APPLIED CRYPTOGRAPHY AND PRACTICAL SCENARIOS FOR CYBER SECURITY DEFENSE

Emil SIMION[1], Alecsandru PĂTRAŞCU[2]

The Internet is a rather new technology for us, but nevertheless it has become a force which is working its way into all our lives and there is no doubt in this. However it is important to realize the security issues that involve the Internet. Cyber security has emerged within the latest years as an issue of vital national and international importance, since governments, businesses and individuals are under constant attack from other governments, cyber criminals and hackers. These attacks consists in stealing large amount of money or personal and business data and compromise sensitive government operations. This paper explains these threats with accent on new forms of it. For in-depth understanding of the concept involved in cyber security, some attack scenarios explained in detail have been included.

Keywords: cryptography, cyber security, Cyber MITRE, malware, Flame, wireless hacking, WordPress hacking

1. Introduction

In our days, Internet Technologies optimize the time needed for taking human and organizational decisions [1]. For example, we are using Internet for electronic communications, commerce, transactions and banking, for accessing different data bases in order to process the information. All these actions, which interact with virtual activities, must be protected from electronic fraud. Thus, we need to implement, in the virtual space, security measures similar to real security measure. But there is a difference: behind real life thefts there are humans which interact with the goods and in the virtual space the thefts are made by viruses, worms, malware applications which interact and monitor the actives of the system. If we think that all these weapons (viruses, worms, malware applications etc.) are produced by humans we can conclude that in the protection of the system actives the human intelligence plays a definitive role and the critical decisions must be taken by humans [2]. The problem is how we can order these decisions, which are in large number, in such a way that to be made, corrected and modified by a limited human intervention [3].

[1] PhD., Advanced Technologies Institute, e-mail: ati@dcti.ro, esimion@fmi.unibuc.ro
[2] Eng., Computer Science Department, University POLITEHNICA of Bucharest, e-mail: alecsandru.patrascu@gmail.com

This paper presents some practical scenarios which can be used for training decision makers in cyber security. These scenarios are based on the Cyber MITRE challenges proposed at the International Conference on Cyber Security organized by Federal Bureau of Investigation and Fordham University. In this document we will present four basic scenarios: identification of encrypted data, identification of steganography and showing the hidden data, identification of a suspect communication between two computers and revealing the stolen data, identification of incorrect cryptographic algorithms usage.

On the other hand, outside the academic area challenges, we find attacking scenarios that affect real-life software that is used by people all around the world [4]. Furthermore, a lot of these attacks are issued by skilled hackers before security companies become aware of them – the so called "zero-days" attacks. At this part we will present the scenario of cracking a Wi-Fi network WPA password on modern routers that use the Wi-Fi Protected Setup (WPS) protocol and hacking a website running the WordPress blog platform.

Botnets and malware are also covered in our document and we will talk about these new threats that have emerged in the Internet and are becoming the fuel of new attack types.

The paper is organized as follows. In section 2 we briefly talk about the tools used in a specific investigation. Section 3 is dedicated entirely for the newer botnets and malware threats. During section 4 we present the complete scenario with details on the input, tasks and output and in section 5 we describe operations done in solving these scenarios and finally section 6 contains conclusions for our paper and an outline of the directions for future research.

2. Tools used in investigation

2.1. CrypTool

Stand-alone tools represent tools that can be used independently over an operating system. Representative to this field is CrypTool. This is a software package dedicated to cryptographic simulation, analysis and cracking which has a user graphic interface.

CrypTool has been developed in cooperation with prestigious universities and thus has become excellent educational software and also a tool for learning cryptology. CrypTool covers both branches of cryptology: cryptography and cryptanalysis. Thus, the product has implement facilities of each field, such as classic cryptography (Caesar and Vigenère ciphers, monoalphabetic substitution, etc.), symmetric cryptography (IDEA, RC2, AES, etc.), asymmetric cryptography (RSA and elliptic curves, etc.), hash functions (MD2, MD5, SHA-1, etc.), cypher text attacks, plaintext attacks, adaptive attacks, side channel attacks.

