Disinformation: A Cybersecurity Perspective

Golden G. Richard III
Center for Computation and Technology
Division of Computer Science and Engineering
Louisiana State University

Abstract

The public is now constantly challenged by disinformation campaigns, on social media and from traditional news outlets, which present both opinions and clear falsehoods as fact. These are expressed either to purposely deceive to further some agenda (e.g., deflection of blame, reelection, etc.) or are themselves based on widespread "supporting" disinformation (essentially, malicious "hearsay"). In the latter case, the supporting disinformation may have simply drowned out the facts and conveniently support a bias already held by the exponent of the opinion. These waves of disinformation can be initiated and amplified by individuals, nation states, and increasingly insidious and sophisticated malware that can operate on an unprecedented scale due to the ever-growing connectivity of people and devices. The facts are in real trouble—and so is democracy. Purposeful deception can now be carried out with an ease never dreamt of by the bad actors and propagandists of the past, and by taking control of users' computer systems, creating fake "armies" of affirmers and dissenters, and targeting individual users, producers of disinformation can easily influence popular opinion. The general public is immersed in technology, creating a fake sense of comfort with computers, smartphones, and Internet-connected devices of all kinds. People underestimate the dangers associated with constant connectivity while overestimating their level of awareness and expertise. Ultimately, deeply technical solutions, powered by digital forensics and memory forensics, must be added to our toolbox to disentangle truth from disinformation.

1. Disinformation as a Cybersecurity Problem

Disinformation can be damaging on a number of different levels, from challenging the integrity of democratic institutions, such as fair elections [96], to slandering specific individuals or corporations (e.g., [103, 74]). Disinformation can consist of text, emails, posts to social media, or convincing "deep fake" videos and other media using generative adversarial networks (GANs) [40, 92, 52, 97]. Because of the dense electronic communication network provided by ubiquitous email and access to social media sites such as Twitter and Facebook, news travels fast and viral disinformation presented as fact can spread

quickly without verification. Importantly, disinformation campaigns can both be directed and appear to be directed from virtually anywhere. Darknet operators are even offering disinformation as a service, with costs ranging from thousands to hundreds of thousands of dollars, depending on the scope of the campaign [46].

It is clear that significant disinformation campaigns have been perpetrated by human actors, frequently with the help of software bots [13] that amplify the information by posting content, following other accounts, upvoting, etc. [60, 35]. Studies have shown that at least 70 countries have participated in disinformation campaigns [29]. A thorough discussion of how social bots operate and some mechanisms for detecting them are presented in [36]. Rather than rehash these issues, this chapter focuses on the enabling technologies poised to make the disinformation problem far worse than it already is, including state-of-the-art malware and botnets, enabled by the appalling state of software and hardware security. In fact, we are creating, maintaining, and expanding a cyber environment which is ripe for disseminating disinformation, supporting attacks against democratic institutions, and fostering attacks against individuals.

The chapter also addresses the problems that individuals face when they are targeted by the same advanced cyber attack techniques that can enable widespread propagation of conspiracy theories or political propaganda. Malware can control their electronic devices, act on their behalf, disrupt electronic voting, destroy their reputations or financial standings, produce "evidence" of crimes allegedly committed, and more. That is, malware can not only amplify the disinformation problem but also make citizens unwilling accomplices, and in some cases, direct targets. Modern malware is stealthy, hard to analyze, frequently undetected by personal security products, such as antivirus, and can perform virtually any action that a computer user might perform. Users, even highly technical ones, are ill-equipped to handle these challenges.

2. The Evolution of Modern Malware

The first self-replicating computer software was postulated by John von Neumann in 1949 and later described in [67]. Initially, computer viruses were written primarily out of curiosity, as a programming exercise, and later, to establish a persona or reputation. In the 1980s and 1990s, malicious versions appeared, which caused damage by deleting files or making systems inaccessible.

The term malware now encompasses many different varieties of malicious software, including viruses, worms, Trojans, backdoors, and more. Generally, viruses replicated "passively", through shared storage media or electronic communication, such as email attachments. Worms intentionally self-replicate, actively targeting and attempting to infect systems. Trojans exhibit some benign and useful behavior with covert, malicious actions mixed in. Backdoors allow remote, unauthorized access to computer systems. Many malware samples exhibit more than one type of replication mechanism and may perform a variety of actions, including stealing personal information, downloading or uploading files, deleting or modifying sensitive data, sending email, activating video or

audio devices, and more. Depending on permissions obtained by a malware sample, virtually any action that the computer's owner might perform can also be performed by the malware.

In addition to propagation methods and intent, malware can also be differentiated into kernel-level malware and user-level malware. The primary difference is that while user-level malware operates with the permissions of individual applications, kernel-level malware can exercise complete control over the entire operating system (and thus every application that runs on the computer system and all user data). The goals of kernel-level malware are typically to provide backdoor functionality (to allow persistent, unauthorized access and control of a system), and to hide malicious data or processes from scrutiny. An instance of kernel-level malware can employ a number of techniques to gain access to needed resources and prevent detection, including hooking or modifying system calls, adding new system calls, inserting new kernel modules, and directly patching kernel code. Kernel-level malware is particularly difficult to detect, because it can deliberately prevent users and applications from viewing incriminating data or observing malicious processes.

Although there have been advances in exploitation prevention, including Data Execution Prevention (DEP) [80], non-executable stacks, user-space and kernel-space address randomization (ASLR and KASLR) [63], stack canaries, control flow [64] and code-pointer integrity [99], heap exploitation prevention techniques [16], signed kernel drivers, etc., attackers are up to the challenge of overriding them. The arms race between offensive and defensive cybersecurity is alive and well. Attackers have developed advanced techniques like Return-Oriented Programming (ROP) [14] to overcome defensive measures like data execution prevention. In response, defenders have formulated strategies to defeat ROP (e.g., [48, 42]; the literature in this area is vast, and detailed coverage is beyond the scope of this chapter), but that has simply resulted in numerous clever workarounds (e.g., [85, 61], among many others). Furthermore, while defensive techniques are somewhat effective in preventing malware from exploiting software components to gain a foothold in a user's systems, they do little to prevent malware from executing when users inadvertently click on an attachment or install software with covert malicious functionality.

2.1. The Incentivization of Malware

A radical difference between early malware and modern malware is incentive. While some of the factors that motivated early virus writers remain at play, monetization, politicization, and adoption by nation-states has completely changed the landscape [78]. Malware is now used to spread propaganda, target individual users or groups, distribute spam and disinformation, extort ransom, steal banking credentials, target critical infrastructure [53], and more [65]. While certain types of malware would seem to be out of the scope of a discussion on disinformation, it's imperative to see that the malware ecosystem is increasingly entwined.

