Honeypot that can bite: Reverse penetration

By Alexey Sintsov, Russian Defcon Group #7812

Introduction

The objectives of this work are to determine the benefits and opportunities in conducting counter attacks on a remote attacker. In this paper, we would like to focus on WEB threats which are some of the most pressing.

These days, the protective equipment is limited to passive defense, trying to prevent possible attacks or detect them. However, these methods have some drawbacks. For an example, as a result of any alarm that was triggered by IDS, we have to conduct several actions such as to parse the logs, to find out the consequences, to assess the damage and to plan our further actions. It is common that after the actions were done, we had found out that it was "false positive" or other case when we had spent the resources exceeding the potential or real EVENTUAL DAMAGES from the event. In addition, in the case of prolonged or targeted attacks it can be important to identify the source of the attack, and the actions of "disclosing the source of the attack" are time consuming and complex. To solve the problems more effectively and to automate the process, we have tested a few methods of "reverse penetration". It has been an experiment in which our WEB site has conducted the counter-attacks on all those who tried to attack us.

Attacker's types

Before we start the practical work it is necessary to define the target groups of attackers. In general, it is obvious that all attackers can be divided into different classes based on their different motivations. There are the following classes:

- Automated attackers (bots/some scans)
- Motivated attackers
- Script Kiddies
- White Hats

Bots

This group is the least interesting because the target system's information is not the main purpose of the attackers. In most cases, an attacker is only interested in system's resources to spread malicious

content. These attacks can be easily identified with the aid of ordinary IDS and are using well- known patterns; so, it is not a big deal for us.

Script Kiddies

This group can be interested in the target system or its data, but, in general, the group acts for common interest of breaking any system. They can have special motivation because of personal reasons such as national or political reasons. Nevertheless, this group is characterized by a lack of persistence in finding vulnerabilities, and they can easily switch to another system. We can say that they are using templates and known patterns for searching the bugs. But, in the case of a successful attack, the group presents real interest for our work because of their harmful impact through disclosure of confidential or private data. In this case there are no differences between targeted attack and just random hack since their effects may be the same.

Motivated attackers

This group is the most interesting and the biggest threat because they are motivated in breaking a particular system. Their techniques may differ from the well-known patterns (script kiddies level) to the highly professional and unique patterns. In fact, the group is of the greatest interest for us.

White Hats

These guys should not be excluded from the list of those who is looking for vulnerabilities because in IDS's context there are no differences between the evil attacks and the sweet bug hunting. It is clear that the identification of the attacker's intent cannot be understood from IDS (in general). But may some honeypots can help us...

From this simplified classification, we can see that it is very important to determine the motives and purposes of attackers before doing anything. Nevertheless, the technical skills of an attacker in terms of his potential negative impact are also in scope of our interests. Also, it is important for us to be able to recognize different signatures/fingerprints such as the signatures for finding bugs, for exploiting or testing vulnerabilities and attacks.

Skill-O-Meter

The technical level of attacks can be determined similar to CTF's tasks. If Honeypot system has a few vulnerabilities with different levels of "finding" and "exploitation", it will be possible to clearly identify the potential of attackers. Also, the same thing can be done by using a single vulnerability if several different steps are needed for an attacker from "finding bug" to a successful attack with exploit.

Our goal is make an attacker to find the honeypot system before he knows the real functionality. Therefore, we need to consider the possible "entry points" that are clear and logical for an attacker, for the scanners and other software that an attacker can use. As an example, we can have the first "point of

entry" as URI - "/admin/". This URL will be found for the first few seconds of scanning of DIRBuster or other software. This script "/admin/" may not contain bugs or contain minor ones (like enumeration of existing logins that the attacker would assume as a username like "admin"). Every time when the honeypot system has found an attempt of authentication with username "admin" and incorrect password, we know there is a Script Kiddie or scanner (if we have detected bruteforce). If we lay out the next script with another URI, for example "/admin/help.php?faq_id=3", containing vulnerability like SQL Injection, then we can proceed to detect the next steps of the attack:

- Finding a vulnerability
- Finding a way of exploitation
- An attack vector

In our example the script can detect each stage separately, as well as the transitions between them, by using only one vulnerability - SQL Injection. This will help us to understand logic, skills and intents of an attacker. For more accurate results, we can use SQLi bug that allows detecting smooth transition between each step and assessing technical skills of an attacker. For a SQL injection, we can do it using filtration of HTTP requests (filtering out certain characters such as spaces). Also, we can use not simple SQL query and not popular DBMS. For example, we can use SQLite database and filter out "spaces". So, our SQLi will be "error-based". Then, the detection of the error is classical:

/admin/help.php?faq_id=3’

This query returns the error code "500”. It indicates that the attacker has found a bug in SQL query. Then we are able to detect the process of finding of vulnerability:

/admin/help.php?faq_id=3’and’1’=’1
/admin/help.php?faq_id=3’and/**/1=1—

Given that the requests:

/admin/help.php?faq_id=3'and'1'='1

and

/admin/help.php?faq_id=3'and'1' = '0

..will returns the same result. It is means the attacker will need to solve two tasks:

- Understand what is a RDBMS
- Find error-based method

Both tasks can be solved only with a little more advanced experience, so, if an attacker gets to the real-world use, we can know better what his skills are.

‘union/**/select(CASE/**/WHEN/**/sqlite_version()like'3.%'THEN/**/select(1)from(lololo)ELSE’BH EU13’END)

If an attacker decides to stop on the early steps, then, he is likely a White Hat since his motivation is limited to bug hunting and exploitation. If an attacker goes to the next step trying to violate the privacy or integrity, we can certainly identify him as an motivated attacker. After that, we can develop an counter attack.

Step 1. Finding bug.
Step 2. Finding how to exploit it.
Step 3. Getting password with SQLi for /admin/.
Step 4. Using this password for authentication.
Step 5. Counterattack.

Counterattack

"Reverse Penetration" can be done in any form, the main purpose of it is to reveal the source of an attack. In general, we are dealing with the logs of a system; therefore, we can get just IP address and User-Agent of attacker’s client. It is obvious that this information is useless in most cases, since, an attacker is using TOR or/and proxies. And, if we can install a backdoor to an attacker's workstation, we will get the direct access to the workstation and receive more information about the attacker. What kind of information can be collected by our backdoor?

- Files from HDD. Config files, images, documents and etc.
- Information about HotSpots - list of BSSIDs. This information can be very useful if we want to get attacker’s geo-location.
- Information about network configuration.
- Run trace route.
- Run nslookup.
- Make mic/video records…

If we want to install backdoor we can use any vulnerabilities in browser and/or its plug-ins. In other words, we can use a classical model of any Exploit-Pack. This method has a good chance to work out. But there are other techniques that can be used in order to hack an attacker. These techniques are based on the methods of Social Engineering. They use the fact that an attacker does not expect counterattack during that time of his attack. That gives us an advantage and the ability to manipulate them in order to run our code on the side of an attacker.

One of such methods is the system of backdoored honey tokens. For an example: an attacker gets into the system and gains the access to the file system. On the next step, he finds a few EXE files with the necessary information. The files can be self-extracting archives, software version with "Unlimited license" or some client for custom software. In any case, it should motivate him to run these files on his

own workstation. In general, the honey token can be a PDF or Word file with 0day exploit inside. The main thing is good motivation to use this file in attacker's network and run it.

A simple example of honey token may be an unique client software, like ActiveX/Java component that can be part of the private system. If there are correct decorations and good disguise, an attacker will not get any suspicions. In particular, this might be useful for counter-attacks in the case of cyber-warfare. If US wants to attack some Russian/Chinese system (army's SMART GRID), it is obvious that in order to get access to such systems (which are based on unique, domestic software), you need to get 'client' software. This 'client' is not possible to download from the public part of Internet or US sources. Similarly, it is true for Russians online banking and other cyberspaces since the local laws make some restriction on use foreign software.

