Phishing Attacks: Learning by Doing

Tom Chothia, Stefan-Ioan Paiu and Michael Oultram
Computer Science, The University of Birmingham

Introduction

Phishing, and particularly spear phishing, is a major security concern, however it is often not taught in any detail on security courses. Showing students examples of what they know to be phishing e-mails tends to give the incorrect impression that phishing is easy to spot and those that fall for phishing e-mails are foolish. Phishing students without their knowledge might be an effective way to teach students the dangers of phishing, but would lead to ethical and legal issues.

We have developed a framework in which students can try to perform phishing attacks against a simulated company[1]. The framework takes the form of a single VM which the students download and run on their own machines. On this VM the students find a website for a fictional company (with employee details), an e-mail client and common tools used for phishing.

Using what they can find out about the company employees the students need to carefully craft spear phishing e-mails. A script in the VM processes every e-mail sent by the student and uses rules to decide if they have produced a realistic spear phishing e-mail. If the e-mail passes this test then any attached executable, or any macros in office documents will be run. Hence, the students need to both craft a successful phishing e-mail and a payload. There is a docker container for each possible phishing victim, successful payloads may give the student a shell on this container, where they can find a flag, which they can submit to show they successfully completed a phishing attack.

Example Spear Phishing Attack

Common phishing attack payloads are macros in office documents, executables and fake websites, our framework has examples of all of these. To perform an attack the students must find and read the public Facebook profiles for the employees listed on the company website. This gives them details of what kinds of e-mails the employees might be expecting, and therefore might lead to a successful phishing attack.

As an example of one attack possible in our framework: while looking at the Facebook profiles the students may spot a post from the company boss to one of the interns, telling the intern that if he sends the boss his CV then she will look at it and give him career advice. This indicates that the boss is expecting a CV from the intern and therefore is likely to open it. For this attack, the student needs to send an e-mail to the boss from the address of the intern. The e-mail body must be address to the boss by name, be from the intern, and the body of the message must include at least 3 of the words ”advice”, ”CV”, thanks/thank you”, ”please”, or ”career”. If these are present, any macros in any attached office document will be run.

The payloads are run in a dedicated docker instance for the employee. The aim of the student is to get a shell on this docker instance. Once this is done they can find a flag for the employee. As well as this attack, the framework has 4 other similar attacks: one that uses a spreadsheet macro, two executable based attacks, and one phishing website attack.

Framework Technical Details

The framework is based on a VirtualBox VM of Ubuntu Linux and uses a docker container for each victim and another docker container to host the company website and a mail server. The advantage of using docker containers is that payloads can be executed and run as they would be on a stand alone machine, so creating a realistic environment. The containers also provide isolation between the student and the rest of the framework, e.g., meaning that the company website code and e-mail server are not visible.

1. Our framework and more information can be found at `www.cs.bham.ac.uk/~tpc/LearnToPhish`.

We use Puppet to automatically configure each docker container, installing the software, files and flags we need starting the web and mail servers, and configuring the network between the docker instances. This has the advantage of automating most of the set up procedure.

Each docker container that represents a possible phishing victim runs a Java program that polls the e-mail server every minute, if an e-mail addressed to the possible victim is found then the program applies checks based on the words and names that are contained in the message. If these checks pass then the script may either look for a Linux executable and run it, or look for a particular type of office document, extract the macros and run those. If any of the checks fail then the program sends a reply to the VM e-mail client (not the from address) giving a hint as to why it failed, (e.g. "I don't trust this e-mail because it wasn't addressed to me" or if it does not contain the words needed in the text "this e-mail doesn't seem relevant to me").

The Java program may also look for a URL in the e-mail, fetch the page and look for particular words and tags on the page, if they are found then the program will post a username and password to a form on the page. So making it possible for students to phish for log in details.

MetaSploit and Libre Office are installed for the students, which they can use to make phishing payloads. A web server is configured so that they can use it to make a phishing website. The e-mail IceDove client is configured on the machine to use the e-mail server in the company container (IceDove allows any from address to be entered for an e-mail, without needing to reconfigure the account details). The web browser is configured to have the companies main page as the default page so it is the first thing the student see when they open the browser.

Using Our Framework for Teaching

The key goal of this exercise is to get students to think carefully about phishing attacks, in particular we want to counter the idea that all phishing attacks are easy to spot and that people who fall for them are stupid. This is done by making students craft convincing e-mails, i.e., ones which they themselves might fall for. Secondary goals are to teach the mechanics of how phishing attacks happen, and to alert the students to the dangers of having personal information publicly available.

We give students an introduction to phishing attacks, briefly describing why people might accept phishing e-mails. We also demonstrate the creations of phishing payloads with MetaSploit and as Office macros, a hand out for students gives details of this and describes how to create a basic phishing website.

Students are asked to work in teams of 3 or 4 to discuss possible ideas for phishing attacks, and try them out on the VM. The aim of this part of the exercise is to encourage students to discuss phishing attacks, and so to gain an understanding of why they work.

While students are developing their attacks we will talk to the groups, discuss their ideas and help them get the attacks working. We found discussion of possible phishing attacks to be particularly useful. Discussion with students also helped clarify how the exercise worked and make sure students where working in the right directions. Some false paths students tried included messaging the people on Facebook, sending e-mails to people to try to gain their trust before sending the payload, searching for more sources of information as well as Facebook and the company website. While all of these were good ideas, they are not supported by our framework, and letting the students know that avoids them wasting time.

The phishing VM has been used at an on site capture the flag competition ($\sim$ 60 students), as an event for cyber security students at the University of New South Wales ($\sim$ 20 students), and as part of a Penetration testing Masters course at the University of Birmingham (22 students). These were all computer science students with an interest in cyber security, although most had not used MetaSploit or Office Macros before. In each case all teams could find 1 or 2 attacks with in 2 hours. The CTF ran for 6 hours and this was one of a range of challenges, teams that focused on this challenge found 4 or 5 of the attacks. For the course, students had a 2 hour supervised lab sessions and then a week to look for more attacks in their own time. It was compulsory to find three attacks, which all students did. Finding the 2 other attacks was for extra credit which 14 out of 22 students did.

The exercise was very popular, part of this seemed to be due to its novelty. Students were also interested in the framework and had a lot of questions about how it worked. Students universally said that the exercise raised their awareness and understanding of phishing attacks.

Further Work

We would like to carry out a formal assessment of our framework as a learning tool. We would also like to add in more sources of information about the employees that could be used for phishing attacks, e.g., Linkdin pages for the employees.

The simple MetaSploit generated and office macro payloads used in this exercise would be picked up by all anti-virus products and many most mail services would detect these and not deliver the e-mails. A way to increase the technical difficulty of the exercises would be to enable different grades of anti virus on the containers. This would require the student to carefully craft their payloads to avoid malware detection.