Ransomware is the poster child of malware monetization. This type of malware encrypts user files, making them unusable, and extorts ransom from victims

in exchange for decryption keys. Early schemes, like the AIDS Information Introductory Diskette [7] were quite primitive and required users to send payment using postal mail to a foreign address (in this case, Panama) in exchange for decryption keys. Current-generation ransomware is much more sophisticated, with payment typically demanded in cryptocurrency. Furthermore, through the use of anonymizing network overlays like Tor [89] and I2P [104], the parties demanding ransom can remain completely anonymous. Many ransomware variants even offer instructions on how to install Tor or I2P, and use a chat system running over these anonymizing network overlays to discuss payment in real time. Recently, ransomware authors have discovered new ways of ensuring that victims pay, by exfiltrating user or corporate data, tying the release of this data with non-payment of ransom.

Malware that mines or steals cryptocurrency [72] is also prevalent. One of the most profitable campaigns analyzed among more than 1.2M malware samples in [72] mined more than $10M in Monero cryptocurrency (using the value of Monero as of 7/8/2020, roughly $65 per XMR). Malware authors have also begun to mix cryptocurrency and ransomware campaigns, by mining cryptocurrency using the victim's computer resources while awaiting payment. Cryptocurrency stealing malware is particularly hard to detect, because it accesses files and other data on a computer system in much the same way users might. One ransomware-stealing variant, called CryptoShuffler [94], monitors a user's Windows clipboard and detects when cryptocurrency addresses, such as those used by Bitcoin, Ethereum, etc., are copied and pasted. The malware silently modifies the addresses, changing them into addresses under the attacker's control, causing payments to be diverted to the attacker. Since cryptocurrency transactions are final, the user loses any diverted funds. Detection is extremely difficult as the malicious payload simply interacts with the system clipboard, an operation performed by many benign applications.

The same distribution systems for infecting users with ransomware can facilitate gaining control of their systems, gathering personal information, interacting with social media in unauthorized ways, distributing and reinforcing disinformation, and more. When groups of machines are under the control of a central operator, botnets are formed.

2.2. Botnets and Disinformation

Aside from targeting the private data of users, modern malware can also enable botnets, which are large groups of infected computer systems under the control of a malicious actor. Advanced command-and-control systems are used to direct the actions of infected computers and botnets can be configured to distribute spam email, conduct email "bombing" campaigns, interact with social media, "war dial" to overload phone systems, and more. Because of widespread and persistent Internet connectivity—a phenomenon that not present when primitive malware first appeared—large-scale botnets can be quickly assembled and do devastating harm. The creation of these botnets is empowered by widespread vulnerabilities in software and hardware, many of which remain in place for years without security patches being offered. A recent study by

Germany's Fraunhofer Institute for Communication (FKIE) [93, 101] evaluated 127 routers manufactured by seven different companies. Every single router had security vulnerabilities, with many using old software components that have not received security updates in almost a decade. Many of the routers had hard-coded credentials to access the administrative interface of the router that could not be changed. Since routers of this type are deployed in almost every home with Internet access, powerful opportunities for cybercriminals are created. Unfortunately, previous studies mirror these results and numerous other hardware and software vulnerabilities open additional avenues for exploitation. Finally, in addition to software and hardware exploitation tactics, botnets are frequently constructed and expanded using social engineering tactics, by tricking users into clicking on attachments in email or visiting malicious websites.

The notorious banking Trojan called Emotet, which emerged in 2014, is frequently spread via infected document files attached to emails. These documents are purported to be invoices, information about package deliveries, etc. When a malicious document is opened, it silently downloads additional software components from a command and control server. The command and control servers also allow the malware authors to update the malware in place on a user's system, to add new functions, exfiltrate user data, and more. Emotet has been significantly expanded since its introduction and is now used to deliver additional malware and distribute spam. In addition to spreading through malicious documents, it has other propagation methods, including brute forcing passwords to other systems on the local network and using contacts discovered on an infected machine to send targeted emails. By maliciously injecting responses into legitimate email threads copied from infected machines, Emotet is able to trick users into opening infected attachments [66]. Newer variations attempt to connect to wireless networks associated with an infected machine, targeting other computers on those networks [68]. Emotet has now grown into a large botnet, organized into multiple tiers which apparently do not communicate with each other. Recently, the actors behind Emotet launched a disinformation campaign targeting Japanese users by sending emails purportedly from a disability service provider in Japan. These emails contain illegitimate reports about COVID-19 infections and urge users to open attached malicious documents, resulting in malware infection [41].

The widespread deployment of Internet of Things (IoT) devices, many of which are plagued with security issues, has made creation of extremely large botnets even easier. The most famous IoT bot, Mirai [43], was estimated to have infected between 800,000 and 2.5M devices [62]. Because the source code for Mirai was leaked, numerous variants based on this malware strain have been released, including Satori, Okiru, Masuta, PureMasuta, OMG, Jenx, Wicked Mirai, covid [100], and many more.

Linux/Moose [69], a botnet that infected routers running Linux, spread rapidly by brute forcing credentials and actively destroying other resident malware. Linux/Moose stole HTTP cookies from social media sites visited by victims, allowing it to issue unauthorized follows, likes, and page views on Facebook, Instagram, Youtube, Twitter, and other sites. Even without posting

messages, the activities of bots like these can be used to dramatically alter the metrics that users rely on to judge public support for particular individuals, points of view, or activities.

3. Malware, Disinformation, and Voting Systems

3.1. Dangers to Voting Systems

We now turn to the interplay between disinformation, malicious actors, malware, and voting. This is particularly important in light of recent concerns about in-person voting raised by the 2020 COVID crisis. But despite a vast amount of literature pointing out vulnerabilities of electronic voting systems [32, 70, 10, 9], states in the US continue to expand electronic voting and some are considering using online voting systems, which allow voters to cast a ballot from a computer or mobile device [38]. Internet-based voting is dramatically more insecure than electronic voting systems that require a voter to visit a polling station. Many of these reasons are outlined in recent letters to the Cybersecurity and Infrastructure Security Agency (CISA) [86] and to governors and secretaries of state [5], authored by prominent cybersecurity experts. Online voting offers a vast number of targets to exploit and experts are near unanimous in believing that we will have no viable, secure methods for online voting in the near future [47]. Furthermore, as discussed in [47], there is a tendency to equate the widespread adoption of "secure" online commerce with the potential for "secure" online voting. Arguments of this variety miss many important differences. First, the motivation to tamper with voting on a large scale is much stronger. Second, the security and privacy requirements for the two activities are quite different. According to [73], ecommerce fraud reached 3.85% in Q2 of 2017, resulting in over $57B in losses across eight industries. Fraud for higher valued purchases, i.e., those exceeding $500, was 11.65%. These losses are "silently" borne by the affected ecommerce sites. There is simply no room for this amount of error in political elections. Furthermore, no mechanisms to recoup or redistribute losses is available. Unlike ecommerce transactions, which record the specific account associated with a purchase, only an individual voter knows which candidate they chose and there is no way to associate a voter with a particular vote after the vote is cast. The latest buzzword in electronic voting is using blockchain technologies. Prominent cybersecurity experts point out that these efforts, too, are misguided [70].