2.2. Statistical tools

In order to test the degree of randomness for input or output for such transportation systems, we need a set of different tools, tests and theoretical models – we need statistical tools. Using this kind of tools we can test for true randomness of functions that are part of the software implementation of these systems. Good examples in this direction are:

- "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications" [5], is a publication of sixteen statistical tests. The authors provide also an implementation;
- Donald Knuth's book [6], describes several empirical tests which include the: frequency, serial, gap, poker, coupon collector's, permutation, run, maximum-of-t, collision, birthday spacing's, and serial correlation;
- The Crypt-XS suite of statistical tests was developed by researchers at the Information Security Research Centre at Queensland University of Technology in Australia. Crypt-XS tests include the frequency, binary derivative, change point, runs, sequence complexity and linear complexity;
- The DIEHARD suite [7] developed by George Marsaglia adds to the tools mentioned before, tests such as random spheres, overlapping sums, etc.

2.3. Security oriented distributions

Penetration testing and security auditing are now part of every system administrator's "other duties as assigned". In this direction, BackTrack Linux (BTL) [8] comes to help. BTL is a custom distribution build on top of Debian Linux, designed for security testing for all skill levels from novice to expert. It is the largest collection of wireless hacking, server exploiting, web application assessing and social-engineering tools available in a single Linux distribution. It is a fine example of a specialized Linux distribution because its only purpose is to test a network, a device and systems for security vulnerabilities.

3. Zero-day vulnerabilities, botnets and malware

3.1. Zero day attacks

A "zero-day" attack or threat is an attack that aims to exploit previously unknown vulnerabilities in software applications. The term comes for the fact that the attack occurs on "day zero" of awareness of the vulnerability and the developers of the affected application had no days available for patching it.

These attacks come in a large number. Malware, viruses and Trojans all represent attacks vectors that target modern software and delivery networks. In

this equation the web browsers and the operating systems on top which they are running represent the most widely targets because they are widespread on all devices, starting with mobile phones and ending with desktops. Mail delivery networks are also targeted because they can carry to a potential victim an infected e-mail attachment. To cope with these threats organizations like **US-CERT** [9] and **Zero Day Initiative** [10] dedicate their work in providing users cyber security.

3.2. Botnets

A botnet is a number of Internet computers that, although their owners are unaware of it, have been mangled and have been set up to send and forward different types of transmissions, including spam or viruses, to the other computers on the Internet. The term comes from the fact that any infected computer becomes a "robot", or "bot" for short, that serves an attacker. A bot is a program attached to one of the computer ports that is left open and through this port a remote program can connect to it.

One example is the DNSChanger bot. This is a Domain Name System (DNS) hijacking Trojan and it was distributed over the Internet as a download claiming to be a video codec needed to view video content on bait pornography sites. Once installed it modifies the target DNS configuration to point to bogus servers over the Internet operated by an Estonian company called Rove Digital and its hosting subsidiary Esthost. By now it is estimated that it infected over 4 million computers worldwide, many of them being at government agencies and large companies. The scheme this botnet implements makes use of its linked Trojans to divert Web traffic from its intended destination to that of advertisers who paid for traffic delivery. This traffic was made to IP addresses falling into the following ranges: 85.255.112.0 – 85.255.127.255, 67.210.0.0 – 67.210.15.255, 93.188.160.0 – 93.188.167.255, 77.67.83.0 – 77.67.83.255, 213.109.64.0 – 213.109.79.255 and 64.28.176.0 – 64.28.191.255.

3.3. Malware

Malware represents the software used or created by hackers to alter computer operations. The goal is to gather sensitive information or to gain access to private computer systems. Its form varies from a full software program to a script. It is a general term that is used to refer to all forms of hostile and intrusive software, like computer viruses, Internet worms, Trojan horses, spyware, adware, and rootkits.

Being such a vast security domain, examples for malware are in great number, but we will present some of the newest form of it, Flame. Even

smartphones are not avoided and we will also talk about the Trojan!MMarketPay.A@Android malware.

3.3.1. Flame

Flame, also known as Flamer or sKyWiper is a computer malware discovered in 2012 that attacks computer running Microsoft Windows operating system. It was used for cyber data espionage in countries such as Iran, Israel, Sudan, Syria, Lebanon, Saudi Arabia and Egypt. Its discovery was made on 28th May 2012 by Iranian MAHER CERT, Kaspersky Labs and CrySyS Lab.

