AV-TEST (Feb. 2016).

[51] S. Shin, G. Gu, N. Reddy, C. P. Lee, A Large-Scale Empirical Study of Conficker, IEEE Transactions on Information Forensics and Security 7 (2) (2012) 676–690. doi:10.1109/TIFS.2011.2173486.

[52] R. Spenneberg, M. Brüggemann, H. Schwartke, PLC-Blaster: A Worm Living Solely in the PLC, in: Black Hat Asia 2016, Singapore, 2016, p. 16.

[53] E. Kovacs, PLC-Blaster Can Pose A Serious Threat to Industrial Networks, SecurityWeek (May 2016). URL http://www.securityweek.com/plc-worms-can-pose-serious-threat-industrial-networks

[54] National Institute of Standards and Technology, National Vulnerability Database, Automating Vulnerability Management, Security Measures and Compliance Checking (Mar. 2016). URL https://nvd.nist.gov/

[55] Mitre Corporation, Common Vulnerabilities and Exposures (CVE) - The Standard for Information Security Vulnerability Names (Mar. 2016). URL http://cve.mitre.org/index.html

[56] P. Ször, P. Ferrie, Hunting for metamorphic, in: Virus Bulletin Conference, CiteSeerX, 2001, pp. 123–144.

[57] M. Roesch, Snort: Lightweight Intrusion Detection for Networks., in: LISA '99: 13 th Systems Administration Conference, Vol. 99, Washington, Seattle, 1999, pp. 229–238.

[58] J. Nazario, Blackenergy DDoS Bot Analysis, White Paper, Arbor Networks (Oct. 2007).

[59] W. Mazurczyk, L. Caviglione, Information Hiding as a Challenge for Malware Detection, IEEE Security Privacy 13 (2) (2015) 89–93. doi:10.1109/MSP.2015.33.

[60] E. Davies, Recommendations for Filtering ICMPv6 Messages in Firewalls, Request for Comments RFC4890, IETF (May 2007).

[61] N. Freed, Behavior of and Requirements for Internet Firewalls, Request for Comments RFC2979, IETF (Oct. 2000).

[62] M. Kaczmarek, Malware Instrumentation Application to Regin Analysis, Le journal de la cybercriminalité & des investigations numériques 1 (1) (2015) 1–12. doi:http://dx.doi.org/10.18464/cybin.v1i1.2.

[63] P. Paganini, Virus Discovered at the Gundremmingen Nuclear Plant in Germany (Apr. 2016). URL http://securityaffairs.co/wordpress/46708/security/virus-gundremmingen-nuclear-plant.html

[64] P. J. Criscuolo, Distributed Denial of Service: Trin00, Tribe Flood Network, Tribe Flood Network 2000, and Stacheldraht ciac-2319, Technical Report UCRL-ID-136939 Rev. 1, Lawrence Livermore National Laboratory (Feb. 2000).

[65] M. Beunardeau, A. Connolly, R. Geraud, D. Naccache, Cdoe Obofsucaitn: Securing Software from Within, IEEE Security Privacy 14 (3) (2016) 78–81. doi:10.1109/MSP.2016.60.

[66] W. Wang, Q. Cai, Y. Sun, H. He, Risk-Aware Attacks and Catastrophic Cascading Failures in U.S. Power Grid, in: 2011 IEEE Global Telecommunications Conference (GLOBECOM 2011), 2011, pp. 1–6. doi:10.1109/GLOCOM.2011.6133788.

[67] M. Rosas-Casals, S. Valverde, R. Sole, Topological Vulnerability of the European Power Grid Under Errors and Attacks, International Journal of Bifurcation and Chaos 17 (7) (2006) 2465–2475. doi:10.1142/S0218127407018531.

[68] X. Jiang, J. Zhang, B. J. Harding, J. J. Makela, A. D. Dominguez-Garcia, Spoofing GPS Receiver Clock Offset of Phasor Measurement Units, IEEE Transactions on Power Systems 28 (3) (2013) 3253–3262. doi:10.1109/TPWRS.2013.2240706.

[69] Center for Disease Control and Prevention, Principles of Epidemiology (May 2016). URL http://www.cdc.gov/ophss/csels/dsepd/SS1978/Lesson1/Section11.html

Peter Eder-Neuhauser received the MSc degree in energy engineering from the University of Applied Sciences Technikum Wien. He is currently pursuing the Ph.D. degree within the interdisciplinary URBEM PhD-kolleg with the University of Technology Vienna (TU Wien). He is working on smart grid security with a focus on system resilience and malware containment in smart grid ICT.

Tanja Zseby is a professor of communication networks at the Faculty of Electrical Engineering and Information Technology at TU Wien. She received her diploma degree (Dipl.-Ing.) in electrical engineering and her doctoral degree (Dr.-Ing.) from TU Berlin, Germany. Before joining TU Wien she led the Competence Center for Network Research at the Fraunhofer Institute for Open Communication Systems (FOKUS) in Berlin and worked as visiting scientist at the University of California, San Diego.

Joachim Fabini received the Dipl.-Ing. degree in computer sciences and the Dr.Techn. degree in electrical engineering from the TU Wien. After five years of research with Ericsson Austria, he joined the Institute of Telecommunications, TU Wien, in 2003. He is a Senior Scientist with the Communication Networks Group with research focus on active measurement methodologies.

Gernot Vormayr received the Dipl.-Ing degree in computer technology from the TU Wien, Vienna, Austria. Since 2016 he is employed as a University Assistant at TU Wien, where he is pursuing his Ph.D. degree. His current research interest is malware communication with a focus on botnets.