Clearly, the malicious techniques discussed in previous sections offer a number of methods to corrupt the voting process. Online authentication of voters is problematic since schemes to validate voters based on knowledge of date of birth, social security numbers, etc., are compromised by vast data leaks that make this information readily available to attackers. Such leaks include information about 150M Adobe customers in 2013; 450M customers for Adult Friend Finder in 2016, with poor password storage resulting in compromise of all account data; 145M eBay users in 2014, including usernames, addresses, dates of birth, and more; 148M consumers during the Equifax breach of 2017, revealing social security numbers, birth dates, addresses, drivers' license numbers; a staggering 3B

user accounts in Yahoo's breach in 2013-4, with names, email addresses, dates of birth, telephone numbers, and more leaked [87]; and 800GB of data affecting 200M U.S. individuals leaked from an unknown source (but suspected to be the US Census), containing full names, email addresses, phone numbers, dates of birth, credit ratings, home addresses, demographic information, mortgage and tax records, and more [83]. In 2015, the U.S. Office of Personnel Management (OPM) was breached, revealing highly sensitive information related to security clearances of more than 20M individuals [95]. Unfortunately, this list is far from exhaustive, with a Wikipedia page tracking roughly 300 breaches like this since 2014 [102]. The real list is likely far longer.

Online voting provides attackers with the opportunity to perform distributed denial-of-service attacks (DDOS), overwhelming servers that collect votes and potentially preventing groups of voters from casting their votes. These servers are also ripe targets for exploitation, to modify or delete votes that have been cast. Any confidence that the systems can be secured is undermined by casual reflection on the data breaches discussed above.

Even hybrid systems, which provide only downloadable ballots, are susceptible to attack. An electronically marked ballot exposes the user not only to data leakage, where information about the vote may persist and be recoverable by malware or memory forensics techniques, but to tampering, as well. Malware infecting the document viewer (e.g., Adobe Acrobat or Preview) can present an invalid ballot, with choices that differ from those on the official ballot, with the choices reordered, or some other mechanism to ultimately cause the ballot to be rejected. Electronically submitting the document is also fraught with user-side risks, as the document viewer may present different content than what is stored in the PDF file. Even printing the document may be insufficient, as print functions in the document viewer could be similarly compromised. Aside from checking the electronic ballot using a dedicated system that has been isolated to prevent infection, the user might still submit a ballot reflecting someone else's choices. Even downloading, printing, and then physically marking up an electronic ballot is subject to interference, as malware can intercept the ballot in the user's web browser and apply the same measures to cause the ballot to be ultimately rejected and the vote lost.

The preferable method for conducting an election where individuals cannot visit a polling station, such as it might be under the COVID-19 crisis, is reception of a paper ballot in the mail, to be marked by hand and then returned via postal mail.

4. Disinformation and the Trojan Defense

Disinformation isn't a strictly global phenomenon. Malicious actors and malware can target individuals or groups and plant false information intended to cause embarrassment, loss of reputation or employment, or civil or criminal liability. Because of the complexity and stealth exhibited by modern attackers and malware, separating the actions performed by a user from those performed by a malicious actor can be exceedingly difficult. State-of-the-art malware can

surreptitiously perform *any* action a computer user can perform, including sending emails, surfing web sites, downloading files and categorizing them, executing attacks against other computer systems, and more. Often, all the owner can reasonably do is plead their innocence.

The Trojan defense, also known as the "some other dude did it" (SODDI) defense, is likely the oldest legal defense. This tactic asserts that the accused is innocent and that some other party is responsible. Since the advent of malware, this defense also includes non-human entities, like computer software. The SODDI defense is traditionally met with skepticism, often well-deserved, but in many cases, the accused person's livelihood and life may depend on an accurate assessment of the facts. Unfortunately, a relatively standard procedure for refuting the Trojan defense in cases where malware is blamed is to perform a forensics investigation and run an antivirus product on the accused's computer system(s). Lack of detection is then essentially a "refutation" of the defense.

The strategy of relying on traditional digital investigative techniques combined with antivirus scans mirrors strategies discussed in "The Trojan Horse Defense in Cybercrime Cases" [12], written more than 15 years ago. That paper suggests a two-pronged approach for prosecutors to refute the SODDI, specifically, that investigators "establish the defendant's computer expertise" and "negate the factual foundation of the defense". The former goal is based on experiences those authors had with defendants using the SODDI also claiming to have little computer expertise, essentially making them particularly vulnerable to malware or an attacker. While it is reasonable to assume that someone accused of a crime would attempt to make their SODDI claim as plausible as possible by introducing the issue of technical aptitude, it is arguable that to some degree, aptitude is a red herring. Not only can modern malware evade detection by antivirus (discussed below), but even technical users have significant difficulty in establishing the legitimacy of websites and differentiating phishing emails from legitimate ones. Recent studies continue to illustrate that users aren't very good at detecting phishing attacks [28] and that technical expertise is not correlated with better detection [4]. Alarmingly, some studies reveal "...educated users and those with high levels of privacy concerns being most susceptible to harm." [8].

The second prong is to conduct an intensive digital forensics investigation. Typical steps taken in such an investigation are discussed in Section 6.1. But as we'll see there, traditional digital forensics techniques are not sufficient for investigating many strains of modern malware, which may be memory-only, fileless, and leave virtually no traces on storage devices. If the investigative effort is truly carried out with all of the technical tools available to us today, including memory forensics (discussed in Section 6.2), then the chances of detecting and analyzing malware (or refuting a SODDI claim) is increased, but this not only requires advanced technical skills (which are in short supply), but is also expensive.

In [84], a number of SODDI cases are discussed in detail. The author points out that while SODDI defenses are often successful in civil cases, that he was aware of no successful acquittals in the criminal domain. Most of the criminal

cases discussed involved child pornography and other compelling evidence was frequently available.

Details of a private case in which the author was involved are presented next, to illustrate some of the technical issues that SODDI defenses may present. The case involved the termination of a female employee who was accused of accessing NSFW materials at work—specifically, visiting and retrieving images from online pornographic sites. The employee claimed innocence, but network logs created by the company's IT staff clearly showed access to these sites from the employee's computer, and furthermore, only when she was in her office. A preliminary forensic examination of her system revealed dozens of pornographic images in the Internet Explorer browser cache and numerous entries in web history, all indicative of intentional access to the pornographic sites.

