Towards Cyber Attribution by Deception

Sampsa Rauti

University of Turku, Finland
sampsa.rauti@utu.fi

Abstract. This paper discusses a technical solution that will help to bring the cyber defenders and investigators one step closer to successful cyber attribution: deception technology. The goal is to detect abnormal activities taking place in the computer system by planting so called fake entities into the system. These fake entities appear to be interesting and valuable for the attacker. The deceptive defense mechanism then waits for the malicious adversary to interact with these fake entities. A fake entity can be anything from a fabricated file to a fake user account in a system. This paper takes a look at how different fake entities can be used for cyber attribution. We conclude that deception technology and fake entities have lots of potential for further development when trying to solve the challenge of cyber attribution.

Keywords: Cyber attribution, Deception, Fake entities

1 Introduction

Malicious cyberattacks often have serious implications for organizations and businesses. As a result of a cyber incident, the organization's reputation and finances can be harmed, and the whole operation of the organization can be disturbed. When an attack has taken place, it is important to launch an effective investigations in order to attribute the cyberattacks to an adversary. This helps in gaining a good understanding of the attack and bringing the perpetrators to justice.

Therefore, the process of *cyber attribution* is about attempting to find the culprit behind a malicious cyber incident [13]. It involves tracing and identifying the attacker, whether there is an individual or a group behind the attack. Cyber attribution is a challenging task. Due to the underlying architecture of the internet, there are several methods attackers can use to hide their traces.

The forensic investigations conducted by cyber experts are made more difficult, as the perpetrators usually use previously infected machines of other victims to launch new attacks. Attackers can also make the tracing process significantly more difficult by taking advantage of proxy servers around the world or IP spoofing to hide tracks, for instance. When the infected machine the attack comes from is located in another country, it is also often challenging to investigate the cyber incident effectively, because help needs to be requested from foreign law enforcement agencies. The fact that most operating systems and user applications have not really been designed with attribution in mind also makes things more difficult.

These challenges are not easy to solve, and one single technology or legislative change alone will not be enough. In this study, we will look at one technical solution that will help to bring the investigators one step closer to successful cyber attribution: *deception technology.* The idea here is to detect abnormal activities taking place in the system by placing *fake entities* withing a device or a computer network [16]. These entities are something that seems interesting and valuable for the attacker [22]. The deceptive defense mechanism then waits for the malicious adversary to use these fake entities. A bogus entity can be anything from a fabricated file to a fake service (honeypot server) in a network. However, fake entities do not need to be technical resources, the attacker can also be fooled by presenting information on nonexistent individuals. In this study, we will focus mainly on fake entities inside a single computer.

The rest of the paper is structured as follows. Section 2 gives a general introduction to fake entities and deception. Section 3 discusses how deception can be used in achieving better results in cyber attribution. Section 4 presents the discussion and Section 5 concludes the paper.

2 Deception

Many traditional countermeasures are not keeping pace with advanced cyber attacks like advanced persistent threats (APT) or malicious insiders [5, 10, 24]. Because these countermeasures have become ineffective against sophisticated threats in many cases, there is a need for novel methods of defending critical systems. One emerging approach to prevent the attacker from accomplishing his or her objectives in the target system is to make use of deception technology [1, 26]. Deceptive security mechanisms give false cues to the attacker in order to detect the attack and learn more about the attacker.

Conventional security measures usually aim to prevent the adversary's malicious actions directly: a firewall helps prevent malicious software and attackers from infiltrating a computer or a network, while encryption prevents the adversary from reading and altering confidential information. Instead, deception aims to manipulate the adversary's actions so that they become beneficial to the defender [7]. Because of this fundamental difference from conventional security solutions, deception can compensate for weaknesses in traditional defensive measures. What is more, deception can be used to gather many kinds of useful information on the adversary when he or she falls into the defender's trap. Yuill [29] says deception is *"planned actions taken to mislead attackers and to thereby cause them to take (or not take) specific actions that aid computer security defenses"*. This definition of deception in cyberspace has the following interesting properties:

– *Improve security proactively.* Fake entities make things more challenging for a piece of malware because it can no longer trust the system it is operating in. This security measure is proactive because it is not necessary to know in advance the fingerprint of the malware or the methods the attackers chooses to use. The goal is not to keep the malicious program out of the system. Instead, the adversary is attracted to interact with fake entities.