"Reverse Penetration" is not limited to counterattacks on attacker's workstation. For disclosure a source of an attack, we can apply all known techniques aimed on uncovering "privacy" or local environment. For an example: DNS Rebinding (Anti DNS pinning) can be useful for discovering local network resources of an attacker [1]. In addition, XXXSS attack [2] on home routers can get a list of BSSID. There are hundreds of techniques that can help in our mission and all of them can be used. In addition, strong relaxation of an attacker can help us to attack him on third-party resources on the Internet. If he is authenticated on some WEB resources like social networks, it is very easy to exploit private (0day) XSS/CSRF on this resources to get information about the attacker. For example, we can use CSRF (it is not CSRF actually but it is a feature) on LinkedIn and it will forward the request to view some profile. Then, we just view the history of this profile to find out who was looking this profile. We can get just the 'Employer's name', but it is better than nothing. By the way, this is a very good method for WhiteHat's identification who are into this social network. In fact, they are a little more into Twitter than they are into LinkedIn.

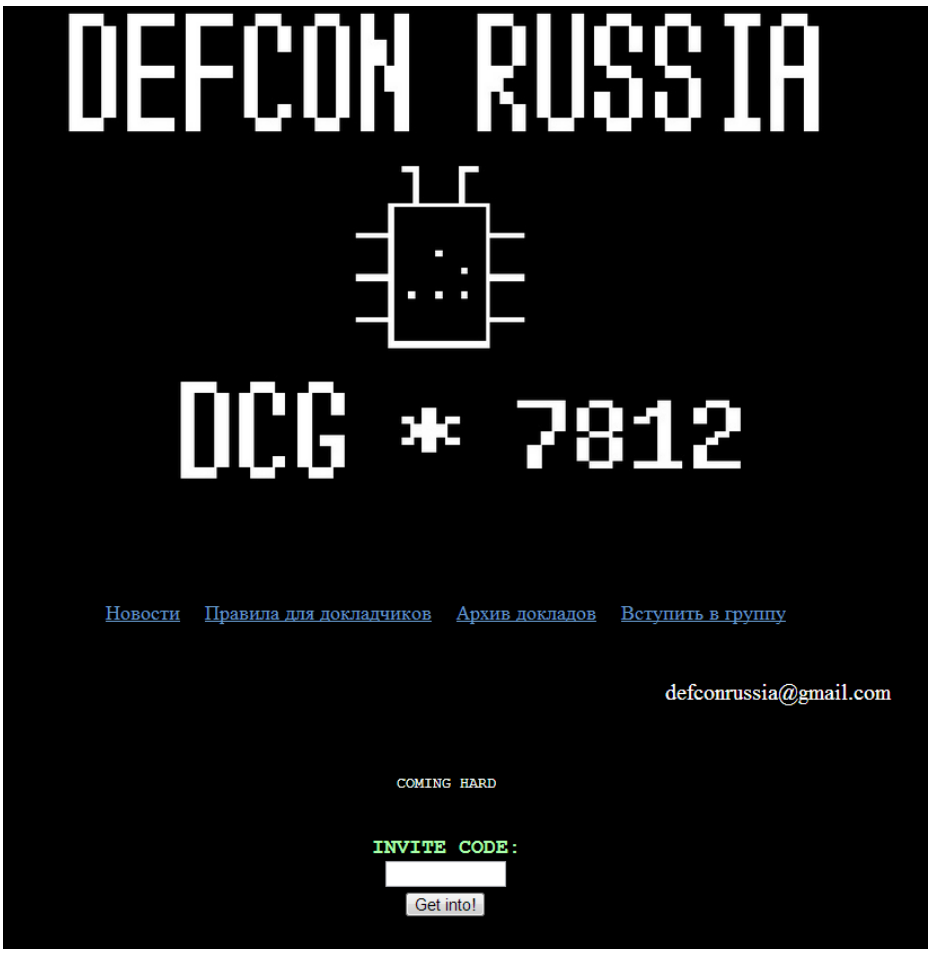
Experiment

In order to test the effectiveness of the methods, we have used a few of the techniques in the real environment. Our web-site of Defcon Group # 7812 [3] was chosen as the experimental site. A system with a bait had been located on the main page since the site was launched a long time ago. The Honeypot represents a simple input form for an "invite code". It can be perceived from the point of view of Social engineering as:

- A certain private code to enter into the community
- Test for "hackers"
- "I don't care. I want to try SQLi in any form"