Further analysis of the computer, performed when the employee continued to insist that she was innocent, revealed malware was present. Prior antivirus scans had failed to detect the malware and it was detected solely because of its use of a very common mechanism for persistence (i.e., manipulation of a RUN key in the Windows registry, which governs which applications start automatically when the machine is rebooted). The use of this relatively easy-to-detect persistence mechanism was curious and a stroke of luck for the employee, as the malware itself was sophisticated (and extremely difficult to analyze manually). It communicated with a remote server to drop additional malicious components on the victim's system and then manipulated Internet Explorer to surf various pornographic sites *without presenting the usual graphical interface* by directly accessing functions in the *ShDocVw.dll* dynamic library (a component of Internet Explorer). The surfing activity was triggered by use of the affected system's keyboard, ensuring the party using the computer would be held liable for accessing the sites. Absolutely no visual signs of the web surfing activities were apparent to the user of the computer system. By controlling Internet Explorer directly, the web history and cache files were populated precisely as if the user had deliberately accessed the sites.

An important question is whether the malware would have been detected if the persistence mechanism had not given it away. When the malware sample was analyzed using 31 different antivirus products, 20 of them tagged it as benign, including ClamAV, F-Prot, Kaspersky, McAfee, and Microsoft's products. Eleven of the products flagged the file as malware, but in every instance, the file was determined simply to be "generic" malware, meaning that it exhibits suspicious behavior, but no details or clues for further investigation of the malicious behavior were provided.

It's important to note that the intention of [12] is not to strip the accused of their potentially only line of defense, but rather to counter spurious use of the SODDI:

> "Our goal, then, is to explain how to negate the defense when it *is* simply a "defensive tactic": a technologically-based SODDI defense. It is not our intention to discredit the Trojan horse defense, as there will no doubt be instances in which its invocation will be well

> founded. Therefore, we seek only to explain how it can be negated when it is being used in an attempt to prevent the conviction of someone who is demonstrably guilty."

The point of the current discussion is similarly not to suggest we throw up hands in dispair and say that electronic evidence simply can't be relied upon. Rather, it is to call attention to the fact that traditional and relatively simplistic digital forensics investigations will typically *not* be effective in unwinding deliberate disinformation campaigns involving sophisticated malware or attackers. Furthermore, computers are not only increasingly involved in crimes, but in many cases provide the *only* evidence that a crime has been committed. Given the capabilities exhibited by modern malware, the extremely difficult task of detecting and analyzing it, and the costs involved in performing such analysis, we should at the very least be more receptive to the possibility that a SODDI defense is valid, especially in situations where the accused faces life-altering (or life-ending) punishment.

5. User Awareness and Perception

Now that some of the threats have been discussed, it is important to address how computer users perceive these threats and what defensive tactics they might employ.

While better user education concerning the threats they face online may help, there are currently few technical solutions available to protect users from advanced malware. Users are often told to run antivirus software and while these products work to some degree, they are most effective against well-known and previously analyzed threats. Users are also instructed to obey the usual "cyber-hygiene" rules, including changing default passwords and not reusing passwords; not clicking on unknown email attachments; not visiting unknown websites; and keeping operating systems and applications up to date. In many cases, users do not correctly perceive what threats are even possible, much less pay adequate attention to them. In numerous studies, user inattention or ambivalence to security issues has been established [1, 33, 44].

Despite the very serious concerns about online voting discussed in the previous section, an alarming 49% of users in a recent poll [90] indicated their support for voting over the Internet in US elections. This is despite prominent experts warning the US government against Internet voting and pointing out woefully inadequate security measures in a US Cybersecurity and Infrastructure Security Agency report [86]. It is therefore very likely that users on the "front line", who encounter new zero-day threats, will be impacted. In fact, experts question whether it is even reasonable for users to be expected to manage their own cybersecurity concerns [49], given that the technical considerations are beyond most users' capabilities and very little concrete assistance is available. Exacerbating this problem is that users often need not explicitly perform any risky actions at all, other than simply purchasing a hardware device or installing an application. This is because many hardware and software components are

vulnerable out of the box. Simply deploying these components exposes users to substantial hidden risks.

Many computer users apparently maintain, either consciously or unconsciously, a "cyber world view" in almost direct contradiction to models like Descartes' imaginary demon [31], described as having 'utmost power and cunning' and having 'employed all his energies in order to deceive me.' This is likely a result of two factors. For the types of threats a particular user is aware of, they may imagine they are taking whatever precautions they can. But more importantly, it is conceivable that most users can scarcely imagine the breadth and depth of cyberattacks that are possible today.

An important aspect of cyberattacks is that they are contagious. Regardless of whether users have attempted to take precautions and are compromised or if they are willfully oblivious, infected computers impact a much broader population than the owner, since they can be used to compromise privileged multi-party communications, as a vector for creating botnets for widespread attacks or dissemination of disinformation, and more. This is in sharp contrast to threats like fire, for example, which are more localized and for which more resources are made available to potential victims by government institutions [49].

6. Technical Solutions

There are promising technical solutions to combat malware and disinformation, but for the most part they are not yet suitable for deployment on end-user systems. Most are investigative and useful to determine the nature of an attack that has already occurred and possibly to identify the attackers. The remainder of this section examines some technical solutions to discovering and analyzing malware that might be used in disinformation campaigns, whether against the general public or specific individuals. While complete solutions are unavailable, digital forensics and memory forensics play an important role in combating disinformation and protecting users.

6.1. Digital Forensics

Traditionally, digital forensics techniques are used to preserve and analyze digital evidence stored on computer systems, cell phones, and other digital devices. These digital forensics techniques target non-volatile storage devices, such as hard drives. The typical workflow is to power down a target machine, make exact copies of its storage devices, and employ a variety of techniques to recover evidence. These techniques include data carving [39] to retrieve deleted data, generation of timelines to determine impacted files [15, 45], and analysis of the Windows registry [18, 17]. An important issue is determining if malware is present and if it is, extracting a sample of the malware for reverse engineering, to see where it came from and what it does.

Data carving techniques use databases of headers and footers, which are strings of bytes at predictable offsets in a file, or more complete file specifications, to identify the start and end locations of files or other data to recreate

deleted files. A common method to infect systems is to spread malicious executables which then download additional components during the infection process, possibly deleting some of thesesP components. This makes file carving an important strategy in recovering deleted executables for analysis. One limitation of the current generation of data carving tools is that the data must generally be stored contiguously on the storage device to be fully recoverable. This is usually not a serious problem on modern systems, however, as malware executables tend to be small and modern computers have abundant storage.

Generating forensic timelines is useful in several ways. First, when investigating malware infections, it allows an investigator to establish when specific files were created, accessed, and modified. Timelining also allows pinpointing the specific times when media was created, which could include the source materials and final product in the generation of deep fakes.

Analysis of system configuration data, such as the Windows registry, can help investigators discover malware as well as establish whether certain malware persistence techniques were employed. As discussed earlier, it was the use of a common persistence technique that upheld a SODDI defense involving access to illicit web sites while at work. These persistence techniques often use the Windows registry to ensure that the malware sample will execute each time the system is restarted. More stealthy persistance mechanisms are in wide use [91] and some are trickier to detect.