– *Observe and analyze the adversary's actions.* As we are aiming to learn more about the attacker so that the cyber incident can be attributed, we need to learn what exactly the attacker does in the system. In some intrusion detection systems and honeypots (fake systems that attract the attackers), the main focus is in detecting suspicious activity and setting off an alarm, but here we are more interested in monitoring the attacker's tracks and behavior, and logging this information for further analysis. With this data, the chances for effective attribution are increased if the investigators are able to use it proficiently.
– *Manipulate the attacker's course of action.* Based on the attacker's actions, the surrounding environment seen by the malware malware sees could be changed. When the malware encounters something unexpected, the course of action it chooses will probably be different and we can learn more about the malware by analyzing its behavior in different situations [2]. Machine learning can made use of here to better deceive the malware. Manipulating the malicious program's actions gives more information to be used for attribution.
– *Waste the perpetrator's resources.* Deception also has a useful property of wasting the attacker's resources. Exploring a honeypot that does not really contain anything useful for the adversary in the end requires time. Analyzing a fabricated reply from a network service uses the attacker's computational resources.
– *Alleviate the adverse effects of malicious program.* The malicious changes that malware or the attacker tries to achieve do not usually have any effect in the deceptive environment, or the system can be easily be returned into a pristine state after adverse changes. A malicious program can also be deceived into believing it has made changes to the system while in reality this is not the case. What is more, the propagation of malware can also be prevented.

3 Using Fake Entities for Attribution

3.1 What is a fake entity?

Anything in the system the attacker is going to interact with is suited to be a fake entity. It can simply be data in a system – a file, a record in the database, or a fake password inside a file. The size of these fake entities varies from a small pieces of data to entire bogus networks. Fake entities can also be embedded into the exchange of messages between the system and the adversary – for example, a reply to a malicious query can be fabricated.

An important characteristic of fake entities is that they have no authorized uses. In other words, interaction with them is always suspicious. This makes it easy for the defender to detect abnormalities in the system and observe malicious activities. For instance, a supposedly sensitive document containing business information could be planted in the system. In fact, however, this file would have no real value and no legitimate use for the users of the system. When the adversary attempts to open the file, we can immediately assume someone is trying to breach the organization's privacy. We can see that in its simplest form, deception with fake entities is very simple; just wait the attacker to access the fake entity, then raise an alert and log the attacker's information. This is much simpler, easier and cheaper than many other detection technologies. Also, the fake entities are one of the only existing ways to detect APTs and zero day exploits.

3.2 Different types of fake entities and attribution

Honeytokens Honeytoken is a piece of data that appears to be valuable and important but is actually fake [22]. These pieces of fallacious information can be planted into files and databases, for instance. For example, in a password file, the security of hashed passwords can be improved by adding honeywords among the real passwords [8]. When an adversary succeeds in cracking the password file and inverting its contents, he or she cannot tell whether a specific entry is a real password or not. In the same manner, false information about user accounts [2] or credit numbers [9] can also be planted into databases or files.

When the honeyword is used somewhere – for instance in an login attempt – we know that malicious activity is going on in the system. After a file or database record containing a honeyword has been accessed, we can change the honeyword. This way, each attacker gets a unique honeyword, and if that word is detected somewhere else, we know that this is the same attacker. This information can help us in attributing the adversary.

Files and directories One relatively easy and stealthy way to make fake entities is to plant fake files into the system. These files files that act as decoys and file accesses are logged [30]. Fake files can be simply traps to catch the adversary and immediately raise an alarm [28], but since we are interested in learning more about the attacker and ultimately attribution, it is more interesting to simply monitor the attacker's actions. The adversary can also be deceived by including supposedly valuable honeytokens, such as credentials, in the file.

The idea of fake files can be generalized to fake directories [19]. The file names and directories are designed to look enticing to the adversary. The files that are displayed in the directory listing do are not necessarily real. Instead, the attacker can be given a generic error message (for example a message stating that the file access is not authorized).

For attribution, the information on which process and user opened the file and when, and whether the file was edited. Information about all file accesses can then be combined to help to build a profile for the adversary. In the same way, the directory listings displayed by the attacker should be logged.

Databases Databases can either be completely fake or honeytokens can be planted into a database. Much like fake information in files, honeytokens act as traps that nobody should ordinarily touch [17]. Fake databases or fake records in a database can then be monitored and all accesses and other operations the adversary carries out are logged for further analysis.