This malware is capable of spreading to other systems over a local area network (LAN) or through USB sticks. It can record audio, screenshots, keyboard activity and network traffic, Skype conversations and can turn mobile infected computers, like laptops and netbooks, into Bluetooth beacons which attempt to connect and download information from other Bluetooth devices nearby.

What is interesting on this malware is its built-in capability to "suicide". The creators of Flame implemented a "kill" command that automatically removed itself from the infected computers. More exactly, when it received that command, Flame located every file existing on the victim PC, deleted it and then overwrote its memory location with random data to prevent a forensic examination.

According to cryptographic experts, Flame was the first malware to use a rather obscure cryptographic technique called "prefix collision attack". This allowed it to fake digital credentials that had helped it to spread. The exact method for this kind of attack was demonstrated in 2008, but the creators of Flame implemented their own variant. All these findings give support to claims that Flame must have been built by a nation state rather than cybercriminals due to the large amount of time, effort and resources that have been put into its creation.

3.3.2. Trojan!MMarketPay.A@Android

This new piece of malware has been discovered on the 9th of July 2012 on more than 100.000 Android smartphones. It generated revenue by silently downloading paid applications and multimedia content from Mobile Market, an Android application store hosted by China Mobile.

It was discovered by the TrustGo security company and it was found on nine other online application stores. The security firm presented to the public eight package names for the malware: *com.mediawoz.goweather, com.mediawoz.gotq, com.mediawoz.gotq1, cn.itkt.travelsky.*

The malware works by placing malicious orders at Mobile Market. Normally, a Mobile Market customer receives a verification code via SMS after purchasing an application or multimedia content, which the customer has to input back into the market to start the download. China Mobile then adds this order to the customer's phone bill. *MMarketPay. A* automates this process and downloads

as much content as it can so that victims end up with huge phone bills. It finds paid content, simulates a click action in the background, intercepts the received SMS messages, and collects the verification code sent by Mobile Market.

4. Scenario descriptions

The following problems were given at MITRE CYBER challenge at International conference of cyber security. There are four scenarios, the first three of them being related one to each other. The investigator can use for solving these problems any software available. In this case we will use CrypTool and Pari software and a little bit of mathematics.

4.1. Identification of encryption system

In this scenario the investigator finds a suspect file on the computer. This file represents an encrypted data with a classical encryption system. The task is to recognize the classical encryption system, decrypt the data and find the password hidden in the encrypted file. The investigator will need this password in solving the second task.

4.2. Identification of steganography

On the investigated computer have been found some images. One of these images has a huge size reported to its format. The task is to find the encrypted data founded in the image, decrypt it and find the file name which will be under the investigation in the third scenario.

4.3. Identifying a suspect communication between two computers and reveal the stolen data

The local administrator gives to the investigator a traffic capture between the computer and an outside unknown source. The task is to find the password which has been stolen.

4.4. Recovering the signature key

You are given two ECDSA signatures. Something about them looks strange. Using that and the known public key and parameters, find a way to recover the private key used to generate the signatures. The next topics will present scenarios of real-life scenarios.

4.5. Cracking a Wi-Fi network WPA password

In this scenario we are trying to gain access to a private Wi-Fi network that is secured using the Wi-Fi Protected Access (WPA) protocol. WPA is a security protocol and security certification program developed by the Wi-Fi Alliance. It is known as the IEEE 802.11i standard. More exactly a Temporal Key

Integrity Protocol (TKIP) is implemented, that involves using a dynamic 128 bit key for every packet transmitted. The newest version, WPA2 also includes Counter Cipher Mode with Block Chaining Message Authentication Code Protocol (CCMP), a new AES based encryption mode with strong security.

4.6. Attacking WordPress-based websites

In this scenario we are trying to gain access to a website that uses the WordPress web framework, exploiting a software bug existing in many themes. WordPress is a wide used free and open-source blogging tool that has also functions derived from Content Management Systems (CMS). It is written in the PHP scripting language and it uses MySQL as a backend database. It is estimated that around 34.2% of existing websites in 2012 are using one of its versions.