A number of other digital forensics techniques are also common. These include the examination of web browser history and caches, to shed light on surfing activities. Network and application logs are also scrutinized to determine when users logged in, whether network scanning activities have been employed, whether applications crashed because of tampering by malware, etc. For many civil as well as criminal cases, especially when supporting physical evidence is available, these traditional digital forensics techniques perform well and are sufficiently powerful to reveal both incriminating and exculpatory evidence. When sophisticated malware is involved, however, these storage forensics techniques fall well short. Modern malware may be file-less or in-memory only [81, 98, 11, 57, 30], meaning that it may leave absolutely no traces on a computer system's storage devices. This may result in signs of malware infection being completely missed, particularly if the machine is powered down to make copies of storage devices, which is typical. Supplementing these techniques to address detection of modern malware is essential, and memory forensics, discussed in the next section, offers great promise.

6.2. Memory Forensics

Over the last 15 years, memory forensics [59, 3, 82, 79, 26, 25, 88, 23, 24, 76, 77, 2] has supplemented digital forensics techniques, offering a better idea of what has occurred and what is happening on a computer system. Memory forensics techniques analyze a snapshot of a system's volatile storage (RAM) instead of concentrating solely on data stored on non-volatile storage devices. Since almost any operation on a computer system induces changes in RAM,

memory forensics can offer a much more complete picture than traditional digital forensics techniques. The evidence recoverable through memory forensics includes lists of processes that have executed on the computer system, active and closed network connections, memory-only malware code, hooks inserted by malware to influence system behaviors, and more.

One problem with many malware detection techniques is that they require a particular malware sample to have been previously analyzed. The appearance of targeted malware, designed explicitly for an attack against an individual, a corporation, or a nation state, requires techniques able to detect all malware and not only malware belonging to previously analyzed families. The author and his collaborators are working on a number of memory forensics techniques designed to discover whenever malware is present on a system and provide deeper analysis capabilities, regardless of whether the malware has been seen before.

In memory forensics research funded by the National Science Foundation, the author, along with Andrew Case from the Volatility Foundation, Mingxuan Sun from LSU, Aisha Ali-Gombe of Towson University, and a number of students from LSU and Towson, are working on important problems in memory forensics. First, we are creating a large and diverse collection of freely available, realistic data sets for memory forensics research and practice. One issue with current memory forensics techniques is that they recover so many artifacts that investigators are easily overwhelmed. To deal with this situation, investigators will frequently start by investigating a "known good" system, running the same operating system and application versions as a targeted system, to understand the system's "normal" state. Once "normal" is understood, anomalous artifacts can quickly be filtered out. Our well-documented set of memory images offer "ground truth" and present a solution to this problem.

Our research effort also includes the creation of tools to automate the tedious and error-prone process of ensuring memory forensics toolsets operate correctly and produce accurate results. Memory forensics frameworks consist of complex code bases. For every artifact an analysis tool recovers from a memory sample, it must typically re-implement one or more algorithms used by an operating system or application. Furthermore, tools must also perfectly replicate the layout of data structures processed by these algorithms to produce correct results. Generating incorrect results (or no results at all) due to coding errors can lead to dire consequences, allowing dangerous malware to go undetected. This is particularly problematic as memory forensics becomes more automated, with no human investigator evaluating each step. Our solution is to develop a massively-parallel fuzzing platform for memory forensics tools, called Gaslight [20, 6, 71], which intelligently modifies memory images to simulate both acquisition errors as well as malicious tampering. Gaslight stress tests memory forensics tools to find errors in the implementations, which are flagged for correction by the developers of the tools. This is critically important to ensure that memory forensics frameworks provide accurate results.

We are also expanding the scope of memory forensics to better detect and analyze userland malware [22]. We have created tools to discover and analyze advanced malware affecting macOS systems [27] as well as techniques to foren-

sically analyze the Windows Subsystem for Linux (WSL) [58]. WSL essentially provides a complete Linux runtime environment inside of Windows 10. Prior to this work, there were no memory forensics tools for analyzing malware that might utilize WSL. We have also conducted pioneering research in the use of emulation in detecting and generating descriptive indicators of compromise for userland malware, based solely on memory images of infected machines [19, 21]. This system is called HookTracer and offers a valuable resource in detecting memory-only and file-less malware, including state-of-the-art keystroke loggers and other malware that hooks operating system and application functions. Finally, we are currently developing new detection and analysis capabilities for malware that has a direct, negative impact on targeted individuals and organizations, motivated by a rash of incidents of this kind [37, 51, 75, 34, 55, 54, 56, 50]. This research involves development of deep memory forensics techniques to investigate compromises of web and database servers. It is our conviction that memory forensics plays a crucial role in battling modern malware and making systems safer.

We acknowledge that a persistent problem with digital forensics and memory forensics techniques is that the investigative procedures are still mostly reactive and require substantial amounts of manual investigative effort, typically performed by an experienced investigator. We are hopeful that our work will ultimately support more autonomous and automated solutions to automatically detect and remediate malicious software of all kinds, including malware that supports disinformation campaigns.

Acknowledgements

This work was supported in part by the National Science Foundation through grant # 1703683. I am very grateful to Elsa Hahne for offering a great deal of helpful feedback on drafts of this chapter. I would also like to acknowledge my many collaborators for helpful discussions, and in particular, Andrew Case, a leading memory forensics researcher and incident response expert.

References

[1] Adrienne Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and Daid Wagner. Android Permissions: User Attention, Comprehension, and Behavior. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, pages 1–14, 2012.

[2] Irfan Ahmed and Golden G. Richard III. Kernel Pool Monitoring for Live Forensics. *Proceedings of the 66th Annual Meeting of the American Academy of Forensic Sciences (AAFS)*, 2014.

[3] Aisha Ali-Gombe, Sneha Sudhakaran, Andrew Case, and Golden G. Richard III. DroidScraper: A Tool for Android In-Memory Object Recovery and Reconstruction. In *22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019*, pages 547–559, 2019.

[4] Mohamed Alsharnouby, Furkan Alaca, and Sonia Chiasson. Why Phishing Still Works: User Strategies for Combating Phishing Attacks. *International Journal of Human-Computer Studies*, 82:69–82, 2015.

[5] American Association for the Advancement of Science. Letter to Governors and Secretaries of State on the Insecurity of Online Voting. `https://www.aaas.org/programs/epi-center/internet-voting-letter`, 2020.

[6] Arian Shahmirza. High Performance Fuzz Testing of Memory Forensics Frameworks, M.S. Thesis, Louisiana State University, 2019.

[7] Jim Bates. Trojan Horse: AIDS Information Introductory Diskette Version 2.0. *Virus Bulletin*, pages 3–6, 1990.

