2017/2500M-0069

ISSN XXXX-XXX (verkkojulkaisu)
ISBN (PDF) xxx-xxx-xx-xxxx-x

SUMMARY REPORT (TIIVISTELMÄRAPORTTI)

ICT Illusions
(In Finnish: Tietotekniset harhautukset)

Sampsa Rauti, Ville Leppänen, Jani Tammi and Jarko Papalitsas
University of Turku, Department of Future Technologies
sjprau@utu.fi, ville.leppanen@utu.fi, jasata@utu.fi, jastpa@utu.fi

Abstract

Cyber attacks and cyber intelligence are growing concerns in today's computer networks. The goal of our research project was to address this threat and study generating deceptive fake services in a new way: by recording the ordinary interaction between a client and a server and then deceiving the adversary by playing back an altered, deceptive reply. We demonstrated the feasibility of such fake service generator tool by creating a proof-of-concept implementation that can create fake services for web environment. The objective was to make the tool as protocol-independent as possible while still giving convincing replies to the adversary. Our results indicate that although fake service creation is a challenging topic, can still achieve very satisfactory results with our proof-of-concept implementation. We therefore believe creating deceptive services is a promising proactive method to lead the adversary astray as he or she is attempting to get access to sensitive data.

1. Introduction

Cyber intelligence and cyber attacks in general are widespread in computer networks today and their relevance will grow in days to come. However, the cyberspace differs from the real world in the sense that an attacker cannot easily decide whether the service it makes use of is in fact a genuine service. This is because the attacker has to make this judgement mostly based on a reply message it receives from the server, and this reply can be fabricated.

Deception is a promising method to achieve software security and privacy. Clifford Stoll described the use of deceptive resources in the context of computer security already in 1989. Bill Cheswick went on to describe the use of an early honeypot system to monitor and study behavior of an adversary in 1992. Since then, the use of several different fake resource categories has been suggested in research as the number and sophistication of cyber-attacks have gone up over the years. Today, it is widely agreed that in many cases, traditional security measures are not enough to defend against advanced attackers. There is a real need for novel proactive approaches.

The idea of our solution is therefore to deceive the adversary by effectively and extensively creating fake services that feed the adversary false information. At the same time, the attacker can also be tricked into giving us valuable information on its objectives. This is important in order to perform some counter-intelligence of our own and to keep in pace with the techniques cyber attackers use.

The main purpose of our project was to implement a tool that is able to generate fake services. In the response messages created by the fake service, the original content has been replaced with fallacious information. To be more specific, messages sent over the network contain *entities*, pieces of data (e.g. names and locations), that are references to things and objects existing in the real world. In the fake services built with our tool, the entities in the re-

sponse messages are replaced with *fake entities* to deceive the adversary.

The real challenge here, of course, revolves around the question how to recognize the important entities (the parts we want to lie about) in the message payloads and how to replace them with the data that is as convincing and enticing for the adversary as possible. Our research also addresses these questions.

In this report, we present the results of our MATINE research project, "ICT Illusions". Section 2 explains our research goals and the objectives for the project. Section 3 covers the theoretical framework and methods used in our study. In Section 4, we summarize the results of the project. Finally, Section 5 concludes the report. In Section 6, publications related to the project are listed and summarized.

2. Research objectives and accomplishment plan

The objective of our project was to study generating deceptive fake services and using them to lead the adversary astray while analyzing his or her actions. As a more concrete goal, the purpose of the project was to show that it is possible to build a practical fake service generator tool. Based on the recorded transactions between a client and a server, the services generated by this tool learn how to deceive the adversary in a convincing way.

This setup is shown in Figure 1. First, ordinary interaction between a client and the genuine service is recorded. Based on the recorded samples (that is, request-response pairs), a fake service then plays an altered deceptive reply to the adversary. We call this the *record and play -approach*.