Obviously, the simplest version of the bypass this "invite code" would be a classic SQL injection:

' or 1=1—



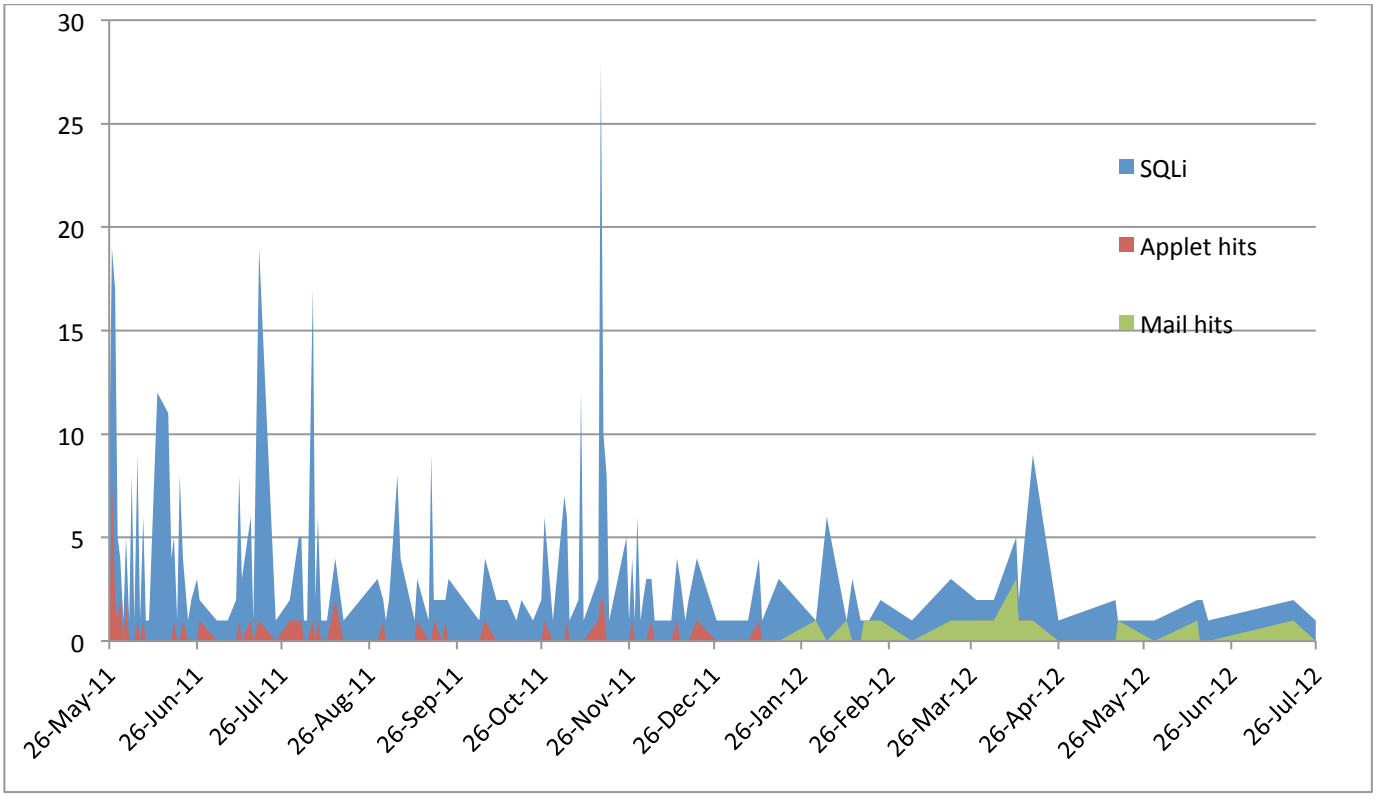
If an attacker had sent SQLi, he got Java applet and message that this applet was a GUI for private zone (Social Engineering). Applet is standard drive-by-download. Dropper is simple an EXE file (backdoor-agent). This Agent will not try to write itself into autorun or etc., its functions include collecting system information and sending it by using revers DNS channel. Remote control was not developed, although this may have been done, but for some ethical and moral reasons, we did not want to have RAD. In addition, this Agent is not trying to get private or personal information. Its functions:

- ipconfig
- tracert 8.8.8.8
- net user

In this experiment, there were done no assessments of attacker's qualification. We were wanted to estimate chance and the benefits of the "Reverse Penetration". An experiment was launched in May 2011. And at the time of launch was only one counterattack - Java applet. In January 2012 we have added two exploits, for two of most popular email services in Russia - mail.ru and yandex.ru. Both exploits were about JSONP Hijacking (CSRF), and as result we got the attacker's mail address (if they have been authenticated through web interfaces and session was active). All three elements of a counterattack (Evil applet and two JSONP CSRF) was activated if SQL injection attack was detected. An experiment was completed in July 2012.

Results

During the experiment were have detected 484 unique SQL attacks to bypass authentication by "Invite code". Also, we had 68 successful "Reverse Penetrations", that is ~ 37% of success in carrying out counter attacks (the data is only about the unique attacks, repeated events were excluded from the statistics). It should be taken into consideration that the information about the applet and its true purposes was published online in April 2011. However, the experiment was successful in spite of that.



Attacks by date.

You can see on the graphs that the relationship between the JSONP exploitation (green) is directly proportional to the number of SQLi attacks. Also, Java applet is not always proportional to the number of attacks, since, the user's action was required (confirmation to run the applet). We have got some interesting results about the attackers. The bulk of the attackers were the main target audience of the website since the site was dedicated to information security problems. Therefore, most of visitors had both skills and motivation for conducting an attack on our website. We can also mention that they have positioned themselves as White Hats during the attacks. Using SQLi was not a moral issue for them, especially, if the attacked website was about IT Security.

```
DNS IP          : 80.[redacted].5
      User:           ro[redacted]in
      DNS:            olympus.[redacted].com
      Local IP:       192.168.1[redacted].55/192.168.[redacted].1(VMware)/192.158.[redacted].1
      Tracert:        *FILTERED

DNS IP          : 195.[redacted].130
      PC:             \\IT-[redacted]
      User:           go[redacted]ov
      DNS:            ru.[redacted].lan
      Local IP:       172.2[redacted].24/192.168.[redacted].1
      Tracert:        -> 172.2[redacted].200 -> cl[redacted].metrocom.ru [213.182[redacted]9]
```

Data attacker's hosts

Because of these reasons, most attackers did not try to hide their IP. There were also registered attempts to SQL injection from external IP of leading Russian information security, the IP addresses of the Ministry of Defense of Russia and the Intelligence Agency of one of the CIS countries. Remarkably, we had successfully counterattacked the workstation owned by Intelligence Agency of one the CIS countries. It was found out later that the workstation has been likely compromised and used as a cover. But, thanks to our methods (and luck) , we found out the probable nickname of the hacker who had got unauthorized access to that workstation (because, a few hours later we had got another successful counterattack to another workstation from the same country, where the account name was the same as know nickname).

```
PC:          \\RE[redacted]
User:        Re[redacted]ed
Local IP:    10.[redacted].1.[redacted]
DOMAIN:      [redacted].gov.[redacted]
```

Data from government attacker

In addition, our applet was used on virtual host of one of antivirus companies, as well as in the Domain network of one of the largest Russian IT Security companies (employee did some research with payload and accidentally ran it with Domain's account).

Conclusions

Obviously, this technique has a number of moral, ethical and legal issues. However, it can be applied with a sufficiently high efficiency for detection and identification of attackers, at least against those who are not careful enough to expect a counterattack. I think that these methods can be used for protection at the state level and, probably, already exist these days. However, using these techniques for the Commercial/Enterprise sector can be difficult because of legal issues of the methods of defense.

[1] http://en.wikipedia.org/wiki/DNS_rebinding
[2] http://www.slideshare.net/M8Rex8b3j/samy-kamkar-geolocation-via-xxxss-2010-cutted-part-of-it
[3] http://defcon-russia.ru

Special thanks to **Alexey Tyurin**, who was helping me with this project!
Thx to all DCG #7812 team, “victims”, hackers and everyone!