[8] Grant Blank and Christoph Lutz. Benefits and Harms from Internet Use: A Differentiated Analysis of Great Britain. *New Media & Society*, 20(2):618–640, 2018.

[9] Matt Blaze, J Braun, H Hursti, D Jefferson, M MacAlpine, and J Moss. DEFCON 26 Voting Village: Report on Cyber Vulnerabilities in U.S. Election Equipment, Databases, and Infrastructure. *DEFCON*, 26, 2018.

[10] Matt Blaze, Jake Braun, and Cambridge Global Advisors. DEFCON 25 Voting Machine Hacking Village. *Proceedings of DEFCON, Washington DC*, pages 1–18, 2017.

[11] Boldizsár Bencsáth, Gábor Pék, Levente Buttyán, and Márk Félegyházi. Duqu: A Stuxnet-like Malware Found in the Wild. CrySyS Lab Technical Report 14, 2011.

[12] Susan W Brenner, Brian Carrier, and Jef Henninger. The Trojan Horse Defense in Cybercrime Cases. *Santa Clara Computer & High Tech. Law Journal*, 21:1, 2004.

[13] John Briar. Disinformation in 5.4 Billion Fake Accounts: A Lesson for the Private Sector. `https://www.securitymagazine.com/articles/91616-disinformation-in-54-billion-fake-accounts-a-lesson-for-the-private-sector`, 2020.

[14] Erik Buchanan, Ryan Roemer, Stefan Savage, and Hovav Shacham. Return-oriented Programming: Exploitation Without Code Injection. In *Blackhat 8*, 2008.

[15] Florian P Buchholz and Courtney Falk. Design and Implementation of Zeitline: a Forensic Timeline Editor. In *Proceedings of the Digital Forensics Research Conference*, 2005.

[16] Byoungyoung Lee, Chengyu Song, Yeongjin Jang, Teilei Wang, Taesoo Kim, Long Lu, and Lee, Wenke. Preventing Use-after-free with Dangling Pointers Nullification. In *NDSS*, 2015.

[17] Harlan Carvey. *Windows Registry Forensics: Advanced Digital Forensic Analysis of the Windows Registry*. Elsevier, 2011.

[18] Harlan Carvey, C. Harrell, and B. Shavers. RegRipper, 2012.

[19] Andrew Case, Aisha Ali-Gombe, Mingxuan Sun, Ryan Maggio, Md Firoz-Ul-Amin, Mohammad Jalalzai, and Golden G. Richard III. HookTracer: A System for Automated and Accessible API Hooks Analysis. *Proceedings of the 18th Annual Digital Forensics Research Conference (DFRWS)*, 2019.

[20] Andrew Case, A. Das, S-J Park, Ram Ramanujam, and Golden G. Richard III. Gaslight: A Comprehensive Fuzzing Architecture for Memory Forensics Frameworks. *Proceedings of the 2017 Digital Forensics Research Conference (DFRWS)*, 2017.

[21] Andrew Case, Ryan Maggio, Md. Firoz-Ul-Amin, Mohammad Jalalzai, Aisha Ali-Gombe, Mingxuan Sun, and Golden G. Richard III. Hooktracer: Automatic Detection and Analysis of Keystroke Loggers Using Memory Forensics. In *Computers and Security*, volume in press, 2020.

[22] Andrew Case, Ryan Maggio, Modhu Manna, and Golden G. Richard III. Memory Analysis of macOS Page Queues. *Digital Forensics Research Conference (DFRWS)*, 2020.

[23] Andrew Case, Lodovico Marziale, Chris Neckar, and Golden G. Richard III. Treasure and Tragedy in kmem_cache Mining for Live Forensics Investigation. *Proceedings of the 10th Annual Digital Forensics Research Workshop (DFRWS)*, 2010.

[24] Andrew Case, Lodovico Marziale, and Golden G. Richard III. Dynamic Recreation of Kernel Data Structures for Live Forensics. *Proceedings of the 10th Annual Digital Forensics Research Workshop (DFRWS)*, 2010.

[25] Andrew Case and Golden G. Richard III. In Lieu of Swap: Analyzing Compressed RAM in Mac OS X and Linux. *Proceedings of the 14th Annual Digital Forensics Research Workshop (DFRWS)*, 2014.

[26] Andrew Case and Golden G. Richard III. Advancing Mac OS X Rootkit Detection. *Proceedings of the 15th Annual Digital Forensics Research Workshop (DFRWS)*, 2015.

[27] Andrew Case and Golden G. Richard III. Detecting Objective-C Malware Through Memory Forensics. *Proceedings of the 16th Annual Digital Forensics Research Workshop (DFRWS)*, 2016.

[28] Christian Iuga, Jason Nurse, and Arnau Erola. Baiting the Hook: Factors Impacting Susceptibility to Phishing Attacks. *Human-centric Computing and Information Sciences*, 6(1), 2016.

[29] Davey Alba and Adam Satariano. At Least 70 Countries Have Had Disinformation Campaigns, Study Finds. https://www.nytimes.com/2019/09/26/technology/government-disinformation-cyber-troops.html, 2019.

[30] Dell SecureWorks Counter Threat Unit Threat Intelligence. Skeleton Key Malware Analysis. https://www.secureworks.com/research/skeleton-key-malware-analysis, 2015.

[31] René Descartes. *René Descartes: Meditations on First Philosophy*. Cambridge University Press, 2013.

[32] Michael Dunn and Laurence Merkle. Overview of Software Security Issues in Direct-Recording Electronic Voting Machines. In *ICCWS 2018 13th International Conference on Cyber Warfare and Security*. Academic Conferences and Publishing Limited, 2018.

[33] David Eargle, Dennis F Galletta, and Jeffrey L Jenkins. What's It Worth to You? Applying Risk Tradeoff Paradigms to Explain User Interactions with Interruptive Security Messages. *Proceedings of the Workshop on Information Security and Privacy*, 2016.

[34] ESET. New Malware Spies on Diplomats, High-Profile Government Targets. https://www.bleepingcomputer.com/news/security/new-malware-spies-on-diplomats-high-profile-government-targets/, 2019.

[35] Emilio Ferrara. Bots, Elections, and Social Media: A Brief Overview. In *Disinformation, Misinformation, and Fake News in Social Media*, pages 95–114. Springer, 2020.

[36] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. The Rise of Social Bots. *Communications of the ACM*, 59(7):96–104, 2016.

[37] FireEye. Connecting the Dots: Syrian Malware Team Uses BlackWorm for Attacks. https://www.fireeye.com/blog/threat-research/2014/08/connecting-the-dots-syrian-malware-team-uses-blackworm-for-attacks.html, 2014.

[38] Eric Geller. Some States Have Embraced Online voting. It's a Huge Risk. https://www.politico.com/news/2020/06/08/online-voting-304013, 2020.