Honey tokens can be quite easily generated and deployed in wide array of various database systems. This kind of database traps can be particularly useful when attempting to spot insider threats [17]. For instance, if an employee inside an organization commits privacy violations, a fake record may help to catch the offender.

Memory Memory areas are also potential fake entities [31]. By reserving certain memory areas for deceptive purposes and possibly placing honeytokens in these areas, memory-scraping malware looking for sensitive information can be caught. It is important to keep track what happens in the memory because sensitive information that is encrypted on the disk is usually edited in memory as plain text, which is why some malicious programs attempt to scan the memory to find this information. When a piece malware accesses a honey entity in memory, the activity is logged.

Metadata So far, we have covered ways to falsify entities with actual data (such as files or databases), but metadata is also used by malicious attackers. Therefore, information associated with the system can be lied to the adversary. Entities associated with system configuration such as environment variables or registry key entries (in Windows) can be faked. Files and databases also have meta information that can be falsified, e.g. file creation and modification times. Fake metadata is also easy to create automatically, unlike some types of actual content (for example, consider a file containing a believable fake business plan).

An interesting practical example about how metadata can be used to deceive the adversary and help with attribution is presented by Spafford [21]. He proposes an approach that causes files to look really large by using sparse files in Unix-based systems. The file size can be just couple of thousand bytes on disk, but when the adversary attempts to copy the file remotely, the copy process will take forever to complete. Meanwhile, the network connection stays open and the defender can start tracing the attacker.

Security patches Software security patches can be used to deceive the adversary [3, 4]. These honey patches are equivalent to ordinary patches in terms of security. However, when the attacker tries to exploit the patched vulnerability, the attack is redirected to a vulnerable, unpatched decoy. The adversary now thinks that the attack has succeeded, and his or her actions can then be monitored and recorded. By deploying several different honey patches in a system, we can raise the probability that an attacker is caught and learn more about his or her intentions and objectives.

Services and interfaces Besides memory, we can also set other traps for the malicious attacker on operating system level. One interesting idea is to deceive the adversary by faking the system call interface. System calls are a mechanism for applications to interact with the operating system and use the critical resources of the computer. All important tasks in the system such as creating files or opening network connections are ultimately carried out by issuing system calls. In a decoy system, it is possible to change the mapping of the system calls so that the trusted programs in the system use new, secret system call interface, but the malware (that does not know about this secret interface) still uses the original, well-known interface [6]. As the malicious program does not know the new system calls and uses the "fake original" interface, calls to the implementations of original system calls can be monitored. We can also pretend that the system calls made by malware have the desired effect by giving the malware convincing replies to its requests. Laurén et al. have presented a proof-of-concept implementation for changing system call interface in the fashion described above [12].

Fake interfaces can also be used in many other contexts. For instance, operating system libraries can be modified (for example by changing function names and leaving the original functions in place to catch malware) [11]. Another example is modifying the command shell language (by changing the commands of the shell language and leaving the old keywords as traps for the malware) [23].

Be deceiving malware and observing what kind of system calls and function calls it attempts to issue in the system, we can often get a deeper understanding on how the malicious adversary operates and what kind of the objectives the adversary has.

Errors We do not always have to return real content to the adversary in form of honeytokens in order to deceive him or her. Another option is to give a false reply to an adversary when he or she tries to interact with a service or tries to access a resource [2]. The attacker's request can be denied and the system can pretend that an error has occurred [19]. We can for example claim that the command adversary has attempted contains a syntactical error or that the requested service is not available. The cyber intruder can also be offered large amount of information or made wait a long time in order to delay him or her. These tricks may draw out new malicious behavior that we are interested in.

There are also several general excuses that one can use to see how a malicious adversary reacts. The attacker can be informed that the system is down, the network is not available or maintenance is being performed in the system and therefore, it may not work as expected [18]. In many situations, we can also lie that an operation has succeeded even though this is not the case. It is worth noting that flooding the adversary with too many error messages may raise suspicions.

Activity in the system Activities happening in the system can also be faked [27]. For example, processes in the system opening and modifying files, and seemingly interesting fake network traffic leaving from and arriving in the system. Having a set of fake activities going on in the system and monitoring how the adversary reacts to them shows us what the adversary is interested in and creates more opportunities to observe his or her actions.