5. Investigator operations

5.1 Using CrypTool

If we take a look at the encrypted file we see that this file contains only 26 characters A to Z. If we perform a statistics of these letters we see that the characters A-Z appears to be random. Thus, we can think that there is a classic encryption such as substitution (Playfair, Caeser, Vigenère etc.). The first scenario can be easy solved using CrypTool. Using cipher text only attack on a Vigenère cipher we find the encryption password <<SQUARE>>. If we take a look at the end of the decrypted file we find ….**STRONGPASSWORDSAREGOOD**, which is the password we are looking for.

5.2 Using UltraEdit and some tricks

For the second task first we take a look at the image. This image is in gif format, but has a huge size aprox 13 MB. If we investigate this image with UltraEdit we can see at the end of the file zip file header (PK). Thus, if we exchange the extension of gif file to zip file and open it with WinZip we find an encrypted archive. The opening password is STRONGPASSWORDSAREGOOD. In this archive we find three gif files. Opening each file we can see that one of these one has the content: hollenger.dll. Thus, the file name with we are searching for is *hollenger.dll*.

5.3 Wireshark

Using Wireshark application we are opening the target file and searching for string hollenger.dll in packet bytes. Using *Follow TCP stream* option we locate the target file and dump it into a file. Opening the dumped file with UltraEdit we see the zip file header (PK), thus we change the extension of the dumped file into zip file and open it. This archive contains a file named

hollenger.dll. Visual inspection with UltraEdit reveals the magic header GIF. Thus, if we exchange the extension of the file into gif and open it with an image viewer we can see an image which has the writing *The Root Password is* ***Pengu1nsR0ck***, which represents the information that we are looking for.

5.4 Incorrect usage of randomness in signing process

5.4.1 ECDSA

Before presenting the results of investigation we need a little bit of mathematics: ECDSA description. The public parameters are the prime number p, an elliptic curve $E[F_p]$, a point $G \in E[F_p]$ with ord $(G)=q$, q prime number. The public key $V \in E[F_p]$ is derived from the signing key $1 \le d \le q-1$: $V=dG$. The **signature** of the hash h is computed using the ephemeral key k mod q is the pair $(r,s)=(x_{kG} \bmod q, (h+dr)k^{-1} \bmod q)$, where x_{eG} is the first component of the point $eG \in E[F_p]$. To verify the signature (r,s) of the hash h we need to check if $x_{v1G+v2V} \bmod q=r$, where $v_1=hs^{-1} \bmod q$ and $v_2=rs^{-1} \bmod q$. It is essential to have for different signatures (r_1,s_1) and (r_2,s_2) different ephemeral keys $k_1 \neq k_2$. If this two keys are equal then the signatures of the two hashes looks like (r,s_1) and (r,s_2). Thus, we can derive $s_1-s_2=k^{-1}(h_1-h_2) \bmod q$ and find the ephemeral key $k=(h_1-h_2)(s_1-s_2)^{-1} \bmod q$. Since $s_1=k^{-1}(h+dr) \bmod q$ we derive the private key $d=(s_1k-h_1)r^{-1} \bmod q$.

5.4.2 Preliminary investigation

The investigator receives three files. The **first file** contains the hash, in hexa codification, of two messages h_1, h_2 and their ECDSA signatures (r_1,s_1) respectively (r_2,s_2) :

h_1=DE37B3145DB7359A0ACC13F0A4AFBD67EB496903
r_1=ACB2C1F5898E7578A8A861BDF1CA39E7EF41EAC0B6AAA49468DD70E2
s_1=BE4FA99C9D261C5F387A3ACE025702F6FB7884DD07CE18CAD48654B8
h_2=28469B02BF0D2CFC86FF43CB612EE8FC05A5DBAA
r_2=ACB2C1F5898E7578A8A861BDF1CA39E7EF41EAC0B6AAA49468DD70E2
s_2=D3540E2B13E51605F5FEB8C87EE8E176E59213F31EA8B8FFDAD077E2

The **second file** parameters.der contains, in der codification, the public parameters of the EC. This file can be interpreted using OpenSSL:

openssl ecparam -inform DER -in /cygdrive/e/parameters.der -outform PEM -out /cygdrive/e/parameters.pem
openssl ecparam -text -in /cygdrive/e/parameters.pem -noout
Field Type: prime-field
Prime:
00:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:fe:ff:ff:e5:6d
A: 0
B: 5 (0x5)
Generator (uncompressed):
04:a1:45:5b:33:4d:f0:99:df:30:fc:28:a1:69:a4:67:e9:e4:70:75:a9:0f:7e:65:0e:b6:b7:a4:5c:7e:

08:9f:ed:7f:ba:34:42:82:ca:fb:d6:f7:e3:19:f7:c0:b0:bd:59:e2:ca:4b:db:55:6d:61:a5
Order:
01:00:00:00:00:00:00:00:00:00:00:00:00:00:00:01:
dc:e8:d2:ec:61:84:ca:f0:a9:71:76:9f:b1:f7
Cofactor: 1 (0x1)

The **third file**, public.oct, contains the public key:

X_V =85CEEE9C98EFDFDFCF64CB522A773F1435D568173677D1D28FC00643
Y_V =58A105CC1AB1A53D77B278850776E144197F3FA4E27AA676408DFE22

5.4.3 Finalizing the investigation

At this point, because the two signatures collide on the first half, we have all the elements to finalize the investigation. The only think we need to do is to compute the private key using the formula $d=(s_1\ k-h_1)r^{-1}$ mod q, where $k=(h_1-h_2)(s_1-s_2)^{-1}$ mod q. We can perform these computations using, for example, MAPLE:

```
> h1:=convert("DE37B3145DB7359A0ACC13F0A4AFBD67EB496903",decimal,hex);
> h2:=convert("28469B02BF0D2CFC86FF43CB612EE8FC05A5DBAA",decimal,hex);
r:=convert("ACB2C1F5898E7578A8A861BDF1CAE[...]C0B6AAA49468DD70E2",decimal,hex);
s1:=convert("BE4FA99C9D261C5F387A3ACE0257[...]DD07CE18CAD48654B8",decimal,hex);
>s2:=convert("D3540E2B13E51605F5FEB8C87EE[...]F31EA8B8FFDAD077E2",decimal,hex);
>q:=convert("0100000000000000000000000000001[...]6184CAF0A971769FB1F7",decimal,hex);
> d:=(-h2*s1+h1*s2)*(r*(s1-s2))^(-1) mod q;
> convert(d,hex,decimal);
```

After the compilation of the program we find the private key:

d=8E88B0433C87D1269173487795C81553AD819A1123AE54854B3C0DA7

5.5 Using Reaver

Reaver implements a brute force attack against Wi-Fi Protected Setup registrar PINs in order to recover WPA/WPA2 passphrases, as described in *http://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf*. On average Reaver will recover the target AP's plain text WPA/WPA2 passphrase in 4-10 hours.

As mentioned before, the Reaver tool takes advantage of a vulnerability existing in WPS. WPS is a feature that exists on many modern routers and its intention is to provide to the users an easy setup process. The problem is that it is tied to a PIN that is hard-coded into the device.

The setup is very simple and it contains the following elements: the BackTrack Linux live DVD, a computer with Wi-Fi and a nearby WPA-secured Wi-Fi network. After booting the Linux distribution we must get our wireless card's interface name and the BSSID of the router we're attempting to crack. The BSSID is a unique series of letters and numbers that identifies a router.

First, we must find out what is our wireless card. Inside a terminal we issue the command *iwconfig* and search the wireless device in the shown list. Most likely it will be named *wlan0*. After that we must put the wireless card into monitor mode by giving the command *airmon-ng start wlan0*. This command will

output the name of the monitor mode interface, *mon0*. Lastly, we have to find the BSSID of the router we want to crack, by typing the command *airodump-ng wlan0* or *airodump-ng mon0*. In the list shown inside the terminal we copy the one for our network. We will assume the following BSSID: *8D:AE:9D:65:1F:B2*.

Now, with the BSSID and the monitor interface name in hand we have everything we need to startup Reaver. Inside a terminal we issue the command *reaver –i mon0 –b 8D:AE:9D:65:1F:B2 –vv* After this, Reaver will try a series of PINs on the router in a brute force attack, one after another. After cracking is complete we will have the output of the total time needed to crack the password, the router PIN and its password.

Unfortunately, even with WPS manually shut down, Reaver is still capable to crack some devices. In our tests, only DD-WRT and Tomato open-source router firmware were immune to the attack since they don't implement the WPS protocol.