[39] Golden G. Richard III and Vassil Roussev. Scalpel: A Frugal, High Performance File Carver. *Digital Forensics Research Conference (DFRWS)*, pages 71–77, 2005.

[40] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[41] Jonathan Greig. Hackers Using Coronavirus Scare to Spread Emotet Malware in Japan. https://www.techrepublic.com/article/hackers-using-coronavirus-scare-to-spread-emotet-malware-in-japan/, 2020.

[42] Aditi Gupta, Sam Kerr, Michael S Kirkpatrick, and Elisa Bertino. Marlin: A Fine Grained Randomization Approach to Defend Against ROP Attacks. In *International Conference on Network and System Security*, pages 293–306. Springer, 2013.

[43] Ben Herzberg, Dima Bekerman, and Igal Zeifman. Breaking Down Mirai: An IoT DDoS Botnet Analysis. https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html, 2016.

[44] Inho Hwang, Daejin Kim, Taeha Kim, and Sanghyun Kim. Why Not Comply with Information Security? An Empirical Approach for the Causes of Non-compliance. *Online Information Review*, 2017.

[45] Bartosz Inglot, Lu Liu, and Nick Antonopoulos. A Framework for Enhanced Timeline Analysis in Digital Forensics. In *2012 IEEE International Conference on Green Computing and Communications*, pages 253–256. IEEE, 2012.

[46] Insikt Group. The Price of Influence: Disinformation in the Private Sector. https://www.recordedfuture.com/disinformation-service-campaigns/, 2019.

[47] David Jefferson. If I Can Shop and Bank Online, Why Can't I Vote Online? — Verified Voting. https://www.verifiedvoting.org/resources/internet-voting/vote-online/, 2020.

[48] Kaan Onarlioglu, Leyla Bilge, Andrea Lanzi, Davide Balzarotti, and Engin Kirda. G-Free: Defeating Return-Oriented Programming Through Gadget-less Binaries. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 49–58, 2010.

[49] Karen Prenaud, Stephen Flowerday, Merrill Warkentin, Paul Cockshott, and Craig Orgeron. Is the Responsibilization of the Cyber Security Risk Reasonable and Judicious? *Computers & Security*, 78:198–211, 2018.

[50] Kaspersky. The Darkhotel APT. https://securelist.com/the-darkhotel-apt/66779/, 2014.

[51] Kaspersky. The Syrian Malware House of Cards. https://securelist.com/the-syrian-malware-house-of-cards/66051/, 2014.

[52] Jan Kietzmann, Linda W Lee, Ian P McCarthy, and Tim C Kietzmann. Deepfakes: Trick or Treat? *Business Horizons*, 63(2):135–146, 2020.

[53] Robert Knake. Top Conflicts to Watch in 2020: A Cyberattack on U.S. Critical Infrastructure. https://www.cfr.org/blog/top-conflicts-watch-2020-cyberattack-us-critical-infrastructure, 2020.

[54] Citizen Lab. NSO Group's iPhone Zero-Days Used Against a UAE Human Rights Defender. https://citizenlab.ca/2016/08/million-dollar-dissident-iphone-zero-day-nso-group-uae/, 2014.

[55] Citizen Lab. Senior Mexican Legislators and Politicians Targeted with NSO Spyware. https://citizenlab.ca/2017/06/more-mexican-nso-targets/, 2014.

[56] Citizen Lab. Between Hong Kong and Burma. https://citizenlab.ca/2016/04/between-hong-kong-and-burma/, 2016.

[57] Kaspersky Lab. Kaspersky Lab Uncovers "The Mask": One of the Most Advanced Global Cyber-espionage Operations to Date Due to the Complexity of the Toolset Used by the Attackers. https://usa.kaspersky.com/about/press-releases/2014, 2014.

[58] Nathan Lewis, Andrew Case, Aisha Ali-Gombe, and Golden G. Richard III. Memory Forensics and the Windows Subsystem for Linux. *Digital Investigation*, 26:S3–S11, 2018.

[59] Michael Hale Ligh, Andrew Case, Jamie Levy, and AAron Walters. *The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory*. Wiley, New York, 2014.

[60] Luca Luceri, Ashok Deb, Adam Badawy, and Emilio Ferrara. Red Bots Do It Better: Comparative Analysis of Social Bot Partisan Behavior. In *Companion Proceedings of the 2019 World Wide Web Conference*, pages 1007–1012, 2019.

[61] Giorgi Maisuradze, Michael Backes, and Christian Rossow. What Cannot be Read, Cannot be Leveraged? Revisiting Assumptions of JIT-ROP Defenses. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 139–156, 2016.

[62] Marc Laliberte. IoT Botnets Are Evolving – How Big Can They Get? https://www.secplicity.org/2018/02/20/iot-botnets-evolving-big-can-get/, 2018.

[63] Hector Marco-Gisbert and Ismael Ripoll. On the Effectiveness of Full-ASLR on 64-bit Linux. In *Proceedings of the In-Depth Security Conference*, 2014.

[64] Martin Abadi, Mihai Budiu, Ulfar Erlingsson, and Jay Ligatti. CFI: Principles, Implementations, and Applications. In *Proc. ACM Conference and Computer and Communications Security (CCS)*, 2005.

[65] Jon Martindale. From Pranks to Nuclear Sabotage, This is the History of Malware. https://www.digitaltrends.com/computing/history-of-malware/, 2018.

[66] Luca Nagy. Virus Bulletin: Exploring Emotet, An Elaborate Everyday Enigma. https://www.virusbulletin.com/virusbulletin/2019/10/vb2019-paper-exploring-emotet-elaborate-everyday-enigma/, 2019.

[67] John V. Neumann. Theory of Self-replicating Automata. *University of Illinois Press, Urbana*, pages 63–78, 1966.

[68] Lindsey O'Donnell. Emotet Now Hacks Nearby Wi-Fi Networks to Spread Like a Worm. https://threatpost.com/emotet-now-hacks-nearby-wi-fi-networks-to-spread-like-a-worm/152725/, 2020.

[69] Olivier Bilodeau and Thomas Dupuy. Dissecting Linux/Moose. https://www.welivesecurity.com/wp-content/uploads/2015/05/Dissecting-LinuxMoose.pdf, 2015.

[70] Sunoo Park, Michael Specter, Neha Narula, and Ronald L. Rivest. Going From Bad to Worse: From Internet Voting to Blockchain Voting, 2020.

[71] Shravya Paruchuri, Andrew Case, and Golden. G. Richard III. Gaslight Revisited: Efficient and Powerful Fuzzing of Digital Forensics Tools. In *Computers and Security*, volume in press, 2020.