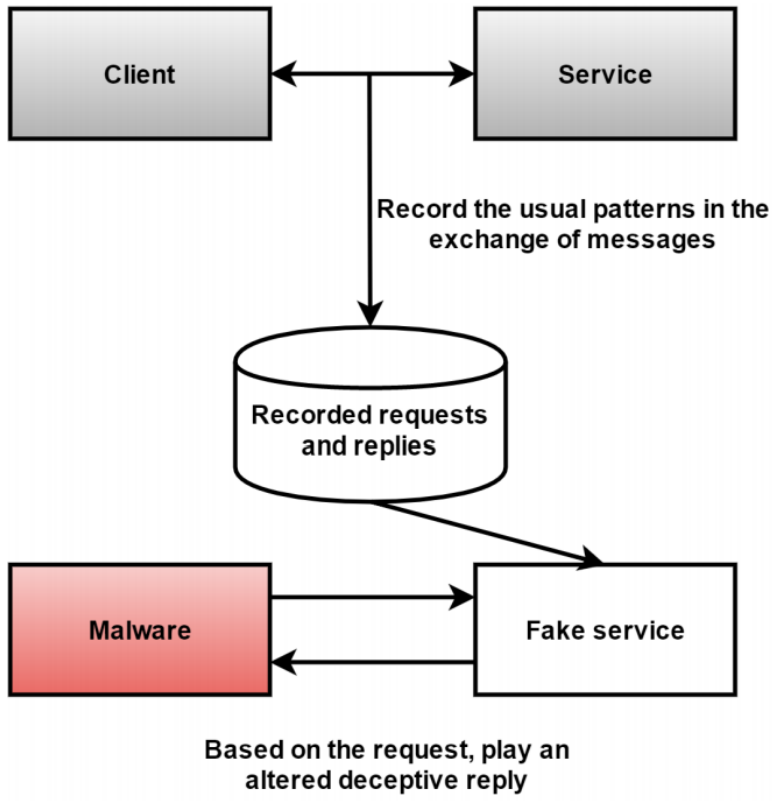

Figure 1. The idea of a record and play -honeypot.

Looking at the concrete objectives in more detail, the following goals were identified:

1. **Create a conceptual scheme for a record and play -honeypot that can create convincing responses for fake services based on previously recorded transactions. Also build a proof-of-concept implementation for this tool.**

   Although we wanted the tool to be general in the sense it can be applied to any text-based protocol with minimal changes, the proof-of-the-concept implementation had to be limited to some specific application area. Therefore, we aimed at building a record and play -honeypot operating in web environment. We also limited our attention to faking the payload of the messages (and not any other related metadata).

   The phases to be implemented were identified as capturing the network traffic, preprocessing the message contents, recognizing the entities in the payload and generating convincing entities. As stated previously, the most significant challenge in the context of deceptive services is entity recognition, which we aimed to implement with minimal manual work required from the user of the tool.

2. **As a part of the previous goal, create mechanisms to learn and generate convincing fake entities. This includes modeling certain entity categories such as names of Finnish persons etc.**

   Naturally, creating convincing entities is an important part of our honeypot tool, especially in the cases where we assume that a human is attacking our service and our fake data is exposed to human scrutiny. From the recorded messages, the tool can learn about typical entities. Naturally, it is not always possible to create convincing data automatically for all entity types, so in many cases we have to make use of manually crafted lists.

3. **Survey the literature to find what kind of fake resource categories have already been used previous work for deceptive purposes and also propose some new categories.**

   Web environment is an interesting application area for deceptive services, but definitely not the only one. Therefore our goal was also to study what kind of deceptive services already exist and what kind of fake resource categories (e.g. files, database entries, system calls?) they employ.

4. **Build a demonstration that showcases the operation and feasibility of our record-and-play honeypot.**

   Using our proof-of-concept implementation as an example, we wanted to make a demonstration (a video) to illustrate how our tool is used when it is applied to a real service in practice.

3. Materials and methods

The methods of our study consist of iteratively innovating conceptual solutions by building proof-of-concept implementations of designed software tools. A thorough survey of previous work in the field of fake service generation and potential fake resource categories was also carried out.

We first created a conceptual scheme for a record and play -honeypot that generates convinc-

ing fake entities based on recorded request-response pairs. In our proof-of-concept implementation, we applied this idea to a web service. In what follows, we will present the phases of our record and play -scheme that also correspond to the phases we went through when developing our proof-of-concept tool. The operation of our tool is depicted in Figure 2.