Persons Information on people that do not exist, also known as honey people, can be published on a social media or websites, for instance [25, 27]. These fake persons are made to look realistic and appear to be connected with the real people in the same community or organization. Fake emails and user accounts can be created for honey people to lure attacker into a honeypot [20]. Behind the fake user account, the system can contain several bogus resources such as files, passwords and address books.

4 Discussion

In the "Guide to Cyber attribution", US Office of the Director of National Intelligence discusses the key indicators that enable attribution [15]. The important indicators include tradecraft, intent, malware and infrastructure. Tradecraft refers to the behavior of the attacker when conducting cyber attacks. By making the attacker interact with honey entities, collecting information about his or her actions, and by analyzing these actions using machine learning and human assessment, we can get a good understanding about the behavior of the attacker. While technical details of the attacks change, it is much more difficult for the attackers to suddenly change their habits and behavior. We believe that analyzing the data collected with the help of diverse fake entities in the target system, profiles can be built for attackers. This fact can be used to breach the apparent anonymity of cyber intruders. Of course, when attack patterns become public, other attackers may also start using them and their significance in profiling perpetrators will diminish.

Along with behavioral patterns, it is important to analyze the intent of the attackers. By monitoring what kind of data the attacker attempts to access or modify and what kind of operations (such as system calls) he or she performs in the system, we can learn a lot about the attackers objectives. Using several kinds of fake entities along with different types of monitoring tools will hopefully help to create a comprehensive picture of the adversary's intent.

We are not only profiling the attackers, but also the malware they use. Many technical details, such as the system call sequences and system call parameters a malicious program uses can be used to detect and profile malware [14]. The way malware interacts with fake entities and reacts the unexpected situations (e.g. receiving a deceptive error message or a fake reply to a request) can tell us a lot about the malware. We can also capture the malicious executable for further analysis in our honeypot system.

Finally, attackers can compromise, lease and use several different resources (e.g. cloud services, servers and networks) in order to build the infrastructure needed to launch cyberattacks. While many attackers change infrastructure frequently to avoid detection, some can launch several attacks from the same platform or platforms.

While analyzing the key indicators discussed above, it is also important to keep in mind that success in attribution has often been a result of some kind of a human error made by the attacker [15]. For example, the intruders have often been careless when it comes to tradecraft and usage of internet infrastructure. We believe that in many ways, fake entities and honey tokens are an excellent method to lure the attacker into making mistakes that can help with the attribution. The more the adversary can be enticed to interact with the deceptive system, the more room there is for critical mistakes.

Cyber deception is an emerging field, and while deceptive cyber defense has been discussed widely in the academic literature, practical real-world solutions are still limited. We hope the ideas discussed in this paper can help to move cyber defenses towards using more deceptive solutions to protect information systems and attribute the adversaries.

5 Conclusions

Many traditional countermeasures against cyber attacks are not keeping up with advanced cyber threats such as advanced persistent threats (APT) or malicious insiders. Still, there is a great need for detecting these attacks and finding the culprit behind the cyber breaches. While cyber attribution is not an easy task, deception technology shows a lot of promise in detecting and helping to attribute advanced cyber attacks. In this paper, we have discussed different fake entities that can help catching advanced malware and profiling the attacker. We believe deception technology and fake entities hold lots of potential for future development when trying to solve the challenge of cyber attribution.

References

1. Al-Shaer, E., Wei, J., Hamlen, K.W., Wang, C.: Towards intelligent cyber deception systems. In: Autonomous Cyber Deception. Springer (2019) 21–33
2. Almeshekah, M., Spafford, E.: Planning and Integrating Deception into Computer Security Defenses. In: Proceedings of the 2014 workshop on New Security Paradigms Workshop, ACM (2014) 127–138
3. Araujo, F., Hamlen, K.W., Biedermann, S., Katzenbeisser, S.: From patches to honey-patches: Lightweight attacker misdirection, deception, and disinformation. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security, ACM (2014) 942–953
4. Araujo, F., Shapouri, M., Pandey, S., Hamlen, K.: Experiences with honey-patching in active cyber security education. In: 8th Workshop on Cyber Security Experimentation and Test ({CSET} 15). (2015)
5. Bejtlich, R.: The Practice of Network Security Monitoring: Understanding Incident Detection and Response. No Starch Press (2013)