[72] Sergio Pastrana and Guillermo Suarez-Tangil. A First Look at the Crypto-mining Malware Ecosystem: A Decade of Unrestricted Wealth. In *Proceedings of the Internet Measurement Conference*, pages 73–86, 2019.

[73] pymnts.com. Global-fraud-index-october-2017-1-min.pdf. https://securecdn.pymnts.com/wp-content/uploads/2019/07/Global-Fraud-Index-October-2017-1-min.pdf, 2017.

[74] Fahmida Y. Rashid. Disinformation Attacks Aren't Just Against Elections. https://duo.com/decipher/disinformation-attacks-aren-t-just-against-elections, 2019.

[75] Reuters. Uzbek Spies Attacked Dissidents With Off-the-Shelf Hacking Tools. https://www.reuters.com/article/us-uzbekistan-cyber/uzbek-spies-attacked-dissidents-with-off-the-shelf-hacking-tools-idUSKBN1WI0YL, 2014.

[76] Golden G. Richard III. Kernel Version-Independent Tools for Deep, Live Digital Forensics Investigation. *Proceedings of the 62nd Annual Meeting of the American Academy of Forensic Sciences (AAFS)*, 2010.

[77] Golden G. Richard III and Irfan Ahmed. Compressed RAM and Live Forensics. *Proceedings of the 66th Annual Meeting of the American Academy of Forensic Sciences (AAFS)*, 2014.

[78] Monica Ruiz. The Changing Landscape of Disinformation and Cybersecurity Threats: A Recap From Verify 2019. https://hewlett.org/the-changing-landscape-of-disinformation-and-cybersecurity-threats-a-recap-from-verify-2019/, 2019.

[79] Brendan Saltaformaggio, Rohit Bhatia, Xiangyu Zhang, Dongyan Xu, and Golden G. Richard III. Screen after Previous Screens: Spatial-Temporal Recreation of Android App Displays from Memory Images. In *USENIX Security*, 2016.

[80] skape and Skywing. Uninformed - vol 2 article 4. http://uninformed.org/?v=2&a=4, 2005.

[81] skape and Jarkko Turkulainen. Remote Library Injection. http://www.hick.org/code/skape/papers/remote-library-injection.pdf, 2004.

[82] Wei Song, Heng Yin, Chang Liu, and Dawn Song. Deepmem: Learning Graph Neural Network Models for Fast and Robust Memory Forensic Analysis. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 606–618, 2018.

[83] Anthony Spadafora. Major Data Breach Exposes Database of 200 Million Users. https://www.techradar.com/news/major-data-breach-exposes-database-of-200-million-users, 2020.

[84] Chad M Steel. Technical SODDI Defenses: The Trojan Horse Defense Revisited. *Journal of Digital Forensics, Security and Law*, 9(4), 2014.

[85] Stephen Checkoway, Lucas Davi, Alexandra Dmitrienko, Ahmad-Reza Sadeghi, Hovav Shacham, and Marcel Winandy. Return-oriented Programming Without Returns. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, pages 559–572, 2010.

[86] Susan Greenhalgh, et al. Letter to CISA. https://context-cdn.washingtonpost.com/notes/prod/default/documents/4ac156b8-9f4d-4df1-95d8-2be023c2559c/note/ba1499e1-4f98-4b96-9dbf-afe99d96e6e0., 2020.

[87] Dan Swinhoe. The 15 Biggest Data Breaches of the 21st Century, CSO Online. https://www.csoonline.com/article/2130877/the-biggest-data-breaches-of-the-21st-century.html, 2020.

[88] Joe Sylve, Andrew Case, Lodovico Marziale, and Golden G. Richard III. Acquisition and Analysis of Volatile Memory from Android Devices. *Digital Investigation*, 8(3), 2012.

[89] Paul Syverson, Roger Dingledine, and Nick Mathewson. Tor: The Second Generation Onion Router. In *Proceedings of Usenix Security*, pages 303–320, 2004.

[90] targetsmart. COVID-19 and Elections – Findings from a National Poll of American Voters. https://insights.targetsmart.com/covid-19-and-elections-findings-from-a-national-poll-of-american-voters.html, 2020.

[91] The MITRE Corporation. Persistence, Tactic TA0003. https://attack.mitre.org/tactics/TA0003/, 2019.

[92] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and Beyond: A Survey of Face Manipulation and Fake Detection. *arXiv preprint arXiv:2001.00179*, 2020.

[93] Liam Tung. Home Router Warning: They're Riddled with Known Flaws and Run Ancient, Unpatched Linux. https://www.zdnet.com/article/home-router-warning-theyre-riddled-with-known-flaws-and-run-ancient-unpatched-linux/, 2020.

[94] Giannis Tziakouris. Cryptocurrencies—A Forensic Challenge or Opportunity for Law Enforcement? An INTERPOL Perspective. *IEEE Security & Privacy*, 16(4):92–94, 2018.

[95] U.S. Office of Personnel Management. Cybersecurity incidents. https://www.opm.gov/cybersecurity/cybersecurity-incidents/, 2019.

[96] U.S. Senate Select Committee on Intelligence. Russian Active Measures Campaigns And Interference In The 2016 U.S. Election, Volume 2: Russia'S Use Of Social Media With Additional Views. https://www.intelligence.senate.gov/sites/default/files/documents/Report_Volume2.pdf, 2019.

[97] Luisa Verdoliva. Media Forensics and Deepfakes: an Overview. *arXiv preprint arXiv:2001.06564*, 2020.

[98] Volexity. PowerDuke: Widespread Post-Election Spear Phishing Campaigns Targeting Think Tanks and NGOs. https://www.volexity.com/blog/2016/11/09/powerduke-post-election-spear-phishing-campaigns-targeting-think-tanks-and-ngos/, 2016.

[99] Volodymyr Kuznetsov, László Szekeres, Mathias Payer, George Candea, R. Sekar, and Dawn Song. Code-pointer Integrity. In *10th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2014.

[100] Doron Voolf and Remi Cohen. Mirai "COVID" Variant Disregards Stay-at-Home Orders. https://www.f5.com/labs/articles/threat-intelligence/mirai-covid-variant-disregards-stay-at-home-orders, 2020.

[101] Peter Weidenbach and Johannes vom Dorp. Home Router Security Report 2020. https://www.fkie.fraunhofer.de/content/dam/fkie/de/documents/HomeRouter/HomeRouterSecurity_2020_Bericht.pdf, 2020.

[102] Wikipedia. List of Data Breaches. https://en.wikipedia.org/wiki/List_of_data_breaches#cite_note-135, 2020.

[103] Ximena Larkin. You Can't Just Get Up and Steal a Police Horse - The New York Times. https://www.nytimes.com/2020/07/01/style/dreadhead-cowboy-chicago.html, 2020.

[104] Bassam Zantout, Ramzi Haraty, et al. I2P Data Communication System. In *Proceedings of ICN*, pages 401–409. Citeseer, 2011.