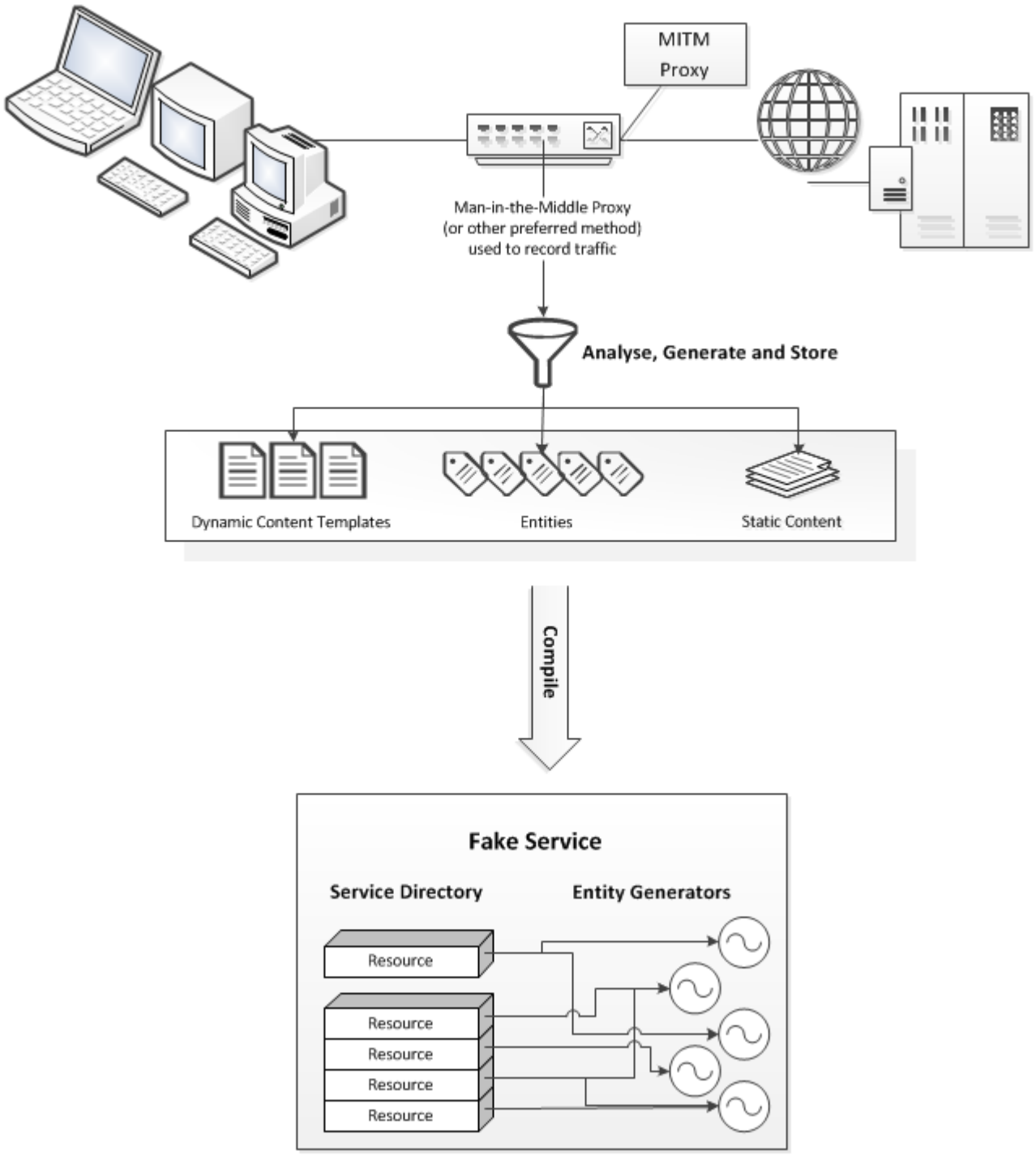

Figure 2. The phases of our record and play -scheme.

First, our tool imports data gathered by a man-in-the-middle proxy that **captures and records network traffic** between the client and the service. Request-response pairs are stored into a database.

The recorded request-response pairs then need to be **pre-processed** to identify which parts of the content are dynamic, which of the dynamic parts are actual entities we want to alter and also which dynamic access parameters -- attributes that affect the content the web server hands out.

The process of locating the portions of the content that show changes within the sample set is in principle quite similar to commonly used diff tools in source code management. A difference, however, is that traffic samples are not incremental and all their content should be analyzed simultaneously as opposed to pairs.