6. Chew, M., Song, D.: Mitigating buffer overflows by operating system randomization. (2002)
7. Cohen, F., Koike, D.: Misleading attackers with deception. In: Proceedings from the Fifth Annual IEEE Information Assurance Workshop, IEEE (2004) 30–37
8. Juels, A., Rivest, R.: Honeywords: Making Passwordcracking Detectable (2013)
9. Kambow, N., Passi, L.K.: Honeypots: The need of network security. International Journal of Computer Science and Information Technologies **5**(5) (2014) 60986101
10. Karuna, P., Purohit, H., Ganesan, R., Jajodia, S.: Generating hard to comprehend fake documents for defensive cyber deception. IEEE Intelligent Systems **33**(5) (2018) 16–25
11. Laurén, S., Mäki, P., Rauti, S., Hosseinzadeh, S., Hyrynsalmi, S., Leppänen, V.: Symbol diversification of Linux binaries. In: World Congress on Internet Security (WorldCIS-2014), IEEE (2014) 74–79
12. Laurén, S., Rauti, S., Leppänen, V.: An interface diversified honeypot for malware analysis. In: Proccedings of the 10th European Conference on Software Architecture Workshops, ACM (2016) 29
13. Lin, H.: Attribution of malicious cyber incidents: From soup to nuts. Journal of International Affairs **70**(1) (2016) 75–137
14. Mutz, D., Valeur, F., Vigna, G., Kruegel, C.: Anomalous system call detection. ACM Transactions on Information and System Security (TISSEC) **9**(1) (2006) 61–93
15. Office of the Director of National Intelligence: A guide to cyber attribution (2018)
16. Rauti, S., Leppänen, V.: A survey on fake entities as a method to detect and monitor malicious activity. In: 2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), IEEE (2017) 386–390
17. Rietta, F.S.: Application layer intrusion detection for sql injection. In: Proceedings of the 44th annual Southeast regional conference, ACM (2006) 531–536
18. Rowe, N.C.: Designing good deceptions in defense of information systems. In: 20th Annual Computer Security Applications Conference, IEEE (2004) 418–427
19. Rowe, N.C.: A model of deception during cyber-attacks on information systems. In: IEEE First Symposium on Multi-Agent Security and Survivability, 2004, IEEE (2004) 21–30
20. Rowe, N.C.: Deception in defense of computer systems from cyber attack. In: Cyber Warfare and Cyber Terrorism. IGI Global (2007) 97–104
21. Spafford, E.: More than passive defense. https://www.cerias.purdue.edu/site/blog/post/more_than_passive_defense/ (2011)
22. Spitzner, L.: Honeypots: Tracking Hackers. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2002)
23. Uitto, J., Rauti, S., Mäkelä, J.M., Leppänen, V.: Preventing malicious attacks by diversifying linux shell commands. In: SPLST. (2015) 206–220
24. Virvilis, N., Gritzalis, D.: The Big Four - What We Did Wrong in Advanced Persistent Threat Detection? In: 2013 International Conference on Availability, Reliability and Security. (2013) 248–254
25. Virvilis, N., Vanautgaerden, B., Serrano, O.S.: Changing the game: The art of deceiving sophisticated attackers. In: 2014 6th International Conference On Cyber Conflict (CyCon 2014), IEEE (2014) 87–97
26. Wang, C., Lu, Z.: Cyber deception: Overview and the road ahead. IEEE Security & Privacy **16**(2) (2018) 80–85
27. Wang, W., Bickford, J., Murynets, I., Subbaraman, R., Forte, A.G., Singaraju, G.: Catching the wily hacker: A multilayer deception system. In: 2012 35th IEEE Sarnoff Symposium, IEEE (2012) 1–6

28. Whitham, B.: Canary files: generating fake files to detect critical data loss from complex computer networks. In: Second International Conference on Cyber Security, Cyber Peacefare and Digital Forensic (CyberSec2013), Malaysia. (2013)
29. Yuill, J.: Defensive Computer-security Deception Operations: Processes, Principles and Techniques. PhD thesis, North Carolina State University (2006)
30. Yuill, J., Zappe, M., Denning, D., Feer, F.: Honeyfiles: deceptive files for intrusion detection. In: Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004., IEEE (2004) 116–122
31. Zeltser, L.: Detecting memory-scraping malware (September 26 2017) US Patent 9,774,627.