5.6 Hacking WordPress

The attack we are going to present next is based on a vulnerability existing in the *timthumb.php* library existing in many themes used by this framework [11] [12]. It has the function of a photo-resizing utility of the pictures uploaded inside a website and in its form allows hackers to write whatever content they point to. For example, the utility can be configured to check the users only pass content from YouTube, but the check that it does inside its source code will only make sure that the word "YouTube" exists within the URL path. Knowing this, we can make our own domain, for example *youtube.com.evil.com*, and it would pass that check and finally could upload a custom PHP file.

To make things even interesting, *timthumb.php* uses a caching mechanism so that it doesn't have to continually re-process images. The cache directory is located under the WordPress root and it is accessible by the visitors of the website. To be clear, the library doesn't actually execute any remote malicious code that causes this vulnerability. It simply gets a remote file and places it in a web accessible directory. As we said before, the problem is in the way this library checks the remote content path. Below in red is the vulnerable code:

```
$allowedSites = array ('flickr.com', 'picassa.com', 'blogger.com', 'img.youtube.com' );
$isAllowedSite = true;
foreach ($allowedSites as $site) {
        if (stristr($url_info['host'], $site) !== false) {$isAllowedSite = false; }
}
```

An attacker can upload a script through a forged web domain to the cache directory. The library will store it as *external_<md5 hash>.php*. Then he will access this script directly from the *timthumb.php* cache directory, for example:
http://myblog.com/wp-content/themes/Memoir/scripts/cache/external_<md5hash>.php

Sample script for exploits that can be used for this vulnerability can be found freely on the Internet and we will not insist on them furthermore. One example is the Alucar shell script. It was uploaded as an encoded base64 file to make finding it a little harder using tool like *grep*. The script gives a web based shell access and It can be used to inject, for example, other base64 code to another WordPress file. In our scenario we injected code in the *wp-blog-header.php*, that is located in the website root. The file looks like this, and the red part is the injected code:

```
<?php
    if ( !isset($wp_did_header) ) {
      […]
      eval(base64_decode("ZnV[…]3BmJyk7"));
      […]
    }
?>
```

The code executes every time a blog page is visited and it downloads a file from a certain URL and writes it to a directory located on the webserver disk. The solution to this kind of attack is to use the newer *timthumb.php* library that has this vulnerability patched.

6. Conclusions

As we can see form the above presented scenarios the cyber security field involves strong knowledge of computer communications techniques, secure programming techniques, algorithm and software implementation analysis, cryptography, steganography, probability and finally applied mathematics. The process of high level of assurance of cyber security must take into account all the above specified domains.

Fast advances in cybercrime technology and techniques have resulted since the beginning of 2012 in an unprecedented rise in data breaches. We think that planning to ensure that our online world is trustworthy and secure we need to consider the fundamental changes that are occurring in the cyberspace and try to adapt to them. In our opinion, looking forward into the future of the 3 billion Internet users existing today we can see four big directions for resolving the cyber security issues: online users security education, mobile devices cryptography, online data obfuscation and cloud services transparency and security.

R E F E R E N C E S

[1] *F. D. Kramer, S. H. Starr, L. Wentz*, Cyberpower and National Security, Potomac Books Inc, 2009

[2] *R. A. Clarke, R. Knake*, Cyber War, Ecco, 2010

[3] *P. Hough*, Understanding Global Security, Routledge, 2009
[4] *P. Engebretson*, The Basics of Hacking and Penetration Testing, Syngress, 2012
[5] *NIST Special Publication*, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, 800-22, 2001
[6] *D. Knuth*, The Art of Computer Programming, Seminumerical Algorithms (3rd ed., Vol. 2). Addison Wesley, Reading, Massachusetts, 1998
[7] *G. Marsaglia*, DIEHARD Statistical Tests. http://stst.fsu.edu/geo/diehard.html.
[8] http://www.backtrack-linux.org/
[9] http://www.us-cert.gov/
[10] http://www.zerodayinitiative.com/
[11] http://markmaunder.com/2011/08/02/technical-details-and-scripts-of-the-word-press timthumb-php-hack/
[12] http://www.sparkyhub.com/timthumb-vulnerability-your-blog-is-under-attack-of-hackers/