Because of the inherent challenges in both automatic dynamic field detection and entity recognition, **operator assistance** is required during the pre-processing phase. We therefore developed a graphical user interface to facilitate this laborious activity. With our interface, a human operator can inspect and edit the dynamic fields in an intuitive manner.

The pre-processing phase concludes with an algorithm that determines the most relevant subset of dynamic access parameters for each dynamic field. All the static parts of a dynamic resource are condensed into a template. Each field will be indexed in relation to the template while retaining their order of appearance. Input to output look-up tables are created for non-generated entities and to assist in the process of entity creation.

From the data produced during the pre-processing phase, **entities and service objects are generated** with the help of previously compiled input-output tables. The specialized modules covering different entity categories are used for entity generation.

This phase also requires a human operator to craft the login and session components using good understanding on the authentication mechanisms employed by the selected live service. The fake service is now ready to be brought online.

4. Results and discussion

Our main contributions are a scheme for a record and play honeypot that can create deceptive fake services, and a practical proof-of-concept implementation of the scheme. This is the tool that was described in detail in Section 3. The conceptual scheme and its implementation correspond to research goal 1 (see Section 2). Publications 5 and 6 (see Section 6) describe the scheme and its implementation.

The implementation of the record-and play honeypot consists of around 7000 lines of code in Python and scripting languages. Many results with this tool are promising: the pre-processor can handle about 1000 request-response pairs in one minute on a normal workstation, and the parameter association mining algorithm (that finds connections between access parameters and dynamic fields) achieved 100 % precision with our testing data (although with some false positives). However, for example the entity recognition and consistence management still need further development. The record and play honeypot has also been quite extensively documented: there is a "master document" (of around 50 pages) describing the scheme and our implementation.

Prior to implementing the record-and-play honeypot, we also implemented “Honeyproxy”, a tool which can replace entities with fake ones but has no real record-and-play functionality. It can considered as a preliminary stage of the more advanced honeypot implementation. Honeyproxy consists of around 600 lines of code in Python. It is a quite simple search and replace -based implementation, but in our experiments it achieved a good precision (0.89) in en-

tity recognition and did not cause a significant performance penalty compared to the genuine service being faked. Honeyproxy is also able to create entities from many different categories such as Finnish names and addresses. This functionality corresponds to research goal 2. Honeyproxy is discussed in publications 2 and 4.

In keeping with research goal 3, a survey study on different fake resource categories used in deceptive services was also carried out (publication 1). We found that many kinds of fake resources – such as files, database records, passwords and complete honey networks – have been proposed in academic literature, although their usage in practice still seems to be quite limited. We also proposed some fake resources of our own (e.g. fake responses to system calls).

In accordance with research goal 4, a demo video that explains all the phases associated with building a fake service and shows how the tool is used was made. The video is publicly available on YouTube: https://www.youtube.com/watch?v=-MrPE4o2qu8

We have also written a publication addressing the challenges associated with record and play honeypots (publication 3). Generally, we also see identifying the many challenges associated with fake service creation as one important result of the project. These challenges include e.g. infeasibility of general and fully automatic entity recognition, creating convincing fake entities, keeping entities consistent inside and between messages, preventing the original data from leaking etc.

Despite the challenges, our results show that record and play -honeypots are a feasible method to deceive and confuse the adversary who is trying to collect information. We believe this is especially true for simple text based protocols that are easier to learn, although our proof-of-concept implementation in complex web environment also demonstrated very satisfactory results.

5. Conclusions

We have presented a conceptual scheme for record and play -honeypot that first records the ordinary interaction (requests and responses) between a client and a service and then plays altered deceptive replies back to the attacker. This scheme is protocol-independent and works proactively against attacks in which the adversary aims to gather information (such as cyber intelligence) in the target service.

We have also built a proof-of-concept implementation to demonstrate the feasibility of our scheme in web environment. Previously, we also built a simpler "Honeyproxy" tool that gives fallacious replies to the adversary. These tools and their components have been discussed in more detail in our publications.

Despite many challenges associated with fake entity recognition, creating convincing entities and dealing with intricacies of web environment, experiments performed with our tools show that with some manual help from an operator, entities can be successfully recognized and the concept of a fake service works in practice.

Of course, our proof-of-concept implementation of record and play honeypot is still far from being an actual product. The major topics for future work are as follows:

- Testing our current record and play -implementation with more diverse set of different web services.
- Combining the fake services with our previous implementation of a diversified honeypot. In this case, the fake service would have a secret diversified interface for the trusted clients, and the original service interface would be left as a decoy for malware.
- Implementing our conceptual scheme for new application areas (other than web envi-

ronment).

- Improving the proof-of-the-concept tool towards an industrial implementation. There are still many aspects in our tool (like automatic entity recognition, the graphical user interface for dynamic field annotation, and the setup process) that would benefit from further development.
- Further addressing and mitigating challenges that were identified during our project (achieving better protocol independence, minimizing the amount of manual work, creating convincing entities automatically etc.).
- Developing a component analyzing attacker's behavior. Although our current record and play -honeypot supports recording traffic (for further analysis), it does not support any automated analysis of adversary's behavior yet.

6. Scientific publishing and other reports produced by the research project

1. **Sampsa Rauti and Ville Leppänen (2017): A Survey on Fake Entities as a Method to Detect and Monitor Malicious Activity. In proceedings of the 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), pp. 386-390, IEEE.**

   This paper is a survey that summarizes previous research concentrating on fake resources as a method to detect and analyze malware. A fake resource is a digital entity (such as a file) that should only be accessed by a malicious attacker. When the resource is accessed, we can immediately know there is an unwanted malicious program in the system. The paper covers fake resources of different sizes, from resources found on one machine to whole networks using groups of honeypots. Some novel fake resource categories are also proposed.

2. **Jarko Papalitsas, Sampsa Rauti and Ville Leppänen (2017): A Comparison of Record and Play Honeypot Designs. In Proceedings of the 18th International Conference on Computer Systems and Technologies (CompSysTech'17), pp. 133-140, ACM.**

   There are several alternative ways to implement a record and play -honeypot system. Each of the options has its pros and cons that vary from the better accuracy of the fake responses to the possibility of causing side effects on the real services. In this paper, we present several potential designs for such honeypots. More specifically, we consider two important aspects in designing honeypots. First, we compare existing named entity recognition methods. Second, we discuss methods for consistently faking these entities. Benefits and drawbacks of each approach are discussed.

3. **Jani Tammi, Sampsa Rauti and Ville Leppänen: Practical Challenges in Building Fake Services with the Record and Play Approach. Accepted for publication in the proceedings of SIN2017, 4 pages.**

   Drawing from literature and our own practical experience, this paper outlines the challenges faced in development of a record-and-play honeypot. These include achieving protocol-independence, recognizing entities, creating convincing entities of different types, entity consistency, and prevention of data leaks, to mention a few. Solutions and recommendations that alleviate the issues are also presented.

4. **Jarko Papalitsas, Sampsa Rauti, Jani Tammi and Ville Leppänen: A honeypot proxy framework for deceiving attackers with fabricated content. Accepted for publication in CTI2017 (book), 20 pages.**

   This paper presents the general idea of deceiving attackers with fake services and fake

content in order to make cyber intelligence more difficult. We also discuss the ideal properties required from a honeypot that generates fabricated entities. We then introduce an implementation of honeypot proxy framework that generates fallacious content and present experiments demonstrating our implementation's accuracy and performance. The experiments show that despite many challenges, deceiving attackers by using fake services is a feasible and promising approach for protecting information systems and analyzing malicious programs.

5. **Jani Tammi, Jarko Paplitsas, Sampsa Rauti and Ville Leppänen: Recognizing Entities in Network Traffic with a Manually Assisted Solution. Submitted to WorldCIST'18, 10 pages.**

   This paper discusses recognition of dynamic fields in the message payloads, which is an important part of a record and play -honeypot but can also be used in many other applications like network traffic analysis. We present a manually assisted approach – specifically focused on web environment – that allows the user to easily annotate dynamic fields. Our graphical user interface for this purpose provides a simple and intuitive way to mark and edit dynamic fields.

6. **Sampsa Rauti, Jarko Papalitsas, Jani Tammi and Ville Leppänen: A record and play honeypot for deceiving adversaries. To be published.**

   This journal article, which is still being written at the moment, will discuss the main deliverable of our project: the complete record and play -honeypot implementation. The design and functionality of the solution will be explained in detail along with the strengths and limitations of the solution. Some experimental results demonstrating the feasibility and performance of the implementation will also be presented.